



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ**  
**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**Πτυχιακή Εργασία**

|                           |                                                                                                                                         |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Τίτλος Πτυχιακής Εργασίας | Εξατομικευμένη διαδικτυακή εφαρμογή ευρέσεως εργασίας για προγραμματιστές<br><br>Personalized job search web application for developers |
| Ονοματεπώνυμο Φοιτητή     | Μάριος Κυρόγλου                                                                                                                         |
| Πατρώνυμο                 | Χρήστος                                                                                                                                 |
| Αριθμός Μητρώου           | Π21080                                                                                                                                  |
| Επιβλέπων                 | Μαρία Βίρβου, Καθηγήτρια                                                                                                                |

Ημερομηνία Παράδοσης **Φεβρουάριος 2026**

**Copyright ©**

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πειραιώς. Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.

## **Ευχαριστίες**

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στην επιβλέπουσα καθηγήτριά μου, κα Μαρία Βίρβου, για την πολύτιμη καθοδήγηση, τη διαρκή υποστήριξη και τις χρήσιμες συμβουλές της καθ' όλη τη διάρκεια της εκπόνησης της παρούσας πτυχιακής εργασίας.

Επίσης, ευχαριστώ από καρδιάς την οικογένειά μου, για την αμέριστη συμπαράσταση και πίστη στις δυνατότητές μου, καθώς και τους φίλους μου, για την ηθική υποστήριξη και την ενθάρρυνση που μου προσέφεραν σε κάθε στάδιο των σπουδών μου.

## Περίληψη

Η παρούσα πτυχιική εργασία πραγματεύεται τη μελέτη, τον σχεδιασμό και την υλοποίηση μιας ολοκληρωμένης διαδικτυακής πλατφόρμας εύρεσης εργασίας για προγραμματιστές (DevJobs). Πρόκειται για ένα σύγχρονο και αναγκαίο εργαλείο διασύνδεσης για την αγορά της πληροφορικής, τόσο για τον ίδιο τον χρήστη που αναζητά εργασία, όσο και για τους ποικίλους εργοδότες, οι οποίοι καλούνται να δημιουργήσουν αγγελίες και να προσλάβουν άτομα, σύμφωνα με τα κριτήρια που οι ίδιοι θέτουν κάθε φορά. Συνεπώς, για την εύρυθμη λειτουργία και ανάπτυξη του ενιαίου ψηφιακού περιβάλλοντος συνδέσεων, κρίνεται αναγκαία η δημιουργία και ύπαρξη προσωπικών ή εταιρικών προφίλ, η δημιουργία και διαχείριση αγγελιών, καθώς και η υποβολή των δεξιοτήτων που κατέχει ο υποψήφιος εργαζόμενος ή επιζητά ο εκάστοτε οργανισμός. Καθορίστηκαν, λοιπόν, δύο βασικοί τύποι χρηστών και διαμορφώθηκαν εξίσου διακριτές ροές χρήσης για κάθε κατηγορία. Η ανάγκη ανάπτυξης του συγκεκριμένου συστήματος έχει τις ρίζες του στην ευρύτερη ψηφιακή μετάβαση της αγοράς εργασίας, η οποία εξελίσσεται και ψηφιοποιείται με ραγδαίους ρυθμούς καθημερινά. Οι παραδοσιακές μέθοδοι εύρεσης προσωπικού φθίνουν, σε αντίθεση με εκείνες που βασίζονται στην συλλογή και ανάλυση δεδομένων του χρήστη, έναντι μηδενικού χρόνου. Στο πλαίσιο αυτό, μέσω της συμβολής του framework της “Spring Boot”, της αντικειμενοστρεφής γλώσσας προγραμματισμού “Java” και διαφόρων άλλων βιβλιοθηκών και εργαλείων που θα συζητηθούν εκτενέστερα στο εγχειρίδιο χρήσης, επιτεύχθηκε η ανάπτυξη ενός αξιόπιστου, επεκτάσιμου και ασφαλούς πληροφοριακού συστήματος, ικανού να λύσει προβλήματα και να συμβάλει ενεργά στην αγορά εργασίας.

## Λέξεις Κλειδιά:

Διαδικτυακή εφαρμογή, εύρεση εργασίας, διασύνδεση

## Abstract

The present thesis deals with the study, design and implementation of an integrated online job search platform for developers (DevJobs). It is a modern and necessary interconnection tool for the tech industry, both for the user himself who is looking for a job, and for the various employers, who are called upon to create advertisements and hire people, according to the criteria that they themselves set each time. Therefore, for the smooth operation and development of the single digital connection environment, it is necessary to create and exist personal or corporate profiles, to create and manage advertisements, as well as to submit the skills that the candidate employee possesses or that the organization is seeking. Therefore, two basic types of users were defined and equally distinct usage flows were formed for each category. The need to develop this specific system has its roots in the broader digital transition of the labor market, which is evolving and digitizing at a rapid pace every day. Traditional methods of finding personnel are declining, in contrast to those based on the collection and analysis of user data, against zero time. In this context, through the contribution of the “Spring Boot” framework, the “Java” object-oriented programming language and various other libraries and tools that will be discussed in more detail in the user manual, the development of a reliable, scalable and secure information system was achieved, capable of solving problems and actively contributing to the labor market.

## Key Words:

Web application, job search, interconnection

## Περιεχόμενα

|                                                                                  |    |
|----------------------------------------------------------------------------------|----|
| Copyright © .....                                                                | 1  |
| Ευχαριστίες .....                                                                | 3  |
| Περίληψη .....                                                                   | 4  |
| Abstract .....                                                                   | 4  |
| ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ .....                                                            | 8  |
| 1 Εισαγωγή .....                                                                 | 10 |
| 1.1 Στόχοι - περιγραφή του προβλήματος .....                                     | 10 |
| 1.2 Αναφορά σε παρόμοιες εφαρμογές .....                                         | 10 |
| 2 Ανασκόπηση πεδίου .....                                                        | 12 |
| 2.1 Η συστηματική εξέλιξη ως προς την εύρεση εργασίας .....                      | 12 |
| 2.2 Αλγόριθμοι – Τεχνητή Νοημοσύνη και Μηχανική Μάθηση στην Αγορά Εργασίας ..... | 13 |
| 2.3 Οφέλη της Αλγοριθμικής Ανάλυσης και Αυτοματοποιημένης Αντιστοίχισης .....    | 14 |
| 2.4 Συνέπειες του αυτοματισμού στην αγορά εργασίας .....                         | 16 |
| 2.5 Προσαρμοστικά Συστήματα .....                                                | 18 |
| 2.6 Προσαρμοστικά Συστήματα στις πλατφόρμες εύρεσης εργασίας .....               | 18 |
| 2.7 Ευφυή συστήματα αντιστοίχισης σε ψηφιακές πλατφόρμες απασχόλησης .....       | 19 |
| 2.8 Ανάλυση της Αγοράς Εργασίας για Προγραμματιστές .....                        | 20 |
| 2.9 Ανάλυση Ανταγωνιστικών Ψηφιακών Πλατφορμών Εύρεσης Εργασίας .....            | 22 |
| 2.9.1 Περιορισμοί και προκλήσεις υφιστάμενων πλατφορμών .....                    | 22 |
| 2.9.2 Λειτουργικοί μηχανισμοί επιλεγμένων πλατφορμών εύρεσης εργασίας .....      | 23 |
| 3 Ανάλυση και Σχεδιασμός .....                                                   | 24 |
| 3.1 Ρόλοι χρηστών και λειτουργίες .....                                          | 24 |
| 3.1.1 Λειτουργίες Υποψηφίου .....                                                | 24 |
| 3.1.2 Λειτουργίες Εργοδότη .....                                                 | 25 |
| 3.1.3 Use Case Diagram .....                                                     | 26 |
| 3.2 Συνολική αρχιτεκτονική Συστήματος .....                                      | 27 |
| 3.3 Λειτουργικές απαιτήσεις .....                                                | 27 |
| 3.3.1 Απαιτήσεις για Υποψήφιο .....                                              | 27 |
| 3.3.2 Απαιτήσεις για Εργοδότη .....                                              | 28 |
| 3.3.3 Απαιτήσεις Διαχείρισης Συστήματος .....                                    | 28 |
| 3.4 Μη λειτουργικές απαιτήσεις .....                                             | 28 |
| 3.4.1 Απόδοση (Performance) .....                                                | 28 |
| 3.4.2 Ασφάλεια (Security) .....                                                  | 29 |
| 3.4.3 Ευχρηστία (Usability) .....                                                | 29 |
| 3.4.4 Αξιοπιστία (Reliability) .....                                             | 30 |
| 3.4.5 Επεκτασιμότητα (Scalability) .....                                         | 30 |
| 3.4.6 Συντηρησιμότητα και Τεκμηρίωση (Maintainability) .....                     | 31 |

|                                                                      |    |
|----------------------------------------------------------------------|----|
| 3.4.7 Συμβατότητα (Compatibility).....                               | 31 |
| 3.5 Τεχνικές Προϋποθέσεις και εξαρτήσεις συστήματος .....            | 32 |
| 3.6 Εκτέλεση της εφαρμογής σε τοπικό περιβάλλον .....                | 32 |
| 4 Υλοποίηση και λειτουργικότητα εφαρμογής.....                       | 33 |
| 4.1 Εισαγωγή αρχιτεκτονικής και τεχνολογιών .....                    | 33 |
| 4.2 Περιγραφή Βάσης Δεδομένων .....                                  | 34 |
| 4.2.1 Πίνακες χρηστών και ρόλων .....                                | 35 |
| 4.2.2 Προφίλ υποψηφίου και εργοδότη .....                            | 35 |
| 4.2.3 Πίνακες δεξιοτήτων .....                                       | 35 |
| 4.2.4 Αγγελίες και αιτήσεις εργασίας.....                            | 36 |
| 4.2.5 Ειδοποιήσεις, αγαπημένα και πρόσθετοι πίνακες .....            | 36 |
| 4.3 Λειτουργία Αντιστοίχισης Υποψηφίων και Αγγελιών.....             | 36 |
| 4.3.1 Ανάλυση δεξιοτήτων και βαθμός κάλυψης απαιτήσεων .....         | 37 |
| 4.3.2 Συμβατότητα τοποθεσίας και τρόπου εργασίας.....                | 37 |
| 4.3.3 Συμβατότητα τύπου απασχόλησης.....                             | 38 |
| 4.3.4 Συνδυασμός επιμέρους κριτηρίων και τελικό αποτέλεσμα .....     | 38 |
| 4.3.5 Πλεονεκτήματα της προσέγγισης.....                             | 38 |
| 4.4 Περιγραφή υλοποίησης διαφόρων εργασιών της εφαρμογής.....        | 40 |
| 4.4.1 Εγγραφή χρηστών .....                                          | 40 |
| 4.4.2 Λειτουργία Σύνδεσης και Διαχείρισης συνεδρίας .....            | 42 |
| 4.4.3 Λειτουργία Συμπλήρωσης και Διαχείρισης Προφίλ Υποψηφίου .....  | 44 |
| 4.4.4 Λειτουργία Δημιουργίας και Διαχείρισης Αγγελιών .....          | 45 |
| 4.4.5 Λειτουργία Αναζήτησης και Φιλτραρίσματος Αγγελιών .....        | 48 |
| 4.4.6 Λειτουργία Υποβολής Αίτησης και Διαχείρισης Συνεντεύξεων ..... | 48 |
| 4.4.7 Λειτουργία Αγαπημένων Αγγελιών και Υποψηφίων .....             | 51 |
| 4.4.8 Λειτουργία ειδοποιήσεων .....                                  | 53 |
| 4.4.9 Λειτουργία Γενικής Πλοήγησης και Κοινών Μοντέλων .....         | 55 |
| 4.4.10 Οργάνωση έργου και δομή αρχείων της εφαρμογής .....           | 56 |
| 4.5 Διεπαφή χρήστη και Πρότυπα παρουσίασης (Frontend).....           | 57 |
| 4.6 Υλοποίηση ασφάλειας και διαχείρισης πρόσβασης.....               | 58 |
| 4.7 Ροή δεδομένων στο σύστημα .....                                  | 58 |
| 5 Παραδείγματα με screenshots από την χρήση του συστήματος .....     | 59 |
| 5.1 Index .....                                                      | 59 |
| 5.2 Επιλογή ρόλου πριν την εγγραφή.....                              | 60 |
| 5.3 Λειτουργίες υποψηφίου εργαζομένου .....                          | 60 |
| 5.3.1 Εγγραφή .....                                                  | 60 |
| 5.3.2 Συμπλήρωση προσωπικών δεδομένων του χρήστη .....               | 61 |
| 5.3.3 Εισαγωγή προτιμήσεων στο προφίλ καριέρας .....                 | 61 |

|                                                                                      |    |
|--------------------------------------------------------------------------------------|----|
| 5.3.4 Εισαγωγή περαιτέρω δεξιοτήτων ή πιθανών πιστοποιήσεων .....                    | 62 |
| 5.3.5 Autocomplete .....                                                             | 63 |
| 5.3.6 Προσθήκη δεξιοτήτων και δυνατότητα βαθμολογίας επάρκειας .....                 | 63 |
| 5.3.7 Ανασκόπηση προφίλ (Profile Dashboard) .....                                    | 64 |
| 5.3.8 Προτεινόμενες αγγελίες βάσει συσχέτισης προσωπικών-ζητούμενων δεξιοτήτων ..... | 64 |
| 5.3.9 Προβολή αγγελίας .....                                                         | 65 |
| 5.3.10 Αίτηση .....                                                                  | 65 |
| 5.3.11 Προσθήκη αγγελίας εργασίας στα αγαπημένα .....                                | 66 |
| 5.3.12 Αιτήσεις που έχουν υποβληθεί .....                                            | 66 |
| 5.3.13 Ειδοποιήσεις σχετικά με την πορεία των αιτήσεων .....                         | 66 |
| 5.3.14 Αποδοχή – Απόρριψη συνέντευξης .....                                          | 67 |
| 5.3.15 Αποδοχή Συνέντευξης (Μήνυμα) .....                                            | 67 |
| 5.4 Εργοδότης .....                                                                  | 67 |
| 5.4.1 Δημιουργία προφίλ εταιρίας .....                                               | 67 |
| 5.4.2 Λογότυπο εταιρίας .....                                                        | 68 |
| 5.4.3 Επισκόπηση προφίλ εταιρίας .....                                               | 68 |
| 5.4.4 Ενεργές αγγελίες επιχείρησης και δυνατότητα επεξεργασίας υπαρχόντων .....      | 69 |
| 5.4.5 Υποβολή νέας αγγελίας εργασίας .....                                           | 70 |
| 5.4.6 Επισκόπηση νέας αγγελίας .....                                                 | 70 |
| 5.4.7 Ειδοποίηση για υποβολή νέας αίτησης από υποψήφιο εργαζόμενο .....              | 71 |
| 5.4.8 Προβολή αίτησης του υποψήφιου εργαζομένου .....                                | 71 |
| 5.4.9 Αποδοχή – Απόρριψη Συνέντευξης .....                                           | 72 |
| 5.4.9 Ενημερωτικό μήνυμα κατάστασης αίτησης .....                                    | 72 |
| 6 Εγχειρίδιο χρήστη .....                                                            | 73 |
| 6.1 Εισαγωγή .....                                                                   | 73 |
| 6.2 Εγγραφή και Σύνδεση Χρηστών .....                                                | 73 |
| 6.3 Αρχική Σελίδα και Πίνακας Ελέγχου .....                                          | 73 |
| 6.4 Προφίλ Χρήστη .....                                                              | 74 |
| 6.5 Αγγελίες Εργασίας και Αιτήσεις .....                                             | 74 |
| 6.6 Αντιστοίχιση και Πιθανές Προτάσεις .....                                         | 74 |
| 6.7 Αγαπημένα και Ειδοποιήσεις .....                                                 | 74 |
| 6.8 Επικοινωνία .....                                                                | 74 |
| 7 Συμπεράσματα – Δυσκολίες κατά την υλοποίηση και μελλοντικές βελτιώσεις .....       | 74 |
| 7.1 Συμπεράσματα .....                                                               | 74 |
| 7.2 Αναμενόμενα οφέλη .....                                                          | 75 |
| 7.3 Μελλοντικές επεκτάσεις και βελτιώσεις .....                                      | 76 |
| 7.3.1 Βελτίωση του μηχανισμού αντιστοίχισης .....                                    | 76 |
| 7.3.2 Προσθήκη συστήματος αναφορών και στατιστικών .....                             | 76 |

|                                                         |    |
|---------------------------------------------------------|----|
| 7.3.3 Ανάπτυξη εφαρμογής για κινητές συσκευές.....      | 76 |
| 7.3.4 Εμπλουτισμός του συστήματος ειδοποιήσεων .....    | 76 |
| 7.3.5 Προσθήκη συστήματος επικοινωνίας (chat) .....     | 77 |
| 7.3.6 Ενίσχυση της ασφάλειας.....                       | 77 |
| 7.3.7 Βελτίωση διαχείρισης αρχείων και βιογραφικών..... | 77 |
| 7.4 Περιορισμοί της παρούσας μελέτης .....              | 77 |
| 6 Βιβλιογραφία .....                                    | 78 |

## ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

|                                                                                                                                                       |    |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Εικόνα 1: Δημοφιλέστεροι ιστότοποι εύρεσης εργασίας .....                                                                                             | 11 |
| Εικόνα 2: Χρήση των ATS κατά την προεπιλογή.....                                                                                                      | 13 |
| Εικόνα 3: Μήνυμα παρακολούθησης για νέες εργασίες που ταιριάζουν σε ένα σύγχρονο ATS .....                                                            | 13 |
| Εικόνα 4: Αρχιτεκτονική συστήματος για την ιεράρχηση αποτελεσμάτων αναζήτησης.....                                                                    | 15 |
| Εικόνα 5: Μερικά από τα πλεονεκτήματα χρήσης AI εργαλείων στο HR.....                                                                                 | 16 |
| Εικόνα 6: Κατά πόσο ηθικό είναι το AI; .....                                                                                                          | 18 |
| Εικόνα 7: Μισθός και εμπειρία ανα τύπο προγραμματιστή (Stack Overflow Survey (2024)) .....                                                            | 21 |
| Εικόνα 8: Κεντρική σελίδα του kariera.gr, από τις δημοφιλέστερες σελίδες εύρεσης εργασίας στην Ελλάδα .....                                           | 24 |
| Εικόνα 9: Use Case Diagram (UML).....                                                                                                                 | 26 |
| Εικόνα 10: Δομή πολυεπίπεδης αρχιτεκτονικής συστήματος .....                                                                                          | 27 |
| Εικόνα 11: Διάγραμμα Οντοτήτων–Σχέσεων (Entity-Relationship Diagram) της βάσης δεδομένων. ..                                                          | 34 |
| Εικόνα 12: Εννοιολογική απεικόνιση της ροής δεδομένων και των βασικών σταδίων του αλγορίθμου αντιστοίχισης στην πλατφόρμα DevJobs. ....               | 37 |
| Εικόνα 13: Κλάση που υλοποιεί τον αλγόριθμο αντιστοίχισης μεταξύ επαγγελματικού προφίλ και αγγελίας εργασίας .....                                    | 39 |
| Εικόνα 14: Κλάση υπεύθυνη για την αποθήκευση του αποτελέσματος με τον αντίστοιχο δείκτη συνάφειας.....                                                | 39 |
| Εικόνα 15: Κλάση που αποδίδει τα αποτελέσματα της αντιστοίχισης, περιλαμβάνοντας την συνολική βαθμολογία και τις κύριες δεξιότητες.....               | 40 |
| Εικόνα 16: Backend στιγμιότυπο της κλάσης εγγραφής που υλοποιεί την δρομολόγηση και την αρχική φόρμα επιλογής ρόλου για την διαδικασία εγγραφής ..... | 41 |
| Εικόνα 17: Μέρος της λειτουργίας αποθήκευσης προφίλ του χρήστη, με έλεγχο ταυτοποίησης, επαλήθευση ρόλου και ενημέρωση του πίνακα "UserProfile".....  | 42 |
| Εικόνα 18: Κλάση για authorization .....                                                                                                              | 43 |
| Εικόνα 19: Κλάση που ελέγχει την ολοκλήρωση του προφίλ, σχετικά με την πρόσβαση σε προστατευμένες πληροφορίες .....                                   | 44 |
| Εικόνα 20: Παράδειγμα κλάσης υπεύθυνης για την καταχώριση των προτιμήσεων του χρήστη .....                                                            | 45 |
| Εικόνα 21: Κλάση που αφορά τον έλεγχο και την διαχείριση των αγγελιών εργασίας .....                                                                  | 46 |
| Εικόνα 22: Παράδειγμα κλάσης που είναι υπεύθυνη για την ανάκτηση, διαγραφή και αποθήκευση των αγγελιών εργασίας .....                                 | 47 |
| Εικόνα 23: Κλάση που συνδράμει στην συσχέτιση των δεξιοτήτων με τις υπάρχουσες αγγελίες.....                                                          | 47 |
| Εικόνα 24: Κλάση υπεύθυνη για την προβολή αγγελιών εργασίας και για την διαχείριση της αρχικής σελίδας.....                                           | 48 |
| Εικόνα 25: Διαχείριση αιτημάτων που αφορούν την προβολή, υποβολή και παρακολούθηση της κατάστασης των αιτήσεων εργασίας.....                          | 49 |



|                                                                                                                                                           |    |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Εικόνα 26: Κλάση υπεύθυνη για τον έλεγχο εγκυρότητας, την αποθήκευση αιτήσεων και την αποστολή σχετικών ειδοποιήσεων .....                                | 50 |
| Εικόνα 27: Παροχή πρόσβασης στην βάση δεδομένων για τις αιτήσεις εργασίας, δίδοντας εξειδικευμένα ερωτήματα με πλήρη φόρτιση σχετιζόμενων δεδομένων ..... | 50 |
| Εικόνα 28: Κλάση που αναλαμβάνει την δημιουργία, την αποθήκευση και την άμεση ενημέρωση των χρηστών μέσω έγκαιρων ειδοποιήσεων .....                      | 51 |
| Εικόνα 29: Μοντέλο που αποθηκεύει πληροφορίες της συνέντευξης, όπως ημερομηνία, ώρα και τρόπο διεξαγωγής .....                                            | 51 |
| Εικόνα 30: Κλάση που αποθηκεύει τους αγαπημένους υποψήφιους του εργοδότη .....                                                                            | 52 |
| Εικόνα 31: Κλάση που διαχειρίζεται αγαπημένους υποψήφιους και αγγελίες .....                                                                              | 52 |
| Εικόνα 32: Υπεύθυνη κλάση για την επιχειρησιακή λογική των αγαπημένων εγγραφών .....                                                                      | 53 |
| Εικόνα 33: Κλάση που συνδράμει στην αποθήκευση αγαπημένων αγγελιών από πλευράς υποψηφίου .....                                                            | 53 |
| Εικόνα 34: Υλοποίηση και δημιουργία ειδοποιήσεων στο σύστημα .....                                                                                        | 54 |
| Εικόνα 35: Προβολή και διαχείριση ειδοποιήσεων του χρήστη .....                                                                                           | 55 |
| Εικόνα 36: Οντότητα βάσης δεδομένων για την αποθήκευση ειδοποιήσεων χρηστών .....                                                                         | 55 |
| Εικόνα 37: Οργάνωση του έργου στο backend .....                                                                                                           | 57 |
| Εικόνα 38: Πρώτη διεπαφή χρήστη-συστήματος .....                                                                                                          | 59 |
| Εικόνα 39: Διαχωρισμός ρόλων κατά την εγγραφή .....                                                                                                       | 60 |
| Εικόνα 40: Φόρμα Εγγραφής .....                                                                                                                           | 60 |
| Εικόνα 41: Εισαγωγή προσωπικών δεδομένων του χρήστη .....                                                                                                 | 61 |
| Εικόνα 42: Προφίλ καριέρας (Μέρος 1) .....                                                                                                                | 62 |
| Εικόνα 43: Προφίλ καριέρας (Μέρος 2) .....                                                                                                                | 62 |
| Εικόνα 44: Πρόταση πιθανών περιοχών αναζήτησης εργασίας .....                                                                                             | 63 |
| Εικόνα 45: Ενσωμάτωση πιθανών δεξιοτήτων του χρήστη κατά την διαμόρφωση του προφίλ του ....                                                               | 63 |
| Εικόνα 46: Προσθήκη προσωπικής δεξιότητας .....                                                                                                           | 63 |
| Εικόνα 47: Profile Dashboard .....                                                                                                                        | 64 |
| Εικόνα 48: Προτεινόμενες Εργασίες .....                                                                                                                   | 65 |
| Εικόνα 49: Προβολή θέσης εργασίας .....                                                                                                                   | 65 |
| Εικόνα 50: Υποβολή αίτησης και συνοδευτικού μηνύματος .....                                                                                               | 66 |
| Εικόνα 51: Προσθήκη στα αγαπημένα .....                                                                                                                   | 66 |
| Εικόνα 52: Ατομικές αιτήσεις .....                                                                                                                        | 66 |
| Εικόνα 53: Notifications .....                                                                                                                            | 67 |
| Εικόνα 54: Διαδικασία αποδοχής ή απόρριψης προτεινόμενης συνέντευξης .....                                                                                | 67 |
| Εικόνα 55: Επιβεβαιωμένη συνέντευξη .....                                                                                                                 | 67 |
| Εικόνα 56: Εισαγωγή πληροφοριών εταιρίας .....                                                                                                            | 68 |
| Εικόνα 57: Εισαγωγή εικόνας εταιρίας .....                                                                                                                | 68 |
| Εικόνα 58: Εταιρικό προφίλ .....                                                                                                                          | 69 |
| Εικόνα 59: Αγγελίες επιχείρησης .....                                                                                                                     | 69 |
| Εικόνα 60: Υποβολή αναζήτησης νέου υπαλλήλου .....                                                                                                        | 70 |
| Εικόνα 61: Προβολή νέας εργασίας .....                                                                                                                    | 71 |
| Εικόνα 62: Notification για νέα υποβολή αίτησης .....                                                                                                     | 71 |
| Εικόνα 63: Επισκόπηση στοιχείων αίτησης νέου υποψηφίου .....                                                                                              | 72 |
| Εικόνα 64: Υποβολή συνέντευξης και λοιπών λεπτομερειών που αφορούν την υποβληθείσα αίτηση .....                                                           | 72 |
| Εικόνα 65: Notification σχετικά με την κατάσταση της αίτησης .....                                                                                        | 73 |

## 1 Εισαγωγή

### 1.1 Στόχοι - περιγραφή του προβλήματος

Κύριος στόχος της παρούσας έρευνας είναι η ανάπτυξη μιας εφαρμογής διασύνδεσης εργοδότη-εργαζομένου, με επίκεντρο θέσεις πληροφορικής και προγραμματισμού, η οποία θα βασίζεται στις ανάγκες και στα κριτήρια που θέτει τόσο ο εργαζόμενος όσο και ο εργοδότης. Όπως αναφέρουν οι Tewari, Joshi και Tewari (2024), ένα από τα βασικά προβλήματα που παρατηρούνται στην τωρινή αγορά εργασίας είναι ότι, παρά το γεγονός ότι ο υποψήφιος εργαζόμενος συχνά πληροί σημαντικά κριτήρια για την προσφερόμενη θέση, όπως γνώση διαφόρων γλωσσών προγραμματισμού ή δημιουργία πολλαπλών έργων, συχνά απορρίπτεται, εξαιτίας ελλιπών μηχανισμών αξιολόγησης και ανάλυσης των δεδομένων του. Έτσι, ένα επικείμενο αυτοματοποιημένο απορριπτικό μήνυμα ή η αγνόηση της αίτησης από το τμήμα του ανθρώπινου δυναμικού, ενδέχεται να επιφέρουν την δυσαρέσκεια και την δυσπιστία του χρήστη, απέναντι στους ήδη υπάρχοντες μηχανισμούς.

Σε παρόμοιο κλίμα, οι εργοδότες με την σειρά τους συχνά αμελούν την τεχνολογική εξέλιξη και συνέχεια του εκάστοτε οργανισμού, μή επενδύοντας σε μηχανισμούς που θα εκμηδενίσουν την διαδικασία εύρεσης νέων υποψηφίων. Για τον λόγο αυτό, συχνά χάνουν ταλέντα ή δεν τα εντοπίζουν έγκαιρα, εξαιτίας του μεγάλου όγκου αιτήσεων και πληροφορίας. Η απουσία δομημένων μεθόδων αξιολόγησης επιβραδύνει την λήψη αποφάσεων και δημιουργεί σημαντικά κενά στην επιλογή προσωπικού. Παράλληλα, η αναζήτηση του κατάλληλου προφίλ είναι ένα εξαιρετικά χρονοβόρο και απαιτητικό στάδιο, γεγονός που ενδέχεται να επηρεάσει την αποδοτικότητα του οργανισμού.

Επιπλέον, ιδίως στα μεγαλύτερα ηλικιακά στρώματα, ένας συμπληρωματικός παράγοντας που ενδέχεται να δυσχεραίνει την πρόσβαση και την εμπειρία των ομάδων αυτών συνιστά η περιορισμένη εξοικείωσή τους με τις σύγχρονες τεχνολογίες και τα αντίστοιχα πληροφοριακά συστήματα. Η μειωμένη ψηφιακή επάρκεια δύναται να λειτουργήσει ανασταλτικά, τόσο ως προς την αναζήτηση ενός είδους απασχόλησης, όσο και ως προς την αποτελεσματική αξιοποίηση ψηφιακών εργαλείων που κατέχουν πλέον σημαντικότερη θέση στον ευρύτερο επαγγελματικό βίο. Σύμφωνα με τους Stojanović, Milošević και Arsić (2023), οι χαμηλές ψηφιακές δεξιότητες συνδέονται άμεσα με μειωμένα ποσοστά απασχόλησης στις ηλικίες 55–64, γεγονός που επιβεβαιώνει τις δυσκολίες που αντιμετωπίζουν οι μεγαλύτεροι ενήλικες στη σύγχρονη ψηφιακή αγορά εργασίας.

Συνεπώς, κρίνεται αναγκαία η δημιουργία ενός συστήματος που θα αντιμετωπίζει τυχών προβλήματα και καθυστερήσεις. Η προτεινόμενη εφαρμογή δίνει ιδιαίτερη έμφαση στη βελτίωση της διαδικασίας αντιστοίχισης υποψηφίων και θέσεων, μέσω της ύπαρξης σαφών κριτηρίων, αλγορίθμων και αυτοματοποιημένων λειτουργιών. Επιδιώκει, επίσης, να ενισχύσει τη διαφάνεια και την αντικειμενικότητα ως προς την λήψη αποφάσεων, μειώνοντας την πιθανότητα λανθασμένων ή άδικων αξιολογήσεων που συναντάει κανείς συχνά στα παραδοσιακά μέσα διαχωρισμού και επιλογής εργαζομένων. Επομένως, συγκροτείται ένα σύστημα αξιόπιστο και αποτελεσματικό, στο οποίο κύριος στόχος τίθεται η προσωπική ανάδειξη των χρηστών και η από κοινού εξέλιξη εταιρίας-εργαζομένων.

### 1.2 Αναφορά σε παρόμοιες εφαρμογές

Η ανάπτυξη ψηφιακών πλατφορμών εύρεσης εργασίας έχει σημειώσει ραγδαία εξέλιξη τα τελευταία έτη, ακολουθώντας τη γενικότερη ανάγκη και τάση για ψηφιοποίηση των κοινωνικών και οικονομικών δραστηριοτήτων. Πλατφόρμες όπως το “LinkedIn”, το “Indeed” και το “Glassdoor” συνιστούν τα βασικά εργαλεία διασύνδεσης υποψηφίων και εργοδοτών σε παγκόσμιο επίπεδο. Οι συγκεκριμένες εφαρμογές επιτρέπουν τη δημιουργία επαγγελματικών προφίλ, τη δημοσίευση αγγελιών, την υποβολή αιτήσεων, αλλά ενισχύουν εξίσου και την ανθρώπινη αλληλεπίδραση, μέσω της δυνατότητας σύνδεσης με άλλα άτομα και ανταλλαγής μηνυμάτων μεταξύ τους. Παράλληλα, προσφέρουν δυνατότητες αναζήτησης και φιλτραρίσματος προσφερόμενων θέσεων ή ενεργών αγγελιών, βάσει κριτηρίων που θέτει ο ίδιος ο χρήστης.



Εικόνα 1: Δημοφιλέστεροι ιστότοποι εύρεσης εργασίας

Η λειτουργικότητα των συστημάτων αυτών βασίζεται σε σύγχρονες τεχνολογίες και μεθόδους, αξιοποιώντας βάσεις δεδομένων, ποικίλους μηχανισμούς και αυτοματισμούς ως προς την αντιστοίχιση προσόντων και απαιτήσεων. Μέσω των αλγορίθμων εξίσου που χρησιμοποιούνται, γίνεται η βέλτιστη δυνατή προσπάθεια για την αποδοτικότερη σύζευξη μεταξύ της προσφοράς και της ζήτησης εργασίας. Έτσι, σε αντίθεση με τις παραδοσιακές μεθόδους εύρεσης εργασίας, ο χρόνος αναμονής τείνει να μειώνεται αισθητά.

Επίσης, τα σύγχρονα μέσα προσδίδουν μεγαλύτερη διαφάνεια στην λειτουργία της αγοράς εργασίας, καθώς παρέχουν πληθώρα πληροφοριών σχετικά με τις απαιτήσεις των θέσεων, την μισθοδοσία και το περιβάλλον εργασίας του χρήστη. Άλλωστε, στην σύγχρονη εποχή των καθημερινών ανακατατάξεων και της μισαλλοδοξίας, το αίσθημα ασφάλειας στον επιχειρηματικό κλάδο είναι ιδιαίτερα θεμιτό και προαπαιτούμενο, τόσο για τον ίδιο τον υποψήφιο, όσο και για τον εκάστοτε οργανισμό.

Άξια αναφοράς αποτελεί και η συνεισφορά των κολοσσών αυτών ως προς την αλλαγή του μοντέλου εργασίας και ειδικότερα κατά την διάρκεια του κορονοϊού. Η υβριδική μέθοδος σε συνδυασμό με την εξ' αποστάσεως εργασία έχουν συνδράμει σημαντικά στον εκμηδενισμό των γεωγραφικών αποστάσεων, στην ελάττωση εξόδων των εταιριών, αλλά και στην αποδοτικότερη συχνά εικόνα των ίδιων των εργαζομένων.

Στην αντίπερα όχθη, παρά την ισχυρή συμβολή της συγκεκριμένης αγοράς ως προς το επιχειρίν, υπάρχουν ορισμένες διαστάσεις που δεν δύνανται να καλυφθούν επαρκώς από τα ήδη υπάρχοντα περιβάλλοντα. Πρωτίστως, οι εφαρμογές αυτές δεν είναι ακόμη σε θέση να υποκαταστήσουν στον υπερθετικό βαθμό την προσωπική επαγγελματική καθοδήγηση, η οποία προϋποθέτει την ανθρώπινη αλληλεπίδραση, την εξατομικευμένη αξιολόγηση δεξιοτήτων του κάθε ατόμου και της εις βάθος κατανόηση των προσωπικών του αναγκών και ιδιαιτεροτήτων.

Όπως προαναφέρθηκε, άτομα μεγαλύτερης ηλικίας, ενδέχεται να αισθανθούν τον αποκλεισμό από διάφορες θέσεις εργασίας, λόγω της ελλιπούς τους ενημέρωσης, σχετικά με την χρήση και τις τεχνολογίες των συγκεκριμένων συστημάτων. Ο ίδιος ισχυρισμός θα μπορούσε να ισχύσει εξίσου και για νεαρότερους ανθρώπους, οι οποίοι δραστηριοποιούνται σε μη τεχνολογικούς τομείς, δημιουργώντας έτσι ένα αθέμιτο πλεονέκτημα υπέρ όσων διαθέτουν αυξημένες ψηφιακές δεξιότητες.

## 2 Ανασκόπηση πεδίου

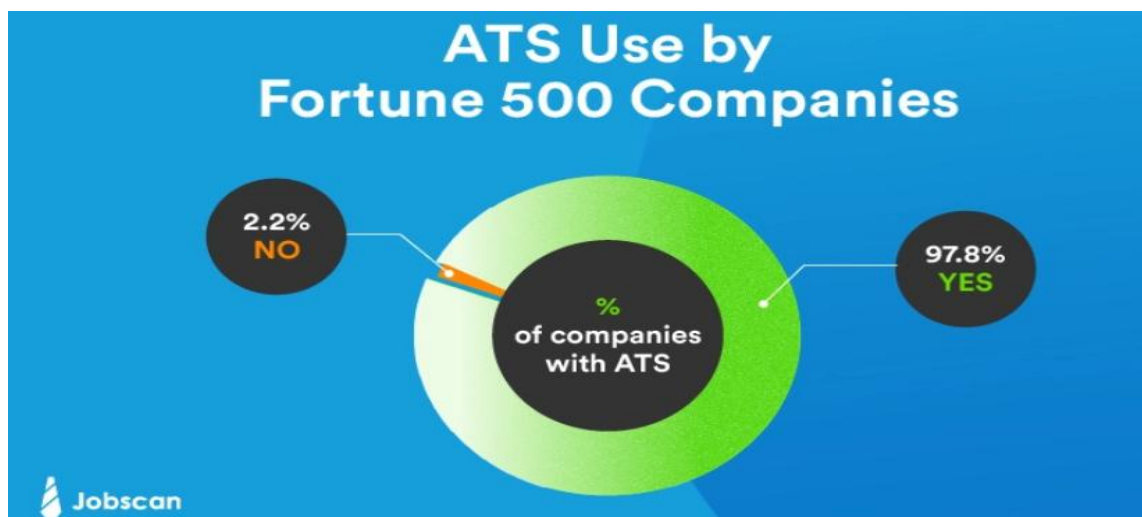
Η παρούσα βιβλιογραφική ανασκόπηση εξετάζει τη σημασία των σύγχρονων συστημάτων εύρεσης εργασίας και την εξέλιξή τους στο σημερινό τεχνολογικό περιβάλλον τόσο γενικότερα όσο και για προγραμματιστές και επαγγελματίες της πληροφορικής. Η πρόοδος στην τεχνητή νοημοσύνη, την μηχανική μάθηση και τα προσαρμοστικά συστήματα έχει αλλάξει ριζικά τον τρόπο με τον οποίο σήμερα εργοδότης και εργαζόμενος αλληλοεπιδρούν μεταξύ τους. Οι τεχνολογίες αυτές, σε συνδυασμό με τα δεδομένα που θέτει ο εκάστοτε χρήστης ως δεξιότητες ή κριτήρια, επιτρέπουν την παροχή εξατομικευμένων λύσεων και προτάσεων, βελτιώνοντας σημαντικά την πιθανότητα επίτευξης συμφωνίας και έναρξης συνεργασίας, σύμφωνα με τις ανάγκες και των δύο πλευρών. Παράλληλα, τα συστήματα αυτά ενδέχεται να δημιουργήσουν νέες προκλήσεις, που αφορούν την προστασία και διαχείριση των δεδομένων και την συμμόρφωση με τα σημερινά κανονιστικά πλαίσια. Στόχος της μελέτης κρίνεται λοιπόν η ανάλυση όλων των πτυχών των νέων τεχνολογικών ανακατατάξεων και η παρουσίαση των τάσεων που διαμορφώνονται στο μέλλον του τραπεζικού τομέα.

### 2.1 Η συστηματική εξέλιξη ως προς την εύρεση εργασίας

Η ανάγκη για καινοτομία και πρόοδο σε όλο το φάσμα της ανθρώπινης ιδιοσυγκρασίας έχει επιφέρει σημαντικότερες εξελίξεις τις τελευταίες δεκαετίες, ακόμη και στην διαδικασία εύρεσης εργασίας. Παραδοσιακά, οι υποψήφιοι εργαζόμενοι βασίζονταν σε έντυπες αγγελίες, προσωπικά δίκτυα και άμεση επικοινωνία με εργοδότες, διαδικασίες συχνά χρονοβόρες, αποσπασματικές και περιορισμένες γεωγραφικά. Ωστόσο, η σταδιακή ψηφιοποίηση της πληροφόρησης και η διεύρυνση του διαδικτύου στις ζωές των πολιτών ήρθαν να αλλάξουν για τα καλά τις μέχρι πρότινος διαδικασίες, δίνοντάς τους αρχικά την δυνατότητα να προσαρμόσουν το πεδίο αναζήτησης σύμφωνα με τις δικές τους προσωπικές ανάγκες, αλλά και βάσει των κριτηρίων που θέτει η εκάστοτε επιχείρηση. Παράλληλα, οργανισμός και μονάδα έχουν εκατοντάδες χιλιάδες επιλογές για επίτευξη συνεργασίας, γεγονός που συντελεί στην διαφάνεια και την αξιοπιστία του κάθε συστήματος.

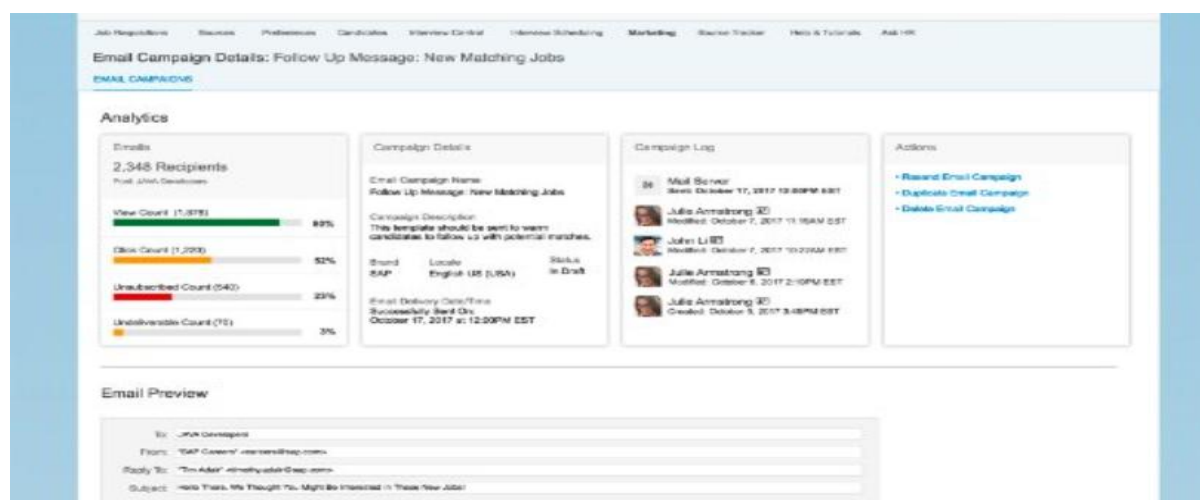
Με την εμφάνιση των πρώτων διαδικτυακών πλατφορμών εύρεσης εργασίας στα τέλη της δεκαετίας του 1990, η διαδικασία επιλογής προσωπικού άρχισε να υιοθετεί πιο συστηματικές και αυτοματοποιημένες πρακτικές. Πλατφόρμες που προαναφέρθηκαν, με καινοτόμα την διαδικτυακή πλατφόρμα “LinkedIn”, διαδραμάτισαν και συνεχίζουν να κατέχουν καθοριστικό ρόλο στην άμεση εύρεση εργασίας, παρέχοντας εξατομικευμένα εργαλεία και υπηρεσίες. Κατά συνέπεια, οι ίδιοι οι υποψήφιοι ανέπτυξαν σημαντικά πλεονεκτήματα έναντι της αγοράς και οι ίδιοι οι εργοδότες είχαν την ικανότητα προεπιλογής και στοχευμένης αναζήτησης, μειώνοντας σημαντικά το κόστος, τον χρόνο πρόσληψης, αλλά και το ρίσκο που ενδέχεται να επιφυλάσσεται από μία άστοχη προσθήκη.

Σήμερα, η εύρεση εργασίας έχει μετασχηματιστεί σε μία εξαιρετικά δομημένη διαδικασία που βασίζεται σε δεδομένα, στη μηχανική μάθηση και στη ψηφιακή αλληλεπίδραση. Οι σύγχρονες πλατφόρμες απασχόλησης ενσωματώνουν εξελιγμένους αλγόριθμους αντιστοίχισης, οι οποίοι αξιολογούν δεξιότητες, εμπειρία και προτιμήσεις, παρέχοντας παράλληλα πληροφορίες για τις συνθήκες, τις αμοιβές και το εργασιακό περιβάλλον. Επιπλέον, η αυξανόμενη χρήση αυτοματοποιημένων εργαλείων, όπως τα Applicant Tracking Systems (ATS), σύμφωνα με την Purcell (2025), έχει αλλάξει ριζικά τον τρόπο που τα συστήματα διαβάζουν, αναλύουν και αξιολογούν τα βιογραφικά. Η ίδια έρευνα καταγράφει ότι το 97.8% των εταιρειών της Fortune 500 χρησιμοποιούν κάποιο ATS στις διαδικασίες πρόσληψης, γεγονός που υπογραμμίζει τον καθοριστικό ρόλο της αυτοματοποίησης.



Εικόνα 2: Χρήση των ATS κατά την προεπιλογή

Παράλληλα, η SAP (2025) επισημαίνει ότι τα ATS είναι ιδιαίτερα κρίσιμα για οργανισμούς με υψηλό όγκο αιτήσεων, καθώς μειώνουν σημαντικά τον χρόνο διαλογής, επιταχύνουν τον κύκλο πρόσληψης και διευκολύνουν την παρακολούθηση της προόδου των υποψηφίων και των εργαζομένων.



Εικόνα 3: Μήνυμα παρακολούθησης για νέες εργασίες που ταιριάζουν σε ένα σύγχρονο ATS

## 2.2 Αλγόριθμοι – Τεχνητή Νοημοσύνη και Μηχανική Μάθηση στην Αγορά Εργασίας

Η ενσωμάτωση αλγορίθμων Τεχνητής Νοημοσύνης και Μηχανικής Μάθησης στην αγορά εργασίας έχει επιφέρει έναν ριζικό μετασχηματισμό στον τρόπο με τον οποίο οι εργοδότες αναζητούν, αξιολογούν και επιλέγουν υποψήφιους εργαζόμενους. Οι σύγχρονες ψηφιακές πλατφόρμες απασχόλησης λειτουργούν πλέον ως σύνθετα οικοσυστήματα δεδομένων, όπου οι εξελιγμένοι αλγόριθμοι επεξεργάζονται πληροφορίες που αφορούν δεξιότητες, επαγγελματική εμπειρία, προτιμήσεις και «μοτίβα» ή συνήθειες των χρηστών. Σκοπός των πρακτικών αυτών συνιστά η βελτίωση της διαδικασίας αντιστοίχισης μεταξύ διαθέσιμων θέσεων και υποψηφίων, εξασφαλίζοντας μεγαλύτερη αποτελεσματικότητα και ακρίβεια, σε

σχέση με παραδοσιακά μοντέλα αναζήτησης, τα οποία βασίζονταν στο χαρτί και στο μολύβι ή στην διαζώσης επικοινωνία.

Στο πλαίσιο αυτό, η εφαρμογή μεθόδων μηχανικής μάθησης, όπως η ταξινόμηση, η ομαδοποίηση και η πρόβλεψη, επιτρέπει στις πλατφόρμες να αναγνωρίζουν πρότυπα στα δεδομένα που συλλέγουν από εκατομμύρια χρήστες. Τα συστήματα ταξινόμησης (classification models) χρησιμοποιούνται για την αξιολόγηση της καταλληλότητας ενός υποψηφίου για μια συγκεκριμένη θέση, λαμβάνοντας υπόψη παραμέτρους όπως το βιογραφικό, τις λέξεις-κλειδιά και το ιστορικό αναζητήσεων. Οι τεχνικές ομαδοποίησης (clustering), από την άλλη, αξιοποιούνται για την κατηγοριοποίηση υποψηφίων ή θέσεων εργασίας, βάσει κοινών χαρακτηριστικών, προσδίδοντας οργάνωση στα δεδομένα και βελτιστοποίηση ως προς τις στοχευμένες προτάσεις. Παράλληλα, αλγόριθμοι πρόβλεψης χρησιμοποιούνται εξίσου για την εκτίμηση ενός ποσοστού πιθανότητας, κατά το οποίο ο εκάστοτε υποψήφιος εξετάζεται αν ταιριάζει επιτυχώς σε μια θέση, βασιζόμενοι σε ιστορικά δεδομένα πρόσληψης και συμπεριφοράς.

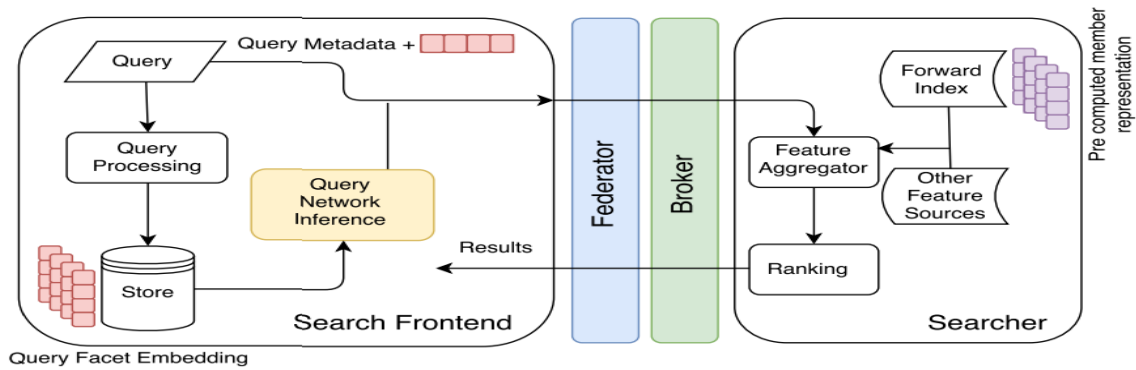
Σε πρώτο επίπεδο, ιδιαίτερα σημαντική είναι η μετάβαση ενός ψηφιακού εγγράφου σε γλώσσα μηχανής. Δεδομένα, όπως το περιεχόμενο ενός βιογραφικού σημειώματος, μετατρέπονται σε αριθμητικές αναπαραστάσεις κατάλληλες για αλγοριθμική επεξεργασία. Όπως αναφέρουν οι Devlin et al. (2019), σύγχρονες τεχνικές ενσωμάτωσης κειμένου, όπως τα deep contextual embeddings, επιτρέπουν την αποτύπωση της σημασιολογικής δομής του κειμένου με υψηλή ακρίβεια. Τεχνικές όπως το TF-IDF, τα text embeddings και συναφείς μέθοδοι αποτυπώνουν σημασιολογικές σχέσεις με επίπεδα λεπτομέρειας πολύ ανώτερα από παραδοσιακές στατιστικές προσεγγίσεις, κάτι που βελτιώνει σημαντικά την αποτελεσματικότητα των συστημάτων πρόσληψης βασισμένων σε τεχνητή νοημοσύνη.

Ως προς το αλγοριθμικό επίπεδο, κυρίαρχο ρόλο διαδραματίζουν τα supervised learning models, τα οποία εκπαιδεύονται πάνω σε ιστορικά δεδομένα προσλήψεων με σκοπό την πρόβλεψη και τη μείωση του ρίσκου σχετικά με μία επικείμενη πρόσληψη. Σύμφωνα με τους Ramanath et al. (2018), τέτοια μοντέλα αποτελούν βασικό πυλώνα των συστημάτων talent search, καθώς επιτρέπουν την πρόβλεψη της καταλληλότητας ενός υποψηφίου βάσει μεγάλων συνόλων δεδομένων αλληλεπιδράσεων και προηγούμενων προσλήψεων. Τα Νευρωνικά Δίκτυα χρησιμοποιούνται ευρέως με τη σειρά τους σε τέτοιες εφαρμογές, καθώς έχουν την ευχέρεια να εντοπίζουν πολύπλοκα πρότυπα συσχέτισης μεταξύ όλων των κριτηρίων και της επιτυχούς τοποθέτησης σε θέση εργασίας. Σημαντικό μειονέκτημα των μοντέλων αυτών μπορεί να θεωρηθεί η εγκυρότητα των πληροφοριών που παρέχουν, καθώς για την ομαλή λειτουργία τους απαιτείται μία διαρκής επικαιροποίηση και εκπαίδευση των μοντέλων.

## **2.3 Οφέλη της Αλγοριθμικής Ανάλυσης και Αυτοματοποιημένης Αντιστοίχισης**

Η αλγοριθμική διαδικασία και ανάλυση, σε συνδυασμό με την αυτοματοποιημένη αντιστοίχιση, έχουν επιφέρει σημαντικά οφέλη στον τρόπο λειτουργίας της σύγχρονης αγοράς εργασίας. Σύμφωνα με τους Ramanath et al. (2018), η αντιστοίχιση υποψηφίων και θέσεων υλοποιείται πλέον μέσω deep learning μοντέλων, τα οποία επεξεργάζονται εκατομμύρια προφίλ και αγγελίες σε πραγματικό χρόνο, παρέχοντας υψηλής ακρίβειας αποτελέσματα. Με τον τρόπο αυτό, τα συστήματα αυτά προσφέρουν

εξατομικευμένες προτάσεις σε ελάχιστο χρόνο, βελτιώνοντας σημαντικά την ακρίβεια και τη συνέπεια της διαδικασίας αντιστοίχισης, χωρίς όμως να εξαλείφουν πλήρως τον κίνδυνο σφαλμάτων.



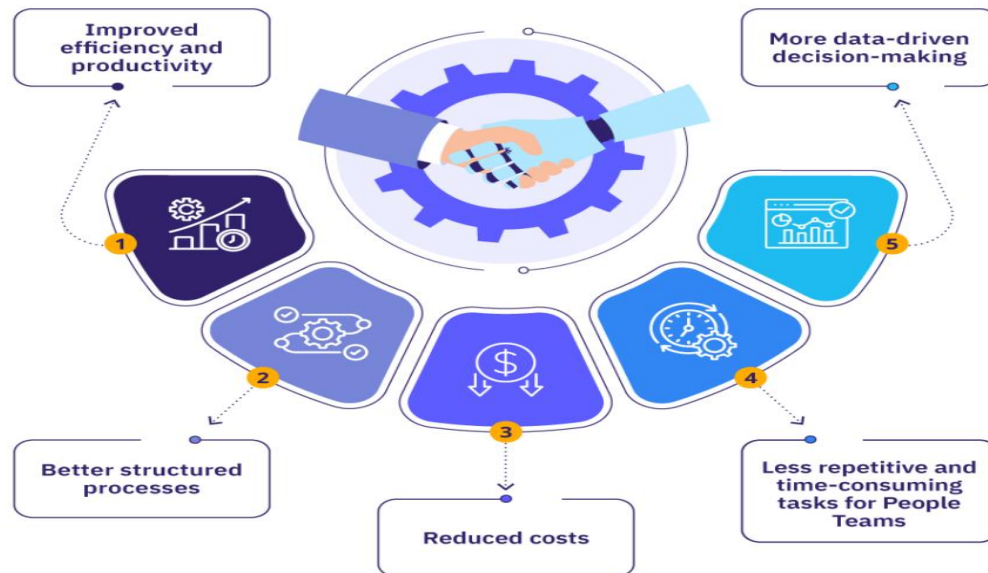
Εικόνα 4: Αρχιτεκτονική συστήματος για την ιεράρχηση αποτελεσμάτων αναζήτησης

Σύμφωνα με την ετήσια έκθεση της Deloitte (2023), η υιοθέτηση προγνωστικών μοντέλων μηχανικής μάθησης συμβάλλει στη μείωση λανθασμένων προσλήψεων και στη βελτίωση της συνολικής αποδοτικότητας του ανθρώπινου δυναμικού. Παράλληλα, η χρήση τεχνικών πρόβλεψης (predictive analytics) επιτρέπει στους οργανισμούς να εντοπίζουν μελλοντικές ανάγκες σε προσωπικό και κρίσιμες δεξιότητες. Όπως αναφέρει το World Economic Forum (2023), τέτοιου είδους αναλυτικά εργαλεία προσδίδουν σημαντικό στρατηγικό πλεονέκτημα και ενισχύουν τον μακροπρόθεσμο σχεδιασμό



ανθρώπινου δυναμικού, μετατρέποντας ουσιαστικά το τμήμα ανθρωπίνου δυναμικού σε στρατηγικό πυλώνα της επιχείρησης.

## Benefits of AI in HR



Εικόνα 5: Μερικά από τα πλεονεκτήματα χρήσης AI εργαλείων στο HR

Η δυνατότητα χαρτογράφησης των τάσεων του εργατικού δυναμικού επιτρέπει την ανάπτυξη αποδοτικότερων εκπαιδευτικών πολιτικών και την ενίσχυση της δια βίου μάθησης, με τρόπο που φέρεται να ευθυγραμμίζεται με τις μελλοντικές και τωρινές απαιτήσεις της αγοράς. Σύμφωνα με το World Economic Forum (2023), έξι στους δέκα εργαζόμενους θα χρειαστούν αναβάθμιση δεξιοτήτων μέχρι το 2027, ενώ μόνο οι μισοί διαθέτουν επαρκείς ευκαιρίες κατάρτισης σήμερα. Σε πρώτο πλάνο, βασικό αντικείμενο αποτελεί η αναλυτική σκέψη, η οποία προβλέπεται να καταλαμβάνει σημαντικό ποσοστό της εκπαιδευτικής διαδικασίας τα επόμενα χρόνια. Τέλος, οι εργαζόμενοι καλούνται να αξιοποιήσουν και να αναλύσουν ογκώδη δεδομένα, καθώς και να ενσωματώσουν την τεχνητή νοημοσύνη στην καθημερινή τους εργασία.

### 2.4 Συνέπειες του αυτοματισμού στην αγορά εργασίας

Η ευρεία ενσωμάτωση αλγοριθμικών συστημάτων και τεχνητής νοημοσύνης στις διαδικασίες πρόσληψης έχει επιφέρει σημαντικές και πολυδιάστατες συνέπειες στο ευρύτερο εργασιακό περιβάλλον. Λόγω των πολλαπλών πληροφοριών και των συνεχών τεχνολογικών εξελίξεων, το άτομο αδυνατεί να ακολουθήσει με τον ίδιο ρυθμό τις ανακατατάξεις στην αγορά εργασίας. Σύμφωνα με το World Economic Forum (2023), μόνο το 30% των εργαζομένων θα έχει πρόσβαση σε ευκαιρίες κατάρτισης μέχρι το 2027, γεγονός που αναμένεται να διευρύνει τα υφιστάμενα κενά δεξιοτήτων και να αυξήσει τις ανισότητες ως προς τις ευκαιρίες και τις θέσεις εργασίας.



Οι αυτοματοποιημένοι αλγόριθμοι που αφορούν τη διαδικασία πρόσληψης ενέχουν κινδύνους ως προς τη λήψη αποφάσεων. Σύμφωνα με τον Lane & Saint-Martin (2021), παρά την ικανότητα των συστημάτων αυτών να περιορίζουν ορισμένες γνωστικές προκαταλήψεις, συχνά αναπαράγουν ή και εντείνουν υποκείμενες ανισότητες που σχετίζονται με την εκπαίδευση και τις δεξιότητες των υποψηφίων. Επιπλέον, όπως αναφέρει η Eurofound (2018), η έλλειψη διαφάνειας στους αλγοριθμικούς μηχανισμούς αξιολόγησης μπορεί να μειώσει τον βαθμό εμπιστοσύνης των υποψηφίων, δημιουργώντας αβεβαιότητα σχετικά με την αξιοπιστία των μοντέλων και την κατάταξη των αιτούντων. Σε συνέχεια αυτών, όπως επισημαίνει η Cath (2018), οι αλγοριθμικές αποφάσεις εγείρουν σημαντικά ηθικά και νομικά ζητήματα, ιδίως ως προς τη λογοδοσία, τη δικαιοσύνη και τη δυνατότητα ερμηνείας των αποτελεσμάτων.

Άλλωστε, ολοένα και περισσότεροι υποψήφιοι δηλώνουν ότι αισθάνονται αδικημένοι ή αδυνατούν να κατανοήσουν τους λόγους για τους οποίους απορρίφθηκαν, καθώς πολλοί οργανισμοί απαντούν πλέον με αυτοματοποιημένα μηνύματα, δίχως επαρκή αξιολόγηση ή τεκμηρίωση.

Ωστόσο, οι συνέπειες έρχονται να επηρεάσουν και το επίπεδο της επαγγελματικής σταθερότητας, καθώς η αξιολόγηση της απόδοσης ενός ατόμου βασίζεται σε αυτοματοποιημένα συστήματα παρακολούθησης και ανάλυσης, μειώνοντας την ικανότητα για επικοινωνία και αλληλεπίδραση. Πλέον, η απόδοση δεν εκτιμάται από την προσωπική γνώμη και κρίση του προϊσταμένου, αλλά από δεδομένα και αυτοματοποιημένες διαδικασίες που συλλέγουν και αναλύουν σε πραγματικό χρόνο δείκτες παραγωγικότητας, συμπεριφορών και επαγγελματικών συνθηκών. Η επαγγελματική εξέλιξη και ανάδειξη εξαρτάται πλέον από την ικανότητα του εργαζομένου να χρησιμοποιεί τα νέα τεχνολογικά μέσα και να παραμένει ενήμερος σχετικά με τις όποιες εξελίξεις στον τεχνολογικό βίο και δευτερευόντως από τα χρόνια εμπειρίας ή κατάρτισής του στο αντικείμενό του. Έτσι, ένα υποκείμενο αίσθημα αβεβαιότητας ή ανασφάλειας που ενδέχεται να αναπτύξει ο εργαζόμενος, θα έχει σημαντικές επιπτώσεις, τόσο ως προς την απόδοσή του, όσο και στην προσωπικότητά του.

Αξίζει επίσης να σημειωθεί ότι ένας από τους πιο άμεσους κινδύνους αφορά την παράνομη ή μη εξουσιοδοτημένη πρόσβαση σε προσωπικά δεδομένα. Σύμφωνα με τον Γενικό Κανονισμό περί Προστασίας Δεδομένων (GDPR), κάθε δεδομένο προσωπικού χαρακτήρα πρέπει να επεξεργάζεται με τρόπο που να εξασφαλίζει την ασφάλεια, την ακεραιότητα και την εμπιστευτικότητά του. Παρόλα αυτά, η ραγδαία ανάπτυξη της τεχνητής νοημοσύνης και ιδίως τα εργαλεία της απαιτούν συχνά πολλαπλές πηγές δεδομένων και εξωτερικούς συνεργάτες, διαδικασίες που δεν είναι φανερές και ενδέχεται να επιφέρουν σοβαρό «πονοκέφαλο» ως προς την ασφάλεια και την προστασία των ατομικών δεδομένων. Όπως αναφέρουν οι Politou et al. (2022), η αυξημένη συγκέντρωση και επεξεργασία πληροφοριών ενισχύει τον κίνδυνο διαρροής προσωπικών στοιχείων και μπορεί να επηρεάσει αρνητικά την εμπιστοσύνη των εργαζομένων απέναντι στον οργανισμό.

Συνολικά, ενώ τα αυτοματοποιημένα συστήματα δύνανται να βελτιώσουν την αποτελεσματικότητα και να μειώσουν μεροληψίες σε ορισμένα στάδια της πρόσληψης, η εκτεταμένη χρήση, επεξεργασία και ανάλυση προσωπικών δεδομένων ενδέχεται να δημιουργήσει ένα σύνολο σοβαρών προκλήσεων για τις ίδιες τις εταιρίες. Προκειμένου να αντιμετωπιστούν αυτές οι συνέπειες, θα ήταν θεμιτή η αυξημένη διαφάνεια και οι τρόποι που επεξεργάζονται και αναλύονται τα δεδομένα, η συνεχής επίβλεψη των αλγορίθμων μηχανικής μάθησης και η σχετική οριοθέτησή τους από τους προγραμματιστές και περαιτέρω θέσπιση ισχυρών μέτρων προστασίας καθόλα τα στάδια πρόσληψης. Θα ήταν εξίσου απαραίτητη η διενέργεια συνεντεύξεων από ανθρώπινο δυναμικό της επιχείρησης και όχι από την μηχανή, η οποία φέρεται για τα καλά να έχει επέλθει στον ευρύτερο βίο.



Εικόνα 6: Κατά πόσο ηθικό είναι το AI;

## 2.5 Προσαρμοστικά Συστήματα

Ο όρος «προσαρμοστικό σύστημα» (adaptive system) αναφέρεται σε ένα σύνολο υπαρκτών ή αφηρημένων οντοτήτων που αλληλοεπιδρούν και αλληλεξαρτώνται, με σκοπό να προσαρμόζουν τη συμπεριφορά τους δυναμικά, ανταποκρινόμενες σε αλλαγές του περιβάλλοντος ή των ιδίων των χρηστών τους. Σύμφωνα με τον Brusilovsky (2001), τα προσαρμοστικά συστήματα χαρακτηρίζονται από τη δυνατότητα μάθησης, αναπροσαρμογής και λήψης αποφάσεων βάσει δεδομένων, γεγονός που τα καθιστά κατάλληλα για περιβάλλοντα υψηλής πολυπλοκότητας και μεταβλητότητας.

Στον τομέα των πληροφοριακών συστημάτων, τα προσαρμοστικά συστήματα στοχεύουν στην παροχή εξατομικευμένου περιεχομένου, υπηρεσιών και λειτουργιών, λαμβάνοντας υπόψη τα χαρακτηριστικά, τις δεξιότητες και τις προτιμήσεις του εκάστοτε χρήστη. Όπως αναφέρει ο Dix et al. (2004), η εξατομίκευση αυτή επιτρέπει στο σύστημα να ανταποκρίνεται αποτελεσματικότερα στις ανάγκες του χρήστη, βελτιώνοντας την εμπειρία χρήσης.

Η διαδραστικότητα αποτελεί βασικό τους γνώρισμα, αφού με βάση τη Shute (2007), η συνεχής αλληλεπίδραση χρήστη-συστήματος και οι μηχανισμοί ανατροφοδότησης αποτελούν κρίσιμα στοιχεία για τη βελτίωση της ποιότητας και της προσαρμοστικότητας των αποτελεσμάτων.

Παράλληλα, τα προσαρμοστικά συστήματα ενσωματώνουν μεθόδους αξιολόγησης τόσο σε πραγματικό χρόνο όσο και σε βάθος χρόνου. Σύμφωνα με τον Sommerville (2016), η διαδικασία συνεχούς αξιολόγησης συμβάλλει στη βελτιστοποίηση της αποδοτικότητας των πληροφοριακών συστημάτων, ενώ όπως σημειώνουν οι Hukkeri & Pol (2025), η αξιοποίηση δεδομένων και μεθόδων τεχνητής νοημοσύνης μπορεί να ενισχύσει σημαντικά την ακρίβεια και τη συνολική αποτελεσματικότητα των συστημάτων αξιολόγησης.

## 2.6 Προσαρμοστικά Συστήματα στις πλατφόρμες εύρεσης εργασίας

Στον τομέα της ψηφιακής απασχόλησης και των πλατφορμών εύρεσης εργασίας, η προσαρμοστικότητα αποτελεί κρίσιμο παράγοντα επιτυχίας, λόγω της συνεχούς μεταβολής των δεξιοτήτων της αγοράς, των απαιτήσεων των εργοδοτών και των επαγγελματικών προφίλ των υποψηφίων. Όπως επισημαίνει το OECD (2020), οι ανάγκες της αγοράς εργασίας μεταβάλλονται διαρκώς εξαιτίας της ψηφιοποίησης και των τεχνολογικών αλλαγών. Υπό αυτό το πλαίσιο, η ενσωμάτωση τεχνικών τεχνητής νοημοσύνης και μηχανικής μάθησης σε πλατφόρμες απασχόλησης, όπως το “DevJobs”, μπορεί να υποστηρίξει πιο αποτελεσματική, αυτοματοποιημένη και προσαρμοστική αντιστοίχιση υποψηφίων και θέσεων εργασίας.

Επιπρόσθετα, η ανάγκη για συνεχή αναπροσαρμογή των συστημάτων εντείνεται σε περιόδους ταχείας τεχνολογικής εξέλιξης ή οικονομικής αβεβαιότητας. Όπως αναφέρει το NIST (2018), η αποτελεσματικότητα και η αξιοπιστία ενός πληροφοριακού συστήματος εξαρτώνται σε μεγάλο βαθμό από την ικανότητά του να ανανεώνει δυναμικά τους μηχανισμούς αξιολόγησης και απόκρισης, ώστε να παραμένει λειτουργικό σε μεταβαλλόμενα περιβάλλοντα εργασίας.

Παράλληλα, ζητήματα όπως η προστασία προσωπικών δεδομένων, η κυβερνοασφάλεια και η ανθεκτικότητα απέναντι σε κακόβουλες επιθέσεις αναδεικνύουν την ανάγκη για συστήματα ικανά να προσαρμόζονται στα σύγχρονα επίπεδα απειλών. Όπως υπογραμμίζει το NIST (2018), η ανθεκτικότητα

ενός συστήματος καθορίζεται από την ικανότητά του να εξελίσσεται μετά από διαταραχές, ενώ η ENISA (2023) καταγράφει σημαντική αύξηση στον αριθμό και τη σοβαρότητα των κυβερνοεπιθέσεων, καθιστώντας αναγκαία την υιοθέτηση προσαρμοστικών μηχανισμών ασφάλειας και ανάκαμψης.

## 2.7 Ευφυή συστήματα αντιστοίχισης σε ψηφιακές πλατφόρμες απασχόλησης

Τα προσαρμοστικά και ευφυή πληροφοριακά συστήματα βασίζονται στη θεωρία της μοντελοποίησης χρήστη και στη δυναμική αναπαράσταση των χαρακτηριστικών του, με στόχο την εξατομικευμένη παροχή υπηρεσιών και τη βελτίωση της αλληλεπίδρασης ανθρώπου–υπολογιστή. Όπως αναφέρουν οι Alerpis και Virvou (2006), αλλά και οι Kabassi και Virvou (2006), η χρήση στερεοτύπων χρηστών και γνωστικών μοντέλων επιτρέπει στα συστήματα να προσαρμόζονται αποτελεσματικά σε ετερογενείς ομάδες χρηστών, με διαφορετικές δεξιότητες, στόχους και συμπεριφορές. Σε αντίστοιχο κλίμα, η έρευνα των Katsionis και Virvou (2004) και της Virvou (1999) υπογραμμίζει τη σημασία της συλλογιστικής και της κατανόησης των ενεργειών του χρήστη, ενώ οι Chrysafiadi, Papadimitriou και Virvou (2022) επισημαίνουν τον ρόλο της συναισθηματικής αλληλεπίδρασης στην περαιτέρω προσαρμοστικότητα των ευφυών συστημάτων.

Επιπλέον, η ενσωμάτωση τεχνικών ασαφούς λογικής και γνωστικών χαρτών έχει αναδειχθεί καθοριστική για την αποδοτική διαχείριση της αβεβαιότητας και της ελλιπούς πληροφορίας. Με βάση τα ευρήματα των Chrysafiadi και Virvou (2013a), η χρήση fuzzy reasoning επιτρέπει την ευέλικτη αναπροσαρμογή των συμπερασμάτων, ενώ οι μελέτες τους σε fuzzy cognitive maps (2013b, 2015) καταδεικνύουν πώς τα συστήματα μπορούν να ανταποκρίνονται σε πολύπλοκες και εξελισσόμενες συνθήκες χρήστη. Όπως τονίζουν, τέλος, οι Moundridou και Virvou (2001), η προσαρμογή αυτή ενισχύει την ποιότητα της εκπαιδευτικής εμπειρίας σε δυναμικά περιβάλλοντα μάθησης.

Η διαδραστικότητα και η συνεχής αξιολόγηση της συμπεριφοράς του χρήστη αποτελούν βασικά χαρακτηριστικά τέτοιων συστημάτων. Σύμφωνα με τους Papadimitriou και Virvou (2017), η συνεχής αλληλεπίδραση επιτρέπει την προσαρμογή του περιεχομένου σε πραγματικό χρόνο, ενώ η μελέτη των Savnoropoulos και Virvou (2010) αναδεικνύει τον ρόλο της ανάλυσης συμπεριφοράς στην προσαρμογή των αλγορίθμων. Αντίστοιχα, η εργασία των Alogogianni και Virvou (2021) επιβεβαιώνει ότι η συνεχής αξιολόγηση και η ανατροφοδότηση βελτιώνουν την απόδοση των υπολογιστικών συστημάτων.

Τέλος, όπως δείχνουν οι Panagoulis et al. (2023) και οι Panagoulis, Sotiropoulos και Tsihrintzis (2021), η επιτυχία των προσαρμοστικών συστημάτων εξαρτάται σε μεγάλο βαθμό από την ποιότητα του μοντέλου χρήστη και την ικανότητα του συστήματος να ερμηνεύει σωστά τις προτιμήσεις, τις ενέργειες και τα πρότυπα συμπεριφοράς του.

Στο πλαίσιο των ψηφιακών πλατφορμών απασχόλησης και επαγγελματικής δικτύωσης, οι παραπάνω αρχές αποκτούν ιδιαίτερη σημασία, καθώς η αποτελεσματική αντιστοίχιση υποψηφίων και θέσεων εργασίας αποτελεί ένα σύνθετο πρόβλημα λήψης αποφάσεων. Όπως αναφέρουν οι Virvou και συνεργάτες (2007), οι τεχνικές συσταδοποίησης αποτελούν βασικό εργαλείο για την εξαγωγή προφίλ χρηστών και την ομαδοποίηση δεξιοτήτων, ενώ η έρευνα των Panagoulis και συνεργατών (2024) αναδεικνύει τη σημασία της ανάλυσης πολλαπλών παραγόντων για τη βελτιστοποίηση των συστημάτων σύστασης. Παράλληλα, οι Chrysafiadi et al. (2023) επισημαίνουν ότι οι προηγμένες τεχνικές τεχνητής νοημοσύνης μπορούν να βελτιώσουν την αποτελεσματικότητα και την ακρίβεια σύνθετων πληροφοριακών συστημάτων.

Η ανάγκη για επεξηγησιμότητα και διαφάνεια καθίσταται κρίσιμη όταν τα συστήματα αυτά επηρεάζουν άμεσα επαγγελματικές επιλογές. Σύμφωνα με τη Virvou (2023), η κατανόηση των βασικών αρχών λειτουργίας των ευφυών συστημάτων αποτελεί κρίσιμο στοιχείο για μια θετική εμπειρία χρήσης και ενισχύει την εμπιστοσύνη των χρηστών απέναντι σε αλγοριθμικές διαδικασίες, ενώ η εργασία των

Mougiakou και Virnou (2017) τονίζει τη σημασία της συμμόρφωσης με τον GDPR και τα αντίστοιχα κανονιστικά πλαίσια, ώστε να ενισχύεται η εμπιστοσύνη των χρηστών στις ψηφιακές πλατφόρμες.

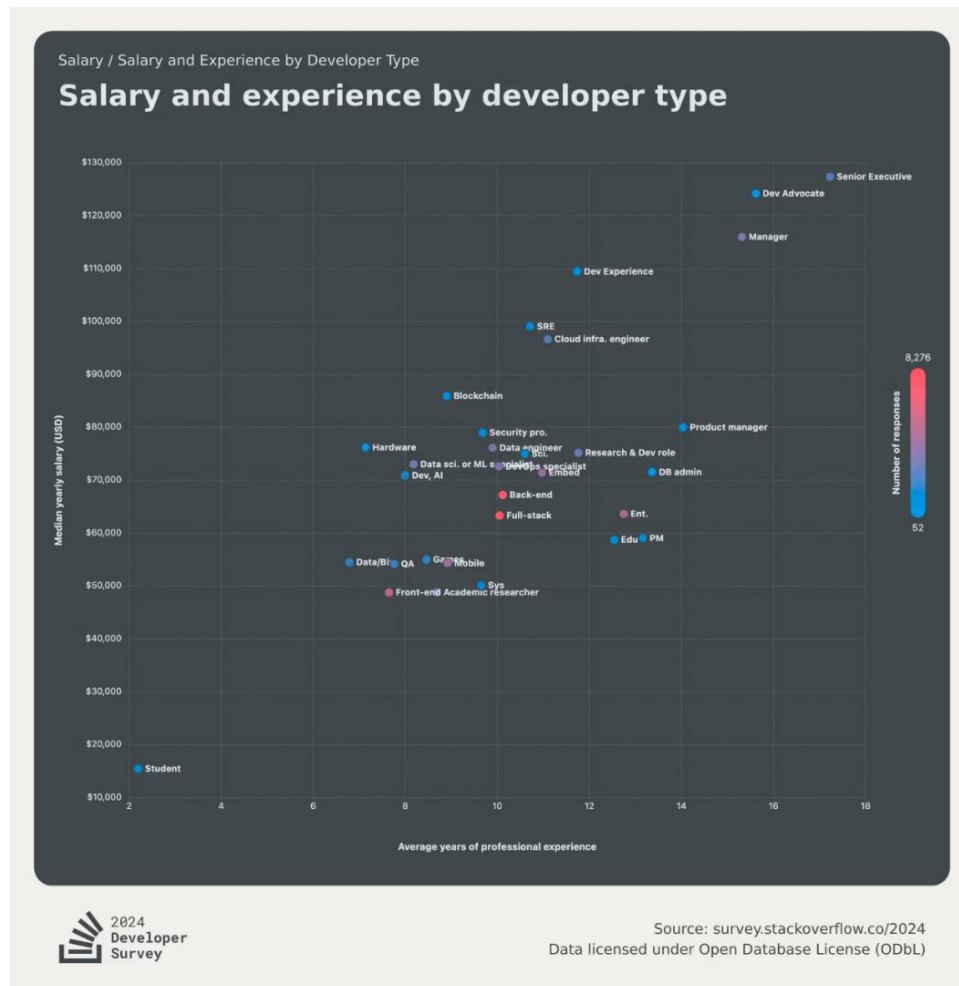
Με βάση τα παραπάνω, η ανάπτυξη πλατφορμών όπως το DevJobs στηρίζεται σε καθιερωμένες επιστημονικές προσεγγίσεις προσαρμοστικότητας και ευφυΐας. Οι προσεγγίσεις αυτές καθιστούν δυνατή τη βέλτιστη και τεκμηριωμένη αντιστοίχιση ανθρώπινου δυναμικού και ψηφιακών ευκαιριών, με τρόπο αξιόπιστο, διαφανή και ανθρωποκεντρικό.

## 2.8 Ανάλυση της Αγοράς Εργασίας για Προγραμματιστές

Η αγορά εργασίας στον τομέα της ανάπτυξης λογισμικού παρουσιάζει πολλές ιδιαιτερότητες και υψηλό βαθμό δυναμικότητας, καθώς οι τεχνολογικές εξελίξεις αλλάζουν συνεχώς τις απαιτήσεις των επιχειρήσεων και τις απαιτούμενες δεξιότητες. Σύμφωνα με το World Economic Forum (2023), οι δεξιότητες που σχετίζονται με την ανάπτυξη λογισμικού, τα δεδομένα και την αυτοματοποίηση είναι μεταξύ εκείνων που αναμένεται να αυξήσουν σημαντικά τη ζήτησή τους τα επόμενα χρόνια. Το γεγονός αυτό αναδεικνύει τον ιδιαίτερα ανταγωνιστικό χαρακτήρα του κλάδου, αλλά και την ανάγκη για συστήματα που μπορούν να υποστηρίξουν αποτελεσματικά την αντιστοίχιση υποψηφίων και θέσεων εργασίας.

Σύμφωνα με την έρευνα Stack Overflow Developer Survey (2024), περίπου 38 % των προγραμματιστών εργάζεται πλήρως απομακρυσμένα και 42 % σε υβριδικό περιβάλλον, με μόνο 20 % να εργάζεται αποκλειστικά στο γραφείο. Η τάση αυτή ευθυγραμμίζεται με τις ευρύτερες μεταβολές στο περιβάλλον της ψηφιακής εργασίας, καθώς οι εταιρείες στρέφονται πλέον σε διεθνείς αγορές και εύρεσης ταλέντων, αναζητώντας εξειδικευμένους επαγγελματίες ανεξαρτήτως γεωγραφικής θέσης. Όπως επισημαίνει το World Economic Forum (2023), η ζήτηση για προγραμματιστές μεσαίας και ανώτερης εμπειρίας αυξάνεται σταθερά, ιδιαίτερα σε τομείς όπως το cloud computing, η DevOps μηχανική και τα συστήματα τεχνητής νοημοσύνης, όπου οι επιχειρήσεις αντιμετωπίζουν σημαντικά κενά στελέχωσης.

Σημαντικές είναι και οι διαφοροποιήσεις στο επίπεδο των ζητούμενων τεχνικών δεξιοτήτων. Σύμφωνα με το World Economic Forum (2023), γλώσσες προγραμματισμού όπως η Python, η JavaScript και η Java παραμένουν σταθερά ανάμεσα στις πιο περιζήτητες δεξιότητες, ενώ ολοένα και μεγαλύτερη σημασία αποκτούν δεξιότητες που σχετίζονται με το cloud computing, τα σύγχρονα framework οικοσυστήματα και τις τεχνολογίες τεχνητής νοημοσύνης. Η σύνθετη αυτή εικόνα δημιουργεί προκλήσεις για εργαζόμενους και εργοδότες, καθώς η ορθή αποτύπωση και κατηγοριοποίηση δεξιοτήτων αποτελεί κρίσιμο παράγοντα επιτυχούς αντιστοίχισης.



Εικόνα 7: Μισθός και εμπειρία ανα τύπο προγραμματιστή (Stack Overflow Survey (2024))

Η αγορά ωστόσο δεν παρουσιάζει ομοιογένεια. Παρατηρούνται έντονες ανισορροπίες μεταξύ ειδικοτήτων, επιπέδων εμπειρίας και γεωγραφικών περιοχών. Με βάση τις προβλέψεις του Bureau of Labor Statistics (2024), οι θέσεις εργασίας για software developers αναμένεται να αυξηθούν κατά 15% την επόμενη δεκαετία, ποσοστό σημαντικά υψηλότερο από τον μέσο όρο όλων των επαγγελματιών. Ωστόσο, η αύξηση αυτή συνοδεύεται και από αυξημένο ανταγωνισμό, ιδιαίτερα για junior προγραμματιστές, οι οποίοι καλούνται να αποδείξουν πρακτικές δεξιότητες σε ένα ιδιαίτερα απαιτητικό περιβάλλον.

Σε αυτό το πλαίσιο, όπως επισημαίνουν οι Bogen και Rieke (2018), η αξιοποίηση αλγοριθμικών τεχνικών για την αντιστοίχιση υποψηφίων–θέσεων αποκτά ολοένα και μεγαλύτερη σημασία, καθώς οι αυτοματοποιημένες διαδικασίες παίζουν πλέον κεντρικό ρόλο στις σύγχρονες πρακτικές πρόσληψης. Σύμφωνα με τους Hukkeri & Pol (2025), τα συστήματα job matching που ενσωματώνουν μηχανική μάθηση και ανάλυση δεδομένων παρουσιάζουν αυξημένη ακρίβεια στην ταξινόμηση και σύσταση θέσεων, ιδίως σε περιβάλλοντα υψηλής τεχνολογικής πολυπλοκότητας. Αντίστοιχα, οι Ramanath et al. (2018) υπογραμμίζουν ότι η χρήση μοντέλων μάθησης βαθιάς αναπαράστασης επιτρέπει στις πλατφόρμες επαγγελματικής δικτύωσης να αναλύουν σύνθετα προφίλ δεξιοτήτων και να αναγνωρίζουν υποψήφιους που ταιριάζουν στις πραγματικές ανάγκες των επιχειρήσεων.

Τέλος, η βιβλιογραφία στον χώρο των προσαρμοστικών συστημάτων αναδεικνύει τη σημασία της ευφυούς μοντελοποίησης χρήστη σε τέτοιες εφαρμογές. Όπως επισημαίνουν οι Chrysafiadi & Vίνου (2015), τα συστήματα που προσαρμόζονται δυναμικά στα χαρακτηριστικά του εκάστοτε χρήστη

επιτυγχάνουν υψηλότερη αποδοτικότητα και μεγαλύτερη ακρίβεια στις παρεχόμενες προτάσεις. Οι αρχές αυτές βρίσκουν άμεση εφαρμογή στις σύγχρονες πλατφόρμες εύρεσης εργασίας για προγραμματιστές, οι οποίες καλούνται να διαχειριστούν μεγάλους όγκους δεδομένων, σύνθετες ανάγκες χρηστών και ταχύτατα μεταβαλλόμενα τεχνολογικά περιβάλλοντα.

Συνοψίζοντας, η αγορά εργασίας για προγραμματιστές αποτελεί ένα πολυδιάστατο και συνεχώς εξελισσόμενο οικοσύστημα, όπου η ανάλυση δεξιοτήτων, η αξιοποίηση δεδομένων και η υιοθέτηση ευφυών αλγοριθμικών μηχανισμών συνιστούν καθοριστικούς παράγοντες επιτυχίας. Η ανάπτυξη εξειδικευμένων πλατφορμών, όπως το DevJobs, εδράζεται ακριβώς σε αυτές τις ανάγκες, αξιοποιώντας τις σύγχρονες τεχνολογικές δυνατότητες για τη βέλτιστη αντιστοίχιση ανθρώπινου δυναμικού και ψηφιακών ευκαιριών.

## **2.9 Ανάλυση Ανταγωνιστικών Ψηφιακών Πλατφορμών Εύρεσης Εργασίας**

### **2.9.1 Περιορισμοί και προκλήσεις υφιστάμενων πλατφορμών**

Οι ψηφιακές πλατφόρμες εύρεσης εργασίας έχουν εξελιχθεί σε κρίσιμες υποδομές για τη διασύνδεση εργοδοτών και υποψηφίων, αξιοποιώντας τεχνικές αυτοματοποίησης, ανάλυσης δεδομένων και τεχνητής νοημοσύνης. Ωστόσο, παρά την εκτεταμένη τους χρήση, πολλές από τις υπάρχουσες λύσεις παρουσιάζουν περιορισμούς που επηρεάζουν την αποτελεσματικότητα της αντιστοίχισης και την εμπειρία των χρηστών.

Σημαντική πλατφόρμα επαγγελματικής δικτύωσης αποτελεί το LinkedIn, το οποίο συνδυάζει στοιχεία κοινωνικού δικτύου και συστήματος αναζήτησης εργασίας. Σύμφωνα με τους Ramanath et al. (2018), το LinkedIn έχει ενσωματώσει τεχνικές βαθιάς μάθησης (deep learning) ώστε να βελτιώνει τη συσχέτιση προφίλ και θέσεων, επεξεργαζόμενο εκατομμύρια δεδομένα δεξιοτήτων και αλληλεπιδράσεων χρηστών. Παρά τα πλεονεκτήματά του, η βιβλιογραφία υπογραμμίζει ότι η αποτελεσματικότητα του συστήματος μπορεί να περιορίζεται όταν οι χρήστες δεν προσφέρουν πλήρη ή ενημερωμένα προφίλ, γεγονός που μειώνει την ακρίβεια της αντιστοίχισης.

Αντίστοιχα, πλατφόρμες όπως το Indeed και το Glassdoor βασίζονται κυρίως σε μηχανές αναζήτησης και μοντέλα λέξεων-κλειδίων. Σύμφωνα με τη μελέτη των Martin και Rao (2022), τέτοια συστήματα παρουσιάζουν συχνά δυσκολίες στην κατανόηση της πραγματικής πρόθεσης του χρήστη και στη σύλληψη σύνθετων δεξιοτήτων, καθώς η αντιστοίχιση βασίζεται περισσότερο στη γλωσσική επιφάνεια παρά σε βαθύτερα σημασιολογικά χαρακτηριστικά. Αυτό οδηγεί σε χαμηλότερη ακρίβεια προτάσεων και σε πληθώρα άσχετων αναζητήσεων, ιδίως σε ειδικότητες υψηλής τεχνολογικής πολυπλοκότητας.

Αντίστοιχα, πλατφόρμες όπως το Indeed και το Glassdoor βασίζονται κυρίως σε μηχανές αναζήτησης και μοντέλα αντιστοίχισης λέξεων-κλειδίων. Σύμφωνα με τους Bogen και Rieke (2018), τέτοια συστήματα παρουσιάζουν συχνά δυσκολίες στην κατανόηση της πραγματικής πρόθεσης του χρήστη και στη σύλληψη σύνθετων δεξιοτήτων, καθώς η αντιστοίχιση βασίζεται επιφανειακά στη γλωσσική ομοιότητα και όχι σε ουσιαστικά σημασιολογικά χαρακτηριστικά. Αυτό οδηγεί σε μειωμένη ακρίβεια προτάσεων και αυξημένο όγκο άσχετων αποτελεσμάτων, ιδιαίτερα σε ειδικότητες με υψηλή τεχνολογική πολυπλοκότητα.

Ένα ακόμη σημαντικό ζήτημα αφορά τη διαφάνεια και την επεξηγησιμότητα των αλγορίθμων. Με βάση τη μελέτη της Cath (2018), η έλλειψη κατανόησης του τρόπου λειτουργίας των αλγοριθμικών συστημάτων μειώνει την εμπιστοσύνη των χρηστών και εγείρει ζητήματα δικαιοσύνης και λογοδοσίας. Αντίστοιχα, η Eurofound (2018) τονίζει ότι η περιορισμένη διαφάνεια στους αλγορίθμους αξιολόγησης μπορεί να δημιουργήσει αβεβαιότητα και χαμηλότερα επίπεδα αποδοχής από τους υποψηφίους.

Συνολικά, η διεθνής βιβλιογραφία δείχνει ότι, παρά την ευρεία διάδοση των ψηφιακών πλατφορμών εύρεσης εργασίας, εξακολουθούν να παρουσιάζουν σημαντικές αδυναμίες. Οι κυριότερες



αφορούν την περιορισμένη ακρίβεια της αντιστοίχισης, τις δυσκολίες στην ορθή ανάλυση και κατηγοριοποίηση δεξιοτήτων, τη συχνή έλλειψη διαφάνειας στους αλγορίθμους αξιολόγησης και, τέλος, την ανεπαρκή προστασία των προσωπικών δεδομένων των χρηστών. Οι προκλήσεις αυτές αναδεικνύουν την ανάγκη για πιο προηγμένα, αξιόπιστα και διαφανή συστήματα, ιδίως σε τομείς υψηλής τεχνολογικής εξειδίκευσης.

Η αντιμετώπιση των ελλείψεων αυτών αποτελεί βασικό πεδίο έρευνας και καινοτομίας. Μέσα σε αυτό το πλαίσιο, η ανάγκη για εξειδικευμένες πλατφόρμες που αξιοποιούν προηγμένες τεχνικές τεχνητής νοημοσύνης και μοντέλα προσαρμοστικότητας —όπως η πλατφόρμα DevJobs που παρουσιάζεται στα επόμενα κεφάλαια— καθίσταται ολοένα και πιο επιτακτική.

### **2.9.2 Λειτουργικοί μηχανισμοί επιλεγμένων πλατφορμών εύρεσης εργασίας**

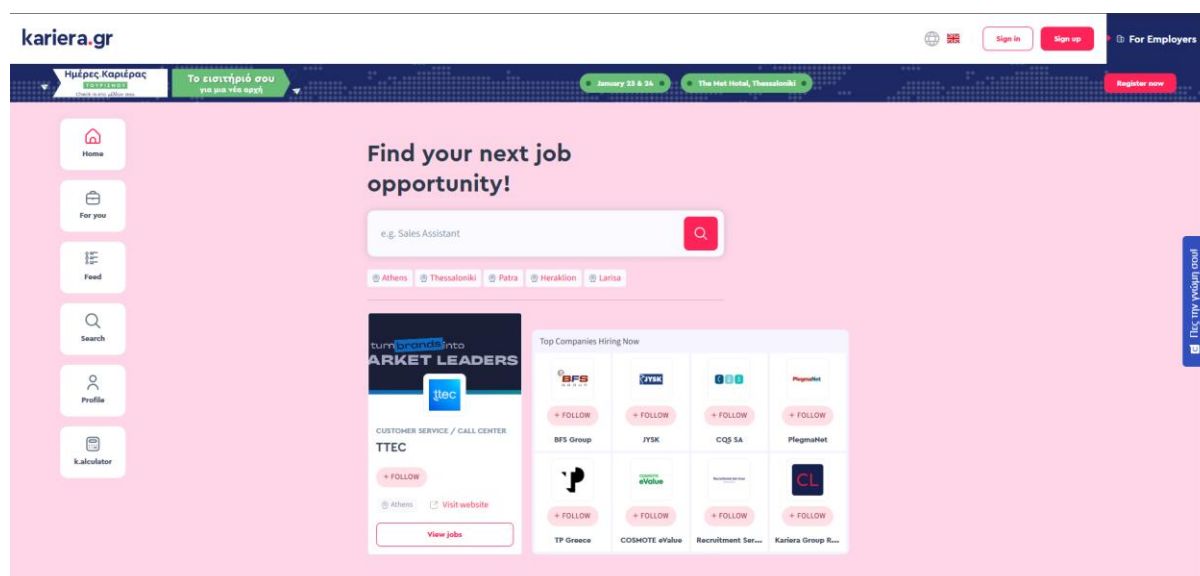
Οι κυριότερες ψηφιακές πλατφόρμες εύρεσης εργασίας ακολουθούν διαφορετικές προσεγγίσεις ως προς τον τρόπο συλλογής, ανάλυσης και αξιοποίησης των δεδομένων υποψηφίων και αγγελιών, γεγονός που επηρεάζει άμεσα την εμπειρία χρήστη και την αποτελεσματικότητα της αντιστοίχισης.

Το LinkedIn αποτελεί υβριδική πλατφόρμα επαγγελματικής δικτύωσης και εύρεσης εργασίας. Η λειτουργία του βασίζεται σε εκτενή προφίλ χρηστών, τα οποία περιλαμβάνουν δεξιότητες, εμπειρία, εκπαίδευση και κοινωνικές συνδέσεις. Οι μηχανισμοί προτάσεων εργασίας αξιοποιούν τόσο τεχνικές μηχανικής μάθησης όσο και δεδομένα κοινωνικού γραφήματος, λαμβάνοντας υπόψη τη δραστηριότητα του χρήστη και τις επαγγελματικές του συνδέσεις. Παρά την τεχνολογική του υπεροχή, η αποτελεσματικότητα του συστήματος επηρεάζεται έντονα από την πληρότητα του προφίλ, ενώ η πολυπλοκότητα των αλγορίθμων δυσχεραίνει την κατανόηση των κριτηρίων κατάταξης από τους χρήστες.

Το Indeed λειτουργεί κυρίως ως μηχανή αναζήτησης αγγελιών εργασίας, συλλέγοντας θέσεις από εταιρικές ιστοσελίδες και τρίτες πηγές. Η αντιστοίχιση βασίζεται σε μεγάλο βαθμό σε λέξεις-κλειδιά, φίλτρα και βασικά στατιστικά αλληλεπίδρασης, όπως οι προβολές και οι αιτήσεις. Η προσέγγιση αυτή επιτρέπει την ταχεία ανάκτηση μεγάλου όγκου αγγελιών, ωστόσο περιορίζει τη σημασιολογική κατανόηση των δεξιοτήτων, οδηγώντας συχνά σε πληθώρα χαμηλής συνάφειας αποτελεσμάτων.

Το Glassdoor διαφοροποιείται ενσωματώνοντας αξιολογήσεις εργαζομένων, πληροφορίες μισθών και δεδομένα για τη διαδικασία προσλήψεων, σε συνδυασμό με αγγελίες εργασίας. Αν και προσφέρει αυξημένη διαφάνεια ως προς τον εργοδότη, οι μηχανισμοί αναζήτησης και αντιστοίχισης παραμένουν σε μεγάλο βαθμό απλοί, βασισμένοι σε λέξεις-κλειδιά, χωρίς βαθιά ανάλυση του συνολικού προφίλ του υποψηφίου.

Η ελληνική πλατφόρμα kariera.gr αποτελεί μία από τις πλέον διαδεδομένες λύσεις εύρεσης εργασίας στην εγχώρια αγορά. Η λειτουργία της βασίζεται κυρίως στη δημοσίευση αγγελιών από εργοδότες και στην αναζήτησή τους μέσω φίλτρων, όπως κατηγορία θέσης, τοποθεσία και επίπεδο εμπειρίας. Η αντιστοίχιση πραγματοποιείται κατά κύριο λόγο μέσω λέξεων-κλειδίων και βασικών κριτηρίων φιλτραρίσματος, χωρίς την εκτεταμένη αξιοποίηση προηγμένων μηχανισμών ανάλυσης δεξιοτήτων ή εξατομικευμένων προτάσεων. Παρότι η πλατφόρμα καλύπτει αποτελεσματικά ένα ευρύ φάσμα επαγγελματικών κλάδων, η προσέγγισή της παρουσιάζει περιορισμούς σε ειδικότητες υψηλής τεχνολογικής εξειδίκευσης, όπου απαιτείται λεπτομερέστερη μοντελοποίηση δεξιοτήτων και εμπειρίας.



Εικόνα 8: Κεντρική σελίδα του kariera.gr, από τις δημοφιλέστερες σελίδες εύρεσης εργασίας στην Ελλάδα

Σε πιο εξειδικευμένες πλατφόρμες για προγραμματιστές, όπως το Stack Overflow Jobs και το Remote OK, η αντιστοίχιση επικεντρώνεται κυρίως στις τεχνολογίες και στα χαρακτηριστικά της εκάστοτε αγγελίας. Οι πλατφόρμες αυτές προσφέρουν στοχευμένο περιεχόμενο για τεχνικούς ρόλους, ωστόσο συχνά δεν λαμβάνουν υπόψη πιο σύνθετες παραμέτρους, όπως το επίπεδο εμπειρίας, τις προτιμήσεις εργασίας ή το συνολικό προφίλ δεξιοτήτων του υποψηφίου.

Συνολικά, οι παραπάνω πλατφόρμες αξιοποιούν διαφορετικούς μηχανισμούς αντιστοίχισης, οι οποίοι όμως παρουσιάζουν κοινά περιοριστικά χαρακτηριστικά: εξάρτηση από λέξεις-κλειδιά, περιορισμένη επεξηγησιμότητα και αδυναμία πλήρους αξιοποίησης σύνθετων δεξιοτήτων. Οι διαπιστώσεις αυτές ενισχύουν την ανάγκη για εξειδικευμένες πλατφόρμες, όπως η DevJobs, οι οποίες στοχεύουν σε πιο διαφανή και προσαρμοστική αντιστοίχιση σε περιβάλλοντα υψηλής τεχνολογικής εξειδίκευσης.

### 3 Ανάλυση και Σχεδιασμός

Στην ενότητα αυτή παρουσιάζονται οι διαφορετικοί ρόλοι χρηστών που αλληλοεπιδρούν με το σύστημα, οι λειτουργίες που θα έχουν στην διάθεσή τους, καθώς και οι περιορισμοί και οι τεχνικές απαιτήσεις για την εγκατάσταση και την χρήση της εφαρμογής. Συγκεκριμένα, αναλύονται οι διαβαθμίσεις πρόσβασης μεταξύ των χρηστών, οι επιτρεπόμενες ενέργειες και διαφοροποιήσεις ως προς την πλοήγηση υποψηφίου-εργοδότη, καθώς και οι ασφαλιστικές δικλίδες που διασφαλίζουν την ορθολογική λειτουργία της εφαρμογής.

#### 3.1 Ρόλοι χρηστών και λειτουργίες

Η εφαρμογή υποστηρίζει δύο κύριους ρόλους χρηστών, τον Υποψήφιο (εργαζόμενο) και τον Εργοδότη, οι οποίοι αλληλοεπιδρούν με ένα σύνολο λειτουργιών που συγκροτούν τον πυρήνα του συστήματος.

##### 3.1.1 Λειτουργίες Υποψηφίου

Αρχικά, ο υποψήφιος (candidate) έχει τη δυνατότητα εγγραφής, σύνδεσης και επιλογής ρόλου, διαδικασίες οι οποίες του επιτρέπουν να αποκτήσει πρόσβαση σε ένα πιο προσωποποιημένο και λειτουργικό περιβάλλον πλοήγησης. Με την είσοδό του στο σύστημα, ο υποψήφιος αποκτά πρόσβαση



σε ένα σύνολο βασικών λειτουργιών, οι οποίες συνθέτουν την κύρια εμπειρία χρήσης της πλατφόρμας DevJobs. Οι βασικές δυνατότητες που παρέχονται στον υποψήφιο περιλαμβάνουν:

- Δημιουργία και διαχείριση προσωπικού προφίλ, μέσω της οποίας ο χρήστης μπορεί να συμπληρώσει και να ενημερώσει τα προσωπικά και επαγγελματικά του στοιχεία, διαμορφώνοντας ένα ολοκληρωμένο προφίλ.
- Καταχώριση δεξιοτήτων και επαγγελματικών χαρακτηριστικών, όπου ο υποψήφιος δηλώνει τον τομέα εξειδίκευσής του, τις γλώσσες προγραμματισμού ή τεχνολογίες που γνωρίζει, καθώς και τα έτη επαγγελματικής του εμπειρίας. Τα δεδομένα αυτά αποτελούν βασική είσοδο για τον μηχανισμό αντιστοίχισης.
- Αναζήτηση διαθέσιμων αγγελιών εργασίας, οι οποίες έχουν αναρτηθεί από οργανισμούς και εργοδότες εγγεγραμμένους στην πλατφόρμα, με δυνατότητα φιλτραρίσματος βάσει κριτηρίων όπως τοποθεσία, τύπος απασχόλησης και επίπεδο εμπειρίας.
- Λήψη προσωποποιημένων προτάσεων εργασίας, οι οποίες παράγονται βάσει των δεδομένων του προφίλ και των προτιμήσεων που έχει ορίσει ο υποψήφιος στον λογαριασμό του, ενισχύοντας τη στοχευμένη αναζήτηση θέσεων.
- Υποβολή αιτήσεων για θέσεις εργασίας και συνεντεύξεις, επιτρέποντας στον υποψήφιο να εκδηλώνει επίσημα το ενδιαφέρον του για συγκεκριμένες αγγελίες μέσω της πλατφόρμας.
- Παρακολούθηση της κατάστασης των αιτήσεων, μέσω ειδικής λειτουργίας προβολής, όπου ο χρήστης μπορεί να ενημερώνεται για την εξέλιξη κάθε αίτησης (π.χ. σε αναμονή, αποδεκτή, απορριφθείσα).

Οι παραπάνω λειτουργίες αποτελούν τον πυρήνα της εμπειρίας χρήσης για τον υποψήφιο στην πλατφόρμα DevJobs, καθώς του παρέχουν τη δυνατότητα να αναζητά και να διεκδικεί θέσεις εργασίας που ευθυγραμμίζονται με το επαγγελματικό του προφίλ και τις τεχνολογικές του δεξιότητες.

### 3.1.2 Λειτουργίες Εργοδότη

Αντίστοιχα, ο εργοδότης (employer – 2nd actor) αλληλοεπιδρά με το σύστημα μέσω διαδικασιών που σχετίζονται κυρίως με τη δημιουργία, τη διαχείριση και την αξιολόγηση αγγελιών και υποψηφίων. Ο ρόλος του εργοδότη είναι καθοριστικός για τη λειτουργία της πλατφόρμας, καθώς αποτελεί την πηγή των διαθέσιμων θέσεων εργασίας και τον βασικό αξιολογητή των αιτήσεων που υποβάλλονται. Οι βασικές λειτουργίες που παρέχονται στον εργοδότη περιλαμβάνουν:

- Δημιουργία και διαχείριση εταιρικού προφίλ, στο οποίο καταχωρίζονται πληροφορίες που αφορούν την επωνυμία του οργανισμού, τη γεωγραφική του τοποθεσία, τον κλάδο δραστηριότητας και τα βασικά στοιχεία επικοινωνίας. Το προφίλ αυτό λειτουργεί ως σημείο αναφοράς για όλες τις αγγελίες που αναρτώνται.
- Δημιουργία νέων αγγελιών εργασίας, όπου ο εργοδότης μπορεί να καθορίσει τον τίτλο της θέσης, την περιγραφή, τον τύπο απασχόλησης, τις απαιτούμενες δεξιότητες και άλλα χαρακτηριστικά που περιγράφουν τη θέση εργασίας.
- Επεξεργασία και ενημέρωση υφιστάμενων αγγελιών, επιτρέποντας τη διόρθωση ή προσαρμογή των στοιχείων μιας θέσης σε περίπτωση αλλαγών στις ανάγκες του οργανισμού.
- Διαγραφή ή απενεργοποίηση αγγελιών, για θέσεις που δεν είναι πλέον ενεργές ή έχουν καλυφθεί, διατηρώντας έτσι το περιεχόμενο της πλατφόρμας επικαιροποιημένο.
- Προβολή και διαχείριση αιτήσεων υποψηφίων, όπου ο εργοδότης έχει πρόσβαση στο σύνολο των αιτήσεων που υποβάλλονται για κάθε αγγελία και μπορεί να εξετάζει τα προφίλ των υποψηφίων.

- Αξιολόγηση αιτήσεων και λήψη αποφάσεων, με δυνατότητα αποδοχής, απόρριψης ή μετάβασης σε επόμενο στάδιο της διαδικασίας πρόσληψης.
- Καθορισμός και διαχείριση συνεντεύξεων, μέσω της επιλογής ημερομηνίας, ώρας και τρόπου διεξαγωγής, υποστηρίζοντας οργανωμένα τη διαδικασία επικοινωνίας με τους υποψηφίους.

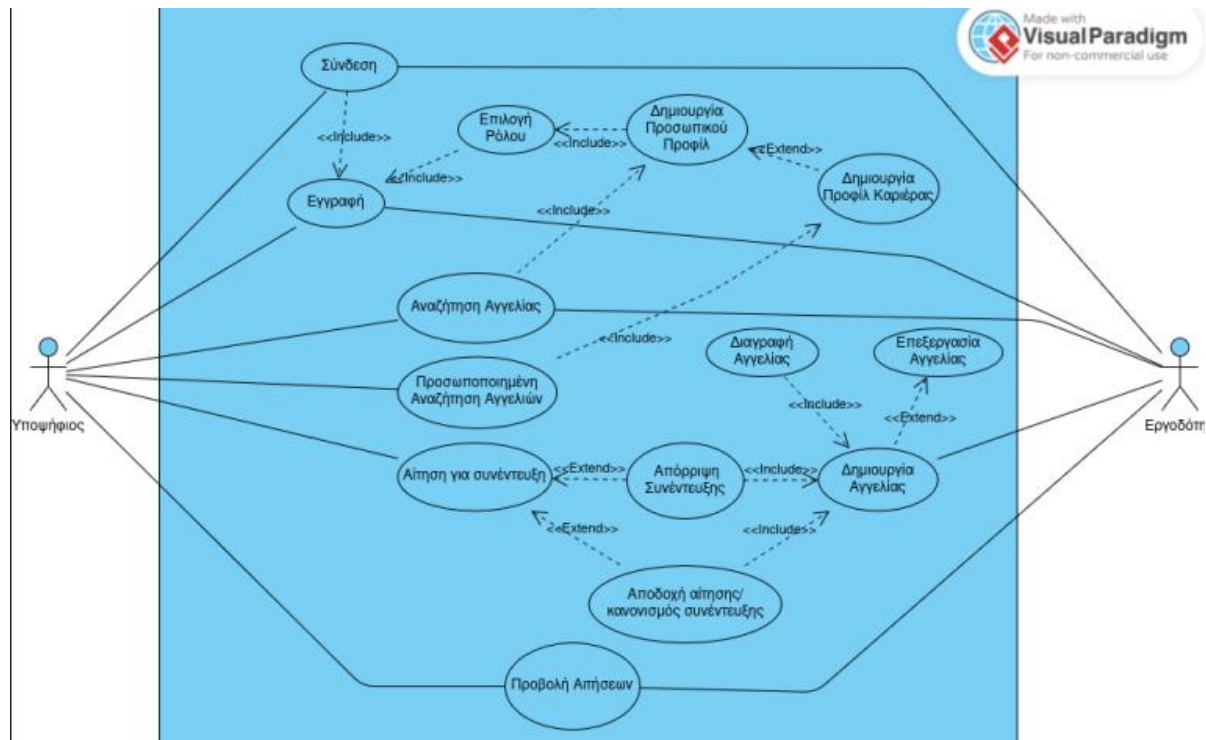
Οι παραπάνω λειτουργίες συνθέτουν έναν ολοκληρωμένο κύκλο επιλογής προσωπικού, ο οποίος υποστηρίζει όλη τη διαδικασία από την ανάρτηση της αγγελίας έως και την τελική αξιολόγηση των υποψηφίων.

Ο σαφής διαχωρισμός και ο καθορισμός των κύριων ρόλων της εφαρμογής συμβάλλουν καθοριστικά στην εύρυθμη λειτουργία της πλατφόρμας DevJobs και αποτελούν τη βάση για τον ορισμό των λειτουργικών απαιτήσεων και των μηχανισμών ασφάλειας που ακολουθούν.

### 3.1.3 Use Case Diagram

Το παρακάτω Use Case Diagram αποτυπώνει τις βασικές αλληλεπιδράσεις των δύο κύριων ρόλων της πλατφόρμας, δηλαδή του Υποψηφίου και του Εργοδότη, με το σύστημα DevJobs. Ο Υποψήφιος έχει πρόσβαση σε λειτουργίες που σχετίζονται με τη δημιουργία προφίλ, την αναζήτηση θέσεων εργασίας, την υποβολή αιτήσεων και την παρακολούθηση της προόδου τους. Αντίστοιχα, ο Εργοδότης μπορεί να δημιουργεί και να διαχειρίζεται αγγελίες, να ελέγχει τις υποβληθείσες αιτήσεις και να προγραμματίζει συνεντεύξεις.

Το διάγραμμα αναδεικνύει τον διαχωρισμό των ευθυνών και των δικαιωμάτων κάθε χρήστη, προσφέροντας μια συνοπτική εικόνα της λειτουργικής δομής της εφαρμογής. Αποτελεί, επομένως, θεμέλιο για την ορθή διαμόρφωση των λειτουργικών απαιτήσεων που παρουσιάζονται στις επόμενες ενότητες.

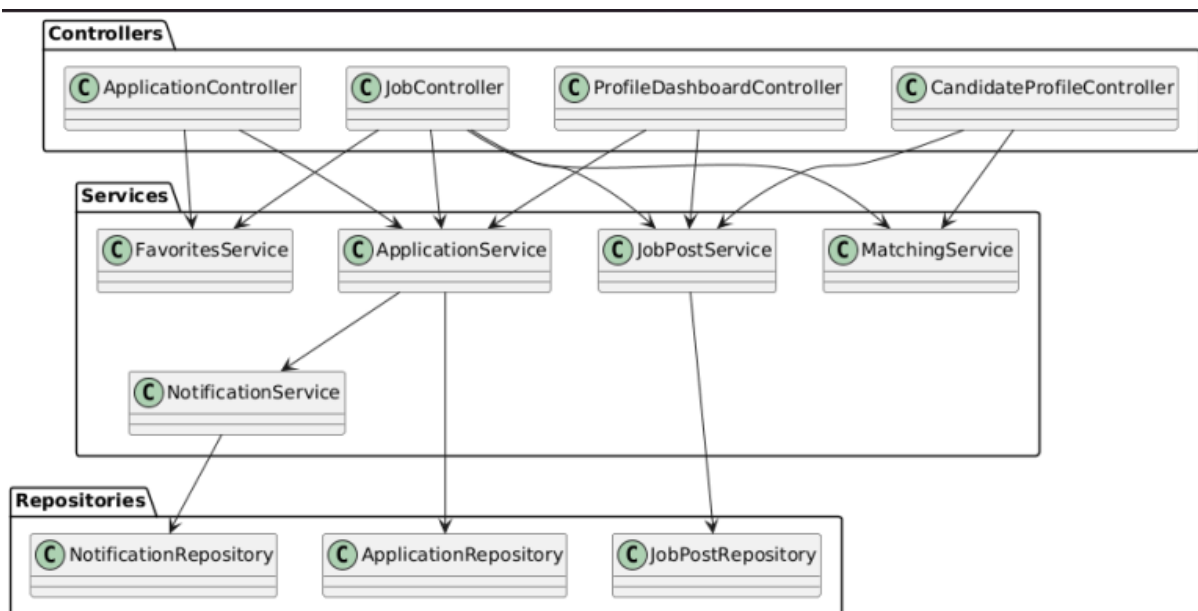


Εικόνα 9: Use Case Diagram (UML)

### 3.2 Συνολική αρχιτεκτονική Συστήματος

Για να παραχθούν οι παραπάνω λειτουργίες, βασίζονται σε μία πολυδιάστατη αρχιτεκτονική, η οποία οργανώνεται σε “Controllers”, “Services” και “Repositories”. Το επίπεδο των “Controllers” διαχειρίζεται τις αιτήσεις των χρηστών και λειτουργεί ως βασικό σημείο εισόδου, όπου κάθε ελεγκτής είναι υπεύθυνος για μία συγκεκριμένη περιοχή λειτουργιών, όπως το “JobController” για την προβολή και αναζήτηση θέσεων. Το ενδιάμεσο επίπεδο των Services συνιστά τον πυρήνα της επιχειρησιακής λογικής της εφαρμογής, καθώς υλοποιεί τις διαδικασίες αντιστοίχισης, διαχείρισης των διαφόρων αιτήσεων-αιτημάτων, λειτουργιών, ειδοποιήσεων και χειρισμού των αγαπημένων θέσεων. Τέλος, το επίπεδο “Repositories” αναλαμβάνει τη διασύνδεση με το σύστημα αποθήκευσης δεδομένων και την μόνιμη καταγραφή των συναλλαγών, διασφαλίζοντας συνέπεια και ακεραιότητα στην ροή πληροφοριών.

Το σύστημα της τριεπίπεδης αρχιτεκτονικής σε συνδυασμό με την αποτύπωση των λειτουργιών στο “Use Case”, αντικατοπτρίζει την ανάγκη ύπαρξης μίας ευέλικτης πλατφόρμας εύρεσης εργασίας, η οποία υποστηρίζει και εξυπηρετεί αποτελεσματικά και με προσωποποιημένο τρόπο τις ανάγκες της αγοράς εργασίας. Η σαφής οργάνωση και η διαφάνεια ως προς τον διαχωρισμό των λειτουργιών των χρηστών της εφαρμογής, συντελούν ως προς την μακροήμερευση και την συνεχή εξέλιξη των οργανισμών-ατόμων έναντι της ατέρμονης τεχνολογικής ανάπτυξης. Έτσι, η συγκεκριμένη εφαρμογή καλείται να φτιάξει ένα περιβάλλον ασφαλές, σύμφωνο με τα κριτήρια του εκάστοτε χρήστη και σεβόμενο την ανάγκη για αντιστοίχιση αυτών με τις ανάγκες του εργοδότη.



Εικόνα 10: Δομή πολυεπίπεδης αρχιτεκτονικής συστήματος

### 3.3 Λειτουργικές απαιτήσεις

Οι λειτουργικές απαιτήσεις περιγράφουν τις ενέργειες που πρέπει να υποστηρίζει το σύστημα ώστε να θεωρείται ολοκληρωμένο.

#### 3.3.1 Απαιτήσεις για Υποψήφιο

Ο Υποψήφιος (candidate) έχει κύριο σκοπό του ως χρήστης την εύρεση αγγελιών εργασίας που ταιριάζουν στις ανάγκες και τις δεξιότητές του. Για να καταστεί αυτό δυνατό ο υποψήφιος θα πρέπει:

- Να μπορεί να πραγματοποιεί εγγραφή/σύνδεση.
- Να δημιουργεί και να επεξεργάζεται το προφίλ του.
- Να προσθέτει δεξιότητες και στοιχεία εμπειρίας.

- Να αναζητά θέσεις εργασίας βάσει φίλτρων: ειδικότητα, γλώσσα προγραμματισμού, εμπειρία, τύπος εργασίας.
- Να λαμβάνει προτάσεις εργασίας προσαρμοσμένες στο προφίλ του.
- Να μπορεί να υποβάλλει αίτηση σε οποιαδήποτε αγγελία.
- Να παρακολουθεί τις αιτήσεις του.
- Να αποθηκεύει αγγελίες ως «αγαπημένες».

### 3.3.2 Απαιτήσεις για Εργοδότη

Ο στόχος του χρήστη με ρόλο εργοδότη έχει την αναζήτηση υποψηφίων που ταιριάζουν στις ανάγκες της εκάστοτε αγγελίας εργασίας. Έτσι, ο εργοδότης θα πρέπει να μπορεί:

- Να δημιουργεί εταιρικό προφίλ.
- Να αναρτά νέες αγγελίες.
- Να διαχειρίζεται υπάρχουσες αγγελίες (τροποποίηση/διαγραφή).
- Να βλέπει όλες τις αιτήσεις που έχουν υποβληθεί για κάθε θέση.
- Να εγκρίνει ή να απορρίπτει υποψηφίους.
- Να προγραμματίζει συνεντεύξεις.

### 3.3.3 Απαιτήσεις Διαχείρισης Συστήματος

Για την αποδοτική και έγκυρη λειτουργία της εφαρμογής, είναι απαραίτητο:

- Το σύστημα πρέπει να επαληθεύει τα διαπιστευτήρια των χρηστών.
- Πρέπει να υποστηρίζεται ασφαλής επικοινωνία με τη βάση δεδομένων.
- Πρέπει να πραγματοποιούνται έλεγχοι εγκυρότητας των εισαγόμενων δεδομένων.

## 3.4 Μη λειτουργικές απαιτήσεις

Οι μη λειτουργικές απαιτήσεις καθορίζουν τα ποιοτικά χαρακτηριστικά που πρέπει να πληροί η εφαρμογή, ώστε να είναι αξιόπιστη, ασφαλής και εύχρηστη. Δεν περιγράφουν συγκεκριμένες λειτουργίες, αλλά το πλαίσιο μέσα στο οποίο το σύστημα οφείλει να λειτουργεί.

### 3.4.1 Απόδοση (Performance)

Η απόδοση της εφαρμογής αποτελεί κρίσιμο μη λειτουργικό χαρακτηριστικό, καθώς επηρεάζει άμεσα την εμπειρία χρήσης και την αποδοχή της πλατφόρμας από τους τελικούς χρήστες. Η εφαρμογή DevJobs οφείλει να ανταποκρίνεται γρήγορα στις βασικές λειτουργίες της, όπως η αναζήτηση αγγελιών εργασίας, η προβολή προφίλ χρηστών και η υποβολή αιτήσεων, χωρίς αισθητές καθυστερήσεις.

Ιδιαίτερη έμφαση δίνεται στον χρόνο απόκρισης του συστήματος, ο οποίος πρέπει να παραμένει χαμηλός ακόμη και σε περιπτώσεις αυξημένου φόρτου. Η ορθή σχεδίαση των queries προς τη βάση δεδομένων, η χρήση κατάλληλων indexes και η αποφυγή περιττών κλήσεων συμβάλλουν καθοριστικά στη διατήρηση ικανοποιητικής απόδοσης. Παράλληλα, η επιχειρησιακή λογική στο επίπεδο των services έχει υλοποιηθεί με τρόπο που αποφεύγει σύνθετες και χρονοβόρες επεξεργασίες κατά την εξυπηρέτηση αιτημάτων.

Επιπλέον, η αξιοποίηση μηχανισμών caching σε κατάλληλα σημεία του συστήματος μπορεί να μειώσει σημαντικά τον αριθμό επαναλαμβανόμενων αιτημάτων προς τη βάση δεδομένων, βελτιώνοντας τον συνολικό χρόνο απόκρισης. Τέτοιες τεχνικές είναι ιδιαίτερα χρήσιμες σε λειτουργίες όπως η προβολή λιστών αγγελιών ή η ανάκτηση σταθερών δεδομένων που δεν μεταβάλλονται συχνά.

Συνολικά, η εφαρμογή έχει σχεδιαστεί με γνώμονα την ομαλή και απρόσκοπτη λειτουργία της, ώστε οι χρήστες να μπορούν να εκτελούν τις βασικές τους ενέργειες γρήγορα και αποτελεσματικά, χωρίς να επηρεάζεται αρνητικά η εμπειρία χρήσης ακόμη και σε περιβάλλοντα με αυξημένη δραστηριότητα.

### 3.4.2 Ασφάλεια (Security)

Η ασφάλεια της εφαρμογής αποτελεί βασική μη λειτουργική απαίτηση, καθώς η πλατφόρμα διαχειρίζεται προσωπικά και επαγγελματικά δεδομένα χρηστών, όπως στοιχεία ταυτοποίησης, προφίλ, αιτήσεις εργασίας και πληροφορίες επικοινωνίας. Η προστασία των δεδομένων αυτών είναι κρίσιμη τόσο για τη συμμόρφωση με βασικές αρχές προστασίας προσωπικών δεδομένων όσο και για την οικοδόμηση εμπιστοσύνης μεταξύ των χρηστών και του συστήματος.

Η πρόσβαση στις λειτουργίες της εφαρμογής πραγματοποιείται μέσω μηχανισμών authentication και authorization, με βάση τον ρόλο κάθε χρήστη. Το σύστημα υποστηρίζει διαφορετικούς ρόλους, όπως υποψήφιος, εργοδότης και διαχειριστής, και περιορίζει την πρόσβαση σε ευαίσθητες λειτουργίες μόνο στους κατάλληλους χρήστες. Με τον τρόπο αυτό αποτρέπεται η μη εξουσιοδοτημένη πρόσβαση σε δεδομένα ή ενέργειες που δεν αντιστοιχούν στον ρόλο του χρήστη.

Οι κωδικοί πρόσβασης δεν αποθηκεύονται σε απλή μορφή, αλλά υποβάλλονται σε διαδικασία κρυπτογράφησης με χρήση ασφαλών hash αλγορίθμων, ώστε να προστατεύονται ακόμη και σε περίπτωση μη εξουσιοδοτημένης πρόσβασης στη βάση δεδομένων. Η πρακτική αυτή αποτελεί θεμελιώδη αρχή ασφάλειας σε σύγχρονες διαδικτυακές εφαρμογές και μειώνει σημαντικά τον κίνδυνο διαρροής ευαίσθητων πληροφοριών.

Παράλληλα, η εφαρμογή έχει σχεδιαστεί ώστε να προστατεύεται από συνήθεις απειλές και ευπάθειες, όπως επιθέσεις τύπου SQL injection. Η χρήση μηχανισμών αντικειμενοστραφούς πρόσβασης στη βάση δεδομένων και παραμετροποιημένων queries συμβάλλει στην ασφαλή διαχείριση των δεδομένων και στη μείωση της πιθανότητας κακόβουλης εκμετάλλευσης.

Τέλος, η γενικότερη αρχιτεκτονική του συστήματος υποστηρίζει την ασφαλή διαχείριση συνεδριών χρηστών και τον έλεγχο πρόσβασης σε προστατευμένες σελίδες, διασφαλίζοντας ότι κάθε χρήστης μπορεί να αλληλεπιδρά μόνο με τα δεδομένα και τις λειτουργίες που του αναλογούν. Η προσέγγιση αυτή ενισχύει τη συνολική αξιοπιστία της πλατφόρμας και καθιστά την εφαρμογή κατάλληλη για χρήση σε πραγματικές συνθήκες.

### 3.4.3 Ευχρηστία (Usability)

Η ευχρηστία της εφαρμογής αποτελεί βασικό παράγοντα για την αποτελεσματική αξιοποίησή της από τους τελικούς χρήστες, καθώς επηρεάζει άμεσα την ευκολία πλοήγησης, την κατανόηση των λειτουργιών και τη συνολική εμπειρία χρήσης. Η εφαρμογή DevJobs έχει σχεδιαστεί με στόχο να είναι απλή, κατανοητή και φιλική προς τον χρήστη, ακόμη και για άτομα με περιορισμένη τεχνική εξοικείωση.

Η δομή της διεπαφής ακολουθεί σαφή και προβλέψιμη λογική πλοήγησης, επιτρέποντας στους χρήστες να εντοπίζουν εύκολα τις βασικές λειτουργίες, όπως η εγγραφή, η σύνδεση, η συμπλήρωση προφίλ και η αναζήτηση θέσεων εργασίας. Οι κύριες ενέργειες οργανώνονται σε διακριτές οθόνες και φόρμες, μειώνοντας τη γνωστική επιβάρυνση και περιορίζοντας τον αριθμό των βημάτων που απαιτούνται για την ολοκλήρωση κάθε διαδικασίας.

Ιδιαίτερη έμφαση έχει δοθεί στη σαφή παρουσίαση των πληροφοριών και στην παροχή άμεσης ανατροφοδότησης προς τον χρήστη. Μηνύματα επιβεβαίωσης, ειδοποιήσεις επιτυχούς ή αποτυχημένης ενέργειας και κατάλληλες ενδείξεις κατάστασης συμβάλλουν στη σωστή καθοδήγηση του χρήστη κατά την αλληλεπίδρασή του με το σύστημα.

Παράλληλα, η διεπαφή έχει σχεδιαστεί ώστε να είναι πλήρως responsive, προσαρμοζόμενη σε διαφορετικά μεγέθη οθόνης και συσκευές. Η υποστήριξη τόσο επιτραπέζιων υπολογιστών όσο και φορητών συσκευών επιτρέπει την απρόσκοπτη χρήση της εφαρμογής ανεξαρτήτως πλατφόρμας, ενισχύοντας την προσβασιμότητα και τη χρηστικότητα του συστήματος.

Συνολικά, η έμφαση στην ευχρηστία διασφαλίζει ότι οι βασικές λειτουργίες της εφαρμογής μπορούν να εκτελεστούν γρήγορα και αποτελεσματικά, βελτιώνοντας την εμπειρία χρήσης και ενισχύοντας τη θετική αποδοχή της πλατφόρμας από τους χρήστες.

### 3.4.4 Αξιοπιστία (Reliability)

Η αξιοπιστία της εφαρμογής αποτελεί κρίσιμη μη λειτουργική απαίτηση, καθώς σχετίζεται με τη σταθερή και προβλέψιμη λειτουργία του συστήματος σε καθημερινή χρήση. Η εφαρμογή DevJobs οφείλει να λειτουργεί απρόσκοπτα, αποφεύγοντας συχνά σφάλματα ή αιφνίδιες διακοπές που θα μπορούσαν να επηρεάσουν αρνητικά την εμπειρία των χρηστών και την αξιοπιστία της πλατφόρμας.

Ιδιαίτερη σημασία έχει η ορθή διαχείριση εξαιρέσεων και σφαλμάτων, ώστε σε περιπτώσεις αποτυχίας κάποιας λειτουργίας να παρέχονται κατανοητά και φιλικά μηνύματα ενημέρωσης προς τον χρήστη. Ως αποτέλεσμα, αποφεύγεται η εμφάνιση τεχνικών σφαλμάτων ή μη αναμενόμενης συμπεριφοράς στη διεπαφή, ενώ παράλληλα διατηρείται η συνοχή της εμπειρίας χρήσης.

Παράλληλα, κρίσιμες λειτουργίες του συστήματος, όπως η υποβολή αιτήσεων, η διαχείριση δεδομένων προφίλ και η ενημέρωση καταστάσεων, πρέπει να εκτελούνται με συνέπεια και ακρίβεια. Η διασφάλιση της ακεραιότητας των δεδομένων αποτελεί βασικό στοιχείο αξιοπιστίας, καθώς αποτρέπει απώλειες ή ασυνέπειες που θα μπορούσαν να επηρεάσουν τη λειτουργικότητα του συστήματος.

Η χρήση μηχανισμών καταγραφής συμβάντων (logging) σε επίπεδο backend συμβάλλει σημαντικά στον εντοπισμό και τη διάγνωση προβλημάτων. Μέσω της καταγραφής κρίσιμων ενεργειών και σφαλμάτων, καθίσταται δυνατή η παρακολούθηση της συμπεριφοράς της εφαρμογής και η έγκαιρη αντιμετώπιση πιθανών δυσλειτουργιών.

Συνολικά, η έμφαση στην αξιοπιστία διασφαλίζει ότι η εφαρμογή μπορεί να υποστηρίξει με συνέπεια τις βασικές της λειτουργίες, προσφέροντας ένα σταθερό και αξιόπιστο περιβάλλον χρήσης, κατάλληλο για την εξυπηρέτηση πραγματικών αναγκών εύρεσης εργασίας.

### 3.4.5 Επεκτασιμότητα (Scalability)

Η επεκτασιμότητα της εφαρμογής αποτελεί σημαντική μη λειτουργική απαίτηση, καθώς σχετίζεται με τη δυνατότητα του συστήματος να προσαρμόζεται σε μελλοντικές ανάγκες χωρίς να απαιτούνται εκτεταμένες αλλαγές στη βασική του δομή. Η εφαρμογή DevJobs έχει σχεδιαστεί με γνώμονα τη σταδιακή εξέλιξη και την υποστήριξη πρόσθετων λειτουργιών, καθώς η φύση των ψηφιακών πλατφορμών εύρεσης εργασίας απαιτεί συνεχή προσαρμογή στις ανάγκες της αγοράς.

Η αρχιτεκτονική MVC που υιοθετείται από το Spring Boot συμβάλλει καθοριστικά στην επεκτασιμότητα του συστήματος, καθώς επιτρέπει τον σαφή διαχωρισμό των επιπέδων παρουσίασης, επιχειρησιακής λογικής και πρόσβασης στα δεδομένα. Επομένως, η προσθήκη νέων λειτουργιών ή η τροποποίηση υφιστάμενων μπορεί να πραγματοποιηθεί χωρίς να επηρεάζεται ο πυρήνας της εφαρμογής.

Επιπλέον, η χρήση της βάσης δεδομένων PostgreSQL προσφέρει δυνατότητες κλιμάκωσης και βελτιστοποίησης της απόδοσης, όπως η δημιουργία indexes, η χρήση κατάλληλων μηχανισμών αναζήτησης και η διαχείριση μεγαλύτερων όγκων δεδομένων. Οι δυνατότητες αυτές καθιστούν τη βάση δεδομένων κατάλληλη για μελλοντική αύξηση του αριθμού χρηστών και των καταχωρήσεων αγγελιών.

Στο πλαίσιο της επεκτασιμότητας, το σύστημα μπορεί να υποστηρίξει:

- την προσθήκη νέων ρόλων χρηστών με διαφορετικά δικαιώματα και λειτουργίες,
- την ενσωμάτωση πιο προηγμένων ή «έξυπνων» αλγορίθμων αντιστοίχισης,



- την ανάπτυξη πρόσθετων υποσυστημάτων, όπως μηχανισμών αξιολόγησης, συστημάτων συνομιλίας ή στατιστικών αναφορών.

Η συνολική σχεδίαση της εφαρμογής επιτρέπει τη σταδιακή εξέλιξή της, χωρίς να απαιτείται επανασχεδιασμός της αρχιτεκτονικής. Η προσέγγιση αυτή καθιστά την DevJobs ένα ευέλικτο και επεκτάσιμο σύστημα, ικανό να προσαρμοστεί σε μελλοντικές τεχνολογικές και λειτουργικές απαιτήσεις.

### 3.4.6 Συντηρησιμότητα και Τεκμηρίωση (Maintainability)

Η συντηρησιμότητα της εφαρμογής αποτελεί σημαντική μη λειτουργική απαίτηση, καθώς καθορίζει τον βαθμό ευκολίας με τον οποίο το σύστημα μπορεί να τροποποιηθεί, να επεκταθεί ή να διορθωθεί στο μέλλον. Η εφαρμογή DevJobs έχει σχεδιαστεί με στόχο τη σαφή οργάνωση του κώδικα και τον διαχωρισμό των επιμέρους υποσυστημάτων, ώστε να διευκολύνεται η κατανόηση και η συντήρησή της από άλλους προγραμματιστές.

Ο διαχωρισμός μεταξύ frontend, backend και βάσης δεδομένων συμβάλλει καθοριστικά στη μείωση της πολυπλοκότητας του συστήματος. Κάθε τμήμα της εφαρμογής λειτουργεί αυτόνομα, επιτρέποντας την ανεξάρτητη τροποποίηση ή αντικατάστασή του χωρίς να επηρεάζεται η συνολική λειτουργία της πλατφόρμας. Παράλληλα, η χρήση της αρχιτεκτονικής MVC ενισχύει τη σαφή κατανομή ευθυνών μεταξύ των επιπέδων, καθιστώντας τον κώδικα πιο αναγνώσιμο και προβλέψιμο.

Η ύπαρξη σαφώς ορισμένων endpoints, αρχείων ρυθμίσεων και δομημένων πακέτων διευκολύνει την κατανόηση της ροής της εφαρμογής και μειώνει τον χρόνο εξοικείωσης νέων προγραμματιστών με το έργο. Επιπλέον, η βασική τεκμηρίωση των λειτουργιών και των ρυθμίσεων αποτελεί απαραίτητο στοιχείο για τη μακροχρόνια συντήρηση του συστήματος και τη μελλοντική του εξέλιξη.

Συνολικά, η έμφαση στη συντηρησιμότητα διασφαλίζει ότι η εφαρμογή μπορεί να υποστηριχθεί και να εξελιχθεί με οργανωμένο τρόπο, χωρίς να απαιτούνται εκτεταμένες επεμβάσεις ή επανασχεδιασμός του κώδικα.

### 3.4.7 Συμβατότητα (Compatibility)

Η συμβατότητα της εφαρμογής αποτελεί σημαντική μη λειτουργική απαίτηση, καθώς διασφαλίζει τη δυνατότητα χρήσης της πλατφόρμας από διαφορετικούς χρήστες και σε ποικίλα τεχνολογικά περιβάλλοντα. Η εφαρμογή DevJobs έχει σχεδιαστεί ώστε να λειτουργεί ομαλά στους σύγχρονους φυλλομετρητές, όπως Google Chrome, Mozilla Firefox και Microsoft Edge, χωρίς να απαιτούνται εξειδικευμένες ρυθμίσεις ή πρόσθετα λογισμικά.

Παράλληλα, η πλατφόρμα είναι συμβατή με τα κύρια λειτουργικά συστήματα, όπως Windows, Linux και macOS, επιτρέποντας την απρόσκοπτη πρόσβαση τόσο από επιτραπέζιους όσο και από φορητούς υπολογιστές. Η χρήση καθιερωμένων τεχνολογιών ιστού και προτύπων HTML και CSS συμβάλλει στη διατήρηση της διαλειτουργικότητας και στη σωστή απόδοση της διεπαφής σε διαφορετικά περιβάλλοντα.

Η προσέγγιση αυτή εξασφαλίζει ότι η εφαρμογή παραμένει ευέλικτη και προσβάσιμη, χωρίς να περιορίζεται σε συγκεκριμένες πλατφόρμες ή τεχνολογικές επιλογές, ενισχύοντας τη συνολική αξιοπιστία και χρηστικότητα του συστήματος.



### 3.5 Τεχνικές Προϋποθέσεις και εξαρτήσεις συστήματος

Προκειμένου νέοι χρήστες να χρησιμοποιήσουν και να επωφεληθούν από τα προνόμια που παρέχει η παρούσα εφαρμογή, απαιτείται αρχικά η πρόσβαση σε ορισμένα τεχνολογικά μέσα. Όπως είναι προφανές, το άτομο χρειάζεται έναν φορητό ή σταθερό υπολογιστή, με σύγχρονο λειτουργικό σύστημα, καθώς και έναν ενημερωμένο φυλλομετρητή (browser), ώστε να εξασφαλίζεται η πλήρης υποστήριξη των λειτουργιών του frontend.

Για προχωρημένους χρήστες ή προγραμματιστές που επιθυμούν την τοπική εγκατάσταση της εφαρμογής, καλούνται να ακολουθήσουν την εξής σειρά προτάσεων. Πρώτο βήμα, καθώς το σύστημα έχει αναπτυχθεί σε Spring Boot, απαιτείται η εγκατάσταση του Java Development Kit (JDK) έκδοσης 17 ή νεότερης, ώστε να εξασφαλιστεί η συμβατότητα με τα επιμέρους συστήματα της εφαρμογής. Σε δεύτερο χρόνο, αναγκαία κρίνεται η παρουσία του Maven ως εργαλείο διαχείρισης εξαρτήσεων, το οποίο αναλαμβάνει τόσο την εγκατάσταση των απαραίτητων βιβλιοθηκών όσο και τη διαδικασία μεταγλώττισης του έργου.

Ένα ακόμη περιβάλλον στο οποίο μπορείτε να επικολλήσετε τα αρχεία του έργου είναι το IntelliJ IDEA, το οποίο προσφέρει πλήρη υποστήριξη για Maven και Spring Boot έργα. Εκεί, θα αναγνωριστούν αυτόματα όλες οι εξαρτήσεις του συστήματος, οι ρυθμίσεις του Maven, καθώς και η δομή των πακέτων που συνιστούν την αρχιτεκτονική του συστήματος. Το περιβάλλον ανάπτυξης έχει επίσης άμεση επικοινωνία με τη βάση δεδομένων της εφαρμογής, η οποία έχει δημιουργηθεί σε PostgreSQL.

Η PostgreSQL λειτουργεί στην προεπιλεγμένη θύρα 8080, η οποία πρέπει να παραμένει ενεργή και προσβάσιμη από τον εξυπηρετητή της εφαρμογής. Στα αρχεία ρυθμίσεων που συμπεριλαμβάνονται στο αρχείο ZIP, είναι ορισμένα τα στοιχεία σύνδεσης, όπως η διεύθυνση, το όνομα της βάσης δεδομένων, το όνομα χρήστη και ο κωδικός πρόσβασης. Οι παράμετροι αυτοί είναι αναγκαίοι για την εκτέλεση των λειτουργιών της βάσης, καθώς μία ενδεχόμενη μη επίτευξη επικοινωνίας (backend–βάσης) μπορεί να ενέχει κινδύνους ως προς την απώλεια ή την αλλοίωση των δεδομένων.

Τέλος, ο προγραμματιστής πρέπει να είναι σίγουρος ότι η θύρα 8080 δεν χρησιμοποιείται από άλλη εφαρμογή, καθώς διαφορετικά δεν θα είναι δυνατή η εκτέλεση των λειτουργιών του συστήματος και η διεπαφή του με το frontend. Επιπλέον, προτείνεται να υπάρχει εγκατεστημένο το Node.js με τον διαχειριστή πακέτων npm, εφόσον το frontend απαιτεί μεταγλώττιση ή εκκίνηση μέσω σχετικών scripts.

### 3.6 Εκτέλεση της εφαρμογής σε τοπικό περιβάλλον

Για να εγκαταστήσετε την εφαρμογή, παρακαλώ ακολουθήστε τα παρακάτω βήματα

- Βεβαιωθείτε ότι το “JDK”, έκδοσης 17 ή νεότερης είναι εγκατεστημένο.
- Εγκατάσταση του “IntelliJ IDEA” (προαιρετικό περιβάλλον ανάπτυξης κώδικα).
- Επικόλληση του φακέλου του έργου στο “IntelliJ”. Εκεί, θα αναγνωριστεί αυτόματα το αρχείο “pom.xml” και θα εγκαταστήσει όλες τις απαραίτητες εξαρτήσεις μέσω “Maven”.
- Ορίστε κατάλληλα την βάση δεδομένων “PostgreSQL”. Ως θύρα του έργου παρακαλώ καταχωρίστε 8080 και βεβαιωθείτε ότι η βάση έχει δημιουργηθεί με το προβλεπόμενο όνομα.
- Στο αρχείο “application.properties” είναι ορισμένες όλες οι πληροφορίες της βάσης, για τις οποίες πρέπει να βεβαιωθείτε για την ορθότητά τους. Η τελευταία είναι αναγκαία για την σωστή επικοινωνία των “Repositories” με το “DBMS”.
- Βεβαιωθείτε ότι είναι εγκατεστημένο το “Node.js” και ο διαχειριστής πακέτων “npm”.
- Εκτέλεση της εντολής “Run” και έπειτα ο χρήστης μπορεί να επιλέξει τον αγαπημένο του “browser” και να κάνει την πλοήγησή του στο σύστημα, μέσω της διεύθυνσης: “http://localhost:8080/”.

## 4 Υλοποίηση και λειτουργικότητα εφαρμογής

### 4.1 Εισαγωγή αρχιτεκτονικής και τεχνολογιών

Η εφαρμογή DevJobs έχει αναπτυχθεί ως ένα πλήρως ολοκληρωμένο πληροφοριακό σύστημα εύρεσης εργασίας, αξιοποιώντας το πλαίσιο ανάπτυξης Spring Boot και τη γλώσσα προγραμματισμού Java. Ο σχεδιασμός της βασίζεται στη λογική της πολυεπίπεδης αρχιτεκτονικής, στην οποία διαχωρίζονται με σαφήνεια οι ευθύνες μεταξύ των διαφορετικών στρωμάτων της εφαρμογής. Η εγκατάσταση και διαχείριση των εξαρτήσεων πραγματοποιείται μέσω του εργαλείου Maven, ενώ η ανάπτυξη και εκτέλεση του έργου υποστηρίζονται από το περιβάλλον IntelliJ IDEA. Παράλληλα, το σύστημα χρησιμοποιεί τη βάση δεδομένων PostgreSQL ως αποθηκευτικό μηχανισμό, εξασφαλίζοντας σταθερή, αξιόπιστη και διαλειτουργική διαχείριση των δεδομένων.

Στην αρχιτεκτονική της εφαρμογής, ο ρόλος των επιμέρους επιπέδων είναι καθοριστικός για την ορθή λειτουργία, τη συντήρηση και την επεκτασιμότητα του συστήματος. Ο διαχωρισμός των ευθυνών υλοποιείται μέσω των ακόλουθων βασικών δομικών στοιχείων:

- **Controllers**, οι οποίοι διαχειρίζονται τα αιτήματα των χρηστών και αποτελούν το σημείο επαφής μεταξύ της διεπαφής χρήστη και της εσωτερικής λογικής της εφαρμογής. Σε αυτό το επίπεδο πραγματοποιείται η λήψη των δεδομένων εισόδου και η δρομολόγησή τους προς τα κατάλληλα services.
- **Services**, στα οποία υλοποιείται η κύρια επιχειρησιακή λογική του συστήματος. Στο επίπεδο αυτό αναπτύσσονται κρίσιμες λειτουργίες, όπως η διαχείριση αιτήσεων, η επεξεργασία προφίλ χρηστών, η αντιστοίχιση υποψηφίων και αγγελιών, καθώς και η δημιουργία και αποστολή ειδοποιήσεων.
- **Repositories**, τα οποία λειτουργούν ως γέφυρα μεταξύ της εφαρμογής και της βάσης δεδομένων. Μέσω του μηχανισμού JPA, τα repositories αναλαμβάνουν την ασφαλή ανάκτηση, αποθήκευση και ενημέρωση των δεδομένων, διασφαλίζοντας τη συνέπεια και την ακεραιότητα των πληροφοριών σε κάθε συναλλαγή.

Η οργάνωση της βάσης δεδομένων στηρίζεται σε οντότητες που αντικατοπτρίζουν την επιχειρησιακή λογική της εφαρμογής και επιτρέπουν τη συστηματική αποθήκευση και διαχείριση των δεδομένων. Ενδεικτικά, χρησιμοποιούνται οι ακόλουθες βασικές οντότητες:

- **User, UserProfile και CareerProfile**, οι οποίες αποθηκεύουν τα στοιχεία ταυτοποίησης και τα επαγγελματικά χαρακτηριστικά των χρηστών.
- **CompanyProfile και JobPost**, που μοντελοποιούν τους εργοδότες και τις αγγελίες εργασίας.
- **Application και InterviewDetails**, μέσω των οποίων υλοποιείται η διαδικασία υποβολής αιτήσεων και προγραμματισμού συνεντεύξεων.
- **JobSkill και UserSkill**, που επιτρέπουν τη σύνδεση δεξιοτήτων με υποψηφίους και αγγελίες.
- **Notification**, η οποία υποστηρίζει τον μηχανισμό ειδοποιήσεων της πλατφόρμας.

Η δημιουργία αρχικών δεδομένων πραγματοποιείται μέσω ειδικών μηχανισμών αρχικοποίησης (seeders), οι οποίοι εκτελούνται κατά την εκκίνηση της εφαρμογής και δημιουργούν βασικές εγγραφές, όπως ρόλους χρηστών, προκαθορισμένες δεξιότητες και βασικές ρυθμίσεις του συστήματος. Οι λεπτομέρειες της δομής και των σχέσεων της βάσης δεδομένων αναλύονται εκτενέστερα στις επόμενες ενότητες του παρόντος κεφαλαίου.

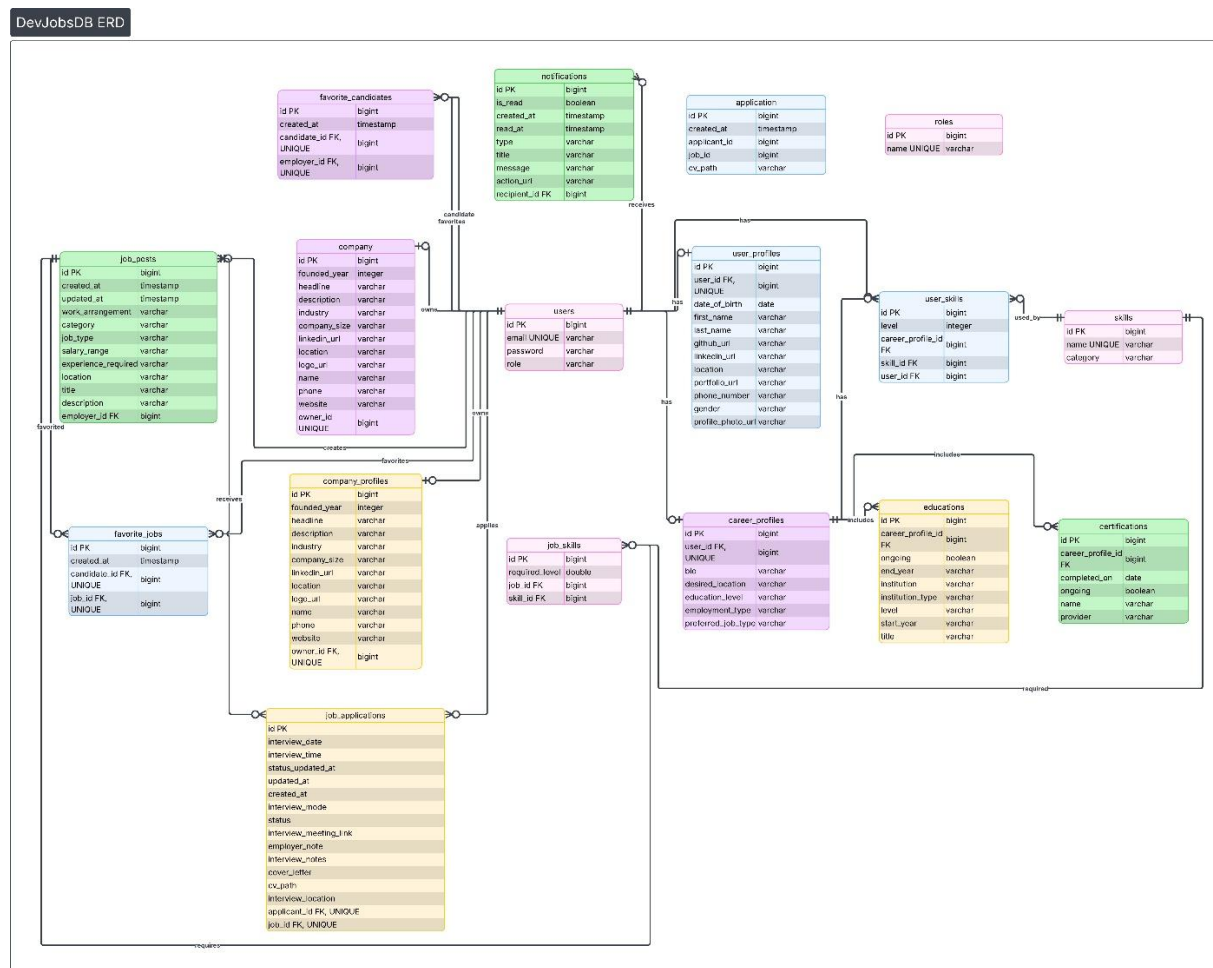
Η λειτουργικότητα της εφαρμογής δομείται γύρω από ένα σύνολο διεργασιών που συνδιαμορφώνουν τη συνολική εμπειρία χρήσης. Η διαδικασία εγγραφής και σύνδεσης, η οποία υλοποιείται μέσω των αντίστοιχων controllers αυθεντικοποίησης, επιτρέπει τη δημιουργία διαφορετικών

τύπων χρηστών και τον έλεγχο πρόσβασης στις διαθέσιμες λειτουργίες. Παράλληλα, μηχανισμοί ελέγχου πληρότητας προφίλ διασφαλίζουν ότι οι χρήστες έχουν συμπληρώσει τα απαραίτητα στοιχεία πριν αξιοποιήσουν τις κύριες λειτουργίες της πλατφόρμας.

Η διαχείριση αγγελιών εργασίας υλοποιείται μέσω εξειδικευμένων υπηρεσιών, οι οποίες υποστηρίζουν τη δημιουργία, την επεξεργασία και την οργάνωση των διαθέσιμων θέσεων. Η υποβολή αιτήσεων και ο προγραμματισμός συνεντεύξεων πραγματοποιούνται μέσω ενός ολοκληρωμένου μηχανισμού που βασίζεται στις οντότητες Application και InterviewDetails, εξασφαλίζοντας την πλήρη παρακολούθηση της πορείας κάθε αίτησης. Ιδιαίτερα σημαντική είναι η λειτουργία αντιστοίχισης, η οποία αξιοποιεί την υπηρεσία MatchingService και υπολογίζει τη συνάφεια μεταξύ υποψηφίων και αγγελιών βάσει δεξιοτήτων, τοποθεσίας και τύπου απασχόλησης.

Συνολικά, η εφαρμογή DevJobs συνιστά ένα καλά οργανωμένο και επεκτάσιμο πληροφοριακό σύστημα, το οποίο συνδυάζει σύγχρονες αρχιτεκτονικές πρακτικές με σαφή διαχωρισμό ευθυνών και ισχυρούς μηχανισμούς διαχείρισης δεδομένων. Η ενσωμάτωση λειτουργιών, όπως τα αγαπημένα, οι ειδοποιήσεις και η διαχείριση λιστών υποψηφίων, προσδίδει δυναμισμό στην πλατφόρμα, ενώ οι μηχανισμοί ασφαλείας διασφαλίζουν την αξιόπιστη και ελεγχόμενη χρήση της. Η δομή του κώδικα επιτρέπει την αυτόνομη ανάπτυξη κάθε υποσυστήματος, μειώνει τον κίνδυνο συγκρούσεων και διευκολύνει τη συντήρηση και τη μελλοντική επεκτασιμότητα του έργου.

## 4.2 Περιγραφή Βάσης Δεδομένων



Εικόνα 11: Διάγραμμα Οντοτήτων-Σχέσεων (Entity-Relationship Diagram) της βάσης δεδομένων.

Η βάση δεδομένων της εφαρμογής DevJobs αποτελεί βασικό δομικό στοιχείο του συστήματος, καθώς υποστηρίζει την αποθήκευση, την ανάκτηση και τη διαχείριση όλων των πληροφοριών που

απαιτούνται για τη λειτουργία της πλατφόρμας. Ο σχεδιασμός της βασίζεται σε αρχές κανονικοποίησης και σαφούς ορισμού σχέσεων, με στόχο τη μείωση του πλεονασμού δεδομένων και τη διασφάλιση της ακεραιότητας των πληροφοριών.

Στο παραπάνω διάγραμμα (Εικόνα 11: Διάγραμμα Οντοτήτων–Σχέσεων (Entity-Relationship Diagram) της βάσης δεδομένων) αποτυπώνονται οι βασικοί πίνακες της βάσης δεδομένων καθώς και οι μεταξύ τους συσχετίσεις, όπως προκύπτουν από το σχήμα της PostgreSQL. Το διάγραμμα αυτό προσφέρει μια συνολική εικόνα της δομής του συστήματος και διευκολύνει την κατανόηση της ροής και της οργάνωσης των δεδομένων.

#### 4.2.1 Πίνακες χρηστών και ρόλων

Κεντρικό σημείο της βάσης δεδομένων αποτελεί ο πίνακας **users**, στον οποίο αποθηκεύονται τα βασικά στοιχεία ταυτοποίησης κάθε χρήστη, όπως το μοναδικό αναγνωριστικό (id), το email και ο κρυπτογραφημένος κωδικός πρόσβασης. Ο πίνακας αυτός λειτουργεί ως αφετηρία για τη σύνδεση όλων των υπόλοιπων πληροφοριών που σχετίζονται με έναν λογαριασμό.

Ο πίνακας **roles** χρησιμοποιείται για τον ορισμό των διαθέσιμων ρόλων του συστήματος (π.χ. CANDIDATE, EMPLOYER, ADMIN), επιτρέποντας τον σαφή διαχωρισμό δικαιωμάτων και λειτουργιών. Η συσχέτιση χρήστη–ρόλου υλοποιείται μέσω ξένων κλειδιών, προσφέροντας ευελιξία στη διαχείριση της πρόσβασης και διευκολύνοντας τη μελλοντική επέκταση του συστήματος με νέους ρόλους.

Σημαντική σχεδιαστική επιλογή αποτελεί το γεγονός ότι τα προσωπικά στοιχεία των χρηστών δεν αποθηκεύονται απευθείας στον πίνακα **users**, αλλά κατανέμονται σε εξειδικευμένους πίνακες. Η προσέγγιση αυτή συμβάλλει στην καλύτερη οργάνωση των δεδομένων και στην αποφυγή περιττών πεδίων για διαφορετικούς τύπους χρηστών.

#### 4.2.2 Προφίλ υποψηφίου και εργοδότη

Ο πίνακας **user\_profiles** περιλαμβάνει γενικές πληροφορίες προφίλ, όπως όνομα, επώνυμο, ημερομηνία γέννησης και στοιχεία επικοινωνίας. Η σύνδεσή του με τον πίνακα **users** πραγματοποιείται μέσω του πεδίου **user\_id**, σχηματίζοντας σχέση ένα-προς-ένα, η οποία εξασφαλίζει ότι κάθε χρήστης διαθέτει ένα μοναδικό προφίλ.

Για τους υποψηφίους χρησιμοποιείται ο πίνακας **career\_profiles**, στον οποίο αποθηκεύονται εξειδικευμένα επαγγελματικά χαρακτηριστικά, όπως έτη εμπειρίας, επίπεδο εκπαίδευσης, επιθυμητός τύπος απασχόλησης και λοιπές προτιμήσεις. Η διάκριση αυτή επιτρέπει την ακριβέστερη μοντελοποίηση των αναγκών των υποψηφίων και υποστηρίζει αποτελεσματικά τον μηχανισμό αντιστοίχισης.

Αντίστοιχα, οι εργοδότες και οι εταιρείες μοντελοποιούνται μέσω των πινάκων **company** και **company\_profiles**, στους οποίους καταγράφονται πληροφορίες όπως η επωνυμία, ο κλάδος δραστηριότητας, η τοποθεσία και τα στοιχεία επικοινωνίας. Οι πίνακες αυτοί συνδέονται επίσης με τον **users**, διασφαλίζοντας ότι κάθε λογαριασμός αντιστοιχεί είτε σε υποψήφιο είτε σε εταιρεία, ανάλογα με τον ρόλο του.

#### 4.2.3 Πίνακες δεξιοτήτων

Οι δεξιότητες αποτελούν κρίσιμο στοιχείο για τη λειτουργία της πλατφόρμας και τον αλγόριθμο αντιστοίχισης. Ο πίνακας **skills** περιέχει το σύνολο των διαθέσιμων δεξιοτήτων, όπως γλώσσες προγραμματισμού, τεχνολογίες και frameworks, οι οποίες μπορούν να χρησιμοποιηθούν τόσο από υποψηφίους όσο και από εργοδότες.

Η σχέση μεταξύ χρηστών και δεξιοτήτων υλοποιείται μέσω του πίνακα `user_skills`, ο οποίος επιτρέπει τη σύνδεση πολλών χρηστών με πολλές δεξιότητες (σχέση πολλά-προς-πολλά). Ο πίνακας αυτός μπορεί να περιλαμβάνει επιπλέον πληροφορίες, όπως επίπεδο γνώσης ή έτη εμπειρίας, προσφέροντας πιο λεπτομερή αποτύπωση του προφίλ κάθε υποψηφίου.

Αντίστοιχα, οι απαιτούμενες δεξιότητες για κάθε αγγελία εργασίας καταγράφονται στον πίνακα `job_skills`, ο οποίος συνδέει τις αγγελίες με συγκεκριμένες δεξιότητες. Η δομή αυτή επιτρέπει την αποδοτική σύγκριση μεταξύ προσόντων υποψηφίων και απαιτήσεων θέσεων εργασίας.

#### 4.2.4 Αγγελίες και αιτήσεις εργασίας

Οι αγγελίες εργασίας αποθηκεύονται στον πίνακα `job_posts`, όπου καταγράφονται στοιχεία όπως ο τίτλος της θέσης, η περιγραφή, η τοποθεσία, ο τύπος απασχόλησης και οι ημερομηνίες δημιουργίας και ενημέρωσης. Κάθε αγγελία συνδέεται με μία εταιρεία μέσω ξένου κλειδιού προς τον πίνακα `company_profiles`, εξασφαλίζοντας σαφή συσχέτιση μεταξύ εργοδότη και θέσης εργασίας.

Η διαδικασία υποβολής αίτησης υλοποιείται μέσω του πίνακα `applications` ή `job_applications`, όπου κάθε εγγραφή συσχετίζει έναν υποψήφιο με μία αγγελία μέσω των ξένων κλειδιών `user_id` και `job_id`. Επιπλέον, καταγράφεται η κατάσταση της αίτησης (π.χ. `PENDING`, `ACCEPTED`, `REJECTED`), ο χρόνος υποβολής και τυχόν πρόσθετες πληροφορίες που αφορούν την αξιολόγηση.

Για τη διαχείριση της διαδικασίας συνεντεύξεων χρησιμοποιείται ο πίνακας `interview_details`, ο οποίος αποθηκεύει στοιχεία όπως ημερομηνία, ώρα και τρόπος διεξαγωγής. Η σύνδεσή του με τις αιτήσεις επιτρέπει την πλήρη παρακολούθηση της πορείας κάθε υποψηφίου.

#### 4.2.5 Ειδοποιήσεις, αγαπημένα και πρόσθετοι πίνακες

Ο μηχανισμός ειδοποιήσεων της εφαρμογής υλοποιείται μέσω του πίνακα `notifications`, ο οποίος συνδέεται με τους χρήστες και αποθηκεύει πληροφορίες όπως το περιεχόμενο του μηνύματος, την ημερομηνία δημιουργίας και την κατάσταση ανάγνωσης. Κατά συνέπεια, διασφαλίζεται η έγκαιρη ενημέρωση των χρηστών για κρίσιμες ενέργειες και αλλαγές.

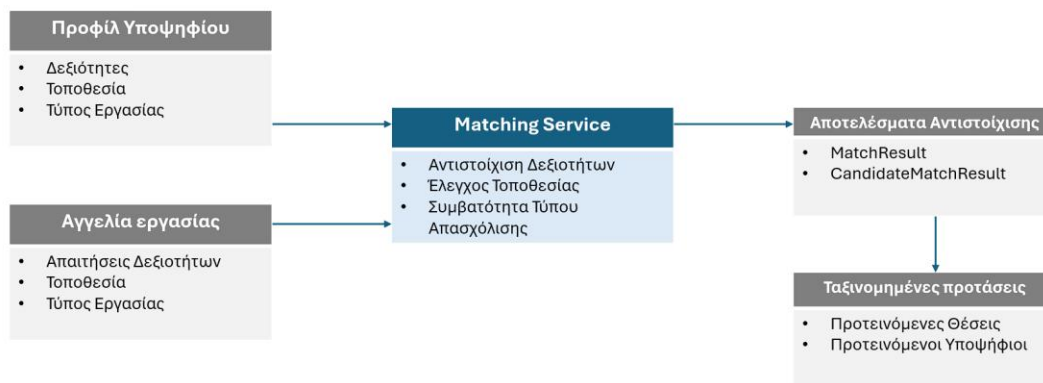
Η δυνατότητα αποθήκευσης αγαπημένων υλοποιείται μέσω των πινάκων `favorite_jobs` και `favorite_candidates`, οι οποίοι επιτρέπουν την ταχεία πρόσβαση σε επιλεγμένες αγγελίες ή υποψηφίους. Οι πίνακες αυτοί ενισχύουν την εμπειρία χρήσης χωρίς να επιβαρύνουν τη βασική επιχειρησιακή λογική του συστήματος.

Τέλος, συμπληρωματικοί πίνακες όπως `educations` και `certifications` επιτρέπουν τη λεπτομερέστερη αποτύπωση του προφίλ των υποψηφίων, καταγράφοντας σπουδές, πιστοποιήσεις και λοιπά προσόντα. Η ύπαρξη αυτών των πινάκων ενισχύει την ακρίβεια της αντιστοίχισης, καθώς προσφέρει στον αλγόριθμο περισσότερες δομημένες και αξιόπιστες πληροφορίες για κάθε χρήστη.

### 4.3 Λειτουργία Αντιστοίχισης Υποψηφίων και Αγγελιών

Η διαδικασία αντιστοίχισης αποτελεί μία από τις πιο κρίσιμες λειτουργίες της πλατφόρμας `DevJobs`, καθώς επηρεάζει άμεσα την ποιότητα των προτάσεων που λαμβάνουν οι χρήστες. Λόγω της σημασίας της λειτουργίας αυτής, ο αλγόριθμος θα αναλυθεί εκτενώς σε ξεχωριστή υποενότητα από τις άλλες λειτουργίες. Η λειτουργία αυτή υλοποιείται μέσω της υπηρεσίας `MatchingService`, η οποία είναι υπεύθυνη για την αξιολόγηση της συμβατότητας μεταξύ ενός επαγγελματικού προφίλ υποψηφίου και μίας αγγελίας εργασίας. Το αποτέλεσμα της διαδικασίας εκφράζεται ως ένας συνολικός βαθμός συνάφειας, ο οποίος χρησιμοποιείται για την ταξινόμηση και παρουσίαση των προτεινόμενων θέσεων ή υποψηφίων.

Σε εννοιολογικό επίπεδο, ο αλγόριθμος ακολουθεί μία πολυπαραγοντική προσέγγιση αξιολόγησης, σύμφωνα με την οποία η συνολική συνάφεια προκύπτει από τον συνδυασμό επιμέρους παραμέτρων που θεωρούνται καθοριστικές για την επιτυχή αντιστοίχιση. Οι παράμετροι αυτές αφορούν κυρίως τις δεξιότητες του υποψηφίου, την τοποθεσία και τον τύπο απασχόλησης, με τις δεξιότητες να αποτελούν το βασικότερο κριτήριο αξιολόγησης.



Εικόνα 12: Εννοιολογική απεικόνιση της ροής δεδομένων και των βασικών σταδίων του αλγορίθμου αντιστοίχισης στην πλατφόρμα DevJobs.

#### 4.3.1 Ανάλυση δεξιοτήτων και βαθμός κάλυψης απαιτήσεων

Ο πυρήνας του αλγορίθμου αντιστοίχισης βασίζεται στη σύγκριση των δεξιοτήτων που έχει δηλώσει ο υποψήφιος με τις δεξιότητες που απαιτεί η αγγελία. Οι δεξιότητες αναπαρίστανται ως σύνολα τιμών, τα οποία αντλούνται από το επαγγελματικό προφίλ του χρήστη και από τα χαρακτηριστικά της αγγελίας αντίστοιχα. Πριν από τη σύγκριση, τα δεδομένα κανονικοποιούνται, ώστε να αποφεύγονται ασυνέπειες που οφείλονται σε διαφορετική γραφή ή μορφοποίηση.

Η αξιολόγηση πραγματοποιείται με βάση το ποσοστό των απαιτούμενων δεξιοτήτων της αγγελίας που καλύπτονται από το προφίλ του υποψηφίου. Έτσι, ο αλγόριθμος δίνει έμφαση στην κάλυψη των αναγκών της θέσης εργασίας, χωρίς να τιμωρεί την ύπαρξη επιπλέον δεξιοτήτων που δεν αναφέρονται ρητά στην αγγελία. Σε περιπτώσεις όπου δεν έχουν δηλωθεί δεξιότητες στην αγγελία, η διαδικασία αντιστοίχισης περιορίζεται, καθώς δεν υπάρχουν επαρκή δεδομένα για ουσιαστική σύγκριση.

#### 4.3.2 Συμβατότητα τοποθεσίας και τρόπου εργασίας

Πέραν των δεξιοτήτων, ο αλγόριθμος λαμβάνει υπόψη τη συμβατότητα ως προς την τοποθεσία και τον τρόπο εργασίας. Για τον σκοπό αυτό, εξετάζονται τόσο οι προτιμήσεις του υποψηφίου (π.χ. επιθυμητή τοποθεσία ή προτίμηση για απομακρυσμένη εργασία) όσο και τα χαρακτηριστικά που δηλώνονται στην αγγελία. Σε περιπτώσεις όπου η θέση υποστηρίζει απομακρυσμένη εργασία, η συμβατότητα θεωρείται υψηλή όταν και ο υποψήφιος εκφράζει αντίστοιχη προτίμηση.



Όταν και οι δύο πλευρές δηλώνουν συγκεκριμένη τοποθεσία, η πλήρης ταύτιση οδηγεί σε υψηλό βαθμό συμβατότητας, ενώ η πλήρης ασυμφωνία μειώνει σημαντικά τη συνάφεια. Σε περιπτώσεις ελλιπών ή ασαφών δεδομένων, ο αλγόριθμος υιοθετεί ουδέτερη στάση, αποφεύγοντας αυστηρές απορρίψεις που θα μπορούσαν να αποκλείσουν δυνητικά κατάλληλες αντιστοιχίσεις.

#### **4.3.3 Συμβατότητα τύπου απασχόλησης**

Ένας επιπλέον παράγοντας που λαμβάνεται υπόψη είναι ο τύπος απασχόλησης, όπως αυτός δηλώνεται από τον υποψήφιο και την αγγελία. Ο αλγόριθμος εξετάζει αν υπάρχει συμφωνία μεταξύ των δύο πλευρών ως προς το είδος της εργασίας, όπως πλήρης ή μερική απασχόληση ή απομακρυσμένη εργασία. Η συμβατότητα σε αυτό το επίπεδο λειτουργεί συμπληρωματικά προς τις δεξιότητες και την τοποθεσία, ενισχύοντας ή αποδυναμώνοντας τη συνολική αξιολόγηση.

#### **4.3.4 Συνδυασμός επιμέρους κριτηρίων και τελικό αποτέλεσμα**

Τα επιμέρους αποτελέσματα των παραπάνω κριτηρίων συνδυάζονται σε έναν ενιαίο βαθμό συνάφειας, ο οποίος αντικατοπτρίζει τη συνολική καταλληλότητα του υποψηφίου για τη συγκεκριμένη θέση εργασίας. Στον συνδυασμό αυτό αποδίδεται μεγαλύτερη βαρύτητα στις δεξιότητες, καθώς αυτές αποτελούν τον βασικότερο δείκτη επαγγελματικής επάρκειας, ενώ η τοποθεσία και ο τύπος απασχόλησης λειτουργούν ως υποστηρικτικοί παράγοντες.

Ο τελικός βαθμός συνάφειας χρησιμοποιείται για την κατάταξη των αποτελεσμάτων που εμφανίζονται στον χρήστη, επιτρέποντας την παρουσίαση των πλέον σχετικών αγγελιών ή υποψηφίων στην κορυφή της λίστας. Τα αποτελέσματα της διαδικασίας επιστρέφονται σε κατάλληλες δομές δεδομένων, όπως MatchResult και CandidateMatchResult, οι οποίες επιτρέπουν την περαιτέρω αξιοποίηση των δεδομένων από τα υπόλοιπα υποσυστήματα της εφαρμογής.

#### **4.3.5 Πλεονεκτήματα της προσέγγισης**

Η υιοθετούμενη προσέγγιση αντιστοίχισης παρουσιάζει σημαντικά πλεονεκτήματα. Αρχικά, προσφέρει διαφάνεια και ερμηνευσιμότητα, καθώς τα κριτήρια που επηρεάζουν το αποτέλεσμα είναι σαφώς καθορισμένα και μπορούν να εξηγηθούν στον τελικό χρήστη. Επιπλέον, η δομή του αλγορίθμου επιτρέπει ευελιξία και επεκτασιμότητα, δεδομένου ότι νέα κριτήρια μπορούν να ενσωματωθούν χωρίς ριζική αναδιάρθρωση της υλοποίησης. Τέλος, η απλότητα της προσέγγισης διασφαλίζει αξιόπιστη απόδοση σε πραγματικό χρόνο, γεγονός που είναι κρίσιμο για εφαρμογές με αυξημένο αριθμό χρηστών και αιτημάτων.



```

package gr.p21080.devjobs.service;

import ...

@Service 6 usages 2 Marios K
public class MatchingService {

    public double calculateMatch(CareerProfile careerProfile, JobPost jobPost) { 4 usages 2 Marios K
        if (careerProfile == null || jobPost == null) {
            return 0.0;
        }

        double skillScore = calculateSkillMatch(careerProfile, jobPost);
        double locationScore = calculateLocationMatch(careerProfile, jobPost);
        double typeScore = calculateTypeMatch(careerProfile, jobPost);

        double weighted = (0.7 * skillScore) + (0.2 * locationScore) + (0.1 * typeScore);
        double scaled = 100 * weighted;
        return Math.max(0.0, Math.min(100.0, scaled));
    }

    private double calculateSkillMatch(CareerProfile careerProfile, JobPost jobPost) { 1 usage 2 Marios K
        Set<String> candidateSkills = safelyCollectSkillNames(careerProfile);
        Set<String> jobSkills = safelyCollectJobSkillNames(jobPost);

        if (jobSkills.isEmpty()) {
            return 0.0;
        }

        long matched = jobSkills.stream().filter(candidateSkills::contains).count();
    }
}

```

Εικόνα 13: Κλάση που υλοποιεί τον αλγόριθμο αντιστοίχισης μεταξύ επαγγελματικού προφίλ και αγγελίας εργασίας

```

package gr.p21080.devjobs.model;

import gr.p21080.devjobs.entity.JobPost;

public class MatchResult { 5 usages 2 Marios K

    private final JobPost job; 2 usages
    private final double score; 2 usages

    public MatchResult(JobPost job, double score) { 2 usages 2 Marios K
        this.job = job;
        this.score = score;
    }

    public JobPost getJob() { return job; }

    public double getScore() { return score; }
}

```

Εικόνα 14: Κλάση υπεύθυνη για την αποθήκευση του αποτελέσματος με τον αντίστοιχο δείκτη συνάφειας

```

package gr.p21080.devjobs.model;

import ...

public class CandidateMatchResult { 9 usages 2 Marios K

    private final User candidate; 2 usages
    private final CareerProfile careerProfile; 2 usages
    private final double score; 2 usages
    private final List<String> topSkills; 2 usages

    public CandidateMatchResult(User candidate, CareerProfile careerProfile, double score) { 2 usages 2 Marios K
        this.candidate = candidate;
        this.careerProfile = careerProfile;
        this.score = score;
        this.topSkills = extractTopSkills(careerProfile);
    }

    public User getCandidate() { return candidate; }

    public CareerProfile getCareerProfile() { return careerProfile; }

    public double getScore() { return score; }

    public List<String> getTopSkills() { return topSkills; }

    private List<String> extractTopSkills(CareerProfile profile) { 1 usage 2 Marios K
        if (profile == null || profile.getSkills() == null || profile.getSkills().isEmpty()) {
            return List.of();
        }
    }
}

```

Εικόνα 15: Κλάση που αποδίδει τα αποτελέσματα της αντιστοίχισης, περιλαμβάνοντας την συνολική βαθμολογία και τις κύριες δεξιότητες

## 4.4 Περιγραφή υλοποίησης διαφόρων εργασιών της εφαρμογής

Αυτή η ενότητα θα παρουσιάσει βήμα προς βήμα όλες τις υλοποιήσεις της εφαρμογής και την οπτικοποίησή τους στο “backend” του συστήματος. Θα ακολουθήσουν επίσης παραδείγματα κώδικα από το εκάστοτε σημείο που θα αναπτυχθεί και θα συζητηθεί.

### 4.4.1 Εγγραφή χρηστών

Η διαδικασία εγγραφής αποτελεί το πρώτο και θεμελιώδες βήμα ως προς την αλληλεπίδραση των χρηστών με την εφαρμογή, καθώς μέσω αυτής πραγματοποιείται η αρχική δημιουργία λογαριασμού και η ένταξή τους στο σύστημα. Ο υποψήφιος χρήστης, είτε πρόκειται για εργοδότη είτε για εργαζόμενο, εισέρχεται στη σελίδα εγγραφής της πλατφόρμας, όπου καλείται να συμπληρώσει βασικά στοιχεία ταυτοποίησης, τα οποία είναι απαραίτητα για την ορθή και ασφαλή λειτουργία του συστήματος.

Τα στοιχεία αυτά περιλαμβάνουν, ενδεικτικά, τη διεύθυνση ηλεκτρονικού ταχυδρομείου, τον κωδικό πρόσβασης, καθώς και την επιλογή του ρόλου με τον οποίο ο χρήστης επιθυμεί να εγγραφεί στην εφαρμογή. Η επιλογή ρόλου είναι ιδιαίτερα σημαντική, καθώς καθορίζει τα δικαιώματα πρόσβασης και τις διαθέσιμες λειτουργίες που θα έχει ο χρήστης κατά τη μετέπειτα χρήση της πλατφόρμας.

Η λειτουργία της εγγραφής υλοποιείται μέσω του “RegistrationController”, ο οποίος είναι υπεύθυνος για τη λήψη των δεδομένων από το περιβάλλον χρήστη, τον αρχικό έλεγχο εγκυρότητας και τη διαβίβασή τους στο αντίστοιχο επίπεδο επιχειρησιακής λογικής. Στο πλαίσιο αυτό, πραγματοποιούνται βασικοί έλεγχοι, όπως η ορθότητα της μορφής του email και η πληρότητα των υποχρεωτικών πεδίων, ώστε να αποφεύγεται η εισαγωγή εσφαλμένων ή ελλιπών δεδομένων.

Στη συνέχεια, τα δεδομένα προωθούνται στο “UserService”, όπου εκτελείται η κύρια επιχειρησιακή λογική της εγγραφής. Σε αυτό το στάδιο πραγματοποιείται η κρυπτογράφηση του κωδικού πρόσβασης, διασφαλίζοντας ότι ευαίσθητες πληροφορίες δεν αποθηκεύονται σε απλή μορφή στη βάση δεδομένων. Παράλληλα, δημιουργούνται οι αντίστοιχες εγγραφές στις οντότητες “User” και

“UserProfile”, οι οποίες αποθηκεύουν τόσο τα στοιχεία ταυτοποίησης όσο και τα βασικά δεδομένα προφίλ του χρήστη.

Με την ολοκλήρωση της διαδικασίας εγγραφής, το σύστημα εξασφαλίζει την ασφαλή αποθήκευση των προσωπικών δεδομένων και προετοιμάζει τον χρήστη για την ομαλή μετάβασή του στις υπόλοιπες λειτουργίες της εφαρμογής. Η συγκεκριμένη προσέγγιση συμβάλλει στη διατήρηση της ασφάλειας, της αξιοπιστίας και της επεκτασιμότητας του συστήματος, θέτοντας τις βάσεις για μια συνεπή και ελεγχόμενη εμπειρία χρήστη από το πρώτο κιόλας στάδιο αλληλεπίδρασης.

```
package gr.p21080.devjobs.controller;

import ...

@Controller @Marios K
@RequiredArgsConstructor
@RequestMapping("/register")
public class RegistrationController {

    private static final String ROLE_SLUG_CANDIDATE = "candidate"; 3 usages
    private static final String ROLE_SLUG_COMPANY = "company"; 3 usages

    private final UserRepository userRepo;
    private final RoleRepository roleRepo;
    private final PasswordEncoder passwordEncoder;
    private final UserProfileRepository userProfileRepository;
    private final CustomUserDetailsService userDetailsService;
    private final HttpServletRequest request;

    @GetMapping @Marios K
    public String chooseRole() { return "auth/register-choose"; }

    @GetMapping("/company") @Marios K
    public String registerCompanyForm(Model model) {
        model.addAttribute( attributeName: "user", new User());
        model.addAttribute( attributeName: "roleSlug", ROLE_SLUG_COMPANY);
        model.addAttribute( attributeName: "roleName", resolveRoleName(ROLE_SLUG_COMPANY));
        return "auth/register-form";
    }
}
```

Εικόνα 16: Backend στιγμιότυπο της κλάσης εγγραφής που υλοποιεί την δρομολόγηση και την αρχική φόρμα επιλογής ρόλου για την διαδικασία εγγραφής

```

@GetMapping("/{candidate}") @Marios K
public String registerCandidateForm(Model model) {
    model.addAttribute("user", new User());
    model.addAttribute("roleSlug", ROLE_SLUG_CANDIDATE);
    model.addAttribute("roleName", resolveRoleName(ROLE_SLUG_CANDIDATE));
    return "auth/register-form";
}

@InitBinder @Marios K
public void initBinder(org.springframework.web.bind.WebDataBinder binder) { binder.setDisallowedFields(

@PostMapping("/{role}") @Marios K
public String processRegistration(@PathVariable String role,
                                @ModelAttribute User user,
                                @RequestParam("roleName") String roleName,
                                Model model) {
    try {
        if (userRepo.findByEmail(user.getEmail()).isPresent()) {
            model.addAttribute("error", "Αυτό το email υπάρχει ήδη! Δοκίμασε με");
            model.addAttribute("user", user);
            String resolvedRole = resolveRoleName(role);
            model.addAttribute("roleSlug", role);
            model.addAttribute("roleName", resolvedRole);
            return "auth/register-form";
        }
    }
}

user.setPassword(passwordEncoder.encode(user.getPassword()));

```

Εικόνα 17: Μέρος της λειτουργίας αποθήκευσης προφίλ του χρήστη, με έλεγχο ταυτοποίησης, επαλήθευση ρόλου και ενημέρωση του πίνακα "UserProfile"

#### 4.4.2 Λειτουργία Σύνδεσης και Διαχείρισης συνεδρίας

Η διαδικασία σύνδεσης στο παρόν σύστημα υλοποιείται μέσω του "AuthController", ο οποίος συνεργάζεται άμεσα με τον μηχανισμό Spring Security για την ασφαλή ταυτοποίηση των διαπιστευτηρίων των χρηστών. Ο χρήστης εισάγει τα στοιχεία σύνδεσής του, όπως το email και τον κωδικό πρόσβασης, και η εφαρμογή προχωρά στον έλεγχο της εγκυρότητάς τους, ακολουθώντας τις πολιτικές ασφάλειας που έχουν οριστεί στο σύστημα.

Κατά τη διάρκεια της διαδικασίας, η εφαρμογή αναζητά τον αντίστοιχο λογαριασμό χρήστη μέσω της υπηρεσίας "CustomUserDetailsService", η οποία είναι υπεύθυνη για την ανάκτηση των αποθηκευμένων στοιχείων ταυτοποίησης από τη βάση δεδομένων. Τα δεδομένα αυτά συγκρίνονται με τις πληροφορίες που υποβλήθηκαν κατά τη σύνδεση, ενώ παράλληλα πραγματοποιείται έλεγχος της κρυπτογραφημένης μορφής του κωδικού πρόσβασης, εξασφαλίζοντας ότι η διαδικασία παραμένει ασφαλής και αξιόπιστη.

Σε περίπτωση επιτυχούς ταυτοποίησης, δημιουργείται η συνεδρία του χρήστη και του παρέχεται πρόσβαση σε προστατευμένες λειτουργίες της εφαρμογής, ανάλογα με τον ρόλο που του έχει αποδοθεί. Η διαχείριση της συνεδρίας επιτρέπει στο σύστημα να αναγνωρίζει τον χρήστη καθ' όλη τη διάρκεια της αλληλεπίδρασής του με την πλατφόρμα, διασφαλίζοντας τη συνέχεια και τη συνοχή της εμπειρίας χρήστη.

Παράλληλα, μετά την επιτυχή σύνδεση, ενεργοποιείται ο "ProfileCompletionInterceptor", ο οποίος ελέγχει αν ο χρήστης έχει συμπληρώσει όλες τις απαραίτητες πληροφορίες που απαιτούνται για τη σωστή διαμόρφωση του προφίλ του. Ο έλεγχος αυτός αφορά βασικά στοιχεία, τα οποία θεωρούνται

κρίσιμα για τη λειτουργικότητα της πλατφόρμας και τη συμμετοχή του χρήστη στις διαδικασίες αντιστοίχισης και αλληλεπίδρασης.

Σε περίπτωση που το προφίλ δεν είναι πλήρως συμπληρωμένο, ο χρήστης ανακατευθύνεται αυτόματα στις αντίστοιχες φόρμες καταχώρισης, ώστε να ολοκληρώσει τα απαιτούμενα δεδομένα. Μέχρι την πλήρη συμμόρφωση με τις προδιαγραφές του συστήματος, η πρόσβαση σε ορισμένες λειτουργίες παραμένει περιορισμένη. Η προσέγγιση αυτή διασφαλίζει την ποιότητα και την πληρότητα των δεδομένων που διαχειρίζεται η πλατφόρμα, ενώ παράλληλα ενισχύει την ορθότητα και την αποτελεσματικότητα των παρεχόμενων υπηρεσιών.

```
package gr.p21080.devjobs.controller;

import ...

@Controller
public class AuthController {

    @GetMapping("/login")
    public String login() { return "auth/login"; }
}
```

Εικόνα 18: Κλάση για authorization

```

package gr.p21080.devjobs.config;

import ...

@Component
public class ProfileCompletionInterceptor implements HandlerInterceptor {

    private static final Set<String> ALLOWED_PREFIXES = Set.of(
        "/profile/personal",
        "/profile/career",
        "/profile/company",
        "/logout",
        "/register",
        "/css",
        "/js",
        "/images",
        "/error",
        "/favicon"
    );

    private static final Set<String> ALLOWED_ENDPOINTS = Set.of(
        "/api/skills"
    );

    private final ProfileCompletionService profileCompletionService;

    public ProfileCompletionInterceptor(ProfileCompletionService profileCompletionService) {
        this.profileCompletionService = profileCompletionService;
    }
}

```

Εικόνα 19: Κλάση που ελέγχει την ολοκλήρωση του προφίλ, σχετικά με την πρόσβαση σε προστατευμένες πληροφορίες

#### 4.4.3 Λειτουργία Συμπλήρωσης και Διαχείρισης Προφίλ Υποψηφίου

Η εφαρμογή απαιτεί την πλήρη και συστηματική καταγραφή των επαγγελματικών χαρακτηριστικών του υποψηφίου πριν από την ενεργοποίηση και χρήση των βασικών υπηρεσιών της πλατφόρμας. Η απαίτηση αυτή διασφαλίζει ότι τα δεδομένα που χρησιμοποιούνται στις διαδικασίες αντιστοίχισης είναι επαρκή, συνεπή και αντιπροσωπευτικά του πραγματικού επαγγελματικού προφίλ κάθε χρήστη.

Η συμπλήρωση και διαχείριση των στοιχείων του υποψηφίου υλοποιείται μέσω των “CandidateProfileController” και “CareerProfileController”, οι οποίοι παρέχουν τις απαραίτητες διεπαφές για την καταχώριση και την ενημέρωση των δεδομένων. Μέσω των συγκεκριμένων controllers, ο χρήστης έχει τη δυνατότητα να εισάγει προσωπικά στοιχεία, πληροφορίες που αφορούν την επαγγελματική του κατάρτιση, το επίπεδο εμπειρίας, καθώς και δεδομένα που σχετίζονται με το επαγγελματικό του υπόβαθρο.

Παράλληλα, δίνεται ιδιαίτερη έμφαση στη συλλογή πληροφοριών που αφορούν τις δεξιότητες και τις επαγγελματικές προτιμήσεις του υποψηφίου. Ο χρήστης μπορεί να δηλώσει τις τεχνικές και μη τεχνικές δεξιότητές του, καθώς και προτιμήσεις που σχετίζονται με τον τρόπο εργασίας, όπως η επιθυμία για απομακρυσμένη, υβριδική ή δια ζώσης απασχόληση. Τα στοιχεία αυτά θεωρούνται κρίσιμα για τη βελτίωση της ποιότητας των προτάσεων που παράγει ο μηχανισμός αντιστοίχισης.

Τα δεδομένα που εισάγονται από τον χρήστη αποθηκεύονται στη βάση δεδομένων μέσω των υπηρεσιών “UserProfileService” και “CareerProfileService”, οι οποίες είναι υπεύθυνες για την επιχειρησιακή λογική και τη διασφάλιση της ακεραιότητας των πληροφοριών. Ο διαχωρισμός αυτός επιτρέπει την καθαρή οργάνωση των δεδομένων και διευκολύνει τη μελλοντική επέκταση ή τροποποίηση της δομής του προφίλ.

Η διαχείριση των δεξιοτήτων πραγματοποιείται μέσω του “SkillController”, ο οποίος επιτρέπει την προσθήκη, την αφαίρεση και την ενημέρωση δεξιοτήτων στο προφίλ του υποψηφίου. Ο controller αυτός αλληλεπιδρά με τις οντότητες “UserSkill” και “Skill”, εξασφαλίζοντας τη σωστή συσχέτιση των δεξιοτήτων με τον αντίστοιχο χρήστη. Η συγκεκριμένη προσέγγιση επιτρέπει την ευέλικτη διαχείριση των δεξιοτήτων και υποστηρίζει την αποτελεσματική λειτουργία του μηχανισμού αντιστοίχισης, καθώς οι δεξιότητες αποτελούν βασικό κριτήριο αξιολόγησης.

Συνολικά, η λειτουργία συμπλήρωσης και διαχείρισης του προφίλ υποψηφίου συμβάλλει καθοριστικά στη δημιουργία ενός ολοκληρωμένου και αξιόπιστου επαγγελματικού προφίλ, το οποίο αξιοποιείται από το σύστημα για την παροχή στοχευμένων και ποιοτικών προτάσεων εργασίας.

```
package gr.p21080.devjobs.controller;

import ...

@Controller
@RequestMapping("/candidates")
public class CandidateProfileController {

    private static final Map<String, String> EMPLOYMENT_TYPE_LABELS = Map.of(
        k1: "Full-time", v1: "Πλήρης απασχόληση",
        k2: "Part-time", v2: "Μερική απασχόληση",
        k3: "Freelance", v3: "Ελεύθερος επαγγελματίας",
        k4: "Internship", v4: "Πρακτική άσκηση",
        k5: "Contract", v5: "Σύμβαση",
        k6: "Temporary", v6: "Προσωρινή"
    );

    private static final Map<String, String> WORK_ARRANGEMENT_LABELS = Map.of(
        k1: "Remote", v1: "Τηλεργασία",
        k2: "Hybrid", v2: "Υβριδική εργασία",
        k3: "On-site", v3: "Στο γραφείο",
        k4: "No Preference", v4: "Χωρίς προτίμηση"
    );

    private final UserRepository userRepository;
    private final UserProfileRepository userProfileRepository;
    private final CareerProfileRepository careerProfileRepository;
    private final UserSkillRepository userSkillRepository;
    private final CertificationRepository certificationRepository;
    private final EducationRepository educationRepository;
```

Εικόνα 20: Παράδειγμα κλάσης υπεύθυνης για την καταχώριση των προτιμήσεων του χρήστη

#### 4.4.4 Λειτουργία Δημιουργίας και Διαχείρισης Αγγελιών

Οι εργοδότες έχουν τη δυνατότητα να δημιουργούν, να επεξεργάζονται και να διαγράφουν αγγελίες εργασίας μέσω του “JobController”, ο οποίος αποτελεί το βασικό σημείο αλληλεπίδρασης μεταξύ του περιβάλλοντος χρήστη και της επιχειρησιακής λογικής του συστήματος. Η λειτουργία αυτή επιτρέπει την οργανωμένη και δομημένη καταχώριση νέων θέσεων εργασίας, συμβάλλοντας στη σαφή παρουσίαση των αναγκών κάθε εργοδότη προς τους υποψήφιους εργαζόμενους.

Κάθε αγγελία περιλαμβάνει ένα σύνολο βασικών χαρακτηριστικών που περιγράφουν τη θέση εργασίας, όπως ο τίτλος της θέσης, το αντικείμενο και το περιβάλλον εργασίας, ο τύπος απασχόλησης (πλήρης, μερική ή απομακρυσμένη εργασία), καθώς και πρόσθετες πληροφορίες που σχετίζονται με τις απαιτήσεις του ρόλου. Η σαφής καταγραφή των στοιχείων αυτών συμβάλλει στη βελτίωση της



ποιότητας των αγγελιών και διευκολύνει την κατανόηση των προσδοκιών του εργοδότη από τους υποψήφιους.

Η επιχειρησιακή λογική που αφορά τη δημιουργία και διαχείριση των αγγελιών υλοποιείται στο “JobPostService”, το οποίο είναι υπεύθυνο για τον έλεγχο εγκυρότητας των δεδομένων, τη διαχείριση των ενημερώσεων και την εφαρμογή των επιχειρησιακών κανόνων. Μέσω του συγκεκριμένου επιπέδου διασφαλίζεται ότι οι αγγελίες πληρούν τις απαιτούμενες προδιαγραφές πριν αποθηκευτούν στη βάση δεδομένων.

Η αποθήκευση και η ανάκτηση των δεδομένων των αγγελιών πραγματοποιούνται μέσω του “JobPostRepository”, το οποίο διαχειρίζεται την επικοινωνία με τη βάση δεδομένων και εξασφαλίζει τη σωστή καταγραφή των πληροφοριών. Ο διαχωρισμός μεταξύ controller, service και repository επιτρέπει την καθαρή αρχιτεκτονική του συστήματος και διευκολύνει τη συντήρηση και τη μελλοντική επέκτασή του.

Ιδιαίτερη σημασία δίνεται στην καταγραφή των απαιτούμενων δεξιοτήτων κάθε αγγελίας, οι οποίες αποθηκεύονται στις οντότητες “JobSkill” και συσχετίζονται με τη συγκεκριμένη θέση εργασίας. Η προσέγγιση αυτή επιτρέπει τη δημιουργία μιας ολοκληρωμένης και αναλυτικής περιγραφής της θέσης, η οποία αξιοποιείται στη συνέχεια από τον μηχανισμό αντιστοίχισης για την παραγωγή στοχευμένων και αξιόπιστων προτάσεων προς τους υποψήφιους.

Συνολικά, η λειτουργία δημιουργίας και διαχείρισης αγγελιών αποτελεί βασικό πυλώνα της πλατφόρμας DevJobs, καθώς συνδέει άμεσα τις ανάγκες των εργοδοτών με τις δυνατότητες και τα χαρακτηριστικά των υποψηφίων, υποστηρίζοντας μια αποτελεσματική και διαφανή διαδικασία εύρεσης εργασίας.

```
package gr.p21080.devjobs.controller;  
  
import ...  
  
@Controller @ Marios K  
@RequestMapping("/jobs")  
public class JobController {  
  
    private static final List<String> JOB_TYPE_OPTIONS = List.of(  
        "Full-time",  
        "Part-time",  
        "Contract",  
        "Temporary",  
        "Internship",  
        "Freelance"  
    );  
  
    private static final List<String> WORK_ARRANGEMENT_OPTIONS = List.of(  
        "On-site",  
        "Hybrid",  
        "Remote"  
    );  
  
    private static final List<String> EXPERIENCE_OPTIONS = List.of(  
        "0-2 years",  
        "3-5 years",  
        "5+ years"  
    );  
};
```

Εικόνα 21: Κλάση που αφορά τον έλεγχο και την διαχείριση των αγγελιών εργασίας

```

package gr.p21080.devjobs.service;

import ...

@Service 12 usages  Marios K
public class JobPostService {

    private final JobPostRepository jobPostRepository; 10 usages

    public JobPostService(JobPostRepository jobPostRepository) { this.jobPostRepository = jobPostRepository; }

    @Transactional(readOnly = true) no usages  Marios K
    public List<JobPost> findAll() { return jobPostRepository.findAllDetailed(); }

    @Transactional(readOnly = true) 1 usage  Marios K
    public List<JobPost> findAllDetailed() { return jobPostRepository.findAllDetailed(); }

    @Transactional(readOnly = true) 3 usages  Marios K
    public List<JobPost> findByEmployer(User employer) {
        return jobPostRepository.findByEmployerOrderByCreatedAtDesc(employer);
    }

    @Transactional  Marios K
    public JobPost save(JobPost jobPost) { return jobPostRepository.save(jobPost); }

    @Transactional  Marios K
    public void delete(Long id) { jobPostRepository.deleteById(id); }

    @Transactional  Marios K
    public void delete(JobPost jobPost) { jobPostRepository.delete(jobPost); }
}

```

Εικόνα 22: Παράδειγμα κλάσης που είναι υπεύθυνο για την ανάκτηση, διαγραφή και αποθήκευση των αγγελιών εργασίας

```

package gr.p21080.devjobs.entity;

import jakarta.persistence.*;

@Entity  Marios K
@Table(name = "job_skills")
public class JobSkill {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @ManyToOne(fetch = FetchType.LAZY) 2 usages
    @JoinColumn(name = "job_id", nullable = false)
    private JobPost jobPost;

    @ManyToOne(fetch = FetchType.LAZY) 2 usages
    @JoinColumn(name = "skill_id", nullable = false)
    private Skill skill;

    // 0.0-1.0: απαιτούμενο επίπεδο
    @Column(nullable = false) 2 usages
    private Double requiredLevel;

    public Long getId() { return id; }

    public JobPost getJobPost() { return jobPost; }

    public void setJobPost(JobPost jobPost) { this.jobPost = jobPost; }
}

```

Εικόνα 23: Κλάση που συνδράμει στην συσχέτιση των δεξιοτήτων με τις υπάρχουσες αγγελίες

#### 4.4.5 Λειτουργία Αναζήτησης και Φιλτραρίσματος Αγγελιών

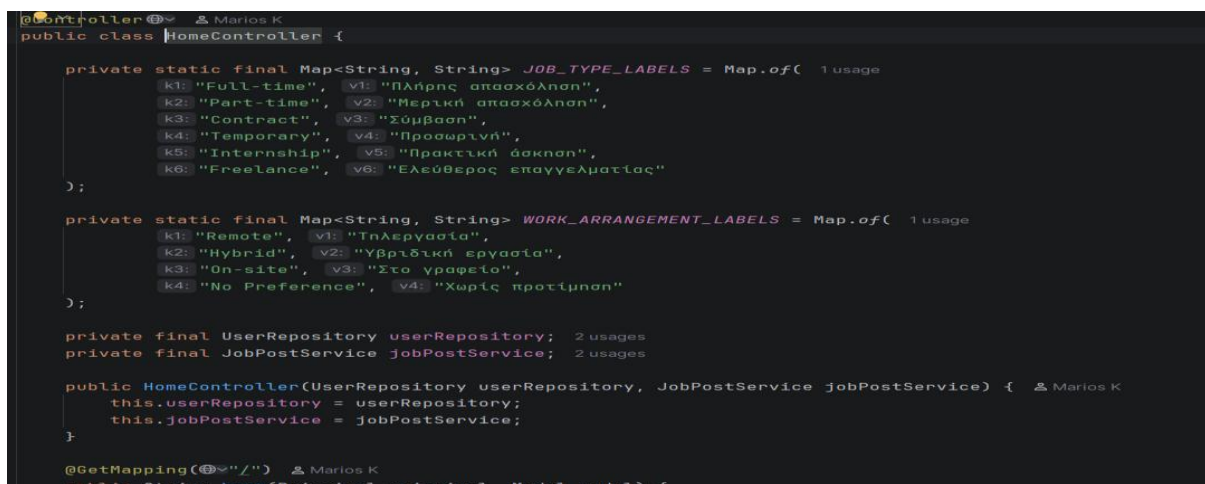
Η εφαρμογή παρέχει έναν ολοκληρωμένο μηχανισμό αναζήτησης και φιλτραρίσματος των διαθέσιμων θέσεων εργασίας, με στόχο τη διευκόλυνση των χρηστών στην εύρεση αγγελιών που ανταποκρίνονται στα επαγγελματικά τους ενδιαφέροντα και προσόντα. Η λειτουργία αυτή αποτελεί βασικό στοιχείο της εμπειρίας χρήστη, καθώς επιτρέπει την άμεση πρόσβαση σε μεγάλο όγκο αγγελιών με οργανωμένο και αποδοτικό τρόπο.

Τα αιτήματα αναζήτησης και φιλτραρίσματος διαχειρίζονται οι “HomeController” και “JobController”, οι οποίοι είναι υπεύθυνοι για τη λήψη των παραμέτρων που εισάγει ο χρήστης και την ανάκτηση των αντίστοιχων αποτελεσμάτων από το σύστημα. Τα δεδομένα που επιστρέφονται προβάλλονται στο περιβάλλον χρήστη μέσω των προτύπων Thymeleaf, εξασφαλίζοντας δυναμική και ευανάγνωστη παρουσίαση των αγγελιών.

Η διαδικασία αναζήτησης βασίζεται σε συνδυασμό λέξεων-κλειδιών και φίλτρων, όπως το επίπεδο επαγγελματικής εμπειρίας, η τοποθεσία, ο τύπος απασχόλησης και άλλα βασικά γνωρίσματα της θέσης εργασίας. Η χρήση πολλαπλών κριτηρίων επιτρέπει στον χρήστη να περιορίσει τα αποτελέσματα και να εστιάσει σε αγγελίες που ανταποκρίνονται πιο ακριβώς στις ανάγκες και τις προτιμήσεις του.

Η δυνατότητα φιλτραρίσματος συμβάλλει ουσιαστικά στη μείωση του χρόνου αναζήτησης και στη βελτίωση της ποιότητας των αποτελεσμάτων, ιδιαίτερα σε περιπτώσεις όπου ο αριθμός των διαθέσιμων αγγελιών είναι μεγάλος. Ως αποτέλεσμα, ο χρήστης αποκτά τη δυνατότητα στοχοθετημένης αναζήτησης, η οποία ευθυγραμμίζεται με τα προσωπικά και επαγγελματικά του χαρακτηριστικά.

Συνολικά, η λειτουργία αναζήτησης και φιλτραρίσματος ενισχύει τη χρηστικότητα της πλατφόρμας και υποστηρίζει την αποτελεσματική πλοήγηση στο περιεχόμενο, προσφέροντας μια πιο οργανωμένη και προσαρμοσμένη εμπειρία εύρεσης εργασίας.

The image shows a code editor window with the following Java code:

```
@Controller
public class HomeController {

    private static final Map<String, String> JOB_TYPE_LABELS = Map.of(
        k1: "Full-time", v1: "Πλήρης απασχόληση",
        k2: "Part-time", v2: "Μερική απασχόληση",
        k3: "Contract", v3: "Σύμβαση",
        k4: "Temporary", v4: "Προσωρινή",
        k5: "Internship", v5: "Πρακτική άσκηση",
        k6: "Freelance", v6: "Ελεύθερος επαγγελματίας"
    );

    private static final Map<String, String> WORK_ARRANGEMENT_LABELS = Map.of(
        k1: "Remote", v1: "Τηλεργασία",
        k2: "Hybrid", v2: "Υβριδική εργασία",
        k3: "On-site", v3: "Στο γραφείο",
        k4: "No Preference", v4: "Χωρίς προτίμηση"
    );

    private final UserRepository userRepository;
    private final JobPostService jobPostService;

    public HomeController(UserRepository userRepository, JobPostService jobPostService) {
        this.userRepository = userRepository;
        this.jobPostService = jobPostService;
    }

    @GetMapping("/")
    public String home(Model model) {
```

Εικόνα 24: Κλάση υπεύθυνη για την προβολή αγγελιών εργασίας και για την διαχείριση της αρχικής σελίδας

#### 4.4.6 Λειτουργία Υποβολής Αίτησης και Διαχείρισης Συνεντεύξεων

Ο υποψήφιος εργαζόμενος έχει τη δυνατότητα να υποβάλει αίτηση για μια διαθέσιμη αγγελία εργασίας μέσω του “ApplicationController”, ο οποίος αναλαμβάνει τη διαχείριση της συγκεκριμένης λειτουργίας και τη σύνδεσή της με τα υπόλοιπα υποσυστήματα της εφαρμογής. Η υποβολή αίτησης αποτελεί

κρίσιμο στάδιο στη διαδικασία εύρεσης εργασίας, καθώς σηματοδοτεί την επίσημη εκδήλωση ενδιαφέροντος του υποψηφίου για μια συγκεκριμένη θέση.

Κατά την υποβολή της αίτησης, δημιουργείται μια νέα εγγραφή στην οντότητα “Application”, η οποία αρχικοποιείται με κατάσταση “PENDING”, υποδηλώνοντας ότι η αίτηση βρίσκεται σε αναμονή αξιολόγησης από τον εργοδότη. Η αποθήκευση και η διαχείριση της αίτησης πραγματοποιούνται μέσω των υπηρεσιών “ApplicationService” και “ApplicationRepository”, οι οποίες διασφαλίζουν τη σωστή καταγραφή των δεδομένων και τη διατήρηση της ακεραιότητας της πληροφορίας στη βάση δεδομένων.

Με την επιτυχή καταχώριση της αίτησης, ο εργοδότης ενημερώνεται άμεσα μέσω του “NotificationService”, γεγονός που επιτρέπει την έγκαιρη παρακολούθηση νέων αιτήσεων και τη γρήγορη ανταπόκριση. Η χρήση του μηχανισμού ειδοποιήσεων ενισχύει την επικοινωνία μεταξύ των δύο πλευρών και μειώνει τις καθυστερήσεις στη διαδικασία αξιολόγησης.

Ο εργοδότης, μέσω της πλατφόρμας, έχει τη δυνατότητα να διαχειριστεί την πορεία κάθε αίτησης, επιλέγοντας να την αποδεχθεί, να την απορρίψει ή να προχωρήσει στον προγραμματισμό συνέντευξης. Ο προγραμματισμός συνεντεύξεων υλοποιείται με τη χρήση της δομής “InterviewDetails”, στην οποία αποθηκεύονται πληροφορίες όπως η ημερομηνία, η ώρα και ο τρόπος διεξαγωγής της συνέντευξης. Έτσι, η διαδικασία παραμένει οργανωμένη και σαφώς καθορισμένη για όλα τα εμπλεκόμενα μέρη.

Η συνολική αυτή λειτουργία εξασφαλίζει την πλήρη παρακολούθηση της πορείας κάθε αίτησης, από την αρχική υποβολή έως την τελική απόφαση. Παράλληλα, συμβάλλει στη διαφάνεια της διαδικασίας πρόσληψης και υποστηρίζει μια πιο αποτελεσματική και δομημένη επιλογή προσωπικού, τόσο για τους εργοδότες όσο και για τους υποψηφίους.

```
package gr.p21080.devjobs.controller;

import ...

@Controller
@RequestMapping("/applications")
public class ApplicationController {

    private final ApplicationService applicationService; // 10 usages
    private final FavoritesService favoritesService; // 3 usages
    private final UserRepository userRepository; // 2 usages
    private static final Map<ApplicationStatus, String> STATUS_BADGE_CLASSES = Stream.of( // 3 usages
        Map.entry(ApplicationStatus.SUBMITTED, "bg-secondary-subtle text-secondary"),
        Map.entry(ApplicationStatus.UNDER_REVIEW, "bg-warning-subtle text-warning"),
        Map.entry(ApplicationStatus.REJECTED, "bg-danger-subtle text-danger"),
        Map.entry(ApplicationStatus.INTERVIEW_REQUESTED, "bg-info-subtle text-info"),
        Map.entry(ApplicationStatus.INTERVIEW_CONFIRMED, "bg-success-subtle text-success"),
        Map.entry(ApplicationStatus.INTERVIEW_CANCELLED, "bg-secondary-subtle text-secondary"),
        Map.entry(ApplicationStatus.INTERVIEW_DECLINED, "bg-danger-subtle text-danger")
    ).collect(Collectors.toUnmodifiableMap(Map.Entry::getKey, Map.Entry::getValue));

    public static Map<ApplicationStatus, String> statusBadgeClasses() { return STATUS_BADGE_CLASSES; }

    public ApplicationController(ApplicationService applicationService, // 10 usages
        FavoritesService favoritesService, // 3 usages
        UserRepository userRepository) {
        this.applicationService = applicationService;
        this.favoritesService = favoritesService;
        this.userRepository = userRepository;
    }
}
```

Εικόνα 25: Διαχείριση αιτημάτων που αφορούν την προβολή, υποβολή και παρακολούθηση της κατάστασης των αιτήσεων εργασίας

```

package gr.p21080.devjobs.service;

import ...

@Service 11 usages Marios K
public class ApplicationService {

    private final ApplicationRepository applicationRepository; 10 usages
    private final NotificationService notificationService; 3 usages

    public ApplicationService(ApplicationRepository applicationRepository, Marios K
        NotificationService notificationService) {
        this.applicationRepository = applicationRepository;
        this.notificationService = notificationService;
    }

    @Transactional 6 usages Marios K
    public Application applyFor(JobPost jobPost, User candidate, String coverLetter) {
        if (jobPost == null || candidate == null) {
            throw new IllegalArgumentException("Δεν ήταν δυνατή η αποθήκευση της αίτησης.");
        }
        Optional<Application> existing = applicationRepository.findByJobIdAndApplicantId(jobPost.getId(),
        if (existing.isPresent()) {
            return existing.get();
        }
        Application application = new Application();
        application.setJob(jobPost);
        application.setApplicant(candidate);
        if (StringUtils.hasText(coverLetter)) {
            application.setCoverLetter(coverLetter.trim());
        }
    }
}

```

Εικόνα 26: Κλάση υπεύθυνη για τον έλεγχο εγκυρότητας, την αποθήκευση αιτήσεων και την αποστολή σχετικών ειδοποιήσεων

```

package gr.p21080.devjobs.repository;

import ...

public interface ApplicationRepository extends JpaRepository<Application, Long> { 3 usages Marios K

    @EntityGraph(attributePaths = { 1 usage Marios K
        "job", "job.employer", "job.employer.companyProfile", "applicant", "applicant.personalProfile"
    })
    List<Application> findByApplicantIdOrderByCreatedAtDesc(Long userId);

    @EntityGraph(attributePaths = { 1 usage Marios K
        "job", "job.employer", "job.employer.companyProfile", "applicant", "applicant.personalProfile"
    })
    List<Application> findByJobIdOrderByCreatedAtDesc(Long jobId);

    @EntityGraph(attributePaths = { 2 usages Marios K
        "job", "job.employer", "job.employer.companyProfile", "applicant", "applicant.personalProfile"
    })
    Optional<Application> findByJobIdAndApplicantId(Long jobId, Long applicantId);

    @EntityGraph(attributePaths = { 1 usage Marios K
        "job", "job.employer", "job.employer.companyProfile", "applicant", "applicant.personalProfile"
    })
    List<Application> findByJobEmployerIdOrderByCreatedAtDesc(Long employerId);

    @EntityGraph(attributePaths = { 1 usage Marios K
        "job", "job.employer", "job.employer.companyProfile", "applicant", "applicant.personalProfile"
    })
    Optional<Application> findDetailedById(Long id);
}

```

Εικόνα 27: Παροχή πρόσβασης στην βάση δεδομένων για τις αιτήσεις εργασίας, δίδοντας εξειδικευμένα ερωτήματα με πλήρη φόρτιση σχετιζόμενων δεδομένων

```

package gr.p21080.devjobs.service;

import ...

@Service 8 usages 2 Marios K
public class NotificationService {

    private final NotificationRepository notificationRepository; 10 usages

    public NotificationService(NotificationRepository notificationRepository) { 2 Marios K
        this.notificationRepository = notificationRepository;
    }

    @Transactional 2 usages 2 Marios K
    public Notification createNotification(User recipient,
                                           NotificationType type,
                                           String title,
                                           String message,
                                           String actionUrl) {
        if (recipient == null || recipient.getId() == null) {
            return null;
        }
        Notification notification = new Notification();
        notification.setRecipient(recipient);
        notification.setType(type != null ? type : NotificationType.GENERAL);
        notification.setTitle(title);
        notification.setMessage(message);
        notification.setActionUrl(actionUrl);
        return notificationRepository.save(notification);
    }
}

```

Εικόνα 28: Κλάση που αναλαμβάνει την δημιουργία, την αποθήκευση και την άμεση ενημέρωση των χρηστών μέσω έγκαιρων ειδοποιήσεων

```

package gr.p21080.devjobs.model;

import ...

public record InterviewDetails(LocalDate date, 9 usages 2 Marios K
                               LocalTime time, 1 usage
                               InterviewMode mode, 3 usages
                               String location, 2 usages
                               String meetingLink, 2 usages
                               String notes) { 2 usages

    public boolean isOnline() { return mode == InterviewMode.ONLINE; }

    public boolean isInPerson() { return mode == InterviewMode.IN_PERSON; }
}

```

Εικόνα 29: Μοντέλο που αποθηκεύει πληροφορίες της συνέντευξης, όπως ημερομηνία, ώρα και τρόπο διεξαγωγής

#### 4.4.7 Λειτουργία Αγαπημένων Αγγελιών και Υποψηφίων

Η εφαρμογή παρέχει τη δυνατότητα αποθήκευσης αγαπημένων αγγελιών από την πλευρά των υποψηφίων, καθώς και αγαπημένων υποψηφίων από την πλευρά των εργοδοτών. Η λειτουργία αυτή επιτρέπει στους χρήστες να διατηρούν εύκολα προσβάσιμες επιλογές που θεωρούν ενδιαφέρουσες, χωρίς να απαιτείται επαναλαμβανόμενη αναζήτηση στο σύστημα.

Η υλοποίηση πραγματοποιείται μέσω του “FavoritesController” και της αντίστοιχης υπηρεσίας “FavoritesService”, οι οποίες διαχειρίζονται την καταχώριση και την ανάκτηση των αγαπημένων



εγγραφών. Τα δεδομένα αποθηκεύονται στις οντότητες “FavoriteJob” και “FavoriteCandidate”, εξασφαλίζοντας τη σωστή συσχέτιση μεταξύ χρηστών και επιλεγμένων αγγελιών ή υποψηφίων.

Με την καταχώριση αγαπημένων εγγραφών, οι χρήστες αποκτούν γρήγορη και άμεση πρόσβαση στις θέσεις εργασίας ή στους υποψηφίους που έχουν επιλέξει, γεγονός που βελτιώνει τη συνολική εμπειρία πλοήγησης και διευκολύνει τη λήψη αποφάσεων. Η συγκεκριμένη λειτουργία λειτουργεί υποστηρικτικά προς τις βασικές υπηρεσίες της πλατφόρμας, προσφέροντας έναν απλό αλλά ουσιαστικό μηχανισμό οργάνωσης ενδιαφερόντων.

```
package gr.p21080.devjobs.entity;

import ...

@Entity @ Marios K
@Table(name = "favorite_candidates", uniqueConstraints = {
    @UniqueConstraint(columnNames = {"employer_id", "candidate_id"})
})
public class FavoriteCandidate {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @ManyToOne(optional = false, fetch = FetchType.LAZY) 2 usages
    @JoinColumn(name = "employer_id", nullable = false)
    private User employer;

    @ManyToOne(optional = false, fetch = FetchType.LAZY) 2 usages
    @JoinColumn(name = "candidate_id", nullable = false)
    private User candidate;

    @Column(name = "created_at", nullable = false, updatable = false) 2 usages
    private Instant createdAt;

    @PrePersist @ Marios K
    public void onCreate() { this.createdAt = Instant.now(); }

    public Long getId() { return id; }
```

Εικόνα 30: Κλάση που αποθηκεύει τους αγαπημένους υποψήφιους του εργοδότη

```
package gr.p21080.devjobs.controller;

import ...

@Controller @ Marios K
@RequestMapping("/favorites")
public class FavoritesController {

    private final FavoritesService favoritesService; 3 usages
    private final UserRepository userRepository; 2 usages

    public FavoritesController(FavoritesService favoritesService, @ Marios K
        UserRepository userRepository) {
        this.favoritesService = favoritesService;
        this.userRepository = userRepository;
    }

    @GetMapping("/jobs") @ Marios K
    public String favoriteJobs(Principal principal, Model model) {
        User candidate = requireCandidate(principal);
        List<FavoriteJob> favorites = favoritesService.findFavoriteJobs(candidate);
        model.addAttribute("favorites", favorites);
        return "favorites/jobs";
    }

    @GetMapping("/candidates") @ Marios K
    public String favoriteCandidates(Principal principal, Model model) {
        User employer = requireEmployer(principal);
        List<FavoriteCandidate> favorites = favoritesService.findFavoriteCandidates(employer);
        model.addAttribute("favorites", favorites);
    }
}
```

Εικόνα 31: Κλάση που διαχειρίζεται αγαπημένους υποψήφιους και αγγελίες



```

package gr.p21080.devjobs.service;

import ...

@Service // 11 usages  ⚡ Marios K
public class FavoritesService {

    private final FavoriteJobRepository favoriteJobRepository; // 6 usages
    private final FavoriteCandidateRepository favoriteCandidateRepository; // 6 usages

    public FavoritesService(FavoriteJobRepository favoriteJobRepository, // ⚡ Marios K
                           FavoriteCandidateRepository favoriteCandidateRepository) {
        this.favoriteJobRepository = favoriteJobRepository;
        this.favoriteCandidateRepository = favoriteCandidateRepository;
    }

    @Transactional // 6 usages  ⚡ Marios K
    public boolean toggleJobFavorite(User candidate, JobPost job) {
        if (candidate == null || job == null) {
            return false;
        }
        Optional<FavoriteJob> existing = favoriteJobRepository.findByCandidateIdAndJobId(candidate.getId(), job.getId());
        if (existing.isPresent()) {
            favoriteJobRepository.delete(existing.get());
            return false;
        }
        FavoriteJob favoriteJob = new FavoriteJob();
        favoriteJob.setCandidate(candidate);
        favoriteJob.setJob(job);
        favoriteJobRepository.save(favoriteJob);
    }
}

```

Εικόνα 32: Υπεύθυνη κλάση για την επιχειρησιακή λογική των αγαπημένων εγγραφών

```

package gr.p21080.devjobs.entity;

import ...

@Entity // ⚡ Marios K
@Table(name = "favorite_jobs", uniqueConstraints = {
    @UniqueConstraint(columnNames = {"candidate_id", "job_id"})
})
public class FavoriteJob {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @ManyToOne(optional = false, fetch = FetchType.LAZY) // 2 usages
    @JoinColumn(name = "candidate_id", nullable = false)
    private User candidate;

    @ManyToOne(optional = false, fetch = FetchType.LAZY) // 2 usages
    @JoinColumn(name = "job_id", nullable = false)
    private JobPost job;

    @Column(name = "created_at", nullable = false, updatable = false) // 2 usages
    private Instant createdAt;

    @PrePersist // ⚡ Marios K
    public void onCreate() { this.createdAt = Instant.now(); }

    public Long getId() { return id; }
}

```

Εικόνα 33: Κλάση που συνδράμει στην αποθήκευση αγαπημένων αγγελιών από πλευράς υποψηφίου

#### 4.4.8 Λειτουργία ειδοποιήσεων

Οι ειδοποιήσεις υλοποιούνται μέσω της υπηρεσίας “NotificationService” και προβάλλονται στους χρήστες μέσω του αντίστοιχου “NotificationController”, ο οποίος αναλαμβάνει τη διαχείριση των αιτημάτων που σχετίζονται με την ανάκτηση και την εμφάνισή τους στο περιβάλλον χρήστη. Ο σχεδιασμός της συγκεκριμένης λειτουργίας στοχεύει στην άμεση και αξιόπιστη ενημέρωση των χρηστών για σημαντικά γεγονότα που προκύπτουν κατά τη χρήση της πλατφόρμας.

Το σύστημα δημιουργεί ειδοποιήσεις για κρίσιμες ενέργειες, όπως η αποδοχή ή απόρριψη μιας αίτησης εργασίας, η αλλαγή της κατάστασης μιας υποβολής, καθώς και η καταχώριση νέας αίτησης από υποψήφιο. Επομένως, διασφαλίζεται ότι τόσο οι εργοδότες όσο και οι υποψήφιοι

ενημερώνονται έγκαιρα για ενέργειες που τους αφορούν άμεσα, χωρίς να απαιτείται συνεχής χειροκίνητος έλεγχος της πλατφόρμας.

Οι ειδοποιήσεις αποθηκεύονται στην οντότητα “Notification”, η οποία περιλαμβάνει πληροφορίες όπως το περιεχόμενο του μηνύματος, τον τύπο της ειδοποίησης, τον χρόνο δημιουργίας και τον παραλήπτη. Η αποθήκευση αυτή επιτρέπει την ιστορικότητα των ειδοποιήσεων, δίνοντας τη δυνατότητα στους χρήστες να ανατρέχουν σε προηγούμενες ενημερώσεις και να παρακολουθούν τη συνολική εξέλιξη των αλληλεπιδράσεών τους.

Η λειτουργία των ειδοποιήσεων ενισχύει ουσιαστικά την επικοινωνία μεταξύ εργοδοτών και υποψηφίων, βελτιώνοντας τη συνολική εμπειρία χρήστη και αυξάνοντας το επίπεδο διαφάνειας στη διαδικασία πρόσληψης. Παράλληλα, συμβάλλει στη διατήρηση μιας οργανωμένης και δομημένης ροής πληροφοριών, μειώνοντας την πιθανότητα παρανοήσεων ή καθυστερήσεων και καθιστώντας τη διαχείριση των ενεργειών πιο αποδοτική.

```
package gr.p21080.devjobs.service;

import ...

@Service 8 usages  Marios K
public class NotificationService {

    private final NotificationRepository notificationRepository; 10 usages

    public NotificationService(NotificationRepository notificationRepository) {  Marios K
        this.notificationRepository = notificationRepository;
    }

    @Transactional 2 usages  Marios K
    public Notification createNotification(User recipient,
                                           NotificationType type,
                                           String title,
                                           String message,
                                           String actionUrl) {

        if (recipient == null || recipient.getId() == null) {
            return null;
        }
        Notification notification = new Notification();
        notification.setRecipient(recipient);
        notification.setType(type != null ? type : NotificationType.GENERAL);
        notification.setTitle(title);
        notification.setMessage(message);
        notification.setActionUrl(actionUrl);
        return notificationRepository.save(notification);
    }
}
```

Εικόνα 34: Υλοποίηση και δημιουργία ειδοποιήσεων στο σύστημα

```

package gr.p21080.devjobs.controller;

import ...

@Controller @Marios K
@RequestMapping("/notifications")
public class NotificationController {

    private final NotificationService notificationService; 5 usages
    private final UserRepository userRepository; 2 usages

    public NotificationController(NotificationService notificationService, @Marios K
        UserRepository userRepository) {
        this.notificationService = notificationService;
        this.userRepository = userRepository;
    }

    @GetMapping @Marios K
    public String listNotifications(Principal principal, Model model) {
        User user = requireUser(principal);
        List<Notification> notifications = notificationService.findAllFor(user);
        model.addAttribute(attributeName: "notifications", notifications);
        return "notifications/list";
    }

    @PostMapping("/{id}/read") @Marios K
    public String markAsRead(@PathVariable Long id,
        @RequestHeader(value = "Referer", required = false) String referer,
        Principal principal,
        RedirectAttributes redirectAttributes) {

```

Εικόνα 35: Προβολή και διαχείριση ειδοποιήσεων του χρήστη

```

package gr.p21080.devjobs.entity;

import ...

@Entity @Marios K
@Table(name = "notifications")
public class Notification {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @ManyToOne(optional = false, fetch = FetchType.LAZY) 2 usages
    @JoinColumn(name = "recipient_id", nullable = false)
    private User recipient;

    @Enumerated(EnumType.STRING) 2 usages
    @Column(length = 50, nullable = false)
    private NotificationType type = NotificationType.GENERAL;

    @Column(nullable = false, length = 200) 2 usages
    private String title;

    @Column(nullable = false, length = 1000) 2 usages
    private String message;

    @Column(name = "action_url") 2 usages
    private String actionUrl;

    @Column(name = "is_read", nullable = false) 2 usages
    private boolean isRead;

```

Εικόνα 36: Οντότητα βάσης δεδομένων για την αποθήκευση ειδοποιήσεων χρηστών

#### 4.4.9 Λειτουργία Γενικής Πλοήγησης και Κοινών Μοντέλων

Η εφαρμογή περιλαμβάνει τον μηχανισμό “GlobalModelAttributes”, ο οποίος χρησιμοποιείται για την παροχή κοινών δεδομένων σε όλες τις σελίδες του συστήματος. Μέσω του συγκεκριμένου μηχανισμού καθίσταται δυνατή η αυτόματη διάθεση πληροφοριών, όπως τα βασικά στοιχεία του συνδεδεμένου χρήστη, οι ενεργές ειδοποιήσεις και δεδομένα που σχετίζονται με το προφίλ, χωρίς να απαιτείται επαναλαμβανόμενος κώδικας σε κάθε επιμέρους controller.

Η προσέγγιση αυτή συμβάλλει σημαντικά στη μείωση της πολυπλοκότητας και του πλεονασμού κώδικα στα επιμέρους Controllers, καθώς οι κοινές πληροφορίες διαχειρίζονται κεντρικά και διατίθενται ομοιόμορφα σε όλη την εφαρμογή. Σε αυτό το πλαίσιο, η συντήρηση του συστήματος καθίσταται πιο αποδοτική και η αναγνωσιμότητα της backend αρχιτεκτονικής ενισχύεται.

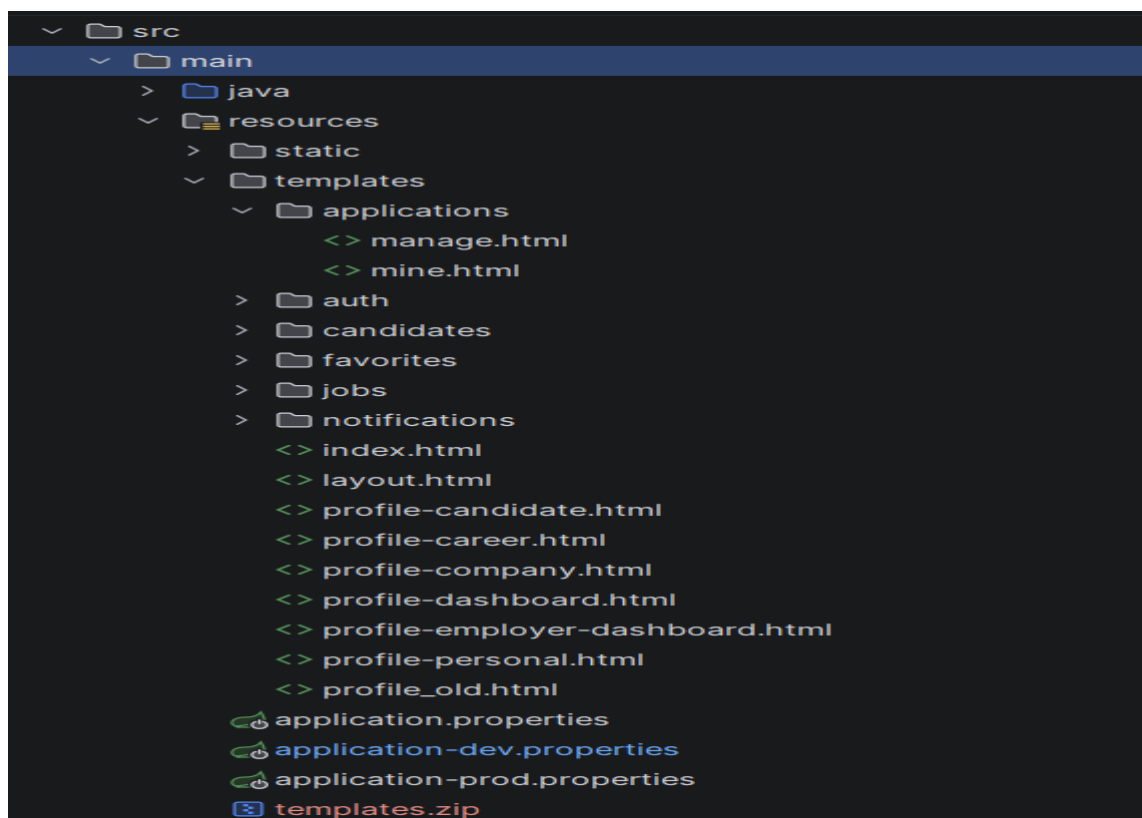
Παράλληλα, η χρήση κοινών μοντέλων ενισχύει τη συνεκτική παρουσίαση των δεδομένων στο περιβάλλον χρήστη, διασφαλίζοντας ότι κρίσιμες πληροφορίες παραμένουν διαθέσιμες ανεξαρτήτως της σελίδας στην οποία βρίσκεται ο χρήστης. Η συγκεκριμένη λειτουργία υποστηρίζει τη γενική πλοήγηση και συμβάλλει στη δημιουργία μιας ενιαίας, ομαλής και βελτιστοποιημένης εμπειρίας χρήσης σε όλα τα τμήματα της εφαρμογής.

#### **4.4.10 Οργάνωση έργου και δομή αρχείων της εφαρμογής**

Η δομή φακέλων που θα περιγραφεί ακολουθεί το καθιερωμένο πρότυπο έργων “Spring Boot”, διαχωρίζοντας με απόλυτη σαφήνεια τον κώδικα με τους πόρους παρουσίασης και ρυθμίσεων του συστήματος. Ο φάκελος “src/main/java” περιέχει με την σειρά του τον πυρήνα της εφαρμογής, δηλαδή όλα τα πακέτα και τις υπηρεσίες που περιγράφηκαν παραπάνω, καθώς και την σύνδεση με την βάση δεδομένων.

Αντίστοιχα, ο φάκελος “src/main/resources” φιλοξενεί τους μή εκτελέσιμους πόρους του συστήματος. Ο υποφάκελος “static” χρησιμοποιείται για στατικά αρχεία, όπως εικόνες και “CSS” κώδικα, τα οποία εξυπηρετούνται απευθείας από τον ίδιο τον διακομιστή. Επίσης, ο φάκελος “templates” εμπεριέχει όλα τα αρχεία “HTML”, τα οποία είναι υπεύθυνα για την αξιοποίηση και την παραγωγή δυναμικού περιεχομένου στο “frontend”.

Τέλος, αρχεία όπως “application.properties”, “application-dev.properties” κ.α. χρησιμοποιούνται για την ρύθμιση της εφαρμογής ανά περιβάλλον εκτέλεσης, με σκοπό την ευέλικτη διαχείριση παραμέτρων που αφορούν την σύνδεση στη βάση δεδομένων και των διαφόρων ρυθμίσεων ασφαλείας. Συνολική, η δομή που αποτυπώθηκε και θα παρουσιαστεί στο στιγμιότυπο παρακάτω, προάγει την καθαρή αρχιτεκτονική, την επεκτασιμότητα και την εύκολη συντήρηση του έργου στο σύστημα.



Εικόνα 37: Οργάνωση του έργου στο backend

#### 4.5 Διεπαφή χρήστη και Πρότυπα παρουσίασης (Frontend)

Η διεπαφή χρήστη της εφαρμογής DevJobs υλοποιείται με τη χρήση προτύπων (templates) Thymeleaf, τα οποία συνδυάζονται με HTML και CSS για την παραγωγή δυναμικών και διαδραστικών σελίδων. Τα πρότυπα οργανώνονται στον φάκελο *templates*, όπου αρχεία όπως *index.html*, *jobs.html*, *profile.html* και *applications.html* αντιστοιχούν στις βασικές λειτουργικές οθόνες της πλατφόρμας. Μέσω της χρήσης ενός βασικού layout, κοινά στοιχεία παρουσίασης, όπως η κεφαλίδα, το μενού πλοήγησης και τα μηνύματα ειδοποιήσεων, επαναχρησιμοποιούνται σε όλες τις σελίδες, προσφέροντας ενιαία και συνεκτική εμπειρία πλοήγησης.

Τα δεδομένα που εμφανίζονται στις σελίδες συνδέονται με τα αντίστοιχα μοντέλα του backend μέσω του μηχανισμού του Thymeleaf, επιτρέποντας την άμεση απόδοση λιστών αγγελιών, αιτήσεων, δεξιοτήτων και λοιπών πληροφοριών. Φόρμες όπως η εγγραφή, η σύνδεση, η συμπλήρωση προφίλ και η δημιουργία αγγελίας υλοποιούνται ως HTML φόρμες, οι οποίες αποστέλλουν τα δεδομένα στους αντίστοιχους controllers του συστήματος. Συνεπώς, επιτυγχάνεται στενή σύνδεση μεταξύ της επιχειρησιακής λογικής και της παρουσίασης, χωρίς την ανάγκη υιοθέτησης ξεχωριστής αρχιτεκτονικής τύπου Single Page Application (SPA).

Για τον σχεδιασμό και τη μορφοποίηση της διεπαφής χρήστη αξιοποιήθηκε το πλαίσιο Bootstrap, σε συνδυασμό με προσαρμοσμένο (custom) θέμα. Η χρήση του Bootstrap επέτρεψε την ταχεία ανάπτυξη επαναχρησιμοποιήσιμων στοιχείων διεπαφής, όπως φόρμες, κουμπιά, πίνακες και στοιχεία πλοήγησης, διασφαλίζοντας παράλληλα τη συμβατότητα με διαφορετικές αναλύσεις οθόνης. Το προσαρμοσμένο θέμα χρησιμοποιήθηκε για την τροποποίηση της χρωματικής παλέτας, της τυπογραφίας και των βασικών οπτικών στοιχείων, προσδίδοντας στην εφαρμογή πιο επαγγελματικό και εξατομικευμένο χαρακτήρα σε σχέση με το προεπιλεγμένο στυλ του Bootstrap.

Παράλληλα, η διάταξη των σελίδων έχει σχεδιαστεί ώστε να παραμένει λειτουργική και ευανάγνωστη σε διαφορετικά μεγέθη οθόνης, υποστηρίζοντας τόσο επιτραπέζιους υπολογιστές όσο

και φορητές συσκευές. Τα στατικά αρχεία της εφαρμογής (CSS, εικόνες και λοιπά βοηθητικά αρχεία) εξυπηρετούνται από τον φάκελο *static*, επιτρέποντας την ευέλικτη προσαρμογή της εμφάνισης της πλατφόρμας χωρίς την ανάγκη τροποποίησης του κώδικα του backend.

Συνολικά, η επιλογή των παραπάνω τεχνολογιών και προτύπων παρουσίασης συμβάλλει στη δημιουργία μιας καθαρής, συνεκτικής και φιλικής προς τον χρήστη διεπαφής. Η προσέγγιση αυτή ενισχύει την εμπειρία χρήσης (User Experience), διευκολύνει τη συντήρηση του frontend και επιτρέπει τη μελλοντική αισθητική ή λειτουργική επέκταση της εφαρμογής με ελάχιστο κόστος ανάπτυξης.

#### 4.6 Υλοποίηση ασφάλειας και διαχείρισης πρόσβασης

Η ασφάλεια της εφαρμογής DevJobs υλοποιείται με τη χρήση του πλαισίου Spring Security, το οποίο αναλαμβάνει τη διαδικασία αυθεντικοποίησης και εξουσιοδότησης των χρηστών. Η κλάση ρυθμίσεων SecurityConfig καθορίζει τους βασικούς κανόνες πρόσβασης, ορίζοντας ποιες διαδρομές (endpoints) είναι δημόσια προσβάσιμες (όπως οι σελίδες εγγραφής και σύνδεσης) και ποιες απαιτούν προηγουμένως επιτυχημένη πιστοποίηση χρήστη. Παράλληλα, γίνεται διαχωρισμός των δικαιωμάτων ανά ρόλο, διασφαλίζοντας ότι λειτουργίες όπως η δημιουργία αγγελιών ή η διαχείριση αιτήσεων είναι διαθέσιμες μόνο σε εξουσιοδοτημένους εργοδότες.

Η ανάκτηση των στοιχείων σύνδεσης από τη βάση δεδομένων υλοποιείται μέσω της υπηρεσίας CustomUserDetailsService, η οποία φορτώνει τον χρήστη με βάση τη διεύθυνση ηλεκτρονικού ταχυδρομείου. Σε αυτό το επίπεδο, γίνεται και η σύνδεση με τα αντίστοιχα δικαιώματα (ρόλους) που είναι αποθηκευμένα στη βάση, ώστε το σύστημα να γνωρίζει αν ο χρήστης θα συμπεριφερθεί ως υποψήφιος ή ως εργοδότης. Οι κωδικοί πρόσβασης αποθηκεύονται κρυπτογραφημένοι, εξασφαλίζοντας ότι σε περίπτωση διαρροής δεδομένων δεν είναι δυνατή η απευθείας ανάγνωσή τους.

Επιπλέον, υλοποιείται μηχανισμός ελέγχου πληρότητας προφίλ, μέσω του ProfileCompletionInterceptor. Ο μηχανισμός αυτός παρεμβάλλεται πριν από την εξυπηρέτηση συγκεκριμένων αιτημάτων και ελέγχει αν ο χρήστης έχει συμπληρώσει τα απαιτούμενα στοιχεία στο προφίλ του (προσωπικά δεδομένα, επαγγελματικές προτιμήσεις, δεξιότητες). Σε περίπτωση που λείπουν κρίσιμες πληροφορίες, ο χρήστης ανακατευθύνεται στις αντίστοιχες φόρμες συμπλήρωσης, ενισχύοντας έτσι τόσο την ποιότητα των δεδομένων όσο και την αξιοπιστία της διαδικασίας αντιστοίχισης.

Τέλος, η εφαρμογή περιορίζει την πρόσβαση σε ευαίσθητα δεδομένα (όπως λίστες αιτήσεων ή στοιχεία άλλων χρηστών) μόνο σε χρήστες που διαθέτουν τα κατάλληλα δικαιώματα. Έτσι, η ασφάλεια και η ιδιωτικότητα των χρηστών ενσωματώνονται ως βασικά χαρακτηριστικά του συστήματος και όχι ως δευτερεύουσα προσθήκη.

#### 4.7 Ροή δεδομένων στο σύστημα

Η ροή δεδομένων στο σύστημα DevJobs ακολουθεί μια σαφώς ορισμένη και δομημένη αρχιτεκτονική, η οποία βασίζεται στη διάκριση των επιπέδων παρουσίασης, επιχειρησιακής λογικής και πρόσβασης δεδομένων. Η προσέγγιση αυτή εξασφαλίζει την ομαλή διαχείριση των αιτημάτων των χρηστών, την ακεραιότητα των δεδομένων και τη σωστή επικοινωνία μεταξύ των επιμέρους υποσυστημάτων.

Σε επίπεδο backend, κάθε ενέργεια του χρήστη ξεκινά με την αποστολή ενός αιτήματος προς τον αντίστοιχο controller. Οι controllers λειτουργούν ως το πρώτο σημείο εισόδου στο σύστημα, λαμβάνοντας τα δεδομένα από το περιβάλλον χρήστη και εκτελώντας βασικούς ελέγχους εγκυρότητας. Στη συνέχεια, τα αιτήματα προωθούνται στο επίπεδο των services, όπου υλοποιείται η κύρια επιχειρησιακή λογική της εφαρμογής. Σε αυτό το στάδιο πραγματοποιούνται οι απαραίτητοι έλεγχοι, οι υπολογισμοί και οι αποφάσεις που καθορίζουν τη συμπεριφορά του συστήματος.

Τα services αλληλεπιδρούν με το επίπεδο πρόσβασης δεδομένων μέσω των repositories, τα οποία είναι υπεύθυνα για την επικοινωνία με τη βάση δεδομένων. Μέσω των repositories εκτελούνται οι απαραίτητες λειτουργίες αποθήκευσης, ανάκτησης και ενημέρωσης των δεδομένων, διασφαλίζοντας ότι όλες οι αλλαγές καταγράφονται με συνέπεια στη βάση δεδομένων. Με την ολοκλήρωση της διαδικασίας, τα αποτελέσματα επιστρέφονται αντίστροφα προς τα services και στη συνέχεια προς τους controllers, οι οποίοι διαμορφώνουν την τελική απόκριση προς τον χρήστη.

Σε επίπεδο frontend, η εφαρμογή αξιοποιεί πρότυπα Thymeleaf για την παραγωγή δυναμικού περιεχομένου. Τα δεδομένα που αποστέλλονται από τους controllers ενσωματώνονται στα πρότυπα αυτά, επιτρέποντας την παρουσίαση ενημερωμένων πληροφοριών χωρίς την ανάγκη άμεσης αλληλεπίδρασης του χρήστη με τη βάση δεδομένων. Με αυτόν τον τρόπο επιτυγχάνεται σαφής διαχωρισμός μεταξύ λογικής παρουσίασης και επιχειρησιακής λογικής.

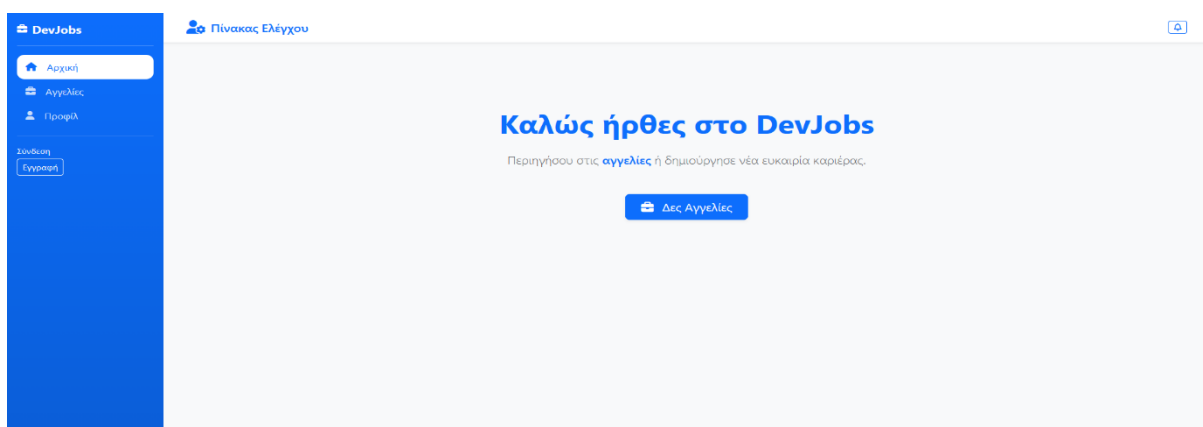
Η συγκεκριμένη ροή δεδομένων διασφαλίζει ότι κάθε ενέργεια του χρήστη επεξεργάζεται με ελεγχόμενο και προβλέψιμο τρόπο, μειώνοντας την πιθανότητα σφαλμάτων και διευκολύνοντας τη συντήρηση και επέκταση του συστήματος. Παράλληλα, η καθαρή οργάνωση της ροής δεδομένων ενισχύει την αναγνωσιμότητα του κώδικα και επιτρέπει την ευκολότερη ενσωμάτωση νέων λειτουργιών στο μέλλον.

## 5 Παραδείγματα με screenshots από την χρήση του συστήματος

Αυτή η ενότητα παραθέτει οπτικά παραδείγματα της χρήσης και αλληλεπίδρασης των χρηστών με το περιβάλλον εύρεσης εργασίας. Ταυτόχρονα, εξηγείται η λειτουργικότητα της διεπαφής στο εκάστοτε παράδειγμα.

### 5.1 Index

Η αρχική οθόνη λειτουργεί ως το πρώτο σημείο επαφής του χρήστη με το σύστημα. Παρέχει βασικές πληροφορίες για την πλατφόρμα, καθώς και δυνατότητα πλοήγησης στις ενότητες εγγραφής και σύνδεσης.

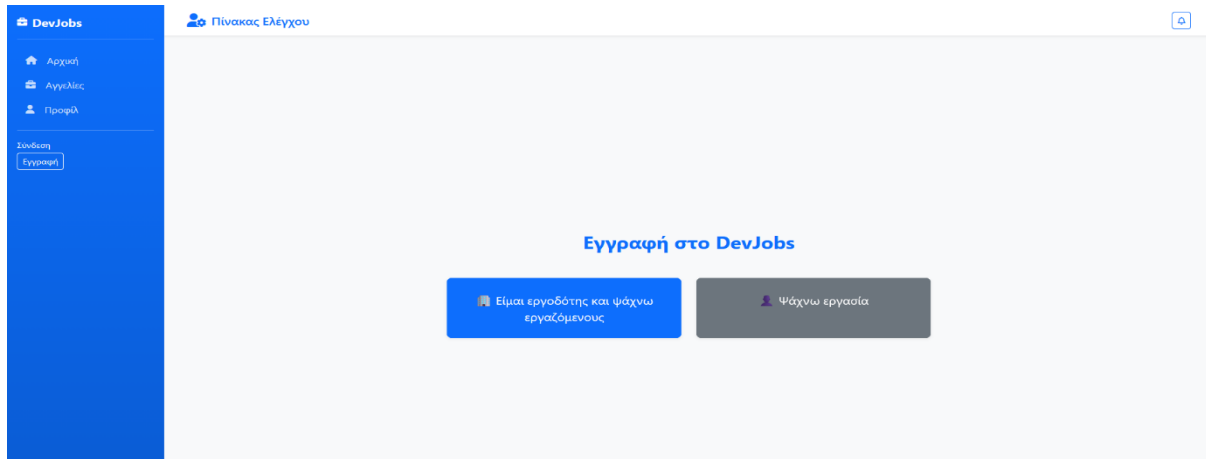


Εικόνα 38: Πρώτη διεπαφή χρήστη-συστήματος



## 5.2 Επιλογή ρόλου πριν την εγγραφή

Πριν την εγγραφή, ο χρήστης καλείται να επιλέξει αν θα συνεχίσει ως υποψήφιος ή ως εργοδότης, ώστε να ενεργοποιηθεί η κατάλληλη ροή δημιουργίας προφίλ.

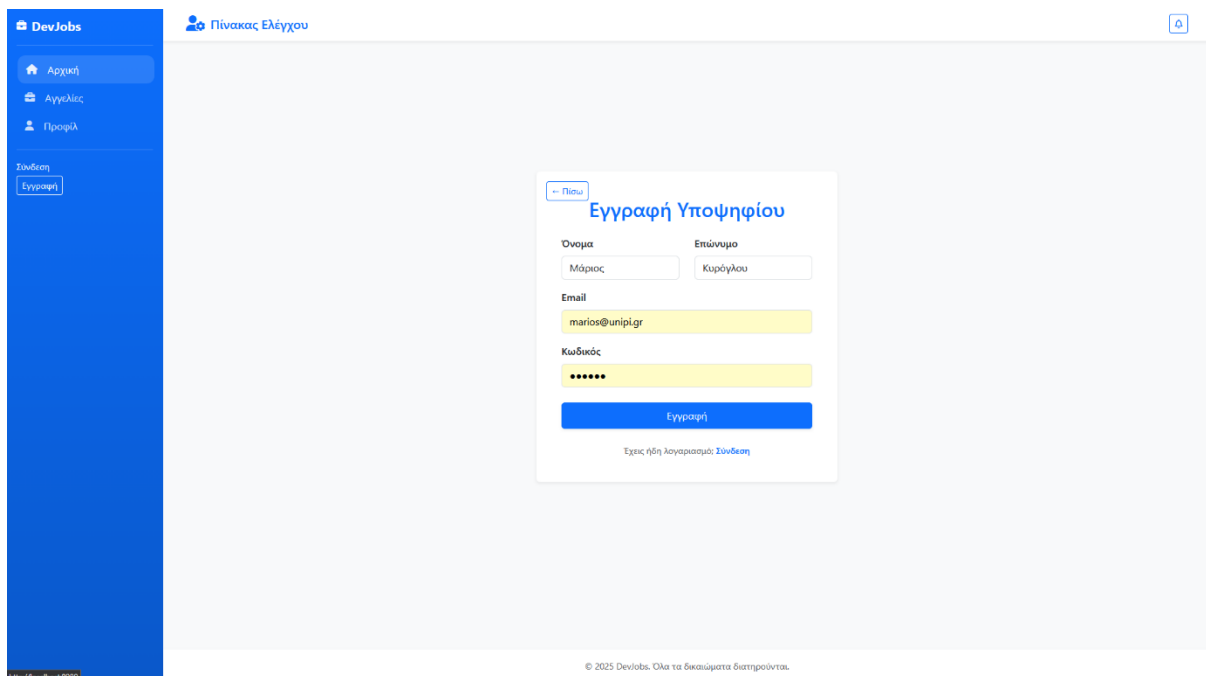


Εικόνα 39: Διαχωρισμός ρόλων κατά την εγγραφή

## 5.3 Λειτουργίες υποψήφιου εργαζομένου

### 5.3.1 Εγγραφή

Η φόρμα εγγραφής συγκεντρώνει τα βασικά στοιχεία ταυτοποίησης, όπως email και κωδικό πρόσβασης. Το σύστημα επαληθεύει τα δεδομένα και δημιουργεί το αρχικό προφίλ.



Εικόνα 40: Φόρμα Εγγραφής

### 5.3.2 Συμπλήρωση προσωπικών δεδομένων του χρήστη

Ο υποψήφιος καταχωρεί στοιχεία όπως όνομα, ηλικία, τοποθεσία και βασικές πληροφορίες που απαιτούνται για την ολοκλήρωση του προσωπικού του προφίλ.

ΒΗΜΑ 1 ΑΠΟ 2

## Δημιούργησε το προσωπικό σου προφίλ

Λίγα ακόμη στοιχεία για να ολοκληρώσουμε την εγγραφή σου.

1

2

Προσωπικό προφίλ

Προφίλ καριέρας

Όνομα

Επώνυμο

Μάριος

Κυρόγλου

Ημερομηνία γέννησης

Φύλο

03 / 07 / 2003

Ανδρας

Φωτογραφία προφίλ

Πόλη / Περιοχή

Browse... profile\_pic.png

Αθήνα, Αττική

Υποστηρίζονται JPG/PNG/WebP έως 5MB.

Πορτφόλιο / Ιστότοπος

Τηλέφωνο

https://...

6912345678

s://www.github.com/marioskyr

LinkedIn

GitHub

//www.linkedin.com/marioskyr

https://...

Συνέχεια στο προφίλ καριέρας

Εικόνα 41: Εισαγωγή προσωπικών δεδομένων του χρήστη

### 5.3.3 Εισαγωγή προτιμήσεων στο προφίλ καριέρας

Σε αυτή την ενότητα ο χρήστης ορίζει το επαγγελματικό του προφίλ. Στοιχεία που ζητούνται είναι η περιοχή ενδιαφέροντος εργασίας, ο τύπος της εργασίας (τηλεργασία, δια ζώσης ή ευέλικτο), το είδος της εργασίας ως προς το ωράριο (πλήρης/ μερική απασχόληση, ελεύθερος επαγγελματίας ή πρακτική άσκηση) καθώς και επίπεδο εκπαίδευσης. Το επίπεδο εκπαίδευσης είναι ένα ευέλικτο πεδίο, στο οποίο μπορούν να προστεθούν δυναμικά πτυχία και σπουδές κάθε είδους, είτε είναι ολοκληρωμένα είτε ακόμη ενεργά, μαζί με διάφορες πληροφορίες όπως το όνομα του ιδρύματος και ο τίτλος σπουδών και πολλά άλλα.

ΒΗΜΑ 2 ΑΠΟ 2

### Προτιμήσεις Καριέρας

Ολοκληρώστε τις επαγγελματικές σου προτιμήσεις για να ξεκινήσεις με προσωποποιημένες προτάσεις. Αφού αποθηκεύσεις, θα μεταφερθείς στο προσωπικό σου dashboard.

1 Προσωπικό προφίλ 2 Προφίλ καριέρας

Περιοχή Εργασίας

Αθήνα, Αττική

Τύπος Εργασίας

Τηλεργασία Υβριδική εργασία Στο γραφείο **Χωρίς προτίμηση**

Είδος Απασχόλησης

**Πλήρης απασχόληση** Μερική απασχόληση Ελεύθερος επαγγελματίας Πρακτική άσκηση

Εκπαίδευση

Προσθέστε προγράμματα σπουδών, έτη και ιδρύμα.

Είδος Σπουδών Τύπος Ιδρύματος Διαγραφή

Προπτυχιακό ΔΕΙ

Πανεπιστήμιο / Ίδρυμα Τμήμα

ΠΑΠΕΙ Πληροφορική

Έτος Εναρξης Έτος Ολοκλήρωσης ☐ Τρέχουσα φοίτηση

2021 2025

Εικόνα 42: Προφίλ καριέρας (Μέρος 1)

### 5.3.4 Εισαγωγή περαιτέρω δεξιοτήτων ή πιθανών πιστοποιήσεων

Παρακάτω στην σελίδα του ορισμού του επαγγελματικού προφίλ μπορούν να προστεθούν επιπλέον πιστοποιήσεις ή σεμινάρια που μπορεί να έχει παρακολουθήσει ο υποψήφιος. Τέλος, το παράθυρο των δεξιοτήτων είναι ίσως το κυριότερο σημείο της σελίδας, καθώς συμβάλει στην κεντρική ιδέα της εφαρμογής, με την αντιστοίχιση των δεξιοτήτων με τις αγγελίες που ταιριάζουν.

Πιστοποιήσεις / Σεμινάρια

Καταχωρήστε τις πιστοποιήσεις ή τα σεμινάρια σας.

+ Προσθήκη

Τίτλος Φορέας / Οργανισμός Διαγραφή

AWS Cloud Practitioner AWS Training

Ημερομηνία Διεξαγωγής

01 / 01 / 2024 ☐ Σε εξέλιξη

Δεξιότητες

Αναζήτηση δεξιοτήτων...

Επιλέξτε δεξιότητες και ορίστε το επιθυμητό επίπεδο (1-5). + Προσθήκη νέας δεξιοτήτας

☐ Docker Επίπεδο: 3 – Πολύ καλή γνώση

☐ Kubernetes Επίπεδο: 3 – Πολύ καλή γνώση

☐ AWS Επίπεδο: 3 – Πολύ καλή γνώση

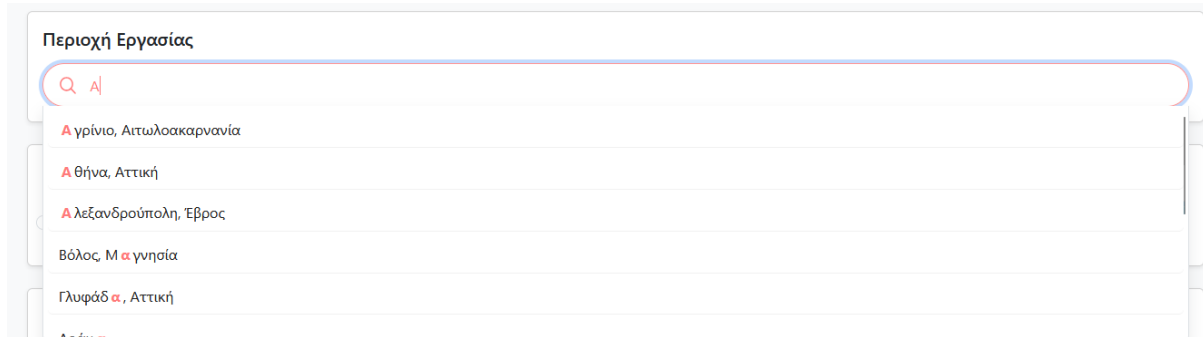
☐ Azure Επίπεδο: 3 – Πολύ καλή γνώση

☐ Google Cloud Επίπεδο: 3 – Πολύ καλή γνώση

Εικόνα 43: Προφίλ καριέρας (Μέρος 2)

### 5.3.5 Autocomplete

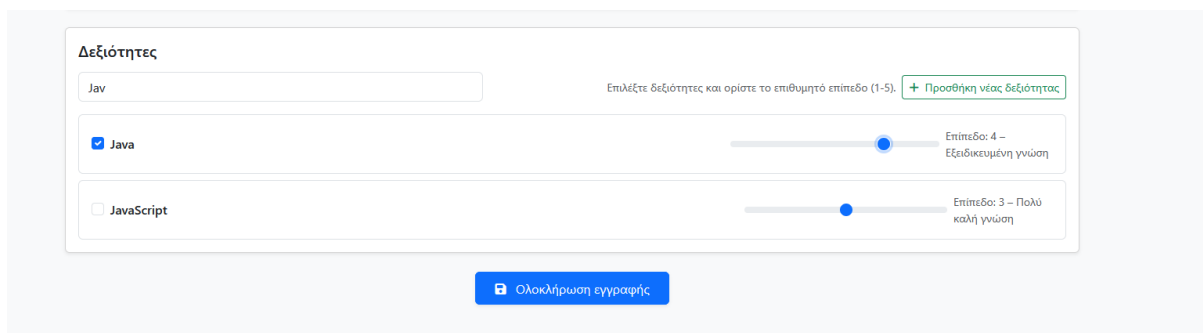
Στο πεδίο περιοχής εργασίας, υπάρχει λειτουργία αυτοσυμπλήρωσης (autocomplete) διευκολύνοντας τόσο τον τελικό χρήστη όσο και την αντιστοίχιση περιοχών με τις αγγελίες στο backend.



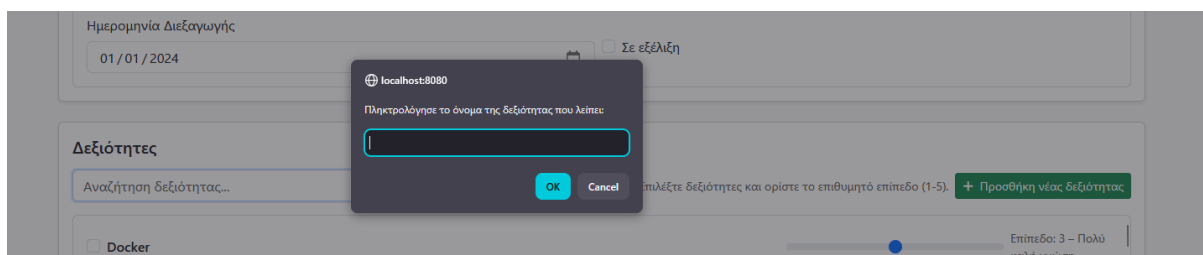
Εικόνα 44: Πρόταση πιθανών περιοχών αναζήτησης εργασίας

### 5.3.6 Προσθήκη δεξιοτήτων και δυνατότητα βαθμολογίας επάρκειας

Ο υποψήφιος χρήστης καλείται να προσθέσει δεξιότητες που κατέχει, καθώς και το επίπεδο το οποίο πιστεύει ότι κατέχει για την κάθε μια. Η λίστα των δεξιοτήτων περιέχει δεκάδες από τις δημοφιλέστερες δεξιότητες που απαιτούνται στην αγορά εργασίας της πληροφορικής (γλώσσες προγραμματισμού, βάσεις δεδομένων, πλατφόρμες, λογισμικά κα). Ωστόσο, ο χρήστης έχει την δυνατότητα να προσθέσει κάποια δικιά του δεξιότητα, εφόσον δεν την βρει στην ήδη υπάρχουσα λίστα.



Εικόνα 45: Ενσωμάτωση πιθανών δεξιοτήτων του χρήστη κατά την διαμόρφωση του προφίλ του



Εικόνα 46: Προσθήκη προσωπικής δεξιότητας

### 5.3.7 Ανασκόπηση προφίλ (Profile Dashboard)

Αφού ο υποψήφιος χρήστης συμπληρώσει τα στοιχεία του προσωπικού προφίλ και του προφίλ καριέρας, ανακατευθύνεται στο profile dashboard, το οποίο έχει πρόσβαση και από το κέντρο ελέγχου (αριστερό μέρος της σελίδας). Στο profile dashboard ο χρήστης μπορεί να δει όλα του τα στοιχεία, καθώς και να προχωρήσει στην επεξεργασία αυτών.

The screenshot displays the Profile Dashboard for Marios Kyroglou. The interface includes a sidebar with navigation links: DevJobs, Αρχική, Αγγελίες, Προφίλ (active), Αιτήσεις, Αισθημένα, and Ειδοποιήσεις. The main header shows the user's name 'Μάριος Κυρόγλου' and a 'Πίνακας Ελέγχου' (Dashboard) link. The main content area is divided into several sections:

- Στοιχεία επικοινωνίας** (Contact Information):

|           |                                                                                               |
|-----------|-----------------------------------------------------------------------------------------------|
| Τηλέφωνο  | 6912345678                                                                                    |
| Email     | marios@unipi.gr                                                                               |
| LinkedIn  | <a href="https://www.linkedin.com/marioskyroglou">https://www.linkedin.com/marioskyroglou</a> |
| GitHub    | <a href="https://www.github.com/marioskyr">https://www.github.com/marioskyr</a>               |
| Πορτοφόλι | —                                                                                             |
- Προτιμήσεις καριέρας** (Career Preferences):

Επιθυμητός τρόπος εργασίας: Χωρίς προτίμηση  
Τύπος απασχόλησης: Πλήρης απασχόληση  
Προτιμώμενη τοποθεσία: Αθήνα, Αττική  
Εκπαίδευση:  
• Προπτυχιακό - ΠΑΤΕΙ (Γίνουροφορεία) — 2021 - 2025  
Σύντομη περιγραφή:  
Ευρεία γνώση σε java και spring boot.
- Πιστοποιήσεις και σεμινάρια** (Certifications and Seminars):

AWS Cloud Practitioner  
Φορέας: AWS Training  
Κατάσταση: 01/01/2024
- Δεξιότητες** (Skills):

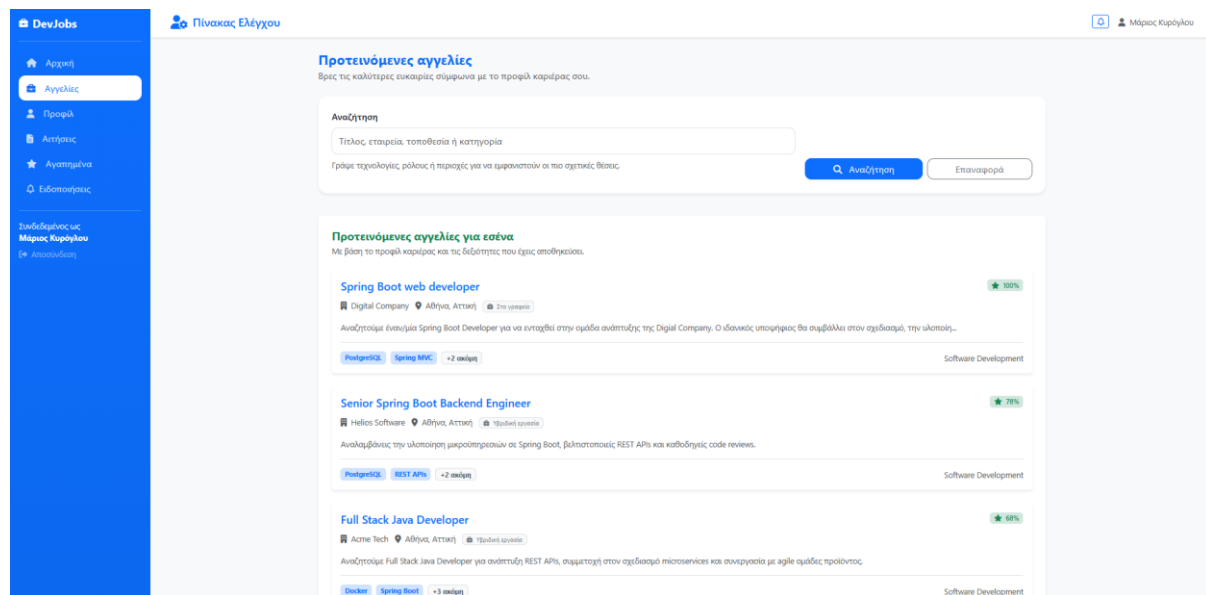
|             |     |
|-------------|-----|
| PostgreSQL  | 5%  |
| Spring Boot | 10% |
| Spring MVC  | 20% |
| Java        | 40% |

At the bottom, there are three buttons: 'Επεξεργασία προσωπικών στοιχείων', 'Ενημέρωση προφίλ καριέρας', and 'Δες τις Προτεινόμενες Αγγελίες σου'.

Εικόνα 47: Profile Dashboard

### 5.3.8 Προτεινόμενες αγγελίες βάσει συσχέτισης προσωπικών-ζητουμένων δεξιοτήτων

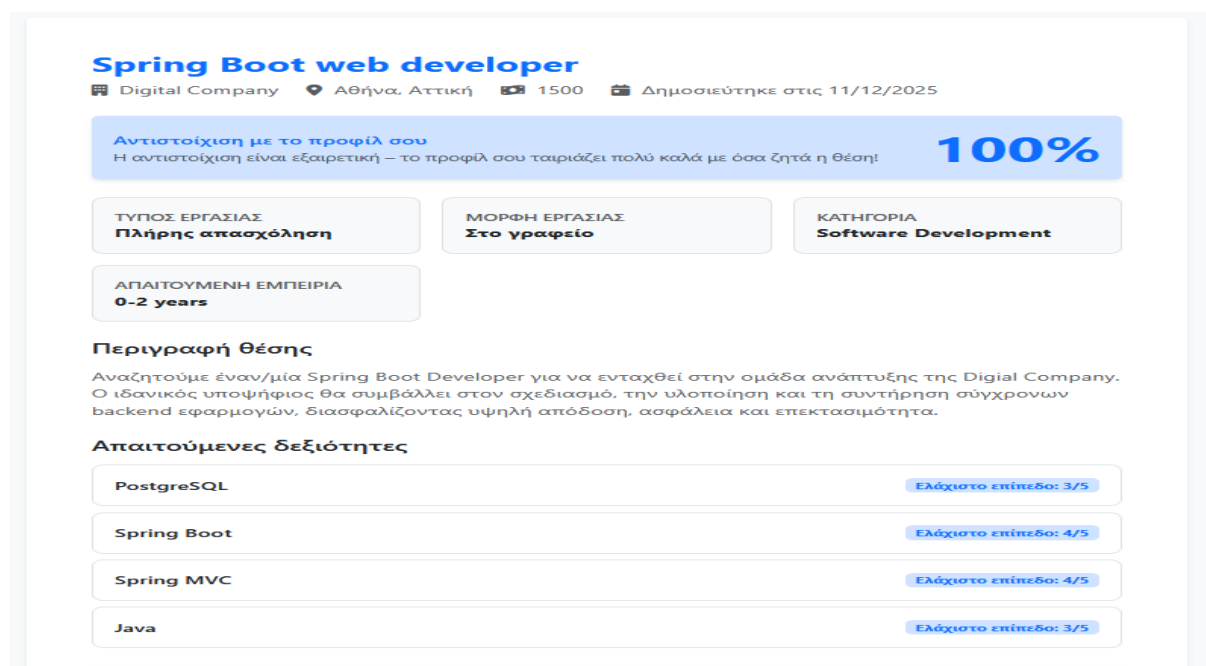
Με την συμπλήρωση των απαραίτητων στοιχείων, ο χρήστης έχει πρόσβαση στη σελίδα των αγγελιών. Εκεί μπορεί να βρει το πλαίσιο "Προτεινόμενες αγγελίες για εσένα" όπου με βάση του Matching Service που εξηγήθηκε στην ενότητα 3 εμφανίζονται οι αγγελίες με το μεγαλύτερο ποσοστό αντιστοίχισης.



Εικόνα 48: Προτεινόμενες Εργασίες

### 5.3.9 Προβολή αγγελίας

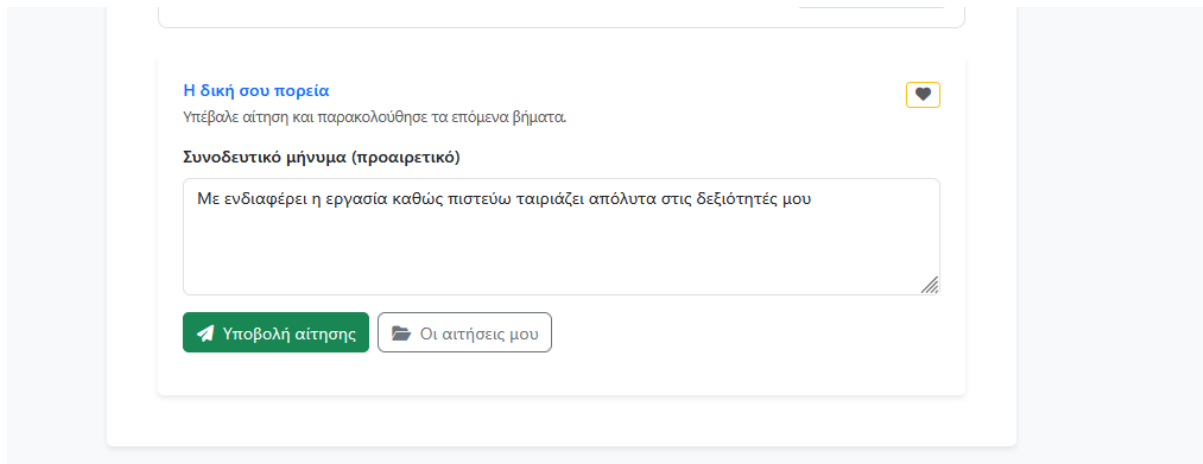
Ο υποψήφιος κάνοντας κλικ σε κάθε αγγελία έχει τη δυνατότητα να δει περισσότερες λεπτομέρειες για την εργασία μαζί με το ποσοστό αντιστοίχισης.



Εικόνα 49: Προβολή θέσης εργασίας

### 5.3.10 Αίτηση

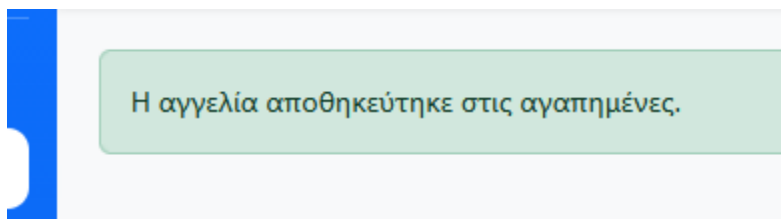
Στη συνέχεια, ο χρήστης έχει τη δυνατότητα να υποβάλει αίτηση για την αγγελία προς τον εργοδότη, συνοδευόμενη από ένα σύντομο μήνυμα.



The screenshot shows a web form for submitting a job application. At the top, it says "Η δική σου πορεία" (Your journey) with a heart icon and a subtext "Υπέβαλε αίτηση και παρακολούθησε τα επόμενα βήματα." (Submitted application and follow the next steps). Below this is a section titled "Συνοδευτικό μήνυμα (προαιρετικό)" (Supporting message (optional)). Inside this section is a text area containing the message: "Με ενδιαφέρει η εργασία καθώς πιστεύω ταιριάζει απόλυτα στις δεξιότητές μου" (I am interested in the job as I believe it fits perfectly with my skills). At the bottom of the form are two buttons: a green button labeled "Υποβολή αίτησης" (Submit application) and a white button labeled "Οι αιτήσεις μου" (My applications).

Εικόνα 50: Υποβολή αίτησης και συνοδευτικού μηνύματος

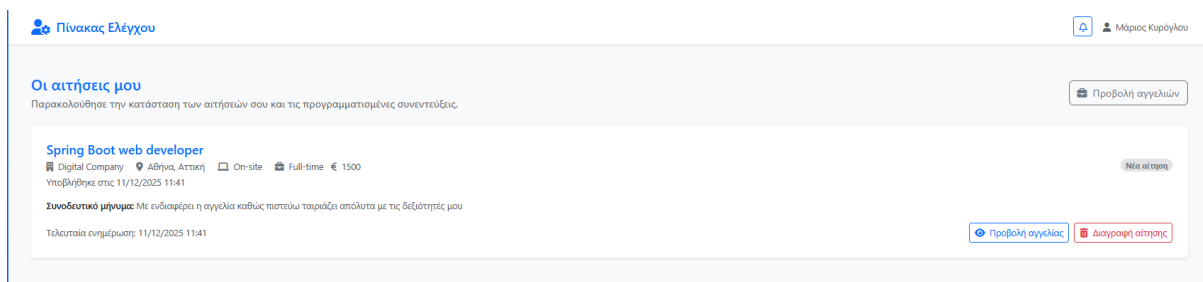
### 5.3.11 Προσθήκη αγγελίας εργασίας στα αγαπημένα



Εικόνα 51: Προσθήκη στα αγαπημένα

### 5.3.12 Αιτήσεις που έχουν υποβληθεί

Στη σελίδα "Οι αιτήσεις μου" ο χρήστης μπορεί να δει τις αιτήσεις που έχει υποβάλει καθώς και να τις επεξεργαστεί ή να τις διαγράψει.



The screenshot shows the "Οι αιτήσεις μου" (My Applications) page. At the top, it says "Πίνακας Ελέγχου" (Dashboard) and "Μάριος Κυρόγλου". Below the title, it says "Παρακολουθείς την κατάσταση των αιτήσεών σου και τις προγραμματισμένες συνεντεύξεις." (You are following the status of your applications and the scheduled interviews). There is a button labeled "Προβολή αγγελιών" (View jobs). The main content area shows a job listing for "Spring Boot web developer" at "Digital Company" in "Αθήνα, Αττική". It includes details like "On-site", "Full-time", and "€ 1500". Below the job listing is a section titled "Συνοδευτικό μήνυμα" (Supporting message) with the same message as in the previous screenshot: "Με ενδιαφέρει η αγγελία καθώς πιστεύω ταιριάζει απόλυτα με τις δεξιότητές μου". At the bottom of the job listing are two buttons: "Προβολή αγγελίας" (View job) and "Διαγραφή αίτησης" (Delete application).

Εικόνα 52: Ατομικές αιτήσεις

### 5.3.13 Ειδοποιήσεις σχετικά με την πορεία των αιτήσεων

Η εφαρμογή διαθέτει ειδοποιήσεις, από τις οποίες ο χρήστης ενημερώνεται για την πορεία των αιτήσεων του.

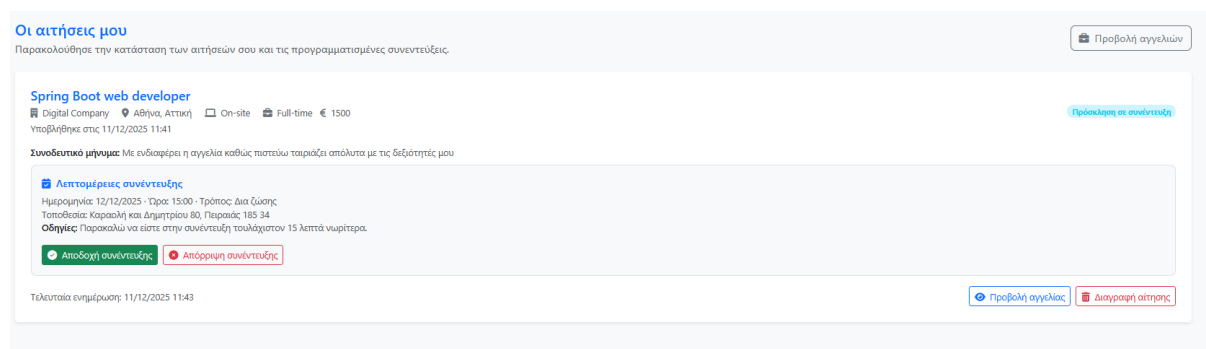




Εικόνα 53: Notifications

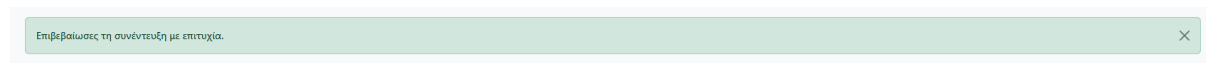
### 5.3.14 Αποδοχή – Απόρριψη συνέντευξης

Εφόσον ο εργοδότης προχωρήσει στην αποδοχή της αίτησης, ο υποψήφιος καλείται να αποδεχτεί τη συνέντευξη συνοδευόμενη από πληροφορίες σχετικά με αυτή (ημερομηνία, ώρα, τοποθεσία κλπ).



Εικόνα 54: Διαδικασία αποδοχής ή απόρριψης προτεινόμενης συνέντευξης

### 5.3.15 Αποδοχή Συνέντευξης (Μήνυμα)



Εικόνα 55: Επιβεβαιωμένη συνέντευξη

## 5.4 Εργοδότης

### 5.4.1 Δημιουργία προφίλ εταιρίας

Κατά την εγγραφή, αν ο χρήστης επιλέξει τον ρόλο του εργοδότη, θα εμφανιστεί η φόρμα δημιουργίας εταιρικού προφίλ, όπου θα πρέπει να συμπληρώσει τα απαραίτητα στοιχεία για την εταιρεία που εκπροσωπεί.

ΚΑΛΩΣΟΡΙΣΕΙΣ  
**Εταιρικό προφίλ**

Συμπλήρωσε τα στοιχεία της εταιρείας σου ώστε οι υποψήφιοι να την γνωρίσουν καλύτερα.  
Μετά την αποθήκευση θα μεταφερθείς στο dashboard για να δεις τις επόμενες ενέργειες.

1

2

Στοιχεία εταιρείας

Δημιούργησε σελίδα

**Ταυτότητα εταιρείας**

Επωνυμία

Digital Company

Σύντομο μόντο

Π.χ. Δημιουργούμε digital εμπειρίες

Κλάδος

Λογισμικό & Τεχνολογία

Μέγεθος

11-50 εργαζόμενοι

Έτος ίδρυσης

2010

**Στοιχεία επικοινωνίας**

Έδρα

Αθήνα, Αττική

Τηλέφωνο επικοινωνίας

2101234567

Ιστότοπος

https://www.digitalcompany.com

Σελίδα LinkedIn

https://www.linkedin.com/digitalcompany

**Παρουσίαση**

Περιγραφή εταιρείας

Η Digital Company είναι μια σύγχρονη εταιρία λογισμικού που ειδικεύεται στην ανάπτυξη ευέλικτων και καινοτόμων ψηφιακών λύσεων. Προσφέρει εφαρμογές υψηλής ποιότητας, προσαρμοσμένες στις ανάγκες των πελατών, με στόχο τη βελτίωση της παραγωγικότητας και τη διευκόλυνση ψηφιακού μετασχηματισμού.

Εικόνα 56: Εισαγωγή πληροφοριών εταιρείας

#### 5.4.2 Λογότυπο εταιρίας

Εν συνεχεία, υπάρχει η δυνατότητα προσθήκης του λογοτύπου της εταιρίας ως εικόνα προφίλ.

**Λογότυπο**

Δεν έχει προστεθεί λογότυπο ακόμη.

Ανέβασε νέο λογότυπο (προαιρετικό)

Browse... company\_logo.png

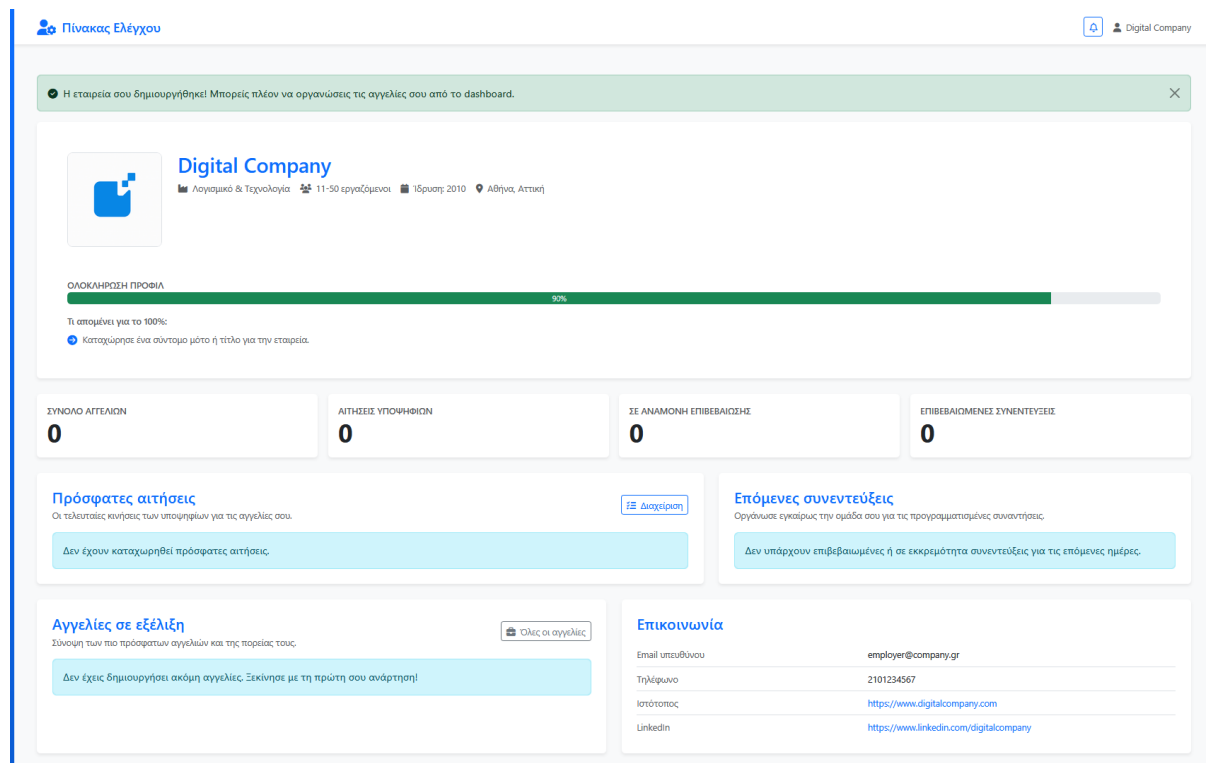
Υποστηρίζονται JPG/PNG/WebP μέχρι 5MB.

Ολοκλήρωση ρύθμισης

Εικόνα 57: Εισαγωγή εικόνας εταιρίας

#### 5.4.3 Επισκόπηση προφίλ εταιρίας

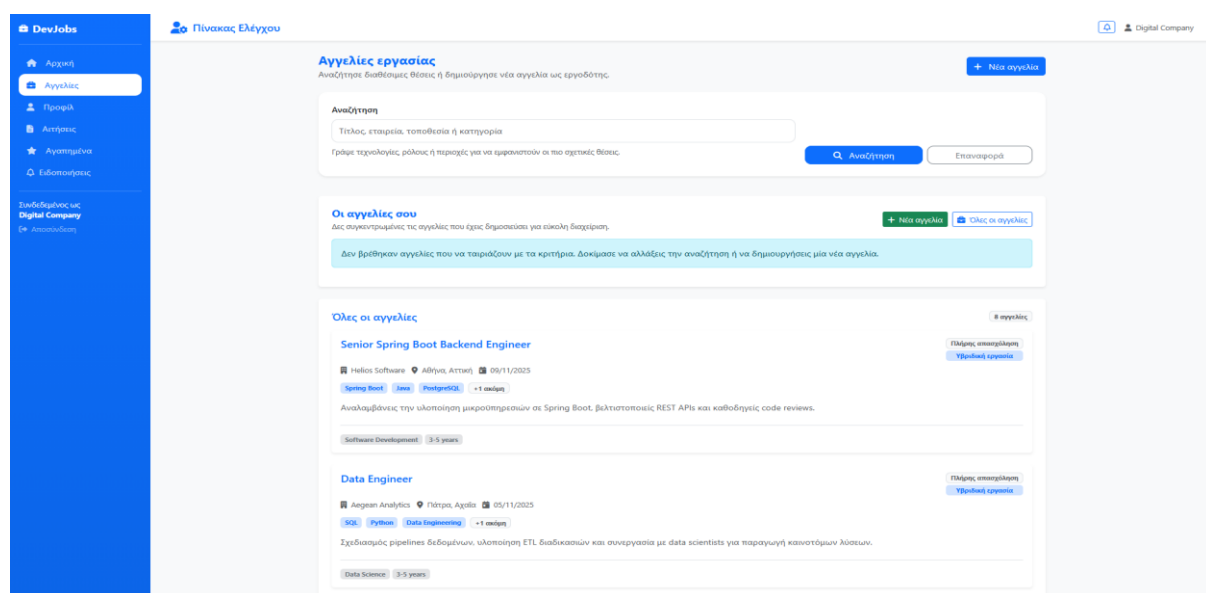
Εδώ παρουσιάζεται το αντίστοιχο profile dashboard της εταιρίας με τις πληροφορίες συγκεντρωμένες σε μια σελίδα.



Εικόνα 58: Εταιρικό προφίλ

### 5.4.4 Ενεργές αγγελίες επιχείρησης και δυνατότητα επεξεργασίας υπαρχόντων

Η σελίδα “Αγγελίες” του εργοδότη έχει διαφορετική παρουσίαση σε σχέση με τη αντίστοιχη σελίδα του υποψηφίου. Όπως φαίνεται στην παρακάτω εικόνα, υπάρχει ένα ξεχωριστό πλαίσιο “Οι αγγελίες μου” με τις αγγελίες που έχει δημιουργήσει ο χρήστης - εργοδότης.



Εικόνα 59: Αγγελίες επιχείρησης

### 5.4.5 Υποβολή νέας αγγελίας εργασίας

Παρακάτω απεικονίζεται η φόρμα δημιουργίας νέας αγγελίας, ζητώντας αντίστοιχα στοιχεία με αυτά που ζητούνταν από τον υποψήφιο, όπως τύπος και μορφή εργασίας, τοποθεσία αλλά και προαπαιτούμενες δεξιότητες.

**Νέα αγγελία** [← Πίσω στις αγγελίες](#)

Συμπλήρωσε τις βασικές πληροφορίες για την αγγελία σου.

**Τίτλος \***

Spring Boot web developer

**Περιγραφή \***

Αναζητούμε έναν/μία Spring Boot Developer για να ενταχθεί στην ομάδα ανάπτυξης της Digital Company. Ο ιδανικός υποψήφιος θα συμβάλει στον σχεδιασμό, την υλοποίηση και τη συντήρηση σύγχρονων **backend** εφαρμογών, διασφαλίζοντας υψηλή απόδοση, ασφάλεια και επεκτασιμότητα.

**Τύπος απασχόλησης \*** **Κατηγορία**

Επιλέξτε Software Development

**Μορφή εργασίας \*** **Απαιτούμενη εμπειρία**

Επιλέξτε 0-2 years

**Τοποθεσία** **Εύρος μισθού**

Αθήνα, Αττική Π.χ. 1800-2300€ μικτά

**Απαιτούμενες δεξιότητες \***

Επιλέξτε δεξιότητες και ορίστε το επιθυμητό επίπεδο (1-5).

[+ Προσθήκη νέας δεξιότητας](#)

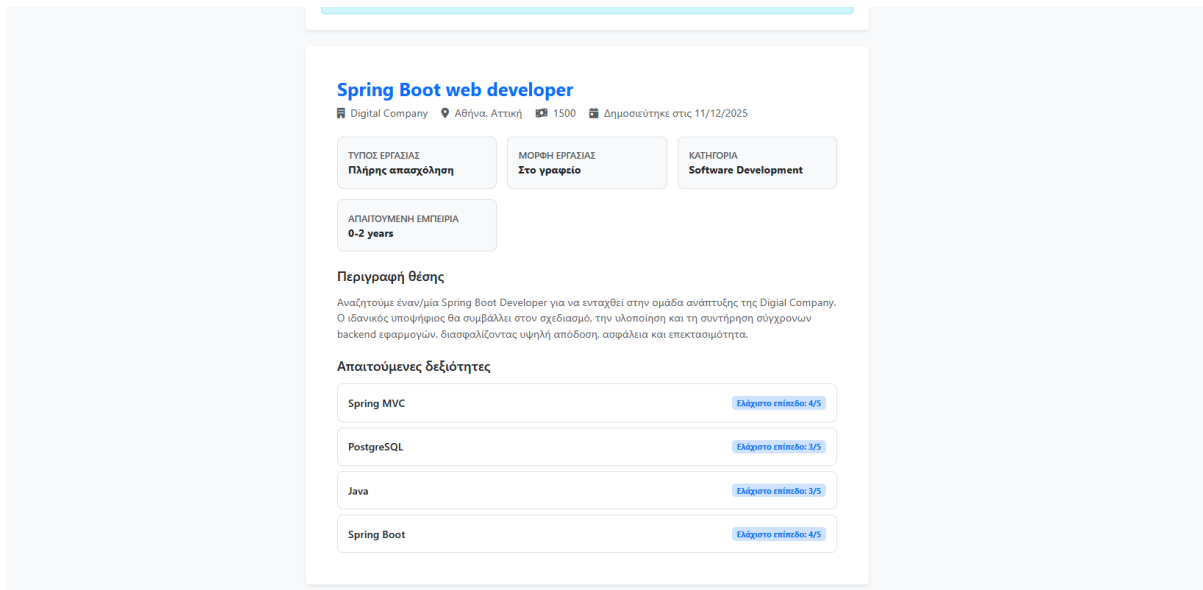
☒ Spring Boot  Επίπεδο: 4 – Εξειδικευμένη γνώση

☒ Spring MVC  Επίπεδο: 4 – Εξειδικευμένη γνώση

Εικόνα 60: Υποβολή αναζήτησης νέου υπαλλήλου

### 5.4.6 Επισκόπηση νέας αγγελίας

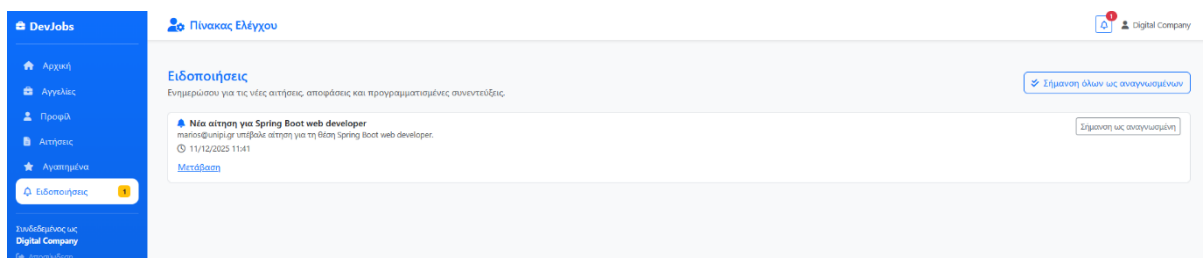
Εφόσον δημιουργηθεί μία νέα αγγελία, παρουσιάζεται στον χρήστη το πλαίσιο επισκόπησης της αγγελίας με συγκεντρωμένες όλες τις απαραίτητες πληροφορίες της.



Εικόνα 61: Προβολή νέας εργασίας

#### 5.4.7 Ειδοποίηση για υποβολή νέας αίτησης από υποψήφιο εργαζόμενο

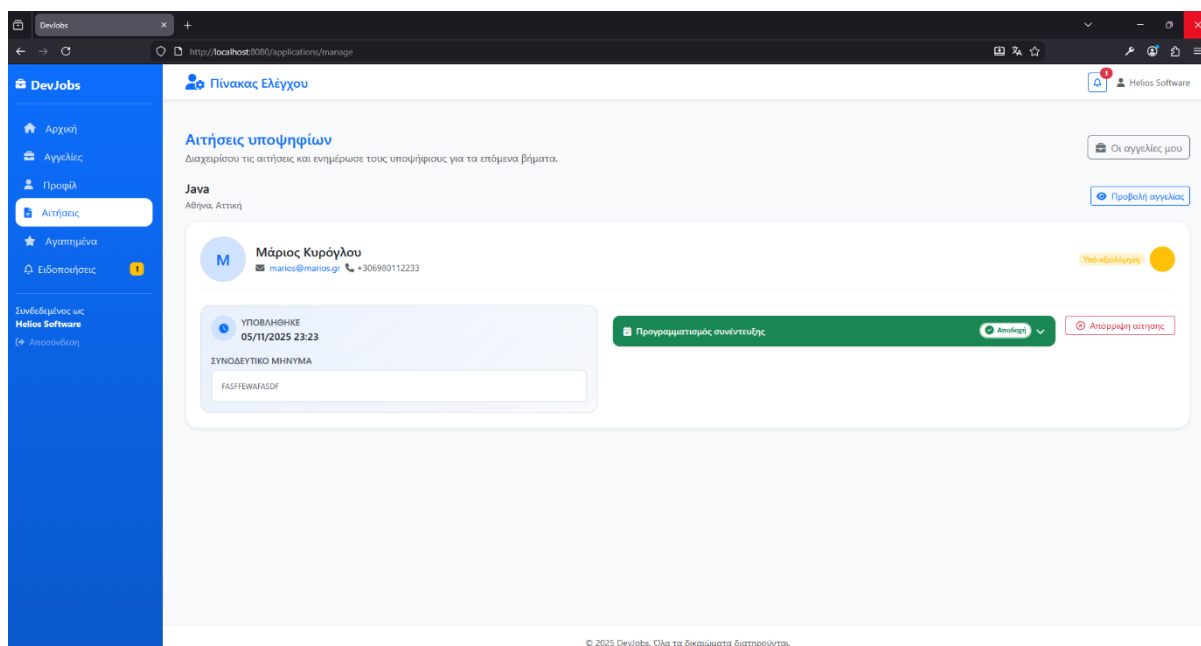
Στην εικόνα 58, απεικονίζεται η σελίδα ειδοποιήσεων, όπου ο χρήστης μπορεί να κάνει ανάγνωση των νέων ή παλιών ειδοποιήσεων και να τις επισημάνει ως αναγνωσμένες.



Εικόνα 62: Notification για νέα υποβολή αίτησης

#### 5.4.8 Προβολή αίτησης του υποψήφιου εργαζομένου

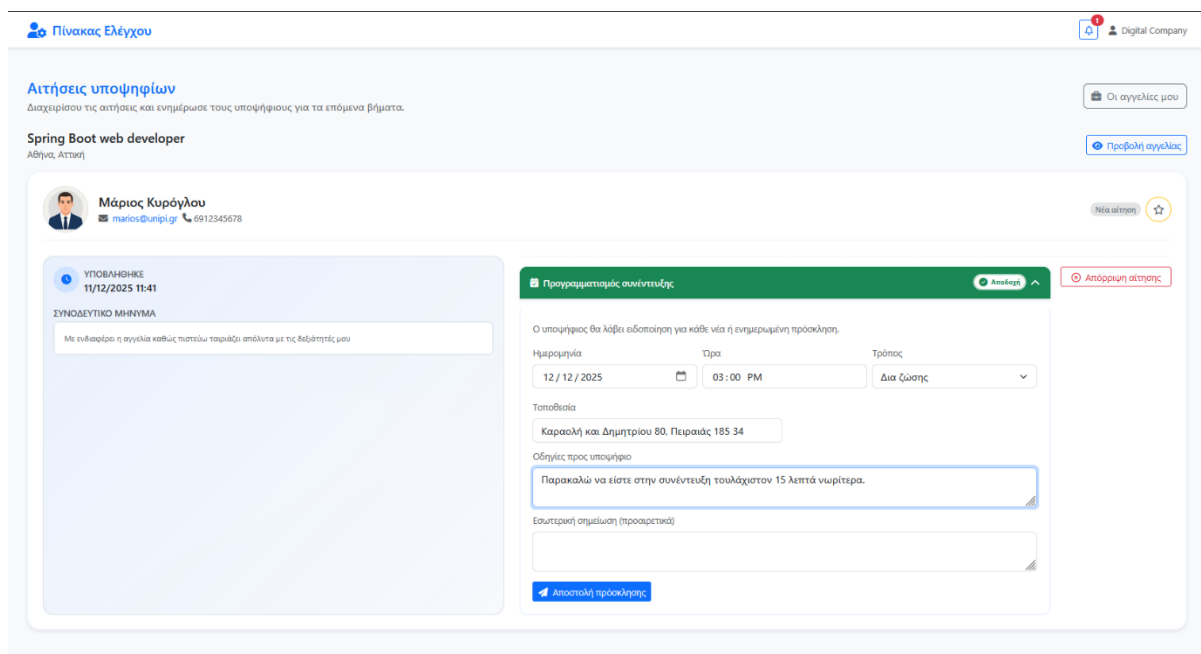
Στην σελίδα αιτήσεων, ο εργοδότης έχει τη δυνατότητα να δει την αίτηση του υποψηφίου για την εκάστοτε αγγελία και ταυτόχρονα έχει την επιλογή να αποδεχθεί ή να απορρίψει την αίτηση.



Εικόνα 63: Επισκόπηση στοιχείων αίτησης νέου υποψηφίου

#### 5.4.9 Αποδοχή – Απόρριψη Συνέντευξης

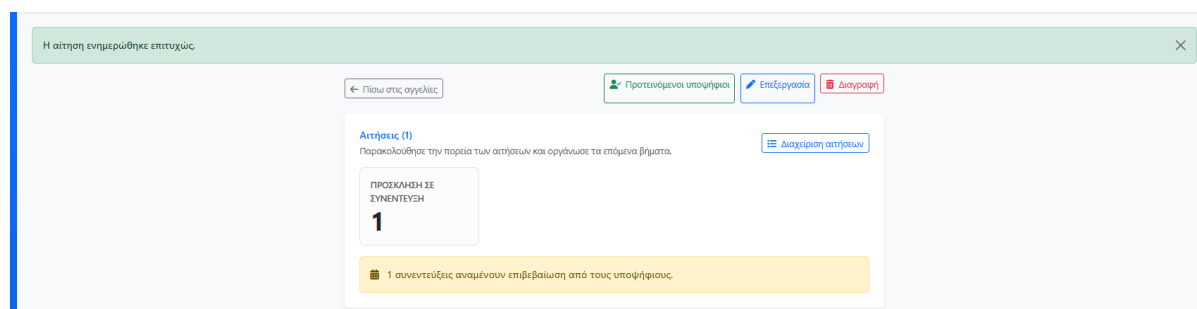
Με την αποδοχή της αίτησης, συνοδεύεται και ο κανονισμός συνέντευξης με τον υποψήφιο, όπου ο εργοδότης θα πρέπει να συμπληρώσει τα απαραίτητα στοιχεία, όπως ημερομηνία και ώρα της συνέντευξης, τρόπος επικοινωνίας, τοποθεσία και άλλα.



Εικόνα 64: Υποβολή συνέντευξης και λοιπών λεπτομερειών που αφορούν την υποβληθείσα αίτηση

#### 5.4.9 Ενημερωτικό μήνυμα κατάστασης αίτησης

Στη σελίδα των αιτήσεων, βρίσκεται και ένα πλαίσιο όπου συγκεντρώνει πληροφορίες σχετικά με τις αιτήσεις που εκκρεμούν από τις αγγελίες του εργοδότη.



Εικόνα 65: Notification σχετικά με την κατάσταση της αίτησης

## 6 Εγχειρίδιο χρήστη

### 6.1 Εισαγωγή

Καλώς ήλθατε στο περιβάλλον εύρεσης εργασίας της “DevJobs”! Ένα σύγχρονο τεχνολογικά σύστημα που αφορά την διαχείριση αγγελιών εργασίας από τους εργοδότες και την ταχεία εύρεση εργασίας από πλευράς του χρήστη. Μέσω της πλοήγησης στην συγκεκριμένη διαδικτυακή εφαρμογή, διευκολύνεται τόσο η εκάστοτε εταιρία, όσο και ο υποψήφιος εργαζόμενος, καθώς η διαδικασία εύρεσης στην “DevJobs” είναι πιο προσωποποιημένη και πιο εύκολη από ποτέ. Βασισμένα στις δεξιότητες, αλλά και στις ανάγκες που υπάρχουν στο εργασιακό τομέα, τα προσαρμοστικά συστήματα και οι αλγόριθμοι που υλοποιήθηκαν στοχεύουν στην πλήρη αντιστοίχιση των «θέλω» και των δύο πλευρών. Με την σειρά του, το παρόν εγχειρίδιο αποσκοπεί στην καθοδήγηση του χρήστη, προκειμένου να κατανοήσει σε βάθος και να αξιοποιήσει αποτελεσματικά τις διαθέσιμες λειτουργίες που προαναφέρθηκαν.

### 6.2 Εγγραφή και Σύνδεση Χρηστών

Προκειμένου οι νέοι χρήστες να απολαμβάνουν τα δικαιώματα και τις υπηρεσίες που τους παρέχει η εφαρμογή, καλούνται να δημιουργήσουν έναν προσωπικό λογαριασμό. Η διαδικασία της εγγραφής πραγματοποιείται μέσω της αντίστοιχης φόρμας, η οποία είναι προσβάσιμη από την σελίδα σύνδεσης. Κατά την εγγραφή, ο υποψήφιος χρήστης καλείται να επιλέξει τον ρόλο του (υποψήφιος – εργοδότης) και να καταχωρίσει τα βασικά στοιχεία ταυτοποίησης.

Ως προς την σύνδεση στο σύστημα, οι εγγεγραμμένοι χρήστες εισαγάγουν τη διεύθυνση του ηλεκτρονικού τους ταχυδρομείου και του προσωπικού τους κωδικού πρόσβασης που είχαν αρχικά ορίσει. Με την επιτυχή αυθεντικοποίηση και την αντιστοίχιση των στοιχείων με την βάση, δημιουργείται ενεργεία συνεδρία, η οποία επιτρέπει την ασφαλή πλοήγηση και την χρήση των λειτουργιών του συστήματος.

### 6.3 Αρχική Σελίδα και Πίνακας Ελέγχου

Μετά την σύνδεση, ο χρήστης μεταφέρεται στην αρχική σελίδα, η οποία λειτουργεί ως ο κεντρικός πίνακας ελέγχου. Εκεί είναι διαθέσιμες όλες οι βασικές επιλογές πλοήγησης, προσαρμοσμένες στον ρόλο του χρήστη. Οι υποψήφιοι έχουν άμεση πρόσβαση σε προτεινόμενες αγγελίες εργασίας, προτάσεις αντιστοίχισης και στο προσωπικό τους προφίλ, ενώ οι εργοδότες με την σειρά τους έχουν την ευχέρεια να διαχειρίζονται τις αγγελίες που έχουν δημοσιεύσει και τις αιτήσεις που έχουν λάβει από τους υποψηφίους. Τέλος, στο επάνω μέρος της σελίδας είναι διαθέσιμα τα προσωπικά στοιχεία του λογαριασμού και ειδοποιήσεις, οι οποίες ενημερώνουν τον χρήστη για σημαντικές πληροφορίες, όπως νέα αιτήματα, αλλαγές κατάστασης αιτήσεων ή προτάσεις υποψηφίων.



## 6.4 Προφίλ Χρήστη

Η εφαρμογή “DevJobs” υποστηρίζει την αναλυτική διαχείριση του προφίλ του χρήστη. Οι υποψήφιοι διαθέτουν προσωπικό ή επαγγελματικό προφίλ αντίστοιχα, στο οποίο καταχωρούν στοιχεία όπως δεξιότητες, επαγγελματική κατάρτιση και εμπειρία και προσωπικές προτιμήσεις. Οι εργοδότες αντίστοιχα, διαχειρίζονται το εταιρικό προφίλ και αγγελίες, οι οποίες περιλαμβάνουν σημαντικές πληροφορίες για τους στόχους και το αντικείμενο της εταιρίας, αλλά και για τις ανοιχτές θέσεις εργασίας. Η συμπλήρωση του προφίλ με ορθές πληροφορίες και ειλικρίνεια είναι καθοριστικής σημασίας, καθώς φέρουν άμεση επιρροή στον μηχανισμό αντιστοίχισης και στην σύνδεση υποψήφιων-εργοδοτών, σχετικά με προτάσεις εργασίας.

## 6.5 Αγγελίες Εργασίας και Αιτήσεις

Οι εργοδότες έχουν την ικανότητα να δημιουργούν, να επεξεργάζονται και να διαγράφουν αγγελίες, ανάλογα με τις ανάγκες και τις ελεύθερες θέσεις εργασίας στο συγκεκριμένο χρονικό διάστημα. Από την άλλη, οι υποψήφιοι έχουν την ικανότητα να αναζητούν αγγελίες, μέσω φίλτρων και να υποβάλλουν απευθείας το ενδιαφέρον τους στο σύστημα. Κάθε αίτηση συνοδεύεται και από την κατάστασή της (π.χ. υποβληθείσα), η οποία ενημερώνεται δυναμικά από τον εργοδότη και είναι ορατή και στα δύο μέρη.

## 6.6 Αντιστοίχιση και Πιθανές Προτάσεις

Η “DevJobs” έχει ενσωματωμένο μηχανισμό, ο οποίος υπολογίζει τον βαθμό αντιστοίχισης ανάμεσα σε εταιρία και εργαζόμενο. Η διαδικασία είναι βασισμένη στις δεξιότητες και στα χαρακτηριστικά που έχει εναποθέσει ο χρήστης από την αρχή, καθώς και στις ανάγκες της επιχείρησης, τον τύπο απασχόλησης, της γεωγραφικής τοποθεσίας κ.α. Ως αποτέλεσμα, οι υποψήφιοι λαμβάνουν προτεινόμενες αγγελίες, ενώ οι εργοδότες αποκτούν εικόνα των καταλληλότερων υποψηφίων.

## 6.7 Αγαπημένα και Ειδοποιήσεις

Η εφαρμογή παρέχει τη δυνατότητα αποθήκευσης αγγελιών και υποψηφίων από το εκάστοτε τύπου χρήστη, διευκολύνοντας σε σημαντικό βαθμό την ταχεία πρόσβαση σε επιλεγμένο και περιορισμένο περιεχόμενο. Παράλληλα, το σύστημα ειδοποιήσεων ενημερώνει τους χρήστες για κρίσιμες ενέργειες, όπως κατάσταση αίτησης, νέες αιτήσεις ή και προγραμματισμό μελλοντικών συνεντεύξεων.

## 6.8 Επικοινωνία

Για οποιοδήποτε τεχνικό ζήτημα που ενδέχεται να αντιμετωπίσετε ή πιθανή απορία σχετικά με τις λειτουργίες του συστήματος, κάθε χρήστης πλέον μπορεί να επικοινωνήσει με την διαχειριστική ομάδα, μέσω του προβλεπόμενου καναλιού επικοινωνίας. Η “DevJobs” είναι εκεί για εσάς σε οποιαδήποτε στιγμή της ημέρας.

# 7 Συμπεράσματα – Δυσκολίες κατά την υλοποίηση και μελλοντικές βελτιώσεις

## 7.1 Συμπεράσματα

Η παρούσα πτυχιακή εργασία επικεντρώθηκε στον σχεδιασμό και την υλοποίηση της εφαρμογής DevJobs, μιας ψηφιακής πλατφόρμας εύρεσης εργασίας με έμφαση στον τομέα της πληροφορικής και των τεχνολογικών ειδικοτήτων. Μέσα από τη μελέτη της διεθνούς βιβλιογραφίας και την ανάλυση υφιστάμενων πλατφορμών, αναδείχθηκαν οι βασικές αδυναμίες των σύγχρονων συστημάτων αντιστοίχισης, οι οποίες αποτέλεσαν το έναυσμα για τον σχεδιασμό μιας πιο στοχευμένης και διαφανούς λύσης.

Η ανάπτυξη της εφαρμογής βασίστηκε σε αρχές καθαρής αρχιτεκτονικής και διαχωρισμού ευθυνών, με σαφή διάκριση των επιπέδων controller, service και repository. Η προσέγγιση αυτή συνέβαλε στη διατηρησιμότητα και την επεκτασιμότητα του συστήματος, ενώ η ορθολογική μοντελοποίηση της βάσης δεδομένων διασφάλισε τη συνέπεια και την ακεραιότητα των αποθηκευμένων πληροφοριών. Ιδιαίτερη έμφαση δόθηκε και σε ζητήματα ασφάλειας, όπως η κρυπτογράφηση κωδικών, η διαχείριση ρόλων χρηστών και η προστασία προσωπικών δεδομένων, τα οποία αποτελούν κρίσιμους παράγοντες για εφαρμογές που διαχειρίζονται ευαίσθητες πληροφορίες.

Κεντρικό στοιχείο της πλατφόρμας αποτέλεσε ο μηχανισμός αυτόματης αντιστοίχισης υποψηφίων και αγγελιών εργασίας. Ο αλγόριθμος σχεδιάστηκε με βάση μια πολυκριτηριακή προσέγγιση, λαμβάνοντας υπόψη κυρίως τη συνάφεια δεξιοτήτων, καθώς και πρακτικούς παράγοντες όπως η τοποθεσία και ο τύπος απασχόλησης. Παρότι δεν αξιοποιήθηκαν προηγμένα μοντέλα μηχανικής μάθησης, η υλοποίηση απέδωσε αξιόπιστα αποτελέσματα και προσέφερε μια ερμηνεύσιμη και διαφανή διαδικασία προτάσεων, θέτοντας σταθερά θεμέλια για μελλοντική εξέλιξη.

Κατά τη διάρκεια της υλοποίησης προέκυψαν τεχνικές προκλήσεις, όπως η ορθή επικοινωνία μεταξύ backend και frontend, η αποφυγή ασυνεπειών στη λογική των services και η ευθυγράμμιση της επιχειρησιακής λογικής με τη ροή της εφαρμογής. Η αντιμετώπιση των ζητημάτων αυτών συνέβαλε ουσιαστικά στην κατανόηση της ανάπτυξης πληροφοριακών συστημάτων σε συνθήκες που προσομοιάζουν πραγματικά περιβάλλοντα παραγωγής. Συνολικά, η εργασία κατέληξε σε μια λειτουργική, επεκτάσιμη και τεχνολογικά συνεπή πλατφόρμα, η οποία μπορεί να αποτελέσει βάση για περαιτέρω έρευνα και ανάπτυξη.

## 7.2 Αναμενόμενα οφέλη

Με την υλοποίηση της παρούσας πτυχιακής εργασίας, η αγορά εργασίας αναμένεται να αποκτήσει αυξημένο δυναμισμό και τεχνολογικό ενδιαφέρον, καθώς το αυτοματοποιημένο περιβάλλον διαχείρισης σε συνδυασμό με τις διαθέσιμες λειτουργίες έχουν ως αντικείμενο ενδιαφέροντος πρωτίστως τον ίδιο τον ανθρώπινο παράγοντα. Η αξιοποίηση ψηφιακών μηχανών αναζήτησης και αντιστοίχισης ενισχύουν τον ευγενή ανταγωνισμό μεταξύ των επιχειρήσεων, προάγοντας την διαφάνεια και τη βελτιστοποίηση των διαδικασιών στελέχωσης. Παράλληλα, οι ίδιοι οι οργανισμοί ενδέχεται να μειώσουν σημαντικά τα λειτουργικά τους κόστη, καθώς η διαδικασία διαλογής και επιλογής υποψηφίων κρίνεται με ορισμένα κριτήρια και δεδομένα.

Ως προς τους υποψηφίους, η εφαρμογή συμβάλλει άμεσα στην βελτίωση της εμπειρίας αναζήτησης εργασίας, προσφέροντας εξατομικευμένες προτάσεις αγγελιών βάσει δεξιοτήτων, προτιμήσεων και επαγγελματικού προφίλ. Η δυνατότητα αποθήκευσης αγαπημένων αγγελιών εργασίας, η διαρκής ικανότητα παρακολούθησης της κατάστασης της αίτησης και η συνεχής ενημέρωση μέσα από τις ποικίλες ειδοποιήσεις ενισχύουν το αίσθημα ελέγχου και διαφάνειας για τον ίδιο το χρήστη, το οποίο φθίνει σήμερα από την ραγδαία ανάπτυξη των τεχνολογικών εξελίξεων. Έτσι, μειώνεται η αβεβαιότητα στο εργασιακό περιβάλλον και ενδέχεται επίσης να παρατηρηθεί η μείωση της ανεργίας, μέσα από την συγκεκριμένη καταχώριση των δεξιοτήτων του χρήστη και την αντιστοίχιση αυτών με τις υπάρχουσες θέσεις εργασίας.

Από την άλλη πλευρά, οι εργοδότες επωφελούνται από την ύπαρξη δομημένων επαγγελματικών προφίλ και μηχανισμών αντιστοίχισης, καθώς μειώνεται σημαντικά ο χρόνος αντιστοίχισης και αυξάνεται η πιθανότητα κάλυψης των κενών θέσεων με υπαλλήλους, έτοιμους να ανταπεξέλθουν σθεναρά στις ανάγκες και στην εξέλιξη της εταιρίας στο μέλλον. Η δυνατότητα φιλτραρίσματος, αξιολόγησης και σύγκρισης αιτήσεων θα οδηγήσουν αναπόφευκτα στην λήξη πιο τεκμηριωμένων αποφάσεων που αφορούν το ανθρώπινο δυναμικό, ενώ η αποθήκευση αγαπημένων υποψηφίων ενδέχεται μακροπρόθεσμα να συμβάλει στην κάλυψη των κενών θέσεων που πιθανόν να δημιουργηθούν.

Τέλος, σε ανθρωπιστικό πλαίσιο η “DevJobs” δύναται να λειτουργήσει ως καταλύτης για την μείωση των ανισοτήτων στον εργασιακό κλάδο, προσφέροντας ίσες ευκαιρίες πρόσβασης σε όλους εκείνους που το αξίζουν. Μέσω της αντικειμενικής αξιολόγησης και της διαφανούς παρουσίασης των αγγελιών, η εφαρμογή προάγει με την σειρά της ένα πιο δίκαιο και αποδοτικό μοντέλο επαγγελματικής αποκατάστασης, θέτοντας τις βάσεις για μία σύγχρονη και βιώσιμη ψηφιακή αγορά εργασίας.

### **7.3 Μελλοντικές επεκτάσεις και βελτιώσεις**

Παρότι η εφαρμογή DevJobs καλύπτει τις βασικές λειτουργικές απαιτήσεις μιας πλατφόρμας εύρεσης εργασίας, υπάρχουν σημαντικά περιθώρια βελτίωσης και εμπλουτισμού, τα οποία μπορούν να ενισχύσουν τόσο την αποδοτικότητα όσο και την εμπειρία των χρηστών.

#### **7.3.1 Βελτίωση του μηχανισμού αντιστοίχισης**

Η τρέχουσα υλοποίηση της αντιστοίχισης βασίζεται σε κανόνες και προκαθορισμένα βάρη συνάφειας. Μελλοντικά, θα μπορούσε να ενσωματωθεί:

- χρήση μοντέλων μηχανικής μάθησης για δυναμική προσαρμογή των κριτηρίων αντιστοίχισης,
- ανάλυση βιογραφικών και αγγελιών μέσω τεχνικών επεξεργασίας φυσικής γλώσσας (NLP),
- αυτόματη πρόταση δεξιοτήτων που λείπουν από το προφίλ του υποψηφίου, βάσει απαιτήσεων της αγοράς.

#### **7.3.2 Προσθήκη συστήματος αναφορών και στατιστικών**

Η ενσωμάτωση αναλυτικών εργαλείων θα μπορούσε να προσφέρει:

- αποτύπωση τάσεων της αγοράς εργασίας,
- στατιστικά στοιχεία για συχνότερες δεξιότητες ανά κλάδο,
- δείκτες επιτυχίας αιτήσεων και προσλήψεων.

Τα δεδομένα αυτά θα ενίσχυαν τη λήψη αποφάσεων τόσο από υποψηφίους όσο και από εργοδότες.

#### **7.3.3 Ανάπτυξη εφαρμογής για κινητές συσκευές**

Η δημιουργία εγγενούς mobile εφαρμογής για Android και iOS θα μπορούσε να:

- βελτιώσει την προσβασιμότητα της πλατφόρμας,
- αυξήσει την αλληλεπίδραση μέσω ειδοποιήσεων σε πραγματικό χρόνο,
- υποστηρίξει άμεση χρήση ανεξαρτήτως συσκευής.

#### **7.3.4 Εμπλουτισμός του συστήματος ειδοποιήσεων**

Μελλοντικές βελτιώσεις περιλαμβάνουν την υλοποίηση πιο έξυπνων και προσωποποιημένων ειδοποιήσεων, όπως:

- προτάσεις θέσεων βάσει νέων δεξιοτήτων,
- ενημερώσεις για μεταβολές στην αγορά εργασίας,
- ειδοποιήσεις για προθεσμίες αιτήσεων.

### 7.3.5 Προσθήκη συστήματος επικοινωνίας (chat)

Η ενσωμάτωση ενός ασφαλούς συστήματος άμεσης επικοινωνίας μεταξύ εργοδοτών και υποψηφίων θα μπορούσε να καταστήσει τη διαδικασία πιο άμεση και αποτελεσματική, μειώνοντας τον χρόνο ανταπόκρισης και διευκολύνοντας την αρχική επαφή.

### 7.3.6 Ενίσχυση της ασφάλειας

Σε επίπεδο ασφάλειας, μελλοντικές επεκτάσεις θα μπορούσαν να περιλαμβάνουν:

- προσθήκη μηχανισμών ταυτοποίησης δύο παραγόντων (2FA),
- καταγραφή και παρακολούθηση συμβάντων ασφαλείας (audit logs),
- αυστηρότερη διαχείριση sessions και πρόσβασης.

### 7.3.7 Βελτίωση διαχείρισης αρχείων και βιογραφικών

Τέλος, θα μπορούσε να προστεθεί προηγμένη διαχείριση βιογραφικών, όπως:

- αποθήκευση και οργάνωση αρχείων CV,
- αυτόματη εξαγωγή στοιχείων από αρχεία PDF,
- δημιουργία προτύπων βιογραφικών μέσω της πλατφόρμας.

## 7.4 Περιορισμοί της παρούσας μελέτης

Παρά το γεγονός ότι η εφαρμογή DevJobs υλοποιήθηκε ως ένα λειτουργικό και επεκτάσιμο πληροφοριακό σύστημα, η παρούσα μελέτη υπόκειται σε ορισμένους περιορισμούς, οι οποίοι σχετίζονται τόσο με το εύρος της ανάπτυξης όσο και με τις συνθήκες εκπόνησης της πτυχιακής εργασίας. Η αναγνώριση και η ανάλυση των περιορισμών αυτών κρίνεται ιδιαίτερα σημαντική, καθώς συμβάλλει στην ορθότερη αξιολόγηση των αποτελεσμάτων και στην καλύτερη κατανόηση των δυνατοτήτων αλλά και των ορίων περαιτέρω εξέλιξης της πλατφόρμας.

Ένας βασικός περιορισμός αφορά τη χρονική διάρκεια υλοποίησης. Η ανάπτυξη της εφαρμογής πραγματοποιήθηκε στο πλαίσιο μιας ακαδημαϊκής εργασίας με συγκεκριμένα χρονικά όρια, γεγονός που επηρέασε τόσο τον βαθμό πολυπλοκότητας του συστήματος όσο και τον αριθμό των λειτουργιών που μπορούσαν να ενσωματωθούν. Ως αποτέλεσμα, ορισμένες δυνατότητες που θα μπορούσαν να προσδώσουν επιπλέον ευφυΐα, αυτοματοποίηση ή προσαρμοστικότητα στο σύστημα παρέμειναν σε θεωρητικό επίπεδο και παρουσιάζονται ενδεικτικά στο κεφάλαιο των μελλοντικών επεκτάσεων.

Επιπλέον, ο μηχανισμός αντιστοίχισης βασίζεται σε κανόνες και προκαθορισμένα κριτήρια συνάφειας, χωρίς τη χρήση πραγματικών ιστορικών δεδομένων ή τεχνικών μηχανικής μάθησης. Η απουσία τέτοιων δεδομένων καθιστά δύσκολη τη δυναμική προσαρμογή του αλγορίθμου στις μεταβαλλόμενες ανάγκες της αγοράς εργασίας και περιορίζει τη δυνατότητα αυτοβελτίωσης του συστήματος μέσω ανατροφοδότησης που θα μπορούσε να προκύψει από τη συμπεριφορά και τις επιλογές των χρηστών σε πραγματικές συνθήκες χρήσης.

Ένας ακόμη σημαντικός περιορισμός αφορά την αξιολόγηση της εφαρμογής από τελικούς χρήστες. Στο πλαίσιο της παρούσας εργασίας δεν πραγματοποιήθηκε εκτεταμένη δοκιμή χρησιμότητας (usability testing) με πραγματικούς υποψηφίους και εργοδότες. Κατά συνέπεια, η αξιολόγηση της εμπειρίας χρήστη βασίστηκε κυρίως σε λειτουργικά κριτήρια και σε δοκιμές ορθής εκτέλεσης, χωρίς τη συστηματική συλλογή ποιοτικών δεδομένων, όπως οργανωμένα σχόλια χρηστών, ερωτηματολόγια ή μετρήσεις ευχρηστίας.

Παράλληλα, η εφαρμογή δεν συνδέθηκε με εξωτερικές πηγές δεδομένων ή πραγματικές αγγελίες εργασίας, γεγονός που περιορίζει τη δυνατότητα γενίκευσης των αποτελεσμάτων της μελέτης. Η χρήση συνθετικών ή περιορισμένων δεδομένων, αν και επαρκής για εκπαιδευτικούς και πειραματικούς σκοπούς, δεν μπορεί να αναπαραστήσει πλήρως τη δυναμική, τον όγκο και την πολυπλοκότητα της πραγματικής αγοράς εργασίας.

Τέλος, η παρούσα μελέτη εστιάζει κυρίως στη λειτουργική και τεχνική υλοποίηση της πλατφόρμας και λιγότερο στη στατιστική ή ποσοτική αξιολόγηση της απόδοσης του συστήματος. Η έλλειψη συγκεκριμένων μετρικών απόδοσης, όπως δείκτες ακρίβειας αντιστοίχισης, χρόνοι απόκρισης υπό αυξημένο φορτίο ή μετρήσεις αποδοτικότητας, αποτελεί έναν επιπλέον περιορισμό που θα μπορούσε να αντιμετωπιστεί σε μελλοντική, πιο εκτενή έρευνα.

Παρά τους παραπάνω περιορισμούς, η εφαρμογή DevJobs εκπληρώνει τους βασικούς στόχους της παρούσας πτυχιακής εργασίας και μπορεί να θεωρηθεί ως ένα αξιόπιστο και σταθερό υπόβαθρο για περαιτέρω ανάπτυξη, πειραματισμό και ερευνητική αξιοποίηση σε πραγματικά σενάρια χρήσης.

## 6 Βιβλιογραφία

1. Tewari, S., Joshi, A. and Tewari, D. (2024). AI Powered HR: Impact, Benefits and Challenges. SSRN. <https://doi.org/10.2139/ssrn.4990927>
2. Stojanović, A., Milošević, I. and Arsić, S. (2023). The Influence of Digital Literacy on the Employment Rate of the Older Population. Proceedings of the XIX International May Conference on Strategic Management (IMCSM23), pp.291–300. University of Belgrade, Technical Faculty in Bor.
3. Kelsey Purcell, CPRW (2025). Applicant Tracking System (ATS) Usage Report: Key Shifts and Strategies for Job Seekers. Jobscan.
4. SAP (2025). Τι είναι το σύστημα παρακολούθησης αιτούντων (ATS).
5. Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2019), pp.4171–4186.
7. Ramanath, R., Inan, H., Polatkan, G., Hu, B., Guo, Q., Ozcaglar, C., Wu, X., Kenthapadi, K. and Geyik, S. C. (2018). Towards Deep and Representation Learning for Talent Search at LinkedIn. Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM 2018), pp.2253–2261. <https://doi.org/10.1145/3269206.3272030>
8. Deloitte (2023). Global Human Capital Trends Report. Deloitte Insights.

9. World Economic Forum (2023) .The Future of Jobs Report.

10. Lane, M. and A. Saint-Martin (2021), "The impact of Artificial Intelligence on the labour market: What do we know so far?", OECD Social, Employment and Migration Working Papers, No. 256, OECD Publishing, Paris, <https://doi.org/10.1787/7c895724-en>.

11. Eurofound (2018). Automation, digitisation and platforms: Implications for work and employment. Publications Office of the European Union. DOI: 10.2806/090974.

12. Cath, C. (2018). Governing artificial intelligence: ethical, legal and technical opportunities and challenges. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 376(2133), 20180080. <https://doi.org/10.1098/rsta.2018.0080>

13. Politou, E.A. Efthimios Alepis, E., Virvou, M., Patsakis, C. : Privacy and Data Protection Challenges in the Distributed Era. Springer 2022, ISBN 978-3-030-85442-3, pp. 1-185, Springer.

14. Brusilovsky, P. (2001). Adaptive Hypermedia. User Modeling and User-Adapted Interaction, 11, 87–110. <https://doi.org/10.1023/A:1011143116306>

15.. Dix, A., Finlay, J., Abowd, G. D. and Beale, R. (2004). Human–Computer Interaction (3rd ed.). Pearson Education Limited.

16. Shute, V. J. (2007). Focus on formative feedback. ETS Research Report RR-07-11. Educational Testing Service. <https://doi.org/10.3102/0034654307313795>

16. Sommerville Ian (2016). Software Engineering (10th ed.). Pearson.

17. Hukkeri, P. and Pol, S. (2025). The impact of artificial intelligence on recruitment efficiency and onboarding effectiveness. IOSR Journal of Multidisciplinary Research, 2(3), 26–33. <https://doi.org/10.9790/7439-0203012633>

18. OECD (2020), OECD Employment Outlook 2020: Worker Security and the COVID-19 Crisis, OECD Publishing, Paris, <https://doi.org/10.1787/1686c758-en>.

20. NIST (2018). Developing Cyber-Resilient Systems: A Systems Security Engineering Approach (NIST SP 800-160, Vol. 2). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.800-160v2>.

22. Alepis, E. and Virvou, M., 2006, October. Emotional Intelligence: Constructing user stereotypes for affective bi-modal interaction. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems* (pp. 435-442). Berlin, Heidelberg: Springer Berlin Heidelberg.
23. Kabassi, K. and Virvou, M., 2006. A knowledge-based software life-cycle framework for the incorporation of multicriteria analysis in intelligent user interfaces. *IEEE Transactions on Knowledge and data Engineering*, 18(9), pp.1265-1277.
24. Katsionis, G. and Virvou, M., 2004, October. A cognitive theory for affective user modelling in a virtual reality educational game. In *2004 IEEE international conference on systems, man and cybernetics* (IEEE Cat. No. 04CH37583) (Vol. 2, pp. 1209-1213). IEEE.
25. Virvou, M., 1999. Automatic reasoning and help about human errors in using an operating system. *Interacting with Computers*, 11(5), pp.545-573.
26. Chrysafiadi, K., Papadimitriou, S. and Virvou, M., 2022. Cognitive-based adaptive scenarios in educational games using fuzzy reasoning. *Knowledge-Based Systems*, 250, p.109111.
27. Chrysafiadi, K. and Virvou, M., 2013. Dynamically personalized e-training in computer programming and the language C. *IEEE transactions on education*, 56(4), pp.385-392.
28. Chrysafiadi, K. and Virvou, M., 2013. A knowledge representation approach using fuzzy cognitive maps for better navigation support in an adaptive learning system. *SpringerPlus*, 2, pp.1-13.
29. Chrysafiadi, K. and Virvou, M., 2015. *Advances in personalized web-based education*. Springer International Publishing.
30. Moundridou, M. and Virvou, M., 2001, August. Authoring and delivering adaptive Web-based textbooks using WEAR. In *Proceedings IEEE International Conference on Advanced Learning Technologies* (pp. 185-188). IEEE.
31. Papadimitriou, S. and Virvou, M., 2017, August. Adaptivity in scenarios in an educational adventure game. In *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)* (pp. 1-6). IEEE..
32. Savvopoulos, A. and Virvou, M., 2010. Evaluating the Generality of a Life-Cycle Framework for Incorporating Clustering Algorithms in Adaptive Systems. In *Multimedia Services in Intelligent Environments: Software Development Challenges and Solutions* (pp. 5-25). Berlin, Heidelberg: Springer Berlin Heidelberg.



33. Alogogianni, E. and Virvou, M., 2021. Addressing the issue of undeclared work—Part I: Applying associative classification per the CRISP-DM methodology. *Intelligent Decision Technologies*, 15(4), pp.721-747.
34. Panagoulas, D.P., Sarvas, E., Marinakis, V., Virvou, M., Tsihrintzis, G.A. and Doukas, H., 2023. Intelligent decision support for energy management: A methodology for tailored explainability of artificial intelligence analytics. *Electronics*, 12(21), p.4430.
35. Panagoulas, D.P., Sotiropoulos, D.N. and Tsihrintzis, G.A., 2021. Nutritional biomarkers and machine learning for personalized nutrition applications and health optimization. *Intelligent Decision Technologies*, 15(4), pp.645-653.
36. Virvou, M., Savvopoulos, A., Tsihrintzis, G.A. and Sotiropoulos, D.N., 2007. Constructing Stereotypes for an Adaptive e-Shop using AIN-based Clustering. In *Adaptive and Natural Computing Algorithms: 8th International Conference, ICANNGA 2007, Warsaw, Poland, April 11-14, 2007, Proceedings, Part I* 8 (pp. 837-845). Springer Berlin Heidelberg.
37. Panagoulas, D.P., Virvou, M. and Tsihrintzis, G.A., 2024. A novel framework for artificial intelligence explainability via the Technology Acceptance Model and Rapid Estimate of Adult Literacy in Medicine using machine learning. *Expert Systems with Applications*, 248, p.123375.
39. Chrysafiadi, K., Virvou, M., Tsihrintzis, G.A. and Hatzilygeroudis, I., (2023). An Adaptive Learning Environment for Programming Based on Fuzzy Logic and Machine Learning. *International Journal on Artificial Intelligence Tools*, 32(05), p.2360011.
40. Virvou, M., 2023. Artificial Intelligence and User Experience in reciprocity: Contributions and state of the art. *Intelligent Decision Technologies* 17 (2023) 73–125 DOI 10.3233/IDT-230092 IOS Press
41. Mouggiakou, E. and Virvou, M., 2017, August. Based on GDPR privacy in UML: Case of e-learning program. In *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)* (pp. 1-8). IEEE.
42. Stack Overflow (2024). Stack Overflow Developer Survey 2024. Stack Overflow. <https://survey.stackoverflow.co/2024>
43. U.S. Bureau of Labor Statistics (2024). Occupational Outlook Handbook: Software Developers. U.S. Department of Labor. <https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm>.

43. Bogen, M. and Rieke, A. (2018). Help wanted: An examination of hiring algorithms, equity, and bias. *Upturn*. <https://www.upturn.org/reports/2018/hiring-algorithms/>