



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Εκπαιδευτική Πλατφόρμα: EduVerse Educational Platform: EduVerse
Όνοματεπώνυμο Φοιτητή	Φώτιος Τσιούμας Fotios Tsioumas
Πατρώνυμο	Κωνσταντίνος
Αριθμός Μητρώου	ΜΠΠΛ21079
Επιβλέπων	Ευθύμιος Αλέπης, Αναπληρωτής Καθηγητής

Ημερομηνία Παράδοσης **Δεκέμβριος 2023**

Τριμελής Επιτροπή Εξέτασης

Μαρία Βίρβου
Καθηγήτρια

Ευθύμιος Αλέπης
Αναπληρωτής Καθηγητής

Διονύσιος Σωτηρόπουλος
Επίκουρος Καθηγητής

Περιεχόμενα

Περίληψη	5
Εισαγωγή.....	6
Ανασκόπηση Πεδίου	7
Ψηφιακή Εκπαιδευτική Πλατφόρμα e-me	7
GUNet2 eClass - Τμήμα Πληροφορικής	7
Brainy	8
Παρουσίαση και Χρήση Εφαρμογής	9
Πλοήγηση ως «επισκέπτης»	9
Αρχική Σελίδα	9
Εγγραφή ανά Ιδιότητα.....	10
Φόρμα Εγγραφής Καθηγητή	11
Φόρμα Εγγραφής Φοιτητή	12
Φόρμα Σύνδεσης.....	13
Πλοήγηση ως «καθηγητής».....	14
Αρχική Σελίδα	15
Δημιουργία Νέου Μαθήματος.....	16
Τα Μαθήματα Μου.....	17
Προβολή Μαθήματος.....	18
Επεξεργασία Μαθήματος.....	19
Δημιουργία Νέου Κεφαλαίου	20
Προβολή Κεφαλαίου	21
Επεξεργασία Κεφαλαίου	22
Προβολή Ερωτήσεων Κεφαλαίου.....	23
Δημιουργία Νέας Ερώτησης.....	24
Επεξεργασία Ερώτησης.....	25
Οι Φοιτητές Μου	26
Προβολή Αποτελεσμάτων Φοιτητή.....	27
Προβολή Ερωτηματολογίου	28
Ο Λογαριασμός Μου.....	29
Επεξεργασία Πληροφοριών Λογαριασμού.....	29
Αλλαγή Κωδικού Πρόσβασης.....	31
Πλοήγηση ως «φοιτητής».....	32
Αρχική Σελίδα	33
Διαθέσιμα Μαθήματα	34
Τα Μαθήματα Μου.....	35
Προβολή Μαθήματος.....	36
Προβολή Κεφαλαίου	36
Ερωτηματολόγιο.....	37

Αποτέλεσμα Ερωτηματολογίου	38
Τα Αποτελέσματά Μου.....	39
Προβολή Ερωτηματολογίου	40
Ο Λογαριασμός Μου.....	40
Επεξεργασία Πληροφοριών Λογαριασμού.....	41
Αλλαγή Κωδικού Πρόσβασης.....	41
Σελίδες Σφαλμάτων.....	42
Σελίδα Σφάλματος 403.....	42
Σελίδα Σφάλματος 404.....	42
Σελίδα Σφάλματος 500.....	43
Σελίδα Υπόλοιπων Σφαλμάτων.....	43
Αρχιτεκτονική Συστήματος.....	44
Γλώσσες, Ρυθμίσεις και Πακέτα	44
Δομή Εφαρμογής.....	45
Entities/Models	46
Repositories	48
Services	49
Controllers.....	54
Configs.....	57
Templates.....	58
Static Files.....	64
Application Properties	65
Δομή Βάσης Δεδομένων	66
Συμπεράσματα και μελλοντικές επεκτάσεις.....	67
Βιβλιογραφία.....	68

Περίληψη

Η μεταπτυχιακή διατριβή, συνεπάγεται την ανάπτυξη και υλοποίηση μιας διαδικτυακής εφαρμογής, ειδικά σχεδιασμένης ως ψηφιακής εκπαιδευτικής πλατφόρμας, με επίκεντρο την ασύγχρονη διδασκαλία. Η εφαρμογή έχει κατασκευαστεί χρησιμοποιώντας το Spring Boot framework και ενσωματώνει κώδικα γραμμένο σε Java, HTML, CSS και JavaScript. Επιπλέον, χρησιμοποιείται μια βάση δεδομένων MySQL, για αποτελεσματική αποθήκευση δεδομένων.

Προσαρμοσμένη στις ανάγκες των ιδρυμάτων τριτοβάθμιας εκπαίδευσης, αυτή η πλατφόρμα απευθύνεται σε τρεις διαφορετικές κατηγορίες χρηστών: καθηγητές, φοιτητές και επισκέπτες. Οι χρήστες αρχικά ορίζονται ως επισκέπτες, όταν δεν είναι εγγεγραμμένοι ή συνδεδεμένοι, με αποτέλεσμα να έχουν περιορισμένη πρόσβαση. Κατά τη σύνδεση, στους χρήστες ανατίθενται συγκεκριμένοι ρόλοι, παρέχοντάς τους αντίστοιχα προνόμια, όπως δημιουργία μαθημάτων, κεφαλαίων και ερωτήσεων, εγγραφή σε μαθήματα και συμμετοχή σε κουίζ.

Ο πρωταρχικός στόχος της εφαρμογής είναι να ενισχύσει την εκπαιδευτική διαδικασία, διευκολύνοντας την οργάνωση του εκπαιδευτικού υλικού, επιτρέποντας στους μαθητές να μελετούν με τον ρυθμό που προτιμούν και δίνοντας τη δυνατότητα συμπλήρωσης ερωτηματολογίων με αυτοματοποιημένη εξαγωγή αποτελεσμάτων και σχολίων. Αυτή η προσέγγιση παρέχει μια άμεση αξιολόγηση του επιπέδου κατανόησης του εκπαιδευτικού αντικειμένου, συμβάλλοντας έτσι σε μια βελτιωμένη εκπαιδευτική εμπειρία.

Abstract

This master's thesis entails the development and implementation of a web application, specifically designed as a digital educational platform focused on asynchronous teaching. The application is constructed utilizing the Spring Boot framework and incorporates code written in Java, HTML, CSS, and JavaScript. Furthermore, a MySQL database is employed for efficient data storage.

Tailored to meet the needs of higher education institutions, this platform caters to three distinct user categories: professors, students, and guests. Users are initially designated as guests when not registered or logged in, consequently having restricted access. Upon logging in, users are assigned specific roles, granting them corresponding privileges such as creating lessons, chapters, and questions, enrolling in courses, and participating in quizzes.

The overarching objective of the application is to enhance the educational process by facilitating the organization of educational material, allowing students to study at their preferred pace, and enabling the completion of questionnaires with automated result extraction and comments. This approach provides an immediate gauge of the level of understanding of the educational matter, thereby contributing to an improved educational experience.

Εισαγωγή

Η εκπαίδευση, αποτελεί ένα θεμελιώδες δικαίωμα του ανθρώπου, το οποίο έχει θεσπιστεί από την Ευρωπαϊκή Ένωση, με το Άρθρο 14, που εμπνεύστηκε από κοινές παραδόσεις των συνταγμάτων των κρατών μελών της (FRA, 2007). Το γεγονός αυτό, αποδεικνύει ότι πρόκειται για έναν από τους σημαντικότερους πυλώνες της κοινωνίας. Κατά συνέπεια, η διασφάλιση της ομαλής λειτουργίας της, είναι σημαντικό τόσο να προστατεύεται όσο και να ενισχύεται, ώστε με το πέρασμα των χρόνων να αναπτύσσεται και να βελτιώνεται όλο και περισσότερο.

Ένας σύμμαχος της εκπαίδευσης, απέναντι σε οποιαδήποτε απειλή που μπορεί να τη διακόψει ή να μην επιτρέψει τη τέλεση της, όπως μια πανδημία, τα έντονα καιρικά φαινόμενα ή η απόσταση διαμονής ενός ανθρώπου από κάποιον φορέα της, είναι η τεχνολογία. Ένα από τα προϊόντα της τεχνολογίας, που τα τελευταία χρόνια έχει επιστρατευτεί για την προστασία και τη βελτίωση της εκπαιδευτικής διαδικασίας, είναι η ανάπτυξη ψηφιακών εκπαιδευτικών πλατφορμών.

Λαμβάνοντας υπόψιν τη σημασία των παραπάνω αναφορών και σε συνδυασμό με το ιδιαίτερο ενδιαφέρον που προσφέρει ο συνδυασμός του τομέα της Εκπαίδευσης και της Πληροφορικής, αναπτύχθηκε η συγκεκριμένη εκπαιδευτική πλατφόρμα.

Οι λειτουργίες που έχουν προστεθεί, όσο και η ασφάλεια των προσωπικών πληροφοριών του χρήστη, που έχει διασφαλιστεί μέσω συγκεκριμένων τεχνικών, οι οποίες θα αναφερθούν στη συνέχεια, έχουν εμπνευστεί από παρόμοιες εφαρμογές, που χρησιμοποιούνται σε όλες τις βαθμίδες εκπαίδευσης, μερικές από τις οποίες παρουσιάζονται στο κεφάλαιο «Ανασκόπηση Πεδίου». Μερικές από τις βασικές ανάγκες που καλύπτει η εφαρμογή, είναι οι παρακάτω:

- Ασύγχρονη διδασκαλία.
- Εξ' αποστάσεως εκπαίδευση.
- Οργάνωση του εκπαιδευτικού υλικού σε διακριτές ενότητες.
- Επιλογή ρυθμού κάλυψης της ύλης από τον ίδιο τον «φοιτητή».
- Δυνατότητα επανάληψης της ύλης.
- Δημιουργία ερωτηματολογίων αξιολόγησης ανά κεφάλαιο.
- Ολοκλήρωση ερωτηματολογίων και άμεση ενημέρωση των αποτελεσμάτων και των προτεινόμενων ενεργειών βάσει των αποτελεσμάτων αυτών.
- Συνολική επισκόπηση των αποτελεσμάτων από τον «φοιτητή» και τον «καθηγητή».

Η παρούσα υλοποίηση, καθώς έγινε στα πλαίσια μεταπτυχιακής διατριβής και κατά συνέπεια ο σκοπός της ήταν εκπαιδευτικός, έχει αρκετά περιθώρια ανάπτυξης ως προς το πλήθος των δυνατοτήτων που προσφέρει. Ωστόσο, οι λειτουργίες που έχουν προστεθεί, έχουν αναπτυχθεί σε βάθος και έχει γίνει χρήση σύγχρονων τεχνικών ανάπτυξης λογισμικού, για τη διασφάλιση τόσο της ποιότητας και της ασφάλειας της εφαρμογής, όσο και της θετικής εμπειρίας που βιώνει ο χρήστης, κατά την διάρκεια πλοήγησής του σε αυτήν.

Ανασκόπηση Πεδίου

Η σημασία των εκπαιδευτικών πλατφορμών φάνηκε σε μεγάλο βαθμό κατά την περίοδο της πανδημίας του COVID, όπου η διακοπή της λειτουργίας των εκπαιδευτικών δομών είχε αποφασιστεί, ως μέτρο περιορισμού της διάδοσης του, αλλά δεν υπήρχε ακριβής εκτίμηση για το πότε θα γίνει η επανέναρξη τους. Φυσικά, εκπαιδευτικές πλατφόρμες είχαν αναπτυχθεί χρόνια πριν, με αποτέλεσμα να υπάρχει η τεχνογνωσία για τη γρηγορότερη ανάπτυξη νέων εφαρμογών. Αυτό που δεν είχε ξανασυμβεί, ήταν η απότομη αύξηση των χρηστών τους, γεγονός που οδήγησε και τις υπάρχουσες εφαρμογές, να δοκιμάσουν τη δυναμική τους και να διορθώσουν τυχόν ελλείψεις τους. Μερικά παραδείγματα τέτοιου είδους εφαρμογών, από την ελληνική επικράτεια είναι τα παρακάτω:

Ψηφιακή Εκπαιδευτική Πλατφόρμα e-me

<https://e-me.edu.gr/>

Η Ψηφιακή Εκπαιδευτική Πλατφόρμα e-me αναπτύχθηκε από το ΙΤΥΕ «ΔΙΟΦΑΝΤΟΣ», πρόκειται για μια ψηφιακή πλατφόρμα που έχει αναπτυχθεί για την υποστήριξη τυπικής, μη-τυπικής και άτυπης μάθησης, για «φοιτητές» και εκπαιδευτικού. Η συγκεκριμένη πλατφόρμα πρωτοεμφανίστηκε 21 Σεπτεμβρίου 2015 και πλέον βρίσκεται στην έκδοση 3.2, μετρώντας 538.571 «φοιτητές», 132.245 εκπαιδευτικούς και 184.344 κυψέλες (χώρων συνεργασίας). Η συγκεκριμένη πλατφόρμα είναι μια από τις εφαρμογές ασύγχρονης διδασκαλίας, που επιστρατεύτηκε από το Υπουργείο Παιδείας & Θρησκευμάτων (ΥΠΑΙΘ) την περίοδο του covid, για τα Ελληνικά σχολεία. Μερικές από τις πιο σημαντικές λειτουργίες που υποστηρίζει είναι οι παρακάτω:

- Η δημιουργία χώρων συνεργασίας (κυψελών), ιδιωτικών και δημόσιων.
- Η επικοινωνία και κοινωνική δικτύωση των χρηστών της.
- Η οργάνωση, αποθήκευση και διαμοιρασμός αρχείων στο περιβάλλον του υπολογιστικού νέφους (cloud).
- Η ανάθεση και παρακολούθηση εργασιών.
- Η δημιουργία προσωπικών ιστολογίων και ιστολογίων κυψέλης (blogs)

(e-me, 2023)

GUNet2 eClass - Τμήμα Πληροφορικής

<https://gunet2.cs.unipi.gr/>

Η συγκεκριμένη πλατφόρμα αναπτύχθηκε και υποστηρίζεται από το Ακαδημαϊκό Διαδίκτυο GUnet όπου ιδρύθηκε το 2000 και έχει ως μέλη όλα τα ακαδημαϊκά ιδρύματα της χώρας. Η έκδοση στην οποία βρίσκεται είναι η Open eClass 2.11.3 και μετράει 353 μαθήματα και 7021 χρήστες.

Πρόκειται για ένα ολοκληρωμένο σύστημα διαχείρισης ηλεκτρονικών μαθημάτων και ασύγχρονης διδασκαλίας, με χρήστες τόσο «καθηγητές» και «φοιτητές» όσο και επισκέπτες. Η φιλοσοφία της πλατφόρμας που ακολουθείται, είναι το ανοικτού κώδικα λογισμικού, γεγονός που δίνει τη δυνατότητα στους επισκέπτες να έχουν πρόσβαση στο μεγαλύτερο σύνολο των μαθημάτων. Οι εγγεγραμμένοι χρήστες είναι καθηγητές και φοιτητές του Τμήματος Πληροφορικής του Πανεπιστημίου Πειραιώς και μερικές από τις δυνατότητες που έχουν είναι οι παρακάτω:

- Δημιουργία μαθημάτων και ανακοινώσεων.
- Επισύναψη υλικού για το κάθε μάθημα.
- Εγγραφή σε μαθήματα.
- Λήψη ειδοποιήσεων για νέες ανακοινώσεις κάθε μαθήματος στο δηλωμένο email.
- Επισύναψη εργασιών ανά μάθημα.

(GUNet2, 2023) | (Open-eClass, 2023)

Brainy

<https://brainy.gr/>

Πρόκειται για μια συνδρομητική εκπαιδευτική πλατφόρμα, η οποία έκανε και αυτή την εμφάνιση της την περίοδο του covid και αποτελεί ένα σύγχρονο ψηφιακό περιβάλλον, απόλυτα εξατομικευμένο στις ανάγκες του μαθητή. Σε αντίθεση με τις προηγούμενες δύο εφαρμογές, οι χρήστες είναι μόνο φοιτητές των βαθμίδων του Δημοτικού και Γυμνασίου, ενώ το υλικό διαμορφώνεται από την ομάδα της brainy η οποία αποτελείται από έμπειρους δασκάλους και καθηγητές. Μερικές από τις δυνατότητες που προσφέρει στον μαθητή είναι οι παρακάτω:

- Παρουσίαση της ύλης των μαθημάτων σε video και quizzes.
- Εξατομικευμένη μάθηση, με ευελιξία στο μέρος, τον χρόνο και τον τρόπο μελέτης του υλικού από τον μαθητή.
- Αποτελεσματική κατανόηση και αφομοίωση των πολύπλοκων εννοιών που εμφανίζονται στην σχολική ύλη.
- Συνδρομή που αλλάζει τα δεδομένα του οικογενειακού προϋπολογισμού.

(Brainy, 2023)

Παρουσίαση και Χρήση Εφαρμογής

Η εφαρμογή, όπως προαναφέρθηκε, έχει τρεις κατηγορίες/ρόλους χρηστών. Τους «καθηγητές», τους «φοιτητές» και τους επισκέπτες. Η παρουσίαση και χρήση της εφαρμογής, έχει διαρθρωθεί αρχικά σε ενότητες με βάση τον ρόλο του χρήστη και στη συνέχεια με βάση τις σελίδες που μπορεί να επισκεφθεί ο κάθε χρήστης. Στις υποενότητες των σελίδων πλοήγησης, αναλύονται και οι δυνατότητες που προσφέρει η εφαρμογή στον χρήστη, αναλόγως τον ρόλο του.

Πλοήγηση ως «επισκέπτης»

Στην κατηγορία «επισκέπτης», ανήκουν όσοι χρήστες δεν έχουν πραγματοποιήσει εγγραφή ή σύνδεση. Στην περίπτωση αυτή, ο χρήστης έχει περιορισμένες επιλογές πλοήγησης μεταξύ των σελίδων της εφαρμογής, οι οποίες είναι οι εξής:

1. Αρχική Σελίδα
2. Εγγραφή ανά Ιδιότητα
3. Φόρμα Εγγραφής Καθηγητή
4. Φόρμα Εγγραφής Φοιτητή
5. Φόρμα Σύνδεσης

Όλες οι παραπάνω σελίδες, αποτελούνται από τρεις περιοχές. Το navigation bar, το content και το footer.

Το navigation bar, είναι κοινό για όλες τις σελίδες του «επισκέπτη» και περιλαμβάνει ένα button στο αριστερό μέρος, το οποίο αποτελείται από το logo και το όνομα της εφαρμογής. Η λειτουργικότητα του είναι η επαναφορά του χρήστη στην αρχική σελίδα. Στην περίπτωση της αρχικής σελίδα, δεν είναι ορατή η λειτουργικότητα του, αλλά στις περιπτώσεις όπου βρίσκεται σε οποιαδήποτε άλλη σελίδα, από τις υπόλοιπες τέσσερις, μόλις το πατήσει, θα μεταφερθεί στην αρχική σελίδα.



Figure 1: Navigation Bar 1

Το footer είναι κοινό για όλες τις σελίδες της εφαρμογής και για όλους τους χρηστών, ανεξαρτήτως ρόλου. Χωρίζεται σε δύο μέρη. Στο αριστερό μέρος βρίσκεται ένα λεκτικό, που αποτελεί τα copyrights της ιστοσελίδας. Στο λεκτικό αυτό υπάρχει ένας αριθμός, ο οποίος αλλάζει δυναμικά και εμφανίζει το έτος, στο οποίο βρισκόμαστε. Στο δεξί μέρος, υπάρχουν δύο button, τα οποία οδηγούν στο GitHub και το LinkedIn account του διαχειριστή της εφαρμογής.



Figure 2: Footer

Οι περισσότερες διαφορές ανά σελίδα, εντοπίζονται στο content και αναλύονται παρακάτω.

Αρχική Σελίδα

Το content της αρχικής σελίδας είναι χωρισμένο σε δύο περιοχές. Η πρώτη περιοχή, που βρίσκεται στο πάνω μέρος της σελίδας, είναι προσαρμοσμένη ανά κατηγορία χρήστη και κατά συνέπεια διαφέρει από ρόλο σε ρόλο. Στην περίπτωση του «επισκέπτη», αποτελείται από μια εικόνα στα αριστερά και έναν πίνακα στα δεξιά. Ο πίνακας αυτός περιλαμβάνει ένα header με μη προσωποποιημένο μήνυμα καλωσορίσματος του «επισκέπτη» στην εφαρμογή και ένα body, το οποίο αποτελείται από δύο buttons, που οδηγούν στις σελίδες «Εγγραφή ανά Ιδιότητα» και «Φόρμα Σύνδεσης», αντίστοιχα. Η δεύτερη περιοχή, που βρίσκεται από κάτω, είναι κοινή για όλους τους ρόλους χρηστών και αποτελείται από δύο μέρη. Στο πρώτο μέρος υπάρχει ένα στατικό κείμενο, που περιγράφει με σύντομο τρόπο την εφαρμογή και στο δεύτερο μέρος υπάρχουν τρία blocks, όπου παρουσιάζουν τον συνολικό αριθμό των μαθημάτων, των καθηγητών και των φοιτητών της πλατφόρμας, τη δεδομένη χρονική στιγμή. Ο αριθμός κάθε block, αποτελεί animation και πραγματοποιεί μέτρηση από το 0 έως τον αριθμό συνόλου της οντότητας που παρουσιάζει το block, αυξάνοντας κατά μία μονάδα σε κάθε ορισμένη στιγμή. Συνεπώς, ο αριθμός

για κάθε οντότητα είναι δυναμικός και αλλάζει με βάση την προσθήκη ή την αφαίρεση ενός αντικειμένου για κάθε μια από αυτές. Κάτω από κάθε αριθμό βρίσκεται το όνομα της οντότητας.

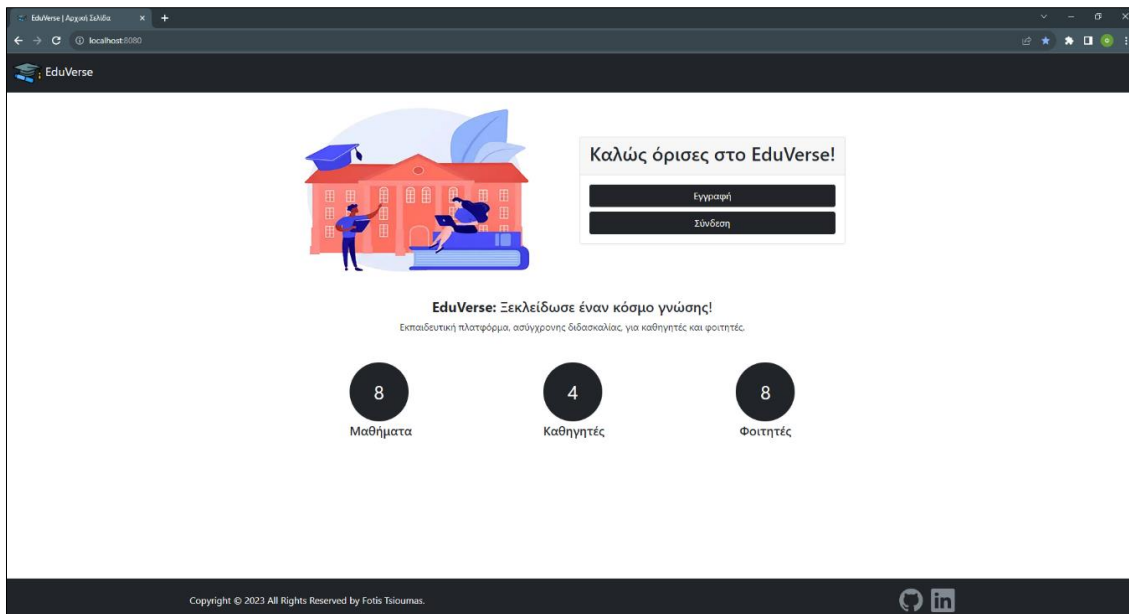


Figure 3: Αρχική Σελίδα – «επισκέπτης»

Εγγραφή ανά Ιδιότητα

Στο content της σελίδας αυτής, υπάρχει πίνακας με δύο buttons, τα οποία οδηγούν στις σελίδες «Φόρμα Εγγραφής Καθηγητή» και «Φόρμα Εγγραφής Φοιτητή», ενώ δεξιά από τον πίνακα υπάρχει μια εικόνα.

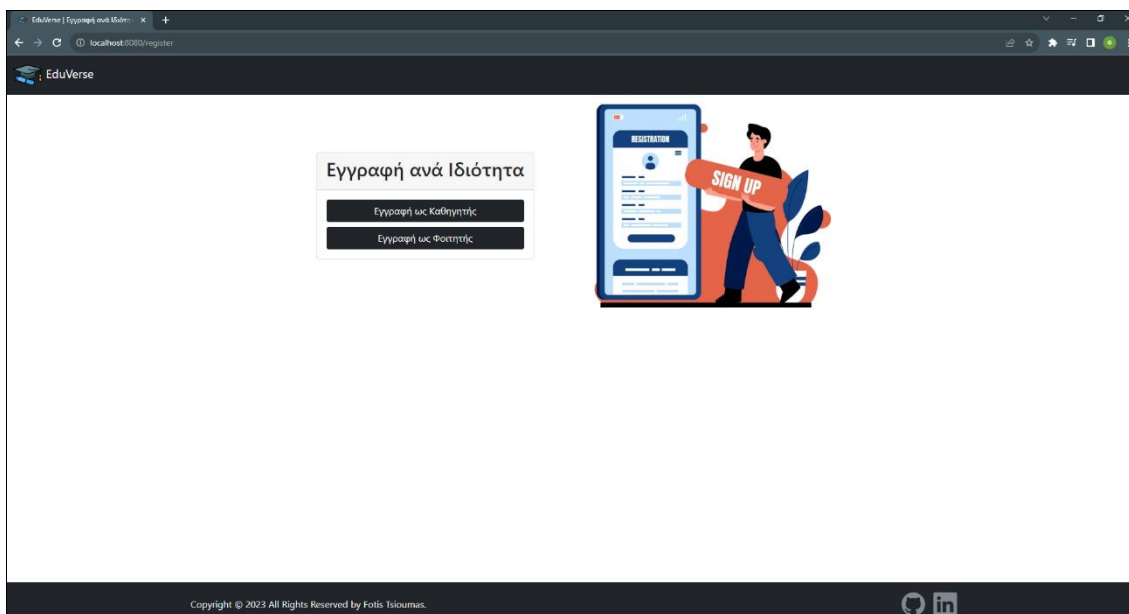


Figure 4: Εγγραφή ανά Ιδιότητα

Φόρμα Εγγραφής Καθηγητή

Το content της σελίδας αποτελείται από μια εικόνα στα αριστερά και μια φόρμα εγγραφής χρήστη στα δεξιά, η οποία αφορά την εγγραφή χρηστών, που μετά την εγγραφή θα πάρουν τον ρόλο «καθηγητή».

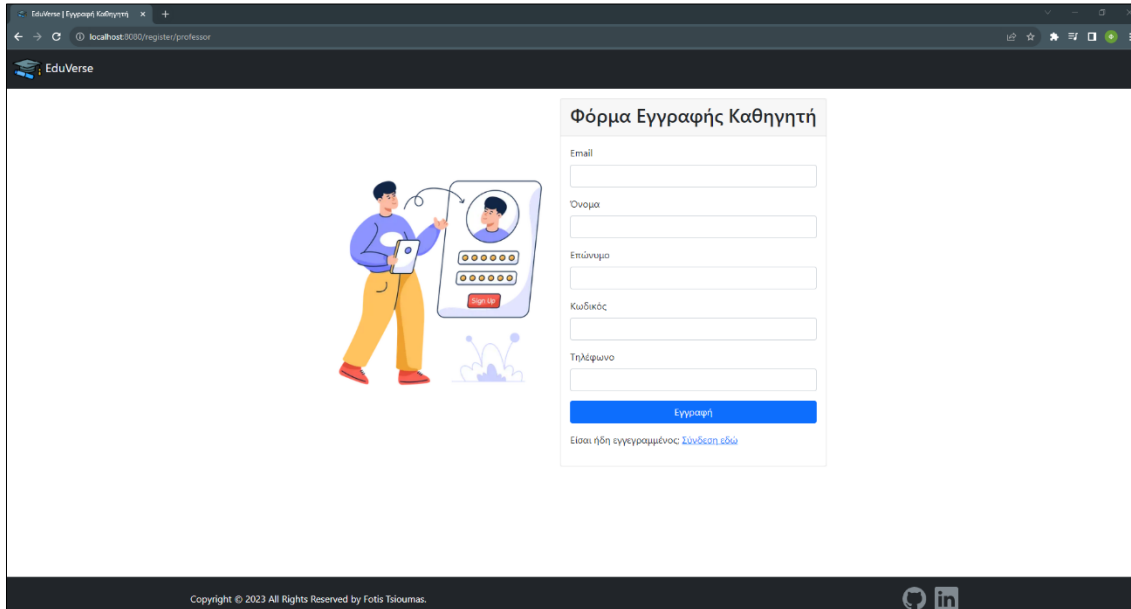


Figure 5: Φόρμα Εγγραφής Καθηγητή

Η φόρμα εγγραφής καθηγητή περιλαμβάνει πέντε πεδία συμπλήρωσης (email, όνομα, επώνυμο, κωδικός, τηλέφωνο), εκ των οποίων τα πρώτα τέσσερα είναι υποχρεωτικά και το πεδίο email έχει και το επιπλέον χαρακτηριστικό του μοναδικού. Ακόμη, υπάρχει το button υποβολής της φόρμας και κάτω από το button υπάρχει ένα λεκτικό με έναν υπερσύνδεσμο, όπου οδηγεί τον χρήστη στην σελίδα σύνδεσης σε περίπτωση που έχει ήδη λογαριασμό. Κατά την υποβολή της φόρμας πατώντας το button «Εγγραφή», ενεργοποιούνται τα παρακάτω validations:

- Τα υποχρεωτικά πεδία ελέγχονται να μην είναι κενά.
- Το πεδίο email ελέγχεται να μην υπάρχει ήδη.

Αν κάποιος κανόνας παραβιαστεί, τότε εμφανίζεται μήνυμα κάτω από το αντίστοιχο πεδίο και δεν επιτρέπει την υποβολή της. Αν ο χρήστης συμπληρώσει σωστά όλα τα στοιχεία και υποβάλει την φόρμα, ανακατευθύνεται στην σελίδα σύνδεσης, όπου ενημερώνεται με μήνυμα στο πάνω μέρος της σελίδας για την επιτυχή εγγραφή του και την δυνατότητα σύνδεσης με τα στοιχεία του.

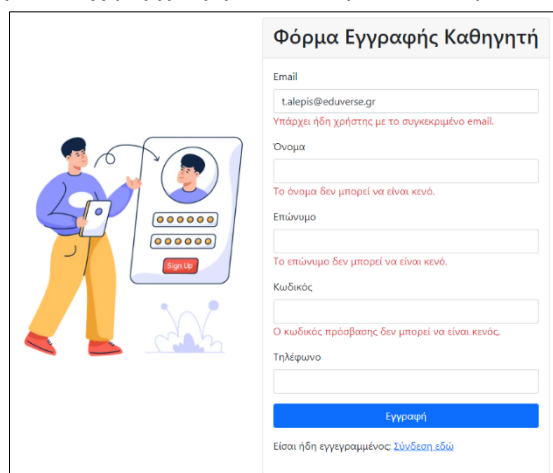


Figure 6: Ανεπιτυχής Εγγραφή Καθηγητή

Έχεις εγγραφεί επιτυχώς στο EduVerse! Συνδέσου για να εξερευνήσεις την εφαρμογή!

Φόρμα Σύνδεσης

Email

Κωδικός

Σύνδεση

Δεν έχεις λογαριασμό;
[Εγγραφή ως Καθηγητής](#) | [Εγγραφή ως Φοιτητής](#)




Figure 7: Επιτυχής Εγγραφή Καθηγητή

Φόρμα Εγγραφής Φοιτητή

Η φόρμα αυτής της σελίδας, έχεις ακριβώς τα ίδια χαρακτηριστικά με την προηγούμενη φόρμα, με μόνες διαφορές, ότι δεν περιλαμβάνει το πεδίο «τηλέφωνο» και ότι μετά την επιτυχή σύνδεση ο χρήστης αποκτά τον ρόλο «φοιτητή».

ΕduVerse

Φόρμα Εγγραφής Φοιτητή

Email
ftsioumas@eduverse.gr

Όνομα
Φώτιος

Επίπλυμο
Τσιούμας

Κωδικός

Εγγραφή

Είσαι ήδη εγγεγραμμένος; [Συνδέσε-σέλι](#)

Copyright © 2023 All Rights Reserved by Fotis Tsioumas.

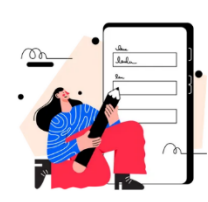


Figure 8: Φόρμα Εγγραφής Φοιτητή

Φόρμα Σύνδεσης

Η συγκεκριμένη φόρμα περιλαμβάνει δύο πεδία (email και κωδικό) και το κείμενο κάτω από το button υποβολής, περιλαμβάνει δύο υπερσυνδέσμους για ανακατεύθυνση στις σελίδες εγγραφής ως «καθηγητής» ή ως «φοιτητής», αντίστοιχα. Η ειδοποιός διαφορά της συγκεκριμένης φόρμας σε σχέση με τις δύο προηγούμενες, είναι ότι αποτελεί φόρμα σύνδεσης ήδη εγγεγραμμένου χρήστη.

Κατά την υποβολή της συγκεκριμένης φόρμας, ελέγχεται αν υπάρχει εγγεγραμμένος χρήστης με τα στοιχεία που δηλώθηκαν. Αν δεν υπάρχει τότε εμφανίζεται στο πάνω μέρος της σελίδας αντίστοιχο μήνυμα, ενώ αν υπάρχει, ο χρήστης οδηγείται στην αρχική σελίδα, συνδεδεμένος πλέον.

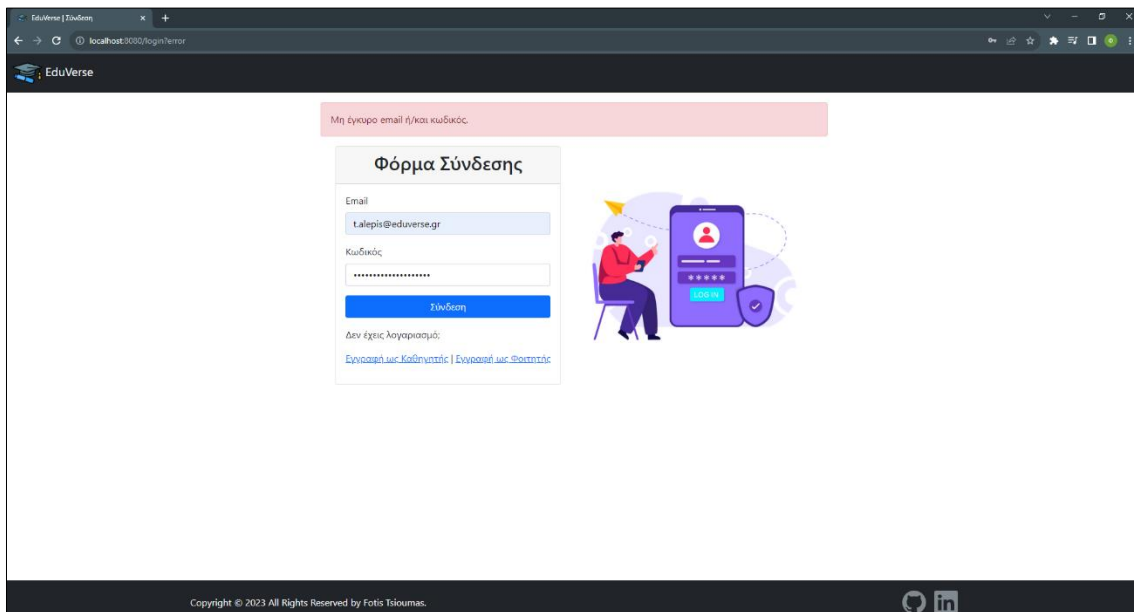


Figure 9: Ανεπιτυχής Σύνδεση Χρήστη

Στη σελίδα αυτή, μπορεί να πλοηγηθεί ένας χρήστης με τους παρακάτω τρόπους:

- Ως «επισκέπτης», μέσω της αρχικής σελίδας.
- Ως «επισκέπτης», μέσω των σελίδων εγγραφής ως «καθηγητής» και ως «φοιτητής».
- Ως «επισκέπτης», αν προσπαθήσει να πλοηγηθεί μέσω url, σε οπουδήποτε άλλη σελίδα, εκτός από τις σελίδες που έχει πρόσβαση.
- Ως συνδεδεμένος χρήστης, αν πραγματοποιήσει αποσύνδεση. Σε αυτή την περίπτωση ενημερώνεται με μήνυμα στο πάνω μέρος της οθόνης για την επιτυχή αποσύνδεσή του.

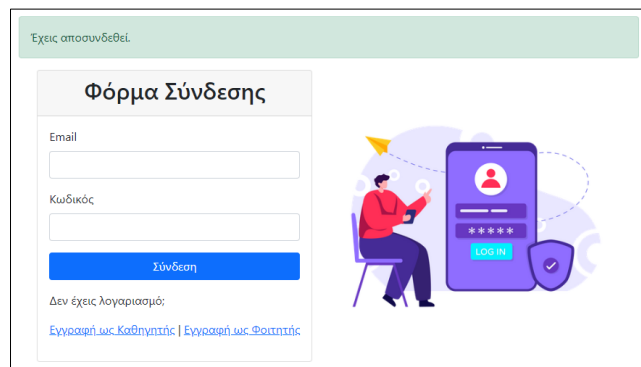


Figure 10: Επιτυχής Αποσύνδεση Χρήστη

Πλοήγηση ως «καθηγητής»

Στην κατηγορία «καθηγητής», ανήκουν όσοι χρήστες έχουν πραγματοποιήσει εγγραφή μέσω της «Φόρμας Εγγραφής Καθηγητή». Στην περίπτωση αυτή, ο χρήστης έχει αυξημένες επιλογές πλοήγησης μεταξύ των σελίδων της εφαρμογής, οι οποίες είναι οι εξής:

1. Αρχική Σελίδα
2. Δημιουργία Νέου Μαθήματος
3. Τα Μαθήματά Μου
4. Προβολή Μαθήματος
5. Επεξεργασία Μαθήματος
6. Δημιουργία Νέου Κεφαλαίου
7. Προβολή Κεφαλαίου
8. Επεξεργασία Κεφαλαίου
9. Προβολή Ερωτήσεων Κεφαλαίου
10. Δημιουργία Νέας Ερώτησης
11. Επεξεργασία Ερώτησης
12. Οι Φοιτητές Μου
13. Προβολή Αποτελεσμάτων Φοιτητή
14. Προβολή Ερωτηματολογίου
15. Ο Λογαριασμός Μου
16. Επεξεργασία Πληροφοριών Λογαριασμού
17. Αλλαγή Κωδικού Πρόσβασης

Όλες οι παραπάνω σελίδες αποτελούνται από τρία μέρη. Το navigation bar, το content και το footer.

Το navigation bar, περιλαμβάνει είτε μόνο το button που οδηγεί στην αρχική σελίδα, το οποίο αναλύσαμε στην περίπτωση του «επισκέπτη» και ένα button αποσύνδεσης στο δεξί μέρος, είτε περιλαμβάνει στα δύο άκρα τα παραπάνω buttons και στη μέση εμφανίζει τις κεντρικές σελίδες πλοήγησης του «καθηγητή».



Figure 11: Navigation Bar 2

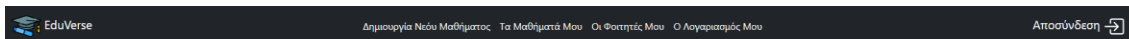


Figure 12: Navigation Bar 3

Το footer, όπως προαναφέρθηκε στην περίπτωση του «επισκέπτη», είναι κοινό για όλες τις σελίδες και στην περίπτωση του καθηγητή. Οι περισσότερες διαφορές ανά σελίδα, εντοπίζονται στο content και αναλύονται παρακάτω.

Αρχική Σελίδα

Το content της αρχικής σελίδας του «καθηγητή», παρουσιάζει ακριβώς την ίδια δομή με αυτή του «επισκέπτη», με μόνες διαφορές στον πίνακα του πρώτου μέρους της σελίδας, τα παρακάτω σημεία:

- Το μήνυμα στο header του πίνακα είναι προσωποποιημένο και εμφανίζει το όνομα του χρήστη.
- Στο body του πίνακα εμφανίζονται με τη μορφή button, οι κεντρικές σελίδες πλοήγησης του χρήστη. Αυτά τα buttons, μεταφέρονται στις υπόλοιπες σελίδες, ως items στο κεντρικό μέρος του navigation bar.

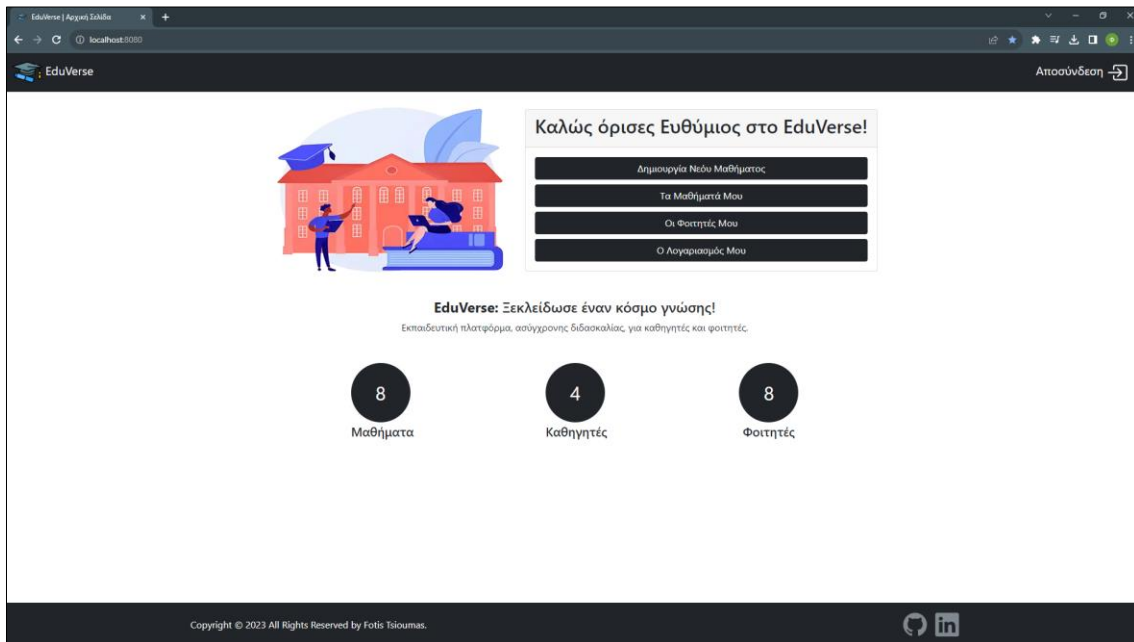


Figure 13: Αρχική Σελίδα – «καθηγητής»

Δημιουργία Νέου Μαθήματος

Στη σελίδα αυτή έχουν πρόσβαση μόνο οι χρήστες με ρόλο «καθηγητή». Το content της αποτελείται από μια φόρμα δημιουργίας μαθήματος στα αριστερά και μια εικόνα στα δεξιά. Η συγκεκριμένη φόρμα αποτελείται από τρία υποχρεωτικά πεδία (κωδικός, τίτλος και σύντομη περιγραφή), τα οποία είναι τα χαρακτηριστικά του μαθήματος. Επίσης περιλαμβάνει δύο buttons, ένα για την υποβολή της φόρμας και ένα για την ακύρωση δημιουργίας μαθήματος, όπου οδηγεί στην αρχική σελίδα, χωρίς την υποβολή της φόρμας. Κατά την υποβολή της φόρμας πατώντας το button «Δημιουργία», ενεργοποιούνται τα παρακάτω validations:

- Τα υποχρεωτικά πεδία ελέγχονται να μην είναι κενά.
- Το πεδίο κωδικός ελέγχεται να μην υπάρχει ήδη.

Αν κάποιος κανόνας παραβιαστεί, τότε εμφανίζεται μήνυμα κάτω από το αντίστοιχο πεδίο και δεν επιτρέπει την υποβολή της. Αν ο χρήστης συμπληρώσει σωστά όλα τα στοιχεία και υποβάλει την φόρμα, ανακατευθύνεται στην σελίδα «Τα Μαθήματά Μου», όπου ενημερώνεται με μήνυμα στο πάνω μέρος της σελίδας για την επιτυχή δημιουργία νέου μαθήματος. Επιπλέον, για να μπορέσει ένας χρήστης να δει τη σελίδα πρέπει να καλύπτει τα παρακάτω κριτήρια:

- Να είναι συνδεδεμένος χρήστης. Αν δεν πληροί το κριτήριο οδηγείται στη σελίδα σύνδεσης.
- Να είναι ο συνδεδεμένος χρήστης με id αυτό που φαίνεται στο url της σελίδας. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι χρήστης με ρόλο «καθηγητή». Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.

Επίπεδο | Δημιουργία Νέου Μ...
localhost:5000/course/new/1
EduVerse
Δημιουργία Νέου Μαθήματος | Τα Μαθήματά Μου | Οι Φοιτητές Μου | Ο Λογαριασμός Μου
Αποσύνδεση

Δημιουργία Νέου Μαθήματος

Κωδικός
TME 105

Τίτλος
ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Σύντομη Περιγραφή
Το μάθημα είναι μεταπτυχιακό και αφορά την εκμάθηση του αντικειμενοστρεφούς προγραμματισμού με βασικό λογισμικό εργαλείο, την JAVA SE7+. Αναπτύσσεται η γλώσσα JAVA 2+ (επίπεδος εκδόσεις) και γενικά η εφαρμογή της αντικειμενοστρεφούς τεχνολογίας σε καθαρά προγραμματιστικό επίπεδο. Είναι προϋπόθεση ο εκπαιδευόμενος να κατέχει τουλάχιστον σε ικανοποιητικό επίπεδο την C/C++, βασικές δομές και ορισμοί, δηλώσεις μεταβλητών, μεθόδων και κλάσεων. Κληρονομικότητα, Εντολές, Πακέτα, Διαφοροποιήσεις, Αρχεία και JAVA. Πολυμέσα, Νήματα. Ειδικά θέματα προγραμματισμού με JAVA 7+.

Δημιουργία
Ακύρωση

Copyright © 2023 All Rights Reserved by Fotis Tsioumas.

Figure 14: Δημιουργία Νέου Μαθήματος

Τα Μαθήματα Μου

Στη συγκεκριμένη σελίδα έχουν πρόσβαση τόσο οι «καθηγητές» όσο και οι «φοιτητές», αλλά το περιεχόμενο του content διαφοροποιείται αναλόγως τον ρόλο. Το content της συγκεκριμένης σελίδας για τον «καθηγητή», αποτελείται από ένα πίνακα που περιλαμβάνει ένα grid με τα μαθήματα που έχει δημιουργήσει ο «καθηγητής» και ένα button στο κάτω μέρος, για δημιουργία νέου μαθήματος. Σε κάθε σειρά του grid υπάρχουν τρεις υπερσύνδεσμοι, οι οποίοι εκτελούν από μια ενέργεια για το κάθε μάθημα, ξεχωριστά. Η προβολή οδηγεί στη σελίδα προβολής του μαθήματος, η επεξεργασία οδηγεί στη σελίδα επεξεργασίας του μαθήματος και η διαγραφή πραγματοποιεί διαγραφή του μαθήματος, αφαιρώντας το μάθημα από το grid και εμφανίζοντας μήνυμα επιτυχημένης διαγραφής στο πάνω μέρος της οθόνης. Για να μπορέσει ένας χρήστης να δει τη σελίδα πρέπει να καλύπτει τα παρακάτω κριτήρια:

- Να είναι συνδεδεμένος χρήστης. Αν δεν πληροί το κριτήριο οδηγείται στη σελίδα σύνδεσης.
- Να είναι ο συνδεδεμένος χρήστης με id αυτό που φαίνεται στο url της σελίδας. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.

#	Κωδικός	Τίτλος	Περιγραφή	Ενέργεια
1	TME105	ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ	Το μάθημα είναι μεταπτυχιακό και αφορά την εκμάθηση του αντικειμενοστρεφούς προγραμματισμού με βασικό λογισμικό εργαλείο, την JAVA SE.7+. Αναπτύσσεται η γλώσσα JAVA 2+ (νιότερες εκδόσεις) και γενικά η εφαρμογή της αντικειμενοστρεφούς τεχνολογίας σε καθαρά προγραμματιστικό επίπεδο. Είναι προϋπόθεση ο εκπαιδευόμενος να κατέχει τουλάχιστον σε ικανοποιητικό επίπεδο την C/C++, βασικές δομές και ορισμοί. Δηλώσεις μεταβλητών, μεθόδων και κλάσεων. Κληρονομικότητα. Εντολές, Πακέτα, Διαπροσωπικές, Αρχεία και JAVA. Πολυμέσα. Νήματα. Ειδικά θέματα προγραμματισμού με JAVA 7+.	Προβολή Επεξεργασία Διαγραφή
2	TME107	ΑΡΧΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C/C++	Το μάθημα Αρχές Προγραμματισμού απευθύνεται στους Σπουδαστές του μεταπτυχιακού προγράμματος Πληροφορική και αποτελείται από τρία βασικά στάδια. Στο πρώτο γίνεται εισαγωγή στην τεχνολογία λογισμικού και στους χώρους δεδομένων. Στο δεύτερο διδάσκεται η επίλυση προβλημάτων με Η/Υ. Στο τρίτο στάδιο γίνεται και εισαγωγή στον αντικειμενοστρεφή προγραμματισμό. Το μάθημα αναφέρεται σε μεταπτυχιακό επίπεδο στον σύγχρονο προγραμματισμό, στην τεχνολογία επίλυσης προβλημάτων με Η/Υ και στον σχεδιασμό και ανάπτυξη προγραμμάτων με C. Προαπαιτούμενο θεωρείται να έχει ο διδασκόμενος γνώσεις γενικά στους Η/Υ. Το μάθημα διδάσκεται στο μεταπτυχιακό ΠΛΗΡΟΦΟΡΙΚΗ.	Προβολή Επεξεργασία Διαγραφή
3	TME154	ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ	Αντικείμενο του μαθήματος είναι η ανάπτυξη εφαρμογών οι οποίες μπορούν να εκτελεστούν από σύγχρονες κινητές συσκευές (smartphones) με ενσωματωμένο λειτουργικό σύστημα. Οι εφαρμογές αυτές δύναται να λειτουργήσουν τόσο στα σύγχρονα «έξυπνα» τηλέφωνα «smartphones», όσο και σε άλλες «κινητές» συσκευές οι οποίες έχουν κάνει την εμφάνισή τους τα τελευταία χρόνια και χρησιμοποιούν λειτουργικό σύστημα (Tablets, Wearables). Στο μάθημα αναλύονται περιληπτικά τα δημοφιλέστερα λειτουργικά συστήματα κινητών συσκευών, καθώς και τα εργαλεία ανάπτυξης εφαρμογών σε αυτά, ωστόσο η ύλη περιλαμβάνει κυρίως τη χρήση της αντικειμενοστρεφούς γλώσσας προγραμματισμού Java, για την ανάπτυξη εφαρμογών σε κινητές συσκευές υπό το λειτουργικό σύστημα Android. Διδακτικά η ανάπτυξη των mobile apps θα υλοποιηθεί με τη χρήση του περιβάλλοντος ανάπτυξης λογισμικού Android Studio. Η Java είναι η πρώτη σε χρήση γλώσσα προγραμματισμού παγκοσμίως (δεδομένα 2019), ενώ το λειτουργικό σύστημα Android κατέχει το μεγαλύτερο μερίδιο της παγκοσμίας αγοράς των smartphones (δεδομένα 2019).	Προβολή Επεξεργασία Διαγραφή

Δημιουργία Νέου Μαθήματος

Figure 15: Τα Μαθήματά Μου – «καθηγητής»

Αν ο «καθηγητής» είναι νέος χρήστης και δεν έχει ακόμα κάποιο μάθημα, τότε αντί για το grid με τα μαθήματα, εμφανίζεται στη θέση του ένα μήνυμα που ενημερώνει ότι δεν υπάρχει κάποιο μάθημα.

Τα Μαθήματά Μου

Δεν έχεις ακόμα κάποιο μάθημα...

Δημιουργία Νέου Μαθήματος

Figure 16: Απουσία μαθημάτων – «καθηγητής»

Προβολή Μαθήματος

Στη συγκεκριμένη σελίδα έχουν πρόσβαση τόσο οι «καθηγητές» όσο και οι «φοιτητές», αλλά το περιεχόμενο του content διαφοροποιείται αναλόγως τον ρόλο. Το content της συγκεκριμένης σελίδας για τον καθηγητή, αποτελείται από ένα πίνακα που περιλαμβάνει στο header το όνομα του μαθήματος και στο body, αρχικά την σύντομη περιγραφή του μαθήματος, μετά ένα grid με τα κεφάλαια του μαθήματος και στο τέλος ένα button για δημιουργία νέου κεφαλαίου. Σε κάθε σειρά του grid υπάρχουν τέσσερις υπερσύνδεσμοι, οι οποίοι εκτελούν από μια ενέργεια για το κάθε κεφάλαιο, ξεχωριστά. Η προβολή οδηγεί στη σελίδα προβολής του κεφαλαίου, η επεξεργασία οδηγεί στη σελίδα επεξεργασίας του κεφαλαίου, η διαγραφή πραγματοποιεί διαγραφή του κεφαλαίου, αφαιρώντας το κεφάλαιο από το grid και εμφανίζοντας μήνυμα επιτυχημένης διαγραφής στο πάνω μέρος της οθόνης και οι ερωτήσεις οδηγούν στη σελίδα ερωτήσεων του κεφαλαίου. Για να μπορέσει ένας χρήστης να δει τη σελίδα πρέπει να καλύπτει τα παρακάτω κριτήρια:

- Να είναι συνδεδεμένος χρήστης. Αν δεν πληροί το κριτήριο οδηγείται στη σελίδα σύνδεσης.
- Να είναι ο συνδεδεμένος χρήστης με id αυτό που φαίνεται στο url της σελίδας. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι ο «καθηγητής» που δημιούργησε το μάθημα. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.

The screenshot shows a web browser window displaying the 'View Course' page. The course title is 'ANTIKEIMENOSTREPHIS PROGRAMMATISMOS'. Below the title is a description of the course. A table lists four chapters with their titles and corresponding actions: 'Εισαγωγή', 'Ιστορικό', 'Έννοιες', and 'Αρχές αντικειμενοστρεφούς σχεδίασης'. Each row has four links: 'Προβολή', 'Επεξεργασία', 'Διαγραφή', and 'Ερωτήσεις'. At the bottom of the table is a blue button labeled 'Δημιουργία Νέου Κεφαλαίου'.

Figure 17: Προβολή Μαθήματος – «καθηγητής»

Αν ο «καθηγητής» έχει δημιουργήσει το μάθημα αλλά δεν έχει δημιουργήσει κεφάλαια, τότε αντί για το grid με τα κεφάλαια του μαθήματος, εμφανίζεται στη θέση του ένα μήνυμα που ενημερώνει ότι δεν υπάρχουν ακόμα κεφάλαια.

The screenshot shows the 'View Course' page for 'ANAGNΩΡΙΣΗ ΠΡΟΤΥΠΩΝ'. The description of the course is visible. At the bottom of the page, there is a red-bordered box containing the message 'Δεν υπάρχουν ακόμα κεφάλαια...'. Below this message is a blue button labeled 'Δημιουργία Νέου Κεφαλαίου'.

Figure 18: Απουσία κεφαλαίων – «καθηγητής»

Επεξεργασία Μαθήματος

Στη σελίδα αυτή έχουν πρόσβαση μόνο οι χρήστες με ρόλο «καθηγητή». Το content της είναι όμοιο με το content της σελίδας δημιουργίας νέου μαθήματος, με τη διαφορά ότι τα πεδία είναι συμπληρωμένα με τα στοιχεία του μαθήματος. Στη συγκεκριμένη σελίδα ο «καθηγητής» μπορεί να επεξεργαστεί τις πληροφορίες του μαθήματος και να πατήσει «Αποθήκευση». Κατά την υποβολή της φόρμας πατώντας το button «Αποθήκευση», ενεργοποιούνται τα παρακάτω validations:

- Τα υποχρεωτικά πεδία ελέγχονται να μην είναι κενά.
- Το πεδίο κωδικός ελέγχεται να μην υπάρχει ήδη.

Αν κάποιος κανόνας παραβιαστεί, τότε εμφανίζεται μήνυμα κάτω από το αντίστοιχο πεδίο και δεν επιτρέπει την αποθήκευση των αλλαγών. Αν ο χρήστης συμπληρώσει σωστά όλα τα στοιχεία και υποβάλει την φόρμα, ανακατευθύνεται στην σελίδα «Τα Μαθήματά Μου», όπου ενημερώνεται με μήνυμα στο πάνω μέρος της σελίδας για την επιτυχή επεξεργασία τους μαθήματος. Επιπλέον, για να μπορέσει ένας χρήστης να δει τη σελίδα πρέπει να καλύπτει τα παρακάτω κριτήρια:

- Να είναι συνδεδεμένος χρήστης. Αν δεν πληροί το κριτήριο οδηγείται στη σελίδα σύνδεσης.
- Να είναι ο συνδεδεμένος χρήστης με id αυτό που φαίνεται στο url της σελίδας. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι χρήστης με ρόλο «καθηγητή». Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι ο «καθηγητής» που δημιούργησε το μάθημα. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.

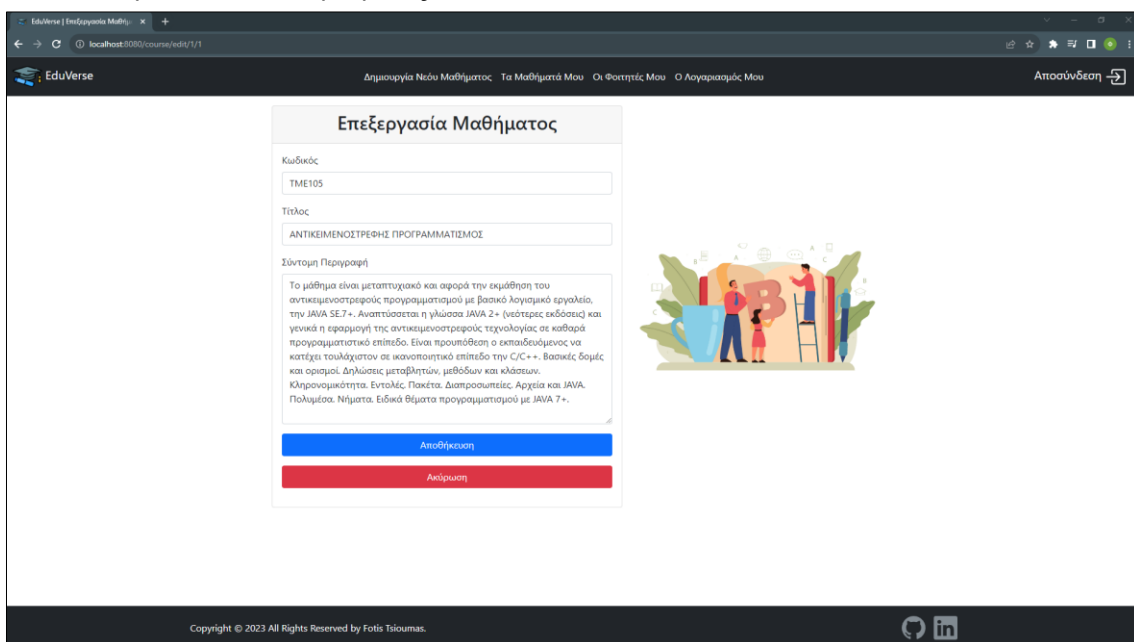


Figure 19: Επεξεργασία Μαθήματος

Δημιουργία Νέου Κεφαλαίου

Στη σελίδα αυτή έχουν πρόσβαση μόνο οι χρήστες με ρόλο «καθηγητή». Το content της αποτελείται από μια φόρμα δημιουργίας μαθήματος στα αριστερά και μια εικόνα στα δεξιά. Η συγκεκριμένη φόρμα αποτελείται από δύο υποχρεωτικά πεδία (τίτλος και περιεχόμενο), τα οποία είναι τα χαρακτηριστικά του κεφαλαίου. Η διαφορά του πεδίου «Περιεχόμενο» της συγκεκριμένης φόρμας σε σχέση με του πεδίου «Σύντομη Περιγραφή» της φόρμας νέου μαθήματος, είναι ότι σε αυτή την περίπτωση ο χρήστης έχει στη διάθεση του έναν rich text editor, ο οποίος του δίνει διάφορα εργαλεία για την καλύτερη δόμηση του περιεχομένου και την πιο 'πλούσια' μορφή του. Επίσης περιλαμβάνει δύο buttons, ένα για την υποβολή της φόρμας και ένα για την ακύρωση δημιουργίας κεφαλαίου, όπου οδηγεί στην προβολή μαθήματος χωρίς την υποβολή της φόρμας. Κατά την υποβολή της φόρμας πατώντας το button «Δημιουργία», ενεργοποιείται το παρακάτω validation:

- Τα υποχρεωτικά πεδία ελέγχονται να μην είναι κενά.

Αν ο κανόνας παραβιαστεί, τότε εμφανίζεται μήνυμα κάτω από το αντίστοιχο πεδίο και δεν επιτρέπεται την υποβολή της. Αν ο χρήστης συμπληρώσει σωστά όλα τα στοιχεία και υποβάλει την φόρμα, ανακατευθύνεται στην σελίδα προβολής μαθήματος, όπου ενημερώνεται με μήνυμα στο πάνω μέρος της σελίδας για την επιτυχή δημιουργία νέου κεφαλαίου. Επιπλέον, για να μπορέσει ένας χρήστης να δει τη σελίδα πρέπει να καλύπτει τα παρακάτω κριτήρια:

- Να είναι συνδεδεμένος χρήστης. Αν δεν πληροί το κριτήριο οδηγείται στη σελίδα σύνδεσης.
- Να είναι ο συνδεδεμένος χρήστης με id αυτό που φαίνεται στο url της σελίδας. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι χρήστης με ρόλο «καθηγητή». Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι ο «καθηγητής» που δημιούργησε το μάθημα. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.

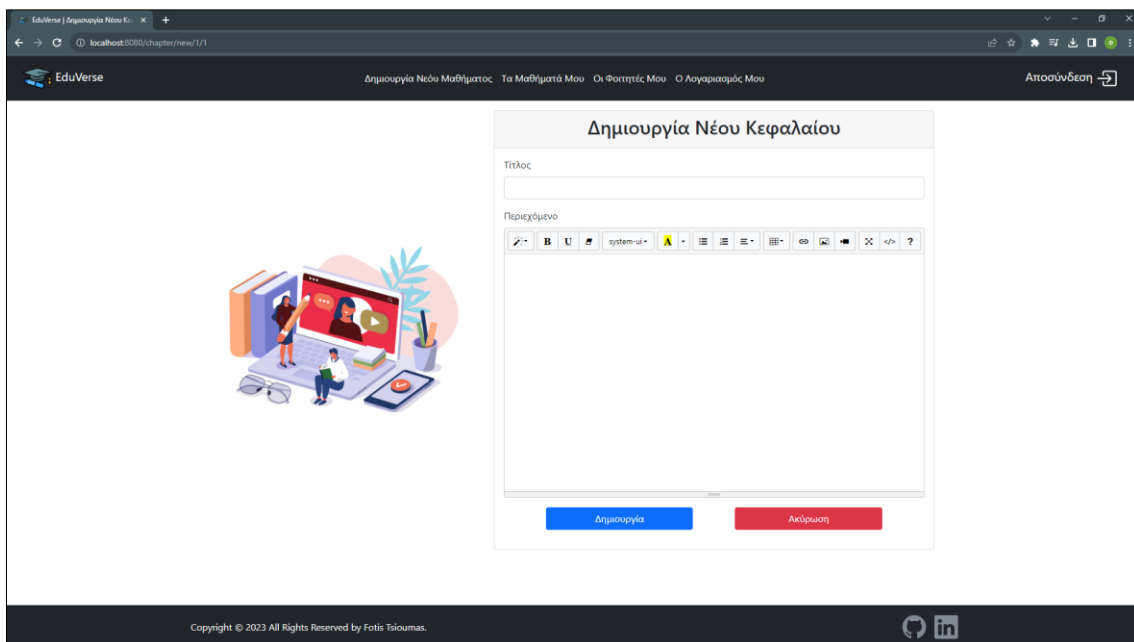


Figure 20: Δημιουργία Νέου Κεφαλαίου

Προβολή Κεφαλαίου

Στη συγκεκριμένη σελίδα έχουν πρόσβαση τόσο οι «καθηγητές» όσο και οι «φοιτητές». Το content της αποτελείται από έναν πίνακα, όπου στο header υπάρχει το όνομα του κεφαλαίου και το όνομα του μαθήματος (με υπερσύνδεσμο προς την προβολή μαθήματος) και στο body υπάρχει το περιεχόμενο του μαθήματος, όπως ακριβώς σχεδιάστηκε κατά τη δημιουργία του. Για να μπορέσει ένας χρήστης να δει τη σελίδα πρέπει να καλύπτει τα παρακάτω κριτήρια:

- Να είναι συνδεδεμένος χρήστης. Αν δεν πληροί το κριτήριο οδηγείται στη σελίδα σύνδεσης.
- Να είναι ο συνδεδεμένος χρήστης με id αυτό που φαίνεται στο url της σελίδας. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι ο «καθηγητής» που δημιούργησε το μάθημα στο οποίο ανήκει το κεφάλαιο. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.

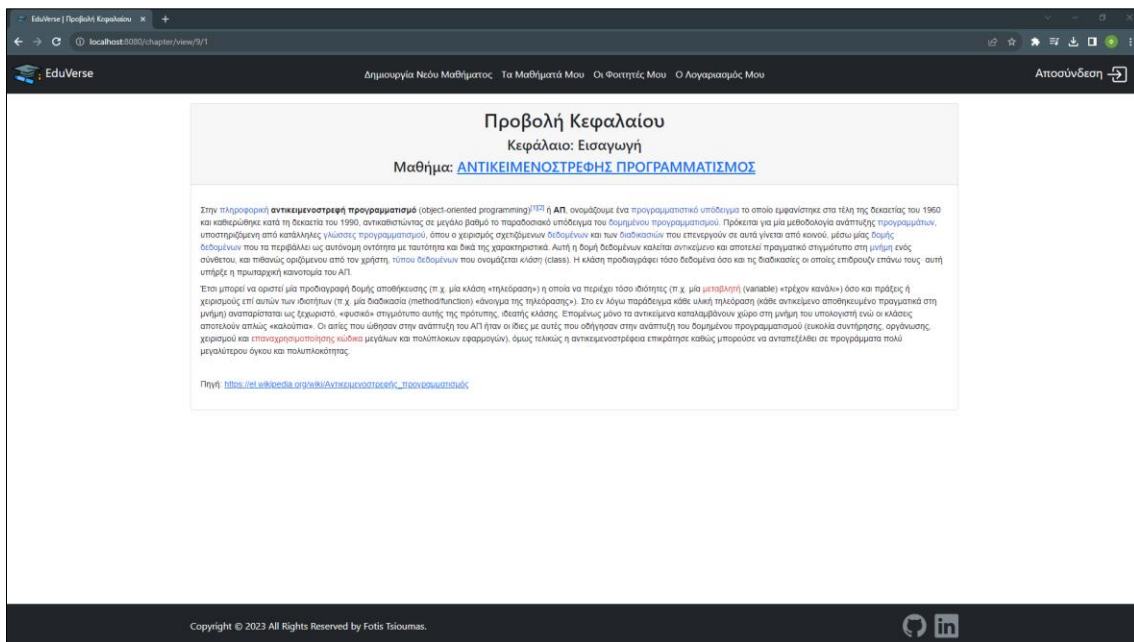


Figure 21: Προβολή Κεφαλαίου – «καθηγητής»

Επεξεργασία Κεφαλαίου

Στη σελίδα αυτή έχουν πρόσβαση μόνο οι χρήστες με ρόλο «καθηγητή». Το content της είναι όμοιο με το content της σελίδας δημιουργίας νέου κεφαλαίου, με τη διαφορά ότι τα πεδία είναι συμπληρωμένα με τα στοιχεία του κεφαλαίου. Στη συγκεκριμένη σελίδα ο «καθηγητής» μπορεί να επεξεργαστεί τις πληροφορίες του κεφαλαίου και να πατήσει «Αποθήκευση». Κατά την υποβολή της φόρμας πατώντας το button «Αποθήκευση», ενεργοποιείται το παρακάτω validation:

- Τα υποχρεωτικά πεδία ελέγχονται να μην είναι κενά.

Αν ο κανόνας παραβιαστεί, τότε εμφανίζεται μήνυμα κάτω από το αντίστοιχο πεδίο και δεν επιτρέπει την αποθήκευση των αλλαγών. Αν ο χρήστης συμπληρώσει σωστά όλα τα στοιχεία και υποβάλει την φόρμα, ανακατευθύνεται στην σελίδα του μαθήματος, όπου ενημερώνεται με μήνυμα στο πάνω μέρος της σελίδας για την επιτυχή επεξεργασίας τους κεφαλαίου. Επιπλέον, για να μπορέσει ένας χρήστης να δει τη σελίδα πρέπει να καλύπτει τα παρακάτω κριτήρια:

- Να είναι συνδεδεμένος χρήστης. Αν δεν πληροί το κριτήριο οδηγείται στη σελίδα σύνδεσης.
- Να είναι ο συνδεδεμένος χρήστης με id αυτό που φαίνεται στο url της σελίδας. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι χρήστης με ρόλο «καθηγητή». Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι ο «καθηγητής» που δημιούργησε το μάθημα στο οποίο ανήκει το κεφάλαιο. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.

The screenshot displays the 'Επεξεργασία Κεφαλαίου' (Edit Chapter) interface in the EduVerse application. The page title is 'Επεξεργασία Κεφαλαίου'. Below the title, there is a 'Τίτλος' (Title) field with the placeholder text 'Εισαγωγή'. The main content area features a rich text editor with a toolbar containing various formatting options like bold, italic, underline, and text color. The editor contains the following text:

Στην πληροφορική **αντικειμενοστρεφή προγραμματισμό** (object-oriented programming) ^{[1][2]} ή **ΑΠ**, αναμόρφωσα ένα προγραμματιστικό υπόδειγμα το οποίο εμφανίστηκε στα τέλη της δεκαετίας του 1960 και καθιερώθηκε κατά τη δεκαετία του 1990, αντικαθιστώντας σε μεγάλο βαθμό το παραδοσιακό υπόδειγμα του δομημένου προγραμματισμού. Πρόκειται για μία μεθοδολογία ανάπτυξης προγραμμάτων, υποστηρίξιμη από κατάλληλες γλώσσες προγραμματισμού, όπου ο χειρισμός σχετιζόμενων δεδομένων και των διαδικασιών που επεξεργάζονται σε αυτά γίνεται από κοινού, μέσω μιας δομής δεδομένων που τα περιβάλλει ως επιτόνια ανάμεσα με ταυτότητα και δικιά της χαρακτηριστικά. Αυτή η δομή δεδομένων καλείται αντικείμενο και αποτελεί πραγματικό σημείο στη μνήμη ενός σιόντιου, και πιθανώς αρθρώνεται από τον χρήστη, όπως δεδομένων που αναμείζονται κλάση (class). Η κλάση προορίζεται τόσο δεδομένα όσο και τις διαδικασίες οι οποίες επηρεάζουν επάνω τους, αυτή υπάρχει η πρωταρχική κανονιστική του ΑΠ.

Είπα μπορεί να οριστεί μία προαγωγή δομής αποθήκευσης (π.χ. μία κλάση «ηλεκτρονική») η οποία να περιέχει τόσο ιδιότητες (π.χ. μία **μεταβλητή** (variable) «τρέχον κανάλι») όσο και πράξεις ή χειρισμούς επί αυτών των ιδιοτήτων (π.χ. μία διαδικασία (method/function) «δυναμική της ηλεκτρονικής»). Στο εν λόγω παράδειγμα κάθε ηλεκτρονική (κάθε αντικείμενο αποθηκευμένο πραγματικά στη μνήμη) αναπαράστα ως εξαρτημένο, «φυσικό» σημείο από τις πρώτες, δεύτερες κλάσεις. Επομένως μόνο τα αντικείμενα καταλαμβάνουν χώρο στη μνήμη του υπολογιστή ενώ οι κλάσεις αποτελούν απλά «καλούπια». Οι αυτές που ώθησαν στην ανάπτυξη του ΑΠ ήταν οι ίδες με αυτές που οδήγησαν στην ανάπτυξη του δομημένου προγραμματισμού (ελακτά συντήρησης, οργάνωσης, χειρισμού και επαναχρησιμοποίησης κώδικα μεγάλων και ποικίλων εφαρμογών), όμως τελικώς η αντικειμενοστρεφής επεξεργασία καθώς μπορούσε να αναπαράγει σε προγράμματα πολύ μεγαλύτερου όγκου και ποικιλοεπίτητες.

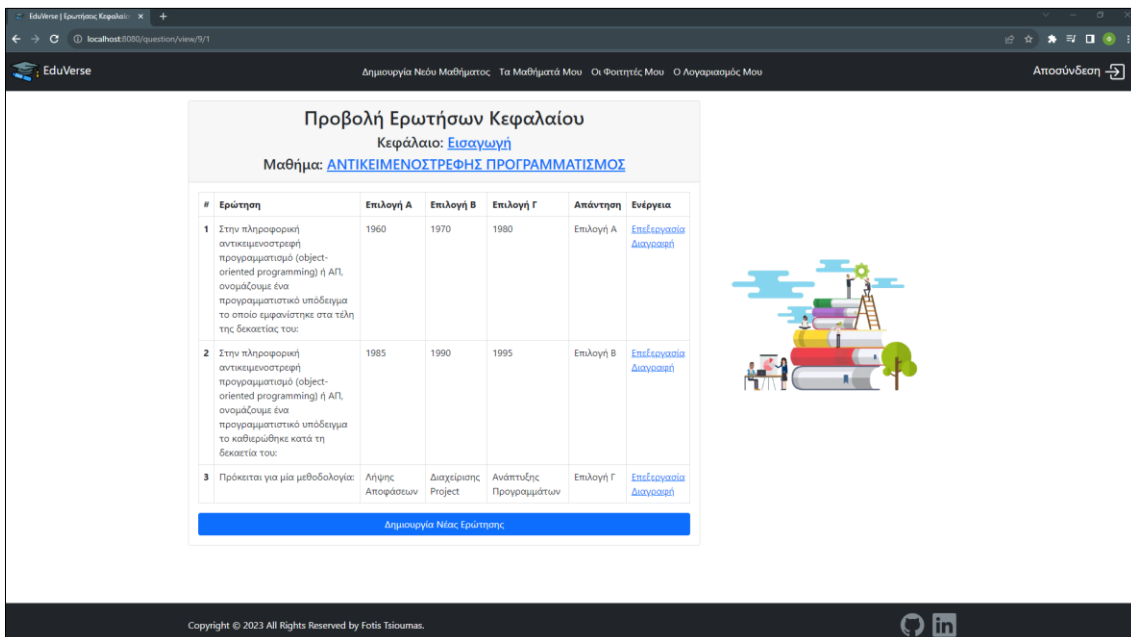
At the bottom of the form, there are two buttons: a blue 'Αποθήκευση' (Save) button and a red 'Ακύρωση' (Cancel) button. The footer of the page contains the text 'Copyright © 2023 All Rights Reserved by Fotis Tsoumas.' and a LinkedIn icon.

Figure 22: Επεξεργασία Κεφαλαίου

Προβολή Ερωτήσεων Κεφαλαίου

Στη σελίδα αυτή έχουν πρόσβαση μόνο οι χρήστες με ρόλο «καθηγητή». Το content της αποτελείται από έναν πίνακα, που στο header έχει το όνομα του κεφαλαίου (με υπερσύνδεσμο προς την προβολή κεφαλαίου) και το όνομα του μαθήματος (με υπερσύνδεσμο προς την προβολή του μαθήματος) και στο body του έχει ένα grid με τις ερωτήσεις του κεφαλαίου, ενώ μετά το grid υπάρχει ένα button που οδηγεί στη σελίδα δημιουργία νέας ερώτησης. Σε κάθε σειρά του grid υπάρχουν δύο υπερσύνδεσμοι, οι οποίοι εκτελούν από μια ενέργεια για την κάθε ερώτηση, ξεχωριστά. Η επεξεργασία οδηγεί στη σελίδα επεξεργασίας της ερώτησης και η διαγραφή πραγματοποιεί διαγραφή της ερώτησης, αφαιρώντας την ερώτηση από το grid και εμφανίζοντας μήνυμα επιτυχημένης διαγραφής στο πάνω μέρος της οθόνης. Για να μπορέσει ένας χρήστης να δει τη σελίδα πρέπει να καλύπτει τα παρακάτω κριτήρια:

- Να είναι συνδεδεμένος χρήστης. Αν δεν πληροί το κριτήριο οδηγείται στη σελίδα σύνδεσης.
- Να είναι ο συνδεδεμένος χρήστης με id αυτό που φαίνεται στο url της σελίδας. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι ο «καθηγητής» που δημιούργησε το μάθημα. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι ο «καθηγητής» που δημιούργησε το μάθημα στο οποίο ανήκει το κεφάλαιο των ερωτήσεων. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.



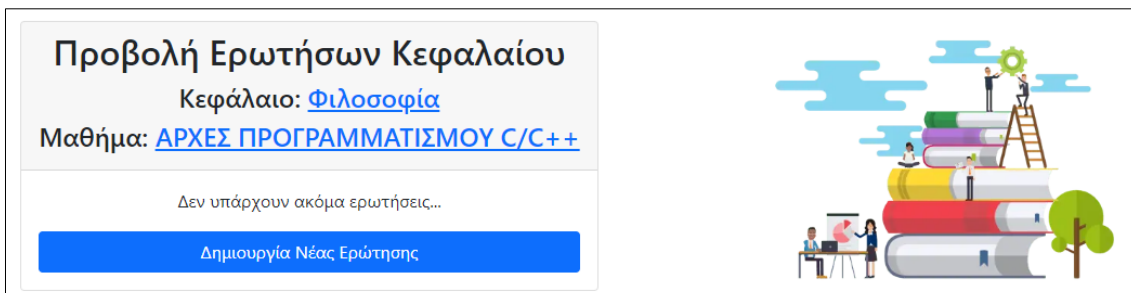
The screenshot shows a web browser displaying the 'EduVerse' interface. The page title is 'Προβολή Ερωτήσεων Κεφαλαίου' (Chapter Questions Display). The header includes the chapter name 'Κεφάλαιο: Εισαγωγή' and the subject 'Μαθήμα: ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ'. Below this is a table with three rows of questions. Each row contains a question number, the question text, and columns for 'Επιλογή Α', 'Επιλογή Β', 'Επιλογή Γ', 'Απάντηση', and 'Ενέργεια'. The 'Ενέργεια' column contains links for 'Επεξεργασία' and 'Διαγραφή'. At the bottom of the table is a blue button labeled 'Δημιουργία Νέας Ερώτησης'. To the right of the table is an illustration of a stack of books with a ladder and a person climbing, symbolizing learning.

#	Ερώτηση	Επιλογή Α	Επιλογή Β	Επιλογή Γ	Απάντηση	Ενέργεια
1	Στην πληροφορική αντικειμενοστρεφή προγραμματισμό (object-oriented programming) ή ΑΠ, ονομάζουμε ένα προγραμματιστικό υπόδειγμα το οποίο εμφανίστηκε στα τέλη της δεκαετίας του:	1960	1970	1980	Επιλογή Α	Επεξεργασία Διαγραφή
2	Στην πληροφορική αντικειμενοστρεφή προγραμματισμό (object-oriented programming) ή ΑΠ, ονομάζουμε ένα προγραμματιστικό υπόδειγμα το καθιερώθηκε κατά τη δεκαετία του:	1985	1990	1995	Επιλογή Β	Επεξεργασία Διαγραφή
3	Πρόκειται για μια μεθοδολογία:	Λήψης Αποφάσεων	Διαχείρισης Project	Ανάπτυξης Προγραμμάτων	Επιλογή Γ	Επεξεργασία Διαγραφή

Copyright © 2023 All Rights Reserved by Fotis Tsioumas.

Figure 23: Προβολή Ερωτήσεων Κεφαλαίου

Αν δεν υπάρχει κάποια ερώτηση για το συγκεκριμένο κεφάλαιο, στη θέση του grid εμφανίζεται μήνυμα ενημέρωσης.



The screenshot shows the same 'EduVerse' interface, but the table is replaced by a message: 'Δεν υπάρχουν ακόμα ερωτήσεις...'. Below the message is a blue button labeled 'Δημιουργία Νέας Ερώτησης'. To the right is the same illustration of a stack of books with a ladder and a person climbing.

Figure 24: Απουσία ερωτήσεων κεφαλαίου

Δημιουργία Νέας Ερώτησης

Στη σελίδα αυτή έχουν πρόσβαση μόνο οι χρήστες με ρόλο «καθηγητή». Το content της αποτελείται από μια φόρμα δημιουργίας ερώτησης πολλαπλής επιλογής, στην οποία ο «καθηγητής» έχει τη δυνατότητα να προσθέσει τρεις επιλογές και να ορίσει ποια από αυτές είναι η σωστή. Η συγκεκριμένη φόρμα αποτελείται από πέντε υποχρεωτικά πεδία (ερώτηση, επιλογή α, επιλογή β, επιλογή γ και σωστή απάντηση), τα οποία είναι τα χαρακτηριστικά της ερώτησης. Επίσης περιλαμβάνει δύο buttons, ένα για την υποβολή της φόρμας και ένα για την ακύρωση δημιουργίας μαθήματος, όπου οδηγεί στην σελίδα προβολή ερωτήσεων του κεφαλαίου. Κατά την υποβολή της φόρμας πατώντας το button «Δημιουργία», ενεργοποιείται το παρακάτω validation:

- Τα υποχρεωτικά πεδία ελέγχονται να μην είναι κενά.

Αν ο κανόνας παραβιαστεί, τότε εμφανίζεται μήνυμα κάτω από το αντίστοιχο πεδίο και δεν επιτρέπει την υποβολή της. Αν ο χρήστης συμπληρώσει σωστά όλα τα στοιχεία και υποβάλει την φόρμα, ανακατευθύνεται στην σελίδα προβολής ερωτήσεων του συγκεκριμένου κεφαλαίου, όπου ενημερώνεται με μήνυμα στο πάνω μέρος της σελίδας για την επιτυχή δημιουργία νέας ερώτησης. Επιπλέον, για να μπορέσει ένας χρήστης να δει τη σελίδα πρέπει να καλύπτει τα παρακάτω κριτήρια:

- Να είναι συνδεδεμένος χρήστης. Αν δεν πληροί το κριτήριο οδηγείται στη σελίδα σύνδεσης.
- Να είναι ο συνδεδεμένος χρήστης με id αυτό που φαίνεται στο url της σελίδας. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι ο «καθηγητής» που δημιούργησε το μάθημα. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι ο «καθηγητής» που δημιούργησε το μάθημα στο οποίο ανήκει το κεφάλαιο, για το οποίο θα δημιουργηθεί η ερώτηση. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.

The screenshot shows a web browser window displaying the 'Δημιουργία Νέας Ερώτησης' form. The form is titled 'Δημιουργία Νέας Ερώτησης' and contains the following fields and buttons:

- Ερώτηση:** A text input field.
- Επιλογή Α:** A text input field.
- Επιλογή Β:** A text input field.
- Επιλογή Γ:** A text input field.
- Σωστή Απάντηση:** A dropdown menu with 'Επιλογή Α' selected.
- Δημιουργία:** A blue button.
- Ακύρωση:** A red button.

On the left side of the form, there is an illustration of a person sitting on a blue beanbag chair, using a laptop. The browser's address bar shows 'localhost:5000/question/new/9/1'. The footer of the page contains the text 'Copyright © 2023 All Rights Reserved by Fotis Tsioumas.' and social media icons for GitHub and LinkedIn.

Figure 25: Δημιουργία Νέας Ερώτησης

Επεξεργασία Ερώτησης

Στη σελίδα αυτή έχουν πρόσβαση μόνο οι χρήστες με ρόλο «καθηγητή». Το content της είναι όμοιο με το content της σελίδας δημιουργίας νέας ερώτησης, με τη διαφορά ότι τα πεδία είναι συμπληρωμένα με τα στοιχεία της ερώτησης. Στη συγκεκριμένη σελίδα ο «καθηγητής» μπορεί να επεξεργαστεί τις πληροφορίες της ερώτησης και να πατήσει «Αποθήκευση». Μια σημαντική πληροφορία στο συγκεκριμένο σημείο είναι, ότι αν ο «καθηγητής» αλλάξει τα στοιχεία μια ερώτησης, δεν αλλάζει κάτι στην προβολή ερωτήσεων ερωτηματολογίου που έχει ολοκληρώσει ήδη ο «φοιτητής». Οι αλλαγές θα εμφανιστούν στην επόμενη προσπάθεια ολοκλήρωσης του ερωτηματολογίου του κεφαλαίου, που θα πραγματοποιήσει. Κατά την υποβολή της φόρμας πατώντας το button «Αποθήκευση», ενεργοποιείται το παρακάτω validation:

- Τα υποχρεωτικά πεδία ελέγχονται να μην είναι κενά.

Αν ο κανόνας παραβιαστεί, τότε εμφανίζεται μήνυμα κάτω από το αντίστοιχο πεδίο και δεν επιτρέπει την αποθήκευση των αλλαγών. Αν ο χρήστης συμπληρώσει σωστά όλα τα στοιχεία και υποβάλει την φόρμα, ανακατευθύνεται στην σελίδα του κεφαλαίου, όπου ενημερώνεται με μήνυμα στο πάνω μέρος της σελίδας για την επιτυχή επεξεργασία της ερώτησης. Επιπλέον, για να μπορέσει ένας χρήστης να δει τη σελίδα πρέπει να καλύπτει τα παρακάτω κριτήρια:

- Να είναι συνδεδεμένος χρήστης. Αν δεν πληροί το κριτήριο οδηγείται στη σελίδα σύνδεσης.
- Να είναι ο συνδεδεμένος χρήστης με id αυτό που φαίνεται στο url της σελίδας. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι χρήστης με ρόλο «καθηγητή». Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι ο «καθηγητής» που δημιούργησε το μάθημα στο οποίο ανήκει το κεφάλαιο της ερώτησης. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.

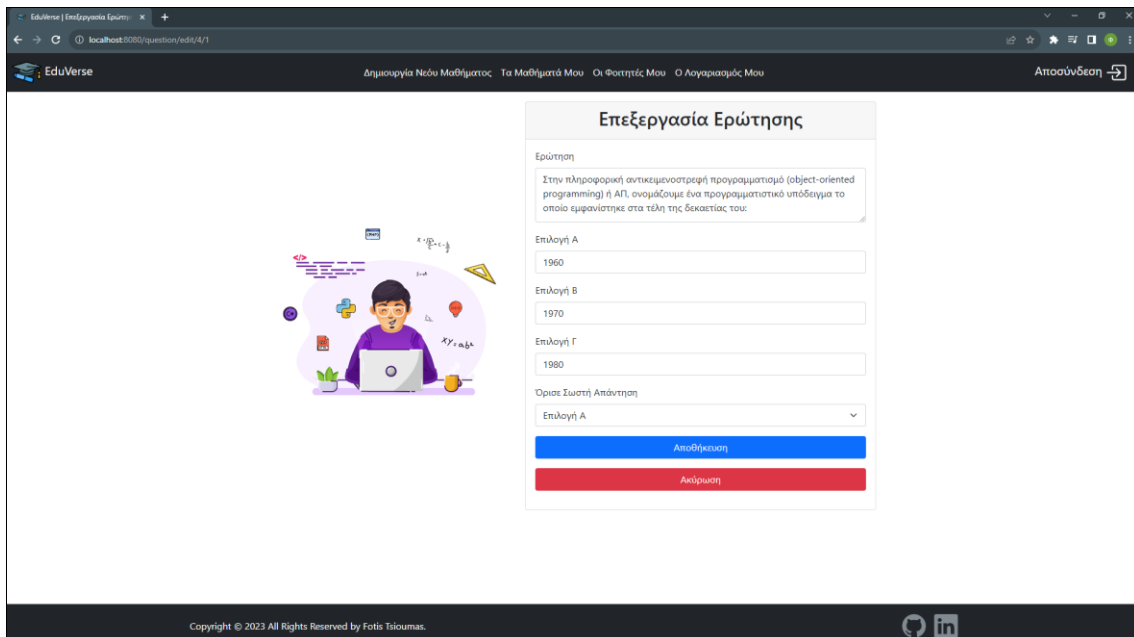


Figure 26: Επεξεργασία Ερώτησης

Οι Φοιτητές Μου

Στη σελίδα αυτή έχουν πρόσβαση μόνο οι χρήστες με ρόλο «καθηγητή». Το content της αποτελείται από ένα πίνακα που περιλαμβάνει ένα grid με τα μαθήματα του καθηγητή και τους εγγεγραμμένους «φοιτητές». Από τα στοιχεία των μαθημάτων περιλαμβάνει τον κωδικό και το όνομα ενώ η τρίτη στήλη σε κάθε γραμμή, αποτελείται από μια λίστα με τα ονόματα και τον αριθμό μητρώου των εγγεγραμμένων φοιτητών. Ο συνδυασμός ονόματος και αριθμού μητρώου κάθε «φοιτητή», αποτελεί υπερσύνδεσμο προς τα συνολικά αποτελέσματα του «φοιτητή» για το μάθημα της γραμμής στην οποία βρίσκεται. Για να μπορέσει ένας χρήστης να δει τη σελίδα πρέπει να καλύπτει τα παρακάτω κριτήρια:

- Να είναι συνδεδεμένος χρήστης. Αν δεν πληροί το κριτήριο οδηγείται στη σελίδα σύνδεσης.
- Να είναι ο συνδεδεμένος χρήστης με id αυτό που φαίνεται στο url της σελίδας. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι χρήστης με ρόλο «καθηγητή». Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.

#	Κωδικός	Τίτλος	Φοιτητές
1	TME105	ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ	Φώτιος Τσιούμας (edustud00002) Γεώργιος Μουσουδής (edustud00011) Δημήτριος Τσιούμας (edustud00006) Ζωστα Σιδέρη (edustud00003)
2	TME107	ΑΡΧΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C/C++	Ευαγγελία Τσιούμα (edustud00004) Δημήτριος Τσιούμας (edustud00006) Παναγιώτης Μαλέσκος (edustud00010)
3	TME154	ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΚΙΝΗΤΕΣ ΣΥΣΤΗΜΕΣ	Ευαγγελία Ανγέλη (edustud00008) Φώτιος Τσιούμας (edustud00002) Ζωστα Σιδέρη (edustud00003) Κωνσταντίνος Τσιούμας (edustud00009) Παναγιώτης Μαλέσκος (edustud00010)

Figure 27: Οι Φοιτητές Μου

Προβολή Αποτελεσμάτων Φοιτητή

Στη σελίδα αυτή έχουν πρόσβαση μόνο οι χρήστες με ρόλο «καθηγητή». Το content της αποτελείται από ένα πίνακα όπου στο header εμφανίζει το όνομα του μαθήματος (με υπερσύνδεσμο προς την προβολή μαθήματος) και το όνομα του μαθητή μαζί με τον αριθμό μητρώου. Το body του πίνακα περιλαμβάνει ένα grid με τα αποτελέσματα του «φοιτητή» στα ερωτηματολόγια που έχει κάνει για το συγκεκριμένο μάθημα. Ο πίνακα αυτός περιλαμβάνει σε κάθε γραμμή, ένα αποτέλεσμα ερωτηματολογίου και προβάλλει το όνομα του κεφαλαίου, το σκορ του «φοιτητή» στο συγκεκριμένο ερωτηματολόγιο, το σχόλιο προς τον «φοιτητή» για το συγκεκριμένο ερωτηματολόγιο και στην τελευταία στήλη υπάρχει ένας υπερσύνδεσμος, ο οποίος οδηγεί στις ερωτήσεις που πραγματοποίησε ο «φοιτητής» στο ερωτηματολόγιο. Στο τέλος του grid υπάρχει και ένα κουμπί που οδηγεί πίσω στους «φοιτητές» του «καθηγητή». Για να μπορέσει ένας χρήστης να δει τη σελίδα πρέπει να καλύπτει τα παρακάτω κριτήρια:

- Να είναι συνδεδεμένος χρήστης. Αν δεν πληροί το κριτήριο οδηγείται στη σελίδα σύνδεσης.
- Να είναι ο συνδεδεμένος χρήστης με id αυτό που φαίνεται στο url της σελίδας. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι χρήστης με ρόλο «καθηγητή». Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι ο «καθηγητής» που έχει δημιουργήσει το συγκεκριμένο μάθημα. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Ο «φοιτητής» με id αυτό που φαίνεται στο url της σελίδας να είναι εγγεγραμμένος στο συγκεκριμένο μάθημα. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.

#	Κεφάλαιο	Σκορ	Σχόλιο	Ενέργεια
1	Εισαγωγή	1/3	Θα πρέπει να διαβάσεις ενάντι το κεφάλαιο και να επαναλάβεις το ερωτηματολόγιο.	Προβολή
2	Εισαγωγή	2/3	Θα ήταν καλό να διαβάσεις ενάντι το κεφάλαιο και να επαναλάβεις το ερωτηματολόγιο.	Προβολή
3	Εισαγωγή	3/3	Μπορείς να προχωρήσεις στο επόμενο κεφάλαιο.	Προβολή
4	Ιστορικό	1/3	Θα πρέπει να διαβάσεις ενάντι το κεφάλαιο και να επαναλάβεις το ερωτηματολόγιο.	Προβολή
5	Ιστορικό	3/3	Μπορείς να προχωρήσεις στο επόμενο κεφάλαιο.	Προβολή

Επιστροφή στους Φοιτητές Μου

Figure 28: Προβολή Αποτελεσμάτων Φοιτητή

Αν ο «φοιτητής» δεν έχει ολοκληρώσει ακόμα κάποιο ερωτηματολόγιο για το συγκεκριμένο μάθημα, αντί για το grid εμφανίζεται μήνυμα ενημέρωσης.

Προβολή Αποτελεσμάτων Φοιτητή
Μαθήμα: [ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΚΙΝΗΤΕΣ ΣΥΣΤΗΜΑΤΑ](#)
Φοιτητής: Ευγενία Αγγέλη (edustud00008)

Δεν έχει ολοκληρώσει ακόμα κάποιο ερωτηματολόγιο...

Επιστροφή στους Φοιτητές Μου

Figure 29: Απουσία αποτελεσμάτων φοιτητή

Προβολή Ερωτηματολογίου

Στη σελίδα αυτή έχουν πρόσβαση χρήστες με ρόλο «καθηγητή» και «φοιτητή». Το content της αποτελείται από ένα πίνακα όπου στο header εμφανίζει το όνομα του κεφαλαίου (με υπερσύνδεσμο προς την προβολή κεφαλαίου), το όνομα του μαθήματος (με υπερσύνδεσμο προς την προβολή μαθήματος) και το όνομα του μαθητή μαζί με τον αριθμό μητρώου. Το body του πίνακα περιλαμβάνει ένα grid με τις ερωτήσεις του ερωτηματολογίου και τις απαντήσεις του «φοιτητή». Οπότε διευκρινίστηκε και νωρίτερα, οι ερωτήσεις από τις οποίες αποτελείται ένα ολοκληρωμένο ερωτηματολόγιο δεν μπορούν να αλλάξουν ή να διαγραφούν. Για να μπορέσει ένας χρήστης να δει τη σελίδα πρέπει να καλύπτει τα παρακάτω κριτήρια:

- Να είναι συνδεδεμένος χρήστης. Αν δεν πληροί το κριτήριο οδηγείται στη σελίδα σύνδεσης.
- Να είναι ο συνδεδεμένος χρήστης με id αυτό που φαίνεται στο url της σελίδας. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι ο «καθηγητής» που δημιούργησε το μάθημα που ανήκει το κεφάλαιο του συγκεκριμένου ερωτηματολογίου.
- Ο «φοιτητής» με id αυτό που φαίνεται στο url της σελίδας να είναι εγγεγραμμένος στο συγκεκριμένο μάθημα.
- Ο «φοιτητής» με id αυτό που φαίνεται στο url της σελίδας να είναι αυτός που έκανε το ερωτηματολόγιο.

Εκπαίδευση | Προβολή Ερωτηματολογίου

localhost:3000/quiz/question/13/2/1

EduVerse

Δημοσίευση Νέου Μαθήματος Τα Μαθήματά Μου Οι Φοιτητές Μου Ο Λογαριασμός Μου Αποσύνδεση

Προβολή Ερωτηματολογίου

Κεφάλαιο: [Εισαγωγή](#)

Μαθήμα: [ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ](#)

Φοιτητής: Φώτιος Τσιούμας (edustud00002)

#	Ερώτηση	Σωστή Απάντηση	Επιλεγμένη Απάντηση	Αποτέλεσμα
1	Στην πληροφορική αντικειμενοστρεφή προγραμματισμό (object-oriented programming) ή ΑΠ, ονομάζουμε ένα προγραμματιστικό υπόδειγμα το οποίο εμφανίστηκε στα τέλη της δεκαετίας του:	1960	1960	Σωστό
2	Στην πληροφορική αντικειμενοστρεφή προγραμματισμό (object-oriented programming) ή ΑΠ, ονομάζουμε ένα προγραμματιστικό υπόδειγμα το καθιερώθηκε κατά τη δεκαετία του:	1990	1985	Λάθος
3	Πρόκειται για μία μεθοδολογία:	Ανάπτυξης Προγραμμάτων	Λήψης Αποφάσεων	Λάθος

[Επιστροφή στα Αποτελέσματα Φοιτητή](#)

Copyright © 2023 All Rights Reserved by Fotis Tsioumas.

Figure 30: Προβολή Ερωτηματολογίου – «καθηγητής»

Ο Λογαριασμός Μου

Στη σελίδα αυτή έχουν πρόσβαση χρήστες με ρόλο «καθηγητή» και «φοιτητή». Το content της αποτελείται από έναν πίνακα στα αριστερά και μια εικόνα στα δεξιά. Ο πίνακας περιλαμβάνει σε ένα grid όλα τα στοιχεία του συνδεδεμένου χρήστη, ενώ του προσφέρεται και η δυνατότητα αλλαγής των στοιχείων ή του κωδικού πρόσβασης, πατώντας στο αντίστοιχο button. Για να μπορέσει ένας χρήστης να δει τη σελίδα πρέπει να καλύπτει τα παρακάτω κριτήρια:

- Να είναι συνδεδεμένος χρήστης. Αν δεν πληροί το κριτήριο οδηγείται στη σελίδα σύνδεσης.
- Να είναι ο συνδεδεμένος χρήστης με id αυτό που φαίνεται στο url της σελίδας. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.

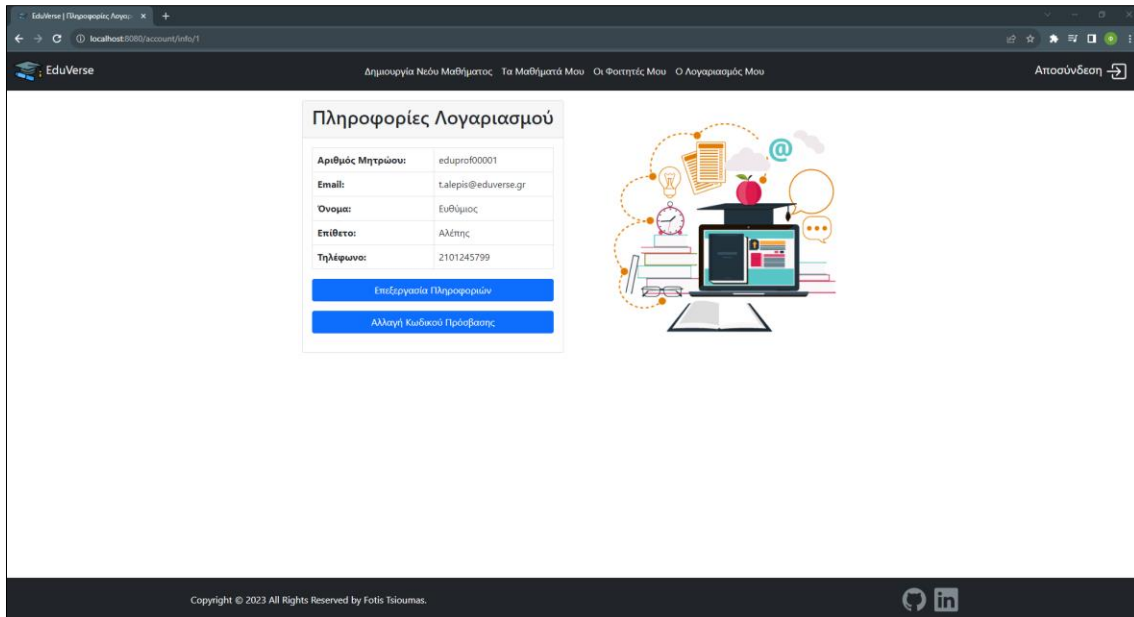


Figure 31: Ο Λογαριασμός Μου – «καθηγητής»

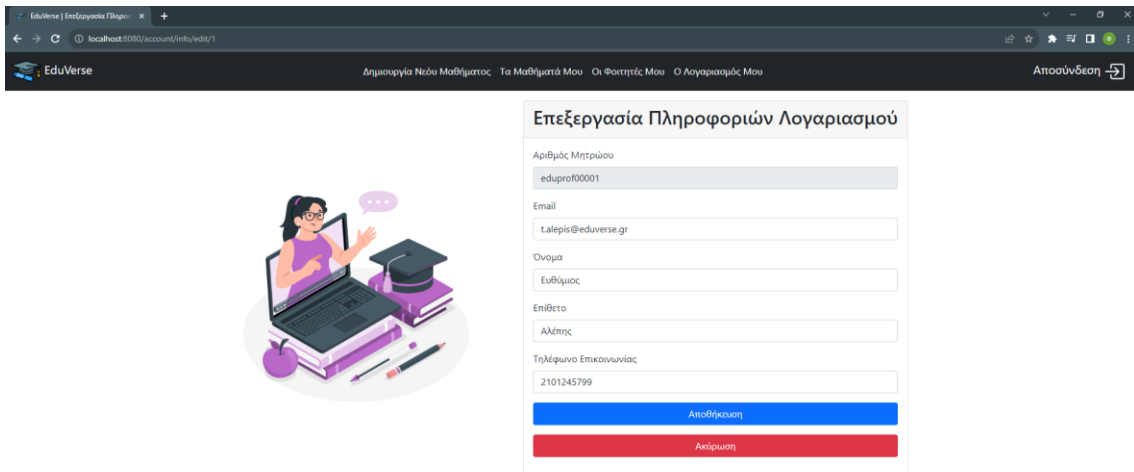
Επεξεργασία Πληροφοριών Λογαριασμού

Στη σελίδα αυτή έχουν πρόσβαση χρήστες με ρόλο «καθηγητή» και «φοιτητή». Το content της αποτελείται από μια φόρμα, στη οποία υπάρχουν συμπληρωμένα τα στοιχεία με τα οποία άνοιξε ο χρήστης τον λογαριασμό του. Σε αυτά τα πεδία προστίθεται ο «Αριθμός Μητρώου», ο οποίος παράγεται αυτόματα κατά τη δημιουργία του λογαριασμού. Το συγκεκριμένο στοιχείο είναι μοναδικό και δεν μπορεί να προστεθεί και σε άλλον χρήστη και δεν μπορεί να το αλλάξει ο χρήστης, όπως μπορεί να κάνει με τα υπόλοιπα στοιχεία του λογαριασμού του. Κατά την υποβολή της φόρμας πατώντας το button «Εγγραφή», ενεργοποιούνται τα παρακάτω validations:

- Τα υποχρεωτικά πεδία ελέγχονται να μην είναι κενά.
- Το πεδίο email ελέγχεται να μην υπάρχει ήδη.

Αν κάποιος κανόνας παραβιαστεί, τότε εμφανίζεται μήνυμα κάτω από το αντίστοιχο πεδίο και δεν επιτρέπει την αποθήκευση των αλλαγών. Αν ο χρήστης συμπληρώσει σωστά όλα τα στοιχεία και υποβάλει την φόρμα, ανακατευθύνεται στην σελίδα «Ο Λογαριασμός Μου», όπου ενημερώνεται με μήνυμα στο πάνω μέρος της σελίδας για την επιτυχή επεξεργασία των στοιχείων του. Επιπλέον, για να μπορέσει ένας χρήστης να δει τη σελίδα πρέπει να καλύπτει τα παρακάτω κριτήρια:

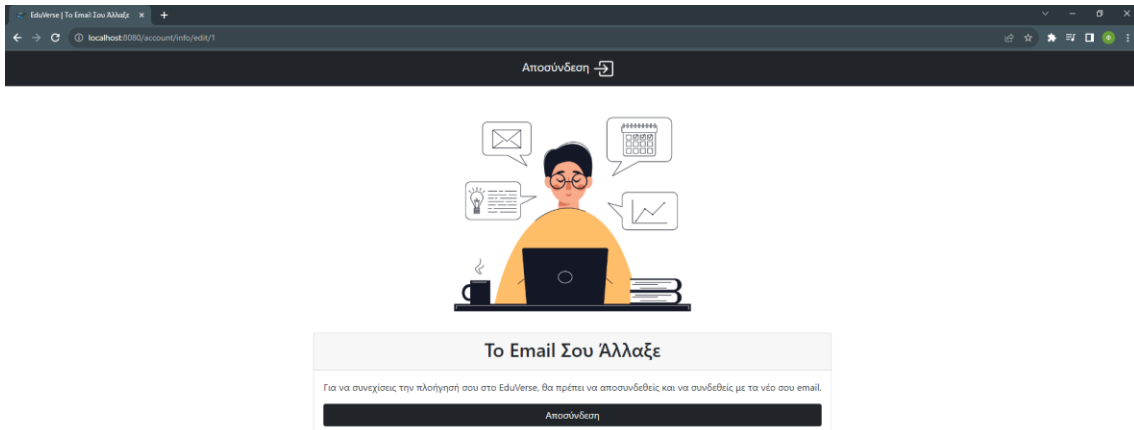
- Να είναι συνδεδεμένος χρήστης. Αν δεν πληροί το κριτήριο οδηγείται στη σελίδα σύνδεσης.
- Να είναι ο συνδεδεμένος χρήστης με id αυτό που φαίνεται στο url της σελίδας. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.



Copyright © 2023 All Rights Reserved by Fotis Tsioumas. 

Figure 32: Επεξεργασία Στοιχείων Λογαριασμού – «καθηγητής»

Σε περίπτωση που ο χρήστης αλλάξει το email των στοιχείων του, οδηγείται στην παρακάτω σελίδα όπου τον ενημερώνει ότι πρέπει να πραγματοποιήσει αποσύνδεση και σύνδεση με τα νέα στοιχεία.



Copyright © 2023 All Rights Reserved by Fotis Tsioumas. 

Figure 33: Αλλαγή Email

Αλλαγή Κωδικού Πρόσβασης

Στη σελίδα αυτή έχουν πρόσβαση χρήστες με ρόλο «καθηγητή» και «φοιτητή». Το content της αποτελείται από μια φόρμα με τρία πεδία, στην οποία ο χρήστης πρέπει να συμπληρώσει τον παλιό κωδικό του, τον νέο κωδικό και να επαληθεύσει τον νέο κωδικό. Κατά την υποβολή της φόρμας πατώντας το button «Εγγραφή», ενεργοποιούνται τα παρακάτω validations:

- Το περιεχόμενο του πεδίου «Παλιός Κωδικός» κατά το submit ελέγχεται να είναι ίδιο με το κωδικό που υπάρχει στη βάση.
- Το περιεχόμενο του πεδίου «Νέος Κωδικός» κατά το submit ελέγχεται να είναι ίδιο με το περιεχόμενο του πεδίου «Επιβεβαίωση Νέου Κωδικού».

Αν κάποιος κανόνας παραβιαστεί, τότε εμφανίζεται μήνυμα κάτω από το αντίστοιχο πεδίο και δεν επιτρέπει την αποθήκευση των αλλαγών. Αν ο χρήστης συμπληρώσει σωστά όλα τα στοιχεία και υποβάλει την φόρμα, ανακατευθύνεται στην σελίδα «Ο Λογαριασμός Μου», όπου ενημερώνεται με μήνυμα στο πάνω μέρος της σελίδας για την επιτυχή αλλαγή κωδικού. Επιπλέον, για να μπορέσει ένας χρήστης να δει τη σελίδα πρέπει να καλύπτει τα παρακάτω κριτήρια:

- Να είναι συνδεδεμένος χρήστης. Αν δεν πληροί το κριτήριο οδηγείται στη σελίδα σύνδεσης.
- Να είναι ο συνδεδεμένος χρήστης με id αυτό που φαίνεται στο url της σελίδας. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.

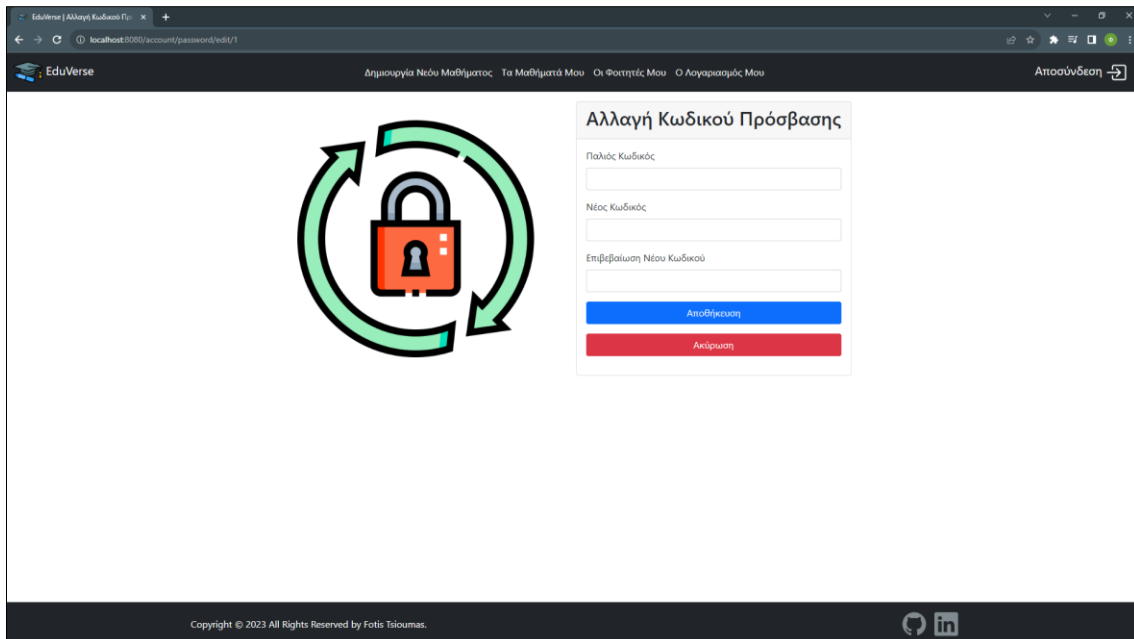


Figure 34: Αλλαγή Κωδικού Πρόσβασης – «καθηγητής»

Πλοήγηση ως «φοιτητής»

Στην κατηγορία «φοιτητής», ανήκουν όσοι χρήστες έχουν πραγματοποιήσει εγγραφή μέσω της «Φόρμας Εγγραφής Φοιτητή». Στην περίπτωση αυτή, ο χρήστης έχει λιγότερες επιλογές πλοήγησης μεταξύ των σελίδων της εφαρμογής από τον «καθηγητή», αλλά και πάλι είναι αυξημένες. Αυτές οι επιλογές, είναι οι εξής:

1. Αρχική Σελίδα
2. Διαθέσιμα Μαθήματα
3. Τα Μαθήματα Μου
4. Προβολή Μαθήματος
5. Προβολή Κεφαλαίου
6. Ερωτηματολόγιο
7. Αποτελέσματα Ερωτηματολογίου
8. Τα Αποτελέσματά Μου
9. Προβολή Ερωτηματολογίου
10. Ο Λογαριασμός Μου
11. Επεξεργασία Πληροφοριών Λογαριασμού
12. Αλλαγή Κωδικού Πρόσβασης

Όλες οι παραπάνω σελίδες αποτελούνται από τρία μέρη. Το navigation bar, το content και το footer.

Το navigation bar, περιλαμβάνει είτε μόνο τα δύο buttons στα άκρα, τα οποία αναφέρθηκαν στην περίπτωση του «καθηγητή», είτε περιλαμβάνει στα δύο άκρα τα παραπάνω buttons και στη μέση εμφανίζει τις κεντρικές σελίδες πλοήγησης του «φοιτητή».

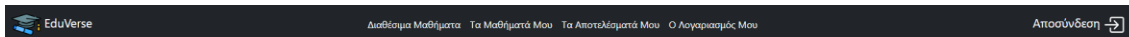


Figure 35: Navigation Bar 4

Το footer, όπως προαναφέρθηκε στην περίπτωση του «επισκέπτη», είναι κοινό για όλες τις σελίδες και στην περίπτωση του «φοιτητή». Οι περισσότερες διαφορές ανά σελίδα, εντοπίζονται στο content και αναλύονται παρακάτω.

Αρχική Σελίδα

Το content της αρχικής σελίδας του «φοιτητή», παρουσιάζει ακριβώς την ίδια δομή με αυτή του «καθηγητή», με μόνη διαφορά τα buttons στον πίνακα του πρώτου μέρους της σελίδας, τα οποία οδηγούν στις κεντρικές σελίδες του «φοιτητή».

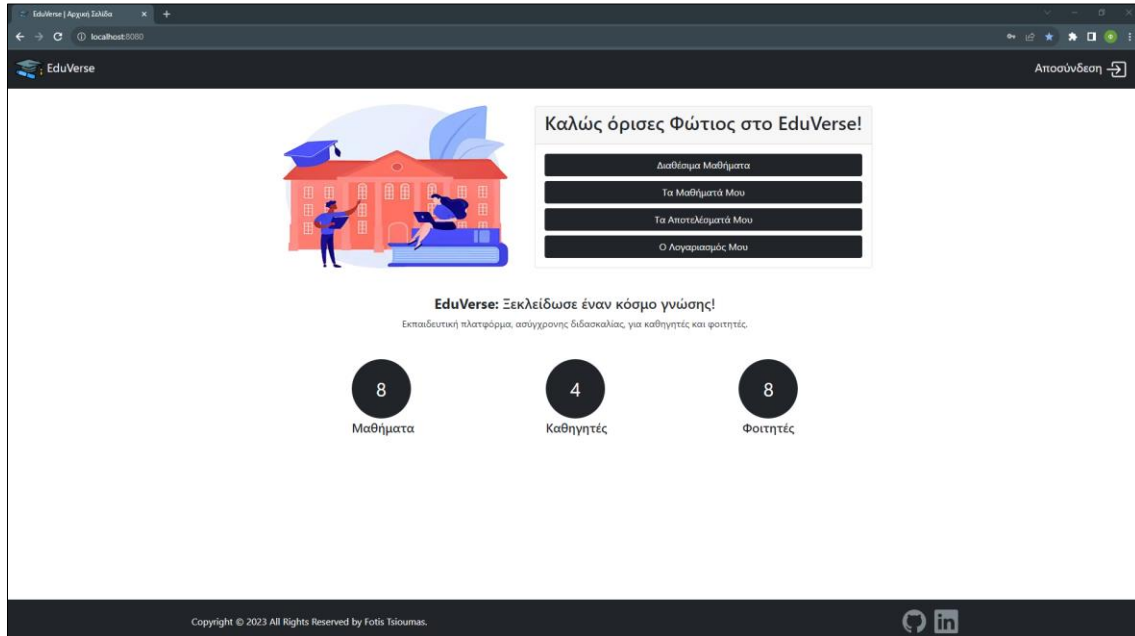


Figure 36: Αρχική Σελίδα – «φοιτητής»

Διαθέσιμα Μαθήματα

Στη συγκεκριμένη σελίδα έχουν πρόσβαση μόνο οι «φοιτητές». Το content της συγκεκριμένης αποτελείται ένα grid με τα μαθήματα που είναι διαθέσιμα στον «φοιτητή» και δεν έχει εγγραφεί ακόμη. Σε κάθε σειρά του grid υπάρχει ένα button «Εγγραφή», το οποίο όταν πατηθεί, αφαιρεί το μάθημα από το συγκεκριμένο grid, μεταφέρει τον «φοιτητή» στη σελίδα «Τα Μαθήματά Μου», τον ενημερώνει με μήνυμα στο πάνω μέρος της οθόνης για την επιτυχημένη εγγραφή και προσθέτη το μάθημα στο grid εκείνης της σελίδα. Για να μπορέσει ένας χρήστης να δει τη σελίδα πρέπει να καλύπτει τα παρακάτω κριτήρια:

- Να είναι συνδεδεμένος χρήστης. Αν δεν πληροί το κριτήριο οδηγείται στη σελίδα σύνδεσης.
- Να είναι ο συνδεδεμένος χρήστης με id αυτό που φαίνεται στο url της σελίδας. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι χρήστης με ρόλο «φοιτητή». Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.

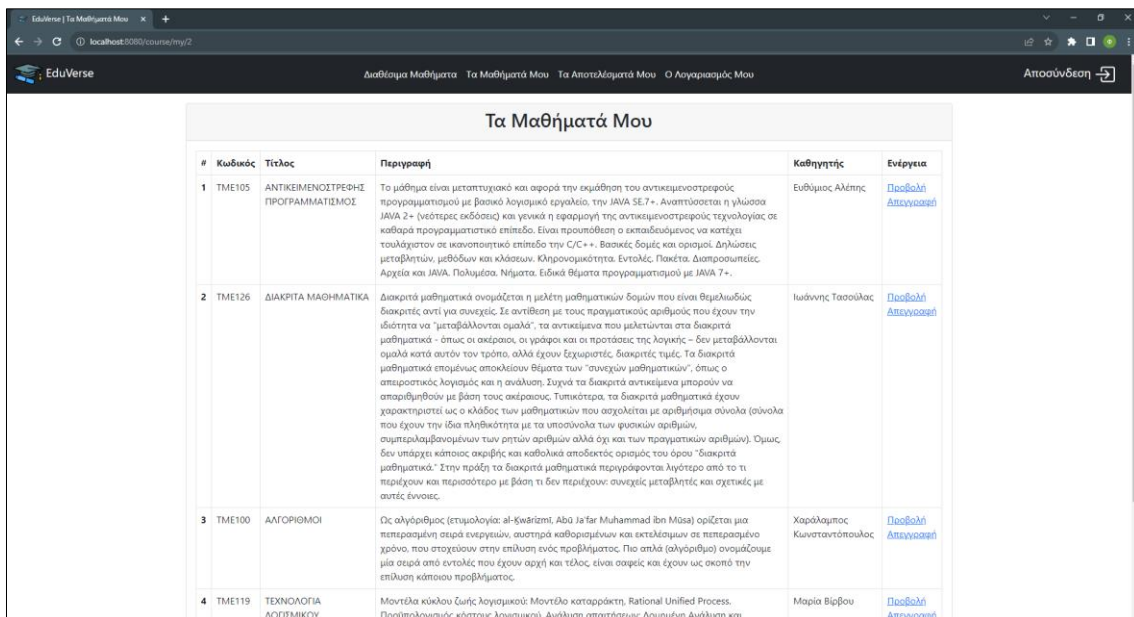
#	Τίτλος	Περιγραφή	Κωδικός	Καθηγητής	Ενέργεια
1	ΑΡΧΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C/C++	Το μάθημα Αρχές Προγραμματισμού απευθύνεται στους Σπουδαστές του μεταπτυχιακού προγράμματος Πληροφορική και αποτελείται από τρία βασικά στάδια. Στο πρώτο γίνεται εισαγωγή στην τεχνολογία Λογισμικού και στους Χώρους δεδομένων. Στο δεύτερο διδάσκονται η επίλυση προβλημάτων με Η/Υ. Στο τρίτο στάδιο γίνεται και εισαγωγή στον αντικειμενοστρεφή προγραμματισμό. Το μάθημα αναφέρεται οι μεταπτυχιακό επίπεδο στον σύγχρονο προγραμματισμό, στην τεχνολογία επίλυσης προβλημάτων με Η/Υ και στην σχεδίαση και ανάπτυξη προγραμμάτων με C. Προσπατούσιμη θεωρείται να έχει ο διδασκόμενος γνώσεις γενικά στους Η/Υ. Το μάθημα διδάσκεται στο μεταπτυχιακό ΠΛΗΡΟΦΟΡΙΚΗ.	TME107	Ευθύμιος Αλέπης	Εγγραφή
2	ΘΕΩΡΙΑ ΚΑΙ ΕΦΑΡΜΟΓΕΣ ΓΡΑΦΗΜΑΤΩΝ	Τα γραφήματα (σημια ενωμένα με δεσμούς ή τασεί σχηματισμοί (κίνων πιο εύλητα) διάφορα προβλήματα. Για την επίλυση των προβλημάτων αναπτύχθηκε η σχετική θεωρία. (Τα γραφήματα δεν έχουν σχέση με τις γραφικές παραστάσεις συναρτήσεων ή με τα γραφήματα της Στατιστικής.) Χρησιμότητα Τα γραφήματα έχουν μεταξύ των άλλων τις παρακάτω εφαρμογές: * Σε οποιοδήποτε είδος Δίκτυα (Άρδευση, Σοκοκωνικές Ύδραυλικά, Δίκτυα Πληροφορικής κλπ). * Οικονομία (Ελαχιστοποίηση κόστους, Χρονικός Προγραμματισμός κλπ) * Οργάνωση επιχειρημάτων (Οργανογράμματα, Κεντρικά σημεία) * Πληροφορική (Δίκτυα, Λογικά δεδομένα), Προγραμματισμός Αλγόριθμος Διάτρηξη δένδρων, Αλγόριθμοι γραφημάτων κλπ)	TME116	Ιωάννης Τσιούμας	Εγγραφή
3	ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΚΑΙ ΜΕΤΑΓΛΩΤΤΙΣΤΕΣ	Εισαγωγή στις Γλώσσες Προγραμματισμού. Ταξινομήσεις γλωσσών προγραμματισμού (διδασκαλίες-αντικειμενοστρεφείς επίπεδα γλωσσών «Λπ.»), Παραδείγματα σύνταξης γλώσσας προγραμματισμού Pascal. Τρόποι προσδιορισμού σύνταξης γλωσσών προγραμματισμού (BNF, EBNF, συντακτικά διαγράμματα). Στοιχεία Θεωρίας Γλωσσών, κανόνες παραγωγής, συντακτικά δένδρα, γραμματικές, η ταξινόμηση Chomsky, κανονικές εκφράσεις, παραφορμικά σύνταξης, αυτόματα στάθμες, μέσα για τον ορισμό σημασιολογικών προδιαγραφών. Εισαγωγή στη μεταγλωττιστή, ανάλυση-σύνθεση, T-διαγράμματα, διεργασίες, συμβολομεταφραστές, λεκτική ανάλυση, ομάδες, λεκτικές μονάδες, πρότυπα, ανόνηση από λάθη, LEX προγράμματα. Συντακτική και Σημασιολογική Ανάλυση, ανάλυση από πάνω προς τα κάτω, γραμματικές LL(k), LR(k), συντακτική ανάλυση αναδρομικής κατάβασης, υλοποίηση συντακτικού ανάλυση με χρήση ΥΑΚΣ, Παραγωγή Κώδικα, ενδιάμεσες αναπαραστάσεις.	TME102	Μαρία Βιβίου	Εγγραφή

Figure 37: Διαθέσιμα Μαθήματα

Τα Μαθήματα Μου

Η συγκεκριμένη σελίδα παρουσιάζει ακριβώς το ίδιο περιεχόμενο με αυτή του «καθηγητή», έχοντας μόνο τις παρακάτω διαφορές:

- Τα μαθήματα του grid είναι τα μαθήματα που έχει εγγραφεί ο «φοιτητής» και όχι αυτά που έχει δημιουργήσει ο καθηγητής.
- Στο grid υπάρχει μια επιπλέον στήλη, τη στήλη «Καθηγητής» και αναφέρει το όνομα του «καθηγητή» που έχει δημιουργήσει το μάθημα.
- Στη στήλη «Ενέργεια», οι δυνατές ενέργειες που έχει ο «φοιτητής» είναι η «Προβολή», η οποία είναι ίδια με αυτή του «καθηγητή» και η «Απεγγραφή», η οποία είναι ενέργεια μόνο του «φοιτητή» και κάνει την αντίστροφη ενέργεια από αυτή που κάνει η «Εγγραφή» της προηγούμενης σελίδας, με τη διαφορά ότι δεν πραγματοποιεί ανακατεύθυνση στην άλλη σελίδα.



#	Κωδικός	Τίτλος	Περιγραφή	Καθηγητής	Ενέργεια
1	TME105	ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ	Το μάθημα είναι μεταπτυχιακό και αφορά την εκμάθηση του αντικειμενοστρεφούς προγραμματισμού με βασικό λογισμικό εργαλείο, την JAVA SE 7+. Αναπτύσσεται η γλώσσα JAVA 2+ (νέότερες εκδόσεις) και γενικά η εφαρμογή της αντικειμενοστρεφούς τεχνολογίας σε καθαρά προγραμματιστικό επίπεδο. Είναι προϋπόθεση ο εκπαιδευόμενος να κατέχει τουλάχιστον σε ικανοποιητικό επίπεδο την C/C++. Βασικές δομές και ορισμοί. Δηλώσεις μεταβλητών, μεθόδων και κλάσεων. Κληρονομικότητα. Εντολές. Πακέτα. Διαπροσωπικές Αρχές και JAVA. Πολύμεσα. Νήματα. Ειδικά θέματα προγραμματισμού με JAVA 7+.	Ευθύμιος Αλέπης	Προβολή Απεγγραφή
2	TME126	ΔΙΑΚΡΙΤΑ ΜΑΘΗΜΑΤΙΚΑ	Διακριτά μαθηματικά ονομάζεται η μελέτη μαθηματικών δομών που είναι θεμελιώδεις διακριτές αντί για συνεχείς. Σε αντίθεση με τους πραγματικούς αριθμούς που έχουν την ιδιότητα να "μεταβάλλονται ομαλά", τα αντικείμενα που μελετώνται στα διακριτά μαθηματικά - όπως οι ακέραιοι, οι γράφοι και οι προτάσεις της λογικής - δεν μεταβάλλονται ομαλά κατά αυτόν τον τρόπο, αλλά έχουν ξεχωριστές, διακριτές τιμές. Τα διακριτά μαθηματικά επομένως αποδίδουν θέματα των "συνεχών μαθηματικών", όπως ο απαριθμητικός λογισμός και η ανάλυση. Συχνά τα διακριτά αντικείμενα μπορούν να αποριθμηθούν με βάση τους ακέραιους. Τυπικότερα, τα διακριτά μαθηματικά έχουν χαρακτηριστικά ως ο κλάδος των μαθηματικών που ασχολείται με αριθμητικά σύνολα (σύνολα που έχουν την ίδια πληθικότητα με το υποσύνολο των φυσικών αριθμών, συμπεριλαμβανομένων των ρητών αριθμών αλλά όχι και των πραγματικών αριθμών). Ωμως, δεν υπάρχει κάποιος ακριβής και καθολικά αποδεκτός ορισμός του όρου "διακριτά μαθηματικά." Στην πράξη τα διακριτά μαθηματικά περιγράφονται λιγότερο από το τι περιέχουν και περισσότερο με βάση τι δεν περιέχουν: συνεχείς μεταβλητές και σχετικές με αυτές έννοιες.	Ιωάννης Τασούλας	Προβολή Απεγγραφή
3	TME100	ΑΛΓΟΡΙΘΜΟΙ	Ος αλγόριθμος (ετυμολογία: al-ḡwāriẓmī, Abū Ja'far Muḥammad ibn Mūsā) ορίζεται μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελεστέων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος. Πιο απλά (αλγόριθμο) ονομάζουμε μία σειρά από εντολές που έχουν αρχή και τέλος, είναι σειρές και έχουν ως σκοπό την επίλυση κάποιου προβλήματος.	Χαράλαμπος Κωνσταντόπουλος	Προβολή Απεγγραφή
4	TME119	ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ	Μοντέλα κύκλου ζωής λογισμικού: Μοντέλο καταρράκτη, Rational Unified Process. Προϊστολογισμός κόστους λογισμικού. Ανάλυση απαιτήσεων: Δομημένη Ανάλυση και	Μαρία Βιρβού	Προβολή Απεγγραφή

Figure 38: Τα Μαθήματα Μου – «φοιτητής»

Αν ο «φοιτητής» είναι νέος χρήστης και δεν έχει ακόμα εγγραφεί σε κάποιο μάθημα, τότε αντί για το grid με τα μαθήματα, εμφανίζεται στη θέση του ένα μήνυμα που ενημερώνει ότι δεν υπάρχει κάποιο μάθημα.

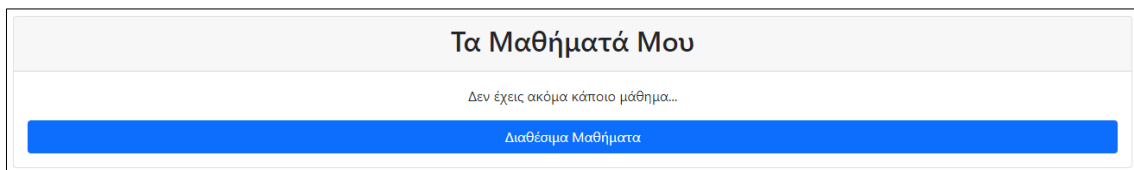


Figure 39: Απουσία μαθημάτων – «φοιτητής»

Προβολή Μαθήματος

Η συγκεκριμένη σελίδα παρουσιάζει ακριβώς το ίδιο περιεχόμενο με αυτή του «καθηγητή», έχοντας μόνο την παρακάτω διαφορά:

- Στο grid των κεφαλαίων του μαθήματος, έχει διαθέσιμες μόνο τις ενέργειες «Προβολή» όπου είναι ίδια με αυτή του «καθηγητή» και «Ερωτηματολόγιο», η οποία είναι ενέργεια που έχει μόνο ο «φοιτητής» και πραγματοποιεί ανακατεύθυνση στη σελίδα «Ερωτηματολόγιο».

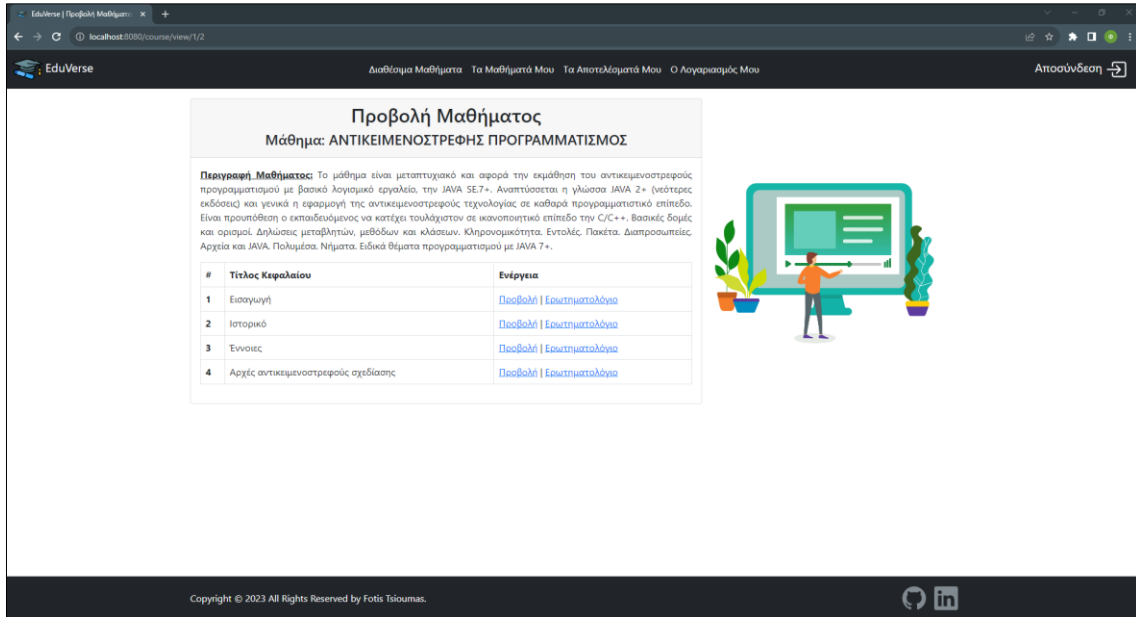


Figure 40: Προβολή Μαθήματος – «φοιτητής»

Αν δεν έχουν δημιουργηθεί κεφάλαια για το μάθημα, τότε αντί για το grid με τα κεφάλαια του μαθήματος, εμφανίζεται στη θέση του ένα μήνυμα που ενημερώνει ότι δεν υπάρχουν ακόμα κεφάλαια.

Προβολή Κεφαλαίου

Η συγκεκριμένη σελίδα παρουσιάζει ακριβώς το ίδιο περιεχόμενο με αυτή του «καθηγητή» χωρίς καμία διαφορά.

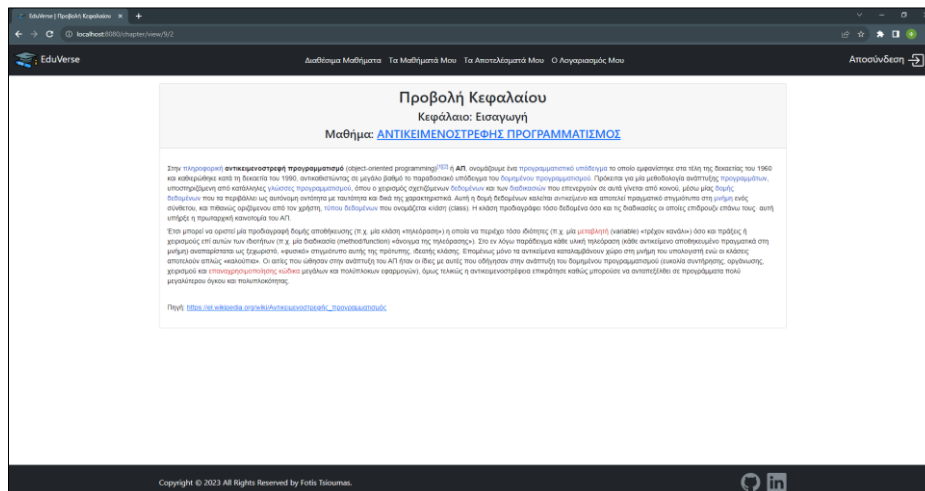


Figure 41: Προβολή Κεφαλαίου – «φοιτητής»

Ερωτηματολόγιο

Στη σελίδα αυτή έχουν πρόσβαση μόνο οι χρήστες με ρόλο «φοιτητή». Το content της αποτελείται από έναν πίνακα, όπου στο header περιλαμβάνει το όνομα του κεφαλαίου (με υπερσύνδεσμο προς το κεφάλαιο) και το όνομα του μαθήματος (με υπερσύνδεσμο προς το μάθημα) και στο body περιλαμβάνει μια φόρμα πραγματοποίησης ερωτηματολογίου και δύο buttons. Η συγκεκριμένη φόρμα αποτελείται από όλες τις ερωτήσεις του αντίστοιχου κεφαλαίου και σε κάθε μια ο «φοιτητής» πρέπει να επιλέξει από μία απάντηση. Τα δύο buttons, όπου το ένα είναι η «Υποβολή» και μεταφέρει τον χρήστη στην σελίδα «Αποτέλεσμα Ερωτηματολογίου» και το άλλο είναι η «Ακύρωση», όπου τον μεταφέρει στην προηγούμενη σελίδα, χωρίς να πραγματοποιήσει υποβολή του ερωτηματολογίου. Για να μπορέσει ένας χρήστης να δει τη σελίδα πρέπει να καλύπτει τα παρακάτω κριτήρια:

- Να είναι συνδεδεμένος χρήστης. Αν δεν πληροί το κριτήριο οδηγείται στη σελίδα σύνδεσης.
- Να είναι ο συνδεδεμένος χρήστης με id αυτό που φαίνεται στο url της σελίδας. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι χρήστης με ρόλο «φοιτητή». Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι «φοιτητής» εγγεγραμμένος στο μάθημα που ανήκει το κεφάλαιο. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.

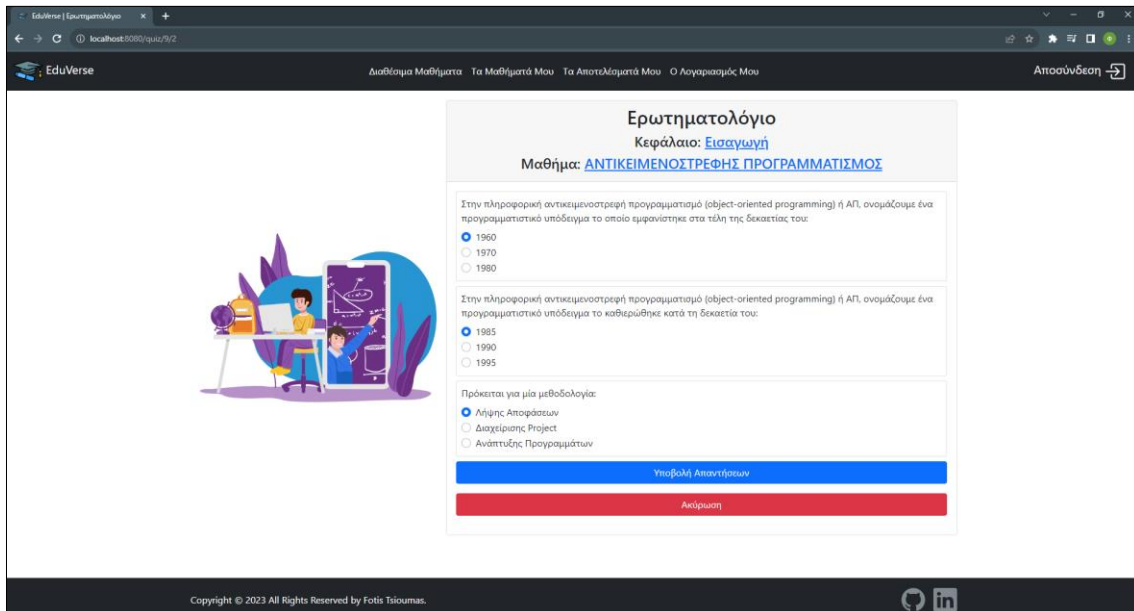


Figure 42: Ερωτηματολόγιο

Αν δεν υπάρχουν ακόμα ερωτήσεις για το συγκεκριμένο κεφάλαιο, αντί για τη φόρμα υποβολής του ερωτηματολογίου, εμφανίζεται στη θέση του ένα μήνυμα ότι δεν έχει δημιουργηθεί ακόμα το ερωτηματολόγιο.

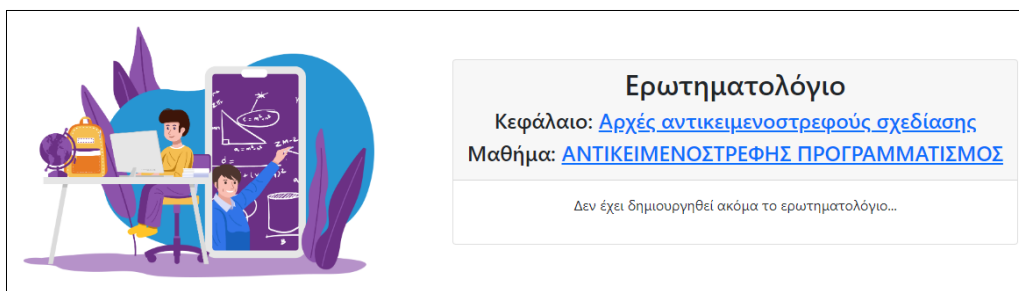


Figure 43: Απουσία ερωτηματολογίου

Αποτέλεσμα Ερωτηματολογίου

Στη σελίδα αυτή έχουν πρόσβαση μόνο οι χρήστες με ρόλο «φοιτητή». Το content της αποτελείται από έναν πίνακα, όπου στο header περιλαμβάνει το όνομα του κεφαλαίου (με υπερσύνδεσμο προς το κεφάλαιο) και το όνομα του μαθήματος (με υπερσύνδεσμο προς το μάθημα) και στο body περιλαμβάνει ένα grid με τον αριθμό των σωστών και λάθος απαντήσεων του «φοιτητή», ένα σχόλιο σχετικό με το αποτέλεσμα του ερωτηματολογίου και ένα κουμπί «Συνολικά Αποτελέσματα» όπου τον οδηγεί στην αντίστοιχη σελίδα. Για να μπορέσει ένας χρήστης να δει τη σελίδα πρέπει να καλύπτει τα παρακάτω κριτήρια:

- Να είναι συνδεδεμένος χρήστης. Αν δεν πληροί το κριτήριο οδηγείται στη σελίδα σύνδεσης.
- Να είναι ο συνδεδεμένος χρήστης με id αυτό που φαίνεται στο url της σελίδας. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι χρήστης με ρόλο «φοιτητή». Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι φοιτητής που πραγματοποίησε το συγκεκριμένο κεφάλαιο. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.

Αποτέλεσμα Ερωτηματολογίου
Κεφάλαιο: [Εισαγωγή](#)
Μαθήμα: [ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ](#)

Σωστές απαντήσεις:	2
Λάθος απαντήσεις:	1

Σχόλιο: Θα ήταν καλό να δοθείτε έναντι το κεφάλαιο και να επαναλάβετε το ερωτηματολόγιο

[Συνολικά Αποτελέσματα](#)

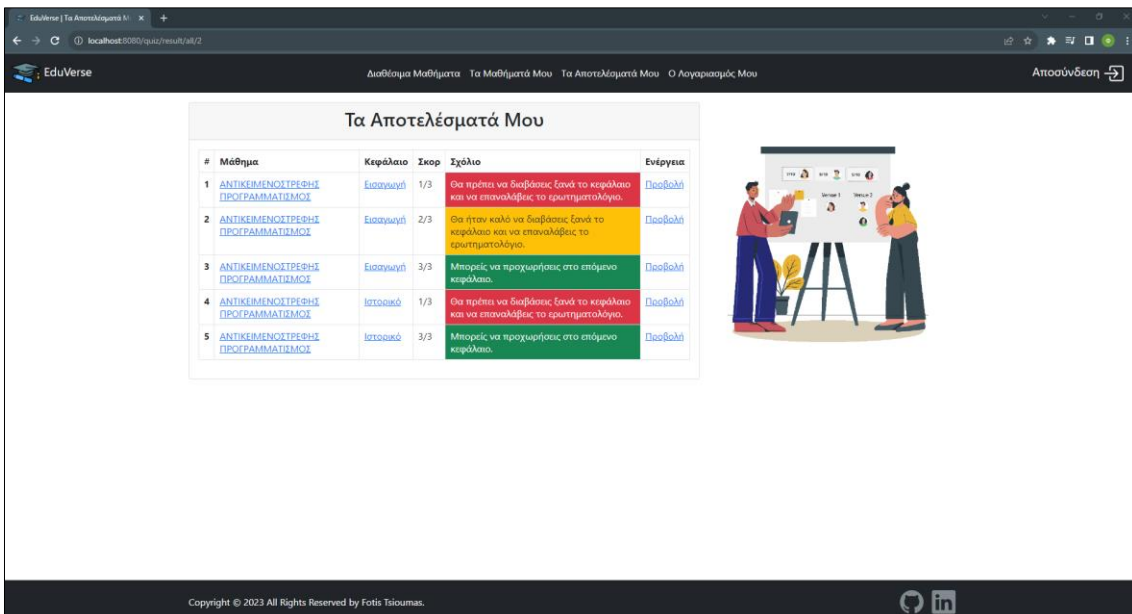
Copyright © 2023 All Rights Reserved by Fotis Tsioumas

Figure 44: Αποτέλεσμα Ερωτηματολογίου

Τα Αποτελέσματά Μου

Στη σελίδα αυτή έχουν πρόσβαση μόνο οι χρήστες με ρόλο «φοιτητή». Το content της αποτελείται από ένα πίνακα όπου περιλαμβάνει ένα grid με τα συνολικά αποτελέσματα του «φοιτητή» στα ερωτηματολόγια που έχει πραγματοποιήσει στα κεφάλαια όλων των μαθημάτων. Ο πίνακας αυτός περιλαμβάνει σε κάθε γραμμή, ένα αποτέλεσμα ερωτηματολογίου και προβάλλει το όνομα του μαθήματος (με υπερσύνδεσμο προς το μάθημα), το όνομα του κεφαλαίου (με υπερσύνδεσμο προς το κεφάλαιο), το σκορ του «φοιτητή» στο συγκεκριμένο ερωτηματολόγιο, το σχόλιο προς τον «φοιτητή» για το συγκεκριμένο ερωτηματολόγιο και στην τελευταία στήλη υπάρχει ένας υπερσύνδεσμος, ο οποίος οδηγεί στις ερωτήσεις που πραγματοποίησε ο «φοιτητής» στο ερωτηματολόγιο. Στο τέλος του grid υπάρχει και ένα κουμπί που οδηγεί πίσω στους «φοιτητές» του «καθηγητή». Για να μπορέσει ένας χρήστης να δει τη σελίδα πρέπει να καλύπτει τα παρακάτω κριτήρια:

- Να είναι συνδεδεμένος χρήστης. Αν δεν πληροί το κριτήριο οδηγείται στη σελίδα σύνδεσης.
- Να είναι ο συνδεδεμένος χρήστης με id αυτό που φαίνεται στο url της σελίδας. Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.
- Να είναι χρήστης με ρόλο «φοιτητής». Αν δεν πληροί το κριτήριο οδηγείται στην 403 σελίδα σφάλματος.



#	Μάθημα	Κεφάλαιο	Σκορ	Σχόλιο	Ενέργεια
1	ΑΝΤΙΚΕΙΜΕΝΟΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ	Εισαγωγή	1/3	Θα πρέπει να διαβάσεις ξανά το κεφάλαιο και να επαναλάβεις το ερωτηματολόγιο.	Παροβλή
2	ΑΝΤΙΚΕΙΜΕΝΟΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ	Εισαγωγή	2/3	Θα ήταν καλό να διαβάσεις ξανά το κεφάλαιο και να επαναλάβεις το ερωτηματολόγιο.	Παροβλή
3	ΑΝΤΙΚΕΙΜΕΝΟΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ	Εισαγωγή	3/3	Μπορείς να προχωρήσεις στο επόμενο κεφάλαιο.	Παροβλή
4	ΑΝΤΙΚΕΙΜΕΝΟΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ	Ιστορικό	1/3	Θα πρέπει να διαβάσεις ξανά το κεφάλαιο και να επαναλάβεις το ερωτηματολόγιο.	Παροβλή
5	ΑΝΤΙΚΕΙΜΕΝΟΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ	Ιστορικό	3/3	Μπορείς να προχωρήσεις στο επόμενο κεφάλαιο.	Παροβλή

Figure 45: Τα Αποτελέσματά Μου

Αν δεν έχει υποβληθεί κάποιο ερωτηματολόγιο, στη θέση του grid, εμφανίζεται ένα μήνυμα όπου ενημερώνει ότι δεν έχει ολοκληρώσει ο χρήστης κάποιο ερωτηματολόγιο και εμφανίζει επίσης το κουμπί τα μαθήματά μου, ώστε να οδηγήσει τον «φοιτητή» στα μαθήματα και να συνεχίσει την μελέτη του.

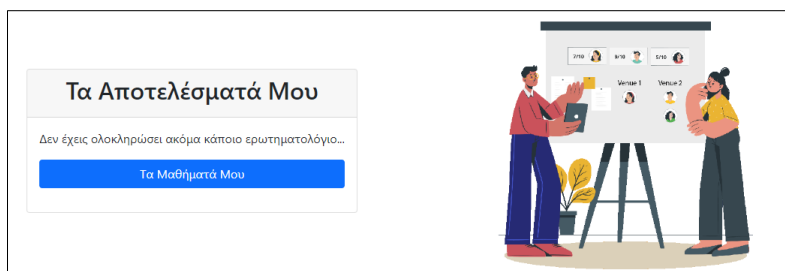
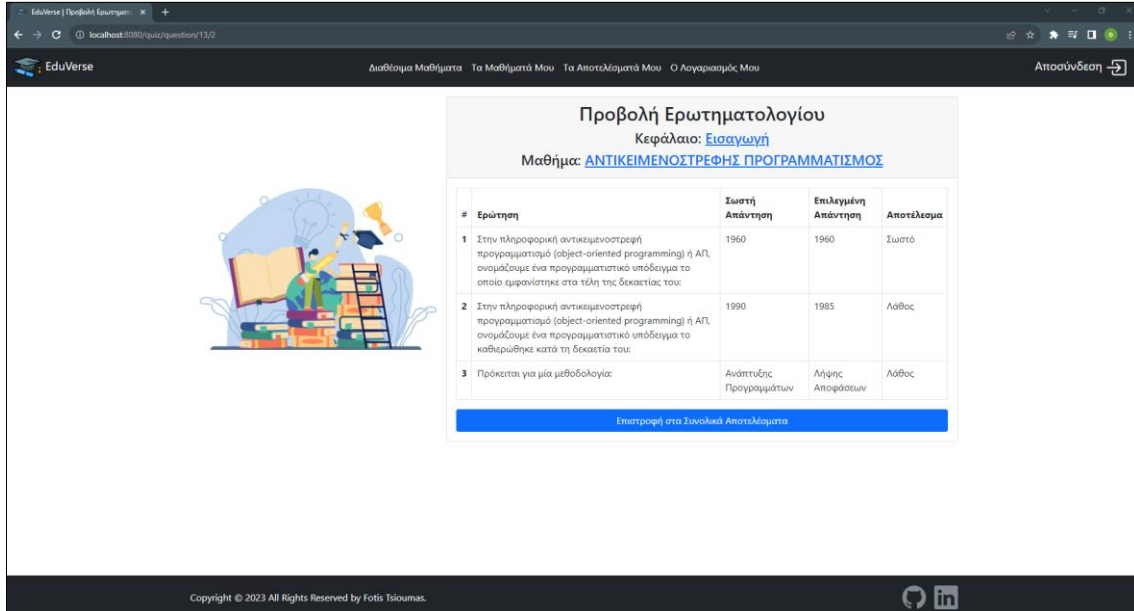


Figure 46: Απουσία αποτελεσμάτων

Προβολή Ερωτηματολογίου

Η συγκεκριμένη σελίδα παρουσιάζει ακριβώς το ίδιο περιεχόμενο με αυτή του «καθηγητή» χωρίς καμία διαφορά.



Επίδειξη Ερωτηματολογίου
Κεφάλαιο: [Εισαγωγή](#)
Μαθήμα: [ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ](#)

#	Ερώτηση	Σωστή Απάντηση	Επιλεγμένη Απάντηση	Αποτέλεσμα
1	Στην πληροφορική αντικειμενοστρεφή προγραμματισμό (object-oriented programming) ή ΑΠ, ονομάζουμε ένα προγραμματιστικό υπόδειγμα το οποίο εμφανίστηκε στα τέλη της δεκαετίας του	1960	1960	Σωστό
2	Στην πληροφορική αντικειμενοστρεφή προγραμματισμό (object-oriented programming) ή ΑΠ, ονομάζουμε ένα προγραμματιστικό υπόδειγμα το καθιερώθηκε κατά τη δεκαετία του	1990	1985	Λάθος
3	Πρόκειται για μία μεθοδολογία:	Ανάπτυξης Προγραμμάτων	Λήψης Αποφάσεων	Λάθος

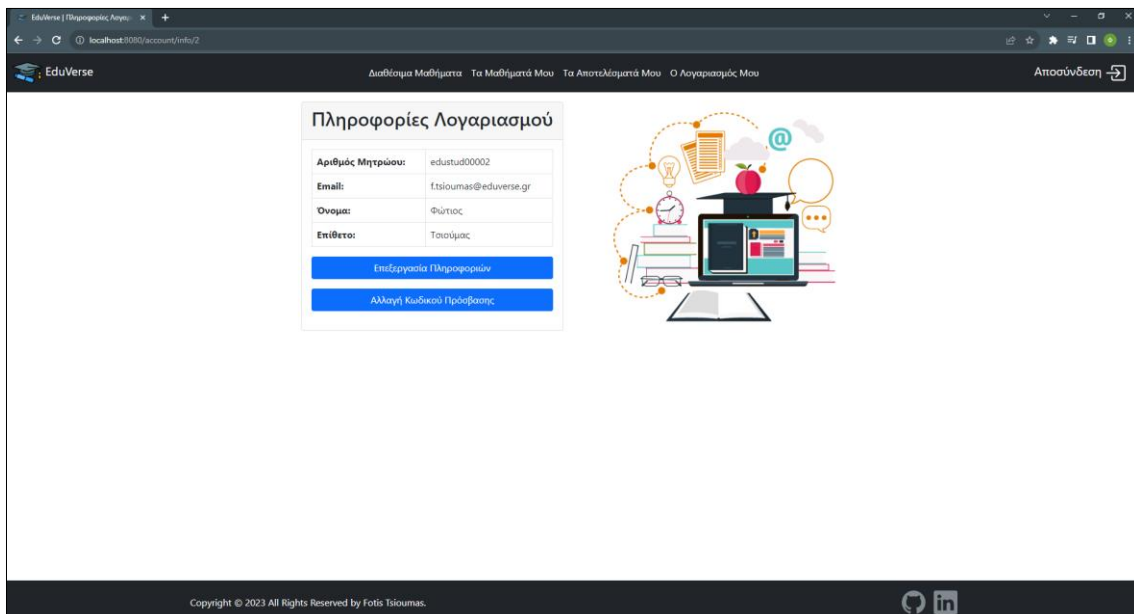
[Επιστροφή στα Συνολικά Αποτελέσματα](#)

Copyright © 2023 All Rights Reserved by Fotis Tsioumas.

Figure 47: Προβολή Ερωτηματολογίου – «φοιτητής»

Ο Λογαριασμός Μου

Η συγκεκριμένη σελίδα παρουσιάζει ακριβώς το ίδιο περιεχόμενο με αυτή του «καθηγητή» με τη διαφορά ότι το grid των στοιχείων δεν περιλαμβάνει το στοιχείο τηλέφωνο.



Πληροφορίες Λογαριασμού

Αριθμός Μητρώου:	edustud00002
Email:	ftsioumas@eduverse.gr
Όνομα:	Φώτιος
Επίθετο:	Τσιούμας

[Επεξεργασία Πληροφοριών](#)

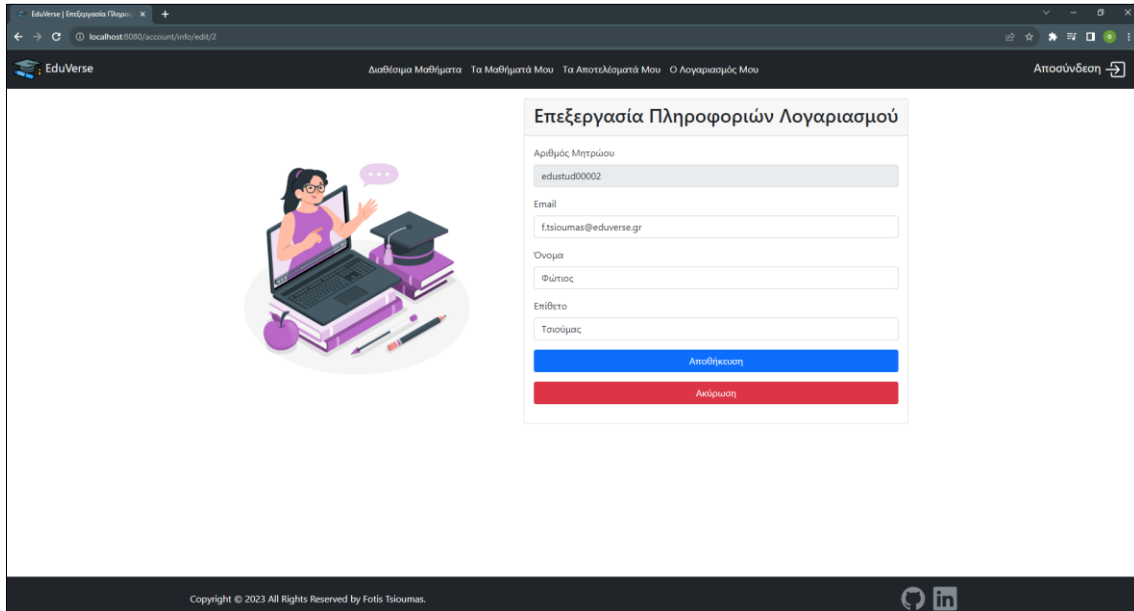
[Αλλαγή Κωδικού Πρόσβασης](#)

Copyright © 2023 All Rights Reserved by Fotis Tsioumas.

Figure 48: Ο Λογαριασμός Μου – «φοιτητής»

Επεξεργασία Πληροφοριών Λογαριασμού

Η συγκεκριμένη σελίδα παρουσιάζει ακριβώς το ίδιο περιεχόμενο με αυτή του «καθηγητή» με τη διαφορά ότι το grid των στοιχείων δεν περιλαμβάνει το στοιχείο τηλέφωνο.

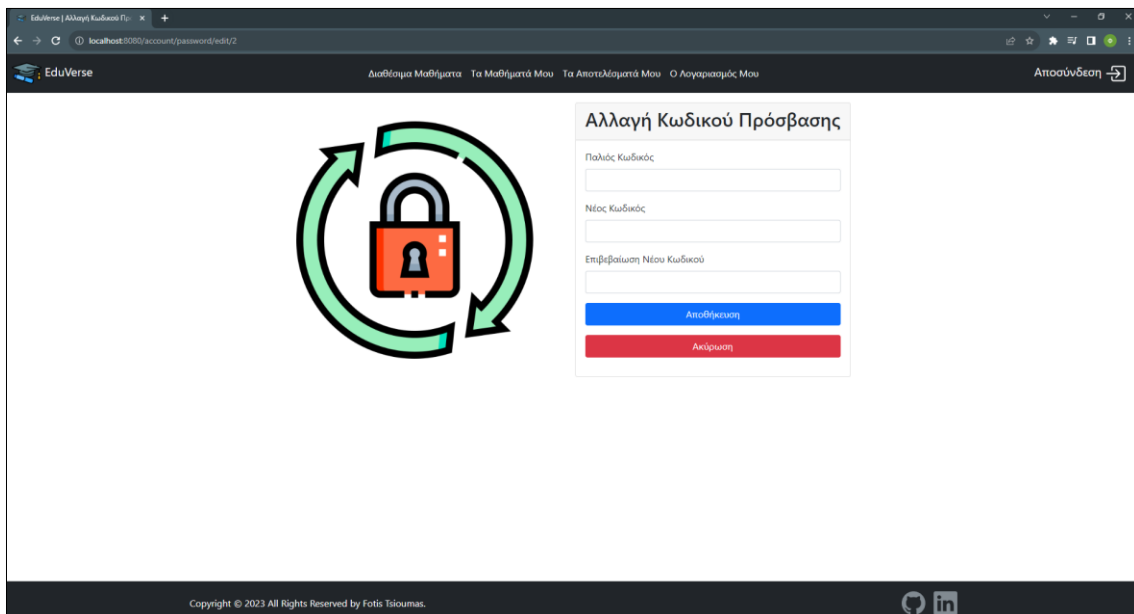


The screenshot shows a web browser window with the URL localhost:5050/account/info/edit/2. The page title is 'Επεξεργασία Πληροφοριών Λογαριασμού'. On the left, there is an illustration of a student sitting at a desk with a laptop, books, and a graduation cap. On the right, there is a form with the following fields: 'Αριθμός Μητρώου' (filled with 'edustud00002'), 'Email' (filled with 'ftsioumas@eduverse.gr'), 'Όνομα' (filled with 'Φώτιος'), and 'Επίθετο' (filled with 'Τσιούμας'). Below the form are two buttons: 'Αποθήκευση' (Save) and 'Ακύρωση' (Cancel). The footer contains 'Copyright © 2023 All Rights Reserved by Fotis Tsioumas.' and social media icons for LinkedIn.

Figure 49: Επεξεργασία Στοιχείων Λογαριασμού – «φοιτητής»

Αλλαγή Κωδικού Πρόσβασης

Η συγκεκριμένη σελίδα παρουσιάζει ακριβώς το ίδιο περιεχόμενο με αυτή του «καθηγητή» χωρίς καμία διαφορά.



The screenshot shows a web browser window with the URL localhost:5050/account/password/edit/2. The page title is 'Αλλαγή Κωδικού Πρόσβασης'. On the left, there is an illustration of a padlock with a keyhole, surrounded by a circular arrow indicating a cycle or refresh. On the right, there is a form with the following fields: 'Παλιός Κωδικός', 'Νέος Κωδικός', and 'Επιβεβαίωση Νέου Κωδικού'. Below the form are two buttons: 'Αποθήκευση' (Save) and 'Ακύρωση' (Cancel). The footer contains 'Copyright © 2023 All Rights Reserved by Fotis Tsioumas.' and social media icons for LinkedIn.

Figure 50: Αλλαγή Κωδικού Πρόσβασης – «φοιτητής»

Σελίδες Σφαλμάτων

Στις προηγούμενες ενότητες και συγκεκριμένα στο σημείο που αναφέρεται ποιος συνδεδεμένος χρήστης μπορεί να δει μια συγκεκριμένη σελίδα, αναφέρθηκαν κάποιες σελίδες σφαλμάτων και υπάρχουν και κάποιες επιπλέον οι οποίες δεν αναφέρθηκαν. Οι σελίδες σφαλμάτων και η σημασία του, ακολουθούν παρακάτω.

Σελίδα Σφάλματος 403

Το σφάλμα 403 αντιστοιχεί στο error 403 Forbidden/Access Denied και εμφανίζεται σε χρήστες όπου προσπαθούν να μπουν σε μια σελίδα όπου δεν έχουν το δικαίωμα πρόσβασης είτε λόγω ρόλου είτε γιατί δεν ανήκει στον συγκεκριμένο χρήστη η σελίδα.

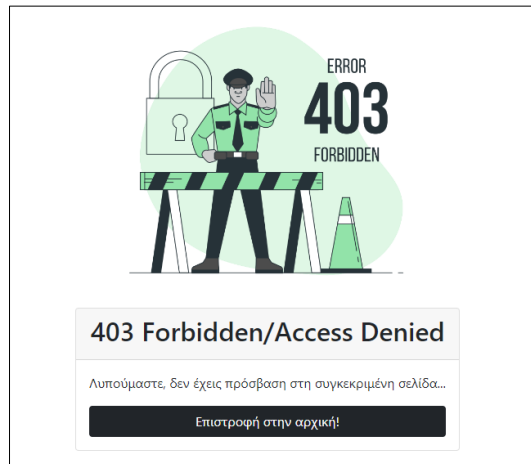


Figure 51: Σελίδα Σφάλματος 403

Σελίδα Σφάλματος 404

Το σφάλμα 404 αντιστοιχεί στο error 404 Page Not Found και εμφανίζεται σε χρήστες όπου προσπαθούν να μπουν σε μια σελίδα όπου δεν υπάρχει.

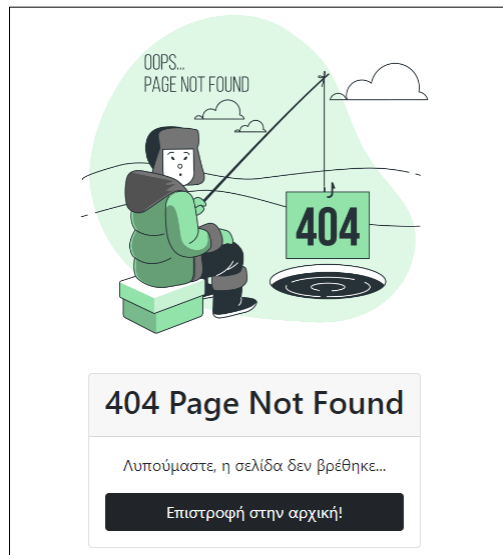


Figure 52: Σελίδα Σφάλματος 404

Σελίδα Σφάλματος 500

Το σφάλμα 500 αντιστοιχεί στο error 500 Internal Server Error και εμφανίζεται σε χρήστες όπου προσπαθούν να μπουν σε μια σελίδα όπου υπάρχει κάποιο συστημικό error. Σε αυτή την περίπτωση για λόγους ασφαλείας του συστήματος, δεν εμφανίζεται το error στην οθόνη, αλλά εμφανίζεται η σελίδα αυτή όπου ενημερώνει τον χρήστη για σφάλμα από τη μεριά του συστήματος και ανακατεύθυνση στην αρχική σελίδα.

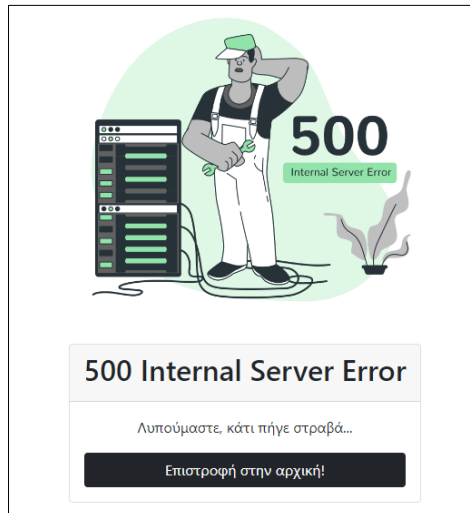


Figure 53: Σελίδα Σφάλματος 500

Σελίδα Υπόλοιπων Σφαλμάτων

Αν προκύψει οποιοδήποτε άλλο σφάλμα, ο πελάτης ενημερώνεται με την παρακάτω σελίδα.

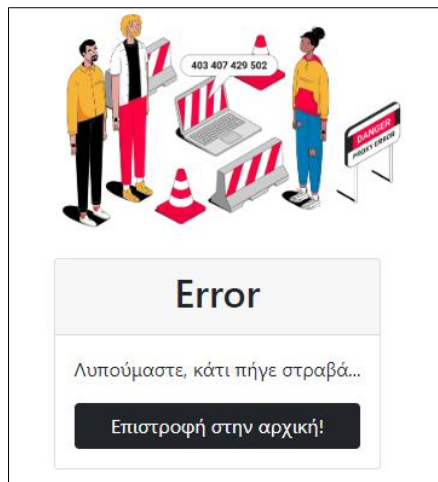


Figure 54: Σελίδα Υπόλοιπων Σφαλμάτων

Αρχιτεκτονική Συστήματος

Η συγκεκριμένη εφαρμογή, έχει αναπτυχθεί με τη χρήση Spring Boot. Το Spring Boot είναι ένα εργαλείο, που βασίζεται στο Spring Framework και επιτρέπει την εύκολη και γρήγορη ανάπτυξη Spring εφαρμογών, με ελάχιστες διαμορφώσεις και ρυθμίσεις του συστήματος. Το Spring Framework είναι ένα δημοφιλές, open-source, enterprise-level framework της Java, που χρησιμοποιείται για τη δημιουργία αυτόνομων (stand-alone), παραγωγικού επιπέδου (production-grade) εφαρμογών, οι οποίες εκτελούνται σε Java Virtual Machine (JVM). Μερικά από τα μεγαλύτερα προτερήματα που προσφέρει το Spring Boot είναι τα παρακάτω:

- Ενσωματωμένο HTTP web server (στη συγκεκριμένη περίπτωση Tomcat).
- Δυνατότητα επιλογής πακέτων/εργαλείων εκκίνησης και αυτόματη διαμόρφωση/ρύθμιση αυτών.
- Αυτόματη εγκατάσταση και ανάπτυξη Spring και 3rd-party βιβλιοθηκών.
- Παροχή λειτουργιών παραγωγικού περιβάλλοντος, όπως εργαλεία για metrics και health checks κ.α.

(Spring-Projects, 2023) | (IBM, 2023)

Γλώσσες, Ρυθμίσεις και Πακέτα

Για την δημιουργία Spring Boot εφαρμογών και την εγκατάσταση των βασικών πακέτων τους, είναι διαθέσιμο το εργαλείο Spring Initializr. Το συγκεκριμένο εργαλείο είναι ένα web application, το οποίο δίνει τη δυνατότητα, κατά τη δημιουργία του project, να επιλεγτούν οι παρακάτω απαραίτητες ρυθμίσεις:

- Η γλώσσα της εφαρμογής (Java, Kotlin, Groovy)
- Η version της γλώσσας
- Ο Builder/Package Manager (Gradle, Maven)
- Το JDK
- Τα dependencies του project

Το Spring Initializr παρέχει ένα API για την ενσωμάτωση του σε άλλες εφαρμογές, όπως IDEs, προσφέροντας την δυνατότητα για απευθείας χρήσης του, για τη δημιουργία Spring Boot projects εντός των εφαρμογών αυτών. Διαφορετικά, υπάρχει η δυνατότητα χρήσης του μέσω του ισότοπου «<https://start.spring.io>». (Spring-io, 2022) | (JetBrains, 2023)

Η συγκεκριμένη εκπαιδευτική πλατφόρμα, αναπτύχθηκε με τη χρήση του IDE IntelliJ, αρχικοποιήθηκε από το εργαλείο Spring Initializr και οι ρυθμίσεις και τα πακέτα που επιλέχθηκαν, αναφέρονται στον πίνακα που ακολουθεί:

Language:	Java
Version:	21
Builder/Package Manager:	Maven
JDK:	openjdk-21
Packaging:	Jar
Dependencies:	Spring Boot DevTools Spring Boot Web Spring Boot Security Spring Boot JPA MySQL Driver Thymeleaf Lombok Validation I/O

Πίνακας 1: Γλώσσα, ρυθμίσεις και πακέτα

Εκτός από τα παραπάνω, χρησιμοποιήθηκαν μια βάση δεδομένων MySQL, κώδικας CSS και JavaScript και εξωτερικά δίκτυα διανομής περιεχομένου (cdn) για τα παρακάτω libraries: Bootstrap v4.4.1 CSS, Bootstrap v5.0.2 CSS, Bootstrap v4.4.1 JS, Summernote v0.8.18 CSS, Summernote v0.8.18 JS, jQuery v3.5.1, popper.js v1.16.0.

Δομή Εφαρμογής

Η δομή των φακέλων και των αρχείων της εφαρμογής, έχει βασιστεί σε έναν συνδυασμό των προτύπων σχεδιασμού (design patterns), MVC (Model View Controller) και Controller-Service-Repository.

Η κάθε λειτουργικότητα του συστήματος, ξεκινάει από μια οντότητα (model/entity), η οποία βρίσκεται στον φάκελο «entity». Η κάθε οντότητα, διαθέτει από ένα repository, το οποίο κάνει extend το JpaRepository, ώστε να αποκτήσει πρόσβαση σε διάφορες μεθόδους που βοηθούν στην επικοινωνία του με την βάση δεδομένων. Σε καθένα από αυτά τα repositories, αποκτά πρόσβαση το αντίστοιχο service, το οποίο περιλαμβάνει μεθόδους με το business logic των λειτουργιών των οντοτήτων. Στη συνέχεια τα services χρησιμοποιούνται στους controllers, οι οποίοι διαχειρίζονται τα http request των users της εφαρμογής. Επίσης υπάρχει ένας φάκελος για τα configurations του Spring Boot Security και την προσθήκη seed data, που εισάγονται στη βάση κατά την πρώτη ανάπτυξη της εφαρμογής. Όλοι οι παραπάνω φάκελοι και τα αρχεία τους, που αποτελούν java classes, βρίσκονται στο path «src/main/java/unipi/fotistsiou/eduverse».

Κάθε controller εξυπηρετεί ένα σύνολο από requests, τα οποία είναι διαθέσιμα στους χρήστες μέσω των views, τα οποία είναι html templates. Για τη μορφοποίηση και την προσθήκη διαδραστικότητας στα templates, έχουν χρησιμοποιηθεί εξωτερικές βιβλιοθήκες και τοπικός κώδικας CSS και JavaScript. Οι εξωτερικές βιβλιοθήκες γίνονται προσβάσιμες μέσω δικτύων διανομής περιεχομένου (cdn) και ο τοπικός κώδικας, βρίσκεται μέσα στους φακέλους static/css και static/js, αντίστοιχα. Επίσης, το περιεχόμενο των templates, έχει εμπλουτιστεί με εικόνες οι οποίες βρίσκονται στον φάκελο static/images. Όλοι οι παραπάνω φάκελοι και τα αρχεία τους βρίσκονται στο path «src/main/resources».

Από τα παραπάνω, γίνεται αντιληπτό ότι το project αποτελεί μια monolithic εφαρμογή, όπου το back-end της βρίσκεται στο path «src/main/java/unipi/fotistsiou/eduverse» και το front-end της βρίσκεται στο path «src/main/resources». Επιπλέον, στον φάκελο resources βρίσκονται και τα application properties της εφαρμογής και έχει προστεθεί το data base schema μαζί με samples data, έπειτα από export της βάσης.

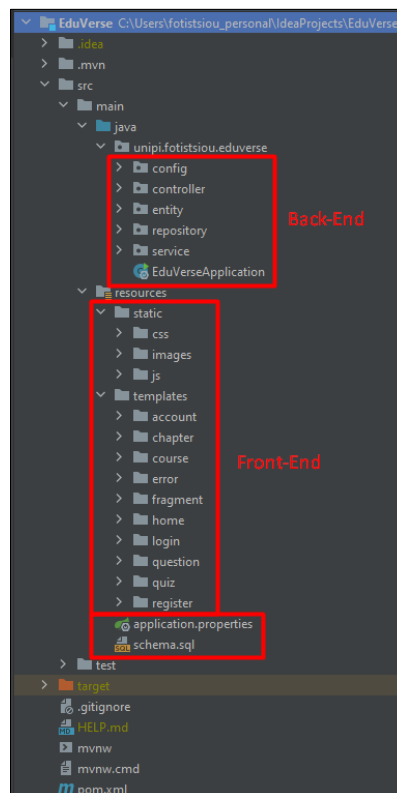


Figure 55: Δομή Εφαρμογής

Entities/Models

Όπως προαναφέρθηκε, οι οντότητες είναι το πιο βασικό δομικό στοιχείο της εφαρμογής. Οι οντότητες, περιλαμβάνουν χαρακτηριστικά (properties) και επιπλέον συσχετίζονται μεταξύ τους. Με την χρήση των κατάλληλων annotations που προσφέρει το Spring Boot, οι οντότητες, τα χαρακτηριστικά του και οι σχέσεις τους, δημιουργούν τους πίνακες της βάσης, τα πεδία των πινάκων, τις ιδιότητες αυτών (data base schema) και τα primary και foreign key τους για την δημιουργία σχέσεων μεταξύ τους. Επίσης, έχουν χρησιμοποιηθεί κατάλληλα annotations για να οριστούν τα ονόματα των πινάκων και των πεδίων τους.

```

13 @NoArgsConstructor
14 @Entity
15 @Table(name="course")
16 public class Course {
17     @Id
18     @GeneratedValue(strategy = GenerationType.IDENTITY)
19     private Long id;
20
21     @NotEmpty(message = "0 κωδικός μαθήματος δεν μπορεί να είναι κενός.")
22     @Column(name="code", nullable=false, unique=true)
23     private String code;
24
25     @NotEmpty(message = "0 τίτλος δεν μπορεί να είναι κενός.")
26     @Column(name="title", nullable=false)
27     private String title;
28
29     @NotEmpty(message = "Η περιγραφή δεν μπορεί να είναι κενή.")
30     @Column(name="description", nullable=false, columnDefinition = "TEXT")
31     private String description;
32
33     @ManyToOne(fetch = FetchType.EAGER)
34     @JoinColumn(name = "professor", referencedColumnName = "id", nullable = false)
35     private User professor;
36
37     @ManyToMany(fetch = FetchType.EAGER)
38     @JoinTable(
39         name = "course_students",
40         joinColumns = @JoinColumn(name = "course_id", referencedColumnName = "id"),
41         inverseJoinColumns = @JoinColumn(name = "student", referencedColumnName = "id")
42     )
43     private Set<User> students = new HashSet<>();

```

Figure 56: Annotations για τη δημιουργία του db schema

Εκτός από τους παραπάνω λόγους, έχουν προστεθεί annotation για έναν ακόμα πολύ σημαντικό λόγο. Αυτός ο λόγος είναι η αυτόματη δημιουργία getters, setters και constructors, χωρίς αυτά να εμφανίζονται στο body της κλάσης της οντότητας. Αυτό είναι ένα μεγάλο πλεονέκτημα, που προσφέρει το dependency «Lombok», το οποίο εγκαταστάθηκε κατά τη δημιουργία του project. Με αυτόν τον τρόπο ο κώδικας παραμένει καθαρός, καθώς αποφεύγεται η παρουσία πολλών γραμμών επαναλαμβανόμενου κώδικα, ο οποίος δεν περιλαμβάνει κάποια νέα λειτουργία. Επίσης, τον καθιστά πιο εύκολα συντηρήσιμο και επεκτάσιμο, διότι στις περιπτώσεις που χρειαστεί να προστεθούν ή να αφαιρεθούν χαρακτηριστικά στις οντοτήτων, προθέτονται ή αφαιρούνται αυτόματα οι getters, setters και τα χαρακτηριστικά στον constructor, κατά το build του project.

```

9 @Getter
10 @Setter
11 @NoArgsConstructor
12 @Entity
13 @Table(name="chapter")
14 public class Chapter {
15     @Id
16     @GeneratedValue(strategy = GenerationType.IDENTITY)
17     private Long id;
18
19     @NotEmpty(message = "0 τίτλος δεν μπορεί να είναι κενός.")
20     @Column(name="title", nullable=false)
21     private String title;
22

```

Figure 57: Annotations για Getters, Setters και Constructors

Επίσης, μέσα στις οντότητες, έχουν χρησιμοποιηθεί τα `@NotEmpty` annotations για να δηλωθεί ότι ένα συγκεκριμένο χαρακτηριστικό, δεν μπορεί να παραμείνει κενό κατά την αρχικοποίηση του object. Επιπλέον, δηλώνεται και το μήνυμα σφάλματος, το οποίο εμφανίζεται στον χρήστη, στην περίπτωση που προσπαθήσει να υποβάλει μια φόρμα χωρίς να συμπληρώσει το συγκεκριμένο χαρακτηριστικό. Προβολή της συγκεκριμένης λειτουργία έγινε στην ενότητα «Φόρμα Εγγραφής Καθηγητή», όπου ο χρήστης προσπαθήσε να υποβάλει τη φόρμα εγγραφής χωρίς να συμπληρώσει τα πεδία. Η παραπάνω δυνατότητα προσφέρεται μέσω του dependency «Validation (I/O)» που εγκαταστάθηκε κατά τη δημιουργία του project.

```

23 private String am;
24 @NotEmpty(message = "Το email δεν μπορεί να είναι κενό.")
25 @Column(name="email", nullable=false, unique=true)
26 private String email;
27
28 @NotEmpty(message = "Ο κωδικός πρόσβασης δεν μπορεί να είναι κενός.")
29 @Column(name="password", nullable=false)
30 private String password;
31
32 @NotEmpty(message = "Το όνομα δεν μπορεί να είναι κενό.")
33 @Column(name="first_name", nullable=false)
34 private String firstName;
35
36 @NotEmpty(message = "Το επώνυμο δεν μπορεί να είναι κενό.")
37 @Column(name="last_name", nullable=false)
38 private String lastName;
39
40 @Column(name="telephone")
41 private String telephone;

```

Figure 58: Annotations για validation fields

Τέλος, υπάρχει μέσα στα entities η κλάση «Quiz», η οποία δεν παράγει κάποιον πίνακα, αλλά αποτελεί ένα component, δηλώνοντάς το με το αντίστοιχο annotation και περιλαμβάνει ένα property, όπου αποτελεί μια λίστα από questions. Η συγκεκριμένη κλάση, έχει σχεδιαστεί για να μεταφέρει μια λίστα ερωτήσεων, από το backend στο frontend και το αντίστροφο, περιέχοντας τα απαραίτητα δεδομένα σε κάθε περίπτωση.

```

1 package unipi.fotistsiou.eduverse.entity;
2
3 import lombok.Getter;
4 import lombok.NoArgsConstructor;
5 import lombok.Setter;
6 import lombok.ToString;
7 import org.springframework.stereotype.Component;
8 import java.util.List;
9
10 usages Fotis Tsioumas
11 @Setter
12 @Getter
13 @NoArgsConstructor
14 @ToString
15 @Component
16 public class Quiz {
17     private List<Question> questions;
18 }

```

Figure 59: Quiz Component

Repositories

Κάθε οντότητα έχει το αντίστοιχο repository της. Το όνομα του repository, φανερώνει την οντότητα όπου εξυπηρετεί. Τα repositories, είναι interfaces, που κάνουν extend το JpaRepository, κατά το οποίο ορίζεται το όνομα της οντότητα/πίνακα και ο τύπος id της. Με τον συγκεκριμένο τρόπο, καταργείται η ανάγκη υλοποίησης CRUD μεθόδων, για την συγκεκριμένη οντότητα. Έτσι γίνεται δυνατή η εισαγωγή, η ανάγνωση, η ενημέρωση και η διαγραφή εγγραφών της βάσης, χωρίς να απαιτείται η υλοποίηση των μεθόδων αυτών. Σε πολλές περιπτώσεις, το repository αποκτά πρόσβαση σε όλες τις απαραίτητες μεθόδους, μόνο μέσω του extend του JpaRepository και το body του παραμένει κενό.

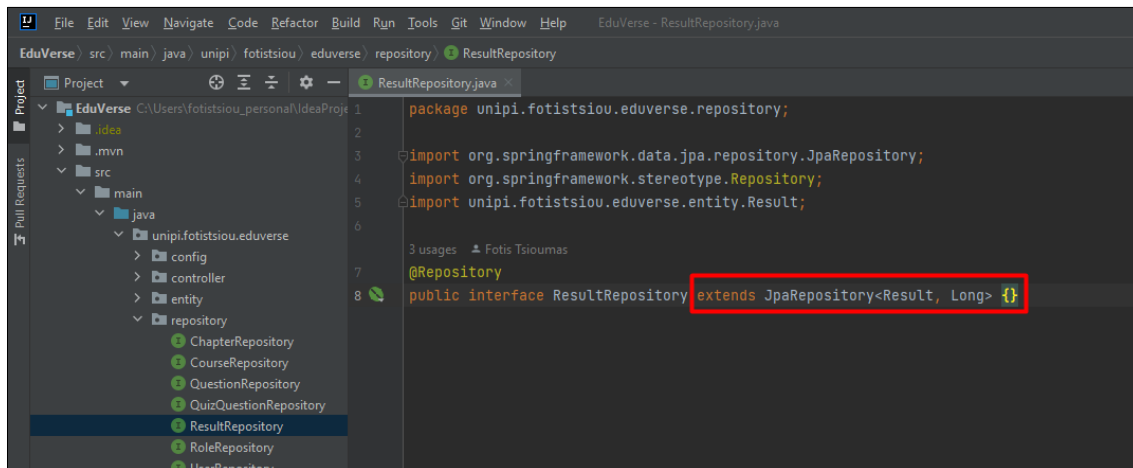


Figure 60: Repository χωρίς επιπλέον μεθόδους

Σε άλλες περιπτώσεις, ακολουθώντας συγκεκριμένες συμβάσεις που αφορούν την ονοματολογία, ορίζονται νέες μέθοδοι, οι οποίες και πάλι δεν χρειάζεται να υλοποιηθούν. Μια τέτοια περίπτωση, είναι η ανάγκη αναζήτησης μιας συγκεκριμένης εγγραφής, αναζητώντας την εγγραφή μέσω ενός πεδίου που δεν αποτελεί το id της οντότητας.

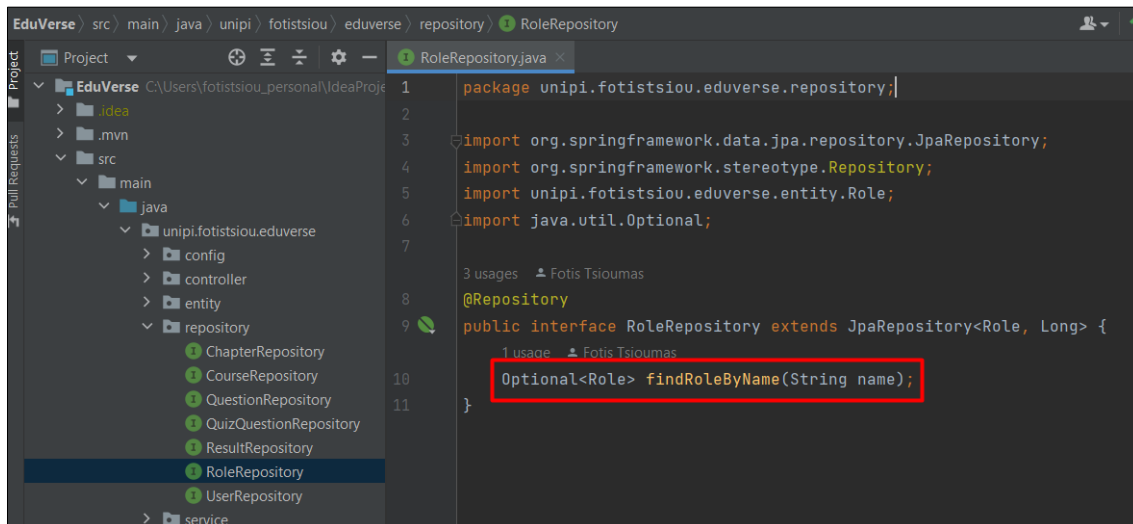


Figure 61: Repository με επιπλέον μεθόδους

Τέλος, υπάρχει η δυνατότητα ορισμού μεθόδου, που εκτελεί custom query προς τη βάση. Σε αυτή την περίπτωση, γίνεται χρήση συγκεκριμένων annotations και το query συντάσσεται μέσα στο αντίστοιχο annotation. Ένα τέτοιο παράδειγμα είναι η ανάγκη ενημέρωσης συγκεκριμένων στοιχείων μια εγγραφής, με τη χρήση ενός update query.


```

12 @Repository
13 public interface UserRepository extends JpaRepository<User, Long> {
14     Optional<User> findUserByEmail(String email);
15
16     @Modifying
17     @Transactional
18     @Query(
19         "UPDATE User u SET " +
20         "u.am = :am " +
21         "WHERE u.id = :id"
22     )
23     void update(Long id, String am);
24     void updateUserDetailsAm(
25         @Param("id") Long id,
26         @Param("am") String am
27     );

```

Figure 62: Repository με query μέθοδο

Services

Για κάθε οντότητα υπάρχει επίσης και ένα service. Τα services είναι ο συνδετικός κρίκος ανάμεσα στους controllers και τα repositories και έχουν επιφορτιστεί όλο το business logic της εφαρμογής. Πριν την έναρξη της κλάσης του service, υπάρχει το annotation @Service και στην αρχή του body της κλάσης, γίνεται η σύνδεση του service με τις εξαρτήσεις του, μέσω του design pattern «dependency injection» και την χρήση του annotation @Autowired. Οι εξαρτήσεις του είναι σίγουρα το repository που φέρει το αντίστοιχο όνομα και σε αρκετές περιπτώσεις είναι και services άλλων οντοτήτων.

```

14 @Service
15 public class UserService {
16     private final UserRepository userRepository;
17     private final RoleService roleService;
18     private final PasswordEncoder passwordEncoder;
19
20     @Autowired
21     public UserService (
22         UserRepository userRepository,
23         RoleService roleService,
24         PasswordEncoder passwordEncoder
25     ){
26         this.userRepository = userRepository;
27         this.roleService = roleService;
28         this.passwordEncoder = passwordEncoder;
29     }

```

Figure 63: Η σύνδεση του service με τις εξαρτήσεις του

Τα services της εφαρμογής έχουν κάποιες κοινές και κάποιες μοναδικές μεθόδους, ως προς την λειτουργικότητά τους. Οι κοινές μεθόδων που συναντώνται στα περισσότερα services, αφορούν την εύρεση ενός record/object μέσω του primary key του πίνακα ή μέσω κάποιου άλλου πεδίου, την εύρεση ενός συνόλου από records/objects που έχουν κάποιο κοινό χαρακτηριστικό, την δημιουργία/αποθήκευση ενός νέου record/object σε έναν πίνακα, την ενημέρωση κάποιων στοιχείων ενός record/object και την διαγραφή ενός record/object.

```

9 usages Fotis Tsioumas
public Optional<Course> findCourseById(Long id) { return courseRepository.findById(id); }

2 usages Fotis Tsioumas
public Optional<Course> findCourseByCode(String code) { return courseRepository.findCourseByCode(code); }

```

Figure 64: Εύρεση μοναδικού record/object

```

1 usage Fotis Tsioumas
public List<Question> findAllChapterQuestions(Long chapterId) {
    List<Question> chapterQuestions = new ArrayList<>();
    List<Question> questions = questionRepository.findAll();
    for (Question question:questions) {
        if (question.getChapter().getId().equals(chapterId)) {
            chapterQuestions.add(question);
        }
    }
    return chapterQuestions;
}

```

Figure 65: Εύρεση συνόλου record/object

```

2 usages
public void saveQuestion(Question question) {
    Fotis Tsioumas
    if (question.getId() == null) {
        question.setChoice(0);
    }
    questionRepository.save(question);
}

```

Figure 66: Δημιουργία νέου record/object

```

1 usage Fotis Tsioumas
public void updateUserDetails(User user) {
    userRepository.updateUserDetails(
        user.getId(),
        user.getEmail(),
        user.getFirstName(),
        user.getLastName(),
        user.getTelephone()
    );
}

```

Figure 67: Ενημέρωση ενός record/object

```

1 usage Fotis Tsioumas
public void deleteCourse(Course course) { courseRepository.delete(course); }

```

Figure 68: Διαγραφή ενός record/object

Υπάρχουν όμως και μοναδικές μέθοδοι για κάθε service, στις οποίες υλοποιούνται λειτουργίες που αφορούν πιο συγκεκριμένες ανάγκες της εφαρμογής. Για παράδειγμα, το πεδίο «Αριθμός Μητρώου» ενός χρήστη, παράγεται αυτόματα μέσω της μεθόδου «addAm» του «UserService» και προστίθεται στην εγγραφή του χρήστη, αφού αυτή δημιουργηθεί με τα στοιχεία που έχει εισαχθεί κατά τη συμπλήρωση της φόρμας εγγραφής. Αυτή η μέθοδος, δέχεται ως argument το object του user που μόλις δημιουργήθηκε στη βάση και παράγει τον αριθμό μητρώου, συνδυάζοντας ένα prefix, το οποίο διαφέρει αν ο user είναι «καθηγητής» ή «φοιτητής» και το id που πήρε η εγγραφή του στον πίνακα, μετατρέποντας το id σε πενταψήφιο αριθμό, προσθέτοντας μηδενικά την αρχή του.

```

55 @
56 public void addAm(User user) {
57     String am = null;
58     if (user.getId() != null) {
59         am = generateAm(user.getId(), user.getRoles().toString());
60         if (!am.isEmpty()) {
61             user.setAm(am);
62         }
63     }
64     userRepository.updateUserDetailsAm(user.getId(), am);
65 }
66
67 private String generateAm(Long userId, String role) {
68     String prefix = role.contains("ROLE_PROFESSOR") ? "eduprof" : "edustud";
69     String userIdStr = String.format("%05d", userId);
70     return prefix + userIdStr;
71 }

```

Figure 69: Μέθοδος «addAm»

Μία άλλη συνάρτηση που είναι μοναδική είναι η «findMyCourses» του «CourseService». Η συγκεκριμένη συνάρτηση έχει δημιουργηθεί για να επιστρέφει τα μαθήματα του grid της σελίδας «Τα Μαθήματα Μου». Δέχεται σαν ορίσματα το userId και το role και στη συνέχεια αναλόγως το role ψάχνει αν τα μαθήματα στα οποία υπάρχει το userId είτε στο πεδίο «professor» εάν πρόκειται για καθηγητή, είτε στη λίστα του πεδίου «students» ένα πρόκειται για μαθητή.

```

public List<Course> findMyCourses(Long userId, String role) {
    List<Course> myCourses = new ArrayList<>();
    List<Course> courses = courseRepository.findAll();
    for (Course course:courses) {
        if (role.contains("ROLE_PROFESSOR")) {
            if (course.getProfessor().getId().equals(userId)) {
                myCourses.add(course);
            }
        } else if (role.contains("ROLE_STUDENT")) {
            boolean isUserEnrolled = false;
            for (User student:course.getStudents()) {
                if (student.getId().equals(userId)) {
                    isUserEnrolled = true;
                    break;
                }
            }
            if (isUserEnrolled) {
                myCourses.add(course);
            }
        }
    }
    return myCourses;
}

```

Figure 70: Μέθοδο «findMyCourses»

Η αντίστοιχη συνάρτηση, η οποία έχει δημιουργηθεί για να γεμίζει την σελίδα της προβολής των μαθημάτων με τα κεφάλαια του είναι πιο απλή. Δέχεται σαν όρισμα μόνο το courseId και στη συνέχεια πραγματοποιεί αναζήτηση του στο πεδίο course του πίνακα «chapter». Η συγκεκριμένη συνάρτηση ονομάζεται «findAllCourseChapters» και βρίσκεται στο «ChapterService».

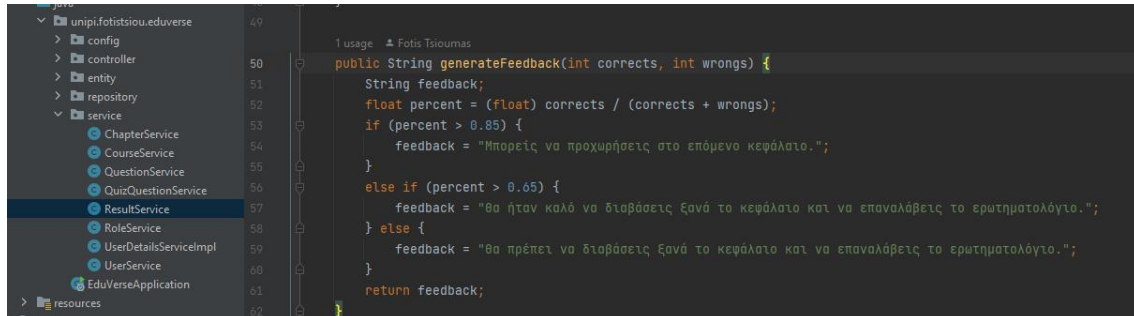
```

1 usage Fotis Tsioumas
public List<Chapter> findAllCourseChapters(Long courseId) {
    List<Chapter> courseChapters = new ArrayList<>();
    List<Chapter> chapters = chapterRepository.findAll();
    for (Chapter chapter:chapters) {
        if (chapter.getCourse().getId().equals(courseId)) {
            courseChapters.add(chapter);
        }
    }
    return courseChapters;
}

```

Figure 71: Μέθοδος «findAllCourseChapters»

Μια ενδιαφέρουσα συνάρτηση είναι επίσης αυτή που παράγει το feedback που λαμβάνει ο φοιτητής όταν ολοκληρώσει ένα ερωτηματολόγιο. Η συγκεκριμένη συνάρτηση δέχεται ως ορίσματα τις σωστές και λάθος απαντήσεις του ερωτηματολογίου και παράγει το αντίστοιχο σχόλιο.



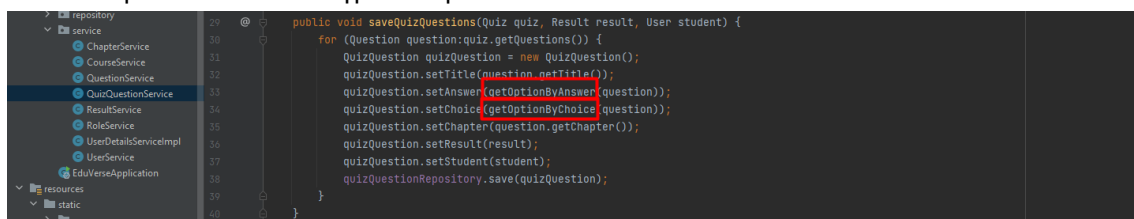
```

1 usage  Fotis Tsioumas
public String generateFeedback(int corrects, int wrongs) {
    String feedback;
    float percent = (float) corrects / (corrects + wrongs);
    if (percent > 0.85) {
        feedback = "Μπορείτε να προχωρήσετε στο επόμενο κεφάλαιο.";
    }
    else if (percent > 0.65) {
        feedback = "Θα ήταν καλό να διαβάσετε ξανά το κεφάλαιο και να επαναλάβετε το ερωτηματολόγιο.";
    }
    else {
        feedback = "Θα πρέπει να διαβάσετε ξανά το κεφάλαιο και να επαναλάβετε το ερωτηματολόγιο.";
    }
    return feedback;
}

```

Figure 72: Μέθοδος «generateFeedback»

Ο τελευταίος μηχανισμός που θα αναφερθεί είναι και από τελείται από περισσότερες από μια συναρτήσεις είναι αυτός που κρατάει τις ερωτήσεις με τις απαντήσεις που έχει δώσει ο χρήστης, μετά την ολοκλήρωση του ερωτηματολογίου και γεμίζει το grid των σελίδων «Προβολή Ερωτηματολογίου». Οι ερωτήσεις με τις απαντήσεις ενός ολοκληρωμένου ερωτηματολογίου, αποθηκεύονται στον πίνακα «quiz_question» μέσω της οντότητας QuizQuestion. Ο σκοπός της αποθήκευσης των ερωτήσεων σε νέο πίνακα μαζί με τις απαντήσεις που έχει δώσει ο χρήστης, είναι να παραμένουν αναλλοίωτες οι πληροφορίες των ερωτήσεων του ολοκληρωμένου ερωτηματολογίου, στην περίπτωση που ο καθηγητής επεξεργαστεί μια υπάρχουσα ερώτηση και επιπλέον να μην επηρεάζεται το ολοκληρωμένο ερωτηματολόγιο σε ένα κεφάλαιο, ακόμα και στην περίπτωση που ο καθηγητής προσθέσει ή αφαιρέσει ερωτήσεις στο συγκεκριμένο κεφάλαιο. Για την αποθήκευση της λίστας των ερωτοαπαντήσεων του ολοκληρωμένου ερωτηματολογίου στον αντίστοιχο πίνακα, έχει δημιουργηθεί η μέθοδος «saveQuizQuestions», η οποία περιλαμβάνει τις βοηθητικές μεθόδους «getOptionByAnswer» και «getOptionByChoice» και όλες μαζί βρίσκονται στο QuizQuestionService. Η μέθοδος «saveQuizQuestions» δέχεται ως ορίσματα, ένα object τύπου Quiz που περιλαμβάνει μία λίστα με τις ερωτοαπαντήσεις του ολοκληρωμένου ερωτηματολογίου, ένα object τύπου Result το οποίο έχει δημιουργηθεί κατά την ολοκλήρωση του ερωτηματολογίου και μια οντότητα τύπου User όπου είναι ο φοιτητής που πραγματοποίησε το ερωτηματολόγιο. Στην συνέχεια η μέθοδος χρησιμοποιεί ένα for loop για να διατρέξει τη λίστα των ερωτοαπαντήσεων του ερωτηματολογίου και για κάθε ερωτοαπάντηση να δημιουργήσει μια εγγραφή στον πίνακα. Σε κάθε εγγραφή αποθηκεύεται ο τίτλος της ερώτησης, η σωστή απάντηση για την ερώτησης, η απάντηση που έδωσε ο χρήστης, το κεφάλαιο της ερώτησης, το result με το οποίο αντιστοιχεί (δευτερεύον κλειδί) και ο user με τον οποίο αντιστοιχεί (δευτερεύον κλειδί). Για την αποθήκευση της σωστής απάντησης της ερώτησης και της απάντησης του χρήστη χρησιμοποιούνται οι μέθοδοι «getOptionByAnswer» και «getOptionByChoice» αντίστοιχα. Οι συγκεκριμένες μέθοδοι δέχονται ως ορίσματα την ερωτοαπάντηση του ερωτηματολογίου, και αφού φορτώσουν την ερώτηση όπως αυτή υπάρχει στον πίνακα question τότε επιστρέφουν το λεκτικό της απάντησης και της επιλογής του χρήστη. Οι συγκεκριμένες διαδικασίες χρειάζονται, καθώς οι ερωτοαπαντήσεις δεν περιλαμβάνουν τα options της ερώτησης αλλά τον αριθμό του σωστού option και του επιλεγμένου option.



```

public void saveQuizQuestions(Quiz quiz, Result result, User student) {
    for (Question question: quiz.getQuestions()) {
        QuizQuestion quizQuestion = new QuizQuestion();
        quizQuestion.setTitle(question.getTitle());
        quizQuestion.setAnswer(getOptionByAnswer(question));
        quizQuestion.setChoice(getOptionByChoice(question));
        quizQuestion.setChapter(question.getChapter());
        quizQuestion.setResult(result);
        quizQuestion.setStudent(student);
        quizQuestionRepository.save(quizQuestion);
    }
}

```

Figure 73: Μέθοδος «saveQuizQuestions»

```

54 // ----- Helper Functions ----- */
55
56 1 usage: Fotis Tsioumas
57 @ private String getOptionByAnswer(Question question) {
58     String optionAnswer = "something went wrong";
59     Optional<Question> optionalQuestion = questionService.findQuestionById(question.getId());
60     if (optionalQuestion.isPresent()) {
61         Question originQuestion = optionalQuestion.get();
62         optionAnswer = switch (question.getAnswer()) {
63             case 1 -> originQuestion.getOption_a();
64             case 2 -> originQuestion.getOption_b();
65             case 3 -> originQuestion.getOption_c();
66             default -> "something went wrong";
67         };
68     }
69     return optionAnswer;
70 }
71
72 1 usage: Fotis Tsioumas
73 @ private String getOptionByChoice(Question question) {
74     String optionChoice = "something went wrong";
75     Optional<Question> optionalQuestion = questionService.findQuestionById(question.getId());
76     if (optionalQuestion.isPresent()) {
77         Question originQuestion = optionalQuestion.get();
78         optionChoice = switch (question.getChoice()) {
79             case 1 -> originQuestion.getOption_a();
80             case 2 -> originQuestion.getOption_b();
81             case 3 -> originQuestion.getOption_c();
82             default -> "something went wrong";
83         };
84     }
85     return optionChoice;
86 }

```

Figure 74: Μέθοδοι «getOptionByAnswer» και «getOptionByChoice»

Τέλος, υπάρχει ένα επιπλέον service όπου είναι απαραίτητο για τη φόρτωση των πληροφοριών του user, κατά την διαδικασία αυθεντικοποίησής του, όταν πραγματοποιεί σύνδεσης στην εφαρμογή. Το service, ονομάζεται «UserDetailsServiceImpl» και κάνει implement το interface «UserService», υλοποιώντας την μέθοδο «loadUserByUsername». Η υλοποίηση της συγκεκριμένης μεθόδους, δέχεται σαν όρισμα ένα username (στη συγκεκριμένη περίπτωση χρησιμοποιείται το email) και με βάση αυτό το value του ορίσματος αναζητείται ένα συχρήστης στη βάση. Εάν βρεθεί ο χρήστης, δημιουργείται ένα αντικείμενο, το οποίο περιλαμβάνει το email του χρήστη, τον κωδικό πρόσβασης και μια λίστα με πεδία, τα οποία θα χρειαστούν για την περεταίρω αυθεντικοποίησής του χρήστη κατά την πλοήγηση του στην εφαρμογή (στη συγκεκριμένη περίπτωση χρειάζεται ο ρόλος του μόνο). Εάν δεν βρεθεί ο χρήστης, στέλνεται ένα UsernameNotFoundException σφάλμα με ένα συγκεκριμένο μήνυμα, που προβάλλεται στο user interface.

```

1 package unipi.fotistsiou.eduverse.service;
2
3 import ...
4
5 1 usage: Fotis Tsioumas
6 @Service
7 @Component("UserDetailsService")
8 public class UserDetailsServiceImpl implements UserDetailsService {
9     2 usages
10     private final UserService userService;
11
12     1 usage: Fotis Tsioumas
13     @Autowired
14     public UserDetailsServiceImpl(UserService userService) { this.userService = userService; }
15
16     no usages: Fotis Tsioumas
17     @Override
18     public UserDetails loadUserByUsername(String email) throws UsernameNotFoundException {
19         Optional<User> optionalUser = userService.findUserByEmail(email);
20         if (optionalUser.isPresent()) {
21             User user = optionalUser.get();
22             return new org.springframework.security.core.userdetails.User(
23                 user.getEmail(),
24                 user.getPassword(),
25                 user.getRoles().set<Role>()
26                     .stream().Stream<Role>()
27                     .map((role) -> new SimpleGrantedAuthority(role.getName())) Stream<SimpleGrantedAuthority>
28                     .collect(Collectors.toList());
29             );
30         } else {
31             throw new UsernameNotFoundException("Μη έγκυρο όνομα ή κωδικός.");
32         }
33     }
34 }

```

Figure 75: Service «UserDetailsServiceImpl»

Controllers

Οι controllers είναι αυτοί που διαχειρίζονται τα http request του front-end και μεταφέρουν πληροφορίες από το front-end το back-end, αλλά και το αντίστροφο. Η ονοματοδοσία τους έχει γίνει με βάση τις σελίδες που εξυπηρετούν. Κάθε controller περιλαμβάνει το annotation `@Controller` και αποκτά πρόσβαση σε τα services που χρειάζεται μέσω dependency injection. Οι controllers έχουν επιφορτιστεί την αυθεντικοποίηση του χρήστη είτε με τη χρήση annotations είτε με τη χρήση manual redirections.

Σε πολλές περιπτώσεις κατά την ανάλυση της λειτουργίας και συγκεκριμένα των περιορισμών σχετικά με το ποιοι χρήστες μπορούν να επισκεφτούν μια σελίδα, έγινε αναφορά στο ότι αν δεν πληροί ένα χρήστης τις συνθήκες οδηγείτε στην σελίδα 403. Μια από αυτές τις συνθήκες ήταν ο user να είναι συνδεδεμένος χρήστης ή να έχει συγκεκριμένο ρόλο. Για να οριστεί ότι μια σελίδα είναι διαθέσιμη μόνο για συνδεδεμένους χρήστες, χρησιμοποιείται στον controller το «`@PreAuthorize("isAuthenticated()")`» annotation, ενώ για να οριστεί ότι μια σελίδα είναι συνδεδεμένη με ένα συγκεκριμένο ρόλο χρηστών, χρησιμοποιούνται τα annotation

- `@PreAuthorize("hasRole('ROLE_PROFESSOR')")` για καθηγητές
- `@PreAuthorize("hasRole('ROLE_STUDENT')")` για μαθητές

Στις παραπάνω περιπτώσεις, όταν δεν πληροί ο χρήστης τις συνθήκες και προσπαθήσει να επισκεφτεί μια σελίδα, προκύπτει συστημικό 403 error. Επίσης, όταν επισκεφτεί μια σελίδα που δεν υπάρχει controller για το url της, τότε προκύπτει συστημικό 404 error. Υπάρχουν όμως και περιπτώσεις, όπου έχει γραφτεί κώδικας για manual redirection στις σελίδες 403 και 404 και σε αυτές τις περιπτώσεις η μη τήρηση συνθηκών έχει να κάνει με λογικές της συγκεκριμένης εφαρμογής και δεν μπορούν να καλυφθούν με τη χρήση annotations. Ένας από τους πιο συχνούς λόγους, για τον οποίο υπάρχει κώδικας σχεδόν σε κάθε controller που αφορά σελίδα που προσφέρεται μόνο σε συνδεδεμένους χρήστες, είναι ο συνδεδεμένος χρήστης να έχει user id αυτό που φαίνεται στο url της σελίδας. Παρακάτω φαίνεται ο κώδικας που αντιστοιχεί στη συγκεκριμένη λογική. Αρχικά ελέγχεται αν υπάρχει user με το id της παραμέτρου του url και στη συνέχεια ελέγχεται αν ο user που προκύπτει από το id της παραμέτρου είναι ο συνδεδεμένος user, οι πληροφορίες του οποίου έχουν φορτώσει στο object τύπου `Principal`, το οποίο γεμίζει κατά την σύνδεση του χρήστη με τις πληροφορίες που έχουν φορτώσει από το `UserDetailsServiceImpl`. Αν δεν υπάρχει ο user με το id της παραμέτρου, ο χρήστης οδηγείται στην 404 σελίδα σφάλματος, ενώ αν υπάρχει αλλά δεν είναι ο χρήστης που τον συνδεδεμένο λογαριασμό, οδηγείται στην 403 σελίδα σφαλμάτων, καθώς δεν πρέπει να έχει πρόσβαση σε σελίδες άλλων χρηστών.

```

123 @GetMapping("/course/student/all/{userId}")
124 @PreAuthorize("hasRole('ROLE_PROFESSOR')")
125 public String getMyCoursesStudentAll(
126     @PathVariable Long userId,
127     Model model,
128     Principal principal
129 ) {
130     String authUsername = "anonymous";
131     if (principal != null) {
132         authUsername = principal.getName();
133     }
134     Optional<User> optionalUser = userService.findUserById(userId);
135     if (optionalUser.isPresent()) {
136         User user = optionalUser.get();
137         if (user.getEmail().equals(authUsername)) {
138             List<Course> courses = courseService.findMyCourses(userId, role: "ROLE_PROFESSOR");
139             model.addAttribute("courses", courses);
140             return "course/course_student_all";
141         }
142         return "error/error_403";
143     }
144     return "error/error_404";
145 }

```

Figure 76: Περιορισμός πρόσβασης σε σελίδα βάσει του id του url

Μια άλλη περίπτωση στην οποία γίνεται custom έλεγχος για να αποφασιστεί αν εμφανιστεί το περιεχόμενο της σελίδας στον χρήστη, είναι η περίπτωση της σελίδας προβολή μαθήματος, όπου πέρα από του παραπάνω περιορισμούς, ελέγχεται και αν ο συνδεδεμένος χρήστης, αφού υπάρχει το μάθημα με id αυτό το url, είναι ως καθηγητής ο δημιουργός του μαθήματος ή ως φοιτητής

εγγεγραμμένος στο συγκεκριμένο μάθημα. Αν δεν πληρείται ο χρήστης το κριτήριο που αντιστοιχεί στον ρόλο του, οδηγείται στην 403 σελίδα σφάλματος.

```

309 @GetMapping("/{courseId}/{userId}")
310 @PreAuthorize("isAuthenticated()")
311 public String getCourse(
312     @PathVariable Long courseId,
313     @PathVariable Long userId,
314     Model model,
315     Principal principal
316 ){
317     String authUsername = "anonymousUser";
318     if (principal != null) {
319         authUsername = principal.getName();
320     }
321     Optional<User> optionalUser = userService.findUserById(userId);
322     if (optionalUser.isPresent()) {
323         User user = optionalUser.get();
324         if (user.getEmail().equals(authUsername)) {
325             Optional<Course> optionalCourse = courseService.findCourseById(courseId);
326             if (optionalCourse.isPresent()) {
327                 Course course = optionalCourse.get();
328                 if (course.getStudents().contains(user) || course.getProfessor().getId().equals(userId)) {
329                     List<Chapter> chapters = chapterService.findAllCourseChapters(courseId);
330                     String role = user.getRoles().toString();
331                     model.addAttribute("course", course);
332                     model.addAttribute("chapters", chapters);
333                     model.addAttribute("userId", userId);
334                     model.addAttribute("role", role);
335                     return "course/course_view";
336                 }
337                 return "error/error_403";
338             }
339             return "error/error_404";
340         }
341         return "error/error_403";
342     }
343     return "error/error_404";
344 }
345

```

Figure 77: Περιορισμός προβολής μαθήματος

Ο πιο σύνθετος έλεγχος για το αν θα προβληθεί η σελίδα σε έναν χρήστη, είναι στην περίπτωση που ένας καθηγητής πλοηγηθεί στη σελίδα προβολής ερωτηματολογίου, ενός φοιτητή. Σε αυτή την περίπτωση, πέρα από τον έλεγχο που γίνεται και αφορά τον αν ο συνδεδεμένος χρήστης είναι ο χρήστης με το professor id της παραμέτρου, γίνεται ένα τριπλός έλεγχος με βάση το student id της παραμέτρου και το result id της παραμέτρου. Αρχικά ελέγχεται το αν το result που προκύπτει από το result id της παραμέτρου είναι ερωτηματολόγιο σε κεφάλαιο μαθήματος που έχει δημιουργήσει ο συγκεκριμένος καθηγητής. Στη συνέχεια εκλέγεται αν το result που προκύπτει από το result id της παραμέτρου είναι ερωτηματολόγιο σε κεφάλαιο μαθήματος που είναι εγγεγραμμένος ο φοιτητής που προκύπτει από το student id της παραμέτρου. Τέλος, ελέγχεται αν το result που προκύπτει από το result id της παραμέτρου είναι ερωτηματολόγιο που ανήκει στον φοιτητή που προκύπτει από το student id της παραμέτρου.

```

230 Optional<User> optionalStudent = userService.findUserById(studentId);
231 if (optionalStudent.isPresent()) {
232     Optional<Result> optionalResult = resultService.findResultById(resultId);
233     if (optionalResult.isPresent()) {
234         User student = optionalStudent.get();
235         Result result = optionalResult.get();
236         if (
237             result.getChapter().getCourse().getProfessor().getId().equals(professorId)
238             &&
239             result.getChapter().getCourse().getStudents().contains(student)
240             &&
241             result.getStudent().getId().equals(studentId)
242         ){
243             List<QuizQuestion> quizQuestions = quizQuestionService.findQuizQuestionsByResultId(resultId);
244             String role = professor.getRoles().toString();
245             Long courseId = result.getChapter().getCourse().getId();
246             model.addAttribute("result", result);
247             model.addAttribute("quizQuestions", quizQuestions);
248             model.addAttribute("studentId", studentId);
249             model.addAttribute("userId", professorId);
250             model.addAttribute("student", student);
251             model.addAttribute("role", role);
252             model.addAttribute("courseId", courseId);
253             return "quiz/quiz_question";
254         }
255         return "error/error_403";
256     }
257     return "error/error_404";
258 }

```

Figure 78: Προβολή ερωτηματολογίου από καθηγητή

Πέρα από την λειτουργία των controllers να ελέγχουν ποιος έχει πρόσβαση σε μια σελίδα, υπάρχει ο ρόλος της μεταφοράς πληροφοριών από το front-end στο back-end και το αντίστροφο.

Η μεταφορά πληροφοριών από το front-end στο back-end, οδηγεί είτε σε προσθήκη είτε σε ενημέρωση είτε σε διαγραφή εγγραφών στη βάση και αυτό γίνεται μέσω post requests. Οι controllers διαχειρίζονται τα post request με τη βοήθεια μεθόδων που περιλαμβάνουν το «@PostMapping» annotation. Αφού λάβουν τα δεδομένα από τις φόρμες του front-end μέσω των post request, καλούν μεθόδους από τα services με τα οποία έχουν συνδεθεί και τη διαχείριση των πληροφοριών μέχρι την ενέργεια στην βάση, το business logic δηλαδή, το αναλαμβάνουν από εκεί και πέρα τα services αυτά.

Ακολουθεί ως παράδειγμα, η αποθήκευση ενός κεφαλαίου στη βάση μετά την υποβολή της φόρμας δημιουργίας κεφαλαίου από έναν καθηγητή. Στο παρακάτω παράδειγμα, φαίνεται επίσης ο έλεγχος που γίνεται, όσον αφορά τη σωστή συμπλήρωση της φόρμας, πριν γίνει η κλήση της μεθόδου που θα αποθηκεύσει το κεφάλαιο στη βάση. Συγκεκριμένα, χρησιμοποιείται ένα object τύπου BindingResult, το οποίο περιλαμβάνει τα σφάλματα, που έχουν προκύψει από την μη τήρηση περιορισμών των πεδίων κάθε οντότητα, όπως το ότι ένα πεδίο δεν πρέπει να είναι κενό κατά την υποβολή της φόρμας, με τη βοήθεια του οποίου ελέγχεται αν ο χρήστης έχει κάνει λάθη κατά τη συμπλήρωση της. Αν δεν έχει κάνει λάθη, μπορεί ο controller να προχωρήσει στην αποθήκευση της φόρμας με τη χρήση της αντίστοιχης μεθόδου και να κάνει redirection τον χρήστη σε μια σελίδα, ορίζοντας τις αντίστοιχες παραμέτρους, που χρειάζεται η σελίδα, για να εμφανίσει το μήνυμα επιτυχής υποβολής της φόρμας. Αν όμως έχει κάνει λάθη, με τη βοήθεια ενός object τύπου Model και ενός object τύπου της αντίστοιχης οντότητα, μαζί με τη χρήση των annotations «@Valid» και «@ModelAttribute», ο χρήστης οδηγείται στη φόρμα που συμπλήρωσε και τα πεδία όπου συμπλήρωσε σωστά, παραμένουν συμπληρωμένα, ενώ αυτά που συμπλήρωσε λάθος, περιλαμβάνουν το μήνυμα λάθους, που έχει οριστεί στο πεδίο της οντότητα μέσα στο entity.

```

75 @PostMapping("/chapter/new/{courseId}/{userId}")
76 @PreAuthorize("hasRole('ROLE_PROFESSOR')")
77 public String createChapter(
78     @PathVariable Long courseId,
79     @PathVariable Long userId,
80     @Valid @ModelAttribute("chapter") Chapter chapter,
81     BindingResult result,
82     Model model,
83     Principal principal
84 ){
85     String authUsername = "anonymousUser";
86     if (principal != null) {
87         authUsername = principal.getName();
88     }
89     Optional<User> optionalUser = userService.findUserById(userId);
90     if (optionalUser.isPresent()) {
91         User user = optionalUser.get();
92         if (user.getEmail().equals(authUsername)) {
93             Optional<Course> optionalCourse = courseService.findCourseById(courseId);
94             if (optionalCourse.isPresent()) {
95                 Course course = optionalCourse.get();
96                 if (course.getProfessor().getId().equals(userId)) {
97                     if (result.hasErrors()) {
98                         model.addAttribute("chapter", chapter);
99                         model.addAttribute("userId", userId);
100                        return "chapter/chapter_new";
101                    }
102                    chapterService.saveChapter(chapter);
103                    return String.format("redirect:/course/view/{courseId}/success_create_chapter", courseId, userId);
104                }
105                return "error/error_403";
106            }
107            return "error/error_404";
108        }
109        return "error/error_405";
110    }
111    }

```

Figure 79: Αποθήκευση νέου κεφαλαίου

Για την αντίστοιχη μεταφορά πληροφοριών από το back-end στο front-end, γίνεται χρήση ενός object τύπου Model, όπου μέσω της δημιουργίας attribute, μεταφέρει στη σελίδα όπου εξυπηρετεί, κάποιες πληροφορίες όπου χρειάζεται. Αυτές οι πληροφορίες, μπορεί να είναι ένα user id μόνο, μπορεί να είναι ένα ολόκληρο object ή μπορεί να είναι μια λίστα από object.

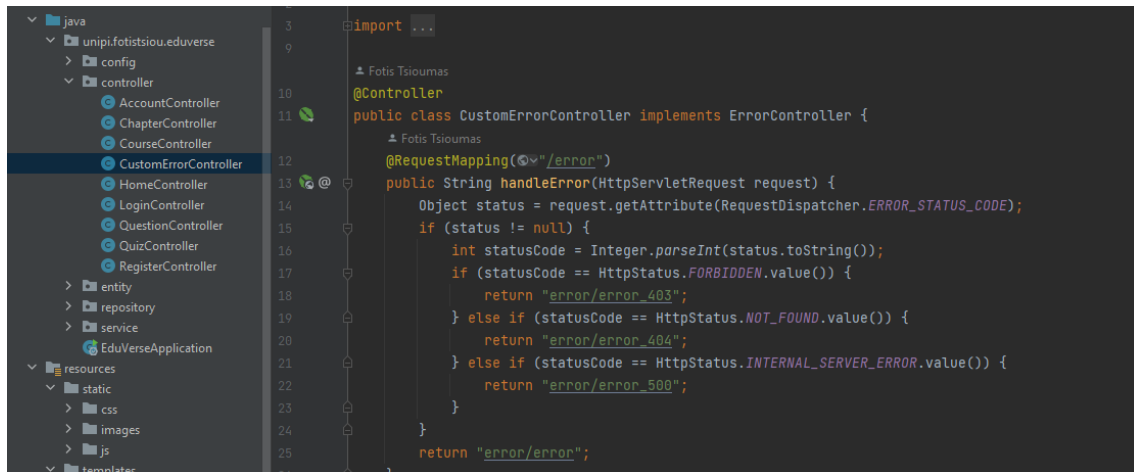
```

55 course.getStudents().contains(student)
56
57 ){
58     List<Result> results = resultService.findStudentCourseResults(courseId, studentId);
59     model.addAttribute("course", course);
60     model.addAttribute("results", results);
61     model.addAttribute("student", student);
62     model.addAttribute("userId", professorId);
63     return "course/course_student_view";
64 }
65 return "error/error_403";

```

Figure 80: Αποστολή πληροφοριών στη σελίδα

Τέλος υπάρχει ένα controller, όπου διαχειρίζεται τα συστημικά errors, που προαναφέρθηκαν παραπάνω και κάποια επιπλέον που αναφέρθηκαν στις ενότητες των σελίδων σφαλμάτων, κάνοντας implement το interface «ErrorController» και με τη βοήθεια μιας ρύθμισης στα application properties, όπου οδηγεί όλα τα errors την end point «/error». Στην μοναδική μέθοδο όπου περιλαμβάνει ο controller, δέχεται ως όρισμα ένα request τύπου HttpServletRequest, και στη συνέχεια μέσω κάποιων διαδικασιών που φαίνονται παρακάτω, ελέγχει τον τύπο σφάλματος και επιστρέφει το περιεχόμενο της αντίστοιχης custom σελίδας σφάλματος που έχει δημιουργηθεί για τη συγκεκριμένη εφαρμογή.



```

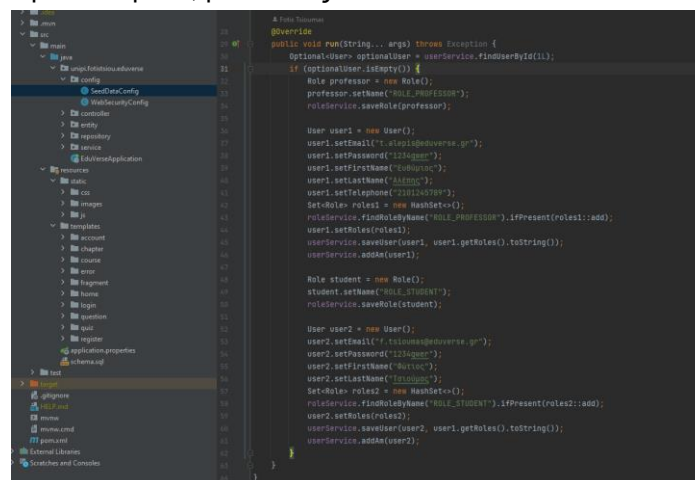
import ...
@Fotis Tsioumas
@Controller
public class CustomErrorController implements ErrorController {
    @Fotis Tsioumas
    @RequestMapping(value = "/error")
    public String handleError(HttpServletRequest request) {
        Object status = request.getAttribute(RequestDispatcher.ERROR_STATUS_CODE);
        if (status != null) {
            int statusCode = Integer.parseInt(status.toString());
            if (statusCode == HttpStatus.FORBIDDEN.value()) {
                return "error/error_403";
            } else if (statusCode == HttpStatus.NOT_FOUND.value()) {
                return "error/error_404";
            } else if (statusCode == HttpStatus.INTERNAL_SERVER_ERROR.value()) {
                return "error/error_500";
            }
        }
        return "error/error";
    }
}

```

Figure 81: Διαχείριση συστημικών σφαλμάτων

Configs

Ο φάκελος configs περιλαμβάνει δύο αρχεία. Το πρώτο περιλαμβάνει την κλάση SeedDataConfig και το δεύτερο την κλάση WebSecurityConfig. Ο ρόλος της πρώτης config κλάσης, είναι αυτός που λέει και το όνομα της, να προσθέσει κάποια αρχικά δεδομένα. Για να συμβεί αυτό, η κλάση SeedDataConfig κάνει implement το interface CommandLineRunner και στη συνέχεια αφού αποκτήσει πρόσβαση στα services που χρειάζεται, μέσω dependency injection, κάνει override την μέθοδο «run». Η υλοποίηση της συγκεκριμένης μεθόδου περιλαμβάνει τη δημιουργία δύο object τύπου Role και τη δημιουργία δύο object τύπου User, όπου τους ανατίθενται τα δύο ρόλους που δημιουργήθηκαν. Η λειτουργία της συγκεκριμένης κλάσης, είναι στην ουσία η προσθήκη των δύο ρόλων όπου χρησιμοποιούνται σε όλο το όλη την υπόλοιπη εφαρμογή και η δημιουργία των πρώτων users, ένα για κάθε ρόλο, μέσω ενός action τύπου command line.



```

@Override
public void run(String... args) throws Exception {
    Optional<User> optionalUser = userService.findById(1L);
    if (optionalUser.isEmpty()) {
        Role professor = new Role();
        professor.setName("ROLE_PROFESSOR");
        roleService.saveRole(professor);

        User user1 = new User();
        user1.setEmail("f.tsioumas@eduverse.gr");
        user1.setPassword("12345678");
        user1.setFirstName("Fotis");
        user1.setLastName("Tsioumas");
        user1.setRoleName("ROLE_PROFESSOR");
        Set<Role> roles1 = new HashSet<>();
        roleService.findRoleByName("ROLE_PROFESSOR").ifPresent(roles1::add);
        user1.setRoles(roles1);
        userService.saveUser(user1, user1.getRoles().toString());
        userService.addUser(user1);

        Role student = new Role();
        student.setName("ROLE_STUDENT");
        roleService.saveRole(student);

        User user2 = new User();
        user2.setEmail("f.tsioumas@eduverse.gr");
        user2.setPassword("12345678");
        user2.setFirstName("Fotis");
        user2.setLastName("Tsioumas");
        Set<Role> roles2 = new HashSet<>();
        roleService.findRoleByName("ROLE_STUDENT").ifPresent(roles2::add);
        user2.setRoles(roles2);
        userService.saveUser(user2, user1.getRoles().toString());
        userService.addUser(user2);
    }
}

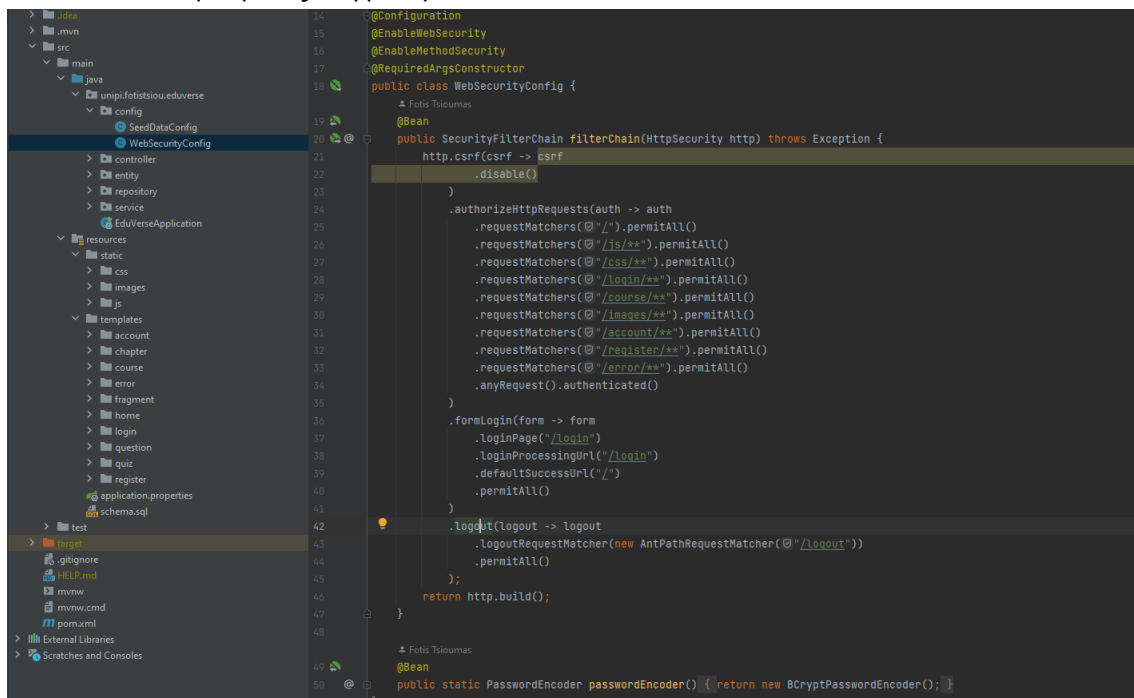
```

Figure 82: Μέθοδος «run»

Ο ρόλος της δεύτερης config κλάσης, είναι πολύ σημαντικός και σχετίζεται με τη διαμόρφωση των ρυθμίσεων ασφαλείας της εφαρμογής, όσον αφορά την ασφάλεια ιστού. Με τη χρήση των annotations «@EnableWebSecurity» και «@EnableMethodSecurity», ενεργοποιείται το Spring Security. Στο body της κλάσης παράγονται τα beans SecurityFilterChain και PasswordEncoder, ώστε να τεθούν στη διάθεση του Spring container. Στη μέθοδο filterChain γίνονται οι εξής ρυθμίσεις:

- Απενεργοποιείται η προστασία CSRF.
- Δηλώνονται τα urls/end points της εφαρμογής.
- Ορίζεται η φόρμα εισόδου του εγγεγραμμένου χρήστη.
- Διαμορφώνεται η λειτουργία αποσύνδεσης με τον καθορισμό του url αποσύνδεσης.

Η μέθοδος passwordEncoder επιστρέφει ένα anonymous object της κλάσης BCryptPasswordEncoder, με τη βοήθεια της οποία κωδικοποιούνται ή αποκωδικοποιούνται οι κωδικοί που ορίζουν οι χρήστες. Η περίπτωση της κωδικοποίησης εμφανίζεται όταν χρειάζεται να αποθηκευτεί ο κωδικός του χρήστη στη βάση, ενώ η περίπτωση της αποκωδικοποίησης εμφανίζεται όταν ο χρήστης προσπαθήσει να αλλάξει κωδικό πρόσβασης και πρέπει να γίνει σύγκριση του input που δίνει στο πεδίο «παλιός κωδικός πρόσβαση» και του κωδικού πρόσβασης που είναι αποθηκευμένος στη βάση.



```

14 @Configuration
15 @EnableWebSecurity
16 @EnableMethodSecurity
17 @RequiredArgsConstructor
18 public class WebSecurityConfig {
19     @Bean
20     public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
21         http.csrf(csrf -> csrf
22             .disable()
23         )
24         .authorizeHttpRequests(auth -> auth
25             .requestMatchers("@/*").permitAll()
26             .requestMatchers("@/*js/*").permitAll()
27             .requestMatchers("@/*css/*").permitAll()
28             .requestMatchers("@/*login/*").permitAll()
29             .requestMatchers("@/*course/*").permitAll()
30             .requestMatchers("@/*images/*").permitAll()
31             .requestMatchers("@/*account/*").permitAll()
32             .requestMatchers("@/*register/*").permitAll()
33             .requestMatchers("@/*enpon/*").permitAll()
34             .anyRequest().authenticated()
35         )
36         .formLogin(form -> form
37             .loginPage("/login")
38             .loginProcessingUrl("/login")
39             .defaultSuccessUrl("/")
40             .permitAll()
41         )
42         .logout(logout -> logout
43             .logoutRequestMatcher(new AntPathRequestMatcher("/logout"))
44             .permitAll()
45         );
46         return http.build();
47     }
48 }
49 @Bean
50 public static PasswordEncoder passwordEncoder() { return new BCryptPasswordEncoder(); }
51 }

```

Figure 83: Κλάση «WebSecurityConfig»

Templates

Τα templates της εφαρμογής, αποτελούνται από html αρχεία, όπου περιλαμβάνουν Thymeleaf κώδικα, ο οποίος βοηθάει στη διαχείριση των objects και των πληροφοριών τους, που στέλνει το back-end στο front-end και το αντίστροφο, μέσω των controllers. Το Thymeleaf είναι ένα σύγχρονο Java template engine, όπου έχει σχεδιαστεί για web και αυτόνομα περιβάλλοντα και έχει τη δυνατότητα να επεξεργάζεται HTML, XML, JavaScript, CSS, ακόμη και απλό κείμενο (Thymeleaf.org, 2018).

Κάθε html αρχείο αποτελεί και από μία σελίδα της πλατφόρμας. Τα αρχεία έχουν χωριστεί σε φακέλους και για κάθε φάκελο υπάρχει ένας controller, ο οποίος φέρει το αντίστοιχο όνομα και εξυπηρετεί τα http request των σελίδων του φακέλου. Έτσι για παράδειγμα, οι σελίδες του φακέλου «account», επικοινωνούν με τον server μέσω του controller «AccountController», ενώ οι

σελίδες του φακέλου «register», επικοινωνούν με τον server μέσω του controller «RegisterController» και ούτω καθεξής.

Οι σελίδες από τη μεριά του χρήστη, αναλύθηκαν στο κεφάλαιο «Παρουσίαση και Χρήση Εφαρμογής», οπότε στη συνέχεια ακολουθεί ανάλυση της δομής τους από μεριάς κώδικα. Από μεριάς χρήστη, ενώ φαίνεται ότι κάθε σελίδα αποτελεί μια ξεχωριστή επιφάνεια, με τα δικά της δομικά στοιχεία, όπως το navigation bar, το content και το footer, από μεριάς κώδικα δεν ισχύει αυτό. Το footer, όπως σημειώθηκε και σε εκείνο το κεφάλαιο, επαναλαμβάνεται ίδιο σε όλες τις σελίδες και για όλους του users, ανεξαρτήτως ρόλου. Έτσι, δημιουργήθηκε ένα template με το όνομα «footer.html», το εξωτερικό <div> element του οποίου περιέχει το Thymeleaf attribute «th:fragment="footer"» και στη συνέχεια το συγκεκριμένο template, προστέθηκε στο τέλος κάθε σελίδας με τη χρήση ενός <nav> element, που προσφέρει το Thymeleaf attribute «th:replace="~{fragment/footer :: footer}"». Αυτός ο μηχανισμός είχε σαν αποτέλεσμα, το footer template να είναι διαχειρίσιμο από ένα μόνο σημείο και να μην επαναλαμβάνεται ο ίδιο κώδικας σε κάθε σελίδα. Παρόμοιος μηχανισμός έχει δημιουργηθεί και για το navigation bar, όπου διαθέτει το template «nav.html» και προστίθεται στην αρχή των templates των σελίδων, με εξαίρεση τις σελίδες που δεν διαθέτουν κεντρικές κατηγορίες πλοήγησης, παρά μόνο τα κουμπιά στα άκρα, όπως συμβαίνει με την αρχική σελίδα ή τις σελίδες σφαλμάτων, στις οποίες το navigation bar έχει συμπεριληφθεί στο template της σελίδας αυτούσιο.

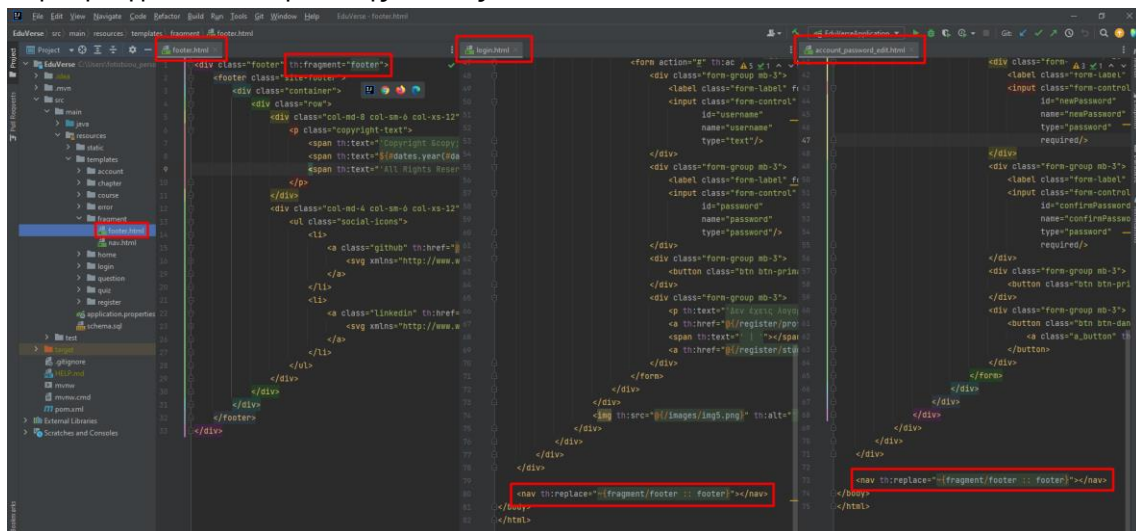


Figure 84: Attributes «th:fragment» και «th:replace»

Από εκεί και πέρα, το content κάθε σελίδα αποτελεί ξεχωριστό στοιχείο για κάθε μία και αυτό αποτυπώνεται και στον κώδικα, αφού έχει δημιουργηθεί ένα ξεχωριστό template για κάθε σελίδα. Ωστόσο, ανατρέχοντας ξανά στο κεφάλαιο που αναλύθηκαν οι σελίδες από τη μεριά του χρήστη, αναφέρεται ότι σε αρκετές σελίδες έχουν πρόσβαση πάνω από μια κατηγορία χρηστών, αλλά το περιεχόμενο φαίνεται διαφορετικό για κάθε κατηγορία. Αυτό επιτυγχάνεται με τη βοήθεια τεσσάρων Thymeleaf attributes, τα οποία προστίθενται σε elements των templates. Το «sec:authorize="!isAuthenticated()"», όπου κάνει το element ορατό μόνο σε μη συνδεδεμένους χρήστες, το «sec:authorize="isAuthenticated()"», όπου κάνει το element ορατό μόνο σε συνδεδεμένους χρήστες, το «sec:authorize="hasRole('ROLE_PROFESSOR')"», όπου κάνει το element ορατό μόνο σε «καθηγητές» και το «sec:authorize="hasRole('ROLE_STUDENT')"», όπου κάνει το element ορατό μόνο σε «φοιτητές». Χαρακτηριστική σελίδα που συμβαίνει αυτό είναι η αρχική σελίδα, όπου ο πίνακας με τα buttons περιλαμβάνει άλλα buttons στην περίπτωση του μη συνδεδεμένου χρήστη, άλλα στην περίπτωση του συνδεδεμένου χρήστη και υπάρχει διαφοροποίηση και στις περιπτώσεις που ο χρήστης είναι «καθηγητής» ή «φοιτητής». Στο συγκεκριμένο template έχουν δημιουργηθεί δύο ξεχωριστά <div>, ένα για την περίπτωση του μη συνδεδεμένου χρήστη, όπου περιλαμβάνει το attribute «sec:authorize="!isAuthenticated()"» και ένα για την περίπτωση του συνδεδεμένου χρήστη που περιλαμβάνει το attribute «sec:authorize="isAuthenticated()"». Στην περίπτωση του συνδεδεμένου χρήστη υπάρχουν buttons, όπου δεν περιλαμβάνουν κανένα attribute και είναι ορατά σε όλους τους συνδεδεμένους χρήστες,

Για να μετατραπεί τελικά το πεδίο «περιεχόμενο» από απλό textarea σε rich text editor, γίνεται χρήση του παρακάτω script, το οποίο είναι τοποθετημένο στο τέλος του body.

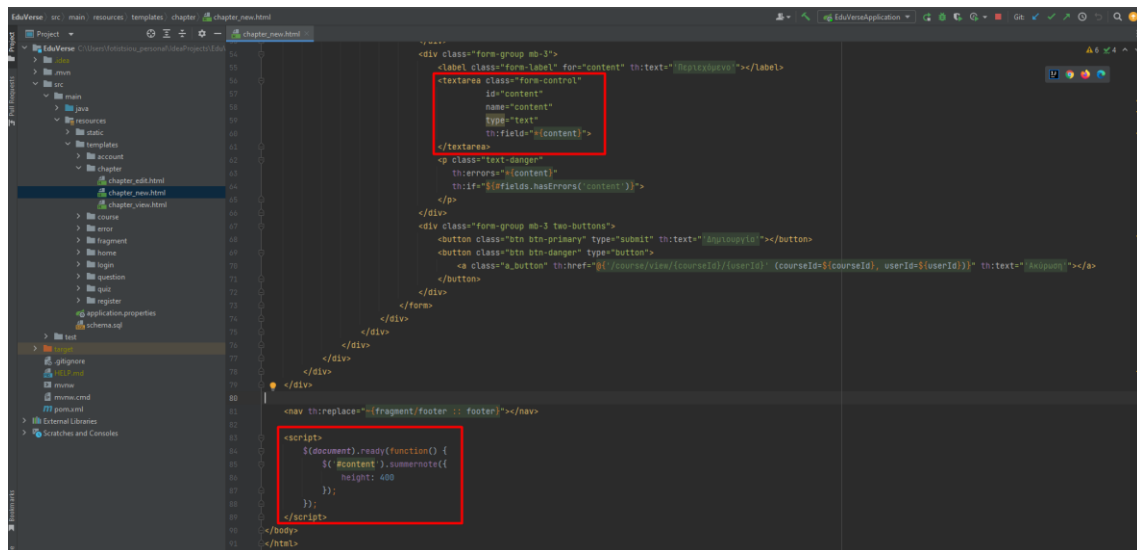


Figure 89: Προσθήκη λειτουργικότητας στο textarea πεδίο

Ένα ακόμα ενδιαφέρον σημείο, είναι ο τρόπος που έχει δομηθεί η φόρμα της υποβολής ερωτηματολογίων. Αποτελεί τη μόνη φόρμα, όπου για να δημιουργηθούν τα πεδία της, γίνεται ένα iteration στην μεταβλητή που έχει στείλει ο controller στο template, η οποία αποτελεί μια λίστα από ερωτήσεις. Ο αριθμός των ερωτήσεων της λίστας είναι δυναμικός και εξαρτάται από τις ερωτήσεις που έχει δημιουργήσει ο καθηγητής για κάθε κεφάλαιο. Έτσι, χρησιμοποιείται ένα εξωτερικό <div>, στο οποίο έχει προστεθεί το Thymeleaf attribute «th:each», το οποίο πραγματοποιεί το iteration της λίστας των ερωτήσεων και σε κάθε πεδίο της φόρμας, τα attributes «th:id», «th:name» και «th:for» παίρνουν τιμές με βάση τον αύξοντα αριθμό της τρέχουσας επανάληψης.

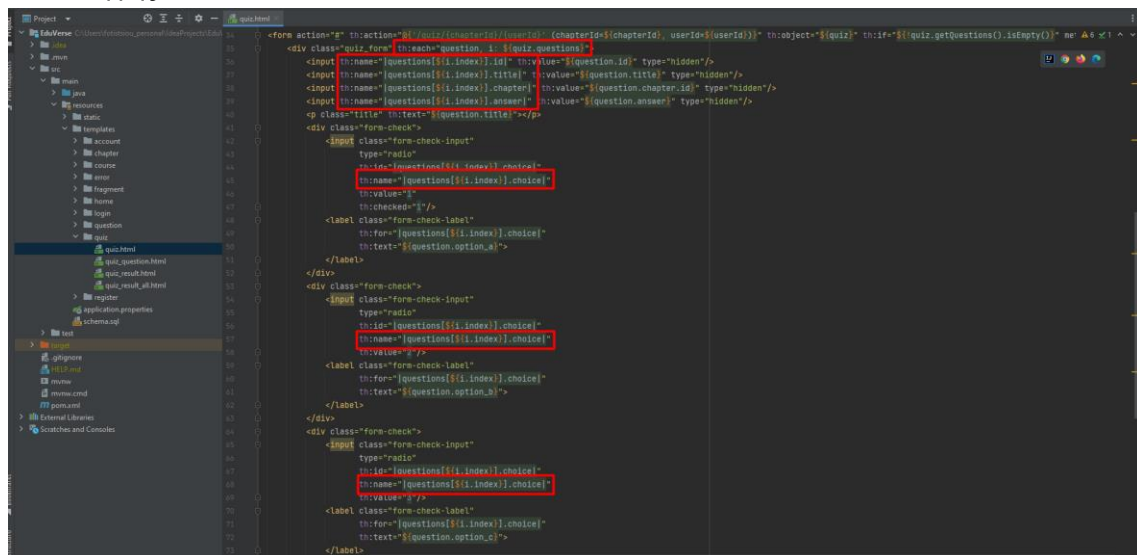


Figure 90: Δομή φόρμας ερωτηματολογίου

Με την παραπάνω δομή, επιτυγχάνεται η αποστολή των δεδομένων τους ερωτηματολογίου, σε κατάλληλη μορφή, ώστε να επεξεργαστούν από το back-end και να αποθηκευτούν ως μια εγγραφή στον πίνακα result και ως ένα σύνολο εγγραφών, όσες και ο αριθμός των ερωτήσεων, στον πίνακα quiz_question. Η μορφή που στέλνονται τα δεδομένα, μετά την υποβολή ενός ερωτηματολογίου, φαίνεται στην παρακάτω εικόνα. Το συγκεκριμένο ερωτηματολόγιο, αποτελείται από δύο ερωτήσεις. Για κάθε ερώτηση, στέλνονται τα στοιχεία id, title, chapter,

answer και choice, τα οποία αποθηκεύονται σε μία λίστα από objects, ο αριθμός των οποίων είναι ίσος με τον αριθμό των ερωτήσεων.

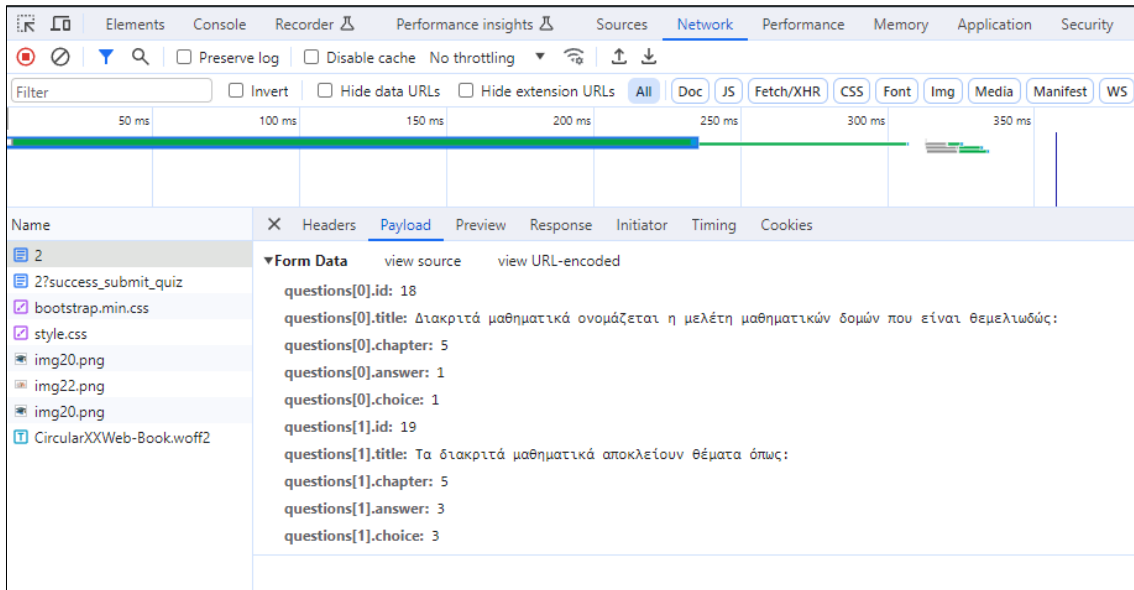


Figure 91: Μορφή δεδομένων υποβολής ερωτηματολογίου

Τέλος, αξίζει να σημειωθεί ο τρόπος με τον οποίο εμφανίζονται τα μηνύματα επιτυχής ή ανεπιτυχής προσπάθεια, μετά από μια ενέργεια του χρήστη, στο πάνω μέρος της οθόνης, όπως για παράδειγμα στην περίπτωση της σελίδα «Φόρμα Σύνδεσης». Στις περιπτώσεις που η ενέργεια του χρήστη, πραγματοποιεί μια αλλαγή στη βάση ή αποστέλλει ένα αίτημα που απαιτεί επιβεβαίωση, όπως για παράδειγμα μετά την προσπάθεια σύνδεσης ενός χρήστη στην εφαρμογή, ο controller που χειρίζεται το αίτημα, ανακατευθύνει τον χρήστη σε μία σελίδα και ανάλογα με το αποτέλεσμα του αιτήματος, προσθέτει μία παράμετρο στο τέλος του url της σελίδας, όπου μεταφέρθηκε ο χρήστης. Το template της σελίδας με τη σειρά του, διαχειρίζεται το σύνολο των πιθανών παραμέτρων, που μπορεί να πάρει το url της σελίδας και μόλις προστεθεί κάποια από αυτές τις παραμέτρους στο url, εμφανίζει ένα μήνυμα μέσα σε ένα πλαίσιο. Το χρώμα του πλαισίου και το περιεχόμενό του μηνύματος, δηλώνουν το αποτέλεσμα του αιτήματος που πραγματοποιήσε ο χρήστης. Ο τρόπος που χειρίζεται το σύνολο των παραμέτρων το template φαίνεται στην παρακάτω εικόνα. Στο συγκεκριμένο παράδειγμα οι παράμετροι είναι τρεις, «error», «logout» και «success_register» και προστίθενται με το σύμβολο του question mark στην αρχή τους, στο τέλος του url (π.χ. /login?**logout**).

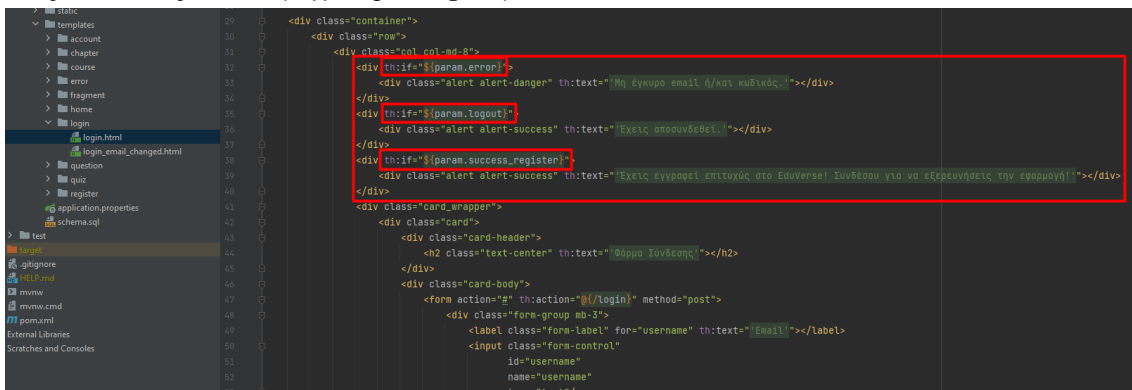


Figure 92: Διαχείριση παραμέτρων του url της σελίδας

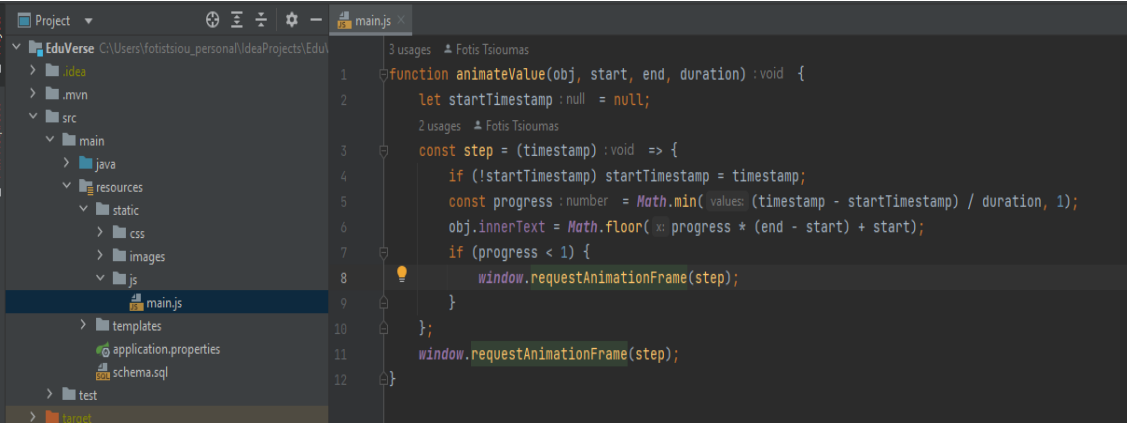
Static Files

Στον φάκελο `resource` του `project`, εκτός από τα `templates` υπάρχει ακόμα ένα φάκελος με το όνομα `static`. Στον συγκεκριμένο φάκελο έχουν τοποθετηθεί η τοπική `css (/css/style.css)`, οι εικόνες της εφαρμογής (`/images/...`) και ένα `javascript` αρχείο (`/js/main.js`).

Το αρχείο `style.css` φορτώνεται στο `head` όλων των `templates` και λειτουργεί συμπληρωματικά με την `bootstrap`, η οποία είναι η βασική εξωτερική βιβλιοθήκη των `templates`, με εξαίρεση τις σελίδες της δημιουργίας και επεξεργασίας κεφαλαίων, στις οποίες χρησιμοποιούνται και άλλες εξωτερικές βιβλιοθήκες για μορφοποίηση, οι οποίες αναλύθηκαν σε προηγούμενη ενότητα.

Οι εικόνες είναι τοποθετημένες στο `body` των `templates` και προστίθενται σε αυτά με τη χρήση του `` element, με την μορφή `" />`. Επίσης, στις εικόνες συμπεριλαμβάνεται και το `favicon` του `site`, το οποίο εμφανίζεται στο `tab` της κάθε σελίδας και βρίσκεται σε αντίθεση με τις υπόλοιπες εικόνες, στο `head` των `templates` στο οποίο έχει προστεθεί με τη χρήση του `<link rel="icon" type="image/png" th:href="@{/images/favicon.png}" />`.

Τέλος, το πιο αξιοσημείωτο κομμάτι του συγκεκριμένου φακέλου είναι η `javascript` που έχει μία μόνο λειτουργία. Να δημιουργεί το εφέ της μέτρησης των μαθημάτων, των καθηγητών και των φοιτητών στην αρχική σελίδα όλων των χρηστών. Όπως αναφέρθηκε και στην ανάλυση της αρχικής σελίδας, ο αριθμός που καταλήγει ο μετρητής κάθε οντότητας από τις παραπάνω, είναι δυναμικός και αποτυπώνει το σύνολο των μαθημάτων που έχουν δημιουργηθεί στην εφαρμογή, το σύνολο των εγγεγραμμένων καθηγητών της εφαρμογής και το σύνολο των εγγεγραμμένων φοιτητών της εφαρμογής. Για την επίτευξη του `animation` της μέτρησης μέχρι τον τελικό αριθμό των συνόλων, όπου και σταματάει η λειτουργία του `animation`, έχει χρησιμοποιηθεί συνάρτηση της παρακάτω εικόνας. Η συγκεκριμένη συνάρτηση δέχεται τέσσερα ορίσματα, το πρώτο δέχεται το `html element` που θα λειτουργήσει ως `animation/μετρητής`, το δεύτερο είναι η αρχική τιμή από την οποία θα ξεκινήσει η μέτρηση, το τρίτο είναι η τελική τιμή στην οποία θα καταλήξει η μέτρηση και το τελευταίο είναι η διάρκεια που θα κρατήσει το `animation` της μέτρησης για να φτάσει από την αρχική τιμή στην τελική. Στη συνέχεια, η συνάρτηση χρησιμοποιεί μια μαθηματική πράξη, για να υπολογίζει κάθε φορά μια τιμή, η οποία στρογγυλοποιείται και ενημερώνει το `text` του `element` που έχει εισαχθεί ως πρώτο όρισμα. Ο τρόπος με τον οποίο ενημερώνεται το `text` του `element` κάθε φορά είναι ιδιαίτερα αποδοτικός, αφού γίνεται με τη χρήση της μεθόδου `«requestAnimationFrame»` του `object «window»`.



```

1 function animateValue(obj, start, end, duration) :void {
2   let startTimestamp : null = null;
3   const step = (timestamp) :void => {
4     if (!startTimestamp) startTimestamp = timestamp;
5     const progress :number = Math.min( values (timestamp - startTimestamp) / duration, 1);
6     obj.innerText = Math.floor( x progress * (end - start) + start);
7     if (progress < 1) {
8       window.requestAnimationFrame(step);
9     }
10  };
11  window.requestAnimationFrame(step);
12 }

```

Figure 93: Script για τη δημιουργία animation/μετρητή

Η συγκεκριμένη συνάρτηση καλείται στην αρχική σελίδα για κάθε ένα από τα `element` μετρητή και η τελική τιμή όπου τροφοδοτείται η συνάρτηση για κάθε μετρητή έχει δοθεί από τον `controller` στο `template` και είναι δυναμική, ενώ η αρχική τιμή είναι πάντα το 0 και η διάρκεια τα 1000msec.


```

91 <p class="text-center" th:text="Εμφαδεντικά παθήματα, μαγιστρός, άσπασκός, για καθήματα και παθήματα.">
92 </p>
93
94 <div class="counters">
95   <div class="counter-wrapper">
96     <span id="course_number" class="counter" th:text="0"></span>
97     <span class="text" th:text="Μαθήματα"></span>
98   </div>
99   <div class="counter-wrapper">
100     <span id="professor_number" class="counter" th:text="0"></span>
101     <span class="text" th:text="Καθηγητές"></span>
102   </div>
103   <div class="counter-wrapper">
104     <span id="students_number" class="counter" th:text="0"></span>
105     <span class="text" th:text="Μαθητές"></span>
106   </div>
107 </div>
108
109 </div>
110
111 <nav th:replace="<fragment/footer :: footer"></nav>
112
113
114 <script th:inline="javascript">
115   let course_number = document.getElementById("course_number");
116   let professor_number = document.getElementById("professor_number");
117   let students_number = document.getElementById("students_number");
118   let start = 0;
119   let numberOfCourses = [[${numberOfCourses}]];
120   let numberOfProfessors = [[${numberOfProfessors}]];
121   let numberOfStudents = [[${numberOfStudents}]];
122   let duration = 1000;
123   window.addEventListener("load", () => animateValue(course_number, start, numberOfCourses, duration));
124   window.addEventListener("load", () => animateValue(professor_number, start, numberOfProfessors, duration));
125   window.addEventListener("load", () => animateValue(students_number, start, numberOfStudents, duration));
126 </script>
127
128 </body>
129 </html>

```

Figure 94: Χρήση animation μέτρησης στην αρχική σελίδα

Application Properties

Το τελευταίο μέρος που αφορά τη δομή της εφαρμογής, είναι τα application properties. Βρίσκονται και αυτά στον φάκελο resource και όπως φαίνεται και από την παρακάτω εικόνα, αφορούν κάποιες ρυθμίσεις του server, όπως στο ποια πόρτα ακούει και σε ποιο path να επιστρέφει τα server errors και κάποιες ρυθμίσεις της σύνδεσης με την βάση δεδομένων.

```

1 # Server settings properties
2 server.port=8080
3 server.error.include-message=always
4 server.error.whitelabel.enabled=false
5 server.error.path=/error
6
7 # Database connection properties
8 spring.datasource.url=jdbc:mysql://127.0.0.1:3306/eduverse
9 spring.datasource.username=root
10 spring.datasource.password=2410
11 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
12 spring.jpa.hibernate.ddl-auto=update

```

Figure 95: Application Properties

Δομή Βάσης Δεδομένων

Η βάση δεδομένων αποτελείται από εννιά πίνακες, έναν για κάθε entity και έναν για κάθε σχέση πολλά προς πολλά μεταξύ των entities course-user (πίνακας course_students) και user-role (πίνακας user_roles). Όλοι οι πίνακες, τα πεδία τους, οι ιδιότητες των πεδίων τους και οι συσχετίσεις μεταξύ τους, παράγονται αυτόματα και προέρχονται από τις οντότητες, που αναλύθηκαν στην αντίστοιχη ενότητα.

Ακολουθεί δύο εικόνες με τα πεδία των πινάκων και τις ιδιότητές τους. Η τελευταία στήλη φανερώνει την προέλευση του πίνακα και την προέλευση των foreign keys του.

Table	Field	Type	AI	PK	FK	Unique	Not Null	Origin
user	id	bigint	NAI	NAI	OXI	OXI	OXI	From User Entity
	am	varchar(255)	OXI	OXI	OXI	OXI	NAI	
	email	varchar(255)	OXI	OXI	OXI	NAI	NAI	
	first_name	varchar(255)	OXI	OXI	OXI	OXI	NAI	
	last_name	varchar(255)	OXI	OXI	OXI	OXI	NAI	
	password	varchar(255)	OXI	OXI	OXI	OXI	NAI	
telephone	varchar(255)	OXI	OXI	OXI	OXI	OXI		
role	id	bigint	NAI	NAI	OXI	OXI	OXI	From Role Entity
	name	varchar(255)	OXI	OXI	OXI	NAI	NAI	
user_roles	user_id	bigint	OXI	NAI	NAI	OXI	NAI	From ManyToMany Relation (user - role)
	role_id	bigint	OXI	NAI	NAI	OXI	NAI	
course	id	bigint	NAI	NAI	OXI	OXI	OXI	From Course Entity
	code	varchar(255)	OXI	OXI	OXI	NAI	NAI	
	title	varchar(255)	OXI	OXI	OXI	OXI	NAI	
	description	text	OXI	OXI	OXI	OXI	NAI	
	professor	bigint	OXI	OXI	NAI	OXI	NAI	From ManyToOne Relation (course - user)
course_students	course_id	bigint	OXI	NAI	NAI	OXI	NAI	From ManyToMany Relation (course - user)
	student	bigint	OXI	NAI	NAI	OXI	NAI	
chapter	id	bigint	NAI	NAI	OXI	OXI	OXI	From Chapter Entity
	title	varchar(255)	OXI	OXI	OXI	OXI	NAI	
	content	text	OXI	OXI	OXI	OXI	NAI	
	course	bigint	OXI	OXI	NAI	OXI	NAI	

Figure 96: Δομή Βάσης Δεδομένων 1

Table	Field	Type	AI	PK	FK	Unique	Not Null	Origin	
question	id	bigint	NAI	NAI	OXI	OXI	OXI	From Question Entity	
	title	text	OXI	OXI	OXI	OXI	NAI		
	option_a	varchar(255)	OXI	OXI	OXI	OXI	NAI		
	option_b	varchar(255)	OXI	OXI	OXI	OXI	NAI		
	option_c	varchar(255)	OXI	OXI	OXI	OXI	NAI		
	answer	int	OXI	OXI	OXI	OXI	NAI		
	choice	int	OXI	OXI	OXI	OXI	NAI		
	chapter	bigint	OXI	OXI	NAI	OXI	NAI	From ManyToOne Relation (question - chapter)	
result	id	bigint	NAI	NAI	OXI	OXI	OXI	From Result Entity	
	correct	int	OXI	OXI	OXI	OXI	NAI		
	wrong	int	OXI	OXI	OXI	OXI	NAI		
	feedback	varchar(255)	OXI	OXI	OXI	OXI	NAI		
	chapter	bigint	OXI	OXI	NAI	OXI	NAI		From ManyToOne Relation (result - chapter)
	student	bigint	OXI	OXI	NAI	OXI	NAI	From ManyToOne Relation (result - user)	
quiz_question	id	bigint	NAI	NAI	OXI	OXI	OXI	From QuizQuestion Entity	
	title	text	OXI	OXI	OXI	OXI	NAI		
	answer	int	OXI	OXI	OXI	OXI	NAI		
	choice	int	OXI	OXI	OXI	OXI	NAI		
	chapter	bigint	OXI	OXI	NAI	OXI	NAI		From ManyToOne Relation (quiz_question- chapter)
	result	bigint	OXI	OXI	NAI	OXI	NAI		From ManyToOne Relation (quiz_question- result)
	student	bigint	OXI	OXI	NAI	OXI	NAI		From ManyToOne Relation (quiz_question- user)

Figure 97: Δομή Βάσης Δεδομένων 2

Συμπεράσματα και μελλοντικές επεκτάσεις

Ο τομέας της Εκπαίδευσης είναι ένας από τους βασικότερους πυλώνες της κοινωνίας μας, καθώς αποτελεί το πιο σημαντικό μέσο για ριζικές αλλαγές απέναντι σε μεγάλες κρίσεις, όπως αυτές που αντιμετωπίζουμε ως ανθρωπότητα τα τελευταία χρόνια. Επιπλέον, ο τομέας της Πληροφορικής, είναι ο τομέας με τη μεγαλύτερη ανάπτυξη τα τελευταία χρόνια, καθώς βρίσκει εφαρμογή σε πάρα πολλούς άλλους τομείς, ένας εκ των οποίων είναι και η Εκπαίδευση. Συνεπώς, ένα από τα πεδία της Εκπαίδευσης που μπορεί να βρει εφαρμογή η Πληροφορική και να παίξει καθοριστικό ρόλο, είναι η ανάπτυξη ψηφιακών εκπαιδευτικών πλατφορμών, καθώς τέτοιου είδους εφαρμογές καθιστούν την εκπαιδευτική διαδικασία δυνατή κάτω και από τις πιο αντίξοες συνθήκες, όπως αυτές που προκύπτουν από μια πανδημία, από την μεγάλη απόσταση από φορείς εκπαίδευσης, ακόμα και από τον πόλεμο.

Με βάση την ανάλυση της συγκεκριμένης υλοποίησης, φαίνεται το ότι πρόκειται για μια πλατφόρμα με τις πολύ βασικές λειτουργίες της εκπαιδευτικής διαδικασίας, σε βάθος όμως υλοποιημένες. Οι μελλοντικές επεκτάσεις που μπορεί να υποστηρίξει και θα μπορούσαν να προτεθούν στο μέλλον είναι πολλές και μερικές από αυτές είναι οι παρακάτω:

- Δυνατότητα συμμετοχής περισσότερης του ενός καθηγητή, ανά μάθημα.
- Δημιουργία forum για αλληλεπίδραση μεταξύ των χρηστών, ανά μάθημα.
- Δημιουργία μηχανισμού αξιολόγησης των καθηγητών και των μαθημάτων, από τους φοιτητές.
- Δημιουργία chat για άμεση επικοινωνία σε προσωπικό επίπεδο, μεταξύ χρηστών της εφαρμογής.
- Δημιουργία chat bot για επίλυση απλών τεχνικών ζητημάτων ή συχνών ερωτήσεων ως προς τη λειτουργία της πλατφόρμας και επικοινωνία με τον διαχειριστή για πιο σύνθετα ζητήματα.

Βιβλιογραφία

- Brainy. (2023). *Brainy*. Ανάκτηση από <https://brainy.gr>
- e-me. (2023). *e-me*. Ανάκτηση από <https://e-me.edu.gr>
- FRA. (2007, Δεκέμβριος 14). *FRA - European Union Agency for Fundamental Rights*. Ανάκτηση από <https://fra.europa.eu/el/eu-charter/article/14-dikaioma-ekpaideysis>
- GUNet2. (2023). *GUNet2 eClass - Τμήμα Πληροφορικής (Πανεπιστήμιο Πειραιώς)*. Ανάκτηση από <https://gunet2.cs.unipi.gr/info/about.php>
- IBM. (2023). *IBM*. Ανάκτηση από <https://www.ibm.com/topics/java-spring-boot>
- JetBrains. (2023, Οκτώβριος 11). *Jetbrains - IntelliJ IDEA*. Ανάκτηση από <https://www.jetbrains.com/help/idea/spring-boot.html>
- Open-eClass. (2023). *Open eclass e-learning platform*. Ανάκτηση από <https://www.openeclass.org>
- Spring-io. (2022, Νοέμβριος 10). *Initializr*. Ανάκτηση από <https://github.com/spring-io/initializr/blob/main/README.adoc>
- Spring-Projects. (2023, Ιουνίου 8). *Spring Boot*. Ανάκτηση από <https://github.com/spring-projects/spring-boot/blob/main/README.adoc>
- Thymeleaf.org. (2018, Οκτώβριος 29). *Thymeleaf - Tutorial: Using Thymeleaf*. Ανάκτηση από <https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html>