ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πτυχιακή Εργασία

Τίτλος Πτυχιακής Εργασίας	Σχεδίαση και αναπτυξη λογισμικού διαδικτυακής διαμοίρασης αρχείων με κεντρικό εξυπηρετητή. Design and develop server based file sharing software
Ονοματεπώνυμο Φοιτητή	Ιωάννης Τζαμπαζάκης
Πατρώνυμο	Αντώνιος Τζαμπαζάκης
Αριθμός Μητρώου	Π17140
Επιβλέπων	Σακκόπουλος Ευάγγελος, Αναπληρωτής Καθηγητής

Ημερομηνία Παράδοσης Σεπτέμβριος 2024

Copyright

Ευχαριστίες

Εκφράζω τις θερμές μου ευχαριστίες στον επιβλέποντα καθηγητή μου Δρ. Σακκόπουλο Ευάγγελο, για την επιστημονική καθοδήγηση που μου παρείχε, την αγαστή συνεργασία και την ευκαιρία που μου έδωσε να υλοποιήσω την παρούσα πτυχιακή στο αντικείμενο που επιθυμούσα.

Ευχαριστώ και όλους ανεξαιρέτως τους διδάσκοντες του Τμήματος Πληροφορικής του Πανεπιστημίου Πειραιώς, για τη συμβολή τους στην επιστημονική μου συγκρότηση και το όμορφο ταξίδι γνώσης που μου προσέφεραν σε όλο το διάστημα των σπουδών μου.

Τέλος, ένα μεγάλο ευχαριστώ στην οικογένειά μου καθώς και σε όλους τους δικούς μου ανθρώπους για τη στήριξη που μου προσέφεραν.

Πίνακας περιεχομένων

Περίληψη

Σκοπός και αντικείμενο της παρούσας πτυχιακής εργασίας συνιστά η εξατομικευμένη διαμοίραση αρχείων με ασφάλεια, καθώς κρίνεται αναγκαία για την προστασία ευαίσθητων προσωπικών δεδομένων, ενώ ταυτόχρονα συνεισφέρει στην ευέλικτη πρόσβαση και συνεργασία σε διαφορετικά περιβάλλοντα. Στόχος, ωστόσο, καθίσταται η ανάπτυξη και αξιολόγηση μιας συγκεκριμένης πρότασης που να επιτρέπει την ασφαλή διαχείριση και τον διαμοιρασμό αρχείων για κάθε χρήστη.

Σε πρώτο στάδιο παρουσιάζονται βασικές έννοιες ώστε να διασαφηνισθεί η κατανόηση της εφαρμογής τους και των λειτουργιών της. Θεμέλιος λίθος στην εφαρμογή είναι ο καθορισμός της ασφάλειας δεδομένων. Στο πλαίσιο αυτό, έχουν θεσπιστεί ποικίλες πρακτικές και τεχνολογίες που κρίνεται αναγκαίο να ληφθούν υπόψη για να επιτευχθεί η ασφάλεια των αρχείων κατά τη μεταφορά και την αποθήκευση.

Σε επόμενο στάδιο, παρουσιάζεται ενδελεχώς η ανάπτυξη μιας διαδικτυακής εφαρμογής η οποία κρυπτογραφεί και διαμοιράζει αρχεία μεταξύ των χρηστών σε τοπικό δίκτυο. Ο σαφής καθορισμός της ασφάλειας των δεδομένων και το πώς επιτελείται, συνιστά τον βασικότερο σκοπό της εφαρμογής.

Abstract

The purpose and object of this bachelor’s dissertation, is the personalized sharing of files with security, as it is deemed necessary for the protection of sensitive personal data, while at the same time it contributes to flexible access and collaboration in different environments. The goal, however, becomes the development and evaluation of a specific proposal that allows secure file management and sharing with personalized settings for each user.

In a first stage, basic concepts related to encryption and security, private/public key, digital signature and 3-tier architecture are presented, in order to clarify the understanding of their application and its functions. A cornerstone of the application is the definition of data security. In this context, various practices and technologies have been established that are deemed necessary to be taken into account to achieve the security of files during transport and storage.

In a next stage, the development of a web application that encrypts and shares files between users on a local network is presented. The clear definition of data security and how it is carried out is the main purpose of the application.

Κεφάλαιο 1
1.1 Εισαγωγή

Στις μέρες μας, καθίσταται θέμα συζήτησης και έρευνας το ζήτημα της ασφάλειας δεδομένων και συναλλαγών. Έστω πως πραγματοποιείται επικοινωνία δύο πλευρών μέσω κλειστού δικτύου τότε αμβλύνεται ο κίνδυνος μη ασφαλών δεδομένων καθώς ο φορέας που ελέγχει το κλειστό αυτό δίκτυο μπορεί να εντοπίσει με ευκολία οποιεσδήποτε παρεμβολές, υποκλοπές ή κενά ασφαλείας στις επικοινωνίες του δικτύου.

Θεσμικά, σύμφωνα με τον κανονισμό της Ευρωπαϊκής Ένωσης, 2016/679 θεσπίζονται κανόνες για την προστασία των φυσικών προσώπων τόσον έναντι της επεξεργασίας των προσωπικών τους δεδομένων όσο και για την ελεύθερη κυκλοφορία των δεδομένων αυτών. Ο εν λόγω κανονισμός συνιστά ουσιαστικό βήμα προς την ενίσχυση των ατομικών θεμελιωδών δικαιωμάτων στην ψηφιακή εποχή. (Regulation (EU), 2016/679, 2016)

Η χρήση συστημάτων διαμοίρασης αρχείων έχει αυξηθεί με την ανάπτυξη των δικτύων και την αύξηση της ποσότητας των κοινόχρηστων δεδομένων. Οι χρήστες δυσκολεύονται να βρουν γρήγορα ένα επιθυμητό αρχείο επειδή πρέπει να αναζητηθεί σε μια περίπλοκη δομή φακέλων. Πέραν αυτού, τα συστήματα διαμοιρασμού αρχείων εμφανίζουν ευπάθειες ασφαλείας λόγω του γεγονότος ότι οι χρήστες δύνανται να έχουν πρόσβαση σε όλα τα αρχεία ενός φακέλου.

Στην παρούσα πτυχιακή, προτείνεται η ανάπτυξη μίας διαδικτυακής εφαρμογής για τη διαμόρφωση εξατομικευμένων φακέλων. Το σύστημα διαμόρφωσης καταλόγου μπορεί να χρησιμοποιηθεί για την ενοποίηση υποκαταλόγων ή το φιλτράρισμα τμημάτων ενός καταλόγου για την κάλυψη των απαιτήσεων των χρηστών. Έπειτα, τα δικαιώματα πρόσβασης μπορούν να καθοριστούν εξαρχής για τον έλεγχο της πρόσβασης σε αρχεία που απαιτούν ασφάλεια. Κατά συνέπεια κρίνεται να σκόπιμο να περιγράφουν λειτουργίες και αλγόριθμοι για τη διαμόρφωση εξατομικευμένων καταλόγων (Park, Lee, 2017).

1.2.Σκοπός

Σκοπός της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη μιας διαδικτυακής εφαρμογής για τον ασφαλή διαμοιρασμό αρχείων μεταξύ χρηστών της σε τοπικό δίκτυο. Καλούμαστε οπότε να βρούμε και να εφαρμόσουμε τεχνικές υβριδικής κρυπτογραφίας καθώς και hashing για να επιτυγχάνεται η ασφάλεια των δεδομένων των χρηστών της εφαρμογής.

Κεφάλαιο 2 Επισκόπηση του Χώρου

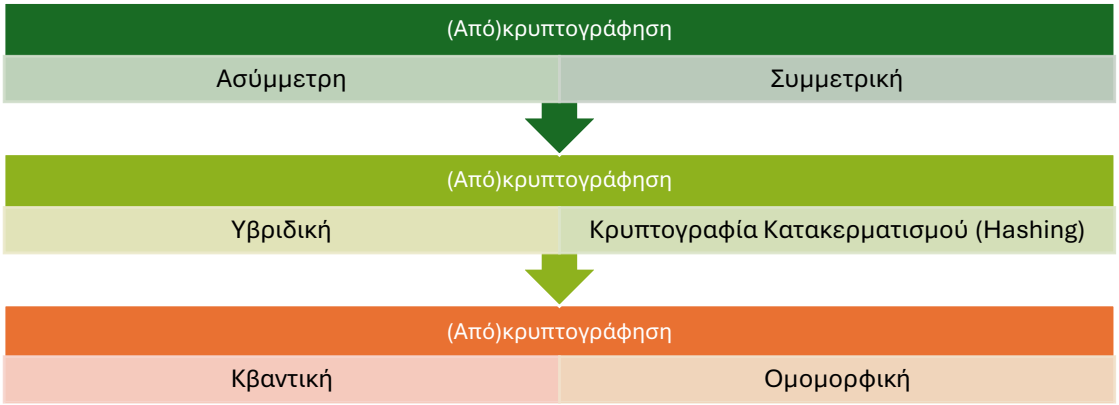
2.1.Βιβλιογραφική επισκόπηση και βασικές έννοιες

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τις βασικές έννοιες, τις υπάρχουσες τεχνολογίες και τις θεωρητικές προσεγγίσεις που σχετίζονται με την εξατομικευμένη διαμοίραση αρχείων με ασφάλεια. Ο στόχος είναι να προσδιοριστεί το πλαίσιο μέσα στο οποίο αναπτύχθηκε η εφαρμογή και να κατανοηθεί η τεχνολογική και θεωρητική βάση της.

2.1.1. Κρυπτογραφία

Η κρυπτογραφία συνιστά την επιστήμη της κρυπτογράφησης και της αποκρυπτογράφησης των δεδομένων, δηλαδή τη μετατροπή των δεδομένων σε μορφή που δεν είναι κατανοητή χωρίς την κατάλληλη "κλειδί". Βασικός στόχος της είναι η **εμπιστευτικότητα** και η **ακεραιότητα** των πληροφοριών, αλλά και η **αυθεντικότητα** και η **μη-αποποίηση** για την προστασία έναντι κακόβουλων επιθέσεων. Αυτή η διαδικασία μετατρέπει δεδομένα απλού κειμένου σε κρυπτογραφημένο κείμενο χρησιμοποιώντας μαθηματικούς αλγόριθμους και καθιστά τα δεδομένα μη αναγνώσιμα, προστατεύοντάς τα.

Η κρυπτογραφία είναι ένα εργαλείο για την επίτευξη της ασφάλειας δεδομένων, αλλά δεν είναι η μόνη μέθοδος. Για πλήρη ασφάλεια, απαιτούνται πολιτικές, πρωτόκολλα και φυσικά μέτρα προστασίας. Οι κρυπτογραφικοί αλγόριθμοι εξασφαλίζουν ότι ακόμα και αν κάποιος αποκτήσει πρόσβαση στα δεδομένα, δεν θα μπορεί να τα κατανοήσει ή να τα εκμεταλλευτεί χωρίς το κατάλληλο κλειδί. Υπάρχουν διάφορα είδη κρυπτογράφησης, που χρησιμοποιούνται για την προστασία δεδομένων και επικοινωνιών. Παρατηρούνται τα παρακάτω βασικά είδη κρυπτογράφησης.



Εικόνα 2: Είδη κρυπτογράφησης

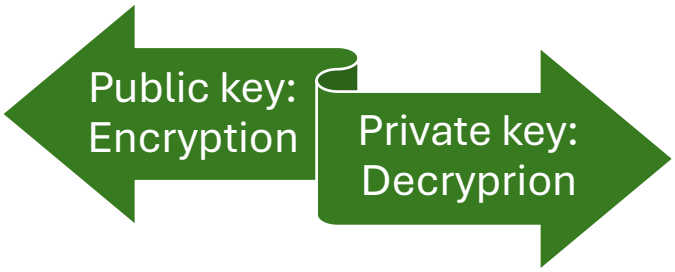
Συμμετρική κρυπτογράφηση
Η συμμετρική κρυπτογράφηση, χρησιμοποιεί το ίδιο κλειδί τόσο για την κρυπτογράφηση όσο και την αποκρυπτογράφηση των δεδομένων. Γνωστοί αλγόριθμοι είναι οι AES (Advanced Encryption Standard) και DES (Data Encryption Standard). Επειδή χρησιμοποιεί το ίδιο κλειδί, η συμμετρική κρυπτογράφηση μπορεί να είναι πιο οικονομική για την ασφάλεια που παρέχει. Κατά συνέπεια είναι σημαντική η επένδυση στην ασφαλή αποθήκευση δεδομένων με την χρήση συμμετρικής κρυπτογράφησης. (Ashtari, 2021) Η συμμετρική κρυπτογραφία ήταν κατάλληλη για οργανισμούς όπως κυβερνήσεις, στρατιωτικοί και μεγάλες χρηματοοικονομικές εταιρείες συμμετείχαν στην διαβαθμισμένη επικοινωνία. (Bhosale, 2020)

Ασύμμετρη πληροφόρηση
Από την άλλη πλευρά, η ασύμμετρη κρυπτογράφηση, χρησιμοποιεί δύο διαφορετικά κλειδιά: για κρυπτογράφηση (δημόσιο κλειδί) και για αποκρυπτογράφηση (ιδιωτικό κλειδί). Ένας από τους πιο γνωστούς αλγόριθμους ασύμμετρης κρυπτογραφίας είναι ο RSA (Rivest–Shamir–

Adleman). Το ιδιωτικό κλειδί δίνεται μόνο σε χρήστες με εξουσιοδοτημένη πρόσβαση. Ως αποτέλεσμα, η ασύμμετρη κρυπτογράφηση μπορεί να είναι πιο αποτελεσματική, αλλά και πιο δαπανηρή. Οι λύσεις κρυπτογράφησης cloud μοιράζονται αυτά τα ιδιωτικά κλειδιά μόνο με τις αρμόδιες αρχές εντός του οργανισμού, μέσω ενός ασφαλούς καναλιού επικοινωνίας.

Αυτά τα κλειδιά πρέπει να αποθηκεύονται με απόλυτη μυστικότητα, καθώς μπορούν να χρησιμοποιηθούν για την αποκρυπτογράφηση όλων των πληροφοριών που έχουν κρυπτογραφηθεί. Ωστόσο, τα δημόσια κλειδιά κοινοποιούνται σε όλους τους σχετικούς μετόχους, τόσο εντός όσο και εκτός του οργανισμού, για να είναι δυνατή η κρυπτογράφηση των δεδομένων πριν από τη μετάδοση. (Bengtsson, 2024)

Το κλειδί κρυπτογράφησης RSA, είναι η τυπική τεχνική κρυπτογράφησης για σημαντική ασφάλεια δεδομένων. Το RSA είναι ασύμμετρη κρυπτογραφία, επομένως υπάρχει ένα δημόσιο κλειδί και ένα ιδιωτικό κλειδί. Ο αλγόριθμος RSA χρησιμοποιεί παραγοντοποίηση πρώτων. Με απλά λόγια, αυτό το κλειδί απαιτεί την παραγοντοποίηση ενός προϊόντος που περιλαμβάνει δύο μεγάλους πρώτους αριθμούς. Αν και φαίνεται εύκολο, το να καταλάβουμε αυτούς τους δύο αριθμούς μπορεί να είναι δύσκολο. Ακόμη και για μεγάλους υπολογιστές, η αποκρυπτογράφηση μπορεί να είναι ακριβή και εξαντλητική. Ενώ το RSA μπορεί να είναι πολύ χρήσιμο, γίνεται όλο και πιο αναποτελεσματικό σε υψηλότερα επίπεδα ασφάλειας. (Cisco,2021).
Τα παραπάνω αποτυπώνονται πιο αναλυτικά στο παρακάτω διάγραμμα:



Εικόνα 3: Ασύμμετρη κρυπτογράφηση και αποκρυπτογράφηση

Κρυπτογραφία κατακερματισμού (Hashing)
Είναι μια μη αντιστρέψιμη λειτουργία που μετατρέπει δεδομένα σε μια σταθερού μεγέθους συμβολοσειρά (hash), όπως οι αλγόριθμοι **SHA (Secure Hash Algorithm)** και **MD5 (Message Digest 5)**. Το hashing είναι μια μαθηματική συνάρτηση που παίρνει μια είσοδο (π.χ. ένα μήνυμα, κείμενο, αρχείο) και παράγει μια σταθερού μεγέθους έξοδο, που ονομάζεται hash ή κατακερματισμένη τιμή. Αυτή η έξοδος είναι μοναδική για κάθε διαφορετική είσοδο και έχει σταθερό μέγεθος, ανεξάρτητα από το μέγεθος των δεδομένων εισόδου. Η κρυπτογραφία κατακερματισμού χρησιμοποιείται σε πολλά πεδία, όπως: αποθήκευση κωδικών πρόσβασης, επαλήθευση ακεραιότητας δεδομένων, ψηφιακές υπογραφές, συγκρίσεις μεγάλων αρχείων.

Υβριδική Κρυπτογράφηση
Η υβριδική κρυπτογράφηση συνδυάζει τις καλύτερες ιδιότητες της συμμετρικής και της ασύμμετρης κρυπτογράφησης, συγχωνεύει δύο ή περισσότερα συστήματα κρυπτογράφησης. Αυτές οι δυνάμεις ορίζονται αντίστοιχα ως ταχύτητα και ασφάλεια. Συχνά, η ασύμμετρη κρυπτογράφηση χρησιμοποιείται για τη διανομή ενός συμμετρικού κλειδιού, το οποίο στη συνέχεια χρησιμοποιείται για την κρυπτογράφηση μεγάλων όγκων δεδομένων. Η υβριδική, χρησιμοποιείται ευρέως σε σύγχρονες εφαρμογές για την επίτευξη του καλύτερου δυνατού αποτελέσματος σε ασφάλεια και απόδοση.

Κβαντική Κρυπτογράφηση
Συνιστά μία ρηξικέλευθη τεχνολογία που στηρίζεται στις αρχές της κβαντικής φυσικής. Παρέχει μεγαλύτερη ασφάλεια, καθώς η παρακολούθηση της επικοινωνίας θα μπορούσε να ανιχνευτεί

αμέσως λόγω της κβαντικής φύσης των σωματιδίων. Το Quantum Key Distribution (QKD) είναι η πιο γνωστή εξ αυτών, επιτρέποντας την ασφαλή μεταφορά κλειδιών. (Ασημάκης, 2015)

Ομομορφική Κρυπτογράφηση
Επιτρέπει τη διενέργεια υπολογισμών πάνω σε κρυπτογραφημένα δεδομένα χωρίς την ανάγκη αποκρυπτογράφησής τους. Αυτό έχει πολλές εφαρμογές σε τομείς όπως το cloud computing, όπου αναπτύχθηκε σε προηγούμενο κεφάλαιο, όπου τα δεδομένα μπορούν να υποβάλλονται σε επεξεργασία ενώ παραμένουν ασφαλή και κρυπτογραφημένα.

2.1.2. Δημόσιο/ Ιδιωτικό Κλειδί

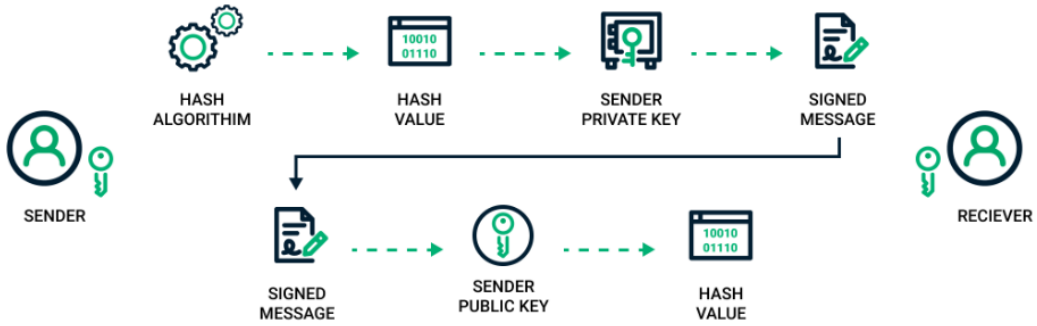
Σε αντίθεση με την κρυπτογραφία συμμετρικού κλειδιού, δεν βρίσκουμε ιστορική χρήση της κρυπτογραφίας δημόσιου κλειδιού. Με την εξάπλωση περισσότερων ανασφαλών δικτύων υπολογιστών τις τελευταίες δεκαετίες, έγινε αισθητή μια πραγματική ανάγκη για χρήση κρυπτογραφίας σε μεγαλύτερη κλίμακα. Το συμμετρικό κλειδί διαπιστώθηκε ότι δεν ήταν πρακτικό λόγω των προκλήσεων που αντιμετώπισε για τη βασική διοίκηση. Αυτό οδήγησε στη δημιουργία κρυπτοσυστημάτων δημόσιου κλειδιού.(Bhosale, 2020)

Η κρυπτογράφηση δημοσίου κλειδιού ή ασύμμετρου κλειδιού επινοήθηκε στο τέλος της δεκαετίας του 1970. Η βασική ιδέα είναι ότι ο αποστολέας και ο παραλήπτης δεν μοιράζονται ένα κοινό μυστικό κλειδί όπως στην περίπτωση της κρυπτογράφησης συμμετρικού κλειδιού, αλλά διαθέτουν διαφορετικά κλειδιά για διαφορετικές λειτουργίες. Συγκεκριμένα κάθε χρήστης διαθέτει δύο κλειδιά κρυπτογράφησης: Το ένα ονομάζεται ιδιωτικό κλειδί (private key) και το άλλο δημόσιο κλειδί (public key). Το ιδιωτικό κλειδί θα πρέπει ο κάθε χρήστης να το προφυλάσσει και να το κρατάει κρυφό, ενώ αντιθέτως το δημόσιο κλειδί μπορεί να το ανακοινώνει σε όλη τη διαδικτυακή κοινότητα ή σε συγκεκριμένους παραλήπτες. Η δημιουργία τους γίνεται από ειδικές συναρτήσεις οι οποίες δέχονται σαν είσοδο έναν μεγάλο τυχαίο αριθμό και στην έξοδο παράγουν το ζεύγος κλειδιών. Είναι προφανές πως όσο πιο τυχαίος ο αριθμός τόσο πιο ασφαλή είναι τα κλειδιά που παράγονται.

Το ιδιωτικό κλειδί διατηρείται μυστικό από τον υπογράφοντα και χρησιμοποιείται για την κρυπτογράφηση της υπογραφής, ενώ το δημόσιο κλειδί είναι διαθέσιμο για οποιονδήποτε να χρησιμοποιήσει για την επαλήθευση της υπογραφής. Είναι σημαντικό να διατηρείται το ιδιωτικό κλειδί ασφαλές για να διασφαλίζεται η γνησιότητα της υπογραφής. Τα πλήκτρα παράγονται χρησιμοποιώντας έναν μαθηματικό αλγόριθμο και η ισχύς των πλήκτρων καθορίζεται από το μήκος του κλειδιού. Όσο μεγαλύτερο είναι το κλειδί, τόσο πιο ασφαλές είναι. Τα κοινά μήκη κλειδιών περιλαμβάνουν 1024, 2048 και 4096 bit. (Wonderbunny, 2023)

2.1.3. Ψηφιακή Υπογραφή

Η ψηφιακή υπογραφή είναι μια από τις πιο σημαντικές εφαρμογές της κρυπτογραφίας, ειδικά της ασύμμετρης κρυπτογράφησης. Χρησιμοποιείται για την επαλήθευση της ταυτότητας του αποστολέα ενός μηνύματος ή εγγράφου και για την εξασφάλιση της ακεραιότητας των δεδομένων. Βασικά της πλεονεκτήματα έναντι άλλων είναι η αυθεντικότητα, η ακεραιότητα και η μη αποποίηση της ευθύνης του παραλήπτη που υπέγραψε το έντυπο. Η ψηφιακή υπογραφή βασίζεται στην ασύμμετρη κρυπτογραφία και στις συναρτήσεις κατακερματισμού για να εξασφαλίσει την ασφάλεια και την ακεραιότητα της επικοινωνίας. Είναι απαραίτητη σε τομείς όπως το ηλεκτρονικό εμπόριο, η διακυβέρνηση, και η προστασία προσωπικών δεδομένων, καθώς εξασφαλίζει τη διαφάνεια και την εμπιστοσύνη στις ψηφιακές διαδικασίες. Ως λύση ψηφιακής ροής εργασιών, οι ψηφιακές υπογραφές επιτρέπουν στους χρήστες να υπογράφουν έγγραφα από οποιαδήποτε τοποθεσία, ανά πάσα στιγμή, προσφέροντας στις επιχειρήσεις ανταγωνιστικό πλεονέκτημα και ενισχύοντας την εμπειρία του πελάτη.



Εικόνα 4: ricoh.com, Πώς δουλεύει η ψηφιακή υπογραφή;

2.1.4. Αρχιτεκτονική Τριών επιπέδων

Τα Συστήματα Διαχείρισης Βάσεων Δεδομένων είναι εφαρμογές λογισμικού που λειτουργούν συνήθως μέσω ενός δικτύου ή του διαδικτύου. Οι κοινοί τρόποι σχεδιασμού του λογισμικού σε ό,τι αφορά τις επιμέρους μονάδες (modules) και τη μεταξύ τους επικοινωνία χαρακτηρίζονται ως αρχιτεκτονικά μοντέλα ή αρχιτεκτονικές.

Η απλούστερη αρχιτεκτονική είναι η αρχιτεκτονική μιας βαθμίδας (1-tier) στην οποία ο υπολογιστής που φιλοξενεί την εφαρμογή είναι ο ίδιος με το οποίο ο χρήστης επικοινωνεί με την εφαρμογή.

Η Αρχιτεκτονική δύο βαθμίδων (2-tier), πρόκειται για την περίπτωση που ένα κεντρικό σύστημα, πολλές φορές με αποκλειστική αρμοδιότητα (server), παρέχει τη λειτουργικότητα για πολλά συστήματα χρηστών (clients).
Το ΣΔΒΔ είναι στον εξυπηρετητή, με τον οποίο επικοινωνούν οι εφαρμογές – πελάτες Η λειτουργικότητα της εφαρμογής του ΣΔΒΔ χωρίζεται σε διαφορετικά επίπεδα που διασυνδέονται λογικά και μοιράζονται πληροφορίες. Κάθε ενδιάμεσο επίπεδο αποτελεί ένα μοντέλο client/server.

Η πιο συνήθης υλοποίηση είναι αυτή των τριών βαθμίδων (3-tier), όπου υπάρχει σαφής φυσικός διαχωρισμός της επεξεργασίας από τη βάση δεδομένων και την παρουσίαση. Συγκρίνοντας τις δύο αρχιτεκτονικές ως προς το κόστος ανάπτυξης και συντήρησης, τα συστήματα 2 βαθμίδων μπορεί να κλιμακώσουν υπερβολικά το κόστος καθώς αυξάνει η πολυπλοκότητα των εφαρμογών δηλ. η επεξεργασία και ο χρόνος ζωής της εφαρμογής. Η Αρχιτεκτονική Τριών Επιπέδων (Three-Tier Architecture) είναι ένα δομικό πρότυπο που χρησιμοποιείται ευρέως στον σχεδιασμό πληροφοριακών συστημάτων, ειδικά σε περιβάλλοντα δικτύων και web εφαρμογών. Χωρίζει το σύστημα σε τρία διακριτά επίπεδα, με σκοπό να επιτευχθεί καλύτερη οργάνωση, επεκτασιμότητα και συντήρηση των συστημάτων. Αυτά τα τρία επίπεδα είναι:

- Επίπεδο Παρουσίασης (Presentation Tier)
- Επίπεδο Λογικής Επιχειρήσεων (Business Logic Tier)
- Επίπεδο Δεδομένων (Data Tier)

Η αρχιτεκτονική τριών επιπέδων είναι μια αποδοτική, επεκτάσιμη και ευέλικτη μέθοδος για την οργάνωση πολύπλοκων εφαρμογών, διαχωρίζοντας τη λογική της εφαρμογής, την αποθήκευση δεδομένων και την παρουσίαση σε διαφορετικά επίπεδα. Αυτή η προσέγγιση εξασφαλίζει καλύτερη απόδοση, ευκολότερη συντήρηση και ευελιξία στην ανάπτυξη.

Κεφάλαιο 3
Απαιτήσεις Συστήματος

3.1 Λειτουργικές Απαιτήσεις

- **Εγγραφή Χρήστη**: Το σύστημα πρέπει να παρέχει τη δυνατότητα εγγραφής νέων χρηστών. Κάθε χρήστης πρέπει να παρέχει διαπιστευτήρια, όπως email και κωδικό πρόσβασης, για τη δημιουργία λογαριασμού.
- **Σύνδεση Χρήστη**: Το σύστημα πρέπει να επιτρέπει στους χρήστες να συνδέονται με τα διαπιστευτήριά τους. Το σύστημα θα επαληθεύει τα στοιχεία του χρήστη μέσω της βάσης δεδομένων.
- **Ανέβασμα Αρχείων**: Οι χρήστες πρέπει να μπορούν να ανεβάζουν αρχεία στην πλατφόρμα, με συγκεκριμένα όρια μεγέθους και τύπους αρχείων που υποστηρίζονται (π.χ., PDF, DOCX).
- **Διαμοιρασμός Αρχείων**: Το σύστημα πρέπει να επιτρέπει στους χρήστες να διαμοιράζονται αρχεία με άλλους χρήστες.
- **Λήψη Αρχείων:** Θα πρέπει οι χρήστες να έχουν την ικανότητα να κατεβάσουν αρχεία από το σύστημα.
- **Κρυπτογράφηση Αρχείων**: Κάθε αρχείο που ανεβαίνει στο σύστημα πρέπει να κρυπτογραφείται για να διασφαλιστεί η ασφάλεια των δεδομένων.
- **Ψηφιακή υπογραφή χρήστη:** Κάθε αρχείο θα πρέπει να συνοδεύεται από μια ψηφιακή υπογραφή και ο παραλήπτης θα πρέπει πάντα να την ελέγχει για να βεβαιωθεί για τον αποστολέα αλλά και για την ακεραιότητα του αρχείου.

3.2 Μη Λειτουργικές Απαιτήσεις

- **Απόδοση**: Το σύστημα πρέπει να ανταποκρίνεται εντός 2 δευτερολέπτων σε αιτήματα χρήστη, όπως την αποστολή και τη λήψη αρχείων.

- **Κλιμάκωση**: Το σύστημα πρέπει να μπορεί να διαχειρίζεται τουλάχιστον 1000 ταυτόχρονους χρήστες χωρίς να επηρεάζεται η απόδοσή του.

- **Ασφάλεια**: Το σύστημα πρέπει να εξασφαλίζει την κρυπτογράφηση των δεδομένων τόσο κατά την αποθήκευση (storage) όσο και κατά τη μεταφορά (transmission).

- **Ευχρηστία**: Η διεπαφή χρήστη (UI) πρέπει να είναι φιλική προς τον χρήστη, με εύκολη πλοήγηση και λειτουργικότητα που να επιτρέπει στους χρήστες να ολοκληρώνουν τις κύριες εργασίες με λίγα βήματα.

- **Συμβατότητα**: Το σύστημα πρέπει να είναι συμβατό με τους πιο δημοφιλείς browsers (π.χ., Chrome, Firefox, Edge) και να υποστηρίζει διάφορες συσκευές, συμπεριλαμβανομένων desktop, tablet και smartphone.

- **Αξιοπιστία**: Το σύστημα πρέπει να έχει διαθεσιμότητα 99.9% για να διασφαλιστεί η συνεχής πρόσβαση των χρηστών.

- **Επεκτασιμότητα**: Το σύστημα πρέπει να είναι επεκτάσιμο και να μπορεί να προσαρμόζεται για μελλοντικές αναβαθμίσεις ή την εισαγωγή νέων λειτουργιών.

3.3 Περιγραφή Σεναρίων Χρήσης (Use Case Description)

Σενάριο Χρήσης 1: Εγγραφή Χρήστη

Περιγραφή:

Ο χρήστης εισάγει τα στοιχεία του για να δημιουργήσει νέο λογαριασμό στην εφαρμογή.

Συμμετέχοντες:

- Χρήστης (Actor)
- Σύστημα (Safe File Sharing Application)

Προϋποθέσεις:

Ο χρήστης πρέπει να μην έχει ήδη λογαριασμό στο σύστημα.

Βασική Ροή:

1. Ο χρήστης ανοίγει τη σελίδα εγγραφής.
2. Ο χρήστης εισάγει το email, το όνομα, το επώνυμο και τον κωδικό πρόσβασης.
3. Το σύστημα ελέγχει αν το email είναι έγκυρο και αν υπάρχει ήδη καταχωρημένο.
4. Το σύστημα αποθηκεύει τα στοιχεία του χρήστη κρυπτογραφημένα στη βάση δεδομένων.
5. Το σύστημα εμφανίζει μήνυμα επιτυχούς εγγραφής και ανακατευθύνει τον χρήστη στη σελίδα σύνδεσης.

Εναλλακτικές Ροές:

- 3a. Αν το email δεν είναι έγκυρο, το σύστημα εμφανίζει μήνυμα λάθους και ζητά από τον χρήστη να διορθώσει το email.
- 3b. Αν το email υπάρχει ήδη, το σύστημα ενημερώνει τον χρήστη να εισάγει διαφορετικό email.

Μετα-συνθήκες:

Ο νέος λογαριασμός έχει δημιουργηθεί και τα στοιχεία του χρήστη είναι αποθηκευμένα στη βάση δεδομένων.

Σενάριο Χρήσης 2: Σύνδεση Χρήστη

Περιγραφή:

Ο χρήστης εισάγει τα στοιχεία του για να συνδεθεί στην εφαρμογή και να αποκτήσει πρόσβαση στα αρχεία του.

Συμμετέχοντες:

- Χρήστης (Actor)
- Σύστημα (Safe File Sharing Application)

Προϋποθέσεις:

Ο χρήστης πρέπει να έχει ήδη λογαριασμό στο σύστημα.

Βασική Ροή:

1. Ο χρήστης ανοίγει τη σελίδα σύνδεσης.
2. Ο χρήστης εισάγει το email και τον κωδικό πρόσβασης.
3. Το σύστημα ελέγχει αν τα διαπιστευτήρια είναι σωστά.
4. Το σύστημα δημιουργεί συνεδρία για τον χρήστη και τον ανακατευθύνει στην κύρια σελίδα της εφαρμογής.

Εναλλακτικές Ροές:

- 3a. Αν τα διαπιστευτήρια είναι λανθασμένα, το σύστημα εμφανίζει μήνυμα λάθους και ζητά από τον χρήστη να επαναλάβει τη διαδικασία.

Μετα-συνθήκες:

Ο χρήστης έχει συνδεθεί επιτυχώς και έχει πρόσβαση στα προσωπικά του αρχεία.

Σενάριο Χρήσης 3: Μεταφόρτωση Αρχείων

Περιγραφή:

Ο χρήστης ανεβάζει ένα αρχείο στην εφαρμογή, το οποίο κρυπτογραφείται πριν αποθηκευτεί στη βάση δεδομένων.

Συμμετέχοντες:

- Χρήστης (Actor)
- Σύστημα (Safe File Sharing Application)

Προϋποθέσεις:

Ο χρήστης πρέπει να είναι συνδεδεμένος στο σύστημα.

Βασική Ροή:

1. Ο χρήστης ανοίγει τη σελίδα επιλογής αρχείων για μεταφόρτωση.
2. Ο χρήστης επιλέγει το αρχείο από τον υπολογιστή του.
3. Το σύστημα δημιουργεί ψηφιακή υπογραφή για το αρχείο
4. Το σύστημα κρυπτογραφεί το αρχείο με AES και αποθηκεύει το κρυπτογραφημένο αρχείο μαζι με την υπογραφή του αλλά και το AES κλειδί αφού έχει κρυπτογραφυθεί με RAS δημόσιο κλειδί στη βάση δεδομένων.
5. Το σύστημα εμφανίζει μήνυμα επιτυχούς μεταφόρτωσης και το αρχείο εμφανίζεται στη λίστα των αρχείων του χρήστη.

Εναλλακτικές Ροές:

- 3a. Αν ο χρήστης επιλέξει κάποιο αρχείο αρκετά μεγάλου μεγέθου, το σύστημα μπορεί εμφανίσει μήνυμα λάθους.

Μετα-συνθήκες:

Το αρχείο έχει κρυπτογραφηθεί και αποθηκευτεί στη βάση δεδομένων με επιτυχία.

Σενάριο Χρήσης 4: Διαμοιρασμός Αρχείων με Άλλους Χρήστες

Περιγραφή:

Ο χρήστης διαμοιράζεται ένα κρυπτογραφημένο αρχείο με έναν άλλο χρήστη.

Συμμετέχοντες:

- Χρήστης (Actor)
- Σύστημα (Safe File Sharing Application)
- Παραλήπτης (Actor)

Προϋποθέσεις:

Ο χρήστης πρέπει να έχει ανεβάσει τουλάχιστον ένα αρχείο στο σύστημα και να γνωρίζει το email του παραλήπτη.

Βασική Ροή:

1. Ο χρήστης ανοίγει τη σελίδα διαμοιρασμού αρχείων.
2. Ο χρήστης επιλέγει το αρχείο που θέλει να διαμοιραστεί και εισάγει το email του παραλήπτη.
3. Το σύστημα κρυπτογραφεί το κλειδί αποκρυπτογράφησης με το δημόσιο κλειδί του παραλήπτη.
4. Το σύστημα στέλνει το αρχείο με το κρυπτογραφημένο κλειδί.
5. Ο παραλήπτης αποκρυπτογραφεί με το RSA κλειδί του το AES κρυπτογραφημένο κλειδί για να ανοίξει το αρχείο .

Μετα-συνθήκες:

Ο παραλήπτης έχει πρόσβαση στο κρυπτογραφημένο αρχείο και μπορεί να το αποκρυπτογραφήσει.

Σενάριο Χρήσης 5: Λήψη Αρχείων

Περιγραφή:

Ο χρήστης κατεβάζει ένα κρυπτογραφημένο αρχείο και το αποκρυπτογραφεί τοπικά.

Συμμετέχοντες:

- Χρήστης (Actor)
- Σύστημα (Safe File Sharing Application)

Προϋποθέσεις:

Ο χρήστης πρέπει να είναι συνδεδεμένος στο σύστημα και να έχει ανεβάσει ή να του έχει διαμοιραστεί κάποιο αρχείο.

Βασική Ροή:

1. Ο χρήστης επιλέγει το αρχείο που θέλει να κατεβάσει.
2. Το σύστημα αποκρυπτογραφεί το κρυπτογραφημένο κλειδί με το ιδιωτικό κλειδί του χρήστη.
3. Το αρχείο κατεβαίνει στον υπολογιστή του χρήστη και αποκρυπτογραφείται τοπικά με το κλειδί AES.
4. Το σύστημα εμφανίζει μήνυμα επιτυχούς λήψης.

Εναλλακτικές Ροές:

- 3a. Αν το κλειδί αποκρυπτογράφησης δεν είναι έγκυρο, το σύστημα εμφανίζει μήνυμα λάθους και η λήψη ακυρώνεται.

Μετα-συνθήκες:

Το αρχείο έχει ληφθεί και αποκρυπτογραφηθεί με επιτυχία.

4. Η ανάπτυξη του συστήματος

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τις τεχνολογίες, τα εργαλεία και τις διαδικασίες που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος ασφαλούς διαμοιρασμού αρχείων. Το σύστημα βασίζεται σε διαδικτυακή εφαρμογή που προσφέρει στους χρήστες τη δυνατότητα να κρυπτογραφούν και να διαμοιράζονται αρχεία μέσω ασφαλών συνδέσεων.

4.1 Αρχιτεκτονική Συστήματος

Το σύστημα βασίζεται στην αρχιτεκτονική **3-Tier**, η οποία αποτελείται από τα εξής επίπεδα:

- **Επίπεδο Παρουσίασης (Presentation Tier)**: Περιλαμβάνει το frontend της εφαρμογής που αποτελείται από HTML, CSS, και JavaScript. Η διεπαφή χρήστη επιτρέπει την αλληλεπίδραση με το σύστημα.
- **Επίπεδο Λογικής Επιχειρήσεων (Business Logic Tier)**: Αποτελεί το backend της εφαρμογής που αναπτύχθηκε σε Java. Χρησιμοποιήθηκαν Java Servlets για την επεξεργασία αιτημάτων των χρηστών και τη διαχείριση των κρυπτογραφικών λειτουργιών.
- **Επίπεδο Δεδομένων (Data Tier)**: Αφορά την αποθήκευση των δεδομένων στη βάση MongoDB. Τα αρχεία των χρηστών αποθηκεύονται κρυπτογραφημένα, ενώ τα κλειδιά κρυπτογράφησης διαχειρίζονται μέσω RSA και AES.

4.2 Τεχνολογίες και Εργαλεία

Για την ανάπτυξη του συστήματος χρησιμοποιήθηκαν σύγχρονες τεχνολογίες που διασφαλίζουν τόσο την αποδοτικότητα όσο και την ασφάλεια των δεδομένων.

4.2.1 Εργαλεία που χρησιμοποιήθηκαν

Σε αυτό το υποκεφάλαιο παρουσιάζονται τα κύρια εργαλεία που χρησιμοποιήθηκαν κατά την ανάπτυξη της εφαρμογής εξατομικευμένου διαμοιρασμού αρχείων.

Eclipse IDE for Enterprise Java and Web Developers

Το **Eclipse IDE** είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment - IDE) που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής σε γλώσσα προγραμματισμού **Java**. Ειδικότερα, η έκδοση **Eclipse IDE for Enterprise Java and Web Developers** προσφέρει εργαλεία για την ανάπτυξη web εφαρμογών βασισμένων σε Java, με ενσωματωμένη υποστήριξη για **Servlets**, **JSP**, και ενσωμάτωση με διάφορους εξυπηρετητές εφαρμογών (application servers) όπως ο **Tomcat**.

Tomcat Apache

Ο **Tomcat Apache** είναι ένας εξυπηρετητής εφαρμογών που χρησιμοποιείται κυρίως για την εκτέλεση Java Servlets και JavaServer Pages (JSP). Στην εφαρμογή χρησιμοποιήθηκε για τη φιλοξενία και εκτέλεση των web components του συστήματος, επιτρέποντας την εξυπηρέτηση αιτήσεων HTTP και την επικοινωνία μεταξύ πελάτη και διακομιστή.

MongoDB Compass

Το **MongoDB Compass** είναι ένα γραφικό εργαλείο διαχείρισης της βάσης δεδομένων **MongoDB**, που επιτρέπει στους χρήστες να παρακολουθούν, να αναλύουν και να διαχειρίζονται δεδομένα. Στην εφαρμογή, η **MongoDB** χρησιμοποιήθηκε για την αποθήκευση και ανάκτηση δεδομένων που αφορούν τους χρήστες και τα αρχεία που διαμοιράζονται, ενώ το **MongoDB Compass** προσέφερε εύκολο τρόπο διαχείρισης της βάσης δεδομένων χωρίς να απαιτείται γνώση εντολών.

4.2.2 Γλώσσες Προγραμματισμού

Οι κύριες γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής είναι οι ακόλουθες.

Java

Η **Java** ήταν η κύρια γλώσσα προγραμματισμού για την ανάπτυξη του backend της εφαρμογής. Χρησιμοποιήθηκε για την υλοποίηση της λογικής της εφαρμογής, τη διαχείριση των αιτημάτων από τον πελάτη και την επικοινωνία με τη βάση δεδομένων.

Jasper

Το **Jasper** είναι η μηχανή μεταγλώττισης (compiler) για JSP (JavaServer Pages). Χρησιμοποιήθηκε για τη μεταγλώττιση των αρχείων **JSP** σε Servlets, επιτρέποντας την εκτέλεση δυναμικού περιεχομένου στον εξυπηρετητή.

JSTL

Η **JSTL (JavaServer Pages Standard Tag Library)** είναι μια βιβλιοθήκη ετικετών που παρέχει διάφορα πρότυπα εργαλεία για JSP, όπως ετικέτες για βρόχους, υπό όρους ελέγχους, επεξεργασία XML και διεθνοποίηση. Χρησιμοποιήθηκε στην εφαρμογή για τη διευκόλυνση της ανάπτυξης δυναμικών ιστοσελίδων με απλό και κατανοητό τρόπο.

HTML

Η **HTML (Hypertext Markup Language)** χρησιμοποιήθηκε για την ανάπτυξη του frontend της εφαρμογής. Παρέχει τη βασική δομή των ιστοσελίδων της εφαρμογής.

Bootstrap

Το **Bootstrap** είναι ένα πλαίσιο (framework) HTML, CSS, και JavaScript, που χρησιμοποιήθηκε για τη δημιουργία responsive και μοντέρνων διεπαφών χρήστη (user interfaces) με ευκολία. Παρέχει προκαθορισμένα στυλ και εργαλεία που βοηθούν στην ανάπτυξη ιστοσελίδων με φιλικό προς το χρήστη σχεδιασμό.

JavaScript

Η **JavaScript** χρησιμοποιήθηκε για την ανάπτυξη διαδραστικών λειτουργιών στην εφαρμογή. Επέτρεψε τη διαχείριση της δυναμικής συμπεριφοράς των ιστοσελίδων, όπως την επεξεργασία φόρμας για τη βελτίωση της εμπειρίας χρήστη.
Με αυτά τα εργαλεία και γλώσσες προγραμματισμού, η εφαρμογή εξασφάλισε τόσο τη σωστή λειτουργία του backend όσο και τη δημιουργία ενός λειτουργικού και αισθητικά άρτιου frontend.

CSS

Η **CSS (Cascading Style Sheets)** είναι μια γλώσσα μορφοποίησης που χρησιμοποιήθηκε για τον καθορισμό της εμφάνισης και διάταξης των στοιχείων της ιστοσελίδας. Μέσω της CSS, η εφαρμογή απέκτησε μια ευπαρουσίαστη και ελκυστική διεπαφή, καθορίζοντας τα στυλ, τα χρώματα, τις γραμματοσειρές, τις αποστάσεις και τις διατάξεις.
Η CSS επιτρέπει την προσαρμογή της αισθητικής της εφαρμογής για διαφορετικές συσκευές και μεγέθη οθόνης, γεγονός που είναι σημαντικό για τη βελτίωση της εμπειρίας χρήστη. Σε συνδυασμό με το **Bootstrap** βοήθησε να διατηρηθεί ένα responsive και συνεπές design σε όλες τις σελίδες της εφαρμογής.
Αυτό επέτρεψε τη βελτίωση της εμπειρίας του χρήστη, καθώς οι σελίδες της εφαρμογής προσαρμόζονται σε διαφορετικές συσκευές και οθόνες, όπως κινητά τηλέφωνα, tablets και επιτραπέζιοι υπολογιστές, παρέχοντας μια άνετη και ευχάριστη πλοήγηση.

4.2.3 Κρυπτογραφικές Τεχνικές

- **AES (Advanced Encryption Standard)**: Χρησιμοποιήθηκε για την κρυπτογράφηση των αρχείων πριν από την αποθήκευση.

- **RSA (Rivest-Shamir-Adleman)**: Χρησιμοποιήθηκε για την κρυπτογράφηση των συμμετρικών κλειδιών (AES keys), διασφαλίζοντας ότι μόνο οι εξουσιοδοτημένοι χρήστες μπορούν να αποκρυπτογραφήσουν τα αρχεία (Υβριδική κρυπτογραφία).
- **Ψηφιακές Υπογραφές**: Χρησιμοποιούνται για την επαλήθευση της ακεραιότητας των αρχείων και την αυθεντικότητα των αποστολέων.

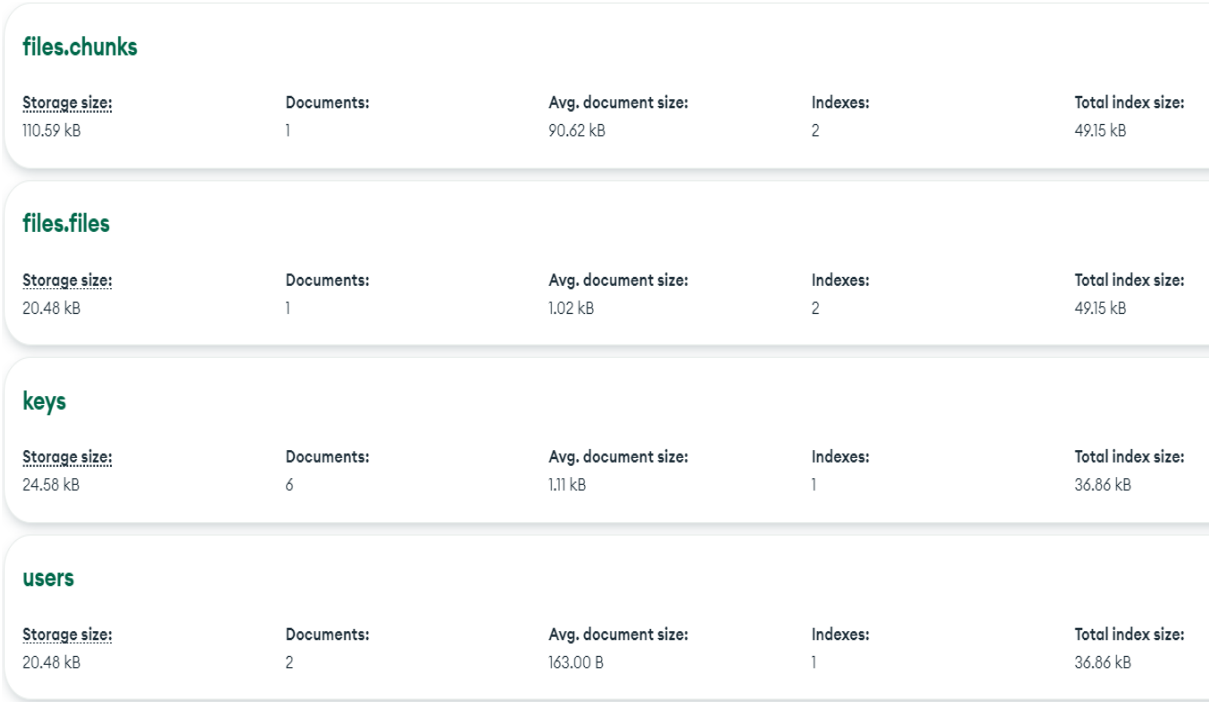
4.3 Διαδικασία Ανάπτυξης

4.3.1 Υλοποίηση Εφαρμογής

4.3.1.1 Βάση Δεδομένων

Για την αποθήκευση και διαχείριση των δεδομένων της εφαρμογής χρησιμοποιήθηκε η **MongoDB**, μια βάση δεδομένων NoSQL. Όλα τα αρχεία των χρηστών και οι πληροφορίες πρόσβασης αποθηκεύονται σε συλλογές (collections). Τα δεδομένα είναι εύκολα διαχειρίσιμα και η MongoDB παρέχει κλιμάκωση για την υποστήριξη μεγάλου αριθμού χρηστών και αρχείων.

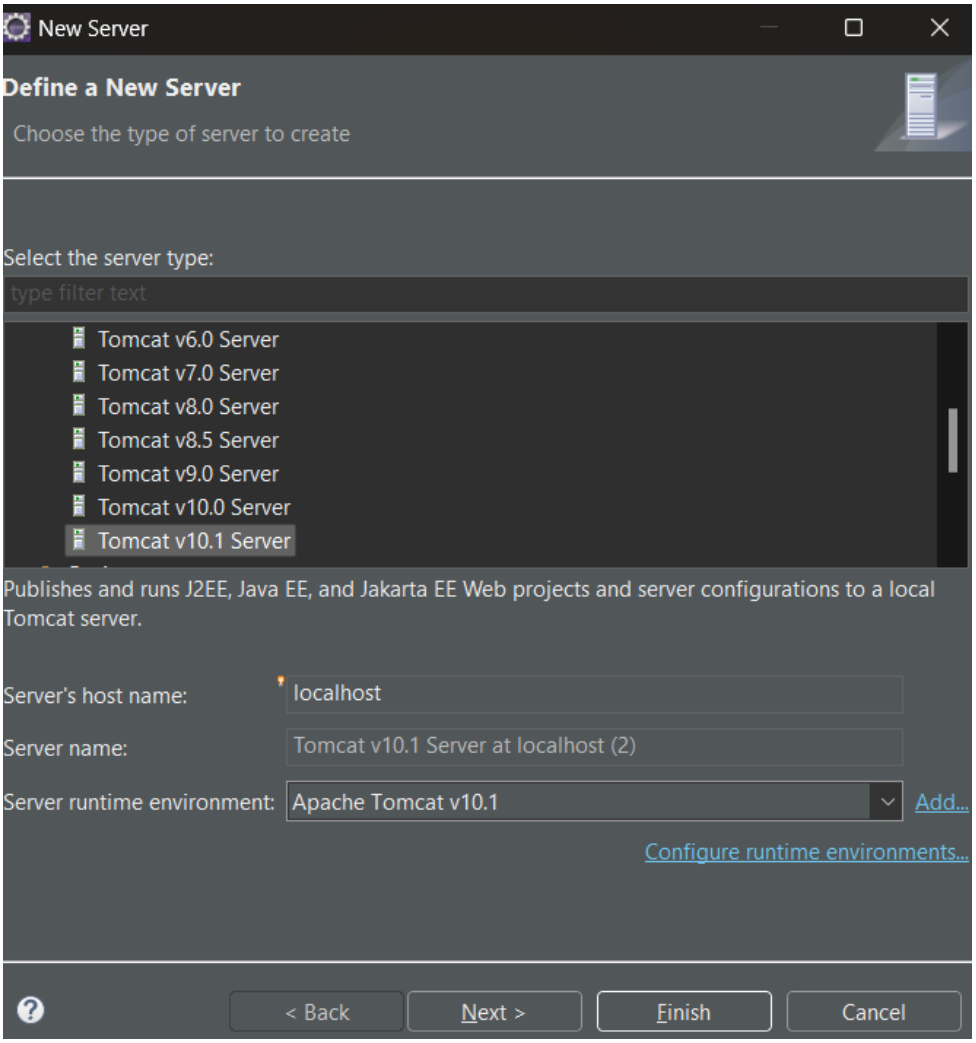
Μέσω του MongoDB Compass δημιουργήσαμε βάση δεδομένων με όνομα filesharing και τις συλλογές (collections) users και keys όπως φαίνεται στην παρακάτω εικόνα:
Σημειώνεται ότι: Οι άλλες δυο δημιουργήθηκαν προγραμματιστικά για την αποθήκευση αρχείων.



Εικόνα 5:Βάση Δεδομένων MongoDB.

4.3.1.2 Σύνδεση Apache Tomcat

Για τη συγκεκριμένη εφαρμογή θα χρησιμοποιήσουμε Apache Tomcat 10.1 εξυπηρετητή (server) όπου μόλις τον εγκαταστήσουμε, ανοίγουμε την πλατφόρμα Eclipse και τον προσθέτουμε στους Servers:



Εικόνα 6: Επιλογή Tomcat Server στο Eclipse.

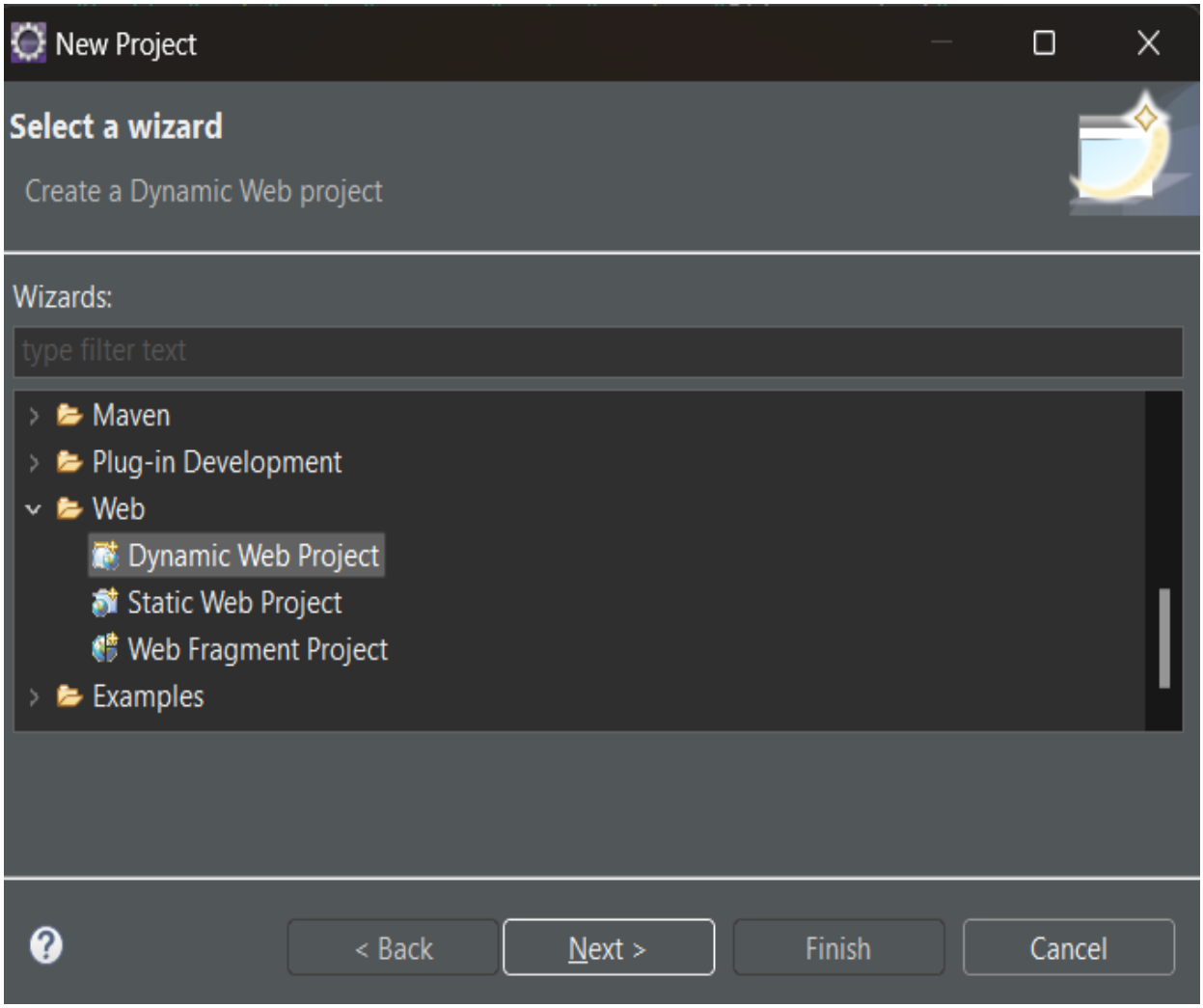
Επόμενο βήμα είναι να ορίσουμε την βάση δεδομένων στον Server και στο αρχείο web.xml θα προσθέσουμε τον εξής κώδικα:

```
<Resource name="mongodb/MyMongoClient"
  auth="Container"
  type="com.mongodb.client.MongoClient"
  closeMethod="close"
  factory="com.mongodb.client.MongoClientFactory"
  singleton="true"
  connectionString="mongodb://localhost:27017"
  maxActive="12"/>
```

Καθώς και την προσθήκη των mongo driver (mongo-java-driver-3.12.14 , mongodb-driver-core-5.1.2 mongodb-driver-sync-5.1.2) στις βιβλιοθήκες του Tomcat.

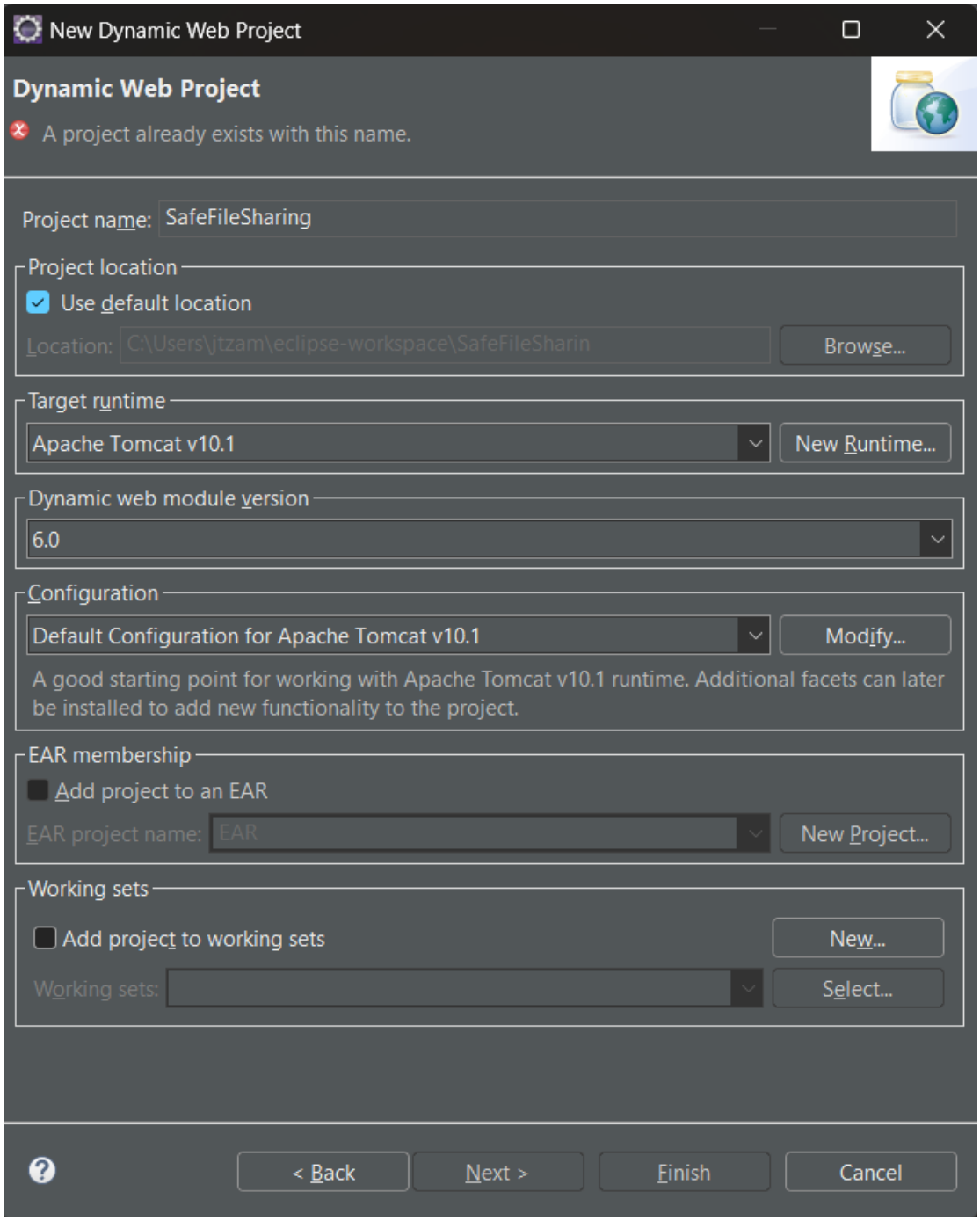
4.3.1.3 Δημιουργία δυναμικής διαδυκτιακης ιστοσελιδας (Dynamic Web app)

Στο Eclipse δημιουργούμε ένα νέο project επιλέγοντας το **Dynamic Web project.**



Εικόνα 7: Δημιουργία δυναμικής διαδικτυακής εφαρμογής.

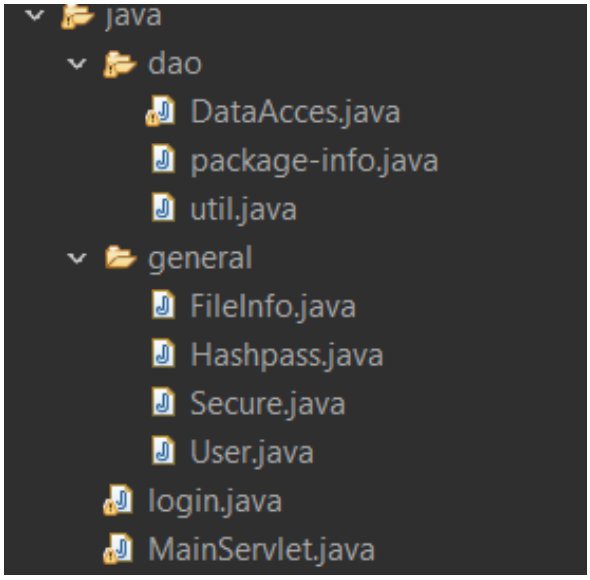
Έπειτα επιλέγουμε να λειτουργεί στο Tomcat 10.1.



Εικόνα 8: Τελευταίες ρυθμίσεις για την δημιουργία της εφαρμογής.

Δημιουργία κώδικα Backend σε Java

Για την εφαρμογή δημιουργούμε δυο πακέτα κλάσεων και δυο επιπλέον κλάσεις Servlets. Αναλυτικότερα τα πακέτα general και dao με κλάσεις FileInfo, Hashpass, Secure, User στο πρώτο και DataAcces, util στο δεύτερο και επιπροσθέτως τις δύο επιπλέον: Servlets login και MainServlet.



Εικόνα 9: Κλάσεις της εφαρμογής.

Λειτουργία πακέτου general

Το πακέτο general περιέχει μεθόδους και αντικείμενα που χρησιμοποιούνται σε κάθε στάδιο της εφαρμογής πιο συγκεκριμένα:

- **User:**
  - Δημιουργεί αντικείμενα user που βοηθάνε στην αποθήκευση και μεταφορά των στοιχείων των χρηστών τόσο στην σύνδεση με την βάση δεδομένων όσο και μεταξύ των Servlets
- **FileInfo:**
  - Όπως και η User έτσι και η FileInfo δημιουργεί αντικείμενα που εξυπηρετούν στην μεταφορά και αποθήκευση στοιχείων αλλά για τα αρχεία.
- **Secure:**
  - Εμπεριέχει όλες της μεθόδους ασφάλειας (εκτός από το hashing), δηλαδή τη δημιουργία κλειδιών (συμμετρικών και ασύμμετρων), κρυπτογράφηση, αποκρυπτογράφηση, την ψηφιακή υπογραφή και τον έλεγχο της ορθότητας της ψηφιακής υπογραφής.
- **Hashpass:**
  - Hashing

Λειτουργία πακέτου dao

Το πακέτο dao χρησιμοποιείται για την πρόσβαση στην βάση δεδομένων, δηλαδή, για τις κλάσεις της ισχύει:

- **Util**
  - Πραγματοποιεί σύνδεση με την βάση δεδομένων.
- **DataAcces**
  - Χρησιμοποιεί την κλάση util για να συνδεθεί στην βάση δεδομένων. Μέσω των μεθόδων της DataAcces τα Servlets μπορούν να επεξεργάζονται και να διαβάζουν τα δεδομένα της βάσης.

Λειτουργία των Servlets

Τα Servlets λειτουργούν ως ενδιάμεσος μεταξύ του χρήστη και του συστήματος. Κάθε φορά που ένας χρήστης στέλνει ένα αίτημα, το αίτημα αποστέλλεται στον διακομιστή, όπου ένα Servlet αναλαμβάνει να επεξεργαστεί το αίτημα και να επιστρέψει την κατάλληλη απάντηση στον χρήστη.

Τα Servlets login και MainServlet διαχειρίζονται τελείως διαφορετικά αιτήματα πιο αναλυτικά:

- **Login**
  - Διαχειρίζεται αιτήματα, για τη σύνδεση και την αποσύνδεση των χρηστών αλλά και της εγγραφής των νέων χρηστών.
- **MainServlet**
  - Διαχειρίζεται αιτήματα μεταφόρτωσης, λήψης, μεταφοράς αρχείων αλλά και την προβολή των λεπτομερειών τους.

Δημιουργία κώδικα Frontend

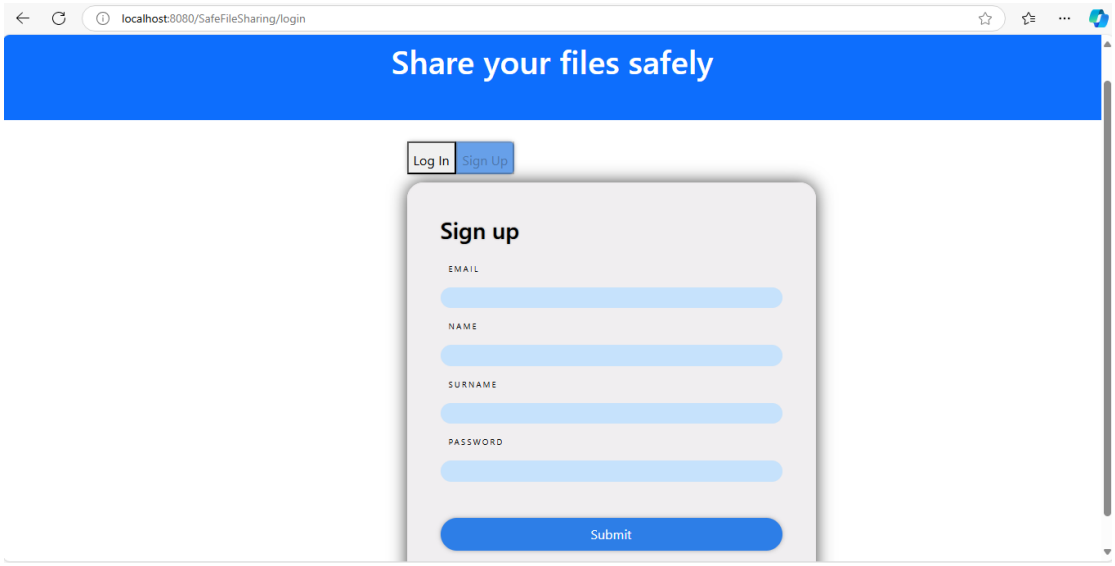
Η ανάπτυξη των frontend έγινε με δυο αρχεία .jsp τα login.jsp και main.jsp και δυο .css style.css και style1.css. Τα css αρχεία ρυθμίζουν την εμφάνιση των δυο jsp αρχείων ενώ τα jsp αρχεία που αποτελούνται από html (για στοιχεία όπως κουμπιά, πίνακες και άλλα) , javascript (για διαδραστικότητα εντός των στοιχείων html όπως λειτουργικότητα κάποιων κουμπιών ή προβολή μηνυμάτων), bootstrap (για τον ίδιο λόγο με τα css αρχεία) και jasper (για αλληλεπίδραση με τα Servlet). Η λειτουργικότητα αυτών των αρχείων είναι η εξής:

- **login.jsp**
  - Περιέχει δυο φόρμες συμπλήρωσης μια για την πραγματοποίηση σύνδεσης του χρήστη και μια για την εγγραφή καινούριου χρήστη. Τα στοιχεία που συλλέγει αποστέλλονται στο Servlet login για επεξεργασία.
- **Main.jsp**
  - Περιέχει πίνακα προβολής στοιχείων από αρχεία με δυο κουμπιά ανά στοιχείο του πίνακα: Ένα για αίτημα λήψης και ένα διαμοίρασης αρχείων. Άλλα τρία κουμπιά χρησιμοποιούνται για το άνοιγμα παραθύρου επιλογής αρχείων από τον υπολογιστή, τη μεταφόρτωση του επιλεγμένου αρχείου και για την αποσύνδεση του χρήστη.

5. Παρουσίαση Εφαρμογής

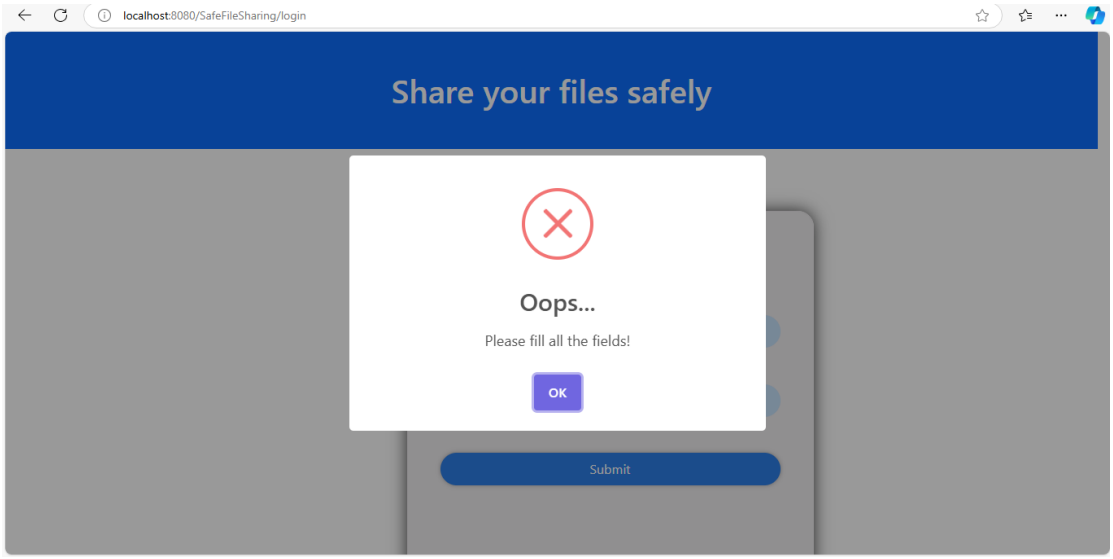
5.1 Εγγραφή Νέου Χρήστη

Αφού ανοίξετε τον browser σας και μεταβείτε στη διεύθυνση της εφαρμογής (localhost:8080/SafeFileSharing/) επιλέγετε το κουμπί «Sign up» στο πάνω μέρος της φόρμας (αν δεν είναι ήδη επιλεγμένο) και συμπληρώνετε τα στοιχεία σας (email, όνομα, επώνυμο, κωδικός πρόσβασης). Έπειτα πατήστε κουμπί «Submit».



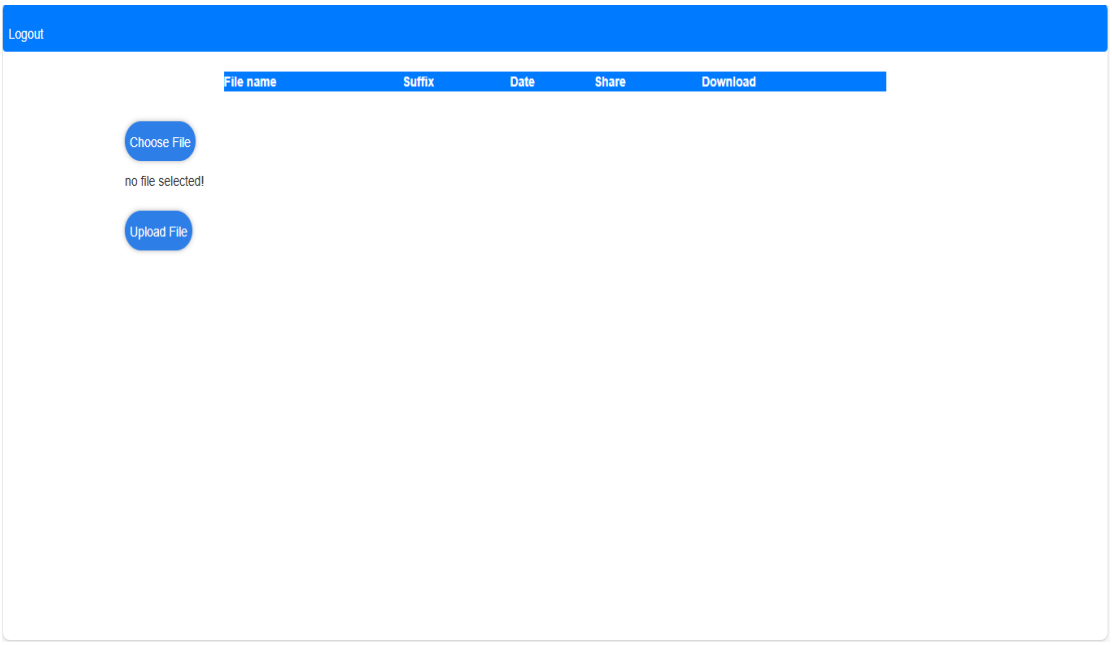
Εικόνα 10: Σελίδα εγγραφής χρήστη.

Σε περίπτωση που υπάρχει ήδη χρήστης με αυτό το email ή κάποιο από τα στοιχεία είναι λανθασμένο (μικρός κωδικός πρόσβασης, κενά στοιχεία κτλ) θα ειδοποιηθείτε ώστε να ξαναπροσπαθήσετε.



Εικόνα 11: Σελίδα σφάλματος χρήστη κατά την εγγραφή.

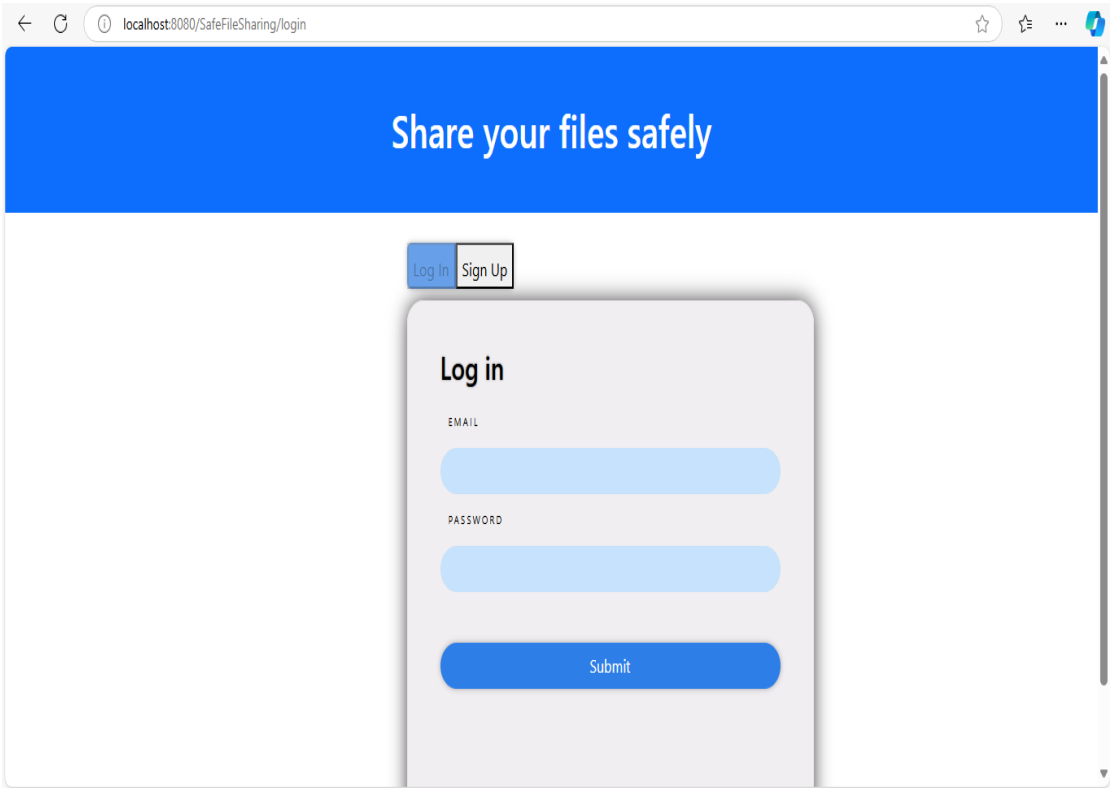
Διαφορετικά συνδέεστε στο αρχικό μενού χρήστη.



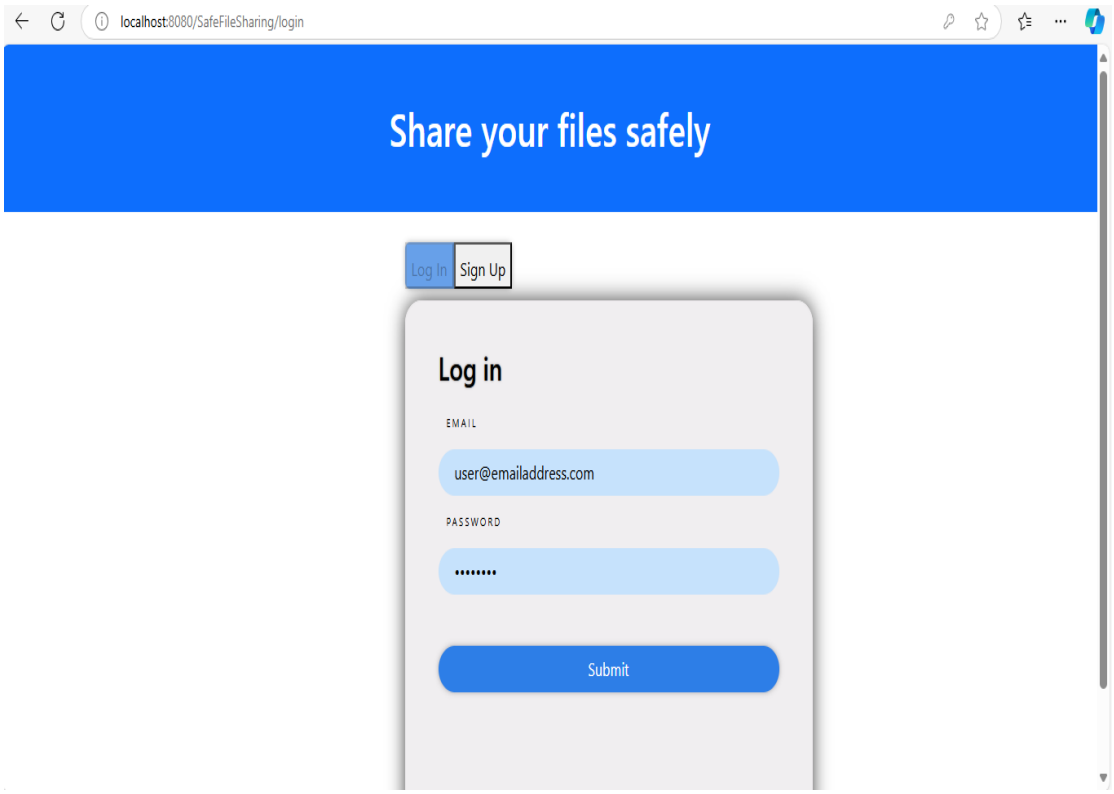
Εικόνα 12: Αρχική σελίδα νέου χρήστη.

5.2 Σύνδεση Χρήστη

Αφού ανοίξτε τον browser σας και μεταβείτε στη διεύθυνση της εφαρμογής (localhost:8080/SafeFileSharing/), επιλέγετε το κουμπί «Log in» στο πάνω μέρος της φόρμας (αν δεν είναι ήδη επιλεγμένο) και συμπληρώστε τα στοιχεία σας (email, κωδικός πρόσβασης). Έπειτα πατήστε κουμπί «Submit».

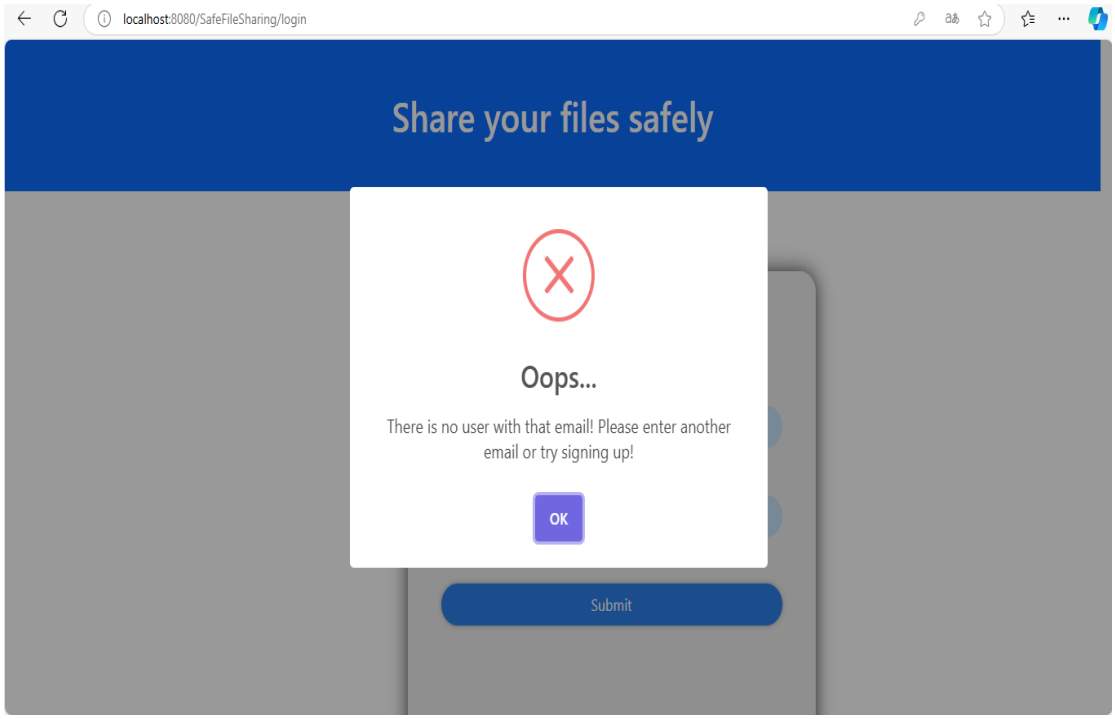


Εικόνα 13: Σελίδα σύνδεσης χρήστη.

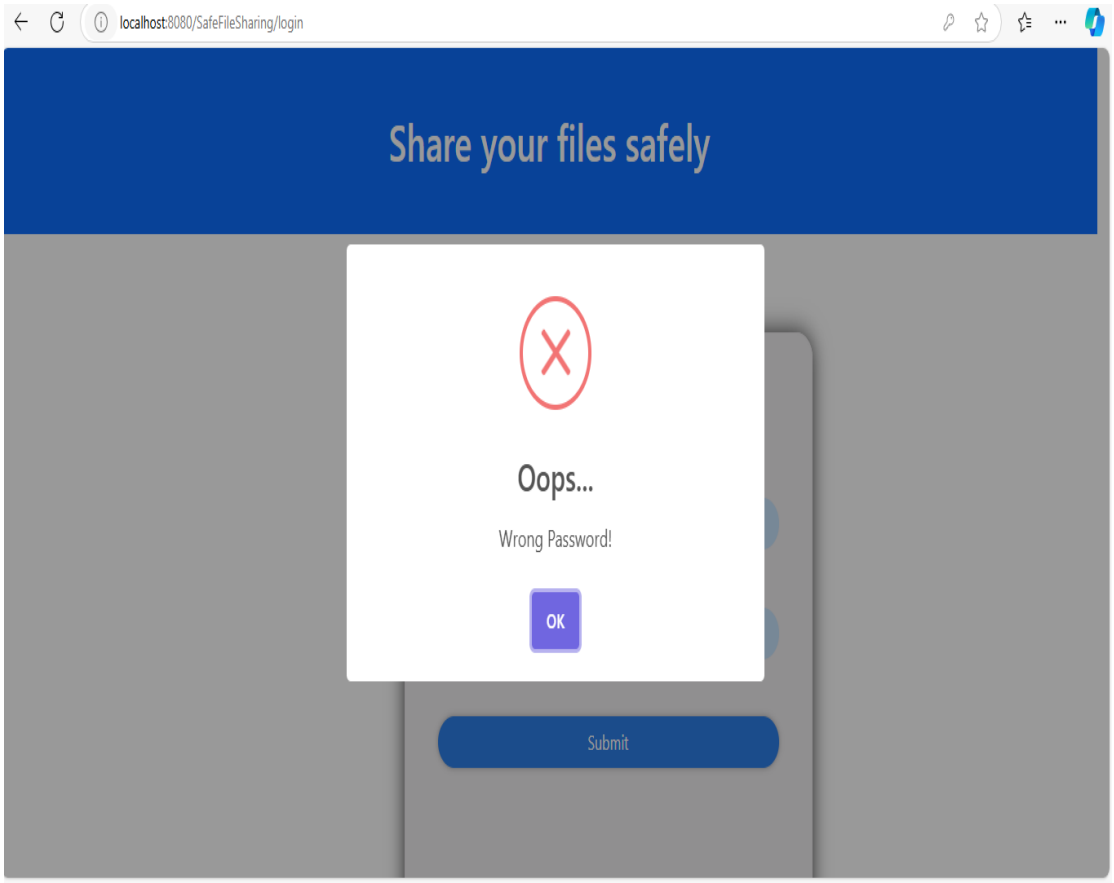


Εικόνα 14: Παράδειγμα σύνδεσης χρήστη.

Σε περίπτωση που δεν υπάρχει χρήστης με αυτό το email ή ο κωδικός είναι λανθασμένος θα ειδοποιηθείτε ώστε να ξαναπροσπαθήσετε.

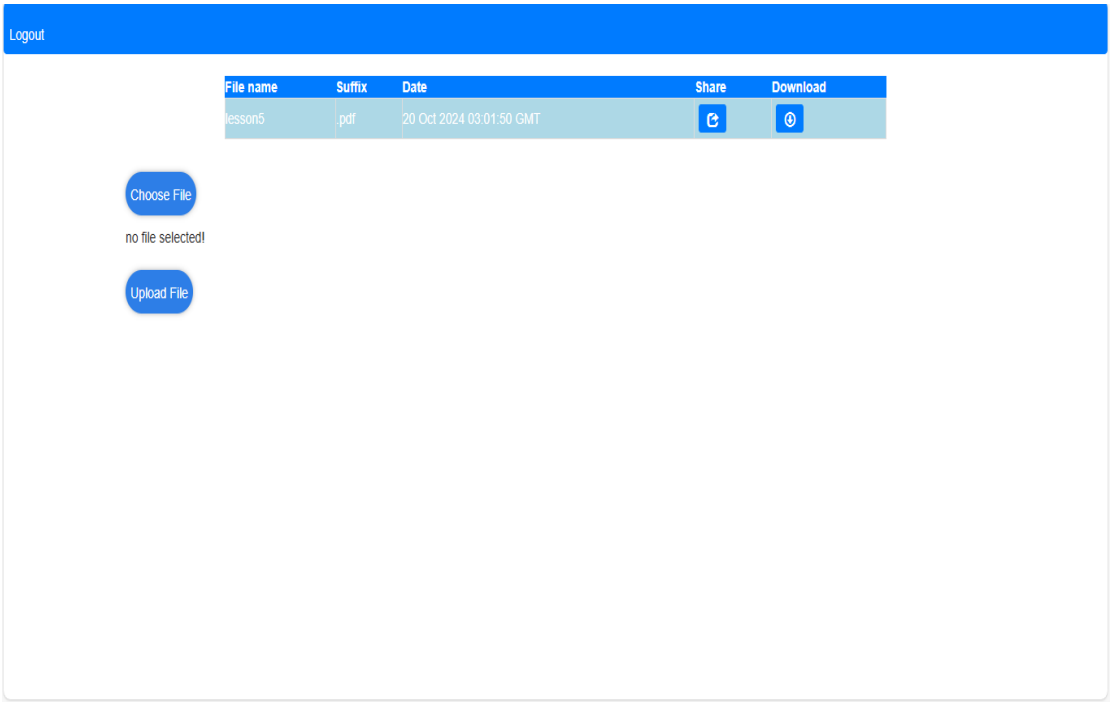


Εικόνα 15: Σελίδα σφάλματος email κατά τη σύνδεση χρήστη.



Εικόνα 16: Σελίδα σφάλματος κωδικού κατά την σύνδεσης χρήστη.

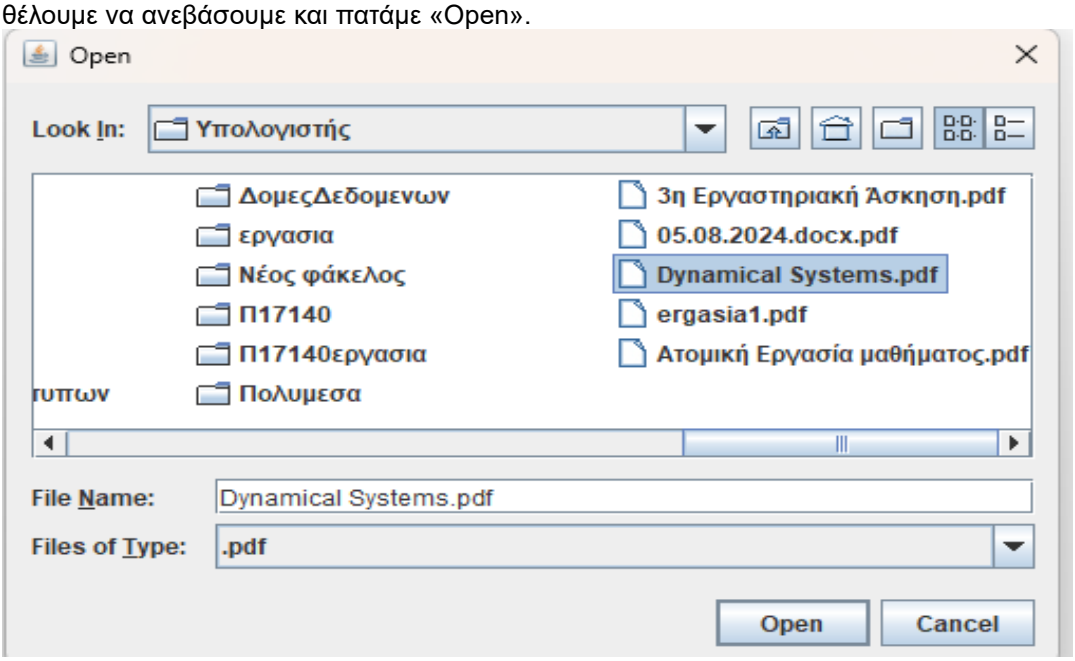
Διαφορετικά συνδέεστε στο αρχικό μενού χρήστη.



Εικόνα 17: Αρχική σελίδα χρήστη.

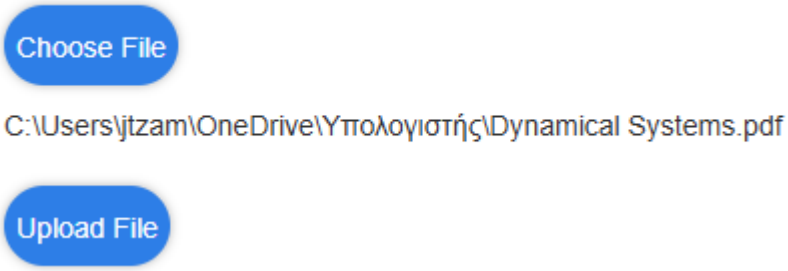
5.3 Μεταφόρτωση αρχείου

Πατάμε το κουμπί «Choose File» και διαλέγουμε από το παράθυρο επιλογής το αρχείο που θέλουμε να ανεβάσουμε και πατάμε «Open».

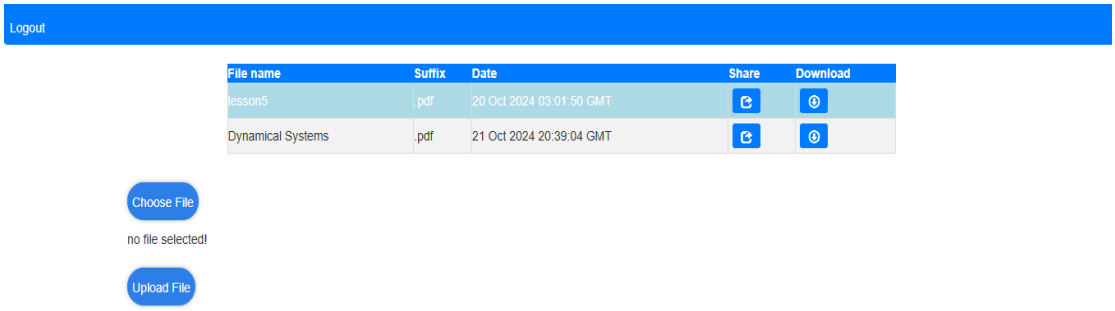


Εικόνα 18: Παράθυρο επιλογής αρχείου.

Αφού δούμε ότι έχει επιλεχθεί το αρχείο που θέλουμε να ανεβάσουμε μπορούμε να πατήσουμε «Upload File».



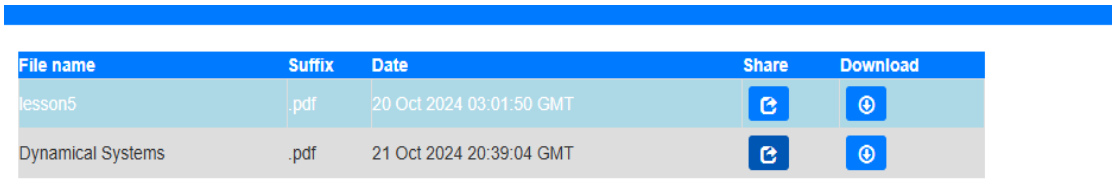
Εικόνα 19: Αλλαγή αρχικής σελίδας χρήστη μετά την επιλογή αρχείου για ανέβασμα.



Εικόνα 20: Αρχική σελίδα χρήστη μετά το ανέβασμα του επιλεγμένου αρχείου.

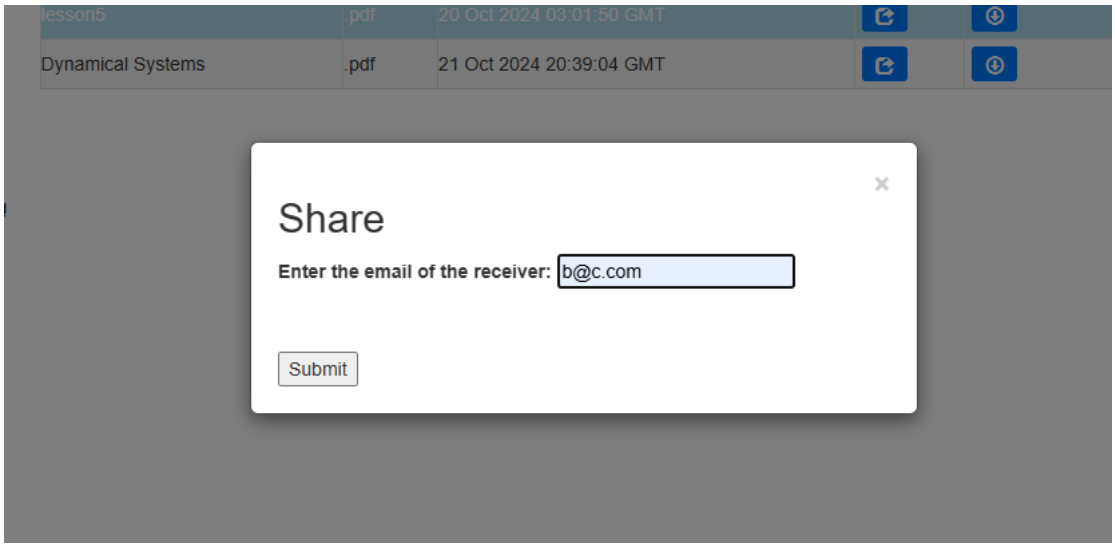
5.4 Διαμοίραση αρχείου

Επιλέγουμε το κουμπί «Share» για το αρχείο του πίνακα που θέλουμε



Εικόνα 21:Αρχεία χρήστη στην αρχική σελίδα.

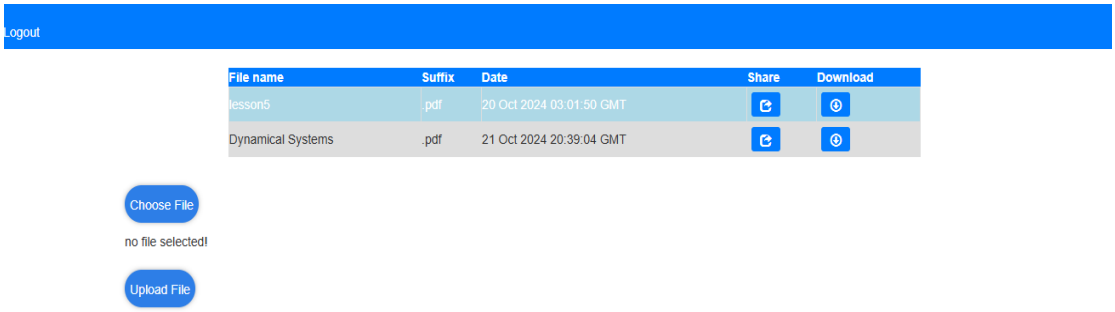
Προσθέτουμε το email του παραλήπτη και πατάμε «Submit».



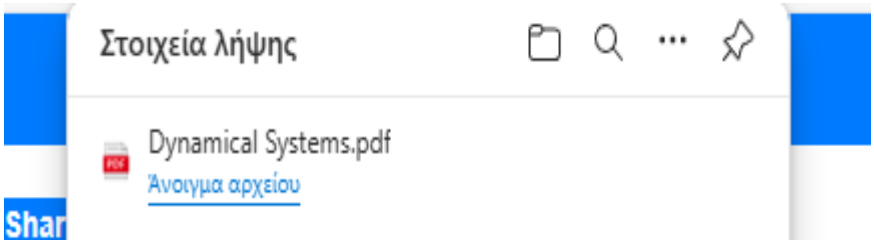
Εικόνα 22: Παράθυρο χρήστη μετά την επιλογή διαμοιρασμού αρχείου.

5.5 Λήψη αρχείου

Επιλέγουμε το κουμπί «Download» για το αρχείο του πίνακα που θέλουμε



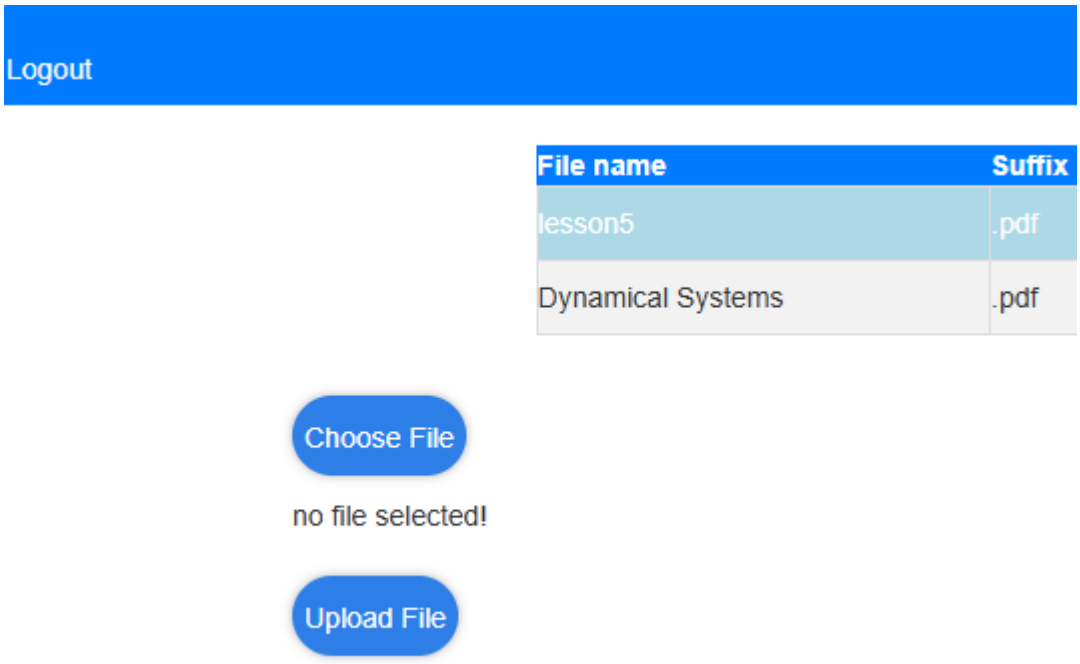
Εικόνα 23: Αρχική σελίδα χρήστη πριν την λήψη αρχείου.



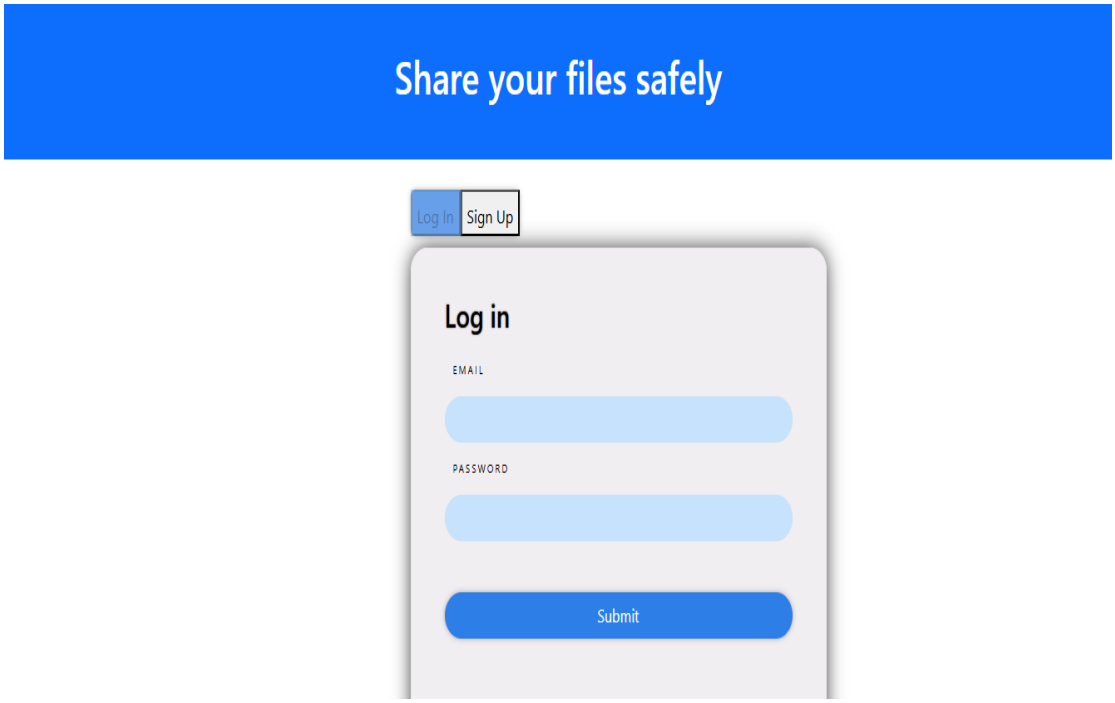
Εικόνα 24: Λήψη αρχείου.

5.6 Αποσύνδεση

Με το πάτημα του κουμπιού «Logout» (αριστερά και πάνω) αποσυνδέεστε.



Εικόνα 24: Κουμπί αποσύνδεσης στην αρχική σελίδα χρήστη.



Εικόνα 25: Επιστροφή στη σελίδα σύνδεσης μετά την αποσύνδεση.

6. Συμπεράσματα

Στην παρούσα πτυχιακή εργασία, αναπτύχθηκε ένα σύστημα ασφαλούς διαμοιρασμού αρχείων που προσφέρει μια πλήρη λύση για την κρυπτογράφηση και τον διαμοιρασμό ευαίσθητων δεδομένων μέσω διαδικτύου. Με την αξιοποίηση των τεχνολογιών Java, JSP, MongoDB και κρυπτογραφικών τεχνικών όπως το AES και RSA, επιτεύχθηκε ένα σύστημα που διασφαλίζει την ασφάλεια των δεδομένων τόσο κατά τη μεταφορά όσο και κατά την αποθήκευση.

6.1 Μελλοντικές Επεκτάσεις

Παρόλο που το σύστημα υλοποιήθηκε με επιτυχία, υπάρχουν ορισμένα σημεία που θα μπορούσαν να αναβαθμιστούν και να επεκταθούν στο μέλλον για τη βελτίωση της λειτουργικότητας και της εμπειρίας χρήστη.

1. Επέκταση σε Cloud Υποδομές
Η μεταφορά του συστήματος σε πλατφόρμες cloud (όπως το AWS ή το Google Cloud) θα επιτρέψει την κλιμάκωση του συστήματος για να υποστηρίξει περισσότερους χρήστες και μεγαλύτερους όγκους αρχείων. Επιπλέον, το cloud προσφέρει τη δυνατότητα αυτόματης δημιουργίας αντιγράφων ασφαλείας και επαναφοράς σε περίπτωση αποτυχίας.
2. Δημιουργία Αναφορών και Καταγραφής Χρήσης
Η εισαγωγή μιας λειτουργίας που επιτρέπει στους χρήστες να βλέπουν το ιστορικό των μεταφορτώσεων και διαμοιρασμών αρχείων θα βελτίωνε τη διαφάνεια και τον έλεγχο για τον χρήστη. Αυτό θα μπορούσε να συνδυαστεί με συστήματα ειδοποιήσεων για κάθε σημαντική ενέργεια, όπως την επιτυχή αποστολή ενός αρχείου.
3. Ενσωμάτωση Πιστοποίησης Δύο Παραγόντων (Two-Factor Authentication)
Η ενσωμάτωση επιπλέον επιπέδων ασφαλείας, όπως η πιστοποίηση δύο παραγόντων (2FA), θα προσέφερε στους χρήστες μεγαλύτερη ασφάλεια, ειδικά σε περιπτώσεις χρήσης όπου οι χρήστες ανταλλάσσουν ευαίσθητα ή απόρρητα δεδομένα.

6.2 Συνοψίζοντας

Το σύστημα ασφαλούς διαμοιρασμού αρχείων που αναπτύχθηκε ανταποκρίνεται στις λειτουργικές και μη λειτουργικές απαιτήσεις, προσφέροντας ένα αξιόπιστο περιβάλλον για την ασφαλή ανταλλαγή αρχείων. Ωστόσο, η εισαγωγή νέων τεχνολογιών και λειτουργιών στο μέλλον μπορεί να ενισχύσει την απόδοση και τη χρηστικότητα του συστήματος, εξασφαλίζοντας

ακόμη μεγαλύτερη ασφάλεια και επεκτασιμότητα. Οι προτεινόμενες μελλοντικές επεκτάσεις παρέχουν μια βάση για περαιτέρω βελτιώσεις και ανάπτυξη του έργου.

Πηγές

Ashtari, H. (2021, September 1). What Is Cloud Encryption? Definition, Importance, Methods, and Best Practices. Spiceworks. https://www.spiceworks.com/tech/cloud/articles/what-is-cloud-encryption/

Bengtsson, J. (2024, July 16). What is Public and Private Key in Cryptography? Nexusgroup. https://www.nexusgroup.com/what-is-public-and-private-key-in-cryptography/

Bhosale, N. (2020). Public key cryptography principles, Indira Gandhi National Tribal University, Amartantak, M.P., India: Dept. of Computer Science Class: BC.A. VI Semester Paper. https://www.igntu.ac.in/eContent/BCA-06Sem-DrNarayanBhosale-%20NETWORK%20SECURITY%20AND%20CYBER%20TECHNOLOGY-Unit3-4.pdf

Cisco (2021) What Is Encryption? Explanation and Types. (n.d.). Cisco. https://www.cisco.com/c/en/us/products/security/encryption-explained.html#~types-of-data-encryption

European Council (2016), Regulation (EU) 2016/679 of the European Parliament and of the Council, https://eur-lex.europa.eu/legal content/EN/TXT/?uri=CELEX:02016R0679-20160504

Hyeok Ju Park, & Yong Kyu Lee. (2017), Configuring Personalized Directories in File Sharing Systems. Advanced Science Letters, 23(10), 9584–9588. https://doi.org/10.1166/asl.2017.9752)

Patil, P., & BasuMallick, C. (2022, February 9). *What Is Cloud Computing? Definition, Benefits, Types, and Trends*. Spiceworks. https://www.spiceworks.com/tech/cloud/articles/what-is-cloud-computing/

Pradeep, K. V., Vijayakumar, V., & Subramaniyaswamy, V. (2019). An Efficient Framework for Sharing a File in a Secure Manner Using Asymmetric Key Distribution Management in Cloud Environment. Journal of Computer Networks and Communications, 2019, 1–8. https://doi.org/10.1155/2019/9852472

Wonderbunny, L. (2023, June 26). How to make a digital signature secure and safe? - Oneflow. Oneflow.com. https://oneflow.com/blog/make-a-digital-signature-secure/

Ασημάκης, Γ. (2015). 9-ΚΒΑΝΤΙΚΗ ΚΡΥΠΤΟΓΡΑΦΙΑ. Gasimakis.gr. https://www.gasimakis.gr/p/blog-page_11.html

Εικόνες

Εικόνα 1: Πηγή: spiceworks.com, Types of Cloud Computing
Εικόνα 2: Κατασκευή μέσω SmartArt, Είδη κρυπτογράφησης
Εικόνα 3: Κατασκευή μέσω SmartArt, Ασύμμετρη κρυπτογράφηση και αποκρυπτογράφηση
Εικόνα 4: ricoh.com, Πώς δουλεύει η ψηφιακή υπογραφή; https://www.ricoh.com.my/blogs/are-digital-signatures-secure