



**ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ**  
ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

**Σχεδιασμός και υλοποίηση Web εφαρμογής για τη διαχείριση φύλλων  
χρονοχρέωσης έργων**

**ΝΙΚΟΛΑΟΣ ΧΑΣΚΑΡΗΣ**

Αθήνα, 2024



**ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ**

**SCHOOL OF DIGITAL TECHNOLOGY**

**DEPARTMENT OF INFORMATICS AND TELEMATICS**

**Design and implementation of a web-based application for the  
management of project timesheets**

**NIKOLAOS CHASKARIS**

Athens, 2024



# **ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ**

**ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ**

**Τσερπές Κωνσταντίνος (Επιβλέπων)**

**Αναπληρωτής Καθηγητής, Πληροφορική & Τηλεματική, Χαροκόπειο**

**Δημητρακόπουλος Γεώργιος**

**Αναπληρωτής Καθηγητής, Πληροφορική & Τηλεματική, Χαροκόπειο**

**Καμαλάκης Θωμάς**

**Καθηγητής, Πληροφορική & Τηλεματική, Χαροκόπειο**

Ο ΝΙΚΟΛΑΟΣ ΧΑΣΚΑΡΗΣ

δηλώνω υπεύθυνα ότι:

- 1)** Είμαι ο κάτοχος των πνευματικών δικαιωμάτων της πρωτότυπης αυτής εργασίας και από όσο γνωρίζω η εργασία μου δε συκοφαντεί πρόσωπα, ούτε προσβάλλει τα πνευματικά δικαιώματα τρίτων.
- 2)** Αποδέχομαι ότι η ΒΚΠ μπορεί, χωρίς να αλλάξει το περιεχόμενο της εργασίας μου, να τη διαθέσει σε ηλεκτρονική μορφή μέσα από τη ψηφιακή Βιβλιοθήκη της, να την αντιγράψει σε οποιοδήποτε μέσο ή/και σε οποιοδήποτε μορφότυπο καθώς και να κρατά περισσότερα από ένα αντίγραφα για λόγους συντήρησης και ασφάλειας.
- 3)** Όπου υφίστανται δικαιώματα άλλων δημιουργών έχουν διασφαλιστεί όλες οι αναγκαίες άδειες χρήσης ενώ το αντίστοιχο υλικό είναι ευδιάκριτο στην υποβληθείσα εργασία.

## ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να εκφράσω την ειλικρινή μου ευγνωμοσύνη στον σύμβουλο της διατριβής μου, καθηγητή Τσερπέ Κωνσταντίνο, για την πολύτιμη καθοδήγηση, την ενθάρρυνση και την υπομονή του καθ' όλη τη διάρκεια αυτού του έργου. Η τεχνογνωσία και οι γνώσεις του καθηγητή Τσερπέ συνέβαλαν καθοριστικά στη διαμόρφωση της κατεύθυνσης αυτής της έρευνας.

Θα ήθελα επίσης να ευχαριστήσω τα μέλη της επιτροπής της διατριβής μου, Δημητρακόπουλο Γεώργιο και Καμαλάκη Θωμά, για τα διορατικά σχόλια και τις υποδείξεις τους. Τα σχόλιά τους με βοήθησαν να βελτιώσω τη σαφήνεια και την εστίαση της εργασίας μου.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια και τους φίλους μου για την αμέριστη υποστήριξη και ενθάρρυνσή τους καθ' όλη τη διάρκεια των σπουδών μου. Η πίστη τους σε μένα με κράτησε παρακινημένο σε δύσκολες στιγμές.

## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΕΥΧΑΡΙΣΤΙΕΣ .....	5
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ .....	6
Περίληψη στα Ελληνικά .....	8
Abstract ή Περίληψη στα Αγγλικά.....	9
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ .....	10
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ.....	11
ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ.....	12
ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ.....	13
ΚΕΦ. 1: ΕΙΣΑΓΩΓΗ .....	14
1.1: Περιβάλλον και Συνθήκες στο χώρο εργασίας .....	14
1.2: Περιγραφή Εφαρμογής.....	14
1.3: Στόχος Εργασίας.....	15
1.4: Σύνοψη Περιεχομένων Εργασίας .....	16
ΚΕΦ.2: Σχεδιασμός εφαρμογής .....	17
2.1: Δομή Εφαρμογής.....	17
2.2: Σχεδίαση Περιβάλλοντος Χρήστη .....	18
2.3: Κατηγορίες χρηστών και σενάρια χρήσης .....	20
2.4: Ανάλυση απαιτήσεων και προδιαγραφές συστήματος .....	21
2.5: Ανάλυση Βάσης Δεδομένων .....	27
ΚΕΦ.3: Υλοποίηση εφαρμογής.....	31
3.1: Backend Υπηρεσία.....	31
3.2: Βάση δεδομένων .....	31
3.3: Firebase υπηρεσία .....	32
3.4: Frontend υπηρεσία .....	33
3.5: Παρουσίαση frontend περιβάλλοντος.....	34
3.6 Εργαλεία Ανάπτυξης.....	41
3.6.1. Git.....	41
3.6.2. Visual Studio Code.....	42
3.7. Διάθεση, Χρήση και Προσαρμογή της Εφαρμογής .....	43
3.7.1. Διάθεση Εφαρμογής.....	43
3.7.2. Χρήση Εφαρμογής .....	43
3.7.3. Προσαρμογή Εφαρμογής .....	44
ΚΕΦ.4: Αξιολόγηση συστήματος/εφαρμογής.....	46
4.1 Αξιολόγηση απόδοσης του συστήματος .....	46
4.2 Συγκριτική Ανάλυση με Παρόμοιες Εφαρμογές.....	47

ΚΕΦ.5: Συμπεράσματα και Μελλοντική Εργασία.....	51
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	52

## Περίληψη στα Ελληνικά

Το θέμα της παρούσας εργασίας αφορά τον σχεδιασμό και την υλοποίηση μιας διαδικτυακής εφαρμογής, για τη διαχείριση φύλλων χρονοχρέωσης έργων, με σκοπό τη διευκόλυνση των ερευνητικών ομάδων στην καταγραφή και διαχείριση του χρόνου που αφιερώνεται σε διάφορα έργα. Οι ομάδες αυτές χρησιμοποιούν επί του παρόντος υπολογιστικά φύλλα Excel, τα οποία παρουσιάζουν περιορισμούς όσον αφορά τη συνεργασία, την ευελιξία και την ακρίβεια στη διαχείριση δεδομένων. Διαδικτυακή εφαρμογή ονομάζεται κάθε εφαρμογή η οποία είναι διαθέσιμη στους χρήστες της μέσω του Διαδικτύου (Internet) ή του ενδοδικτύου μιας εταιρίας και ο χρήστης χρειάζεται μόνο τον περιηγητή του για να την χρησιμοποιήσει.

Το αντικείμενο της εργασίας περιλαμβάνει τη δημιουργία ενός φιλικού προς τον χρήστη περιβάλλοντος, που θα επιτρέπει την εύκολη καταχώρηση, επεξεργασία και παρακολούθηση των ωρών εργασίας που αφιερώνει το κάθε μέλος της ομάδας ανά έργο. Επιπλέον, στόχος είναι η παροχή δυνατοτήτων για τη δημιουργία αναφορών και στατιστικών, που θα βοηθούν στη λήψη αποφάσεων και στην καλύτερη διαχείριση των πόρων.

Η μεθοδολογία που ακολουθήθηκε περιλαμβάνει την ανάλυση των απαιτήσεων των ερευνητικών ομάδων, τον σχεδιασμό της αρχιτεκτονικής της εφαρμογής, την ανάπτυξη της πλατφόρμας με τη χρήση σύγχρονων web τεχνολογιών και τη δοκιμαστική εφαρμογή της σε πραγματικές συνθήκες. Η υλοποίηση βασίστηκε σε μια responsive και cloud-based αρχιτεκτονική για να διασφαλιστεί η προσβασιμότητα από διαφορετικές συσκευές και την εύκολη επεκτασιμότητα.

Αρχικά, προσδιορίζονται οι ομάδες των χρηστών που θα χρησιμοποιήσουν την εν λόγω εφαρμογή, ώστε να καταγραφούν οι απαιτήσεις τους από αυτήν και εν τέλει να προκύψουν οι προδιαγραφές που πρέπει να υλοποιηθούν. Αυτές οι προδιαγραφές θα αποτελέσουν τη βάση για την ανάπτυξη μιας διεπαφής χρήστη που θα ανταποκρίνεται στις ανάγκες των χρηστών. Στο πλαίσιο της εργασίας, περιλαμβάνονται εικόνες που δείχνουν τις βασικές λειτουργίες της διεπαφής.

Εν συνεχεία, ακολουθεί διάγραμμα της βάσης δεδομένων που απεικονίζει το σύνολο των απαραίτητων πληροφοριών για αυτή τη διεπαφή. Για την υλοποίηση της εφαρμογής, χρησιμοποιήθηκαν τα εργαλεία της MongoDB, του NodeJS και του NextJS. Τέλος θα αναφερθούν μελλοντικές βελτιώσεις, για την απλοποίηση του συστήματος και την ενίσχυση της ασφάλειας.

**Λέξεις κλειδιά:** Φύλλα χρονοχρέωσης έργων, Ερευνητικές Ομάδες, Ανάλυση Απαιτήσεων, Σχεδιασμός Βάσης Δεδομένων



## Abstract ή Περίληψη στα Αγγλικά

The subject of this paper concerns the design and implementation of a web application for the management of project timesheets, aimed at facilitating research teams in recording and managing the time spent on various projects. These teams are currently using Excel spreadsheets, which present limitations in terms of collaboration, flexibility, and data management accuracy.

The objective of this work involves creating a user-friendly environment that allows easy entry, editing, and monitoring of working hours per project and team member. Additionally, the goal is to provide features for generating reports and statistics that assist in decision-making and better resource planning.

The methodology followed includes the analysis of the research teams' requirements, the design of the application's architecture, the development of the platform using modern web technologies, and its trial deployment in real-world conditions. The implementation was based on a responsive and cloud-based architecture to ensure accessibility from different devices and easy scalability.

Initially, the user groups that will use this application are identified, so that their requirements can be recorded, leading to the specifications that need to be implemented. These specifications will form the basis for creating a user interface that meets the users' goals. A set of images, illustrating the basic functions, is presented throughout the paper, depicting how the application operates.

Subsequently, the database schema is presented, capturing all the required information for this interface. To complete the design, the tools MongoDB, Node.js, and Next.js were used to implement the application. Lastly, future improvements will be mentioned, aimed at simplifying the system and enhancing its security.

**Keywords:** Project time sheets, Research teams, Requirements analysis, Database design

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Σελίδα Έργων .....	35
Εικόνα 2: Δημιουργία Έργου .....	35
Εικόνα 3: Σελίδα Συμβολαίων .....	36
Εικόνα 4: Δημιουργία Συμβολαίου .....	36
Εικόνα 5: Σελίδα Εργαζομένων .....	37
Εικόνα 6: Προσθήκη νέου εργαζομένου .....	37
Εικόνα 7: Σελίδα Διασυνδέσεων .....	38
Εικόνα 8: Δημιουργία Διασύνδεσης .....	38
Εικόνα 9: Σελίδα φύλλων χρονοχρέωσης .....	39
Εικόνα 10: Δημιουργία φύλλου χρονοχρέωσης .....	39
Εικόνα 11: Εισαγωγή ωρών εργασίας .....	40
Εικόνα 12: Προβολή χρηστών .....	40
Εικόνα 13: Δημιουργία νέου χρήστη .....	41
Εικόνα 14: Παράδειγμα Git διακλαδώσεων .....	42

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Απαίτηση - Τρόπος σύνδεσης .....	21
Πίνακας 2: Απαίτηση - Αναλυτική Αναφορά .....	22
Πίνακας 3: Απαίτηση - GANTT διάγραμμα .....	22
Πίνακας 4: Απαίτηση - Επεξεργασία Δεδομένων .....	22
Πίνακας 5: Απαίτηση - Ευκολία εισαγωγής δεδομένων .....	23
Πίνακας 6: Απαίτηση - Αναζήτηση .....	23
Πίνακας 7: Απαίτηση - Εισαγωγή φύλλων χρονοχρέωσης.....	24
Πίνακας 8: Απαίτηση - Ειδοποιήσεις.....	24
Πίνακας 9: Απαίτηση - Συνεργασία διαχειριστών .....	24
Πίνακας 10: Απαίτηση – Τρόπος αποθήκευσης .....	25
Πίνακας 11: Απαίτηση - Συνεχής λειτουργία .....	25
Πίνακας 12: Απαίτηση - Αλληλεπίδρασή χρηστών με ιστοσελίδα .....	26
Πίνακας 13: Απαίτηση - Αξιοπιστία ιστοσελίδας.....	26
Πίνακας 14: Απαίτηση - Συντήρηση και ενημέρωση ιστοσελίδας.....	26
Πίνακας 15: Απαίτηση - Τεκμηρίωση χρήσης ιστοσελίδας και παραγωγής κώδικα.....	27
Πίνακας 16: Απαίτηση - Ικανοποίηση χρηστών .....	27
Πίνακας 17: Ορισμός Πίνακα Παραδοτέων (Wps).....	28
Πίνακας 18: Ορισμός Πίνακα Έργων (Projects).....	28
Πίνακας 19: Ορισμός Πίνακα Συμβολαίων (Contracts) .....	29
Πίνακας 20: Ορισμός Πίνακα Ανθρώπινου δυναμικού(Humans) .....	29
Πίνακας 21: Ορισμός Πίνακα Φύλλων Χρονοχρέωσης(Timesheets) .....	29
Πίνακας 22: Ορισμός Πίνακα Χρήστες(Users).....	30
Πίνακας 23: Ορισμός Πίνακα Αργιών (Holidays) .....	30
Πίνακας 24: Προδιαγραφές τοπικού υπολογιστή .....	46
Πίνακας 25: Αποτελέσματα δοκιμών απόδοσης τοπικού υπολογιστή.....	46
Πίνακας 26: Προδιαγραφές διακομιστή.....	47
Πίνακας 27: Αποτελέσματα δοκιμών απόδοσης διακομιστή.....	47
Πίνακας 28: Συγκριτική Ανάλυση με Παρόμοιες Εφαρμογές .....	50

## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1: Διάγραμμα αρχιτεκτονικής συστήματος .....	18
--	----

## ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

BaaS	Backend as a Service
HTTP	Hypertext Transfer Protocol
API	Application Programming Interface
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
DOM	Document Object Model
SOC 2	Service Organization Control 2

## **ΚΕΦ. 1: ΕΙΣΑΓΩΓΗ**

### **1.1: Περιβάλλον και Συνθήκες στο χώρο εργασίας**

Χάρη στις τεχνολογίες αιχμής, οι εργασιακές συνθήκες και το περιβάλλον στα οποία εργάζονται οι ερευνητικές ομάδες, έχουν αναπτυχθεί σημαντικά, και έχει διευκολυνθεί η διαχείριση των έργων. Την τρέχουσα εποχή, οι ερευνητικές ομάδες είναι επιβαρυνμένες με την διαχείριση πολλών έργων ταυτόχρονα, απαιτώντας την συνεχή οργάνωση του χρόνου και των διαθέσιμων πόρων. Η σύνθετη δομή αυτή προϋποθέτει εργαλεία και συστήματα για την αποδοτική διαχείριση των έργων, για την κατανομή των πόρων και για τη έγκαιρη ολοκλήρωση των εργασιών. Επομένως, τα συστήματα διαχείρισης φύλλων χρονοχρέωσης έργων γίνονται όλο και περισσότερο αναγκαία για την αποτελεσματική και σωστή καταγραφή των δεδομένων.

Έως τώρα, πολλές ερευνητικές ομάδες αντιμετωπίζουν προβλήματα στην παρακολούθηση του χρόνου ολοκλήρωσης διαφορετικών έργων και εργασιών. Οι παραδοσιακές μέθοδοι, όπως τα λογιστικά φύλλα, είναι συχνά μη αποδοτικές και επιρρεπείς σε ανθρώπινα λάθη. Το παραπάνω σύστημα μπορεί να οδηγήσει σε λάθη όσον αφορά τον χρόνο, με αποτέλεσμα να δυσκολεύει η σωστή αξιολόγηση της προσπάθειας που αναλώνεται σε κάθε έργο. Πέρα από την διαχείριση των έργων, το παραπάνω φαινόμενο επηρεάζει και την κατανομή των πόρων, καθώς τα πλήρη δεδομένα είναι αναγκαία για επικυρωμένες αποφάσεις.

Το περιβάλλον στο χώρο εργασίας των ερευνητικών ομάδων συντελεί στον καθορισμό των λειτουργικών και μη απαιτήσεων της εφαρμογής. Οι ερευνητικές ομάδες λειτουργούν στα πλαίσια πανεπιστημίων, όπου στοχεύουν περισσότερο σε επιστημονικές επιδιώξεις παρά σε διοικητικά καθήκοντα. Σε τέτοια περιβάλλοντα πολλαπλές ομάδες εργάζονται ταυτόχρονα σε διάφορα έργα, συχνά σε διαφορετικούς τομείς και σε διαφορετικές τοποθεσίες. Μία εφαρμογή για την διαχείριση φύλλων χρονοχρέωσης έργων που λειτουργεί σε πραγματικό χρόνο μπορεί να βελτιώσει τις καθημερινές εργασίες των ερευνητικών ομάδων.

Επιπλέον, οι συνθήκες υπό τις οποίες λειτουργούν οι ερευνητικές ομάδες συχνά περιλαμβάνουν βραχυχρόνιες προθεσμίες και περιορισμένη χρηματοδότηση. Έτσι, οι ερευνητές πρέπει να διαχειρίζονται ορθά των χρόνο τους μεταξύ των εργασιών, και την καθοδήγηση νέων μελών, καθιστώντας απαραίτητη την ευχρηστία και την αποτελεσματικότητα του οποιουδήποτε εργαλείου διαχείρισης χρόνου. Η εφαρμογή θα έχει την δυνατότητα υποστήριξης διαφορετικών ειδών έργων, είτε αυτά είναι βραχυπρόθεσμα είτε είναι μακροχρόνια, υπολογίζοντας πάντα τον παράγοντα της μοναδικότητας του καθενός έργου.

Η ανάγκη επίλυσης αυτών των προβλημάτων οδήγησε στην δημιουργία της παρούσας διατριβής. Η ανάπτυξη μιας εφαρμογής που περιέχει όλες τις παραπάνω λειτουργίες, έχει ως σκοπό την δημιουργία ενός φιλικού και εύχρηστου περιβάλλοντος για τον χρήστη.

### **1.2: Περιγραφή Εφαρμογής**

Η εφαρμογή διαχείρισης φύλλων χρονοχρέωσης και τιμολόγησης έργων στοχεύει στην απλοποίηση και στη βελτιστοποίηση της διαδικασίας της καταγραφής, της παρακολούθησης και των αξιολογήσεων των ωρών εργασίας της ερευνητικής ομάδας. Μέσω ενός φιλικού και εύχρηστου, προς τον χρήστη περιβάλλοντος, η χρήση της εφαρμογής θα επιτρέπει στους

εργαζόμενους την εύκολη διαχείριση των παραπάνω. Με την δυσκολία διαχείρισης πολλαπλών έργων να αυξάνεται συνεχώς, η ανάγκη για το παραπάνω σύστημα είναι απόλυτα απαραίτητη. Το σύστημα αυτό εγγυάται την καταγραφή του χρόνου λεπτομερώς, παρέχοντας αναγκαία δεδομένα για την διαχείριση των έργων και την οικονομική σχεδίαση.

Από τη σκοπιά του χρήστη, η εφαρμογή προσφέρει έναν απλό τρόπο εισαγωγής των ημερήσιων ωρών που αφιερώνονται σε κάθε έργο. Οι διαχειριστές μπορούν να καταγράψουν τον χρόνο εργασίας των μελών της ομάδας σε δομημένη μορφή, για παράδειγμα σε μορφή ημερολογίου, και την συσχέτιση της εργασίας με συγκεκριμένα παραδοτέα έργων, συμβολαίων κλπ.

Για τους διευθυντές και τους διαχειριστές των ερευνητικών ομάδων, η εφαρμογή παρέχει ωφέλιμα εργαλεία για την εποπτεία των χρονοδιαγραμμάτων και των πόρων των έργων. Παρέχονται λειτουργίες αναφορών, διευκολύνοντας έτσι τους διαχειριστές με την συγκέντρωση των δεδομένων σχετικά με τις συνολικές ώρες απασχόλησης, με την αξιολόγηση της εξέλιξης των έργων και με την εφαρμογή του οικονομικού σχεδιασμού. Επίσης, εξαλείφεται ως ένα βαθμό η εισαγωγή δεδομένων χειροκίνητα, όπως θα ήταν στα λογιστικά φύλλα (Excel), κάνοντας έτσι όλη την διαδικασία πιο φιλική και αυτοματοποιημένη, μειώνοντας τυχόν λάθη και ώρες εργασίας.

Το σύστημα έχει κατασκευαστεί με τέτοια αρχιτεκτονική και τεχνολογίες ώστε να μπορεί να επεκταθεί εύκολα και να είναι ασφαλές. Παρέχει μία ανανεωμένη λύση στους οργανισμούς και στις ερευνητικές ομάδες, απλοποιώντας τις διαδικασίες έργων και φύλλων χρονοχρέωσης. Συνολικά, το σύστημα στοχεύει στην αύξηση της παραγωγικότητας, της διαφάνειας των ενεργειών των διαχειριστών και της υπευθυνότητας που φέρουν οι ενέργειες αυτές.

### **1.3: Στόχος Εργασίας**

Ο σκοπός του παρόντος έργου είναι ο σχεδιασμός και η υλοποίηση μιας διαδικτυακής εφαρμογής για τη διαχείριση των φύλλων χρονοχρέωσης, έπειτα από αναλυτικές μελέτες των απαιτήσεων των τελικών χρηστών. Στόχος της εν λόγω εργασίας είναι η δημιουργία ενός συστήματος που δίνει την δυνατότητα στους διαχειριστές να επεξεργάζονται με ευκολία τις ώρες εργασίας των ομάδων τους πιο αποτελεσματικά. Οι διαχειριστές με την ευκολία στην παρακολούθηση των ωρών που αναλώνουν οι εργαζόμενοι σε κάθε έργο, μπορούν να συντονίσουν τους διαθέσιμους πόρους με μεγαλύτερη ευελιξία. Η εφαρμογή θα φροντίσει πέρα από την ανάγκη των χρηστών για εύκολη διαχείριση των φύλλων χρονοχρέωσης, να δημιουργήσει επίσης ένα φιλικό και εύχρηστο περιβάλλον ώστε να διευκολυνθεί η παραπάνω οργάνωση.

Μέσω της ανάλυσης των απαιτήσεων των χρηστών οι προδιαγραφές που πρέπει να υλοποιήσει το σύστημα, όπως ο σχεδιασμός της διεπαφής χρήστη, οι δυνατότητες επεξεργασίας δεδομένων και η αποθήκευση πληροφοριών, γίνονται εμφανείς. Η διασύνδεση μεταξύ του χρήστη και του συστήματος θα βελτιστοποιηθεί, ώστε λειτουργίες όπως η αποθήκευση/επεξεργασία δεδομένων, οι καταχωρήσεις χρόνου και οι διασυνδέσεις μεταξύ των χρηστών να παρέχονται έως ένα ικανοποιητικό επίπεδο.

Επιπλέον, θα εξεταστεί ο σχεδιασμός της αρχιτεκτονικής των πληροφοριών της εφαρμογής, διασφαλίζοντας ότι η υποκείμενη βάση δεδομένων και το σχήμα της μπορούν να χειριστούν

δυναμικά, διαρκώς μεταβαλλόμενα δεδομένα. Αυτή η αρχιτεκτονική θα υποστηρίζει την αποτελεσματική ροή δεδομένων μεταξύ των χρηστών και του συστήματος, επιτρέποντας την απρόσκοπτη δημιουργία, την διαχείριση και την κοινή χρήση των πληροφοριών του φύλλου εργασίας σε πολλαπλά έργα και τμήματα.

Επιπλέον, το έργο θα παρουσιάσει τις τεχνικές απαιτήσεις για κάθε στοιχείο του συστήματος. Αυτό θα παρέχει μια σαφή κατανόηση του τρόπου με τον οποίο οι χρήστες θα αλληλεπιδρούν με την εφαρμογή και τον τρόπο με τον οποίο το σύστημα θα υποστηρίζει τη ροή της εργασίας τους.

Τέλος, θα παρουσιαστούν οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη κάθε συνιστώσας της εφαρμογής, παράλληλα με εικόνες που απεικονίζουν τη διάταξη και τη λειτουργικότητα της διεπαφής.

## **1.4: Σύνοψη Περιεχομένων Εργασίας**

Η εργασία διαρθρώνεται σε 4 συνολικά κεφάλαια, με την εξής μορφή:

1. Κεφάλαιο 1: Εισαγωγή: Λειτουργεί ως πρόλογος της εργασίας, αναφέροντας τους στόχους που φιλοδοξεί να επιτύχει, το αντικείμενο μελέτης και η μεθοδολογία που ακολούθησε.
2. Κεφάλαιο 2: Σχεδιασμός Εφαρμογής: Εδώ αναλύεται ο τρόπος σχεδίασης της εφαρμογής.
3. Κεφάλαιο 3: Υλοποίηση Εφαρμογής: Παρουσιάζονται αναλυτικά τα εργαλεία υλοποίησης της κάθε συνιστώσας της εφαρμογής, προσαρμοσμένα στον σχεδιασμό που έχει προηγηθεί.
4. Κεφάλαιο 4: Συμπεράσματα και Μελλοντική εργασία: Περιλαμβάνει τα τελικά συμπεράσματα της εργασίας, παρουσιάζοντας συμπληρωματικά ιδέες για μελλοντική ανάπτυξη και βελτίωση.



## ΚΕΦ.2: Σχεδιασμός εφαρμογής

### 2.1: Δομή Εφαρμογής

Η εφαρμογή θα ακολουθεί την αρχιτεκτονική client-server, όπου ο client (frontend) θα είναι υπεύθυνος για την αλληλεπίδραση με τον χρήστη, ενώ ο server (backend) θα διαχειρίζεται τα δεδομένα και θα εκτελεί τους απαραίτητους υπολογισμούς. Η αρχιτεκτονική αυτή επιτρέπει την καλύτερη κλιμάκωση και την πιο αποδοτική διαχείριση των πόρων. Επιπλέον, διασφαλίζει την ευελιξία του συστήματος, καθώς ο frontend και ο backend μπορούν να αναπτυχθούν ανεξάρτητα και να επικοινωνούν μέσω HTTP/HTTPS και RESTful API.

Το frontend της εφαρμογής θα είναι υπεύθυνο για την αλληλεπίδραση με τον χρήστη και θα παρέχει μια φιλική και εύχρηστη διεπαφή για τη διαχείριση των φύλλων χρονοχρέωσης. Θα αναπτυχθεί με χρήση τεχνολογιών HTML, CSS, και Typescript, ενώ θα βασιστεί σε σύγχρονο framework, την NextJS, που επιτρέπουν τη δημιουργία δυναμικών και διαδραστικών διεπαφών. Η διεπαφή θα παρέχει δυνατότητες όπως η εισαγωγή δεδομένων, η προβολή στατιστικών στοιχείων και η αναζήτηση παλαιότερων χρεώσεων.

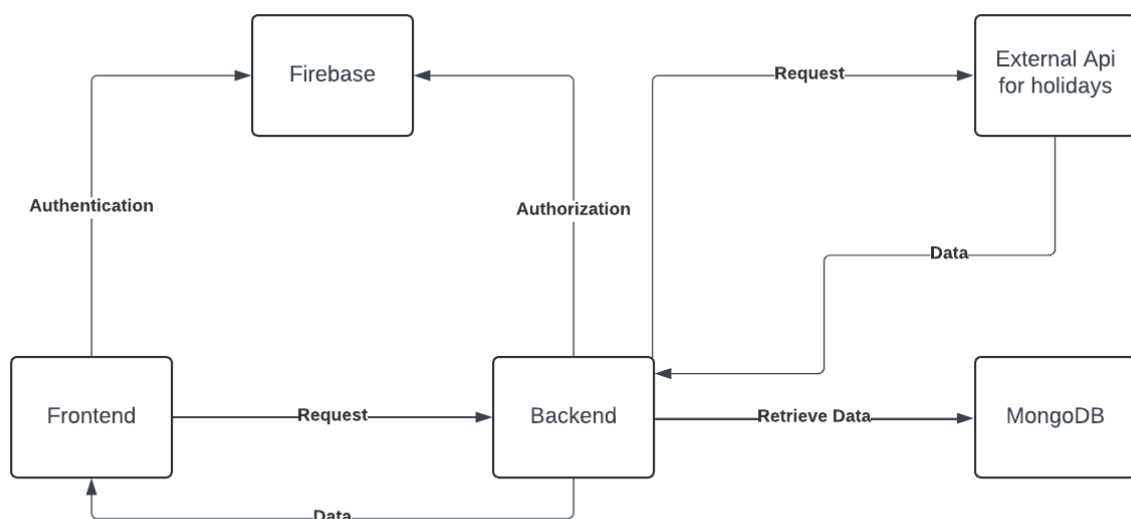
Το backend της εφαρμογής θα είναι υπεύθυνο για την αποθήκευση και επεξεργασία των δεδομένων, καθώς και για την επικοινωνία με τη βάση δεδομένων. Θα αναπτυχθεί με χρήση NodeJS και συγκεκριμένα με το framework NestJS, τα οποία εργαλεία υποστηρίζουν τη δημιουργία ασφαλών και αποδοτικών web εφαρμογών. Το backend θα περιλαμβάνει API που θα δέχεται αιτήσεις από το frontend, θα πραγματοποιεί τις απαραίτητες λειτουργίες και θα επιστρέφει τα δεδομένα στον χρήστη. Επίσης θα επικοινωνεί το backend με ένα εξωτερικό RESTful API που είναι υπεύθυνο για την ανάκτηση των αργιών.

Η βάση δεδομένων θα είναι υπεύθυνη για την αποθήκευση των στοιχείων που αφορούν τα έργα, τις χρεώσεις και τις χρονικές καταγραφές. Μια μη σχεσιακή βάση δεδομένων, MongoDB, θα επιλεγεί για την αποθήκευση δεδομένων, όπως τα φύλλα χρονοχρέωσης, οι χρήστες και οι σχετικοί πόροι. Η δομή της βάσης δεδομένων θα περιλαμβάνει πίνακες για τα έργα, τους υπαλλήλους, τις καταγραφές χρονοχρέωσης, τα συμβόλαια, τις αργίες, του ανθρώπους που εργάζονται στην ομάδα και τα παραδοτέα κάθε έργου. Θα δοθεί ιδιαίτερη προσοχή στη σωστή ομαλοποίηση των δεδομένων, ώστε να αποφεύγεται η περιττή επανάληψη και να διασφαλίζεται η ακεραιότητα τους.

Για την εξασφάλιση της ασφάλειας τόσο στο frontend αλλά και τόσο στο backend κομμάτι της εφαρμογής θα χρησιμοποιηθεί η υπηρεσία Firebase Authentication. Η εξουσιοδότηση και επαλήθευση της ταυτότητας θα γίνονται μέσω του Google sign in. Το frontend μόλις ο χρήστης συνδεθεί θα στέλνει τον ηλεκτρονικό λογαριασμό στη Firebase και αν είναι χρήστης της εφαρμογής, θα επιστρέφεται ένα μοναδικό σύμβολο. Όταν ο χρήστης θέλει να κάνει την οποιαδήποτε ενέργεια θα στέλνεται το μοναδικό σύμβολο αυτό στο backend όπου γίνεται ταυτοποίηση ξανά με το firebase για την διασφάλιση της ασφάλειας.

Η επικοινωνία μεταξύ του frontend και του backend θα γίνεται μέσω RESTful API, όπου το frontend θα στέλνει αιτήσεις HTTP (GET, POST, PUT, DELETE) στο backend για τη λήψη ή την ενημέρωση των δεδομένων. Το backend θα επεξεργάζεται τις αιτήσεις αυτές και θα επιστρέφει τα δεδομένα σε μορφή JSON. Αυτή η προσέγγιση προσφέρει μια σαφή δομή για την επικοινωνία μεταξύ των δύο μερών της εφαρμογής και επιτρέπει την επέκταση ή την τροποποίηση μελλοντικά, χωρίς σημαντικές αλλαγές στην υποδομή.

Το κεφάλαιο αυτό, λοιπόν, συνδυάζει αρμονικά το frontend, το backend και τη βάση δεδομένων, δημιουργώντας μια ολοκληρωμένη και ασφαλή λύση για τη διαχείριση φύλλων χρονοχρέωσης. Κάθε στοιχείο της αρχιτεκτονικής έχει σαφή ρόλο και λειτουργεί συνεργατικά, ώστε να παρέχει μια απρόσκοπτη εμπειρία στον χρήστη. Το frontend προσφέρει ένα εύχρηστο περιβάλλον αλληλεπίδρασης, το backend εξασφαλίζει την ορθή διαχείριση των δεδομένων και την ασφαλή επικοινωνία, ενώ η βάση δεδομένων οργανώνει τα δεδομένα με τρόπο που να είναι εύκολα προσβάσιμα και αναλυτικά. Η δομή αυτή επιτρέπει την επέκταση και βελτίωση της εφαρμογής στο μέλλον, διατηρώντας παράλληλα τη σταθερότητα και την απόδοση της.



Σχήμα 1: Διάγραμμα αρχιτεκτονικής συστήματος

## 2.2: Σχεδίαση Περιβάλλοντος Χρήστη

Η σχεδίαση του περιβάλλοντος χρήστη βασίζεται σε αρχές που στοχεύουν στη βέλτιστη εμπειρία χρήσης (User Experience - UX). Αυτές περιλαμβάνουν την απλότητα, την ευκολία πλοήγησης, τη σαφήνεια της πληροφορίας και την ταχύτητα αλληλεπίδρασης. Η δομή της διεπαφής θα είναι όσο το δυνατόν πιο καθαρή και απλή, με στόχο να διευκολύνει την εκτέλεση καθημερινών λειτουργιών, όπως η καταχώρηση χρόνου, η προβολή ιστορικών δεδομένων και η επεξεργασία πληροφοριών. Οι χρήστες θα πρέπει να έχουν την δυνατότητα να κατανοήσουν αμέσως πώς να χρησιμοποιήσουν την εφαρμογή, χωρίς περιττές διαδικασίες ή περίπλοκα μενού.

Η πλοήγηση στην εφαρμογή θα είναι ξεκάθαρη και οργανωμένη, με κεντρικό μενού που θα επιτρέπει άμεση πρόσβαση στις βασικές λειτουργίες. Οι χρήστες θα έχουν τη δυνατότητα να περιηγηθούν σε βασικές ενότητες όπως τα Έργα, τα Φύλλα Χρονοχρέωση, τα Συμβόλαια, το ανθρώπινο δυναμικό και τέλος τα Συνεργαζόμενα Άτομα. Η γραμμή πλοήγησης θα διευκολύνει την πρόσβαση στις υποενότητες και σε παλιότερες εγγραφές, ενώ οι χρήστες θα μπορούν εύκολα να επιστρέφουν σε προηγούμενες σελίδες. Η δομή αυτή συμβάλλει στην καλύτερη ροή εργασίας, εξοικονομώντας χρόνο και μειώνοντας την πολυπλοκότητα.

Οι κύριες σελίδες της εφαρμογής, όπως η σελίδα καταχώρησης χρονοχρέωσης, η σελίδα προβολής έργων και η οθόνη συμβολαίων, θα σχεδιαστούν με βάση τη λειτουργικότητα και την ευχρηστία. Στη σελίδα καταχώρησης χρονοχρέωσης, ο χρήστης θα μπορεί να προσθέτει ώρες εργασίας για συγκεκριμένα έργα. Οι φόρμες θα είναι απλές και κατηγοριοποιημένες, ώστε ο χρήστης να βλέπει μόνο τις απαραίτητες πληροφορίες. Στη σελίδα προβολής έργων, θα παρουσιάζονται όλα τα έργα μαζί με ένα διάγραμμα GANTT, ενώ η οθόνη συμβολαίων θα παρέχει εύκολη σύνοψη όλων των συμβολαίων με τις απαραίτητες πληροφορίες.

Η εφαρμογή σχεδιάστηκε με πρωταρχικό σκοπό τη συμβατότητα σε πολλαπλές συσκευές (responsive design). Αυτό σημαίνει ότι η διεπαφή θα προσαρμόζεται αυτόματα στις διαστάσεις της οθόνης του χρήστη, είτε πρόκειται για desktop, tablet ή κινητό τηλέφωνο. Τα στοιχεία της διεπαφής θα αλλάζουν μορφή ανάλογα με το μέγεθος της οθόνης, ώστε να διασφαλίζεται η ευκολία στη χρήση, ανεξάρτητα από τη συσκευή που χρησιμοποιείται. Για την σχεδίαση αυτή μελετήθηκαν οι ευρετικές αξιολογήσεις του Nielsen οι οποίες είναι οι εξής:

1. Ορατότητα συστήματος: Το σύστημα πρέπει να ενημερώνει τους χρήστες για οποιαδήποτε ενέργεια η οποία γίνεται στο υπόβαθρο, σε ένα ευνοϊκό σύντομο χρόνο από την έναρξη της ενέργειας αυτής.
2. Ομοιότητες μεταξύ του πραγματικού κόσμου και της εφαρμογής: Το σύστημα πρέπει να βρίσκεται σε γλώσσα και διάλεκτο την οποία καταλαβαίνει ο χρήστης. Χρήση λέξεων, προτάσεων και ιδεολογιών που ο χρήστης βρίσκει οικία χωρίς την προσθήκη προγραμματιστικών ορολογιών. Η εμφάνιση των λέξεων προτάσεων και ιδεολογιών πρέπει να γίνεται με λογική σειρά ώστε να είναι εύκολη στην κατανόηση.
3. Ελευθερία στον χειρισμό της εφαρμογής από τον χρήστη: Ο χρήστης θα πρέπει να μπορεί με ευκολία να απομακρυνθεί από μία ενέργεια που έγινε κατά λάθος χωρίς να χρειάζεται να παρέμβει σε μία εκτενής και κουραστική διαδικασία.
4. Συνοχή και Προδιαγραφές: Ο χρήστης πέρα από τη εφαρμογή αυτή, αξιοποιεί τον χρόνο του και σε πολλές άλλες. Οι εμπειρίες με αυτές προδιαθέτουν τον χρήστη στο να έχει κάποιες προσδοκίες, οπότε η εφαρμογή θα πρέπει να συμβαδίζει με την γενική σχεδιαστική λογική στην αγορά.
5. Αναγνώριση αντί για Ανάκληση: Πληροφορίες οι οποίες χρησιμοποιούνται ξανά από τον χρήστη σε διαφορετικά σημεία πρέπει να εμφανίζονται αυτόματα χωρίς τον χρήστη να πρέπει να τα θυμάται, δεδομένου ότι οι άνθρωποι έχουν περιορισμένη βραχυπρόθεσμη μνήμη.
6. Ευελιξία και αποτελεσματικότητα χρήσης: Παρακάμψεις κρυμμένες, από τους αρχάριους χρήστες, μπορούν να βοηθήσουν αυτούς με περισσότερη εμπειρία

επιταχύνοντας την οποιαδήποτε διαδικασία τους. Η επιλογή για την παραμετροποίηση διάφορων ενεργειών από τους χρήστες για την ατομική διευκόλυνση τους είναι επίσης ευνοϊκή.

7. Αισθητικός και Μινιμαλιστικός Σχεδιασμός: Οι διεπαφές δεν πρέπει να περιέχουν άχρηστες πληροφορίες ή πληροφορίες που χρησιμοποιούνται/χρειάζονται σπάνια. Κάθε έξτρα κομμάτι τέτοιας πληροφορίας ανταγωνίζεται τις σχετικές πληροφορίες με αποτέλεσμα να χάνουν την σημαντικότητα τους.
8. Βοήθεια στους χρήστες να αναγνωρίσουν, να διαγνώσουν και να ανακτήσουν από σφάλματα: Τα μηνύματα σφαλμάτων θα πρέπει να είναι αποτυπωμένα με κανονική διάλεκτο χωρίς την ύπαρξη μοναδικών κωδικών προβλημάτων. Θα πρέπει να αποτυπώνουν το πρόβλημα με σαφήνεια και να δίνουν ξεκάθαρες οδηγίες στους χρήστες για τα επόμενα βήματα.
9. Τεκμηρίωση: Είναι προτιμότερο αν το σύστημα είναι σχεδιασμένο με τέτοιο τρόπο που δεν χρειάζεται επιπλέον τεκμηρίωση. Στις περιπτώσεις που αυτό όμως δεν είναι εφικτό, είναι πολύ σημαντικό να υπάρχει τεκμηρίωση αν όχι για όλη την εφαρμογή, τουλάχιστον για κριτικά σημεία της.

## **2.3: Κατηγορίες χρηστών και σενάρια χρήσης**

Η εφαρμογή έχει ως σκοπό να βελτιώσει και να διευκολύνει την εμπειρία των διαχειριστών των ερευνητικών ομάδων. Ένας διαχειριστής χρησιμοποιώντας την εφαρμογή, θα έχει την δυνατότητα να συγκεντρώσει τις συλλεγμένες πληροφορίες σε ένα σημείο. Παρακάτω παρουσιάζονται δύο σενάρια χρήσης από τις πιο σημαντικές λειτουργίες.

Ο διαχειριστής, αφού συνδεθεί στην εφαρμογή, έχει την επιλογή να δημιουργήσει ένα νέο ερευνητικό έργο, μέσω της σελίδας προβολής όλων των έργων. Κατά την δημιουργία του έργου θα προσθέσει τις πληροφορίες, το μοναδικό χαρακτηριστικό του έργου, τον τίτλο του, την περιγραφή του, την ημερομηνία έναρξης, την διάρκεια σε μήνες και τέλος τουλάχιστον ένα παραδοτέο το οποίο έχει και αυτό τίτλο, ημερομηνία έναρξης και διάρκεια σε μήνες. Μόλις ολοκληρωθεί η διαδικασία και τα στοιχεία που εισήχθησαν θεωρηθούν σωστά, το έργο πλέον θα εμφανίζεται στη σελίδα προβολής έργων.

Παρόμοια ο χρήστης στην επιλογή της δημιουργίας νέου συμβολαίου, θα πρέπει να έχει ήδη δημιουργήσει έναν εργαζόμενο και ένα έργο για να μπορέσει να τους εισάγει στο συμβόλαιο μαζί με την αρχή του συμβολαίου, την διάρκεια του, το ποσό χρέωσης ανά ώρα του εργαζομένου και το συνολικό κόστος του συμβολαίου. Αν δεν υπάρχει σφάλμα μεταξύ της ωριαίας χρέωσης και του συνολικού κόστους, το συμβόλαιο θα εμφανίζεται στη σελίδα προβολής συμβολαίων.

Ακόμα, ύψιστης σημασίας σενάριο χρήσης είναι η δημιουργία ενός φύλλου χρονοχρέωσης. Προϋποθέτει ότι ο χρήστης έχει ήδη δημιουργήσει ένα συμβόλαιο για κάποιο άτομο και επιχειρεί να προσθέσει τις ώρες εργασίες του για τον τρέχοντα μήνα του εργαζομένου. Τα σαββατοκύριακα και οι αργίες είναι απενεργοποιημένες εξ αρχής και πρέπει να επιλέξει ο χρήστης ρητά για να τις ενεργοποιήσει. Έπειτα επιλέγει κάθε μέρα και προσθέτει ανάλογα σε ποιο παραδοτέο δούλεψε το άτομο τις ώρες του. Ο συνολικός αριθμός των ωρών δεν μπορεί να

υπερβαίνει τις 143. Μόλις ο χρήστης επικυρώσει το φύλλο, μπορεί να το αναζητήσει στην σελίδα προβολής φύλλων χρονομέτρησης.

## 2.4: Ανάλυση απαιτήσεων και προδιαγραφές συστήματος

Με βάση τα σενάρια χρήσης και έχοντας υπόψη την σχεδίαση του περιβάλλοντος χρήστη, μπορούμε να αντιληφθούμε τις απαιτήσεις και τις προδιαγραφές του συστήματος. Η διαδικασία συλλογής και καταγραφής των απαιτήσεων βασίζεται στην αλληλεπίδραση με τους εμπλεκόμενους χρήστες και στον προσδιορισμό των βασικών λειτουργιών που θα προσφέρει η εφαρμογή. Έτσι λοιπόν μετά από συναντήσεις με τον κύριο Τσερπέ, καταγράφηκαν όλες οι απαιτήσεις και χωρίστηκαν σε λειτουργικές και μη λειτουργικές. Οι λειτουργικές προσδιορίζουν τις βασικές λειτουργίες που θα προσφέρει η εφαρμογή. Οι μη λειτουργικές αφορούν την απόδοση, την ασφάλεια, την επεκτασιμότητα και άλλα τεχνικά χαρακτηριστικά. Στη συνέχεια, οι απαιτήσεις αυτές παρουσιάζονται με τη μορφή πίνακα για καλύτερη κατανόηση και ανάλυση.

Τίτλος απαίτησης	Τρόπος σύνδεσης
Σύντομη περιγραφή	Οι χρήστες έχουν τη δυνατότητα να συνδεθούν με τον λογαριασμό Google τους εφόσον έχει δημιουργηθεί λογαριασμός για αυτούς από κάποιον διαχειριστή.
Τύπος απαίτησης	Λειτουργική Απαίτηση
Επεξήγηση	Δεδομένου ότι οι περισσότερες ερευνητικές ομάδες υπάγονται σε κάποιο πανεπιστήμιο και κατέχουν έναν ιδρυματικό λογαριασμό, είναι ιδανικό για την δημιουργία ενός καταλόγου εγκεκριμένων μελών (white list)

Πίνακας 1: Απαίτηση - Τρόπος σύνδεσης

Τίτλος απαίτησης	Αναλυτική Αναφορά
Σύντομη περιγραφή	Υπάρχει η δυνατότητα εξαγωγής φύλλων χρονοχρέωσης για ένα συγκεκριμένο μήνα από τους χρήστες σε μορφή CSV. Η αναμονή για τη λήψη του συγκεκριμένου αρχείου μπορεί να είναι έως και 5 δευτερόλεπτα ανάλογα τον όγκο των δεδομένων.
Τύπος απαίτησης	Λειτουργική Απαίτηση

Επεξήγηση	Προσφέρει την δυνατότητα στους διαχειριστές των ομάδων, να δημιουργήσουν μία αναλυτική αναφορά για την αξιολόγηση της απόδοσης για παράδειγμα.
-----------	--

Πίνακας 2: Απαίτηση - Αναλυτική Αναφορά

Τίτλος απαίτησης	GANTT διάγραμμα
Σύντομη περιγραφή	GANTT διάγραμμα στην σελίδα προβολής έργων.
Τύπος απαίτησης	Λειτουργική Απαίτηση
Επεξήγηση	Για την καλύτερη κατανόηση και διαχείριση των έργων, υπάρχει ένα GANTT διάγραμμα το οποίο παρουσιάζει το όνομα του έργου, την ημερομηνία έναρξης έως και την ημερομηνία λήξης. Υπάρχει επίσης ένας οπτικός πίνακας που περιλαμβάνει όλες τις ημερομηνίες, χρωματικές αναπαραστάσεις των ημερομηνιών έναρξης και λήξης. Ο χρήστης μπορεί να αιωρηθεί πάνω από το κάθε έργο για να παρατηρήσει της ημέρες που απομένουν πριν την λήξη του.

Πίνακας 3: Απαίτηση - GANTT διάγραμμα

Τίτλος απαίτησης	Επεξεργασία Δεδομένων
Σύντομη περιγραφή	Ο χρήστης έχει τη δυνατότητα να επεξεργαστεί τα δεδομένα, έργα, συμβάσεις, φύλλα χρονοχρέωσης.
Τύπος απαίτησης	Λειτουργική Απαίτηση
Επεξήγηση	Αυτή η απαίτηση διασφαλίζει ότι ο χρήστης μπορεί να επεξεργαστεί δεδομένα με απόλυτη ασφάλεια. Ο χρήστης πρέπει να έχει τη δυνατότητα να κάνει αλλαγές στα δεδομένα του χωρίς να υπάρχει κίνδυνος να χαθούν πληροφορίες κατά τη διάρκεια της επεξεργασίας ή να μην αποθηκευτούν σωστά οι αλλαγές που πραγματοποιεί.

Πίνακας 4: Απαίτηση - Επεξεργασία Δεδομένων

Τίτλος απαίτησης	Ευκολία εισαγωγής δεδομένων
------------------	-----------------------------

Σύντομη περιγραφή	Η εισαγωγή συγκεκριμένων δεδομένων πρέπει να γίνεται με τέτοιο τρόπο ώστε να διευκολύνει τον χρήστη.
Τύπος απαίτησης	Λειτουργική Απαίτηση
Επεξήγηση	Ο χρήστης αντί για τις ημερομηνίες λήξης εισάγει έναν αριθμό διάρκειας. Επίσης, ο χρήστης θα εισάγει τον μήνα που αρχίζει μία δέσμη εργασιών με μορφή ΜΧΧ, η οποία χρησιμοποιείται ευρέως από τις ερευνητικές ομάδες. Αυτές οι πληροφορίες μεταφράζονται σωστά όταν τις λαμβάνει η εφαρμογή και τις παρουσιάζει σε κανονική ημερολογιακή ημερομηνία.

Πίνακας 5: Απαίτηση - Ευκολία εισαγωγής δεδομένων

Τίτλος απαίτησης	Αναζήτηση
Σύντομη περιγραφή	Ο χρήστης μπορεί να αναζητήσει κάποιο συγκεκριμένο δεδομένο σε όλες σελίδες προβολής.
Τύπος απαίτησης	Λειτουργική Απαίτηση
Επεξήγηση	Οι χρήστες έχουν τη δυνατότητα αναζήτηση συγκεκριμένου δεδομένου με ένα μοναδικό αναγνωριστικό πεδίο ανάλογα τον τύπο. Αυτή η αναζήτηση έχει μια καθυστέρηση έως και 3 δευτερόλεπτα σε μεγάλες ομάδες από πληροφορίες.

Πίνακας 6: Απαίτηση - Αναζήτηση

Τίτλος απαίτησης	Εισαγωγή φύλλων χρονοχρέωσης
Σύντομη περιγραφή	Φιλικό και γνώριμο περιβάλλον εισαγωγής φύλλων χρονοχρέωσης.
Τύπος απαίτησης	Λειτουργική Απαίτηση
Επεξήγηση	Στα φύλλα χρονοχρέωσης όταν ο χρήστης επισκέπτεται τη σελίδα για εισαγωγή δεδομένων εμφανίζεται μια μορφή ημερολογίου για την πιο εύκολη κατανόηση. Στο ημερολόγιο αυτό τα σαββατοκύριακα και οι αργίες είναι απενεργοποιημένες εξ αρχής και μόνο με παρέμβαση του

	χρήστη μπορούν να εισαχθούν δεδομένα σε αυτές. Αυτό το ημερολόγιο εμφανίζεται στον χρήστη στιγμιαία. Οι αργίες ανανεώνονται καθημερινά κάθε μεσάνυχτα ώστε να μειωθεί η πιθανότητα αντιπαράθεσης με τους χρήστες.
--	---

Πίνακας 7: Απαίτηση - Εισαγωγή φύλλων χρονοχρέωσης

Τίτλος απαίτησης	Ειδοποιήσεις
Σύντομη περιγραφή	Μηχανισμός ενημέρωσης χρηστών μέσω ειδοποιήσεων
Τύπος απαίτησης	Λειτουργική Απαίτηση
Επεξήγηση	Υπάρχουν ειδοποιήσεις που ενημερώνουν τον χρήστη για οποιαδήποτε λάθος ενέργεια ή πληροφορία. Αυτό μπορεί να είναι κάποιο διπλότυπο μοναδικό δεδομένο όπως το χαρακτηριστικό νούμερο των έργων

Πίνακας 8: Απαίτηση - Ειδοποιήσεις

Τίτλος απαίτησης	Συνεργασία διαχειριστών
Σύντομη περιγραφή	Πολλαπλοί διαχειριστές για μία ομάδα
Τύπος απαίτησης	Λειτουργική Απαίτηση
Επεξήγηση	Ο χρήστης έχει τη δυνατότητα να συνδεθεί με άλλους χρήστες για τη δημιουργία μίας ομάδας, παρέχοντας έτσι την δυνατότητα, εισαγωγής, επεξεργασίας και διαγραφής κοινού περιεχομένου.

Πίνακας 9: Απαίτηση - Συνεργασία διαχειριστών

Τίτλος απαίτησης	Τρόπος αποθήκευσης
Σύντομη περιγραφή	Τα δεδομένα αποθηκεύονται σε βάση δεδομένων.
Τύπος απαίτησης	Μη Λειτουργική Απαίτηση
Επεξήγηση	Αυτή η απαίτηση διασφαλίζει ότι όλα τα δεδομένα των χρηστών αποθηκεύονται σε μια βάση δεδομένων που



	<p>ακολουθεί αυστηρά τα πρότυπα ασφαλείας. Η βάση δεδομένων εφαρμόζει βέλτιστες πρακτικές όπως προστασία από μη εξουσιοδοτημένη πρόσβαση και τακτική ενημέρωση των μέτρων ασφαλείας. Επιπλέον, υποβάλλεται σε ελέγχους ασφαλείας, εξασφαλίζοντας την προστασία από παραβιάσεις και επιθέσεις. Περνάει το 100% των τεστ ασφαλείας, κάτι που αποδεικνύει ότι δεν παρουσιάζει κενά ασφαλείας ή αδυναμίες που θα μπορούσαν να εκθέσουν τα δεδομένα των χρηστών σε κίνδυνο.</p>
--	--

Πίνακας 10: Απαίτηση – Τρόπος αποθήκευσης

Τίτλος απαίτησης	Συνεχής λειτουργία
Σύντομη περιγραφή	Η εφαρμογή πρέπει να είναι πάντα διαθέσιμη στους χρήστες με ποσοστό χρόνου εκτός λειτουργίας 5 τις εκατό.
Τύπος απαίτησης	Μη Λειτουργική Απαίτηση
Επεξήγηση	<p>Η απαίτηση αυτή καθορίζει ότι η εφαρμογή πρέπει να είναι διαθέσιμη στους χρήστες το 95% του χρόνου, επιτρέποντας ένα μέγιστο ποσοστό χρόνου εκτός λειτουργίας (downtime) 5%. Αυτό σημαίνει ότι η εφαρμογή θα πρέπει να είναι σχεδόν πάντα προσβάσιμη, ενώ οποιαδήποτε διακοπή λειτουργίας (π.χ., λόγω συντήρησης ή τεχνικών προβλημάτων) δεν πρέπει να ξεπερνά το συγκεκριμένο ποσοστό. Η διαθεσιμότητα της εφαρμογής είναι κρίσιμη για την απρόσκοπτη χρήση της από τους χρήστες, και το ποσοστό αυτό διασφαλίζει μια υψηλή απόδοση και αξιοπιστία του συστήματος.</p>

Πίνακας 11: Απαίτηση - Συνεχής λειτουργία

Τίτλος απαίτησης	Αλληλεπίδραση χρηστών με ιστοσελίδα
Σύντομη περιγραφή	Οι αλληλεπιδράσεις των χρηστών με την ιστοσελίδα πρέπει να είναι γρήγορες.
Τύπος απαίτησης	Μη Λειτουργική Απαίτηση

Επεξήγηση	Οι αλληλεπιδράσεις των χρηστών με την ιστοσελίδα πρέπει να είναι γρήγορες, κρατώντας υψηλή επίδοση ακόμη και σε ώρες αιχμής. Η μέση τιμή απόκρισης δε θα πρέπει να υπερβαίνει τα 3 δευτερόλεπτα.
-----------	--

Πίνακας 12: Απαίτηση - Αλληλεπίδρασή χρηστών με ιστοσελίδα

Τίτλος απαίτησης	Αξιοπιστία ιστοσελίδας
Σύντομη περιγραφή	Η ιστοσελίδα θα πρέπει να λειτουργεί αξιόπιστα χωρίς απρόοπτες αποτυχίες.
Τύπος απαίτησης	Μη Λειτουργική Απαίτηση
Επεξήγηση	Η ιστοσελίδα θα πρέπει να λειτουργεί αξιόπιστα χωρίς απρόοπτες αποτυχίες. 99.99 τις εκατό η εφαρμογή θα είναι αξιόπιστη, με το υπόλοιπο ποσοστό να οφείλεται σε τυχαίες αποτυχίες καθ' όλη τη διάρκεια του έτους.

Πίνακας 13: Απαίτηση - Αξιοπιστία ιστοσελίδας

Τίτλος απαίτησης	Συντήρηση και ενημέρωση ιστοσελίδας
Σύντομη περιγραφή	Η εφαρμογή θα πρέπει να είναι εύκολο να συντηρηθεί και να ενημερωθεί
Τύπος απαίτησης	Μη Λειτουργική Απαίτηση
Επεξήγηση	Η εφαρμογή θα πρέπει να είναι εύκολο να συντηρηθεί και να ενημερωθεί. Η αλλαγές σε κώδικα και ενημερώσεις θα πρέπει να γίνονται μέσα σε 2 ώρες για μικρές ενημερώσεις και μέσα σε 24 ώρες για σημαντικές χωρίς ιδιαίτερα προβλήματα.

Πίνακας 14: Απαίτηση - Συντήρηση και ενημέρωση ιστοσελίδας

Τίτλος απαίτησης	Τεκμηρίωση χρήσης ιστοσελίδας και παραγωγής κώδικα
Σύντομη περιγραφή	Θα πρέπει να υπάρχει κατανοητή τεκμηρίωση και για τους χρήστες αλλά και για τους προγραμματιστές.

Τύπος απαίτησης	Μη Λειτουργική Απαίτηση
Επεξήγηση	Θα πρέπει να υπάρχει κατανοητή τεκμηρίωση και για τους χρήστες αλλά και για τους προγραμματιστές. Θα πρέπει η τεκμηρίωση να καλύπτει 100 τις εκατό της εφαρμογής και να έχει οριστεί βοηθητική τουλάχιστον από το 80 τις εκατό των χρηστών.

Πίνακας 15: Απαίτηση - Τεκμηρίωση χρήσης ιστοσελίδας και παραγωγής κώδικα

Τίτλος απαίτησης	Ικανοποίηση χρηστών
Σύντομη περιγραφή	Πρέπει η χρήση της εφαρμογής να είναι εύκολη και να υπάρχει άνετη περιήγηση της.
Τύπος απαίτησης	Μη Λειτουργική Απαίτηση
Επεξήγηση	Πρέπει η χρήση της εφαρμογής να είναι εύκολη και να υπάρχει άνετη περιήγηση της. Η ικανοποίηση των χρηστών θα πρέπει να μετριέται από αξιολογήσεις με τουλάχιστον 80 τις εκατό θετική κριτική.

Πίνακας 16: Απαίτηση - Ικανοποίηση χρηστών

## 2.5: Ανάλυση Βάσης Δεδομένων

Μια βάση δεδομένων είναι μία αποθήκη από δεδομένα, όπου αυτά μπορούν να υποστούν διάφορες ενέργειες, όπως την αποθήκευση καινούργιων δεδομένων ή την ενημέρωση των ήδη υπαρχόντων δεδομένων. Τα δύο είδη που χρησιμοποιούνται ευρέως είναι οι σχεσιακές και αντιστοίχως η μη σχεσιακές βάσεις. Στις σχεσιακές βάσεις δεδομένων όπως αναφέρει και το όνομα τους, υπάρχουν σχέσεις μεταξύ των δεδομένων. Δημιουργούνται πίνακες και αυτοί οι πίνακες συνδέονται ανάλογα με τις σχέσεις. Τα δεδομένα μέσα στους πίνακες είναι αποθηκευμένα σε γραμμές και στήλες. Κάθε γραμμή σε ένα πίνακα έχει ένα μοναδικό κλειδί το οποίο είναι μέρος της παραπάνω σύνδεσης. Οι μη σχεσιακές βάσεις δεδομένων δεν χρησιμοποιούν τον παραπάνω τρόπο, αποθηκεύοντας σε γραμμές και στήλες, αλλά αντιθέτως χρησιμοποιούν ένα μοντέλο το οποίο είναι βελτιστοποιημένο για τις προϋποθέσεις του συγκεκριμένου τύπου δεδομένων που χρειάζεται να αποθηκευτεί. Συνήθης τρόπος αποθήκευσης δεδομένων είναι σε μορφή εγγράφου. Αυτός ο τρόπος διαχειρίζεται το έγγραφο σε ένα σύνολο από πεδία σε μορφή γραμματοσειράς και την τιμή αυτών σε μορφή αντικειμένου.

Και οι δύο τύποι βάσεων προσφέρουν τις βασικές λειτουργίες όπως αποθήκευση και διαχείριση των δεδομένων, όμως η αρχιτεκτονική, τα πλεονεκτήματα και οι περιορισμοί τους τα καθιστούν

κατάλληλα για διαφορετικές εφαρμογές και φόρτους εργασίας. Οι συγκεκριμένες έρευνες [1, 3] αναφέρουν ότι οι μη σχεσιακές βάσεις έχουν υψηλό βαθμό διαπερατότητας και εύκολη επέκταση. Η απόδοση μπορεί να αυξηθεί με την χρήση της μνήμης του συστήματος στις μη σχεσιακές ενώ αντίθετα στις σχεσιακές πρέπει να χρησιμοποιηθούν εργαλεία τρίτων. Ταυτόχρονα τα δεδομένα στις μη σχεσιακές μπορούν να εισαχθούν χωρίς κάποιο συγκεκριμένο σχήμα. Εκεί όπου μειώνεται η απόδοση της μη σχεσιακής βάσης, όμως, είναι στην περιορισμένη συνοχή και στην αποδεκτή εισαγωγή διπλότυπων εγγράφων που μπορεί ανάλογα με την εφαρμογή να μην είναι επιθυμητό. Τέλος η αναζήτηση δυσχερεί σε σχέση με τις σχεσιακές βάσεις, ειδικά όταν γίνεται σε πολλαπλά ταυτόχρονα πεδία. Στις επόμενες έρευνες [2, 4] που μελετήθηκαν, δοκιμάστηκαν συγκεκριμένα ενέργειες όπως η εισαγωγή δεδομένων, η επεξεργασία τους, η διαγραφή τους και η αναζήτηση τους. Τα αποτελέσματα που διαπιστώθηκαν είναι παρόμοια με αυτά της έρευνας [1], τονίζοντας την εξής σημαντική διαφορά: όταν δοκιμάστηκαν οι τρεις πρώτες λειτουργίες με τις μη σχεσιακές βάσεις, αποδείχτηκε ότι αυτές υπερτερούν. Αντιθέτως, στην αναζήτηση οι σχεσιακές βάσεις είχαν τα ταχύτερα αποτελέσματα. Επίσης οι έρευνες αναφέρουν ότι για μικρές έως και μεσαίες εταιρίες χρησιμοποιείται γενικά μια βάση δεδομένων σχεσιακή, ενώ αντίστοιχα για μεγαλύτερες εταιρίες θα ήταν προτιμότερο να χρησιμοποιηθεί μια μη σχεσιακή βάση. Τα κριτήρια επιλογής πάντα επηρεάζονται από την εφαρμογή και τον τύπο δεδομένων που χρησιμοποιεί. Η τελευταία έρευνα [5] επικεντρώνεται κυρίως στο γεγονός ότι ένας από τους σκοπούς των μη σχεσιακών βάσεων είναι η δημιουργία κώδικα στην προτιμώμενη γλώσσα του προγραμματιστή σε αντικειμενοστραφή περιβάλλον. Με τα παραπάνω δεδομένα επιλέχθηκε η MongoDB.

Όνομα πεδίου	Τύπος πεδίου	Ιδιότητες
title	Συμβολοσειρά	Μοναδικό
activeIntervals	Πίνακας τύπου Interval	-

Πίνακας 17: Ορισμός Πίνακα Παραδοτέων (Wps)

Όνομα πεδίου	Τύπος πεδίου	Ιδιότητες
id	Συμβολοσειρά	Μοναδικό
title	Συμβολοσειρά	-
description	Συμβολοσειρά	-
wps	Συμβολοσειρά	Παραπομπή(reference) στο πεδίο wps._id
interval	Interval	-

Πίνακας 18: Ορισμός Πίνακα Έργων (Projects)

Όνομα πεδίου	Τύπος πεδίου	Ιδιότητες
--------------	--------------	-----------

project	Συμβολοσειρά	Παραπομπή(reference) στο πεδίο project._id
human	Συμβολοσειρά	Παραπομπή(reference) στο πεδίο human._id
wps	Συμβολοσειρά	Παραπομπή(reference) στο πεδίο wps._id
interval	Interval	-
hourlyRate	Αριθμός	-
totalCost	Αριθμός	-

Πίνακας 19: Ορισμός Πίνακα Συμβολαίων (Contracts)

Ο τύπος Interval που χρησιμοποιείται στους παραπάνω πίνακες αποτελείται από δύο πεδία, το startDate και το duration όπου το ένα είναι η ημερομηνία έναρξης της οποιασδήποτε διαδικασίες και η διάρκεια της σε μήνες.

Όνομα πεδίου	Τύπος πεδίου	Ιδιότητες
vat	Συμβολοσειρά	Μοναδικό
lastName	Συμβολοσειρά	-
firstName	Συμβολοσειρά	-

Πίνακας 20: Ορισμός Πίνακα Ανθρώπινου δυναμικού(Humans)

Όνομα πεδίου	Τύπος πεδίου	Ιδιότητες
days	Πίνακας τύπου Day	-
contract	Συμβολοσειρά	Παραπομπή(reference) στο πεδίο contract._id
timestamp_created	Συμβολοσειρά	-

Πίνακας 21: Ορισμός Πίνακα Φύλλων Χρονοχρέωσης(Timesheets)

Ο τύπος days αντικατοπτρίζει τις ημέρες του μήνα που ο διαχειριστής θα προσθέσει τις ώρες εργασίες για κάθε συμβόλαιο. Εμπεριέχει τα πεδία timestamp όπου είναι η συγκεκριμένη μέρα και έναν πίνακα που περιέχει αντικείμενα με πεδία wr όπου είναι μια παραπομπή στο παραδοτέο στο οποίο δούλεψε το άτομο και hours όπου είναι οι ώρες που δούλεψε.

Όνομα πεδίου	Τύπος πεδίου	Ιδιότητες
projects	Πίνακας από συμβολοσειρές	Παραπομπές(reference) στο πεδίο project._id

humans	Πίνακας από συμβολοσειρές	Παραπομπές (reference) στο πεδίο human._id
contracts	Πίνακας από συμβολοσειρές	Παραπομπές(reference) στο πεδίο contract._id
timesheets	Πίνακας από συμβολοσειρές	Παραπομπές (reference) στο πεδίο timesheet._id
email	Συμβολοσειρά	-
uid	Συμβολοσειρά	-
role	Συμβολοσειρά	-

Πίνακας 22: Ορισμός Πίνακα Χρήστες(Users)

Το πεδίο uid αναφέρεται στο μοναδικό χαρακτηριστικό που έχει κάθε χρήστης στην firebase για να πραγματοποιείται η αυθεντικοποίηση του χρήστη.

Όνομα πεδίου	Τύπος πεδίου	Ιδιότητες
name	Συμβολοσειρά	-
description	Συμβολοσειρά	-
dateIso	Συμβολοσειρά	-
type	Πίνακας από Συμβολοσειρές	-

Πίνακας 23: Ορισμός Πίνακα Αργιών (Holidays)

Αποθηκεύονται σε αυτόν τον πίνακα οι αργίες για την χρησιμοποίησή τους στα φύλλα χρονοχρέωσης και για να μην γίνεται επανειλημμένα αίτημα στο εξωτερικό API.

## ΚΕΦ.3: Υλοποίηση εφαρμογής

### 3.1: Backend Υπηρεσία

Για την υλοποίηση αυτής της υπηρεσίας, backend, υπάρχει η επιλογή της δημιουργίας της από την αρχή ή η επιλογή ενός BaaS συστήματος [16]. Ένα BaaS σύστημα παρέχει υπηρεσίες όπως βάση δεδομένων, εξουσιοδότηση, ειδοποιήσεις ανάλογα την πλατφόρμα, την λογική που χρειάζεται για να δουλέψει η εφαρμογή. Όλες αυτές οι λειτουργίες δημιουργούνται και παρέχονται από τρίτους. Το BaaS είναι ιδανικό για μικρές εταιρείες, αυξάνοντας τον γενικό ρυθμό ανάπτυξης της ιστοσελίδας. Προσφέρει αυτόματη επέκταση, μειώνει την συντήρηση που χρειάζεται η εφαρμογή και είναι βασισμένο στο μοντέλο, που χρεώνει ανάλογα με τους πόρους που χρησιμοποιούνται. Σε αντίθεση η υλοποίηση ενός συστήματος από την αρχή προσφέρει ολικό έλεγχο στην δομή της εφαρμογής, ευκολία στην δημιουργία προσαρμοσμένων χαρακτηριστικών, μεγαλύτερη ασφάλεια στα δεδομένα, εφόσον δεν τα διαχειρίζεται κάποια άλλη οντότητα. Με βάση τα παραπάνω και τις τωρινές αλλά και μελλοντικές ανάγκες της εφαρμογής η πτυχιακή υλοποιεί την backend υπηρεσία από την αρχή.

Η εφαρμογή γράφτηκε σε NodeJS για να χρησιμοποιήσει την ομαδοποίηση που προσφέρει σε σχέση με το frontend, υψηλή απόδοση σε πολλαπλές ενέργειες και βιβλιοθήκες για την εξομάλυνση και διευκόλυνση της ανάπτυξης κώδικα. Δεδομένου ότι στον χώρο του προγραμματισμού κυριαρχεί κυρίως ο αντικειμενοστραφής προγραμματισμός, επιλέχθηκε συγκεκριμένα να χρησιμοποιηθεί η Typescript [7, 8 ,9 ,10]. Η Typescript μεταφράζεται σε JavaScript για να εκτελεστεί. Κατά την ανάπτυξη του κώδικα, επιτρέπει την εισαγωγή κλάσεων, τύπων, διεπαφών. Έχει αυτομάτους ελέγχους για συνηθισμένα λάθη τα οποία τα εμφανίζει σε οποιαδήποτε IDE χρησιμοποιείται. Διευκολύνει την επέκταση και διατήρηση του προγράμματος καθ' όλη την διάρκεια ζωής ενός προγραμματικού έργου. Για την διευκόλυνση της ανάπτυξης της υπηρεσίας backend, χρησιμοποιήθηκε το framework NestJS[6].

Το NestJS δίνει έμφαση στη δημιουργία ενός καθαρού, δομημένου κώδικα και έχει επίσημη υποστήριξη για διάφορα γνωστά εργαλεία σαν το GraphQL, WebSocket, Microservices και άλλα. Προτιμήθηκε το NestJS διότι προσφέρει ένα δομημένο περιβάλλον το οποίο θα βοηθήσει καθώς επεκτείνεται η εφαρμογή και κατά την διάρκεια της υλοποίησης της. Τέλος σημαντικό επίσης χαρακτηριστικό είναι η εσωτερική υποστήριξη για βιβλιοθήκες που θα χρησιμοποιηθούν στην υλοποίηση όπως το mongoose, το πακέτο για την σύνδεση με την MongoDB βάση.

### 3.2: Βάση δεδομένων

Σε αυτήν την πτυχιακή δεν υπάρχουν πολλές διασυνδέσεις μεταξύ των τύπων δεδομένων, με αποτέλεσμα να μεριμνά την επεκτασιμότητα της εφαρμογής μαζί με την ανάγκη για γρήγορες συναλλαγές και την ευκολία στην ενσωμάτωση μιας τέτοιας βάσης στο σύστημα. Έτσι λοιπόν επιλέχθηκε μία μη σχεσιακή βάση για την αποθήκευση των δεδομένων και συγκεκριμένα η MongoDB. Χρησιμοποιεί μορφή εγγράφων JSON (BSON στη MongoDB), επιτρέποντας την αποθήκευση δεδομένων με πολύπλοκες δομές χωρίς την ανάγκη προκαθορισμένων σχημάτων. Εξ ορισμού η MongoDB δημιουργεί ένα πεδίο `_id` σε κάθε έγγραφο για την εύκολη διαχείριση και αναζήτηση τους. Αυτή η ευελιξία κάνει τη MongoDB ιδανική για εφαρμογές που απαιτούν την

αποθήκευση μεγάλου όγκου δεδομένων ή δεδομένων που μπορεί να μεταβάλλονται συχνά. Οι χρήστες μπορούν να καταχωρούν δεδομένα χωρίς να ακολουθούν αυστηρές δομές, κάτι που διευκολύνει την ταχεία ανάπτυξη εφαρμογών. Χάρη στην αρχιτεκτονική της, υποστηρίζει κλιμάκωση (scaling) τόσο κάθετα όσο και οριζόντια, επιτρέποντας σε συστήματα να προσαρμόζονται στις ανάγκες αυξημένης επεξεργαστικής ισχύος ή αποθήκευσης. Επιπλέον, η MongoDB προσφέρει ενσωματωμένα χαρακτηριστικά όπως αντιγραφή δεδομένων (replication) για εξασφάλιση αξιοπιστίας και ανοχής σφαλμάτων, καθώς και μηχανισμούς διαμοιρασμού δεδομένων (sharding) για την κατανομή μεγάλων βάσεων δεδομένων σε πολλαπλούς διακομιστές.

### 3.3: Firebase υπηρεσία

Ένα από τα σημαντικά αν όχι το πιο σημαντικό σημείο είναι η ασφάλεια και η εγκυρότητα των διαπιστευτηρίων των χρηστών. Η ταυτοποίηση είναι η πρώτη άμυνα ενάντια σε μη εξουσιοδοτημένη πρόσβαση, που οδηγεί έτσι στην ανάγκη για σωστή και ισχυρή δημιουργία μεθόδων για αποθήκευση αυτών των δεδομένων. Για τον παραπάνω σκοπό μπορούν να χρησιμοποιηθούν δύο τρόποι: η δημιουργία ενός συστήματος πιστοποίησης μαζί με τις άλλες υπηρεσίες ή η χρήση εξωτερικών συστημάτων που έχουν ήδη δημιουργηθεί για αυτό τον σκοπό. Πριν την ανάλυση των δύο επιλογών να αναφερθεί ότι από τη #1 λειτουργική απαίτηση οι χρήστες θέλουν να έχουν τη δυνατότητα εισόδου με τον λογαριασμό τους στην Google.

Η δημιουργία ενός τέτοιου εσωτερικού συστήματος εμπεριέχει τη διαχείριση των εγγραφών των χρηστών, των στοιχείων σύνδεσης και τη γενική διαχείριση του συστήματος. Με το να επιλεγθεί εσωτερικό σύστημα, επιτρέπει τη δημιουργία προσαρμοσμένων χαρακτηριστικών ανάλογα με τις ανάγκες της εφαρμογής. Επίσης, δεν βασίζεται σε τρίτες υπηρεσίες με αποτέλεσμα να μειώνονται τα σημεία όπου η εφαρμογή μπορεί να αποτύχει. Ταυτόχρονα μειώνονται και τα σημεία από τα οποία μπορούν τα διαπιστευτήρια των χρηστών να διαρρεύσουν. Για να δημιουργηθεί όμως όλη αυτή η υπηρεσία χρειάζεται ένα μεγάλο ανθρώπινο δυναμικό, αρκετό χρόνο για την υλοποίηση του και ειδικές γνώσεις επί του θέματος. Όμως πέρα από τη δημιουργία του, υπάρχει και συνεχής ανάγκη για αναβαθμίσεις ασφαλείας και συντήρησης. Πρέπει επίσης το παραπάνω σύστημα να τηρεί και τους γενικούς κανόνες και κανονισμούς ασφαλείας και να ανανεώνεται όταν αλλάζουν αυτοί. Οι παραπάνω λόγοι οδήγησαν στη χρήση ενός εξωτερικού συστήματος για την είσοδο των χρηστών στην ιστοσελίδα, την υπηρεσία Firebase.

Σε λεπτομερή αναφορά οι λειτουργίες που μπορεί να παρέχει η Firebase είναι οι εξής: Προσφέρουν έτοιμα για χρήση παραδείγματα που περιέχουν τόσο κώδικα για λόγους σχεδιασμού όσο και μόνο τη λογική. Παρέχονται παραδείγματα κώδικα για το πώς μπορεί κάποιος να εφαρμόσει την εν λόγω βιβλιοθήκη. Στην περίπτωση της πτυχιακής δημιουργούνται αντικείμενα τα οποία εφαρμόζουν στον ήδη υπάρχοντα σχεδιασμό της ιστοσελίδας. Για τις μικρές εταιρίες, περιπτώσεις δοκιμών ή και σε περιπτώσεις νεοφυών επιχειρήσεων δίνεται δωρεάν μια βαθμίδα που επιτρέπει να είναι ενεργοί έως και 3000 χρήστες ταυτόχρονα στην ιστοσελίδα, με τη δυνατότητα επέκτασης οποιαδήποτε στιγμή. Αποθηκεύει στη μνήμη τον συνδεδεμένο χρήστη ώστε ακόμη και να γίνει ανανέωση της σελίδας να μην χρειαστεί να



συνδεθεί ξανά. Αυτή η πληροφορία χρησιμοποιείται για την ορθή εμφάνιση ορισμένων στοιχείων για τους χρήστες ανάλογα και με τον ρόλο τους. Δίνονται ορισμένες συναρτήσεις ακροατών(listeners) για τις διάφορες καταστάσεις που μπορεί να βρίσκεται ένας χρήστης όπως το αν συνδέθηκε, αν αποσυνδέθηκε ή αν το τεκμήριο ανανεώθηκε για οποιοδήποτε λόγο. Τέλος, ο χρήστης έχει εξ αρχής τη δυνατότητα δημιουργίας και διαγραφής λογαριασμού στο περιβάλλον Firebase, το οποίο του δίνει τον πλήρη έλεγχο στον λογαριασμό του. Στην περίπτωση της πτυχιακής όπως αναφέρεται και στη λειτουργική απαίτηση #1, μόνο ένας διαχειριστής μπορεί να δημιουργήσει χρήστες για την ιστοσελίδα.

### 3.4: Frontend υπηρεσία

Για να μπορέσουν να εφαρμοστούν όλες οι λειτουργικές απαιτήσεις και οι χρήστες να μπορούν να ολοκληρώσουν τις διεργασίες τους, πρέπει να επιλεγθούν τα σωστά εργαλεία. Η γλώσσα επιλογής για αυτή την διαδικασία παραμένει η JavaScript, διότι κυριαρχεί σε αυτόν τομέα και κυρίως, επειδή μας προσφέρει μια ομαδοποίηση της διαδικασίας μεταξύ των δύο πλαισίων Backend και Frontend.

Για την ανάπτυξη του Frontend πλαισίου της εφαρμογής πρέπει επίσης να επιλεγθεί και ο τρόπος. Υπάρχει η επιλογή της χρήσης JavaScript χωρίς την προσθήκη τρίτου λογισμικού αλλά και η χρήση κάποιας βιβλιοθήκης. Με την χρήση JavaScript δεν χρειάζεται η χρήση εξωτερικής βιβλιοθήκης, μειώνοντας τον συνολικό αριθμό κώδικα που χρησιμοποιείται. Επίσης ο προγραμματιστής έχει τον πλήρη έλεγχο του κώδικα και το πώς εκτελείται. Μπορεί να είναι πολύ αποδοτική η παραπάνω χρήση αν υλοποιηθεί σωστά. Όμως από την άλλη πλευρά η υλοποίηση αυτή θα αυξήσει τον χρόνο ανάπτυξης για να μπορέσει να γίνει σωστά, διότι προστίθεται στον φόρτο και ο χειρισμός του DOM, με αποτέλεσμα να γίνεται πιο δύσκολη η συντήρηση του προγράμματος, γιατί αυξάνεται το μέγεθος του κώδικα. Για να επιταχυνθεί η διαδικασία επιλέχθηκε η React και επίσης ένα framework που βασίζεται πάνω της, το NextJS.

Αναλυτικά η React: επιτρέπει την εισαγωγή δεδομένων από JavaScript κατευθείαν μέσα στο περιεχόμενο. Διαθέτει εξαρτώμενη απεικόνιση με χρήση απλής JavaScript και εύκολη εμφάνιση λιστών μέσω της συνάρτησης map. Επιτρέπει την δημιουργία handlers για την εύκολη λειτουργία αντικειμένων όπως για παράδειγμα το πάτημα ενός κουμπιού. Παρέχει την δυναμική παρουσίαση δεδομένων είτε σε ένα αντικείμενο μιας σελίδας, είτε σε πολλά, στα οποία επιτρέπει την σύνδεση μεταξύ των τιμών που μπορεί να μεταβάλλονται. Η React θα προσφέρει την παρουσίαση της λειτουργικότητας στον χρήστη. Για την σχεδίαση με την οποία μεταφέρεται αυτή η πληροφορία θα χρησιμοποιηθεί βιβλιοθήκη για το UI το οποίο θα προσφέρει εύκολη CSS χωρίς ιδιαίτερη δυσκολία για τον προγραμματιστή. Η βιβλιοθήκη που χρησιμοποιήθηκε ήταν η Tailwind CSS. Η Tailwind προσφέρει ευελιξία, με κλάσεις χαμηλού επιπέδου, διευκολύνει την παραμετροποίηση χωρίς την ανάγκη για αλλαγή της συνηθισμένης δομής. Αναπαράγει μόνο την CSS την οποία χρειάζεται, για να λειτουργήσει, μειώνοντας έτσι το συνολικό μέγεθος. Έχει σημαντικές διαφορές από την απλή υλοποίηση της CSS οπότε έχει αυξημένη δυσκολία και σε ένα αρχείο HTML αυξάνεται η πολυπλοκότητα του κώδικα. Η Bootstrap προσφέρει ένα μεγάλος πλήθος από αντικείμενα και λειτουργίες, όπως: δική της

διάταξη και συνοχή από εφαρμογή σε εφαρμογή. Οι παραπάνω λειτουργίες χρειάζονται αρκετή παραμετροποίηση αυξάνοντας έτσι την δυσκολία και ταυτόχρονα το μέγεθος της εφαρμογής με την προσθήκη των καινούργιων λειτουργιών.

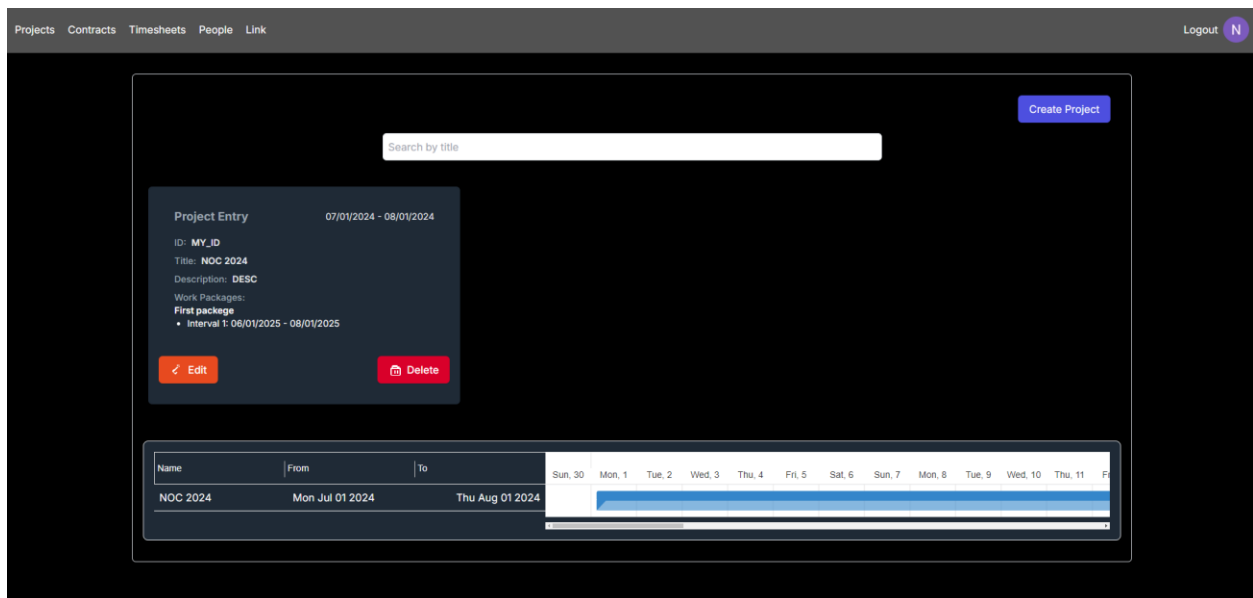
Το framework, NextJS, σε συνδυασμό με την React διευκολύνει την δημιουργία των αντικειμένων. Αναλυτικά η NextJS: διαχωρίζει τις καταστάσεις σε δύο κατηγορίες, την κατηγορία των χρηστών και την κατηγορία του προγράμματος. Για οτιδήποτε υπάγεται στην δεύτερη κατηγορία μπορεί να δημιουργηθεί ένα στατικό περιβάλλον, διότι δεν αλλάζει ποτέ, αυξάνοντας έτσι την ταχύτητα φόρτωσης περιεχομένου. Για την παραλαβή δεδομένων παρουσιάζονται επιπλέον λειτουργίες όπως η δυνατότητα αποθήκευσης των αποτελεσμάτων στην μνήμη για γρήγορη επαναχρησιμοποίηση, πιο γρήγορα αιτήματα και την επιλογή επικύρωσης των ήδη αποθηκευμένων. Προσφέρει εύκολη εισαγωγή Tailwind CSS και άλλων βιβλιοθηκών, για την χρήση στον σχεδιασμό της εφαρμογής. Βελτιστοποιεί την φόρτωση εικόνων, γραμματοσειρών, και άλλων διάφορων σεναρίων, αυξάνοντας έτσι τον ολικό χρόνο ανταπόκρισης του Frontend. Υποστηρίζει επίσης τη γραφή σε Typescript, όπως χρησιμοποιείται και στο Backend αυξάνοντας την ενοποίηση ολόκληρου του συστήματος.

### **3.5: Παρουσίαση frontend περιβάλλοντος**

Στην παρούσα ενότητα, θα παρουσιαστούν τα κύρια χαρακτηριστικά και οι λειτουργίες του frontend περιβάλλοντος της εφαρμογής, όπως αυτά έχουν υλοποιηθεί μέχρι τώρα. Μέσω μιας σειράς από στιγμιότυπα οθόνης(screenshots), θα δοθεί μια αναλυτική εικόνα του τρόπου με τον οποίο οι χρήστες αλληλεπιδρούν με το σύστημα, από την αρχική σελίδα έως τις πιο εξειδικευμένες λειτουργίες. Κάθε εικόνα θα συνοδεύεται από μια επεξήγηση των λειτουργιών που παρουσιάζονται, προσφέροντας μια πλήρη εικόνα της εμπειρίας του χρήστη.

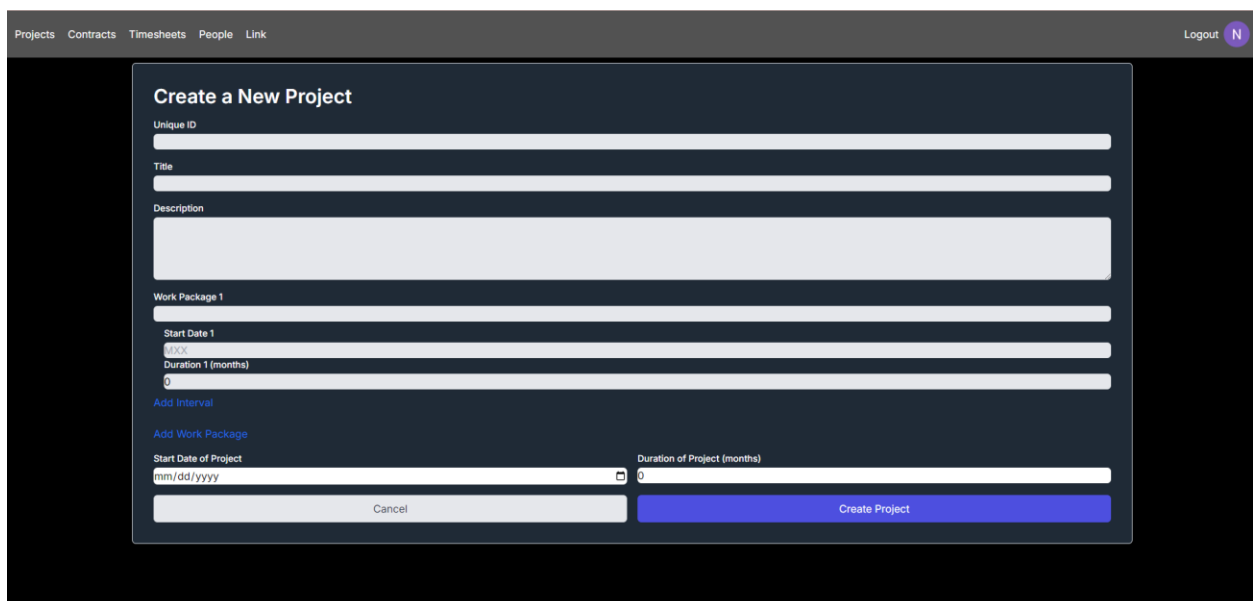
Αυτή η προσέγγιση επιτρέπει να κατανοηθούν καλύτερα οι διεπαφές και η πλοήγηση της εφαρμογής. Θα αναλυθούν βασικές σελίδες όπως η σελίδα σύνδεσης, η διαχείριση έργων, οι επιλογές επεξεργασίας δεδομένων και άλλες κρίσιμες λειτουργίες. Η παρουσίαση αυτή στοχεύει στο να αναδείξει την εργονομία, την ευχρηστία και τη συνολική αισθητική της εφαρμογής, ενώ ταυτόχρονα υπογραμμίζει τον τρόπο με τον οποίο επιτυγχάνεται η αλληλεπίδραση του χρήστη με το σύστημα.

Η πρώτη σελίδα που συναντά ο χρήστης όταν επισκέπτεται την ιστοσελίδα είναι η σελίδα σύνδεσης. Η σελίδα σύνδεσης της εφαρμογής παρέχει στους χρήστες τη δυνατότητα να συνδεθούν εύκολα και γρήγορα μέσω του Google Sign-In. Αυτός ο τρόπος σύνδεσης επιτρέπει στους χρήστες να χρησιμοποιούν τους υπάρχοντες λογαριασμούς τους Google για να αποκτήσουν πρόσβαση στην εφαρμογή, χωρίς να χρειάζεται να δημιουργήσουν νέους κωδικούς πρόσβασης. Η ενσωμάτωση του Google Sign-In ενισχύει την ασφάλεια και την ευχρηστία της εφαρμογής, αφού μειώνει τον κίνδυνο ξεχασμένων κωδικών και προσφέρει μια ομαλή εμπειρία σύνδεσης, με την προστασία των δεδομένων να εξασφαλίζεται από τα πρωτόκολλα ασφαλείας της Google. Ακόμη και αν κάποιος κακόβουλος χρήστης παρακάμψει τις μεταβλητές στον αποθηκευτικό χώρο του προγράμματος περιήγησης, δεν θα μπορεί να επικοινωνεί με την backend υπηρεσία και το μόνο που θα βλέπει θα είναι τις σελίδες χωρίς δεδομένα.



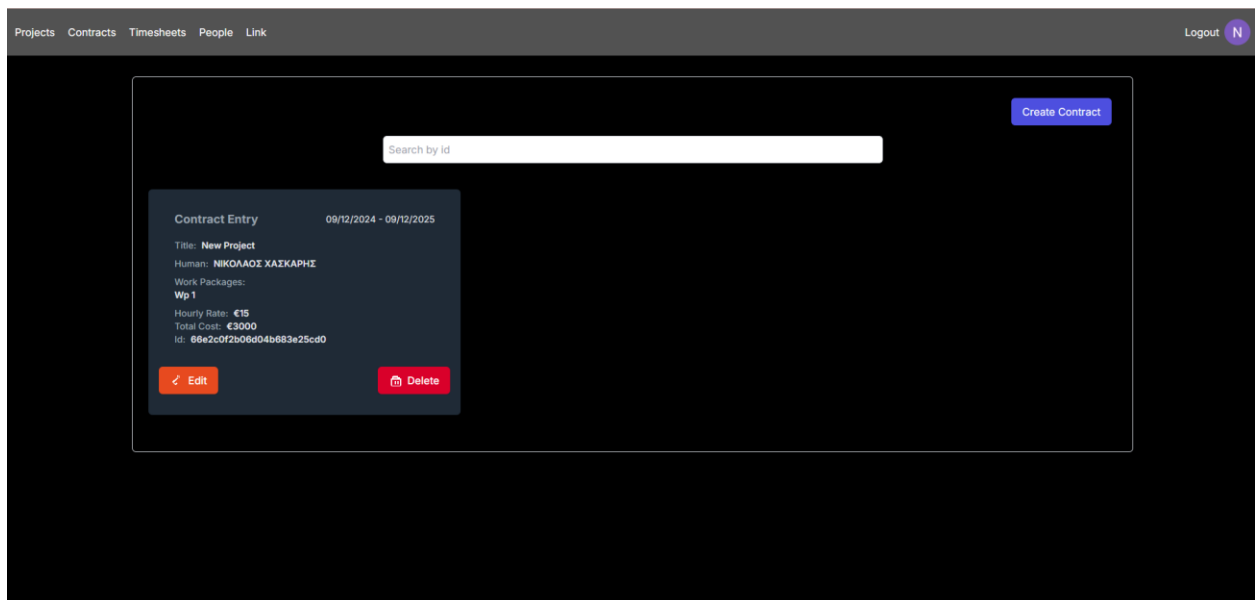
Εικόνα 1: Σελίδα Έργων

Μόλις ο χρήστης συνδεθεί θα γίνει αυτόματη ανακατεύθυνση στην σελίδα με τα έργα, όπου μπορεί να παρατηρήσει όλα τα έργα που έχει δημιουργήσει, να τα επεξεργαστεί και να τα διαγράψει. Επίσης εμφανίζεται στο κάτω μέρος της σελίδας ένα GANTT διάγραμμα το οποίο δείχνει με έναν οπτικό τρόπο τα έργα. Ο χρήστης έχει επίσης την δυνατότητα να αναζητήσει κάποιο έργο από την μπάρα αναζήτησης.

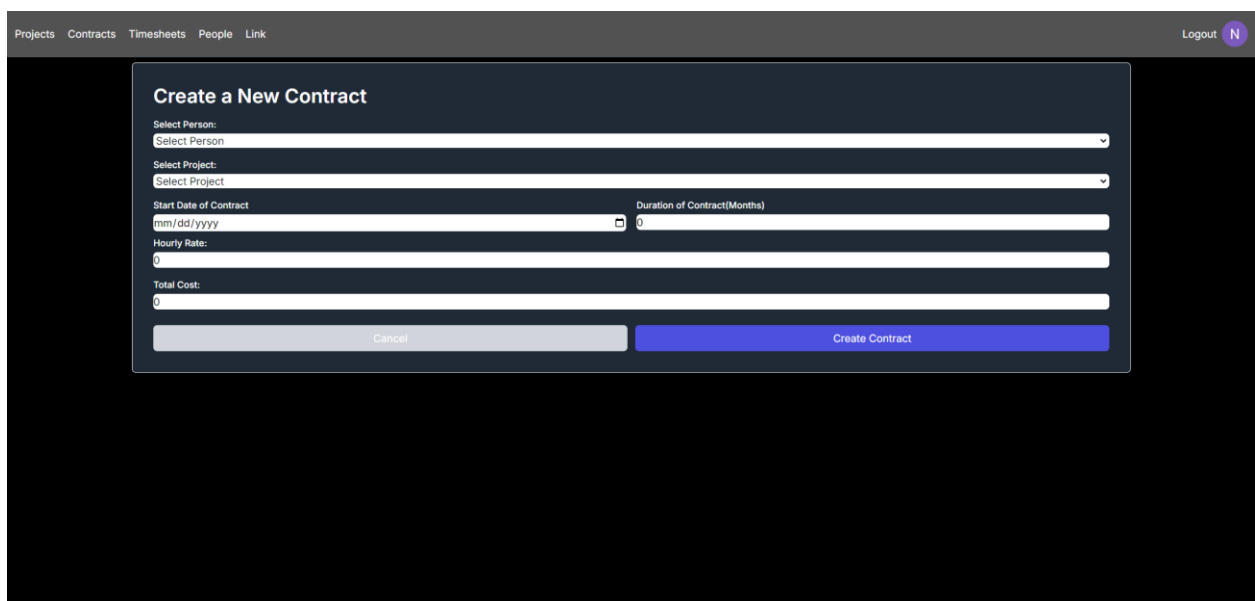


Εικόνα 2: Δημιουργία Έργου

Αν ο χρήστης επιλέξει να δημιουργήσει ένα καινούργιο έργο θα μεταφερθεί σε μία καινούργια σελίδα όπου πρέπει να συμπληρώσει όλα τα απαραίτητα στοιχεία για να προχωρήσει. Ο χρήστης πρέπει να προσθέσει τουλάχιστον ένα παραδοτέο για το έργο. Μπορεί να προσθέσει όσα μεσοδιαστήματα (Intervals) θέλει για το κάθε παραδοτέο και όσα παραδοτέα θέλει αρκεί να βρίσκονται μέσα στο διάστημα του έργου. Είτε ο χρήστης επιλέξει την δημιουργία ενός έργου ή την διακοπή της διαδικασίας θα μεταφερθεί αυτόματα πίσω στην σελίδα με όλα τα έργα.

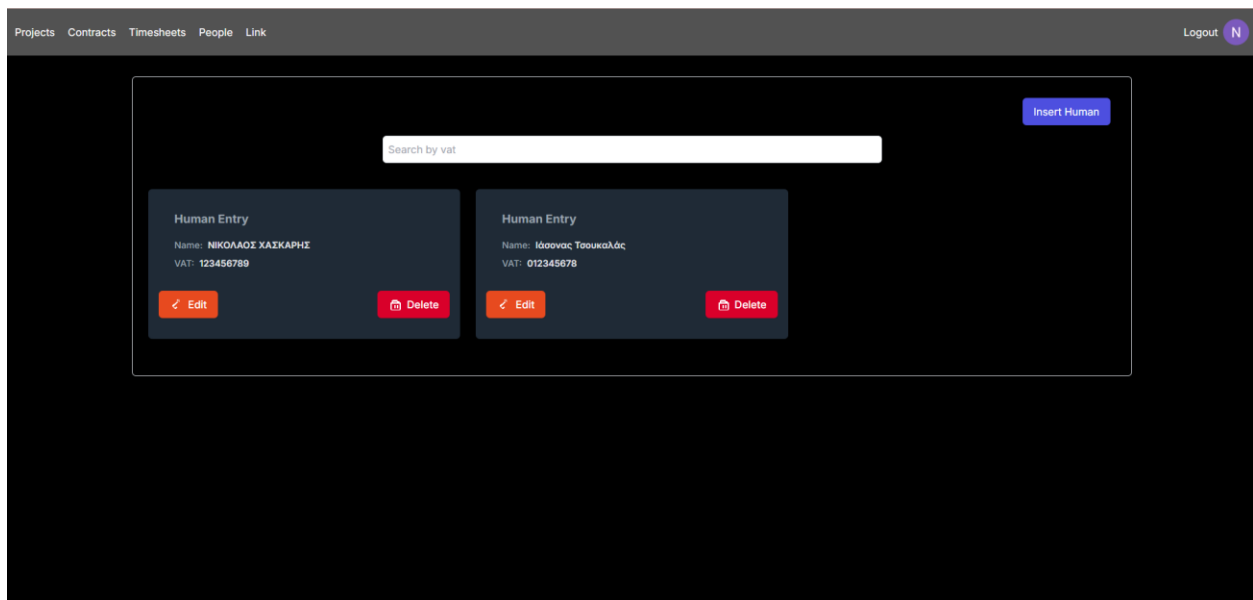


Εικόνα 3: Σελίδα Συμβολαίων

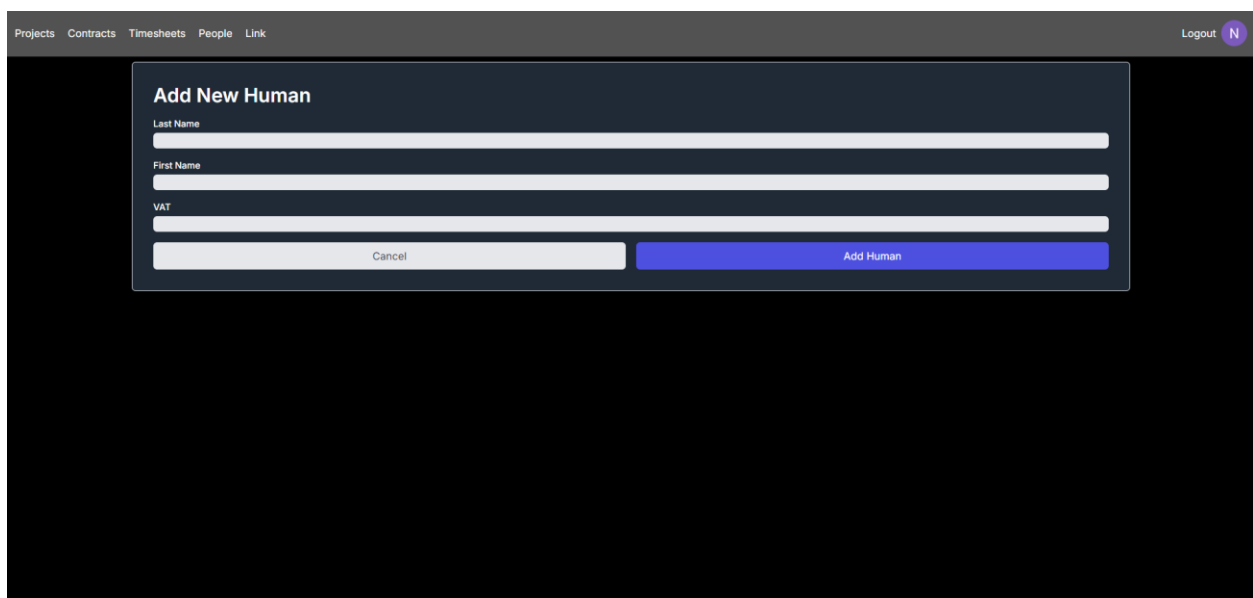


Εικόνα 4: Δημιουργία Συμβολαίου

Κατά την δημιουργία ενός καινούργιου συμβολαίου ο χρήστης επιλέγει ένα άτομο από το ανθρώπινο δυναμικό και ένα έργο από αυτά που έχει δημιουργήσει και είναι ενεργά. Κατά την συμπλήρωση του μισθού ανά ώρα και του συνολικού κόστους του συμβολαίου, θα υπάρχει αυτόματη ενημέρωση αν οι μήνες εργασίας του εργαζομένου ξεπερνούν τους συνολικούς μήνες του έργου.

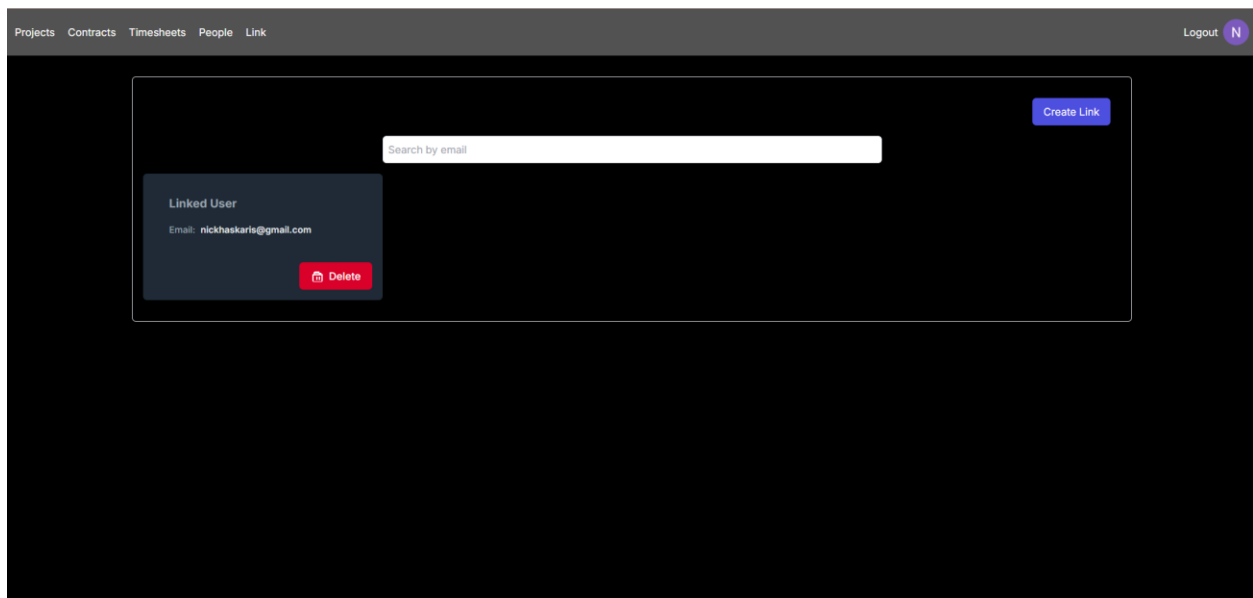


Εικόνα 5: Σελίδα Εργαζομένων

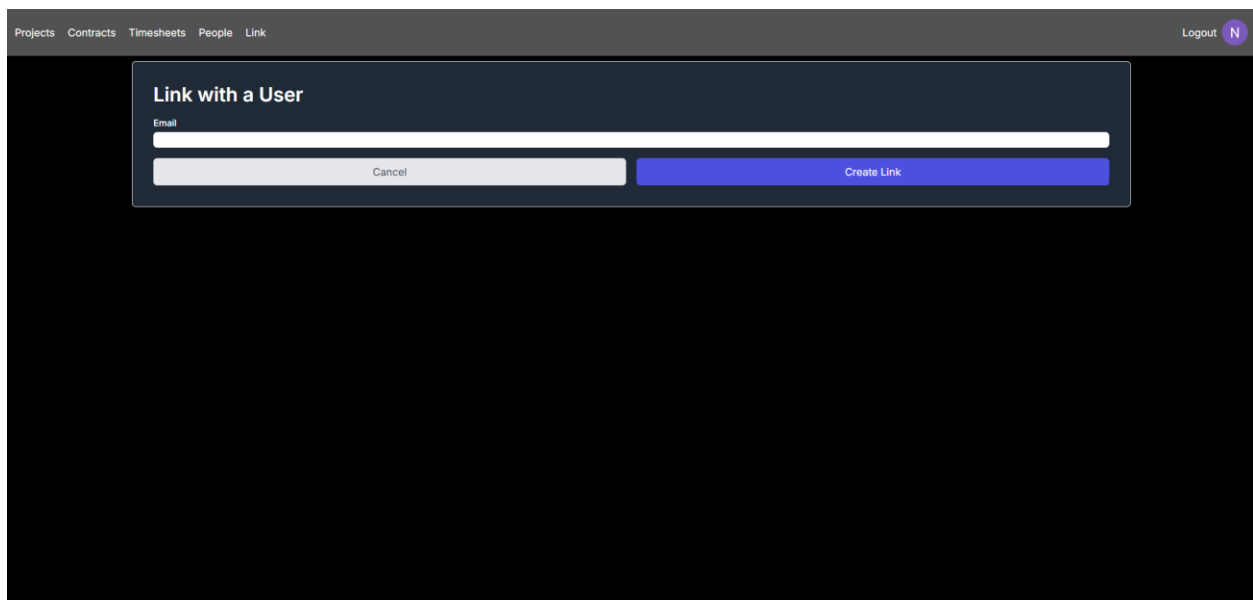


Εικόνα 6: Προσθήκη νέου εργαζομένου

Το μόνο που πρέπει να προσέξει ο χρήστης κατά την προσθήκη νέου εργαζομένου είναι το vat να είναι μοναδικό.

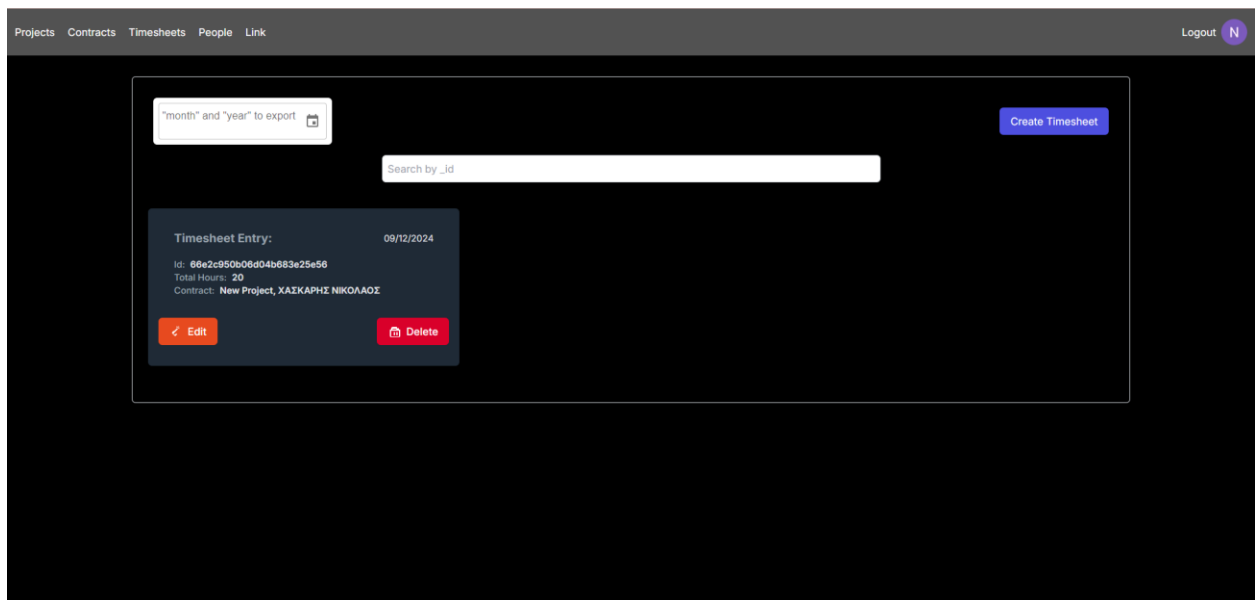


Εικόνα 7: Σελίδα Διασυνδέσεων

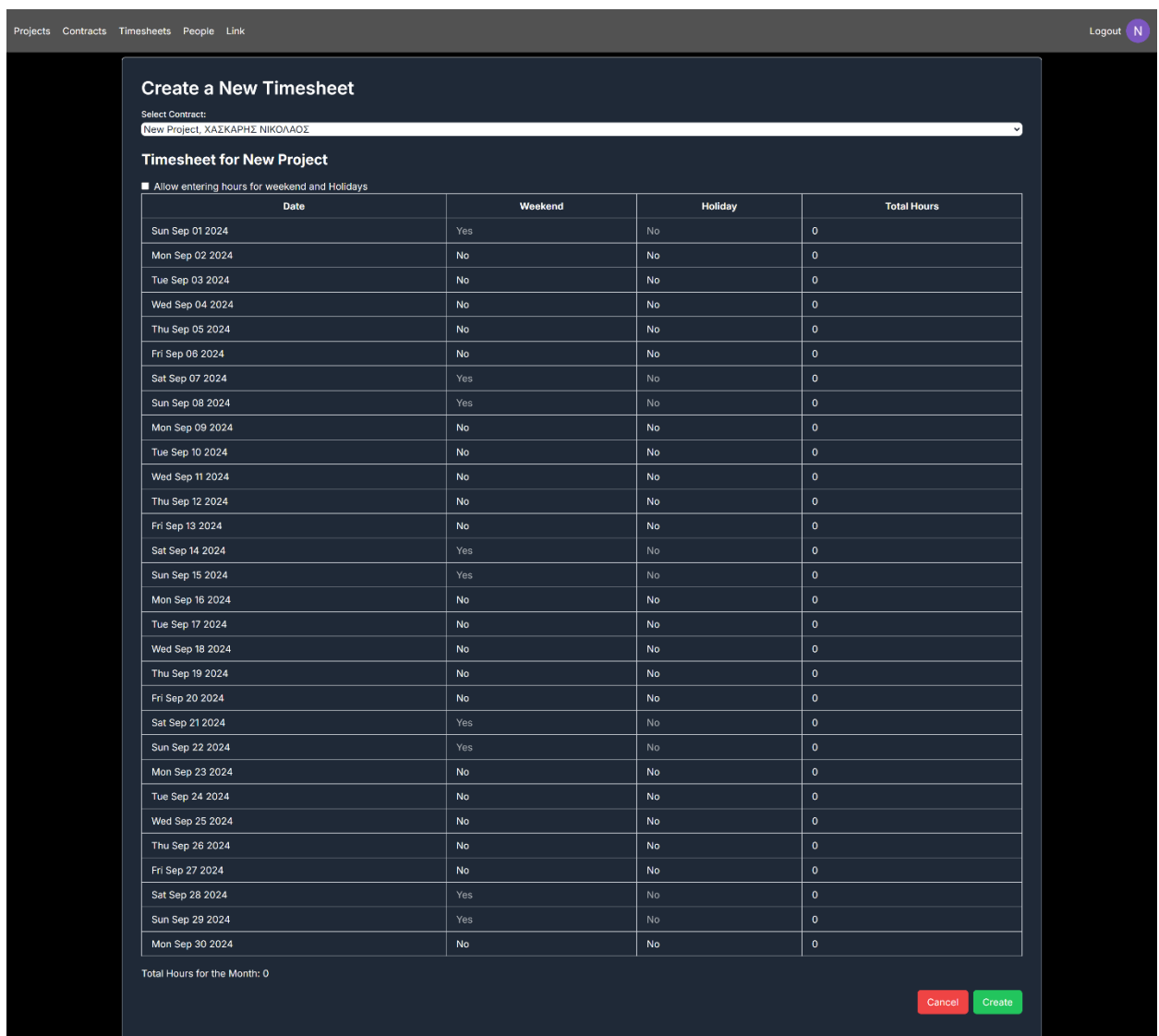


Εικόνα 8: Δημιουργία Διασύνδεσης

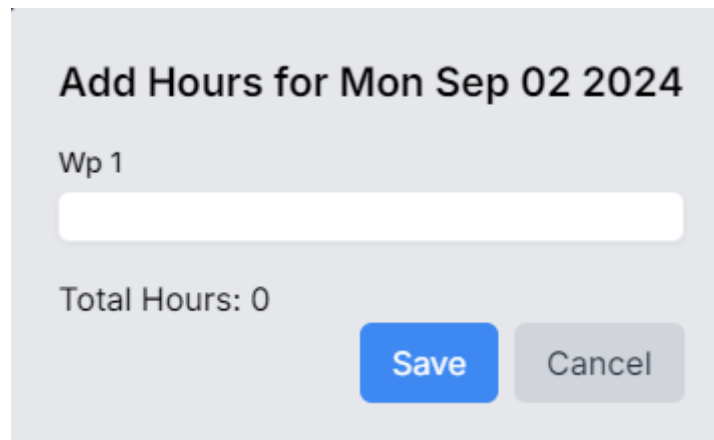
Για την δημιουργία μίας νέας διασύνδεσης μεταξύ των διαχειριστών των ερευνητικών ομάδων ο χρήστης πρέπει απλά να συμπληρώσει τον ηλεκτρονικό ταχυδρομικό τους λογαριασμό και να πατήσει δημιουργία.



Εικόνα 9: Σελίδα φύλλων χρονοχρέωσης



Εικόνα 10: Δημιουργία φύλλου χρονοχρέωσης



Add Hours for Mon Sep 02 2024

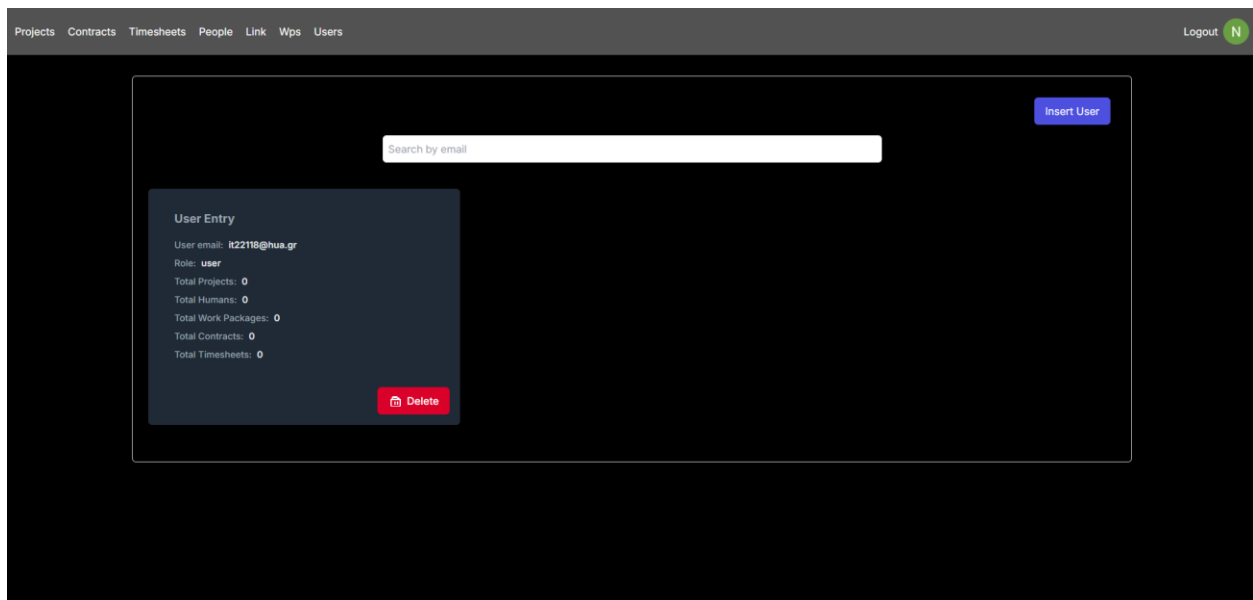
Wp 1

Total Hours: 0

Save Cancel

Εικόνα 11: Εισαγωγή ωρών εργασίας

Στην σελίδα προβολής των φύλλων χρονοχρέωσης ο χρήστης μπορεί να επιλέξει έναν συγκεκριμένο μήνα και να κατεβάσει ένα excel αρχείο με τις πληροφορίες των φύλλων χρονοχρέωσης που βρίσκονται μέσα σε αυτό το διάστημα. Κατά την δημιουργία ενός καινούργιου φύλλου χρονοχρέωσης ο χρήστης επιλέγει ένα από τα διαθέσιμα συμβόλαια και εμφανίζεται μια μορφή τύπου ημερολογίου. Πατώντας πάνω σε κάθε μία από τις ημέρες εμφανίζεται ένα pop-up όπου ο χρήστης μπορεί να συμπληρώσει τις ώρες που δούλεψε ο εργαζόμενος σε κάθε παραδοτέο. Να σημειωθεί ότι δεν μπορούν οι συνολικές ώρες της ημέρας να ξεπερνούν τις 8. Στην τυχόν περίπτωση που ο χρήστης επιθυμεί να συμπληρώσει ώρες σε μία ημέρα η οποία είναι σαββατοκύριακο ή αργία τότε πρέπει να επιλέξει το κουτάκι που βρίσκεται στην κορυφή του ημερολογίου. Τέλος δεν μπορεί να δημιουργηθεί φύλλο χρονοχρέωσης όταν οι συνολικές ώρες του μήνα ξεπερνούν τις 143.



Projects Contracts Timesheets People Link Wps Users Logout N

Search by email

Insert User

User Entry

User email: it22118@hua.gr

Role: user

Total Projects: 0

Total Humans: 0

Total Work Packages: 0

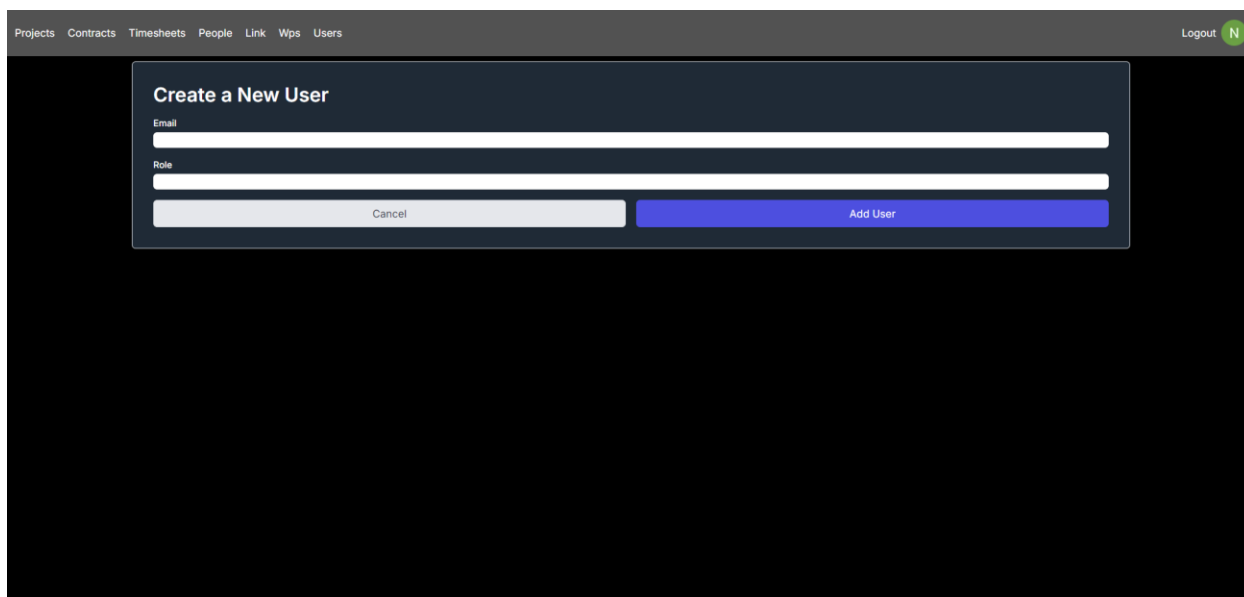
Total Contracts: 0

Total Timesheets: 0

Delete

Εικόνα 12: Προβολή χρηστών





The screenshot displays a web interface for creating a new user. At the top, a navigation bar contains links for 'Projects', 'Contracts', 'Timesheets', 'People', 'Link', 'Wps', and 'Users'. On the right side of the navigation bar is a 'Logout' button next to a user profile icon. The main content area features a dark-themed modal window titled 'Create a New User'. Inside this modal, there are two text input fields: the first is labeled 'Email' and the second is labeled 'Role'. Below these fields are two buttons: a light blue 'Cancel' button and a dark blue 'Add User' button.

Εικόνα 13: Δημιουργία νέου χρήστη

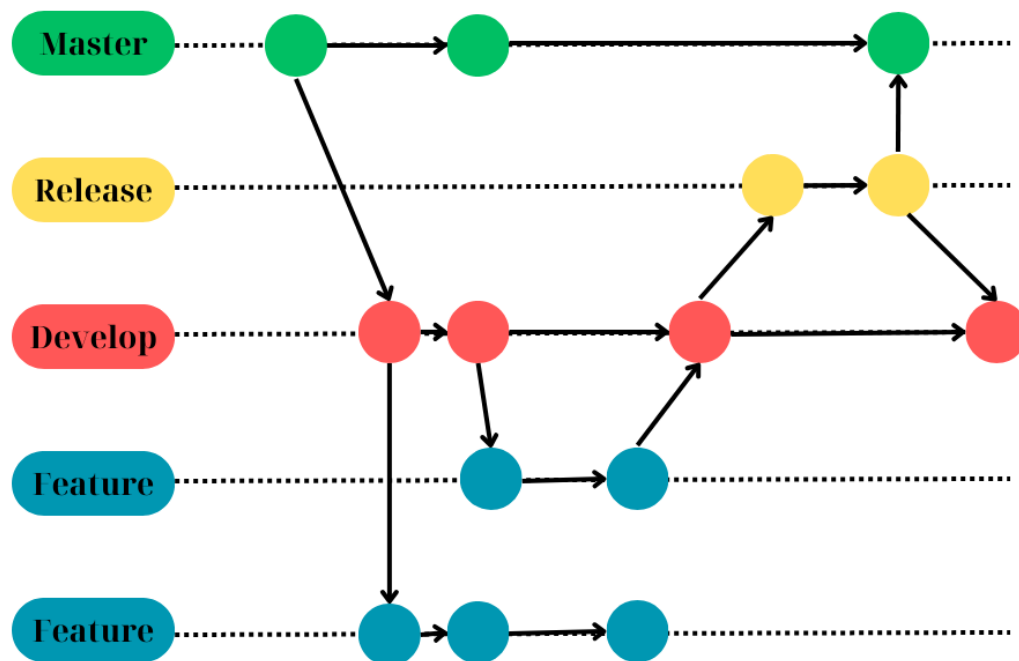
Αν ο χρήστης που είναι συνδεδεμένος είναι admin τότε έχει την δυνατότητα δημιουργίας νέων χρηστών στην ιστοσελίδα. Ο χρήστης πρέπει απλά να συμπληρώσει τον ηλεκτρονικό ταχυδρομικό λογαριασμό και τον ρόλο που θα έχει ο νέος χρήστης στην ιστοσελίδα, είτε αυτός είναι απλός χρήστης είτε admin.

## 3.6 Εργαλεία Ανάπτυξης

### 3.6.1. Git

Το Git είναι ένα σύστημα ελέγχου εκδόσεων για την παρακολούθηση αλλαγών σε αρχεία του υπολογιστή, που δημιουργήθηκε το 2005 από τον Linus Torvalds. Δεν είναι απαραίτητη κάποια συγκεκριμένη γλώσσα προγραμματισμού ή framework και μπορεί να χρησιμοποιηθεί ακόμη και σε απλές στατικές HTML ιστοσελίδες. Το σύστημα ελέγχου εκδόσεων αποθηκεύει συνεχώς παλιές εκδόσεις του έργου και μπορεί κάποιος να επαναφέρει κάποια συγκεκριμένη έκδοση. Περαιτέρω, ο σκοπός του Git είναι η συνεργασία, δεδομένου ότι προσφέρει στους προγραμματιστές ένα κοινό περιβάλλον, και συνεχώς ενημερώνει ποιος έκανε την κάθε αλλαγή και τι ακριβώς άλλαξε. Για αυτό τον λόγο, το έργο θα αναπτύσσεται σαν ένα σύνολο από την αρχή και θα υπάρξει εξοικονόμηση χρόνου λόγω της μειωμένης ύπαρξης συγκρούσεων. Η συλλογή των εκδόσεων μπορεί να είναι πολύ κρίσιμη για τους προγραμματιστές, αφού η αποθήκευση εκδόσεων ενός έργου μετά την επεξεργασία του είναι πολύ ουσιώδης. Επομένως, το Git επιτρέπει στους προγραμματιστές να δημιουργήσουν σημεία αποθήκευσης (commits) κατά την διάρκεια ενός έργου. Αυτά τα σημεία αποθήκευσης επιτρέπουν στους προγραμματιστές να επιστρέψουν σε κάποιο σημείο στο οποίο κάθε συνάρτηση στο έργο λειτουργεί κανονικά, μετά από κάποια λανθασμένη αλλαγή που μπορεί να έγινε. Στο Git οι διακλαδώσεις (branches) διασυνδέουν τα σημεία αποθήκευσης και όταν ένα Git αποθετήριο (repository) δημιουργείται για πρώτη φορά το Git αρχικοποιεί μία διακλάδωση η οποία ονομάζεται master, και όσο οι προγραμματιστές δημιουργούν νέα σημεία αποθήκευσης, τόσο η διακλάδωση master αναπτύσσεται. Επίσης προσφέρει την δυνατότητα της δημιουργίας διακλάδωσης συγκεκριμένου χαρακτηριστικού (feature branch), ο οποίος είναι ένας όρος που

χρησιμοποιείται κατά την ανάπτυξη της εφαρμογής για την εισαγωγή ενός νέου χαρακτηριστικού χωρίς να αλλάξει η ροή της κύριας διακλάδωσης. Με αποτέλεσμα, ένας προγραμματιστής να μπορεί να δουλέψει στην διακλάδωση του χαρακτηριστικού και οι υπόλοιποι να ασχολούνται με την επεξεργασία της κεντρικής διακλάδωσης. Όταν το χαρακτηριστικό είναι έτοιμο και λειτουργικό, μπορεί να συγχωνευτεί με την κεντρική διακλάδωση και δημιουργείται ένα καινούργιο σημείο αποθήκευσης.



Εικόνα 14: Παράδειγμα Git διακλαδώσεων

### 3.6.2. Visual Studio Code

Το Visual Studio Code (VsCode) προσφέρει στους προγραμματιστές μία νέα επιλογή, η οποία συνδυάζει την απλότητα και διευκολύνει την εμπειρία της χρήσης ενός επεξεργαστή κώδικα, παρέχοντας ό,τι οι προγραμματιστές χρειάζονται για τον βασικό κύκλο επεξεργασίας κώδικα και αποσφαλμάτωσης. Είναι ο πρώτος επεξεργαστής κώδικα στην οικογένεια με τα εργαλεία του visual studio, με την δυνατότητα εκτέλεσης σε περιβάλλοντα Windows, OS X και Linux. Το VsCode διαθέτει ένα ισχυρό και γρήγορο επεξεργαστή κώδικα, ιδανικό για καθημερινή χρήση, συμπεριλαμβανομένου εύκολη πλοήγηση στον κώδικα, προσαρμόσιμες συνδέσεις πλήκτρων, υπογράμμιση λάθους σύνταξης, αυτόματη αντιστοίχιση παρενθέσεων και ενσωματωμένη υποστήριξη για δεκάδες γλώσσες. Οι προγραμματιστές συνήθως χρειάζονται να δουλέψουν με κώδικα περισσότερο από ότι με απλό κείμενο. Το VsCode υποστηρίζει την αυτόματη συμπλήρωση κώδικα και πλούσια σημασιολογική πλοήγηση. Ενσωματώνεται με υπάρχοντα εργαλεία και μπορεί να εκτελεί δοκιμές και να εκτελεί σχεδόν οποιαδήποτε εντολή μέσα από το ίδιο. Τα σφάλματα και οι προειδοποιήσεις είναι ορατά και καθιστούν τη ροή εργασιών πολύ εύκολη και ταχύτερη.

## 3.7. Διάθεση, Χρήση και Προσαρμογή της Εφαρμογής

### 3.7.1. Διάθεση Εφαρμογής

Ο κώδικας της εφαρμογής είναι διαθέσιμος σε ένα αποθετήριο στο GitHub, παρέχοντας εύκολη πρόσβαση σε τρίτα άτομα που επιθυμούν να εγκαταστήσουν, να δοκιμάσουν ή να προσαρμόσουν την εφαρμογή. Το αποθετήριο περιέχει όλα τα αρχεία, τις εξαρτήσεις και τις οδηγίες που απαιτούνται για τη σωστή λειτουργία της εφαρμογής, ενώ ο σύνδεσμος προς το αποθετήριο βρίσκεται [εδώ](#). Για να εγκαταστήσετε την εφαρμογή τοπικά ή σε απομακρυσμένο server, θα πρέπει να ακολουθήσετε μια σειρά από βήματα που περιλαμβάνουν τη λήψη του κώδικα από το αποθετήριο GitHub, την εγκατάσταση των απαραίτητων εξαρτήσεων, και τη ρύθμιση του περιβάλλοντος λειτουργίας. Ο χρήστης πρέπει να έχει εγκαταστήσει, στο μηχάνημα στο οποίο επιχειρεί να εκτελέσει την εφαρμογή, την βάση MongoDB, το NodeJS και τέλος το NestJS. Αφού έχει γίνει ήδη η λήψη του κώδικα και η εγκατάσταση των απαραίτητων εργαλείων, συνέχεια έχει η ρύθμιση του περιβάλλοντος. Ο χρήστης θα πρέπει να επισκεφτεί τον ιστότοπο του Firebase για να δημιουργήσει ένα καινούργιο έργο και να ενεργοποιήσει την υπηρεσία της επαλήθευσης. Οι μέθοδοι εισόδου που θα πρέπει να ενεργοποιηθούν είναι μέσω Google και μέσω λογαριασμού ηλεκτρονικού ταχυδρομείου/κωδικού για τον πρώτο διαχειριστή της εφαρμογής. Να σημειωθεί ότι η δημιουργία νέων λογαριασμών από τους χρήστες στην ιστοσελίδα πρέπει να είναι απενεργοποιημένη. Μόνο ο διαχειριστής έχει αυτή την δυνατότητα. Στον φάκελο του Backend ο χρήστης πρέπει να δημιουργήσει ένα αρχείο ονόματι `private_key.json` και να το συμπληρώσει με τις πληροφορίες που βρίσκονται στις ρυθμίσεις του έργου στο Firebase, στην υπό-κατηγορία Service Accounts. Υπάρχει και ένα αρχείο για τις μεταβλητές του συστήματος που περιλαμβάνει την θύρα στην οποία εκτελείται το πρόγραμμα και ένας σύνδεσμος για την βάση δεδομένων. Να σημειωθεί ότι δεν είναι απαραίτητα αυτά τα στοιχεία και υπάρχουν προεπιλεγμένες ρυθμίσεις στην περίπτωση που δεν συμπληρωθούν. Στον φάκελο του Frontend υπάρχει ένα αρχείο πρότυπο το οποίο ο χρήστης πρέπει να συμπληρώσει με πληροφορίες πάλι από τις ρυθμίσεις του Firebase, στην υπό-κατηγορία General και έπειτα να το μετονομάσει για να μην περιέχει την λέξη πρότυπο.

### 3.7.2. Χρήση Εφαρμογής

Εφόσον ο χρήστης έχει ακολουθήσει τα παραπάνω βήματα, έχει πλέον την δυνατότητα εκτέλεσης της εφαρμογής. Για τη χρήση της εφαρμογής, υπάρχουν τρεις βασικοί τρόποι εκτέλεσης: μέσω Docker, μέσω PM2, και απλά μέσω Node.js. Κάθε μέθοδος προσφέρει διαφορετικά πλεονεκτήματα, ανάλογα με το επίπεδο ευελιξίας και διαχείρισης που απαιτείται.

- Η εκτέλεση μέσω **Docker** είναι ιδανική για όσους θέλουν μια απομονωμένη, τυποποιημένη ρύθμιση. Το Docker εξασφαλίζει ότι η εφαρμογή θα τρέχει σε ένα ελεγχόμενο περιβάλλον, ανεξάρτητα από τις διαφορές του λειτουργικού συστήματος του χρήστη, προσφέροντας σταθερότητα και συνέπεια.
- Η εκτέλεση μέσω **PM2** είναι ιδανική για τη διαχείριση της εφαρμογής σε περιβάλλον τύπου server, επιτρέποντας την παρακολούθηση και την αυτόματη επανεκκίνηση της

εφαρμογής σε περίπτωση σφαλμάτων. Το PM2 προσφέρει λειτουργίες για αυτόματη διαχείριση των διεργασιών και είναι κατάλληλο για παραγωγικά περιβάλλοντα.

- Τέλος, η εκτέλεση απλά μέσω **NodeJS** προσφέρει έναν γρήγορο και απλό τρόπο για να τρέξετε την εφαρμογή τοπικά. Αυτή η μέθοδος είναι ιδανική για ανάπτυξη και debugging, επιτρέποντας στους χρήστες να εκτελέσουν την εφαρμογή απευθείας μέσω του Node.js runtime.

Για την εκτέλεση μέσω Docker ο χρήστης πρέπει απλά να εκτελέσει την εντολή `docker compose up -d`. Στις περιπτώσεις που ο χρήστης χρησιμοποιήσει pm2 ή απλά NodeJS θα πρέπει πρώτα να εγκαταστήσει τα απαραίτητα πακέτα και να κάνει build τον κώδικα. Η διαδικασία αυτή πρέπει να γίνει τόσο στο Backend όσο και στο Frontend. Για την χρήση του pm2 ο χρήστης θα τρέξει την εντολή `npm start pm2` για το Frontend και για το Backend. Στην περίπτωση του NodeJS, ο χρήστης θα χρειαστεί να εκτελέσει τις εντολές `npm run start -- -p PORT_NUMBER` για το Frontend και `npm run start:prod` για το Backend.

### 3.7.3. Προσαρμογή Εφαρμογής

Η προσαρμογή της εφαρμογής επιτρέπει την τροποποίηση και επέκταση της βασικής λειτουργικότητας ώστε να καλύψει ειδικές ανάγκες ή απαιτήσεις. Η δομή των φακέλων του Backend της εφαρμογής είναι σχεδιασμένη με τρόπο που διευκολύνει την οργάνωση και την επέκταση των λειτουργιών. Στον κεντρικό φάκελο `src`, κάθε οντότητα της εφαρμογής (π.χ. έργα, συμβάσεις, χρήστες) έχει τον δικό της ξεχωριστό φάκελο. Κάθε φάκελος περιέχει όλα τα απαραίτητα αρχεία για τη λειτουργία της αντίστοιχης οντότητας, όπως η δομή της βάσης δεδομένων, ο controller που διαχειρίζεται τις αιτήσεις και αποκρίσεις για τη συγκεκριμένη οντότητα, και τα services που εκτελούν την επιχειρηματική λογική. Αυτή η οργάνωση διευκολύνει την ανάπτυξη και την τροποποίηση του κώδικα, καθώς κάθε οντότητα έχει τη δική της αυτοτελή διαχείριση και επέκταση. Μέσα στον Backend φάκελο επίσης υπάρχει ένας φάκελος με `scripts` τα οποία μπορεί ο χρήστης να αξιοποιήσει, όπως την δημιουργία του πρώτου διαχειριστή της εφαρμογής. Η δομή του frontend φακέλου, ακολουθεί μια ξεκάθαρη οργάνωση, διατηρώντας τη λειτουργικότητα και την ευελιξία. Στον φάκελο `src` και ειδικά στον υποφάκελο `app`, υπάρχουν διάφοροι υποφάκελοι που αντιπροσωπεύουν διαφορετικές οντότητες και λειτουργίες της εφαρμογής.

- Ο φάκελος `api` περιέχει τα endpoints που χρησιμοποιούνται για την επικοινωνία με το backend.
- Οι φάκελοι `contracts`, `people`, `projects`, `timesheets`, `users` και `wps` περιέχουν τις σελίδες και τα components που σχετίζονται με τις αντίστοιχες οντότητες, όπως συμβάσεις, μέλη, έργα, φύλλα χρονοχρέωσης, χρήστες και εργασίες.
- Τα αρχεία όπως `_app.tsx`, `layout.tsx` και `globals.css` αφορούν τη γενική ρύθμιση της εφαρμογής, με το `_app.tsx` να καθορίζει το global wrapper της εφαρμογής, το `layout.tsx` να χειρίζεται τη δομή του layout και το `globals.css` να περιέχει τα global στυλ.

Επιπλέον, οι φάκελοι `components`, `context`, `firebase`, `tools`, `types` και `utils` είναι υπεύθυνοι για τη διαχείριση και επαναχρησιμοποίηση διάφορων τμημάτων κώδικα, όπως τα κοινά `components`, η διαχείριση του `state`, οι ενσωματώσεις με το `Firebase`, τα εργαλεία και οι βοηθητικές συναρτήσεις που διευκολύνουν τις λειτουργίες της εφαρμογής.

## ΚΕΦ.4: Αξιολόγηση συστήματος/εφαρμογής

### 4.1 Αξιολόγηση απόδοσης του συστήματος

Η αξιολόγηση της απόδοσης του συστήματος γίνεται μέσω μίας σειράς δοκιμών, σχεδιασμένες για την μετριοποίηση της ανταπόκρισης, της σταθερότητας και της αποτελεσματικότητας υπό διαφορετικές συνθήκες. Για την συγκεκριμένη μέτρηση οι βασικές ρυθμίσεις που χρησιμοποιήθηκαν ήταν η κάθε δοκιμή να είναι δέκα λεπτά ακριβώς, με τους συνολικούς χρήστες να ανέρχονται σε είκοσι. Δοκιμάστηκαν οι λειτουργίες Fixed, Ramp Up, Spike Up, Peak. Τα αποτελέσματα τα οποία μελετήθηκαν είναι ο χρόνος ανταπόκρισης του συστήματος, πόσα αιτήματα το σύστημα μπορούσε να εξυπηρετήσει κάθε δευτερόλεπτο και ο συνολικός αριθμός των αιτημάτων. Στην λειτουργία Fixed οι 20 χρήστες στέλνουν σταθερά αιτήματα κατά την διάρκεια της δοκιμής για να μετρηθεί η απόδοση του συστήματος υπό σταθερό φόρτο. Στην Ramp Up οι χρήστες αυξάνονται από έναν έως είκοσι κατά την διάρκεια των πρώτων πέντε λεπτών και για τα επόμενα πέντε λεπτά παραμένουν στους είκοσι. Στην Spike Up υπάρχουν ξαφνικές αυξήσεις των χρηστών από μηδέν έως και είκοσι σε ένα μικρό διάστημα των τριάντα δευτερολέπτων. Τέλος στην Peak λειτουργία μεταβάλλεται ο φόρτος των χρηστών από υψηλές και μικρές ποσότητες. Για παράδειγμα είκοσι χρήστες στέλνουν αιτήματα για ένα λεπτό και έπειτα πέντε χρήστες στέλνουν αιτήματα για επίσης ένα λεπτό, με την διαδικασία αυτή να επαναλαμβάνεται μέχρι την λήξη της δοκιμής. Το πείραμα έλαβε μέρος σε δύο συσκευές, στον τοπικό υπολογιστή που έγινε και η ανάπτυξη της εφαρμογής και σε έναν διακομιστή(server) με τοποθεσία την Γερμανία. Οι προδιαγραφές των συσκευών και τα αποτελέσματα της έρευνας παρουσιάζονται στους πίνακες 24,25,26,27.

CPU	AMD Ryzen 5 5600
RAM	24GB

Πίνακας 24: Προδιαγραφές τοπικού υπολογιστή

	Συνολικές Δοκιμές	Αριθμός εξυπηρέτησης ανά δευτερόλεπτο	Χρόνος ανταπόκρισης
Fixed	18000 req	31.13 req/sec	50ms
Ramp Up	13703 req	22.68 req/sec	45ms
Spike Up	2800 req	4.63 req/sec	35ms
Peak	10052 req	16.53 req/sec	40ms

Πίνακας 25: Αποτελέσματα δοκιμών απόδοσης τοπικού υπολογιστή

CPU	Intel Xeon
RAM	2GB

Πίνακας 26: Προδιαγραφές διακομιστή

	Συνολικές Δοκιμές	Αριθμός εξυπηρέτησης ανά δευτερόλεπτο	Χρόνος ανταπόκρισης
Fixed	10495 req	17.31 req/sec	57ms
Ramp Up	7613 req	12.54 req/sec	56ms
Spike Up	2045 req	3.37 req/sec	56ms
Peak	5463 req	8.99 req/sec	57ms

Πίνακας 27: Αποτελέσματα δοκιμών απόδοσης διακομιστή

Για το παραπάνω πείραμα χρησιμοποιήθηκε η εφαρμογή Postman για να παραχθούν οι δοκιμές. Το Postman είναι ένα ευρέως χρησιμοποιούμενο εργαλείο ανάπτυξης και δοκιμών API, το οποίο επιτρέπει στους προγραμματιστές να σχεδιάζουν, να αναλύουν και να δοκιμάζουν τα APIs που δημιουργούν. Μέσω της φιλικής προς το χρήστη διεπαφής του, οι χρήστες μπορούν εύκολα να κάνουν αιτήσεις HTTP, όπως GET, POST, PUT, και DELETE, καθώς και να βλέπουν τις αντίστοιχες αποκρίσεις από τον server. Το Postman είναι ιδιαίτερα χρήσιμο στη φάση της ανάπτυξης, καθώς βοηθά στην απομόνωση προβλημάτων, τον έλεγχο της λειτουργικότητας των APIs και τη βελτίωση της δια λειτουργικότητας μεταξύ των διαφορετικών στοιχείων μιας εφαρμογής. Παρέχει επίσης δυνατότητες για τη δημιουργία collections από αιτήματα, τα οποία μπορούν να αποθηκευτούν και να μοιραστούν για μελλοντική χρήση ή ομαδικές δοκιμές.

Σημαντική λειτουργία του Postman είναι τα performance runs, τα οποία χρησιμοποιούνται για τη μέτρηση της απόδοσης των APIs υπό διαφορετικά φορτία και συνθήκες. Τα performance runs επιτρέπουν τη διενέργεια stress testing και load testing, προσομοιώνοντας μεγάλο αριθμό αιτήσεων προς ένα API για να εξεταστεί πώς ανταποκρίνεται το σύστημα σε περιόδους υψηλής χρήσης. Μέσα από αυτά τα τεστ, μπορεί κανείς να αναλύσει την απόκριση του API σε όρους ταχύτητας, σταθερότητας και ανθεκτικότητας, εντοπίζοντας τυχόν σημεία συμφόρησης ή καθυστερήσεις. Οι δοκιμές που εκτελέστηκαν μέσω του Postman χρησιμοποιήθηκαν για να διασφαλίσουν ότι το API της εφαρμογής λειτουργεί αποτελεσματικά και μπορεί να αντέξει την αυξημένη χρήση χωρίς απώλεια απόδοσης.

## 4.2 Συγκριτική Ανάλυση με Παρόμοιες Εφαρμογές

Ένας επίσης χρήσιμος τρόπος για την αξιολόγηση ενός συστήματος είναι η σύγκριση του προγράμματος που δημιουργήθηκε μαζί με υπάρχοντα συστήματα που προσφέρουν παρόμοιες

ή και ακριβώς ίδιες λειτουργίες. Η σύγκριση αυτή θα τονίσει τα πλεονεκτήματα της τωρινής εφαρμογής και τι λειτουργίες μπορούν να υλοποιηθούν στο μέλλον. Για την επιλογή των εξωτερικών παρόμοιων εφαρμογών χρησιμοποιήθηκαν τα εξής κριτήρια. Οι εφαρμογές πρέπει να προσφέρουν σχετικά παρόμοιες λειτουργίες, να στοχεύουν να εξυπηρετήσουν παρόμοιο κοινό χρηστών, να έχουν αναπτυχθεί με παρόμοια ή ίδια εργαλεία και τεχνολογίες και τα εξωτερικά συστήματα να είναι αναγνωρίσιμα στην αγορά. Τα κριτήρια αυτά οδήγησαν στην επιλογή των τριών εφαρμογών: Asana, Trello, Microsoft Project. Η σύγκριση αυτών των εργαλείων θα επικεντρωθεί κυρίως στις λειτουργίες που παρέχουν, τον σχεδιασμό της διεπαφής που χρησιμοποιούν οι χρήστες, την επίδοση του συστήματος, την δυνατότητα κλιμάκωσης, το ποσοστό ευχαρίστησης χρηστών και τέλος την προσβασιμότητα. Παρακάτω παρουσιάζεται η σύγκριση μεταξύ των εφαρμογών σε μορφή λίστας και ένας συνοπτικός Πίνακας 28: Συγκριτική Ανάλυση με Παρόμοιες Εφαρμογές.

## 1. Λειτουργικότητα

- **Asana:** Προσφέρει δυνατότητες διαχείρισης έργων, αναθέσεων εργασιών, παρακολούθησης προόδου, συνεργασίας ομάδας και ενσωματώνεται με πολλές άλλες πλατφόρμες (π.χ. Slack, Google Drive). Είναι κατάλληλο για ομάδες που διαχειρίζονται πολλαπλά έργα και καθήκοντα.
- **Trello:** Χρησιμοποιεί έναν απλό και οπτικά οργανωμένο πίνακα Kanban για τη διαχείριση έργων. Εστιάζει στη διαχείριση εργασιών με κάρτες, λίστες και ταμπλό. Έχει λιγότερες δυνατότητες από το Asana, αλλά είναι πολύ ευέλικτο και εύκολο στη χρήση.
- **Microsoft Project:** Ισχυρό εργαλείο διαχείρισης έργων με έμφαση στην κατανομή πόρων, προϋπολογισμούς, προγραμματισμό και χρονοδιαγράμματα. Είναι πιο κατάλληλο για μεγάλες επιχειρήσεις και σύνθετα έργα που απαιτούν αυστηρή διαχείριση πόρων.
- **Πτυχιακή:** Εστιάζει αποκλειστικά στη διαχείριση φύλλων χρονοχρέωσης έργων. Είναι πιο εξειδικευμένη στη διαχείριση χρόνου και καταχώρησης χρονοχρέωσης, ενώ προσφέρει και προσαρμοσμένες αναφορές για τους διαχειριστές των έργων.

## 2. Ευχρηστία

- **Asana:** Παρά τις πολλές δυνατότητες, η καμπύλη εκμάθησης μπορεί να είναι απότομη για νέους χρήστες. Ωστόσο, η διεπαφή είναι καθαρή και προσφέρει πληθώρα επιλογών για την παρακολούθηση της προόδου.
- **Trello:** Εξαιρετικά εύκολο στη χρήση, ακόμα και για νέους χρήστες. Η απλή διάταξη Kanban διευκολύνει την παρακολούθηση εργασιών με λίγες κινήσεις.
- **Microsoft Project:** Πολύπλοκο εργαλείο, με μια σχετικά υψηλή καμπύλη εκμάθησης λόγω της πληθώρας χαρακτηριστικών. Απαιτεί εκπαίδευση για την πλήρη αξιοποίηση των δυνατοτήτων του.



- **Πτυχιακή:** Εστιάζει στην απλότητα και την ευχρηστία για τη συγκεκριμένη χρήση. Οι διαχειριστές μπορούν να καταχωρούν και να παρακολουθούν τις χρονοχρεώσεις των μελών εύκολα, με μια καθαρή και διαισθητική διεπαφή που δεν απαιτεί πολύπλοκες ρυθμίσεις.

### 3. Σχεδίαση και Οπτική Διάταξη

- **Asana:** Η διεπαφή είναι σύγχρονη και οπτικά ελκυστική, με πολλές επιλογές για την προβολή των εργασιών (π.χ. λίστα, πίνακας, ημερολόγιο).
- **Trello:** Η βασική σχεδίαση του Trello με τους πίνακες Kanban είναι απλή, καθαρή και οπτικά ευχάριστη. Ιδανικό για χρήστες που προτιμούν να βλέπουν τα έργα τους οργανωμένα σε πίνακες.
- **Microsoft Project:** Η σχεδίαση είναι περισσότερο λειτουργική παρά οπτικά ελκυστική. Δίνεται έμφαση στην πληροφόρηση και τα δεδομένα παρά στην οπτική διάταξη.
- **Πτυχιακή:** Επικεντρώνεται σε μια μινιμαλιστική σχεδίαση για ευκολία στη χρήση. Η απλότητα της διεπαφής βοηθά τους διαχειριστές να εστιάσουν στην καταγραφή και την ανάλυση του χρόνου χωρίς περιττές πληροφορίες.

### 4. Απόδοση και Αξιοπιστία

- **Asana:** Γενικά γρήγορο και αξιόπιστο, αλλά μπορεί να επιβραδυνθεί σε πολύ μεγάλα έργα με πολλούς χρήστες.
- **Trello:** Ελαφρύ και ταχύτατο, καθώς είναι βασισμένο στην απλή αρχιτεκτονική Kanban. Οι επιδόσεις είναι σταθερές ακόμα και με πολλές κάρτες.
- **Microsoft Project:** Ισχυρό αλλά μπορεί να γίνει βαρύ και πιο αργό όταν διαχειρίζεται πολύ μεγάλα και σύνθετα έργα. Παρόλα αυτά, είναι αξιόπιστο και χρησιμοποιείται ευρέως σε μεγάλα έργα.
- **Πτυχιακή:** Προσαρμοσμένη στη διαχείριση χρονοχρεώσεων με έμφαση στη σταθερότητα και την ταχύτητα για συγκεκριμένη χρήση. Αξιόπιστη ακόμα και με μεγάλο όγκο δεδομένων χρονοχρέωσης.

### 5. Ασφάλεια

- **Asana:** Παρέχει ισχυρά πρωτόκολλα ασφαλείας και ενσωματώνει πιστοποιήσεις όπως το SOC 2 για την προστασία δεδομένων.
- **Trello:** Διαθέτει επίσης ισχυρή ασφάλεια, συμπεριλαμβανομένης της κρυπτογράφησης δεδομένων σε μετάδοση και αποθήκευση.
- **Microsoft Project:** Χρησιμοποιεί τα ισχυρά μέτρα ασφαλείας της Microsoft, με αυθεντικοποίηση πολλαπλών παραγόντων και προστασία δεδομένων σε επίπεδο εταιρείας.

- **Πτυχιακή:** Ακολουθεί σύγχρονες πρακτικές ασφαλείας με έμφαση στην προστασία των δεδομένων χρονοχρέωσης των έργων και των χρηστών, με αυστηρές διαδικασίες πρόσβασης.

Κριτήριο	Asana	Trello	Microsoft Project	Πτυχιακή
<b>Λειτουργικότητα</b>	Διαχείριση έργων και συνεργασίας.	Πίνακες Kanban για απλή διαχείριση εργασιών.	Σύνθετη διαχείριση έργων και πόρων.	Εξειδίκευση στη διαχείριση χρονοχρεώσεων.
<b>Ευχρηστία</b>	Εύχρηστη αλλά απαιτεί εκμάθηση.	Πολύ απλή και διαισθητική.	Πολύπλοκη, απαιτεί εκπαίδευση.	Απλή και εξειδικευμένη για χρονοχρεώσεις.
<b>Σχεδίαση</b>	Καθαρή και σύγχρονη.	Απλή και οπτική.	Λειτουργική και πρακτική.	Μινιμαλιστική και ευέλικτη.
<b>Απόδοση</b>	Σταθερή, αλλά αργή σε μεγάλα έργα.	Ελαφριά και γρήγορη.	Ισχυρή αλλά βαρύτερη.	Σταθερή και προσαρμοσμένη σε μεγάλο όγκο δεδομένων.
<b>Ασφάλεια</b>	Ισχυρή ασφάλεια (SOC 2).	Κρυπτογράφηση δεδομένων.	Προστασία εταιρικών δεδομένων.	Σύγχρονη ασφάλεια.

Πίνακας 28: Συγκριτική Ανάλυση με Παρόμοιες Εφαρμογές

Για να εξηγήσω καλύτερα τους όρους που αναφέρθηκαν στη σύγκριση, όπως οι πίνακες Kanban και η πιστοποίηση SOC 2, θα παρέχω μια σύντομη επεξήγηση της σημασίας τους. Οι πίνακες Kanban είναι ένα εργαλείο οπτικής διαχείρισης έργων που οργανώνει τις εργασίες σε κολώνες, οι οποίες αντιπροσωπεύουν στάδια της διαδικασίας, όπως "To Do", "In Progress" και "Done". Οι εργασίες αναπαρίστανται ως κάρτες που μετακινούνται μεταξύ των κολώνων καθώς προχωρούν, παρέχοντας μια σαφή εικόνα της προόδου. Η μέθοδος Kanban βοηθά τις ομάδες να διαχειρίζονται τη ροή εργασιών τους, βελτιστοποιώντας την παραγωγικότητα και αποφεύγοντας τα μπουτιλιαρίσματα, ενώ επιτρέπει συνεχή βελτίωση στη διαχείριση χρόνου και πόρων. Το SOC 2 (Service Organization Control 2) είναι ένα πρότυπο πιστοποίησης που διασφαλίζει ότι οι πάροχοι υπηρεσιών τηρούν αυστηρές διαδικασίες και ελέγχους για την προστασία της ασφάλειας, της διαθεσιμότητας, της ακεραιότητας επεξεργασίας, της εμπιστευτικότητας και της ιδιωτικότητας των δεδομένων. Αφορά κυρίως οργανισμούς που διαχειρίζονται δεδομένα πελατών στο cloud και απαιτεί τακτικούς ελέγχους για να διασφαλίζεται η συμμόρφωση με αυτά τα κριτήρια. Η πιστοποίηση SOC 2 είναι σημαντική για την οικοδόμηση εμπιστοσύνης και την ασφάλεια των πληροφοριών σε εταιρείες τεχνολογίας.

## ΚΕΦ.5: Συμπεράσματα και Μελλοντική Εργασία

Συμπερασματικά, η ανάπτυξη της εφαρμογής για τη διαχείριση φύλλων χρονοχρέωσης έργων πέτυχε τον στόχο της, προσφέροντας ένα λειτουργικό, φιλικό προς τον χρήστη και ασφαλές περιβάλλον εργασίας. Η υλοποίηση τόσο του backend όσο και του frontend διασφάλισε την ομαλή και αποτελεσματική διαχείριση των δεδομένων, ενώ οι δυνατότητες επεξεργασίας και αποθήκευσης καλύπτουν τις ανάγκες των χρηστών για αξιοπιστία και ευελιξία. Η ενσωμάτωση σύγχρονων τεχνολογιών, όπως η MongoDB και το Google Sign-In, βελτίωσε σημαντικά την εμπειρία του χρήστη και την ασφάλεια. Η εφαρμογή ανταποκρίνεται πλήρως στις απαιτήσεις, προσφέροντας μια λύση που συμβάλλει στην αποτελεσματική οργάνωση και παρακολούθηση των έργων. Το σύστημα μπορεί να έχει πετύχει τους βασικούς σκοπούς και λειτουργίες που απαιτήθηκαν αλλά υπάρχουν μελλοντικές ευκαιρίες για περαιτέρω προσθήκες. Στην δημιουργία ομάδας χρηστών για να μπορούν να διαχειρίζονται τα ίδια δεδομένα θα μπορούσε να προστεθεί ένας μηχανισμός ειδοποιήσεων για την αποδοχή ή όχι του αιτήματος αυτού. Οι ειδοποιήσεις που παρέχονται στους χρήστες θα μπορούσαν να είναι πιο αναλυτικές για την καλύτερη κατανόηση και καθοδήγηση τους. Δεδομένου ότι κάποιος χρήστης μπορεί να μην προλάβει να εισάγει τα φύλλα χρονοχρέωσης για κάποιον μήνα, θα ήταν χρήσιμο να επιλέγει τον συγκεκριμένου μήνα που τον ενδιαφέρει αντί για την αυτόματη επιλογή του τρέχοντος. Τέλος στις περιπτώσεις που υπάρχουν επιλογές με αναπτυσσόμενες λίστες(drop-down lists), θα ήταν ευνοϊκό να δημιουργηθεί ένας μηχανισμός αναζήτησης, χωρίς ο χρήστης να χρειάζεται να ψάξει όλη την λίστα.

## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

- [1] Jatana, N., Puri, S., Ahuja, M., Kathuria, I., & Gosain, D. (2012). A survey and comparison of relational and non-relational database. *International Journal of Engineering Research & Technology*, 1(6), 1-5.
- [2] Gyorödi, C., Gyorödi, R., & Sotoc, R. (2015). A comparative study of relational and non-relational database models in a web-based application. *International Journal of Advanced Computer Science and Applications*, 6(11), 78-83.
- [3] Kashyap, N. K., Pandey, B. K., Mandoria, H. L., & Kumar, A. (2016). A review of leading databases: Relational & non-relational database. *i-Manager's Journal on Information Technology*, 5(2), 34.
- [4] Thakur, N., & Gupta, N. (2021). Relational and Non-Relational Databases: A Review. *Journal of University of Shanghai for Science and Technology*, 23(8), 117-121.
- [5] Kolonko, K. (2018). Performance comparison of the most popular relational and non-relational database management systems.
- [6] Pham, A. D. (2020). Developing back-end of a web application with NestJS framework: Case: Integrity Oy's student management system.
- [7] Cherny, B. (2019). *Programming TypeScript: making your JavaScript applications scale*. O'Reilly Media.
- [8] Jansen, R. H., Vane, V., & De Wolff, I. G. (2016). *TypeScript: Modern JavaScript Development*. Paöckt Publishing Ltd.
- [9] Goldberg, J. (2022). *Learning TypeScript*. " O'Reilly Media, Inc."
- [10] Maharry, D. (2013). *TypeScript revealed*. Apress.