



Ανάπτυξη Κατανεμημένου
Συστήματος για Διαχείριση
Συσκευής DRMS μέσω
Ασύρματης Επικοινωνίας

Γεράσιμος Στεφάτος (2013030047)

Επιβλέπων: Μιχαήλ Γ. Λαγουδάκης

Καθηγητής: Ευτύχιος Κουτρούλης

Καθηγητής: Παναγιώτης Παρτσινέβελος (Σχολή ΜΗΧΟΠ)

Ημερομηνία: 31 Ιουλίου 2023

TECHNICAL UNIVERSITY OF CRETE
SCHOOL OF ELECTRICAL & COMPUTER ENGINEERING



Development of a Distributed System for Managing a DRMS Device through Wireless Communication

Gerasimos Stefatos (2013030047)

Supervisor: Michail Lagoudakis
Professor: Eutuxios Koutroulis
Professor: Panayiotis Partsinevelos (School of MRE)
Date: July 31, 2023

Περίληψη

Η ραγδαία πρόοδος της τεχνολογίας οδήγησε στην ανάγκη για αποτελεσματικότερη, ευέλικτη και απομακρυσμένη διαχείριση των τεχνολογικών συστημάτων, αλλά και για την ολοένα και πιο ομαλή ενσωμάτωσή τους στην καθημερινότητά μας. Τα κατανεμημένα συστήματα αποτελούν μια εξαιρετικά σημαντική πτυχή αυτής της τεχνολογικής προόδου. Η ενσωμάτωση αυτών των τεχνολογιών στον τομέα της γεώτρησης, και πιο συγκεκριμένα στα συστήματα μέτρησης αντίστασης διάτρησης (DRMS - Drilling Resistance Measurement Systems), δημιουργεί ευκαιρίες για ανάπτυξη καινοτομίας και επιτυγχάνει πιο αποτελεσματική διαχείριση. Σε αυτό το ευρύτερο πλαίσιο, η παρούσα διπλωματική εργασία προσπαθεί να συνδυάσει τις προηγμένες τεχνολογίες IoT (Internet-of-Things) με την εξειδίκευση των DRMS, δημιουργώντας ένα ολοκληρωμένο κατανεμημένο σύστημα, για τη βελτίωση και επέκταση των δυνατοτήτων του DRMS. Μέσω της αναβάθμισης της αρχιτεκτονικής ενός υφιστάμενου DRMS με έναν μικροελεγκτή ESP32, είμαστε σε θέση να ενσωματώσουμε δυνατότητες ασύρματης επικοινωνίας, επιτρέποντας την απομακρυσμένη παρακολούθηση και διαχείριση του DRMS. Αυτό το νέο κατανεμημένο σύστημα επιτρέπει στους χρήστες να ελέγχουν και να οργανώνουν το DRMS από οποιαδήποτε τοποθεσία στον κόσμο, μέσω μιας εύκολα προσβάσιμης βάσης δεδομένων για την αποθήκευση όλων των συλλεγόμενων δεδομένων και ενός γραφικού web interface για την παρακολούθηση των μετρήσεων και τον έλεγχο του συστήματος, προσφέροντας έτσι αυξημένη ευελιξία και αποδοτικότητα. Η σχεσιακή βάση δεδομένων βασίζεται σε SQL, ενώ η μεταφορά δεδομένων γίνεται εφικτή μέσω ενός διακομιστή Flask, SQL-Alchemy και REST-API. Η web interface εφαρμογή αναπτύχθηκε χρησιμοποιώντας JavaScript. Η ενσωμάτωση της τεχνολογίας IoT μεταμορφώνει τον τρόπο λειτουργίας και εφαρμογής των DRMS, ανοίγοντας νέες δυνατότητες για τη βιομηχανία και την επιστήμη.

Abstract

The rapid advancement of technology has led to a need for more effective, flexible, and remote management of technological systems, as well as their seamless integration into our daily lives. Distributed systems represent an exceptionally important aspect of this technological progress. The integration of these technologies into the field of drilling, more specifically into Drilling Resistance Measurement Systems (DRMS), creates opportunities for innovation and achieves more efficient management. In this broader context, this diploma thesis attempts to combine advanced Internet-of-Things (IoT) technologies with the specialization of DRMS, creating a comprehensive distributed system to improve and expand the capabilities of DRMS. By upgrading the architecture of an existing DRMS with an ESP32 microcontroller, we are able to incorporate wireless communication capabilities, allowing remote monitoring and management of the DRMS. This new distributed system allows users to control and organize DRMS from any location in the world, through an easily-accessible database for storing all the collected data and a graphical web interface for monitoring the measurements and controlling the system, thus offering increased flexibility and efficiency. The relational database relies on SQL, while data transfer becomes feasible through a Flask server, SQL-Alchemy, and REST-API. The web interface app was developed using JavaScript. The integration of IoT technology transforms the way DRMS operate and are applied, opening new possibilities for the industry and science.

Ευχαριστίες

Πρώτα απ'όλα, θα ήθελα να ευχαριστήσω την οικογένειά μου για την ανεκτίμητη στήριξή τους κατά τη διάρκεια των χρόνων σπουδών μου. Επίσης, τον επιβλέποντα καθηγητή μου, για τη στήριξη και την καθοδήγησή του κατά τη διάρκεια της διπλωματικής μου εργασίας. Μου εμπιστεύθηκε ένα ανεξερεύνητο πεδίο και η πρόκληση διεύρυνε τον πνευματικό μου ορίζοντα. Τέλος, τους συναδέλφους μου, για την αμέριστη στήριξή τους στη διάρκεια των σπουδών μου στα Χανιά.

Περιεχόμενα

1	Εισαγωγή	1
1.1	Κίνητρα	1
1.2	Συνεισφορά	1
1.3	Περίγραμμα	2
2	Εργαλεία και Τεχνολογίες	4
2.1	Εργαλεία	5
2.1.1	VsCode	5
2.1.2	Python	5
2.1.3	Pip (Package Installer for Python)	7
2.1.4	Flask	7
2.1.5	SQLAlchemy	8
2.1.6	Javascript	9
2.1.7	Postman	11
2.1.8	SQL	12
2.1.9	Bootstrap	13
2.1.10	Docker	13
2.1.11	ESP32	14
2.2	Τεχνολογίες	16
2.2.1	Representational State Transfer (REST)-Application Programming Interface (API)	16
2.2.2	Distrubed systems	17
3	Σχεδιασμός και Ανάπτυξη Συστήματος	19
3.1	Εισαγωγή	19

3.2	Υπάρχον Σύστημα	19
3.3	Περιορισμοί Υπάρχοντος Συστήματος	20
3.4	Αναβάθμιση Συστήματος	21
3.5	Απομακρυσμένος Έλεγχος και Παρακολούθηση	22
3.6	Εγκατάσταση και επικοινωνία	23
4	Υλοποίηση	24
4.1	Ανάλυση στοιχείων του συστήματος	26
4.2	Esp32	26
4.2.1	Υλοποίηση και Προγραμματισμός Esp32	26
4.3	Database	31
4.3.1	Υλοποίηση Βάσης Δεδομένων	31
4.4	Flask Server	34
4.4.1	Υλοποίηση Flask Server	34
4.5	Web-Γραφικό Περιβάλλον	56
4.5.1	Υλοποίηση Γραφικού Περιβάλλοντος	57
4.6	Dockerization	68
5	Συμπεράσματα	69
5.1	Μελλοντικές εργασίες	70
	Βιβλιογραφία	71

Κατάλογος Σχημάτων

2.1	Διάγραμμα Λειτουργίας ESP32	16
3.1	Υπάρχον Σύστημα	20
4.1	Διάγραμμα του Κατανεμημένου DRMS Συστήματος	25
4.2	ESP32 Δίκτυο	28
4.3	WIFI Μενού	29
4.4	Διαθέσιμα Δίκτυα	30
4.5	Καταχώρηση Στοιχείων Δικτύου	30
4.6	Επιτυχής Σύνδεση	31
4.7	Σχήμα Βάσης Δεδομένων	32
4.8	Διακομιστής Flask	36
4.9	Αποτυχημένη Εγγραφή Χρήστη: Ελλιπή Στοιχεία	37
4.10	Αποτυχημένη Εγγραφή Χρήστη: Διπλοκαταχώρηση	38
4.11	Επιτυχής Εγγραφή Χρήστη	39
4.12	Επιτυχής Εγγραφή Χρήστη στην Βάση Δεδομένων	39
4.13	Αποτυχημένο Αίτημα Σύνδεσης: Ελλιπή Στοιχεία	40
4.14	Αποτυχημένη Σύνδεση Χρήστη: Ο Χρήστης δεν Υπάρχει	41
4.15	Λάθος Κωδικός Πρόσβασης	42
4.16	Επιτυχής Σύνδεση Χρήστη	43
4.17	Καταχώρηση Μοναδικού Αναγνωριστικού στην Βάση Δεδομένων	43
4.18	Αποτυχημένη Αποσύνδεση	44
4.19	Επιτυχής Αποσύνδεση	45
4.20	Μη Εξουσιοδοτημένος Χρήστης	46
4.21	Προφίλ Υλικών	47
4.22	Αποτυχημένο Αίτημα Προσθήκης Υλικού: Ελλιπή Στοιχεία	48

4.23 Επιτυχής Προσθήκη Υλικού	49
4.24 Προφίλ Υλικών στη Βάση Δεδομένων	49
4.25 Επιτυχής Διαγραφή Υλικού	50
4.26 Αποτυχημένη Προσθήκη Μέτρησης: Ελλιπή Στοιχεία	51
4.27 Αποτυχημένη Καταχώρηση Μέτρησης: Διπλοκαταχώρηση Μέτρη- σης	52
4.28 Επιτυχής Αποθήκευση Ονόματος Μέτρησης	53
4.29 Αίτημα Λήψης Μετρήσεων	54
4.30 Διαγραφή Μέτρησης	55
4.31 Διάγραμμα Ροής Εγγραφής/Σύνδεσης Χρήστη	57
4.32 Αποτυχημένη Εγγραφή Χρήστη	58
4.33 Αποτυχημένη Σύνδεση Χρήστη: Μη Έγκυρα στοιχεία Χρήστη .	59
4.34 Μπάρα Πλοήγησης	60
4.35 Αρχική Σελίδα	60
4.36 Διάγραμμα Ροής Έναρξης Μέτρησης	61
4.37 Διπλοκαταχώρηση Μέτρησης	62
4.38 Δεδομένα Αισθητήρων	63
4.39 Διάγραμμα Ροής Εισαγωγής Προφίλ Υλικού	64
4.40 Διπλοκαταχώρηση Μέτρησης	65
4.41 Επιτυχημένη Εγγραφή Προφίλ Υλικού	66
4.42 Διαγραφή Προφίλ Υλικού	66
4.43 Λήψη Μέτρησης	67
4.44 Διαγραφή Μέτρησης	67

Κατάλογος Ακρωνυμίων

IoT Internet of Things

DRMS Drilling Resistance Measurement System

API Application Programming Interface

REST Representational State Transfer

Pip Package Installer for Python

HTTP Hypertext Transfer Protocol

SQL Structured Query Language

RDBMS Relational Database Management System

CSS Cascading Style Sheets

VSCode Visual Studio Code

UART Universal Asynchronous Receiver/Transmitter

SPI Serial Peripheral Interface

I2C Inter-Integrated Circuit

EEPROM Electrically Erasable Programmable Read-Only Memory

SRAM Static Random Access Memory

JSON JavaScript Object Notation

ID Identifier

SD Secure Digital

DIY Do It Yourself

Κεφάλαιο 1

Εισαγωγή

1.1 Κίνητρα

Η σημασία των κατανεμημένων συστημάτων στον σύγχρονο κόσμο αυξάνεται διαρκώς, καθώς τα περιβάλλοντα υπολογιστών και τα δίκτυα γίνονται ολοένα πιο πολύπλοκα και διασυνδεδεμένα. Τα κατανεμημένα συστήματα επιτρέπουν την αποδοτική και αξιόπιστη διαχείριση πόρων, τη βελτίωση της απόδοσης και της ευελιξίας, καθώς και τη μείωση του κόστους και της πολυπλοκότητας της διαχείρισης των διαφόρων συστατικών ενός συστήματος. Επιπλέον, τα κατανεμημένα συστήματα ενθαρρύνουν την αυτονομία των μεμονωμένων στοιχείων, επιτρέποντας την αυτόματη προσαρμογή και ανάπτυξη του συστήματος με βάση τις απαιτήσεις και τις αλλαγές στο περιβάλλον. Ως αποτέλεσμα, τα κατανεμημένα συστήματα προσφέρουν ισχυρές δυνατότητες για την υποστήριξη εφαρμογών και υπηρεσιών σε μια πληθώρα τομέων, όπως οι τηλεπικοινωνίες, η επιστήμη των δεδομένων, η παραγωγή και η διαχείριση ενέργειας, καθιστώντας τα ένα κρίσιμο συστατικό στην κατασκευή και τη λειτουργία των υποδομών του 21ου αιώνα.

1.2 Συνεισφορά

Στην παρούσα διπλωματική εργασία, έχουμε υλοποιήσει ένα κατανεμημένο σύστημα DRMS προκειμένου να ανταποκριθούμε στις αυξημένες απαιτήσεις της τεχνολογίας γεωτρήσεων και της διαχείρισης πληροφοριών. Αυτό το κατανεμη-

μένο σύστημα προσφέρει μια αποτελεσματική και ευέλικτη λύση για τη συλλογή, αποθήκευση, ανάλυση και παρουσίαση των δεδομένων που προκύπτουν από τις διαδικασίες γεωτρήσεων. Με τη χρήση τεχνολογιών, όπως ο ESP32 για ασύρματη επικοινωνία, καινοτόμων βάσεων δεδομένων για τη διαχείριση των πληροφοριών, καθώς και web-client εργαλείων για την επικοινωνία με τους χρήστες, το κατανεμημένο DRMS επιτρέπει τη βελτίωση της απόδοσης, της αξιοπιστίας και της χρηστικότητας του συστήματος. Η υλοποίηση αυτή αναμένεται να συμβάλει στη βελτίωση των διαδικασιών λήψης αποφάσεων σχετικά με τις γεωτρήσεις, καθώς και στην εξασφάλιση της βιωσιμότητας των έργων γεωτρήσεων.

1.3 Περίγραμμα

Η εν λόγω διπλωματική εργασία αναδύεται από την αποστολή να αναλύσει, ερευνήσει και εφαρμόσει σύγχρονες πρακτικές και τεχνολογίες. Αναδεικνύεται μέσα από πέντε περιεκτικά κεφάλαια, καθένα εξ αυτών προσφέρει εποικοδομητικά στοιχεία στο συνολικό πλαίσιο της εργασίας.

Κεφάλαιο 1: Στο Κεφάλαιο 1 (Εισαγωγή) γίνεται παρουσίαση των κινήτρων και των συνεισφορών της έρευνας, καθορίζοντας τον τόνο και το πεδίο του ακαδημαϊκού εγχειρήματος.

Κεφάλαιο 2: Το Κεφάλαιο 2 (Εργαλεία και Τεχνολογίες) αφιερώνεται στην περιγραφή των εργαλείων και των τεχνολογιών που επιλέχθηκαν για την υλοποίηση του έργου, από την εξειδίκευση σε συγκεκριμένες γλώσσες προγραμματισμού όπως Python, C/C++ και JavaScript, έως τη χρήση προηγμένων εργαλείων και πλατφορμών, όπως το Docker και ο ESP32.

Κεφάλαιο 3: Η ουσία του συστήματος αναδύεται μέσα από το Κεφάλαιο 3 (Σχεδιασμός και Ανάπτυξη Συστήματος), όπου γίνεται εκτενής αναφορά στον σχεδιασμό του, με έμφαση στην κατανεμημένη αρχιτεκτονική, την απομακρυσμένη εποπτεία και τον έλεγχο, καθώς και στην εγκατάσταση και επικοινωνία του συστήματος.

Κεφάλαιο 4: Το Κεφάλαιο 4 (Υλοποίηση) αποκαλύπτει την ολοκληρωμένη υλοποίηση του συστήματος, από την ανάλυση των συστατικών του συστήματος, την προγραμματιστική αποτύπωση του ESP32, την δημιουργία της βάσης δεδομένων, την υλοποίηση του εξυπηρετητή Flask, την ανάπτυξη του web γραφικού περιβάλλοντος της ιστοσελίδας και την εφαρμογή της διαδικασίας Dockerization.

Κεφάλαιο 5: Κλείνοντας, το Κεφάλαιο 5 (Συμπεράσματα) αποτελεί τον καθρέφτη της εργασίας, παρουσιάζοντας τα συμπεράσματα και τις διδαχές που απορρέουν από το παρόν εγχείρημα.

Κεφάλαιο 2

Εργαλεία και Τεχνολογίες

Στο παρόν κεφάλαιο της διπλωματικής εργασίας εξετάζουμε τα εργαλεία και τις τεχνολογίες που χρησιμοποιήθηκαν στη συγκεκριμένη έρευνα, αποσκοπώντας στην κατανόηση της σημασίας τους και του ρόλου που διαδραματίζουν στην επίτευξη του στόχου μας.

Τα οποία ονομαστικά είναι:

1. *VsCode*
2. *Python*
3. *Pip*
4. *Flask*
5. *SQLAlchemy*
6. *Javascript*
7. *Postman*
8. *SQL*
9. *ESP32*
10. *Bootstrap*

11. *Docker*
12. *REST – API*
13. *Distrubed systems*

2.1 Εργαλεία

2.1.1 VsCode

Το Visual Studio Code (VSCode) είναι ένας δωρεάν και ανοιχτού κώδικα επεξεργαστής κώδικα που προσφέρει υποστήριξη για πολλαπλές γλώσσες προγραμματισμού, καθιστώντας τον ιδανικό για προγραμματιστές που εργάζονται με διαφορετικά περιβάλλοντα και πλαίσια. Το VSCode προσφέρει επίσης εξαιρετική ευελιξία, καθώς επιτρέπει την προσαρμογή του περιβάλλοντος εργασίας μέσω επεκτάσεων και θεμάτων. Επιπροσθέτως το Visual Studio Code περιλαμβάνει ισχυρές δυνατότητες αποσφαλμάτωσης και διαχείρισης κώδικα, καθώς και ενσωματωμένη υποστήριξη για το Git, το δημοφιλές σύστημα ελέγχου εκδόσεων. Αυτό επιτρέπει στους προγραμματιστές να διαχειρίζονται εύκολα τις αλλαγές στον κώδικα, να παρακολουθούν το ιστορικό των αλλαγών και να συνεργάζονται με άλλους προγραμματιστές σε έργα. Τέλος, το VSCode διαθέτει μια ενεργή και αναπτυσσόμενη κοινότητα που συνεισφέρει στη βελτίωση και τη δημιουργία επεκτάσεων για το περιβάλλον εργασίας. Αυτές οι επεκτάσεις προσθέτουν νέες λειτουργίες, εργαλεία και χαρακτηριστικά που βελτιώνουν την παραγωγικότητα και την εμπειρία του χρήστη. Με τη χρήση των επεκτάσεων, οι προγραμματιστές μπορούν να προσαρμόσουν το VSCode στις ανάγκες τους, καθιστώντας το ένα από τα πιο δημοφιλή εργαλεία ανάπτυξης λογισμικού [19].

2.1.2 Python

Η Python αποτελεί μια αντικειμενοστραφή και υψηλού επιπέδου γλώσσα προγραμματισμού, η οποία έχει αποκτήσει ιδιαίτερη δημοτικότητα τα τελευταία

χρόνια, χάρη στη ραγδαία ανάπτυξη της τεχνολογίας και των απαιτήσεων των σύγχρονων συστημάτων. Καταξιωμένη για τις εκτεταμένες δυνατότητές της και την ευχρηστιά της, η Python βρίσκεται στο επίκεντρο της προγραμματιστικής κοινότητας. Μεταξύ των εφαρμογών που μπορεί να υποστηρίξει η Python περιλαμβάνονται η Τεχνητή Νοημοσύνη, το Web Development Scripting, η Διαχείριση Μεγάλων Δεδομένων και η Σύνδεση με Βάσεις Δεδομένων. Η επιλογή της γλώσσας Python βασίζεται στα πλεονεκτήματα που προσφέρει σε σχέση με άλλες γλώσσες, μερικά εκ των οποίων παρουσιάζονται παρακάτω [20].

Ευκολία στη χρήση: Η Python διαθέτει ένα εξαιρετικά απλό και καθαρό συντακτικό, που την καθιστά προσιτή και ελκυστική επιλογή για όσους αναζητούν μια γλώσσα προγραμματισμού. Χάρη στην απλότητα αυτή, τα προγράμματα που δημιουργούνται με την Python είναι εύκολα αναγνώσιμα και η ανίχνευση των σφαλμάτων γίνεται με αυξημένη ταχύτητα και αποτελεσματικότητα. Αυτό επιτρέπει στους προγραμματιστές να αναπτύσσουν λογισμικό υψηλής ποιότητας, ενισχύοντας την παραγωγικότητα και τη δημιουργικότητα.

Ευελιξία: Η Python αποτελεί μια εξαιρετικά ευέλικτη γλώσσα προγραμματισμού, καθώς προσφέρει στους χρήστες της τη δυνατότητα να επιλέξουν ανάμεσα στον διαδικαστικό ή τον αντικειμενοστρεφή προγραμματισμό. Πέραν αυτού, χάρη στη χρήση των ενσωματωμένων, αλλά και των third party πακέτων, η Python καθίσταται εξαιρετικά προσαρμοστική στη δημιουργία διαφορετικών εφαρμογών. Αυτή η ευελιξία και προσαρμοστικότητα επιτρέπει στους προγραμματιστές να αναπτύξουν λύσεις για ποικίλες εφαρμογές με αυξημένη αποδοτικότητα και καινοτομία.

Φορητότητα: Ένα από τα σημαντικά πλεονεκτήματα της Python είναι η δυνατότητά της να εκτελείται χωρίς προβλήματα σε διαφορετικά λειτουργικά συστήματα. Αυτό οφείλεται στο γεγονός ότι ένα πρόγραμμα Python δεν απαιτεί compiler, επιτρέποντας έτσι στους χρήστες να εκτελούν απευθείας τον κώδικα Python. Κατ' αυτόν τον τρόπο, η Python προσφέρει μια ευέλικτη και κομψή λύση για την ανάπτυξη εφαρμογών σε ποικίλα περιβάλλοντα λειτουργικών συστημάτων.

Μηδενική διαδικασία μεταγλώττισης: Η Python είναι μια interpreted γλώσσα προγραμματισμού, πρακτικά σημαίνοντας ότι το compile του κώδικα πραγματοποιείται κατά τη διάρκεια της εκτέλεσης. Αυτό έχει ως αποτέλεσμα την απουσία ανάγκης για έναν ξεχωριστό compiler, ελαχιστοποίηση του χρόνου που απαιτείται για το compile και την εξάλειψη των compile errors. Κατ' αυτόν τον τρόπο, η Python προσφέρει μια ευέλικτη και αποτελεσματική λύση για την ανάπτυξη και την εκτέλεση προγραμμάτων, ενώ ταυτόχρονα εξοικονομεί χρόνο και πόρους.

2.1.3 Pip (Package Installer for Python)

Το pip είναι ένα εργαλείο διαχείρισης πακέτων που επιτρέπει στους προγραμματιστές να εγκαταστήσουν, αναβαθμίσουν και απεγκαταστήσουν εύκολα βιβλιοθήκες Python και πακέτα. Αυτό καθιστά τη διαδικασία εγκατάστασης των απαραίτητων εξαρτήσεων για τα έργα Python απλή και αυτοματοποιημένη, βελτιώνοντας την παραγωγικότητα των προγραμματιστών. Επιπλέον, το pip είναι εύκολο στη χρήση και διαθέτει ένα απλό σύνολο εντολών. Αυτό καθιστά το pip μια βασική εργαλειοθήκη για προγραμματιστές Python ανεξαρτήτως του επιπέδου εμπειρίας τους. Τέλος το pip δίνει πρόσβαση στον προγραμματιστή σε μια τεράστια συλλογή πακέτων και βιβλιοθηκών που διατίθενται στο Python Package Index (PyPI), το οποίο είναι ένα αποθετήριο που περιέχει χιλιάδες πακέτα ανοιχτού κώδικα από την κοινότητα της Python. Αυτό επιτρέπει στους προγραμματιστές να εκμεταλλεύονται τις γνώσεις και την εμπειρία της κοινότητας για να βρουν λύσεις σε κοινά προβλήματα και να επιταχύνουν την ανάπτυξη των έργων τους [21].

2.1.4 Flask

Το Flask είναι ένα ελαφρύ web framework για τη γλώσσα προγραμματισμού Python, ιδανικό για τη δημιουργία μικρών και μεσαίου μεγέθους ιστοσελίδων

και εφαρμογών. Χάρη στην απλότητα και την ευελιξία του, το Flask επιτρέπει στους προγραμματιστές να αρχίσουν γρήγορα την ανάπτυξη των ιστοσελίδων, εστιάζοντας στη λειτουργικότητα και την εμπειρία χρήστη. Μερικά από τα πλεονεκτήματα του Flask παρατίθενται παρακάτω [22].

Απλότητα: Η απλότητα και η ευκολία χρήσης του Flask καθιστούν την ανάπτυξη και τη διαχείριση της εφαρμογής γρήγορη και αποτελεσματική. Χάρη στην ελαχιστοποίηση των εξαρτήσεων και της πολυπλοκότητας, ο Flask επιτρέπει την εστίαση στη λειτουργικότητα της εφαρμογής, αποφεύγοντας την ανάγκη για περιττές προσαρμογές.

Ευελιξία: Η ευελιξία του Flask επιτρέπει την εύκολη επέκταση και προσαρμογή της εφαρμογής, χάρη στην υποστήριξη για επιπλέον επεκτάσεις και πρόσθετα. Αυτό καθιστά το Flask ιδανικό για τη δημιουργία εφαρμογών που προσαρμόζονται εύκολα σε διαφορετικές απαιτήσεις και περιβάλλοντα, προσφέροντας τη δυνατότητα για κλιμακούμενες και εξατομικευμένες λύσεις.

Υποστήριξη και κοινότητα: Η κοινότητα και η υποστήριξη που περιβάλλει το Flask είναι εξαιρετικά ενεργή και αναπτυσσόμενη. Αυτό σημαίνει ότι υπάρχει μια πλούσια πηγή γνώσης και πόρων, που περιλαμβάνει τεκμηρίωση, οδηγούς, παραδείγματα κώδικα και συμβουλές για τη βελτιστοποίηση της ανάπτυξης και της λειτουργίας της εφαρμογής.

Ευκολία: Το Flask υποστηρίζει τη δημιουργία RESTful-APIs, προσφέροντας μια ισχυρή βάση για την ανάπτυξη και την επικοινωνία μεταξύ των διαφορετικών συστατικών της εφαρμογής. Αυτό επιτρέπει την εύκολη ολοκλήρωση με άλλες υπηρεσίες και συστήματα, δημιουργώντας μια αποδοτική και εύχρηστη αρχιτεκτονική.

2.1.5 SQLAlchemy

Το SQLAlchemy αποτελεί μια ευέλικτη και ισχυρή βιβλιοθήκη γλώσσας προγραμματισμού Python, η οποία παρέχει έναν πλήρη και ευέλικτο SQL toolkit,

καθώς και έναν object-relational mapper (ORM).

Ο σκοπός του SQLAlchemy είναι διττός: αφενός, να δώσει τη δυνατότητα στον προγραμματιστή να εκμεταλλευτεί πλήρως την ισχύ και την ευελιξία της γλώσσας SQL, αφετέρου, να παράσχει ένα απλό και pythonic περιβάλλον για την εργασία με βάσεις δεδομένων SQL [10].

Ας προχωρήσουμε σε μια πιο λεπτομερή εξέταση των κύριων πλεονεκτημάτων του SQLAlchemy:

Πολυ-διάλεκτος SQL: Το SQLAlchemy υποστηρίζει μια πληθώρα διαλέκτων SQL, όπως MySQL, PostgreSQL, SQLite κ.ά., επιτρέποντας στους προγραμματιστές να επιλέξουν την βάση δεδομένων που καλύπτει καλύτερα τις ανάγκες τους.

Object-Relational Mapping (ORM): Η δυνατότητα ORM του SQLAlchemy καθιστά την αλληλεπίδραση με βάσεις δεδομένων απλούστερη και πιο αποτελεσματική, καθώς μετατρέπει τις κλήσεις στην βάση δεδομένων σε αντικείμενα Python, τα οποία μπορούν να χειριστούν και να τροποποιηθούν απευθείας από τον κώδικα.

Ευελιξία στις ερωτήσεις SQL: Το SQLAlchemy επιτρέπει τη διατύπωση σύνθετων ερωτημάτων SQL με τη χρήση φυσικής σύνταξης Python, διευκολύνοντας την εκφραστικότητα και τη δυνατότητα δημιουργίας πιο πολύπλοκων ερωτημάτων.

Ασφάλεια: Το SQLAlchemy παρέχει ενσωματωμένη προστασία από SQL Injection attacks, ενός από τα πιο συνηθισμένα είδη επιθέσεων σε εφαρμογές βάσεων δεδομένων, χρησιμοποιώντας bound parameters για τις εντολές SQL.

2.1.6 Javascript

Η JavaScript είναι μια από τις πιο δημοφιλείς γλώσσες προγραμματισμού στον κόσμο, κυρίως λόγω της ευκολίας χρήσης και εκμάθησής της. Είναι μια υψηλού επιπέδου γλώσσα που προσφέρει απλές συντακτικές δομές και ευέλικτες δυνατότητες, καθιστώντας την κατάλληλη για αρχάριους και έμπειρους

προγραμματιστές. Μερικά απο τα πλεονεκτήματα της παρατίθενται παρακάτω [16].

Υποστήριξη σε όλα τα μεγάλα προγράμματα περιήγησης: Ένα από τα μεγάλα πλεονεκτήματα της JavaScript είναι η ευρεία υποστήριξη σε όλα τα μεγάλα προγράμματα περιήγησης, όπως το Google Chrome, το Mozilla Firefox, το Safari και το Microsoft Edge. Αυτό σημαίνει ότι οι εφαρμογές που χρησιμοποιούν JavaScript μπορούν να λειτουργήσουν απρόσκοπτα σε διάφορες πλατφόρμες και συσκευές, καθιστώντας την ιδανική για την ανάπτυξη διαδραστικών ιστοσελίδων και εφαρμογών.

Ευελιξία και δυνατότητες επέκτασης: Η JavaScript είναι μια ευέλικτη γλώσσα προγραμματισμού που υποστηρίζει πολλαπλά παραδείγματα προγραμματισμού, όπως τον αντικειμενοστρεφή, τον συναρτησιακό και τον ανταγωνιστικό προγραμματισμό. Αυτό επιτρέπει στους προγραμματιστές να χρησιμοποιούν τη γλώσσα με διαφορετικούς τρόπους ανάλογα με τις ανάγκες τους και το είδος του έργου. Επιπλέον, η JavaScript διαθέτει μια τεράστια βιβλιοθήκη από πρόσθετα και πλατφόρμες ανάπτυξης, όπως το Node.js, το React και το Angular, που επεκτείνουν τις δυνατότητές της γλώσσας και παρέχουν πρόσθετες λειτουργίες.

Υποστήριξη ασύγχρονου προγραμματισμού: Η JavaScript υποστηρίζει τον ασύγχρονο προγραμματισμό, κάτι που σημαίνει ότι οι προγραμματιστές μπορούν να δημιουργήσουν εφαρμογές που μπορούν να εκτελέσουν πολλαπλές λειτουργίες ταυτόχρονα, χωρίς να επηρεάζεται η απόδοση του συστήματος. Αυτό επιτρέπει τη δημιουργία γρήγορων και αποκρίσιμων εφαρμογών, οι οποίες μπορούν να ανταποκριθούν άμεσα στις ενέργειες των χρηστών και να επεξεργαστούν μεγάλους όγκους δεδομένων χωρίς καθυστερήσεις. Αυτή η ικανότητα είναι ιδιαίτερα σημαντική για την ανάπτυξη προηγμένων web εφαρμογών και πλατφόρμες πραγματικού χρόνου, όπως οι διαδραστικές ιστοσελίδες, τα συστήματα διαχείρισης περιεχομένου (CMS) και τα online παιχνίδια.

2.1.7 Postman

Το Postman είναι ένα δημοφιλές εργαλείο που χρησιμοποιείται από προγραμματιστές και δοκιμαστές για την ανάπτυξη, δοκιμή και τεκμηρίωση RESTful APIs. Αυτό το εργαλείο παρέχει ένα εύχρηστο γραφικό περιβάλλον που επιτρέπει στους χρήστες να δημιουργούν, επεξεργάζονται και εκτελούν αιτήματα Hypertext Transfer Protocol (HTTP), να ελέγχουν τις αποκρίσεις και να διαχειρίζονται τις πληροφορίες που σχετίζονται με τα APIs [13] [14].

Τα πλεονεκτήματα του Postman περιλαμβάνουν την ευκολία χρήσης, τη δυνατότητα οργάνωσης και διαχείρισης των APIs σε συλλογές, καθώς και την ικανότητα να εξάγει και να εισάγει τα αρχεία παραμετροποίησης για εύκολη συνεργασία και κοινή χρήση. Το Postman υποστηρίζει επίσης την αυτοματοποίηση των δοκιμών και τη δημιουργία αυτόματων δοκιμών με τη χρήση διαφόρων γλωσσών προγραμματισμού.

Ένα ακόμη πλεονέκτημα του Postman είναι η δυνατότητα να προσαρμοστεί στις διαφορετικές ανάγκες των χρηστών, καθώς παρέχει υποστήριξη για διάφορα πρωτόκολλα και μεθόδους αυθεντικοποίησης. Επιπλέον, το Postman ενσωματώνει δυνατότητες παρακολούθησης και επίβλεψης της απόδοσης των APIs σε πραγματικό χρόνο, βοηθώντας τους χρήστες να εντοπίζουν και να επιλύουν προβλήματα γρηγορότερα.

Με τη χρήση του Postman, οι προγραμματιστές και οι δοκιμαστές μπορούν να επικεντρωθούν στη δημιουργία και τη βελτίωση των APIs, αφού το εργαλείο απλοποιεί τη διαδικασία δοκιμής και τεκμηρίωσης. Επιπλέον, η ομάδα ανάπτυξης μπορεί να βελτιώσει τη συνεργασία και την επικοινωνία μεταξύ των μελών της, χάρη στις δυνατότητες κοινής χρήσης και συγχρονισμού του Postman.

Συνολικά, το Postman αποτελεί ένα πολύτιμο εργαλείο για την ανάπτυξη, τη δοκιμή και τη συντήρηση των APIs, καθώς παρέχει μια ευρεία γκάμα λειτουργιών και πλεονεκτημάτων που βοηθούν τους προγραμματιστές και τους δοκιμαστές να εργάζονται αποτελεσματικά και αποδοτικά.

2.1.8 SQL

Η Structured Query Language, γνωστή ως SQL, είναι μια γλώσσα προγραμματισμού που χρησιμοποιείται για τη διαχείριση και την επεξεργασία δεδομένων που αποθηκεύονται σε σχεσιακές βάσεις δεδομένων (RDBMS). Αυτή η γλώσσα είναι το αποτέλεσμα ετών έρευνας και ανάπτυξης στην επιστήμη των δεδομένων, και έχει καταστεί το πρότυπο για τη διαχείριση δεδομένων σε σχεσιακά συστήματα. Η SQL προσφέρει μια ισχυρή, ευέλικτη και διαισθητική σύνταξη για τη διαχείριση δεδομένων, επιτρέποντας την αποτελεσματική επεξεργασία μεγάλων όγκων δεδομένων με ακρίβεια και αποδοτικότητα. Παρακάτω, θα εξετάσουμε σε βάθος τα πλεονεκτήματα που προσφέρει η SQL στον τομέα της διαχείρισης δεδομένων [4].

Διαχείριση περίπλοκων σχέσεων: Μέσω της SQL βάσης δεδομένων επιτυγχάνεται η ακριβής και αποτελεσματική διαχείριση περίπλοκων και διασυνδεδεμένων σχέσεων μεταξύ των πινάκων.

Ευελιξία και επεκτασιμότητα: Οι σχεσιακές βάσεις δεδομένων είναι εξαιρετικά ευέλικτες και επεκτάσιμες, δίνοντας τη δυνατότητα για προσαρμογή στις νέες απαιτήσεις και αλλαγές στην εφαρμογή.

Ασφάλεια και συνέπεια δεδομένων: Η SQL βάση δεδομένων προσφέρει αυξημένη ασφάλεια και συνέπεια στη διαχείριση των δεδομένων, λόγω της τήρησης προτύπων και κανόνων όπως το ACID.

Εύκολη πρόσβαση και ανάκτηση δεδομένων: Χάρη στη γλώσσα SQL, η ανάκτηση και η επεξεργασία δεδομένων γίνεται εύκολα και αποτελεσματικά. Οι προγραμματιστές μπορούν να χρησιμοποιήσουν σύνθετα ερωτήματα για να λάβουν ακριβώς τα αποτελέσματα που χρειάζονται, εξοικονομώντας χρόνο και πόρους.

Υποστήριξη και κοινότητα: Οι SQL βάσεις δεδομένων έχουν μια μεγάλη κοινότητα προγραμματιστών και εκτεταμένη υποστήριξη, που παρέχει πλο-

ύσια πηγή γνώσης, βοήθειας και εργαλείων. Αυτό καθιστά ευκολότερη την ανάπτυξη και τη συντήρηση της βάσης δεδομένων.

2.1.9 Bootstrap

Το Bootstrap είναι ένα από τα πιο δημοφιλή front-end frameworks για την ανάπτυξη responsive ιστοσελίδων. Προσφέρει ευκολία χρήσης, καθώς περιλαμβάνει έτοιμα στοιχεία σχεδίασης, όπως κουμπιά, φόρμες, πλαίσια κειμένου και πολλά άλλα. Οι προγραμματιστές μπορούν να επωφεληθούν από τις ενσωματωμένες κλάσεις Cascading Style Sheets (CSS) και τα πρότυπα σχεδίασης για να δημιουργήσουν σταθερές και ελκυστικές ιστοσελίδες, χωρίς την ανάγκη να ξεκινήσουν από το μηδέν.

Ένα από τα βασικά πλεονεκτήματα του Bootstrap είναι η υποστήριξη responsive σχεδιασμού. Το framework αυτόματα προσαρμόζει τη διάταξη και τα στοιχεία της ιστοσελίδας σε διάφορες συσκευές και μεγέθη οθόνης, όπως κινητά, tablet και desktop υπολογιστές. Αυτό εξασφαλίζει ότι οι ιστοσελίδες θα φαίνονται καλά και θα λειτουργούν ομαλά σε διάφορα περιβάλλοντα και πλατφόρμες. Η συμβατότητα με πολλαπλές συσκευές είναι ουσιαστική για την προσφορά μιας ευχάριστης εμπειρίας χρήστη και τη μεγιστοποίηση της προβολής των ιστοσελίδων [17] [18].

2.1.10 Docker

Το Docker είναι ένα ανοιχτού κώδικα εργαλείο που επιτρέπει τη δημιουργία, την ανάπτυξη και τη διανομή εφαρμογών μέσω containers, εξασφαλίζοντας την απόλυτη απομόνωση των περιβαλλόντων εκτέλεσης. Το Docker έχει επαναπροσδιορίσει τον τρόπο με τον οποίο οι επιχειρήσεις και οι προγραμματιστές διαχειρίζονται τις εφαρμογές και τις υπηρεσίες, προσφέροντας αξεπέραστη ευελιξία, συνέπεια και αποδοτικότητα. Μέσω της χρήσης των containers, το Docker επιτρέπει την απομόνωση των εφαρμογών και των εξαρτήσεών τους, διευκολύνοντας την ανάπτυξη, την εκτέλεση και την κλιμάκωση εφαρμογών σε

οποιοδήποτε περιβάλλον. Ακολουθεί μια αναλυτική παρουσίαση των βασικών πλεονεκτημάτων που προσφέρει το Docker [15].

Ευελιξία: Τα Docker containers μπορούν να εκτελεστούν σε διάφορες πλατφόρμες και λειτουργικά συστήματα, διευκολύνοντας την εγκατάσταση και τη χρήση της εφαρμογής.

Επαναληψιμότητα: Η χρήση των Docker containers εξασφαλίζει ότι η εφαρμογή θα λειτουργεί με τον ίδιο τρόπο σε διαφορετικά περιβάλλοντα, αποφεύγοντας τα προβλήματα συμβατότητας.

Εύκολη υπηρεσιοκεντρική αρχιτεκτονική (Microservices): Τα Docker containers διευκολύνουν την κατασκευή και τον συντονισμό των υπηρεσιών που αποτελούν την εφαρμογή, βελτιώνοντας τη διαχείριση και την επεκτασιμότητα του συστήματος.

Διαχείριση πόρων: Τα Docker containers επιτρέπουν την αποδοτικότερη χρήση των πόρων του συστήματος, καθώς μοιράζονται το λειτουργικό σύστημα και χρησιμοποιούν περιορισμένους πόρους όταν δεν είναι σε λειτουργία.

Ασφάλεια: Τα Docker containers παρέχουν έναν αυξημένο βαθμό ασφάλειας, καθώς οι εφαρμογές εκτελούνται σε απομονωμένα περιβάλλοντα, μειώνοντας τους κινδύνους διαρροής πληροφοριών και επιθέσεων.

2.1.11 ESP32

Ο ESP32 είναι ένας ισχυρός και ευέλικτος μικροελεγκτής που έχει σχεδιαστεί ειδικά για IoT (Internet of Things) εφαρμογές. Προσφέρει μια ολοκληρωμένη λύση για τους προγραμματιστές, καθώς συνδυάζει Wi-Fi, Bluetooth, και πλούσιες δυνατότητες εισόδου-εξόδου (GPIO) σε ένα μικρό και οικονομικό

ολοκληρωμένο κύκλωμα. Αυτό τον καθιστά ιδανικό για την ανάπτυξη εφαρμογών, όπως ο ασύρματος έλεγχος συσκευών, η αυτοματοποίηση σπιτιού, και η παρακολούθηση περιβάλλοντος [5].

Ένα από τα βασικά πλεονεκτήματα του ESP32 είναι η χαμηλή κατανάλωση ενέργειας, που τον καθιστά κατάλληλο για εφαρμογές όπου η αυτονομία της μπαταρίας και η ενεργειακή απόδοση είναι κρίσιμης σημασίας. Χάρη στην προηγμένη τεχνολογία λειτουργίας χαμηλής ισχύος, ο ESP32 μπορεί να μεταβαίνει σε καταστάσεις χαμηλής κατανάλωσης όταν δεν χρησιμοποιείται, εξοικονομώντας έτσι ενέργεια και παρατείνοντας τη διάρκεια ζωής της μπαταρίας. Αυτή η ικανότητα είναι ιδιαίτερα σημαντική σε απομακρυσμένες και φορητές εφαρμογές, όπως συστήματα παρακολούθησης καιρικών συνθηκών, αισθητήρες περιβάλλοντος και εφαρμογές wearables.

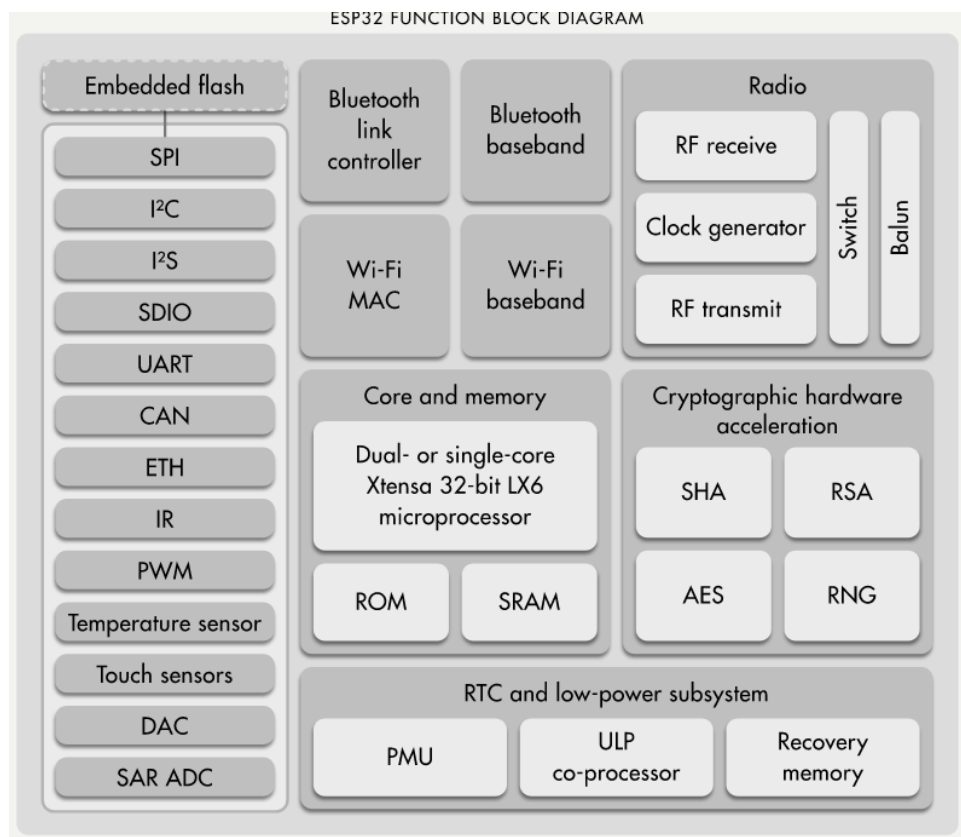
Ο ESP32 είναι ένας ισχυρός microcontroller με δύο πυρήνες και συχνότητα λειτουργίας έως και 240 MHz. Επίσης, ο ESP32 διαθέτει ενσωματωμένο Wi-Fi και Bluetooth. Η δυνατότητα αυτή σημαίνει ότι ο ESP32 μπορεί να συνδεθεί ασύρματα στο δίκτυο και να επικοινωνήσει με άλλες συσκευές, επιτρέποντας μια πιο ενσωματωμένη και αξιόπιστη λειτουργία του συστήματος.

Επίσης διαθέτει 512KB μνήμης RAM και 4MB μνήμης flash, αυτό καθιστά τον ESP32 ιδανικό για τη διαχείριση μεγάλου όγκου δεδομένων από αισθητήρες.

Επιπλέον μπορεί να προγραμματιστεί με διάφορους τρόπους, συμπεριλαμβανομένων των γλωσσών προγραμματισμού Arduino, MicroPython και Lua. Αυτό τον καθιστά εξαιρετικά εύκολο στη χρήση από επαγγελματίες και ερασιτέχνες.

Τέλος ο ESP32 προσφέρει ευελιξία και επεκτασιμότητα στους προγραμματιστές, καθώς υποστηρίζει μια πληθώρα περιφερειακών συσκευών και αισθητήρων. Οι προγραμματιστές μπορούν να επωφεληθούν από τις δυνατότητες GPIO του ESP32, όπως οι ψηφιακές και αναλογικές εισόδους-εξόδους, σειριακές επικοινωνίες (Universal Asynchronous Receiver/Transmitter (UART), Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I2C)), και άλλα πρωτόκολλα, για να συνδέσουν και να ελέγξουν διάφορες συσκευές και αισθητήρες. Αυτό τον καθιστά κατάλληλο για μια ευρεία γκάμα εφαρμογών, από απλές λύσεις όπως η αυτοματοποίηση φωτισμού, μέχρι πιο περίπλοκες, όπως η

ρομποτική και τα αυτοκινούμενα οχήματα.



Σχήμα 2.1: Διάγραμμα Λειτουργίας ESP32
Source: esp32.net

2.2 Τεχνολογίες

2.2.1 REST-API

Το REST-API (Representational State Transfer Application Programming Interface) είναι ένα μοντέλο αρχιτεκτονικής λογισμικού που έχει γίνει ιδιαίτερα δημοφιλές στην ανάπτυξη και επικοινωνία μεταξύ διαφορετικών υπηρεσιών και εφαρμογών μέσω του διαδικτύου. Χρησιμοποιεί το HTTP ως πρωτόκολλο επικοινωνίας και επιτρέπει στις εφαρμογές να αλληλεπιδρούν μεταξύ τους μέσω απλών και κατανοητών αιτημάτων και απαντήσεων. Με την χρήση του REST-API, οι προγραμματιστές μπορούν να δημιουργήσουν εύκολα, ευ-

έλικτα και επεκτάσιμα συστήματα [1].

Η αρχιτεκτονική του REST-API βασίζεται στην αρχή των πόρων (resources), που είναι οι βασικές οντότητες του συστήματος και μπορούν να αναπαρασταθούν με διευθύνσεις URL. Οι πόροι ανταποκρίνονται σε διάφορα στοιχεία και οντότητες της εφαρμογής, όπως δεδομένα, αρχεία και λειτουργίες. Το REST-API χρησιμοποιεί τις τυπικές μεθόδους του HTTP, όπως GET, POST, PUT και DELETE, για την εκτέλεση των βασικών λειτουργιών επί των πόρων.

Τέλος, ένα από τα μεγάλα πλεονεκτήματα του REST-API είναι η απλότητα και η κατανοησιμότητα της αρχιτεκτονικής του, καθώς και η ευρεία υποστήριξη από πληθώρα γλωσσών και πλατφορμών. Το REST-API είναι ανεξάρτητο από τη γλώσσα προγραμματισμού και την πλατφόρμα, προσφέροντας ευελιξία στους προγραμματιστές. Επιπλέον, τα REST-API μπορούν να επικοινωνήσουν με διαφορετικά μορφότυπα δεδομένων, όπως XML, JavaScript Object Notation (JSON) και HTML, καθιστώντας την ανταλλαγή πληροφοριών εύκολη και αποτελεσματική.

2.2.2 Distrubed systems

Τα καταναμημένα συστήματα υπολογιστών αποτελούν μια αρχιτεκτονική υπολογιστών που συνδυάζει πολλούς αυτόνομους υπολογιστές μέσω ενός δικτύου, με σκοπό την καλύτερη απόδοση, αξιοπιστία και αποτελεσματικότητα στην επεξεργασία και διαχείριση των δεδομένων. Ένα από τα κύρια πλεονεκτήματα των καταναμημένων συστημάτων είναι η ικανότητα να διαχειρίζονται μεγάλους όγκους δεδομένων και υπολογιστικές εργασίες με υψηλό βαθμό ταυτοχρονισμού, καθιστώντας τα ιδανικά για εφαρμογές, όπως τα cloud computing, τα κέντρα δεδομένων και η επεξεργασία μεγάλων όγκων δεδομένων (big data) [7].

Επιπλέον, τα καταναμημένα συστήματα προσφέρουν υψηλή ανθεκτικότητα και αξιοπιστία, καθώς η απώλεια ή η αποτυχία ενός κόμβου δεν οδηγεί στην αποτυχία ολόκληρου του συστήματος. Αυτό επιτρέπει στα καταναμημένα συστήματα να επιτυγχάνουν υψηλό βαθμό αντοχής σε σφάλματα και διαταρα-

χές, ενώ ταυτόχρονα διασφαλίζουν τη συνέχιση της λειτουργίας τους χωρίς σημαντική απώλεια απόδοσης. Μάλιστα, στην περίπτωση που κάποιος κόμβος αποτύχει, τα κατανεμημένα συστήματα μπορούν να ανακατανείμουν το φόρτο εργασίας στους υπόλοιπους κόμβους, εξασφαλίζοντας τη συνεχή λειτουργία της εφαρμογής.

Τέλος, τα κατανεμημένα συστήματα υπολογιστών επιτρέπουν την καλύτερη χρήση των πόρων και της υπολογιστικής ισχύος, καθώς επιτρέπουν την παράλληλη επεξεργασία δεδομένων και την αποδοτική κατανομή του φορτίου εργασίας μεταξύ των διαφόρων κόμβων. Αυτό οδηγεί σε μια πιο αποδοτική και οικονομική λύση συγκριτικά με τα κεντρικά συστήματα, καθώς μειώνει το κόστος απόκτησης και συντήρησης πιο ισχυρών και δαπανηρών υπολογιστών, καθώς και το κόστος ενέργειας και ψύξης.

Κεφάλαιο 3

Σχεδιασμός και Ανάπτυξη Συστήματος

3.1 Εισαγωγή

Το Drilling Resistance Measuring System (DRMS) είναι ένα ρομποτικό σύστημα διάτρησης, το οποίο μετρά τα χαρακτηριστικά των υλικών, όπως σκληρότητα και σαθρότητα, μέσω της καταγραφής τιμών από αισθητήρες όπως ροπής, πίεσης και στροφών, κατά τη διάρκεια της διάτρησης.

3.2 Υπάρχον Σύστημα

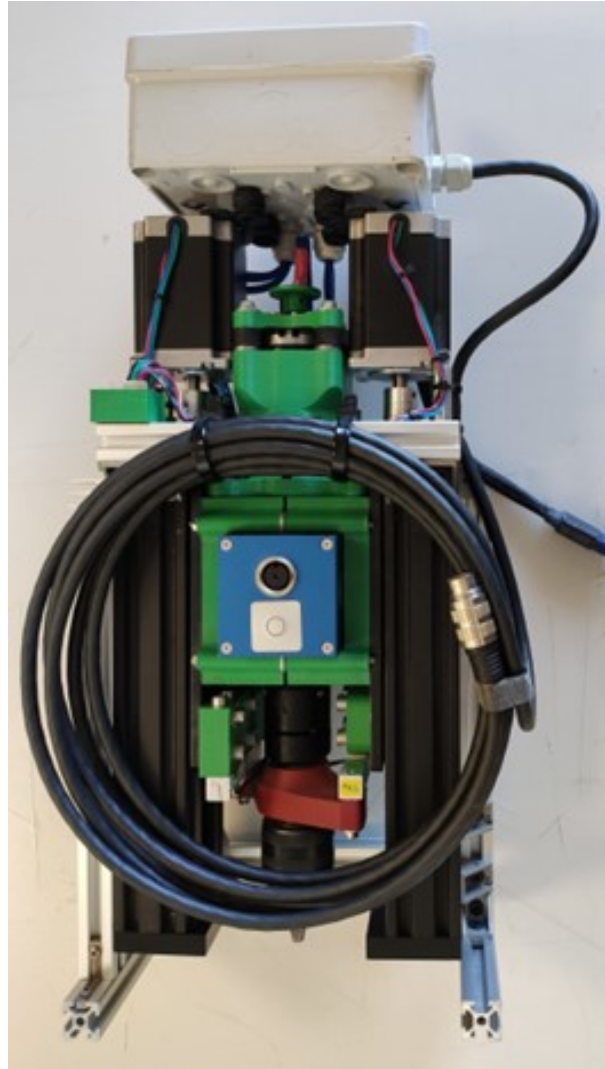
Το υπάρχον σύστημα χρησιμοποιούσε έναν μικροελεγκτή ATMEGA328P και κατέγραφε τα δεδομένα σε μία κάρτα Secure Digital (SD). Στη συνέχεια, τα δεδομένα αναλύονταν με τη χρήση προγραμμάτων MATLAB ή Python.

Ο ATMEGA328P είναι εξοπλισμένος με έναν 8-bit AVR RISC επεξεργαστή, 32KB flash μνήμη, 2KB Static Random Access Memory (SRAM), 1KB Electrically Erasable Programmable Read-Only Memory (EEPROM) και μια πληθώρα χαρακτηριστικών που περιλαμβάνουν 23 γενικής χρήσης I/O γραμμές, 6 κανάλια 10-bit ADC, 6 PWM κανάλια, καθώς και δυνατότητες σειριακής επικοινωνίας μέσω USART, SPI, και I2C.

Η λειτουργία του υπάρχοντος συστήματος επικεντρώνεται γύρω από τον ATMEGA328P, χρησιμοποιώντας τις δυνατότητές του για την άμεση διαχείρι-

3.3. Περιορισμοί Υπάρχοντος Συστήματος

ση συσκευών και επικοινωνία με άλλα συστατικά του συστήματος. Η επέκταση και βελτίωση αυτού του συστήματος αποτελεί το κεντρικό θέμα της παρούσας εργασίας.



Σχήμα 3.1: Υπάρχον Σύστημα

3.3 Περιορισμοί Υπάρχοντος Συστήματος

Ο ATMEGA328P είναι ένας από τους πιο δημοφιλείς μικροελεγκτές που χρησιμοποιούνται σε πολλά Do It Yourself (DIY) και εκπαιδευτικά έργα, όπως το Arduino Uno. Ωστόσο, παρόλη την απλότητα και την ευκολία χρήσης του, υπάρχουν κάποια περιορισμένα χαρακτηριστικά που μπορούν να αποτελέσουν αδυναμίες σε πιο προηγμένες εφαρμογές:

Έλλειψη Ενσωματωμένης Υποστήριξης Wi-Fi ή Bluetooth: Ο ATmega328P δεν περιλαμβάνει ενσωματωμένες λειτουργίες για Wi-Fi ή Bluetooth. Αυτό σημαίνει ότι για εφαρμογές που απαιτούν ασύρματη επικοινωνία, θα πρέπει να προστεθούν εξωτερικά modules, τα οποία αυξάνουν την πολυπλοκότητα του σχεδίου και το κόστος.

Περιορισμένη Μνήμη: Ο ATmega328P έχει μόνο 32KB μνήμη flash για την αποθήκευση του προγράμματος και 2KB SRAM. Αυτός ο περιορισμός μπορεί να είναι αρκετός για απλές εφαρμογές, αλλά μπορεί να αποτελέσει εμπόδιο για πιο προηγμένες εφαρμογές που απαιτούν μεγαλύτερη χωρητικότητα και περισσότερη μνήμη για δεδομένα.

Περιορισμένη Επεξεργαστική Ικανότητα: Ο ATmega328P διαθέτει μόνο έναν επεξεργαστή 8-bit με συχνότητα λειτουργίας που φτάνει τα 20MHz. Αυτό μπορεί να είναι αρκετό για απλές εργασίες, αλλά για πιο απαιτητικές εφαρμογές, ιδιαίτερα αυτές που απαιτούν υψηλότερες υπολογιστικές επιδόσεις, ο ATmega328P μπορεί να μην είναι ιδανικός.

Τέλος αν και ο ATmega328P είναι ένας καλός μικροελεγκτής για αρκετές εφαρμογές, οι παραπάνω αδυναμίες τον καθιστούν λιγότερο ιδανικό για πιο προηγμένες ή απαιτητικές εφαρμογές, όπως τα συστήματα IoT που απαιτούν ενσωματωμένη ασύρματη επικοινωνία, μεγάλη χωρητικότητα μνήμης, και ισχυρότερες υπολογιστικές επιδόσεις.

3.4 Αναβάθμιση Συστήματος

Μετά την αναλυτική εξέταση των περιορισμών του υπάρχοντος συστήματος, αναδείχθηκε η ανάγκη για σημαντική αναβάθμιση του. Το επίκεντρο της αναβάθμισης αυτής αφορούσε την αντικατάσταση του μικροελεγκτή ATmega328P με έναν ESP32. Το κίνητρο για αυτή την αλλαγή ήταν πολυεπίπεδο και θα το αναλύσουμε παρακάτω.

Ο ESP32 είναι ένας πιο προηγμένος μικροελεγκτής που προσφέρει σημαντικές βελτιώσεις σε σύγκριση με τον ATmega328P. Η κύρια διαφορά που τον καθιστά επιθυμητό για την αναβάθμιση του συστήματος είναι η ενσωματωμένη

υποστήριξη για Wi-Fi και Bluetooth. Η δυνατότητα αυτή ανοίγει ένα ευρύ φάσμα νέων εφαρμογών και λειτουργιών για το σύστημά μας, προσφέροντας τη δυνατότητα απομακρυσμένου ελέγχου και παρακολούθησης.

Με τη χρήση του ESP32, το σύστημα θα μπορεί να συνδεθεί στο Διαδίκτυο, επιτρέποντας την απομακρυσμένη πρόσβαση και διαχείριση. Αυτό δημιουργεί μια ευέλικτη και προσαρμόσιμη λύση που επιτρέπει την αυξημένη παρακολούθηση και τον ευέλικτο έλεγχο του συστήματος, ανεξάρτητα από την τοποθεσία του χρήστη.

Σε συνδυασμό με τις πιο προηγμένες επεξεργαστικές ικανότητες και την μεγαλύτερη μνήμη του ESP32, αυτές οι βελτιώσεις αποτελούν σημαντική αναβάθμιση στο υπάρχον σύστημα, ενισχύοντας την απόδοση, την λειτουργικότητα και την αξιοπιστία του.

Η επανασχεδίαση του συστήματος με τη χρήση του ESP32, ως εκ τούτου, παρέχει μια ριζική βελτίωση στην απόδοση και την ευελιξία του συστήματος, διευρύνοντας τις δυνατότητές του και ανοίγοντας νέες δυνατότητες εφαρμογών.

3.5 Απομακρυσμένος Έλεγχος και Παρακολούθηση

Η απόφαση για την αναβάθμιση του μικροελεγκτή σε ESP32 δεν ήταν απλώς μια αναβάθμιση εξοπλισμού, αλλά σηματοδότησε την έναρξη μιας νέας εποχής για το σύστημα. Η αναβάθμιση άνοιξε νέες δυνατότητες για την ανάπτυξη και εφαρμογή του λογισμικού, με την προσθήκη των λειτουργιών Wi-Fi και Bluetooth. Αυτό επέτρεψε στο σύστημα να ελέγχεται και να παρακολουθείται απομακρυσμένα, προσφέροντας μια νέα διάσταση ευελιξίας και δυνατότητας προσαρμογής.

Παράλληλα, έγινε ανάπτυξη ενός ευέλικτου και διαδραστικού web-γραφικού περιβάλλοντος ιστού, καθώς και ενός back-end συστήματος με χρήση Flask Python. Η διεπαφή αυτή επέτρεψε την άμεση αλληλεπίδραση των χρηστών με το σύστημα, επιτρέποντας την παρακολούθηση των αισθητήρων αλλά και τον έλεγχο των βηματικών κινητήρων μέσω του web γραφικού περιβάλλοντος.

Η ενσωμάτωση με μια βάση δεδομένων παρείχε έναν αποθηκευτικό χώρο

για την καταγραφή και την ανάλυση των δεδομένων από το σύστημα, δίνοντας τη δυνατότητα για αναλυτική εξέταση και βελτίωση των λειτουργιών του.

Έπειτα από αυτήν την σημαντική αναβάθμιση, το σύστημα έχει αυξήσει την αποδοτικότητά του, προσφέροντας μια ολοκληρωμένη, ευέλικτη και επεκτάσιμη πλατφόρμα για την καλύτερη χρήση του ESP32 και των ευκαιριών που παρέχει η ψηφιακή εποχή.

3.6 Εγκατάσταση και επικοινωνία

Κάθε στοιχείο του συστήματος, εκτός από τον ESP32, τρέχει μέσα σε Docker container, προσφέροντας έτσι αυξημένη ευελιξία και επεκτασιμότητα. Τα Docker containers διευκολύνουν την εγκατάσταση και τη λειτουργία του συστήματος σε διαφορετικές πλατφόρμες και λειτουργικά συστήματα, εξασφαλίζοντας τη συμβατότητα και την απρόσκοπτη λειτουργία του DRMS.

Κεφάλαιο 4

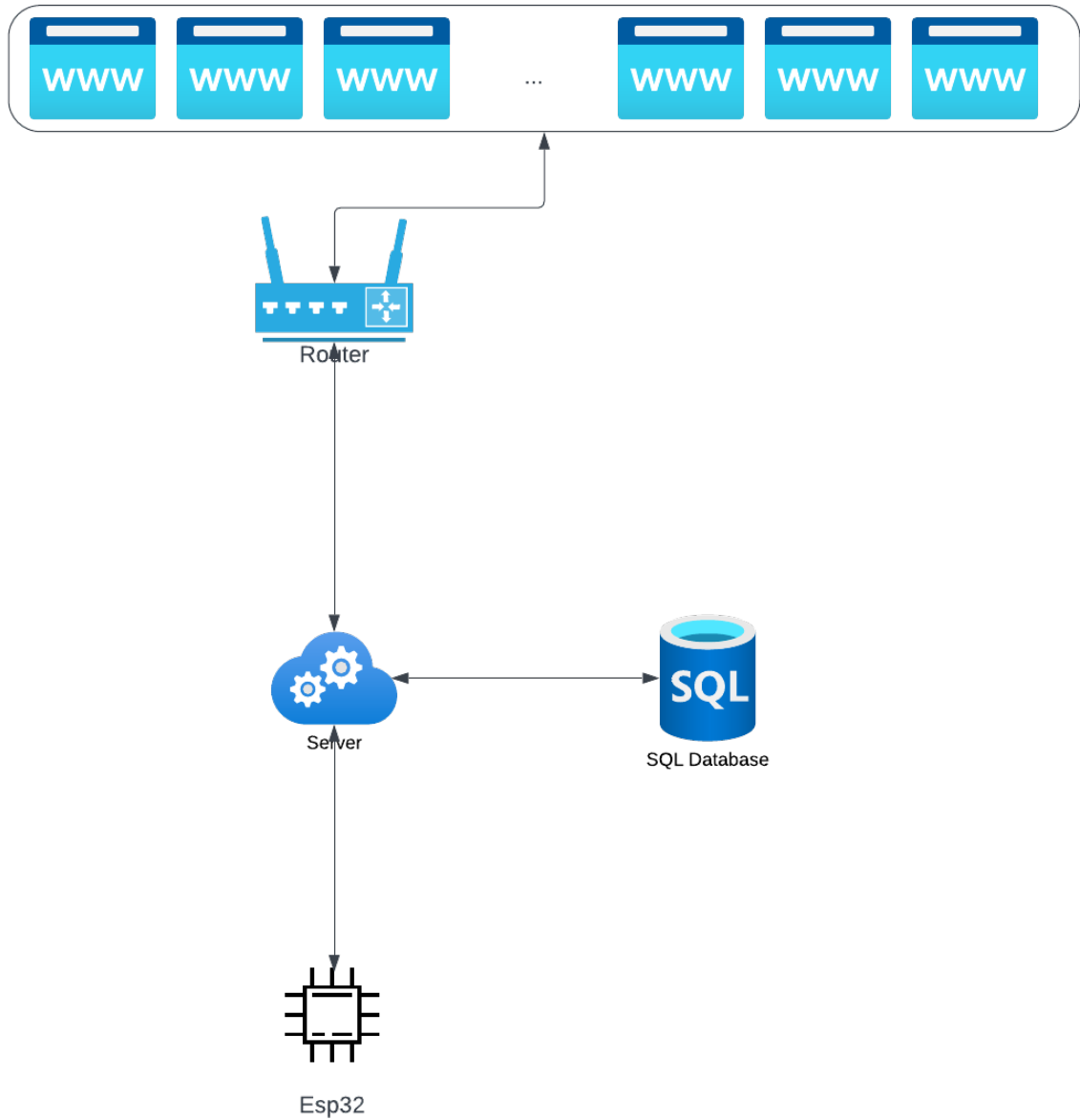
Υλοποίηση

Στο κεφάλαιο αυτό της διπλωματικής εργασίας, θα παρουσιάσουμε την υλοποίηση του κατανεμημένου συστήματος για τη συλλογή, αποστολή και επεξεργασία δεδομένων από αισθητήρες, αλλά καθώς και τον έλεγχο των δυο βηματικών κινητήρων με τη χρήση ενός μικροελεγκτή ESP32, ενός διακομιστή Flask, μιας βάσης δεδομένων και ενός web γραφικού περιβάλλοντος. Το σύστημα αυτό επιτρέπει στους χρήστες να παρακολουθούν τα δεδομένα των αισθητήρων σε πραγματικό χρόνο, να ελέγχουν τον μικροελεγκτή και να αποθηκεύουν τις μετρήσεις σε αρχεία CSV για περαιτέρω ανάλυση.

Η επικοινωνία μεταξύ των διαφόρων στοιχείων του συστήματος βασίζεται στην αρχιτεκτονική REST-API, που επιτρέπει την αποτελεσματική ανταλλαγή πληροφοριών μεταξύ των διαφορετικών συνιστωσών. Στη συνέχεια του κεφαλαίου, θα εξετάσουμε αναλυτικά τις τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν, την αρχιτεκτονική του συστήματος και τις κύριες λειτουργίες που παρέχει.

Τέλος, στόχος μας είναι να παράσχουμε μια κατανοητή και συνοπτική παρουσίαση της υλοποίησης, ώστε οι αναγνώστες να κατανοήσουν τις βασικές αρχές λειτουργίας του κατανεμημένου συστήματος, τις τεχνολογίες που χρησιμοποιήθηκαν και τις λύσεις που εφαρμόσαμε για την επίτευξη των στόχων της εργασίας

Στο διάγραμμα 4.1 παρουσιάζεται η διάταξη αλλά και το πως έχουν χωριστεί τα διάφορα components του συστήματος.



Σχήμα 4.1: Διάγραμμα του Κατανεμημένου DRMS Συστήματος

4.1 Ανάλυση στοιχείων του συστήματος

Στην παρούσα ενότητα, θα αναλύσουμε το διάγραμμα 4.1 του κατανεμημένου συστήματος μας, το οποίο αποτελείται από τέσσερα κύρια συστατικά:

- ESP32
- Βάση Δεδομένων
- Python Διακομιστής
- Web-Γραφικό Περιβάλλον

4.2 Esp32

Το DRMS σύστημα χρησιμοποιούσε μέχρι πρότινος τον Atmel ATMEGA328P microcontroller για τη διαχείριση δεδομένων από αισθητήρες και τον έλεγχο κινητήρων βηματικής κίνησης. Ωστόσο, για την επίτευξη πιο αξιόπιστης και αποδοτικής επικοινωνίας με τα υπόλοιπα στοιχεία του συστήματος, αποφασίσαμε να επιλέξουμε τον ESP32 microcontroller.

Συνολικά, η αναβάθμιση του συστήματος DRMS στο ESP32 θα επιτρέψει την ανάπτυξη ενός πιο ισχυρού, ευέλικτου και αξιόπιστου συστήματος, που θα επιτρέπει στους χρήστες να συλλέγουν και να αναλύουν δεδομένα από πολλές πηγές, και να επικοινωνούν ασύρματα με άλλες συσκευές και το διαδίκτυο. Με τον ESP32, οι δυνατότητες του συστήματος DRMS επεκτείνονται και βελτιώνονται, καθιστώντας το έτοιμο για τις απαιτητικές αναλύσεις δεδομένων.

4.2.1 Υλοποίηση και Προγραμματισμός Esp32

Κατά την υλοποίηση του ελεγκτή ESP32, αναπτύχθηκε ένας WiFi Manager, με τη χρήση της γλώσσας προγραμματισμού C/C++, ο οποίος επιτρέπει στον

χρήστη να συνδέεται εύκολα σε ένα δίκτυο WiFi της επιλογής του. Αυτή η λειτουργικότητα διασφαλίζει την ευελιξία και την προσαρμοστικότητα του συστήματος σε διαφορετικά περιβάλλοντα και συνθήκες λειτουργίας.

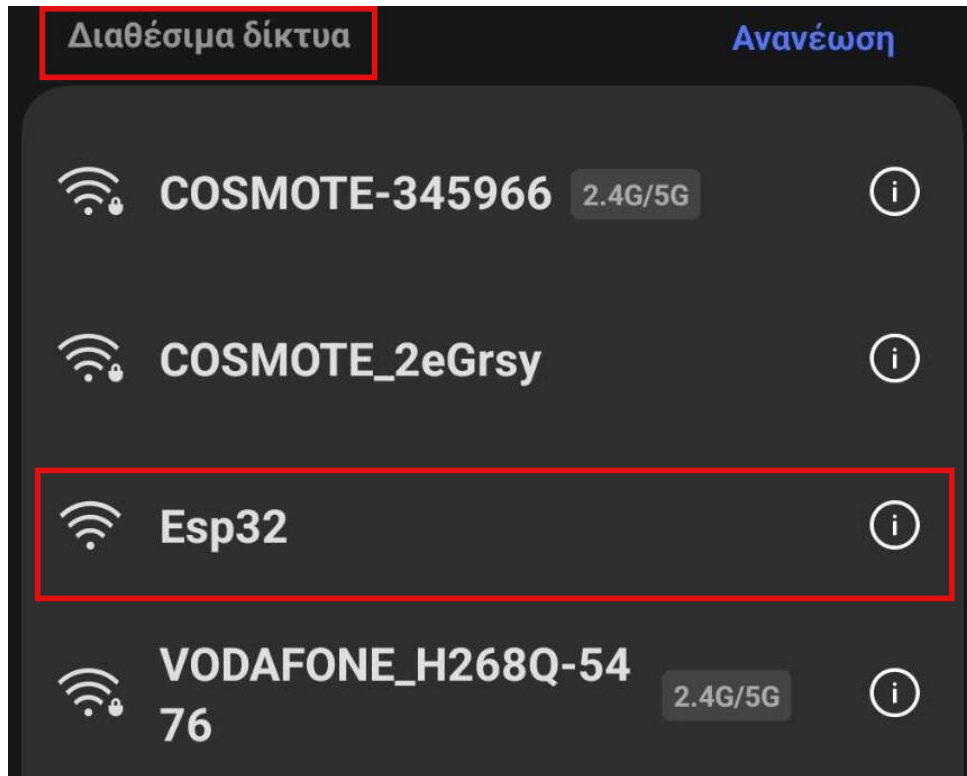
Μετά τη σύνδεση στο WiFi, ο ελεγκτής ESP32 αποκτά τη δυνατότητα να αποστέλλει και να λαμβάνει δεδομένα μέσω JSON, χρησιμοποιώντας τις μεθόδους GET και POST. Συγκεκριμένα, ο ελεγκτής αποστέλλει τις τιμές από τους αισθητήρες στο endpoint data, εξασφαλίζοντας έτσι την άμεση και ακριβή καταγραφή των παραμέτρων που παράγονται από το σύστημα.

Επιπλέον, ο ελεγκτής ESP32 επιτρέπει την απομακρυσμένη παρακολούθηση και ρύθμιση των βηματικών κινητήρων μέσω της μεθόδου getMotorValue. Με αυτόν τον τρόπο, ο χρήστης μπορεί πλέον να ελέγχει και να τροποποιεί τις τιμές των βηματικών κινητήρων απομακρυσμένα, επιτυγχάνοντας έτσι την ομαλή, ακριβή και ελεγχόμενη κίνηση που επιθυμεί. Αυτή η λειτουργικότητα βελτιστοποιεί την απόδοση του συστήματος, ενώ παράλληλα αυξάνει την ευκολία χρήσης για τον τελικό χρήστη, καθώς προσφέρει μεγαλύτερο έλεγχο και ευελιξία στη διαχείριση των κινητήρων.

Συνοψίζοντας, η παρούσα υλοποίηση του ελεγκτή ESP32 αποτελεί έναν καίριο και κεντρικό παράγοντα του κατανεμημένου συστήματος, παρέχοντας την απαραίτητη επικοινωνία, έλεγχο και διαχείριση των διαφόρων συστατικών που συνθέτουν την εφαρμογή. Η λεπτομερής ανάλυση και η προσεγμένη υλοποίηση εξασφαλίζουν την αξιοπιστία, την απόδοση και την αποτελεσματικότητα του συστήματος, δημιουργώντας μια ισχυρή βάση για την περαιτέρω ανάπτυξη και βελτίωση της εφαρμογής.

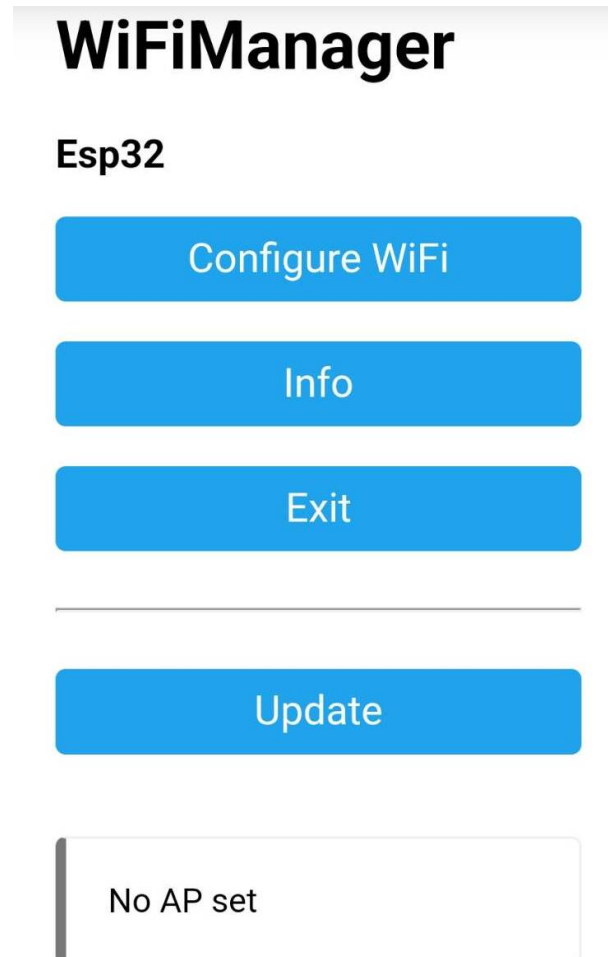
Η διαδικασία σύνδεσης σε ένα δίκτυο παρατίθεται παρακάτω:

Αρχικά δημιουργείται ένα δίκτυο με όνομα ESP32 στο οποίο ο χρήστης μπορεί να κάνει σύνδεση (Σχήμα 4.2).



Σχήμα 4.2: ESP32 Δίκτυο

Μετά την επιτυχημένη σύνδεση του χρήστη, αυτός καθοδηγείται αυτόματα προς την επόμενη οθόνη. Κατ' αυτόν τον τρόπο, η διεπαφή παρέχει μια ομαλή και εύχρηστη εμπειρία στον χρήστη.



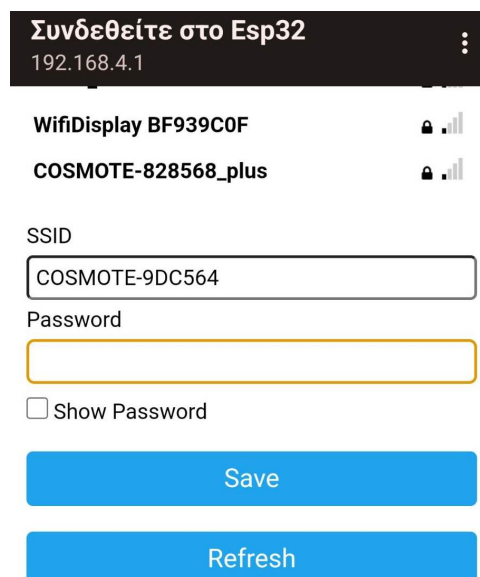
Σχήμα 4.3: WIFI Μενού

Όπως φαίνεται στο Σχήμα 4.3, ο χρήστης έχει πολλές επιλογές, όπως να συνδεθεί σε ένα δίκτυο, να δει γενικές πληροφορίες ή να αποχωρήσει. Για να συνδεθεί στο δίκτυο της επιλογής του, απαιτείται να επιλέξει το “Configure WiFi”, το οποίο θα τον οδηγήσει σε μια οθόνη με τα διαθέσιμα ασύρματα δίκτυα που υπάρχουν όπως απεικονίζεται στο Σχήμα 4.4:



Σχήμα 4.4: Διαθέσιμα Δίκτυα

Στην συνέχεια επιλέγει το δίκτυο της αρεσκείας του για να κάνει σύνδεση (Σχήμα 4.5).



Σχήμα 4.5: Καταχώρηση Στοιχείων Δικτύου

Μετά την εισαγωγή του ονόματος του ασύρματου δικτύου και του α-

4.3. Database

ντίστοιχου κωδικού πρόσβασης, ο μικροελεγκτής ESP32 προσπαθεί να συνδεθεί στο συγκεκριμένο δίκτυο. Εάν η σύνδεση είναι επιτυχής, τότε εμφανίζεται ένα μήνυμα επιβεβαίωσης στο Serial Monitor (Σχήμα 4.6).

```
*wm:Connecting to NEW AP: COSMOTE-9DC564
*wm:connectTimeout not set, ESP waitForConnectResult...
*wm:Connect to new AP [SUCCESS]
*wm:Got IP Address:
*wm:192.168.1.16
[ 75594] [E] [WebServer.cpp:649] _handleRequest(): request handler not found
*wm:config portal exiting
connected...yeey :)
```

Σχήμα 4.6: Επιτυχής Σύνδεση

Μόλις ο μικροελεγκτής ESP32 συνδεθεί στο διαδίκτυο, είναι έτοιμος να ξεκινήσει την αμφίδρομη επικοινωνία με τα υπόλοιπα στοιχεία του κατανεμημένου συστήματός μας, αποστέλλοντας και λαμβάνοντας δεδομένα μέσω του HTTP πρωτοκόλλου που χρησιμοποιεί η εφαρμογή μας.

Τέλος ο ESP32 χρησιμοποιεί το πρωτόκολλο REST-API για να λαμβάνει και να αποστέλλει δεδομένα σε μορφή JSON με τα υπόλοιπα στοιχεία του κατανεμημένου συστήματος.

4.3 Database

Η ανάγκη για διαχείριση διαφορετικών χρηστών και αποθήκευση δεδομένων μας οδήγησε στη λύση μιας σχεσιακής βάσης δεδομένων SQL. Αυτή η επιλογή είναι εξαιρετικά χρήσιμη γιατί μας επιτρέπει να αποθηκεύουμε δεδομένα με δομημένο τρόπο, επιτρέποντάς μας να τα ανακτούμε και να τα επεξεργαζόμαστε εύκολα. Επιπλέον, η χρήση μιας σχεσιακής βάσης δεδομένων SQL μας παρέχει ευέλικτες επιλογές για την αναζήτηση, την ταξινόμηση και την ανάλυση των δεδομένων μας, κάνοντας τη διαχείρισή τους γρήγορη και αποτελεσματική.

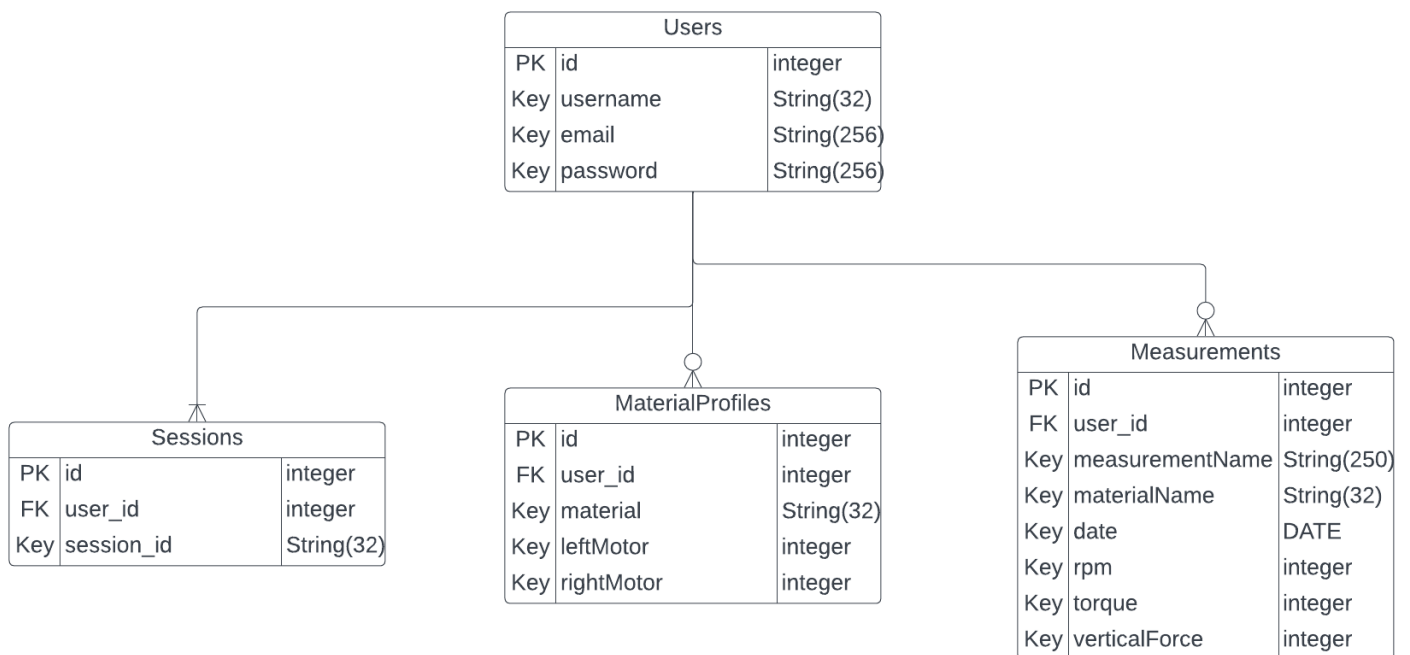
4.3.1 Υλοποίηση Βάσης Δεδομένων

Ένα διάγραμμα βάσης δεδομένων αποτελεί ένα σημαντικό εργαλείο στη διαχείριση μιας σύνθετης εφαρμογής. Απεικονίζει τη δομή της βάσης δεδομένων, τους πίνακες που περιλαμβάνει και τις σχέσεις μεταξύ τους. Παρέχει μια οπτική

4.3. Database

αναπαράσταση των δεδομένων και των σχέσεων τους, και είναι απαραίτητο για την καλή σχεδίαση και υλοποίηση της βάσης δεδομένων.

Το διάγραμμα 4.7 αναπαριστά την σχεσιακή βάση δεδομένων που αναπτύχθηκε για την εφαρμογή DRMS. Στο διάγραμμα αυτό φαίνονται οι πίνακες και οι σχέσεις μεταξύ τους, καθώς και οι περιορισμοί στα δεδομένα που μπορούν να αποθηκευτούν σε κάθε πίνακα. Περαιτέρω εξηγήσεις για το διάγραμμα και τη λειτουργία των πινάκων και των σχέσεων τους, θα παρουσιαστούν παρακάτω.



Σχήμα 4.7: Σχήμα Βάσης Δεδομένων

Η βάση δεδομένων του DRMS συστήματος περιέχει 4 πίνακες.

1. *Users*
2. *Sessions*
3. *Material Profiles*
4. *Measurements*

Users: Ο πίνακας Users φέρει την ευθύνη της αποθήκευσης και διαχείρισης των χρηστών του συστήματος DRMS. Συγκεκριμένα, ο πίνακας περιλαμβάνει ένα πρωτεύον κλειδί id, το οποίο αυξάνεται κατά ένα μετά την επιτυχή εγγραφή ενός νέου χρήστη. Επιπρόσθετα, ο πίνακας περιέχει το όνομα χρήστη, τη διεύθυνση email και τον κρυπτογραφημένο κωδικό πρόσβασης του χρήστη, εξασφαλίζοντας έτσι την προστασία του λογαριασμού.

Sessions: Ο πίνακας Sessions είναι υπεύθυνος για την αποθήκευση και διαχείριση των συνδέσεων των χρηστών στο σύστημα DRMS. Περιλαμβάνει ένα πρωτεύον κλειδί id, το οποίο αυξάνεται αυτόματα κατά την είσοδο του χρήστη. Επιπλέον, ο πίνακας περιέχει ένα δευτερεύον κλειδί από τον πίνακα Users και ένα μοναδικό κλειδί sessionId. Όπως γίνεται αντιληπτό, κάθε χρήστης μπορεί να έχει από “zero to many” συνδέσεις. Η τιμή μηδέν αντιπροσωπεύει την περίπτωση κατά την οποία ένας χρήστης δεν έχει καμία ενεργή σύνδεση στο DRMS, ενώ το N υποδηλώνει την περίπτωση κατά την οποία ένας χρήστης έχει μία ή περισσότερες συνδέσεις από διαφορετικούς περιηγητές ή συσκευές.

Material Profiles: Ο πίνακας Material Profiles καταγράφει τις τιμές των βηματικών κινητήρων για συγκεκριμένα υλικά. Πιο συγκεκριμένα, περιλαμβάνει ένα πρωτεύον κλειδί που αυξάνεται κατά ένα αυτόματα μετά από κάθε εισαγωγή, ένα δευτερεύον κλειδί που αναφέρεται στο id του χρήστη, το όνομα του υλικού, καθώς και τις τιμές των δύο βηματικών κινητήρων. Η σχέση μεταξύ του πίνακα Users και του πίνακα Material Profiles είναι “zero to many“. Δηλαδή, κάθε χρήστης μπορεί να έχει από 0 υλικά προφίλ (material profiles) έως N.

Measurements: Ο πίνακας Μετρήσεις (Measurements) είναι υπεύθυνος για την αποθήκευση και διαχείριση των μετρήσεων που πραγματοποιεί το σύστημα DRMS. Αυτός ο πίνακας περιλαμβάνει ένα πρωτεύον κλειδί id, το

οποίο αυξάνεται αυτόματα, ένα δευτερεύον κλειδί που αναφέρεται στο id του χρήστη, το μοναδικό όνομα της μέτρησης, το όνομα του υλικού που μετράται, την ημερομηνία καταχώρησης στον πίνακα, καθώς και τις τιμές των αισθητήρων στη συγκεκριμένη χρονική στιγμή. Τέλος, η συσχέτιση μεταξύ του πίνακα Χρήστες (Users) και του πίνακα Μετρήσεις (Measurements) είναι “zero to many“, δηλαδή ένας χρήστης μπορεί να έχει από 0 μετρήσεις έως N.

4.4 Flask Server

Σε αυτή την ενότητα, εμβαθύνουμε στην υλοποίηση του διακομιστή μας Flask, ένα κρίσιμο στοιχείο του έργου μας που θα παίζει θεμελιώδη ρόλο στη διευκόλυνση της επικοινωνίας και της μεταφοράς δεδομένων μεταξύ της εφαρμογής μας, της βάσης δεδομένων και του web-client. Πριν μπούμε στις λεπτομέρειες της υλοποίησης, είναι σημαντικό να κατανοήσουμε ότι ο διακομιστής μας δεν θα διαχειρίζεται μόνο τα εισερχόμενα αιτήματα και απαντήσεις, αλλά επίσης θα διαχειρίζεται βασικές εργασίες όπως η δρομολόγηση, η διαχείριση σφαλμάτων και η παροχή στατικών αρχείων. Στόχος μας είναι να δημιουργήσουμε μια ισχυρή και αποδοτική υποδομή διακομιστή που μπορεί να χειριστεί μεγάλους όγκους δεδομένων και να παρέχει μια αρμονική εμπειρία χρήστη.

4.4.1 Υλοποίηση Flask Server

Στο παρούσα ενότητα, παρουσιάζεται το διάγραμμα αρχιτεκτονικής του Flask server (Σχήμα 4.8), που αποτελεί τον πυρήνα της εφαρμογής που αναπτύχθηκε στο πλαίσιο της παρούσας διπλωματικής εργασίας. Σκοπός αυτού του διαγράμματος είναι να παρέχει μια σαφή και συνοπτική εικόνα των βασικών συνιστωσών της εφαρμογής, των εξαρτήσεων ανάμεσά τους, καθώς και της ροής των δεδομένων και των αιτήσεων μέσα στο σύστημα.

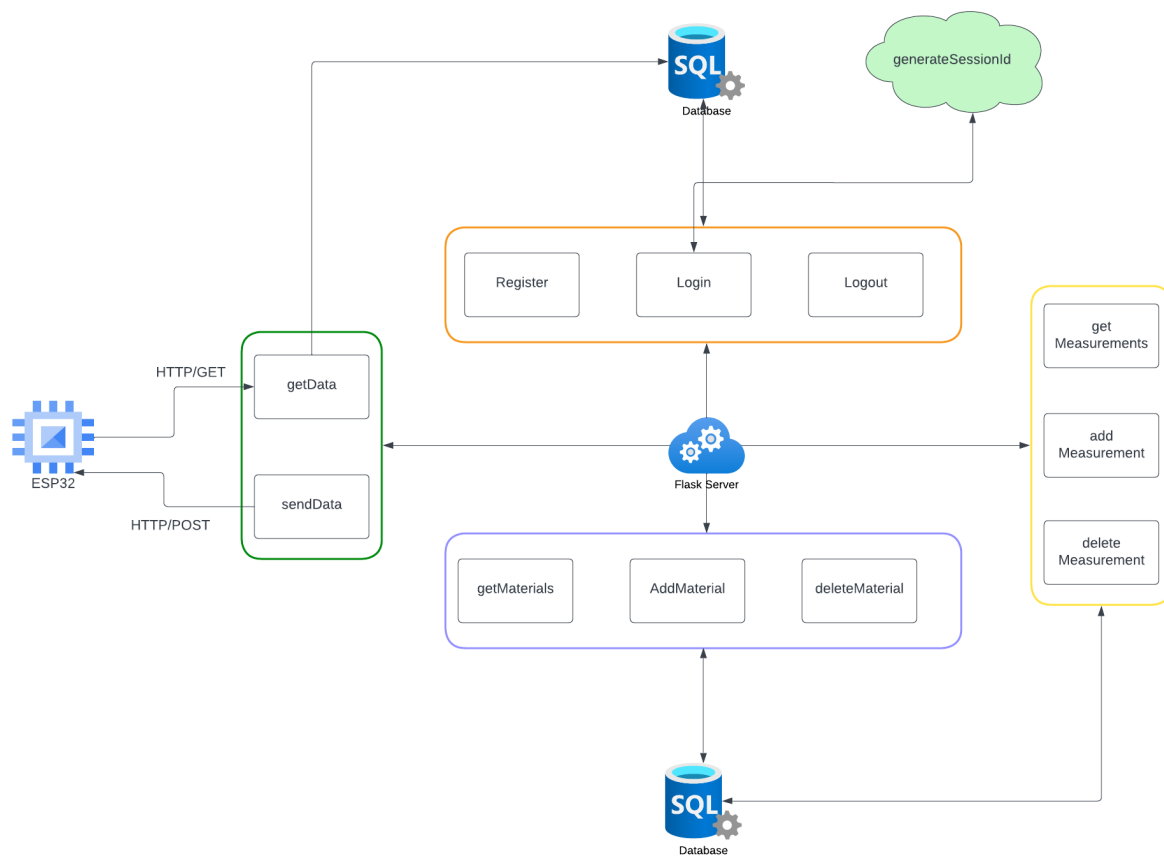
Το διάγραμμα αρχιτεκτονικής δίνει έμφαση στην εξερεύνηση και την κα-

τανόηση της εσωτερικής δομής της εφαρμογής, όπως τα routes, τα controllers, τα μοντέλα, τα views και τα templates, καθώς και στην επικοινωνία της εφαρμογής με τις εξωτερικές υπηρεσίες, όπως βάσεις δεδομένων.

Αυτό το διάγραμμα έχει ως στόχο να διευκολύνει την ανάλυση και τη συζήτηση γύρω από την αρχιτεκτονική της εφαρμογής, καθώς και να βοηθήσει την ομάδα ανάπτυξης να εντοπίσει περιθώρια βελτίωσης και να αντιμετωπίσει τυχόν προκλήσεις που προκύπτουν από την εφαρμογή του Flask framework. Επιπλέον, το διάγραμμα αρχιτεκτονικής συμβάλλει στην τεκμηρίωση του έργου, παρέχοντας μια βάση γνώσης για τη συνεχή ανάπτυξη και συντήρηση της εφαρμογής.

Η παρουσίαση του διαγράμματος αρχιτεκτονικής στη διπλωματική εργασία επιτρέπει στον αναγνώστη να αποκτήσει μια καλύτερη κατανόηση της λειτουργίας και της δομής του Flask διακομιστή, καθιστώντας την εξέταση των λεπτομερειών της υλοποίησης πιο προσβάσιμη και ευκολότερη.

Καθώς το έργο εξελίσσεται, το διάγραμμα αρχιτεκτονικής μπορεί να ενημερωθεί για να αντικατοπτρίζει τυχόν αλλαγές στη δομή και τις σχέσεις μεταξύ των συνιστωσών του συστήματος. Επομένως, το διάγραμμα αρχιτεκτονικής αποτελεί ένα ζωντανό έγγραφο που προσαρμόζεται στις ανάγκες του έργου και στηρίζει τη διαδικασία λήψης αποφάσεων της ομάδας ανάπτυξης.

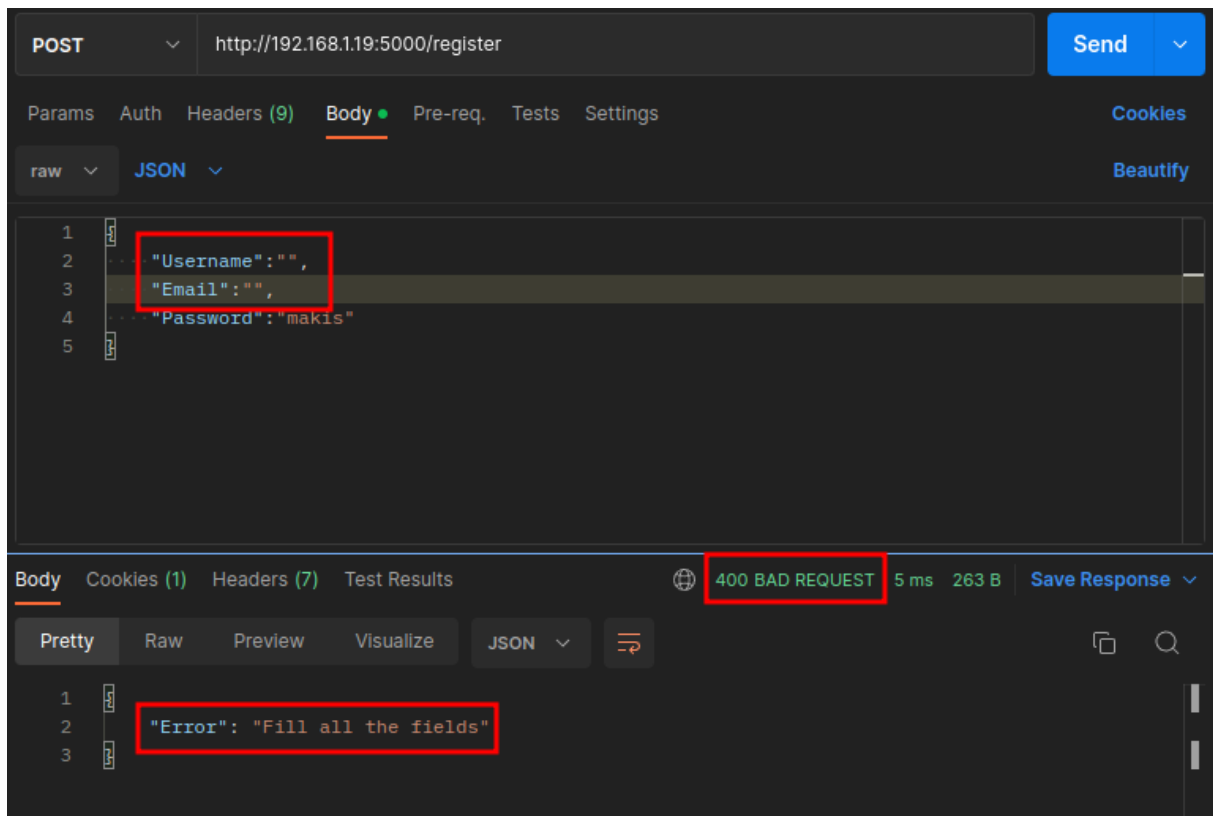


Σχήμα 4.8: Διακομιστής Flask

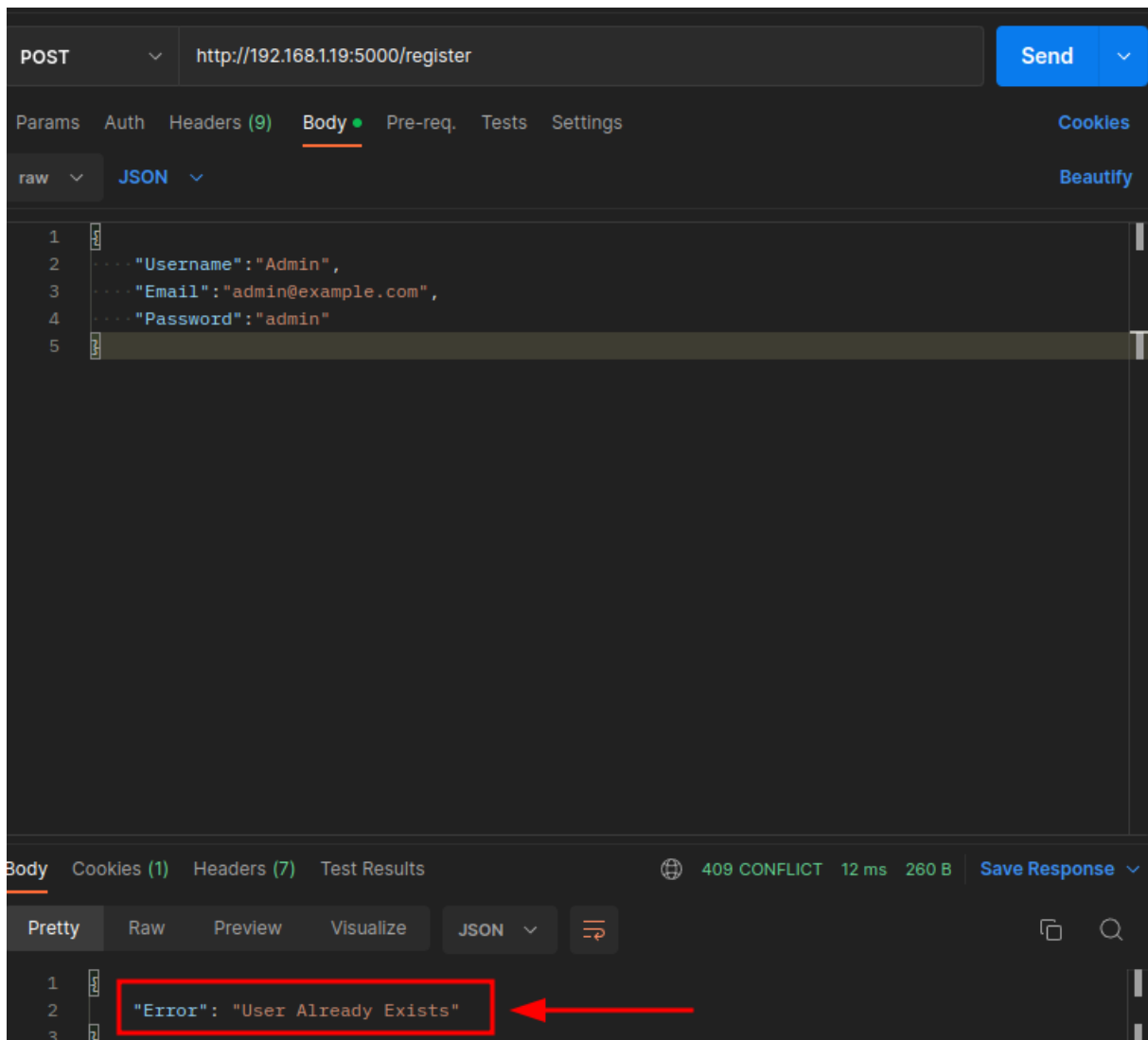
4.4. Flask Server

Register: Το Register αποτελεί μια διαδικασία του flask server το οποίο διαχειρίζεται τη διαδικασία εγγραφής νέων χρηστών. Όταν λαμβάνεται ένα POST αίτημα από το web-client σε μορφή JSON, περιέχοντας το όνομα χρήστη (username), το ηλεκτρονικό ταχυδρομείο (email) και τον κωδικό πρόσβασης (password), ο server επεξεργάζεται τα δεδομένα.

Αρχικά, ελέγχει αν όλα τα πεδία έχουν τιμές. Σε περίπτωση που κάποιο από αυτά είναι κενό, επιστρέφει ένα αποτέλεσμα (response) 400 όπως φαίνεται στο Σχήμα 4.9. Στη συνέχεια, εκτελείται ένα ερώτημα στη βάση δεδομένων για να διαπιστωθεί αν το όνομα χρήστη υπάρχει ήδη. Εάν υπάρχει, επιστρέφεται ένα αποτέλεσμα 409 με το μήνυμα “username already exists” όπως φαίνεται στο Σχήμα 4.10.



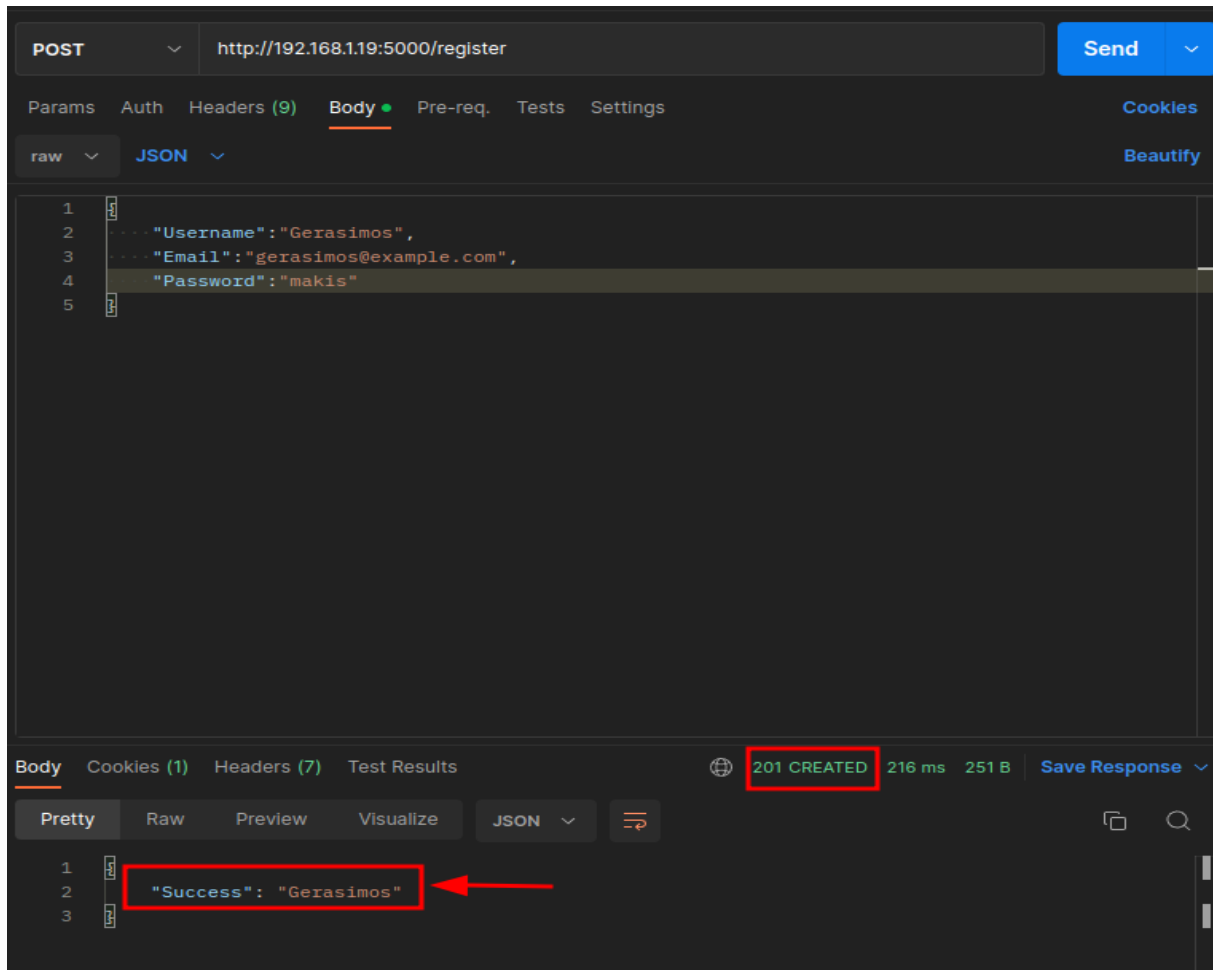
Σχήμα 4.9: Αποτυχημένη Εγγραφή Χρήστη: Ελλιπή Στοιχεία



Σχήμα 4.10: Αποτυχημένη Εγγραφή Χρήστη: Διπλοκαταχώρηση

Εάν το όνομα χρήστη δεν υπάρχει, ο κωδικός πρόσβασης κωδικοποιείται για λόγους ασφαλείας. Μετά από αυτό, ο νέος χρήστης καταχωρείται στη βάση δεδομένων και επιστρέφεται ένα αποτέλεσμα 201, συνοδευόμενο από ένα μήνυμα που περιέχει το όνομα χρήστη του νέου λογαριασμού. Ένα παράδειγμα εισαγωγής απεικονίζεται στο Σχήμα 4.11 ενώ στο Σχήμα 4.12 φαίνεται η εισαγωγή στην βάση δεδομένων.

4.4. Flask Server



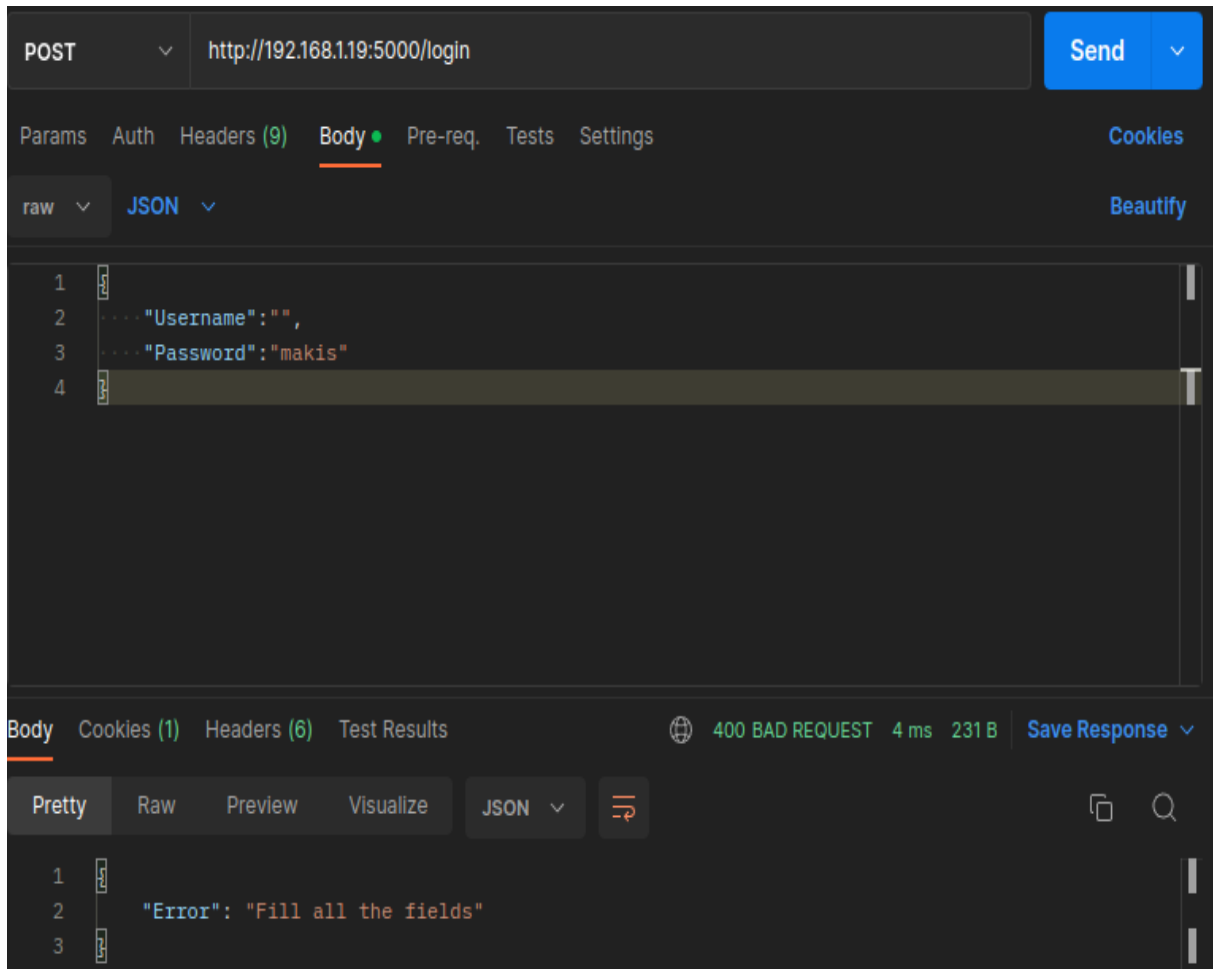
Σχήμα 4.11: Επιτυχής Εγγραφή Χρήστη

id	Username	email	Password
1	makis	gerasimosstefatos@gmail.c...	60 Bytes
2	Admin	admin@example.com	60 Bytes
3	Gerasimos	gerasimos@example.com	60 Bytes

Σχήμα 4.12: Επιτυχής Εγγραφή Χρήστη στην Βάση Δεδομένων

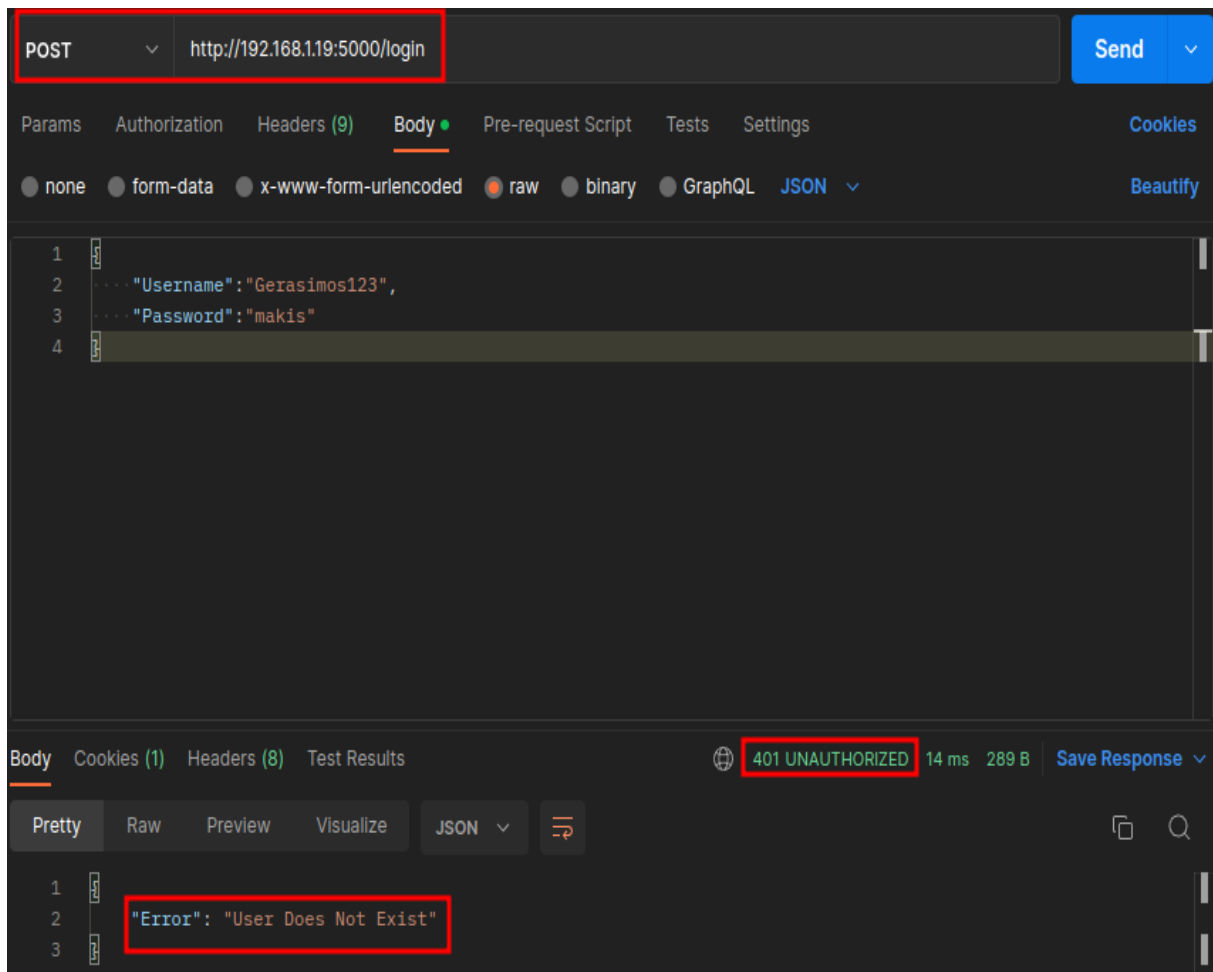
4.4. Flask Server

Login: Η μέθοδος του Flask διακομιστή Login αναλαμβάνει τη διαδικασία εισόδου των χρηστών. Αρχικά, λαμβάνει το όνομα χρήστη και τον κωδικό πρόσβασης μέσω ενός POST αιτήματος από το web-client. Στη συνέχεια, ελέγχει εάν κάποιο από τα πεδία είναι κενό. Σε αυτή την περίπτωση, στέλνει ένα response με κωδικό 400 και το μήνυμα “Fill all the fields“ προς το web-client (Σχήμα 4.13).



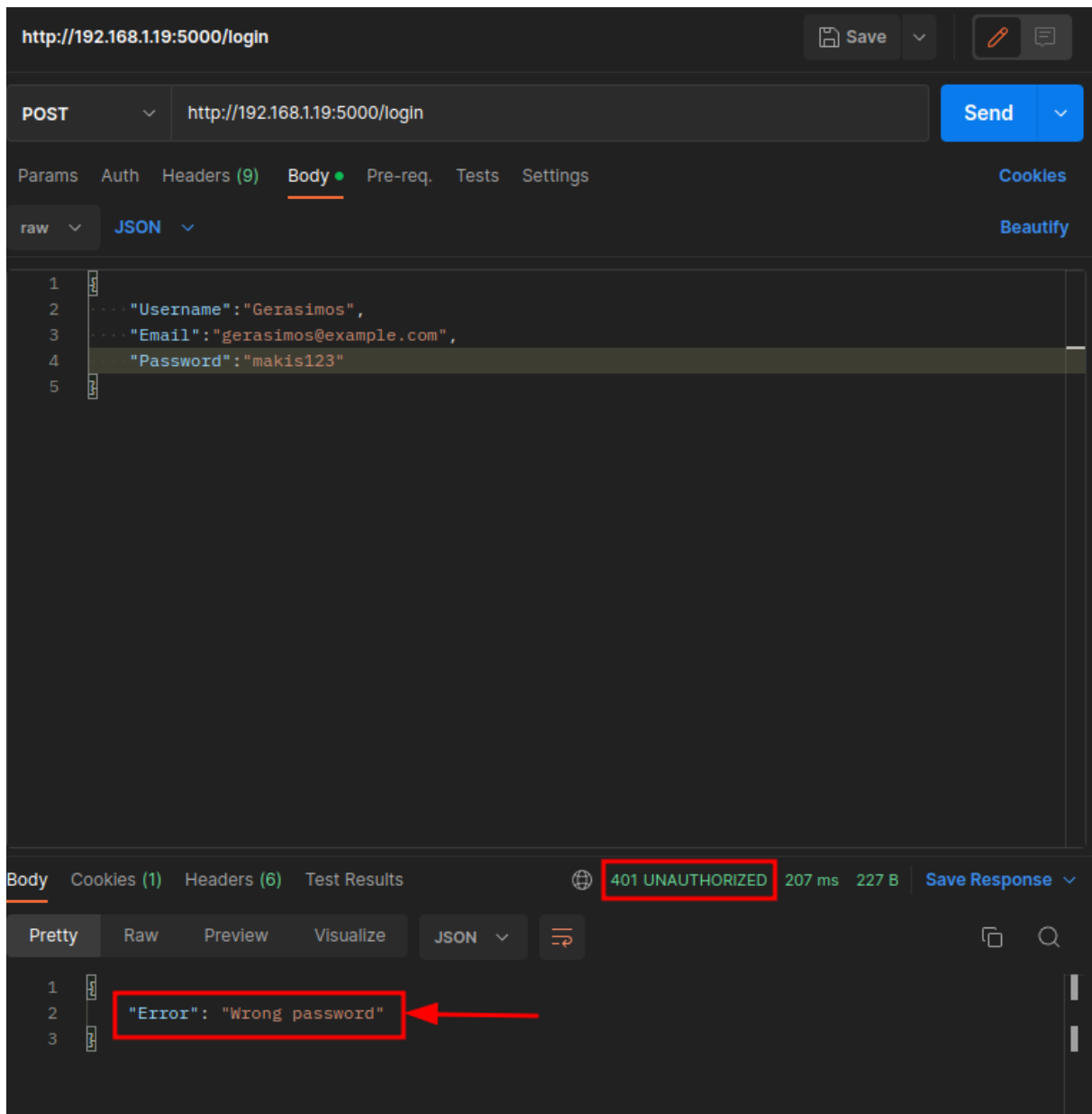
Σχήμα 4.13: Αποτυχημένο Αίτημα Σύνδεσης: Ελλιπή Στοιχεία

Βασισμένο στο όνομα χρήστη, το σύστημα εκτελεί ένα ερώτημα στη βάση δεδομένων για να βρει τον χρήστη. Εάν ο χρήστης δεν υπάρχει, στέλνει ένα response με κωδικό 401 και το μήνυμα “User Does not exist“ (Σχήμα 4.14) προς το web-client. Εάν ο χρήστης υπάρχει, το σύστημα ελέγχει αν ο κωδικός πρόσβασης είναι σωστός, αποκωδικοποιώντας τον αποθηκευμένο κωδικό και συγκρίνοντάς τον με αυτόν που πληκτρολόγησε ο χρήστης.



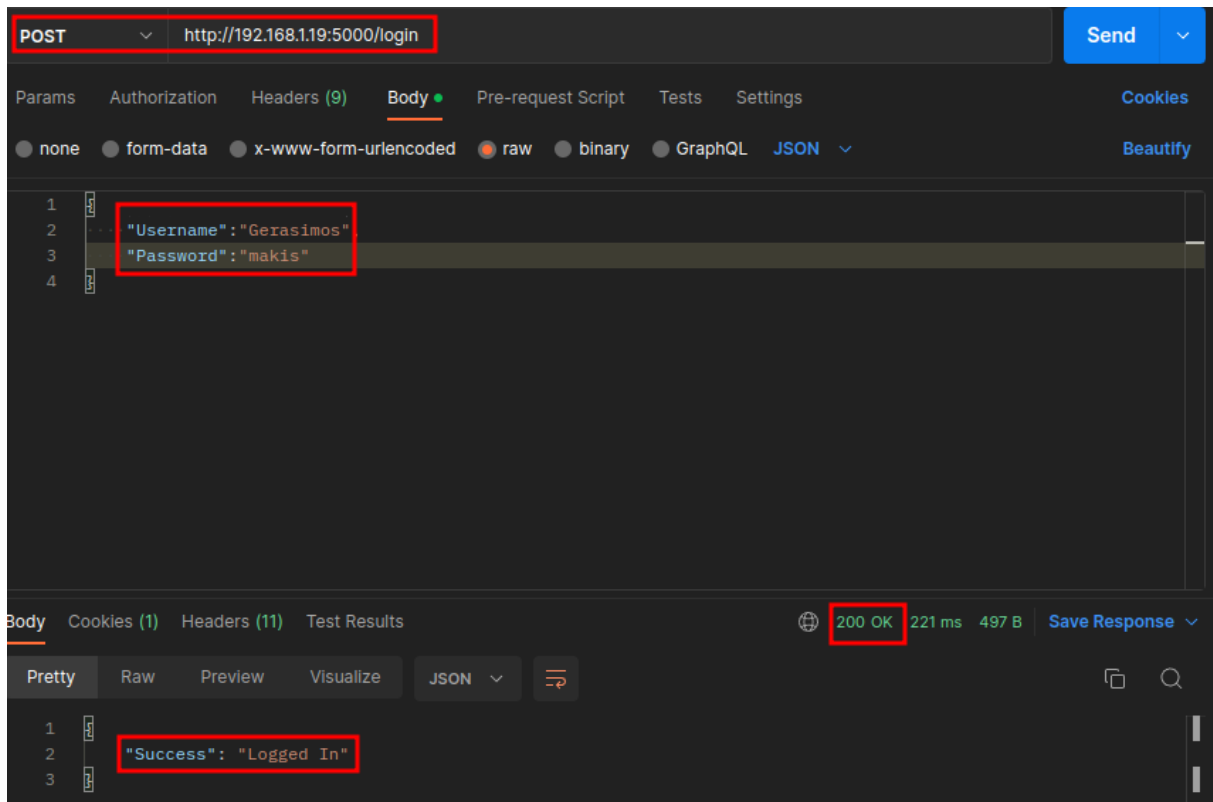
Σχήμα 4.14: Αποτυχημένη Σύνδεση Χρήστη: Ο Χρήστης δεν Υπάρχει

Εάν ο κωδικός που έστειλε ο χρήστης είναι λάθος, το σύστημα στέλνει ένα response με κωδικό 401 και το μήνυμα “wrong password” (Σχήμα 4.15) προς το web-client. Αντίθετα, αν ο κωδικός είναι σωστός, δημιουργείται ένα μοναδικό sessionId, το οποίο καταχωρείται στη βάση δεδομένων (Σχήμα 4.17) και χρησιμοποιείται στη συνέχεια για τον έλεγχο της εξουσιοδότησης του χρήστη. Εφόσον το sessionId δημιουργηθεί και καταχωρηθεί στη βάση δεδομένων, ο Flask server αποστέλλει ένα response με κωδικό 200, το μήνυμα “Logged In” καθώς και το sessionId ως cookie προς το web-client (Σχήμα 4.16).



Σχήμα 4.15: Λάθος Κωδικός Πρόσβασης

4.4. Flask Server



Σχήμα 4.16: Επιτυχής Σύνδεση Χρήστη

The screenshot shows a database client interface with a table named `sessions`. The table has columns `id`, `user_id`, and `session_id`. A single record is displayed with `id` 1, `user_id` 1, and `session_id` 829b9e25-f85f-4071...

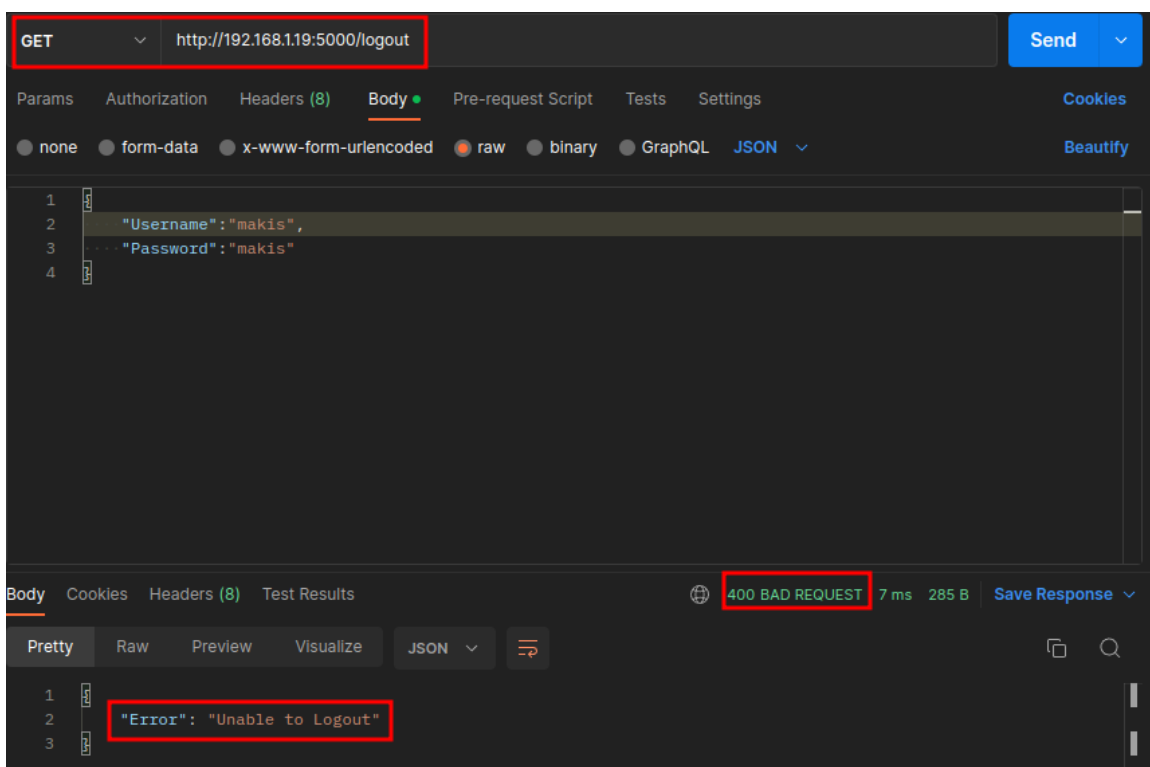
id	user_id	session_id
1	1	829b9e25-f85f-4071...

Σχήμα 4.17: Καταχώρηση Μοναδικού Αναγνωριστικού στην Βάση Δεσομένων

4.4. Flask Server

Με αυτό τον τρόπο, η μέθοδος Login εξασφαλίζει την απρόσκοπτη είσοδο των χρηστών στο σύστημα, ελέγχοντας τα απαραίτητα στοιχεία και παρέχοντας την κατάλληλη απάντηση σε κάθε περίπτωση.

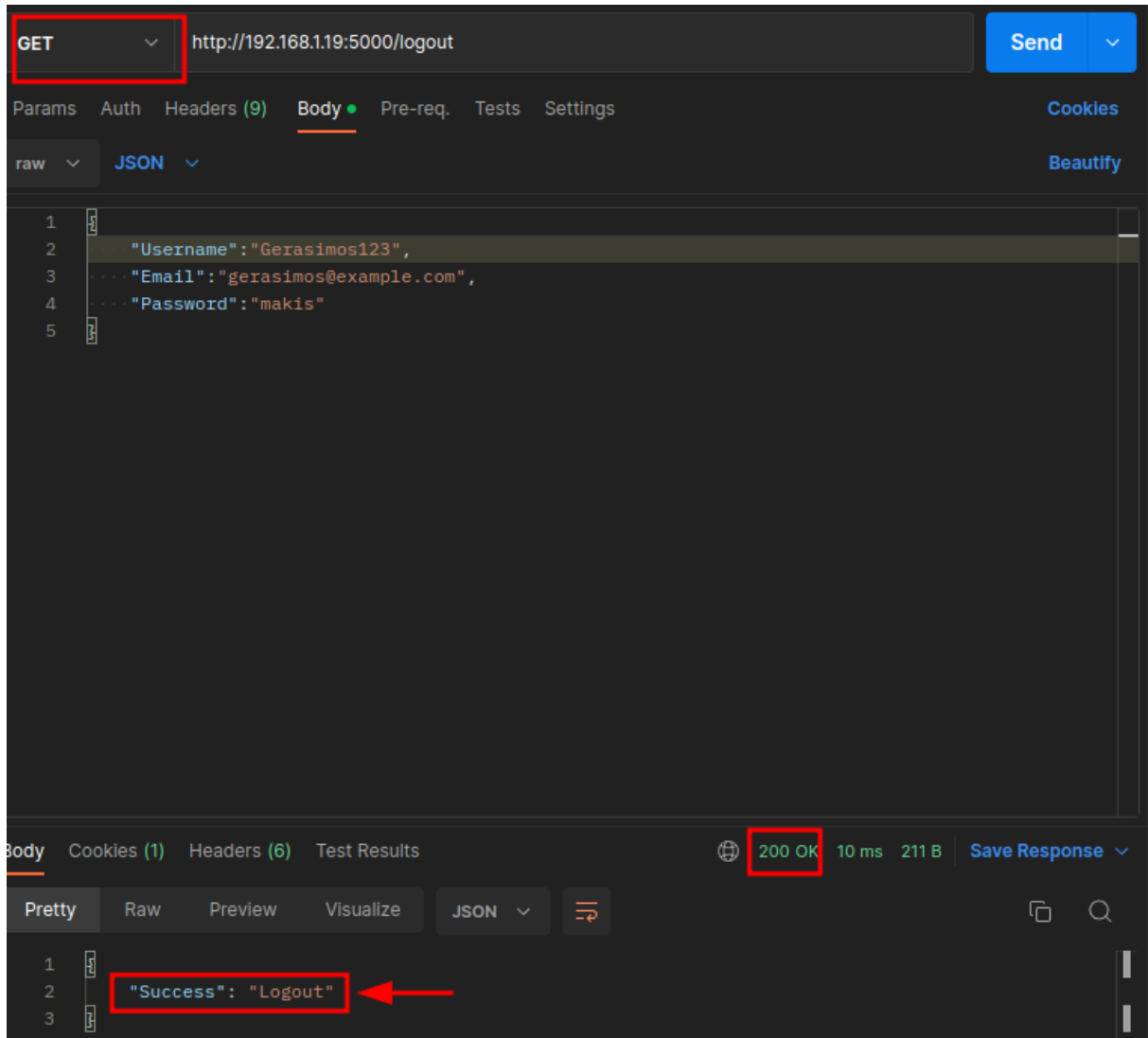
Logout: Κατά τη διαδικασία της αποσύνδεσης, ο Flask διακομιστής λαμβάνει το sessionId ως cookie από τον web-client. Στη συνέχεια, ελέγχει εάν το συγκεκριμένο sessionId υπάρχει καταχωρημένο στη βάση δεδομένων για τον εν λόγω χρήστη. Εάν δεν υπάρχει, τότε ο Flask server επιστρέφει ένα response στο web-client με κωδικό 400 και μήνυμα “Unable to Logout” (Σχήμα 4.18).



Σχήμα 4.18: Αποτυχημένη Αποσύνδεση

4.4. Flask Server

Εάν, αντίθετα, το sessionId υπάρχει στη βάση δεδομένων για τον συγκεκριμένο χρήστη, τότε διαγράφεται από τη βάση δεδομένων και ο διακομιστής επιστρέφει ένα response στον web-client με κωδικό 200 και μήνυμα “Logout” (Σχήμα 4.19).

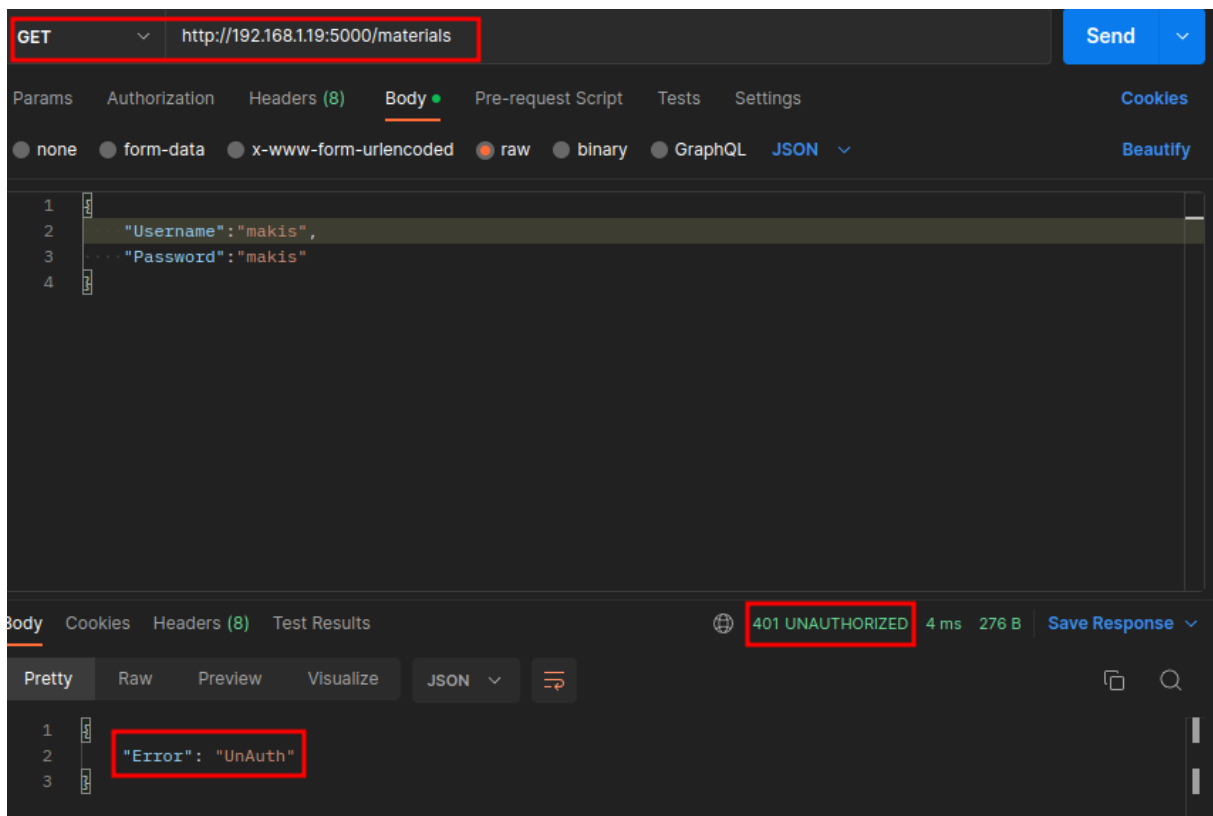


Σχήμα 4.19: Επιτυχής Αποσύνδεση

4.4. Flask Server

Με αυτόν τον τρόπο, το route Logout διασφαλίζει την ομαλή αποσύνδεση των χρηστών, ελέγχοντας την εγκυρότητα του sessionId και παρέχοντας την κατάλληλη απάντηση ανάλογα με την περίπτωση.

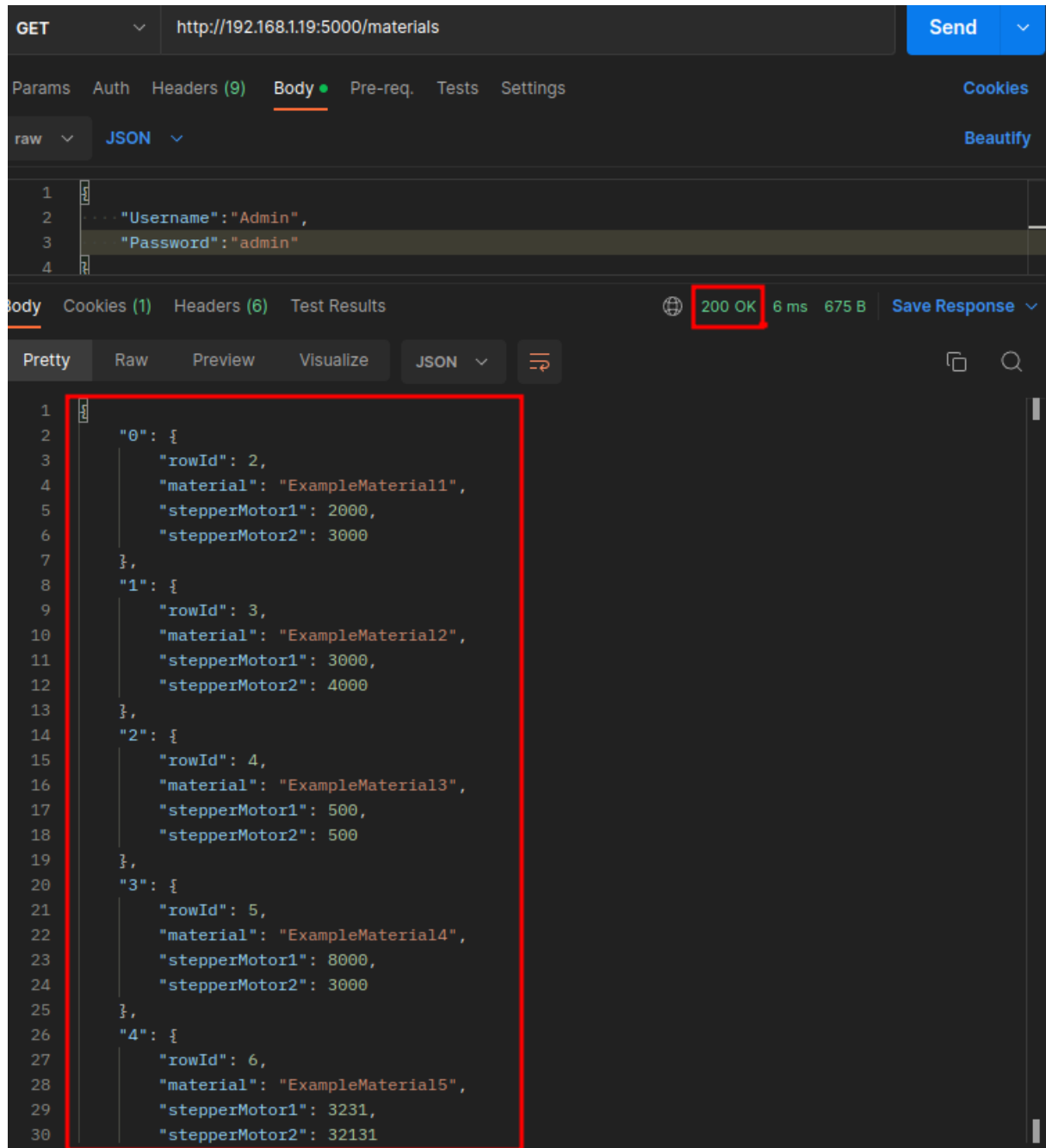
getMaterials: Στη διαδικασία του materials, αρχικά ο Flask διακομιστής λαμβάνει το cookie του χρήστη από τον web-client και ελέγχει αν ο χρήστης είναι εξουσιοδοτημένος, βεβαιώνοντας ότι το sessionId είναι καταχωρημένο στη βάση δεδομένων για τον συγκεκριμένο χρήστη. Εάν ο χρήστης δεν έχει εξουσιοδότηση, ο Flask server επιστρέφει ένα status code 401 και ένα μήνυμα “Unauthorized” σε μορφή JSON (Σχήμα 4.20).



Σχήμα 4.20: Μη Εξουσιοδοτημένος Χρήστης

4.4. Flask Server

Ωστόσο, εάν ο χρήστης είναι εξουσιοδοτημένος, τότε ανακτώνται όλα τα material Profiles που είναι αποθηκευμένα στη βάση δεδομένων για τον εν λόγω χρήστη όπως απεικονίζεται στο Σχήμα 4.21. Στη συνέχεια, αυτά τα προφίλ επιστρέφονται στον web-client με τη μορφή JSON, παρέχοντας έτσι μια ομαλή και αποτελεσματική απόκριση που περιλαμβάνει όλες τις απαραίτητες πληροφορίες για τα υλικά που σχετίζονται με τον χρήστη.

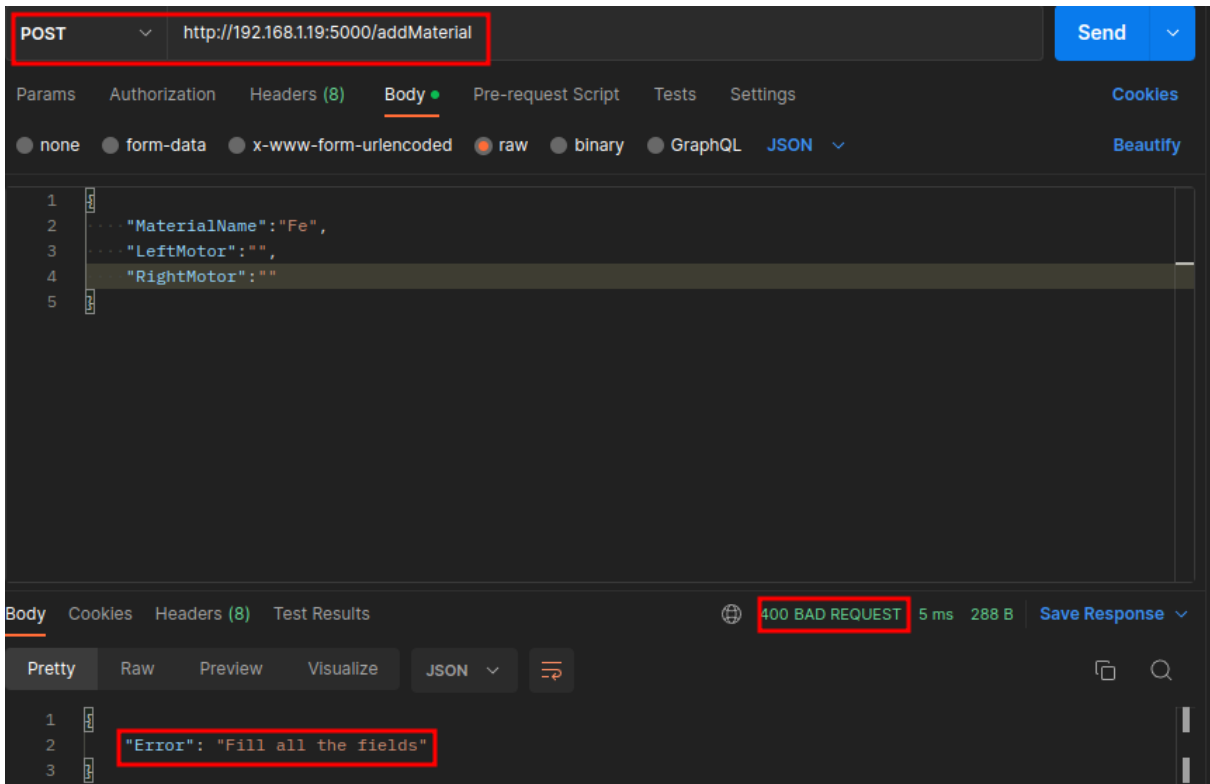


```
GET http://192.168.1.19:5000/materials

Body
JSON
[
  {
    "rowId": 2,
    "material": "ExampleMaterial1",
    "stepperMotor1": 2000,
    "stepperMotor2": 3000
  },
  {
    "rowId": 3,
    "material": "ExampleMaterial2",
    "stepperMotor1": 3000,
    "stepperMotor2": 4000
  },
  {
    "rowId": 4,
    "material": "ExampleMaterial3",
    "stepperMotor1": 500,
    "stepperMotor2": 500
  },
  {
    "rowId": 5,
    "material": "ExampleMaterial4",
    "stepperMotor1": 8000,
    "stepperMotor2": 3000
  },
  {
    "rowId": 6,
    "material": "ExampleMaterial5",
    "stepperMotor1": 3231,
    "stepperMotor2": 32131
  }
]
```

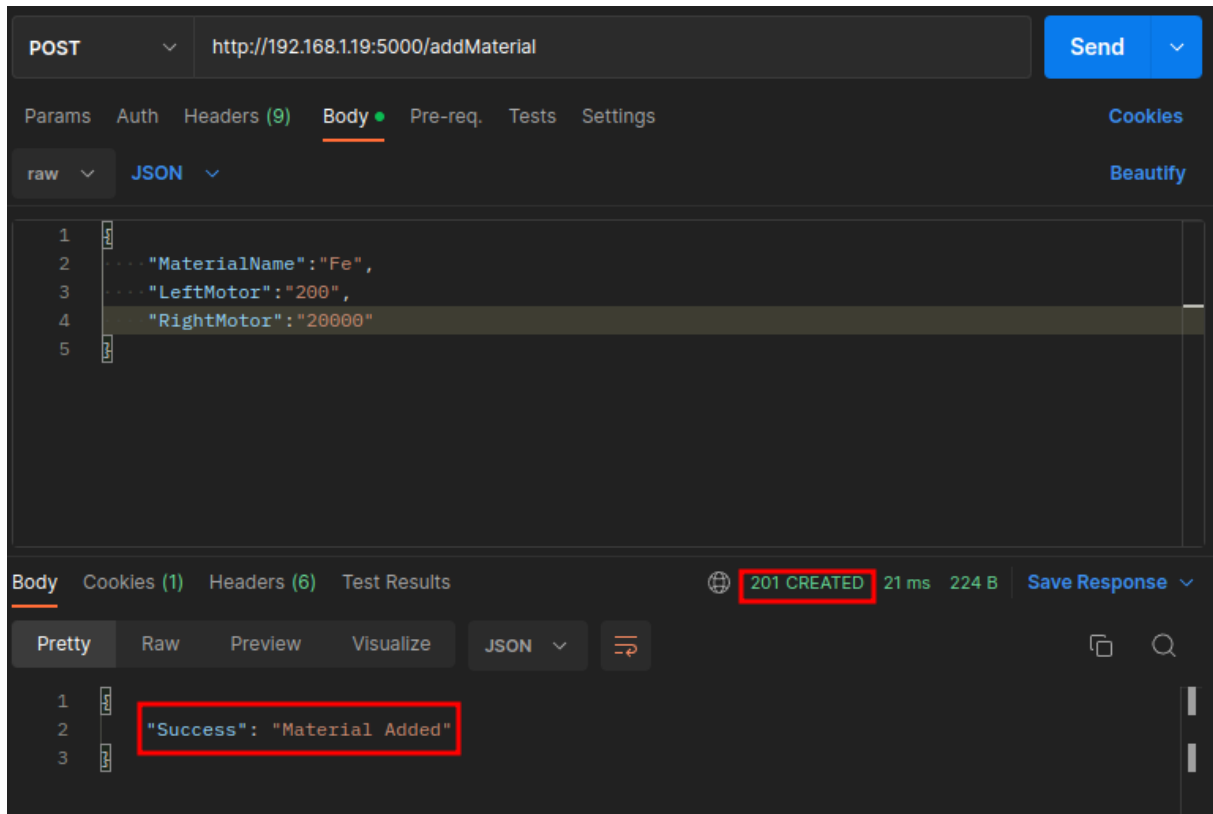
Σχήμα 4.21: Προφίλ Υλικών

AddMaterial: Κατά τη διαδικασία του “addMaterial“, ο διακομιστής λαμβάνει ένα POST αίτημα από τον web-client με το όνομα του υλικού και τις δύο τιμές των βηματικών κινητήρων σε μορφή JSON. Έπειτα, ελέγχει αν αυτές οι τιμές περιέχουν κάποια έγκυρη τιμή. Σε περίπτωση που οι τιμές είναι κενές, ο server επιστρέφει ένα response στον web-client με status code 400 και ένα μήνυμα “Fill all the fields“ σε μορφή JSON (Σχήμα 4.22).



Σχήμα 4.22: Αποτυχημένο Αίτημα Προσθήκης Υλικού: Ελλιπή Στοιχεία

Εάν, ωστόσο, οι τιμές δεν είναι κενές, το υλικό καθώς και οι τιμές των βηματικών κινητήρων καταχωρούνται στη βάση δεδομένων για τον συγκεκριμένο χρήστη. Στη συνέχεια, ο server επιστρέφει ένα response με status code 201 και ένα μήνυμα “Material Added“ σε μορφή JSON, επιβεβαιώνοντας την επιτυχή καταχώρηση του νέου υλικού στη βάση δεδομένων όπως φαίνεται στο Σχήμα 4.23.

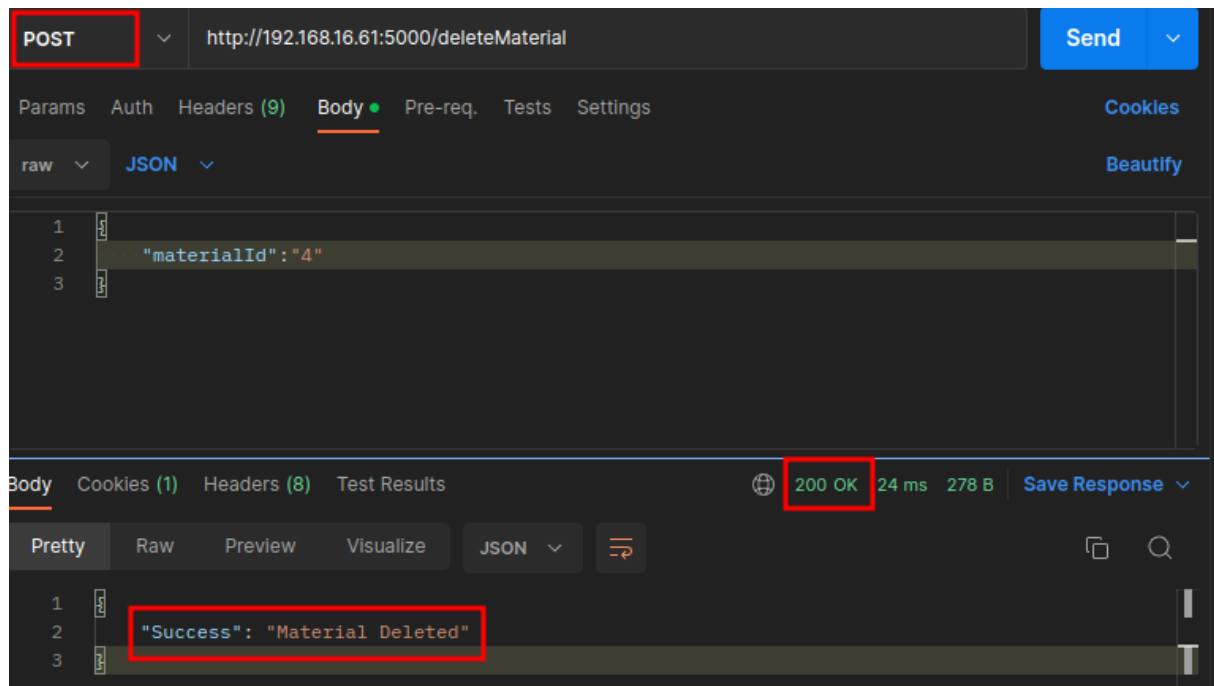


Σχήμα 4.23: Επιτυχής Προσθήκη Υλικού

id	user_id	material	firstMotor	secondMotor
1	2	ExampleMaterial1	2000	3000
2	3	ExampleMaterial2	3000	4000
3	4	ExampleMaterial3	500	500
4	5	ExampleMaterial4	8000	3000
5	6	ExampleMaterial5	3231	32131
6	7	Fe	200	20000

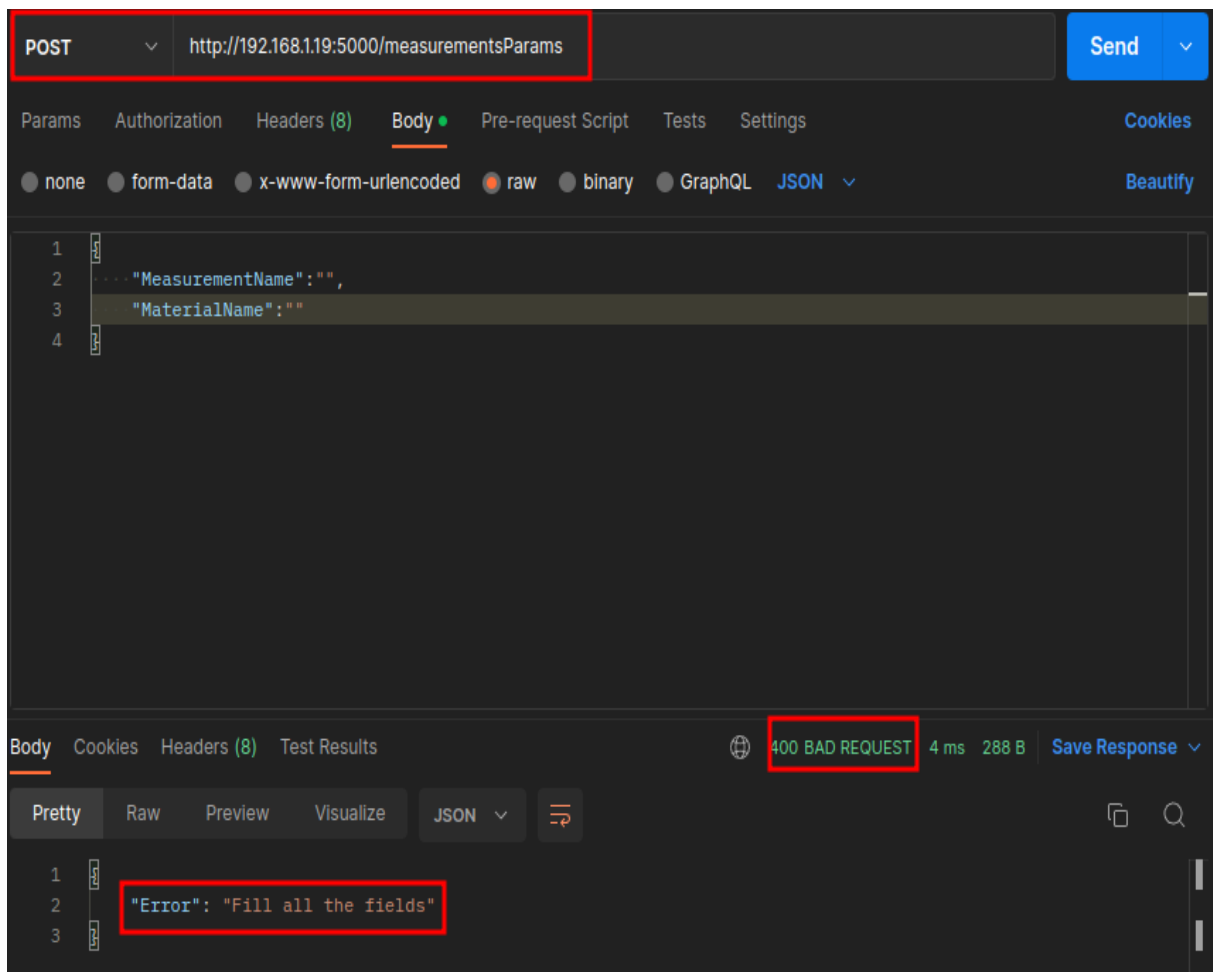
Σχήμα 4.24: Προφίλ Υλικών στη Βάση Δεδομένων

Delete Material: Κατά τη διαδικασία του “deleteMaterial“, πρώτα ελέγχεται αν ο χρήστης είναι εξουσιοδοτημένος. Σε περίπτωση που δεν είναι, επιστρέφεται ένα response στον web-client με ένα μήνυμα “Unauthorized“. Εάν, αντιθέτως, ο χρήστης είναι εξουσιοδοτημένος, τότε ο διακομιστής εκτελεί μια κλήση στη βάση δεδομένων για να διαγράψει το υλικό που ζητείται από τον χρήστη. Κατόπιν, επιστρέφεται ένα response σε μορφή JSON με status code 201 και το μήνυμα “Material Deleted“, επιβεβαιώνοντας την επιτυχημένη διαγραφή του υλικού (Σχήμα 4.25).



Σχήμα 4.25: Επιτυχής Διαγραφή Υλικού

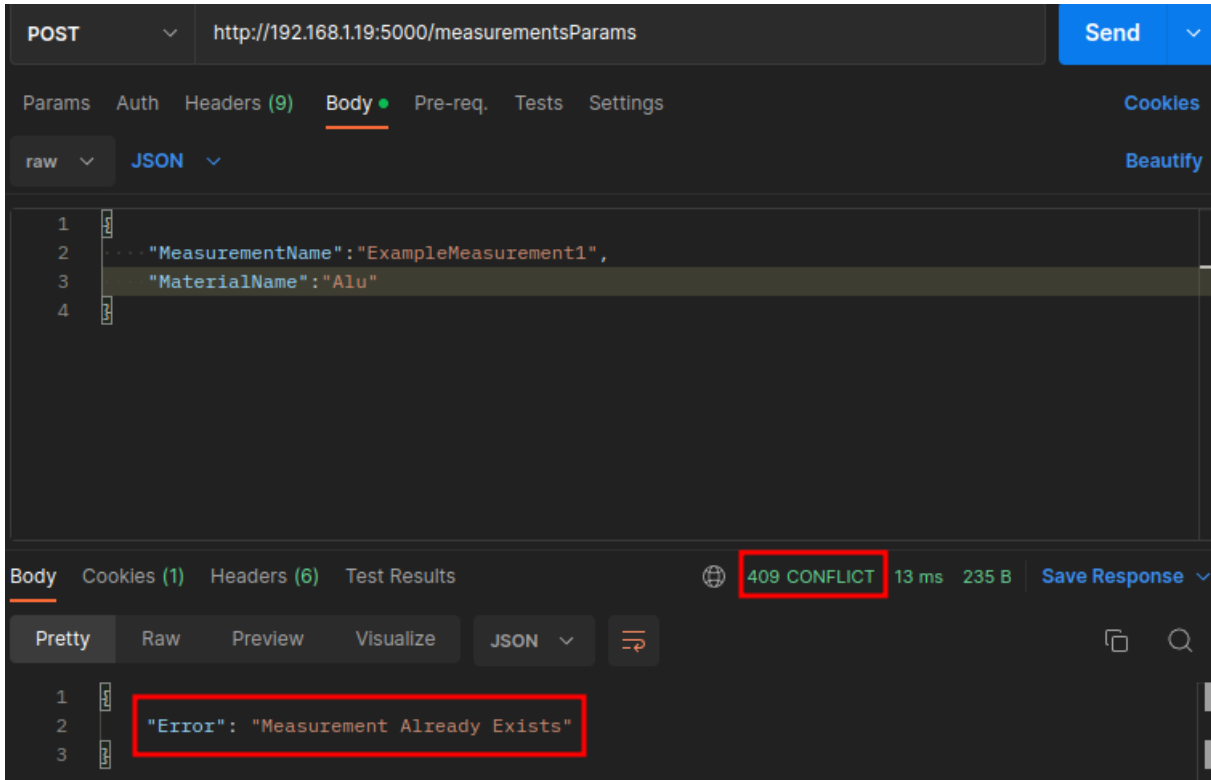
addMeasurement: Κατά τη διαδικασία του “measurementsParams“, ο διακομιστής λαμβάνει ένα POST αίτημα που περιέχει το όνομα της μέτρησης και το όνομα του υλικού σε μορφή JSON. Αρχικά, ο server ελέγχει αν και τα δύο πεδία είναι συμπληρωμένα. Εάν κάποιο από τα δεδομένα λείπει, ο server επιστρέφει ένα response στον web-client με κωδικό κατάστασης 400 και ένα μήνυμα λάθους “Fill all the fields“ (Σχήμα 4.26).



Σχήμα 4.26: Αποτυχημένη Προσθήκη Μέτρησης: Ελλιπή Στοιχεία

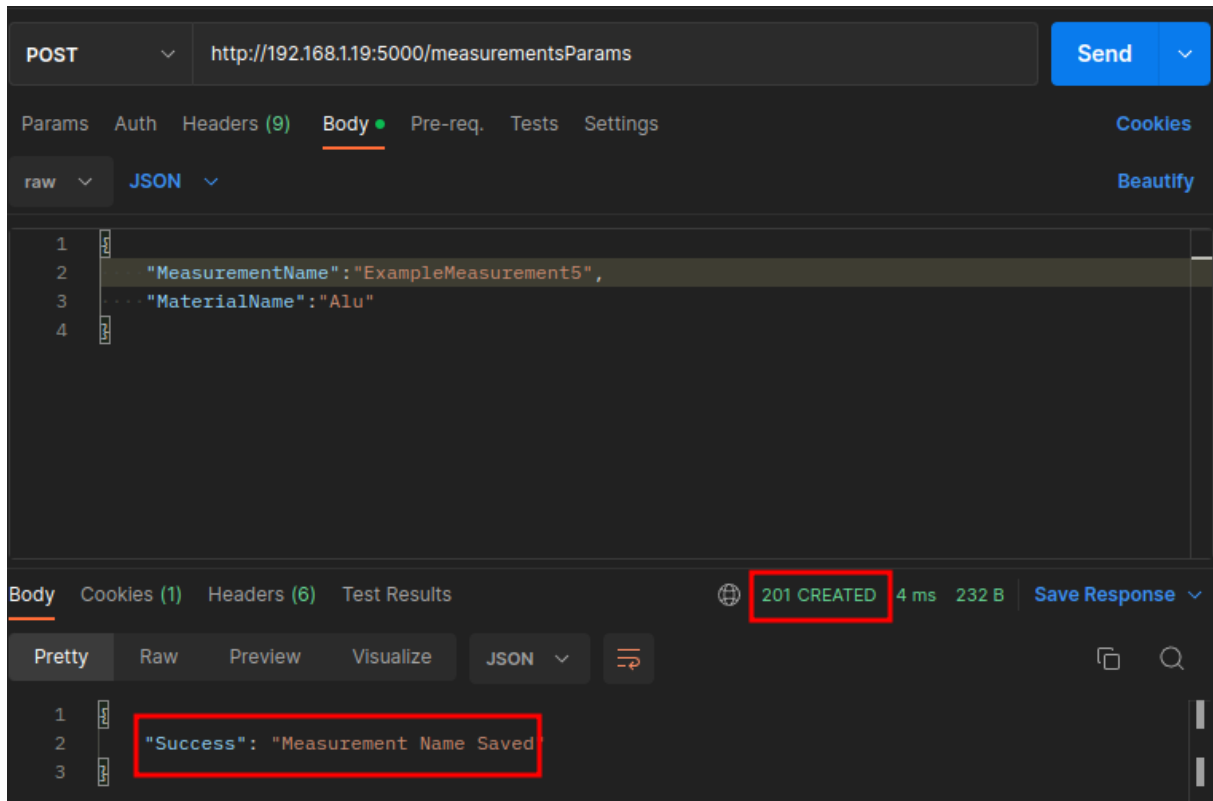
4.4. Flask Server

Σε περίπτωση που και τα δύο πεδία είναι συμπληρωμένα, ο διακομιστής κάνει μια κλήση στη βάση δεδομένων για να ελέγξει αν το όνομα της μέτρησης υπάρχει ήδη. Εάν το όνομα της μέτρησης υπάρχει στη βάση δεδομένων, ο διακομιστής επιστρέφει ένα response με κωδικό κατάστασης 409 και ένα μήνυμα λάθους “Measurement Already Exists” όπως φαίνεται στο Σχήμα 4.27.



Σχήμα 4.27: Αποτυχημένη Καταχώρηση Μέτρησης: Διπλοκαταχώρηση Μέτρησης

Αντίθετα, αν το όνομα της μέτρησης δεν υπάρχει, ο διακομιστής επιστρέφει ένα επιτυχημένο response με κωδικό κατάστασης 201 και ένα μήνυμα “Measurement Name Saved” (Σχήμα 4.28). Με αυτόν τον τρόπο, η διαδικασία διαχείρισης των ονομάτων μετρήσεων γίνεται με κομψό και αποτελεσματικό τρόπο, διασφαλίζοντας την ορθότητα των δεδομένων και την εύκολη πρόσβαση των χρηστών στις πληροφορίες τους.



Σχήμα 4.28: Επιτυχής Αποθήκευση Ονόματος Μέτρησης

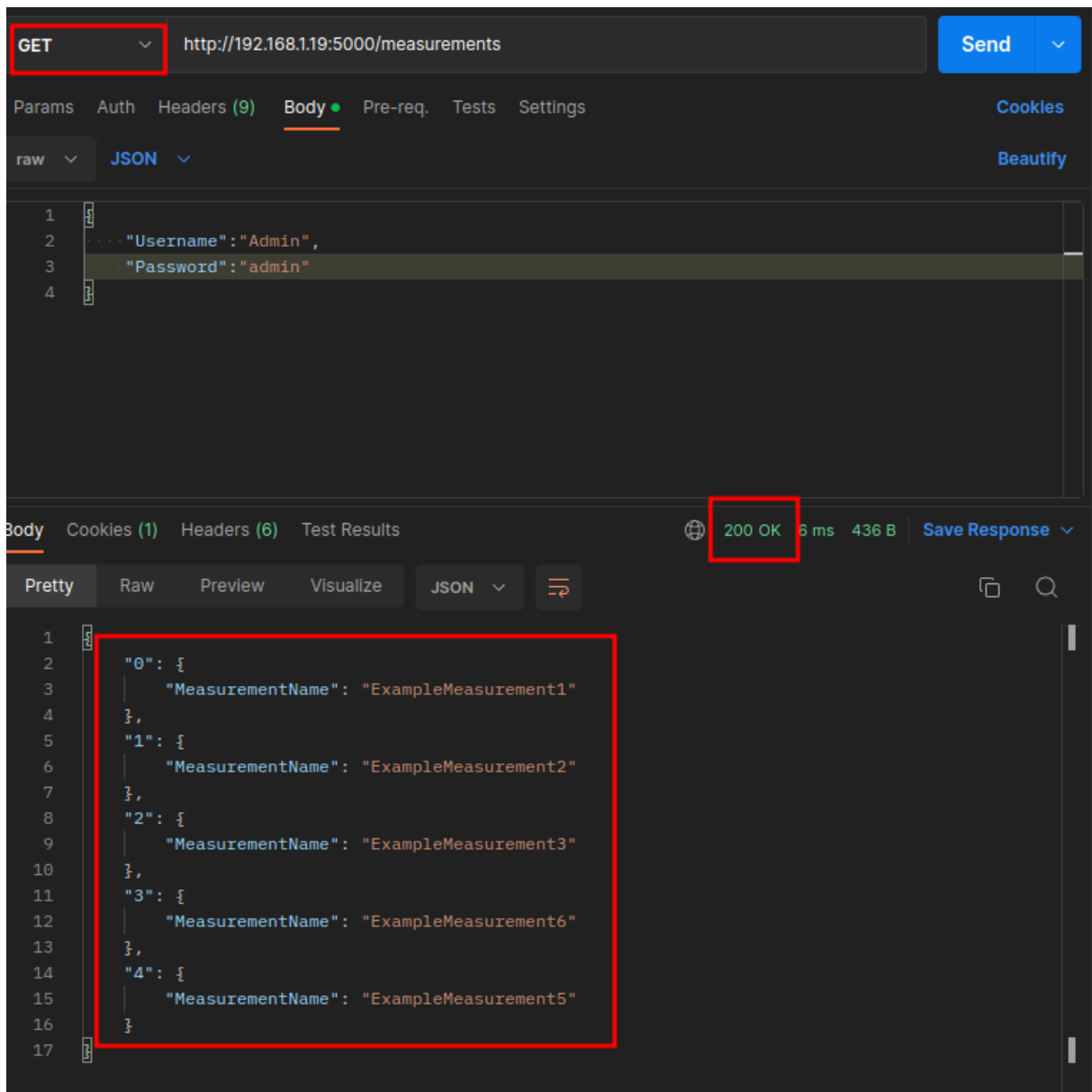
Με την εφαρμογή αυτής της διαδικασίας, οι χρήστες εξασφαλίζουν τη σωστή οργάνωση και αποθήκευση των μετρήσεων τους. Επιπλέον, η αυτοματοποιημένη διαδικασία ελέγχου των ονομάτων μετρήσεων εξαλείφει τον κίνδυνο διπλοκαταχωρήσεων ή λανθασμένων εισαγωγών. Με αυτό το σύστημα, οι χρήστες μπορούν να επικεντρωθούν στην παρακολούθηση και την ανάλυση των αποτελεσμάτων των μετρήσεων, χωρίς να ανησυχούν για τυχόν τεχνικά ζητήματα που σχετίζονται με τη διαχείριση των δεδομένων τους.

getMeasurements: Κατά τη διαδικασία του “measurements“, ο Flask server αρχικά λαμβάνει το cookie του χρήστη από τον web-client και πραγματοποιεί έλεγχο εξουσιοδότησης, επιβεβαιώνοντας την ύπαρξη του sessionId για τον συγκεκριμένο χρήστη στη βάση δεδομένων. Εάν η εξουσιοδότηση δεν επιτρέπεται, ο Flask server αποστέλλει ένα status code 401 και ένα μήνυμα “Unauthorized“ σε μορφή JSON.

Ωστόσο, εφόσον ο χρήστης είναι εξουσιοδοτημένος, οι μετρήσεις που αποθηκεύτηκαν για τον χρήστη ανακτώνται από τη βάση δεδομένων. Στη συνέχεια, αυτές οι μετρήσεις επιστρέφονται στον web-client ως JSON όπως φα-

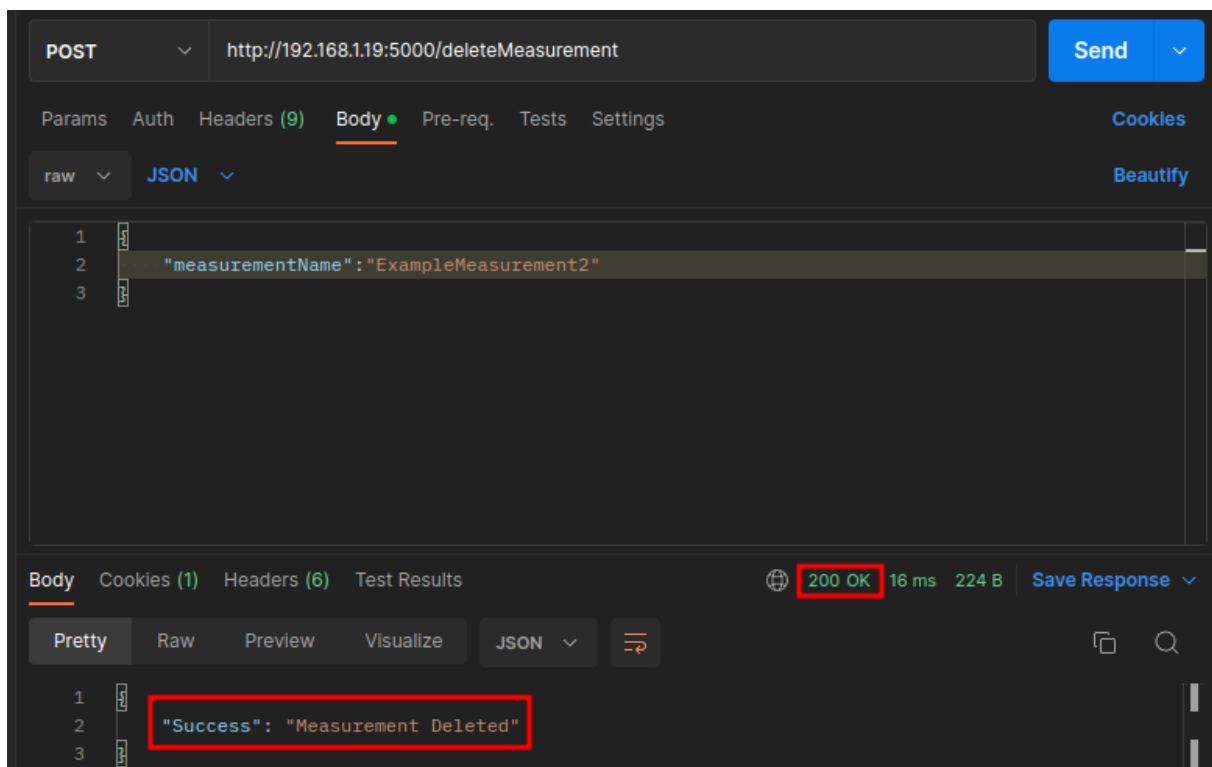
4.4. Flask Server

ίνεται στο Σχήμα 4.29, προσφέροντας μια εύκολη διαχείριση των μετρήσεων που έχει κάνει ο χρήστης.



Σχήμα 4.29: Αίτημα Λήψης Μετρήσεων

deleteMeasurement: Κατά τη διαδικασία του “deleteMeasurement”, πρώτα ελέγχεται αν ο χρήστης είναι εξουσιοδοτημένος. Σε περίπτωση που δεν είναι, επιστρέφεται ένα response στον web-client με ένα μήνυμα “Unauthorized”. Εάν, αντιθέτως, ο χρήστης είναι εξουσιοδοτημένος, τότε ο Flask server εκτελεί μια κλήση στη βάση δεδομένων για να διαγράψει τη μέτρηση που ζητείται από τον web-client. Κατόπιν, επιστρέφεται ένα response σε μορφή JSON με status code 201 και το μήνυμα “Measurement Deleted”, επιβεβαιώνοντας την επιτυχημένη διαγραφή της μέτρησης όπως φαίνεται στο Σχήμα στο 4.30.



Σχήμα 4.30: Διαγραφή Μέτρησης

getData: Το end-point getData λαμβάνει τα δεδομένα των αισθητήρων από τον ESP32 μέσω ενός POST αίτημα. Μετά τη λήψη των δεδομένων για τη ροπή, τις στροφές και την κάθετη δύναμη, δημιουργείται ένα στιγμιότυπο της ώρας και πραγματοποιείται μια κλήση στη βάση δεδομένων για την αποθήκευσή τους.

Στη συνέχεια, ο διακομιστής προωθεί αυτά τα δεδομένα προς τον web-client μέσω web-sockets, παρέχοντας μια ταχύτερη απόκριση σε σύγκριση με ένα POST αίτημα.

sendData: Κατά τη διαδικασία του `sendData`, ο Flask server χρησιμοποιεί `web-sockets` για να αποστείλει τις τιμές των βηματικών κινητήρων σε μορφή JSON με ταχύτερη απόκριση στο ESP32. Αυτό επιτρέπει μια πιο άμεση και αποτελεσματική επικοινωνία μεταξύ του διακομιστή και των κινητήρων, επιταχύνοντας την ανταλλαγή δεδομένων και βελτιώνοντας την απόδοση της εφαρμογής.

generateSessionId: Η συνάρτηση `generateSessionId` είναι υπεύθυνη για τη δημιουργία ενός μοναδικού `sessionId` για κάθε χρήστη. Αυτό επιτυγχάνεται με τη χρήση της συνάρτησης `uuid.uuid4()` του `module UUID` της Python, η οποία δημιουργεί ένα τυχαίο και μοναδικό αναγνωριστικό 128-bit.

Το UUID που παράγεται από τη συνάρτηση `uuid.uuid4()` βασίζεται σε τυχαίους αριθμούς, όπως τον χρόνο, τον αριθμό υπολογιστή και τη διεύθυνση MAC, εξασφαλίζοντας ότι κάθε `sessionId` θα είναι διαφορετικό. Η πιθανότητα δημιουργίας δύο ίδιων UUIDs είναι εξαιρετικά μικρή, καθιστώντας αυτή τη μέθοδο ασφαλή και αξιόπιστη για τη διαχείριση των `sessionIDs` των χρηστών. Η συνάρτηση επιστρέφει το παραγόμενο UUID ως συμβολοσειρά χαρακτήρων, έτοιμο για χρήση στην εφαρμογή.

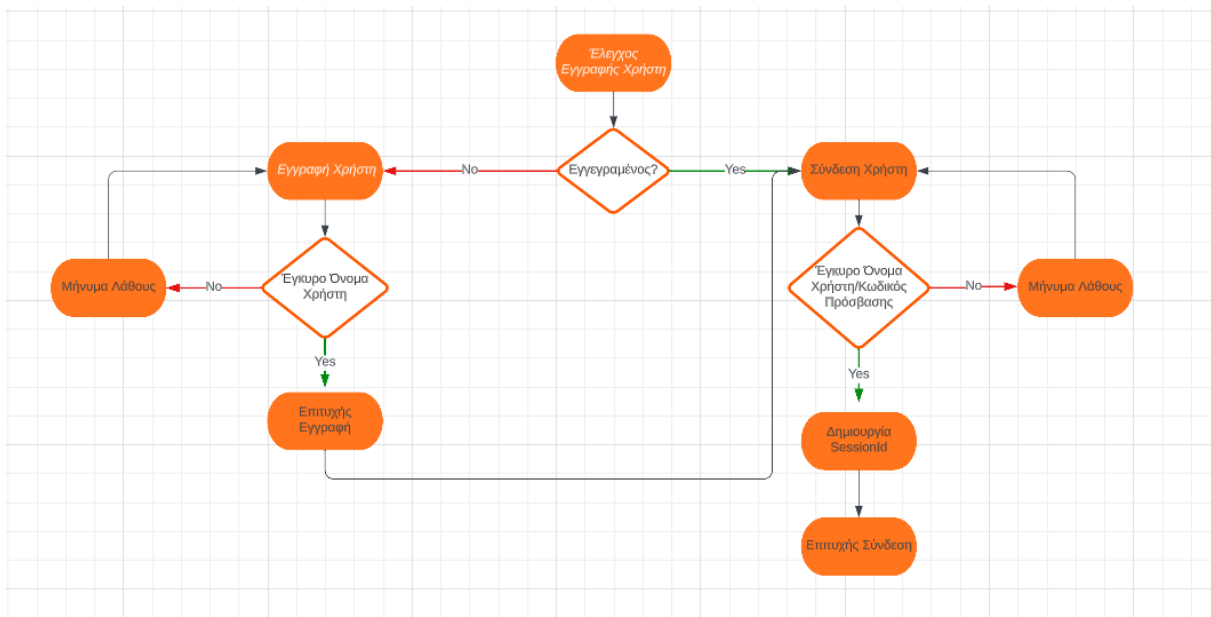
4.5 Web-Γραφικό Περιβάλλον

Στο κατανεμημένο σύστημα που έχουμε αναπτύξει, το γραφικό περιβάλλον διαδραματίζει έναν ζωτικό ρόλο στη διασύνδεση των χρηστών με την πλατφόρμα. Το γραφικό περιβάλλον αποτελεί την προσωπική πύλη των χρηστών προς τις λειτουργίες και τις υπηρεσίες που παρέχονται από το σύστημα, επιτρέποντας τους να διαχειρίζονται τα δεδομένα και να επικοινωνούν με το σύστημα μέσω ενός φιλικού προς τον χρήστη και εύχρηστου περιβάλλοντος.

Με τη χρήση αυτού, οι χρήστες μπορούν να κάνουν εγγραφή, να συνδεθούν, να αποκτήσουν πρόσβαση στα δεδομένα και τις υπηρεσίες που είναι διαθέσιμες στην πλατφόρμα, καθώς και να επικοινωνούν το ίδιο το σύστημα. Η εμπειρία των χρηστών σχεδιάστηκε έτσι ώστε να είναι απλή, κατανοητή και αποτελεσματική, εξασφαλίζοντας την ομαλή λειτουργία του συστήματος και την ικανοποίηση των χρηστών.

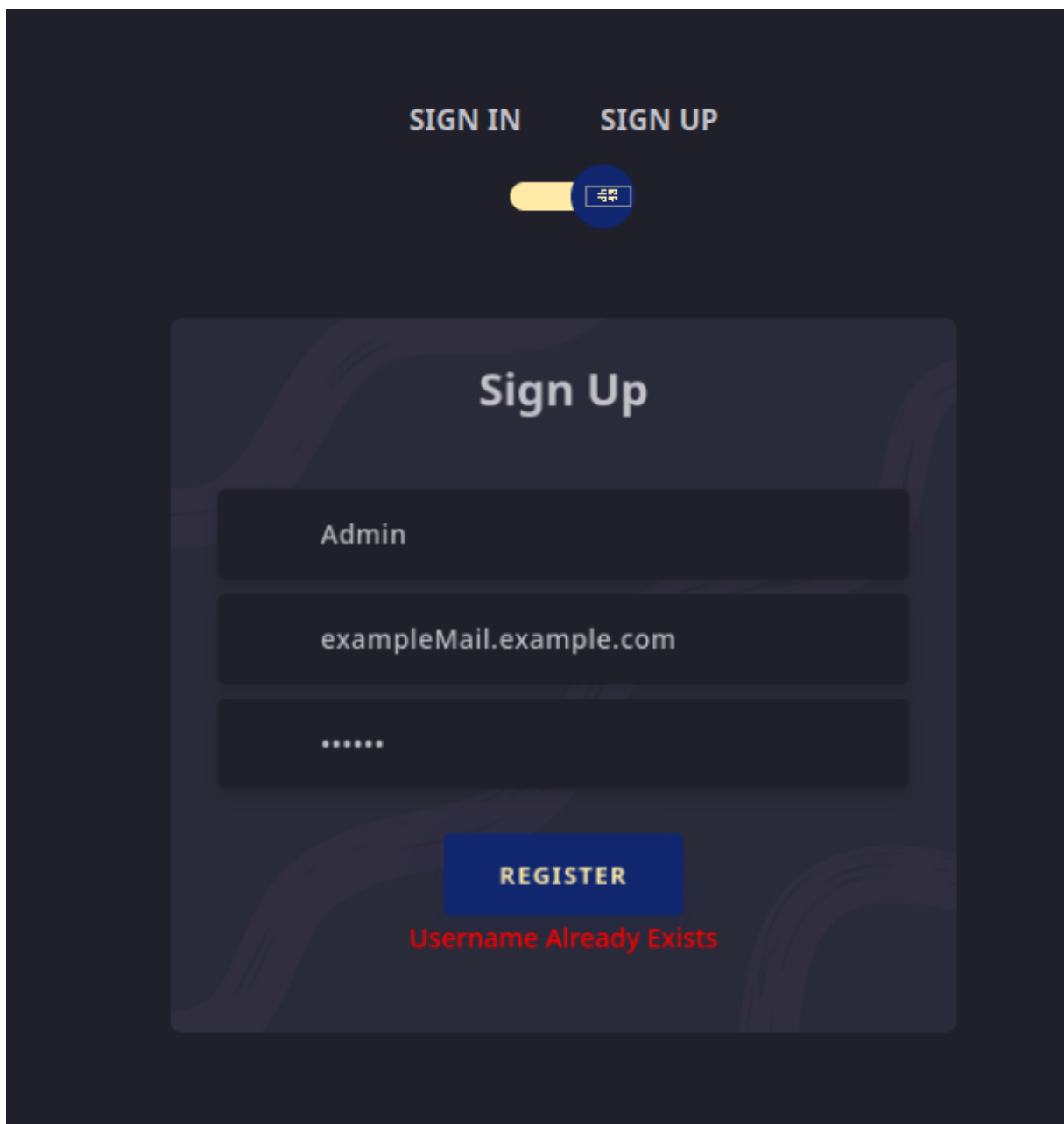
4.5.1 Υλοποίηση Γραφικού Περιβάλλοντος

Εγγραφή και Σύνδεση στο Σύστημα: Στο διάγραμμα ροής που απεικονίζεται στο σχήμα 4.31, το οποίο θα απεικονίζει τη διαδικασία εγγραφής ή σύνδεσης του χρήστη στο σύστημα. Μέσω αυτού του διαγράμματος, θα γίνει πιο κατανοητή η διαδικασία που ακολουθούν οι χρήστες για να δημιουργήσουν έναν λογαριασμό ή να συνδεθούν στην πλατφόρμα, αποκαλύπτοντας τις βασικές επιλογές και τις λειτουργίες που προσφέρονται κατά τη διάρκεια αυτής της διαδικασίας.



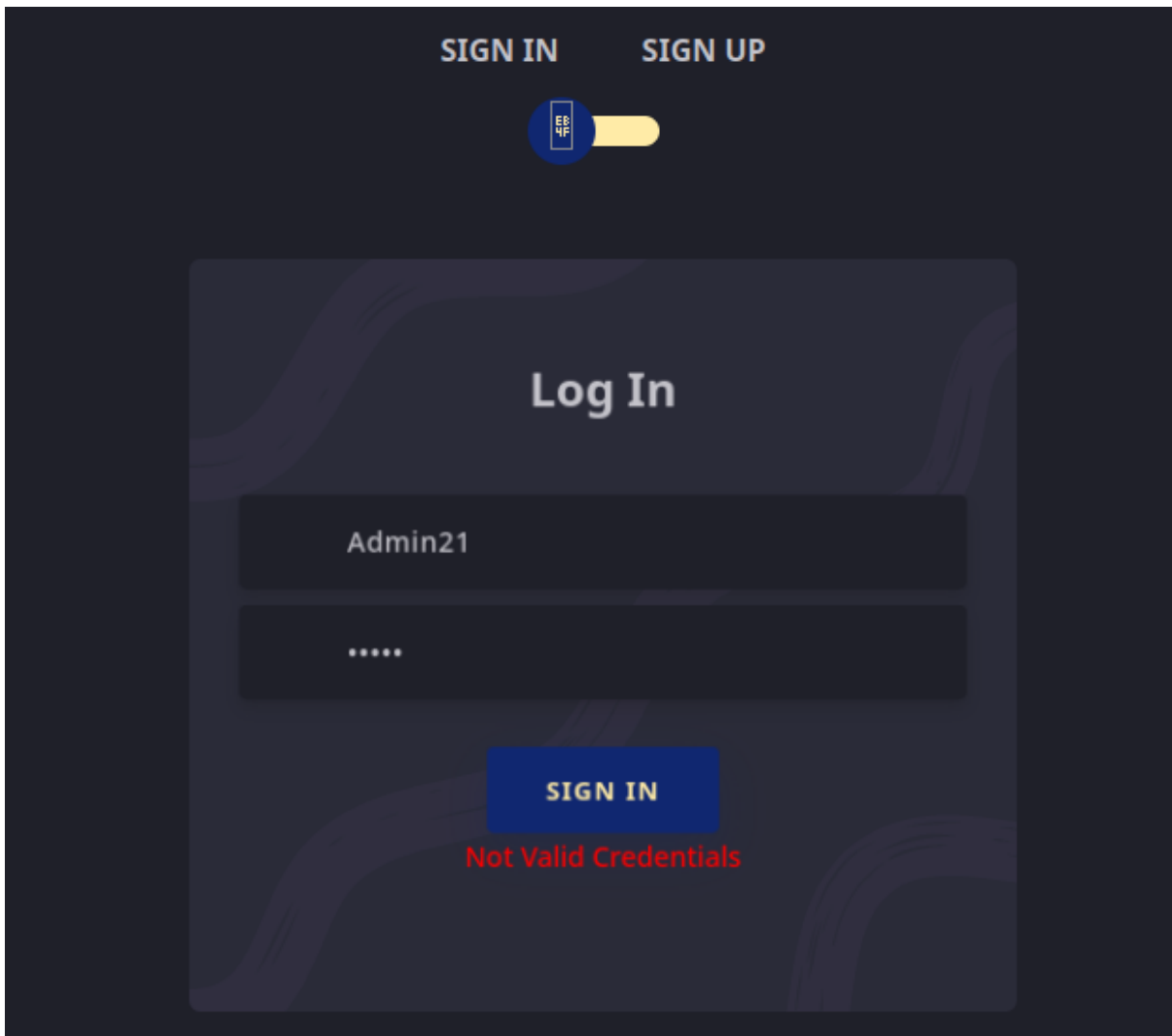
Σχήμα 4.31: Διάγραμμα Ροής Εγγραφής/Σύνδεσης Χρήστη

Αρχικά, όταν ο χρήστης συνδέεται στο γραφικό περιβάλλον, μέσω κάποιου κινητού τηλεφώνου ή ενός υπολογιστή, χωρίς να έχει ήδη λογαριασμό θα πρέπει να εγγραφεί στο σύστημα. Για να το πραγματοποιήσει αυτό, καταχωρεί τα προσωπικά του στοιχεία (όνομα χρήστη, διεύθυνση ηλεκτρονικού ταχυδρομείου, κωδικός πρόσβασης). Στη συνέχεια, το σύστημα ελέγχει αν το όνομα χρήστη υπάρχει ήδη στη βάση δεδομένων. Εάν το όνομα χρήστη υπάρχει, εμφανίζεται ένα μήνυμα λάθους στην οθόνη του χρήστη (Σχήμα 4.32).



Σχήμα 4.32: Αποτυχημένη Εγγραφή Χρήστη

Αντίθετα, εάν το όνομα χρήστη δεν υπάρχει στη βάση δεδομένων, η εγγραφή του χρήστη θεωρείται επιτυχής και αυτόματα μεταφέρεται στη σελίδα σύνδεσης. Εάν ο χρήστης έχει ήδη εγγραφεί στο σύστημα, εισάγει τα προσωπικά του στοιχεία (όνομα χρήστη, κωδικός πρόσβασης) για να συνδεθεί. Το σύστημα ελέγχει τα δεδομένα προκειμένου να επιβεβαιώσει την ταυτότητα του χρήστη.



Σχήμα 4.33: Αποτυχημένη Σύνδεση Χρήστη: Μη Έγκυρα στοιχεία Χρήστη

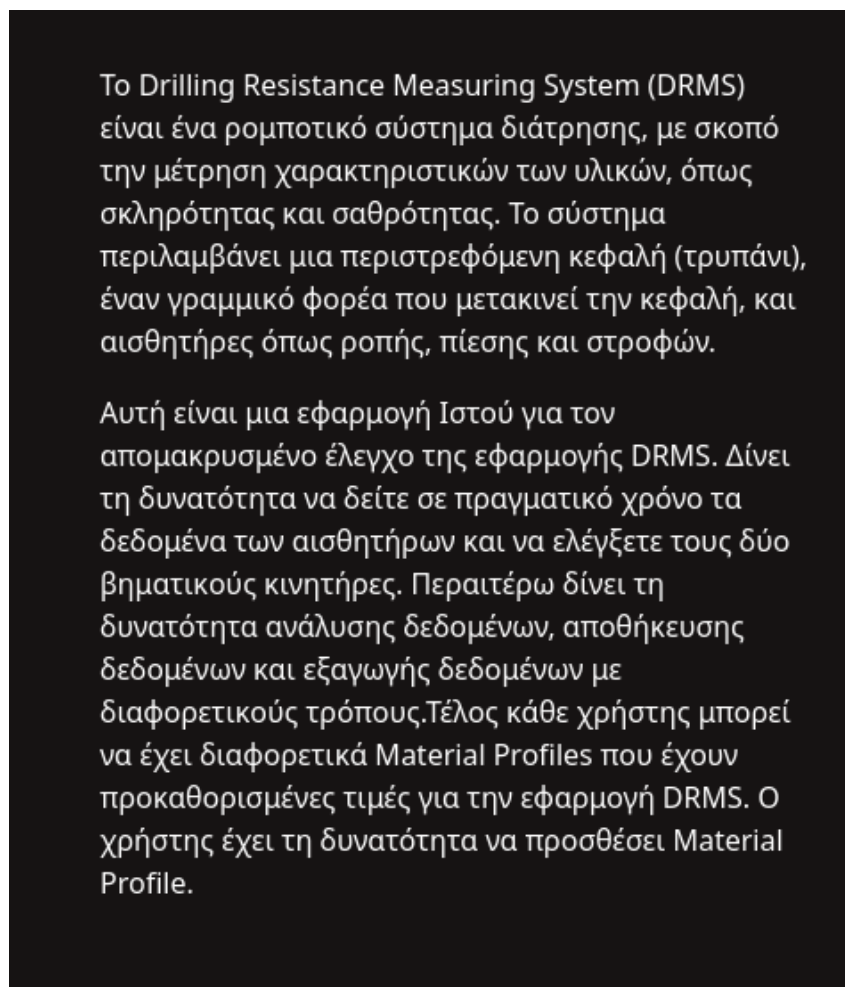
Εάν τα δεδομένα δεν είναι έγκυρα, εμφανίζεται ένα μήνυμα λάθους (Σχήμα 4.33) και ο χρήστης επιστρέφει στη σελίδα σύνδεσης. Εάν τα δεδομένα είναι έγκυρα, τότε παράγεται ένα μοναδικό `sessionId` και η σύνδεση του χρήστη θεωρείται επιτυχής. Με αυτόν τον τρόπο, ο χρήστης είναι πλέον συνδεδεμένος στο σύστημα και μπορεί να αξιοποιήσει τις λειτουργίες και τις υπηρεσίες που παρέχονται από την πλατφόρμα.

Πλοήγηση: Μετά από μια επιτυχημένη σύνδεση στο DRMS σύστημα, ο χρήστης αποκτά τη δυνατότητα να πλοηγηθεί και να εξερευνήσει τις υπηρεσίες που παρέχει η πλατφόρμα μας.



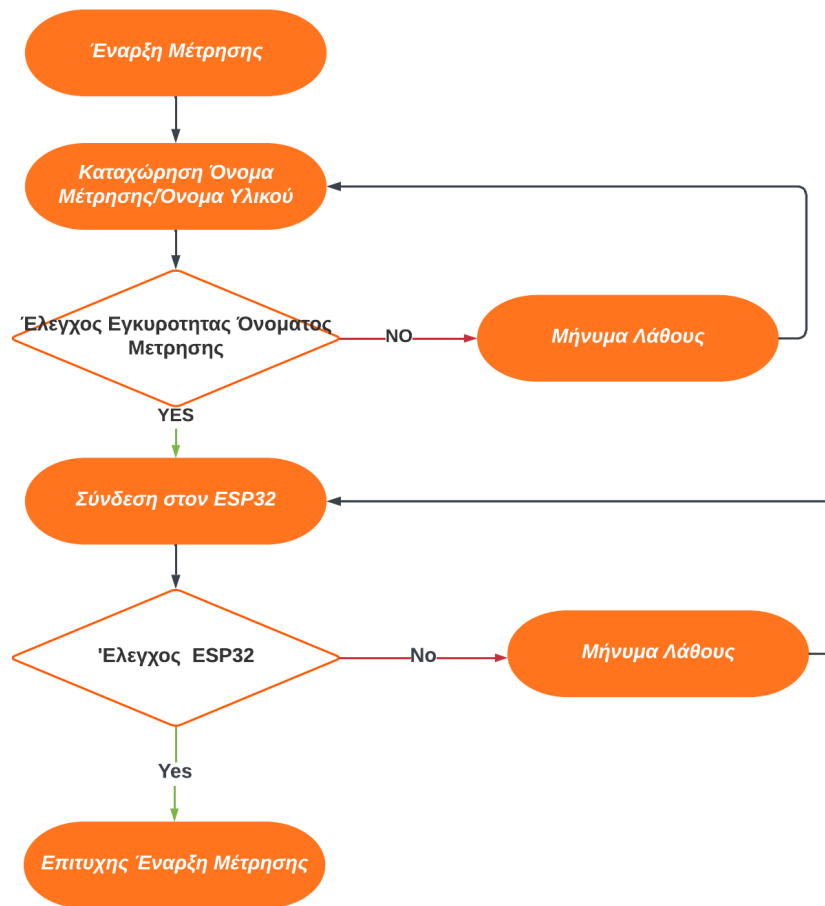
Σχήμα 4.34: Μπάρα Πλοήγησης

Home: Στην αρχική ιστοσελίδα, ο χρήστης έχει τη δυνατότητα να ενημερωθεί σχετικά με τη χρήση, τις λειτουργίες και τις δυνατότητες που προσφέρει το σύστημά μας.



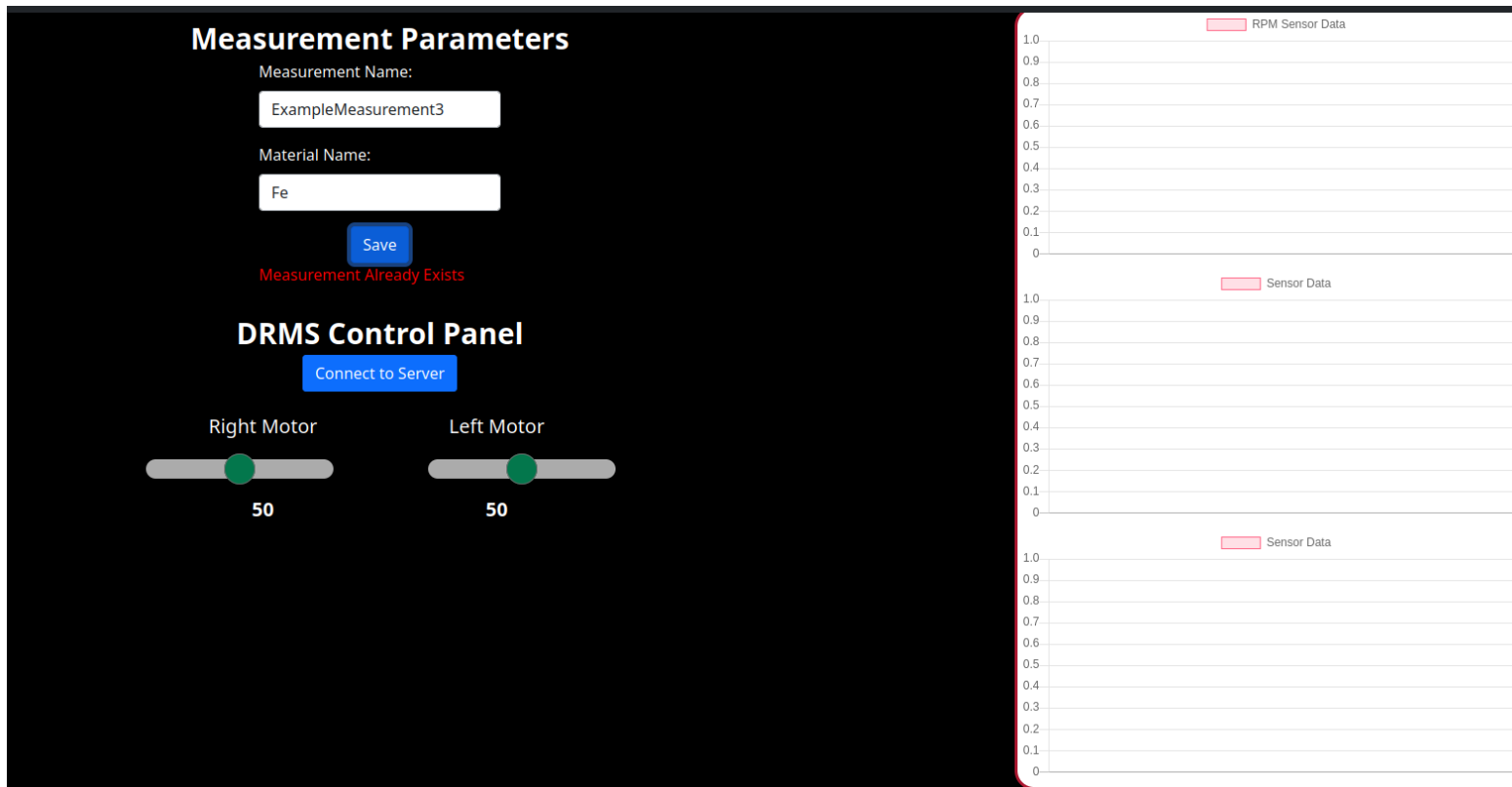
Σχήμα 4.35: Αρχική Σελίδα

DRMS: Στη σελίδα DRMS, ο χρήστης έχει την ευκαιρία να ξεκινήσει μια μέτρηση. Στο σχήμα 4.36 παρουσιάζεται ένα διάγραμμα ροής που απεικονίζει τη διαδικασία έναρξης μιας μέτρησης στο σύστημά μας. Το διάγραμμα ροής έχει σχεδιαστεί για να κατανοήσουν οι χρήστες τα βήματα και τις επιλογές που πρέπει να ακολουθήσουν, προκειμένου να ξεκινήσουν μια επιτυχημένη μέτρηση.



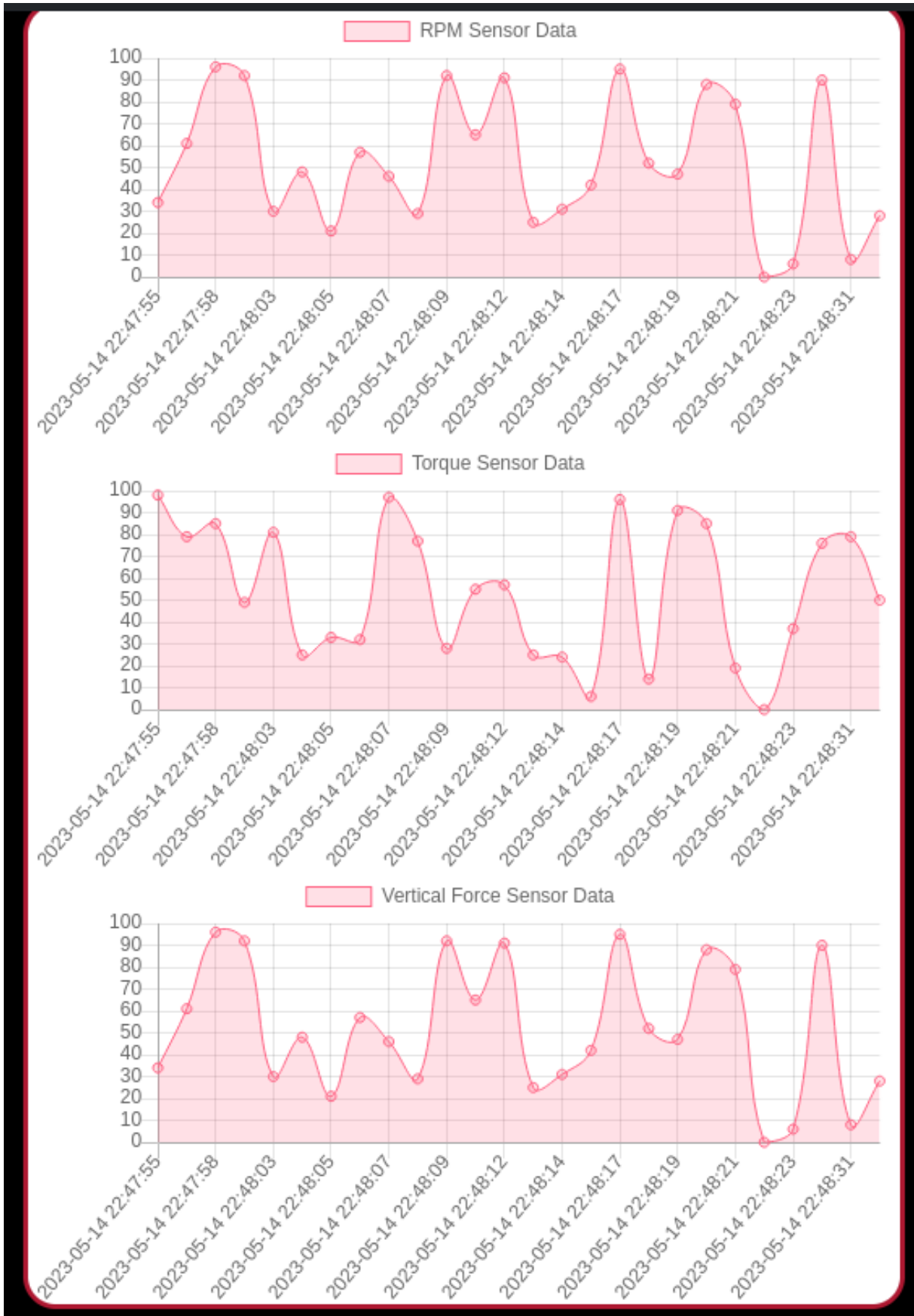
Σχήμα 4.36: Διάγραμμα Ροής Έναρξης Μέτρησης

Διαδικασία Έναρξης Μέτρησης: Για να ξεκινήσει ο χρήστης μια μέτρηση, θα πρέπει αρχικά να καταχωρήσει το όνομα της μέτρησης και το όνομα του υλικού που επιθυμεί να μετρήσει. Εάν το όνομα της μέτρησης υπάρχει ήδη στη βάση δεδομένων του συστήματος, τότε εμφανίζεται ένα μήνυμα λάθους στον χρήστη όπως απεικονίζεται στο Σχήμα 4.37.



Σχήμα 4.37: Διπλοκαταχώρηση Μέτρησης

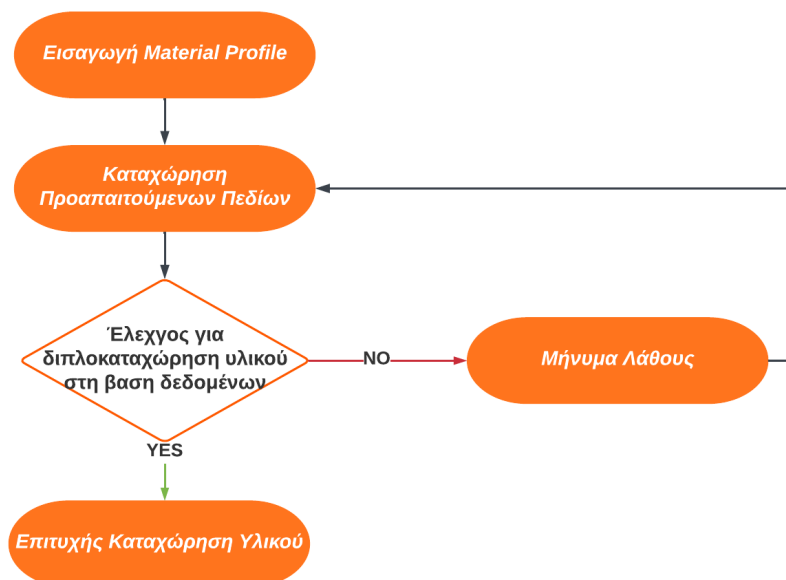
Αντίθετα, αν το όνομα δεν υπάρχει, ο χρήστης συνεχίζει κάνοντας σύνδεση στον server. Στη συνέχεια, πραγματοποιείται ένας έλεγχος για τη διαθεσιμότητα του server εάν δεν είναι διαθέσιμος, εμφανίζεται ένα μήνυμα λάθους στον χρήστη. Σε αντίθετη περίπτωση, ο χρήστης πραγματοποιεί μια επιτυχημένη έναρξη μέτρησης.



Σχήμα 4.38: Δεδομένα Αισθητήρων

Material Profiles: Στην ενότητα “Material Profiles“, ο χρήστης έχει τη δυνατότητα να προβάλλει τα υπάρχοντα προφίλ υλικών, να εισάγει νέα και να διαγράψει υπάρχοντα. Η σελίδα αυτή στοχεύει στην παροχή προκαθορισμένων τιμών για τους βηματικούς κινητήρες σε συγκεκριμένα υλικά. Οι τιμές αυτές πιθανότατα προέρχονται από κάποια ανάλυση που έχει πραγματοποιηθεί επάνω στο εκάστοτε υλικό, εξασφαλίζοντας κατ'αυτόν τον τρόπο βέλτιστες ρυθμίσεις για την επεξεργασία των υλικών.

Στο Σχήμα 4.39 παρουσιάζεται ένα διάγραμμα ροής, το οποίο απεικονίζει την διαδικασία εισαγωγής ενός Material Profile. Με αυτό το διάγραμμα, στόχος μας είναι η σαφής απόδοση των βημάτων που θα πρέπει να ακολουθήσει ο χρήστης, ώστε να έχει μια επιτυχημένη καταχώριση υλικού.

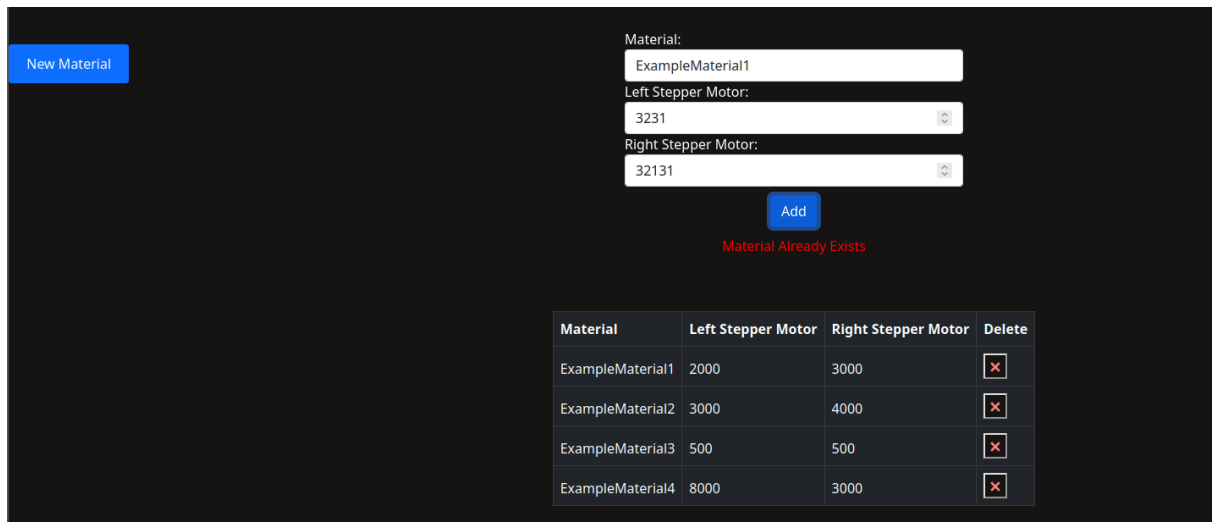


Σχήμα 4.39: Διάγραμμα Ροής Εισαγωγής Προφίλ Υλικού

Διαδικασία Εισαγωγής Υλικού: Η διαδικασία ξεκινά με την επιλογή του επιθυμητού υλικού και την πληκτρολόγηση του ονόματός του, καθώς και

4.5. Web-Γραφικό Περιβάλλον

τις τιμές των βηματικών κινητήρων. Στη συνέχεια, το σύστημα ελέγχει αν το όνομα του υλικού υπάρχει ήδη στη βάση δεδομένων. Εάν υπάρχει, εμφανίζεται ένα μήνυμα λάθους (Σχήμα 4.40) και ο χρήστης καλείται να επαναλάβει τη διαδικασία.



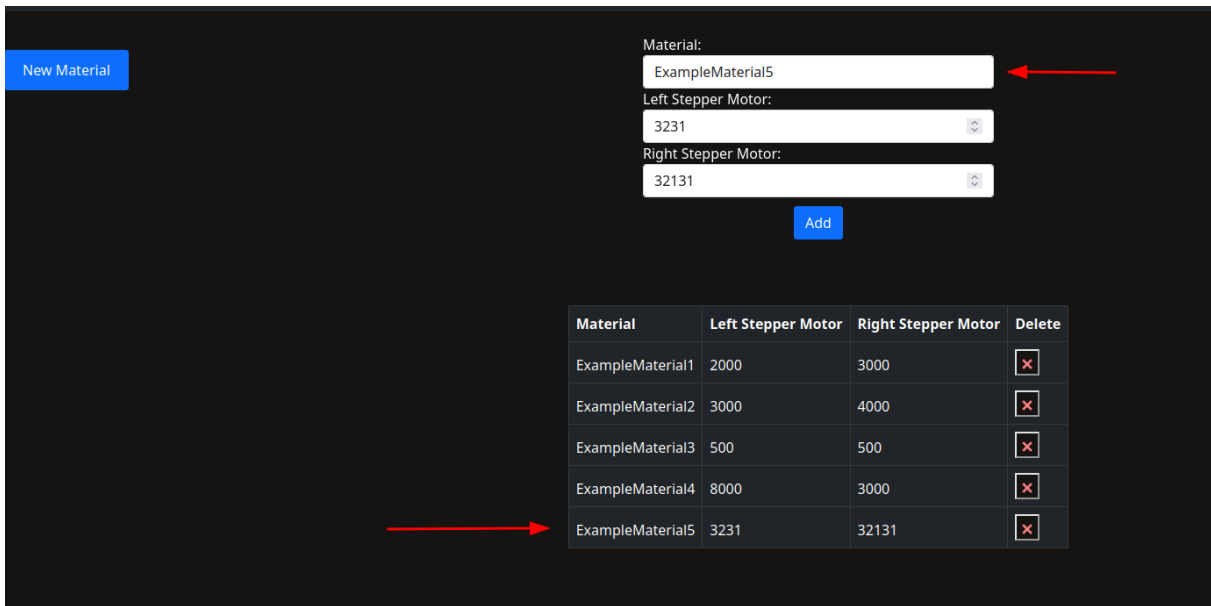
The screenshot shows a web interface for adding a new material. It features a 'New Material' button, a form with three input fields: 'Material' (containing 'ExampleMaterial1'), 'Left Stepper Motor' (containing '3231'), and 'Right Stepper Motor' (containing '32131'). An 'Add' button is below the form. A red error message 'Material Already Exists' is displayed. Below the form is a table of existing materials.

Material	Left Stepper Motor	Right Stepper Motor	Delete
ExampleMaterial1	2000	3000	<input type="checkbox"/>
ExampleMaterial2	3000	4000	<input type="checkbox"/>
ExampleMaterial3	500	500	<input type="checkbox"/>
ExampleMaterial4	8000	3000	<input type="checkbox"/>

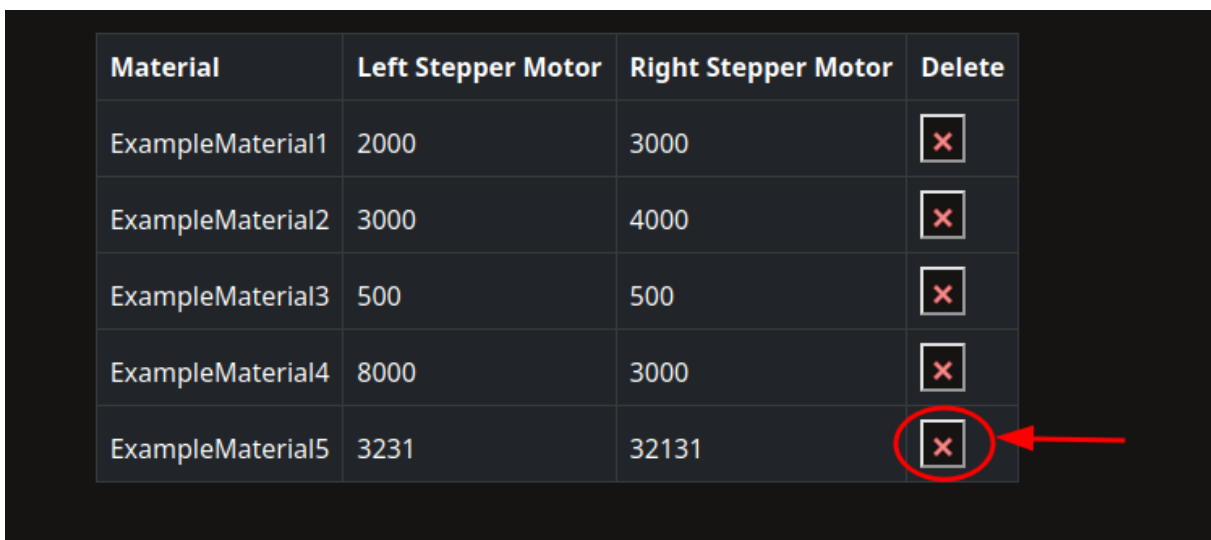
Σχήμα 4.40: Διπλοκαταχώρηση Μέτρησης

Εφόσον το όνομα του υλικού είναι μοναδικό, το επόμενο βήμα είναι η συμπλήρωση των απαραίτητων πληροφοριών και η επιβεβαίωση των λεπτομερειών. Τέλος, ο χρήστης υποβάλλει το νέο Material Profile, το οποίο προστίθεται στη βάση δεδομένων και εμφανίζεται στη λίστα των διαθέσιμων Material Profiles του χρήστη όπως φαίνεται και στο Σχήμα 4.41.

Η δυνατότητα διαγραφής ενός υπάρχοντος Material Profile παρέχεται επίσης στη σελίδα. Ο χρήστης μπορεί να επιλέξει το επιθυμητό προφίλ και να πατήσει το κουμπί Διαγραφή για να το αφαιρέσει από τη λίστα (Σχήμα 4.42).

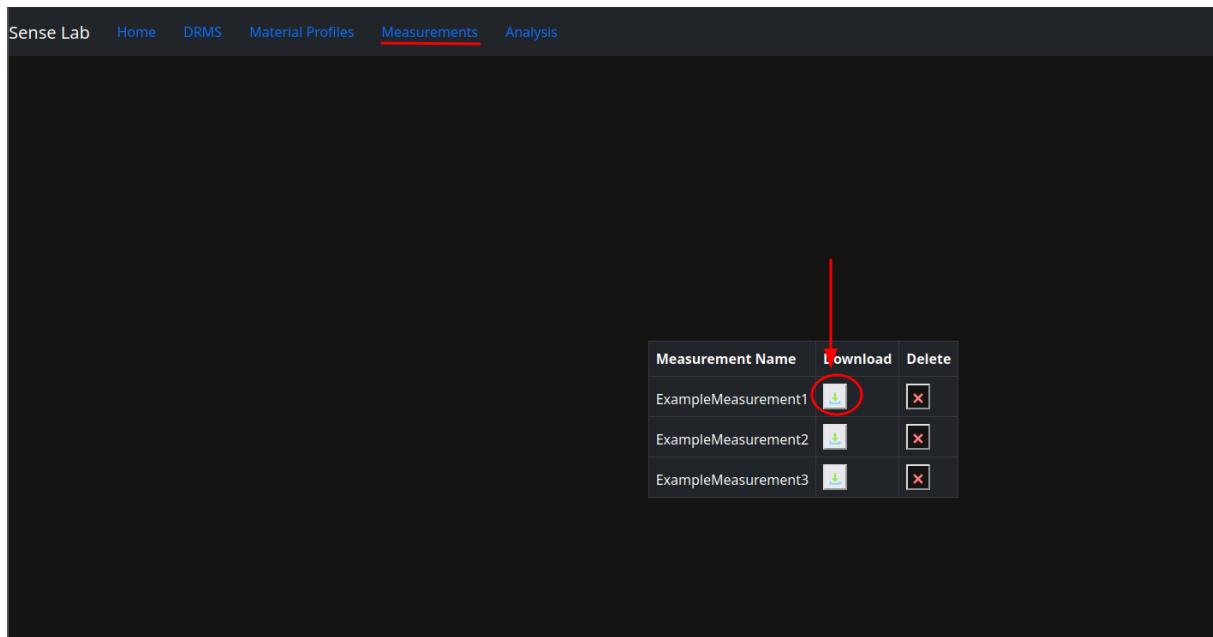


Σχήμα 4.41: Επιτυχημένη Εγγραφή Προφίλ Υλικού

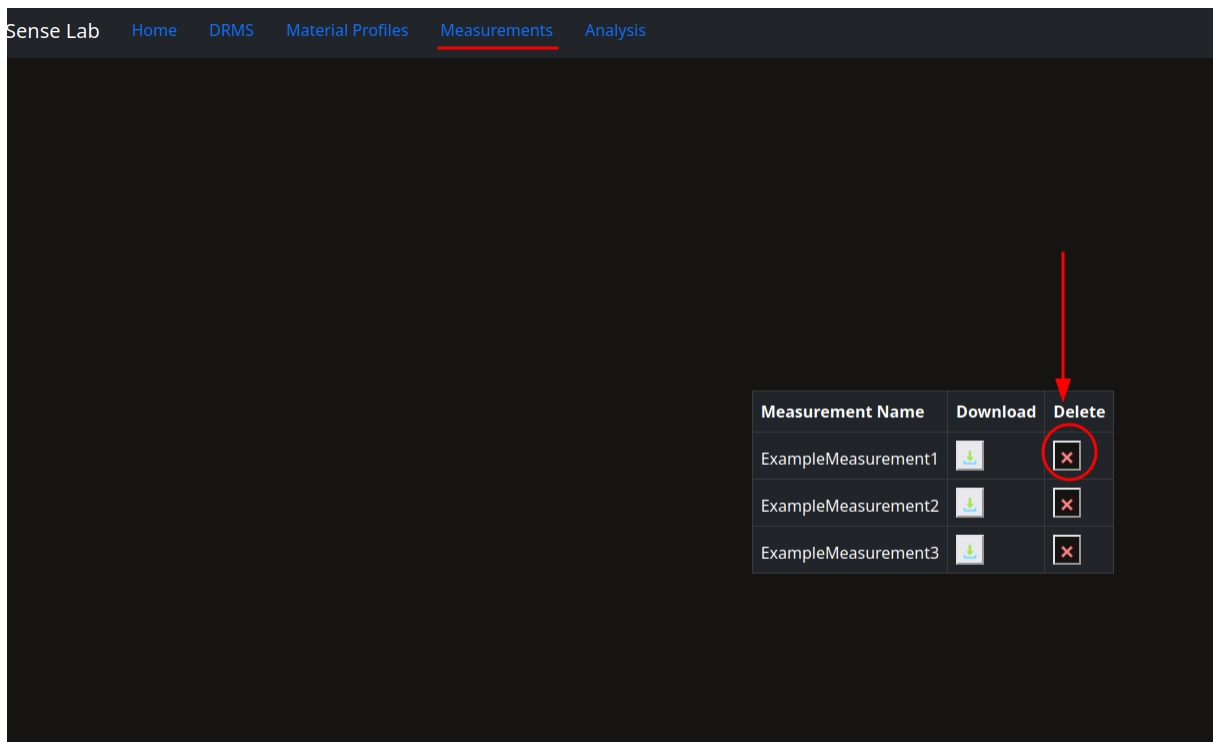


Σχήμα 4.42: Διαγραφή Προφίλ Υλικού

Measurements: Στην ενότητα Measurements, ο χρήστης έχει τη δυνατότητα να προβάλλει τις μετρήσεις που έχει πραγματοποιήσει, να λάβει κάποια μέτρηση ως αρχείο (Σχήμα 4.43)csv, αλλά και να διαγράψει κάποια μέτρηση (Σχήμα 4.44). Ο στόχος είναι η οργάνωση και η εύκολη επεξεργασία των δεδομένων.



Σχήμα 4.43: Λήψη Μέτρησης



Σχήμα 4.44: Διαγραφή Μέτρησης

4.6 Dockerization

Στο πλαίσιο της υλοποίησης, κρίθηκε απαραίτητη η χρήση της τεχνολογίας Docker για την απομόνωση και την αυτοματοποίηση των διάφορων στοιχείων του συστήματος. Κάθε στοιχείο του συστήματος όπως η βάση δεδομένων, ο Flask server και το web γραφικό περιβάλλον διαμορφώθηκαν ως ξεχωριστά Docker containers.

Οι εντολές κατασκευής των Docker εικόνων για τον Flask server αλλά και για το web-γραφικό περιβάλλον είναι οι ακόλουθες:

```
# Building the Docker image for the Flask application  
build -t server .  
# Building the Docker image for the Web application  
build -t drms .
```

Οι εντολές για την εκκίνηση του Flask container αλλά και του Web container είναι οι εξής:

```
# Run the Docker image for the Flask application  
docker run -it -p 5000:5000 name server server  
# Run the Docker image for the Web application  
docker run -it -p 5500:5500 name drms drms
```

Οι παράμετροι της εντολής περιγράφονται παρακάτω:

- **-it**: Αυτή η επιλογή εξασφαλίζει ότι η εντολή εκτελείται σε διαδραστική λειτουργία (interactive mode), δηλαδή οι χρήστες μπορούν να εισάγουν εντολές στο terminal του container.
- **-p 5000:5000**: Αυτή η επιλογή καθορίζει την ανακατεύθυνση των θυρών (port mapping). Εδώ, η θύρα 5000 του host συστήματος (το μηχάνημα στο οποίο τρέχει ο Docker daemon) ανακατευθύνεται στη θύρα 5000 του Docker container.
- **name server server**: Εδώ, το “name” αντιπροσωπεύει το όνομα του Docker container, ενώ το “server” που ακολουθεί είναι το όνομα της εικόνας Docker που θα χρησιμοποιηθεί για να δημιουργήσει το container.

Κεφάλαιο 5

Συμπεράσματα

Συνοψίζοντας τη διπλωματική εργασία, έχουμε εξετάσει και αναπτύξει ένα σύστημα που αποσκοπεί στην ευελιξία και βελτίωση της διαχείρισης του DRMS συστήματος. Κατά τη διάρκεια της εργασίας, αναπτύχθηκαν διάφορα στοιχεία και τεχνικές για την κατασκευή και τη βελτίωση του συστήματος.

Αρχικά, πραγματοποιήθηκε μια λεπτομερής ανάλυση των απαιτήσεων και των προδιαγραφών του συστήματος, καθορίζοντας τις βασικές λειτουργίες και την αρχιτεκτονική του. Στη συνέχεια, αναπτύχθηκαν τα διάφορα στοιχεία του συστήματος, όπως η βάση δεδομένων, το back-end και το front-end, χρησιμοποιώντας τεχνολογίες όπως Flask, SQLAlchemy, REST-API και JavaScript.

Η εφαρμογή Docker χρησιμοποιήθηκε για τη διαχείριση και την αυτοματοποίηση των επιμέρους στοιχείων του συστήματος, καθιστώντας τη διαδικασία ανάπτυξης, εγκατάστασης και αναβάθμισης πιο εύκολη και αποτελεσματική.

Το σύστημα παρέχει στους χρήστες τη δυνατότητα να βλέπουν σε πραγματικό χρόνο, να διαχειρίζονται, και να εξάγουν δεδομένα μετρήσεων με εύκολο και αποτελεσματικό τρόπο. Οι χρήστες μπορούν να οργανώνουν τις μετρήσεις τους, να κατεβάζουν αρχεία CSV με τα δεδομένα των μετρήσεων και να διαγράφουν μετρήσεις που δεν χρειάζονται πλέον. Αυτό συμβάλλει στη βελτίωση της οργάνωσης και της επεξεργασίας των δεδομένων, επιτρέποντας στους χρήστες να εστιάσουν στην ανάλυση και την ερμηνεία των αποτελεσμάτων.

Κατά τη διάρκεια της ανάπτυξης του συστήματος, χρησιμοποιήθηκαν εργαλεία όπως το Postman για τη δοκιμή και την αξιολόγηση των διαφόρων API endpoints. Το Postman αποδείχθηκε ιδιαίτερα χρήσιμο για την επικοινωνία με

το back-end, την επιβεβαίωση της σωστής λειτουργίας των διαφόρων λειτουργιών και την εύρεση και διόρθωση προβλημάτων.

Με την ολοκλήρωση αυτής της διπλωματικής εργασίας, καταφέραμε να αναπτύξουμε ένα καταναμημένο σύστημα που ανταποκρίνεται στις αρχικές απαιτήσεις και προδιαγραφές, προσφέροντας μια εύκολη στη χρήση και αποτελεσματική λύση για τη διαχείριση και την επίβλεψη του DRMS. Το σύστημα έχει τη δυνατότητα να επεκταθεί και να βελτιωθεί περαιτέρω, προσθέτοντας νέες λειτουργίες και βελτιώνοντας την απόδοση και τη χρηστικότητα του.

Συνολικά, η παρούσα διπλωματική εργασία αποτελεί μια επιτυχημένη προσπάθεια για την ανάπτυξη ενός συστήματος που στοχεύει στη βελτίωση της διαχείρισης του DRMS, παρέχοντας χρήσιμες λειτουργίες και εύχρηστες διεπαφές για τους χρήστες του.

5.1 Μελλοντικές εργασίες

Μελλοντικές εργασίες θα μπορούσαν να περιλαμβάνουν την προσθήκη δυνατοτήτων αυτόματης ανάλυσης και εξαγωγής αποτελεσμάτων, την ενσωμάτωση με άλλες πλατφόρμες και εργαλεία για την ανάλυση δεδομένων, καθώς και τη βελτίωση της ασφάλειας και της προστασίας των δεδομένων των χρηστών.

Βιβλιογραφία

- [1] What is RESTAPI <https://www.redhat.com/en/topics/api/what-is-a-rest-api>.
- [2] Flask web development: developing web applications with python.
- [3] Essential SQLAlchemy: Mapping Python to Databases.
- [4] Learning SQL: master SQL fundamentals.
- [5] Esp32 Features and Specifications [HTTP://esp32.net/](http://esp32.net/).
- [6] WiFi Manager with ESP32 [HTTPS://www.youtube.com/watch?v=Z_HWJ80Dh6s&t=3s](https://www.youtube.com/watch?v=Z_HWJ80Dh6s&t=3s).
- [7] What Are Distributed Systems? [HTTPS://www.splunk.com/en_us/data-insider/what-are-distributed-systems.html](https://www.splunk.com/en_us/data-insider/what-are-distributed-systems.html).
- [8] Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. Scotts Valley, CA: CreateSpace.
- [9] Jang, J. (2015). RESTful Web APIs with Flask: Build Powerful, Scalable, and Maintainable RESTful Web Services with Flask. New York, NY: Packt Publishing.
- [10] Bayer, M. (2019). SQLAlchemy: The Python SQL Toolkit and Object-Relational Mapper. [HTTPS://www.sqlalchemy.org/](https://www.sqlalchemy.org/).
- [11] Post, J. (2017). Flask-SQLAlchemy: An Extension That Adds SQLAlchemy Support to Your Flask Application. [HTTPS://flask-sqlalchemy.palletsprojects.com/](https://flask-sqlalchemy.palletsprojects.com/).

- [12] Kleppmann, M. (2017). Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. Sebastopol, CA: O'Reilly Media.
- [13] Asthana, A., & Scott, K. (2019). Postman: The Complete Guide. [HTTps://learning.postman.com/docs/getting-started/introduction/](https://learning.postman.com/docs/getting-started/introduction/)
- [14] Scott, K. (2020). Postman for API Development and Testing. New York, NY: Packt Publishing.
- [15] Miell, I., & Sayers, A. (2016). Docker Cookbook: Solutions and Examples for Building Distributed Applications. Sebastopol, CA: O'Reilly Media.
- [16] Javascript
<https://codeinstitute.net/global/blog/advantages-of-javascript/>.
- [17] Spurlock, J., Grant, K., & Kaden, J. (2019). Bootstrap 4: Responsive Web Design and Development. New York, NY: Apress.
- [18] Otwell, D. (2018). Bootstrap 4 By Example. New York, NY: Packt Publishing.
- [19] VsCode <https://code.visualstudio.com>.
- [20] Python Advantages
<https://techvidvan.com/tutorials/python-advantages-and-disadvantages>
- [21] What it is Pip
<https://realpython.com/what-is-pip/>
- [22] Advantages of Flask
<https://careerfoundry.com/en/blog/web-development/what-is-flask/#advantages-and-disadvantages-of-flask>