



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
«Πλατφόρμα Καταχώρησης Ιδεών για Hackathon με
Κοινωνική Συμμετοχή»



Του φοιτητή
Γερόπουλος Απόστολου
Αρ. Μητρώου: 174865

Επιβλέπων
Όνοματεπώνυμο: Μπράτσας
Χαράλαμπος.
Βαθμίδα: Επίκουρος Καθηγητής

Ημερομηνία 26/01/2025

Πλατφόρμα Καταχώρησης Ιδεών για Hackathon με Κοινωνική ΣυμμετοχήΚωδικός Δ.Ε. ...

Γερόπουλος Απόστολος

κ. Μπράτσας Χαράλαμπος

Ημερομηνία ανάληψης : 05-07-2024

Ημερομηνία περάτωσης Δ.Ε. : 26/01/2025

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Γερόπουλου Απόστολου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Αφιερώνω αυτήν τη διπλωματική εργασία, τον τελευταίο σταθμό για την απόκτηση του πτυχίου μου, στον αγαπημένο μου παππού, Γερόπουλο Απόστολο. Ήταν εκείνος που πάντα έλεγε: “Αυτό το παιδί θα προκόψει μια μέρα.” Δυστυχώς, έφυγε πριν προλάβει να με δει να πραγματοποιώ το όνειρό μου, αλλά η μνήμη και τα λόγια του αποτελούν για μένα αστείρευτη πηγή έμπνευσης και δύναμης.»

Πρόλογος

Ο τομέας της πληροφορικής αποτελεί πλέον έναν από τους πιο κρίσιμους παράγοντες στην ανάπτυξη και εξέλιξη όλων των επαγγελματικών τομέων. Σκοπός αυτής της τεχνολογικής προόδου είναι η βελτίωση της καθημερινότητας των ανθρώπων, διευκολύνοντας τις διαδικασίες και ενισχύοντας την αποτελεσματικότητα σε διάφορες δραστηριότητες. Σήμερα, η τεχνολογία έχει γίνει αναπόσπαστο κομμάτι της ζωής μας, επηρεάζοντας κάθε πτυχή της καθημερινότητας και προσφέροντας λύσεις σε πλήθος προβλημάτων που αντιμετωπίζουμε καθημερινά. Ο αριθμός των ατόμων που εξειδικεύονται στον τομέα της πληροφορικής συνεχώς αυξάνεται, ενώ ταυτόχρονα οι ειδικότητες γίνονται όλο και πιο διαφοροποιημένες και εξειδικευμένες. Η τεχνολογία, σήμερα, προσφέρει λύσεις σε προβλήματα που, αν και φαίνονται απλά στην καθημερινότητα, μπορεί να έχουν σημαντικό αντίκτυπο σε ατομικό ή κοινωνικό επίπεδο. Από τις πιο απλές ανάγκες ενός πολίτη μέχρι τα πιο σύνθετα ζητήματα που αφορούν δημόσιους φορείς, η πληροφορική έχει τη δυνατότητα να προσφέρει καινοτόμες λύσεις σε κοινωνικά, οικονομικά, στατιστικά, αλλά και σε πολλές άλλες μορφές προβλημάτων.

Για να επιτευχθεί, όμως, η επίλυση αυτών των προβλημάτων μέσω της πληροφορικής, απαιτείται η δημιουργία μιας γέφυρας επικοινωνίας μεταξύ των τεχνολόγων και των χρηστών, ώστε τα πραγματικά προβλήματα να αναγνωρίζονται και να αντιμετωπίζονται με στοχευμένες λύσεις. Στην περίπτωση που η επιστήμη της πληροφορικής λειτουργεί αποκομμένη από τις πραγματικές ανάγκες της κοινωνίας, ενδέχεται να οδηγηθούμε σε λύσεις που δεν έχουν πρακτική εφαρμογή ή δεν ανταποκρίνονται στις πραγματικές απαιτήσεις της καθημερινότητας. Επομένως, η συνεργασία μεταξύ των εμπλεκόμενων φορέων και η διαρκής σύνδεση της τεχνολογίας με τα προβλήματα που καλείται να επιλύσει είναι θεμελιώδης για την αποτελεσματική χρήση της πληροφορικής στην πράξη.

Περίληψη

Η παρούσα διπλωματική εργασία αφορά τη δημιουργία μιας πλατφόρμας που θα επιτρέπει στους πολίτες και δημόσιους φορείς να καταθέτουν ιδέες για εφαρμογές και πλατφόρμες, με στόχο την επίλυση προβλημάτων που αντιμετωπίζουν. Η πλατφόρμα επικεντρώνεται κυρίως σε περιβαλλοντικά ζητήματα και αναπτύσσεται στο πλαίσιο ενός hackathon που θα διοργανωθεί στη Θεσσαλονίκη. Σκοπός είναι να δίνεται η δυνατότητα σε κάθε ενδιαφερόμενο να καταθέτει τις ιδέες του, προσδιορίζοντας τα δεδομένα που απαιτούνται για την υλοποίηση των εφαρμογών, ενώ παράλληλα προγραμματιστές θα έχουν την ευκαιρία να αναλάβουν τις προτάσεις και να τις υλοποιήσουν.

Η πλατφόρμα θα σχεδιαστεί με τέτοιο τρόπο ώστε να είναι απλή και εύχρηστη, εξασφαλίζοντας την ευχέρεια τροποποίησης ακόμα και από άτομα με περιορισμένες γνώσεις στον τομέα της πληροφορικής. Η ανάπτυξή της θα πραγματοποιηθεί με την χρήση του Laravel, αξιοποιώντας templates βασισμένα στο Bootstrap για την κατασκευή του frontend και SQL για τη διαχείριση των βάσεων δεδομένων. Μέσα από την ανάλυση αυτών των τεχνολογιών, η εργασία εξετάζει τη διαδικασία σχεδίασης και υλοποίησης της πλατφόρμας, καθώς και τις δυνατότητες που παρέχει για την ενίσχυση της καινοτομίας και της συνεργασίας μεταξύ πολιτών, φορέων και προγραμματιστών στο πλαίσιο της επίλυσης σύγχρονων περιβαλλοντικών προβλημάτων.

Λέξεις κλειδιά: Web Development, Hackathon, Αποθετήριο Ιδεών, Laravel

«Idea Submission Platform for Hackathon with Social Participation»

«APOSTOLOS GEROPOULOS»

Abstract

Βεβαίως! Ορίστε μια άρτια και προσεγμένη μετάφραση του κειμένου σου:

This thesis focuses on the development of a platform that allows citizens and public entities to submit ideas for applications and platforms, aiming to address the problems they face. The platform primarily focuses on environmental issues and is being developed within the framework of a hackathon to be held in Thessaloniki. The objective is to provide every interested party with the opportunity to submit their ideas, specifying the data required for the implementation of these applications, while developers will have the chance to take on these proposals and bring them to life.

The platform will be designed in such a way that it is simple and user-friendly, ensuring ease of modification even by individuals with limited knowledge in the field of information technology. Its development will be carried out using Laravel, utilizing Bootstrap-based templates for building the frontend, and SQL for database management. Through the analysis of these technologies, the thesis examines the design and implementation process of the platform, as well as the possibilities it offers for enhancing innovation and collaboration among citizens, entities, and developers in the context of addressing modern environmental challenges.

Keywords: Web Development, Hackathon, Idea Repository, Laravel

Ευχαριστίες

«Θα ήθελα να εκφράσω τις θερμότερες ευχαριστίες μου στους καθηγητές μου, για την πολύτιμη καθοδήγηση και υποστήριξή τους καθ' όλη τη διάρκεια της ακαδημαϊκής μου πορείας. Επίσης, ευχαριστώ την οικογένειά μου, που στάθηκε δίπλα μου με αγάπη και αδιάκοπη στήριξη σε κάθε μου βήμα, καθώς και τους φίλους μου, για την ενθάρρυνση και τη συμπαράστασή τους στις στιγμές που τη χρειαζόμουν περισσότερο.»

Περιεχόμενα

Πρόλογος.....	v
Περίληψη.....	vi
Abstract.....	vii
Ευχαριστίες.....	viii
Περιεχόμενα	ix
Κατάλογος Σχημάτων.....	xii
Κατάλογος Πινάκων.....	Error! Bookmark not defined.
Συνομογραφίες.....	xiii
Κεφάλαιο 1ο: Εισαγωγή.....	1
Κεφάλαιο 2ο: Hackathons και οι τεχνολογίες τους.....	3
2.1 Εισαγωγή.....	3
2.2 Υφιστάμενες Πλατφόρμες Hackathon: Ανάλυση και Σύγκριση.....	3
2.2.1 Devpost.....	3
2.2.2 Hackerearth.....	4
2.2.3 IdeaScale.....	4
2.2.4 GitHub.....	4
2.2.5 Slack.....	4
2.2.6 Figma.....	4
2.2.7 Trello.....	4
2.2.8 Miro.....	4
2.2.9 Airtable.....	5
2.2.10 OpenIDEO.....	5
2.2.11 Kaggle.....	5
2.2.12 HackMD.....	5
2.2.13 GitHub.....	5
2.2.14 ChallengePost.....	5
2.2.15 HackClub.....	6
2.2.16 LeetCode.....	6
2.2.17 Major League Hacking (MLH).....	6
2.3 Επίλογος.....	6
Κεφάλαιο 3ο: Διαμόρφωση και Εγκατάσταση Περιβάλλοντος Ανάπτυξης για την Υλοποίηση της Εφαρμογής.....	8

3.1	Εισαγωγή	8
3.2	Xampp	8
3.2.1	Τι περιλαμβάνει το XAMPP;	8
3.2.2	Πώς λειτουργεί;.....	9
3.2.3	Αναλυτική περιγραφή του Apache;	9
3.2.4	Αναλυτική περιγραφή του MySQL;	10
3.2.5	Πώς συνεργάζονται;.....	10
3.3	Composer.....	11
3.3.1	Λειτουργία του Composer	11
3.3.2	Η επιλογή του Composer	11
3.4	Visual Studio Code (VS Code).....	12
3.4.1	Τι είναι το VS Code;.....	12
3.4.2	Η επιλογή του VS Code.....	13
3.4.3	Visual Studio Code σαν καλύτερη επιλογή	13
3.4.4	Visual Studio Code Extensions.....	14
3.5	Επίλογος	16
Κεφάλαιο 4ο:	Αρχιτεκτονική Συστήματος.....	18
4.1	Εισαγωγή	18
4.2	Idea Lists UML.....	19
4.3	Use Case Diagram (χρηστών).....	20
4.4	Sequence Diagrams	21
4.4.1	Ανάλυση για το Admin User Sequence Diagram.....	21
4.4.2	Ανάλυση για το Guest User Sequence Diagram	24
4.4.3	Ανάλυση για το Registered User Sequence Diagram.....	25
4.5	Επίλογος	26
Κεφάλαιο 5ο:	Ανάπτυξη εφαρμογής.....	27
5.1	Εισαγωγή	27
5.2	Βάσεις Δεδομένων.....	27
5.2.1	Ανάλυση πίνακα Cache.....	28
5.2.2	Ανάλυση πίνακα Cache locks	29
5.2.3	Ανάλυση πίνακα Idea_tags	29
5.2.4	Ανάλυση πίνακα Ideas	30
5.2.5	Migrations και Ανάλυση Πίνακα migrations	31
5.2.6	Ανάλυση πίνακα Users.....	33
5.2.7	Ανάλυση πίνακα tags	34

5.2.8	Ανάλυση πίνακα password_resets.....	34
5.2.9	Ανάλυση πίνακα sessions.....	35
5.3	Σύνδεση βάση δεδομένων με εφαρμογή.....	35
5.3.1	Εισαγωγή.....	35
5.3.2	.env.....	35
5.3.3	Database.php.....	36
5.3.4	Migrations.....	39
5.3.5	Controllers.....	39
5.3.6	Models.....	40
5.3.7	API.....	41
5.4	Blades.....	44
5.4.1	Τι είναι τα Blades.....	44
5.4.2	Πως λειτουργεί το Blade.....	44
5.4.3	Σε τι χρησιμεύει το Blade.....	44
5.4.4	Γιατί να επιλέξουμε τα Blades.....	44
5.4.5	Παραδείγμα.....	45
5.4.6	Επίλογος.....	47
Κεφάλαιο 6ο:	Case study.....	47
6.1	Εισαγωγή.....	47
6.2	Home Page.....	48
6.3	Register/Log in.....	50
6.4	Dashboard.....	51
6.5	Idea Post.....	54
6.6	Idea Listing.....	54
6.7	Single Idea.....	56
Κεφάλαιο 7ο:	Προτάσεις Βελτίωσης και Επέκτασης της Πλατφόρμας Hackathon.....	57
7.1	Chat.....	57
7.2	Κατοχύρωση Ιδεών και Παρακολούθηση Υλοποίησης.....	58
7.3	Δημιουργία Υπερπλατφόρμας για Ανάπτυξη Εξατομικευμένων Πλατφορμών.....	58
7.4	Δημιουργία chat bot.....	59
7.5	Ενσωμάτωση Τεχνητής Νοημοσύνης για Έγκριση και Δημιουργία Ιδεών.....	60
7.6	Πιστοποίηση Χρήστη και Σύστημα Κατάταξης.....	61
7.7	Επίλογος.....	63
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		64
ΠΑΡΑΡΤΗΜΑ Α : ΤΙΤΛΟΣ ΠΑΡΑΡΤΗΜΑΤΟΣ.....		66

Κατάλογος Σχημάτων

Εικόνα 1 : Ο διαγωνισμός (hackathon) που θα χρησιμοποιηθεί η εφαρμογή

Εικόνα 2: Guest Sequence Diagram

Εικόνα 3: Registered User Sequence Diagram

Εικόνα 4: Στιγμιότυπο Βάσης Δεδομένων

Εικόνα 5: Στιγμιότυπο Πίνακα Cache

Εικόνα 6: Στιγμιότυπο Πίνακα Cache

Εικόνα 7: Στιγμιότυπο Πίνακα Idea Tags

Εικόνα 8: Στιγμιότυπο Πίνακα migrations

Εικόνα 9: Στιγμιότυπο Πίνακα Users

Εικόνα 10: Στιγμιότυπο Πίνακα tags

Εικόνα 11: Στιγμιότυπο Πίνακα password_resets

Εικόνα 12: Στιγμιότυπο Πίνακα sessions

Εικόνα 13: Στιγμιότυπο κώδικα από αρχείο .env

Εικόνα 14: Στιγμιότυπο κώδικα Database.php

Εικόνα 15: Στιγμιότυπο κώδικα από migrations

Εικόνα 16: Στιγμιότυπο κώδικα από Controller

Εικόνα 17: Στιγμιότυπο κώδικα από routes

Εικόνα 18: Στιγμιότυπο κώδικα από api users

Εικόνα 19: Στιγμιότυπο κώδικα Blade

Εικόνα 20: Στιγμιότυπο από Homepage 1

Εικόνα 21: Στιγμιότυπα από Homepage 2

Εικόνα 22 : Στιγμιότυπο από Register

Εικόνα 23: Στιγμιότυπο από Log in

Εικόνα 24: Στιγμιότυπο από Dashboard 1

Εικόνα 25: Στιγμιότυπο από Dashboard 2

Εικόνα 26 : Στιγμιότυπο από Idea Listing

Εικόνα 27: Στιγμιότυπο από Idea Listing

Εικόνα 28: Στιγμιότυπο από Single Idea

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία

Κεφάλαιο 1ο: Εισαγωγή

Η ραγδαία τεχνολογική ανάπτυξη των τελευταίων δεκαετιών έχει επηρεάσει βαθύτατα τον τρόπο με τον οποίο οι κοινωνίες αντιμετωπίζουν καθημερινά προβλήματα, ειδικά στις πτυχές που αφορούν το περιβάλλον και τη βιώσιμη ανάπτυξη. Στη σύγχρονη εποχή, η καινοτομία και η συνεργασία αποτελούν καταλυτικά εργαλεία για την αναζήτηση λύσεων σε προβλήματα που αποτελούν προκλήσεις για τις τοπικές κοινωνίες, αλλά και για την ανθρωπότητα συνολικά.

Στο πλαίσιο αυτό, η συγκεκριμένη διπλωματική εργασία αποσκοπεί στην ανάπτυξη μιας διαδικτυακής πλατφόρμας, η οποία θα επιτρέπει στους πολίτες και στους δημόσιους φορείς να καταθέτουν ιδέες για εφαρμογές και πλατφόρμες που στοχεύουν στην αντιμετώπιση συγκεκριμένων προβλημάτων. Ιδιαίτερη έμφαση δίνεται σε περιβαλλοντικά ζητήματα, τα οποία απαιτούν άμεσες και αποτελεσματικές παρεμβάσεις.

Η πλατφόρμα αυτή στηρίζεται σε ένα πλαίσιο ανοιχτής και συμμετοχικής καινοτομίας, όπως αυτές εκφράζονται μέσα από δραστηριότητες όπως τα hackathons. Τα hackathons είναι διοργανώσεις στις οποίες προγραμματιστές, σχεδιαστές και λοιποί ειδικοί συναντιούνται για να δημιουργήσουν καινοτόμες λύσεις για συγκεκριμένα προβλήματα, συνδυάζοντας γνώση, δεξιότητες και δημιουργικότητα [1].

Η σημασία των Hackathons και Ο ρόλος τους στη συνεργασία και καινοτομία

Η συμμετοχή σε hackathons ανοικτών δεδομένων και άλλες προκλήσεις ψηφιακής καινοτομίας συνεχίζει να αυξάνεται. Οι διαγωνισμοί αυτοί έχουν γίνει δημοφιλείς και συμβάλλουν στην ενίσχυση της ανοικτής καινοτομίας και της ψηφιακής επιχειρηματικότητας [2][3]. Ένας από τους κύριους στόχους τους είναι η ανάπτυξη υπηρεσιών που βασίζονται σε ανοικτά δεδομένα, οι οποίες αυξάνουν την προστιθέμενη αξία για την κοινωνία και τους δημόσιους φορείς [4][5].

Παρότι τα hackathons αποτελούν έναν ταχύ και αποτελεσματικό τρόπο για τη δημιουργία νέων ιδεών και πρωτοτύπων, μόνο ένας περιορισμένος αριθμός από αυτά τα πρωτότυπα καταλήγει στην αγορά και χρησιμοποιείται από τελικούς χρήστες [6]. Επιπλέον, οι οργανωτές hackathons προωθούν την ενδοεπιχειρηματικότητα, ενθαρρύνοντας τους υπαλλήλους να συμμετέχουν σε διαγωνισμούς και να προτείνουν βελτιώσεις σε υφιστάμενες υπηρεσίες [7].

Ο ρόλος της πλατφόρμας και οι τεχνολογικές της υποδομές

Η πλατφόρμα που προτείνεται θα λειτουργήσει ως αποθετήριο συγκέντρωσης ιδεών και προτάσεων, οι οποίες θα κατατεθούν στο πλαίσιο του hackathon που θα διοργανωθεί στη Θεσσαλονίκη υπό την αιγίδα της Open Knowledge Greece Foundation [8].

Η ανάπτυξη της πλατφόρμας βασίζεται στο Laravel, ένα δημοφιλές PHP framework, και στο Bootstrap για το frontend, με τη χρήση SQL για τη διαχείριση δεδομένων. Οι επιλογές αυτές εξασφαλίζουν την αποδοτικότητα, τη συντήρηση και την επεκτασιμότητα του συστήματος.

Συμπεράσματα και προοπτικές

Η συμμετοχή πολιτών και δημόσιων φορέων στην πλατφόρμα μπορεί να οδηγήσει στη δημιουργία εφαρμογών που συμβάλλουν στην αντιμετώπιση περιβαλλοντικών προβλημάτων. Για παράδειγμα, προτάσεις μπορούν να περιλαμβάνουν εφαρμογές για τη μέτρηση της ποιότητας του αέρα, τη διαχείριση απορριμμάτων ή την εξοικονόμηση ενέργειας.

Επιπλέον, η πλατφόρμα φιλοδοξεί να ενισχύσει τη συνεργασία και τη διάχυση της καινοτομίας σε τοπικό και εθνικό επίπεδο, ενώ η διοργάνωση hackathons προάγει τη χρήση τεχνολογίας για τη δημιουργία λύσεων που ανταποκρίνονται στις σύγχρονες προκλήσεις και προωθούν τη βιώσιμη ανάπτυξη [9][10].



Εικόνα 1.1 Ο διαγωνισμός (hackathon) που θα χρησιμοποιηθεί η εφαρμογή

Κεφάλαιο 2ο: Hackathons και ΟΙ τεχνολογίες τους

2.1 Εισαγωγή

Τα hackathons έχουν καταστεί θεμελιώδη γεγονότα στον τομέα της τεχνολογίας και της καινοτομίας. Πρόκειται για εκδηλώσεις διάρκειας 24 έως 48 ωρών, στις οποίες ομάδες προγραμματιστών, σχεδιαστών, επιχειρηματιών και άλλων επαγγελματιών εργάζονται συνεργατικά για να αναπτύξουν καινοτόμες λύσεις που αντιμετωπίζουν πραγματικά προβλήματα. Παρά την έντονη ανταγωνιστικότητα, το βασικό χαρακτηριστικό των hackathons είναι η συνεργασία και η ταχύτητα, που επιτρέπουν στους συμμετέχοντες να αξιοποιήσουν το δημιουργικό τους δυναμικό και να αναπτύξουν νέα προϊόντα ή εφαρμογές [10].

Ιστορικά, τα hackathons ξεκίνησαν ως ανεπίσημες συναντήσεις για την ανάπτυξη λογισμικού ή υλικού μέσα σε περιορισμένο χρονικό διάστημα. Ο όρος "hackathon" προέρχεται από τη σύνθεση των λέξεων "hack" και "marathon" και εμφανίστηκε στα τέλη της δεκαετίας του 1990. Η πρώτη hackathon πραγματοποιήθηκε από την κοινότητα του OpenBSD το 1999 στον Καναδά, ενώ μια διαφορετική άποψη αναφέρει την πρώτη σχετική εκδήλωση στο συνέδριο JavaOne την ίδια χρονιά, με στόχο συγκεκριμένες προγραμματιστικές προκλήσεις [11].

Με την πάροδο του χρόνου, τα hackathons εξελίχθηκαν ώστε να περιλαμβάνουν ευρύτερες κατηγορίες εκδηλώσεων που δεν περιορίζονται μόνο σε τεχνικά ζητήματα, αλλά ενσωματώνουν και μη τεχνικές προσεγγίσεις. Επίσης, μπορούν να λάβουν διάφορες μορφές, όπως φυσικές ή εικονικές, ανάλογα με τις ανάγκες της εκάστοτε διοργάνωσης. Παρά τη μεγάλη εξάπλωσή τους, η εις βάθος κατανόηση του φαινομένου παραμένει περιορισμένη. Οι προγενέστερες μελέτες επισημαίνουν ότι τα hackathons μπορούν να λειτουργήσουν ως μέσο για τη δομή της μάθησης, την οργάνωση διαδικασιών και την ενίσχυση της συμμετοχής, συνδέοντας τις σχεδιαστικές τους πτυχές με βιώσιμα αποτελέσματα.

Η σημασία των hackathons είναι ανεκτίμητη, καθώς αποτελούν ένα κύριο εργαλείο για την ενθάρρυνση της καινοτομίας, την επίλυση κοινωνικών και τεχνολογικών προβλημάτων, αλλά και την προώθηση της συνεργασίας. Σε αυτές τις εκδηλώσεις, αξιοποιούνται τεχνολογίες όπως APIs, εργαλεία ανάπτυξης λογισμικού και συνεργατικές πλατφόρμες, που παίζουν καθοριστικό ρόλο στην επιτυχία τους [11]. Παράλληλα, οι φαινόμενα που συνδυάζουν ανθρώπινη συμμετοχή και τεχνολογία, όπως τα hackathons, απαιτούν μια διεπιστημονική προσέγγιση για την πληρέστερη κατανόησή τους.

2.2 Υφιστάμενες Πλατφόρμες Hackathon: Ανάλυση και Σύγκριση

Υπάρχουν πολλές πλατφόρμες που χρησιμοποιούνται για hackathons και συνεργατικά έργα, οι οποίες προσφέρουν διαφορετικές δυνατότητες, από τη διαχείριση των συμμετοχών μέχρι την ανάπτυξη εφαρμογών και τη συνεργασία μεταξύ των συμμετεχόντων. Οι πιο γνωστές είναι οι παρακάτω:

2.2.1 Devpost

Η **Devpost** είναι μία από τις πιο δημοφιλείς πλατφόρμες για hackathons, όπου οι συμμετέχοντες καταθέτουν τα έργα τους, ανταγωνίζονται για βραβεία και προβάλλουν τις δημιουργίες τους. Η πλατφόρμα παρέχει τη δυνατότητα στους χρήστες να ανεβάσουν την πρόοδό τους, να μοιραστούν τον

κώδικα και να συνεργαστούν με άλλους. Ενώ η Devpost είναι εξαιρετική για την παρουσίαση και την υποβολή ολοκληρωμένων έργων, δεν παρέχει τις ίδιες δυνατότητες για την υποβολή και την επεξεργασία ιδεών σε πραγματικό χρόνο από τους χρήστες, όπως η πλατφόρμα που αναπτύσσουμε.

2.2.2 Hackerearth

Η **Hackerearth** είναι μία πλατφόρμα που συνδυάζει τη διεξαγωγή hackathons με την εκπαίδευση και την αξιολόγηση δεξιοτήτων προγραμματιστών. Ενώ είναι εξαιρετική για την εκπαίδευση και την ανάπτυξη δεξιοτήτων, η Hackerearth δεν επικεντρώνεται τόσο στην καταγραφή και την επεξεργασία των αρχικών ιδεών που κατατίθενται από το κοινό, όπως συμβαίνει στην πλατφόρμα που σχεδιάζουμε.

2.2.3 IdeaScale

Η **IdeaScale** είναι μια πλατφόρμα ιδεών που επιτρέπει στους χρήστες να υποβάλλουν και να αξιολογούν ιδέες για νέες εφαρμογές ή καινοτόμες λύσεις. Αν και παρέχει τη δυνατότητα για υποβολή και διαχείριση ιδεών, δεν εστιάζει στον προγραμματισμό ή την ανάπτυξη αυτών των ιδεών σε εφαρμογές, ενώ η δική μας πλατφόρμα θα συνδυάζει την κατάθεση ιδεών με την ενεργή υλοποίηση από προγραμματιστές.

2.2.4 GitHub

Το **GitHub** είναι η πλατφόρμα που χρησιμοποιείται ευρέως για τη διαχείριση του κώδικα και τη συνεργασία στην ανάπτυξη λογισμικού. Αν και εξαιρετική για τον κώδικα, δεν παρέχει την ίδια ευχέρεια για την οργάνωση hackathons ή τη συλλογή και διαχείριση των αρχικών ιδεών από το κοινό, κάτι που είναι κεντρικό στη δική μας πλατφόρμα.

2.2.5 Slack

Το **Slack** είναι μια πλατφόρμα επικοινωνίας που χρησιμοποιείται ευρέως για την οργάνωση ομάδων και τη διαχείριση έργων. Παρά τη μεγάλη χρησιμότητά του για την επικοινωνία σε πραγματικό χρόνο, δεν παρέχει τη δυνατότητα για την υποβολή ιδεών και την ολοκληρωμένη διαχείριση hackathons, όπως η πλατφόρμα που προτείνουμε.

2.2.6 Figma

Το **Figma** είναι ένα εργαλείο σχεδίασης που επιτρέπει την κοινή εργασία σε σχέδια και πρωτότυπα. Παρόλο που είναι εξαιρετικό για τη συνεργασία στην ανάπτυξη UI/UX, δεν παρέχει τη δυνατότητα για την καταγραφή ιδεών και τη δημιουργία λύσεων από τον χρήστη προς την κατεύθυνση ανάπτυξης λογισμικού, όπως η δική μας πλατφόρμα.

2.2.7 Trello

Το **Trello** είναι ένα εργαλείο διαχείρισης έργων που επιτρέπει στους χρήστες να οργανώνουν και να παρακολουθούν την πρόοδο των εργασιών τους. Αν και χρήσιμο για τη διαχείριση του έργου, δεν είναι κατάλληλο για την υποβολή και ανάπτυξη ιδεών, ενώ η πλατφόρμα μας ενσωματώνει την υποβολή ιδεών και τη δυνατότητα ανάπτυξής τους σε πραγματικό χρόνο.

2.2.8 Miro

Το **Miro** είναι ένα εργαλείο για δημιουργία διαγραμμάτων και brainstorming, που βοηθά στην οργάνωση ιδεών. Ενώ είναι εξαιρετικό για οπτική παρουσίαση και συνεργασία, δεν έχει τη δυνατότητα

υποβολής ιδεών και την ενσωμάτωση τους με την ανάπτυξη εφαρμογών και την αξιολόγηση τους σε hackathon, όπως προσφέρει η πλατφόρμα μας.

2.2.9 Airtable

Το **Airtable** είναι μια πλατφόρμα για τη δημιουργία βάσεων δεδομένων που μπορούν να χρησιμοποιηθούν για τη διαχείριση έργων και την καταγραφή ιδεών. Παρόλο που μπορεί να χρησιμοποιηθεί για την αποθήκευση και τη διαχείριση δεδομένων, δεν προσφέρει την ολοκληρωμένη υποστήριξη για τη δημιουργία και την υλοποίηση ιδεών από πολίτες και φορείς, όπως η δική μας πλατφόρμα.

2.2.10 OpenIDEO

Η **OpenIDEO** είναι μια πλατφόρμα καινοτομίας που συνδυάζει τη συλλογή ιδεών με την επίλυση κοινωνικών προβλημάτων. Αν και επικεντρώνεται στην κοινωνική καινοτομία, δεν εστιάζει στην υλοποίηση των ιδεών μέσω hackathons με τη συμμετοχή προγραμματιστών και τη διαχείριση δεδομένων σε πραγματικό χρόνο, όπως η δική μας πλατφόρμα.

2.2.11 Kaggle

Το **Kaggle** είναι μια πλατφόρμα που επικεντρώνεται στη διεξαγωγή διαγωνισμών δεδομένων και ανάλυσης δεδομένων. Αν και εξαιρετική για τη διαχείριση διαγωνισμών σε σχέση με τα δεδομένα, δεν παρέχει τις ίδιες δυνατότητες για την υποβολή ιδεών και την οργάνωση hackathons όπως η πλατφόρμα μας.

2.2.12 HackMD

Το **HackMD** είναι μια πλατφόρμα συνεργασίας για σημειώσεις και έγγραφα σε πραγματικό χρόνο. Παρόλο που είναι χρήσιμη για συνεργασία, δεν καλύπτει τη διαχείριση hackathons ή την υποβολή ιδεών για ανάπτυξη εφαρμογών όπως η πλατφόρμα μας.

2.2.13 GitHub

Το **GitHub** είναι η πλατφόρμα που χρησιμοποιείται ευρέως για τη διαχείριση του κώδικα και τη συνεργασία στην ανάπτυξη λογισμικού. Αν και εξαιρετική για τον κώδικα, δεν παρέχει την ίδια ευχέρεια για την οργάνωση hackathons ή τη συλλογή και διαχείριση των αρχικών ιδεών από το κοινό, κάτι που είναι κεντρικό στη δική μας πλατφόρμα.

2.2.14 ChallengePost

Το **ChallengePost** είναι μια πλατφόρμα που επιτρέπει τη διοργάνωση διαγωνισμών, προσφέροντας εργαλεία για τη δημοσίευση και τη συμμετοχή σε hackathons. Αν και είναι πολύ χρήσιμη για τη διαχείριση διαγωνισμών και την παρουσίαση των έργων, δεν παρέχει την ίδια δυνατότητα για την καταγραφή και υλοποίηση ιδεών σε συνεργασία με δημόσιους φορείς ή πολίτες. Επίσης, η έμφαση δίνεται κυρίως στην ολοκλήρωση των έργων και όχι στην ανάπτυξη ή την καταγραφή των προβλημάτων από την αρχή. Η πλατφόρμα μας επικεντρώνεται στην καταγραφή προβλημάτων και την άμεση συνεργασία για την ανάπτυξή τους, κάτι που δεν παρέχει το ChallengePost..

2.2.15 HackClub

Το **HackClub** είναι μια πλατφόρμα που υποστηρίζει και οργανώνει hackathons και άλλες δράσεις για μαθητές και νέους. Παρόλο που παρέχει καλές δυνατότητες για εκπαίδευση και υποστήριξη της κοινότητας, δεν έχει τη δυνατότητα για υποβολή και υλοποίηση ιδεών από δημόσιους φορείς ή πολίτες. Επίσης, δεν παρέχει τα εργαλεία για να συνδεθούν οι συμμετέχοντες με φορείς που θα μπορούσαν να χρησιμοποιήσουν τις λύσεις που αναπτύσσονται. Στη δική μας πλατφόρμα, παρέχεται η δυνατότητα στους χρήστες να καταθέτουν προβλήματα και να συνεργάζονται άμεσα με προγραμματιστές για την ανάπτυξη βιώσιμων λύσεων.

2.2.16 LeetCode

Το **LeetCode** είναι μια πλατφόρμα για την εκπαίδευση στον προγραμματισμό και την επίλυση αλγοριθμικών προβλημάτων. Παρόλο που η πλατφόρμα είναι ιδανική για προγραμματιστές που θέλουν να βελτιώσουν τις δεξιότητές τους, δεν παρέχει δυνατότητες για την υποβολή και συνεργασία σε πραγματικά κοινωνικά ή περιβαλλοντικά προβλήματα. Η δική μας πλατφόρμα, αντίθετα, επικεντρώνεται στην υποβολή και επίλυση προβλημάτων με άμεση αλληλεπίδραση από την κοινότητα και τους δημόσιους φορείς, επιτρέποντας την ανάπτυξη εφαρμογών που απευθύνονται σε πραγματικές ανάγκες.

2.2.17 Major League Hacking (MLH)

Η **Major League Hacking (MLH)** είναι μια κοινότητα hackathon που οργανώνει διαγωνισμούς για φοιτητές, με στόχο την ανάπτυξη δεξιοτήτων στον προγραμματισμό και την καινοτομία. Παρόλο που η MLH είναι εξαιρετική για εκπαιδευτικούς διαγωνισμούς και για τη δημιουργία κοινότητας, δεν παρέχει έναν τρόπο για τις τοπικές κοινότητες ή δημόσιους φορείς να καταθέτουν προβλήματα και να συνεργάζονται με τους προγραμματιστές για την ανάπτυξή τους. Αντιθέτως, η πλατφόρμα μας επιτρέπει την καταγραφή κοινωνικών ή περιβαλλοντικών προβλημάτων από τους φορείς και την άμεση συμμετοχή των προγραμματιστών στην υλοποίησή τους.

2.3 Επίλογος

Αν και υπάρχουν πολλές πλατφόρμες που υποστηρίζουν hackathons και διαδικασίες συλλογής καινοτόμων ιδεών, καμία δεν καλύπτει πλήρως τη λειτουργικότητα και τον σκοπό της πλατφόρμας που αναπτύξαμε. Οι υφιστάμενες λύσεις είτε επικεντρώνονται αποκλειστικά στη διαχείριση διαγωνισμών και την ανάπτυξη λογισμικού, είτε διευκολύνουν τη συλλογή ιδεών χωρίς να εστιάζουν στην υλοποίησή τους.

Η πλατφόρμα μας διαφέρει ουσιαστικά καθώς επικεντρώνεται στην άμεση γεφύρωση του χάσματος ανάμεσα στα πραγματικά κοινωνικά προβλήματα και τις τεχνολογικές λύσεις που μπορούν να τα αντιμετωπίσουν. Επιτρέπει σε πολίτες και φορείς να καταθέτουν προβλήματα και ανάγκες, τα οποία γίνονται άμεσα προσβάσιμα σε προγραμματιστές και ειδικούς της πληροφορικής για συνεργασία και ανάπτυξη βιώσιμων λύσεων.

Ένα ακόμα χαρακτηριστικό που καθιστά την πλατφόρμα μοναδική είναι η ευελιξία της στη διαμόρφωση και προσαρμογή ανάλογα με τη θεματολογία και τη φύση των προβλημάτων που στοχεύει να επιλύσει. Χάρη στον ευέλικτο σχεδιασμό της, η πλατφόρμα μπορεί να προσαρμόζεται εύκολα για εξειδικευμένες ανάγκες. Για παράδειγμα, στην τρέχουσα υλοποίησή της για το Open Up Hackathon, η πλατφόρμα έχει διαμορφωθεί ώστε να εστιάζει σε περιβαλλοντικά δεδομένα, ανταποκρινόμενη στις απαιτήσεις του συγκεκριμένου θέματος.

Αυτή η δυνατότητα τροποποίησης ενισχύει περαιτέρω την αξία της πλατφόρμας, καθώς μπορεί να προσαρμόζεται σε διαφορετικά πλαίσια, διευκολύνοντας την επίλυση προβλημάτων με στοχευμένο και αποτελεσματικό τρόπο. Έτσι, η πλατφόρμα που αναπτύξαμε δεν αποτελεί μόνο μια τεχνολογική λύση αλλά και ένα σημαντικό κοινωνικό εργαλείο που προάγει τη συνεργασία, τη δημιουργικότητα και την καινοτομία, παραμένοντας ταυτόχρονα ευέλικτο και προσαρμοστικό στις ανάγκες κάθε hackathon και κοινωνικής πρόκλησης..

Κεφάλαιο 3ο: Διαμόρφωση και Εγκατάσταση Περιβάλλοντος Ανάπτυξης για την Υλοποίηση της Εφαρμογής

3.1 Εισαγωγή

Η προετοιμασία του υπολογιστή και η εγκατάσταση των απαραίτητων εργαλείων αποτελούν ένα από τα πιο σημαντικά βήματα για την επιτυχή ανάπτυξη εφαρμογών. Στο πλαίσιο αυτής της διπλωματικής εργασίας, η σωστή διαμόρφωση του συστήματος εξασφαλίζει ότι οι προγραμματιστές μπορούν να εργάζονται αποδοτικά, διασφαλίζοντας τη σταθερότητα και την αποτελεσματικότητα της εφαρμογής που πρόκειται να αναπτυχθεί.

Κάθε εφαρμογή απαιτεί συγκεκριμένες τεχνολογίες, γλώσσες προγραμματισμού, εργαλεία και βιβλιοθήκες. Έτσι, η εγκατάσταση και η ρύθμιση αυτών των στοιχείων πρέπει να γίνεται με ακρίβεια και προσοχή, ώστε να αποφεύγονται λάθη που μπορεί να επηρεάσουν την ανάπτυξη.

Σε αυτό το κεφάλαιο, θα περιγράψουμε τη διαδικασία διαμόρφωσης του υπολογιστή, συμπεριλαμβάνοντας όλα τα προγράμματα και τα λογισμικά που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Αρχικά, θα εξεταστούν τα απαραίτητα λογισμικά, όπως οι γλώσσες προγραμματισμού, οι βιβλιοθήκες και τα εργαλεία που επιλέχθηκαν σύμφωνα με τις ανάγκες της εφαρμογής. Στη συνέχεια, θα παρουσιαστούν οι πλατφόρμες και τα συστήματα που χρησιμοποιήθηκαν για τη δημιουργία του περιβάλλοντος ανάπτυξης, εστιάζοντας στις ρυθμίσεις και παραμέτρους που διασφαλίζουν την ομαλή λειτουργία του συστήματος.

Η σωστή επιλογή και διαμόρφωση των εργαλείων είναι καθοριστική για την αποτελεσματικότητα της ανάπτυξης. Παράλληλα, εξασφαλίζει ότι η αποσφαλμάτωση και οι δοκιμές θα γίνονται με ταχύτητα και ακρίβεια. Αυτό το κεφάλαιο έχει σκοπό να προσφέρει μια ολοκληρωμένη γενική εικόνα των βημάτων που ακολούθησαν για τη ρύθμιση του περιβάλλοντος εργασίας της συγκεκριμένης εφαρμογής.

3.2 Xampp

Το XAMPP είναι μια εντελώς δωρεάν, εύκολη στην εγκατάσταση διανομή Apache που περιέχει MariaDB, PHP και Perl. Το πακέτο ανοικτού κώδικα XAMPP έχει ρυθμιστεί έτσι ώστε να είναι όσο πιο εύκολο γίνεται τόσο στην εγκατάσταση όσο και τη χρήση [12].

3.2.1 Τι περιλαμβάνει το XAMPP;

- **Apache:** Ο διακομιστής ιστού που εξυπηρετεί τα αιτήματα για ιστοσελίδες.
- **MySQL/MariaDB:** Το σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων.
- **PHP και Perl:** Οι γλώσσες προγραμματισμού που χρησιμοποιούνται για την κατασκευή δυναμικών ιστοσελίδων.
- **phpMyAdmin:** Ένα εργαλείο με γραφικό περιβάλλον για τη διαχείριση βάσεων δεδομένων.
- **FileZilla FTP Server:** Ένας διακομιστής FTP για τη μεταφορά αρχείων.

- **Mercury Mail:** Ένας τοπικός mail server για δοκιμές.

3.2.2 Πώς λειτουργεί;

1. **Τοπικός server:** Το XAMPP εγκαθιστά έναν τοπικό server στον υπολογιστή έτσι ώστε να μπορούν να εκτελεστούν οι ιστοσελίδες χωρίς να χρειάζεστε την εγκατάσταση σε τρίτο σερβερ.
2. **Εύκολη χρήση:** Όλα τα εργαλεία που χρειάζονται περιλαμβάνονται σε πακέτα που κάνει την χρήση πολύ πιο εύκολη.
3. **Ευελιξία:** Μπορεί να αναπτυχθεί σε μια πληθώρα λειτουργικών όπως Linux, Windows και MacOS

Το XAMPP είναι ιδανικό για χρήση στην εκπαίδευση, ανάπτυξη πρωτοτύπων των εφαρμογών, και την γρήγορη δοκιμή τους.

3.2.3 Αναλυτική περιγραφή του Apache;

Ο **Apache HTTP Server** είναι ένας από τους παλαιότερους και πιο διαδεδομένους διακομιστές ιστού στον κόσμο. Αποτελεί τη ραχοκοκαλιά του παγκόσμιου ιστού (WWW), καθώς εξυπηρετεί εκατομμύρια ιστοσελίδες.

1. **Πώς λειτουργεί;**
2. **Λήψη αιτημάτων:** Ο Apache λειτουργεί ως διακομιστής που ακούει αιτήματα από προγράμματα περιήγησης (browsers), όπως το Chrome ή το Firefox. Όταν εισάγεται μια διεύθυνση URL, το πρόγραμμα περιήγησης επικοινωνεί με τον Apache μέσω πρωτοκόλλου HTTP ή HTTPS.
3. **Επεξεργασία:** Ο Apache εξυπηρετεί **στατικά αρχεία** (π.χ., HTML, CSS, εικόνες) εκτελεί **δυναμικά scripts** με τη βοήθεια γλωσσών προγραμματισμού όπως PHP και **Επιστρέφει αποτελέσματα** (συνήθως μια ιστοσελίδα στο πρόγραμμα περιήγησης).
4. **Βασικά χαρακτηριστικά:**
 - **Προσαρμοστικότητα:** Χρησιμοποιεί αρθρωτή δομή (modules) για να προσαρμόζεται σε διαφορετικές ανάγκες, όπως η υποστήριξη PHP, SSL, και άλλων πρωτοκόλλων.
 - **Ασφάλεια:** Παρέχει λειτουργίες όπως SSL/TLS για κρυπτογράφηση.
 - **Επεκτασιμότητα:** Υποστηρίζει μεγάλο αριθμό ταυτόχρονων αιτημάτων.
 - Στο πλαίσιο του XAMPP, ο Apache λειτουργεί ως ο κύριος server που εξυπηρετεί τις ιστοσελίδες τοπικά στον υπολογιστή που έχει εγκατασταθεί.

3.2.4 Αναλυτική περιγραφή του MySQL;

Η MySQL είναι ένα δημοφιλές σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS), σχεδιασμένο για την αποθήκευση, ανάκτηση και επεξεργασία μεγάλου όγκου δεδομένων. Στο XAMPP χρησιμοποιείται για τη διαχείριση των βάσεων δεδομένων στις δυναμικές ιστοσελίδες.

3.2.4.1 Πως λειτουργεί η MySQL

- **Αποθήκευση δεδομένων:** Τα δεδομένα οργανώνονται σε **πίνακες** (tables), που αποτελούνται από στήλες (columns) και γραμμές (rows). Για παράδειγμα, ένας πίνακας χρηστών μπορεί να περιέχει στήλες όπως Όνομα, Email, και Κωδικός και η κάθε γραμμή αποτελεί και μια εγγραφή δεδομένων στο σύστημα.
- **Χρήση SQL:** Η γλώσσα SQL (Structured Query Language) χρησιμοποιείται για:
 - Δημιουργία δομών δεδομένων (CREATE TABLE).
 - Εισαγωγή δεδομένων (INSERT).
 - Ενημέρωση δεδομένων (UPDATE).
 - Ανάκτηση δεδομένων (SELECT).
 - Διαγραφή δεδομένων (DELETE).
- **Αλληλεπίδραση με εφαρμογές:** Η MySQL συνεργάζεται με άλλες τεχνολογίες, όπως η PHP, για να εξυπηρετήσει τα αιτήματα των χρηστών. Για παράδειγμα:
 - Ένας χρήστης συνδέεται σε μια ιστοσελίδα.
 - Η PHP στέλνει ένα αίτημα στη MySQL για έλεγχο των διαπιστευτηρίων του.
 - Η MySQL επιστρέφει τα αποτελέσματα στην PHP, και αυτά εμφανίζονται στον χρήστη.

3.2.4.2 Χαρακτηριστικά

- **Ταχύτητα και αξιοπιστία:** Η MySQL είναι σχεδιασμένη για υψηλές επιδόσεις.
- **Κλιμάκωση:** Μπορεί να χειριστεί μεγάλα σύνολα δεδομένων και πολλαπλούς χρήστες ταυτόχρονα.
- **Υποστήριξη διεθνούς χαρακτήρα:** Διαχειρίζεται δεδομένα σε διαφορετικές γλώσσες και κωδικοποιήσεις.

Στο XAMPP, η MySQL διαχειρίζεται το back-end των δεδομένων, επιτρέποντας τη δημιουργία δυναμικών εφαρμογών ιστού.

3.2.5 Πώς συνεργάζονται;

Στο XAMPP, τα Apache, MySQL και PHP συνεργάζονται αρμονικά για την εξυπηρέτηση δυναμικών ιστοσελίδων. Ο Apache λειτουργεί ως ο διακομιστής ιστού που λαμβάνει τα αιτήματα των χρηστών μέσω του πρωτοκόλλου HTTP/HTTPS. Αν το αίτημα αφορά δυναμικό περιεχόμενο, ο Apache το διαβιβάζει στον PHP επεξεργαστή, ο οποίος εκτελεί τον κώδικα και παράγει την κατάλληλη έξοδο. Η PHP συχνά αλληλεπιδρά με τον διακομιστή MySQL για την ανάκτηση, εισαγωγή ή ενημέρωση

δεδομένων στη βάση δεδομένων. Η MySQL οργανώνει και αποθηκεύει τα δεδομένα σε σχεσιακούς πίνακες, παρέχοντας τη βάση για την ομαλή λειτουργία της εφαρμογής. Αφού η PHP επεξεργαστεί τον κώδικα και επιστρέψει τα αποτελέσματα, ο Apache αποστέλλει την τελική σελίδα HTML στον χρήστη. Αυτή η συνεργασία επιτρέπει την ομαλή δημιουργία, διαχείριση και παρουσίαση δυναμικού περιεχομένου στο περιβάλλον του XAMPP.

3.3 Composer

Το **Composer** είναι ένα σύγχρονο εργαλείο διαχείρισης εξαρτήσεων (dependency manager) για τη γλώσσα προγραμματισμού **PHP**. Επιτρέπει στους προγραμματιστές να διαχειρίζονται τις βιβλιοθήκες και τα πακέτα που χρειάζονται οι εφαρμογές τους, καθιστώντας την ανάπτυξη πιο οργανωμένη και αποδοτική [13][14].

3.3.1 Λειτουργία του Composer

Το Composer λειτουργεί βασισμένο στο αρχείο `composer.json`, όπου καθορίζονται οι εξαρτήσεις της εφαρμογής, όπως βιβλιοθήκες και πακέτα, καθώς και οι εκδόσεις τους. Το εργαλείο:

1. **Αναλύει** το αρχείο `composer.json` για να καταλάβει τι εξαρτήσεις απαιτούνται.
2. **Κατεβάζει** τις απαραίτητες βιβλιοθήκες από το αποθετήριο (repository) [Packagist](#) [15], που αποτελεί την κύρια πλατφόρμα φιλοξενίας πακέτων για το Composer.
3. **Διαχειρίζεται τις εκδόσεις** των βιβλιοθηκών, διασφαλίζοντας τη συμβατότητα με την εφαρμογή.

3.3.2 Η επιλογή του Composer

Το Composer έχει επικρατήσει ως το **καθιερωμένο εργαλείο διαχείρισης εξαρτήσεων** στην κοινότητα PHP για τους εξής λόγους:

- **Απλότητα και Εργονομία:** Το Composer ξεχωρίζει για την απλότητα και την εργονομία του, καθιστώντας το εύκολο στη χρήση, ακόμη και για προγραμματιστές με περιορισμένη εμπειρία. Μέσα από βασικές εντολές, όπως το `composer install` και το `composer update`, επιτυγχάνεται η αποτελεσματική διαχείριση των εξαρτήσεων, διευκολύνοντας σημαντικά τη διαδικασία ανάπτυξης λογισμικού.
- **Ευελιξία:** Παρέχει τη δυνατότητα καθορισμού συγκεκριμένων εκδόσεων βιβλιοθηκών ή εύρους εκδόσεων, χρησιμοποιώντας τη μέθοδο του `semantic versioning`. Έτσι, προσφέρει μεγαλύτερο έλεγχο στις εξαρτήσεις της εφαρμογής, διασφαλίζοντας την ομαλή ενσωμάτωσή τους στο έργο.
- **Διασφάλιση Συνεκτικότητας:** Το Composer δημιουργεί το αρχείο `composer.lock`, όπου αποθηκεύονται όλες οι ακριβείς εκδόσεις των πακέτων που εγκαθίστανται. Αυτό το χαρακτηριστικό εξασφαλίζει ότι η εφαρμογή θα λειτουργεί ομοίμορφα σε διαφορετικά περιβάλλοντα, ανεξάρτητα από την πλατφόρμα ή τις ενημερώσεις.

- **Υποστήριξη από την Κοινότητα:** Η ενεργή κοινότητα χρηστών του Composer προσφέρει πληθώρα διαθέσιμων πακέτων, καθώς και εκτενή τεκμηρίωση. Η υποστήριξη αυτή διευκολύνει την επίλυση προβλημάτων και ενισχύει την αξιοπιστία του εργαλείου για τους προγραμματιστές.
- **Προσαρμοστικότητα σε έργα μεγάλης κλίμακας:** Το Composer έχει σχεδιαστεί με τέτοιο τρόπο ώστε να μπορεί να διαχειρίζεται αποτελεσματικά έργα μεγάλης κλίμακας. Η δυνατότητά του να υποστηρίζει πολλαπλές εξαρτήσεις χωρίς συγκρούσεις καθιστά το εργαλείο κατάλληλο για την ανάπτυξη σύνθετων εφαρμογών.

Συνοψίζοντας, το Composer αποτελεί την καλύτερη επιλογή για τη διαχείριση εξαρτήσεων στις εφαρμογές PHP λόγω της αυτοματοποίησης, της αξιοπιστίας, και της ευκολίας που προσφέρει. Με την ικανότητά του να διαχειρίζεται βιβλιοθήκες, να διασφαλίζει τη συμβατότητα, και να ενσωματώνεται ομαλά με την υπάρχουσα υποδομή, το Composer έχει καθιερωθεί ως απαραίτητο εργαλείο για κάθε προγραμματιστή PHP.

3.4 Visual Studio Code (VS Code)

Το **Visual Studio Code (VS Code)** είναι ένα ελαφρύ και ταυτόχρονα ισχυρό εργαλείο ανάπτυξης λογισμικού (source code editor), που έχει αναπτυχθεί από τη Microsoft. Πρόκειται για ένα από τα πιο δημοφιλή εργαλεία στον κόσμο των προγραμματιστών, χάρη στην ευελιξία, την επεκτασιμότητα και τις προηγμένες δυνατότητες που προσφέρει.

3.4.1 Τι είναι το VS Code;

Το VS Code είναι ένας **επεξεργαστής κώδικα** που συνδυάζει τις δυνατότητες ενός επαγγελματικού περιβάλλοντος ανάπτυξης (IDE) με την απλότητα ενός απλού text editor. Υποστηρίζει δεκάδες γλώσσες προγραμματισμού και μπορεί να προσαρμοστεί πλήρως για να καλύψει τις ανάγκες του προγραμματιστή. Μερικά από τα χαρακτηριστικά του είναι τα παρακάτω :

- **Επεξεργασία κώδικα:** Παρέχει λειτουργίες όπως αυτόματη συμπλήρωση κώδικα, επισήμανση σύνταξης (syntax highlighting), και διαχείριση tabs για εύκολη πλοήγηση μεταξύ αρχείων.
- **Ενσωματωμένο Terminal:** Ο προγραμματιστής μπορεί να εκτελεί εντολές και scripts χωρίς να εγκαταλείψει το περιβάλλον του VS Code.
- **Εντοπισμός σφαλμάτων (Debugging):** Υποστηρίζει τη σύνδεση με debugging tools για την εκτέλεση και ανάλυση κώδικα σε πραγματικό χρόνο.
- **Ενσωμάτωση με Git:** Παρέχει ενσωματωμένη υποστήριξη για συστήματα ελέγχου έκδοσης όπως το Git, επιτρέποντας στους χρήστες να κάνουν commits, branches και merges απευθείας από το εργαλείο.

3.4.2 Η επιλογή του VS Code

Η χρήση του VS Code προσφέρει πολλαπλά οφέλη για τους προγραμματιστές, καθώς συνδυάζει απόδοση, ευκολία στη χρήση, και πλούσια λειτουργικότητα. Ορισμένοι από τους λόγους για τους οποίους αποτελεί δημοφιλή επιλογή είναι οι εξής:

1. **Υποστήριξη Πολλαπλών Γλωσσών:** Υποστηρίζει γλώσσες όπως Python, JavaScript, PHP, C++, Java, και πολλές άλλες. Κάθε γλώσσα μπορεί να προσαρμοστεί περαιτέρω μέσω επεκτάσεων (extensions).
2. **Επεκτάσεις και Marketplace:** Το VS Code διαθέτει ένα ευρύ φάσμα επεκτάσεων, οι οποίες βελτιώνουν τις δυνατότητές του. Οι επεκτάσεις περιλαμβάνουν εργαλεία για debugging, υποστήριξη νέων γλωσσών, θεματικές διατάξεις (themes), και πολλά άλλα.
3. **Ελαφρύ και γρήγορο:** Αν και προσφέρει δυνατότητες παρόμοιες με αυτές ενός IDE, παραμένει γρήγορο και δεν απαιτεί μεγάλο φόρτο στους πόρους του συστήματος.
4. **Διαλειτουργικότητα:** Το VS Code μπορεί να χρησιμοποιηθεί σε οποιοδήποτε λειτουργικό σύστημα (Windows, macOS, Linux), κάνοντάς το ιδανικό για προγραμματιστές που εργάζονται σε πολλαπλές πλατφόρμες.

Συνοψίζοντας, το Composer αποτελεί την καλύτερη επιλογή για τη διαχείριση εξαρτήσεων στις εφαρμογές PHP λόγω της αυτοματοποίησης, της αξιοπιστίας, και της ευκολίας που προσφέρει. Με την ικανότητά του να διαχειρίζεται βιβλιοθήκες, να διασφαλίζει τη συμβατότητα, και να ενσωματώνεται ομαλά με την υπάρχουσα υποδομή, το Composer έχει καθιερωθεί ως απαραίτητο εργαλείο για κάθε προγραμματιστή PHP.

3.4.3 Visual Studio Code σαν καλύτερη επιλογή

Το VS Code έχει αναδειχθεί ως η κορυφαία επιλογή για προγραμματιστές παγκοσμίως, και αυτό οφείλεται σε αρκετούς παράγοντες:

1. **Προσαρμοστικότητα:** Κάθε προγραμματιστής μπορεί να προσαρμόσει το περιβάλλον σύμφωνα με τις ανάγκες του, χρησιμοποιώντας ρυθμίσεις, επεκτάσεις, και θεματικές διατάξεις.
2. **Κοινότητα και Υποστήριξη:** Με μια τεράστια κοινότητα χρηστών, είναι εύκολο να βρείτε λύσεις σε προβλήματα, να μάθετε νέες τεχνικές, και να έχετε πρόσβαση σε εκατοντάδες δωρεάν επεκτάσεις.
3. **Ενσωματωμένες λειτουργίες:** Η ενσωμάτωση λειτουργιών όπως debugging, Git, και terminal το καθιστούν ολοκληρωμένο εργαλείο για την ανάπτυξη λογισμικού χωρίς να απαιτείται επιπλέον λογισμικό.
4. **Open Source:** Παρόλο που αναπτύσσεται από τη Microsoft, το VS Code είναι ανοιχτού κώδικα, γεγονός που το καθιστά διαφανές και ευέλικτο για παραμετροποίηση από την κοινότητα.
5. **Υποστήριξη για Remote Development:** Το VS Code υποστηρίζει εργαλεία για απομακρυσμένη ανάπτυξη (Remote Development), επιτρέποντας στους χρήστες να εργάζονται σε απομακρυσμένα περιβάλλοντα ή containers.

Συμπερνεύοντας, το VS Code είναι ένα εξαιρετικό εργαλείο ανάπτυξης λογισμικού που καλύπτει τις ανάγκες προγραμματιστών κάθε επιπέδου. Η προσαρμοστικότητα, οι ισχυρές λειτουργίες, και η ευκολία στη χρήση το καθιστούν ιδανική επιλογή τόσο για μικρά έργα όσο και για μεγάλης κλίμακας εφαρμογές. Με τη συνεχή ανάπτυξη και τη στήριξη της κοινότητας, το VS Code παραμένει στην κορυφή των επιλογών για σύγχρονους προγραμματιστές.

3.4.4 Visual Studio Code Extensions

Το **Visual Studio Code (VS Code)** έχει καθιερωθεί ως ένα από τα πιο δημοφιλή εργαλεία ανάπτυξης λογισμικού, κυρίως λόγω της ευελιξίας και της προσαρμοστικότητάς του. Ένας από τους σημαντικότερους παράγοντες που συμβάλλουν σε αυτό είναι η υποστήριξη επεκτάσεων (**extensions**). Οι επεκτάσεις επιτρέπουν στους προγραμματιστές να προσαρμόζουν το περιβάλλον τους και να το επεκτείνουν, ώστε να καλύπτει πλήρως τις απαιτήσεις κάθε έργου.

Με τη χρήση των **extensions**, το VS Code μπορεί να υποστηρίξει ένα ευρύ φάσμα γλωσσών προγραμματισμού, εργαλείων **debugging**, μορφοποίησης κώδικα, και πολλά άλλα. Αυτά τα εργαλεία ενισχύουν την παραγωγικότητα, βελτιώνουν την ποιότητα του κώδικα και απλοποιούν τις πολύπλοκες διεργασίες ανάπτυξης. Επιπλέον, η κοινότητα γύρω από το VS Code αναπτύσσει συνεχώς νέες επεκτάσεις, εμπλουτίζοντας το οικοσύστημά του.

Η χρήση των κατάλληλων επεκτάσεων δεν είναι απλώς βοηθητική· είναι απαραίτητη για την ταχύτερη, αποδοτικότερη, και πιο οργανωμένη ανάπτυξη λογισμικού. Στις παρακάτω παραγράφους, παρουσιάζονται ορισμένες από τις πιο χρήσιμες επεκτάσεις που μπορούν να μεταμορφώσουν την εμπειρία χρήσης του VS Code αλλά και που χρησιμοποιήθηκαν για να γίνει η εφαρμογή.

3.4.1.1 Command Palette

Η **Command Palette** αποτελεί ένα από τα πιο βασικά και ευέλικτα εργαλεία του VS Code, επιτρέποντας την εκτέλεση εντολών χωρίς τη χρήση μενού ή τη μνήμη περίπλοκων συντομεύσεων. Μέσω του συνδυασμού πλήκτρων *Ctrl+Shift+P* (ή *Cmd+Shift+P* σε macOS), οι χρήστες μπορούν να εκτελούν γρήγορα ενέργειες όπως η εγκατάσταση επεκτάσεων, η αλλαγή ρυθμίσεων, η εκτέλεση **debugging** και πολλές άλλες λειτουργίες. Είναι ιδιαίτερα χρήσιμη για νέους προγραμματιστές ή όσους προτιμούν έναν καθαρό και λιτό τρόπο εργασίας. Επιπλέον, η **Command Palette** είναι πλήρως αναζητήσιμη, επιτρέποντας την εύκολη εύρεση εντολών ακόμη και σε πολύπλοκα έργα, ενώ μπορεί να προσαρμοστεί για την εκτέλεση συγκεκριμένων **task**. Η ευελιξία αυτή την καθιστά αναντικατάστατη για κάθε χρήστη του VS Code.

3.4.1.2 NPM Script Runner

Το **NPM Script Runner** διευκολύνει την εκτέλεση **scripts** που περιλαμβάνονται στο αρχείο `package.json` μιας εφαρμογής JavaScript ή Node.js. Με αυτήν την επέκταση, οι προγραμματιστές μπορούν να εκτελούν **scripts** όπως *start*, *build*, *test* ή οποιοδήποτε άλλο προσαρμοσμένο **script**, απευθείας μέσα από το περιβάλλον του VS Code. Αντί να χρειάζεται να μεταβούν στο **terminal** για να γράψουν εντολές, η επέκταση προσφέρει ένα γραφικό περιβάλλον για την επιλογή και εκτέλεση **scripts**, εξοικονομώντας χρόνο και μειώνοντας τα λάθη. Επίσης, παρέχει χρήσιμες πληροφορίες για την κατάσταση του κάθε **script** και τα αποτελέσματα εκτέλεσης, καθιστώντας τη διαδικασία ανάπτυξης πιο οργανωμένη και αποδοτική.

3.4.1.3 Format HTML in PHP

Η επέκταση **Format HTML in PHP** λύνει ένα από τα πιο συνηθισμένα προβλήματα που συναντούν οι προγραμματιστές PHP: την κακή μορφοποίηση HTML κώδικα μέσα σε PHP αρχεία. Δεδομένου ότι πολλά έργα PHP περιλαμβάνουν HTML ενσωματωμένο στον κώδικα, αυτή η επέκταση προσφέρει τη δυνατότητα αυτόματης στοίχισης και μορφοποίησης του HTML, κάνοντας τον κώδικα πιο ευανάγνωστο και οργανωμένο. Επιπλέον, ενισχύει τη συμβατότητα μεταξύ HTML και PHP

συντακτικού, διασφαλίζοντας ότι δεν θα δημιουργηθούν προβλήματα μορφοποίησης κατά την επεξεργασία του αρχείου. Αυτή η λειτουργία είναι ιδιαίτερα χρήσιμη σε έργα που απαιτούν συχνές αλλαγές στο front-end και το back-end.

3.4.1.4 Live Server

Το **Live Server** είναι μία από τις πιο δημοφιλείς επεκτάσεις για την ανάπτυξη ιστοσελίδων, καθώς παρέχει έναν τοπικό διακομιστή με δυνατότητα αυτόματης ανανέωσης του browser κάθε φορά που αποθηκεύεται μια αλλαγή στον κώδικα. Αυτή η λειτουργία, γνωστή ως live reload, εξοικονομεί πολύτιμο χρόνο και επιτρέπει στους προγραμματιστές να βλέπουν άμεσα τα αποτελέσματα των τροποποιήσεών τους. Το Live Server υποστηρίζει HTML, CSS, JavaScript και πολλά άλλα αρχεία, ενώ είναι εξαιρετικά απλό στη χρήση. Η άμεση ανατροφοδότηση που παρέχει βοηθά στην ταχύτερη επίλυση προβλημάτων και στη βελτίωση της εμπειρίας ανάπτυξης.

3.4.1.5 Material Icon Theme

Το **Material Icon Theme** είναι μια επέκταση που προσφέρει ένα σύνολο από καλαίσθητα και λειτουργικά εικονίδια για την απεικόνιση αρχείων και φακέλων στο VS Code. Βοηθά τους προγραμματιστές να εντοπίζουν γρήγορα τα αρχεία που χρειάζονται, καθώς κάθε τύπος αρχείου (π.χ., .js, .php, .json) έχει το δικό του μοναδικό εικονίδιο. Αυτή η βελτιωμένη οπτική αναπαράσταση ενισχύει την οργανωτικότητα και καθιστά πιο ευχάριστη την εμπειρία χρήσης του VS Code. Επιπλέον, το Material Icon Theme προσφέρει επιλογές προσαρμογής, επιτρέποντας στους χρήστες να τροποποιούν τα εικονίδια σύμφωνα με τις προσωπικές τους προτιμήσεις.

3.4.1.6 PHP Debug

Η επέκταση **PHP Debug** προσθέτει προηγμένες δυνατότητες εντοπισμού σφαλμάτων για εφαρμογές PHP στο VS Code, επιτρέποντας τη σύνδεση με το Xdebug, ένα δημοφιλές εργαλείο debugging. Οι χρήστες μπορούν να ρυθμίζουν σημεία διακοπής (breakpoints), να εκτελούν τον κώδικα βήμα προς βήμα και να αναλύουν τις τιμές των μεταβλητών κατά τη διάρκεια της εκτέλεσης. Επιπλέον, η επέκταση παρέχει λεπτομερείς πληροφορίες για την ροή του κώδικα και τις εξαιρέσεις που μπορεί να προκύψουν. Αυτές οι λειτουργίες καθιστούν τον εντοπισμό και τη διόρθωση σφαλμάτων ταχύτερη και πιο αποδοτική, ειδικά σε πολύπλοκα έργα.

3.4.1.7 PHP Extension Pack

Το **PHP Extension Pack** είναι ένα πακέτο επεκτάσεων που προσφέρει όλα τα βασικά εργαλεία για την ανάπτυξη εφαρμογών PHP. Περιλαμβάνει εργαλεία όπως PHP Debug, PHP Intelephense, και PHP IntelliSense, τα οποία συνδυάζονται για να δημιουργήσουν ένα πλήρως λειτουργικό και αποδοτικό περιβάλλον ανάπτυξης. Με το Extension Pack, οι προγραμματιστές έχουν πρόσβαση σε αυτόματη συμπλήρωση κώδικα, debugging, και ανάλυση σύνταξης, καθιστώντας το VS Code ισχυρό εργαλείο για την ανάπτυξη PHP εφαρμογών.

3.4.1.8 PHP Intelephense

Το **PHP Intelephense** είναι μια από τις πιο δημοφιλείς επεκτάσεις για τη γλώσσα PHP, προσφέροντας δυνατότητες όπως αυτόματη συμπλήρωση κώδικα, ανάλυση σύνταξης, και εντοπισμό σφαλμάτων. Χάρη στη βελτιωμένη μηχανή της, μπορεί να επεξεργάζεται μεγάλα έργα PHP γρήγορα και αποτελεσματικά. Επιπλέον, παρέχει πληροφορίες σχετικά με τις συναρτήσεις, τις κλάσεις, και τις παραμέτρους, βοηθώντας τους προγραμματιστές να γράφουν καλύτερο κώδικα.

3.4.1.9 PHP IntelliSense

Η **PHP IntelliSense** επεκτείνει τη λειτουργικότητα του VS Code για τη γλώσσα PHP, προσφέροντας δυνατότητες αυτόματης συμπλήρωσης και επισήμανσης σύνταξης. Παρέχει προτάσεις κώδικα σε πραγματικό χρόνο, βοηθώντας στην αποφυγή λαθών και στη γρήγορη συγγραφή του κώδικα. Είναι ιδιαίτερα χρήσιμη για νέους προγραμματιστές που θέλουν να εξοικειωθούν με τη σύνταξη της PHP.

3.4.1.10 Prettier - Code Formatter

Το **Prettier** είναι ένα εργαλείο μορφοποίησης κώδικα που εφαρμόζει ένα συνεπές στυλ σε πολλές γλώσσες όπως JavaScript, HTML, CSS και PHP. Χάρη στις δυνατότητές του, μπορεί να αναδιατάσσει τη στοίχιση, τα διαστήματα, και τη δομή του κώδικα με στόχο την ομοιομορφία και τη βελτίωση της αναγνωσιμότητας.

3.4.1.11 Prettier ESLint

Το **Prettier ESLint** συνδυάζει τις δυνατότητες του Prettier με τους κανόνες του ESLint, εξασφαλίζοντας ότι ο κώδικας είναι καλογραμμένος και συμβατός με τις καλύτερες πρακτικές της γλώσσας. Είναι ιδανικό για έργα που απαιτούν υψηλή ποιότητα και συνέπεια στον κώδικα.

3.5 Επίλογος

Συμπερασματικά, η ορθή διαμόρφωση του αναπτυξιακού περιβάλλοντος αποτελεί θεμελιώδη διαδικασία για την επιτυχή ανάπτυξη εφαρμογών και λογισμικού. Η επιλογή των κατάλληλων εργαλείων και η σωστή εγκατάσταση και ρύθμισή τους συμβάλλει καθοριστικά στη σταθερότητα, την αποδοτικότητα και τη διατήρηση της ποιότητας κατά τη διάρκεια όλου του κύκλου ζωής μιας εφαρμογής.

Το κεφάλαιο αυτό ανέδειξε τη σημασία χρήσης εργαλείων όπως το XAMPP, το Composer και το Visual Studio Code, τα οποία επιτρέπουν στους προγραμματιστές να εργάζονται με τρόπο οργανωμένο και αποδοτικό. Το XAMPP, ως μια πλήρης λύση διαχείρισης τοπικών διακομιστών, παρέχει όλα τα

απαραίτητα εργαλεία για τη φιλοξενία και τη διαχείριση δυναμικών ιστοσελίδων σε ένα ενιαίο πακέτο, καθιστώντας την ανάπτυξη τοπικών εφαρμογών απλή και προσβάσιμη. Ο Apache, η MySQL και η PHP που περιλαμβάνονται στο XAMPP, συνεργάζονται άψογα για την παροχή ενός ολοκληρωμένου περιβάλλοντος ανάπτυξης.

Η χρήση του Composer για τη διαχείριση εξαρτήσεων στις PHP εφαρμογές επιβεβαιώνει τη σημασία της αυτοματοποίησης στη διαχείριση βιβλιοθηκών και πακέτων. Η δυνατότητα διατήρησης μιας σταθερής και ενημερωμένης λίστας εξαρτήσεων μέσω του αρχείου `composer.json` απλοποιεί τη διαδικασία ανάπτυξης, ενώ παράλληλα διασφαλίζει τη συμβατότητα μεταξύ διαφορετικών εκδόσεων λογισμικού και τη σταθερότητα του κώδικα.

Το Visual Studio Code (VS Code) αναδείχθηκε ως ένα ισχυρό εργαλείο ανάπτυξης λογισμικού, χάρη στην ευελιξία και τις προηγμένες δυνατότητές του. Με λειτουργίες όπως η ενσωματωμένη υποστήριξη για Git, το ενσωματωμένο τερματικό και η εκτεταμένη γκάμα επεκτάσεων, το VS Code προσφέρει στους προγραμματιστές ένα πλήρως εξοπλισμένο περιβάλλον εργασίας που υποστηρίζει τόσο αρχάριους όσο και έμπειρους προγραμματιστές. Οι επεκτάσεις, όπως το Live Server, το PHP Debug και το PHP Intelephense, ενισχύουν ακόμη περισσότερο τη λειτουργικότητα του εργαλείου, καθιστώντας το ιδανικό για ανάπτυξη PHP εφαρμογών.

Η διαμόρφωση ενός τέτοιου ολοκληρωμένου περιβάλλοντος ανάπτυξης δεν συμβάλλει μόνο στην αποτελεσματικότητα της ανάπτυξης, αλλά και στη βελτίωση της ποιότητας του παραγόμενου κώδικα. Μέσω εργαλείων μορφοποίησης όπως το Prettier και επεκτάσεων debugging όπως το PHP Debug, οι προγραμματιστές μπορούν να διασφαλίσουν ότι ο κώδικας παραμένει ευανάγνωστος, οργανωμένος και απαλλαγμένος από λάθη.

Επιπλέον, η χρήση αυτών των εργαλείων και τεχνολογιών προσφέρει σημαντικά πλεονεκτήματα στη διαδικασία δοκιμών και αποσφαλμάτωσης. Η δυνατότητα εκτέλεσης τοπικών servers, η αυτοματοποίηση της εγκατάστασης βιβλιοθηκών και η εύκολη διαχείριση των αρχείων μέσω του VS Code εξασφαλίζουν ότι η ανάπτυξη παραμένει αποδοτική και χωρίς περιττά εμπόδια.

Εν κατακλείδι, η παρουσίαση αυτών των εργαλείων και της μεταξύ τους συνεργασίας στο πλαίσιο της διπλωματικής εργασίας καταδεικνύει τη σημασία της κατάλληλης προετοιμασίας κατά την ανάπτυξη λογισμικού. Η επιλογή και ορθολογική χρήση του XAMPP, του Composer και του Visual Studio Code παρέχουν ένα σταθερό, οργανωμένο και αποτελεσματικό περιβάλλον εργασίας, το οποίο υποστηρίζει όχι μόνο την ανάπτυξη αλλά και τη διατήρηση και εξέλιξη του λογισμικού με συνέπεια και επαγγελματισμό. Αυτή η μεθοδολογία διαμόρφωσης περιβάλλοντος εργασίας αποτελεί αναπόσπαστο μέρος κάθε σοβαρής προγραμματιστικής εργασίας, διασφαλίζοντας την ποιότητα, τη σταθερότητα και την ευκολία συντήρησης των εφαρμογών που αναπτύσσονται.

Κεφάλαιο 4ο: Αρχιτεκτονική Συστήματος

4.1 Εισαγωγή

Η UML, συντομογραφία της Unified Modeling Language, είναι μια τυποποιημένη γλώσσα μοντελοποίησης που αποτελείται από ένα ολοκληρωμένο σύνολο διαγραμμάτων, η οποία αναπτύχθηκε για να βοηθήσει τους προγραμματιστές συστημάτων και λογισμικού για τον προσδιορισμό, την οπτικοποίηση, την κατασκευή και την τεκμηρίωση των αντικειμένων των συστημάτων λογισμικού, καθώς και για τη μοντελοποίηση επιχειρήσεων και άλλων συστημάτων που δεν είναι λογισμικού. Η UML αντιπροσωπεύει μια συλλογή βέλτιστων πρακτικών μηχανικής που έχουν αποδειχθεί επιτυχείς στη μοντελοποίηση μεγάλων και πολύπλοκων συστημάτων. Η UML αποτελεί πολύ σημαντικό μέρος της ανάπτυξης αντικειμενοστραφούς λογισμικού και της διαδικασίας ανάπτυξης λογισμικού. Η UML χρησιμοποιεί ως επί το πλείστον γραφικές σημειώσεις για να εκφράσει τη σχεδίαση έργων λογισμικού. Η χρήση της UML βοηθά τις ομάδες έργου να επικοινωνούν, να διερευνούν τα πιθανά σχέδια και να επικυρώνουν τον αρχιτεκτονικό σχεδιασμό του λογισμικού. Σε αυτό το άρθρο, θα σας δώσουμε λεπτομερείς ιδέες σχετικά με το τι είναι η UML, την ιστορία της UML και μια περιγραφή κάθε τύπου διαγράμματος UML, μαζί με παραδείγματα UML[16].

Καθώς η στρατηγική αξία του λογισμικού αυξάνεται για πολλές εταιρείες, ο κλάδος αναζητά τεχνικές για την αυτοματοποίηση της παραγωγής λογισμικού, τη βελτίωση της ποιότητας και τη μείωση του κόστους και του χρόνου διάθεσης στην αγορά. Οι τεχνικές αυτές περιλαμβάνουν την τεχνολογία συστατικών, τον οπτικό προγραμματισμό, τα πρότυπα και τα πλαίσια. Οι επιχειρήσεις αναζητούν επίσης τεχνικές για τη διαχείριση της πολυπλοκότητας των συστημάτων καθώς αυτά αυξάνονται σε έκταση και κλίμακα. Ειδικότερα, αναγνωρίζουν την ανάγκη επίλυσης επαναλαμβανόμενων αρχιτεκτονικών προβλημάτων, όπως η φυσική κατανομή, η ταυτόχρονη χρήση, η αντιγραφή, η ασφάλεια, η εξισορρόπηση φορτίου και η ανοχή σε σφάλματα. Επιπλέον, η ανάπτυξη του Παγκόσμιου Ιστού, ενώ έκανε κάποια πράγματα απλούστερα, επιδείνωσε αυτά τα αρχιτεκτονικά προβλήματα. Η ενοποιημένη γλώσσα μοντελοποίησης (UML) σχεδιάστηκε για να ανταποκριθεί σε αυτές τις ανάγκες. Οι πρωταρχικοί στόχοι του σχεδιασμού της UML συνοψίζονται από τον Page-Jones στο Fundamental Object-Oriented Design in UML ως εξής:

- Παροχή στους χρήστες μιας έτοιμης προς χρήση, εκφραστικής οπτικής γλώσσας μοντελοποίησης, ώστε να μπορούν να αναπτύξουν και να ανταλλάξουν μοντέλα με νόημα.
- Παροχή μηχανισμών επεκτασιμότητας και εξειδίκευσης για την επέκταση των βασικών εννοιών.
- Να είναι ανεξάρτητη από συγκεκριμένες γλώσσες προγραμματισμού και διαδικασίες ανάπτυξης.
- Να παρέχει μια τυπική βάση για την κατανόηση της γλώσσας μοντελοποίησης.
- Να ενθαρρύνει την ανάπτυξη της αγοράς εργαλείων ΟΟ.
- Να υποστηρίζει έννοιες ανάπτυξης υψηλότερου επιπέδου, όπως συνεργασίες, πλαίσια, πρότυπα και συστατικά.
- Ενσωμάτωση βέλτιστων πρακτικών.

4.2 Idea Lists UML

Στη διπλωματική αυτή εργασία, η χρήση των διαγραμμάτων UML (Unified Modeling Language) αποτελεί βασικό εργαλείο για την ανάλυση και τη μοντελοποίηση της λειτουργικότητας της πλατφόρμας καταχώρησης ιδεών για Hackathon. Η ενσωμάτωση αυτών των διαγραμμάτων κρίνεται απαραίτητη, καθώς επιτρέπει την καλύτερη κατανόηση της ροής δεδομένων, των αλληλεπιδράσεων μεταξύ των χρηστών και του συστήματος, καθώς και τη δομή της εφαρμογής.

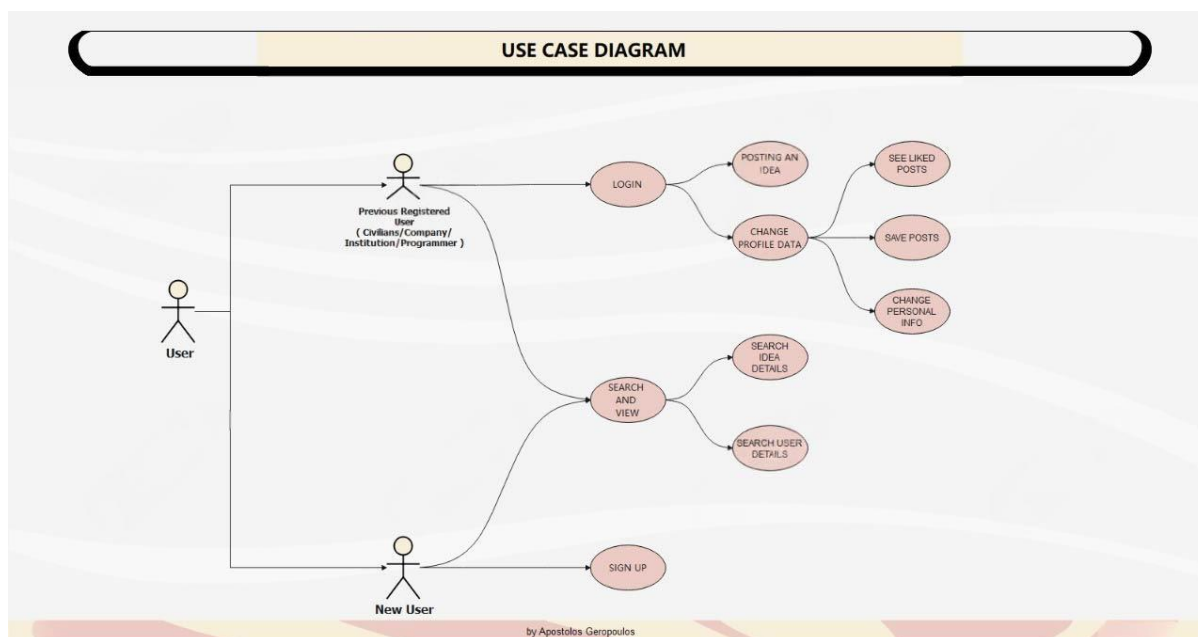
Το Sequence Diagram χρησιμοποιείται για να απεικονίσει τη διαδοχική ροή των αλληλεπιδράσεων μεταξύ των χρηστών και του συστήματος. Στη συγκεκριμένη πλατφόρμα, παρουσιάζει τη ροή των ενεργειών όπως η περιήγηση στις ιδέες, η προβολή λεπτομερειών, η εγγραφή/είσοδος χρηστών, η αποθήκευση αγαπημένων ιδεών, η αξιολόγηση με like και η υποβολή νέων ιδεών. Το Sequence Diagram επιτρέπει τη σαφή κατανόηση των χρονικών αλληλουχιών και της επικοινωνίας μεταξύ των οντοτήτων του συστήματος.

Το Use Case Diagram, από την άλλη, εστιάζει στις λειτουργικές απαιτήσεις της πλατφόρμας, περιγράφοντας ποιοι χρήστες (Guest User, Registered User, Admin, Company/Organization) αλληλεπιδρούν με το σύστημα και ποιες δυνατότητες έχουν στη διάθεσή τους. Μέσω αυτού του διαγράμματος καθίσταται σαφές ποιοι ρόλοι επιτρέπεται να εκτελούν συγκεκριμένες ενέργειες, όπως η καταχώρηση ιδεών ή η έγκριση αυτών από τον διαχειριστή.

Η χρήση αυτών των διαγραμμάτων είναι κρίσιμη για διάφορους λόγους. Πρώτον, βοηθούν στην οπτικοποίηση της ροής πληροφοριών και στη διάκριση των διαφορετικών σεναρίων χρήσης. Δεύτερον, υποστηρίζουν τη σαφή επικοινωνία μεταξύ των μελών της ομάδας ανάπτυξης, εξασφαλίζοντας κοινή κατανόηση των απαιτήσεων και της λογικής λειτουργίας του συστήματος. Επιπλέον, παρέχουν μια ισχυρή βάση για τον έλεγχο της πληρότητας των λειτουργικών απαιτήσεων και τη διασφάλιση της συνοχής στη σχεδίαση της εφαρμογής.

Πριν την παρουσίαση των διαγραμμάτων, κρίνεται σημαντικό να εξηγηθεί η συνολική λειτουργικότητα της πλατφόρμας, δηλαδή ο σκοπός της, οι χρήστες που την αξιοποιούν και οι βασικές δυνατότητες που παρέχει. Η κατανόηση αυτών των στοιχείων διευκολύνει την ερμηνεία των διαγραμμάτων που ακολουθούν και την πληρέστερη κατανόηση της δομής και των ροών της εφαρμογής. Σε αυτό το πλαίσιο, στις επόμενες ενότητες θα εξηγήσουμε το Use Case Diagram της εφαρμογής αλλά και την αλληλεπίδραση των τριών (3) ειδών χρηστών που έχουμε (Guest, Registered, Admin) με την εφαρμογή.

4.3 Use Case Diagram (χρηστών)



Το παραπάνω διάγραμμα Use Case απεικονίζει τη λειτουργικότητα ενός συστήματος που παρέχει δυνατότητες σε χρήστες να αλληλεπιδρούν με την πλατφόρμα μέσω διαφορετικών δραστηριοτήτων. Το σύστημα περιλαμβάνει δύο κύριες κατηγορίες χρηστών: τον "Νέο Χρήστη" και τον "Προηγούμενος Εγγεγραμμένο Χρήστη". Ο "Νέος Χρήστης" είναι κάποιος που δεν έχει ήδη λογαριασμό και θέλει να εγγραφεί στην πλατφόρμα, ενώ ο "Προηγούμενος Εγγεγραμμένος Χρήστης" αναφέρεται σε χρήστες που έχουν ήδη λογαριασμό και μπορούν να χρησιμοποιήσουν όλες τις παρεχόμενες λειτουργίες.

Η πρώτη λειτουργία του συστήματος είναι η εγγραφή, η οποία αφορά αποκλειστικά τον "Νέο Χρήστη". Μέσω αυτής της διαδικασίας, ο χρήστης υποβάλλει τα απαραίτητα στοιχεία για να δημιουργήσει έναν λογαριασμό στην πλατφόρμα. Μετά την επιτυχή εγγραφή, ο "Νέος Χρήστης" αποκτά πρόσβαση στις υπόλοιπες δυνατότητες του συστήματος.

Η επόμενη λειτουργία είναι η σύνδεση, η οποία απευθύνεται μόνο στους εγγεγραμμένους χρήστες. Μέσω αυτής της διαδικασίας, οι χρήστες εισάγουν τα διαπιστευτήριά τους και αποκτούν πρόσβαση στον προσωπικό τους λογαριασμό. Μόλις ολοκληρωθεί η σύνδεση, οι χρήστες μπορούν να προχωρήσουν σε άλλες ενέργειες, όπως η δημοσίευση ιδεών, η διαχείριση του προφίλ τους και η αναζήτηση πληροφοριών.

Μία από τις κύριες λειτουργίες της πλατφόρμας είναι η δυνατότητα δημοσίευσης ιδεών. Ο "Προηγούμενος Εγγεγραμμένος Χρήστης" μπορεί να υποβάλει τις ιδέες του, συμπεριλαμβάνοντας λεπτομέρειες, όπως περιγραφές, εικόνες ή άλλες σχετικές πληροφορίες. Αυτή η λειτουργία στοχεύει στη διαμοίραση γνώσεων, σκέψεων ή έργων μεταξύ των χρηστών.

Η διαχείριση των δεδομένων προφίλ είναι μια ακόμη σημαντική δυνατότητα που παρέχεται στους εγγεγραμμένους χρήστες. Μέσω αυτής της λειτουργίας, οι χρήστες μπορούν να αλλάζουν προσωπικές πληροφορίες, όπως το όνομά τους, τη διεύθυνση email ή άλλες ρυθμίσεις λογαριασμού. Επιπλέον, μπορούν να αποθηκεύουν αναρτήσεις που τους ενδιαφέρουν ή να βλέπουν αναρτήσεις που έχουν προηγουμένως εκφράσει την προτίμησή τους.

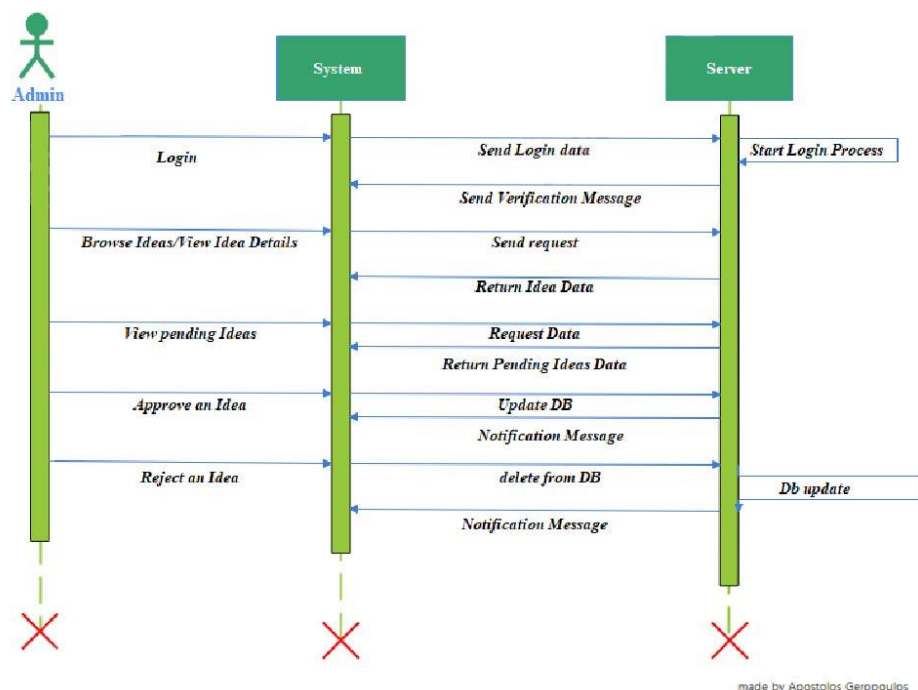
Η πλατφόρμα προσφέρει επίσης δυνατότητα αναζήτησης πληροφοριών, είτε πρόκειται για ιδέες είτε για άλλους χρήστες. Οι εγγεγραμμένοι χρήστες έχουν τη δυνατότητα να αναζητήσουν και να δουν

λεπτομέρειες για συγκεκριμένες ιδέες που έχουν δημοσιευθεί στην πλατφόρμα. Παράλληλα, μπορούν να αναζητήσουν πληροφορίες για άλλους χρήστες, όπως το προφίλ τους ή τις δραστηριότητές τους στην πλατφόρμα.

Συνοψίζοντας, το διάγραμμα Use Case παρουσιάζει με σαφήνεια τις διάφορες λειτουργίες του συστήματος, κατηγοριοποιώντας τις ανάλογα με τον τύπο του χρήστη. Η πλατφόρμα υποστηρίζει πλήρως τόσο νέους όσο και ήδη εγγεγραμμένους χρήστες, παρέχοντας μια σειρά από λειτουργίες που καλύπτουν τις ανάγκες δημιουργικότητας, διαχείρισης δεδομένων και αναζήτησης πληροφοριών.

4.4 Sequence Diagrams

ADMIN SEQUENCE DIAGRAM



4.4.1 Ανάλυση για το Admin User Sequence Diagram

Το Admin Sequence Diagram περιγράφει τις αλληλεπιδράσεις που πραγματοποιούνται ανάμεσα στον διαχειριστή (Admin), το σύστημα και τον διακομιστή για την εκτέλεση διάφορων διαχειριστικών λειτουργιών.

1. Σύνδεση (Login):

Ο διαχειριστής ξεκινά τη διαδικασία σύνδεσης εισάγοντας τα διαπιστευτήριά του. Το σύστημα στέλνει τα δεδομένα σύνδεσης στον διακομιστή, ο οποίος αρχίζει τη διαδικασία επαλήθευσης. Μετά την ολοκλήρωση της επαλήθευσης, ο διαχειριστής αποκτά πρόσβαση στις υπόλοιπες λειτουργίες.

2. Περιήγηση Ιδεών / Προβολή Λεπτομερειών Ιδεών (Browse Ideas/View Idea Details):

Ο διαχειριστής ζητά από το σύστημα να προβάλει ιδέες ή λεπτομέρειες για μια συγκεκριμένη

ιδέα. Το σύστημα στέλνει αίτημα στον διακομιστή, ο οποίος επιστρέφει τα αντίστοιχα δεδομένα στον διαχειριστή μέσω του συστήματος.

3. **Προβολή Εκκρεμών Ιδεών (View Pending Ideas):**

Ο διαχειριστής έχει τη δυνατότητα να δει ιδέες που βρίσκονται σε κατάσταση εκκρεμότητας. Το σύστημα ζητά τα δεδομένα από τον διακομιστή, ο οποίος επιστρέφει τις εκκρεμείς ιδέες.

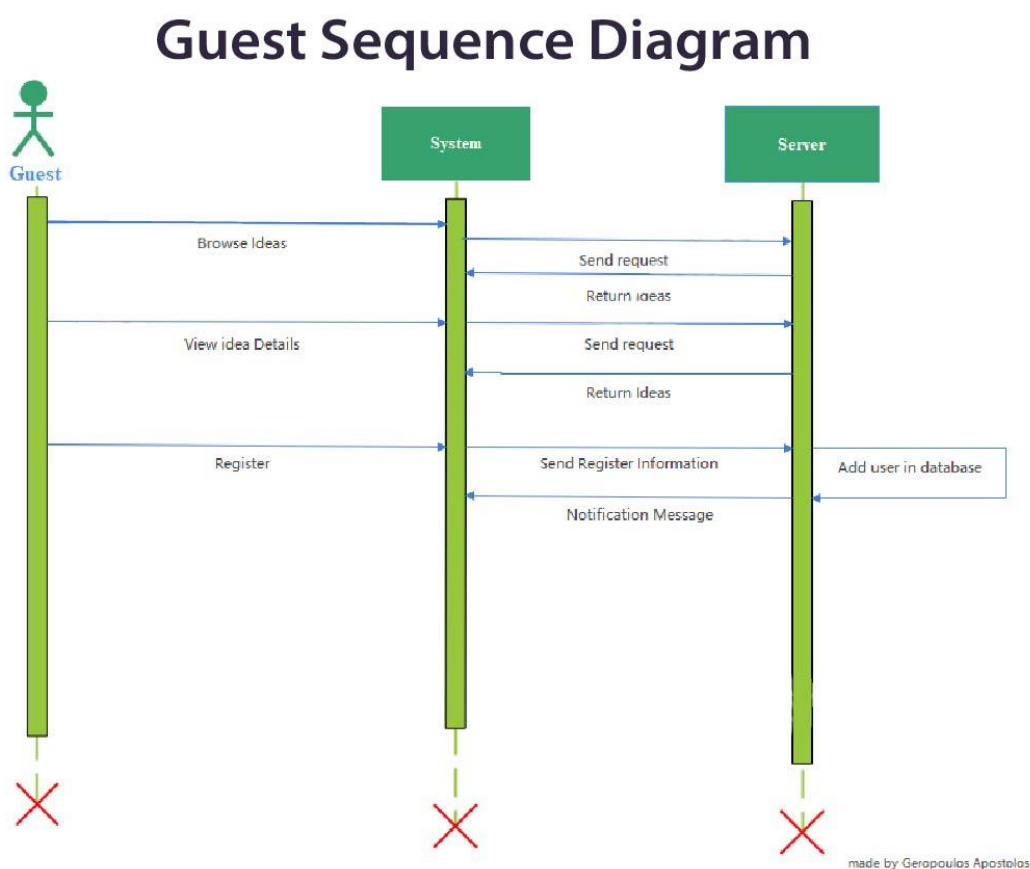
4. **Έγκριση Ιδέας (Approve an Idea):**

Ο διαχειριστής μπορεί να εγκρίνει μια ιδέα. Το σύστημα στέλνει αίτημα ενημέρωσης στη βάση δεδομένων μέσω του διακομιστή, ο οποίος ενημερώνει τα δεδομένα και επιστρέφει μήνυμα ειδοποίησης για την επιτυχή ολοκλήρωση.

5. **Απόρριψη Ιδέας (Reject an Idea):**

Ο διαχειριστής μπορεί να απορρίψει μια ιδέα. Το σύστημα διαγράφει τα δεδομένα της ιδέας από τη βάση δεδομένων μέσω του διακομιστή και αποστέλλεται ειδοποίηση για την απόρριψη.

Το διάγραμμα αυτό αναδεικνύει την κεντρική διαχείριση από τον διαχειριστή, καθώς και τη συνεργασία μεταξύ του συστήματος και του διακομιστή για την επεξεργασία δεδομένων και τη λήψη αποφάσεων.



Εικόνα 2: Guest Sequence Diagram

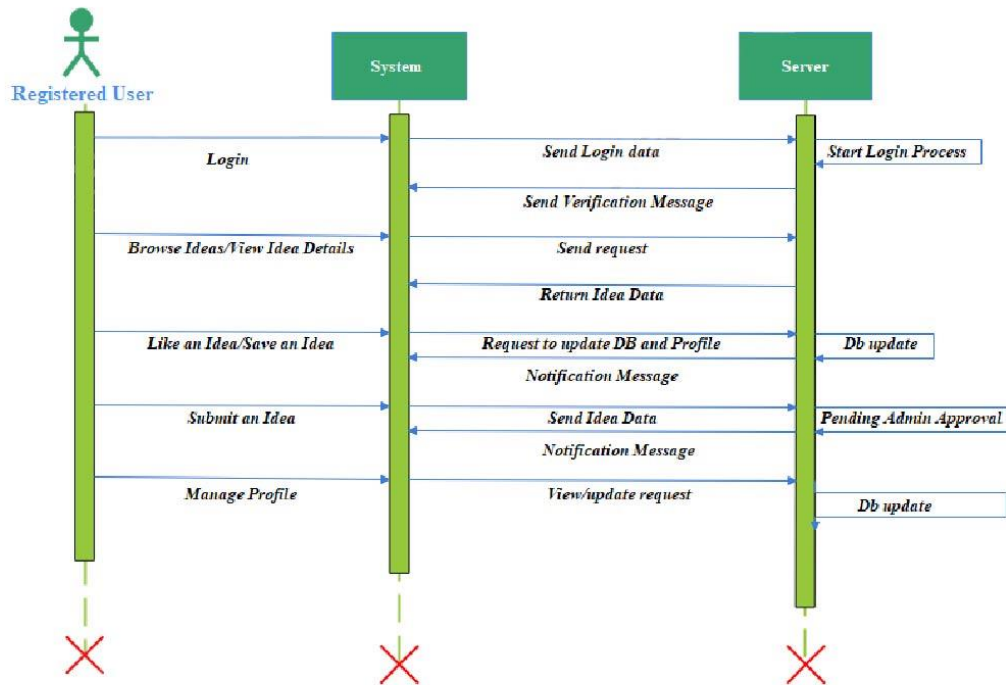
4.4.2 Ανάλυση για το Guest User Sequence Diagram

Το Guest User Sequence Diagram απεικονίζει τις λειτουργίες που είναι διαθέσιμες στους επισκέπτες της πλατφόρμας, δηλαδή τους χρήστες που δεν έχουν δημιουργήσει ακόμα λογαριασμό.

1. Περιήγηση Ιδεών (Browse Ideas):
2. Ο επισκέπτης μπορεί να περιηγηθεί στις ιδέες που έχουν δημοσιευτεί. Το σύστημα στέλνει αίτημα στον διακομιστή για τις διαθέσιμες ιδέες, ο οποίος επιστρέφει τα δεδομένα στο σύστημα και στη συνέχεια στον επισκέπτη.
3. Προβολή Λεπτομερειών Ιδέας (View Idea Details):
4. Ο επισκέπτης έχει τη δυνατότητα να δει περισσότερες λεπτομέρειες για μια συγκεκριμένη ιδέα. Το σύστημα στέλνει αίτημα στον διακομιστή για την παροχή των σχετικών δεδομένων, τα οποία επιστρέφονται στον χρήστη.
5. Εγγραφή (Register):
6. Ο επισκέπτης μπορεί να δημιουργήσει λογαριασμό. Το σύστημα στέλνει τις πληροφορίες εγγραφής στον διακομιστή, ο οποίος τις αποθηκεύει στη βάση δεδομένων. Μετά την ολοκλήρωση, ο διακομιστής στέλνει ένα μήνυμα ειδοποίησης για την επιτυχή εγγραφή.

Το διάγραμμα αυτό τονίζει τις περιορισμένες αλλά σημαντικές λειτουργίες που είναι διαθέσιμες στους επισκέπτες, παρέχοντας τους τη δυνατότητα να περιηγηθούν στο περιεχόμενο της πλατφόρμας και να εγγραφούν για περισσότερες δυνατότητες.

REGISTERED USER SEQUENCE DIAGRAM



Εικόνα 3: Registered User Sequence Diagram

4.4.3 Ανάλυση για το Registered User Sequence Diagram

Το Registered User Sequence Diagram περιγράφει τις λειτουργίες που είναι διαθέσιμες στους εγγεγραμμένους χρήστες της πλατφόρμας.

1. Σύνδεση (Login):

Ο εγγεγραμμένος χρήστης ξεκινά τη διαδικασία σύνδεσης εισάγοντας τα διαπιστευτήριά του. Το σύστημα στέλνει τα δεδομένα σύνδεσης στον διακομιστή, ο οποίος επαληθεύει την ταυτότητα του χρήστη. Μετά την επαλήθευση, ο χρήστης αποκτά πρόσβαση στις διαθέσιμες λειτουργίες.

2. Περιήγηση Ιδεών / Προβολή Λεπτομερειών Ιδεών (Browse Ideas/View Idea Details):

Ο χρήστης μπορεί να περιηγηθεί στις διαθέσιμες ιδέες και να δει λεπτομέρειες για αυτές. Το σύστημα στέλνει αιτήματα στον διακομιστή, ο οποίος επιστρέφει τα σχετικά δεδομένα.

3. Αποθήκευση ή "Μου Αρέσει" σε Ιδέα (Like an Idea/Save an Idea):

Ο χρήστης έχει τη δυνατότητα να αποθηκεύσει ή να δηλώσει ότι του αρέσει μια ιδέα. Το σύστημα στέλνει αίτημα ενημέρωσης στη βάση δεδομένων μέσω του διακομιστή, ο οποίος επιστρέφει μήνυμα επιβεβαίωσης.

4. Υποβολή Ιδέας (Submit an Idea):

Ο εγγεγραμμένος χρήστης μπορεί να υποβάλει μια ιδέα. Το σύστημα στέλνει τα δεδομένα της ιδέας στον διακομιστή, ο οποίος την αποθηκεύει στη βάση δεδομένων και αποστέλλει μήνυμα επιβεβαίωσης.

5. Διαχείριση Προφίλ (Manage Profile):

Ο χρήστης μπορεί να τροποποιήσει τις πληροφορίες του προφίλ του. Το σύστημα στέλνει αίτημα ενημέρωσης στον διακομιστή, ο οποίος ενημερώνει τη βάση δεδομένων και στέλνει μήνυμα επιβεβαίωσης.

Το διάγραμμα αυτό δείχνει την πλήρη γκάμα λειτουργιών που είναι διαθέσιμες στους εγγεγραμμένους χρήστες, τονίζοντας την αλληλεπίδραση με το σύστημα και τον διακομιστή για τη διαχείριση δεδομένων, την περιήγηση και την υποβολή ιδεών.

4.5 Επίλογος

Στο παρόν κεφάλαιο παρουσιάστηκαν και αναλύθηκαν τα διαγράμματα ακολουθίας (sequence diagrams) που περιγράφουν τις βασικές λειτουργίες και τις αλληλεπιδράσεις των τριών κύριων κατηγοριών χρηστών με το σύστημα: του διαχειριστή (Admin), του επισκέπτη (Guest User) και του εγγεγραμμένου χρήστη (Registered User). Μέσα από τα διαγράμματα αυτά αναδείχθηκε η λειτουργική υποδομή του συστήματος και η ροή των δεδομένων από και προς τους χρήστες, το σύστημα και τον διακομιστή.

Αρχικά, στο διάγραμμα του διαχειριστή, παρουσιάστηκαν οι ενέργειες διαχείρισης που περιλαμβάνουν τη σύνδεση, την περιήγηση στις ιδέες, την αξιολόγηση εκκρεμών ιδεών και την έγκριση ή απόρριψή τους. Ο διαχειριστής έχει κεντρικό ρόλο στη λειτουργία της πλατφόρμας, εξασφαλίζοντας την ποιότητα και την εγκυρότητα του περιεχομένου.

Στο διάγραμμα του επισκέπτη, αναλύθηκαν οι διαδικασίες περιήγησης και εγγραφής, όπου ο επισκέπτης μπορεί να εξερευνήσει τις ιδέες της πλατφόρμας πριν αποφασίσει να εγγραφεί ως χρήστης. Η λειτουργία αυτή είναι κρίσιμη για την ενθάρρυνση νέων χρηστών να γίνουν μέλη της κοινότητας.

Στο διάγραμμα του εγγεγραμμένου χρήστη, αναλύθηκαν ενέργειες όπως η περιήγηση, η αποθήκευση ιδεών, η υποβολή νέων προτάσεων και η διαχείριση του προσωπικού προφίλ. Η πλατφόρμα εξασφαλίζει την εξατομικευμένη εμπειρία του χρήστη, προσφέροντας εργαλεία για ενεργή συμμετοχή και συμβολή στο περιεχόμενο.

Η ανάλυση αυτών των διαγραμμάτων προσφέρει μια ολοκληρωμένη εικόνα της λειτουργικότητας του συστήματος, αναδεικνύοντας τη σημασία της σωστής διαχείρισης των διαδικασιών και της ροής δεδομένων. Παράλληλα, επιτρέπει τη βαθύτερη κατανόηση της αρχιτεκτονικής του συστήματος, θέτοντας τις βάσεις για μελλοντική ανάπτυξη και βελτιώσεις που θα ανταποκρίνονται στις ανάγκες των χρηστών.

Κεφάλαιο 5ο: Ανάπτυξη εφαρμογής

5.1 Εισαγωγή

Στο παρόν κεφάλαιο παρουσιάζεται η διαδικασία ανάπτυξης της εφαρμογής που σχεδιάστηκε και υλοποιήθηκε στο πλαίσιο της διπλωματικής εργασίας. Η εφαρμογή συνιστά ένα ολοκληρωμένο σύστημα που αξιοποιεί σύγχρονες τεχνολογίες για την επίτευξη των στόχων της. Η ανάπτυξή της στηρίχθηκε στην πλατφόρμα Laravel, ένα από τα πιο διαδεδομένα και ισχυρά frameworks για την ανάπτυξη web εφαρμογών, σε συνδυασμό με τη βάση δεδομένων MySQL.

Η υλοποίηση της εφαρμογής περιλαμβάνει διάφορες διακριτές φάσεις, οι οποίες καλύπτονται λεπτομερώς σε αυτό το κεφάλαιο. Αρχικά, δίνεται έμφαση στη σχεδίαση και διαχείριση της βάσης δεδομένων. Μέσα από τη χρήση της MySQL, εξασφαλίστηκε η αποθήκευση και διαχείριση των δεδομένων με τρόπο που εγγυάται αποδοτικότητα και αξιοπιστία. Παρουσιάζονται οι βασικοί πίνακες, τα σχήματα και οι σχέσεις που ορίστηκαν, καθώς και τα εργαλεία που χρησιμοποιήθηκαν για τη σύνδεση της βάσης δεδομένων με την εφαρμογή.

Στη συνέχεια, περιγράφεται η διαδικασία ανάπτυξης του κώδικα με τη χρήση του Laravel, εστιάζοντας στις τεχνολογίες, τις μεθόδους και τις αρχές σχεδιασμού που ακολουθήθηκαν. Έμφαση δίνεται στην υλοποίηση του MVC (Model-View-Controller) μοντέλου, καθώς και στον τρόπο που ενσωματώθηκαν τα διάφορα στοιχεία του backend και του frontend.

Επιπλέον, αναλύεται η χρήση των Blade templates, μιας πανίσχυρης λειτουργίας του Laravel για τη δημιουργία δυναμικών και προσαρμόσιμων διεπαφών χρήστη. Εξηγείται ο τρόπος υλοποίησης των templates και ο ρόλος τους στην ενοποίηση της εφαρμογής.

Το κεφάλαιο αυτό αποσκοπεί στην παροχή μιας πλήρους εικόνας της διαδικασίας ανάπτυξης της εφαρμογής, παρέχοντας τεχνικές λεπτομέρειες που καταδεικνύουν την οργάνωση και την καινοτομία που υιοθετήθηκαν.

5.2 Βάσεις Δεδομένων

Η βάση δεδομένων αποτελεί τον πυρήνα της εφαρμογής, καθώς εξασφαλίζει την αποθήκευση, διαχείριση και ανάκτηση των δεδομένων με τρόπο αξιόπιστο και αποδοτικό. Για την υλοποίηση της εφαρμογής, επιλέχθηκε η **MySQL**, ένα από τα πιο δημοφιλή συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS), λόγω της σταθερότητας, της ευελιξίας και της ευρείας υποστήριξής της. Για την υλοποίηση των βάσεων αρχικά πρέπει να έχει γίνει η εγκατάσταση του Xampp όπως έχει προαναφερθεί σε προηγούμενο κεφάλαιο της διπλωματικής. Με την βοήθεια το Xampp – Control Panel δίνεται η δυνατότητα για διαχείριση των βάσεων μέσω του phpMyAdmin όπου θα χρησιμοποιηθεί για να γίνει η δημιουργία της βάσης δεδομένων μας αλλά και η διαχείριση της. Για τις ανάγκες του προγράμματος η βάση δεδομένων αρχικοποιήθηκε με τους εξής πίνακες :

Κεφάλαιο 4

- Cache
- Cache_locks
- Idea-tags
- Ideas
- Migrations
- Password-reset-tokens
- Sessions
- Tags
- users

Table	Action	Rows	Type	Collation	Size	Overhead
cache	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
cache_locks	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
ideas	★ Browse Structure Search Insert Empty Drop	18	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
idea_tags	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
likes	★ Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
migrations	★ Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
password_reset_tokens	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
sessions	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
tags	★ Browse Structure Search Insert Empty Drop	20	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
users	★ Browse Structure Search Insert Empty Drop	15	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
10 tables	Sum	73	InnoDB	utf8mb4_general_ci	256.0 KiB	0 B

Εικόνα 4: Στιγμιότυπο Βάσης Δεδομένων

5.2.1 Ανάλυση πίνακα Cache

Ο πίνακας cache έχει δημιουργηθεί για να χρησιμοποιηθεί για την αποθήκευση προσωρινών δεδομένων (caching) που βελτιώνουν την απόδοση της εφαρμογής. Περιέχει δεδομένα που μπορούν να ανακτηθούν πιο άμεσα χωρίς να γίνουν νέες ερωτήσεις στην βάση των δεδομένων μας. Βέβαια στην συγκεκριμένη εφαρμογή δεν έχει μεγάλη χρησιμότητα όμως στα πλαίσια της δημιουργίας μιας πλατφόρμας όπου θα γίνεται εύκολη η μετατροπή σε διαφορετικές πλατφόρμες, δημιουργήθηκε για να δώσει τις δυνατότητες σε μελλοντικές μετατροπές να κάνουν χρήση σε περίπτωση που κρίνεται αναγκαίο.

Η δομή του πίνακα Cache έχει την παρακάτω μορφή :

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	key	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
2	value	mediumtext	utf8mb4_unicode_ci		No	None			Change Drop More
3	expiration	int(11)			No	None			Change Drop More

Εικόνα 5: Στιγμιότυπο Πίνακα Cache

Το key αποτελεί το κύριο κλειδί (primary key) του πίνακα και είναι μορφή χαρακτήρων. Ενώ τα δεδομένα που θέλουμε να αποθηκεύσουμε βρίσκονται στο value (τύπου: medium text) και τέλος μια μεταβλητή expiration (τύπου: integer) που ο στόχος είναι η αποθήκευση μιας χρονικής σφραγίδας (timestamp) κατά την οποία το cache entry θα λήγει ή θα παύει να ισχύει για να μπορεί να καθαρίζει τον πίνακα μας χωρίς να διατηρεί στοιχεία που δεν είναι πλέον χρήσιμα.

5.2.2 Ανάλυση πίνακα Cache locks

Ο πίνακας `cache_locks` διαδραματίζει σημαντικό ρόλο στη διαχείριση συγχρονισμού δεδομένων, διασφαλίζοντας την ασφαλή πρόσβαση σε πόρους ή λειτουργίες της εφαρμογής όταν αυτή χρησιμοποιείται από πολλαπλούς χρήστες ή διεργασίες. Κύριος στόχος του είναι να αποτρέψει τις συγκρούσεις που μπορεί να προκύψουν κατά την ταυτόχρονη πρόσβαση σε κοινόχρηστα δεδομένα.

Η δομή του πίνακα περιλαμβάνει τρεις κύριες στήλες. Η πρώτη στήλη, με την ονομασία `key`, αποτελεί το πρωτεύον κλειδί του πίνακα και χρησιμεύει για τη μοναδική αναγνώριση κάθε εγγραφής `lock`. Στην ουσία, το `key` αντιπροσωπεύει το αναγνωριστικό ενός συγκεκριμένου πόρου ή μίας λειτουργίας που έχει «κλειδωθεί», ώστε να αποτραπεί η πρόσβαση από άλλους χρήστες ή διεργασίες.

Η δεύτερη στήλη, με την ονομασία `owner`, καταγράφει τον ιδιοκτήτη του `lock`, δηλαδή τον χρήστη, τη διαδικασία ή την υπηρεσία που το έχει αποκτήσει. Αυτή η πληροφορία είναι χρήσιμη, καθώς επιτρέπει την αναγνώριση του ποιος διαχειρίζεται τον συγκεκριμένο πόρο, ενισχύοντας την ανιχνευσιμότητα στη διαχείριση των δεδομένων.

Τέλος, η τρίτη στήλη, `expiration`, αποθηκεύει την χρονική στιγμή λήξης του `lock` σε μορφή UNIX `timestamp`, δηλαδή ως αριθμό δευτερολέπτων από την 1η Ιανουαρίου 1970. Η χρήση του πεδίου αυτού είναι ζωτικής σημασίας για την αποφυγή της διατήρησης κλειδώματος επ' αόριστον. Εφόσον το χρονικό όριο που ορίζεται από το `expiration` ξεπεραστεί, το `lock` θεωρείται παρωχημένο και μπορεί να αφαιρεθεί, επιτρέποντας έτσι σε άλλες διεργασίες ή χρήστες να αποκτήσουν πρόσβαση στον ίδιο πόρο.

Η λειτουργία του πίνακα `cache_locks` είναι απαραίτητη για την αποφυγή φαινομένων `deadlock`, όπου οι πόροι της εφαρμογής παραμένουν κλειδωμένοι λόγω αποτυχημένων ή μη ολοκληρωμένων διεργασιών. Σε περιπτώσεις μεγάλης ταυτόχρονης δραστηριότητας, αυτός ο πίνακας διασφαλίζει τη σωστή λειτουργία του συστήματος, επιτρέποντας την ασφαλή και οργανωμένη πρόσβαση σε δεδομένα, χωρίς να δημιουργούνται συγκρούσεις ή καθυστερήσεις. Η δομή του πίνακα φαίνεται παρακάτω :

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 key	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	2 owner	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	3 expiration	int(11)			No	None			Change Drop More

Εικόνα 6: Στιγμιότυπο Πίνακα Cache

5.2.3 Ανάλυση πίνακα Idea_tags

Ο πίνακας `idea_tags` αποτελεί έναν πίνακα συσχέτισης, σχεδιασμένο να διαχειρίζεται τη σύνδεση μεταξύ ιδεών και ετικετών. Πρόκειται για μια κλασική υλοποίηση σχέσης πολλών-προς-πολλούς (`many-to-many`), όπου μια ιδέα μπορεί να σχετίζεται με πολλές ετικέτες, ενώ παράλληλα μια ετικέτα μπορεί να σχετίζεται με πολλές ιδέες. Η δομή του πίνακα είναι απλή αλλά απαραίτητη για την κατηγοριοποίηση και την ευκολότερη αναζήτηση περιεχομένου.

Ο πίνακας περιλαμβάνει τις εξής στήλες:

- **idea_id**: Αυτή η στήλη αποθηκεύει το μοναδικό αναγνωριστικό μιας ιδέας, όπως ορίζεται στον πίνακα ideas. Το πεδίο αυτό είναι τύπου bigint, υποδηλώνοντας ότι μπορεί να αποθηκεύσει μεγάλες αριθμητικές τιμές. Λειτουργεί ως ξένο κλειδί (foreign key) προς τον πίνακα ideas, εξασφαλίζοντας ότι κάθε καταχώριση αναφέρεται σε μια έγκυρη ιδέα. Η σύνδεση αυτή είναι σημαντική για τη διατήρηση της ακεραιότητας των δεδομένων.
- **tag_id**: Αυτή η στήλη αποθηκεύει το μοναδικό αναγνωριστικό μιας ετικέτας, όπως ορίζεται στον πίνακα tags. Όπως και το idea_id, είναι τύπου bigint και λειτουργεί ως ξένο κλειδί (foreign key) προς τον πίνακα tags. Η χρήση της εξασφαλίζει ότι κάθε καταχώριση στον πίνακα idea_tags αναφέρεται σε μια έγκυρη ετικέτα.

Η συνδυαστική χρήση των πεδίων idea_id και tag_id ως primary key διασφαλίζει ότι δεν θα υπάρξει διπλή εγγραφή για την ίδια συσχέτιση, ενισχύοντας έτσι την αποδοτικότητα και την ακεραιότητα της βάσης δεδομένων.

Ο συγκεκριμένος πίνακας καθιστά δυνατή την κατηγοριοποίηση των ιδεών με βάση τις σχετικές ετικέτες, επιτρέποντας πιο στοχευμένες αναζητήσεις και φιλτραρίσματα. Η δομή του πίνακα είναι η παρακάτω:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 idea_id	bigint(20)		UNSIGNED	No	None			Change Drop More
<input type="checkbox"/>	2 tag_id	bigint(20)		UNSIGNED	No	None			Change Drop More

Εικόνα 7:Στιγμιότυπο Πίνακα Idea Tags

5.2.4 Ανάλυση πίνακα Ideas

Ο πίνακας ideas είναι ένας από τους κεντρικούς πίνακες της βάσης δεδομένων και έχει σχεδιαστεί για την αποθήκευση και διαχείριση των ιδεών που υποβάλλονται από τους χρήστες. Ο πίνακας αυτός παρέχει όλες τις απαραίτητες πληροφορίες για κάθε ιδέα, όπως τον τίτλο, το περιεχόμενο, την κατάσταση και επιπλέον μεταδεδομένα.

Οι στήλες που περιλαμβάνονται στον πίνακα είναι οι εξής:

- **id**: Το id είναι το μοναδικό αναγνωριστικό κάθε ιδέας, το οποίο χρησιμοποιείται ως primary key. Είναι τύπου bigint και ορίζεται ως auto_increment, διασφαλίζοντας ότι κάθε ιδέα θα έχει μοναδική και αυτόματα αυξανόμενη τιμή. Αυτό το πεδίο αποτελεί τη βάση για τη σύνδεση με άλλους πίνακες που σχετίζονται με τις ιδέες, όπως ο idea_tags.
- **userId**: Αυτή η στήλη αποθηκεύει το μοναδικό αναγνωριστικό του χρήστη που δημιούργησε την ιδέα. Είναι τύπου bigint και λειτουργεί ως ξένο κλειδί προς τον πίνακα users. Η χρήση αυτού του πεδίου καθιστά δυνατή την αντιστοίχιση κάθε ιδέας με τον δημιουργό της, διευκολύνοντας τη διαχείριση των δικαιωμάτων πρόσβασης και την παρακολούθηση της δραστηριότητας του χρήστη.
- **title**: Η στήλη title περιέχει τον τίτλο της ιδέας. Είναι τύπου varchar με μέγιστο μήκος 255 χαρακτήρες. Ο τίτλος είναι μια σύντομη περιγραφή που βοηθά στην αναγνώριση της ιδέας από τον χρήστη ή τον διαχειριστή.

- **content:** Το content είναι τύπου text και περιέχει την περιγραφή ή τις λεπτομέρειες της ιδέας. Σε αντίθεση με τον τίτλο, αυτή η στήλη μπορεί να περιέχει μεγαλύτερο όγκο δεδομένων, επιτρέποντας την αναλυτική παρουσίαση της ιδέας.
- **status:** Η στήλη status είναι τύπου enum και περιλαμβάνει προκαθορισμένες τιμές όπως "pending", "active" και "declined". Η τιμή αυτή αποθηκεύει την κατάσταση της ιδέας, υποδεικνύοντας αν η ιδέα βρίσκεται υπό αξιολόγηση, έχει εγκριθεί ή έχει απορριφθεί.
- **created_at** και **updated_at:** Αυτά τα πεδία είναι τύπου timestamp και καταγράφουν τη χρονική στιγμή δημιουργίας και τελευταίας ενημέρωσης της ιδέας, αντίστοιχα. Οι χρονικές σημάνσεις αυτές είναι ιδιαίτερα χρήσιμες για την παρακολούθηση αλλαγών και την αναζήτηση με βάση τον χρόνο.
- **likes:** Η στήλη likes είναι τύπου int και αποθηκεύει τον αριθμό των "likes" που έχει λάβει η ιδέα. Η ύπαρξη αυτού του πεδίου προσθέτει μια κοινωνική διάσταση, επιτρέποντας στους χρήστες να αξιολογούν τις ιδέες.
- **area:** Το area είναι τύπου varchar και αποθηκεύει την περιοχή ή το θέμα στο οποίο ανήκει η ιδέα. Αυτή η στήλη επιτρέπει την καλύτερη ομαδοποίηση και φιλτράρισμα των ιδεών με βάση το περιεχόμενό τους.

Ο πίνακας ideas είναι βασικός για την αποθήκευση του περιεχομένου της εφαρμογής, καθώς κάθε εγγραφή αντιπροσωπεύει μια μοναδική ιδέα.

5.2.5 Migrations και Ανάλυση Πίνακα migrations

Ο πίνακας migrations αποτελεί έναν από τους πιο κρίσιμους πίνακες στη διαχείριση της υποδομής μιας βάσης δεδομένων, ιδιαίτερα σε περιβάλλοντα όπου η ανάπτυξη ακολουθεί το μοντέλο DevOps ή CI/CD (Continuous Integration/Continuous Deployment). Ο ρόλος του είναι να καταγράφει όλες τις αλλαγές (μεταναστεύσεις) που έχουν πραγματοποιηθεί στη δομή της βάσης δεδομένων, εξασφαλίζοντας ότι η ανάπτυξη και η συντήρηση της βάσης δεδομένων είναι συνεπής μεταξύ των διαφορετικών περιβαλλόντων (π.χ. ανάπτυξη, staging, παραγωγή) [18].

Στήλες του Πίνακα migrations:

1. **id**
 - **Τύπος:** bigint
 - **Περιγραφή:** Το πεδίο id είναι το μοναδικό αναγνωριστικό κάθε εγγραφής. Είναι auto_increment, το οποίο σημαίνει ότι η τιμή του αυξάνεται αυτόματα κάθε φορά που προστίθεται μια νέα εγγραφή στον πίνακα.
 - **Χρήση:** Απλοποιεί την αναζήτηση και την παρακολούθηση συγκεκριμένων μεταναστεύσεων.
2. **migration**
 - **Τύπος:** varchar(255)
 - **Περιγραφή:** Το πεδίο αυτό αποθηκεύει το όνομα του αρχείου μετανάστευσης. Συνήθως, το όνομα του αρχείου ακολουθεί μια προκαθορισμένη δομή, όπως η

ημερομηνία και η ώρα δημιουργίας, για να διασφαλίζεται η χρονολογική σειρά εκτέλεσης των μεταναστεύσεων (π.χ. 202501010101_create_users_table).

- **Χρήση:** Επιτρέπει στον μηχανισμό μετανάστευσης να γνωρίζει ποιες αλλαγές έχουν ήδη εφαρμοστεί στη βάση δεδομένων, ώστε να αποφεύγεται η επανεκτέλεση.

3. batch

- **Τύπος:** int
- **Περιγραφή:** Το πεδίο batch αποθηκεύει τον αριθμό της παρτίδας (batch) στην οποία ανήκει η συγκεκριμένη μετανάστευση. Κάθε παρτίδα αντιπροσωπεύει μια ομάδα μεταναστεύσεων που εκτελούνται μαζί. Για παράδειγμα, αν σε μια ανάπτυξη εκτελεστούν τρεις μεταναστεύσεις, όλες αυτές θα έχουν τον ίδιο αριθμό παρτίδας.
- **Χρήση:** Χρησιμοποιείται για την ευκολότερη διαχείριση της αναίρεσης μεταναστεύσεων (rollback), καθώς επιτρέπει την αναίρεση ολόκληρης της παρτίδας με μία εντολή.

Η δομή του πίνακα migrations είναι η παρακάτω:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 migration	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	3 batch	int(11)			No	None			Change Drop More

Εικόνα 8: Στιγμιότυπο Πίνακα migrations

Ρόλος και Λειτουργία

Ο πίνακας migrations χρησιμοποιείται από το σύστημα μετανάστευσης για να καταγράφει ποια scripts έχουν ήδη εκτελεστεί. Οι βασικές λειτουργίες του πίνακα περιλαμβάνουν:

1. **Καταγραφή Εκτελεσμένων Scripts**
Όταν εκτελείται ένα script μετανάστευσης, το όνομά του (που περιγράφεται στη στήλη migration) καταγράφεται στον πίνακα. Αυτό επιτρέπει στο σύστημα να γνωρίζει ότι οι συγκεκριμένες αλλαγές έχουν ήδη εφαρμοστεί.
2. **Αποφυγή Επαναληπτικών Ενεργειών**
Πριν εκτελεστεί μια νέα μετανάστευση, το σύστημα ελέγχει τον πίνακα migrations. Αν το όνομα του script υπάρχει ήδη, παρακάμπτει την εκτέλεση.
3. **Rollback (Αναίρεση Μεταναστεύσεων)**
Ο αριθμός παρτίδας (batch) διευκολύνει την αναίρεση των αλλαγών. Για παράδειγμα, αν πρέπει να αναιρεθούν οι αλλαγές μιας συγκεκριμένης ανάπτυξης, όλα τα scripts της ίδιας παρτίδας μπορούν να αναιρεθούν με μία εντολή.
4. **Διαχείριση Συνεχούς Εξέλιξης**
Ο πίνακας διασφαλίζει ότι η βάση δεδομένων παραμένει συνεπής μεταξύ διαφορετικών περιβαλλόντων. Όταν οι μεταναστεύσεις μεταφέρονται από περιβάλλον ανάπτυξης σε staging ή παραγωγή, το σύστημα εκτελεί μόνο τις νέες μεταναστεύσεις που δεν υπάρχουν στον πίνακα.

Παράδειγμα Χρήσης

Ας υποθέσουμε ότι έχουμε τρία scripts μετανάστευσης:

1. 202501010101_create_users_table
2. 202501020202_add_status_to_users
3. 202501030303_create_ideas_table

Όταν εκτελεστούν αυτά τα scripts, ο πίνακας migrations θα περιέχει:

id migration	batch
1 202501010101_create_users_table	1
2 202501020202_add_status_to_users	1
3 202501030303_create_ideas_table	1

Αν στη συνέχεια προστεθούν δύο νέα scripts (202501040404_add_likes_to_ideas και 202501050505_create_tags_table) και εκτελεστούν, ο πίνακας θα γίνει:

id migration	batch
1 202501010101_create_users_table	1
2 202501020202_add_status_to_users	1
3 202501030303_create_ideas_table	1
4 202501040404_add_likes_to_ideas	2
5 202501050505_create_tags_table	2

Αυτό σημαίνει ότι οι πρώτες τρεις μεταναστεύσεις ανήκουν στην παρτίδα 1, ενώ οι δύο επόμενες στην παρτίδα 2.

Πλεονεκτήματα του Πίνακα migrations

- **Αυτοματοποίηση Διαδικασιών:** Ο πίνακας επιτρέπει στο σύστημα να εκτελεί και να αναίρει μεταναστεύσεις αυτόματα.
- **Ιχνηλασιμότητα:** Καταγράφει λεπτομερώς ποια scripts έχουν εκτελεστεί και πότε.
- **Ακεραιότητα Δεδομένων:** Εξασφαλίζει ότι όλες οι αλλαγές στη δομή της βάσης δεδομένων είναι συνεπείς μεταξύ διαφορετικών περιβαλλόντων.
- **Ευελξία Rollback:** Η ύπαρξη του πεδίου batch επιτρέπει την αναίρεση συγκεκριμένων παρτίδων.

5.2.6 Ανάλυση πίνακα Users

Ο πίνακας **users** αποτελεί τον βασικό πίνακα για την αποθήκευση πληροφοριών των χρηστών του συστήματος. Περιλαμβάνει τις απαραίτητες στήλες για την αναγνώριση και τη διαχείριση των χρηστών. Η πρώτη στήλη, **id**, είναι ένας μοναδικός αριθμητικός δείκτης τύπου **bigint**, ο οποίος λειτουργεί ως το κύριο κλειδί του πίνακα και διασφαλίζει την ταυτοποίηση κάθε χρήστη. Η στήλη **name** αποθηκεύει το όνομα του χρήστη σε μορφή κειμένου με μέγιστο μήκος 255 χαρακτήρες, ενώ η

στήλη **email** καταγράφει τη διεύθυνση email κάθε χρήστη, η οποία είναι μοναδική για την αποφυγή διπλότυπων εγγραφών από ίδιο χρήστη. Η στήλη **password** περιέχει τον κωδικό πρόσβασης του χρήστη σε κρυπτογραφημένη μορφή, διασφαλίζοντας έτσι την προστασία των δεδομένων. Τέλος, οι στήλες **created_at** και **updated_at** αποθηκεύουν χρονικές ενδείξεις που δηλώνουν πότε δημιουργήθηκε και πότε ενημερώθηκε για τελευταία φορά η εγγραφή, αντίστοιχα. Η δομή του πίνακα `users` είναι η παρακάτω:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 name	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	3 email	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	4 email_verified_at	timestamp			Yes	NULL			Change Drop More
<input type="checkbox"/>	5 password	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	6 remember_token	varchar(100)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	7 created_at	timestamp			Yes	NULL			Change Drop More
<input type="checkbox"/>	8 updated_at	timestamp			Yes	NULL			Change Drop More

Εικόνα 9: Στιγμιότυπο Πίνακα Users

5.2.7 Ανάλυση πίνακα tags

Ο πίνακας `tags` έχει σχεδιαστεί για να διαχειρίζεται ετικέτες που μπορούν να συσχετιστούν με άλλα δεδομένα, όπως ιδέες ή περιεχόμενο. Περιλαμβάνει μια στήλη `id`, η οποία αποτελεί το κύριο κλειδί του πίνακα και είναι τύπου `bigint`. Η στήλη `tagName` αποθηκεύει το όνομα της ετικέτας, που μπορεί να περιέχει έως 255 χαρακτήρες. Επιπλέον, οι στήλες `created_at` και `updated_at` χρησιμοποιούνται για την καταγραφή της ημερομηνίας και ώρας δημιουργίας και τελευταίας ενημέρωσης της κάθε εγγραφής.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 tagName	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	3 created_at	timestamp			Yes	NULL			Change Drop More
<input type="checkbox"/>	4 updated_at	timestamp			Yes	NULL			Change Drop More

Εικόνα 10: Στιγμιότυπο Πίνακα tags

5.2.8 Ανάλυση πίνακα password_resets

Ο πίνακας `password_resets` έχει σχεδιαστεί για τη διαχείριση αιτημάτων επαναφοράς κωδικών πρόσβασης. Η στήλη `email` αποθηκεύει τη διεύθυνση email του χρήστη που έχει ζητήσει επαναφορά κωδικού. Είναι τύπου `varchar` με μέγιστο μήκος 255 χαρακτήρες και χρησιμοποιείται για την ταυτοποίηση του χρήστη. Η στήλη `token` καταγράφει έναν μοναδικό κωδικό επαναφοράς κωδικού, ο οποίος δημιουργείται δυναμικά και χρησιμεύει για την επιβεβαίωση της διαδικασίας. Είναι επίσης τύπου `varchar` με μέγιστο μήκος 255 χαρακτήρες. Τέλος, η στήλη `created_at` αποθηκεύει την ημερομηνία και ώρα δημιουργίας του αιτήματος, ώστε να ελέγχεται η χρονική ισχύς του κωδικού επαναφοράς. Η δομή του πίνακα `password_resets` είναι η παρακάτω:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 email	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	2 token	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	3 created_at	timestamp			Yes	NULL			Change Drop More

Εικόνα 11: Στιγμιότυπο Πίνακα password_resets

5.2.9 Ανάλυση πίνακα sessions

Ο πίνακας **sessions** καταγράφει τις ενεργές ή πρόσφατες συνδέσεις χρηστών. Η στήλη **id**, τύπου **varchar**, είναι το κύριο κλειδί και αποθηκεύει ένα μοναδικό αναγνωριστικό για κάθε συνεδρία. Η στήλη **user_id** αναφέρεται στον πίνακα **users** και δηλώνει σε ποιον χρήστη αντιστοιχεί η συνεδρία. Η στήλη **ip_address** αποθηκεύει τη διεύθυνση IP από την οποία έγινε η σύνδεση, ενώ η στήλη **user_agent** περιέχει πληροφορίες σχετικά με το πρόγραμμα περιήγησης ή τη συσκευή που χρησιμοποιήθηκε. Η στήλη **payload** είναι τύπου **longtext** και χρησιμοποιείται για την αποθήκευση δεδομένων σχετικών με τη συνεδρία, όπως προσωρινές ρυθμίσεις ή πληροφορίες κατάστασης. Τέλος, η στήλη **last_activity** καταγράφει την τελευταία δραστηριότητα του χρήστη στη συνεδρία, σε μορφή αριθμού που αντιπροσωπεύει την ημερομηνία και ώρα. Η δομή του πίνακα **sessions** είναι η παρακάτω:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	2 user_id	bigint(20)		UNSIGNED	Yes	NULL			Change Drop More
<input type="checkbox"/>	3 ip_address	varchar(45)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	4 user_agent	text	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	5 payload	longtext	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	6 last_activity	int(11)			No	None			Change Drop More

Εικόνα 12: Στιγμιότυπο Πίνακα sessions

5.3 Σύνδεση βάση δεδομένων με εφαρμογή

5.3.1 Εισαγωγή

Η πλατφόρμα που αναπτύχθηκε βασίζεται στο Laravel, ένα ισχυρό framework PHP που παρέχει εργαλεία για την ανάπτυξη εφαρμογών ιστού και τη διαχείριση βάσεων δεδομένων. Στη συγκεκριμένη ενότητα περιγράφεται η διαδικασία σύνδεσης της βάσης δεδομένων, που διαχειρίζεται μέσω phpMyAdmin [17], με την πλατφόρμα, καθώς και οι λειτουργίες του Laravel που διευκολύνουν την επικοινωνία μεταξύ εφαρμογής και βάσης δεδομένων.

5.3.2 .env

Το Laravel χρησιμοποιεί το αρχείο περιβάλλοντος **.env** για να αποθηκεύσει βασικές ρυθμίσεις, όπως αυτές της βάσης δεδομένων. Αυτή η προσέγγιση επιτρέπει τη διαχείριση ρυθμίσεων χωρίς να αποκαλύπτονται ευαίσθητα δεδομένα στον πηγαίο κώδικα.

Στο αρχείο **.env** περιλαμβάνονται:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=myprojectdb
DB_USERNAME=root
DB_PASSWORD=

SESSION_DRIVER=database
SESSION_LIFETIME=120
SESSION_ENCRYPT=false
SESSION_PATH=/
SESSION_DOMAIN=null

BROADCAST_CONNECTION=log
FILESYSTEM_DISK=local
QUEUE_CONNECTION=database

CACHE_DRIVER=database
CACHE_PREFIX=
```

Εικόνα 13: Στιγμιότυπο κώδικα από αρχείο .env

Ανάλυση:

DB_CONNECTION: Ο τύπος βάσης δεδομένων που χρησιμοποιείται (MySQL στην προκειμένη περίπτωση).

DB_HOST: Η διεύθυνση του server (συνήθως localhost).

DB_PORT: Η πόρτα του server (προεπιλογή: 3306 για MySQL).

DB_DATABASE: Το όνομα της βάσης δεδομένων.

DB_USERNAME & DB_PASSWORD: Τα διαπιστευτήρια πρόσβασης.

5.3.3 Database.php

Το αρχείο config/database.php περιέχει όλες τις παραμέτρους σύνδεσης που αντλούνται από το .env. Ένα τυπικό απόσπασμα του κώδικα φαίνεται παρακάτω:

```
'mysql' => [
    'driver' => 'mysql',
    'url' => env('DB_URL'),
    'host' => env('DB_HOST', '127.0.0.1'),
    'port' => env('DB_PORT', '3306'),
    'database' => env('DB_DATABASE', 'laravel'),
    'username' => env('DB_USERNAME', 'root'),
    'password' => env('DB_PASSWORD', ''),
    'unix_socket' => env('DB_SOCKET', ''),
    'charset' => env('DB_CHARSET', 'utf8mb4'),
    'collation' => env('DB_COLLATION', 'utf8mb4_unicode_ci'),
    'prefix' => '',
    'prefix_indexes' => true,
    'strict' => true,
    'engine' => null,
    'options' => extension_loaded('pdo_mysql') ? array_filter([
        PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),
    ]) : [],
],
```

Εικόνα 14: Στιγμιότυπο κώδικα Database.php

Ανάλυση Κάθε Παράμετρου :

1. driver

- Τιμή: 'mysql'
- Ρόλος: Ορίζει τον τύπο βάσης δεδομένων που χρησιμοποιείται. Στην περίπτωση αυτή, η εφαρμογή χρησιμοποιεί MySQL, έναν από τους πιο διαδεδομένους τύπους βάσεων δεδομένων.

2. url

- Τιμή: env('DB_URL')
- Ρόλος: Παρέχει τη δυνατότητα να οριστεί μία πλήρης URL σύνδεσης στη βάση δεδομένων. Αν αυτή η τιμή είναι παρούσα, παρακάμπτει τις υπόλοιπες παραμέτρους όπως host, port, database, username και password.

3. host

- Τιμή: env('DB_HOST', '127.0.0.1')
- Ρόλος: Ορίζει τη διεύθυνση του διακομιστή της βάσης δεδομένων. Η προεπιλεγμένη τιμή είναι 127.0.0.1, που αντιστοιχεί στον τοπικό διακομιστή.

4. port

- Τιμή: env('DB_PORT', '3306')
- Ρόλος: Ορίζει την πόρτα του διακομιστή MySQL. Η προεπιλεγμένη τιμή είναι 3306, που αποτελεί την τυπική πόρτα για MySQL.

5. database

- Τιμή: env('DB_DATABASE', 'laravel')
- Ρόλος: Καθορίζει το όνομα της βάσης δεδομένων στην οποία θα συνδεθεί η εφαρμογή.

6. **username**

- Τιμή: env('DB_USERNAME', 'root')
- Ρόλος: Ορίζει το όνομα χρήστη για τη σύνδεση στη βάση δεδομένων.

7. **password**

- Τιμή: env('DB_PASSWORD', '')
- Ρόλος: Καθορίζει τον κωδικό πρόσβασης για τη σύνδεση. Εάν δεν έχει οριστεί, το Laravel αφήνει κενό αυτό το πεδίο.

8. **unix_socket**

- Τιμή: env('DB_SOCKET', '')
- Ρόλος: Παρέχει την επιλογή χρήσης Unix socket για τη σύνδεση, αντί για τον παραδοσιακό τρόπο μέσω host και port.

9. **charset**

- Τιμή: env('DB_CHARSET', 'utf8mb4')
- Ρόλος: Καθορίζει το character set που χρησιμοποιείται στη βάση δεδομένων. Το utf8mb4 υποστηρίζει πλήρη Unicode χαρακτήρες, συμπεριλαμβανομένων emoji.

10. **collation**

- Τιμή: env('DB_COLLATION', 'utf8mb4_unicode_ci')
- Ρόλος: Ορίζει την αντιστοίχιση χαρακτήρων (collation) που χρησιμοποιείται, επηρεάζοντας τη σειρά ταξινόμησης δεδομένων.

11. **prefix**

- Τιμή: ''
- Ρόλος: Εάν οριστεί, προστίθεται ως πρόθεμα (prefix) σε όλα τα ονόματα πινάκων, χρήσιμο όταν διαφορετικές εφαρμογές χρησιμοποιούν την ίδια βάση δεδομένων.

12. **prefix_indexes**

- Τιμή: true
- Ρόλος: Ενεργοποιεί τη δημιουργία index ονομάτων με το πρόθεμα που ορίζεται στο prefix.

13. **strict**

- Τιμή: true
- Ρόλος: Όταν είναι ενεργό, το Laravel επιβάλλει αυστηρούς κανόνες για την εκτέλεση ερωτημάτων, όπως την απαγόρευση μη έγκυρων δεδομένων σε ερωτήματα.

14. **engine**

- Τιμή: null
- Ρόλος: Επιτρέπει τον καθορισμό μηχανής αποθήκευσης MySQL, όπως InnoDB ή MyISAM. Αν παραμείνει null, χρησιμοποιείται η προεπιλεγμένη μηχανή της MySQL.

15. options

- Τιμή:
extension_loaded('pdo_mysql') ? array_filter([PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),]) : []
- Ρόλος: Παρέχει επιπλέον επιλογές για τη σύνδεση, όπως η χρήση SSL για ασφαλή επικοινωνία με τον διακομιστή. Εδώ, ελέγχεται αν είναι ενεργό το extension pdo_mysql.

5.3.4 Migrations

Οι μεταναστεύσεις είναι ένας τύπος ελέγχου έκδοσης για τη βάση δεδομένων σας. Επιτρέπουν σε μια ομάδα να τροποποιεί το σχήμα της βάσης δεδομένων και να παραμένει ενήμερη για την τρέχουσα κατάσταση του σχήματος. Οι μεταναστεύσεις συνήθως συνδυάζονται με το Schema Builder για την εύκολη διαχείριση του σχήματος της εφαρμογής σας. Την λειτουργία και περισσότερες πληροφορίες για τα migrations αναφέραμε στην ενότητα 5.2.5

Παραδείγματα migrations:

- Δημιουργία του πίνακα users:

```
Schema::create('users', function (Blueprint $table) {
    $table->id();
    $table->string('name');
    $table->string('email')->unique();
    $table->timestamp('email_verified_at')->nullable();
    $table->string('password');
    $table->rememberToken();
    $table->timestamps();
});
```

Εικόνα 15: Στιγμιότυπο κώδικα από migrations

Με αυτό τον τρόπο, κάθε αλλαγή στη βάση δεδομένων γίνεται με προγραμματισμένο τρόπο, διασφαλίζοντας την εύκολη επαναδημιουργία της.

5.3.5 Controllers

Οι **Controllers** λειτουργούν ως "διαμεσολαβητές" ανάμεσα στον χρήστη και την εφαρμογή. Είναι υπεύθυνοι για τη λογική που χειρίζεται τις αιτήσεις που λαμβάνονται από το χρήστη, την επεξεργασία αυτών των αιτήσεων και την επιστροφή των κατάλληλων απαντήσεων, είτε αυτές είναι δεδομένα είτε HTML σελίδες.

Ρόλος των Controllers:

1. **Επεξεργασία Αιτήσεων:** Επεξεργάζονται τις HTTP αιτήσεις (GET, POST, PUT, DELETE).
2. **Επικοινωνία με τα Models:** Ανακτούν ή ενημερώνουν δεδομένα στη βάση μέσω των Models.

3. **Απάντηση στον Χρήστη:** Επιστρέφουν τα δεδομένα ή τις σελίδες στον χρήστη μέσω Views ή JSON για APIs.
4. **Ασφάλεια:** Εφαρμόζουν κανόνες επαλήθευσης δεδομένων και πρόσβασης.

Παράδειγμα Controller: RegisteredUserController.php

Ο παρακάτω κώδικας αποτελεί παράδειγμα ενός Controller που χειρίζεται την εγγραφή νέων χρηστών.

```
public function store(Request $request): RedirectResponse
{
    $request->validate([
        'name' => ['required', 'string', 'max:255'],
        'email' => ['required', 'string', 'lowercase', 'email', 'max:255', 'unique:'.User::class],
        'password' => ['required', 'confirmed', Rules\Password::defaults()],
    ]);

    $user = User::create([
        'name' => $request->name,
        'email' => $request->email,
        'password' => Hash::make($request->password),
    ]);

    event(new Registered($user));

    Auth::login($user);

    return redirect(route('dashboard', absolute: false));
}
```

Εικόνα 16: Στιγμιότυπο κώδικα από Controller

Ανάλυση:

- **\$request->validate([...]):** Ελέγχει ότι τα δεδομένα εισόδου (όνομα, email, κωδικός) είναι έγκυρα σύμφωνα με τους ορισμένους κανόνες.
- **User::create([...]):** Δημιουργεί έναν νέο χρήστη στη βάση δεδομένων με τα δεδομένα που εισήγαγε ο χρήστης.
- **Hash::make(\$request->password):** Κρυπτογραφεί τον κωδικό πρόσβασης για λόγους ασφαλείας.
- **event(new Registered(\$user)):** Πυροδοτεί ένα συμβάν για την εγγραφή του χρήστη, το οποίο μπορεί να χρησιμοποιηθεί για αποστολή email επιβεβαίωσης.
- **Auth::login(\$user):** Συνδέει αυτόματα τον χρήστη αφού εγγραφεί.
- **return redirect(route('dashboard')):** Ανακατευθύνει τον χρήστη στον πίνακα ελέγχου της εφαρμογής.

5.3.6 Models

Τα **Models** αναπαριστούν τη βάση δεδομένων και είναι υπεύθυνα για τη διαχείριση και την επεξεργασία των δεδομένων. Σε ένα τυπικό περιβάλλον Laravel, κάθε πίνακας της βάσης δεδομένων έχει ένα αντίστοιχο Model που χειρίζεται τις λειτουργίες CRUD (Create, Read, Update, Delete).

Ρόλος των Models:

1. **Αναπαράσταση Δεδομένων:** Κάθε Model αναπαριστά έναν πίνακα της βάσης δεδομένων.

2. **Λειτουργίες CRUD:** Διευκολύνει τη δημιουργία, ανάκτηση, ενημέρωση και διαγραφή δεδομένων.
3. **Σχέσεις:** Ορίζει τις σχέσεις μεταξύ διαφορετικών πινάκων, όπως one-to-many, many-to-many.
4. **Λογική Επιπέδου Δεδομένων:** Περιέχει λογική σχετική με τα δεδομένα, όπως scope queries ή accessor/mutator methods.

Παράδειγμα Model: User.php

Ο παρακάτω κώδικας είναι ένα τυπικό Model για τον πίνακα users.

5.3.7 API

Η ανάπτυξη ενός API στο Laravel βασίζεται στη συνεργασία πολλών υποσυστημάτων της πλατφόρμας, που περιλαμβάνουν ρυθμίσεις, δομές και αρχεία κώδικα. Η σύνδεση όλων αυτών των στοιχείων εξασφαλίζει την εύρυθμη λειτουργία του API, παρέχοντας ασφάλεια, απόδοση και επεκτασιμότητα.

Ρόλος και Συνεργασία των Υποσυστημάτων

Το **.env** αποτελεί το θεμέλιο για τη διαχείριση των ρυθμίσεων του συστήματος. Μέσω αυτού, η εφαρμογή αποκτά πρόσβαση σε κρίσιμες πληροφορίες όπως η σύνδεση με τη βάση δεδομένων, οι παράμετροι ασφαλείας και άλλα χαρακτηριστικά που είναι απαραίτητα για τη λειτουργία του API. Το αρχείο αυτό, σε συνδυασμό με το **config/database.php**, εξασφαλίζει τη σωστή επικοινωνία της εφαρμογής με τη βάση δεδομένων. Ο driver MySQL που καθορίζεται στο **database.php** χρησιμοποιεί τις πληροφορίες από το **.env** για τη δημιουργία της σύνδεσης.

Τα **migrations** διασφαλίζουν τη δομή της βάσης δεδομένων που υποστηρίζει το API. Για παράδειγμα, πίνακες όπως ο **users**, ο **tags**, ή ο **password_resets** δημιουργούνται μέσω των **migrations** και λειτουργούν ως το αποθετήριο δεδομένων που διαχειρίζεται το API. Η λειτουργικότητα αυτή επιτρέπει την εύκολη αναβάθμιση και συντήρηση της βάσης δεδομένων, χωρίς να χρειάζεται χειροκίνητη παρέμβαση.

Οι **Controllers**, όπως ο **RegisteredUserController**, αναλαμβάνουν την επιχειρησιακή λογική του API. Αποτελούν τη γέφυρα μεταξύ των αιτημάτων που λαμβάνει η εφαρμογή και των δεδομένων που επεξεργάζεται. Στο παράδειγμα του **store()** method, η μέθοδος ελέγχει την εγκυρότητα των δεδομένων εισόδου, δημιουργεί έναν νέο χρήστη και επιστρέφει το κατάλληλο αποτέλεσμα. Αυτό σημαίνει ότι κάθε αίτημα προς το API περνά από έναν **Controller**, ο οποίος καθορίζει πώς θα διαχειριστεί τα δεδομένα.

Τα **Models**, όπως το **User**, συνεργάζονται άμεσα με τη βάση δεδομένων. Παρέχουν μια αφαιρετική διεπαφή για την πρόσβαση στα δεδομένα, κάνοντας τη διαχείριση πιο απλή και αποτελεσματική. Επιπλέον, τα **Models** χρησιμοποιούνται στους **Controllers** για να δημιουργήσουν, ανακτήσουν ή ενημερώσουν εγγραφές στη βάση δεδομένων. Η έννοια της "μαζικής ανάθεσης" (**fillable**) και η κρυπτογράφηση κωδικών μέσω του **Hash::make()** προσδίδουν ασφάλεια στη διαχείριση δεδομένων χρηστών.

Το **web.php** καθορίζει τις διαδρομές της εφαρμογής, συνδέοντας τα αιτήματα HTTP με συγκεκριμένες μεθόδους των **Controllers**. Για το API, το αρχείο αυτό περιέχει διαδρομές που κατευθύνουν τα αιτήματα (π.χ., **POST /register**) προς τις αντίστοιχες μεθόδους των **Controllers**. Με αυτόν τον τρόπο, οι **Controllers** διαχειρίζονται διαφορετικές ενέργειες βάσει της διαδρομής.

Τέλος, η συνολική υποδομή βασίζεται στα **Migrations**, τα **Models**, το `database.php`, το `.env` και το `web.php` για να παρέχει ένα λειτουργικό API. Κάθε μέρος συνεργάζεται στενά:

- Το `.env` και το `database.php` εξασφαλίζουν τη σύνδεση με τη βάση δεδομένων.
- Τα **Migrations** διαμορφώνουν τη βάση δεδομένων.
- Τα **Models** και οι **Controllers** υλοποιούν τη διαχείριση και επεξεργασία των δεδομένων.
- Το `web.php` δρομολογεί τα αιτήματα στις σωστές λειτουργίες.

Αυτή η συνεργασία εξασφαλίζει ότι κάθε αίτημα στο API διαχειρίζεται αποτελεσματικά, με ασφαλή πρόσβαση στα δεδομένα και ευελιξία στην ανάπτυξη νέων λειτουργιών. Ένα παράδειγμα με **UPDATE**, **DELETE**, **GET** αιτήματα είναι το παρακάτω :

```
Route::middleware('auth')->group(function () {
    Route::get('/profile', [ProfileController::class, 'edit']->name('profile.edit'));
    Route::patch('/profile', [ProfileController::class, 'update']->name('profile.update'));
    Route::delete('/profile', [ProfileController::class, 'destroy']->name('profile.destroy'));
});
```

Εικόνα 17: Στιγμιότυπο κώδικα από routes

```

class User extends Authenticatable
{
    use HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'email',
        'password',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * Get the attributes that should be cast.
     *
     * @return array<string, string>
     */
    protected function casts(): array
    {
        return [
            'email_verified_at' => 'datetime',
            'password' => 'hashed',
        ];
    }
}

```

Εικόνα 18: Στιγμιότυπο κώδικα από api users

Ανάλυση:

- **\$fillable:** Ορίζει τις στήλες που επιτρέπεται να οριστούν μαζικά (mass assignment). Εδώ επιτρέπεται η εισαγωγή τιμών στις στήλες name, email και password.
- **\$hidden:** Ορίζει τις στήλες που δεν θα επιστρέφονται όταν το Model μετατρέπεται σε JSON, όπως το password και το remember_token.
- **casts():** Ορίζει τη μετατροπή συγκεκριμένων δεδομένων. Για παράδειγμα, το email_verified_at αποθηκεύεται ως ημερομηνία και ο κωδικός κρυπτογραφείται.

5.4 Blades

5.4.1 Τι είναι τα Blades

Το Blade είναι ο προεπιλεγμένος template engine του Laravel και αποτελεί ένα εργαλείο για τη δημιουργία δυναμικών views που ενσωματώνουν δεδομένα και λογική. Εισάγει μια σύνταξη που βασίζεται στην PHP αλλά περιλαμβάνει και δικές του συντομεύσεις, με στόχο να διευκολύνει τη δημιουργία και συντήρηση διεπαφών χρήστη. Τα αρχεία Blade έχουν κατάληξη `.blade.php` και βρίσκονται στον φάκελο `resources/views`, οργανώνοντας τη δομή των views. Το Blade δεν περιορίζει τους προγραμματιστές να γράφουν καθαρό PHP κώδικα μέσα στα αρχεία του, αλλά προσφέρει εναλλακτικές επιλογές που είναι πιο καθαρές και εύχρηστες. Το σύστημα λειτουργεί μεταγλωττίζοντας τα αρχεία `.blade.php` σε κανονικά αρχεία PHP, γεγονός που διασφαλίζει υψηλή απόδοση κατά την εκτέλεση της εφαρμογής. Έτσι, το Blade συνδυάζει την απλότητα στη σύνταξη με την ευελιξία της PHP, καθιστώντας το κατάλληλο για οποιοδήποτε είδος εφαρμογής.

5.4.2 Πως λειτουργεί το Blade

Το Blade template engine λειτουργεί μεταγλωττίζοντας τα αρχεία `.blade.php` σε καθαρά PHP αρχεία, προσφέροντας μεγάλη ευελιξία και αποδοτικότητα. Παρέχει ειδικές "directives" που επιτρέπουν την εισαγωγή λογικής μέσα στα views, όπως οι `@if`, `@foreach`, και `@extends`, που διευκολύνουν την επεξεργασία δυναμικών δεδομένων. Επιπλέον, το Blade διευκολύνει τη χρήση layouts, τα οποία προσφέρουν ένα κεντρικό σημείο διαχείρισης του κοινού σχεδιασμού της εφαρμογής, ενώ παράλληλα υποστηρίζει την εκτύπωση δεδομένων με τη χρήση της σύνταξης `{{ }}`, η οποία προστατεύει αυτόματα από XSS επιθέσεις. Ακόμη, μέσω της δυνατότητας ανάπτυξης components, παρέχεται η δυνατότητα επαναχρησιμοποίησης στοιχείων διεπαφής. Αυτή η modular προσέγγιση καθιστά το Blade ένα εργαλείο που προάγει την οργάνωση και τη διαχειρισιμότητα της εφαρμογής. Τέλος, ο συνδυασμός της απλότητας στη σύνταξη και της ευελιξίας του Blade το καθιστά έναν από τους πιο αποδοτικούς τρόπους δημιουργίας διεπαφών.

5.4.3 Σε τι χρησιμεύει το Blade

Τα Blade templates εξυπηρετούν βασικές ανάγκες για την ανάπτυξη διεπαφών χρήστη, συνδυάζοντας λογική και δεδομένα. Αρχικά, διευκολύνουν τη διαχείριση περιεχομένου, καθώς επιτρέπουν την οργάνωση των δεδομένων που αποστέλλονται από τους Controllers και την προβολή τους στο χρήστη. Επιπλέον, μέσω των layouts και των components, τα Blade templates προάγουν την επαναχρησιμοποίηση κώδικα, μειώνοντας την επαναληψιμότητα και βελτιώνοντας τη συντηρησιμότητα της εφαρμογής. Ταυτόχρονα, παρέχουν δυναμικό περιεχόμενο, καθιστώντας εύκολη την εμφάνιση δεδομένων που προκύπτουν από βάσεις δεδομένων ή χρήστες, όπως λίστες προϊόντων ή αποτελέσματα αναζητήσεων. Η ασφάλεια αποτελεί άλλο ένα σημαντικό χαρακτηριστικό, καθώς η εκτύπωση δεδομένων προστατεύεται αυτόματα από επιθέσεις XSS. Με την ενσωμάτωσή τους στις δυνατότητες του Laravel, τα Blade templates εξυπηρετούν απόλυτα τη δημιουργία ασφαλών, δυναμικών και επαναχρησιμοποιήσιμων διεπαφών χρήστη.

5.4.4 Γιατί να επιλέξουμε τα Blades

Η χρήση των Blade templates προσφέρει πολλαπλά πλεονεκτήματα που τα καθιστούν απαραίτητα για την ανάπτυξη εφαρμογών με το Laravel. Πρώτον, είναι απλά και εύκολα στη χρήση, με σύνταξη που μειώνει την πολυπλοκότητα του HTML και PHP κώδικα, καθιστώντας τον πιο ευανάγνωστο και διαχειρίσιμο. Δεύτερον, το Blade είναι απόλυτα ενσωματωμένο με το Laravel, αξιοποιώντας πλήρως τις δυνατότητες του framework, όπως η διαχείριση δεδομένων από Controllers και η χρήση middleware για ασφαλείς λειτουργίες. Επιπλέον, παρέχουν εξαιρετική απόδοση, καθώς μετατρέπονται

σε PHP αρχεία κατά την εκτέλεση, διατηρώντας την ταχύτητα της εφαρμογής. Τέλος, οι δυνατότητες επέκτασης, όπως η δημιουργία custom directives και components, επιτρέπουν την προσαρμογή των templates στις ανάγκες κάθε εφαρμογής. Συνολικά, τα Blade templates εξασφαλίζουν οργανωμένη, ασφαλή και αποδοτική ανάπτυξη διεπαφών χρήστη.

5.4.5 Παραδείγμα

Στο επόμενο κομμάτι παρουσιάζεται ένα παράδειγμα από τον κώδικα της εφαρμογής που έχει υλοποιηθεί με Blade. Στο συγκεκριμένο σημείο ο κώδικας εμφανίζει μια λίστα θέσεων εργασίας που αντλούνται από τη βάση δεδομένων μέσω του \$posts. Κάθε εγγραφή παρουσιάζεται ως ένα στοιχείο της λίστας με δυναμικά δεδομένα (τίτλος, περιοχή, likes) και μια σύντομη περιγραφή. Χρησιμοποιούνται Blade directives (@foreach) για την επανάληψη και helper functions (Str::limit, asset) για τη διαχείριση του περιεχομένου. Τέλος, παρέχεται η δυνατότητα πλοήγησης μεταξύ σελίδων με τη σελιδοποίηση.

```

<section class="site-section" id="next">
  <div class="container">
    <div class="row mb-5 justify-content-center">
      <div class="col-md-7 text-center">
        <h2 class="section-title mb-2">43,167 Job Listed</h2>
      </div>
    </div>
    <div class="job-listings mb-5">
      <div class="bg" style="background-color: white;">
        <ul class="job-listings m-0 p-2">
          @foreach($posts as $post)
            <li class="job-listing d-block d-sm-flex pb-3 pb-sm-0 align-items-center" style="margin-top: 50px; background-color:#eeffee">
              <!-- Job Details Link -->
              <a href="{{ asset('import/assets/job-single.html') }}"></a>
              <!-- Job Logo -->
              <div class="job-listing-logo">
                
              </div>
              <!-- Job Details -->
              <div
                class="job-listing-about d-sm-flex custom-width w-100 justify-content-between mx-4">
                <div class="job-listing-position custom-width w-50 mb-3 mb-sm-0">
                  <!-- Injecting Post Title -->
                  <h2 >{{ $post->title }}</h2>
                  <!-- Injecting Post Content (can be truncated) -->
                  <strong >{{ \Illuminate\Support\Str::limit($post->content, 50) }}</strong>
                </div>
                <!-- Location Div -->
                <div class="job-listing-location mb-3 mb-sm-0 custom-width w-25">
                  <span class="icon-room"></span> {{ $post->area }}
                </div>
                <!-- Status Placeholder -->
                <div class="job-listing-meta">
                  <span>likes : </span>
                  <span class="badge badge-danger">{{ $post->likes }}</span>
                </div>
              </div>
            </li>
          @endforeach
        </ul>
        <!-- Pagination Links -->
        <div >
          {{ $posts->links('pagination:bootstrap-4') }}
        </div>
      </div>
    </div>
  </section>

```

Εικόνα 19: Στιγμιότυπο κώδικα Blade

Ανάλυση κώδικα :

- **<section class="site-section" id="next">**
Δημιουργεί ένα section HTML, το οποίο χρησιμεύει για την οργάνωση της σελίδας. Το id="next" επιτρέπει την αναφορά σε αυτό το τμήμα μέσω συνδέσμων.
- **<div class="container">**
Χρησιμοποιείται για την οριοθέτηση του περιεχομένου σε μια προκαθορισμένη διάταξη, συχνά καθορισμένη από ένα CSS framework όπως το Bootstrap.
- Τμήμα Τίτλου:
 - **<div class="row mb-5 justify-content-center">**
Δημιουργεί μια ευθυγραμμισμένη στο κέντρο γραμμή με κάτω περιθώριο (mb-5).
 - **<div class="col-md-7 text-center">**
Χωρίζει τον χώρο της γραμμής σε μια στήλη, ευθυγραμμίζοντας το κείμενο στο κέντρο.
 - **<h2 class="section-title mb-2">43,167 Job Listed</h2>**
Εμφανίζει τον τίτλο της ενότητας με κατάλληλη μορφοποίηση.
- Τμήμα Λίστας Θέσεων Εργασίας:
 - **<ul class="job-listings mb-5">**
Δημιουργεί μια λίστα (ul) με περιθώριο στο κάτω μέρος.
 - **@foreach(\$posts as \$post)**
Χρησιμοποιεί έναν βρόχο foreach για να επαναλάβει κάθε εγγραφή της μεταβλητής \$posts. Κάθε εγγραφή αναφέρεται ως \$post.
- Εμφάνιση Καθεμίας Θέσης Εργασίας:
 - **<li class="bg" style="background-color: white;">**
Δημιουργεί ένα στοιχείο λίστας με λευκό φόντο.
 - **<div class="job-listing d-block ...">**
Εμφανίζει ένα block για τη θέση εργασίας, με διάφορες κλάσεις CSS για την στοίχιση.
 - ****
Δημιουργεί έναν σύνδεσμο που οδηγεί σε μια σελίδα με λεπτομέρειες για τη θέση.
 - ****
Εισάγει μια εικόνα για τη θέση, π.χ., το λογότυπο της εταιρείας.
- Τίτλος και Περιγραφή Θέσης:
 - **<h2>{{ \$post->title }}</h2>**
Εμφανίζει τον τίτλο της θέσης, που προέρχεται από το αντικείμενο \$post.
 - **{{ Illuminate\Support\Str::limit(\$post->content, 50) }}**
Εμφανίζει μέρος της περιγραφής, περιορίζοντάς την στους πρώτους 50 χαρακτήρες μέσω της συνάρτησης Str::limit.
- Περιοχή και Likes:
 - **<div class="job-listing-location">**
Εμφανίζει την περιοχή της θέσης, με χρήση του πεδίου \$post->area.
 - **{{ \$post->likes }}**
Εμφανίζει τον αριθμό των likes με τη χρήση CSS για εμφάνιση ως "badge".
- Σελιδοποίηση:
 - **{{ \$posts->links('pagination::bootstrap-4') }}**
Δημιουργεί σελιδοποίηση χρησιμοποιώντας τις εγγενείς δυνατότητες του Laravel και τις κλάσεις του Bootstrap 4.

5.4.6 Επίλογος

Το Blade, ως η επίσημη μηχανή εμφάνισης του Laravel, αποτελεί έναν από τους ακρογωνιαίους λίθους της ανάπτυξης εφαρμογών, προσφέροντας έναν αποτελεσματικό και ευέλικτο τρόπο διαχείρισης του front-end. Μέσα από τη χρήση του Blade, ο διαχωρισμός της λογικής από την εμφάνιση επιτυγχάνεται με έναν απλό αλλά ισχυρό τρόπο, επιτρέποντας τη δημιουργία δυναμικών διεπαφών χρήστη που είναι εύκολες στη συντήρηση και την επεκτασιμότητα.

Κατά την ανάλυση των χαρακτηριστικών του Blade, φάνηκε πως οι δυνατότητες που προσφέρει, όπως οι directives, οι layouts και τα components, συμβάλλουν στη βελτιστοποίηση της διαδικασίας ανάπτυξης. Παρέχουν τη δυνατότητα επαναχρησιμοποίησης κώδικα και τη διατήρηση μιας ενιαίας αρχιτεκτονικής στον σχεδιασμό των views, ενώ παράλληλα επιτρέπουν την ενσωμάτωση CSS, JavaScript και άλλων εξωτερικών πόρων με ευκολία. Ένα σημαντικό πλεονέκτημα του Blade είναι η απλότητα στη χρήση του, που καθιστά τον κώδικα πιο αναγνώσιμο και τη διαδικασία ανάπτυξης πιο αποδοτική.

Αναλύθηκαν επίσης πρακτικά παραδείγματα για τη δημιουργία views με τη χρήση του Blade, όπου έγινε εμφανές πως η χρήση directives, όπως οι @foreach, @if και @include, διευκολύνει την επεξεργασία και την εμφάνιση δεδομένων από τη βάση, ενώ παράλληλα ενισχύει τη δυναμικότητα της εφαρμογής. Επιπλέον, μέσω του Blade επιτυγχάνεται η εύκολη διαχείριση των δεδομένων που λαμβάνονται από τους controllers και τα models.

Συνοψίζοντας, το Blade αποτελεί αναπόσπαστο εργαλείο στο οικοσύστημα του Laravel, διευκολύνοντας τη δημιουργία καλαισθητών και λειτουργικών εφαρμογών. Η απλότητα, η ευελιξία και η απόδοσή του το καθιστούν ιδανικό για κάθε φοιτητή και επαγγελματία προγραμματιστή που επιδιώκει να αναπτύξει σύγχρονες και προηγμένες web εφαρμογές.

Κεφάλαιο 6ο: Case study

6.1 Εισαγωγή

Στο παρόν κεφάλαιο, θα παρουσιαστεί η λειτουργικότητα και η εμπειρία χρήστη της πλατφόρμας hackathon που έχει αναπτυχθεί στο πλαίσιο της διπλωματικής αυτής εργασίας. Η πλατφόρμα σχεδιάστηκε με σκοπό να παρέχει ένα οργανωμένο περιβάλλον για τη συμμετοχή χρηστών σε διαγωνισμούς καινοτομίας, διευκολύνοντας τη διαδικασία εγγραφής, υποβολής ιδεών και αλληλεπίδρασης μεταξύ των συμμετεχόντων.

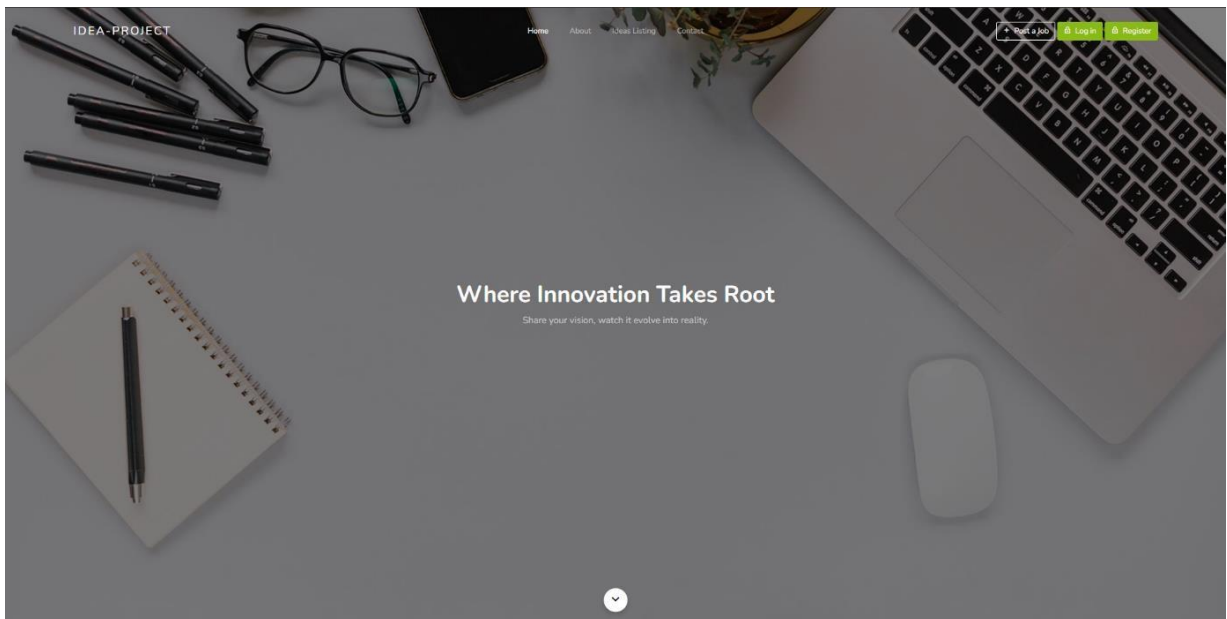
Το κεφάλαιο αυτό περιλαμβάνει την παρουσίαση ενός σεναρίου χρήσης που καλύπτει τα βασικά βήματα που εκτελεί ένας χρήστης κατά τη συμμετοχή του σε ένα hackathon. Αρχικά, θα αναλυθεί η

διαδικασία εγγραφής και η πρόσβαση στο προσωπικό προφίλ. Στη συνέχεια, θα περιγραφεί η διαδικασία υποβολής ιδεών και η προβολή τους στον δημόσιο πίνακα ιδεών της πλατφόρμας. Τέλος, θα παρουσιαστεί η σελίδα λεπτομερειών ιδέας, η οποία επιτρέπει στους χρήστες να δουν πιο αναλυτικά τις υποβληθείσες ιδέες.

Η ανάλυση αυτή έχει ως στόχο να αναδείξει τη λειτουργικότητα της πλατφόρμας και να παρέχει σαφή εικόνα της εμπειρίας χρήστη, ώστε να αξιολογηθεί η πληρότητα και η χρηστικότητα του συστήματος.

6.2 Home Page

Με την είσοδο του χρήστη στην σελίδα της πλατφόρμας, το πρώτο πράγμα που αντικρίζει είναι το home page. Μια σελίδα κατα την οποία μπορεί να δει γενικά πράγματα για τον διαγωνισμό αλλά και να κάνει εγγραφή ή είσοδο. Βέβαια στα πλαίσια της προσαρμοστικότητας της σελίδας και για άλλους διαγωνισμούς μπορεί να χρησιμοποιηθεί για την δημιουργία των επίκαιρων ενημερώσεων για τον διαγωνισμό ή ότι άλλο κριθεί απαραίτητο. Η αρχική (home page) σελίδα φαίνεται παρακάτω στις φωτογραφίες.



Εικόνα 20: Στιγμιότυπο από Homepage 1



Καλώς ήρθατε στο Hackathon – Το Απόλυτο Τεχνολογικό Γεγονός!

Ετοιμαστείτε για ένα συναρπαστικό κούρσι κανονικής και ανοικτής! Το Hackathon έρχεται για να συγκεντρώσει προγραμματιστές, designers, επιχειρηματίες και δημιουργικά μυαλά από όλη τη χώρα σε μια ασφαλή εμπειρία δημιουργίας.

Τι να Περιμένεις:

- 48 ώρες απεριόριστη δημιουργικότητα
- Πρωτότυπος που θα σε εμπνεύσουν να καινοτομήσεις
- Βραβεία για τις καλύτερες ιδέες!

📅 Πότε: 1/1/2025
📍 Πού: (Ευρασιακό) / Ονέιξ

Έχεις φανταστεί ποτέ μια πλατφόρμα που μπορεί να κάνει την πλάση σου καλύτερη; Σήμερα είναι η καλύτερη σου! Το Hackathon είναι ένας συναρπαστικός, δοκιμαστικός θησαυρός πλατφορμών για το καινούριο κατό!

Εκείνο με προγραμματιστές, designers, επιχειρηματίες και καινοτομίες για να συμβάλει μαζί τους, που θα βελτιώσουν τη ζωή των πολιτών της πατρίδας και της Ελλάδας!

Γιατί να Συμμετέχεις:

- 🏆 **Καλύτεροι Προβλεπόμενοι** - Δωμάτια σε αντάλλαξη προγραμμάτων καινοτομίας
- 📄 **Επιχειρησιακά Βραβεία** - Κιβερνοεπιχειρήματα και ανταμειβόμενοι
- 🌟 **Διακρίσεις** - Γνώριση ανταμειβόμενων με καινούριο ενδιαφέρον!

Μην χάσεις την ευκαιρία να φέρεις οτιδήποτε φανταστικό στην πλατφόρμα που θα αλλάξει τις ταχθείς δυνατότητες σου! Συμμετείχε! Σήμερα και βρήκαμε να κάνουμε τον κόσμο λίγο καλύτερο, μία ώρα τη φορά! 🌟



Εικόνα 21: Στιγμιότυπα από Homerpage 2

Στο επάνω μέρος της σελίδας υπάρχει μια λίστα επιλογών περιήγησης στην πλατφόρμα όπου είναι σε μορφή navigation bar. Εκεί δίνεται η δυνατότητα στον χρήστη να μεταβεί σε σελίδες όπως είναι:

NAVIGATION BAR
Αρχική σελίδα
Σχετικά με εμάς (About)
Λίστα ιδεών (Ideas Listing)
Επικοινωνία (Contact)
Εγγραφή (Register)
Σύνδεση (Log in)

Το navigation bar είναι ίδιο σε κάθε σελίδα που υπάρχει μέσα στην πλατφόρμα και δίνονται οι ίδιες δυνατότητες αλληλεπίδρασης του χρήστη από παντού με την διαφορά ότι με την σύνδεση του χρήστη στην πλατφόρμα του δίνεται η δυνατότητα του DashBoard , ένα μέρος που θα αναλυθεί παρακάτω όπου μπορεί να δει τα στοιχεία του κ.α.. Με την σύνδεση στην πλατφόρμα το dashboard τον ακολουθεί παντού.

Στο επόμενο μέρος της σελίδας εμφανίζονται φωτογραφίες του διαγωνισμού και ένα κείμενο υποδοχής/ενημέρωσης για τον χρήστη που μπήκε στην πλατφόρμα. Το section της σελίδας είναι δομημένο με τέτοιο τρόπο ώστε να μπορεί να φιλοξενήσει τόσο κείμενο όσο και φωτογραφίες ή βίντεο που θα ήθελε να χρησιμοποιήσει ο κάθε διαχειριστής της σελίδας.

6.3 Register/Log in

Ο χρήστης για να μπορέσει να αλληλεπιδράσει πλήρως με την πλατφόρμα θα χρειαστεί να κάνει εγγραφή/σύνδεση. Η δυνατότητα αυτή δίνεται από το navigation bar σε κάθε σημείο της πλατφόρμας. Με την επιλογή της εγγραφής (Register) έρχεται αντιμέτωπος με την εξής διεπαφή :

The image shows a screenshot of a web application interface. At the top, there is a navigation bar with the text 'IDEA-PROJECT' on the left and links for 'Home', 'About', 'Ideas Listing', and 'Contact' in the center. On the right side of the navigation bar, there are three buttons: 'Post a Job', 'Log In', and 'Register'. Below the navigation bar, the main content area is titled 'Register' and contains a registration form. The form has the following fields: 'Name' (with the value 'tester'), 'Email' (with the value 'tester@example.com'), 'Password' (with masked characters), and 'Confirm Password' (with masked characters). There is a green 'Sign Up' button at the bottom of the form.

Εικόνα 22 : Στιγμιότυπο από Register

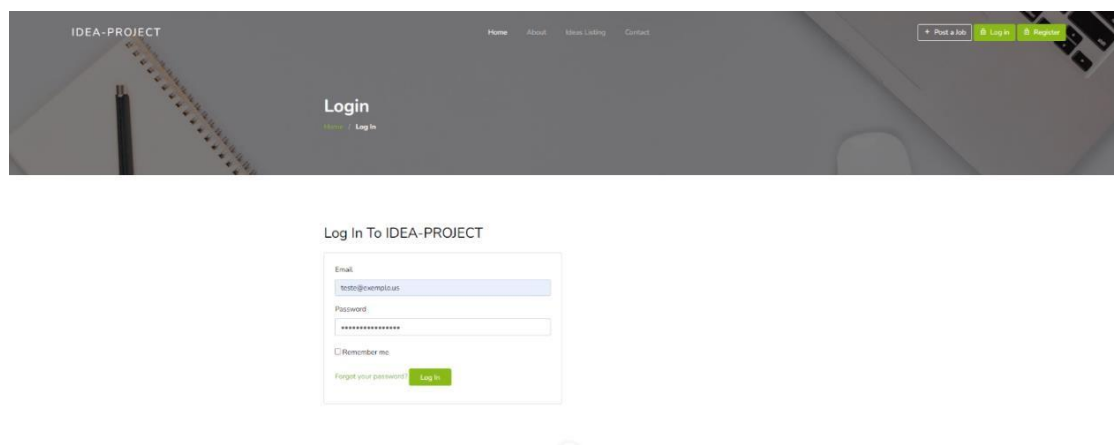
Τα στοιχεία που πρέπει να δώσει ο χρήστης είναι το όνομα (username), το email, και έναν κωδικό (password) . Μόλις κάνει την εγγραφή και διαπιστωθεί ότι δεν υπάρχει ίδιος χρήστης με ίδιο email ή username τότε γίνεται η εγγραφή του στην βάση δεδομένων. Στο συγκεκριμένο παράδειγμα δόθηκαν τα εξής στοιχεία :

ΣΤΟΙΧΕΙΑ ΧΡΗΣΤΗ	
Username :	teste
Email :	teste@exemplo.us
Password :	teste@exemplo.us

Με την υποβολή του αιτήματος της εγγραφής μπορούμε να δούμε και την εγγραφή του στην βάση δεδομένων.

<input type="checkbox"/>				15 teste	teste@exemplo.us	NULL	\$2y\$12\$UwRRwmmGF7o2LGXMy0YFK.fkm5ox1N3kJZGbr53V3nv...	NULL	2024-11-12 14:44:57	2024-11-12 14:44:57
--------------------------	--	--	--	----------	------------------	------	--	------	---------------------	---------------------

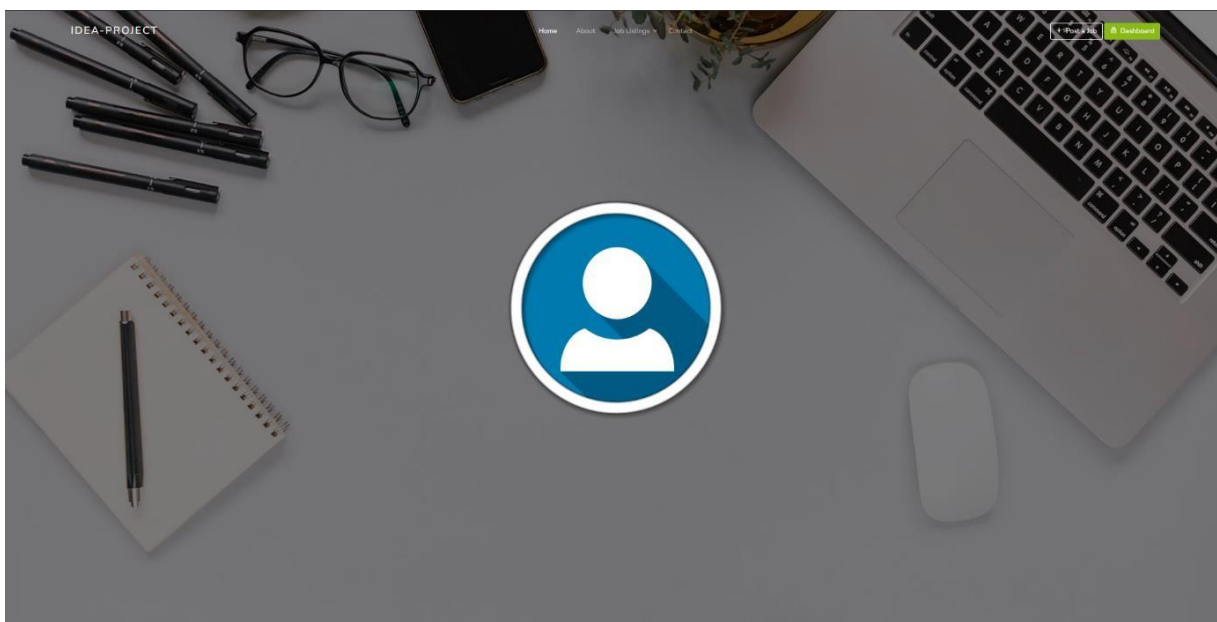
Με την εγγραφή του πλέον μπορεί να κάνει σύνδεση όποτε επιθυμεί στην αντίστοιχη διεπαφή σύνδεσης (Log In). Με τα στοιχεία του email και του κωδικού που έχει τοποθετήσει.



Εικόνα 23: Στιγμιότυπο από Log in

6.4 Dashboard

Αφού γίνει η σύνδεση βλέπουμε ότι τα κομμάτια της σύνδεση/εγγραφής έχουν μετατραπεί σε Dashboard όπου ο χρήστης μπορεί να εισέλθει στον προσωπικό του λογαριασμό.



Εικόνα 24: Στιγμιότυπο από Dashboard 1

Profile
 id: 15
 Name: teste
 Email: teste@exemplo.us

Liked Ideas

Name	Title	Content
Alice Smith	Machine Learning Model	Develop a machine learning model to predict customer churn based on various metrics.
Alice Smith	Data Cleansing Tool	Develop a tool to clean and preprocess data before analysis to improve accuracy.

My ideas

ID	Title	Content
17	Fraud Detection System	Design a fraud detection system that uses machine learning to identify fraudulent transactions.
18	Customer Feedback Analysis	Develop a tool to analyze customer feedback from surveys and reviews.

Εικόνα 25: Στιγμιότυπο από Dashboard 2

Για την καλύτερη απόδοση του παραδείγματος αλλά και για να μπορέσει κανείς να καταλάβει καλύτερα την λειτουργία της πλατφόρμας έχουν δοθεί κάποια προεπιλεγμένα δεδομένα στον χρήστη για την πλήρη εμφάνιση των στοιχείων όπως και εμφανίζονται περισσότερα πράγματα από ότι στον πραγματικό χρήστη όπως το id του το οποίο του δόθηκε από την πλατφόρμα αυτόματα με την εγγραφή του.

Στην συγκεκριμένη διεπαφή ο χρήστης μπορεί να δει τα στοιχεία του αλλά και να δει δύο (2) πίνακες με ιδέες. Ο πρώτος πίνακας είναι οι ιδέες που έχει επιλέξει να αποθηκεύσει με την μορφή Μου αρέσει (Like) και ο δεύτερος πίνακας είναι οι ιδέες που έχει καταθέσει ο ίδιος στην πλατφόρμα. Για την

ευκολότερη πρόσβαση στα δεδομένα και στους δύο πίνακες δίνεται η δυνατότητα πρόσβασης απευθείας στις λεπτομέρεις των ιδεών μέσω υπερσύνδεσης από τον ίδιο τον πίνακα.

6.5 Idea Post

Η δυνατότητα κατάθεσης μια ιδέας δίνεται μόνο στους χρήστες που είναι εγγεγραμμένοι στην πλατφόρμα. Με την επιλογή για δημοσίευση μια ιδέας ο χρήστης έρχεται στην διεπαφή όπου υπάρχει μια φόρμα κατάθεσης ιδεών. Στόχος της διεπαφής είναι να γίνει με τον πιο απλοποιημένο τρόπο για αυτό και το μόνο που χρειάζεται ο χρήστης να κάνει είναι η υποβολή των εξής στοιχείων της ιδέας:


ΦΟΡΜΑ ΥΠΟΒΟΛΗΣ ΙΔΕΩΝ
Τίτλος
Περιοχή
Περιγραφή ιδέας

Με τη είσοδο αυτών των στοιχείων γίνεται η καταγραφή της ιδέας στην βάση δεδομένων σε κατάσταση pending για επιβεβαίωση από τους διαχειριστές. Στο συγκεκριμένο παράδειγμα δόθηκε ένα τυχαίο κείμενο με τα εξής στοιχεία:

ΦΟΡΜΑ ΥΠΟΒΟΛΗΣ ΙΔΕΩΝ	
Τίτλος	Hello
Περιοχή	Pavloy Mela
Περιγραφή ιδέας	Hello wotrld

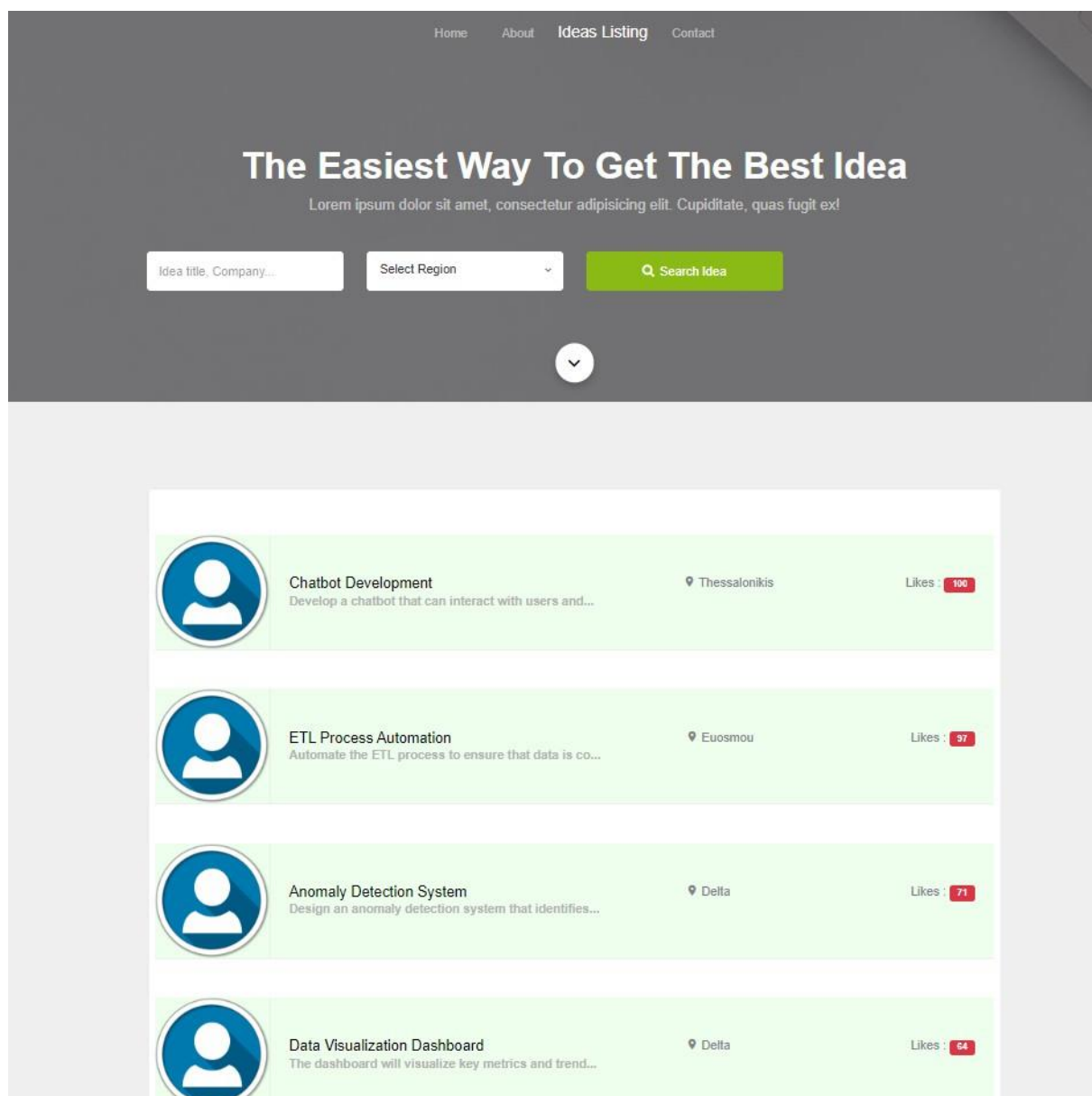
Είναι σημαντικό να τονιστεί πως η επιλογή της περιοχής είναι ένας drop down πίνακας με προεπιλογές τους δήμους της Θεσσαλονίκης και την δυνατότητα Anywhere το οποίο όμως μπορεί να μεταβληθεί ανάλογα με τις απαιτήσεις τις εφαρμογής.

Η καταγραφή στην βάση δεδομένων έχει αυτήν την μορφή :



6.6 Idea Listing

Μόλις ενημερωθεί η βάση δεδομένων και αφού δοθεί από τους διαχειριστές η έγκριση της ιδέας (για τις απαιτήσεις του παραδείγματος δόθηκε χειροκίνητα και η έγκριση αλλά και 1 like από τον ίδιο τον χρήστη) τότε εμφανίζεται στον πίνακα Idea Listing ο οποίος είναι προσπελάσιμος από και αυτός από το navigation bar. Εκεί μπορεί ο χρήστης να δει την λίστα με τις ιδέες ταξινομημένες σε φθίνουσα σειρά βάση των Like των οποίων έχει η κάθε μια, τον τίτλο της ιδέας αλλά και την περιοχή στην οποία αναφέρεται. Όπως και δίνεται η δυνατότητα για αναζήτηση ιδεών με βάση τον τίτλο ή την περιοχή. Η διεπαφή έχει αυτήν την μορφή:



Εικόνα 26 : Στιγμιότυπο από Idea Listing

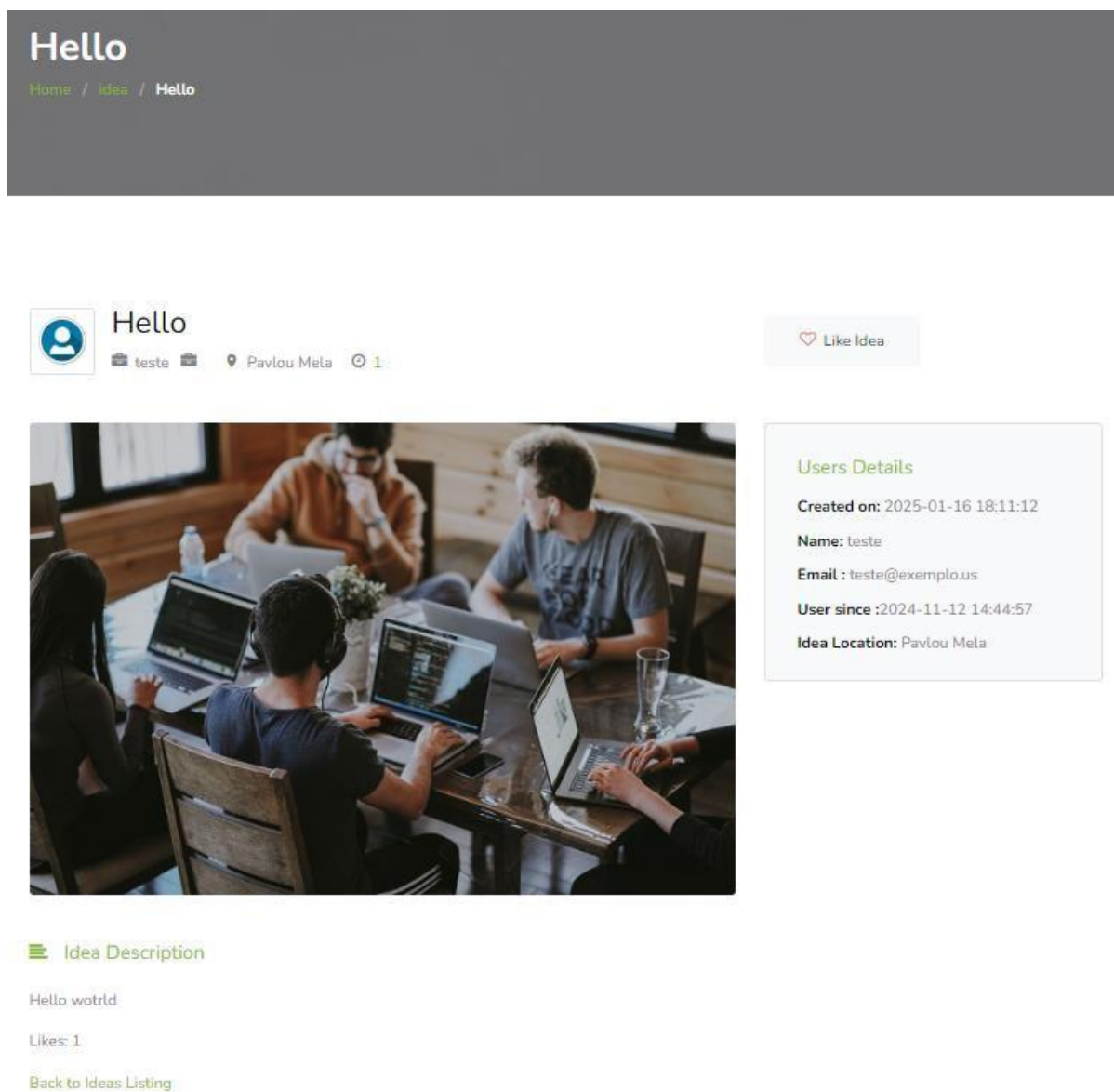
Με περιήγηση μέσω της σελιδοποίησης που έχει η σελίδα θα μπορεί ο χρήστης να βρει τη ιδέα του (Η συγκεκριμένη είναι αυτή που κατατέθηκε στα πλαίσια του παραδείγματος). Όπου με την επιλογή της ιδέας μεταφερόμαστε στις λεπτομέρειες της ιδέας σε μια ξεχωριστή διεπαφή.



Εικόνα 27: Στιγμιότυπο από Idea Listing

6.7 Single Idea

Στην διεπαφή μια ιδέας ξεχωριστά ο χρήστης μπορεί να δει τις λεπτομέρεις της ιδέας, να κάνει Like, να δει τα στοιχεία αλλά και τα στοιχεία επικοινωνίας του χρήστη που κατέθεσε την ιδέα, και την περιοχή την οποία αναφέρεται, την ημερομηνία δημιουργίας της ιδέας και επίσης μπορεί να επιστρέψει πίσω στην λίστα με τις ιδέες με ξεχωριστή υπερσύνδεση πέρα από το navigation bar. Η διεπαφή έχει αυτήν την μορφή :



Εικόνα 28: Στιγμιότυπο από Single Idea

Κεφάλαιο 7ο: Προτάσεις Βελτίωσης και Επέκτασης της Πλατφόρμας Hackathon

Η συνεχής εξέλιξη των τεχνολογικών εργαλείων και οι διαρκώς μεταβαλλόμενες ανάγκες των χρηστών επιβάλλουν τη συνεχή βελτίωση και αναπροσαρμογή κάθε ψηφιακής πλατφόρμας. Η ανάπτυξη μίας πλατφόρμας δεν αποτελεί μία στατική διαδικασία, αλλά έναν δυναμικό κύκλο που απαιτεί στρατηγικό σχεδιασμό για την ενσωμάτωση νέων λειτουργιών, τη βελτίωση της απόδοσης και την προσαρμογή σε σύγχρονες τεχνολογικές καινοτομίες.

Στο συγκεκριμένο κεφάλαιο θα παρουσιαστούν προτάσεις για μελλοντικές αλλαγές και βελτιστοποιήσεις που στοχεύουν στη μεγιστοποίηση της αποτελεσματικότητας, της χρηστικότητας και της ανθεκτικότητας της πλατφόρμας. Παράλληλα, θα δοθεί έμφαση στην αξιοποίηση των τεχνολογικών εξελίξεων, καθώς και στη διαρκή ανατροφοδότηση από τους χρήστες, ώστε να εξασφαλιστεί η μακροπρόθεσμη βιωσιμότητα και ανταγωνιστικότητα του συστήματος.

Αυτές οι βελτιστοποιήσεις δεν περιορίζονται μόνο σε τεχνικές πτυχές, όπως η βελτίωση της ταχύτητας και της ασφάλειας, αλλά επεκτείνονται και σε στρατηγικές ενσωμάτωσης καινοτόμων εργαλείων και πρακτικών, με σκοπό τη δημιουργία μιας πλατφόρμας που ανταποκρίνεται στις εκάστοτε σύγχρονες απαιτήσεις και προσδοκίες.

7.1 Chat

Μία από τις σημαντικότερες προτάσεις για τη μελλοντική βελτιστοποίηση της πλατφόρμας είναι η ενσωμάτωση μιας λειτουργίας chat, η οποία θα επιτρέπει την άμεση και απρόσκοπτη επικοινωνία μεταξύ των χρηστών. Αυτή η δυνατότητα στοχεύει στην ενίσχυση της διαδραστικότητας, της συνεργασίας και της ανταλλαγής πληροφοριών, δημιουργώντας ένα περιβάλλον που προάγει τη συμμετοχή και την κοινότητα.

Η λειτουργία του chat θα προσφέρει πολλαπλές δυνατότητες, όπως:

- **Άμεση Επικοινωνία:** Οι χρήστες θα μπορούν να ανταλλάσσουν μηνύματα σε πραγματικό χρόνο, διευκολύνοντας τη συνεργασία σε κοινά θέματα ή έργα.
- **Ομαδικές Συζητήσεις:** Θα υποστηρίζεται η δημιουργία ομάδων συνομιλίας, επιτρέποντας σε ομάδες χρηστών να ανταλλάσσουν ιδέες και να εργάζονται συλλογικά.
- **Κοινή Χρήση Αρχείων:** Η λειτουργία θα επιτρέπει την αποστολή αρχείων, όπως έγγραφα, εικόνες ή παρουσιάσεις, διευκολύνοντας την ανταλλαγή πληροφοριών.
- **Προσαρμογή Προφίλ:** Οι χρήστες θα μπορούν να διαχειρίζονται τις ρυθμίσεις του chat, όπως οι ειδοποιήσεις και η ορατότητα, διασφαλίζοντας μια προσωποποιημένη εμπειρία.

Η ενσωμάτωση της λειτουργίας chat απαιτεί προσεκτικό σχεδιασμό, ώστε να εξασφαλιστεί η ασφάλεια των δεδομένων και η προστασία της ιδιωτικότητας. Για τον σκοπό αυτό, θα χρησιμοποιηθούν σύγχρονες τεχνολογίες κρυπτογράφησης και θα εφαρμοστούν αυστηρές πολιτικές για την αποτροπή κακόβουλων ενεργειών.

Η λειτουργία chat δεν θα αποτελέσει μόνο ένα εργαλείο επικοινωνίας, αλλά και ένα μέσο για την ενδυνάμωση της κοινότητας των χρηστών της πλατφόρμας, δημιουργώντας ένα πιο διαδραστικό και υποστηρικτικό περιβάλλον. Η εισαγωγή αυτής της δυνατότητας αναμένεται να ενισχύσει τη συνολική εμπειρία χρήσης, καθιστώντας την πλατφόρμα πιο ελκυστική και λειτουργική.

7.2 Κατοχύρωση Ιδεών και Παρακολούθηση Υλοποίησης

Μία σημαντική μελλοντική βελτιστοποίηση της πλατφόρμας είναι η δυνατότητα κατοχύρωσης των ιδεών από τους χρήστες που αναλαμβάνουν την υλοποίησή τους. Αυτή η λειτουργία αποσκοπεί στη δημιουργία διαφάνειας και στη διευκόλυνση της παρακολούθησης των έργων, προσφέροντας μια ξεκάθαρη εικόνα για το ποιος εργάζεται πάνω σε κάθε ιδέα και ποια είναι η πρόοδος της υλοποίησης.

Η λειτουργία κατοχύρωσης ιδεών θα περιλαμβάνει τα εξής:

- **Ανάθεση και Κατοχύρωση:** Κάθε χρήστης που επιθυμεί να αναλάβει την υλοποίηση μιας ιδέας θα μπορεί να την κατοχυρώσει μέσω της πλατφόρμας. Η κατοχύρωση θα συνοδεύεται από την καταγραφή των στοιχείων του χρήστη και την ημερομηνία ανάθεσης.
- **Εμφάνιση Ενεργών Συμμετοχών:** Σε κάθε ιδέα θα εμφανίζεται ένας πίνακας με τα άτομα που ασχολούνται με την υλοποίησή της. Αυτό θα παρέχει στους χρήστες τη δυνατότητα να βλέπουν πόσα άτομα συμμετέχουν και ποια είναι τα προφίλ τους.
- **Παρακολούθηση Προόδου:** Οι χρήστες που αναλαμβάνουν την ιδέα θα έχουν τη δυνατότητα να ενημερώνουν την πλατφόρμα για την πρόοδο των εργασιών, είτε μέσω σχολίων είτε με την προσθήκη ενημερώσεων στα σχετικά αρχεία.
- **Συνεργατική Υποστήριξη:** Η λειτουργία θα επιτρέπει τη σύνδεση χρηστών που ενδιαφέρονται να συμβάλουν στην ίδια ιδέα, προάγοντας τη συνεργασία και την ανταλλαγή τεχνολογίας.

Η δυνατότητα κατοχύρωσης ιδεών ενισχύει την υπευθυνότητα και τη διαφάνεια, δημιουργώντας ένα περιβάλλον όπου οι χρήστες μπορούν να παρακολουθούν την πρόοδο και να αναγνωρίζουν τη συνεισφορά των συμμετεχόντων. Επιπλέον, συμβάλλει στην αποφυγή διπλής εργασίας ή σύγχυσης, αφού κάθε ιδέα θα είναι ξεκάθαρα συνδεδεμένη με τα άτομα που εργάζονται πάνω σε αυτή.

Αυτή η λειτουργία αναμένεται να ενισχύσει την εμπιστοσύνη και τη συνεργατικότητα στην πλατφόρμα, ενώ παράλληλα θα παρέχει ένα εργαλείο για την αποτελεσματική διαχείριση των ιδεών και των πόρων που απαιτούνται για την υλοποίησή τους.

7.3 Δημιουργία Υπερπλατφόρμας για Ανάπτυξη Εξατομικευμένων Πλατφορμών

Μία από τις πιο καινοτόμες προτάσεις για τη μελλοντική εξέλιξη είναι η δημιουργία μιας υπερπλατφόρμας, η οποία θα επιτρέπει στους χρήστες να δημιουργούν εύκολα και γρήγορα τις δικές τους πλατφόρμες, παρόμοιες με την υπάρχουσα, προσαρμοσμένες στις δικές τους ανάγκες και στόχους. Παράλληλα, η υπερπλατφόρμα θα παρέχει λειτουργίες ιστορικού, διατηρώντας πληροφορίες για τη συμμετοχή και την απόδοση σε διάφορους διαγωνισμούς, καθιστώντας τη χρήσιμη για παρακολούθηση και ανάλυση δεδομένων.

Λειτουργίες και Δυνατότητες της Υπερπλατφόρμας

- **Δημιουργία Πλατφορμών:** Οι χρήστες θα μπορούν να δημιουργούν τις δικές τους εξατομικευμένες πλατφόρμες, επιλέγοντας από προκαθορισμένα πρότυπα και παραμετροποιώντας τις λειτουργίες που επιθυμούν, όπως η υποβολή ιδεών, η επικοινωνία, η διαχείριση έργων, κ.λπ.
- **Ιστορικό Διαγωνισμών:** Η υπερπλατφόρμα θα καταγράφει αυτόματα το ιστορικό συμμετοχής κάθε πλατφόρμας στους διαγωνισμούς, συμπεριλαμβανομένων στοιχείων όπως η ημερομηνία συμμετοχής, οι ιδέες που υποβλήθηκαν, η πρόοδος υλοποίησης και τα αποτελέσματα. Αυτό το ιστορικό θα είναι διαθέσιμο τόσο στους διαχειριστές όσο και στους χρήστες της πλατφόρμας για ανάλυση και βελτιστοποίηση στρατηγικών.
- **Κοινόχρηστα Εργαλεία και Υποστήριξη:** Οι νέες πλατφόρμες θα ενσωματώνουν κοινά εργαλεία της υπερπλατφόρμας, όπως λειτουργίες chat, παρακολούθηση ιδεών, και κατοχύρωση, ενώ θα υποστηρίζονται από κεντρικές τεχνολογικές υποδομές.
- **Συνεργατική Κοινότητα:** Η υπερπλατφόρμα θα λειτουργεί ως σημείο συνάντησης, όπου οι δημιουργοί πλατφορμών θα μπορούν να μοιράζονται βέλτιστες πρακτικές, να ανταλλάσσουν ιδέες και να συνεργάζονται σε κοινούς στόχους.
- **Εύκολη Ενσωμάτωση Νέων Τεχνολογιών:** Η υπερπλατφόρμα θα διατηρεί ενσωματωμένες λειτουργίες για την υποστήριξη τεχνολογιών αιχμής, όπως η τεχνητή νοημοσύνη, η ανάλυση δεδομένων και τα εργαλεία αυτοματοποίησης.
- Οφέλη της Υπερπλατφόρμας

Η εισαγωγή μιας τέτοιας υπερπλατφόρμας αναμένεται να προσφέρει πολλαπλά οφέλη:

- **Εξεικονόμενη Πύρωση:** Οι χρήστες δεν θα χρειάζονται εξειδικευμένες γνώσεις ή σημαντική επένδυση για να δημιουργήσουν και να διαχειριστούν τις δικές τους πλατφόρμες.
- **Ευελξία και Επεκτασιμότητα:** Η δυνατότητα προσαρμογής της κάθε πλατφόρμας στις ανάγκες του χρήστη εξασφαλίζει ότι μπορεί να καλύψει διαφορετικούς σκοπούς, από επιχειρηματικούς διαγωνισμούς μέχρι εκπαιδευτικές πρωτοβουλίες.
- **Κοινή Μάθηση και Συνεργασία:** Το ιστορικό και η συλλογή δεδομένων από διαφορετικές πλατφόρμες θα δημιουργούν μια πλούσια βάση γνώσης που θα βοηθά στη λήψη αποφάσεων και στην υιοθέτηση βέλτιστων πρακτικών.

Η υλοποίηση της υπερπλατφόρμας θα αποτελέσει ένα σημαντικό βήμα προς την παροχή εργαλείων που ενισχύουν τη δημιουργικότητα και τη συνεργασία, δίνοντας τη δυνατότητα στους χρήστες να αναπτύσσουν εξατομικευμένες λύσεις και να παρακολουθούν την πρόοδο και τα επιτεύγματά τους σε βάθος χρόνου.

7.4 Δημιουργία chat bot

Η προσθήκη ενός chatbot στην πλατφόρμα, το οποίο θα είναι διαθέσιμο για την παροχή απαντήσεων σε συχνές ερωτήσεις και για την παροχή ενημερώσεων σχετικά με τους διαγωνισμούς, αποτελεί μία ακόμα καινοτόμα πρόταση για τη βελτίωση της εμπειρίας των χρηστών. Το chatbot θα λειτουργεί ως ένας έξυπνος και άμεσος ψηφιακός βοηθός, ενισχύοντας την επικοινωνία και διευκολύνοντας την πρόσβαση σε πληροφορίες.

Λειτουργίες του Chatbot

- **Απαντήσεις σε Συχνές Ερωτήσεις (FAQ):** Το chatbot θα μπορεί να παρέχει αυτόματες απαντήσεις σε ερωτήσεις σχετικά με τη διαδικασία του διαγωνισμού, τα κριτήρια

αξιολόγησης, τις προθεσμίες, τη μορφή υποβολής και οποιαδήποτε άλλη πληροφορία είναι κρίσιμη για τους χρήστες.

- **Προσωποποιημένες Ενημερώσεις:** Οι χρήστες θα μπορούν να λαμβάνουν εξατομικευμένες ενημερώσεις σχετικά με την πρόοδο των αιτήσεών τους, τη συμμετοχή τους σε διαγωνισμούς ή τις αλλαγές στους όρους και τις προθεσμίες.
- **Υπενθυμίσεις:** Το chatbot θα παρέχει υπενθυμίσεις για κρίσιμες ημερομηνίες, όπως η προθεσμία υποβολής ιδεών ή η ανακοίνωση αποτελεσμάτων.
- **Οδηγίες και Υποστήριξη:** Οι χρήστες θα μπορούν να ζητούν οδηγίες για τη χρήση της πλατφόρμας, όπως πώς να υποβάλουν μια ιδέα, πώς να συμμετάσχουν σε μια ομάδα ή πώς να παρακολουθούν την πρόδό τους.
- **24/7 Διαθεσιμότητα:** Το chatbot θα είναι διαθέσιμο οποιαδήποτε στιγμή, εξασφαλίζοντας συνεχή υποστήριξη και άμεσες απαντήσεις στις ανάγκες των χρηστών.

Τεχνολογίες και Επεκτασιμότητα

Το chatbot θα βασίζεται σε τεχνολογίες τεχνητής νοημοσύνης, όπως φυσική επεξεργασία γλώσσας (NLP), για να κατανοεί και να ανταποκρίνεται αποτελεσματικά στις ερωτήσεις των χρηστών. Η δυνατότητα εκμάθησης (machine learning) θα επιτρέπει στο chatbot να βελτιώνεται με την πάροδο του χρόνου, αναγνωρίζοντας καλύτερα τις ανάγκες των χρηστών και προσφέροντας πιο στοχευμένες απαντήσεις.

Οφέλη του Chatbot

- **Άμεση Πρόσβαση σε Πληροφορίες:** Οι χρήστες δεν θα χρειάζεται να ανατρέχουν σε μακροσκελή έγγραφα ή να περιμένουν υποστήριξη από διαχειριστές, καθώς θα λαμβάνουν γρήγορα απαντήσεις.
- **Αποτελεσματικότητα και Εξοικονόμηση Χρόνου:** Η αυτοματοποίηση της επικοινωνίας μειώνει τον χρόνο που απαιτείται για την εξυπηρέτηση των χρηστών, επιτρέποντας στους διαχειριστές να εστιάζουν σε πιο στρατηγικές εργασίες.
- **Ενίσχυση της Εμπειρίας Χρήστη:** Το chatbot παρέχει μια πιο διαδραστική και ευχάριστη εμπειρία για τους χρήστες, βελτιώνοντας τη συνολική ικανοποίησή τους από την πλατφόρμα.

Η ενσωμάτωση του chatbot αναμένεται να αναβαθμίσει σημαντικά τη λειτουργικότητα της πλατφόρμας, ενισχύοντας την προσβασιμότητα, την υποστήριξη και την αποτελεσματικότητα της επικοινωνίας με τους χρήστες.

7.5 Ενσωμάτωση Τεχνητής Νοημοσύνης για Έγκριση και Δημιουργία Ιδεών

Η ενσωμάτωση τεχνητής νοημοσύνης (AI) στην πλατφόρμα αποτελεί μία καινοτόμα πρόταση που στοχεύει στην αυτοματοποίηση της διαδικασίας έγκρισης ιδεών και στη δημιουργία πρωτότυπων ιδεών σε ξεχωριστή κατηγορία "AI Generated Ideas". Αυτή η λειτουργία θα συμβάλλει στη βελτίωση της αποτελεσματικότητας και στη διεύρυνση του δημιουργικού δυναμικού της πλατφόρμας.

Λειτουργίες Τεχνητής Νοημοσύνης

- **Αυτόματη Έγκριση Ιδεών:** Το σύστημα AI θα αξιολογεί τις υποβαλλόμενες ιδέες που βρίσκονται σε κατάσταση αναμονής (pending) με βάση προκαθορισμένα κριτήρια, όπως η πρωτοτυπία, η συνάφεια με τον διαγωνισμό, η πληρότητα της περιγραφής και η βιωσιμότητα της υλοποίησης. Η AI θα επισημαίνει ιδέες που απαιτούν περαιτέρω ανασκόπηση από διαχειριστές, μειώνοντας τον φόρτο εργασίας και επιταχύνοντας τη διαδικασία έγκρισης.

- **Δημιουργία Ιδεών από ΑΙ:** Η πλατφόρμα θα ενσωματώνει ένα εργαλείο δημιουργίας ιδεών με τη βοήθεια ΑΙ, το οποίο θα βασίζεται σε σύγχρονες τεχνολογίες επεξεργασίας φυσικής γλώσσας (NLP) και μοντέλα δημιουργίας περιεχομένου. Οι ΑΙ-generated ιδέες θα ανήκουν σε ξεχωριστή κατηγορία "ΑΙ Generated Ideas", με στόχο την έμπνευση χρηστών, τη δημιουργία νέων ευκαιριών και τη διεύρυνση του φάσματος των προτάσεων.

Οι χρήστες θα έχουν τη δυνατότητα να υιοθετούν, να προσαρμόζουν ή να συνεργάζονται πάνω σε αυτές τις ιδέες.

Οφέλη της Ενσωμάτωσης ΑΙ

- **Αυτοματοποίηση και Ταχύτητα:** Η διαδικασία έγκρισης γίνεται ταχύτερη και πιο αντικειμενική, μειώνοντας τον χρόνο αναμονής για τους χρήστες.
- **Ενίσχυση Δημιουργικότητας:** Η δημιουργία ιδεών από ΑΙ λειτουργεί ως εργαλείο έμπνευσης, δίνοντας στους χρήστες αφετηρίες για να αναπτύξουν καινοτόμες προτάσεις.
- **Μείωση Ανθρώπινου Λάθους:** Η ΑΙ μπορεί να αναλύσει μεγάλο όγκο δεδομένων με ακρίβεια, εντοπίζοντας σημαντικές λεπτομέρειες που ενδέχεται να παραβλεφθούν από τους διαχειριστές.
- **Διαφάνεια και Δικαιοσύνη:** Η χρήση προκαθορισμένων κριτηρίων έγκρισης από την ΑΙ διασφαλίζει ότι όλες οι ιδέες αξιολογούνται με συνέπεια και αμεροληψία.
- **Προσαρμογή στις Ανάγκες:** Οι ΑΙ-generated ιδέες μπορούν να προσαρμόζονται στις ανάγκες διαφορετικών κατηγοριών διαγωνισμών ή προγραμμάτων, εξασφαλίζοντας μεγαλύτερη συνάφεια.

Προκλήσεις και Διαχείρισή τους

- **Διασφάλιση Ποιότητας:** Οι ΑΙ-generated ιδέες θα υποβάλλονται σε επιπλέον έλεγχο από διαχειριστές, ώστε να διασφαλίζεται η ποιότητά τους.
- **Αποδοχή από Χρήστες:** Θα υπάρχει σαφής διαχωρισμός ανάμεσα σε ιδέες που υποβάλλονται από χρήστες και σε αυτές που δημιουργούνται από ΑΙ, ώστε να αποφεύγονται παρεξηγήσεις.

Η ενσωμάτωση τεχνητής νοημοσύνης στην πλατφόρμα θα αποτελέσει έναν σημαντικό παράγοντα ενίσχυσης της λειτουργικότητας και της καινοτομίας, συμβάλλοντας στη διαμόρφωση ενός σύγχρονου και δυναμικού περιβάλλοντος για τους χρήστες.

7.6 Πιστοποίηση Χρήστη και Σύστημα Κατάταξης

Η εισαγωγή λειτουργιών για τον χαρακτηρισμό των χρηστών ως πιστοποιημένων (User Authentication) και για τη δημιουργία συστήματος κατάταξης (Ranking System) αποτελεί ένα σημαντικό βήμα για την ενίσχυση της αξιοπιστίας και της συμμετοχής στην πλατφόρμα.

- Πιστοποιημένος Χρήστης (User Authentication)

Η πιστοποίηση των χρηστών θα εξασφαλίζει ότι τα προφίλ τους είναι αυθεντικά και αξιόπιστα, ενισχύοντας τη διαφάνεια και την ασφάλεια στην πλατφόρμα.

- Διαδικασία Πιστοποίησης:

Οι χρήστες θα μπορούν να πιστοποιούν το προφίλ τους μέσω:

- Επαλήθευσης email.
- Σύνδεσης με επίσημες ταυτότητες (π.χ., μέσω εγκεκριμένων τρίτων όπως το eIDAS).
- Επαλήθευσης κοινωνικών προφίλ.
-
- Διακριτικά Πιστοποίησης:

Ένα ειδικό σήμα πιστοποίησης θα εμφανίζεται στο προφίλ του χρήστη, υποδεικνύοντας ότι έχει ολοκληρώσει τη διαδικασία επαλήθευσης.

- Οφέλη Πιστοποίησης:

Πρόσβαση σε επιπλέον δυνατότητες, όπως δικαίωμα να κατοχυρώνουν ιδέες ή να συμμετέχουν σε ειδικούς διαγωνισμούς.

Αυξημένη εμπιστοσύνη από την κοινότητα.

- Σύστημα Κατάταξης (Ranking System)

Το σύστημα κατάταξης θα βασίζεται σε μια δυναμική αξιολόγηση των χρηστών, λαμβάνοντας υπόψη τη συμμετοχή και την αποδοχή των ιδεών τους από την κοινότητα.

Κριτήρια Κατάταξης:

- Η κατάταξη του χρήστη θα υπολογίζεται με βάση:
- **Ποσότητα Ιδεών:** Ο συνολικός αριθμός ιδεών που έχει υποβάλει ο χρήστης.
- **Αποδοχή Ιδεών:** Ο αριθμός των likes ή άλλων μορφών θετικής αξιολόγησης που έχουν λάβει οι ιδέες του.
- Μηχανισμός Βαθμολόγησης:

Κάθε ιδέα θα συνεισφέρει σε έναν συντελεστή κατάταξης (ranking score), ο οποίος θα βασίζεται στον τύπο:

$$\text{Ranking Score} = \alpha \cdot \text{Ideas Count} + \beta \cdot \text{Total Likes}$$

όπου α και β είναι προσαρμόσιμοι συντελεστές βαρύτητας.

- Κατηγορίες Κατάταξης:

Οι χρήστες θα κατηγοριοποιούνται σε επίπεδα, π.χ.:

- **Newbie Innovator:** Χρήστες με λίγες ιδέες ή μικρή αποδοχή.
- **Creative Contributor:** Χρήστες μεσαίας συνεισφοράς και αποδοχής.
- **Top Innovator:** Χρήστες με σημαντική δραστηριότητα και επιρροή στην πλατφόρμα.

- Οφέλη του Συστήματος
- **Αναγνώριση και Κίνητρο:** Οι χρήστες με υψηλότερη κατάταξη θα αποκτούν προβολή και πρόσβαση σε αποκλειστικά προνόμια, παρακινώντας τους να συμμετέχουν ενεργά.
- **Διαφάνεια και Δικαιοσύνη:** Το σύστημα κατάταξης παρέχει αντικειμενική μέτρηση της συνεισφοράς κάθε χρήστη, ενισχύοντας τη δέσμευση και τη συμμετοχή στην κοινότητα.
- **Βελτίωση Περιεχομένου:** Η αξιολόγηση από την κοινότητα ενθαρρύνει τη δημιουργία ποιοτικών ιδεών, βελτιώνοντας συνολικά το περιεχόμενο της πλατφόρμας.

Αυτή η λειτουργία θα προσδώσει στην πλατφόρμα μια νέα διάσταση αλληλεπίδρασης και ανταγωνισμού, ενισχύοντας τη δυναμική της κοινότητας και την αξία των συμμετοχών.

7.7 Επίλογος

Συνοψίζοντας, οι προτεινόμενες βελτιστοποιήσεις και λειτουργίες αποσκοπούν στη συνεχή αναβάθμιση της πλατφόρμας, ώστε να προσφέρει ένα δυναμικό, διαδραστικό και καινοτόμο περιβάλλον για τους χρήστες της. Η ενσωμάτωση εργαλείων όπως το chatbot, η τεχνητή νοημοσύνη για αξιολόγηση και δημιουργία ιδεών, το σύστημα κατάταξης, η δυνατότητα πιστοποίησης χρηστών, καθώς και η ανάπτυξη υπερπλατφόρμας, ενισχύουν τη λειτουργικότητα, την αποδοτικότητα και την εμπειρία χρήστη.

Κάθε μία από αυτές τις βελτιώσεις έχει σχεδιαστεί για να ανταποκρίνεται στις ανάγκες της κοινότητας, ενδυναμώνοντας τη συνεργασία, την καινοτομία και τη διαφάνεια. Παράλληλα, αποτελούν τη βάση για τη διαμόρφωση ενός περιβάλλοντος που προωθεί τη δημιουργικότητα, ενώ παραμένει φιλικό και προσβάσιμο για όλους.

Η υλοποίηση αυτών των αλλαγών δεν αποτελεί μόνο ένα τεχνολογικό επίτευγμα, αλλά και μια στρατηγική επένδυση στο μέλλον της πλατφόρμας, καθώς θέτει τις βάσεις για συνεχή εξέλιξη και ανάπτυξη. Μέσω αυτών των βελτιώσεων, η πλατφόρμα φιλοδοξεί να αποτελέσει πρότυπο για παρόμοιες πρωτοβουλίες, ενισχύοντας τη συμβολή της στον κόσμο της καινοτομίας και της δημιουργικότητας.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Websites :

- [1] J. Alves, W. Yang, Cognitive mechanisms in entrepreneurship competence: its implication for open innovation, *J. Open Innov.: Technol., Market, Complex.* 8 (2022), <https://doi.org/10.3390/joitmc8020065>. Article 65.
- [2] S. Adamczyk, A.C. Bullinger, K.M. Möslin, Innovation contests: a review, classification and outlook, *Creativ. Innovat. Manag.* 21 (2012) 335–360, <https://doi.org/10.1111/caim.12003>.
- [3] M. Alba, M. Avalos, C. Guzmán, V.M. Larios, Synergy between smart cities' hackathons and living labs as a Vehicle for Accelerating Tangible Innovations on Cities, in: *Proceedings of the 2016 IEEE International Smart Cities Conference (ISC2)*, Trento, Italy, 2016, pp. 1–6, <https://doi.org/10.1109/ISC2.2016.7580877>.
- [4] J. Chan, K. Husted, Dual allegiance and knowledge sharing in open source software firms, *Creativ. Innovat. Manag.* 19 (2010) 314–326, <https://doi.org/10.1111/j.1467-8691.2010.00566.x>.
- [5] G. Juell-Skielse, A. Hjalmarsson, P. Johannesson, D. Rudmark, Is the public motivated to engage in open data innovation? in: M. Janssen, H.J. Scholl, M.A. Wimmer, F. Bannister (Eds.), *Electronic Government, EGOV 2014. Lecture Notes in Computer Science Springer*, 2014, pp. 277–288, https://doi.org/10.1007/978-3-662-44426-9_23.
- [6] B. Rosell, S. Kumar, J. Shepherd, Unleashing innovation through internal hackathons, in: *Proceedings of the 2014 IEEE Innovations in Technology Conference (InnoTek)*, Warwick, 2014, pp. 1–8, <https://doi.org/10.1109/InnoTek.2014.6877369>.
- [7] P. Hernández-Dionis, D. Pérez-Jorge, O. Curbelo-González, O.M. Alegre de la Rosa, The coordinator of information and communication technologies: its implication for open innovation, *J. Open Innov.: Technol., Market, Complex.* 8 (2022), <https://doi.org/10.3390/joitmc8010042>. Article 42.
- [8] Open Knowledge Foundation Greece, "Open Knowledge Foundation Greece," [Online]. Available: <https://okfn.gr/>. [Accessed: 17-Jan-2025].
- [9] Kitsios, F. and Kamariotou, M. (2023). Digital innovation and entrepreneurship through open data-based platforms: Critical success factors for hackathons. *Heliyon*, 9(4), p.e14868. doi:<https://doi.org/10.1016/j.heliyon.2023.e14868>.
- [10] Chau, C.W. and Gerber, E.M. (2023). On Hackathons: A Multidisciplinary Literature Review. doi:<https://doi.org/10.1145/3544548.3581234>.
- [11] M. Rys, Invention Development. The Hackathon Method, *Knowledge Management Research & Practice* (2021) 1–13, <https://doi.org/10.1080/14778238.2021.1911607>.
- [12] Apache Friends, "XAMPP," [Online]. Available: <https://www.apachefriends.org/>. [Accessed: 17-Jan-2025].
- [13] Composer, "Introduction to Composer," [Online]. Available: <https://getcomposer.org/doc/00-intro.md>. [Accessed: 17-Jan-2025].
- [14] Composer, "Basic Usage," [Online]. Available: <https://getcomposer.org/doc/01-basic-usage.md>. [Accessed: 17-Jan-2025].
- [15] Packagist, "Packagist," [Online]. Available: <https://packagist.org/>. [Accessed: 17-Jan-2025].

[16] Visual Paradigm, "What is UML?," [Online]. Available: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/>. [Accessed: 17-Jan-2025].

[17] phpMyAdmin, "phpMyAdmin," [Online]. Available: <https://www.phpmyadmin.net/>. [Accessed: 17-Jan-2025].

[18] Laravel, "Migrations," [Online]. Available: <https://laravel.com/docs/5.0/migrations>. [Accessed: 17-Jan-2025].

ΠΑΡΑΡΤΗΜΑ Α : ΤΙΤΛΟΣ ΠΑΡΑΡΤΗΜΑΤΟΣ

...