

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Διαδικτυακή εφαρμογή διαχείρισης ηλεκτρονικών αιτημάτων υποστήριξης»

Ticket System Dashboard Logout

Διαχείριση Αιτημάτων

Δείξε 10 εγγραφές Αναζήτηση

#	Χρήστης	Κατηγορία	Τίτλος	Περιγραφή	Κατάσταση	Ημερομηνία Δημιουργίας	Ενέργειες
1	Βουδούρης Κώστας	Πρόβλημα με τη σύνδεση	Δεν αναγνωρίζει το email μου. Δεν μπορώ να συνδεθώ με το voua@gmail.com.	Σε Εξέλιξη	14-01-2025 16:11	Ακύρωση Απόκριση Αλλαγή Κατάστασης Γραβιάρι Αποκρίσεων	
2	Βουδούρης Κώστας	Πρόβλημα το ανέβασμα αρχείου	Δεν μπορώ να ανεβάσω το αρχείο μου	Δεν μπορώ να ανεβάσω το αρχείο μου	Ακύρωση	15-01-2025 09:29	Επιστροφή Απόκριση Αλλαγή Κατάστασης Γραβιάρι Αποκρίσεων
3	Φοιτητής 3	Πρόβλημα την αποθήκευση	Πρόβλημα με αποθήκευση	Απώλεια δεδομένων.	Ανοικτό	15-01-2025 14:20	Ακύρωση Απόκριση Αλλαγή Κατάστασης Γραβιάρι Αποκρίσεων
4	Φοιτητής 3	Γενικό πρόβλημα	Γενικό πρόβλημα	Καθυστέρηση στη φόρτωση σελίδας	Κλειστό	15-01-2025 14:30	Ακύρωση Απόκριση Αλλαγή Κατάστασης Γραβιάρι Αποκρίσεων

Φοιτητής

ΚΩΝΣΤΑΝΤΙΝΟΣ ΒΟΥΔΟΥΡΗΣ 516012

Επιβλέπων

Δρ. Κυριάκος Τσιακμάκης

Φεβρουάριος 2025

Διαδικτυακή εφαρμογή διαχείρισης ηλεκτρονικών αιτημάτων υποστήριξης

Κωδικός: **24171**

Φοιτητής: ΚΩΝΣΤΑΝΤΙΝΟΣ ΒΟΥΔΟΥΡΗΣ

Εισηγητής: Δρ Κυριάκος Τσιακμάκης

Ημερομηνία ανάληψης Π.Ε. 27-03-2024

Ημερομηνία περάτωσης Π.Ε. 20-01-2025

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 πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή **Κωνσταντίνου Βουδούρη** 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 πτυχιακής 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.

Περίληψη

Η εργασία αυτή αφορά τη δημιουργία μιας διαδικτυακής εφαρμογής διαχείρισης αιτημάτων υποστήριξης για τη διευκόλυνση των φοιτητών και τη βελτίωση της λειτουργίας ενός τμήματος. Οι χρήστες (φοιτητές) έχουν τη δυνατότητα να υποβάλουν αιτήματα υποστήριξης σχετικά με προβλήματα όπως σύνδεση, ανέβασμα αρχείων, αποθήκευση δεδομένων ή γενικά θέματα. Οι διαχειριστές (admins) μπορούν να διαχειρίζονται αυτά τα αιτήματα, να αλλάζουν την κατάστασή τους (π.χ. Ανοιχτό, Σε Εξέλιξη, Κλειστό), να απαντούν και να καταγράφουν όλες τις ενέργειες. Η εφαρμογή είναι υλοποιημένη σε Laravel, χρησιμοποιεί MySQL για τη διαχείριση δεδομένων και Bootstrap για ένα σύγχρονο και φιλικό προς τον χρήστη περιβάλλον.

« Online Support Ticket Management Application »

Abstract

This project involves the development of an online application for managing support requests to assist students and enhance the operations of a department. Users (students) can submit support requests related to issues such as connectivity, file uploads, data storage, or general concerns. Administrators (admins) can manage these requests, update their statuses (e.g., Open, In Progress, Closed), respond to them, and log all user actions. The application is built using Laravel, utilizes MySQL for data management, and employs Bootstrap for the interface.

Ευχαριστίες

Θέλω να εκφράσω την ευγνωμοσύνη μου στους γονείς μου για την υποστήριξή τους και στον επιβλέποντα για την πολύτιμη καθοδήγηση, τις επιστημονικές του συμβουλές και τη συμβολή του στην ανάπτυξη του κώδικα της εφαρμογής.

Περιεχόμενα

Περίληψη	iv
Abstract	v
Ευχαριστίες	vi
Περιεχόμενα.....	vii
Κατάλογος Σχημάτων	viii
Κεφάλαιο 1ο: Εισαγωγή.....	9
1.1 Εισαγωγή.....	9
1.2 Συνεισφορά της εργασίας.....	10
1.3 Δομή της εργασίας.....	10
Κεφάλαιο 2ο: Βιβλιογραφική Ανασκόπηση – Παρόμοια συστήματα.....	11
2.1 Εισαγωγή.....	11
2.2 Zendesk.....	11
2.3 Freshdesk.....	12
2.4 Jira Issue & Project Tracking Software Atlassian.....	13
2.5 Ticketing System: A Descriptive Research on the Use of Ticketing System for Project Management and Issue Tracking in IT Companies.....	15
2.6 Enhancing a Ticketing System with AI Intelligence	16
Κεφάλαιο 3ο: Τεχνολογία και εργαλεία	17
3.1 Laravel.....	17
3.2 PHP.....	21
3.3 Bootstrap.....	36
3.4 MySQL.....	39
Κεφάλαιο 4ο: Το σύστημα Ticket	42
4.1 Το σύστημα και οι διαδικασίες.....	42
4.1.1 Υποβολή Αιτήματος από τον Χρήστη.....	42
4.1.2 Διαχείριση Αιτημάτων από τους Διαχειριστές.....	42
4.1.3 Καταγραφή Ενεργειών	43
4.1.4 Τεχνολογίες και Σχεδιασμός του Συστήματος	43
4.1.5 Ειδικές Λειτουργίες.....	43

4.2	Παρουσίαση Διεπαφής Χρήστη.....	49
4.3	Η Βάση.....	56
4.4	Ασφάλεια στο σύστημα και στα δεδομένα	59
Κεφάλαιο 5ο:	Συμπεράσματα και προτάσεις βελτίωσης	62
BIBΛΙΟΓΡΑΦΙΑ.....		64
ΠΑΡΑΡΤΗΜΑ Α		66

Κατάλογος Σχημάτων

Εικόνα 4.1:	Διαδικασία Αιτήματος.....	45
Εικόνα 4.2:	Διαδικασία Αιτήματος σε προβολή Use Case Diagram.....	46
Εικόνα 4.3:	Λειτουργίες του χρήστη και του διαχειριστή.....	47
Εικόνα 4.4:	Διάγραμμα ροής αλληλεπίδρασης συστήματος.....	48
Εικόνα 4.5:	Σελίδα υποδοχής	49
Εικόνα 4.6:	Σελίδα Εγγραφής	50
Εικόνα 4.7:	Σελίδα σύνδεσης	50
Εικόνα 4.8:	Η κεντρική σελίδα Dashboard για τον απλό χρήστη.....	51
Εικόνα 4.9:	Η σελίδα με τα αιτήματα που έχει κάνει ο χρήστης	51
Εικόνα 4.10:	Η κεντρική σελίδα Dashboard για τον admin	52
Εικόνα 4.11:	Διαχείριση Αιτημάτων από τον Admin	53
Εικόνα 4.12:	Παράθυρο διαλόγου για την υποβολή απάντησης σε αίτημα	53
Εικόνα 4.13:	Παράθυρο διαλόγου για την αλλαγή κατάστασης ενός αιτήματος	54
Εικόνα 4.14:	Παράθυρο διαλόγου για τις απαντήσεις για ένα αίτημα	54
Εικόνα 4.15:	Παράθυρο διαλόγου για την ακύρωση.....	55
Εικόνα 4.16:	Πίνακα καταγραφής ενεργειών του admin.....	55
Εικόνα 4.17:	Προβολή αιτήματος όταν ο admin πατήσει πάνω στο αίτημα στον πίνακα καταγραφής ενεργειών	56
Εικόνα 4.18:	Η βάση με τους πίνακες	56

Κεφάλαιο 1ο: Εισαγωγή

1.1 Εισαγωγή

Η διαχείριση αιτημάτων υποστήριξης αποτελεί βασικό στοιχείο για τη βελτίωση της αποτελεσματικότητας σε διάφορους οργανισμούς. Η ανάπτυξη τεχνολογικών λύσεων σε αυτόν τον τομέα αποσκοπεί στη γρήγορη και οργανωμένη επίλυση προβλημάτων που αντιμετωπίζουν οι χρήστες, εξοικονομώντας χρόνο και πόρους. Επιπλέον, ενισχύεται η διαφάνεια στις διαδικασίες, αφού οι χρήστες έχουν πρόσβαση σε ένα σύστημα που καταγράφει όλες τις ενέργειες που σχετίζονται με το αίτημά τους.

Ιδιαίτερα στα εκπαιδευτικά ιδρύματα, η διαχείριση αιτημάτων υποστήριξης διαδραματίζει σημαντικό ρόλο. Οι φοιτητές, συχνά, αντιμετωπίζουν ζητήματα όπως προβλήματα πρόσβασης σε ηλεκτρονικές πλατφόρμες, ελλιπής σύνδεση στο δίκτυο ή αποτυχία αποθήκευσης δεδομένων. Ένα καλά οργανωμένο σύστημα υποστήριξης εξασφαλίζει ότι τα προβλήματα αυτά μπορούν να αναγνωριστούν και να επιλυθούν άμεσα, προσφέροντας παράλληλα καλύτερη εμπειρία στους χρήστες.

Η ανάγκη για ψηφιοποίηση και αυτοματοποίηση τέτοιων διαδικασιών γίνεται ολοένα και πιο σημαντική. Με την εφαρμογή σύγχρονων τεχνολογιών, όπως τα διαδικτυακά συστήματα διαχείρισης αιτημάτων, οι οργανισμοί αποκτούν τη δυνατότητα να ανταποκρίνονται γρήγορα στις ανάγκες των χρηστών τους, ενισχύοντας τη συνεργασία και την αποτελεσματικότητα μεταξύ των εμπλεκόμενων μερών.

Η εργασία αυτή αφορά την ανάπτυξη μιας διαδικτυακής εφαρμογής διαχείρισης αιτημάτων υποστήριξης, η οποία επιτρέπει στους φοιτητές να υποβάλουν αιτήματα για τεχνική υποστήριξη και στους διαχειριστές να τα διαχειρίζονται με δομημένο και αποτελεσματικό τρόπο. Το σύστημα βασίζεται στο PHP framework Laravel, με χρήση MySQL για τη διαχείριση των δεδομένων και Bootstrap.

Η εφαρμογή προσφέρει δυνατότητες όπως:

- Υποβολή αιτημάτων από φοιτητές.
- Ανάθεση αιτημάτων σε συγκεκριμένες κατηγορίες.
- Απάντηση και αλλαγή κατάστασης αιτημάτων από διαχειριστές.
- Καταγραφή όλων των ενεργειών για διαφάνεια και παρακολούθηση.

Ο κύριος στόχος της εφαρμογής είναι να παρέχει ένα ασφαλές και φιλικό προς τον χρήστη περιβάλλον, επιτρέποντας τη γρήγορη επίλυση προβλημάτων και τη βελτίωση της εξυπηρέτησης εντός του Τμήματος.

1.2 Συνεισφορά της εργασίας

Η εργασία αυτή παρουσιάζει ένα ολοκληρωμένο σύστημα διαχείρισης αιτημάτων υποστήριξης, ειδικά προσαρμοσμένο στις ανάγκες ενός εκπαιδευτικού ιδρύματος. Η συνεισφορά της εργασίας περιλαμβάνει:

- **Διαφάνεια και οργάνωση:** Όλες οι ενέργειες καταγράφονται, διασφαλίζοντας την ιχνηλασιμότητα.
- **Απλοποίηση διαδικασιών:** Οι φοιτητές έχουν τη δυνατότητα να υποβάλουν αιτήματα εύκολα, μέσω μιας σύγχρονης διεπαφής.
- **Αποτελεσματική διαχείριση:** Οι διαχειριστές μπορούν να οργανώνουν, να παρακολουθούν και να απαντούν στα αιτήματα γρήγορα.
- **Ενσωμάτωση σύγχρονων τεχνολογιών:** Χρήση Laravel, MySQL και Bootstrap για την ανάπτυξη μιας ασφαλούς και λειτουργικής εφαρμογής.

Το μεγαλύτερο πλεονέκτημα της εφαρμογής είναι η δυνατότητα άμεσης επικοινωνίας μεταξύ φοιτητών και διαχειριστών, μειώνοντας το χρόνο επίλυσης προβλημάτων.

1.3 Δομή της εργασίας

Το περιεχόμενο της εργασίας διαρθρώνεται ως εξής:

1. **Εισαγωγή:** Παρουσιάζεται η αναγκαιότητα του συστήματος, οι στόχοι και η συνεισφορά της εφαρμογής.
2. **Βιβλιογραφική Ανασκόπηση:** Εξετάζονται παρόμοια συστήματα και η συμβολή τους στη διαχείριση αιτημάτων.
3. **Τεχνολογία και Εργαλεία:** Αναλύονται οι τεχνολογίες που χρησιμοποιήθηκαν και τα εργαλεία που ενσωματώθηκαν.
4. **Υλοποίηση και Λειτουργικότητα:** Παρουσιάζονται τα βασικά χαρακτηριστικά της εφαρμογής, με διαγράμματα και στιγμιότυπα οθόνης.
5. **Συμπεράσματα και Μελλοντική Έρευνα:** Συνοψίζονται τα αποτελέσματα της εργασίας και προτείνονται βελτιώσεις.

Η δομή της εργασίας εξασφαλίζει μια ολοκληρωμένη προσέγγιση στην ανάλυση, την ανάπτυξη και την αξιολόγηση της εφαρμογής.

Κεφάλαιο 2ο: Βιβλιογραφική Ανασκόπηση – Παρόμοια συστήματα

2.1 Εισαγωγή

Σκοπός αυτού του κεφαλαίου είναι να παρουσιαστούν υπάρχοντα online συστήματα που έχουν αναπτυχθεί για τη διαχείριση αιτημάτων υποστήριξης σε διάφορους τομείς, όπως η εκπαίδευση, η υγεία, η τεχνολογία και οι επιχειρήσεις. Μέσα από την ανάλυση αυτών των συστημάτων, θα εντοπιστούν καλές πρακτικές που μπορούν να εφαρμοστούν στο παρόν έργο, αλλά και τα κενά που υπάρχουν και μπορούν να καλυφθούν με νέες λειτουργικότητες.

Σε κάθε υποκεφάλαιο, παρουσιάζεται ένα συγκεκριμένο σύστημα, περιγράφονται τα χαρακτηριστικά του, οι τεχνολογίες που χρησιμοποιούνται και οι βασικές λειτουργίες του. Στη συνέχεια, αξιολογούνται τα πλεονεκτήματα και οι αδυναμίες του σε σχέση με τις ανάγκες που εξυπηρετεί.

2.2 Zendesk

Το Zendesk είναι ένα από τα πιο διαδεδομένα συστήματα διαχείρισης αιτημάτων υποστήριξης, που χρησιμοποιείται ευρέως από επιχειρήσεις για την παροχή υποστήριξης στους πελάτες τους. Είναι μια πλατφόρμα που βασίζεται στο cloud, επιτρέποντας στους χρήστες να καταγράφουν, να οργανώνουν και να παρακολουθούν τα αιτήματα υποστήριξης μέσω ενός φιλικού περιβάλλοντος χρήστη.

Κύρια Χαρακτηριστικά:

1. Πολυκαναλική υποστήριξη: Το Zendesk επιτρέπει στους χρήστες να υποβάλλουν αιτήματα μέσω email, chat, τηλεφώνου, κοινωνικών δικτύων και μιας προσαρμοσμένης πύλης πελατών.
2. Αυτοματοποιημένες ροές εργασίας: Παρέχει αυτοματισμούς που κατευθύνουν τα αιτήματα στις κατάλληλες ομάδες ή υπαλλήλους, μειώνοντας το χρόνο διαχείρισης.
3. Διαχείριση SLA (Service Level Agreements): Το σύστημα παρακολουθεί τα επίπεδα εξυπηρέτησης, διασφαλίζοντας την έγκαιρη απάντηση σε αιτήματα.
4. Αναφορές και Στατιστικά: Το Zendesk διαθέτει εργαλεία ανάλυσης που παρέχουν δεδομένα σχετικά με την απόδοση της ομάδας υποστήριξης και τη διαχείριση των αιτημάτων.
5. Ενσωμάτωση με άλλες πλατφόρμες: Μπορεί να συνδεθεί με εργαλεία όπως το Slack, το Jira, το Salesforce κ.ά.

Τεχνολογίες: Το Zendesk είναι κατασκευασμένο με τεχνολογίες web, βασισμένο σε πλατφόρμες όπως Ruby on Rails για το backend και JavaScript για το frontend. Χρησιμοποιεί REST APIs για ενσωμάτωση με άλλα συστήματα και υπηρεσίες.

Πλεονεκτήματα:

- Ευελιξία στη χρήση διαφορετικών καναλιών επικοινωνίας.
- Φιλικό περιβάλλον χρήστη, ακόμα και για μη εξειδικευμένους χρήστες.
- Ισχυρά εργαλεία ανάλυσης και αναφοράς.

Περιορισμοί:

- Υψηλό κόστος συνδρομής για μικρές επιχειρήσεις ή μη κερδοσκοπικούς οργανισμούς.
- Περιορισμένη προσαρμογή στις απαιτήσεις συγκεκριμένων κλάδων.

Το Zendesk αποτελεί ένα πλήρες και επαγγελματικό σύστημα διαχείρισης αιτημάτων υποστήριξης, ιδανικό για μεγάλες επιχειρήσεις. Ωστόσο, η πολυπλοκότητά του και το υψηλό κόστος ενδέχεται να το καθιστούν απαγορευτικό για μικρότερους οργανισμούς, όπως εκπαιδευτικά ιδρύματα. Η ανάλυση του Zendesk προσφέρει ιδέες για την ενσωμάτωση λειτουργιών όπως η πολυκαναλική υποστήριξη και οι αυτοματισμοί στην ανάπτυξη του παρόντος έργου. [1]

2.3 Freshdesk

Το **Freshdesk** είναι μια ολοκληρωμένη πλατφόρμα υποστήριξης πελατών που προσφέρει διαχείριση αιτημάτων υποστήριξης για επιχειρήσεις και οργανισμούς διαφόρων μεγεθών. Η πλατφόρμα αυτή είναι σχεδιασμένη για να ενισχύει την επικοινωνία μεταξύ των ομάδων υποστήριξης και των πελατών, με έμφαση στη βελτίωση της παραγωγικότητας και της εμπειρίας χρήστη.

Βασικές Λειτουργίες

1. **Εισιτήρια Υποστήριξης:** Τα αιτήματα οργανώνονται ως εισιτήρια, επιτρέποντας την παρακολούθηση και τη διαχείριση τους μέσω ενός κεντρικού πίνακα ελέγχου.
2. **Αυτοματοποίηση Εργασιών:** Το Freshdesk προσφέρει ροές εργασίας που αυτοματοποιούν επαναλαμβανόμενες εργασίες, όπως η δρομολόγηση εισιτηρίων στις κατάλληλες ομάδες.
3. **Γνωσιακή Βάση:** Παρέχει στους χρήστες τη δυνατότητα πρόσβασης σε αυτοεξυπηρέτηση μέσω άρθρων και οδηγιών.

4. **Συνεργασία Ομάδων:** Διευκολύνει τη συνεργασία μεταξύ μελών της ομάδας μέσω κοινών σημειώσεων και εσωτερικών μηνυμάτων.
5. **Υποστήριξη Πολλαπλών Καναλιών:** Οι χρήστες μπορούν να υποβάλουν αιτήματα μέσω email, τηλεφώνου, chat, κοινωνικών δικτύων και της πύλης πελατών.

Τεχνολογίες

Το Freshdesk βασίζεται σε τεχνολογίες cloud και υποστηρίζει ενσωμάτωση με APIs για προσαρμογή στις ανάγκες του χρήστη. Η πλατφόρμα χρησιμοποιεί σύγχρονες τεχνολογίες web για την ανάπτυξη, όπως React για το frontend και Node.js για το backend.

Πλεονεκτήματα

- Εύχρηστο περιβάλλον με φιλικό σχεδιασμό.
- Δυνατότητα προσαρμογής μέσω πρόσθετων και ενσωματώσεων.
- Ισχυρή γνωσιακή βάση που μειώνει τα εισερχόμενα αιτήματα.

Μειονεκτήματα

- Περιορισμένες λειτουργίες σε χαμηλότερα πακέτα συνδρομών.
- Δεν προσφέρει ευκολίες σε οργανισμούς που χρειάζονται εξειδικευμένα χαρακτηριστικά.

Το Freshdesk είναι μια εξαιρετική επιλογή για οργανισμούς που χρειάζονται μια ευέλικτη και φιλική πλατφόρμα υποστήριξης. Παρότι στοχεύει κυρίως σε επιχειρήσεις, πολλά από τα χαρακτηριστικά του, όπως η γνωσιακή βάση και η πολυκαναλική υποστήριξη, μπορούν να εμπνεύσουν τη δημιουργία ενός προσαρμοσμένου συστήματος διαχείρισης αιτημάτων για εκπαιδευτικά ιδρύματα. [2]

2.4 Jira | Issue & Project Tracking Software | Atlassian

Το **Jira Service Management** της Atlassian είναι ένα σύστημα υποστήριξης που σχεδιάστηκε για να εξυπηρετεί οργανισμούς διαφόρων μεγεθών, κυρίως σε τεχνολογικές ομάδες και τμήματα πληροφορικής. Το σύστημα αυτό παρέχει μια ολοκληρωμένη λύση για τη διαχείριση αιτημάτων, τη συνεργασία και την παρακολούθηση διαδικασιών.

Βασικά Χαρακτηριστικά

1. **Διαχείριση Αιτημάτων ITSM (IT Service Management):** Παρέχει εργαλεία για την οργάνωση αιτημάτων που σχετίζονται με την πληροφορική, όπως προβλήματα συστημάτων ή εγκατάσταση νέου εξοπλισμού.
2. **Καταγραφή και Παρακολούθηση Εργασιών:** Οι ομάδες μπορούν να παρακολουθούν την πρόοδο κάθε αιτήματος μέσω προσαρμοσμένων ροών εργασίας.
3. **Ενσωμάτωση με Jira Software:** Διευκολύνει τη συνεργασία μεταξύ των ομάδων υποστήριξης και ανάπτυξης λογισμικού.
4. **SLA Management:** Παρέχει δυνατότητες για τον καθορισμό και την παρακολούθηση Συμφωνιών Επίπεδου Εξυπηρέτησης (SLAs).
5. **Αυτοματοποίηση:** Υποστηρίζει τη δημιουργία κανόνων για την αυτοματοποίηση επαναλαμβανόμενων εργασιών, όπως η δρομολόγηση αιτημάτων ή η αποστολή ειδοποιήσεων.

Τεχνολογίες

Το Jira Service Management χρησιμοποιεί τεχνολογίες cloud και υποστηρίζει REST APIs για ενσωμάτωση με άλλα εργαλεία. Το backend είναι βασισμένο σε Java, ενώ το frontend αναπτύσσεται με React και Atlassian UI Framework.

Πλεονεκτήματα

- Ενσωμάτωση με άλλα προϊόντα της Atlassian, όπως το Confluence και το Jira Software.
- Εξαιρετική προσαρμοστικότητα για τη δημιουργία εξατομικευμένων ροών εργασιών.
- Αναλυτικά εργαλεία για την παρακολούθηση της απόδοσης και της διαχείρισης αιτημάτων.

Μειονεκτήματα

- Μπορεί να είναι περίπλοκο για μικρές ομάδες ή οργανισμούς με περιορισμένη εμπειρία σε τεχνολογίες.
- Το κόστος ανεβαίνει καθώς αυξάνονται οι χρήστες ή τα πρόσθετα που απαιτούνται.

Το Jira Service Management είναι ιδανικό για οργανισμούς που επιθυμούν να συνδυάσουν τη διαχείριση αιτημάτων υποστήριξης με τις διαδικασίες ανάπτυξης λογισμικού. Παρότι στοχεύει σε επαγγελματίες τεχνολογίας, χαρακτηριστικά όπως η αυτοματοποίηση και η παρακολούθηση SLA

μπορούν να αποτελέσουν βάση για τη βελτιστοποίηση συστημάτων διαχείρισης αιτημάτων σε άλλα πεδία, όπως τα εκπαιδευτικά ιδρύματα. [3]

2.5 Ticketing System: A Descriptive Research on the Use of Ticketing System for Project Management and Issue Tracking in IT Companies

Το άρθρο "**Ticketing System: A Descriptive Research on the Use of Ticketing System for Project Management and Issue Tracking in IT Companies**" εξετάζει τον ρόλο που διαδραματίζουν τα συστήματα ticketing στην αποτελεσματική διαχείριση έργων και την παρακολούθηση προβλημάτων σε εταιρείες πληροφορικής. Η έρευνα εστιάζει στη χρήση του συστήματος JIRA, ενός δημοφιλούς εργαλείου που χρησιμοποιείται ευρέως για τη διαχείριση έργων Agile, παρέχοντας ένα πλαίσιο για την ανάλυση και επίλυση προβλημάτων σε πραγματικό χρόνο.[4]

Το JIRA, ένα εργαλείο που αναπτύχθηκε από την Atlassian, έχει σχεδιαστεί για να διευκολύνει την παρακολούθηση προβλημάτων και τη διαχείριση έργων. Το σύστημα υποστηρίζει τη δημιουργία αιτημάτων υποστήριξης (tickets), τα οποία κατηγοριοποιούνται και ανατίθενται σε συγκεκριμένα μέλη της ομάδας. Μέσα από τις λειτουργίες του, όπως η δημιουργία backlogs, οι ενημερώσεις προόδου και οι ενσωματώσεις με άλλα εργαλεία ανάπτυξης λογισμικού, το JIRA βελτιώνει την επικοινωνία και την παραγωγικότητα στις ομάδες εργασίας.

Η έρευνα αναδεικνύει τα σημαντικά οφέλη του JIRA για τις IT εταιρείες. Μεταξύ αυτών περιλαμβάνονται η βελτιωμένη ορατότητα στα έργα, η ταχύτερη επίλυση προβλημάτων και η ενίσχυση της συνεργασίας μεταξύ των μελών της ομάδας. Επιπλέον, το σύστημα υποστηρίζει την προσαρμογή στις ανάγκες της κάθε ομάδας, επιτρέποντας τη δημιουργία custom workflows που ανταποκρίνονται στις ειδικές απαιτήσεις της εκάστοτε εταιρείας.

Παρά τα πλεονεκτήματά του, το JIRA παρουσιάζει και ορισμένους περιορισμούς. Ορισμένοι χρήστες αναφέρουν ότι η αρχική εκμάθηση του συστήματος είναι δύσκολη λόγω της πολυπλοκότητάς του. Επιπλέον, το κόστος χρήσης του συστήματος, ιδιαίτερα για μικρότερες εταιρείες, μπορεί να αποτελεί εμπόδιο. Η έρευνα προτείνει την ανάγκη για εκπαίδευση των χρηστών και τη δυνατότητα εξερεύνησης εναλλακτικών, πιο οικονομικών λύσεων.

Το JIRA αποτελεί ένα ισχυρό εργαλείο που ενισχύει την αποτελεσματικότητα στη διαχείριση έργων και την παρακολούθηση προβλημάτων. Η έρευνα καταλήγει στο συμπέρασμα ότι τα συστήματα ticketing, όπως το JIRA, είναι απαραίτητα για τη διαχείριση της πολυπλοκότητας σε IT εταιρείες, ενισχύοντας την ευελιξία και την ταχύτητα στην αντιμετώπιση των προκλήσεων. Ωστόσο, απαιτούνται

περαιτέρω προσαρμογές και υποστήριξη για να εξυπηρετήσουν πλήρως τις ανάγκες μικρότερων εταιρειών.

2.6 Enhancing a Ticketing System with AI Intelligence

Το άρθρο "Enhancing a Ticketing System with AI Intelligence" παρουσιάζει την ενσωμάτωση της τεχνητής νοημοσύνης (AI) σε παραδοσιακά συστήματα διαχείρισης αιτημάτων υποστήριξης (ticketing systems). Η χρήση της AI στοχεύει στην αυτοματοποίηση διαδικασιών, όπως η ταξινόμηση και η ανάθεση αιτημάτων, συμβάλλοντας στη μείωση του χρόνου επίλυσης προβλημάτων και στην αύξηση της αποτελεσματικότητας στις επιχειρήσεις. Η μελέτη εστιάζει στα πλεονεκτήματα που προσφέρει η AI και στις τεχνικές υλοποίησής της. [5]

Μία από τις βασικές βελτιώσεις που εξετάζονται στο άρθρο είναι η αυτόματη ταξινόμηση αιτημάτων με τη χρήση αλγορίθμων μηχανικής μάθησης (machine learning). Τα αιτήματα αναλύονται βάσει λέξεων-κλειδιών και προτύπων στο περιεχόμενό τους, επιτρέποντας στο σύστημα να αναθέτει τα αιτήματα αυτόματα στα κατάλληλα τμήματα ή υπεύθυνους. Η αυτοματοποίηση αυτή εξοικονομεί χρόνο και μειώνει τα ανθρώπινα λάθη, διευκολύνοντας την άμεση ανταπόκριση σε επείγοντα αιτήματα.

Η μελέτη παρουσιάζει επίσης τη χρήση της προγνωστικής ανάλυσης (predictive analytics) για την εκτίμηση των χρόνων επίλυσης αιτημάτων. Η AI χρησιμοποιεί ιστορικά δεδομένα για να προβλέψει πόσος χρόνος θα απαιτηθεί για την επίλυση ενός αιτήματος, βοηθώντας τις ομάδες υποστήριξης να προγραμματίσουν αποτελεσματικά τους πόρους τους. Επιπλέον, οι προβλέψεις αυτές μπορούν να χρησιμοποιηθούν για την ενημέρωση των χρηστών σχετικά με την πρόοδο του αιτήματός τους.

Η ενσωμάτωση της AI σε ticketing systems επιτρέπει την εξατομίκευση της εξυπηρέτησης χρηστών. Με τη χρήση ανάλυσης δεδομένων χρηστών και ιστορικών αιτημάτων, τα συστήματα μπορούν να προτείνουν λύσεις πριν ο χρήστης υποβάλει αίτημα. Επιπλέον, τα chatbots που βασίζονται στην AI προσφέρουν άμεση υποστήριξη, απαντώντας σε κοινές ερωτήσεις χωρίς την παρέμβαση ανθρώπων.

Η τεχνητή νοημοσύνη μπορεί να μεταμορφώσει τα παραδοσιακά συστήματα διαχείρισης αιτημάτων υποστήριξης, βελτιώνοντας την αποδοτικότητα και την εμπειρία των χρηστών. Το άρθρο υπογραμμίζει την ανάγκη για περαιτέρω επενδύσεις σε τεχνολογίες AI, ενώ παράλληλα αναγνωρίζει τις προκλήσεις που σχετίζονται με την υιοθέτηση αυτών των συστημάτων, όπως το κόστος και η εκπαίδευση του προσωπικού. Παρόλα αυτά, η ενσωμάτωση της AI προσφέρει σημαντικά οφέλη, καθιστώντας τη μία απαραίτητη προσθήκη για σύγχρονα ticketing systems.

Κεφάλαιο 3ο: Τεχνολογία και εργαλεία

Στο κεφάλαιο αυτό θα περιγράψει η τεχνολογία που χρησιμοποιήθηκε για την υλοποίηση της ηλεκτρονικής ψηφοφορίας.

3.1 Laravel

Το Laravel είναι ένα από τα πιο δημοφιλή PHP frameworks που χρησιμοποιείται για την ανάπτυξη web εφαρμογών. Σχεδιάστηκε για να κάνει τη διαδικασία ανάπτυξης ευκολότερη και πιο ευχάριστη, παρέχοντας ισχυρά εργαλεία για κοινές εργασίες, όπως η διαχείριση δεδομένων, η επικοινωνία με βάσεις δεδομένων, και η δημιουργία APIs. Το Laravel υποστηρίζει τη δημιουργία σύγχρονων εφαρμογών, με έμφαση στη σαφήνεια και τη συντηρησιμότητα του κώδικα. [6]

Βασικά Χαρακτηριστικά

1. MVC Αρχιτεκτονική

Το Laravel βασίζεται στην αρχιτεκτονική Model-View-Controller (MVC), η οποία διαχωρίζει τον κώδικα σε τρεις βασικές ενότητες: το μοντέλο για τη διαχείριση δεδομένων, τη θέα για την παρουσίαση του περιεχομένου, και τον ελεγκτή για την επεξεργασία αιτημάτων και δεδομένων. Αυτός ο διαχωρισμός καθιστά τον κώδικα πιο οργανωμένο και ευκολότερο στη συντήρηση.

2. Blade Template Engine

Το Laravel περιλαμβάνει το Blade, μια μηχανή template που επιτρέπει στους προγραμματιστές να δημιουργούν δυναμικές ιστοσελίδες με απλό τρόπο, χρησιμοποιώντας τις δομές ελέγχου της PHP.

3. Eloquent ORM

Η Eloquent ORM παρέχει ένα απλό και εκφραστικό API για την αλληλεπίδραση με βάσεις δεδομένων. Οι προγραμματιστές μπορούν να διαχειρίζονται πίνακες και εγγραφές μέσω μοντέλων, μειώνοντας την ανάγκη για χειροκίνητη SQL γραφή.

4. Artisan CLI

Το Laravel περιλαμβάνει το Artisan, ένα ισχυρό εργαλείο γραμμής εντολών για την αυτοματοποίηση κοινών εργασιών ανάπτυξης, όπως η δημιουργία μοντέλων, η μεταφορά δεδομένων, και η διαχείριση βάσεων δεδομένων.

5. Μέθοδοι Authentication και Authorization

Το Laravel παρέχει ενσωματωμένα εργαλεία για την αυθεντικοποίηση και την εξουσιοδότηση χρηστών, προσφέροντας ασφαλή και προσαρμόσιμα συστήματα πρόσβασης.

Το Laravel αποτελεί κορυφαίο PHP framework, ιδανικό για την ανάπτυξη σύγχρονων web εφαρμογών. Με την ενσωμάτωση ισχυρών εργαλείων και λειτουργιών, το Laravel προσφέρει στους προγραμματιστές την ευελιξία και την απλότητα που χρειάζονται για να δημιουργήσουν εφαρμογές υψηλής ποιότητας.[6]

Πλεονεκτήματα του Laravel

Ενσωματωμένο Σύστημα Αυθεντικοποίησης

Το Laravel παρέχει ενσωματωμένες λειτουργίες αυθεντικοποίησης χρηστών, διευκολύνοντας τη διαχείριση λογαριασμών και δικαιωμάτων πρόσβασης. Οι προγραμματιστές μπορούν εύκολα να προσθέσουν login, εγγραφή και επαναφορά κωδικού στις εφαρμογές τους.

Eloquent ORM: Ευκολία στη Διαχείριση Δεδομένων

Το Eloquent ORM (Object-Relational Mapping) προσφέρει έναν εκφραστικό και απλό τρόπο διαχείρισης βάσεων δεδομένων. Οι προγραμματιστές μπορούν να εργάζονται με πίνακες και εγγραφές σαν να είναι αντικείμενα, μειώνοντας την ανάγκη για περίπλοκες SQL εντολές.

Ευέλικτη Αρχιτεκτονική MVC

Η αρχιτεκτονική Model-View-Controller επιτρέπει την καλύτερη οργάνωση του κώδικα, χωρίζοντας τη λογική της εφαρμογής από την παρουσίαση. Αυτό καθιστά το Laravel ιδανικό για μεγάλες και σύνθετες εφαρμογές.

Blade Template Engine: Δυναμική Παρουσίαση Δεδομένων

Το Laravel ενσωματώνει τη μηχανή Blade, που επιτρέπει τη δημιουργία templates με καθαρό και επαναχρησιμοποιήσιμο κώδικα. Με δυνατότητες όπως η κληρονομικότητα views, οι προγραμματιστές μπορούν να διαχειρίζονται εύκολα τη διάταξη του περιεχομένου.

Αυτοματισμός μέσω Artisan CLI

Το Artisan Command-Line Interface επιτρέπει στους προγραμματιστές να αυτοματοποιούν κοινές εργασίες, όπως η δημιουργία μοντέλων, η εκτέλεση migrations και η διαχείριση seeds.

Ενσωματωμένα Middleware

Το Laravel υποστηρίζει middleware, που επιτρέπει τον έλεγχο των αιτημάτων και απαντήσεων της εφαρμογής. Αυτό προσφέρει πρόσθετη ασφάλεια και ευελιξία στη διαχείριση δεδομένων.

API Routing και RESTful Υποστήριξη

Το Laravel κάνει εύκολη τη δημιουργία RESTful APIs, παρέχοντας εργαλεία για τη διαχείριση routes, controllers και απαντήσεων. Η ενσωμάτωση με εργαλεία όπως το Postman και το Swagger απλοποιεί τη διαδικασία ανάπτυξης APIs.

Υποστήριξη για Unit Testing

Το Laravel ενσωματώνει εργαλεία για τη δημιουργία και την εκτέλεση unit tests. Αυτό εξασφαλίζει τη σταθερότητα της εφαρμογής κατά την ανάπτυξη νέων χαρακτηριστικών.

Μεγάλη Κοινότητα και Πηγές Υποστήριξης

Η ενεργή κοινότητα του Laravel παρέχει οδηγούς, tutorials και πακέτα που βοηθούν τους προγραμματιστές να αντιμετωπίσουν κοινές προκλήσεις. Η πληθώρα plugins και extensions μειώνει σημαντικά τον χρόνο ανάπτυξης.

Μειονεκτήματα του Laravel

Αρχική Καμπύλη Μάθησης

Αν και το Laravel προσφέρει πολλά εργαλεία, η πληθώρα των λειτουργιών του μπορεί να είναι συντριπτική για νέους προγραμματιστές. Η πλήρης κατανόηση της λειτουργίας του απαιτεί χρόνο.

Αυξημένη Χρήση Πόρων

Το Laravel είναι πιο απαιτητικό από άποψη πόρων σε σύγκριση με άλλα frameworks. Αυτό μπορεί να είναι πρόβλημα για εφαρμογές που φιλοξενούνται σε shared hosting.

Συντήρηση Μεγάλων Εφαρμογών

Σε μεγάλες εφαρμογές, η διαχείριση του κώδικα μπορεί να γίνει δύσκολη χωρίς αυστηρές πρακτικές οργάνωσης. Η χρήση πολλαπλών πακέτων μπορεί να προκαλέσει προβλήματα συμβατότητας.

Εφαρμογές

Το Laravel χρησιμοποιείται ευρέως για την ανάπτυξη σύγχρονων web εφαρμογών, χάρη στην ευελιξία, την ασφάλεια και τη μεγάλη κοινότητα υποστήριξής του. [7-11]

1. Εφαρμογές eCommerce

- Το Laravel χρησιμοποιείται για τη δημιουργία eCommerce πλατφορμών λόγω της ευκολίας στη διαχείριση παραγγελιών, χρηστών και πληρωμών.
- Παραδείγματα:
 - Bagisto (eCommerce πλατφόρμα βασισμένη στο Laravel).
 - AvonRed (ανοιχτού κώδικα eCommerce πλατφόρμα).

2. Συστήματα Διαχείρισης Περιεχομένου (CMS)

- Το Laravel υποστηρίζει την ανάπτυξη CMS που επιτρέπουν την εύκολη διαχείριση περιεχομένου από μη τεχνικούς χρήστες.
- Παραδείγματα:
 - OctoberCMS.
 - Lavalite.

3. Εφαρμογές Social Networking

- Το Laravel χρησιμοποιείται για τη δημιουργία κοινωνικών πλατφορμών που απαιτούν προηγμένη διαχείριση χρηστών και δυναμικό περιεχόμενο.
- Παράδειγμα:
 - Laravel.io, μια κοινότητα για προγραμματιστές Laravel.

4. Εφαρμογές API και Microservices

- Η υποστήριξη RESTful APIs καθιστά το Laravel ιδανικό για τη δημιουργία backend εφαρμογών που συνεργάζονται με διαφορετικές πλατφόρμες.

5. Πλατφόρμες Εκπαίδευσης

- Εφαρμογές για online εκπαίδευση, όπου οι φοιτητές και οι εκπαιδευτές αλληλεπιδρούν μέσω ενός συστήματος διαχείρισης μαθημάτων.
- Παράδειγμα:
 - Koel, μια εφαρμογή διαχείρισης μουσικής βασισμένη στο Laravel.

Ποσοστά Χρήσης Laravel

- **Παγκόσμια Χρήση:** Το Laravel είναι ένα από τα πιο δημοφιλή PHP frameworks, με πάνω από **25% των προγραμματιστών PHP** να το χρησιμοποιούν για τα έργα τους.
- **Χρήση στις Επιχειρήσεις:** Σύμφωνα με αναφορές, περίπου το **10% των επιχειρηματικών εφαρμογών PHP** βασίζεται στο Laravel.

- **Github Popularity:** Το Laravel έχει πάνω από **70.000 αστέρια (stars)** στο GitHub, δείχνοντας την ευρεία αποδοχή του στην κοινότητα των προγραμματιστών.
- **Marketplace Adoption:** Περισσότερες από **1 εκατομμύριο εφαρμογές** έχουν αναπτυχθεί χρησιμοποιώντας Laravel, σύμφωνα με αναφορές της κοινότητας.

3.2 PHP

Η PHP (Hypertext Preprocessor) είναι μια από τις πιο διαδεδομένες γλώσσες προγραμματισμού παγκοσμίως, σχεδιασμένη κυρίως για την ανάπτυξη δυναμικών ιστοσελίδων και εφαρμογών στο διαδίκτυο. Δημιουργήθηκε το 1994 από τον Rasmus Lerdorf ως ένα προσωπικό εργαλείο για την παρακολούθηση επισκεπτών στον ιστότοπό του. Η επιτυχία της πρώτης έκδοσης οδήγησε στην ευρύτερη χρήση της και στη συνεχή εξέλιξή της μέσω μιας ενεργής κοινότητας προγραμματιστών.

Αν και αρχικά χρησιμοποιήθηκε ως μια απλή γλώσσα scripting για τη δημιουργία σελίδων με HTML, η PHP εξελίχθηκε γρήγορα σε ένα ισχυρό εργαλείο ανάπτυξης ιστοσελίδων, με υποστήριξη για σύνθετα χαρακτηριστικά όπως η επικοινωνία με βάσεις δεδομένων, η διαχείριση συνεδριών και η ενσωμάτωση με διάφορες τεχνολογίες. Η γλώσσα έχει αναδειχθεί ως θεμελιώδες εργαλείο για τη δημιουργία περιεχομένου που ανταποκρίνεται στις απαιτήσεις των χρηστών σε πραγματικό χρόνο.

Η Φιλοσοφία της PHP

Η PHP έχει σχεδιαστεί με γνώμονα τη χρηστικότητα και την απλότητα. Η φιλοσοφία της βασίζεται στη λογική του “Just Work”, επιτρέποντας στους προγραμματιστές να επικεντρωθούν στη λειτουργικότητα των εφαρμογών τους χωρίς να ασχολούνται με σύνθετες διαδικασίες ρύθμισης. Η εύκολη κατανόηση της σύνταξης και η ενσωμάτωση με HTML καθιστούν τη γλώσσα φιλική προς αρχάριους, ενώ ταυτόχρονα παρέχει εργαλεία για πιο έμπειρους προγραμματιστές να δημιουργήσουν κλιμακούμενες εφαρμογές.

Η Πορεία της PHP

Από την αρχική της μορφή έως σήμερα, η PHP έχει γνωρίσει πολλές βελτιώσεις:

PHP 3.0 (1998): Η πρώτη ευρέως διαδεδομένη έκδοση, με ενσωματωμένη υποστήριξη για βάσεις δεδομένων.

PHP 4.0 (2000): Εισήγαγε το Zend Engine, βελτιώνοντας τη διαχείριση μνήμης και την απόδοση.

PHP 5.0 (2004): Πρόσθεσε αντικειμενοστραφή χαρακτηριστικά, διευρύνοντας τις δυνατότητες ανάπτυξης.

PHP 7.0 (2015): Έφερε βελτιώσεις στην ταχύτητα και εισήγαγε νέα χαρακτηριστικά, όπως το type hinting.

PHP 8.0 (2020): Περιλαμβάνει τον JIT compiler, βελτιώνοντας ακόμα περισσότερο την απόδοση.

Η PHP στη Σύγχρονη Τεχνολογία

Η PHP παραμένει ένας πυλώνας στον κόσμο του web development. Με περισσότερες από 1 εκατομμύριο ιστοσελίδες να βασίζονται σε αυτήν, η γλώσσα εξακολουθεί να διαδραματίζει πρωταγωνιστικό ρόλο. Οι δυνατότητές της επεκτείνονται από την απλή ανάπτυξη ιστοσελίδων μέχρι τη δημιουργία πλήρως λειτουργικών web εφαρμογών και APIs.

Γιατί PHP;

Η επιτυχία της PHP οφείλεται στην προσαρμοστικότητά της. Μπορεί να χρησιμοποιηθεί σε διάφορες πλατφόρμες, όπως Windows, macOS, και Linux, και υποστηρίζει πλήθος βάσεων δεδομένων. Επιπλέον, η ενεργή κοινότητα προγραμματιστών έχει δημιουργήσει πλήθος βιβλιοθηκών και εργαλείων που επεκτείνουν τις δυνατότητές της. [12-14]

Χαρακτηριστικά της PHP

Η PHP είναι γνωστή για τα χαρακτηριστικά της που την καθιστούν μια από τις πιο δημοφιλείς και ευέλικτες γλώσσες για την ανάπτυξη δυναμικών ιστοσελίδων και εφαρμογών. Αυτά τα χαρακτηριστικά την κάνουν κατάλληλη για προγραμματιστές όλων των επιπέδων εμπειρίας, από αρχάριους έως επαγγελματίες. [13-15]

1. Εκτέλεση στην Πλευρά του Διακομιστή (Server-Side Execution)

Η PHP εκτελείται αποκλειστικά στην πλευρά του διακομιστή, γεγονός που σημαίνει ότι ο χρήστης βλέπει μόνο το αποτέλεσμα του επεξεργασμένου κώδικα, όπως μια HTML σελίδα. Αυτό ενισχύει την ασφάλεια, καθώς ο κώδικας δεν εκτίθεται στους τελικούς χρήστες.

- Η δυνατότητα αυτή την καθιστά ιδανική για εφαρμογές που απαιτούν επεξεργασία δεδομένων, διαχείριση βάσεων δεδομένων και ασφαλή λειτουργία.

2. Εύκολη Ενσωμάτωση με HTML

Η PHP μπορεί να ενσωματωθεί απευθείας σε κώδικα HTML, επιτρέποντας στους προγραμματιστές να δημιουργούν δυναμικό περιεχόμενο εύκολα και γρήγορα. Αυτή η λειτουργία καθιστά τη μετάβαση από τη στατική στη δυναμική ιστοσελίδα μια απλή διαδικασία.
Παράδειγμα:

```
<!DOCTYPE html>

<html>

<body>

  <h1>Καλώς ήρθατε!</h1>

  <?php echo "Ημερομηνία: " . date("d-m-Y"); ?>

</body>

</html>
```

3. Υποστήριξη για Πολλαπλές Βάσεις Δεδομένων

Η PHP υποστηρίζει πολλούς τύπους βάσεων δεδομένων, όπως:

- **MySQL**
- **PostgreSQL**
- **SQLite**
- **MariaDB**
- **Microsoft SQL Server**

Αυτό δίνει στους προγραμματιστές τη δυνατότητα να επιλέξουν τη βάση δεδομένων που ταιριάζει στις ανάγκες του έργου τους.

4. Ευρεία Υποστήριξη για Πλατφόρμες και Λειτουργικά Συστήματα

Η PHP είναι πολυπλατφορμική και μπορεί να λειτουργήσει σε:

- **Windows**
- **Linux**
- **macOS**

Η δυνατότητα αυτή την καθιστά ιδιαίτερα ευέλικτη και ιδανική για διάφορες εφαρμογές.

5. Υποστήριξη για Αντικειμενοστραφή Προγραμματισμό (OOP)

Από την έκδοση PHP 5 και μετά, η γλώσσα παρέχει υποστήριξη για αντικειμενοστραφή προγραμματισμό, που επιτρέπει τη χρήση αντικειμένων, κλάσεων και κληρονομικότητας. Αυτό κάνει τη γλώσσα πιο ισχυρή και προσαρμόσιμη για σύνθετες εφαρμογές.

6. Εκτεταμένη Κοινότητα και Εργαλεία

Η PHP διαθέτει μια από τις μεγαλύτερες κοινότητες προγραμματιστών στον κόσμο. Αυτό μεταφράζεται σε:

- Αμέτρητα δωρεάν εργαλεία και βιβλιοθήκες.
- Υποστήριξη μέσω forums, οδηγών και βίντεο.

Παραδείγματα βιβλιοθηκών:

- **PHPMailer**: Για αποστολή email.
- **PHPExcel**: Για επεξεργασία Excel αρχείων.

7. Επεκτασιμότητα και Επεκτάσεις

Η PHP υποστηρίζει επεκτάσεις για κρυπτογράφηση, διαχείριση αρχείων, επεξεργασία εικόνων και πολλά άλλα.

Παραδείγματα:

- **GD Library**: Επεξεργασία εικόνων.
- **cURL**: Επικοινωνία με APIs.

8. Υποστήριξη για Συνεδρίες και Cookies

Η PHP διευκολύνει τη διαχείριση συνεδριών και cookies, επιτρέποντας την αποθήκευση πληροφοριών χρήστη όπως στοιχεία σύνδεσης, προτιμήσεις και άλλα.

9. Υψηλή Απόδοση και Απόκριση

Με την εισαγωγή του PHP 7 και PHP 8, η γλώσσα έχει βελτιώσει σημαντικά την ταχύτητα εκτέλεσης, κάνοντας την ιδανική για εφαρμογές που απαιτούν υψηλή απόδοση.

10. Συμβατότητα με APIs και Web Services

Η PHP υποστηρίζει τη δημιουργία και κατανάλωση APIs μέσω REST και SOAP, καθιστώντας τη κατάλληλη για εφαρμογές που χρειάζονται αλληλεπίδραση με άλλες υπηρεσίες.

Εφαρμογές της PHP

Η PHP αποτελεί τη βάση για πολλές εφαρμογές στο διαδίκτυο και είναι γνωστή για την ευελιξία και την προσαρμοστικότητά της. Από την ανάπτυξη μικρών ιστοσελίδων μέχρι τη δημιουργία σύνθετων web εφαρμογών, η PHP καλύπτει ένα ευρύ φάσμα αναγκών. [12-15]

1. Δυναμικές Ιστοσελίδες

Η PHP χρησιμοποιείται για τη δημιουργία δυναμικών ιστοσελίδων που παρέχουν περιεχόμενο προσαρμοσμένο στις ανάγκες του χρήστη.

Παραδείγματα:

- Εμφάνιση εξατομικευμένου περιεχομένου, όπως προτάσεις προϊόντων ή προσωπικές ειδοποιήσεις.
- Δημιουργία σελίδων με μεταβλητά δεδομένα, όπως ειδήσεις, blogs ή forums.

Παραδείγματα CMS (Content Management Systems) βασισμένα σε PHP:

- **WordPress:** Καλύπτει περίπου το 43% όλων των ιστοσελίδων παγκοσμίως.
- **Drupal** και **Joomla:** Ευρέως γνωστά για ευελιξία και παραμετροποίηση.

2. Web Εφαρμογές

Η PHP χρησιμοποιείται συχνά για την ανάπτυξη διαδικτυακών εφαρμογών, όπως:

- **Πλατφόρμες κοινωνικής δικτύωσης:** Η PHP αποτέλεσε τη βάση για το Facebook κατά τα πρώτα χρόνια λειτουργίας του.
- **Ηλεκτρονικό εμπόριο:** Εφαρμογές όπως το Magento και το OpenCart προσφέρουν εργαλεία διαχείρισης e-shop.

- **Εφαρμογές SaaS (Software as a Service):** Πλατφόρμες που παρέχουν υπηρεσίες μέσω διαδικτύου, όπως εργαλεία τιμολόγησης και λογισμικό διαχείρισης πελατών.

3. Συστήματα Ηλεκτρονικής Εκπαίδευσης

Η PHP αποτελεί την πλατφόρμα πίσω από δημοφιλή συστήματα ηλεκτρονικής εκπαίδευσης, όπως:

- **Moodle:** Ένα ανοιχτού κώδικα σύστημα διαχείρισης μαθημάτων που χρησιμοποιείται ευρέως σε εκπαιδευτικά ιδρύματα.
- **Chamilo:** Ένα εργαλείο εκπαίδευσης εξ αποστάσεως που βασίζεται στην PHP.

4. Επεξεργασία Δεδομένων και APIs

Η PHP χρησιμοποιείται για τη δημιουργία και κατανάλωση APIs.

Παραδείγματα:

- REST APIs για επικοινωνία με εξωτερικές εφαρμογές.
- Συλλογή και ανάλυση δεδομένων από άλλες πηγές μέσω της βιβλιοθήκης **cURL**.
- Επεξεργασία αρχείων JSON ή XML για αλληλεπίδραση με τρίτα συστήματα.

5. Δημιουργία Συστημάτων Διαχείρισης Περιεχομένου

Η PHP υποστηρίζει τη δημιουργία CMS που βοηθούν τους χρήστες να διαχειρίζονται εύκολα το περιεχόμενό τους.

Παραδείγματα:

- Εφαρμογές blogging (WordPress).
- Συστήματα διαχείρισης έργων (Project Management Systems).

6. Διαχείριση Συνεδριών και Cookies

Η PHP διευκολύνει τη δημιουργία εφαρμογών που απαιτούν διαχείριση συνεδριών και cookies.

Παραδείγματα εφαρμογών:

- Διαδικτυακές πλατφόρμες σύνδεσης χρηστών.
- Σελίδες που αποθηκεύουν προτιμήσεις χρηστών.

7. Ηλεκτρονικό Εμπόριο

Η PHP αποτελεί τη βάση για πολλά δημοφιλή εργαλεία ηλεκτρονικού εμπορίου, όπως:

- **WooCommerce:** Ενσωματώνεται με το WordPress για τη δημιουργία e-shops.
- **Magento:** Μια πλατφόρμα για μεγάλα καταστήματα.
- **PrestaShop:** Μια ελαφριά και ευέλικτη λύση για μικρομεσαίες επιχειρήσεις.

8. Chatbots και Συστήματα Υποστήριξης

Η PHP χρησιμοποιείται για την ανάπτυξη:

- Chatbots που προσφέρουν αυτόματη υποστήριξη.
- Συστήματα υποστήριξης πελατών, όπως πλατφόρμες ticketing.

9. Εφαρμογές για Εσωτερική Χρήση

Πολλοί οργανισμοί χρησιμοποιούν PHP για την ανάπτυξη εσωτερικών εφαρμογών που καλύπτουν ανάγκες όπως:

- Συστήματα διαχείρισης πελατών (CRM).
- Εργαλεία διαχείρισης εργατικού δυναμικού.

10. Δημιουργία Εργαλείων Ανάλυσης

Η PHP υποστηρίζει τη δημιουργία εργαλείων ανάλυσης δεδομένων και αναφορών, όπως:

- Αναλυτικά στατιστικά για επισκέψεις σε ιστοσελίδες.
- Εργαλεία αναφορών πωλήσεων.

Πλεονεκτήματα και Μειονεκτήματα της PHP

Η PHP είναι μια από τις πιο δημοφιλείς γλώσσες προγραμματισμού για την ανάπτυξη διαδικτυακών εφαρμογών. Παρά την ευρεία αποδοχή της, όπως κάθε τεχνολογία, διαθέτει πλεονεκτήματα αλλά και μειονεκτήματα. Στην ενότητα αυτή θα εξετάσουμε λεπτομερώς τα χαρακτηριστικά που την κάνουν ιδιαίτερα δημοφιλή, καθώς και τις περιορισμένες αδυναμίες της.[12-16]

Πλεονεκτήματα της PHP

1. Ευκολία Μάθησης και Χρήσης

Η PHP είναι ιδιαίτερα προσιτή για αρχάριους. Ο συνδυασμός της απλής σύνταξης και της πλούσιας τεκμηρίωσης την καθιστά ιδανική επιλογή για νέους προγραμματιστές.

- Η χρήση της μοιάζει πολύ με άλλες γλώσσες προγραμματισμού, όπως η C και η Perl, διευκολύνοντας τη μετάβαση.
- Οι νέοι προγραμματιστές μπορούν να ξεκινήσουν γρήγορα, γράφοντας απλά scripts και σταδιακά να περάσουν σε πιο σύνθετες εφαρμογές.

2. Ευρεία Υποστήριξη Πλατφορμών και Διακομιστών

Η PHP μπορεί να τρέξει σε σχεδόν κάθε λειτουργικό σύστημα και διακομιστή, όπως:

- **Λειτουργικά συστήματα:** Windows, Linux, macOS, Unix.
- **Διακομιστές:** Apache, Nginx, Microsoft IIS.
Η δυνατότητα αυτή επιτρέπει την ανάπτυξη εφαρμογών ανεξάρτητα από την υποδομή του server.

3. Εκτεταμένη Κοινότητα και Πόροι

Η PHP έχει μια από τις μεγαλύτερες κοινότητες προγραμματιστών στον κόσμο.

- Δεκάδες forums, ομάδες στο GitHub, και διαδικτυακά μαθήματα παρέχουν συνεχή υποστήριξη.
- Ανοιχτού κώδικα εργαλεία, όπως frameworks (Laravel, Symfony) και CMS (WordPress, Joomla), ενισχύουν την παραγωγικότητα.

4. Υψηλή Απόδοση και Βελτιστοποίηση

Η PHP είναι βελτιστοποιημένη για την ταχεία ανάπτυξη και εκτέλεση διαδικτυακών εφαρμογών.

- Χρησιμοποιεί **caching engines** για τη βελτίωση της ταχύτητας.
- Υποστηρίζει ακριβή επεξεργασία δεδομένων μέσω βιβλιοθηκών, όπως PDO για ασφαλείς βάσεις δεδομένων.

5. Ευελιξία και Παραμετροποίηση

Η PHP προσφέρει εργαλεία και επεκτάσεις για κάθε είδους εφαρμογή:

- **Επεξεργασία αρχείων:** JSON, XML, PDF, εικόνες.
- **Αλληλεπίδραση με APIs:** REST και SOAP.
- **Αυτοματισμοί:** Εργαλεία για cron jobs και αποστολή email.

6. Χαμηλό Κόστος Ανάπτυξης

Η PHP είναι ανοιχτού κώδικα, που σημαίνει ότι δεν απαιτείται καμία χρέωση για τη χρήση της.

- Υποστηρίζεται από δωρεάν διακομιστές, όπως Apache.
- Διαθέτει τεράστια βιβλιοθήκη έτοιμων κώδικων και εργαλείων που μειώνουν το κόστος ανάπτυξης.

Μειονεκτήματα της PHP

1. Ασφάλεια

Η PHP μπορεί να είναι ευάλωτη σε επιθέσεις αν δεν ακολουθηθούν οι βέλτιστες πρακτικές ασφάλειας.

- Συνηθισμένες ευπάθειες περιλαμβάνουν SQL Injection και XSS (Cross-Site Scripting).
- Χρειάζεται προσεκτική χρήση βιβλιοθηκών, όπως το PDO, για να μειωθούν οι πιθανότητες παραβίασης.

2. Απόδοση σε Μεγάλες Εφαρμογές

Η απόδοση της PHP ενδέχεται να μειωθεί σε πολύ μεγάλες και περίπλοκες εφαρμογές:

- Ενώ είναι ιδανική για μικρές εφαρμογές, συχνά απαιτείται συνεργασία με **caching tools** και υπηρεσίες cloud για μεγάλης κλίμακας έργα.

3. Συγκριτικά Περιορισμένη Υποστήριξη για Multithreading

Η PHP δεν υποστηρίζει natively το **multithreading**, που περιορίζει την απόδοσή της σε εφαρμογές με μεγάλο όγκο ταυτόχρονων εργασιών.

4. Αντίληψη Παρωχημένης Τεχνολογίας

Παρά την εξέλιξή της, η PHP συχνά θεωρείται από ορισμένους ως "παρωχημένη" γλώσσα:

- Οι επικριτές της προτιμούν πιο σύγχρονες τεχνολογίες, όπως Node.js ή Python.

- Ωστόσο, η PHP συνεχίζει να εξελίσσεται, με την έκδοση 8 να φέρνει βελτιώσεις στην ταχύτητα και τα χαρακτηριστικά της.

5. Κώδικας Δύσκολα Διαχειρίσιμος

Σε έργα μεγάλης κλίμακας, η PHP συχνά οδηγεί σε κώδικα δύσκολο στη διαχείριση, εάν δεν χρησιμοποιηθεί σωστή δομή (π.χ., χρήση frameworks).

Σύγκριση της PHP με Άλλες Γλώσσες Προγραμματισμού

Η PHP είναι μια γλώσσα προγραμματισμού ειδικά σχεδιασμένη για την ανάπτυξη διαδικτυακών εφαρμογών. Παρόλο που είναι μία από τις πιο δημοφιλείς γλώσσες στον κόσμο, υπάρχουν πολλές εναλλακτικές που προσφέρουν διαφορετικά χαρακτηριστικά και προσεγγίσεις.

1. PHP vs Python

Ομοιότητες

Ευκολία Μάθησης: Και οι δύο γλώσσες είναι εύκολες για νέους προγραμματιστές, με σαφή σύνταξη και πλούσια τεκμηρίωση.

Υποστήριξη Web Development: Και οι δύο χρησιμοποιούνται ευρέως για την ανάπτυξη διαδικτυακών εφαρμογών. Η PHP με frameworks όπως το Laravel και η Python με frameworks όπως το Django και Flask.

Ανοιχτού Κώδικα: Και οι δύο γλώσσες είναι ανοιχτού κώδικα, προσφέροντας πρόσβαση σε πληθώρα βιβλιοθηκών.

Διαφορές

Γενική Χρήση: Η PHP εστιάζει περισσότερο σε εφαρμογές ιστού, ενώ η Python είναι πιο γενικής χρήσης και χρησιμοποιείται για επιστήμη δεδομένων, τεχνητή νοημοσύνη και ανάπτυξη λογισμικού.

Απόδοση: Η Python είναι ελαφρώς πιο αργή σε web εφαρμογές, αλλά καλύτερα δομημένη για σύνθετες εφαρμογές και machine learning.

Κοινότητα: Η PHP έχει μεγαλύτερη βάση χρηστών στον χώρο του web, ενώ η Python υπερέχει σε ακαδημαϊκούς και επιστημονικούς τομείς.

2. PHP vs Node.js

Ομοιότητες

Ανοιχτού Κώδικα: Και οι δύο είναι δωρεάν και διαθέσιμες για χρήση σε οποιοδήποτε περιβάλλον.

Υποστήριξη Μεγάλων Έργων: Τόσο η PHP όσο και το Node.js υποστηρίζουν μεγάλης κλίμακας εφαρμογές με τη χρήση caching και cloud υπηρεσιών.

Διαφορές

Αρχιτεκτονική: Η PHP είναι synchronously executed, ενώ το Node.js υποστηρίζει asynchronous λειτουργίες, καθιστώντας το ιδανικό για εφαρμογές πραγματικού χρόνου.

Εκτέλεση: Η PHP απαιτεί διακομιστή (π.χ. Apache), ενώ το Node.js λειτουργεί σε runtime περιβάλλον που βασίζεται στη μηχανή V8 της Google.

Ταχύτητα: Το Node.js είναι ταχύτερο σε εφαρμογές πραγματικού χρόνου (π.χ. chats, streaming).

3. PHP vs Ruby

Ομοιότητες

Web Development: Και οι δύο είναι σχεδιασμένες για web εφαρμογές και διαθέτουν ισχυρά frameworks, όπως το Laravel για PHP και το Ruby on Rails για Ruby.

Κοινότητα και Πόροι: Διαθέτουν ισχυρή υποστήριξη από κοινότητες προγραμματιστών.

Διαφορές

Προσανατολισμός: Η PHP είναι πιο πρακτική και απλή, ενώ η Ruby δίνει έμφαση στην κομψότητα και την παραγωγικότητα.

Εκτέλεση: Η Ruby είναι πιο αργή σε απόδοση, αλλά τα Rails προσφέρουν περισσότερη αυτοματοποίηση.

4. PHP vs JavaScript

Ομοιότητες

Υποστήριξη Web Development: Και οι δύο είναι κρίσιμες για την ανάπτυξη ιστοσελίδων.

Ανοιχτός Κώδικας: Ελεύθερες για χρήση, με εκτεταμένη υποστήριξη.

Διαφορές

Server vs Client: Η PHP είναι server-side, ενώ η JavaScript είναι κυρίως client-side (αν και με το Node.js έχει μεταφερθεί και στον server).

Χρήσεις: Η PHP επικεντρώνεται στη διαχείριση δεδομένων και στον χειρισμό server, ενώ η JavaScript προσφέρει διαδραστικότητα στις ιστοσελίδες.

Απόδοση: Η JavaScript (με Node.js) θεωρείται ταχύτερη για real-time εφαρμογές.

Η PHP είναι ιδιαίτερα αποδοτική για εφαρμογές διαδικτύου και διαχείρισης δεδομένων, καθιστώντας την εξαιρετική επιλογή για CMS, blogs και forums.. Άλλες γλώσσες, όπως η Python και το Node.js, προσφέρουν πιο ευρείες δυνατότητες, ιδίως σε τομείς πέρα από το web development. [17-18]

Μερίδιο της PHP στην Web Αγορά και Χρήσεις της

Η PHP παραμένει μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού για την ανάπτυξη web εφαρμογών, παρόλο που έχει εμφανιστεί έντονος ανταγωνισμός από νεότερες τεχνολογίες, όπως το Python και το Node.js. Στο κεφάλαιο αυτό, θα αναλύσουμε το μερίδιο της PHP στην παγκόσμια αγορά και θα εξετάσουμε πού χρησιμοποιείται κυρίως.

Στατιστικά Χρήσης PHP στην Αγορά

Σύμφωνα με δεδομένα του [W3Techs](#), [20] περίπου το **77.4%** όλων των ιστοσελίδων παγκοσμίως που χρησιμοποιούν γλώσσα server-side είναι γραμμένες σε PHP (Ιανουάριος 2025). Παρόλο που το ποσοστό αυτό έχει μειωθεί ελαφρώς με τα χρόνια λόγω του ανταγωνισμού, η PHP εξακολουθεί να είναι η κυρίαρχη γλώσσα στην αγορά. [22-24]

Πίνακας 1: Μερίδιο Server-Side Γλωσσών

Γλώσσα	Ποσοστό (%)
PHP	77.4
ASP.NET	7.5
Node.js	2.3
Ruby	1.1
Python	1.4
Άλλες	10.3

Η PHP βρίσκεται πίσω από μεγάλα συστήματα διαχείρισης περιεχομένου (CMS), όπως το WordPress, το οποίο από μόνο του κατέχει περίπου **43% της παγκόσμιας αγοράς CMS**. Επίσης, χρησιμοποιείται ευρέως σε πλατφόρμες όπως το Joomla, το Drupal, και το Magento.

Τομείς Χρήσης της PHP

1. Συστήματα Διαχείρισης Περιεχομένου (CMS)

- Όπως αναφέρθηκε, πλατφόρμες όπως το **WordPress**, το **Drupal**, και το **Joomla** βασίζονται σε μεγάλο βαθμό στην PHP για τη λειτουργία τους. Αυτά τα CMS επιτρέπουν τη δημιουργία δυναμικών ιστοσελίδων με ελάχιστη ή καθόλου γνώση προγραμματισμού.

2. E-Commerce

- Το **Magento**, το **WooCommerce** και το **OpenCart** είναι πλατφόρμες ηλεκτρονικού εμπορίου γραμμένες σε PHP. Χρησιμοποιούνται από χιλιάδες επιχειρήσεις για τη δημιουργία online καταστημάτων.

3. Web Applications

- Η PHP χρησιμοποιείται ευρέως σε web εφαρμογές, όπως **CRM συστήματα** (Customer Relationship Management) και **ERP** (Enterprise Resource Planning). Frameworks όπως το **Laravel** και το **Symfony** έχουν συμβάλει στην ανάπτυξη εφαρμογών υψηλής ποιότητας.

4. Blogs και Forums

- Πλατφόρμες όπως το **WordPress** (για blogs) και το **phpBB** (για forums) βασίζονται στην PHP.

5. Κοινωνικά Δίκτυα

- Γνωστές πλατφόρμες, όπως το **Facebook** στην αρχική του μορφή, χρησιμοποιούσαν PHP για τη δημιουργία των βασικών λειτουργιών.

Πλεονεκτήματα που Οδηγούν στη Δημοφιλία

1. **Ευκολία Χρήσης:** Η PHP είναι εύκολη για νέους προγραμματιστές, με μια απλή σύνταξη που μοιάζει με αυτή των γλωσσών όπως η C.
2. **Ευρεία Υποστήριξη:** Διαθέτει πληθώρα βιβλιοθηκών και εργαλείων για τη δημιουργία εφαρμογών.

3. **Συμβατότητα:** Λειτουργεί άψογα με σχεδόν όλους τους web servers (Apache, Nginx, IIS).
4. **Χαμηλό Κόστος:** Είναι ανοιχτού κώδικα και δωρεάν, ενώ παράλληλα υποστηρίζεται από πολλά φθηνά hosting πακέτα.

Παρότι η PHP κατέχει μεγάλο μερίδιο της αγοράς, αντιμετωπίζει προκλήσεις:

- **Ανταγωνισμός:** Γλώσσες όπως η Python και το Node.js κερδίζουν έδαφος, ειδικά σε εφαρμογές real-time και επιστημονικές εφαρμογές.
- **Κριτική για την Απόδοση:** Οι παλαιότερες εκδόσεις της PHP είχαν χαμηλότερη απόδοση σε σχέση με άλλες γλώσσες. Ωστόσο, η PHP 7 και PHP 8 έχουν βελτιώσει σημαντικά αυτήν την εικόνα.

Χρήση της PHP με το Laravel

Η συνδυαστική χρήση της PHP με το Laravel αποτελεί μία από τις πιο δημοφιλείς πρακτικές στον τομέα του web development. Το Laravel είναι ένα από τα πιο σύγχρονα και ισχυρά frameworks της PHP, προσφέροντας εργαλεία και χαρακτηριστικά που διευκολύνουν την ανάπτυξη εφαρμογών υψηλής ποιότητας. Σε αυτό το κεφάλαιο θα αναλύσουμε τη σύνδεση της PHP με το Laravel, τις δυνατότητες που προσφέρει, καθώς και παραδείγματα εφαρμογών. [6,25]

Το Laravel είναι ένα MVC (Model-View-Controller) framework που έχει δημιουργηθεί πάνω στην PHP. Σχεδιάστηκε για να απλοποιήσει τη διαδικασία ανάπτυξης web εφαρμογών και να προσφέρει μια καθαρή, δομημένη προσέγγιση στη συγγραφή κώδικα.

Κύριες Αρχές που Κληρονομεί από την PHP:

1. **Ευκολία Μάθησης:** Όπως και η PHP, το Laravel έχει εύκολη καμπύλη εκμάθησης, καθιστώντας το ιδανικό για νέους προγραμματιστές.
2. **Επεκτασιμότητα:** Το Laravel επεκτείνει τη λειτουργικότητα της PHP με εργαλεία όπως το Artisan CLI και το Eloquent ORM.
3. **Διαλειτουργικότητα:** Υποστηρίζει πλήρως PHP βιβλιοθήκες και πακέτα, επιτρέποντας την εύκολη ενσωμάτωσή τους.

Δυνατότητες που Παρέχει το Laravel Μέσω της PHP

1. **Eloquent ORM:** Το Laravel χρησιμοποιεί το Eloquent ORM για την αλληλεπίδραση με βάσεις δεδομένων. Αυτό επιτρέπει στους προγραμματιστές να γράφουν PHP κώδικα αντί για SQL ερωτήματα.

```
// Ανάκτηση δεδομένων από τον πίνακα "users"
```

```
$users = User::where('active', 1)->get();
```

2. **Blade Templating Engine:** Το Laravel εισάγει το Blade, ένα απλό και ισχυρό εργαλείο για τη διαχείριση HTML και PHP μέσα στα views.

Παράδειγμα:

```
@foreach ($users as $user)
```

```
<p>{{ $user->name }}</p>
```

```
@endforeach
```

3. **Artisan CLI:** Το Laravel προσφέρει το Artisan, ένα command-line εργαλείο που διευκολύνει την εκτέλεση καθημερινών εργασιών, όπως η δημιουργία μοντέλων ή ο έλεγχος της βάσης δεδομένων.

Παράδειγμα:

```
php artisan make:model Ticket
```

4. **Αυθεντικοποίηση και Ασφάλεια:** Το Laravel παρέχει έτοιμες λύσεις αυθεντικοποίησης, όπως το sanctum και το passport, που είναι βασισμένα στην PHP.

Παράδειγμα:

```
if (Auth::check()) {
```

```
    return view('dashboard');
```

```
}
```

5. **Routing:** Με το Laravel, ο ορισμός των routes γίνεται εξαιρετικά εύκολος και καθαρός.

```
Route::get('/tickets', [TicketController::class, 'index']);
```

Εφαρμογές του Laravel και Παραδείγματα

- **Διαχείριση Αιτημάτων Υποστήριξης:** Η χρήση του Laravel, όπως στο παρόν έργο, διευκολύνει τη δημιουργία ενός συστήματος υποστήριξης με δυναμική διαχείριση αιτημάτων.
- **Πλατφόρμες E-commerce:** Πλατφόρμες όπως το **Bagisto** βασίζονται στο Laravel για την ανάπτυξη ηλεκτρονικών καταστημάτων.

- **Συστήματα CMS:** Εφαρμογές όπως το **October CMS** χρησιμοποιούν το Laravel για τη διαχείριση περιεχομένου.
- **Προσωπικές Ιστοσελίδες και Portfolios:** Εύκολα και γρήγορα δημιουργούνται ιστότοποι με τη χρήση του Blade και των helpers του Laravel.

Πλεονεκτήματα Χρήσης Laravel με PHP

1. **Ταχύτητα Ανάπτυξης:** Ο συνδυασμός Laravel και PHP επιταχύνει τη διαδικασία ανάπτυξης μέσω αυτοματοποιημένων εργαλείων και βιβλιοθηκών.
2. **Αναγνωσιμότητα Κώδικα:** Η καθαρή δομή και τα patterns του Laravel βοηθούν στη συντήρηση και την επεκτασιμότητα.
3. **Ασφάλεια:** Παρέχει προστασία ενάντια σε κοινές απειλές, όπως SQL injections και XSS attacks.
4. **Ενεργή Κοινότητα:** Το Laravel διαθέτει μια από τις μεγαλύτερες κοινότητες PHP developers, προσφέροντας υποστήριξη και οδηγούς.

Μειονεκτήματα και Περιορισμοί

1. **Καμπύλη Εκμάθησης:** Παρόλο που είναι φιλικό, οι προχωρημένες δυνατότητες του Laravel μπορεί να φανούν περίπλοκες για αρχάριους.
2. **Απαιτήσεις Πόρων:** Οι εφαρμογές Laravel απαιτούν περισσότερους πόρους από απλές εφαρμογές PHP.

3.3 Bootstrap

Το Bootstrap είναι ένα από τα πιο διαδεδομένα frameworks για την ανάπτυξη responsive και mobile-first ιστοσελίδων. Δημιουργήθηκε από την ομάδα του Twitter το 2011, και από τότε έχει καθιερωθεί ως βασικό εργαλείο για τους front-end developers. Προσφέρει μια μεγάλη συλλογή από εργαλεία, έτοιμες βιβλιοθήκες CSS και JavaScript, που επιταχύνουν τη διαδικασία σχεδιασμού και ανάπτυξης. [20,26-28]

Τι Είναι το Bootstrap;

Το Bootstrap είναι ένα open-source framework που παρέχει έτοιμα components και εργαλεία για τη δημιουργία σύγχρονων και φιλικών προς το χρήστη ιστοσελίδων. Βασίζεται σε HTML, CSS, και JavaScript, και υποστηρίζει όλες τις σύγχρονες τεχνολογίες web.

Βασικές Αρχές:

1. **Responsive Design:** Όλες οι ιστοσελίδες που φτιάχνονται με Bootstrap προσαρμόζονται αυτόματα σε οποιοδήποτε μέγεθος οθόνης.
2. **Mobile-First Approach:** Το Bootstrap έχει σχεδιαστεί με προτεραιότητα στις κινητές συσκευές.
3. **Προσβασιμότητα:** Υποστηρίζει πρότυπα όπως το ARIA για χρήστες με αναπηρίες.

Κύρια Στοιχεία του Bootstrap

Το Bootstrap περιλαμβάνει τρία κύρια στοιχεία που το καθιστούν πανίσχυρο:

1. **Grid System:** Το grid system του Bootstrap βασίζεται σε ένα ευέλικτο και ισχυρό δίκτυο 12 στηλών. Με αυτό, οι προγραμματιστές μπορούν να οργανώσουν το περιεχόμενο με ακρίβεια.

```
<div class="container">
  <div class="row">
    <div class="col-md-6">Περιεχόμενο A</div>
    <div class="col-md-6">Περιεχόμενο B</div>
  </div>
</div>
```

2. **Components:** Το Bootstrap διαθέτει πληθώρα έτοιμων components όπως κουμπιά, φόρμες, navbar, cards και modal dialogs.

```
<button class="btn btn-primary">Πάτησέ με</button>
```

3. **Utilities:** Παρέχει έτοιμες κλάσεις CSS για τη διαχείριση του spacing, του alignment και άλλων χαρακτηριστικών.

```
<div class="text-center mt-3">Κεντραρισμένο κείμενο με απόσταση</div>
```

Εφαρμογές του Bootstrap

1. **Responsive Ιστοσελίδες:** Το Bootstrap είναι ιδανικό για την ανάπτυξη ιστοσελίδων που λειτουργούν εξίσου καλά σε κινητά, tablets και desktop.
2. **Εφαρμογές Διαχείρισης:** Χρησιμοποιείται ευρέως για admin dashboards και συστήματα διαχείρισης περιεχομένου (CMS).
3. **Ηλεκτρονικά Καταστήματα:** Με το Bootstrap, η δημιουργία responsive πλατφορμών e-commerce γίνεται πιο εύκολη.
4. **Προσωπικά Portfolios:** Πολλοί σχεδιαστές και developers χρησιμοποιούν το Bootstrap για τη δημιουργία online portfolios.

Πλεονεκτήματα του Bootstrap

1. **Ταχύτητα Ανάπτυξης:** Προσφέρει έτοιμες λύσεις που μειώνουν τον χρόνο σχεδιασμού και ανάπτυξης.
2. **Συμβατότητα με Browsers:** Το Bootstrap υποστηρίζει όλα τα σύγχρονα προγράμματα περιήγησης.
3. **Εκτεταμένη Τεκμηρίωση:** Η τεκμηρίωση του Bootstrap είναι μια από τις καλύτερες στην αγορά, παρέχοντας παραδείγματα και οδηγούς.
4. **Ενεργή Κοινότητα:** Με χιλιάδες ενεργούς χρήστες, το Bootstrap διαθέτει τεράστια κοινότητα για υποστήριξη.
5. **Επεκτασιμότητα:** Επιτρέπει την προσαρμογή του κώδικα σύμφωνα με τις ανάγκες κάθε έργου.

Μειονεκτήματα του Bootstrap

1. **Ομοιογένεια Σχεδίασης:** Επειδή πολλοί χρησιμοποιούν το Bootstrap, οι ιστοσελίδες συχνά έχουν παρόμοια εμφάνιση.
2. **Περιορισμός Δημιουργικότητας:** Οι έτοιμες λύσεις του framework μπορεί να περιορίσουν τη φαντασία του designer.
3. **Μεγάλο Μέγεθος Αρχείων:** Το Bootstrap περιλαμβάνει πολλά χαρακτηριστικά, τα οποία μπορεί να μην είναι όλα απαραίτητα για κάθε έργο.

Μερίδιο Αγοράς και Δημοτικότητα

Το Bootstrap παραμένει το πιο διαδεδομένο front-end framework:

- Χρησιμοποιείται από το **19.2%** των ιστοσελίδων παγκοσμίως (πηγή: **W3Techs**).
- Είναι το προτιμώμενο framework για το **80% των developers**, σύμφωνα με έρευνα του **Stack Overflow**.

Χρήση του Bootstrap σε Laravel Εφαρμογές

Το Bootstrap συνεργάζεται άψογα με το Laravel, καθώς προσφέρει τα εξής πλεονεκτήματα:

- Δημιουργία γρήγορων και responsive user interfaces.
- Εύκολη ενσωμάτωση με το Blade templating engine του Laravel.
- Δυνατότητα χρήσης του Laravel Mix για την παραμετροποίηση των CSS και JavaScript αρχείων.

Το Bootstrap έχει καθιερωθεί ως ένα από τα πιο ισχυρά και αξιόπιστα εργαλεία για την ανάπτυξη responsive ιστοσελίδων. Σε συνδυασμό με το Laravel και άλλες σύγχρονες τεχνολογίες, δίνει στους developers τη δυνατότητα να δημιουργούν επαγγελματικές εφαρμογές με ταχύτητα και αποτελεσματικότητα.

3.4 MySQL

Η MySQL είναι ένα από τα πιο δημοφιλή συστήματα διαχείρισης βάσεων δεδομένων (DBMS) στον κόσμο. Ανοιχτού κώδικα, ευέλικτη και ισχυρή, έχει καταστεί απαραίτητη για την ανάπτυξη εφαρμογών web και επιχειρηματικών λύσεων. Από προσωπικά έργα έως παγκόσμιες επιχειρήσεις, η MySQL αποτελεί την καρδιά εκατομμυρίων εφαρμογών και ιστοσελίδων. [29-32]

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS), βασισμένο στη γλώσσα SQL (Structured Query Language). Αναπτύχθηκε από τη σουηδική εταιρεία MySQL AB το 1995 και πλέον ανήκει στην Oracle Corporation. Χρησιμοποιείται κυρίως για την αποθήκευση και διαχείριση δεδομένων που υποστηρίζουν εφαρμογές web.

Χαρακτηριστικά:

- **Ανοιχτού Κώδικα:** Ελεύθερη χρήση και προσαρμογή με επιλογές commercial licensing.
- **Υποστήριξη Σχέσεων:** Βάσεις δεδομένων που αποτελούνται από πίνακες συνδεδεμένους με σχέσεις.
- **Αποδοτικότητα:** Υψηλές επιδόσεις σε μεγάλα δεδομένα.

Πλεονεκτήματα της MySQL

1. **Ευκολία στη Χρήση:** Παρέχει φιλικά εργαλεία για προγραμματιστές, όπως το MySQL Workbench, που επιτρέπουν τη δημιουργία και διαχείριση βάσεων δεδομένων.
2. **Ευελιξία:** Υποστηρίζει διάφορες γλώσσες προγραμματισμού όπως PHP, Python, Java και C++.
3. **Ταχύτητα και Απόδοση:** Η MySQL είναι σχεδιασμένη για υψηλές επιδόσεις, ακόμα και σε μεγάλες εφαρμογές.
4. **Ασφάλεια:** Προσφέρει προηγμένα χαρακτηριστικά ασφαλείας, όπως έλεγχο πρόσβασης και πιστοποίηση χρηστών.
5. **Υποστήριξη και Κοινότητα:** Με τεράστια βάση χρηστών και ενεργή κοινότητα, η MySQL διαθέτει άφθονους πόρους και οδηγίες.

Μειονεκτήματα της MySQL

1. **Περιορισμοί σε Σύνθετες Εφαρμογές:** Σε ορισμένες περιπτώσεις, δεν είναι τόσο ισχυρή όσο άλλες βάσεις δεδομένων όπως η PostgreSQL.
2. **Ελλιπής Υποστήριξη για Μεγάλα Συστήματα:** Για πολύ μεγάλα δεδομένα και απαιτητικές εφαρμογές, ίσως απαιτούνται άλλες λύσεις.
3. **Περιορισμοί στην Αναλυτική Διαχείριση:** Σε σύνθετες αναλύσεις δεδομένων, μπορεί να χρειαστεί συνδυασμός με άλλα εργαλεία.

Εφαρμογές της MySQL

1. **Συστήματα Διαχείρισης Περιεχομένου (CMS):** Πλατφόρμες όπως το WordPress, το Joomla και το Drupal βασίζονται στη MySQL για την αποθήκευση περιεχομένου.
2. **E-commerce Εφαρμογές:** Ιστοσελίδες όπως το Magento και το Shopify χρησιμοποιούν MySQL για την αποθήκευση πληροφοριών πελατών και προϊόντων.
3. **Social Media:** Το Facebook χρησιμοποίησε τη MySQL ως τη βασική του βάση δεδομένων στα πρώτα του στάδια.
4. **Εφαρμογές SaaS (Software as a Service):** Πολλές πλατφόρμες SaaS, όπως το Zendesk και το HubSpot, βασίζονται στη MySQL για την αποθήκευση δεδομένων χρηστών.

5. Πλεονεκτήματα στη Χρήση με Laravel

Η MySQL συνδυάζεται εξαιρετικά με το Laravel, προσφέροντας:

- **Εύκολη Διαχείριση:** Το Eloquent ORM του Laravel απλοποιεί τη διαχείριση δεδομένων MySQL.
- **Υψηλή Απόδοση:** Το Laravel βελτιστοποιεί τη χρήση των ερωτημάτων MySQL.
- **Migration Tools:** Παρέχει εργαλεία για την παρακολούθηση αλλαγών στις βάσεις δεδομένων.

6. Μερίδιο Αγοράς

Η MySQL παραμένει η πιο διαδεδομένη βάση δεδομένων στον κόσμο:

- **Δημοτικότητα:** Σύμφωνα με το **DB-Engines**, είναι η #1 RDBMS πλατφόρμα με περισσότερους από 5 εκατομμύρια χρήστες.
- **Χρήση:** Το 58% των εφαρμογών web χρησιμοποιούν MySQL.

Η MySQL συνεχίζει να είναι το βασικό εργαλείο για την αποθήκευση δεδομένων στις περισσότερες εφαρμογές web. Η σταθερότητα, η ευελιξία και η υποστήριξή της την καθιστούν αδιαμφισβήτητη επιλογή για προγραμματιστές και επιχειρήσεις. Σε συνδυασμό με το Laravel, η MySQL δημιουργεί ένα ισχυρό θεμέλιο για την ανάπτυξη σύγχρονων εφαρμογών.

Κεφάλαιο 4ο: Το σύστημα Ticket

4.1 Το σύστημα και οι διαδικασίες

Το σύστημα Ticket είναι μια διαδικτυακή εφαρμογή που έχει σχεδιαστεί για τη διαχείριση αιτημάτων υποστήριξης. Σκοπός του είναι να παρέχει στους χρήστες έναν αποτελεσματικό, γρήγορο και οργανωμένο τρόπο καταγραφής προβλημάτων ή αιτημάτων, επιτρέποντας στους διαχειριστές να τα διαχειρίζονται και να τα επιλύουν με μεθοδικό τρόπο.

Η εφαρμογή αποτελείται από δύο βασικούς ρόλους: τους χρήστες (users), οι οποίοι δημιουργούν αιτήματα, και τους διαχειριστές (admins), που αναλαμβάνουν τη διαχείριση και επίλυση των αιτημάτων. Κάθε αίτημα (ticket) συνοδεύεται από μια σειρά δεδομένων, όπως η κατηγορία, η περιγραφή, η κατάσταση του και τυχόν απαντήσεις ή σχόλια από τους διαχειριστές.

Η λειτουργία του συστήματος Ticket χωρίζεται σε βήματα, τα οποία περιγράφονται παρακάτω:

4.1.1 Υποβολή Αιτήματος από τον Χρήστη

Ο χρήστης εισέρχεται στο σύστημα μέσω της σελίδας σύνδεσης, χρησιμοποιώντας τα διαπιστευτήριά του.

Ο χρήστης μεταβαίνει στη σελίδα δημιουργίας αιτήματος.

Επιλέγει την κατηγορία του αιτήματος (π.χ. «Πρόβλημα με τη σύνδεση», «Πρόβλημα αποθήκευσης»).

Συμπληρώνει τον τίτλο και την περιγραφή του προβλήματος.

Το αίτημα καταχωρείται στη βάση δεδομένων και αποδίδεται σε κατάσταση «Ανοιχτό».

Οι διαχειριστές ειδοποιούνται για το νέο αίτημα.

4.1.2 Διαχείριση Αιτημάτων από τους Διαχειριστές

Οι διαχειριστές μπορούν να δουν όλα τα αιτήματα στη σελίδα διαχείρισης.

Τα αιτήματα εμφανίζονται με πληροφορίες όπως το όνομα του χρήστη, η κατηγορία, ο τίτλος, η περιγραφή και η τρέχουσα κατάστασή τους.

Οι διαχειριστές μπορούν να αλλάξουν την κατάσταση ενός αιτήματος (π.χ. «Σε Εξέλιξη», «Κλειστό», «Ακύρωση»).

Έχουν τη δυνατότητα να προσθέσουν σχόλια ή οδηγίες μέσω της λειτουργίας απαντήσεων.

Εάν το αίτημα είναι άκυρο ή άσχετο, μπορούν να το ακυρώσουν.

4.1.3 Καταγραφή Ενεργειών

Κάθε ενέργεια του διαχειριστή (αλλαγή κατάστασης, απάντηση, ακύρωση) καταγράφεται στο σύστημα logs.

Οι χρήστες μπορούν να δουν την ιστορία των ενεργειών μέσω της σελίδας προβολής του αιτήματος.

Ιστορικό Ενεργειών: Οι χρήστες έχουν πρόσβαση στο ιστορικό ενεργειών του αιτήματός τους, βλέποντας πότε έγινε κάθε ενέργεια και ποιος τη διενήργησε.

Ειδοποιήσεις: Οι χρήστες λαμβάνουν ειδοποιήσεις για αλλαγές στην κατάσταση του αιτήματος ή για νέες απαντήσεις από τους διαχειριστές.

4.1.4 Τεχνολογίες και Σχεδιασμός του Συστήματος

Το σύστημα Ticket αναπτύχθηκε χρησιμοποιώντας σύγχρονες τεχνολογίες και πρότυπα ανάπτυξης web. Οι βασικοί πυλώνες του συστήματος είναι οι εξής:

Backend: Το Laravel framework για τη διαχείριση της λογικής της εφαρμογής και τη σύνδεση με τη βάση δεδομένων MySQL.

Frontend: Το Bootstrap χρησιμοποιήθηκε για το σχεδιασμό του φιλικού προς το χρήστη περιβάλλοντος.

Βάση Δεδομένων: Η MySQL χρησιμοποιήθηκε για την αποθήκευση όλων των δεδομένων, όπως τα αιτήματα, οι απαντήσεις, οι χρήστες και τα logs.

Ασφάλεια: Ενσωματώθηκαν μηχανισμοί ασφαλείας, όπως η επικύρωση στοιχείων σύνδεσης και η διαχείριση δικαιωμάτων πρόσβασης.

4.1.5 Ειδικές Λειτουργίες

Κατηγοριοποίηση Αιτημάτων:

Τα αιτήματα κατηγοριοποιούνται για την ευκολότερη διαχείριση από τους διαχειριστές.

Οι κατηγορίες είναι δυναμικές και μπορούν να προστεθούν νέες σύμφωνα με τις ανάγκες του οργανισμού.

Προβολή Απαντήσεων:

Ο χρήστης μπορεί να βλέπει όλες τις απαντήσεις που έχουν δοθεί από τους διαχειριστές.

Οι απαντήσεις συνοδεύονται από την ημερομηνία και το όνομα του διαχειριστή που τις παρέιχε.

Διαχείριση Logs:

Όλες οι ενέργειες καταγράφονται με λεπτομέρειες.

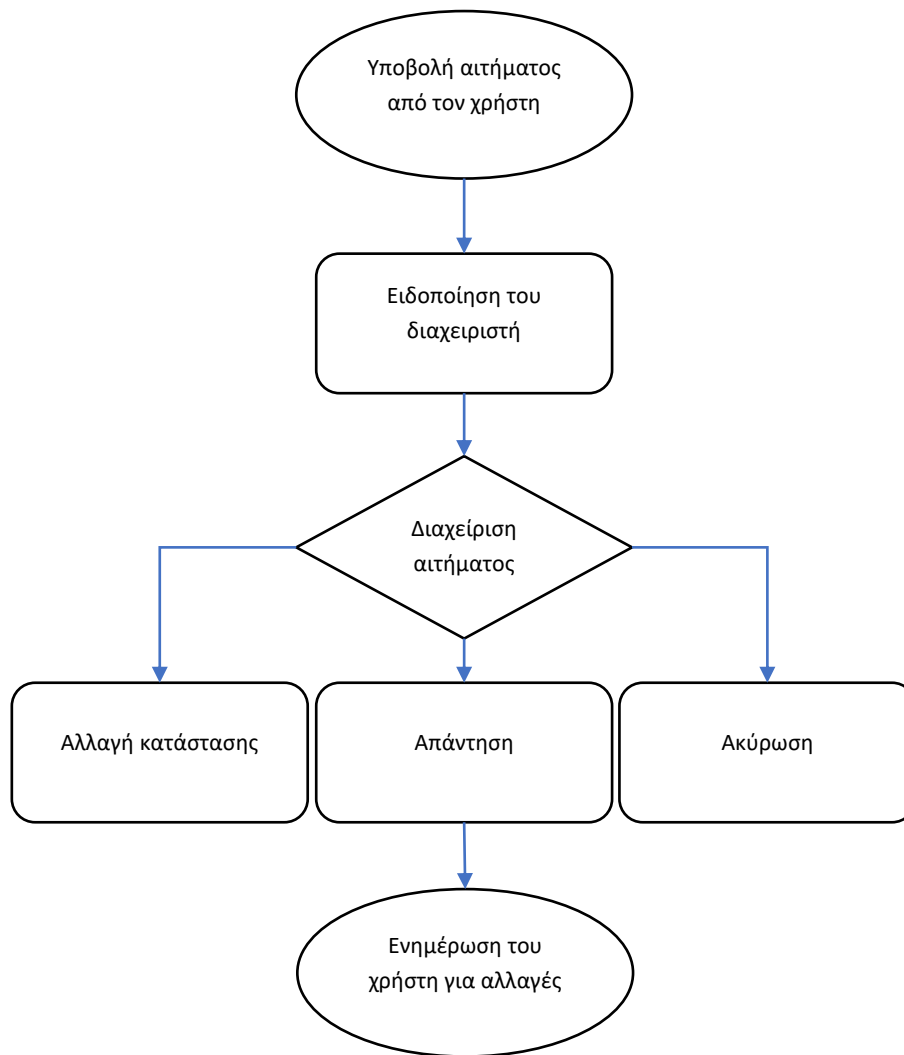
Οι διαχειριστές έχουν πρόσβαση σε αναλυτικές αναφορές για τη διαφάνεια και τη βελτίωση της διαδικασίας.

Το σύστημα Ticket αποτελεί ένα σύγχρονο εργαλείο που αναβαθμίζει την εμπειρία των χρηστών, ενώ παράλληλα διευκολύνει τη δουλειά των διαχειριστών. Με απλό και αποτελεσματικό τρόπο, ενσωματώνει όλες τις απαραίτητες λειτουργίες για τη διαχείριση αιτημάτων, προσφέροντας μια ολοκληρωμένη λύση για εκπαιδευτικά και επαγγελματικά περιβάλλοντα.

Η Εικόνα 4.1 απεικονίζει τη διαδικασία υποβολής και διαχείρισης αιτήματος στο σύστημα υποστήριξης. Η διαδικασία ξεκινά με την υποβολή ενός αιτήματος από τον χρήστη, όπου περιγράφεται το πρόβλημα ή η ανάγκη του. Στη συνέχεια, ο διαχειριστής ενημερώνεται μέσω ειδοποίησης για το νέο αίτημα.

Αφού ο διαχειριστής λάβει γνώση του αιτήματος, προχωρά στη διαχείρισή του μέσω τριών δυνατών επιλογών: αλλαγή της κατάστασης του αιτήματος, προσθήκη απάντησης ή ακύρωση του αιτήματος εάν κρίνεται απαραίτητο. Ο τελικός στόχος είναι να ενημερωθεί ο χρήστης σχετικά με τις αλλαγές που έγιναν στο αίτημά του, εξασφαλίζοντας διαφάνεια και αποτελεσματικότητα στη διαδικασία επίλυσης.

Το διάγραμμα ροής προσφέρει μια καθαρή και συνοπτική απεικόνιση των βημάτων, διευκολύνοντας την κατανόηση του πώς συνεργάζονται χρήστες και διαχειριστές για την ταχεία διευθέτηση προβλημάτων.

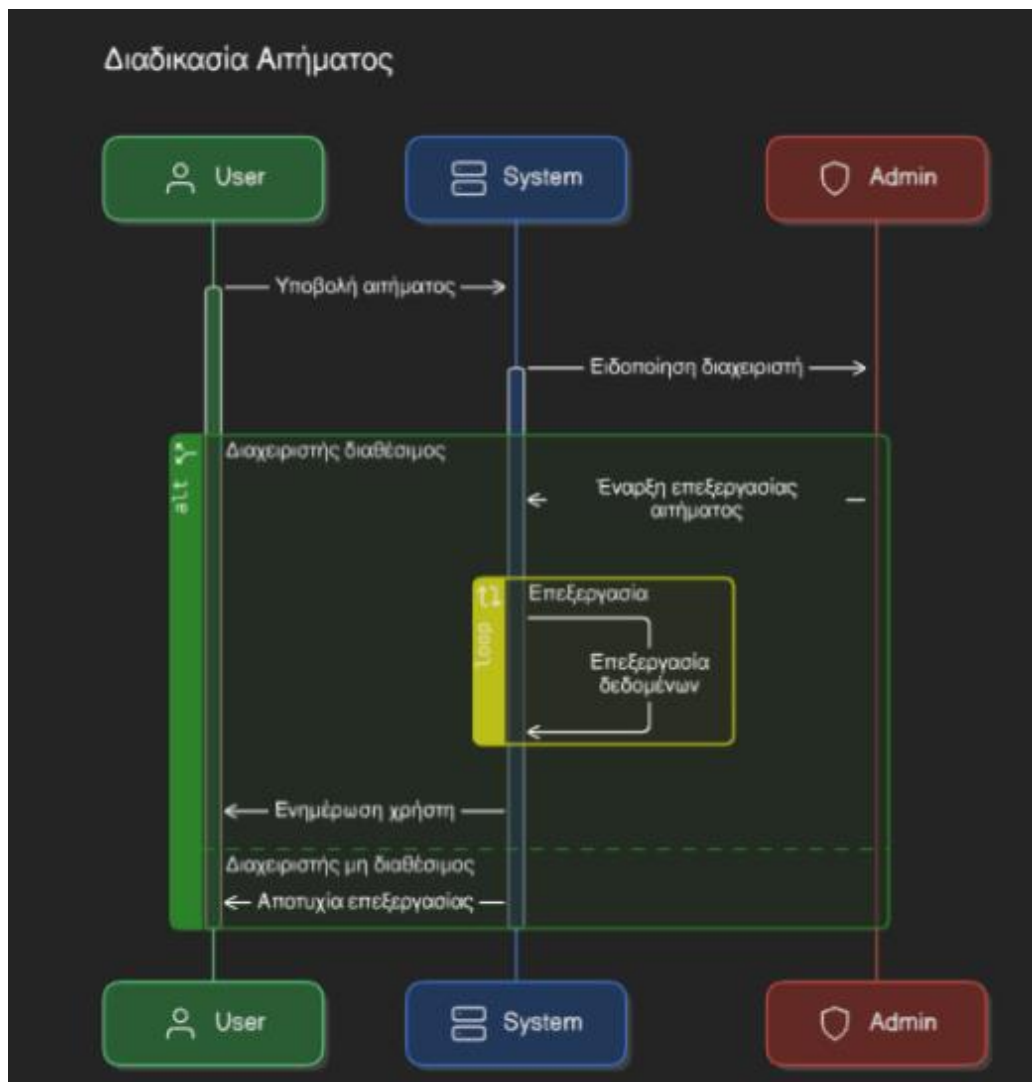


Εικόνα 4.1: Διαδικασία Αιτήματος

Η Εικόνα 4.2 παρουσιάζει τη διαδικασία αιτήματος σε προβολή Use Case Diagram, αποτυπώνοντας τις αλληλεπιδράσεις μεταξύ χρηστών, συστήματος και διαχειριστών. Ο χρήστης ξεκινά τη διαδικασία με την υποβολή ενός αιτήματος προς το σύστημα. Το σύστημα στη συνέχεια ενημερώνει τον διαχειριστή για την ύπαρξη του αιτήματος, ενεργοποιώντας τη διαδικασία επεξεργασίας.

Ο διαχειριστής, με βάση τις δυνατότητές του, μπορεί να ξεκινήσει την επεξεργασία του αιτήματος, να πραγματοποιήσει αλλαγές ή να δώσει απαντήσεις. Η επεξεργασία δεδομένων γίνεται κυκλικά, ώστε να διασφαλίζεται η ολοκληρωμένη κάλυψη των απαιτήσεων του αιτήματος.

Η διαδικασία ολοκληρώνεται είτε με την ενημέρωση του χρήστη για τις αλλαγές που πραγματοποιήθηκαν είτε με την αποτυχία της επεξεργασίας, εάν ο διαχειριστής δεν είναι διαθέσιμος. Το Use Case Diagram παρέχει μια σαφή και διαδοχική απεικόνιση της διαδικασίας, αναδεικνύοντας τους βασικούς ρόλους και τις αλληλεπιδράσεις τους.

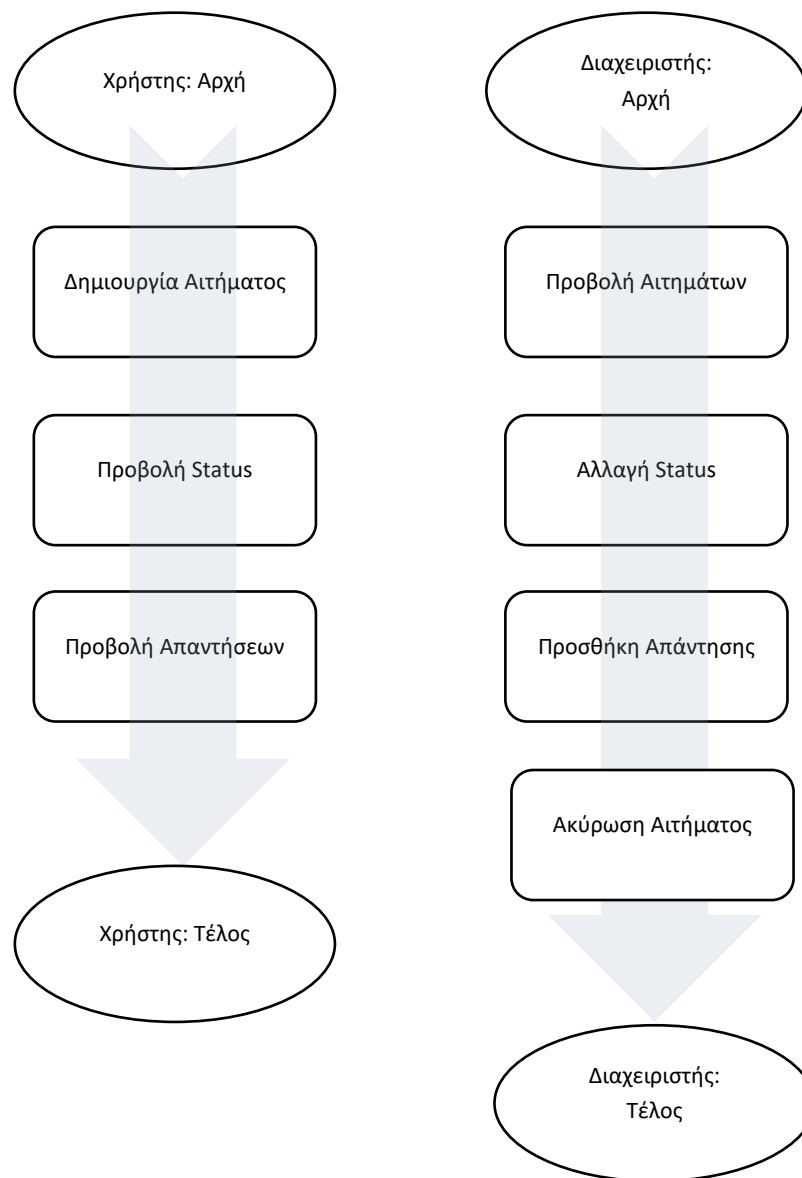


Εικόνα 4.2: Διαδικασία Αιτήματος σε προβολή Use Case Diagram

Η Εικόνα 4.3 παρουσιάζει τη ροή αλληλεπίδρασης μεταξύ χρηστών και διαχειριστών στο σύστημα υποστήριξης αιτημάτων. Στην αριστερή στήλη περιγράφονται οι ενέργειες του χρήστη, ξεκινώντας από τη δημιουργία αιτήματος, την προβολή της κατάστασης του αιτήματος και τη δυνατότητα ανάγνωσης των απαντήσεων που έχουν δοθεί από τον διαχειριστή. Η διαδικασία καταλήγει στο τέλος της ενέργειας του χρήστη.

Στην αντίστοιχη δεξιά στήλη αναλύονται οι ενέργειες του διαχειριστή, που περιλαμβάνουν την προβολή των αιτημάτων, την αλλαγή της κατάστασης, την προσθήκη απαντήσεων και την ακύρωση ενός αιτήματος, αν απαιτείται. Η διαδικασία αυτή ολοκληρώνεται με την επιτυχή διαχείριση των αιτημάτων από τον διαχειριστή.

Το διάγραμμα υπογραμμίζει τις σαφείς και διακριτές λειτουργίες του κάθε ρόλου και τονίζει τη ροή εργασιών που εξασφαλίζει τη συνεργασία μεταξύ των δύο πλευρών.

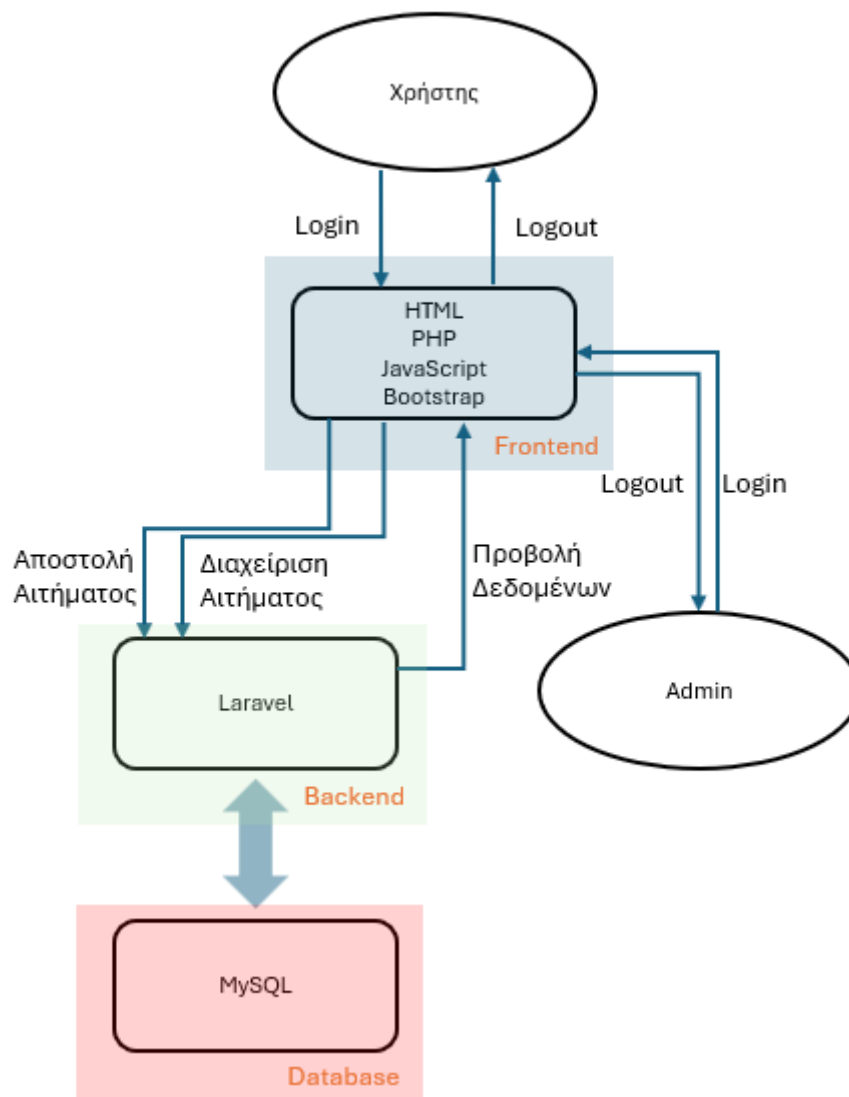


Εικόνα 4.3: Λειτουργίες του χρήστη και του διαχειριστή

Η Εικόνα 4.4 παρουσιάζει το διάγραμμα ροής αλληλεπίδρασης του συστήματος Ticket, αναδεικνύοντας τη σύνδεση μεταξύ των διαφόρων επιπέδων του συστήματος και των ρόλων που εμπλέκονται. Η αρχιτεκτονική αποτελείται από τρία κύρια επίπεδα: το **Frontend**, το **Backend** και τη **βάση δεδομένων**.

1. **Frontend:** Στο επίπεδο του Frontend, ο χρήστης και ο διαχειριστής αλληλεπιδρούν με το σύστημα μέσω τεχνολογιών όπως **HTML**, **PHP**, **JavaScript** και **Bootstrap**. Το περιβάλλον αυτό επιτρέπει την εισαγωγή αιτημάτων, την προβολή των καταστάσεων και την υποβολή αλλαγών.

2. **Backend:** Το επίπεδο Backend υλοποιείται με το framework **Laravel**, το οποίο λειτουργεί ως γέφυρα μεταξύ του Frontend και της βάσης δεδομένων. Το Laravel διαχειρίζεται τις αιτήσεις που υποβάλλουν οι χρήστες, ενώ εξασφαλίζει την επεξεργασία και τον χειρισμό των δεδομένων.
3. **Database:** Η βάση δεδομένων **MySQL** είναι το θεμέλιο αποθήκευσης πληροφοριών. Όλες οι ενέργειες, όπως η δημιουργία αιτημάτων, οι αλλαγές καταστάσεων και οι απαντήσεις, καταγράφονται και οργανώνονται σε σχεσιακούς πίνακες.

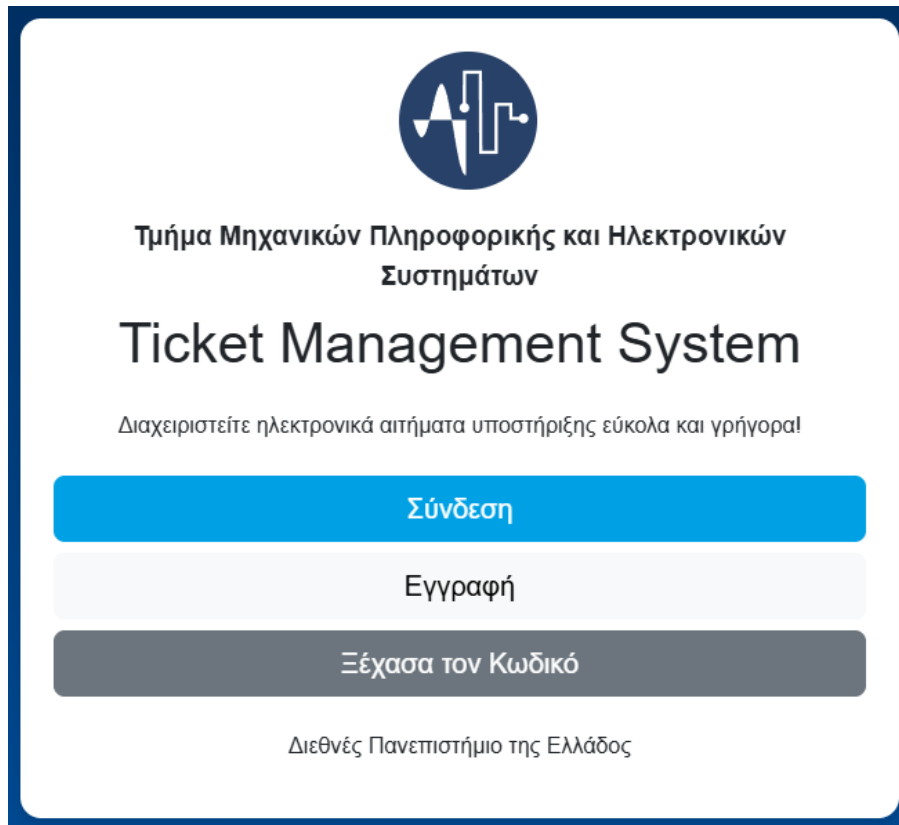


Εικόνα 4.4: Διάγραμμα ροής αλληλεπίδρασης συστήματος

Η ροή δεδομένων ξεκινά με τον χρήστη, ο οποίος υποβάλλει το αίτημα. Το αίτημα περνά στο Backend, όπου επεξεργάζεται και αποθηκεύεται στη βάση δεδομένων. Στη συνέχεια, το σύστημα ενημερώνει τον

διαχειριστή ή επιστρέφει πληροφορίες στον χρήστη. Το διάγραμμα απεικονίζει αυτή την αλληλεπίδραση με σαφήνεια, δείχνοντας τις συνδέσεις και την αμφίδρομη επικοινωνία που διασφαλίζει τη λειτουργικότητα του συστήματος.

4.2 Παρουσίαση Διεπαφής Χρήστη



Εικόνα 4.5: Σελίδα υποδοχής

Στην Εικόνα 4.5 παρουσιάζεται η σελίδα υποδοχής, η οποία αποτελεί την πρώτη επαφή του χρήστη με το σύστημα. Περιλαμβάνει ένα καλοσχεδιασμένο περιβάλλον με σαφείς επιλογές για εγγραφή ή σύνδεση.

Στην Εικόνα 4.6 παρουσιάζεται η σελίδα εγγραφής, που επιτρέπει στους χρήστες να δημιουργήσουν λογαριασμό παρέχοντας βασικά στοιχεία όπως όνομα, email, και κωδικό πρόσβασης. Διαθέτει πεδία με οδηγίες και ελέγχους εγκυρότητας για τη διασφάλιση σωστών εισαγωγών δεδομένων.

Εγγραφή

Όνομα

Email

Κωδικός

Εγγραφή

Εικόνα 4.6: Σελίδα Εγγραφής

Στην Εικόνα 4.7 φαίνεται η σελίδα σύνδεσης, η οποία επιτρέπει στους ήδη εγγεγραμμένους χρήστες να έχουν πρόσβαση στο σύστημα. Με ένα απλό και καθαρό layout, οι χρήστες μπορούν να εισάγουν τα στοιχεία τους και να μεταβούν εύκολα στον λογαριασμό τους.

Σύνδεση

Email

Κωδικός

Σύνδεση

[Ξέχασα τον Κωδικό](#)

[Δεν έχετε λογαριασμό, Εγγραφείτε](#)

Εικόνα 4.7: Σελίδα σύνδεσης

Dashboard

Καλωσορίσατε στο σύστημα διαχείρισης αιτημάτων υποστήριξης!

Τα Αιτήματά Μου

Δείτε τα αιτήματα που έχετε υποβάλει.

Προβολή

Εικόνα 4.8: Η κεντρική σελίδα Dashboard για τον απλό χρήστη

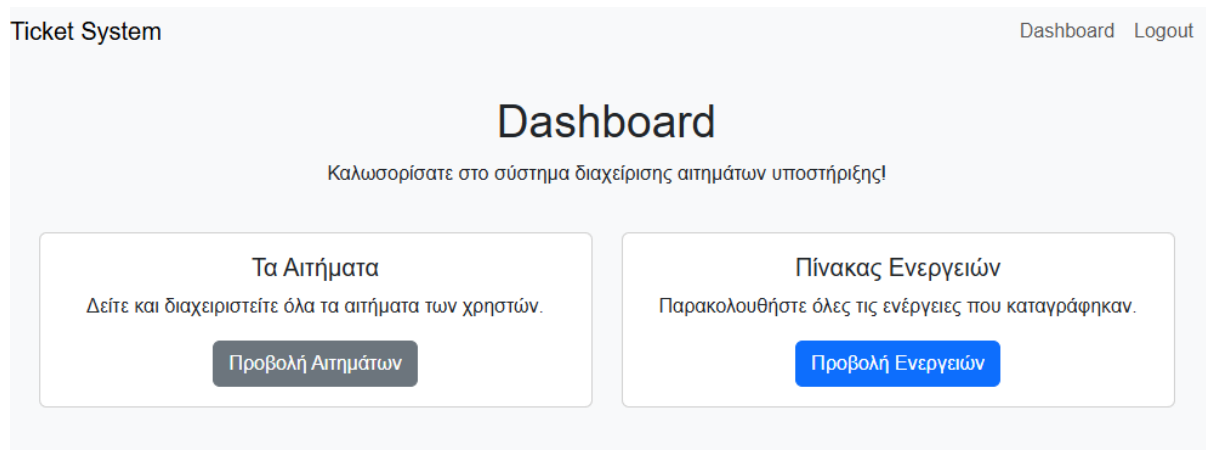
Στην Εικόνα 4.8 παρουσιάζεται η κεντρική σελίδα για τον απλό χρήστη, η οποία εμφανίζει πληροφορίες όπως αιτήματα, ειδοποιήσεις, και τρέχουσες ενέργειες. Παρέχει εύκολη πρόσβαση στη δημιουργία αιτημάτων ή την παρακολούθηση της κατάστασης αυτών.

Τα Αιτήματά Μου

#	Κατηγορία	Τίτλος	Περιγραφή	Κατάσταση	Ημερομηνία Δημιουργίας	Ενέργειες
1	Πρόβλημα με τη σύνδεση	Δεν αναγνωρίζει το email μου	Δεν αναγνωρίζει το email μου. Δεν μπορώ να συνδεθώ με το voud@gmail.com.	Σε Εξέλιξη	2025-01-14 16:11:56	Ακύρωση
Απαντήσεις:						
<ul style="list-style-type: none"> 2025-01-14 17:00:00 - Παρακαλώ δοκιμάστε ξανά. 						
2	Πρόβλημα το ανέβασμα αρχείου	Δεν μπορώ να ανεβάσω το αρχείο μου	Δεν μπορώ να ανεβάσω το αρχείο μου	Ακύρωση	2025-01-15 09:29:42	
Απαντήσεις:						
<ul style="list-style-type: none"> 2025-01-15 14:15:00 - Θα το εξετάσουμε. 						
5	Πρόβλημα με τη σύνδεση	Πρόβλημα σύνδεσης WiFi	Δεν μπορώ να συνδεθώ στο WiFi του πανεπιστημίου.	Ανοιχτό	2025-01-15 15:10:00	Ακύρωση
Απαντήσεις:						
<ul style="list-style-type: none"> Δεν υπάρχουν απαντήσεις. 						

Εικόνα 4.9: Η σελίδα με τα αιτήματα που έχει κάνει ο χρήστης

Στην Εικόνα 4.9 παρουσιάζεται η σελίδα που εμφανίζει αναλυτικά όλα τα αιτήματα που έχει υποβάλει ο χρήστης. Περιλαμβάνει πίνακα με τις κατηγορίες αιτημάτων, τον τίτλο, την περιγραφή, την κατάσταση και τις ημερομηνίες δημιουργίας.



Εικόνα 4.10: Η κεντρική σελίδα Dashboard για τον admin

Στην Εικόνα 4.10 φαίνεται η κεντρική σελίδα για τον admin, η οποία παρέχει πρόσβαση σε εργαλεία διαχείρισης όπως η προβολή όλων των αιτημάτων, η αλλαγή κατάστασης, και η διαχείριση απαντήσεων. Η διάταξη είναι προσαρμοσμένη για να υποστηρίξει τη γρήγορη εκτέλεση διαχειριστικών εργασιών.

Στην Εικόνα 4.11 παρουσιάζεται η σελίδα διαχείρισης αιτημάτων, η οποία επιτρέπει στον admin να προβάλει όλα τα αιτήματα σε μορφή πίνακα. Ενσωματώνει λειτουργίες όπως απαντήσεις, αλλαγή κατάστασης, και ακύρωση, με εύχρηστα κουμπιά για κάθε ενέργεια.

Διαχείριση Αιτημάτων

Δείξε εγγραφές Αναζήτηση:

#	Χρήστης	Κατηγορία	Τίτλος	Περιγραφή	Κατάσταση	Ημερομηνία Δημιουργίας	Ενέργειες
1	Βουδούρης Κώστας	Πρόβλημα με τη σύνδεση	Δεν αναγνωρίζει το email μου	Δεν αναγνωρίζει το email μου. Δεν μπορώ να συνδεθώ με το vourd@gmail.com.	Σε Εξέλιξη	14-01-2025 16:11	<div style="display: flex; flex-direction: column; gap: 5px;"> Ακύρωση Απάντηση Αλλαγή Κατάστασης Προβολή Απαντήσεων </div>
2	Βουδούρης Κώστας	Πρόβλημα το ανέβασμα αρχείου	Δεν μπορώ να ανεβάσω το αρχείο μου	Δεν μπορώ να ανεβάσω το αρχείο μου	Ακύρωση	15-01-2025 09:29	<div style="display: flex; flex-direction: column; gap: 5px;"> Επικοινωνία Απάντηση Αλλαγή Κατάστασης Προβολή Απαντήσεων </div>
3	Φοιτητής 3	Πρόβλημα την αποθήκευση	Πρόβλημα με αποθήκευση	Απώλεια δεδομένων.	Ανοιχτό	15-01-2025 14:20	<div style="display: flex; flex-direction: column; gap: 5px;"> Ακύρωση Απάντηση Αλλαγή Κατάστασης Προβολή Απαντήσεων </div>
4	Φοιτητής 3	Γενικό πρόβλημα	Γενικό πρόβλημα	Καθυστέρηση στη φόρτωση σελίδας.	Κλειστό	15-01-2025 14:30	<div style="display: flex; flex-direction: column; gap: 5px;"> Ακύρωση Απάντηση Αλλαγή Κατάστασης Προβολή Απαντήσεων </div>

Εικόνα 4.11: Διαχείριση Αιτημάτων από τον Admin

✕

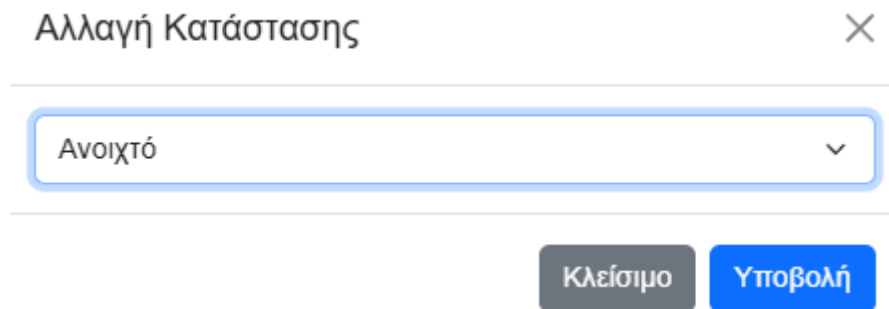
Απάντηση στο Αίτημα

Γράψτε την απάντησή σας εδώ...

Κλείσιμο
Υποβολή

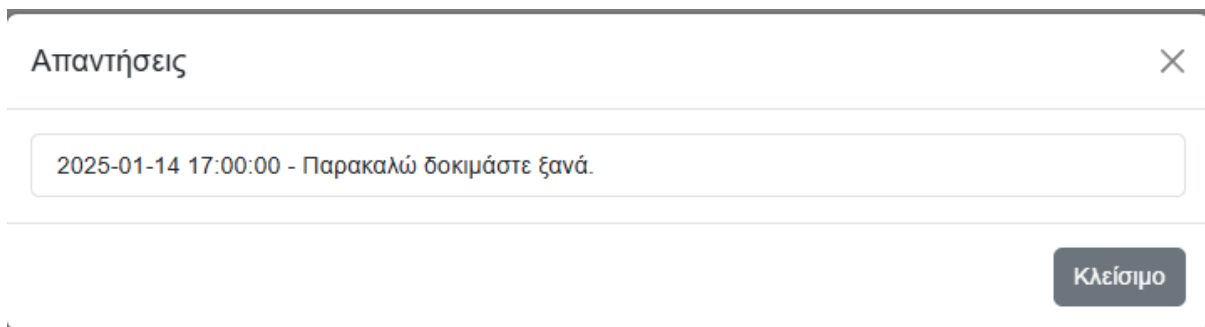
Εικόνα 4.12: Παράθυρο διαλόγου για την υποβολή απάντησης σε αίτημα

Στην Εικόνα 4.12 φαίνεται το παράθυρο διαλόγου για την απάντηση σε αίτημα. Επιτρέπει στον admin να εισαγάγει μια απάντηση και να την αποστείλει, προσφέροντας ένα απλό περιβάλλον με πεδία κειμένου και κουμπιά υποβολής για εύκολη χρήση.



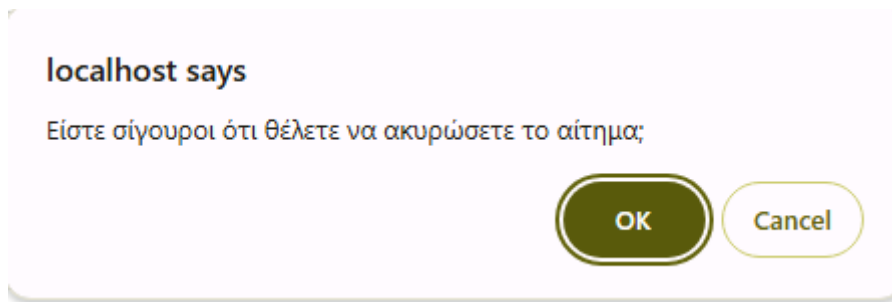
Εικόνα 4.13: Παράθυρο διαλόγου για την αλλαγή κατάστασης ενός αιτήματος

Στην Εικόνα 4.13 παρουσιάζεται το παράθυρο αλλαγής κατάστασης, το οποίο δίνει στον admin τη δυνατότητα να επιλέξει και να ενημερώσει την κατάσταση ενός αιτήματος (π.χ., "Ανοιχτό", "Σε Εξέλιξη", "Κλειστό"). Η διαδικασία γίνεται εύκολα και με ασφάλεια.



Εικόνα 4.14: Παράθυρο διαλόγου για τις απαντήσεις για ένα αίτημα

Στην Εικόνα 4.14 φαίνεται το παράθυρο που παρουσιάζει όλες τις απαντήσεις που έχουν δοθεί σε ένα αίτημα. Με καθαρό και οργανωμένο τρόπο, εμφανίζει τις ημερομηνίες και τα περιεχόμενα των απαντήσεων.



Εικόνα 4.15: Παράθυρο διαλόγου για την ακύρωση

Στην Εικόνα 4.15 παρουσιάζεται το παράθυρο ακύρωσης, το οποίο παρέχει στον admin τη δυνατότητα να ακυρώσει ένα αίτημα. Περιλαμβάνει επιβεβαιωτικά μηνύματα για την αποφυγή τυχόν λαθών και διασφαλίζει την ακρίβεια της ενέργειας.

Καταγραφή Ενεργειών

Δείξε 10 εγγραφές Αναζήτηση:

#	Αίτημα	Χρήστης	Ενέργεια	Ημερομηνία
1	Δεν μπορώ να ανεβάσω το αρχείο μου	Βουδούρης Κώστας	Cancelled	15-01-2025 09:41
2	Δεν αναγνωρίζει το email μου	Καθηγητής1	Responded	15-01-2025 12:16
3	Δεν αναγνωρίζει το email μου	Καθηγητής1	Responded	15-01-2025 12:16
4	Δεν αναγνωρίζει το email μου	Καθηγητής1	Responded	15-01-2025 12:17
5	Δεν αναγνωρίζει το email μου	Καθηγητής1	Responded	15-01-2025 12:18
6	Δεν αναγνωρίζει το email μου	Καθηγητής1	Status_changed	15-01-2025 12:24
7	Δεν μπορώ να ανεβάσω το αρχείο μου	Καθηγητής1	Reopened	15-01-2025 13:07
8	Δεν μπορώ να ανεβάσω το αρχείο μου	Καθηγητής1	Status_changed	15-01-2025 13:08

Εμφανίζονται 1 έως 8 από 8 εγγραφές Προηγούμενη 1 Επόμενη

Εικόνα 4.16: Πίνακα καταγραφής ενεργειών του admin

Στην Εικόνα 4.16 παρουσιάζεται ο πίνακας που καταγράφει λεπτομερώς τις ενέργειες του admin, όπως απαντήσεις, αλλαγές κατάστασης, ή ακυρώσεις αιτημάτων. Παρέχει ένα πλήρες ιστορικό για λόγους διαφάνειας και ελέγχου.

Στην Εικόνα 4.17 φαίνεται η προβολή ενός αιτήματος στον πίνακα καταγραφής ενεργειών, η οποία επιτρέπει στον admin να δει λεπτομέρειες για τη συγκεκριμένη ενέργεια. Παρουσιάζει πληροφορίες για το ποιος την εκτέλεσε και πότε, μαζί με το σχετικό αίτημα.

Προβολή Αιτήματος

Δεν μπορώ να ανεβάσω το αρχείο μου

Κατηγορία: Πρόβλημα το ανέβασμα αρχείου

Χρήστης: Βουδούρης Κώστας

Κατάσταση: Ακύρωση

Περιγραφή:
Δεν μπορώ να ανεβάσω το αρχείο μου

Ημερομηνία Δημιουργίας: 15-01-2025 09:29

Τελευταία Ενημέρωση: 15-01-2025 13:08

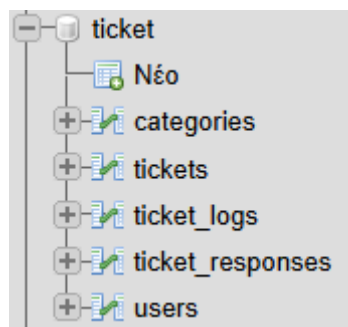
Απαντήσεις

15-01-2025 14:15: Θα το εξετάσουμε.

[Πίσω](#)

Εικόνα 4.17: Προβολή αιτήματος όταν ο admin πατήσει πάνω στο αίτημα στον πίνακα καταγραφής ενεργειών

4.3 Η Βάση



Εικόνα 4.18: Η βάση με τους πίνακες

Η βάση κατασκευάστηκε με MySQL MariaDB και στο Εικόνα 4.18 παρουσιάζεται το σχέδιο με τους πίνακες που χρησιμοποιήθηκαν.

Η βάση δεδομένων του συστήματος Ticket έχει σχεδιαστεί για να υποστηρίζει όλες τις βασικές λειτουργίες του συστήματος, όπως η υποβολή, η διαχείριση, η καταγραφή και η παρακολούθηση των αιτημάτων υποστήριξης. Παρακάτω παρουσιάζεται μια λεπτομερής ανάλυση των πινάκων, των πεδίων και της μεταξύ τους σχέσης.

1. Πίνακας categories

Ο πίνακας categories αποθηκεύει τις διαφορετικές κατηγορίες αιτημάτων που μπορούν να υποβάλλουν οι χρήστες. Κάθε κατηγορία έχει μια μοναδική αναγνωριστική τιμή.

- **Πεδία:**
 - id (Πρωτεύον κλειδί): Αναγνωριστικό κατηγορίας.
 - name: Όνομα της κατηγορίας (π.χ. "Πρόβλημα σύνδεσης").
 - description: Περιγραφή της κατηγορίας.
 - enable: Δείκτης ενεργοποίησης της κατηγορίας.
 - created_at, updated_at: Χρονosφραγίδες δημιουργίας και ενημέρωσης.
- **Λειτουργία:** Οι κατηγορίες χρησιμοποιούνται για την κατηγοριοποίηση των αιτημάτων, διευκολύνοντας τη διαχείριση από τον διαχειριστή.

2. Πίνακας tickets

Ο πίνακας tickets αποθηκεύει όλες τις πληροφορίες που σχετίζονται με τα αιτήματα υποστήριξης.

- **Πεδία:**
 - id (Πρωτεύον κλειδί): Αναγνωριστικό αιτήματος.
 - user_id: Αναγνωριστικό του χρήστη που υπέβαλε το αίτημα.
 - category_id: Συσχέτιση με τον πίνακα categories για την κατηγορία του αιτήματος.
 - title: Τίτλος του αιτήματος.
 - description: Περιγραφή του προβλήματος.
 - status: Κατάσταση του αιτήματος (Ανοιχτό, Σε Εξέλιξη, Κλειστό, Ακύρωση).
 - enable: Δείκτης ενεργοποίησης του αιτήματος.
 - created_at, updated_at: Χρονosφραγίδες δημιουργίας και ενημέρωσης.
- **Λειτουργία:** Τα αιτήματα καταχωρούνται σε αυτόν τον πίνακα με πληροφορίες για τον χρήστη, την κατηγορία, και την κατάσταση του αιτήματος.

3. Πίνακας ticket_logs

Ο πίνακας ticket_logs χρησιμοποιείται για την καταγραφή κάθε ενέργειας που σχετίζεται με ένα αίτημα.

- **Πεδία:**
 - id (Πρωτεύον κλειδί): Αναγνωριστικό της καταγραφής.
 - ticket_id: Συσχέτιση με το αναγνωριστικό του αιτήματος στον πίνακα tickets.

- user_id: Αναγνωριστικό του χρήστη που εκτέλεσε την ενέργεια.
- action: Περιγραφή της ενέργειας (π.χ., cancelled, responded, status_changed).
- note: Προαιρετική σημείωση για την ενέργεια.
- created_at: Χρονοσφραγίδα καταγραφής της ενέργειας.
- **Λειτουργία:** Ο πίνακας αυτός εξασφαλίζει πλήρη ιστορικότητα για κάθε αίτημα, επιτρέποντας την παρακολούθηση αλλαγών και ενεργειών.

4. Πίνακας ticket_responses

Ο πίνακας ticket_responses αποθηκεύει τις απαντήσεις που παρέχονται από τους διαχειριστές για τα αιτήματα υποστήριξης.

- **Πεδία:**
 - id (Πρωτεύον κλειδί): Αναγνωριστικό απάντησης.
 - ticket_id: Συσχέτιση με το αναγνωριστικό του αιτήματος στον πίνακα tickets.
 - admin_id: Αναγνωριστικό του διαχειριστή που έδωσε την απάντηση.
 - response: Το κείμενο της απάντησης.
 - created_at, updated_at: Χρονοσφραγίδες δημιουργίας και ενημέρωσης.
- **Λειτουργία:** Οι απαντήσεις διαχειριστών καταχωρούνται και συνδέονται με τα αντίστοιχα αιτήματα, επιτρέποντας στους χρήστες να δουν τις λύσεις που προσφέρονται.

5. Πίνακας users

Ο πίνακας users αποθηκεύει πληροφορίες για όλους τους χρήστες του συστήματος, συμπεριλαμβανομένων των διαχειριστών.

- **Πεδία:**
 - id (Πρωτεύον κλειδί): Αναγνωριστικό χρήστη.
 - name: Ονοματεπώνυμο του χρήστη.
 - email: Διεύθυνση email για ταυτοποίηση.
 - password: Κρυπτογραφημένος κωδικός πρόσβασης.
 - role: Ρόλος χρήστη (user, admin).
 - enable: Δείκτης ενεργοποίησης του λογαριασμού.

- created_at, updated_at: Χρονοσφραγίδες δημιουργίας και ενημέρωσης.
- **Λειτουργία:** Ο πίνακας αυτός περιλαμβάνει όλους τους χρήστες του συστήματος, υποστηρίζοντας τη διαχείριση δικαιωμάτων πρόσβασης.

Συσχέτιση των Πινάκων

Η βάση δεδομένων είναι σχεδιασμένη με τρόπο που να εξασφαλίζει τη συσχέτιση μεταξύ των πινάκων:

- Οι πίνακες tickets, ticket_logs, και ticket_responses συνδέονται μέσω του πεδίου ticket_id.
- Οι πίνακες users και tickets συνδέονται μέσω του user_id.
- Ο πίνακας categories συνδέεται με τον πίνακα tickets μέσω του category_id.

Η βάση δεδομένων του συστήματος Ticket έχει σχεδιαστεί για να υποστηρίζει με αποτελεσματικότητα τις λειτουργίες του συστήματος. Ο σωστός σχεδιασμός της επιτρέπει την εύκολη επέκταση, την ταχεία επεξεργασία δεδομένων και την εξασφάλιση της ιστορικότητας όλων των ενεργειών.

4.4 Ασφάλεια στο σύστημα και στα δεδομένα

Η ασφάλεια είναι ένας κρίσιμος παράγοντας για οποιοδήποτε σύστημα που διαχειρίζεται ευαίσθητες πληροφορίες και προσωπικά δεδομένα. Το σύστημα "Ticket" σχεδιάστηκε με στόχο να εξασφαλίζει την προστασία των δεδομένων χρηστών και τη διασφάλιση της ακεραιότητας του συστήματος.

Αρχές Ασφάλειας

Το σύστημα "Ticket" βασίζεται στις εξής βασικές αρχές ασφάλειας:

1. **Εμπιστευτικότητα (Confidentiality):** Εξασφαλίζεται ότι μόνο εξουσιοδοτημένοι χρήστες μπορούν να έχουν πρόσβαση στα δεδομένα.
2. **Ακεραιότητα (Integrity):** Διασφαλίζεται ότι τα δεδομένα είναι ακριβή και προστατευμένα από μη εξουσιοδοτημένες αλλαγές.
3. **Διαθεσιμότητα (Availability):** Εξασφαλίζεται ότι το σύστημα και τα δεδομένα είναι διαθέσιμα ανά πάσα στιγμή για τους εξουσιοδοτημένους χρήστες.
4. **Λογοδοσία (Accountability):** Όλες οι ενέργειες των χρηστών καταγράφονται, ώστε να υπάρχει πλήρες ιστορικό ενεργειών.

Μέτρα Ασφαλείας στο Σύστημα

Για την επίτευξη των παραπάνω στόχων, το σύστημα "Ticket" περιλαμβάνει τα εξής μέτρα ασφαλείας:

Αυθεντικοποίηση Χρηστών

Όλοι οι χρήστες (διαχειριστές και τελικοί χρήστες) πρέπει να συνδεθούν μέσω μοναδικών διαπιστευτηρίων (email και κωδικός πρόσβασης). Οι κωδικοί πρόσβασης αποθηκεύονται στη βάση δεδομένων σε κρυπτογραφημένη μορφή χρησιμοποιώντας ισχυρούς αλγορίθμους όπως το bcrypt.

Διαχείριση Ρόλων και Δικαιωμάτων

Το σύστημα εφαρμόζει αυστηρή πολιτική πρόσβασης βάσει ρόλων:

- Οι χρήστες με ρόλο **admin** έχουν πρόσβαση στις λειτουργίες διαχείρισης (προβολή και επεξεργασία όλων των αιτημάτων).
- Οι χρήστες με ρόλο **user** έχουν πρόσβαση μόνο στα αιτήματα που έχουν δημιουργήσει.

Προστασία Δεδομένων κατά τη Μεταφορά

Όλες οι επικοινωνίες μεταξύ του πελάτη (browser) και του διακομιστή προστατεύονται με κρυπτογράφηση SSL/TLS, ώστε να αποτρέπεται η υποκλοπή δεδομένων κατά τη μεταφορά.

Προστασία Δεδομένων στη Βάση

- Όλα τα ευαίσθητα δεδομένα, όπως οι κωδικοί πρόσβασης, αποθηκεύονται σε κρυπτογραφημένη μορφή.
- Η βάση δεδομένων προστατεύεται από μη εξουσιοδοτημένη πρόσβαση μέσω firewall και περιορισμένων δικαιωμάτων πρόσβασης.

Logging και Παρακολούθηση Ενεργειών

Κάθε ενέργεια των χρηστών, όπως αλλαγή κατάστασης αιτημάτων, ακύρωση, απαντήσεις ή προσθήκες, καταγράφεται στον πίνακα ticket_logs. Τα logs αυτά παρέχουν πλήρη διαφάνεια και λογοδοσία.

Προστασία από Επιθέσεις

Το σύστημα προστατεύεται από κοινές ευπάθειες, όπως:

- **SQL Injection:** Όλες οι εισερχόμενες τιμές ελέγχονται και περνούν από μηχανισμούς καθαρισμού πριν χρησιμοποιηθούν σε ερωτήματα βάσης δεδομένων.
- **Cross-Site Scripting (XSS):** Οι εισερχόμενες πληροφορίες φιλτράρονται για επικίνδυνες εντολές.
- **Cross-Site Request Forgery (CSRF):** Ενσωματώνονται CSRF tokens για την προστασία από ανεπιθύμητες ενέργειες.

Αντιμετώπιση Ενδεχόμενων Απειλών

Παρά τα προληπτικά μέτρα, το σύστημα σχεδιάστηκε ώστε να μπορεί να ανιχνεύει και να αποκρίνεται σε απειλές:

- **Ανίχνευση ύποπτων δραστηριοτήτων:** Ασυνήθιστη δραστηριότητα χρήστη, όπως πολλαπλές αποτυχημένες προσπάθειες σύνδεσης, καταγράφεται και μπορεί να οδηγήσει σε προσωρινή απενεργοποίηση του λογαριασμού.
- **Ανάκτηση από επιθέσεις:** Τακτικά backups της βάσης δεδομένων διασφαλίζουν την ανάκτηση σε περίπτωση απώλειας δεδομένων.

Κεφάλαιο 5ο: Συμπεράσματα και προτάσεις βελτίωσης

Η ανάπτυξη και υλοποίηση του συστήματος διαχείρισης αιτημάτων υποστήριξης που περιγράφηκε στην παρούσα εργασία αποτελεί μια σημαντική πρόοδο στην κατεύθυνση της αποτελεσματικής διαχείρισης των τεχνικών προβλημάτων και αιτημάτων των χρηστών. Το σύστημα σχεδιάστηκε για να είναι απλό στη χρήση, ευέλικτο και επεκτάσιμο, ενώ ενσωματώνει σύγχρονες τεχνολογίες, όπως το Laravel, το PHP, το Bootstrap και το MySQL.

Μέσα από την ανάλυση και την υλοποίηση που παρουσιάστηκαν, προέκυψαν τα εξής κύρια συμπεράσματα:

- **Αύξηση της αποτελεσματικότητας:** Το σύστημα επιτρέπει την άμεση διαχείριση και παρακολούθηση των αιτημάτων τόσο από τους χρήστες όσο και από τους διαχειριστές. Ο χρόνος απόκρισης μειώθηκε σημαντικά, επιτυγχάνοντας καλύτερη εμπειρία χρήστη.
- **Βελτίωση της διαφάνειας:** Η δυνατότητα καταγραφής όλων των ενεργειών (log actions) διασφαλίζει την πλήρη διαφάνεια και την υπευθυνότητα στη διαχείριση των αιτημάτων.
- **Ασφάλεια δεδομένων:** Ενσωματώθηκαν μηχανισμοί ασφαλείας που διασφαλίζουν την προστασία των προσωπικών δεδομένων των χρηστών και την ασφάλεια κατά την ανταλλαγή πληροφοριών.
- **Ευελξία και προσαρμοστικότητα:** Η χρήση μοντέρνων εργαλείων και τεχνολογιών καθιστά το σύστημα εύκολα επεκτάσιμο και προσαρμόσιμο στις ανάγκες του οργανισμού.

Παράλληλα, η εργασία ανέδειξε την αναγκαιότητα της ύπαρξης τέτοιων συστημάτων σε σύγχρονους οργανισμούς, καθώς και τη σημασία της απλότητας και της φιλικότητας προς τον χρήστη. Η προσέγγιση αυτή ενισχύει την εμπιστοσύνη των χρηστών προς το σύστημα και αυξάνει τη συμμετοχή τους στη χρήση του.

Προτάσεις Βελτίωσης

Παρότι το σύστημα υλοποιήθηκε με επιτυχία και πληροί τις βασικές απαιτήσεις, υπάρχει πάντα περιθώριο για βελτιώσεις και μελλοντική εξέλιξη. Η ενσωμάτωση τεχνητής νοημοσύνης μπορεί να βοηθήσει στην αυτόματη κατηγοριοποίηση των αιτημάτων και την πρόβλεψη πιθανών λύσεων με βάση τη βάση δεδομένων παλαιότερων αιτημάτων. Επιπλέον, η χρήση chatbots για την άμεση επικοινωνία με τους χρήστες μπορεί να μειώσει την πίεση στους διαχειριστές.

Η ανάπτυξη μιας εφαρμογής για κινητές συσκευές θα προσφέρει στους χρήστες την ευελξία να υποβάλλουν αιτήματα και να παρακολουθούν την κατάσταση αυτών από οπουδήποτε και οποτεδήποτε. Η mobile-first προσέγγιση είναι πλέον απαραίτητη σε κάθε σύγχρονο σύστημα.

Για να καταστεί το σύστημα προσιτό σε ένα ευρύτερο φάσμα χρηστών, είναι σημαντικό να ενσωματωθεί υποστήριξη για πολλές γλώσσες. Αυτό θα επιτρέψει τη χρήση του συστήματος σε πολυπολιτισμικούς οργανισμούς.

Η ανάπτυξη πιο προηγμένων εργαλείων αναφορών και στατιστικών μπορεί να παρέχει πολύτιμα δεδομένα σχετικά με τις επιδόσεις του συστήματος και τα μοτίβα αιτημάτων. Οι διαχειριστές μπορούν να χρησιμοποιήσουν αυτές τις πληροφορίες για τη λήψη στρατηγικών αποφάσεων.

Η ενσωμάτωση του συστήματος με εξωτερικές πλατφόρμες, όπως CRM ή ERP, μπορεί να βελτιώσει τη ροή εργασίας και να επιτρέψει την ανταλλαγή δεδομένων μεταξύ διαφορετικών τμημάτων ενός οργανισμού.

Η εργασία αυτή προσφέρει μια σταθερή βάση για τη μελλοντική βελτίωση και ενίσχυση του συστήματος καθιστώντας το ένα πολύτιμο εργαλείο για την υποστήριξη χρηστών και την ενίσχυση της αποτελεσματικότητας σε οργανισμούς. Επίσης χρησιμοποιήθηκε το chat της open ai σε διάφορα σημεία της εργασίας για τη παραγωγή και διόρθωση του συντακτικού και της σωστής διαμόρφωσης του κειμένου.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] <https://www.zendesk.com/>
- [2] <https://www.freshworks.com/freshdesk/>
- [3] <https://www.atlassian.com/software/jira>
- [4] https://www.researchgate.net/publication/358603069_Ticketing_System_A_Descriptive_Research_on_the_Use_of_Ticketing_System_for_Project_Management_and_Issue_Tracking_in_IT_Companies
- [5] <https://www.ijert.org/research/enhancing-a-ticketing-system-with-ai-intelligence-IJERTV13IS070012.pdf>
- [6] <https://laravel.com/docs/11.x>
- [7] <https://trends.builtwith.com/framework/Laravel>
- [8] <https://infostride.com/sites-built-with-laravel/>
- [9] <https://www.addwebsolution.com/blog/11-successful-websites-developed-using-laravel>
- [10] <https://datafloq.com/read/laravel-usage-statistics-what-they-tell-us-about-the-future-of-web-development-2/>
- [11] <https://www.glorywebs.com/blog/laravel-usage-statistics>
- [12] <https://codeinstitute.net/global/blog/what-is-php-programming/>
- [13] <https://skillcrush.com/blog/php/>
- [14] https://www.w3schools.com/php/php_intro.asp
- [15] <https://www.freecodecamp.org/news/what-is-php-the-php-programming-language-meaning-explained/>
- [16] <https://builtin.com/software-engineering-perspectives/php>
- [17] <https://www.pixelcrayons.com/blog/dedicated-teams/php-vs-python-vs-ruby-comparison/>
- [18] <https://www.techaheadcorp.com/blog/exploring-the-javascript-vs-other-languages-debate-a-balanced-perspective-for-backend-development/>
- [19] <https://dev.to/alexmercedcoder/10-programming-languages-side-by-side-js-python-ruby-php-go-rust-dart-c-java-ballerina-3b0f>
- [20] <https://w3techs.com/>
- [21] <https://www.statista.com/>
- [22] <https://aundigital.ae/blog/laravel-usage-statistics/>
- [23] <https://builtin.com/>
- [24] <https://www.coursera.org/articles/popular-programming-languages>
- [25] <https://www.w3schools.com/php/>
- [26] <https://getbootstrap.com/>
- [27] <https://www.w3schools.com/bootstrap5/>

- [28] <https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/>
- [29] <https://www.mysql.com/>
- [30] <https://db-engines.com/en/ranking>
- [31] https://www.w3schools.com/mysql/mysql_intro.asp
- [32] <https://laravel.com/docs/11.x/database>

ΠΑΡΑΡΤΗΜΑ Α

Στο παράρτημα αυτό αναφέρονται τα βασικά κομμάτια του κώδικα που χρησιμοποιήθηκε.

AdminController

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Session;

class AdminController extends Controller
{
    // Επιστρέφει τη λίστα αιτημάτων για τον διαχειριστή
    public function adminIndex()
    {
        // Έλεγχος αν ο χρήστης έχει δικαιώματα admin
        if (!Session::has('user_id') || Session::get('user_role') !== 'admin') {
            return redirect()->route('login')->withErrors('Δεν έχετε δικαίωμα πρόσβασης.');
```

```

{
    // Έλεγχος αν ο χρήστης έχει δικαιώματα admin
    if (!Session::has('user_id') || Session::get('user_role') !== 'admin') {
        return response()->json(['error' => 'Unauthorized'], 403);
    }

    // Ανάκτηση απαντήσεων με ταξινόμηση κατά ημερομηνία
    $responses = DB::table('ticket_responses')
        ->where('ticket_id', $ticketId)
        ->orderBy('created_at', 'desc')
        ->get(['response', 'created_at']);

    return response()->json(['responses' => $responses]);
}

// Ακυρώνει ένα αίτημα
public function cancelTicket(Request $request, $id)
{
    // Έλεγχος αν ο χρήστης έχει δικαιώματα admin
    if (!Session::has('user_id') || Session::get('user_role') !== 'admin') {
        return redirect()->route('login')->withErrors('Δεν έχετε δικαίωμα πρόσβασης.');
```

```

// Ενημέρωση κατάστασης του αιτήματος σε 'Ανοιχτό'
DB::table('tickets')->where('id', $id)->update([
    'status' => 'Ανοιχτό',
    'updated_at' => now(),
]);

// Καταγραφή της ενέργειας στα logs
DB::table('ticket_logs')->insert([
    'ticket_id' => $id,
    'action' => 'reopened',
    'user_id' => Session::get('user_id'),
    'created_at' => now(),
]);

return redirect()->route('admin.ticket.index')->with('success', 'Το αίτημα επανήλθε
επιτυχώς.');
```

```

    }

// Καταχωρεί απάντηση για συγκεκριμένο αίτημα
public function respondTicket(Request $request, $id)
{
    // Επικύρωση δεδομένων απάντησης
    $request->validate([
        'response' => 'required|string|max:1000',
    ]);

    // Έλεγχος αν ο χρήστης έχει δικαιώματα admin
    if (!Session::has('user_id') || Session::get('user_role') !== 'admin') {
        return redirect()->route('login')->withErrors('Δεν έχετε δικαίωμα πρόσβασης.');
```

```

    }

// Εισαγωγή απάντησης στη βάση δεδομένων
DB::table('ticket_responses')->insert([
    'ticket_id' => $id,
    'admin_id' => Session::get('user_id'),
    'response' => $request->response,
    'created_at' => now(),
]);

// Καταγραφή της ενέργειας στα logs
DB::table('ticket_logs')->insert([
    'ticket_id' => $id,
    'action' => 'responded',
    'user_id' => Session::get('user_id'),
    'created_at' => now(),
]);

```

```

        return redirect()->route('admin.ticket.index')->with('success', 'Η απάντηση καταχωρήθηκε με
επιτυχία.');
```

```

    }

    // Αλλάζει την κατάσταση ενός αιτήματος
    public function changeStatus(Request $request, $id)
    {
        // Επικύρωση δεδομένων κατάστασης
        $request->validate([
            'status' => 'required|string|in:Ανοιχτό,Σε Εξέλιξη,Κλειστό,Ακύρωση',
        ]);

        // Έλεγχος αν ο χρήστης έχει δικαιώματα admin
        if (!Session::has('user_id') || Session::get('user_role') !== 'admin') {
            return redirect()->route('login')->withErrors('Δεν έχετε δικαίωμα πρόσβασης.');
```

```

        }

        // Ενημέρωση κατάστασης του αιτήματος
        DB::table('tickets')->where('id', $id)->update([
            'status' => $request->status,
            'updated_at' => now(),
        ]);

        // Καταγραφή της ενέργειας στα logs
        DB::table('ticket_logs')->insert([
            'ticket_id' => $id,
            'action' => 'status_changed',
            'user_id' => Session::get('user_id'),
            'created_at' => now(),
        ]);

        return redirect()->route('admin.ticket.index')->with('success', 'Η κατάσταση του αιτήματος
άλλαξε με επιτυχία.');
```

```

    }

    // Επιστρέφει λεπτομέρειες για συγκεκριμένο αίτημα
    public function viewTicket($id)
    {
        // Έλεγχος αν ο χρήστης έχει δικαιώματα admin
        if (!Session::has('user_id') || Session::get('user_role') !== 'admin') {
            return redirect()->route('login')->withErrors('Δεν έχετε δικαίωμα πρόσβασης.');
```

```

        }

        // Ανάκτηση δεδομένων του αιτήματος
        $ticket = DB::table('tickets')
            ->join('categories', 'tickets.category_id', '=', 'categories.id')
            ->join('users', 'tickets.user_id', '=', 'users.id')
            ->select(

```

```

        'tickets.*',
        'categories.name as category_name',
        'users.name as user_name'
    )
    ->where('tickets.id', $id)
    ->first();

    if (!$ticket) {
        return redirect()->route('admin.ticket.index')->withErrors('Το αίτημα δεν βρέθηκε.');
```

```
    }
```

```

    // Ανάκτηση των απαντήσεων του αιτήματος
    $responses = DB::table('ticket_responses')
        ->where('ticket_id', $id)
        ->orderBy('created_at', 'asc')
        ->get();

```

```

    $ticket->responses = $responses;

```

```

    return view('ticket.view', ['ticket' => $ticket]);

```

```

    }
}

```

TicketController

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Session;

class TicketController extends Controller
{
    // Εμφάνιση των αιτημάτων του χρήστη
    public function userIndex()
    {
        // Έλεγχος αν ο χρήστης είναι συνδεδεμένος
        if (!Session::has('user_id')) {
            return redirect()->route('login');
        }

        $userId = Session::get('user_id');

        // Ανάκτηση αιτημάτων που ανήκουν στον συνδεδεμένο χρήστη
        $tickets = DB::table('tickets')
            ->join('categories', 'tickets.category_id', '=', 'categories.id')
            ->where('tickets.user_id', $userId)

```

```

->select('tickets.*', 'categories.name as category_name')
->get();

// Ανάκτηση των απαντήσεων για κάθε αίτημα
foreach ($tickets as $ticket) {
    $ticket->responses = DB::table('ticket_responses')
        ->where('ticket_id', $ticket->id)
        ->orderBy('created_at', 'asc') // Ταξινόμηση από την παλαιότερη στην πιο πρόσφατη
    απάντηση
        ->get();
}

return view('ticket.userindex', ['tickets' => $tickets]);
}

// Επιστροφή φόρμας για δημιουργία νέου αιτήματος
public function userCreate()
{
    // Έλεγχος αν ο χρήστης είναι συνδεδεμένος
    if (!Session::has('user_id')) {
        return redirect()->route('login');
    }

    // Ανάκτηση διαθέσιμων κατηγοριών
    $categories = DB::table('categories')->where('enable', 1)->get();

    return view('ticket.usercreate', ['categories' => $categories]);
}

// Αποθήκευση νέου αιτήματος στη βάση δεδομένων
public function store(Request $request)
{
    // Επικύρωση δεδομένων
    $request->validate([
        'category_id' => 'required|integer',
        'title' => 'required|string|max:255',
        'description' => 'required|string',
    ]);

    // Έλεγχος αν ο χρήστης είναι συνδεδεμένος
    if (!Session::has('user_id')) {
        return redirect()->route('login');
    }

    // Δημιουργία αιτήματος
    $ticketId = DB::table('tickets')->insertGetId([
        'user_id' => Session::get('user_id'),
        'category_id' => $request->category_id,
    ]);
}

```

```

        'title' => $request->title,
        'description' => $request->description,
        'status' => 'open',
        'enable' => 1,
        'created_at' => now(),
    ]);

    // Καταγραφή δημιουργίας αιτήματος στα logs
    DB::table('ticket_logs')->insert([
        'ticket_id' => $ticketId,
        'action' => 'created',
        'user_id' => Session::get('user_id'),
        'created_at' => now(),
    ]);

    return redirect()->route('ticket.userindex')->with('success', 'Το αίτημα δημιουργήθηκε με
επιτυχία.');
```

```

    }

    // Ακύρωση ενός αιτήματος
    public function cancel(Request $request, $id)
    {
        // Έλεγχος αν ο χρήστης είναι συνδεδεμένος
        if (!Session::has('user_id')) {
            return redirect()->route('login');
        }

        // Ανάκτηση του αιτήματος
        $ticket = DB::table('tickets')->where('id', $id)->first();

        // Έλεγχος αν το αίτημα ανήκει στον συνδεδεμένο χρήστη
        if (!$ticket || $ticket->user_id != Session::get('user_id')) {
            return redirect()->route('ticket.index')->withErrors('Δεν έχετε πρόσβαση σε αυτό το
αίτημα.');
```

```

        }

        // Ενημέρωση κατάστασης του αιτήματος σε "Ακυρώθηκε"
        DB::table('tickets')->where('id', $id)->update([
            'enable' => 0, // Απενεργοποίηση του αιτήματος
            'status' => 'cancelled',
            'updated_at' => now(),
        ]);

        // Καταγραφή ακύρωσης του αιτήματος στα logs
        DB::table('ticket_logs')->insert([
            'ticket_id' => $id,
            'action' => 'cancelled',
            'user_id' => Session::get('user_id'),

```

```
        'created_at' => now(),
    ]);

    return redirect()->route('ticket.index')->with('success', 'Το αίτημα ακυρώθηκε με επιτυχία.');
```

```
}

// Εμφάνιση καταγραφών ενεργειών (logs)
public function logs()
{
    // Ανάκτηση καταγραφών ενεργειών
    $logs = DB::table('ticket_logs')
        ->join('tickets', 'ticket_logs.ticket_id', '=', 'tickets.id')
        ->join('users', 'ticket_logs.user_id', '=', 'users.id')
        ->select('ticket_logs.*', 'tickets.title as ticket_title', 'users.name as user_name')
        ->orderBy('ticket_logs.created_at', 'desc')
        ->get();

    return view('ticket.logs', ['logs' => $logs]);
}
}
```