

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Σύστημα καταγραφής και ανάλυσης παρουσιών φοιτητών
με χρήση ψηφιακών τεχνολογιών»

Λίστα Ερωτήσεων

+ Νέα Ερώτηση

Show 10 entries

Search:

#	Ερώτηση	Μάθημα	Κωδικός	Ισχύει Από	Ισχύει Μέχρι	Ενέργειες
1	Τι είναι το TCP/IP?	Δίκτυα	1101	13-05-2025 22:00	13-05-2025 22:20	Επεξεργασία Διαγραφή Αποτελέσματα
2	Ένας διακομιστής είναι	Δίκτυα	2101	13-05-2025 22:20	13-05-2025 22:40	Επεξεργασία Διαγραφή Αποτελέσματα
3	Ποια από τις παρακάτω κλάσεις χρησιμοποιείται για...	Προγραμματισμός Java	3102	14-05-2025 12:30	14-05-2025 12:40	Επεξεργασία Διαγραφή Αποτελέσματα

Showing 1 to 3 of 3 entries

Previous 1 Next

Φοιτητής

134031

ΚΩΝΣΤΑΝΤΙΝΟΣ ΜΠΡΑΒΟΣ

Επιβλέπων

Δρ. Κυριάκος Τσιακμάκης

Ιούνιος 2025

Σύστημα καταγραφής και ανάλυσης παρουσιών φοιτητών με χρήση ψηφιακών τεχνολογιών

Κωδικός: 25151

Φοιτητής: Μπράβος Κωνσταντίνος

Εισηγητής: Δρ Κυριάκος Τσιακμάκης

Ημερομηνία ανάληψης Π.Ε. 08-03-2025

Ημερομηνία περάτωσης Π.Ε. 31-05-2025

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 Μπράβου Κωνσταντίνου 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

Το έργο αφορά την ανάπτυξη ενός σύγχρονου συστήματος καταγραφής και ανάλυσης παρουσιών φοιτητών με χρήση ψηφιακών τεχνολογιών. Στόχος του συστήματος είναι να διευκολύνει τους καθηγητές στην άμεση καταγραφή παρουσιών κατά τη διάρκεια των μαθημάτων και την αξιολόγηση της συμμετοχής των φοιτητών. Η υλοποίηση βασίστηκε στο Laravel framework για τη διαχείριση της εφαρμογής και το MySQL για την αποθήκευση των δεδομένων. Παράλληλα χρησιμοποιήθηκαν οι βιβλιοθήκες Chart.js και DataTables για την παρουσίαση των αποτελεσμάτων και των στατιστικών στοιχείων με εύληπτο και γραφικό τρόπο. Το σύστημα υποστηρίζει την καταγραφή παρουσίας μέσω μοναδικού κωδικού που δημιουργεί ο καθηγητής και παρέχει ασφάλεια και αξιοπιστία. Η εφαρμογή είναι πλήρως responsive και φιλική προς τον χρήστη, προσφέροντας μια ολοκληρωμένη λύση για τη διαχείριση και ανάλυση παρουσιών σε ακαδημαϊκά ιδρύματα.

« Student attendance monitoring and analysis system using digital technologies»

Abstract

The project concerns the development of a modern system for recording and analyzing student attendance using digital technologies. The aim of the system is to facilitate teachers in directly recording attendance during classes and evaluating student participation. The implementation was based on the Laravel framework for managing the application and MySQL for storing data. At the same time, the Chart.js and DataTables libraries were used to present the results and statistics in an understandable and graphical way. The system supports attendance recording through a unique code created by the teacher, offering security and reliability. The application is fully responsive and user-friendly, offering a comprehensive solution for managing and analyzing attendance in academic institutions.

Ευχαριστίες

Θέλω να ευχαριστήσω τους γονείς μου και τους φίλους μου.

Περιεχόμενα

Περίληψη	iv
Abstract	v
Ευχαριστίες	vi
Περιεχόμενα.....	vii
Κατάλογος Σχημάτων	viii
Κεφάλαιο 1ο: Εισαγωγή.....	9
1.1 Εισαγωγή.....	9
Κεφάλαιο 2ο: Συστήματα καταγραφής παρουσιών φοιτητών με χρήση ψηφιακών τεχνολογιών 12	
2.1 Top Hat [1-2].....	12
2.2 QuickSchools [3].....	14
2.3 MyAttendanceTracker [4]	15
2.4 Attendance Radar [5].....	16
2.5 ScholarChip [6].....	18
Κεφάλαιο 3ο: Χρήση τεχνολογιών για το έργο.....	19
3.1 Laravel.....	19
3.2 MySQL.....	21
3.3 Chart.js.....	23
3.4 Datatables	23
3.5 Bootstrap.....	25
Κεφάλαιο 4ο: Το σύστημα καταγραφής και ανάλυσης παρουσιών φοιτητών με χρήση ψηφιακών τεχνολογιών.....	27
4.1 Μια μικρή περιγραφή	27
4.2 Η βάση δεδομένων του συστήματος μας.....	27
4.3 Το σύστημα.....	31
4.3.1 Διαγράμματα Ροής	31
4.3.2 Περιγραφή και Παρουσίαση της Ιστοσελίδας.....	33
Κεφάλαιο 5ο: Τα συμπεράσματα της εργασίας και τι μπορεί να βελτιωθεί.....	45
ΒΙΒΛΙΟΓΡΑΦΙΑ	46
ΠΑΡΑΡΤΗΜΑ.....	47

Κατάλογος Σχημάτων

Εικόνα 2.1: Tophat πλατφόρμα [https://tophat.com/]	13
Εικόνα 2.2: QuickSchools πλατφόρμα	15
Εικόνα 2.2: MyAttendanceTracker πλατφόρμα	16
Εικόνα 2.3: Attendance Radar πλατφόρμα	17
Εικόνα 3.1: Πλεονεκτήματα Laravel	19
Εικόνα 3.1: Διαδικασία Διαχείρισης από τον Διαχειριστή (Admin) - Διαχείριση Μαθημάτων και Καθηγητών από Admin	31
Εικόνα 3.2: Ροή Διαχείρισης από τον Καθηγητή - Διαδικασία Δημιουργίας και Παρακολούθησης Ερωτήσεων	32
Εικόνα 3.3: Ροή Χρήσης από τον Φοιτητή - Διαδικασία Δήλωσης Παρουσίας από τον Φοιτητή ...	33
Εικόνα 3.4: Σύνδεση χρήστη – εκτός φοιτητή	34
Εικόνα 3.5: Εγγραφή χρήστη – εκτός φοιτητή	35
Εικόνα 3.6: Πίνακας Ελέγχου Διαχειριστή - Admin	36
Εικόνα 3.7: Λίστα Μαθημάτων - Admin	36
Εικόνα 3.8: Επεξεργασία Μαθήματος - Admin	37
Εικόνα 3.9: Δημιουργία Νέου Μαθήματος - Admin	37
Εικόνα 3.10: Πίνακας Ελέγχου Καθηγητή - Καθηγητής	38
Εικόνα 3.11: Λίστα Ερωτήσεων - Καθηγητής	38
Εικόνα 3.12: Δημιουργία Νέας Ερώτησης - Καθηγητής	39
Εικόνα 3.13: Επιλογές dropdown λίστα για Μάθημα και Σωστής Επιλογής και φόρμα για τον καθορισμό ημερομηνιών κατά τη Δημιουργία Νέας Ερώτησης - Καθηγητής	40
Εικόνα 3.14: Επεξεργασία Ερώτησης - Καθηγητής	41
Εικόνα 3.15: Αποτελέσματα Ερώτησης με προβολή της σωστής απόκρισης - Καθηγητής	41
Εικόνα 3.16: Στατιστικά απαντήσεων σε bar chart - Καθηγητής	42
Εικόνα 3.17: Excel αρχείο με τα αποτελέσματα - Καθηγητής	42
Εικόνα 3.18: Δήλωση Παρουσίας – Εισαγωγή κωδικού – Φοιτητής	43
Εικόνα 3.19: Εισαγωγή κωδικού σε ερώτηση που δεν υπάρχει ή είναι εκτός προθεσμίας – Φοιτητής	43
Εικόνα 3.20: Απάντηση στην Ερώτηση – Εισαγωγή όλων των απαραίτητων στοιχείων – Φοιτητής	44

Κεφάλαιο 1ο: Εισαγωγή

1.1 Εισαγωγή

Η παρακολούθηση παρουσιών αποτελεί μια από τις βασικές λειτουργίες στον χώρο της εκπαίδευσης, τόσο στη δευτεροβάθμια όσο και στην ανώτατη εκπαίδευση. Παραδοσιακά η καταγραφή παρουσιών πραγματοποιείται με χειρόγραφες μεθόδους όπως είναι η συμπλήρωση παρουσιολογίων ή η προφορική επιβεβαίωση της παρουσίας από τον καθηγητή. Αυτές οι μέθοδοι αν και απλές παρουσιάζουν σημαντικά μειονεκτήματα όπως καθυστερήσεις στη διαδικασία και σφάλματα και αδυναμία σωστής επεξεργασίας των δεδομένων που συλλέγονται.

Με την πρόοδο της τεχνολογίας και τη μεγάλη είσοδο ψηφιακών εργαλείων στην εκπαίδευση αναπτύχθηκαν νέα αυτοματοποιημένα συστήματα που στοχεύουν στη βελτίωση αυτής της διαδικασίας. Τα συστήματα αυτά δίνουν τη δυνατότητα στους εκπαιδευτικούς να παρακολουθούν την προσέλευση των φοιτητών εύκολα και γρήγορα και παρέχουν δυνατότητες επεξεργασίας, στατιστικής ανάλυσης και εξαγωγής δεδομένων για περαιτέρω αξιοποίηση.

Συναντάμε συστήματα που χρησιμοποιούν τεχνολογίες όπως η αναγνώριση προσώπου, οι κωδικοί QR, οι κάρτες RFID και οι διαδικτυακές πλατφόρμες. Αυτές οι λύσεις έχουν μεγαλύτερη ακρίβεια, ασφάλεια και αυτοματοποίηση και απαιτούν και μεγαλύτερους πόρους υλικού και λογισμικού. Σε πολλές περιπτώσεις ειδικά σε πανεπιστημιακά ιδρύματα με μεγάλο αριθμό φοιτητών, οι παραδοσιακές μέθοδοι δεν επαρκούν και η ανάγκη για πιο σύγχρονες και αποδοτικές λύσεις γίνεται αναγκαία.

Η ένταξη τέτοιων συστημάτων διευκολύνει όχι μόνο την καταγραφή παρουσιών αλλά και τη γενικότερη διαχείριση της εκπαιδευτικής διαδικασίας. Με άμεση ανατροφοδότηση στατιστικά δεδομένα και εύκολη πρόσβαση σε αναφορές τα συστήματα αυτά αποτελούν πολύτιμο εργαλείο τόσο για τη βελτίωση της διδασκαλίας και για τη διοικητική διαχείριση των εκπαιδευτικών ιδρυμάτων.

Στο πλαίσιο της παρούσας εργασίας αναπτύχθηκε ένα διαδικτυακό σύστημα καταγραφής παρουσιών με τη χρήση ερωτήσεων πολλαπλής επιλογής. Η βασική ιδέα του συστήματος είναι απλή και άμεσα εφαρμόσιμη στην πράξη. Οι φοιτητές δηλώνουν την παρουσία τους απαντώντας σε μία σύντομη ερώτηση που θέτει ο καθηγητής κατά τη διάρκεια του μαθήματος. Με αυτόν τον τρόπο η καταγραφή παρουσιών γίνεται πιο οργανωμένα και αποφεύγονται φαινόμενα τυπικής παρουσίας χωρίς συμμετοχή και παράλληλα παρέχεται ένα μέσο άμεσης αξιολόγησης της προσοχής και κατανόησης του μαθήματος από τους φοιτητές.

Το σύστημα προσφέρει ξεχωριστές διεπαφές για τους διαχειριστές, τους καθηγητές και τους φοιτητές. Οι διαχειριστές έχουν τη δυνατότητα να δημιουργούν και να διαχειρίζονται μαθήματα, να μπορούν να ορίζουν καθηγητές υπεύθυνους για κάθε μάθημα για να επιβλέπουν τη λειτουργία του συστήματος. Οι

καθηγητές μπορούν να δημιουργούν εύκολα ερωτήσεις πολλαπλής επιλογής, να καθορίζουν το χρονικό διάστημα μέσα στο οποίο οι ερωτήσεις θα είναι διαθέσιμες και να παρακολουθούν σε πραγματικό χρόνο τις απαντήσεις των φοιτητών.

Από την πλευρά των φοιτητών η διαδικασία είναι απλή και γρήγορη. Κατά τη διάρκεια του μαθήματος, ο καθηγητής ανακοινώνει έναν μοναδικό κωδικό ερώτησης. Οι φοιτητές, μέσω ενός ειδικού συνδέσμου, εισάγουν τον κωδικό, απαντούν στην ερώτηση και συμπληρώνουν τα προσωπικά τους στοιχεία για την επιβεβαίωση της παρουσίας τους. Το σύστημα καταγράφει επιπλέον την IP διεύθυνση της συσκευής του φοιτητή για να διασφαλίζεται η μοναδικότητα της συμμετοχής του.

Το σύστημα παρέχει τη δυνατότητα στον καθηγητή να δει άμεσα τα αποτελέσματα των απαντήσεων και να εξάγει στατιστικά διαγράμματα και αναφορές. Με αυτόν τον τρόπο διευκολύνεται όχι μόνο η καταγραφή παρουσιών αλλά και η αξιολόγηση της συμμετοχής και της κατανόησης των φοιτητών κατά τη διάρκεια του μαθήματος.

Η ανάπτυξη του συστήματος βασίστηκε σε σύγχρονες τεχνολογίες όπως η πλατφόρμα Laravel για την υλοποίηση της διαδικτυακής εφαρμογής, η βάση δεδομένων MySQL για την αποθήκευση των στοιχείων και η χρήση εργαλείων όπως DataTables και Chart.js για την παρουσίαση των δεδομένων με εύχρηστα και ωραία γραφήματα.

Η εργασία δομείται σε πέντε βασικά κεφάλαια, μέσα από τα οποία αναπτύσσεται τόσο το θεωρητικό υπόβαθρο όσο και η πρακτική υλοποίηση του συστήματος καταγραφής παρουσιών φοιτητών με χρήση ψηφιακών τεχνολογιών.

Στο Κεφάλαιο 2ο παρουσιάζονται τα συστήματα καταγραφής παρουσιών φοιτητών, με έμφαση στις σύγχρονες ψηφιακές τεχνολογίες που χρησιμοποιούνται για την αυτοματοποίηση και την αποτελεσματική παρακολούθηση της προσέλευσης των φοιτητών σε μαθήματα και εκπαιδευτικές δραστηριότητες. Γίνεται αναφορά τόσο στις παραδοσιακές μεθόδους όσο και σε εξελιγμένες λύσεις, όπως οι ηλεκτρονικές πλατφόρμες, τα συστήματα αναγνώρισης ταυτότητας και οι φορητές συσκευές.

Στο Κεφάλαιο 3ο αναλύονται οι τεχνολογίες που αξιοποιήθηκαν για την υλοποίηση του συγκεκριμένου έργου. Περιγράφονται τα εργαλεία και τα λογισμικά που χρησιμοποιήθηκαν, όπως η πλατφόρμα Laravel για την ανάπτυξη της εφαρμογής, η MySQL για τη διαχείριση της βάσης δεδομένων, καθώς και βιβλιοθήκες όπως DataTables και Chart.js για την παρουσίαση των δεδομένων.

Στο Κεφάλαιο 4ο παρουσιάζεται αναλυτικά το σύστημα καταγραφής και ανάλυσης παρουσιών φοιτητών που αναπτύχθηκε στο πλαίσιο της εργασίας. Περιγράφεται η λειτουργικότητα του συστήματος από την πλευρά του διαχειριστή, του καθηγητή και του φοιτητή, ενώ γίνεται εκτενής αναφορά στα επιμέρους υποσυστήματα, όπως η διαχείριση μαθημάτων, η δημιουργία ερωτήσεων, η καταγραφή απαντήσεων και η εξαγωγή στατιστικών στοιχείων. Το κεφάλαιο συνοδεύεται από

παραδείγματα, εικόνες και διαγράμματα που αποτυπώνουν με σαφήνεια την οργάνωση και τη λειτουργία του συστήματος.

Στο Κεφάλαιο 5ο παρατίθενται τα συμπεράσματα που προέκυψαν από την ανάπτυξη και υλοποίηση του συστήματος, καθώς και προτάσεις για μελλοντικές βελτιώσεις. Γίνεται αποτίμηση της αποτελεσματικότητας του συστήματος και επισημαίνονται τα σημεία που θα μπορούσαν να ενισχυθούν για να καλύψουν ακόμη περισσότερες ανάγκες της εκπαιδευτικής κοινότητας.

Η εργασία ολοκληρώνεται με τη Βιβλιογραφία, όπου καταγράφονται όλες οι πηγές που χρησιμοποιήθηκαν για την εκπόνηση της μελέτης, και το Παράρτημα, το οποίο περιλαμβάνει πρόσθετο υλικό όπως κομμάτια κώδικα.

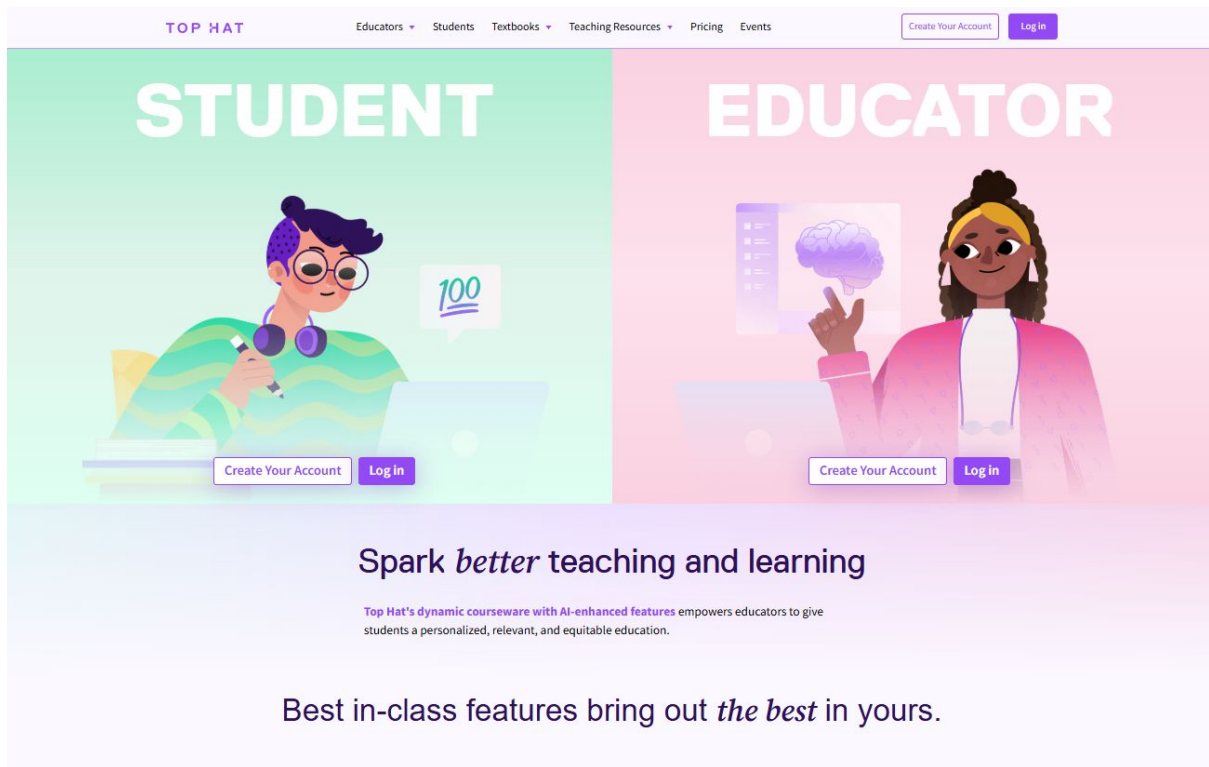
Κεφάλαιο 2ο: Συστήματα καταγραφής παρουσιών φοιτητών με χρήση ψηφιακών τεχνολογιών

2.1 Top Hat [1-2]

Το Top Hat αποτελεί μια ολοκληρωμένη πλατφόρμα εκπαιδευτικής τεχνολογίας που σχεδιάστηκε για να ενισχύσει την αλληλεπίδραση μεταξύ διδασκόντων και φοιτητών, παρέχοντας εργαλεία για την παρακολούθηση παρουσιών, την αξιολόγηση και την ενεργή συμμετοχή στην εκπαιδευτική διαδικασία.

Βασικά Χαρακτηριστικά

- **Καταγραφή Παρουσιών με Κωδικό:** Οι καθηγητές μπορούν να προβάλλουν έναν μοναδικό 4-ψήφιο κωδικό στην αρχή ή κατά τη διάρκεια του μαθήματος. Οι φοιτητές εισάγουν αυτόν τον κωδικό μέσω της εφαρμογής ή του προγράμματος περιήγησης για να δηλώσουν την παρουσία τους.
- **Secure Attendance (Ασφαλής Καταγραφή Παρουσιών):** Μια προαιρετική λειτουργία που χρησιμοποιεί γεωεντοπισμό και Bluetooth για να επιβεβαιώσει ότι οι φοιτητές βρίσκονται φυσικά στην αίθουσα κατά την καταγραφή παρουσίας.
- **Διαδραστικές Δραστηριότητες:** Προσφέρει δυνατότητες για ζωντανές δημοσκοπήσεις, κουίζ, συζητήσεις και αναθέσεις εργασιών για να έχουμε ενεργή συμμετοχή των φοιτητών.
- **Ενσωμάτωση με LMS:** Ενσωματώνεται με συστήματα διαχείρισης μάθησης όπως το Brightspace και επιτρέπει τον συγχρονισμό βαθμολογιών και παρουσιών.



Εικόνα 2.1: Tophat πλατφόρμα [<https://tophat.com/>]

Διαδικασία Καταγραφής Παρουσιών

1. Ενεργοποίηση Καταγραφής: Ο καθηγητής επιλέγει την επιλογή "Take Attendance" στην πλατφόρμα, προβάλλοντας έναν μοναδικό 4-ψήφιο κωδικό στην οθόνη.
2. Εισαγωγή Κωδικού από Φοιτητές: Οι φοιτητές εισάγουν τον κωδικό μέσω της εφαρμογής ή του προγράμματος περιήγησης για να δηλώσουν την παρουσία τους.
3. Καταγραφή και Αποθήκευση: Η παρουσία καταγράφεται αυτόματα στο σύστημα, επιτρέποντας στον καθηγητή να παρακολουθεί και να εξάγει τα δεδομένα.

Πλεονεκτήματα και Περιορισμοί

Πλεονεκτήματα:

- Ευκολία Χρήσης: Η διαδικασία καταγραφής παρουσιών είναι απλή και γρήγορη, τόσο για τους καθηγητές όσο και για τους φοιτητές.
- Ενίσχυση Συμμετοχής: Οι διαδραστικές δυνατότητες ενισχύουν την ενεργή συμμετοχή των φοιτητών κατά τη διάρκεια του μαθήματος.

- **Ενσωμάτωση με Υφιστάμενα Συστήματα:** Η δυνατότητα ενσωμάτωσης με LMS διευκολύνει τη διαχείριση των δεδομένων παρουσίας και βαθμολογιών.

Περιορισμοί:

- **Απαιτήσεις Τεχνολογίας:** Η χρήση του Secure Attendance απαιτεί ενεργοποίηση γεωεντοπισμού και Bluetooth, κάτι που μπορεί να προκαλέσει δυσκολίες σε ορισμένους φοιτητές.
- **Πιθανότητα Κατάχρησης:** Η βασική λειτουργία καταγραφής με κωδικό μπορεί να είναι ευάλωτη σε καταχρήσεις, όπως η κοινοποίηση του κωδικού σε απόντες φοιτητές.

2.2 QuickSchools [3]

Το QuickSchools είναι ένα διαδικτυακό σύστημα διαχείρισης σχολικών πληροφοριών που στοχεύει στην απλοποίηση των διοικητικών διαδικασιών των εκπαιδευτικών ιδρυμάτων. Μέσω μιας φιλικής προς τον χρήστη πλατφόρμας επιτρέπει σε εκπαιδευτικούς, διαχειριστές και γονείς να παρακολουθούν και να διαχειρίζονται δεδομένα που σχετίζονται με την παρουσία, τις βαθμολογίες, τα προγράμματα και άλλες σημαντικές πληροφορίες.

Η καταγραφή παρουσιών στο QuickSchools είναι ιδιαίτερα απλή. Οι εκπαιδευτικοί μπορούν να σημειώνουν την παρουσία των μαθητών με ένα μόνο κλικ, είτε σε καθημερινή βάση, είτε ανά περίοδο ή μάθημα. Το σύστημα υποστηρίζει έγχρωμα γραφήματα για εύκολη αναγνώριση των παρόντων, απόντων και καθυστερημένων μαθητών. Αυτή η διαδικασία μειώνει τα λάθη και εξοικονομεί χρόνο για τους εκπαιδευτικούς. Επιπλέον, οι γονείς έχουν πρόσβαση σε πληροφορίες σχετικά με την παρουσία, τις βαθμολογίες και τις εργασίες των παιδιών τους μέσω μιας ειδικής πύλης.

Το QuickSchools προσφέρει επίσης μια εφαρμογή για την καταγραφή των ωρών εισόδου και εξόδου των μαθητών και των γονέων, βελτιώνοντας την ασφάλεια και την ακρίβεια των δεδομένων παρουσίας. Αυτή η εφαρμογή επιτρέπει την αυτόματη καταγραφή των ωρών εισόδου και εξόδου ενώ παρέχει τη δυνατότητα προσθήκης σχολίων και διαχείρισης των δεδομένων μέσω ενός εύχρηστου περιβάλλοντος.

Το σύστημα παρέχει αναλυτικές αναφορές για την παρακολούθηση της παρουσίας και επιτρέπει στους διαχειριστές να εντοπίζουν τάσεις και να λαμβάνουν ενημερωμένες αποφάσεις. Αυτές οι αναφορές βοηθούν στην κατανόηση των προτύπων απουσιών και καθυστερήσεων διευκολύνοντας την έγκαιρη παρέμβαση όταν χρειάζεται.

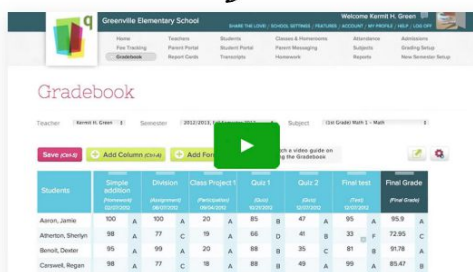
Το QuickSchools αποτελεί μια ολοκληρωμένη λύση για την ψηφιακή διαχείριση των σχολικών διαδικασιών και δίνει ευκολία στη χρήση, προσβασιμότητα από οποιαδήποτε συσκευή με σύνδεση στο διαδίκτυο και ενσωμάτωση με άλλα εργαλεία και πλατφόρμες όπως το Google Calendar και το Canvas.

Online School Management Software

- Customizable gradebook software
- Homework and assignments management
- Powerful Student Information System (SIS)
- Classroom and attendance management
- Parents portal to track child's progress
- Admissions management platform

SIGN UP FOR FREE

Watch 90-second demo



A SCHOOL MANAGEMENT SYSTEM THAT IS SHOCKINGLY EASY TO USE

Εικόνα 2.2: QuickSchools πλατφόρμα

[<https://www.quickschools.com/>]

2.3 MyAttendanceTracker [4]

Το MyAttendanceTracker είναι μια δωρεάν διαδικτυακή εφαρμογή που απλοποιεί την καταγραφή παρουσιών και την παρακολούθηση της προόδου των μαθητών για εκπαιδευτικούς και οργανωτές ομάδων. Δημιουργήθηκε από τον Trent O., προγραμματιστή. Η εφαρμογή έχει εξελιχθεί και χρησιμοποιείται από πάνω από 50.000 χρήστες για την παρακολούθηση περισσότερων από 650.000 μαθητών σε διάφορα εκπαιδευτικά και μη περιβάλλοντα.

Η πλατφόρμα επιτρέπει την εύκολη καταγραφή παρουσιών μέσω οποιασδήποτε συσκευής με σύνδεση στο διαδίκτυο, προσφέροντας επίσης δυνατότητες όπως ενσωματωμένο βαθμολόγιο, αποστολή μηνυμάτων σε γονείς και προσαρμοσμένες αναφορές. Οι χρήστες μπορούν να δημιουργήσουν απεριόριστες τάξεις, να εισάγουν ή να εξάγουν δεδομένα και να παρακολουθούν την πρόοδο των μαθητών μέσω διαδραστικών πινάκων ελέγχου.



Attendance Tracking Online: [MyAttendanceTracker](#)

Have you ever wanted a simple and convenient way to track attendance for your classroom or group? Well, now there's a way. MyAttendanceTracker.com is an online attendance tracker that helps you simplify the tedious task of keeping attendance.

[SIGN UP FOR FREE!](#)

Εικόνα 2.3: MyAttendanceTracker πλατφόρμα


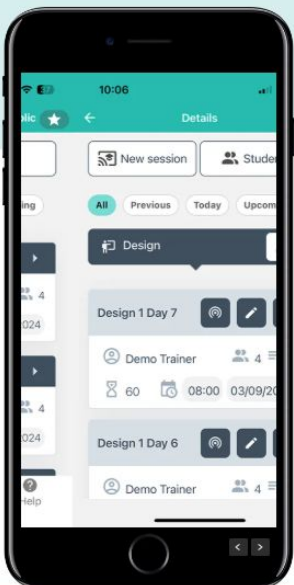
[<https://www.myattendancetracker.com/>]

Το MyAttendanceTracker προσφέρει μια φιλική προς τον χρήστη διεπαφή και το κάνει το ιδανικό για εκπαιδευτικούς που επιθυμούν μια απλή αλλά αποτελεσματική λύση για την παρακολούθηση της παρουσίας και της απόδοσης των μαθητών τους. Η εφαρμογή είναι προσβάσιμη από κινητές συσκευές, επιτρέποντας στους χρήστες να καταγράφουν δεδομένα εν κινήσει.

2.4 Attendance Radar [5]

Το Attendance Radar είναι μια σύγχρονη εφαρμογή που απλοποιεί και αυτοματοποιεί τη διαδικασία καταγραφής παρουσιών σε εκπαιδευτικά ιδρύματα. Χρησιμοποιώντας τεχνολογία Bluetooth επιτρέπει στους καθηγητές να καταγράφουν την παρουσία των φοιτητών με ταχύτητα και ακρίβεια και εξαλείφει την ανάγκη για παραδοσιακές μεθόδους όπως είναι οι χειρόγραφες λίστες ή οι κωδικοί QR.

The easiest, fastest and most reliable way to track student attendance.

- Fast**
Mark the attendance of a whole class in seconds with our Bluetooth beacon.
- Reliable**
Ensure only the attendance of those students present is marked.
- Evolving**
Receive frequent updates with exciting new features.
- Easy**
Track student attendance in just a couple of clicks.
- Free**
Download Attendance Radar for free both on Google Play and the App Store.
- Versatile**
Take student attendance in tutorials, lectures, seminars or any other event.

Εικόνα 2.4: Attendance Radar πλατφόρμα

[<https://attendanceradar.com/>]

Η λειτουργία της εφαρμογής είναι απλή: ο καθηγητής ενεργοποιεί ένα ασφαλές σήμα Bluetooth από την εφαρμογή του το οποίο ανιχνεύεται από τις συσκευές των φοιτητών που βρίσκονται στην αίθουσα. Οι φοιτητές με ένα μόνο πάτημα στην εφαρμογή τους και επιβεβαιώνουν την παρουσία τους. Αυτή η μέθοδος διασφαλίζει ότι μόνο οι παρόντες φοιτητές μπορούν να καταγράψουν την παρουσία τους, μειώνοντας τις πιθανότητες απάτης.

Η εφαρμογή προσφέρει επίσης δυνατότητες όπως η εισαγωγή και εξαγωγή δεδομένων παρουσίας μέσω αρχείων Excel, η δημιουργία επαναλαμβανόμενων συνεδριών, και η καταγραφή σημειώσεων για τη συμμετοχή και την απόδοση των φοιτητών. Παρέχει στατιστικά στοιχεία για την παρακολούθηση της παρουσίας και επιτρέπει στους διαχειριστές να εντοπίζουν τάσεις και να λαμβάνουν ενημερωμένες αποφάσεις.

Το Attendance Radar διατίθεται σε διάφορες εκδόσεις με μια δωρεάν έκδοση με βασικές λειτουργίες, μιας premium έκδοσης με πρόσθετες δυνατότητες, και μιας έκδοσης για πανεπιστήμια που προσφέρει επιπλέον δυνατότητες διαχείρισης και ενσωμάτωσης με πανεπιστημιακά συστήματα. Η εφαρμογή έχει

υιοθετηθεί από πολλά εκπαιδευτικά ιδρύματα παγκοσμίως όπως το Geneva Business School, το KU Leuven, το University of Ghent, το Free University Brussels και το University of Antwerp.

2.5 ScholarChip [6]

Το ScholarChip είναι ένα ολοκληρωμένο σύστημα διαχείρισης ταυτοτήτων και παρακολούθησης παρουσιών, σχεδιασμένο για να ενισχύει την ασφάλεια και την αποτελεσματικότητα των σχολικών ιδρυμάτων. Ιδρύθηκε το 2000 από τον Dr. Maged Atiya και εδρεύει στη Νέα Υόρκη.

Η βασική λειτουργία του ScholarChip βασίζεται στη χρήση έξυπνων καρτών με τεχνολογία NFC, οι οποίες επιτρέπουν την ανέπαφη καταγραφή παρουσιών. Οι μαθητές και το προσωπικό μπορούν να "χτυπούν" τις κάρτες τους σε ειδικά τερματικά κατά την είσοδο ή έξοδο από τις αίθουσες ή το σχολείο, καταγράφοντας έτσι την παρουσία τους σε πραγματικό χρόνο. Αυτή η διαδικασία αυτοματοποιεί την παρακολούθηση παρουσιών και μειώνει την ανάγκη για χειροκίνητη καταγραφή.

Το ScholarChip προσφέρει δυνατότητες διαχείρισης πρόσβασης σε διάφορους χώρους του σχολείου, όπως βιβλιοθήκες, εργαστήρια και καφετέριες, μέσω των ίδιων έξυπνων καρτών. Αυτό ενισχύει την ασφάλεια αφού μόνο εξουσιοδοτημένα άτομα μπορούν να εισέλθουν σε συγκεκριμένους χώρους.

Το σύστημα υποστηρίζει την ενσωμάτωση με υπάρχοντα Συστήματα Διαχείρισης Μαθητών (SIS) και επιτρέπει την εύκολη ανταλλαγή δεδομένων και τη δημιουργία αναφορών για την παρακολούθηση παρουσιών και άλλων σχετικών πληροφοριών. Αυτές οι αναφορές μπορούν να χρησιμοποιηθούν για την ανάλυση τάσεων απουσιών και τη λήψη αποφάσεων για την αντιμετώπιση προβλημάτων όπως η χρόνια απουσία.

Το ScholarChip έχει υιοθετηθεί από διάφορα σχολικά συστήματα στις Ηνωμένες Πολιτείες, συμπεριλαμβανομένων των δημόσιων σχολείων της Φιλαδέλφειας και της Βαλτιμόρης, για τη βελτίωση της ασφάλειας και της διαχείρισης παρουσιών. Η χρήση του έχει συμβάλει στη μείωση της απουσίας και στην ενίσχυση της συνολικής ασφάλειας των σχολικών εγκαταστάσεων.

Κεφάλαιο 3ο: Χρήση τεχνολογιών για το έργο

3.1 Laravel

Το Laravel είναι ένα ισχυρό και δημοφιλές framework για την ανάπτυξη διαδικτυακών εφαρμογών, ιδιαίτερα κατάλληλο για έργα που απαιτούν ευελιξία, ασφάλεια και ταχύτητα ανάπτυξης. Στην εργασία μας το Laravel αποτέλεσε τον βασικό κορμό για την υλοποίηση του συστήματος καταγραφής και ανάλυσης παρουσιών φοιτητών και έχει ένα σταθερό και επεκτάσιμο περιβάλλον ανάπτυξης. [7-10]

Το Laravel είναι ένα ανοιχτού κώδικα framework για την ανάπτυξη εφαρμογών ιστού, βασισμένο στη γλώσσα προγραμματισμού PHP. Ακολουθεί το πρότυπο αρχιτεκτονικής MVC (Model-View-Controller) το οποίο διαχωρίζει τη λογική της εφαρμογής από την παρουσίαση και μας δίνει συντήρηση και επεκτασιμότητα του κώδικα. Η φιλοσοφία του Laravel επικεντρώνεται στην απλότητα, την κομψότητα και την αναγνωσιμότητα του κώδικα και είναι το ιδανικό για προγραμματιστές που επιθυμούν να δημιουργήσουν ισχυρές και ασφαλείς εφαρμογές με λιγότερο κόπο.

Το Laravel προσφέρει ένα ευρύ φάσμα εργαλείων και λειτουργιών που καθιστούν την ανάπτυξη εφαρμογών πιο αποδοτική και οργανωμένη. Με το ενσωματωμένο σύστημα δρομολόγησης, τη διαχείριση βάσεων δεδομένων μέσω του Eloquent ORM και το Blade templating engine για την εύκολη δημιουργία προβολών, το Laravel επιτρέπει στους προγραμματιστές να δημιουργούν εφαρμογές με καθαρό και συντηρήσιμο κώδικα. Το Artisan CLI έχει μια σειρά από εντολές που αυτοματοποιούν κοινές εργασίες ανάπτυξης όπως η δημιουργία controllers, models και migrations.



Εικόνα 3.1: Πλεονεκτήματα Laravel

Ένα από τα κύρια πλεονεκτήματα του Laravel είναι η ευκολία χρήσης και η καθαρή σύνταξη του κώδικα, που επιτρέπει την ταχεία ανάπτυξη εφαρμογών. Η εκτενής τεκμηρίωση και η ενεργή κοινότητα υποστήριξης κάνουν την εκμάθηση και την επίλυση προβλημάτων πιο προσιτή. Το Laravel παρέχει ενσωματωμένες λειτουργίες ασφαλείας, όπως η προστασία από επιθέσεις CSRF και XSS και μηχανισμούς αυθεντικοποίησης και εξουσιοδότησης χρηστών.

Το Laravel έχει και ορισμένα μειονεκτήματα όπως είναι η απόδοση που μπορεί να είναι χαμηλότερη σε σύγκριση με ελαφρύτερα frameworks ιδιαίτερα σε εφαρμογές που απαιτούν υψηλή ταχύτητα απόκρισης. Η καμπύλη εκμάθησης μπορεί να είναι απότομη για αρχάριους προγραμματιστές λόγω της πληθώρας των διαθέσιμων λειτουργιών και της πολυπλοκότητας ορισμένων εννοιών.

Το Laravel συνεχίζει να κυριαρχεί στον χώρο των PHP frameworks το 2024 και έχει ηγετική θέση στην ανάπτυξη διαδικτυακών εφαρμογών. Η δημοτικότητά του οφείλεται στην ευκολία χρήσης την εκτενή τεκμηρίωση και την ενεργή κοινότητα υποστήριξης.

Σύμφωνα με την ετήσια έρευνα του JetBrains για το 2024 το Laravel παραμένει το πιο δημοφιλές PHP framework με σημαντικό αριθμό προγραμματιστών να το χρησιμοποιούν τακτικά στα έργα τους.

Η έρευνα "State of Laravel 2024" αποκάλυψε ότι το Laravel χρησιμοποιείται κυρίως για επιχειρηματικές εφαρμογές (95,4%) και χόμπι projects (57,9%) και έτσι φαίνεται η ευελιξία του σε διάφορα είδη έργων. [11]

Στην εργασία μας το Laravel επιλέχθηκε ως το βασικό framework για την ανάπτυξη του συστήματος καταγραφής και ανάλυσης παρουσιών φοιτητών. Η επιλογή αυτή βασίστηκε στην ανάγκη για ένα σταθερό, ασφαλές και επεκτάσιμο περιβάλλον ανάπτυξης και να υποστηρίζει την γρήγορη υλοποίηση και την εύκολη συντήρηση της εφαρμογής.

Η χρήση του Laravel επέτρεψε την οργάνωση του κώδικα σύμφωνα με το πρότυπο MVC ξεχωρίζοντας τη λογική της εφαρμογής από την παρουσίαση. Το Eloquent ORM διευκόλυνε τη διαχείριση των δεδομένων παρουσιών, την εύκολη δημιουργία, ανάγνωση, ενημέρωση και διαγραφή εγγραφών στη βάση δεδομένων. Το Blade templating engine χρησιμοποιήθηκε για τη δημιουργία δυναμικών και επαναχρησιμοποιήσιμων προβολών και το Artisan CLI επιτάχυνε τη διαδικασία ανάπτυξης μέσω αυτοματοποιημένων εντολών.

Το Laravel προσέφερε ενσωματωμένες λειτουργίες αυθεντικοποίησης και εξουσιοδότησης χρηστών, διασφαλίζοντας ότι μόνο εξουσιοδοτημένοι χρήστες μπορούν να έχουν πρόσβαση σε συγκεκριμένες λειτουργίες της εφαρμογής. Η δυνατότητα δημιουργίας και διαχείρισης migrations διευκόλυνε την παρακολούθηση των αλλαγών στη δομή της βάσης δεδομένων και η υποστήριξη για unit testing έδωσε τη διασφάλιση της ποιότητας του κώδικα.

Η επιλογή του Laravel για την ανάπτυξη του συστήματος καταγραφής και ανάλυσης παρουσιών φοιτητών αποδείχθηκε επιτυχής, προσφέροντας ένα ισχυρό και ευέλικτο περιβάλλον που μας κάλυψε πλήρως στις απαιτήσεις του έργου.

3.2 MySQL

Το MySQL είναι ένα από τα πιο διαδεδομένα συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS), ευρέως χρησιμοποιούμενο σε εφαρμογές ιστού και επιχειρηματικά περιβάλλοντα. [12-15]

Το MySQL είναι ένα ανοικτού κώδικα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων, το οποίο χρησιμοποιεί τη γλώσσα SQL (Structured Query Language) για την πρόσβαση και διαχείριση των δεδομένων. Λόγω της ευκολίας χρήσης, της αξιοπιστίας και της ευελιξίας του, έχει καθιερωθεί ως βασικό εργαλείο για την ανάπτυξη εφαρμογών που απαιτούν αποθήκευση και ανάκτηση δεδομένων.

Το MySQL δημιουργήθηκε το 1995 από τους Σουηδούς David Axmark και Allan Larsson, μαζί με τον Φινλανδό Michael "Monty" Widenius. Αρχικά αναπτύχθηκε ως εναλλακτική λύση στο mSQL, με στόχο την παροχή ενός πιο ευέλικτου και αποδοτικού συστήματος διαχείρισης βάσεων δεδομένων. Το 2008, η MySQL AB εξαγοράστηκε από τη Sun Microsystems, η οποία με τη σειρά της εξαγοράστηκε από την Oracle Corporation το 2010. Σήμερα το MySQL συνεχίζει να αναπτύσσεται υπό την αιγίδα της Oracle, με τακτικές ενημερώσεις και βελτιώσεις.

Το MySQL χρησιμοποιείται ευρέως για την αποθήκευση, διαχείριση και ανάκτηση δεδομένων σε διάφορες εφαρμογές από απλές ιστοσελίδες έως πολύπλοκα επιχειρηματικά συστήματα. Η υποστήριξή του για πολλαπλές πλατφόρμες και η συμβατότητά του με διάφορες γλώσσες προγραμματισμού το καθιστούν ιδανικό για ανάπτυξη εφαρμογών που απαιτούν αξιόπιστη και αποδοτική διαχείριση δεδομένων.

Πλεονεκτήματα:

- **Ανοικτός Κώδικας:** Το MySQL είναι διαθέσιμο δωρεάν υπό την άδεια GPL, επιτρέποντας την ευρεία υιοθέτηση και προσαρμογή του.
- **Υψηλή Απόδοση:** Χάρη στην αρχιτεκτονική του και την υποστήριξη για διάφορες μηχανές αποθήκευσης, όπως το InnoDB, προσφέρει υψηλή απόδοση σε ποικίλες εφαρμογές.
- **Ευκολία Χρήσης:** Η απλή σύνταξη της SQL και τα διαθέσιμα εργαλεία διαχείρισης καθιστούν το MySQL προσβάσιμο ακόμα και σε αρχάριους χρήστες.
- **Ευρεία Υποστήριξη:** Διαθέτει μεγάλη κοινότητα χρηστών και προγραμματιστών, προσφέροντας εκτενή τεκμηρίωση και υποστήριξη.

Μειονεκτήματα:

- **Περιορισμένη Υποστήριξη για Προηγμένα Χαρακτηριστικά:** Σε σύγκριση με άλλες εμπορικές βάσεις δεδομένων, το MySQL μπορεί να υστερεί σε προηγμένα χαρακτηριστικά, όπως πλήρης υποστήριξη για αποθηκευμένες διαδικασίες και triggers.
- **Περιορισμένη Κλιμάκωση:** Αν και κατάλληλο για πολλές εφαρμογές, μπορεί να αντιμετωπίσει προκλήσεις σε πολύ μεγάλες και απαιτητικές βάσεις δεδομένων.
- **Διαχείριση Αντιγράφων Ασφαλείας:** Η διαδικασία δημιουργίας και αποκατάστασης αντιγράφων ασφαλείας μπορεί να είναι πιο περίπλοκη σε σύγκριση με άλλα συστήματα.

Στην εργασία το MySQL επιλέχθηκε ως το βασικό σύστημα διαχείρισης βάσεων δεδομένων για την αποθήκευση και διαχείριση των δεδομένων παρουσιών φοιτητών. Η επιλογή αυτή βασίστηκε στην αξιοπιστία, την απόδοση και την ευκολία ενσωμάτωσης του MySQL με το Laravel framework το οποίο χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής.

Χρησιμοποιήθηκε για την αποτελεσματική διαχείριση των δεδομένων παρουσιών, την εκτέλεση πολύπλοκων ερωτημάτων για την ανάλυση των δεδομένων και την εξαγωγή χρήσιμων στατιστικών. Η υποστήριξή του για συναλλαγές και η δυνατότητα δημιουργίας αποθηκευμένων διαδικασιών συνέβαλαν στη διασφάλιση της ακεραιότητας των δεδομένων και στην αυτοματοποίηση επαναλαμβανόμενων εργασιών.

Η σύνδεση του Laravel με τη βάση δεδομένων MySQL αποτέλεσε κρίσιμο στοιχείο για την υλοποίηση του συστήματος καταγραφής και ανάλυσης παρουσιών φοιτητών. Ο συνδυασμός αυτών των δύο τεχνολογιών προσέφερε ένα σταθερό, ευέλικτο και αποδοτικό περιβάλλον ανάπτυξης για την αποτελεσματική διαχείριση των δεδομένων και την απρόσκοπτη λειτουργία της εφαρμογής.

Το Laravel χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής καταγραφής παρουσιών και το MySQL αποτέλεσε το σύστημα διαχείρισης της βάσης δεδομένων. Η χρήση του Eloquent ORM του Laravel επέτρεψε την εύκολη διαχείριση των δεδομένων με τη δημιουργία μοντέλων που αντιστοιχούν στους πίνακες της βάσης. Οι λειτουργίες όπως οι μεταναστεύσεις (migrations) και οι σπορές (seeders) διευκόλυναν τη δημιουργία και τον αρχικό πληθυσμό της βάσης δεδομένων.

3.3 Chart.js

Η οπτικοποίηση των δεδομένων αποτελεί σημαντικό μέρος κάθε σύγχρονης εφαρμογής και βοηθά στην κατανόηση πολύπλοκων πληροφοριών μέσω γραφικών αναπαραστάσεων. Ένα από τα πιο δημοφιλή εργαλεία για την υλοποίηση αυτής της ανάγκης είναι το Chart.js. [16-17]

Το Chart.js είναι μια ελαφριά και εύχρηστη JavaScript βιβλιοθήκη για τη δημιουργία διαδραστικών γραφημάτων. Χρησιμοποιεί το στοιχείο HTML5 Canvas για την απεικόνιση δεδομένων με υψηλή απόδοση και ευχάριστη αισθητική.

Υποστηριζόμενοι Τύποι Διαγραμμάτων:

- Γραμμικά (Line Charts)
- Ραβδογράμματα (Bar Charts)
- Κυκλικά (Pie Charts)
- Δακτυλοειδή (Doughnut Charts)
- Ραντάρ (Radar Charts)
- Πολικών Περιοχών (Polar Area Charts)

Το Chart.js χρησιμοποιήθηκε για την οπτικοποίηση των δεδομένων παρουσιών φοιτητών. Η βιβλιοθήκη επέτρεψε την εύκολη δημιουργία γραφημάτων που απεικονίζουν τις παρουσίες ανά φοιτητή, μάθημα ή χρονική περίοδο. Η ενσωμάτωση με το Laravel και το MySQL ήταν απλή για την απευθείας άντληση δεδομένων από τη βάση και την απόδοσή τους σε γραφήματα. Η χρήση του Chart.js βελτίωσε σημαντικά την κατανόηση των δεδομένων και συνέβαλε στη λήψη τεκμηριωμένων αποφάσεων βάσει των παρουσιών.

3.4 Datatables

Το DataTables είναι μια ισχυρή βιβλιοθήκη JavaScript που ενισχύει τα HTML tables με προηγμένες δυνατότητες αλληλεπίδρασης όπως ταξινόμηση, αναζήτηση, σελιδοποίηση και πολλά άλλα. Είναι ιδιαίτερα χρήσιμη για την παρουσίαση μεγάλων συνόλων δεδομένων με τρόπο φιλικό προς τον χρήστη. [18-19]

Το DataTables είναι ένα πρόσθετο (plugin) για τη βιβλιοθήκη jQuery το οποίο προσθέτει λειτουργίες όπως ταξινόμηση, αναζήτηση και σελιδοποίηση σε απλά HTML tables. Αναπτύχθηκε με στόχο να παρέχει ένα εύχρηστο εργαλείο για την ενίσχυση των δυνατοτήτων παρουσίασης δεδομένων σε πίνακες.

Η βιβλιοθήκη DataTables δημιουργήθηκε από τον Allan Jardine και έχει εξελιχθεί με τη συμβολή μιας ενεργής κοινότητας προγραμματιστών. Από την αρχική της έκδοση έχει υποστεί σημαντικές βελτιώσεις και επεκτάσεις και έχει γίνει ένα από τα πιο δημοφιλή εργαλεία για την ενίσχυση των HTML tables.

Το DataTables επιτρέπει στους προγραμματιστές να μετατρέπουν απλά HTML tables σε δυναμικά και διαδραστικά στοιχεία. Είναι ιδιαίτερα χρήσιμο σε εφαρμογές που απαιτούν την παρουσίαση μεγάλων ποσοτήτων δεδομένων με δυνατότητες όπως:

- Ταξινόμηση δεδομένων κατά στήλη.
- Αναζήτηση και φιλτράρισμα δεδομένων.
- Σελιδοποίηση για τη διαχείριση μεγάλων συνόλων δεδομένων.
- Υποστήριξη για responsive σχεδιασμό, προσαρμόζοντας τον πίνακα σε διάφορες συσκευές.

Πλεονεκτήματα:

- Εύκολη ενσωμάτωση με υπάρχοντα HTML tables.
- Πλούσια τεκμηρίωση και ενεργή κοινότητα υποστήριξης.
- Υποστήριξη για διάφορες επεκτάσεις και plugins που προσθέτουν επιπλέον λειτουργίες.
- Συμβατότητα με διάφορα frameworks και βιβλιοθήκες.

Μειονεκτήματα:

- Εξάρτηση από τη βιβλιοθήκη jQuery, η οποία μπορεί να μην είναι επιθυμητή σε σύγχρονες εφαρμογές.
- Η απόδοση μπορεί να επηρεαστεί όταν διαχειρίζεται πολύ μεγάλα σύνολα δεδομένων χωρίς τη χρήση server-side processing.

Το DataTables χρησιμοποιήθηκε για την παρουσίαση των δεδομένων παρουσιών φοιτητών με τρόπο διαδραστικό και φιλικό προς τον χρήστη. Η ενσωμάτωσή του επέτρεψε τη γρήγορη αναζήτηση και φιλτράρισμα των δεδομένων και τη σελιδοποίηση για την καλύτερη διαχείριση μεγάλων συνόλων πληροφοριών. Η χρήση του DataTables συνέβαλε σημαντικά στη βελτίωση της εμπειρίας χρήστη και στην αποτελεσματική παρουσίαση των δεδομένων.

3.5 Bootstrap

Το Bootstrap είναι ένα από τα πιο διαδεδομένα και δημοφιλή front-end frameworks που χρησιμοποιούνται για την ανάπτυξη ιστοσελίδων και εφαρμογών ιστού. Προσφέρει ένα σύνολο εργαλείων για την εύκολη και γρήγορη δημιουργία ελκυστικών, λειτουργικών και πλήρως responsive ιστοσελίδων. Δημιουργήθηκε αρχικά από την ομάδα του Twitter το 2011, με σκοπό την απλοποίηση της διαδικασίας ανάπτυξης του front-end και είναι ένα συνεκτικό και ομοιόμορφο περιβάλλον εργασίας για τους προγραμματιστές. [20-21]

Το Bootstrap περιλαμβάνει έτοιμα CSS και JavaScript components που καλύπτουν σχεδόν όλες τις ανάγκες ενός σύγχρονου ιστότοπου, από απλά κουμπιά και φόρμες μέχρι πολύπλοκα εργαλεία πλοήγησης, modals και sliders. Το μεγαλύτερο πλεονέκτημά του είναι ότι επιτρέπει τη δημιουργία responsive ιστοσελίδων με ελάχιστη προσπάθεια και εξασφαλίζει ότι το περιεχόμενο προσαρμόζεται αυτόματα σε όλες τις συσκευές όπως υπολογιστές, tablets και κινητά τηλέφωνα.

Το Bootstrap αναπτύχθηκε αρχικά ως ένα εσωτερικό εργαλείο της Twitter για τη διαχείριση των διεπαφών χρήστη. Σύντομα διατέθηκε ως open-source πλατφόρμα γνωρίζοντας τεράστια απήχηση στη διεθνή κοινότητα των προγραμματιστών. Από τότε το Bootstrap έχει περάσει από πολλές σημαντικές εκδόσεις, με την πιο πρόσφατη να υποστηρίζει σύγχρονες τεχνολογίες CSS όπως Flexbox και Grid, ενώ απομακρύνεται από την εξάρτηση σε παλαιότερες βιβλιοθήκες όπως το jQuery.

Η χρήση του Bootstrap επιταχύνει σημαντικά την ανάπτυξη ιστοσελίδων και παρέχει ένα οργανωμένο και καλοσχεδιασμένο σύνολο προκαθορισμένων στοιχείων διεπαφής. Με τη βοήθεια του Bootstrap, ακόμα και αρχάριοι προγραμματιστές μπορούν να δημιουργήσουν μοντέρνες και καλοσχεδιασμένες ιστοσελίδες χωρίς να χρειάζεται να γράψουν πολύπλοκο CSS ή JavaScript. Δίνει τη δυνατότητα εύκολης παραμετροποίησης ώστε οι ιστοσελίδες να αποκτούν μοναδικό χαρακτήρα και να προσαρμόζονται στις ανάγκες του εκάστοτε έργου.

Το μεγαλύτερο πλεονέκτημα του Bootstrap είναι η ταχύτητα ανάπτυξης. Παρέχει έτοιμα και πλήρως δοκιμασμένα components και προσέχει ώστε η σελίδα να είναι συμβατή με όλους τους σύγχρονους browsers. Η υποστήριξη responsive design είναι επίσης ενσωματωμένη χωρίς να απαιτείται επιπλέον προσπάθεια από τον προγραμματιστή. Το framework διαθέτει εκτενή τεκμηρίωση, που το καθιστά εξαιρετικά φιλικό ακόμα και σε νέους χρήστες.

Κάποιο από τα μειονεκτήματα είναι ότι οι ιστοσελίδες που βασίζονται εξ ολοκλήρου στο Bootstrap συχνά παρουσιάζουν παρόμοια εμφάνισ, αν δεν γίνουν οι απαραίτητες προσαρμογές. Αν και η βιβλιοθήκη παρέχει πληθώρα έτοιμων κλάσεων, η κατάχρησή τους μπορεί να οδηγήσει σε δυσανάγνωστο και δύσκολο διαχειρίσιμο κώδικα. Το μέγεθος των αρχείων του Bootstrap μπορεί να

είναι μεγάλο και επηρεάζει την απόδοση της ιστοσελίδας εκτός αν γίνει χρήση μόνο των απαραίτητων modules.

Το Bootstrap χρησιμοποιήθηκε για την ανάπτυξη του γραφικού περιβάλλοντος της εφαρμογής καταγραφής και ανάλυσης παρουσιών φοιτητών. Η χρήση του επέτρεψε την ταχεία δημιουργία ενός αισθητικά ευχάριστου και πλήρως λειτουργικού interface, χωρίς την ανάγκη εκτενούς συγγραφής CSS. Εξασφαλίστηκε η σωστή απεικόνιση του περιεχομένου τόσο σε υπολογιστές όσο και σε φορητές συσκευές, χάρη στο responsive design που προσφέρει το framework.

Το Bootstrap συνέβαλε τη διασφάλιση της ομοιόμορφης εμφάνισης όλων των σελίδων της εφαρμογής, καθώς και στην εύκολη ενσωμάτωση διαδραστικών στοιχείων, όπως modals, dropdowns και alert μηνυμάτων. Μέσω της απλής παραμετροποίησης των προκαθορισμένων κλάσεων, επιτεύχθηκε ένας καθαρός και σύγχρονος σχεδιασμός, βελτιώνοντας σημαντικά την εμπειρία χρήστη.

Η επιλογή του Bootstrap για την εργασία μας αποδείχθηκε ιδανική και επέτρεψε την γρήγορη ανάπτυξη ενός καλοσχεδιασμένου και λειτουργικού συστήματος χωρίς να απαιτείται η συγγραφή πολύπλοκου CSS ή JavaScript κώδικα. Παρόλο που το Bootstrap έχει ορισμένους περιορισμούς, τα πλεονεκτήματα που έχει όπως η ταχύτητα ανάπτυξης και η υποστήριξη responsive design το κάνουν εξαιρετικά χρήσιμο για την ανάπτυξη μοντέρνων διαδικτυακών εφαρμογών.

Κεφάλαιο 4ο: Το σύστημα καταγραφής και ανάλυσης παρουσιών φοιτητών με χρήση ψηφιακών τεχνολογιών

4.1 Μια μικρή περιγραφή

Το σύστημα παρέχει έναν σύγχρονο και αξιόπιστο τρόπο καταγραφής παρουσιών φοιτητών κατά τη διάρκεια των μαθημάτων, συνδυάζοντας ταυτόχρονα αξιολόγηση της συμμετοχής μέσω απαντήσεων σε ερωτήσεις πολλαπλής επιλογής.

Ο διαχειριστής (Admin) έχει τη δυνατότητα να δημιουργεί και να διαχειρίζεται τα μαθήματα, καθώς και να αναθέτει καθηγητές σε αυτά. Οι καθηγητές, μέσω του δικού τους πίνακα ελέγχου, μπορούν να δημιουργούν ερωτήσεις πολλαπλής επιλογής για τα μαθήματά τους, να ορίζουν το χρονικό διάστημα ισχύος τους και να παράγουν έναν μοναδικό κωδικό για κάθε ερώτηση. Ο κωδικός αυτός ισχύει για περιορισμένο χρονικό διάστημα και χρησιμοποιείται από τους φοιτητές για να δηλώσουν την παρουσία τους.

Οι φοιτητές, εισάγοντας τον κωδικό σε ειδική σελίδα του συστήματος, καλούνται να συμπληρώσουν τα προσωπικά τους στοιχεία (Αριθμό Μητρώου, Επώνυμο, Email) και να απαντήσουν στην ερώτηση. Το σύστημα καταγράφει την IP του χρήστη για λόγους ασφαλείας και διασφάλισης μοναδικότητας συμμετοχής.

Οι καθηγητές μπορούν να παρακολουθούν σε πραγματικό χρόνο τις απαντήσεις των φοιτητών, να αξιολογούν την ορθότητα των απαντήσεων και να πραγματοποιούν στατιστική ανάλυση μέσω δυναμικών διαγραμμάτων και εξαγωγής δεδομένων σε Excel. Οι απαντήσεις εμφανίζονται με οπτική ένδειξη ορθότητας, παρέχοντας άμεση εικόνα της κατανόησης του μαθήματος από τους φοιτητές.

Το σύστημα ενισχύει τη διαδραστικότητα της εκπαιδευτικής διαδικασίας και παρέχει ένα αδιάβλητο και αντικειμενικό μέσο παρακολούθησης της συμμετοχής και της κατανόησης του διδακτικού αντικειμένου.

4.2 Η βάση δεδομένων του συστήματος μας

Η βάση δεδομένων αποτελεί την καρδιά του συστήματος καταγραφής παρουσιών μέσω ερωτήσεων πολλαπλής επιλογής. Είναι υπεύθυνη για την ασφαλή αποθήκευση των χρηστών, των μαθημάτων, των ερωτήσεων που δημιουργούνται από τους καθηγητές, καθώς και των απαντήσεων που υποβάλλονται

από τους φοιτητές. Η σχεδίαση της βάσης ακολουθεί τις αρχές της απλότητας και της λειτουργικότητας, αποφεύγοντας τη χρήση περιορισμών όπως ξένων κλειδιών (foreign keys), ώστε να διατηρείται μεγαλύτερη ευελιξία στη διαχείριση των δεδομένων μέσω του backend.

Η βάση δεδομένων αναπτύχθηκε σε MySQL και περιλαμβάνει τους εξής βασικούς πίνακες:

Ο πίνακας users διαχειρίζεται όλα τα εγγεγραμμένα μέλη του συστήματος, τα οποία διακρίνονται σε δύο βασικούς ρόλους:

- **Admin (Διαχειριστής):** Έχει πλήρη δικαιώματα διαχείρισης του συστήματος.
- **Teacher (Καθηγητής):** Έχει δικαιώματα διαχείρισης των μαθημάτων του και δημιουργίας ερωτήσεων.

Δομή Πίνακα:

Πίνακας 4.1. Πίνακας Users

Πεδίο	Τύπος Δεδομένων	Περιγραφή
id	INT (PK)	Μοναδικός αναγνωριστικός αριθμός χρήστη.
name	VARCHAR(100)	Όνοματεπώνυμο χρήστη.
email	VARCHAR(150)	Ηλεκτρονική διεύθυνση (μοναδική).
password	VARCHAR(150)	Κωδικός πρόσβασης.
role	ENUM	Ο ρόλος του χρήστη (admin, teacher).
created_at	DATETIME	Ημερομηνία δημιουργίας λογαριασμού.

Ο πίνακας courses αποθηκεύει πληροφορίες για τα μαθήματα του συστήματος. Κάθε μάθημα μπορεί να έχει έναν υπεύθυνο καθηγητή.

Δομή Πίνακα:

Πίνακας 4.2. Πίνακας Courses

Πεδίο	Τύπος Δεδομένων	Περιγραφή
id	INT (PK)	Μοναδικός αναγνωριστικός αριθμός μαθήματος.

Πεδίο	Τύπος Δεδομένων	Περιγραφή
title	VARCHAR(150)	Τίτλος μαθήματος.
description	TEXT	Περιγραφή του μαθήματος.
teacher_id	INT	ID του καθηγητή που έχει αναλάβει το μάθημα.
created_at	DATETIME	Ημερομηνία δημιουργίας μαθήματος.

Ο πίνακας questions καταγράφει τις ερωτήσεις πολλαπλής επιλογής που δημιουργούν οι καθηγητές για τα μαθήματά τους. Κάθε ερώτηση μπορεί να έχει έως τέσσερις επιλογές απάντησης.

Δομή Πίνακα:

Πίνακας 4.3. Πίνακας Questions

Πεδίο	Τύπος Δεδομένων	Περιγραφή
id	INT (PK)	Μοναδικός αναγνωριστικός αριθμός ερώτησης.
course_id	INT	ID μαθήματος στο οποίο ανήκει η ερώτηση.
question_text	TEXT	Το κείμενο της ερώτησης.
option_a	VARCHAR(150)	Επιλογή Α.
option_b	VARCHAR(150)	Επιλογή Β.
option_c	VARCHAR(150)	Επιλογή Γ (προαιρετική).
option_d	VARCHAR(150)	Επιλογή Δ (προαιρετική).
correct_option	CHAR(1)	Η σωστή απάντηση (A, B, C, D).
code	VARCHAR(50)	Μοναδικός κωδικός ερώτησης (μορφή: ιδερώτησης+10+ιδμαθήματος).
starts_at	DATETIME	Χρόνος έναρξης ισχύος της ερώτησης.
expires_at	DATETIME	Χρόνος λήξης ισχύος της ερώτησης.
created_at	DATETIME	Ημερομηνία δημιουργίας ερώτησης.

Η δημιουργία του κωδικού code εξασφαλίζει την ασφάλεια και την ταχύτητα εισαγωγής της ερώτησης από τους φοιτητές.

Ο πίνακας responses καταγράφει τις απαντήσεις που υποβάλλουν οι φοιτητές, λειτουργώντας παράλληλα και ως μηχανισμός καταγραφής παρουσιών.

Δομή Πίνακα:

Πίνακας 4.4. Πίνακας Responses

Πεδίο	Τύπος Δεδομένων	Περιγραφή
id	INT (PK)	Μοναδικός αναγνωριστικός αριθμός απάντησης.
question_id	INT	ID της ερώτησης.
am	VARCHAR(20)	Αριθμός Μητρώου φοιτητή.
surname	VARCHAR(100)	Επώνυμο φοιτητή.
email	VARCHAR(150)	Ηλεκτρονική διεύθυνση φοιτητή.
answer	CHAR(1)	Επιλεγμένη απάντηση (A, B, C, D).
ip_address	VARCHAR(45)	IP Διεύθυνση υποβολής (για έλεγχο μοναδικότητας).
responded_at	DATETIME	Ημερομηνία και ώρα υποβολής απάντησης.

Η βάση δεδομένων μας εξυπηρετεί κρίσιμες λειτουργίες του συστήματος:

- Διαχείριση Χρηστών και Ρόλων: Ορίζει και διαχωρίζει τα δικαιώματα admin και καθηγητών.
- Οργάνωση Μαθημάτων: Επιτρέπει στον admin να δημιουργεί και να εκχωρεί μαθήματα σε καθηγητές.
- Υποστήριξη Διαδικασίας Καταγραφής Παρουσιών: Μέσω της δημιουργίας ερωτήσεων, των κωδικών και της υποβολής απαντήσεων.
- Στατιστική Ανάλυση: Παρέχει τα δεδομένα για την παραγωγή στατιστικών αναφορών και διαγραμμάτων σχετικά με την επίδοση και τη συμμετοχή των φοιτητών.

- Διασφάλιση Αδιάβλητης Διαδικασίας: Με την καταγραφή IP διευθύνσεων και την απαγόρευση πολλαπλών συμμετοχών.

4.3 Το σύστημα

4.3.1 Διαγράμματα Ροής



Εικόνα 4.1: Διαδικασία Διαχείρισης από τον Διαχειριστή (Admin) - Διαχείριση Μαθημάτων και Καθηγητών από Admin



Εικόνα 4.2: Ροή Διαχείρισης από τον Καθηγητή - Διαδικασία Δημιουργίας και Παρακολούθησης Ερωτήσεων



Εικόνα 4.3: Ροή Χρήσης από τον Φοιτητή - Διαδικασία Δήλωσης Παρουσίας από τον Φοιτητή

4.3.2 Περιγραφή και Παρουσίαση της Ιστοσελίδας

Η διαδικτυακή εφαρμογή σχεδιάστηκε με στόχο την απλότητα, την ευχρηστία και την άμεση πρόσβαση σε όλες τις απαραίτητες λειτουργίες τόσο για τους διαχειριστές όσο και για τους καθηγητές και φοιτητές. Η αρχιτεκτονική του συστήματος ακολουθεί την αρχή του καθαρού σχεδιασμού επιτρέποντας στους χρήστες να εκτελούν τις ενέργειές τους με τον ελάχιστο δυνατό αριθμό βημάτων.

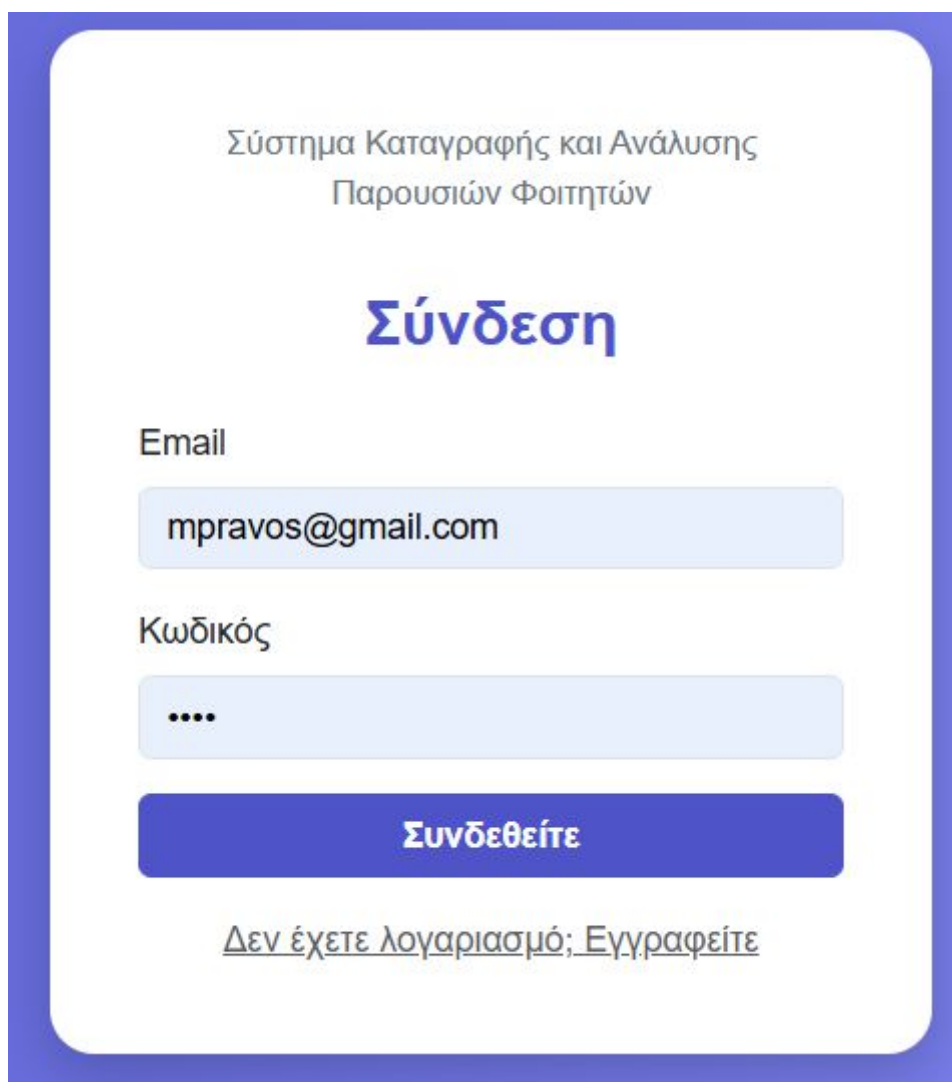
Η ιστοσελίδα προσφέρει προσωποποιημένο περιβάλλον για κάθε κατηγορία χρήστη:

- Οι Διαχειριστές (Admins) έχουν πρόσβαση σε εργαλεία διαχείρισης μαθημάτων και χρηστών.

- Οι Καθηγητές διαχειρίζονται τα μαθήματα και τις ερωτήσεις πολλαπλής επιλογής, ενώ παρακολουθούν και αναλύουν τις απαντήσεις των φοιτητών.
- Οι Φοιτητές χρησιμοποιούν το σύστημα για να δηλώσουν την παρουσία τους απαντώντας σε ερωτήσεις, με εύκολη και γρήγορη διαδικασία.

Η διεπαφή (User Interface) αναπτύχθηκε με χρήση μοντέρνων τεχνολογιών όπως Laravel, Bootstrap, DataTables, και Chart.js, ώστε να έχει ταχύτητα, απόκριση και καλή εμφάνιση.

Στη συνέχεια παρουσιάζονται αναλυτικά τα βασικά σημεία της ιστοσελίδας με αντίστοιχες εικόνες που δείχνουν την πλήρη λειτουργικότητα του συστήματος.



The image shows a login form titled "Σύστημα Καταγραφής και Ανάλυσης Παρουσιών Φοιτητών" (System for Student Registration and Analysis). The main heading is "Σύνδεση" (Login). The form includes an "Email" field with the text "mpravos@gmail.com" and a "Κωδικός" (Password) field with four dots. Below the fields is a blue button labeled "Συνδεθείτε" (Login). At the bottom, there is a link that says "Δεν έχετε λογαριασμό; Εγγραφείτε" (Don't have an account? Register).

Εικόνα 4.4: Σύνδεση χρήστη – εκτός φοιτητή

Στην Εικόνα 3.4 παρουσιάζεται η αρχική οθόνη εισόδου στο σύστημα μέσω της οποίας οι διαχειριστές και καθηγητές μπορούν να συνδεθούν για να αποκτήσουν πρόσβαση στις λειτουργίες του συστήματος.

Στην Εικόνα 3.5 εμφανίζεται η φόρμα εγγραφής νέου χρήστη, όπου δίνεται η δυνατότητα δημιουργίας λογαριασμού για διαχειριστές και καθηγητές με την εισαγωγή βασικών στοιχείων.

Πλατφόρμα Υποβολής και Διαχείρισης
Παρουσιών Φοιτητών

Εγγραφή

Επώνυμο

Όνομα

Email

Κωδικός

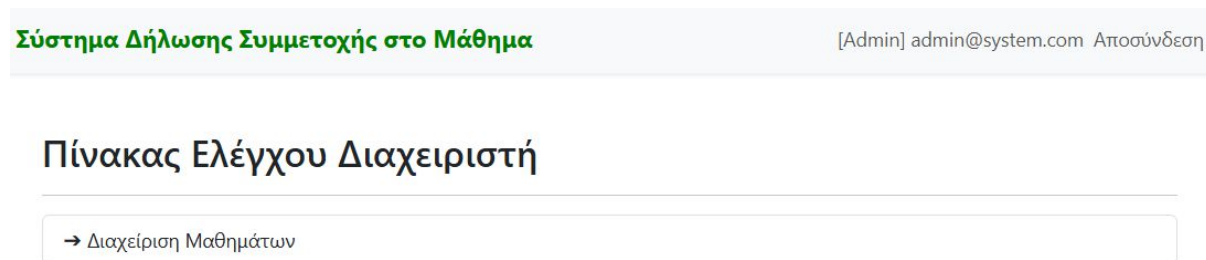
Εγγραφή

[Έχετε ήδη λογαριασμό; Σύνδεση](#)

Εικόνα 4.5: Εγγραφή χρήστη – εκτός φοιτητή

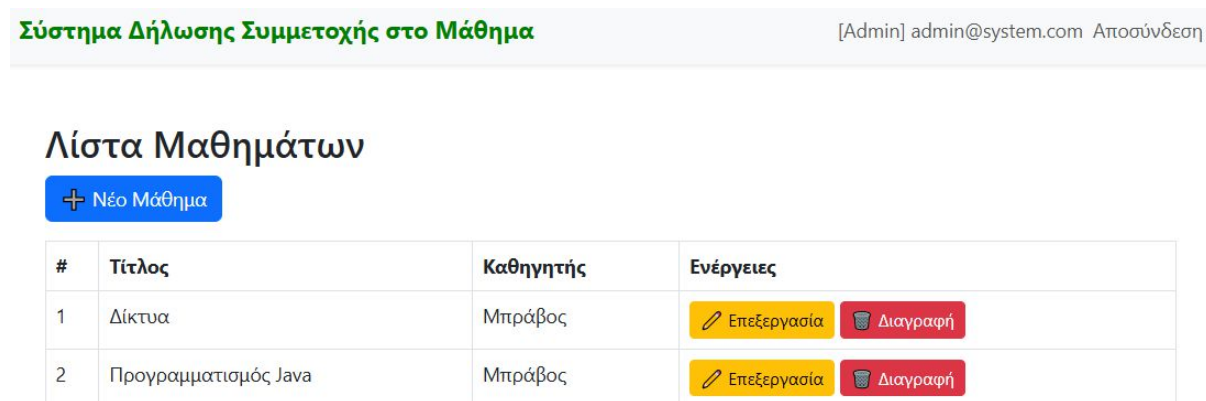
4.3.2.1 Από τη μεριά του Admin

Στην Εικόνα 3.6 παρουσιάζεται ο πίνακας ελέγχου του διαχειριστή από όπου έχει συνολική εποπτεία του συστήματος και πρόσβαση σε βασικές λειτουργίες διαχείρισης.



Εικόνα 4.6: Πίνακας Ελέγχου Διαχειριστή - Admin

Στην Εικόνα 3.7 απεικονίζεται η λίστα των μαθημάτων που έχουν καταχωρηθεί στο σύστημα. Ο διαχειριστής μπορεί να δει αναλυτικές πληροφορίες για κάθε μάθημα και να εκτελέσει σχετικές ενέργειες.



Εικόνα 4.7: Λίστα Μαθημάτων - Admin

Στην Εικόνα 3.8 παρουσιάζεται η φόρμα επεξεργασίας μαθήματος, όπου ο διαχειριστής μπορεί να τροποποιήσει τον τίτλο, την περιγραφή και να αλλάξει τον καθηγητή που είναι υπεύθυνος για το μάθημα.

Στην Εικόνα 3.9 εμφανίζεται η φόρμα δημιουργίας νέου μαθήματος, όπου ο διαχειριστής μπορεί να εισάγει τα στοιχεία του μαθήματος και να το αναθέσει σε συγκεκριμένο καθηγητή.

Επεξεργασία Μαθήματος

Τίτλος Μαθήματος

Περιγραφή

Καθηγητής

Εικόνα 4.8: Επεξεργασία Μαθήματος - Admin

Δημιουργία Νέου Μαθήματος

Τίτλος Μαθήματος

Περιγραφή

Καθηγητής

Εικόνα 4.9: Δημιουργία Νέου Μαθήματος - Admin

4.3.2.2 Από τη μεριά του Teacher

Πίνακας Ελέγχου Καθηγητή

→ Διαχείριση Ερωτήσεων

Εικόνα 4.10: Πίνακας Ελέγχου Καθηγητή - Καθηγητής

Στην Εικόνα 3.10 παρουσιάζεται ο πίνακας ελέγχου του καθηγητή, μέσω του οποίου μπορεί να διαχειρίζεται τα μαθήματά του και να δημιουργεί ερωτήσεις πολλαπλής επιλογής.

Λίστα Ερωτήσεων

[+ Νέα Ερώτηση](#)

Show entries Search:

#	Ερώτηση	Μάθημα	Κωδικός	Ισχύει Από	Ισχύει Μέχρι	Ενέργειες
1	Τι είναι το TCP/IP?	Δίκτυα	1101	13-05-2025 22:00	13-05-2025 22:20	Επεξεργασία Διαγραφή Αποτελέσματα
2	Ένας διακομιστής είναι	Δίκτυα	2101	13-05-2025 22:20	13-05-2025 22:40	Επεξεργασία Διαγραφή Αποτελέσματα
3	Ποια από τις παρακάτω κλάσεις χρησιμοποιείται για...	Προγραμματισμός Java	3102	14-05-2025 12:30	14-05-2025 12:40	Επεξεργασία Διαγραφή Αποτελέσματα

Showing 1 to 3 of 3 entries Previous 1 Next

Εικόνα 4.11: Λίστα Ερωτήσεων - Καθηγητής

Στην Εικόνα 3.11 απεικονίζεται η λίστα των ερωτήσεων που έχει δημιουργήσει ο καθηγητής, με δυνατότητα προβολής, επεξεργασίας ή διαγραφής κάθε ερώτησης.

Δημιουργία Νέας Ερώτησης

Μάθημα

-- Επιλέξτε Μάθημα --

Ερώτηση

Επιλογές

Επιλογή A

Επιλογή B

Επιλογή C

Επιλογή D

Σωστή Επιλογή

-- Επιλέξτε --

Έναρξη Ισχύος

mm/dd/yyyy --:-- --

Λήξη Ισχύος

mm/dd/yyyy --:-- --

 Αποθήκευση

Εικόνα 4.12: Δημιουργία Νέας Ερώτησης - Καθηγητής

Στην Εικόνα 3.12 παρουσιάζεται η φόρμα δημιουργίας νέας ερώτησης πολλαπλής επιλογής για την καταγραφή παρουσιών των φοιτητών.

Δημιουργία Νέας Ερώτησης

Μάθημα

-- Επιλέξτε Μάθημα --

-- Επιλέξτε Μάθημα --

Δίκτυα

Προγραμματισμός Java

Σωστή Επιλογή

-- Επιλέξτε --

-- Επιλέξτε --

A

B

C

D

mm/dd/yyyy --:-- --

Επιλογές

Επιλογή Α

May 2025

Su Mo Tu We Th Fr Sa

27 28 29 30 1 2 3

4 5 6 7 8 9 10

11 12 13 14 15 16 17

18 19 20 21 22 23 24

25 26 27 28 29 30 31

1 2 3 4 5 6 7

Clear Today

09 00 PM

10 AM

11 01

12 02

01 03

02 04

03 05

06

05/01/2025 09:00 PM

Εικόνα 4.13: Επιλογές dropdown λίστα για Μάθημα και Σωστής Επιλογής και φόρμα για τον καθορισμό ημερομηνιών κατά τη Δημιουργία Νέας Ερώτησης - Καθηγητής

Στην Εικόνα 3.13 παρουσιάζεται η φόρμα μέσω της οποίας ο καθηγητής έχει τη δυνατότητα να δημιουργήσει μια νέα ερώτηση πολλαπλής επιλογής, σχετίζοντάς την άμεσα με ένα από τα διαθέσιμα μαθήματά του. Μέσα από την αναπτυσσόμενη λίστα (dropdown) επιλέγει το μάθημα στο οποίο θα αναρτηθεί η ερώτηση, ενώ παράλληλα έχει τη δυνατότητα να ορίσει τη σωστή απάντηση μεταξύ των διαθέσιμων επιλογών (Α, Β, Γ, Δ).

Επιπλέον, μέσω των πεδίων ημερομηνίας και ώρας, ο καθηγητής καθορίζει το χρονικό διάστημα κατά το οποίο η ερώτηση θα είναι ενεργή και διαθέσιμη προς απάντηση από τους φοιτητές. Ο ακριβής καθορισμός της ημερομηνίας έναρξης και λήξης εξασφαλίζει τον έλεγχο της διαθεσιμότητας της ερώτησης, διασφαλίζοντας ότι η καταγραφή παρουσίας πραγματοποιείται αποκλειστικά εντός του προκαθορισμένου χρονικού πλαισίου.

Στην Εικόνα 3.14 παρουσιάζεται η φόρμα επεξεργασίας ερώτησης, όπου ο καθηγητής μπορεί να τροποποιήσει το κείμενο της ερώτησης, τις επιλογές και το χρονικό πλαίσιο ισχύος.

Επεξεργασία Ερώτησης

Μάθημα

Δίκτυα

Ερώτηση

Τι είναι το TCP/IP?

Επιλογές

Δίκτυο

Πρωτόκολλο

Μέθοδος

Κρυπτογράφηση

Σωστή Επιλογή

B

Έναρξη Ισχύος

05/13/2025 10:00 PM

Λήξη Ισχύος

05/13/2025 10:20 PM

Ενημέρωση

Πίσω

Εικόνα 4.14: Επεξεργασία Ερώτησης - Καθηγητής

Στην Εικόνα 3.15 απεικονίζεται η σελίδα προβολής των απαντήσεων των φοιτητών, με έγχρωμη επισήμανση για το αν η απάντηση ήταν σωστή ή λανθασμένη.

Εικόνα 4.15: Αποτελέσματα Ερώτησης με προβολή της σωστής απόκρισης - Καθηγητής

Στην Εικόνα 3.16 παρουσιάζεται το στατιστικό γράφημα (bar chart) που εμφανίζει την κατανομή των απαντήσεων των φοιτητών και το ποσοστό επιτυχίας.

Αποτελέσματα Ερώτησης

Ερώτηση: Τι είναι το TCP/IP?

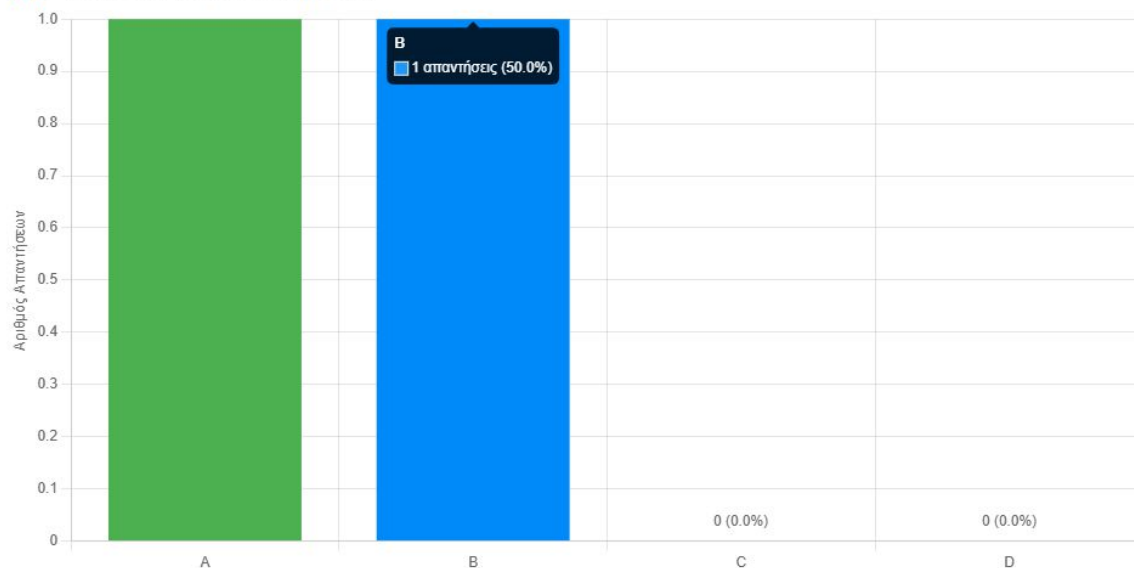
Κωδικός: 1101

Εξαγωγή σε Excel Search:

#	AM	Επώνυμο	Email	Απάντηση	IP	Ημ/νία Υποβολής	Κατάσταση
1	134031	Μπραβίδης	mpravos@gmail.com	B	127.0.0.2	13-05-2025 14:37	Σωστό
2	134032	Τακίδης	mpira@gmail.com	A	127.0.0.1	13-05-2025 14:40	Λάθος

Showing 1 to 2 of 2 entries Previous Next

Στατιστικά Απαντήσεων



Εικόνα 4.16: Στατιστικά απαντήσεων σε bar chart - Καθηγητής

#	AM	Επώνυμο	Email	Απάντηση	IP	Ημ/νία Υποβολής	Κατάσταση
1	134031	Μπραβίδης	mpravos@gmail.com	B	127.0.0.2	13-05-2025 14:37	Σωστό
2	134032	Τακίδης	mpira@gmail.com	A	127.0.0.1	13-05-2025 14:40	Λάθος

Εικόνα 4.17: Excel αρχείο με τα αποτελέσματα - Καθηγητής

Στην Εικόνα 3.17 απεικονίζεται το αρχείο Excel που μπορεί να εξαχθεί από το σύστημα, περιέχοντας αναλυτικά τις απαντήσεις και τα στοιχεία των φοιτητών.

4.3.2.3 Από τη μεριά του Student

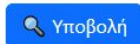
Ο καθηγητής μέσα στην αίθουσα δίνει στον φοιτητή:

1. Το link που θα μπει πχ <http://.../.../attendance>
2. Και το κωδικό της ερώτησης που θα βάλει ο φοιτητής για να μπορέσει να δει την εκφώνηση και τις πιθανές απαντήσεις.

Στην Εικόνα 3.18 παρουσιάζεται η αρχική φόρμα όπου ο φοιτητής καλείται να εισάγει τον μοναδικό κωδικό της ερώτησης που του έχει δοθεί από τον καθηγητή. Η εισαγωγή του κωδικού αποτελεί το πρώτο βήμα για τη δήλωση παρουσίας και την πρόσβαση στην αντίστοιχη ερώτηση.

Δήλωση Παρουσίας

Κωδικός Ερώτησης



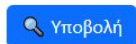
Εικόνα 4.18: Δήλωση Παρουσίας – Εισαγωγή κωδικού – Φοιτητής

Στην Εικόνα 3.19 εμφανίζεται το μήνυμα σφάλματος που λαμβάνει ο φοιτητής όταν εισάγει έναν μη έγκυρο κωδικό ή όταν η ερώτηση έχει λήξει και δεν είναι πλέον διαθέσιμη. Το σύστημα ενημερώνει ξεκάθαρα για την αιτία της απόρριψης ώστε να αποφευχθούν λανθασμένες ενέργειες.

Δήλωση Παρουσίας

- Ο κωδικός δεν είναι ενεργός.

Κωδικός Ερώτησης



Εικόνα 4.19: Εισαγωγή κωδικού σε ερώτηση που δεν υπάρχει ή είναι εκτός προθεσμίας – Φοιτητής

Στην Εικόνα 3.20 παρουσιάζεται η φόρμα υποβολής απάντησης, όπου ο φοιτητής συμπληρώνει τα απαραίτητα προσωπικά στοιχεία (Αριθμό Μητρώου, Επώνυμο, Email) και επιλέγει την απάντησή του στην ερώτηση. Η διαδικασία αυτή ολοκληρώνει τη δήλωση παρουσίας και καταγράφει την απάντηση στο σύστημα.

Απάντηση στην Ερώτηση

Ερώτηση: Ένας διακομιστής είναι

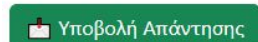
Αριθμός Μητρώου (AM)

Επώνυμο

Email

Απάντηση

- A: Κεντρικός κόμβος στο δίκτυο που μεταβιβάζει πακέτα
- B: Τερματικός σταθμός
- C: Τερματικός σταθμός με ασύρματη σύνδεση
- D: Τίποτα από τα αναφερθέντα

 Υποβολή Απάντησης

Εικόνα 4.20: Απάντηση στην Ερώτηση – Εισαγωγή όλων των απαραίτητων στοιχείων – Φοιτητής

Κεφάλαιο 5ο: Τα συμπεράσματα της εργασίας και τι μπορεί να βελτιωθεί

Η ανάπτυξη και υλοποίηση του παρόντος συστήματος καταγραφής παρουσιών βασισμένου σε ερωτήσεις πολλαπλής επιλογής ανέδειξε τις δυνατότητες που προσφέρουν οι ψηφιακές τεχνολογίες στην εκπαιδευτική διαδικασία. Το σύστημα συνδύασε λειτουργικότητα, αξιοπιστία και απλότητα, προσφέροντας ένα καλό μοντέλο παρακολούθησης παρουσιών φοιτητών το οποίο εμπλέκει ενεργά τους συμμετέχοντες μέσω της απάντησης σε σύντομες ερωτήσεις. Το κεφάλαιο αυτό παρουσιάζει τα βασικά συμπεράσματα που προέκυψαν από την υλοποίηση της εργασίας, καθώς και προτάσεις για μελλοντική βελτίωση και εξέλιξη του συστήματος.

Πρώτον, το σύστημα κατάφερε να επιτύχει τον αρχικό στόχο του: την αξιόπιστη καταγραφή παρουσιών με τρόπο που ενισχύει τη συμμετοχή και τη συγκέντρωση των φοιτητών. Η χρήση μιας ερώτησης πολλαπλής επιλογής ως μηχανισμός παρουσιολογίου όχι μόνο καταγράφει την παρουσία αλλά λειτουργεί ταυτόχρονα και ως εργαλείο γρήγορης αξιολόγησης.

Δεύτερον, η πλατφόρμα αποδείχθηκε ιδιαίτερα φιλική προς τον χρήστη, τόσο για τους φοιτητές όσο και για τους καθηγητές. Η διεπαφή είναι καθαρή, άμεση και επιτρέπει την ολοκλήρωση των διαδικασιών χωρίς πολύπλοκα βήματα με διαλειτουργικότητα και ταχύτητα πρόσβασης.

Τρίτον, το σύστημα προσφέρει ουσιαστική ανατροφοδότηση στους καθηγητές μέσω των γραφημάτων και των στατιστικών ανάλυσης που τους επιτρέπει να αξιολογήσουν σε πραγματικό χρόνο την κατανόηση των φοιτητών σε συγκεκριμένα σημεία του μαθήματος. Η δυνατότητα εξαγωγής αποτελεσμάτων σε Excel διευκολύνει επιπλέον την αποθήκευση και επεξεργασία δεδομένων εκτός της εφαρμογής.

Αν και το σύστημα καλύπτει τον βασικό του σκοπό υπάρχουν περιθώρια περαιτέρω ενίσχυσης των δυνατοτήτων του. Μια πρώτη κατεύθυνση θα μπορούσε να είναι η προσθήκη μηχανισμού ειδοποιήσεων (email ή ειδοποίηση εντός συστήματος) προς τους φοιτητές όταν μια νέα ερώτηση δημιουργείται για να αυξηθεί η ανταπόκριση και να μειωθούν οι απώλειες παρουσιών.

Μια δεύτερη βελτίωση αφορά την ασφάλεια και την ακρίβεια ταυτοποίησης των φοιτητών. Θα μπορούσε να ενσωματωθεί ένας ελαφρύς μηχανισμός επιβεβαίωσης ταυτότητας (όπως π.χ. ένας προσωρινός κωδικός που αποστέλλεται στο email τους), προκειμένου να διασφαλιστεί ότι η απάντηση καταχωρείται από τον πραγματικό κάτοχο του Αριθμού Μητρώου.

Κατά τη διάρκεια της συγγραφής της παρούσας εργασίας έγινε χρήση εργαλείων τεχνητής νοημοσύνης με σκοπό την υποβοήθηση στη διατύπωση συντακτικών εκφράσεων. Τα εργαλεία αυτά χρησιμοποιήθηκαν αποκλειστικά ως υποστηρικτικά μέσα, με την τελική επιμέλεια, επιλογή περιεχομένου και σύνταξη να πραγματοποιείται από τον συντάκτη της εργασίας.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] <https://tophat.com/features/attendance-software/>
- [2] <https://otl.du.edu/knowledgebase/introduction-to-top-hat/>
- [3] <https://www.quickschools.com/>
- [4] <https://www.myattendancetracker.com/>
- [5] <https://attendanceradar.com>
- [6] <https://www.scholarchip.com/>
- [7] https://www.tutorialspoint.com/laravel/laravel_overview.htm
- [8] <https://scaleupally.io/blog/benefits-of-laravel-framework/>
- [9] <https://www.ranktracker.com/el/blog/why-laravel-remains-the-best-php-framework-in-2024/>
- [10] <https://belitsoft.com/laravel-development-services/10-benefits-using-laravel-php-framework>
- [11] https://blog.jetbrains.com/phpstorm/2024/09/laravel-trends-2024-the-latest-market-insights/?utm_source=chatgpt.com
- [12] <https://www.w3schools.com/MySQL/default.asp>
- [13] <https://blueclawdb.com/mysql/advantages-disadvantages-mysql/>
- [14] <https://www.w3schools.blog/mysql-advantages-disadvantages>
- [15] <https://planetscale.com/blog/build-a-laravel-application-with-a-mysql-database>
- [16] <https://www.chartjs.org/>
- [17] https://www.w3schools.com/ai/ai_chartjs.asp
- [18] <https://datatables.net/>
- [19] <https://medium.com/@patrickwanchinyeep/basic-usage-of-datatable-c2fef17db558>
- [20] <https://getbootstrap.com/>
- [21] https://www.w3schools.com/bootstrap/bootstrap_get_started.asp

ΠΑΡΑΡΤΗΜΑ

Κώδικες που χρησιμοποίησα:

AdminController

```
<?php

namespace App\Http\Controllers;

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

class AdminController extends Controller
{
    // Έλεγχος σύνδεσης και ρόλου admin
    private function checkAdmin()
    {
        if (!Session::has('user_id') || Session::get('user_role') != 'admin') {
            abort(403, 'Access Denied');
        }
    }

    // Admin Dashboard
    public function dashboard()
    {
        $this->checkAdmin();
        return view('admin.dashboard');
    }

    // Προβολή όλων των μαθημάτων
    public function courses()
    {
        $this->checkAdmin();
        $courses = DB::table('courses')->get();
        return view('admin.courses.index', compact('courses'));
    }
}
```

```

// Φόρμα δημιουργίας μαθήματος
public function createCourse()
{
    $this->checkAdmin();
    $teachers = DB::table('users')->where('role', 'teacher')->get();
    return view('admin.courses.create', compact('teachers'));
}

// Αποθήκευση νέου μαθήματος
public function storeCourse(Request $request)
{
    $this->checkAdmin();

    $request->validate([
        'title' => 'required|string|max:150',
        'description' => 'nullable|string',
        'teacher_id' => 'required|integer',
    ]);

    DB::table('courses')->insert([
        'title' => $request->title,
        'description' => $request->description,
        'teacher_id' => $request->teacher_id,
        'created_at' => now(),
    ]);

    return redirect()->route('admin.courses')->with('success', 'Το μάθημα δημιουργήθηκε.');
```

```

}

// Επεξεργασία μαθήματος
public function editCourse($id)
{
    $this->checkAdmin();

    $course = DB::table('courses')->where('id', $id)->first();
    $teachers = DB::table('users')->where('role', 'teacher')->get();

    if (!$course) {
        abort(404, 'Το μάθημα δεν βρέθηκε.');
```

```

    }

    return view('admin.courses.edit', compact('course', 'teachers'));
}

```

```

// Ενημέρωση μαθήματος
public function updateCourse(Request $request, $id)
{
    $this->checkAdmin();

    $request->validate([
        'title' => 'required|string|max:150',
        'description' => 'nullable|string',
        'teacher_id' => 'required|integer',
    ]);

    DB::table('courses')->where('id', $id)->update([
        'title' => $request->title,
        'description' => $request->description,
        'teacher_id' => $request->teacher_id,
    ]);

    return redirect()->route('admin.courses')->with('success', 'Το μάθημα ενημερώθηκε.');
```

```

}

// Διαγραφή μαθήματος
public function deleteCourse($id)
{
    $this->checkAdmin();

    DB::table('courses')->where('id', $id)->delete();

    return redirect()->route('admin.courses')->with('success', 'Το μάθημα διαγράφηκε.');
```

```

}
}

```

TeacherController

```

<?php

namespace App\Http\Controllers;
```

```

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

class TeacherController extends Controller
{
    // Έλεγχος σύνδεσης και ρόλου teacher
    private function checkTeacher()
    {
        if (!Session::has('user_id') || Session::get('user_role') != 'teacher') {
            abort(403, 'Access Denied');
        }
    }

    // Teacher Dashboard
    public function dashboard()
    {
        $this->checkTeacher();
        return view('teacher.dashboard');
    }

    // Λίστα Ερωτήσεων
    public function questions()
    {
        $this->checkTeacher();
        $teacherId = Session::get('user_id');

        $questions = DB::table('questions')
            ->join('courses', 'questions.course_id', '=', 'courses.id')
            ->where('courses.teacher_id', $teacherId)
            ->select('questions.*', 'courses.title as course_title')
            ->get();

        return view('teacher.questions.index', compact('questions'));
    }

    // Φόρμα Δημιουργίας Ερώτησης
    public function createQuestion()
    {
        $this->checkTeacher();
    }
}

```

```

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

$Courses = DB::table('courses')->where('teacher_id', $teacherId)->get();

return view('teacher.questions.create', compact('courses'));
}

// Αποθήκευση Ερώτησης
public function storeQuestion(Request $request)
{
    $this->checkTeacher();

    $request->validate([
        'course_id' => 'required|integer',
        'question_text' => 'required|string',
        'option_a' => 'required|string',
        'option_b' => 'required|string',
        'option_c' => 'nullable|string',
        'option_d' => 'nullable|string',
        'correct_option' => 'required|in:A,B,C,D',
        'starts_at' => 'required|date',
        'expires_at' => 'required|date|after:starts_at',
    ]);

    // Δημιουργία κωδικού ερώτησης: ιδερότητας+10+ιδμαθήματος (προσωρινά το ιδερότητας το θεωρούμε
    0, θα γίνει update μετά)
    $tempCode = 'TMP';

    $id = DB::table('questions')->insertGetId([
        'course_id' => $request->course_id,
        'question_text' => $request->question_text,
        'option_a' => $request->option_a,
        'option_b' => $request->option_b,
        'option_c' => $request->option_c,
        'option_d' => $request->option_d,
        'correct_option' => $request->correct_option,
        'starts_at' => $request->starts_at,
        'expires_at' => $request->expires_at,
        'code' => $tempCode, // προσωρινά
        'created_at' => now(),
    ]);
}

```

```

]);

// Οριστική τιμή κωδικού
$finalCode = $id . '10' . $request->course_id;
DB::table('questions')->where('id', $id)->update(['code' => $finalCode]);

return redirect()->route('teacher.questions')->with('success', 'Η ερώτηση δημιουργήθηκε.');
```

```

}

// Προβολή Απαντήσεων Ερώτησης
public function viewResults($id)
{
    $this->checkTeacher();

    $question = DB::table('questions')->where('id', $id)->first();

    if (!$question) {
        abort(404, 'Η ερώτηση δεν βρέθηκε.');
```

```

    }

    $responses = DB::table('responses')->where('question_id', $id)->get();

    return view('teacher.questions.results', compact('question', 'responses'));
}

// Επεξεργασία Ερώτησης
public function editQuestion($id)
{
    $this->checkTeacher();
    $question = DB::table('questions')->where('id', $id)->first();
    $teacherId = Session::get('user_id');
    $courses = DB::table('courses')->where('teacher_id', $teacherId)->get();

    if (!$question) {
        abort(404, 'Η ερώτηση δεν βρέθηκε.');
```

```

    }

    return view('teacher.questions.edit', compact('question', 'courses'));
}

```

```

// Ενημέρωση Ερώτησης
public function updateQuestion(Request $request, $id)
{
    $this->checkTeacher();

    $request->validate([
        'course_id' => 'required|integer',
        'question_text' => 'required|string',
        'option_a' => 'required|string',
        'option_b' => 'required|string',
        'option_c' => 'nullable|string',
        'option_d' => 'nullable|string',
        'correct_option' => 'required|in:A,B,C,D',
        'starts_at' => 'required|date',
        'expires_at' => 'required|date|after:starts_at',
    ]);

    DB::table('questions')->where('id', $id)->update([
        'course_id' => $request->course_id,
        'question_text' => $request->question_text,
        'option_a' => $request->option_a,
        'option_b' => $request->option_b,
        'option_c' => $request->option_c,
        'option_d' => $request->option_d,
        'correct_option' => $request->correct_option,
        'starts_at' => $request->starts_at,
        'expires_at' => $request->expires_at,
    ]);

    return redirect()->route('teacher.questions')->with('success', 'Η ερώτηση ενημερώθηκε.');
```

```

// Διαγραφή Ερώτησης
```

```

public function deleteQuestion($id)
{
    $this->checkTeacher();
    DB::table('questions')->where('id', $id)->delete();

    return redirect()->route('teacher.questions')->with('success', 'Η ερώτηση διαγράφηκε.');
```

```
}
```

AttendanceController

```
<?php

namespace App\Http\Controllers;

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

class AttendanceController extends Controller
{
    // Εμφάνιση φόρμας εισαγωγής κωδικού
    public function enterCode()
    {
        return view('attendance.enter_code');
    }

    // Έλεγχος κωδικού και μετάβαση στη φόρμα απάντησης
    public function verifyCode(Request $request)
    {
        $request->validate([
            'code' => 'required|string'
        ]);

        $code = $request->code;
        $question = DB::table('questions')->where('code', $code)->first();

        if (!$question) {
            return back()->withErrors(['code' => 'Μη έγκυρος κωδικός.']);
        }

        $now = now();

        if ($now < $question->starts_at || $now > $question->expires_at) {
            return back()->withErrors(['code' => 'Ο κωδικός δεν είναι ενεργός.']);
        }
    }
}
```

```

return redirect()->route('attendance.form', ['code' => $code]);
}

// Φόρμα Υποβολής Απάντησης
public function showForm($code)
{
    $question = DB::table('questions')->where('code', $code)->first();

    if (!$question) {
        abort(404, 'Η ερώτηση δεν βρέθηκε. ');
    }

    return view('attendance.answer_form', compact('question'));
}

// Υποβολή Απάντησης
public function submitResponse(Request $request, $code)
{
    $request->validate([
        'am' => 'required|string|max:20',
        'surname' => 'required|string|max:100',
        'email' => 'required|email|max:150',
        'answer' => 'required|in:A,B,C,D',
    ]);

    $question = DB::table('questions')->where('code', $code)->first();

    if (!$question) {
        abort(404, 'Η ερώτηση δεν βρέθηκε. ');
    }

    $ip = $request->ip();

    // Έλεγχος αν έχει ήδη απαντήσει με το ίδιο am ή ip
    $exists = DB::table('responses')
        ->where('question_id', $question->id)
        ->where(function ($q) use ($request, $ip) {
            $q->where('am', $request->am)
                ->orWhere('ip_address', $ip);
        })
        ->exists();

```

```

if ($exists) {
    return back()->withErrors(['am' => 'Έχετε ήδη υποβάλει απάντηση από αυτό το ΑΜ ή IP.']);
}

DB::table('responses')->insert([
    'question_id' => $question->id,
    'am' => $request->am,
    'surname' => $request->surname,
    'email' => $request->email,
    'answer' => $request->answer,
    'ip_address' => $ip,
    'responded_at' => now(),
]);

return redirect()->route('attendance.code')->with('success', 'Η απάντησή σας καταγράφηκε με επιτυχία.');
```

Web.php

```

<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\AdminController;
use App\Http\Controllers\TeacherController;
use App\Http\Controllers\UserController;

use App\Http\Controllers\AttendanceController;

// Υπάρχοντα Routes
Route::get('/', [UserController::class, 'mywelcome']->name('mywelcome'));
Route::get('/login', [UserController::class, 'loginform']->name('login'));
Route::post('/login', [UserController::class, 'login']);
Route::get('/register', [UserController::class, 'registerform']->name('register'));
Route::post('/register', [UserController::class, 'register']);
Route::post('/logout', [UserController::class, 'logout']->name('logout'));
Route::get('/dashboard', [UserController::class, 'dashboard']->name('dashboard'));
```

```

// ----- ADMIN ROUTES -----

Route::get('/admin/dashboard', [AdminController::class, 'dashboard']->name('admin.dashboard'));

Route::get('/admin/courses', [AdminController::class, 'courses']->name('admin.courses'));

Route::get('/admin/courses/create', [AdminController::class, 'createCourse']->name('admin.courses.create'));

Route::post('/admin/courses/store', [AdminController::class, 'storeCourse']->name('admin.courses.store'));

Route::get('/admin/courses/{id}/edit', [AdminController::class, 'editCourse']->name('admin.courses.edit'));

Route::post('/admin/courses/{id}/update', [AdminController::class, 'updateCourse']->name('admin.courses.update'));

Route::post('/admin/courses/{id}/delete', [AdminController::class, 'deleteCourse']->name('admin.courses.delete'));

// ----- TEACHER ROUTES -----

Route::get('/teacher/dashboard', [TeacherController::class, 'dashboard']->name('teacher.dashboard'));

// Διαχείριση Ερωτήσεων

Route::get('/teacher/questions', [TeacherController::class, 'questions']->name('teacher.questions'));

Route::get('/teacher/questions/create', [TeacherController::class, 'createQuestion']->name('teacher.questions.create'));

Route::post('/teacher/questions/store', [TeacherController::class, 'storeQuestion']->name('teacher.questions.store'));

Route::get('/teacher/questions/{id}/results', [TeacherController::class, 'viewResults']->name('teacher.questions.results'));

Route::get('/teacher/questions/{id}/edit', [TeacherController::class, 'editQuestion']->name('teacher.questions.edit'));

Route::post('/teacher/questions/{id}/update', [TeacherController::class, 'updateQuestion']->name('teacher.questions.update'));

Route::post('/teacher/questions/{id}/delete', [TeacherController::class, 'deleteQuestion']->name('teacher.questions.delete'));

// ----- ATTENDANCE (ΔΗΛΩΣΗ ΠΑΡΟΥΣΙΑΣ) -----

```

```
// Εισαγωγή Κωδικού Ερώτησης
Route::get('/attendance', [AttendanceController::class, 'enterCode']->name('attendance.code'));

Route::post('/attendance/verify', [AttendanceController::class, 'verifyCode']->name('attendance.verify'));

// Υποβολή Απάντησης
Route::get('/attendance/{code}/form', [AttendanceController::class, 'showForm']->name('attendance.form'));

Route::post('/attendance/{code}/submit', [AttendanceController::class, 'submitResponse']->name('attendance.submit'));
```

Results blade

```
@extends('layouts.app')

@section('content')
<div class="container mt-4">
    <h2>Αποτελέσματα Ερώτησης</h2>
    <p><strong>Ερώτηση:</strong> {{ $question->question_text }}</p>
    <p><strong>Κωδικός:</strong> {{ $question->code }}</p>
    <hr>

    <table id="resultsTable" class="table table-bordered table-striped">
        <thead>
            <tr>
                <th>#</th>
                <th>ΑΜ</th>
                <th>Επώνυμο</th>
                <th>Email</th>
                <th>Απάντηση</th>
                <th>IP</th>
                <th>Ημ/νία Υποβολής</th>
                <th>Κατάσταση</th>
            </tr>
        </thead>
        <tbody>
            @foreach ($responses as $response)
```

```

    @php
        $isCorrect = $response->answer === $question->correct_option;
    @endphp
    <tr>
        <td>{{ $response->id }}</td>
        <td>{{ $response->am }}</td>
        <td>{{ $response->surname }}</td>
        <td>{{ $response->email }}</td>
        <td>{{ $response->answer }}</td>
        <td>{{ $response->ip_address }}</td>
        <td>{{ \Carbon\Carbon::parse($response->responded_at)->format('d-m-Y H:i') }}</td>
        <td>
            <span class="badge {{ $isCorrect ? 'bg-success' : 'bg-danger' }}">
                {{ $isCorrect ? 'Σωστό' : 'Λάθος' }}
            </span>
        </td>
    </tr>
@endforeach
</tbody>
</table>

<hr>
<h4><img alt="Bar chart icon" data-bbox="183 538 198 553"/> Στατιστικά Απαντήσεων</h4>
<canvas id="answersChart" width="400" height="200"></canvas>
</div>

<!-- DataTables + Buttons Extension -->
<link rel="stylesheet" href="https://cdn.datatables.net/1.13.4/css/jquery.dataTables.min.css">
<link rel="stylesheet" href="https://cdn.datatables.net/buttons/2.4.1/css/buttons.dataTables.min.css">
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script src="https://cdn.datatables.net/1.13.4/js/jquery.dataTables.min.js"></script>
<script src="https://cdn.datatables.net/buttons/2.4.1/js/dataTables.buttons.min.js"></script>
<script src="https://cdn.datatables.net/buttons/2.4.1/js/buttons.html5.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/jszip@3.10.0/dist/jszip.min.js"></script>

<!-- Chart.js -->
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<script>
    $(document).ready(function() {

```

```

$('#resultsTable').DataTable({
  language: {
    url: '//cdn.datatables.net/plug-ins/1.13.4/i18n/el.json'
  },
  dom: 'Bfrtip',
  buttons: [
    {
      extend: 'excelHtml5',
      text: '📄 Εξαγωγή σε Excel',
      title: 'Αποτελέσματα_Ερώτησης'
    }
  ]
});

const responses = @json($responses);
const correctOption = '{{ $question->correct_option }}';

const counts = {'A': 0, 'B': 0, 'C': 0, 'D': 0};
responses.forEach(res => {
  if (counts[res.answer] !== undefined) {
    counts[res.answer]++;
  }
});

const total = responses.length;
const labels = ['A', 'B', 'C', 'D'];
const dataValues = labels.map(opt => counts[opt]);
const percentages = dataValues.map(val => total ? (val / total * 100).toFixed(1) : 0);

const ctx = document.getElementById('answersChart').getContext('2d');
new Chart(ctx, {
  type: 'bar',
  data: {
    labels: labels,
    datasets: [{
      label: 'Απαντήσεις',
      data: dataValues,
      backgroundColor: ['#4CAF50', '#2196F3', '#FFC107', '#FF5722']
    }]
  },
});

```

```

options: {
  responsive: true,
  scales: {
    y: {
      beginAtZero: true,
      title: { display: true, text: 'Αριθμός Απαντήσεων' }
    }
  },
  plugins: {
    tooltip: {
      callbacks: {
        label: function(context) {
          const idx = context.dataIndex;
          return `${dataValues[idx]} απαντήσεις (${percentages[idx]}%`;
        }
      }
    },
    legend: { display: false },
    datalabels: {
      anchor: 'end',
      align: 'top',
      formatter: (value, ctx) => `${value} (${percentages[ctx.dataIndex]}%`
    }
  },
  plugins: [ChartDataLabels]
});
});
</script>

<!-- Chart.js Data Labels Plugin -->
<script src="https://cdn.jsdelivr.net/npm/chartjs-plugin-datalabels@2.2.0"></script>
@endsection

```

Questions index

```
@extends('layouts.app')
```

```
@section('content')
```

```

<div class="container mt-4">
  <h2>Λίστα Ερωτήσεων</h2>
  <a href="{{ route('teacher.questions.create') }}" class="btn btn-primary mb-3">✚ Νέα Ερώτηση</a>

  {{-- @if (session('success'))
    <div class="alert alert-success">{{ session('success') }}</div>
  @endif --}}

  <table id="questionsTable" class="table table-bordered table-striped">
    <thead>
      <tr>
        <th>#</th>
        <th>Ερώτηση</th>
        <th>Μάθημα</th>
        <th>Κωδικός</th>
        <th>Ισχύει Από</th>
        <th>Ισχύει Μέχρι</th>
        <th>Ενέργειες</th>
      </tr>
    </thead>
    <tbody>
      @foreach ($questions as $question)
        <tr>
          <td>{{ $question->id }}</td>
          <td>{{ Str::limit($question->question_text, 50) }}</td>
          <td>{{ $question->course_title }}</td>
          <td>{{ $question->code }}</td>
          <td>{{ \Carbon\Carbon::parse($question->starts_at)->format('d-m-Y H:i') }}</td>
          <td>{{ \Carbon\Carbon::parse($question->expires_at)->format('d-m-Y H:i') }}</td>
          <td>
            <a href="{{ route('teacher.questions.edit', $question->id) }}" class="btn btn-
warning btn-sm">
               Επεξεργασία
            </a>

            <form action="{{ route('teacher.questions.delete', $question->id) }}"
method="POST"
              style="display:inline-block;"
              onsubmit="return confirm('Είστε σίγουροι ότι θέλετε να διαγράψετε αυτή
την ερώτηση;');">
              @csrf

```

```

        <button type="submit" class="btn btn-danger btn-sm">🗑️ Διαγραφή</button>
    </form>

    <a href="{ route('teacher.questions.results', $question->id) }" class="btn
btn-info btn-sm">
        <img alt="Bar chart icon" data-bbox="328 185 342 198" style="vertical-align: middle;"/> Αποτελέσματα
    </a>
</td>
</tr>
</tbody>
@endforeach
</table>
</div>
<!-- DataTables για πιο όμορφη παρουσίαση -->
<link rel="stylesheet" href="https://cdn.datatables.net/1.13.4/css/jquery.dataTables.min.css">
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script src="https://cdn.datatables.net/1.13.4/js/jquery.dataTables.min.js"></script>

<script>
    $(document).ready(function() {
        $('#questionsTable').DataTable({
            language: {
                url: '//cdn.datatables.net/plug-ins/1.13.4/i18n/el.json'
            }
        });
    });
</script>
@endsection

```