

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη εφαρμογής διαχείρισης εργασιών μαθήματος»

Task Management ktsiak@ihu.gr (teacher) Home Courses Logout

[Dashboard](#) / [Course](#) / Task Details

Task Details

Εργασία 1

Εργασία 1 για Προγραμματισμό Ι

Due Date: 24-05-2024 21:38

Grade Submissions

Μάριος Τζώλης	6.00	Grade
Νίκος Πέτρου	5.00	Grade

Submissions

Student	ID	File	Submission Date	Actions
Μάριος Τζώλης	1	View File	17-05-2024 19:01	Delete
Μάριος Τζώλης	2	View File	17-05-2024 19:07	Delete
Νίκος Πέτρου	3	View File	18-05-2024 08:20	Delete
Νίκος Πέτρου	8	View File	18-05-2024 09:19	Delete
Νίκος Πέτρου	9	View File	18-05-2024 12:13	Delete

[Edit Task](#) [Delete Task](#)

Φοιτητής

ΜΑΡΙΟΣ ΤΖΩΛΗΣ
515144

Επιβλέπων

Δρ. Κυριάκος Τσιακμάκης

Σεπτέμβριος 2024

Ανάπτυξη εφαρμογής διαχείρισης εργασιών μαθήματος

Κωδικός: 23321

Φοιτητής: Τζώλης Μάριος

Εισηγητής: Δρ Κυριάκος Τσιακμάκης

Ημερομηνία ανάληψης Π.Ε. 05-11-2023

Ημερομηνία περάτωσης Π.Ε. 06-09-2024

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 **Μάριου Τζώλη** 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 σύστημα διαχείρισης εργασιών μαθημάτων που υλοποιήσαμε με τη χρήση του Laravel προσφέρει ένα ασφαλές και ευέλικτο περιβάλλον για διδάσκοντες και φοιτητές. Περιλαμβάνει διαχείριση χρηστών με ασφαλείς κωδικούς πρόσβασης, δημιουργία και διαχείριση μαθημάτων και εργασιών, καθώς και υποβολή και αξιολόγηση εργασιών. Η ενσωμάτωση σύγχρονων πρακτικών ασφαλείας, όπως hash, προστασία από SQL injection και CSRF tokens, διασφαλίζει την ακεραιότητα των δεδομένων. Το σύστημα προσφέρει φιλική προς τον χρήστη διεπαφή με τη χρήση του Bootstrap, καθιστώντας την πλοήγηση και τη χρήση του εύκολη και αποδοτική.

« Development of a coursework management application »

Abstract

The coursework management system we implemented using Laravel offers a secure and flexible environment for teachers and students. It includes managing users with secure passwords, creating and managing courses and assignments, and submitting and grading assignments. Incorporating modern security practices such as hashing, SQL injection protection and CSRF tokens ensures data integrity. The system offers a user-friendly interface using Bootstrap, making it easy and efficient to navigate and use.

Ευχαριστίες

Θα ήθελα να εκφράσω τις βαθύτερες ευχαριστίες μου στους γονείς μου για την αμέριστη υποστήριξή τους. Ιδιαίτερες ευχαριστίες απευθύνω στον επιβλέπων μου κ. Τσιακμάκη Κυριάκο για τη συνεχή παιδαγωγική καθοδήγηση, τις επιστημονικές οδηγίες και την ανεκτίμητη συνεισφορά του στον κώδικα εφαρμογής. Οι γνώσεις και η βοήθειά του ήταν καθοριστικές για την επιτυχή ολοκλήρωση αυτού του έργου.

Περιεχόμενα

Περίληψη	iv
Abstract	v
Ευχαριστίες	vi
Περιεχόμενα	vii
Κατάλογος Σχημάτων	viii
Κεφάλαιο 1ο: Εισαγωγή.....	9
1.1 Εισαγωγή.....	9
1.2 Δομή της εργασίας	11
Κεφάλαιο 2ο: Βιβλιογραφική Ανασκόπηση	13
2.1 Εισαγωγή.....	13
2.2 Moodle	14
2.3 Blackboard	17
Κεφάλαιο 3ο: Τεχνολογία και εργαλεία.....	19
3.1 PHP	19
3.2 Laravel	23
3.3 Bootstrap	26
3.4 MySQL.....	28
Κεφάλαιο 4ο: Το σύστημα διαχείρισης εργασιών μαθήματος.....	31
4.1 Περιγραφή διαδικασίας.....	31
4.1.1 Διαχείριση Χρηστών και Αυθεντικοποίηση.....	31
4.1.2 Διαχείριση Μαθημάτων και Εγγραφές.....	32
4.1.3 Διαχείριση Εργασιών και Υποβολές.....	33
4.1.4 Αξιολόγηση και Ανατροφοδότηση	34
4.2 Εισαγωγή – Η διαδικασία	34
4.3 Η Βάση.....	43
4.3.1 Δομή και Σχέσεις της Βάσης Δεδομένων για το Σύστημα Διαχείρισης Εργασιών Μαθήματος.....	43
4.3.2 Δομή και περιεχόμενα των πινάκων	47
4.3.3 Η βάση με το Laravel.....	50

4.4	Ασφάλεια στο σύστημα και στα δεδομένα.....	56
Κεφάλαιο 5ο:	Συμπεράσματα και προτάσεις βελτίωσης.....	62
BIBΛΙΟΓΡΑΦΙΑ.....		64
ΠΑΡΑΡΤΗΜΑ Α.....		66

Κατάλογος Σχημάτων

Εικόνα	3.1:	PHP	8
[https://qirolab.com/images/202011vFuWvJgQjJ0FdX2dsZb9zrva4HCpmUguqEfnGnp0.jpeg]			
Εικόνα 4.1:	Το σύστημα – η πρώτη σελίδα για είσοδο ή εγγραφή		35
Εικόνα 4.2:	Σελίδα Εγγραφής.....		35
Εικόνα 4.3:	Σελίδα για σύνδεση του χρήστη.....		36
Εικόνα 4.4:	Κεντρική σελίδα dashboard για παρακολούθηση μαθημάτων και εργασιών: Ρόλος Φοιτητής.....		37
Εικόνα 4.5:	Εγγραφή του φοιτητή σε κάποιο μάθημα αφού έχει στη διάθεση του τον κατάλληλο κωδικό μαθήματος που του έχει δοθεί από τον διδάσκοντα		37
Εικόνα 4.6:	Επιλογή ενός μαθήματος από την κεντρική σελίδα για προβολή των διαθέσιμων και εντός προθεσμιών εργασιών του μαθήματος. Ρόλος Φοιτητής		38
Εικόνα 4.7:	Προβολή των αρχείων, της βαθμολογίας και των χαρακτηριστικών για μια εργασία ενός μαθήματος. Ρόλος Φοιτητής.....		38
Εικόνα 4.8:	Προβολή μηνύματος για εκτός προθεσμίας για μια εργασία ενός μαθήματος. Ρόλος Φοιτητής.....		39
Εικόνα 4.9:	Σύνδεση του διδάσκοντα		40
Εικόνα 4.10:	Προβολή του dashboard για τον διδάσκοντα. Διαθέσιμα τα μαθήματα του και οι εργασίες τους με κάθε επιλογή, όπως προβολή, επεξεργασία και διαγραφή. Ρόλος Διάσκοντα		40
Εικόνα 4.11:	Επεξεργασία ενός μαθήματος από τον διδάσκοντα.....		41
Εικόνα 4.12:	Επεξεργασία μιας εργασίας ενός μαθήματος από τον διδάσκοντα. Καθορισμός των απαραίτητων πεδίων.		41
Εικόνα 4.13:	Προβολή και επεξεργασία των αρχείων και της βαθμολογίας μιας εργασίας για όλους τους φοιτητές που συμμετέχουν σε αυτή.		42
Εικόνα 4.14:	Η βάση με τους πίνακες		47
Εικόνα 4.15:	Η δομή του Πίνακα user.....		47
Εικόνα 4.16:	Τα περιεχόμενα του Πίνακα user.....		47
Εικόνα 4.17:	Η δομή του Πίνακα course		48
Εικόνα 4.18:	Τα περιεχόμενα του Πίνακα course		48
Εικόνα 4.19:	Η δομή του Πίνακα courseattend		48
Εικόνα 4.20:	Τα περιεχόμενα του Πίνακα courseattend		48

Εικόνα 4.21: Η δομή του Πίνακα coursetask	49
Εικόνα 4.22: Τα περιεχόμενα του Πίνακα coursetask	49
Εικόνα 4.23: Η δομή του Πίνακα coursetaskattend	49
Εικόνα 4.24: Τα περιεχόμενα του Πίνακα coursetaskattend	49
Εικόνα 4.25: Η δομή του Πίνακα files	50
Εικόνα 4.26: Τα περιεχόμενα του Πίνακα files	50

Κεφάλαιο 1ο: Εισαγωγή

1.1 Εισαγωγή

Το σύστημα διαχείρισης εργασιών μαθημάτων αποτελεί ένα κρίσιμο εργαλείο στη σύγχρονη εκπαιδευτική διαδικασία, επιτρέποντας στους διδάσκοντες και τους φοιτητές να οργανώνουν, να παρακολουθούν και να αξιολογούν τις ακαδημαϊκές εργασίες με αποτελεσματικότητα και ακρίβεια. Σε έναν ψηφιακά μεταβαλλόμενο κόσμο, η χρήση τέτοιων συστημάτων γίνεται όλο και πιο επιτακτική, καθώς προσφέρει τη δυνατότητα για ευέλικτη και διαδραστική μάθηση, ανεξαρτήτως γεωγραφικής τοποθεσίας.

Η εφαρμογή ενός συστήματος διαχείρισης εργασιών μαθημάτων παρουσιάζει πολλαπλά οφέλη. Διευκολύνει τη δημιουργία, την ανάθεση και την παρακολούθηση των εργασιών, παρέχοντας στους διδάσκοντες τη δυνατότητα να οργανώνουν και να διαχειρίζονται τις εκπαιδευτικές τους δραστηριότητες με δομημένο τρόπο. Οι φοιτητές, από την άλλη, μπορούν να παρακολουθούν τις προθεσμίες τους, να υποβάλλουν τις εργασίες τους ηλεκτρονικά και να λαμβάνουν ανατροφοδότηση σε πραγματικό χρόνο.

Το σύστημα χρησιμοποιείται ευρέως σε εκπαιδευτικά ιδρύματα, όπως πανεπιστήμια, σχολεία και επαγγελματικά κέντρα εκπαίδευσης, καθώς και σε εταιρικά περιβάλλοντα για την εκπαίδευση προσωπικού. Στα πανεπιστήμια, η ανάγκη για τη διαχείριση μεγάλου όγκου εργασιών και την παρακολούθηση της προόδου των φοιτητών καθιστά τέτοια συστήματα απαραίτητα. Βοηθούν στην εξασφάλιση της ακαδημαϊκής ακεραιότητας, στην αποδοτική διαχείριση του χρόνου και στην προαγωγή της συνεργασίας μεταξύ φοιτητών και διδασκόντων.

Η ανάπτυξη του συστήματος διαχείρισης εργασιών μαθημάτων απαιτεί τη χρήση σύγχρονων τεχνολογιών και εργαλείων για την εξασφάλιση της αποδοτικότητας, της ασφάλειας και της επεκτασιμότητας. Ένα από τα βασικά εργαλεία που χρησιμοποιήθηκαν είναι το Laravel, ένα PHP framework που επιτρέπει την ανάπτυξη ισχυρών και ευέλικτων web εφαρμογών. Το Laravel προσφέρει πλούσια χαρακτηριστικά, όπως το Eloquent ORM για την αλληλεπίδραση με τη βάση δεδομένων, το Blade templating engine για τη δημιουργία δυναμικών σελίδων και τις ισχυρές δυνατότητες routing για τη διαχείριση των αιτημάτων.

Η βάση δεδομένων του συστήματος υλοποιήθηκε χρησιμοποιώντας MySQL, ένα δημοφιλές σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων. Η MySQL επιτρέπει την αποθήκευση και την ανάκτηση δεδομένων με αποδοτικό και ασφαλή τρόπο, υποστηρίζοντας ταυτόχρονα τις πολύπλοκες σχέσεις μεταξύ των οντοτήτων του συστήματος, όπως οι χρήστες, τα μαθήματα και οι εργασίες.

Για το front-end μέρος της εφαρμογής, χρησιμοποιήθηκε το Bootstrap, ένα ισχυρό CSS framework που διευκολύνει τη δημιουργία φιλικών προς τον χρήστη και προσαρμόσιμων διεπαφών. Με τη χρήση του Bootstrap, η εφαρμογή αποκτά μια μοντέρνα και ανταποκρινόμενη σχεδίαση, εξασφαλίζοντας μια βέλτιστη εμπειρία χρήστη σε όλες τις συσκευές και τα μεγέθη οθόνης.

Η ασφάλεια της εφαρμογής διασφαλίστηκε μέσω της εφαρμογής βέλτιστων πρακτικών, όπως η χρήση hash για τους κωδικούς πρόσβασης, η προστασία από επιθέσεις SQL injection και CSRF, καθώς και η χρήση sessions για την αυθεντικοποίηση των χρηστών. Αυτές οι τεχνικές διασφαλίζουν την προστασία των ευαίσθητων δεδομένων και την ακεραιότητα των πληροφοριών.

Το σύστημα διαχείρισης εργασιών μαθημάτων αποτελεί ένα ουσιαστικό εργαλείο για την αποτελεσματική διαχείριση της εκπαιδευτικής διαδικασίας, ενσωματώνοντας προηγμένες τεχνολογίες και πρακτικές που εξασφαλίζουν την αποδοτικότητα και την ασφάλεια. Η υλοποίηση με τη χρήση του Laravel, της MySQL και του Bootstrap προσφέρει μια ευέλικτη και ισχυρή πλατφόρμα που ανταποκρίνεται στις ανάγκες των εκπαιδευτικών ιδρυμάτων και των φοιτητών, προωθώντας τη μάθηση και την ακαδημαϊκή αριστεία.

Η ανάπτυξη του συστήματος διαχείρισης εργασιών μαθημάτων συνεισφέρει σημαντικά στην εκπαιδευτική διαδικασία, προσφέροντας μια ολοκληρωμένη και ευέλικτη λύση που διευκολύνει τόσο τους διδάσκοντες όσο και τους φοιτητές. Η παρούσα εργασία αναλύει και υλοποιεί ένα σύστημα που παρέχει πολλαπλά οφέλη και βελτιώνει τη λειτουργικότητα των εκπαιδευτικών ιδρυμάτων.

Ένα από τα βασικά πλεονεκτήματα του συστήματος είναι η βελτίωση της οργάνωσης και διαχείρισης των μαθημάτων. Οι διδάσκοντες μπορούν εύκολα να δημιουργούν και να διαχειρίζονται μαθήματα, να αναθέτουν εργασίες και να παρακολουθούν την πρόοδο των φοιτητών. Η πλατφόρμα παρέχει ένα δομημένο περιβάλλον όπου οι διδάσκοντες μπορούν να ορίσουν προθεσμίες, να προσθέτουν κριτήρια αξιολόγησης και να παρέχουν ανατροφοδότηση στους φοιτητές. Αυτό διευκολύνει τη διαχείριση του φόρτου εργασίας και την παρακολούθηση της εκπαιδευτικής διαδικασίας.

Ένα από τα βασικά πλεονεκτήματα του συστήματος είναι η βελτίωση της οργάνωσης και διαχείρισης των μαθημάτων. Οι διδάσκοντες μπορούν εύκολα να δημιουργούν και να διαχειρίζονται μαθήματα, να αναθέτουν εργασίες και να παρακολουθούν την πρόοδο των φοιτητών. Η πλατφόρμα παρέχει ένα δομημένο περιβάλλον όπου οι διδάσκοντες μπορούν να ορίσουν προθεσμίες, να προσθέτουν κριτήρια αξιολόγησης και να παρέχουν ανατροφοδότηση στους φοιτητές. Αυτό διευκολύνει τη διαχείριση του φόρτου εργασίας και την παρακολούθηση της εκπαιδευτικής διαδικασίας.

Η αξιολόγηση των εργασιών και η παροχή ανατροφοδότησης είναι ζωτικής σημασίας για την ακαδημαϊκή πρόοδο των φοιτητών. Το σύστημα που υλοποιήθηκε επιτρέπει στους διδάσκοντες να αξιολογούν τις εργασίες με αντικειμενικό τρόπο, χρησιμοποιώντας ρουμπρικές και συγκεκριμένα κριτήρια αξιολόγησης. Οι φοιτητές λαμβάνουν αναλυτική ανατροφοδότηση που τους βοηθά να κατανοήσουν τα δυνατά και αδύνατα σημεία τους και να βελτιωθούν. Η δυνατότητα αποθήκευσης

όλων των υποβολών και των αξιολογήσεων ηλεκτρονικά διευκολύνει την παρακολούθηση της προόδου των φοιτητών και την εξαγωγή στατιστικών στοιχείων για τη βελτίωση της διδασκαλίας.

Η διασφάλιση της ακαδημαϊκής ακεραιότητας αποτελεί ένα από τα κύρια ζητήματα στην εκπαιδευτική διαδικασία. Η πλατφόρμα παρέχει εργαλεία για την ανίχνευση λογοκλοπής και την προστασία των δεδομένων των χρηστών. Χρησιμοποιώντας τεχνολογίες όπως η ανίχνευση λογοκλοπής και η κρυπτογράφηση δεδομένων, το σύστημα διασφαλίζει ότι οι εργασίες των φοιτητών είναι αυθεντικές και ότι οι προσωπικές πληροφορίες τους παραμένουν ασφαλείς. Αυτό ενισχύει την εμπιστοσύνη των φοιτητών και των διδασκόντων στο σύστημα και προάγει την ακαδημαϊκή αριστεία.

Η υλοποίηση του συστήματος με τη χρήση σύγχρονων τεχνολογιών όπως το Laravel, η MySQL και το Bootstrap προσφέρει μια ευέλικτη, επεκτάσιμη και φιλική προς τον χρήστη λύση. Το Laravel παρέχει ένα ισχυρό framework για την ανάπτυξη της εφαρμογής, ενώ η MySQL εξασφαλίζει την ασφαλή και αποδοτική διαχείριση των δεδομένων. Το Bootstrap προσφέρει μια μοντέρνα και ανταποκρινόμενη σχεδίαση που βελτιώνει την εμπειρία χρήστη. Αυτές οι τεχνολογίες επιτρέπουν την εύκολη συντήρηση και αναβάθμιση του συστήματος, καθιστώντας το ένα αξιόπιστο εργαλείο για τα εκπαιδευτικά ιδρύματα.

Η συνεισφορά της παρούσας εργασίας είναι πολυδιάστατη, καλύπτοντας την οργάνωση, τη συνεργασία, την αξιολόγηση, την ακαδημαϊκή ακεραιότητα και την τεχνολογική καινοτομία. Το σύστημα διαχείρισης εργασιών μαθημάτων προσφέρει ένα ολοκληρωμένο και αποδοτικό περιβάλλον που διευκολύνει την εκπαιδευτική διαδικασία και υποστηρίζει την ακαδημαϊκή πρόοδο των φοιτητών. Με την υιοθέτηση τέτοιων συστημάτων, τα εκπαιδευτικά ιδρύματα μπορούν να βελτιώσουν σημαντικά την ποιότητα της εκπαίδευσης και να ανταποκριθούν στις απαιτήσεις του σύγχρονου εκπαιδευτικού περιβάλλοντος.

1.2 Δομή της εργασίας

Το πρώτο κεφάλαιο παρέχει μια εισαγωγή στο θέμα της εργασίας και τη δομή της.

Το δεύτερο κεφάλαιο παρέχει μια βιβλιογραφική ανασκόπηση σχετικών συστημάτων διαχείρισης μάθησης.

Το τρίτο κεφάλαιο περιγράφει τις τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος.

Το τέταρτο κεφάλαιο περιγράφει την ανάπτυξη και τη λειτουργία του συστήματος διαχείρισης εργασιών μαθήματος.

Το πέμπτο κεφάλαιο παρουσιάζει τα συμπεράσματα της εργασίας και προτείνει βελτιώσεις για μελλοντική ανάπτυξη. Περιλαμβάνει ανάλυση των αποτελεσμάτων και προτάσεις για την περαιτέρω βελτίωση του συστήματος.

Η βιβλιογραφία παρέχει μια λίστα με όλες τις πηγές που χρησιμοποιήθηκαν στην εργασία και βρίσκεται στην τελευταία ενότητα της εργασίας.

Κεφάλαιο 2ο: Βιβλιογραφική Ανασκόπηση

2.1 Εισαγωγή

Η διαχείριση εργασιών μαθήματος αποτελεί έναν από τους κεντρικούς πυλώνες της εκπαιδευτικής διαδικασίας, ειδικά σε ψηφιακά περιβάλλοντα μάθησης. Οι πλατφόρμες και οι εφαρμογές διαχείρισης εργασιών έχουν αναδειχθεί ως κρίσιμα εργαλεία για τη διευκόλυνση της αλληλεπίδρασης μεταξύ διδασκόντων και φοιτητών, την οργάνωση της διδακτικής ύλης, την παρακολούθηση της προόδου και την αξιολόγηση των επιδόσεων. Οι πλατφόρμες διαχείρισης εργασιών μαθήματος παρέχουν μια πληθώρα λειτουργιών που διευκολύνουν την εκπαιδευτική διαδικασία. Οι βασικές λειτουργίες περιλαμβάνουν τη δημιουργία και διαχείριση μαθημάτων και εργασιών, την υποβολή και αξιολόγηση των εργασιών, την παροχή ανατροφοδότησης και την επικοινωνία μεταξύ διδασκόντων και φοιτητών.

Οι διδάσκοντες μπορούν να δημιουργούν και να οργανώνουν μαθήματα, να ορίζουν τις εργασίες, και να θέτουν προθεσμίες. Η πλατφόρμα επιτρέπει την ευέλικτη διαχείριση του εκπαιδευτικού υλικού και τη σύνδεσή του με συγκεκριμένα μαθήματα και εργασίες. Αυτή η δυνατότητα βοηθά στην τυποποίηση και την οργάνωση της διδασκαλίας, ενώ παράλληλα διευκολύνει την πρόσβαση των φοιτητών στο υλικό.

Οι φοιτητές μπορούν να υποβάλλουν τις εργασίες τους ηλεκτρονικά μέσω της πλατφόρμας, επιτρέποντας στους διδάσκοντες να αξιολογήσουν και να βαθμολογήσουν τις εργασίες απευθείας. Η παροχή ανατροφοδότησης μέσω της πλατφόρμας διευκολύνει τη συνεχή βελτίωση των φοιτητών, ενώ η αυτόματη διατήρηση αρχείου των υποβολών και των βαθμολογιών εξασφαλίζει την ακεραιότητα των δεδομένων.

Οι πλατφόρμες διαχείρισης εργασιών μαθήματος συχνά περιλαμβάνουν εργαλεία επικοινωνίας, όπως μηνύματα, φόρουμ και σχόλια, που διευκολύνουν την επικοινωνία μεταξύ διδασκόντων και φοιτητών. Αυτά τα εργαλεία ενισχύουν την συνεργατικότητα και επιτρέπουν την άμεση επίλυση αποριών και προβλημάτων. Η ανάπτυξη και η υλοποίηση των πλατφορμών διαχείρισης εργασιών μαθήματος βασίζεται σε σύγχρονες τεχνολογίες και βέλτιστες πρακτικές ανάπτυξης λογισμικού. Η χρήση frameworks όπως το Laravel επιτρέπει την αποτελεσματική και ασφαλή ανάπτυξη της εφαρμογής.

Η ασφάλεια των δεδομένων είναι ζωτικής σημασίας για τις πλατφόρμες διαχείρισης εργασιών μαθήματος. Μέτρα όπως η χρήση hash για τους κωδικούς πρόσβασης, η προστασία από SQL injection και η χρήση CSRF tokens διασφαλίζουν την ακεραιότητα και την εμπιστευτικότητα των δεδομένων των χρηστών.

Οι σύγχρονες πλατφόρμες είναι σχεδιασμένες να είναι επεκτάσιμες και ευέλικτες, επιτρέποντας την εύκολη προσαρμογή και προσθήκη νέων λειτουργιών. Η χρήση του Eloquent ORM και του Query Builder της Laravel διευκολύνει την ανάπτυξη και τη συντήρηση της βάσης δεδομένων, ενώ τα migrations επιτρέπουν την οργάνωση και την παρακολούθηση των αλλαγών στη βάση δεδομένων.

Πλατφόρμες όπως το Moodle και το Blackboard έχουν καθιερωθεί ως κορυφαία εργαλεία στην κατηγορία τους, παρέχοντας ολοκληρωμένες λύσεις για τη διαχείριση της εκπαιδευτικής διαδικασίας. Αυτές οι πλατφόρμες προσφέρουν μια πληθώρα λειτουργιών που καλύπτουν όλες τις ανάγκες των εκπαιδευτικών και των φοιτητών, από την παρακολούθηση της προόδου έως την αξιολόγηση των επιδόσεων. [1-6]

Moodle

Το Moodle είναι μια ανοικτού κώδικα πλατφόρμα μάθησης που χρησιμοποιείται ευρέως σε σχολεία, πανεπιστήμια και άλλους εκπαιδευτικούς οργανισμούς. Παρέχει εργαλεία για τη δημιουργία και τη διαχείριση μαθημάτων, την υποβολή εργασιών, τη συνεργασία και την επικοινωνία. Η ευελιξία του Moodle και η υποστήριξη πολλών πρόσθετων το καθιστούν μια από τις πιο δημοφιλείς πλατφόρμες μάθησης.

Blackboard

Το Blackboard είναι μια εμπορική πλατφόρμα μάθησης που προσφέρει ολοκληρωμένες λύσεις για την εκπαίδευση. Παρέχει εργαλεία για τη διαχείριση μαθημάτων και εργασιών, την αξιολόγηση και την ανατροφοδότηση, καθώς και εργαλεία επικοινωνίας και συνεργασίας. Το Blackboard χρησιμοποιείται από πολλά πανεπιστήμια και εκπαιδευτικούς οργανισμούς παγκοσμίως.

2.2 Moodle

Το Moodle είναι μια ανοικτού κώδικα πλατφόρμα μάθησης, που σχεδιάστηκε για να προσφέρει εκπαιδευτικές λύσεις σε σχολεία, πανεπιστήμια, και άλλους εκπαιδευτικούς οργανισμούς. Αναπτύχθηκε το 2002 από τον Martin Dougiamas, και έκτοτε έχει εξελιχθεί σε ένα από τα πιο δημοφιλή εργαλεία διαχείρισης μάθησης παγκοσμίως. Το όνομα Moodle προέρχεται από τα αρχικά "Modular Object-Oriented Dynamic Learning Environment", κάτι που αναδεικνύει τη δυνατότητα προσαρμογής και ευελιξίας της πλατφόρμας. Η αρχιτεκτονική του Moodle επιτρέπει στους χρήστες να προσθέτουν και να διαμορφώνουν διάφορα modules και plugins, κάνοντάς το ένα εξαιρετικά παραμετροποιήσιμο εργαλείο που μπορεί να καλύψει ένα ευρύ φάσμα εκπαιδευτικών αναγκών.

Το Moodle παρέχει ένα εκτενές σύνολο δυνατοτήτων που διευκολύνουν την εκπαιδευτική διαδικασία. Αυτές περιλαμβάνουν τη δημιουργία και διαχείριση μαθημάτων, την ανάθεση και αξιολόγηση εργασιών, τη διεξαγωγή διαδικτυακών διαλέξεων και την παρακολούθηση της προόδου των φοιτητών. Οι διδάσκοντες μπορούν να δημιουργούν διαδραστικά μαθήματα με πολυμέσα, όπως βίντεο και

παρουσιάσεις, και να ενσωματώνουν δραστηριότητες όπως κουίζ, συζητήσεις και εργασίες. Το σύστημα υποστηρίζει επίσης τη συνεργατική μάθηση, επιτρέποντας στους φοιτητές να συνεργάζονται μέσω φόρουμ, ομαδικών εργασιών και wiki. Αυτές οι λειτουργίες καθιστούν το Moodle ένα ολοκληρωμένο εργαλείο μάθησης που προάγει την ενεργή συμμετοχή και την αλληλεπίδραση των φοιτητών.

Ένα από τα μεγαλύτερα πλεονεκτήματα του Moodle είναι η φύση του ως λογισμικό ανοικτού κώδικα. Αυτό σημαίνει ότι ο πηγαίος κώδικας είναι διαθέσιμος δωρεάν και μπορεί να τροποποιηθεί από οποιονδήποτε. Η μεγάλη και ενεργή κοινότητα των χρηστών και προγραμματιστών του Moodle συμβάλλει συνεχώς στη βελτίωση και την επέκταση της πλατφόρμας. Χιλιάδες εκπαιδευτικοί, προγραμματιστές και οργανισμοί συνεργάζονται για την ανάπτυξη νέων λειτουργιών, τη διόρθωση σφαλμάτων και την παροχή υποστήριξης. Αυτό το συλλογικό πνεύμα συνεργασίας και η συνεχής ανατροφοδότηση από την κοινότητα διασφαλίζουν ότι το Moodle παραμένει ένα σύγχρονο και αξιόπιστο εργαλείο μάθησης.

Η προσαρμοστικότητα του Moodle είναι ένα από τα βασικά χαρακτηριστικά που το ξεχωρίζουν. Οι εκπαιδευτικοί οργανισμοί μπορούν να προσαρμόσουν την πλατφόρμα στις δικές τους ανάγκες, επιλέγοντας από χιλιάδες διαθέσιμα plugins και themes. Αυτή η ευελιξία επιτρέπει τη δημιουργία εξατομικευμένων εκπαιδευτικών περιβαλλόντων που ανταποκρίνονται στις συγκεκριμένες απαιτήσεις κάθε οργανισμού. Επιπλέον, η πλατφόρμα είναι επεκτάσιμη και μπορεί να υποστηρίξει μεγάλους αριθμούς χρηστών και μαθημάτων χωρίς να επηρεάζεται η απόδοση. Αυτό καθιστά το Moodle ιδανικό για εκπαιδευτικούς οργανισμούς κάθε μεγέθους, από μικρά σχολεία μέχρι μεγάλα πανεπιστήμια και εταιρικές εκπαιδευτικές μονάδες.

Η ασφάλεια των δεδομένων και η προστασία της ιδιωτικότητας είναι κρίσιμα ζητήματα για κάθε πλατφόρμα μάθησης, και το Moodle δεν αποτελεί εξαίρεση. Η πλατφόρμα είναι σχεδιασμένη για να πληροί αυστηρά πρότυπα ασφαλείας και συμμόρφωσης με κανονισμούς όπως το Γενικό Κανονισμό για την Προστασία Δεδομένων (GDPR). Το Moodle παρέχει εργαλεία για τη διαχείριση των δικαιωμάτων πρόσβασης και τη διασφάλιση της ασφαλείας των δεδομένων των χρηστών. Τα δεδομένα κρυπτογραφούνται και τα συστήματα ασφαλείας ενημερώνονται τακτικά για να προστατεύουν από νέες απειλές. Αυτές οι πρακτικές διασφαλίζουν ότι οι πληροφορίες των χρηστών είναι προστατευμένες και ότι η πλατφόρμα μπορεί να χρησιμοποιηθεί με εμπιστοσύνη από εκπαιδευτικούς οργανισμούς παγκοσμίως.

Η δυνατότητα ανάθεσης εργασιών σε φοιτητές είναι μία από τις πιο σημαντικές και ευέλικτες λειτουργίες του Moodle, επιτρέποντας στους διδάσκοντες να ενσωματώσουν μια μεγάλη ποικιλία εκπαιδευτικών δραστηριοτήτων και αξιολογήσεων στην εκπαιδευτική διαδικασία. Οι εργασίες στο Moodle μπορούν να διαμορφωθούν με τέτοιο τρόπο ώστε να καλύπτουν τις ανάγκες διάφορων μαθησιακών στόχων και να προάγουν την ενεργό συμμετοχή των φοιτητών. Από απλές γραπτές

εργασίες έως πολύπλοκα πρότζεκτ ομαδικής συνεργασίας, οι δυνατότητες είναι σχεδόν απεριόριστες. Οι διδάσκοντες μπορούν να δημιουργούν εργασίες που απαιτούν από τους φοιτητές να υποβάλλουν γραπτά έγγραφα, παρουσιάσεις, βίντεο, αρχεία ή ακόμα και να συμπληρώνουν online φόρμες και ερωτηματολόγια. Η πλατφόρμα υποστηρίζει επίσης τη χρήση διαδραστικών δραστηριοτήτων, όπως κουίζ, τα οποία μπορούν να αξιοποιηθούν για την άμεση αξιολόγηση της κατανόησης των φοιτητών σε συγκεκριμένα θέματα.

Ένα από τα βασικά χαρακτηριστικά της διαχείρισης εργασιών στο Moodle είναι η δυνατότητα για την καθορισμό προθεσμιών και κριτηρίων αξιολόγησης. Οι διδάσκοντες μπορούν να ορίσουν συγκεκριμένες ημερομηνίες και ώρες υποβολής, δίνοντας έτσι στους φοιτητές σαφή εικόνα των προσδοκιών και των χρονοδιαγραμμάτων. Επιπλέον, μπορούν να προσθέσουν κριτήρια αξιολόγησης ή ρουμπρίκες, οι οποίες επιτρέπουν μια πιο αντικειμενική και δομημένη αξιολόγηση των εργασιών. Οι ρουμπρίκες μπορούν να περιλαμβάνουν διάφορους παράγοντες αξιολόγησης, όπως η πρωτοτυπία, η ακρίβεια, η πληρότητα, η γραμματική και η παρουσίαση, παρέχοντας έτσι αναλυτική ανατροφοδότηση στους φοιτητές. Η δυνατότητα προσθήκης σχολίων και σημειώσεων στις υποβληθείσες εργασίες ενισχύει περαιτέρω την ποιοτική ανατροφοδότηση και βοηθά τους φοιτητές να κατανοήσουν τα δυνατά και αδύνατα σημεία τους.

Η διαχείριση των υποβολών εργασιών στο Moodle είναι επίσης πολύ ευέλικτη. Οι διδάσκοντες μπορούν να επιτρέπουν πολλαπλές υποβολές, κάτι που δίνει στους φοιτητές τη δυνατότητα να βελτιώσουν και να επανυποβάλουν τις εργασίες τους μέχρι την τελική προθεσμία. Επιπλέον, το Moodle υποστηρίζει διάφορους τύπους αρχείων για τις υποβολές, επιτρέποντας στους φοιτητές να χρησιμοποιήσουν το εργαλείο που τους εξυπηρετεί καλύτερα. Μια άλλη σημαντική δυνατότητα είναι η ανίχνευση λογοκλοπής, η οποία μπορεί να ενσωματωθεί στην πλατφόρμα για να διασφαλιστεί η ακαδημαϊκή ακεραιότητα. Τα εργαλεία ανίχνευσης λογοκλοπής μπορούν να αναλύσουν τις υποβολές και να συγκρίνουν το περιεχόμενο με άλλες πηγές, παρέχοντας αναφορές στους διδάσκοντες για την αυθεντικότητα των εργασιών.

Η συνεργατική μάθηση είναι επίσης ένα σημαντικό στοιχείο που μπορεί να ενσωματωθεί στις εργασίες του Moodle. Οι διδάσκοντες μπορούν να δημιουργούν ομαδικές εργασίες, όπου οι φοιτητές συνεργάζονται σε κοινά πρότζεκτ. Αυτή η δυνατότητα ενισχύει τις δεξιότητες συνεργασίας και επικοινωνίας των φοιτητών και τους προετοιμάζει για πραγματικά περιβάλλοντα εργασίας. Οι ομαδικές εργασίες μπορούν να περιλαμβάνουν κοινές υποβολές, φόρουμ συζητήσεων και συνεργατικά έγγραφα, τα οποία διευκολύνουν την ανταλλαγή ιδεών και την από κοινού επίλυση προβλημάτων. Οι διδάσκοντες έχουν τη δυνατότητα να παρακολουθούν την πρόοδο των ομάδων και να παρεμβαίνουν όταν χρειάζεται, προσφέροντας καθοδήγηση και υποστήριξη.

Η παρακολούθηση της προόδου των φοιτητών είναι επίσης ένα σημαντικό πλεονέκτημα των εργασιών στο Moodle. Οι διδάσκοντες μπορούν να δουν ποιοι φοιτητές έχουν υποβάλει τις εργασίες τους, να

παρακολουθήσουν την επίδοση και τη συμμετοχή τους και να αναγνωρίσουν έγκαιρα τυχόν προβλήματα ή δυσκολίες που μπορεί να αντιμετωπίζουν. Η πλατφόρμα παρέχει λεπτομερείς αναφορές και στατιστικά στοιχεία, τα οποία μπορούν να χρησιμοποιηθούν για τη βελτίωση της διδασκαλίας και της μάθησης. Η δυνατότητα για συνεχή ανατροφοδότηση και επικοινωνία μέσω των εργαλείων του Moodle συμβάλλει στην ενίσχυση της εκπαιδευτικής διαδικασίας και στη δημιουργία ενός υποστηρικτικού μαθησιακού περιβάλλοντος.

Οι εργασίες στο Moodle αποτελούν ένα πολυδιάστατο εργαλείο που ενσωματώνει πολλές και ποικίλες εκπαιδευτικές δραστηριότητες. Από την απλή ανάθεση γραπτών εργασιών μέχρι τη σύνθετη διαχείριση ομαδικών πρότζεκτ και τη χρήση διαδραστικών δραστηριοτήτων, το Moodle προσφέρει όλα τα απαραίτητα εργαλεία για την ενίσχυση της μάθησης και την αποτελεσματική αξιολόγηση των φοιτητών. Η ευελιξία και η προσαρμοστικότητα της πλατφόρμας την καθιστούν ιδανική για κάθε είδους εκπαιδευτικό περιβάλλον, προάγοντας την ενεργή συμμετοχή και την ακαδημαϊκή αριστεία.

2.3 Blackboard

Το Blackboard είναι μια εμπορική πλατφόρμα μάθησης που έχει καταστεί ένα από τα πιο αναγνωρισμένα και ευρέως χρησιμοποιούμενα εργαλεία διαχείρισης μάθησης παγκοσμίως. Αναπτύχθηκε με στόχο να προσφέρει ολοκληρωμένες λύσεις για την υποστήριξη της εκπαιδευτικής διαδικασίας σε σχολεία, πανεπιστήμια, και εταιρικά περιβάλλοντα. Το Blackboard παρέχει ένα πλήρες σύνολο εργαλείων που καλύπτουν τις ανάγκες τόσο των διδασκόντων όσο και των μαθητών, προσφέροντας ευελιξία, ασφάλεια και επεκτασιμότητα. Η πλατφόρμα έχει σχεδιαστεί για να είναι εύχρηστη και προσβάσιμη, υποστηρίζοντας πολυγλωσσικά περιβάλλοντα και διάφορες μορφές εκπαίδευσης, συμπεριλαμβανομένης της παραδοσιακής, της εξ αποστάσεως και της μικτής μάθησης.

Το Blackboard προσφέρει μια πληθώρα δυνατοτήτων που διευκολύνουν την εκπαιδευτική διαδικασία και την αλληλεπίδραση μεταξύ διδασκόντων και μαθητών. Οι διδάσκοντες μπορούν να δημιουργούν και να διαχειρίζονται μαθήματα, να αναθέτουν εργασίες, να διεξάγουν διαδικτυακές διαλέξεις και να παρέχουν ανατροφοδότηση στους μαθητές. Οι μαθητές μπορούν να υποβάλλουν τις εργασίες τους ηλεκτρονικά, να συμμετέχουν σε διαδραστικές δραστηριότητες όπως φόρουμ και συζητήσεις, και να λαμβάνουν ανατροφοδότηση σε πραγματικό χρόνο. Η πλατφόρμα υποστηρίζει επίσης τη διαχείριση βαθμολογιών και την παρακολούθηση της προόδου των μαθητών, προσφέροντας λεπτομερείς αναφορές και στατιστικά στοιχεία που βοηθούν τους διδάσκοντες να αναγνωρίσουν έγκαιρα τυχόν προβλήματα ή δυσκολίες.

Η ασφάλεια των δεδομένων και η συμμόρφωση με τους κανονισμούς είναι από τις κορυφαίες προτεραιότητες του Blackboard. Η πλατφόρμα είναι σχεδιασμένη για να πληροί αυστηρά πρότυπα

ασφαλείας και να συμμορφώνεται με κανονισμούς όπως ο Γενικός Κανονισμός για την Προστασία Δεδομένων (GDPR). Το Blackboard παρέχει εργαλεία για τη διαχείριση των δικαιωμάτων πρόσβασης, την κρυπτογράφηση των δεδομένων και την προστασία των προσωπικών πληροφοριών των χρηστών. Η τακτική ενημέρωση των συστημάτων ασφαλείας και η ενσωμάτωση τεχνολογιών όπως το Single Sign-On (SSO) και η διπλή ταυτοποίηση (2FA) διασφαλίζουν την προστασία των δεδομένων από κακόβουλες επιθέσεις και παραβιάσεις.

Η προσαρμοστικότητα και η επεκτασιμότητα είναι από τα κύρια χαρακτηριστικά του Blackboard. Η πλατφόρμα επιτρέπει στους εκπαιδευτικούς οργανισμούς να προσαρμόσουν τη λειτουργικότητα και την εμφάνισή της σύμφωνα με τις δικές τους ανάγκες και απαιτήσεις. Οι χρήστες μπορούν να επιλέξουν από μια πληθώρα διαθέσιμων plugins και modules που επεκτείνουν τις δυνατότητες του Blackboard, προσθέτοντας νέα εργαλεία και λειτουργίες που βελτιώνουν την εκπαιδευτική εμπειρία. Επιπλέον, το Blackboard μπορεί να υποστηρίξει μεγάλο αριθμό χρηστών και μαθημάτων, καθιστώντας το ιδανικό για εκπαιδευτικούς οργανισμούς κάθε μεγέθους, από μικρά σχολεία μέχρι μεγάλα πανεπιστήμια και πολυεθνικές εταιρείες.

Το Blackboard υποστηρίζεται από μια μεγάλη και ενεργή κοινότητα χρηστών και επαγγελματιών. Η πλατφόρμα παρέχει εκτενή τεκμηρίωση, online forums, και εκπαιδευτικά σεμινάρια που βοηθούν τους χρήστες να αξιοποιήσουν στο έπακρο τις δυνατότητες του συστήματος. Επιπλέον, η Blackboard Inc. προσφέρει επαγγελματική υποστήριξη και συμβουλευτικές υπηρεσίες, βοηθώντας τους εκπαιδευτικούς οργανισμούς να υιοθετήσουν και να προσαρμόσουν την πλατφόρμα στις δικές τους ανάγκες. Η συνεχής ανατροφοδότηση από την κοινότητα και οι τακτικές ενημερώσεις λογισμικού διασφαλίζουν ότι η πλατφόρμα παραμένει σύγχρονη και ανταποκρίνεται στις συνεχώς μεταβαλλόμενες ανάγκες της εκπαιδευτικής κοινότητας.

Το Blackboard αποτελεί μια ολοκληρωμένη και αξιόπιστη λύση διαχείρισης μάθησης που προσφέρει ευελιξία, ασφάλεια και επεκτασιμότητα. Με τις πολυάριθμες δυνατότητες και τη συνεχή υποστήριξη, η πλατφόρμα διευκολύνει την εκπαιδευτική διαδικασία και ενισχύει την αλληλεπίδραση και την εμπλοκή των μαθητών, καθιστώντας το ιδανικό εργαλείο για εκπαιδευτικούς οργανισμούς σε όλο τον κόσμο.

Κεφάλαιο 3ο: Τεχνολογία και εργαλεία

Στο κεφάλαιο αυτό θα περιγράψει η τεχνολογία που χρησιμοποιήθηκε για την υλοποίηση του συστήματος.

3.1 PHP

Η PHP (Hypertext Preprocessor) είναι μια δημοφιλής γλώσσα προγραμματισμού που χρησιμοποιείται κυρίως για την ανάπτυξη δυναμικών ιστοσελίδων και web εφαρμογών. Είναι μια γλώσσα διακομιστή (server-side), που σημαίνει ότι ο κώδικας εκτελείται στον διακομιστή και τα αποτελέσματα αποστέλλονται στον περιηγητή του χρήστη. Η PHP είναι γνωστή για την ευκολία χρήσης της, την ισχυρή κοινότητα υποστήριξης και την ενσωμάτωση με πολλές βάσεις δεδομένων και web servers. [7-8]



Εικόνα 3.1: PHP 8

[<https://qirolab.com/images/202011vFuWvJgQjJ0FdX2dsZb9zpva4HCpmUguqEfnGnp0.jpeg>]

Η PHP δημιουργήθηκε αρχικά από τον Rasmus Lerdorf το 1994 ως ένα απλό σύνολο εργαλείων γραμμένο σε Perl, το οποίο αργότερα επανέγραψε σε C για καλύτερη απόδοση και επέκταση. Το 1995, ο Lerdorf κυκλοφόρησε τη PHP/FI (Personal Home Page / Forms Interpreter), την πρώτη έκδοση της PHP. Από τότε, η PHP έχει εξελιχθεί σημαντικά, με την υποστήριξη πολλών προγραμματιστών και την εισαγωγή νέων χαρακτηριστικών και βελτιώσεων σε κάθε έκδοση.

Η PHP έχει υποστεί πολλές αναβαθμίσεις και βελτιώσεις από την αρχική της έκδοση. Μερικές από τις σημαντικότερες εκδόσεις περιλαμβάνουν:

PHP 3 (1998): Αυτή η έκδοση ήταν η πρώτη που υιοθέτησε η κοινότητα και περιελάμβανε υποστήριξη για μια πιο δομημένη και εκτεταμένη γλώσσα.

PHP 4 (2000): Εισήγαγε το Zend Engine, έναν νέο κινητήρα εκτέλεσης, βελτιώνοντας σημαντικά την απόδοση.

PHP 5 (2004): Έφερε υποστήριξη για αντικειμενοστραφή προγραμματισμό και πρόσθεσε πολλά νέα χαρακτηριστικά, όπως τη βελτίωση της διαχείρισης εξαιρέσεων.

PHP 7 (2015): Προσέφερε μεγάλες βελτιώσεις στην απόδοση και την αποδοτικότητα της μνήμης, καθιστώντας την PHP πιο γρήγορη και ελαφριά.

PHP 8 (2020): Εισήγαγε νέα χαρακτηριστικά, όπως το JIT (Just In Time) compiler, βελτιώνοντας περαιτέρω την απόδοση και προσθέτοντας σύγχρονες δυνατότητες γλώσσας.

Απόδοση της PHP

Η απόδοση της PHP έχει βελτιωθεί σημαντικά με τις πρόσφατες εκδόσεις, ιδιαίτερα με την εισαγωγή του PHP 7 και του PHP 8. Οι βελτιώσεις περιλαμβάνουν ταχύτερη εκτέλεση κώδικα, μειωμένη κατανάλωση μνήμης και την εισαγωγή του JIT compiler στην PHP 8, ο οποίος επιτρέπει την εκτέλεση του κώδικα σε πραγματικό χρόνο, βελτιώνοντας την απόδοση σε εφαρμογές υψηλών απαιτήσεων.

Χρήση της PHP

Η χρήση της PHP είναι απλή και ευέλικτη. Μπορείτε να ενσωματώσετε PHP κώδικα σε HTML αρχεία, επιτρέποντάς σας να δημιουργείτε δυναμικές ιστοσελίδες εύκολα.

```
<!DOCTYPE html>

<html>

<head>

  <title>PHP Example</title>

</head>

<body>
```

```
<h1>Welcome to my website</h1>

<?php
echo "Hello, world!";
?>

</body>
</html>
```

Για να εκτελέσετε αυτόν τον κώδικα, πρέπει να έχετε εγκατεστημένο έναν διακομιστή web που υποστηρίζει PHP, όπως το Apache ή το Nginx, και να αποθηκεύσετε το αρχείο με την κατάληξη .php.

Υπάρχουν πολλοί λόγοι για τους οποίους κάποιος μπορεί να προτιμήσει την PHP για την ανάπτυξη web εφαρμογών.

Η PHP είναι εύκολη στη μάθηση και τη χρήση, ακόμα και για αρχάριους προγραμματιστές.

Η PHP μπορεί να χρησιμοποιηθεί για την ανάπτυξη μικρών ιστοσελίδων έως μεγάλων, πολύπλοκων web εφαρμογών.

Υπάρχει μια μεγάλη και ενεργή κοινότητα προγραμματιστών PHP που προσφέρει υποστήριξη και πλήθος βιβλιοθηκών και εργαλείων. Η PHP ενσωματώνεται εύκολα με πολλές βάσεις δεδομένων (όπως MySQL, PostgreSQL) και web servers (όπως Apache, Nginx).

Οι πρόσφατες εκδόσεις της PHP έχουν βελτιώσει σημαντικά την απόδοση, καθιστώντας την κατάλληλη για εφαρμογές υψηλών απαιτήσεων.

Η PHP είναι ελεύθερη και ανοικτού κώδικα, καθιστώντας την προσβάσιμη για όλους χωρίς κόστος.

Η επιλογή της κατάλληλης γλώσσας προγραμματισμού για την ανάπτυξη ενός συστήματος διαχείρισης εργασιών μαθημάτων είναι ζωτικής σημασίας για την επιτυχία του έργου. Η PHP αποτελεί μια ιδανική επιλογή για την υλοποίηση αυτού του συστήματος λόγω των πολλαπλών πλεονεκτημάτων που προσφέρει. Σε αυτό το κεφάλαιο, θα αναλύσουμε τους κύριους λόγους για τους οποίους η PHP είναι η κατάλληλη γλώσσα για την ανάπτυξη του συστήματος διαχείρισης εργασιών μαθημάτων.

Η PHP είναι γνωστή για την απλότητά της, καθιστώντας την εύκολη στη μάθηση και τη χρήση, ακόμα και για αρχάριους προγραμματιστές. Η σύνταξη της PHP είναι απλή και κατανοητή, επιτρέποντας στους προγραμματιστές να αναπτύξουν γρήγορα λειτουργικές web εφαρμογές. Αυτή η ευκολία στη χρήση επιτρέπει στους προγραμματιστές να επικεντρωθούν περισσότερο στη λογική της εφαρμογής και λιγότερο στις τεχνικές λεπτομέρειες της γλώσσας.

Η PHP προσφέρει εξαιρετική υποστήριξη για ενσωμάτωση με πολλές δημοφιλείς βάσεις δεδομένων, όπως MySQL, PostgreSQL, SQLite, και άλλες. Η ευκολία με την οποία μπορεί να συνδεθεί με μια βάση δεδομένων καθιστά την PHP ιδανική για την ανάπτυξη συστημάτων διαχείρισης δεδομένων, όπως ένα σύστημα διαχείρισης εργασιών μαθημάτων. Οι ενσωματωμένες συναρτήσεις της PHP για διαχείριση βάσεων δεδομένων επιτρέπουν την εύκολη αποθήκευση, ανάκτηση και διαχείριση δεδομένων φοιτητών, εργασιών και μαθημάτων.

Η PHP είναι εξαιρετικά ευέλικτη και μπορεί να χρησιμοποιηθεί για την ανάπτυξη από μικρές ιστοσελίδες έως μεγάλες και πολύπλοκες web εφαρμογές. Το σύστημα διαχείρισης εργασιών μαθημάτων μπορεί να ξεκινήσει με βασικές λειτουργίες και να επεκταθεί σταδιακά με νέα χαρακτηριστικά και βελτιώσεις, χωρίς να απαιτούνται ριζικές αλλαγές στην αρχιτεκτονική του. Η επεκτασιμότητα της PHP εξασφαλίζει ότι το σύστημα μπορεί να αναπτυχθεί παράλληλα με τις αυξανόμενες ανάγκες του εκπαιδευτικού ιδρύματος.

Η PHP διαθέτει μια μεγάλη και ενεργή κοινότητα προγραμματιστών που παρέχει συνεχή υποστήριξη και συνεισφέρει στην ανάπτυξη της γλώσσας και των εργαλείων της. Υπάρχει πληθώρα δωρεάν βιβλιοθηκών, εργαλείων και πηγών που μπορούν να χρησιμοποιηθούν για την επιτάχυνση της ανάπτυξης του συστήματος διαχείρισης εργασιών μαθημάτων. Αυτή η ισχυρή κοινότητα σημαίνει ότι οποιοδήποτε πρόβλημα ή απορία μπορεί να επιλυθεί γρήγορα μέσω των διαθέσιμων πόρων.

Οι πρόσφατες εκδόσεις της PHP, ειδικά η PHP 7 και η PHP 8, έχουν φέρει σημαντικές βελτιώσεις στην απόδοση και την ασφάλεια. Η PHP είναι πλέον πολύ ταχύτερη και αποδοτικότερη στη χρήση μνήμης, καθιστώντας την ιδανική για εφαρμογές υψηλών απαιτήσεων. Επιπλέον, η PHP παρέχει εργαλεία και καλές πρακτικές για την εξασφάλιση της ασφάλειας των web εφαρμογών, όπως η χρήση prepared statements για την πρόληψη SQL injection και η ενσωμάτωση μηχανισμών CSRF protection.

Η PHP μπορεί εύκολα να ενσωματωθεί με άλλα συστήματα και εργαλεία, όπως CMS (Content Management Systems) και πλατφόρμες ηλεκτρονικού εμπορίου. Αυτό επιτρέπει την ευκολότερη διαχείριση περιεχομένου και την παροχή επιπρόσθετων υπηρεσιών στους χρήστες του συστήματος διαχείρισης εργασιών μαθημάτων. Η PHP υποστηρίζει επίσης την ανάπτυξη APIs, επιτρέποντας την εύκολη διασύνδεση με άλλες εφαρμογές και υπηρεσίες.

Η PHP είναι μια δωρεάν και ανοικτού κώδικα γλώσσα προγραμματισμού, που σημαίνει ότι δεν υπάρχουν κόστι αδειοδότησης για τη χρήση της. Επιπλέον, το γεγονός ότι η PHP είναι τόσο διαδεδομένη σημαίνει ότι υπάρχει μεγάλη διαθεσιμότητα προγραμματιστών, μειώνοντας το κόστος ανάπτυξης και συντήρησης του συστήματος. Το χαμηλό κόστος ανάπτυξης και η ευκολία συντήρησης καθιστούν την PHP μια οικονομικά αποδοτική επιλογή για την ανάπτυξη του συστήματος διαχείρισης εργασιών μαθημάτων. Η χρήση της PHP για την υλοποίηση του συστήματος διαχείρισης εργασιών μαθημάτων προσφέρει πολλαπλά πλεονεκτήματα, από την ευκολία χρήσης και την ευελιξία έως την υψηλή απόδοση και την ασφάλεια. Με την υποστήριξη μιας μεγάλης κοινότητας και την ενσωμάτωση

με πολλαπλές τεχνολογίες, η PHP αποτελεί μια ισχυρή και αξιόπιστη επιλογή για την ανάπτυξη web εφαρμογών που ανταποκρίνονται στις σύγχρονες εκπαιδευτικές ανάγκες.

3.2 Laravel

Το Laravel είναι ένα δημοφιλές PHP framework ανοικτού κώδικα, σχεδιασμένο για την ανάπτυξη web εφαρμογών με κομψή σύνταξη και αποτελεσματικότητα. Έχει κατασκευαστεί με γνώμονα την ευκολία χρήσης και την παραγωγικότητα των προγραμματιστών, προσφέροντας μια πλούσια γκάμα εργαλείων και βιβλιοθηκών που διευκολύνουν την ανάπτυξη σύγχρονων web εφαρμογών. Το Laravel ακολουθεί το αρχιτεκτονικό πρότυπο Model-View-Controller (MVC), το οποίο βοηθά στον διαχωρισμό της λογικής της εφαρμογής από την παρουσίαση και τη διαχείριση των δεδομένων. [10-12]

Το Laravel δημιουργήθηκε από τον Taylor Otwell το 2011 ως μια προσπάθεια να προσφέρει μια πιο προηγμένη εναλλακτική λύση στα υπάρχοντα PHP frameworks της εποχής, όπως το CodeIgniter. Η πρώτη έκδοση του Laravel κυκλοφόρησε τον Ιούνιο του 2011 και έκτοτε έχει υποστεί πολλές βελτιώσεις και αναβαθμίσεις. Κάθε νέα έκδοση του Laravel εισάγει νέες δυνατότητες και βελτιώσεις, κάνοντάς το ένα από τα πιο δημοφιλή και αξιόπιστα frameworks για την ανάπτυξη web εφαρμογών.

Το Laravel έχει εξελιχθεί σημαντικά από την αρχική του έκδοση.

Laravel 1 (2011): Η πρώτη έκδοση του Laravel προσέφερε βασικά χαρακτηριστικά για την ανάπτυξη web εφαρμογών.

Laravel 2 (2011): Εισήγαγε βελτιώσεις και νέα χαρακτηριστικά, όπως το σύστημα CLI (Artisan).

Laravel 3 (2012): Εισήγαγε πακέτα, υποστήριξη για unit testing, και το σύστημα migrations για τη διαχείριση της βάσης δεδομένων.

Laravel 4 (2013): Μια πλήρης ανακατασκευή που βασίστηκε στο Composer για τη διαχείριση εξαρτήσεων και την επέκταση της λειτουργικότητας μέσω πακέτων.

Laravel 5 (2015): Εισήγαγε πολλά νέα χαρακτηριστικά, όπως το Scheduler για τη διαχείριση εργασιών cron, το Flysystem για την αποθήκευση αρχείων και το βελτιωμένο σύστημα routing.

Laravel 6 (2019): Εισήγαγε το Laravel Vapor, μια πλατφόρμα serverless για την ανάπτυξη εφαρμογών Laravel.

Laravel 7 (2020): Προσέφερε βελτιώσεις στις επιδόσεις και νέα χαρακτηριστικά, όπως το Laravel Airlock για την αυθεντικοποίηση API.

Laravel 8 (2020): Εισήγαγε νέα χαρακτηριστικά, όπως το Laravel Jetstream για την ανάπτυξη εφαρμογών με αυθεντικοποίηση και διαχείριση χρηστών.

Το Laravel είναι σχεδιασμένο για να προσφέρει υψηλή απόδοση και αποδοτικότητα. Οι πιο πρόσφατες εκδόσεις του Laravel έχουν εισαγάγει σημαντικές βελτιώσεις στις επιδόσεις, καθιστώντας το ένα από τα ταχύτερα PHP frameworks. Η ενσωμάτωση εργαλείων όπως το OPcache και το JIT compiler της PHP 8 βοηθούν στην επιτάχυνση της εκτέλεσης του κώδικα, μειώνοντας τον χρόνο απόκρισης και βελτιώνοντας την εμπειρία χρήστη.

Η χρήση του Laravel είναι απλή και ευέλικτη.

Για την εγκατάσταση βεβαιωθείτε ότι έχετε εγκαταστήσει το Composer, το εργαλείο διαχείρισης εξαρτήσεων για PHP.

Εκτελέστε την εντολή `composer create-project --prefer-dist laravel/laravel project-name` για να δημιουργήσετε ένα νέο Laravel project.

Η δομή του Laravel project περιλαμβάνει φακέλους όπως `app`, `config`, `database`, `resources`, `routes`, και `storage`, που βοηθούν στην οργάνωση του κώδικα και των πόρων της εφαρμογής.

Υπάρχουν πολλοί λόγοι για τους οποίους το Laravel είναι η προτιμώμενη επιλογή για την ανάπτυξη web εφαρμογών

Το Laravel έχει σχεδιαστεί για να είναι φιλικό προς τους προγραμματιστές, με σαφή σύνταξη και εκτενή τεκμηρίωση.

Η αρχιτεκτονική MVC βοηθά στον διαχωρισμό της λογικής από την παρουσίαση, καθιστώντας την ανάπτυξη και συντήρηση της εφαρμογής πιο αποδοτική. Το Laravel περιλαμβάνει εργαλεία όπως το Artisan CLI, το Eloquent ORM, και το Blade templating engine, που διευκολύνουν την ανάπτυξη και τη διαχείριση της εφαρμογής. Το Laravel προσφέρει ενσωματωμένες δυνατότητες ασφάλειας, όπως η προστασία από SQL injection, XSS, και CSRF επιθέσεις. Η μεγάλη και ενεργή κοινότητα του Laravel προσφέρει συνεχή υποστήριξη και πλήθος πόρων για την επίλυση προβλημάτων και την ανταλλαγή γνώσεων. Το Laravel είναι εξαιρετικά επεκτάσιμο, επιτρέποντας την ανάπτυξη μικρών εφαρμογών έως μεγάλων επιχειρησιακών λύσεων.

Το Laravel είναι ένα ισχυρό και ευέλικτο PHP framework που προσφέρει όλα τα απαραίτητα εργαλεία και χαρακτηριστικά για την ανάπτυξη σύγχρονων, αποδοτικών και ασφαλών web εφαρμογών. Η ευκολία χρήσης, η υψηλή απόδοση και η εκτενής υποστήριξη από την κοινότητα καθιστούν το Laravel την ιδανική επιλογή για προγραμματιστές που επιθυμούν να δημιουργήσουν και να διαχειριστούν δυναμικές web εφαρμογές.

Είναι ένα PHP framework που έχει σχεδιαστεί για να κάνει την ανάπτυξη web εφαρμογών πιο απλή, ευχάριστη και αποδοτική. Η καμπύλη εκμάθησης του Laravel είναι ήπια, καθώς παρέχει σαφή και κατανοητή τεκμηρίωση, πλούσια εργαλεία ανάπτυξης και προ-ενσωματωμένα χαρακτηριστικά. Αυτό το καθιστά ιδανικό για προγραμματιστές όλων των επιπέδων εμπειρίας, από αρχάριους έως έμπειρους επαγγελματίες. Με το Laravel, οι προγραμματιστές μπορούν να επικεντρωθούν περισσότερο στη

λογική της εφαρμογής και λιγότερο στις λεπτομέρειες της υποδομής, κάτι που επιταχύνει την ανάπτυξη και την παράδοση του έργου.

Το Laravel προσφέρει ενσωματωμένη υποστήριξη για πολλαπλές βάσεις δεδομένων, όπως MySQL, PostgreSQL, SQLite και SQL Server. Το Eloquent ORM (Object-Relational Mapping) του Laravel επιτρέπει την εύκολη αλληλεπίδραση με τη βάση δεδομένων μέσω ενός καθαρού και ευανάγνωστου API. Οι προγραμματιστές μπορούν να διαχειριστούν τις βάσεις δεδομένων τους, να δημιουργήσουν και να τροποποιήσουν πίνακες, καθώς και να χειριστούν σύνθετες σχέσεις δεδομένων χωρίς να χρειάζεται να γράψουν περίπλοκα SQL ερωτήματα. Αυτό διευκολύνει την ανάπτυξη και τη συντήρηση της βάσης δεδομένων, καθιστώντας τη διαχείριση δεδομένων στο σύστημα διαχείρισης εργασιών μαθημάτων πιο αποδοτική.

Είναι εξαιρετικά ευέλικτο και μπορεί να χρησιμοποιηθεί για την ανάπτυξη από μικρές έως μεγάλες web εφαρμογές. Η αρχιτεκτονική του βασίζεται σε ένα μοντέλο MVC (Model-View-Controller), που διαχωρίζει την επιχειρησιακή λογική από την παρουσίαση και τη διαχείριση δεδομένων. Αυτό επιτρέπει στους προγραμματιστές να οργανώνουν τον κώδικά τους με τρόπο που διευκολύνει τη συντήρηση και την επεκτασιμότητα της εφαρμογής. Η ευελιξία του Laravel επιτρέπει την προσθήκη νέων λειτουργιών και βελτιώσεων στο σύστημα διαχείρισης εργασιών μαθημάτων χωρίς να διακυβεύεται η σταθερότητα του υπάρχοντος κώδικα.

Το Laravel υποστηρίζεται από μια μεγάλη και ενεργή κοινότητα προγραμματιστών. Αυτή η κοινότητα παρέχει συνεχή υποστήριξη, εκτενή τεκμηρίωση, και ένα ευρύ φάσμα δωρεάν βιβλιοθηκών και πακέτων που μπορούν να χρησιμοποιηθούν για την επέκταση της λειτουργικότητας του Laravel. Η ύπαρξη αυτής της κοινότητας σημαίνει ότι οι προγραμματιστές μπορούν να βρουν λύσεις σε προβλήματα, να μοιραστούν γνώσεις και να συνεργαστούν για την ανάπτυξη και βελτίωση του framework. Αυτό είναι ιδιαίτερα χρήσιμο για την ανάπτυξη και συντήρηση του συστήματος διαχείρισης εργασιών μαθημάτων, καθώς οι προγραμματιστές μπορούν να βασιστούν στην κοινότητα για υποστήριξη και καινοτομία.

Έχει σχεδιαστεί με γνώμονα την απόδοση και την ασφάλεια. Περιλαμβάνει πολλά εργαλεία και βέλτιστες πρακτικές για τη διασφάλιση της ασφάλειας των web εφαρμογών, όπως η προστασία από SQL injection, η διαχείριση των sessions και η προστασία από CSRF (Cross-Site Request Forgery). Επιπλέον, το Laravel προσφέρει εργαλεία για την αποδοτική διαχείριση των πόρων και την βελτιστοποίηση της απόδοσης, όπως το caching και η υποστήριξη για προφόρτωση (eager loading) και τεμπέλικη φόρτωση (lazy loading) δεδομένων. Αυτά τα χαρακτηριστικά καθιστούν το Laravel κατάλληλο για την ανάπτυξη ασφαλών και αποδοτικών web εφαρμογών, όπως το σύστημα διαχείρισης εργασιών μαθημάτων.

Το Laravel είναι σχεδιασμένο να ενσωματώνεται εύκολα με άλλα συστήματα και εργαλεία. Υποστηρίζει την ανάπτυξη RESTful APIs, επιτρέποντας την εύκολη διασύνδεση με άλλες εφαρμογές

και υπηρεσίες. Επιπλέον, το Laravel περιλαμβάνει εργαλεία για τη διαχείριση εργασιών (task scheduling) και την επεξεργασία ουρών (queue processing), διευκολύνοντας την ανάπτυξη σύνθετων και διαλειτουργικών web εφαρμογών. Η δυνατότητα ενσωμάτωσης με άλλες τεχνολογίες και υπηρεσίες καθιστά το Laravel ιδανικό για την ανάπτυξη ενός πλήρους και διαλειτουργικού συστήματος διαχείρισης εργασιών μαθημάτων.

Το Laravel είναι ένα δωρεάν και ανοικτού κώδικα framework, γεγονός που μειώνει το κόστος ανάπτυξης και συντήρησης του συστήματος. Η χρήση του Laravel επιτρέπει την γρήγορη ανάπτυξη λειτουργιών, μειώνοντας το χρόνο και τους πόρους που απαιτούνται για την ανάπτυξη της εφαρμογής. Επιπλέον, η μεγάλη κοινότητα του Laravel και η πληθώρα διαθέσιμων βιβλιοθηκών και πακέτων σημαίνουν ότι οι προγραμματιστές μπορούν να χρησιμοποιήσουν προϋπάρχοντα εργαλεία και λύσεις, εξοικονομώντας χρόνο και κόπο.

Η χρήση του Laravel για την υλοποίηση του συστήματος διαχείρισης εργασιών μαθημάτων προσφέρει πολλαπλά πλεονεκτήματα, από την ευκολία χρήσης και την ευελιξία έως την υψηλή απόδοση και την ασφάλεια. Με την υποστήριξη μιας μεγάλης και ενεργής κοινότητας και την ενσωμάτωση με πολλαπλές τεχνολογίες, το Laravel αποτελεί μια ισχυρή και αξιόπιστη επιλογή για την ανάπτυξη web εφαρμογών που ανταποκρίνονται στις σύγχρονες εκπαιδευτικές ανάγκες.

3.3 Bootstrap

Το Bootstrap είναι ένα από τα πιο δημοφιλή ανοικτού κώδικα frameworks για την ανάπτυξη responsive και mobile-first ιστοσελίδων. Δημιουργήθηκε από τους Mark Otto και Jacob Thornton στο Twitter και κυκλοφόρησε για πρώτη φορά το 2011. Το Bootstrap παρέχει ένα σύνολο εργαλείων που περιλαμβάνει HTML, CSS και JavaScript, τα οποία βοηθούν τους προγραμματιστές να δημιουργήσουν όμορφες και λειτουργικές ιστοσελίδες γρήγορα και αποτελεσματικά. [13-15]

Το κύριο χαρακτηριστικό του Bootstrap είναι η ευκολία χρήσης του και η δυνατότητα να δημιουργούνται ιστοσελίδες που προσαρμόζονται αυτόματα σε διαφορετικά μεγέθη οθόνης, από κινητά τηλέφωνα μέχρι μεγάλες επιτραπέζιες οθόνες. Το Bootstrap βασίζεται σε ένα σύστημα πλέγματος (grid system) που επιτρέπει την εύελικτη και ακριβή τοποθέτηση των στοιχείων της σελίδας, εξασφαλίζοντας μια συνεπή εμφάνιση και αίσθηση ανεξάρτητα από τη συσκευή που χρησιμοποιείται για την προβολή της.

Το Bootstrap είναι εξαιρετικά χρήσιμο για πολλούς λόγους, και έχει καταστεί το προτιμώμενο εργαλείο για πολλούς προγραμματιστές και σχεδιαστές ιστοσελίδων. Το Bootstrap είναι σχεδιασμένο να είναι εύκολο στη χρήση, ακόμα και για αρχάριους προγραμματιστές. Η τεκμηρίωση του είναι εκτενής και

περιλαμβάνει πολλά παραδείγματα και οδηγίες που διευκολύνουν τη γρήγορη εκμάθηση και την εφαρμογή του.

Το Bootstrap κάνει την ανάπτυξη responsive ιστοσελίδων πολύ πιο απλή. Το σύστημα πλέγματος του Bootstrap επιτρέπει την εύκολη δημιουργία διατάξεων που προσαρμόζονται αυτόματα σε διαφορετικά μεγέθη οθόνης, εξασφαλίζοντας ότι η ιστοσελίδα σας θα φαίνεται καλά σε οποιαδήποτε συσκευή. Περιλαμβάνει μια μεγάλη συλλογή προκαθορισμένων στυλ και συστατικών (components), όπως κουμπιά, φόρμες, κάρτες, πλοήγηση και άλλα, που μπορούν να χρησιμοποιηθούν άμεσα. Αυτό εξοικονομεί χρόνο και προσπάθεια, καθώς δεν χρειάζεται να δημιουργηθούν από το μηδέν. Έχει σχεδιαστεί για να λειτουργεί ομαλά σε όλα τα σύγχρονα προγράμματα περιήγησης, εξασφαλίζοντας μια συνεπή εμπειρία χρήστη ανεξάρτητα από το πρόγραμμα περιήγησης που χρησιμοποιείται. Είναι εξαιρετικά επεκτάσιμο και προσαρμόσιμο. Οι προγραμματιστές μπορούν εύκολα να προσαρμόσουν τα στυλ και τις λειτουργίες του Bootstrap για να ταιριάζουν στις συγκεκριμένες ανάγκες του έργου τους.

Η χρήση του Bootstrap στο σύστημα διαχείρισης εργασιών μαθημάτων προσφέρει πολλά πλεονεκτήματα που διευκολύνουν την ανάπτυξη, τη συντήρηση και τη βελτίωση της εφαρμογής.

Το Bootstrap παρέχει ένα συνεπές και επαγγελματικό σχέδιο που βελτιώνει την εμφάνιση και την αίσθηση της εφαρμογής. Τα προκαθορισμένα στυλ και τα συστατικά του Bootstrap εξασφαλίζουν ότι όλες οι σελίδες και τα στοιχεία της εφαρμογής θα έχουν ένα ομοιόμορφο και μοντέρνο σχέδιο.

Το σύστημα διαχείρισης εργασιών μαθημάτων χρειάζεται να είναι προσβάσιμο από διάφορες συσκευές, όπως υπολογιστές, tablet και κινητά τηλέφωνα. Η χρήση του Bootstrap εξασφαλίζει ότι η εφαρμογή θα προσαρμόζεται αυτόματα σε διαφορετικά μεγέθη οθόνης, παρέχοντας μια βέλτιστη εμπειρία χρήστη ανεξαρτήτως της συσκευής.

Τα προκαθορισμένα στυλ και τα συστατικά του Bootstrap επιταχύνουν τη διαδικασία ανάπτυξης, επιτρέποντας στους προγραμματιστές να δημιουργούν λειτουργικές και όμορφες σελίδες γρήγορα και εύκολα. Αυτό εξοικονομεί χρόνο και πόρους, επιτρέποντας στην ομάδα ανάπτυξης να επικεντρωθεί στις βασικές λειτουργίες και τη λογική της εφαρμογής.

Η χρήση του Bootstrap διευκολύνει τη συντήρηση και την ενημέρωση της εφαρμογής. Οι προγραμματιστές μπορούν εύκολα να κάνουν αλλαγές στα στυλ και τη διάταξη της εφαρμογής χωρίς να χρειάζεται να τροποποιήσουν κάθε σελίδα ξεχωριστά. Αυτό καθιστά την εφαρμογή πιο ευέλικτη και επεκτάσιμη.

Το Bootstrap περιλαμβάνει μια μεγάλη ποικιλία από προκαθορισμένα συστατικά και πρόσθετα (plugins) που μπορούν να χρησιμοποιηθούν για την προσθήκη επιπλέον λειτουργιών στην εφαρμογή. Αυτά τα συστατικά περιλαμβάνουν στοιχεία όπως modal windows, dropdowns, tooltips, και πολλά άλλα, τα οποία μπορούν να βελτιώσουν σημαντικά τη λειτουργικότητα και την εμπειρία χρήστη της εφαρμογής. Είναι συμβατό με όλα τα σύγχρονα προγράμματα περιήγησης και υποστηρίζει τη χρήση

HTML5 και CSS3, εξασφαλίζοντας ότι η εφαρμογή θα λειτουργεί ομαλά και θα είναι ενημερωμένη με τις τελευταίες τεχνολογίες web. Επιπλέον, η μεγάλη και ενεργή κοινότητα του Bootstrap παρέχει συνεχή υποστήριξη και ενημερώσεις, διασφαλίζοντας ότι το framework θα παραμένει επίκαιρο και αξιόπιστο.

Το Bootstrap αποτελεί ένα ισχυρό εργαλείο για την ανάπτυξη web εφαρμογών, προσφέροντας ευκολία χρήσης, responsive σχεδίαση, και πλούσια λειτουργικότητα. Η ενσωμάτωση του Bootstrap στο σύστημα διαχείρισης εργασιών μαθημάτων προσφέρει πολλαπλά πλεονεκτήματα, βελτιώνοντας την εμφάνιση, την απόδοση και τη συντήρηση της εφαρμογής, και εξασφαλίζοντας μια βέλτιστη εμπειρία χρήστη για όλους τους χρήστες της εφαρμογής.

3.4 MySQL

Η MySQL είναι ένα από τα πιο δημοφιλή και ευρέως χρησιμοποιούμενα συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) στον κόσμο. Αναπτύχθηκε αρχικά από τη σουηδική εταιρεία MySQL AB το 1995 και αργότερα αποκτήθηκε από την Oracle Corporation. Η MySQL είναι ανοικτού κώδικα και διατίθεται δωρεάν υπό την άδεια GPL (General Public License), αν και υπάρχουν και εμπορικές εκδόσεις με πρόσθετες δυνατότητες και υποστήριξη. Χρησιμοποιεί τη γλώσσα SQL (Structured Query Language) για την αλληλεπίδραση με τη βάση δεδομένων και υποστηρίζει πολλαπλές πλατφόρμες, όπως Linux, Windows, και macOS. Είναι γνωστή για την υψηλή απόδοση, την αξιοπιστία και την ευκολία χρήσης της, καθιστώντας την ιδανική επιλογή για μια μεγάλη ποικιλία εφαρμογών, από μικρές προσωπικές ιστοσελίδες έως μεγάλες επιχειρηματικές εφαρμογές και πλατφόρμες κοινωνικής δικτύωσης. [16-18]

Η MySQL είναι εξαιρετικά χρήσιμη για πολλούς λόγους, που την καθιστούν ένα από τα πιο προτιμώμενα συστήματα διαχείρισης βάσεων δεδομένων για προγραμματιστές και επιχειρήσεις. Έχει σχεδιαστεί για υψηλή απόδοση και μπορεί να διαχειριστεί μεγάλο όγκο δεδομένων και υψηλά φορτία ταυτόχρονων χρηστών. Είναι κλιμακώσιμη, επιτρέποντας την ανάπτυξη από μικρές εφαρμογές έως μεγάλης κλίμακας επιχειρηματικές εφαρμογές. Η MySQL είναι γνωστή για την αξιοπιστία και τη σταθερότητά της. Διαθέτει μηχανισμούς ανάκτησης από σφάλματα, λειτουργίες αντιγράφων ασφαλείας και αποκατάστασης, και υποστηρίζει ACID (Atomicity, Consistency, Isolation, Durability) συναλλαγές, εξασφαλίζοντας την ακεραιότητα των δεδομένων.

Είναι εύκολη στη μάθηση και τη χρήση, με μια απλή και κατανοητή σύνταξη SQL. Διαθέτει επίσης μια πληθώρα εργαλείων διαχείρισης και διεπαφών χρήστη που διευκολύνουν τη διαχείριση της βάσης δεδομένων, όπως το MySQL Workbench. Υποστηρίζει πολλαπλές πλατφόρμες, επιτρέποντας την ανάπτυξη εφαρμογών σε διάφορα λειτουργικά συστήματα χωρίς προβλήματα συμβατότητας. Επίσης,

υποστηρίζει πολλαπλές μηχανές αποθήκευσης (storage engines), όπως InnoDB και MyISAM, επιτρέποντας στους προγραμματιστές να επιλέξουν την κατάλληλη μηχανή για τις ανάγκες της εφαρμογής τους. Η δυνατότητα αλλαγής μηχανών αποθήκευσης επιτρέπει την προσαρμογή και την επέκταση της βάσης δεδομένων με βάση τις απαιτήσεις της εφαρμογής.

Η MySQL παρέχει ισχυρούς μηχανισμούς ασφαλείας, όπως έλεγχο ταυτότητας χρηστών, διαχείριση δικαιωμάτων πρόσβασης και κρυπτογράφηση δεδομένων, διασφαλίζοντας την προστασία των δεδομένων από μη εξουσιοδοτημένη πρόσβαση και έχει μια μεγάλη και ενεργή κοινότητα χρηστών και προγραμματιστών που παρέχουν συνεχή υποστήριξη, τεκμηρίωση και πόρους. Επιπλέον, υπάρχουν πολλές εταιρείες που προσφέρουν επαγγελματική υποστήριξη και υπηρεσίες για τη MySQL.

Η χρήση της MySQL στο σύστημα διαχείρισης εργασιών μαθημάτων προσφέρει πολλαπλά πλεονεκτήματα που βελτιώνουν την αποδοτικότητα, την αξιοπιστία και την ευκολία διαχείρισης της εφαρμογής. Ακολουθούν μερικοί από τους κύριους λόγους για τους οποίους η MySQL είναι τόσο χρήσιμη στο σύστημα μας:

Το σύστημα διαχείρισης εργασιών μαθημάτων χρειάζεται να αποθηκεύει και να διαχειρίζεται μεγάλες ποσότητες δεδομένων, όπως πληροφορίες φοιτητών, λεπτομέρειες μαθημάτων και υποβολές εργασιών. Η MySQL προσφέρει τα εργαλεία και τις δυνατότητες για την αποθήκευση, ανάκτηση και διαχείριση αυτών των δεδομένων με αποδοτικό και ασφαλή τρόπο.

Η MySQL υποστηρίζει ACID συναλλαγές και παρέχει μηχανισμούς ανάκτησης από σφάλματα, διασφαλίζοντας την ακεραιότητα και την αξιοπιστία των δεδομένων. Αυτό είναι ιδιαίτερα σημαντικό για το σύστημα διαχείρισης εργασιών μαθημάτων, καθώς εξασφαλίζει ότι οι υποβολές εργασιών και οι βαθμολογίες των φοιτητών αποθηκεύονται με ασφάλεια και ακρίβεια. Είναι σχεδιασμένη για υψηλή απόδοση και μπορεί να διαχειριστεί μεγάλο όγκο δεδομένων και πολλαπλές ταυτόχρονες συνδέσεις χρηστών. Αυτό εξασφαλίζει ότι το σύστημα διαχείρισης εργασιών μαθημάτων μπορεί να ανταποκριθεί στις απαιτήσεις ενός μεγάλου αριθμού χρηστών, χωρίς να επηρεάζεται η ταχύτητα και η απόδοση της εφαρμογής. Παρέχει εύχρηστα εργαλεία διαχείρισης, όπως το MySQL Workbench, που διευκολύνουν τη δημιουργία, τη συντήρηση και την παρακολούθηση της βάσης δεδομένων. Αυτό επιτρέπει στους διαχειριστές του συστήματος να διαχειρίζονται αποτελεσματικά τη βάση δεδομένων και να εξασφαλίζουν τη συνεχή λειτουργία της εφαρμογής.

Η υποστήριξη πολλαπλών μηχανών αποθήκευσης επιτρέπει την προσαρμογή της βάσης δεδομένων στις συγκεκριμένες ανάγκες του συστήματος διαχείρισης εργασιών μαθημάτων. Αυτό διευκολύνει την επέκταση της εφαρμογής με νέα χαρακτηριστικά και δυνατότητες, καθώς και την προσαρμογή της βάσης δεδομένων στις αυξανόμενες απαιτήσεις της εφαρμογής.

Η MySQL παρέχει ισχυρούς μηχανισμούς ασφαλείας, όπως έλεγχο ταυτότητας χρηστών και διαχείριση δικαιωμάτων πρόσβασης, διασφαλίζοντας την προστασία των δεδομένων των φοιτητών και των διδασκόντων. Αυτό είναι ιδιαίτερα σημαντικό για την προστασία των ευαίσθητων πληροφοριών και την εξασφάλιση της εμπιστευτικότητας των δεδομένων. Έχει μια μεγάλη και ενεργή κοινότητα χρηστών και προγραμματιστών που παρέχουν συνεχή υποστήριξη, τεκμηρίωση και πόρους. Αυτό σημαίνει ότι οποιοδήποτε πρόβλημα ή απορία μπορεί να επιλυθεί γρήγορα και αποτελεσματικά, εξασφαλίζοντας την απρόσκοπτη λειτουργία του συστήματος. Η χρήση της MySQL για την υλοποίηση του συστήματος διαχείρισης εργασιών μαθημάτων προσφέρει πολλαπλά πλεονεκτήματα, από την υψηλή απόδοση και την αξιοπιστία έως την ευκολία διαχείρισης και την ασφάλεια. Με την υποστήριξη μιας μεγάλης κοινότητας και την ευελιξία της να προσαρμόζεται στις ανάγκες της εφαρμογής, η MySQL αποτελεί μια ισχυρή και αξιόπιστη επιλογή για την ανάπτυξη και τη διαχείριση βάσεων δεδομένων στο σύστημα διαχείρισης εργασιών μαθημάτων.

Κεφάλαιο 4ο: Το σύστημα διαχείρισης εργασιών μαθήματος

4.1 Περιγραφή διαδικασίας

Η διαχείριση εργασιών μαθημάτων αποτελεί κρίσιμο μέρος της εκπαιδευτικής διαδικασίας, τόσο για τους διδάσκοντες όσο και για τους φοιτητές. Ένα ολοκληρωμένο σύστημα διαχείρισης εργασιών επιτρέπει στους διδάσκοντες να δημιουργούν, να διαχειρίζονται και να βαθμολογούν τις εργασίες, ενώ παρέχει στους φοιτητές τη δυνατότητα να υποβάλλουν τις εργασίες τους ηλεκτρονικά, να παρακολουθούν τις προθεσμίες και να λαμβάνουν ανατροφοδότηση και βαθμολογίες.

Η πλατφόρμα που έχει αναπτυχθεί με τη χρήση του Laravel, ενός δημοφιλούς PHP framework, προσφέρει ένα σύνολο λειτουργιών που εξυπηρετούν τόσο τους διδάσκοντες όσο και τους φοιτητές. Η εφαρμογή διαθέτει ένα ολοκληρωμένο σύστημα αυθεντικοποίησης και διαχείρισης χρηστών, υποστηρίζοντας ρόλους όπως διδάσκων, φοιτητής και διαχειριστής. Επιπλέον, η πλατφόρμα επιτρέπει τη δημιουργία και διαχείριση μαθημάτων, την ανάθεση εργασιών, την υποβολή αρχείων και την αξιολόγηση των εργασιών.

Μέσω της ενσωμάτωσης του reCAPTCHA της Google, η εφαρμογή διασφαλίζει την προστασία από κακόβουλες επιθέσεις και spam. Το σύστημα υποστηρίζει επίσης τη δυνατότητα ανάκτησης κωδικού πρόσβασης, παρέχοντας έναν εύκολο τρόπο για τους χρήστες να ανακτήσουν την πρόσβασή τους σε περίπτωση που ξεχάσουν τον κωδικό τους.

Το περιβάλλον χρήστη της εφαρμογής έχει σχεδιαστεί με τη χρήση του Bootstrap, προσφέροντας μια φιλική και εύχρηστη διεπαφή. Οι διδάσκοντες μπορούν να δουν τα μαθήματα και τις εργασίες που έχουν δημιουργήσει, να προσθέσουν νέες εργασίες και να βαθμολογήσουν τις υποβολές των φοιτητών. Οι φοιτητές, από την άλλη πλευρά, μπορούν να εγγραφούν σε μαθήματα με τη χρήση ενός μοναδικού κωδικού μαθήματος, να υποβάλλουν τις εργασίες τους και να παρακολουθούν την πρόοδο και τις βαθμολογίες τους.

4.1.1 Διαχείριση Χρηστών και Αυθεντικοποίηση

Η διαχείριση χρηστών και η αυθεντικοποίηση είναι θεμελιώδεις λειτουργίες της εφαρμογής διαχείρισης εργασιών μαθημάτων. Κάθε χρήστης μπορεί να δημιουργήσει έναν λογαριασμό είτε ως διδάσκων είτε ως φοιτητής μέσω της φόρμας εγγραφής. Η εγγραφή περιλαμβάνει τη συμπλήρωση προσωπικών στοιχείων, όπως το όνομα, το επώνυμο, το email και τον κωδικό πρόσβασης, καθώς και την επιλογή του ρόλου χρήστη.

Κατά τη διάρκεια της εγγραφής και της σύνδεσης, η εφαρμογή χρησιμοποιεί το reCAPTCHA της Google για την αποτροπή κακόβουλων επιθέσεων και spam. Η επαλήθευση του reCAPTCHA προσθέτει ένα επιπλέον επίπεδο ασφάλειας, διασφαλίζοντας ότι οι εγγραφές και οι συνδέσεις πραγματοποιούνται από πραγματικούς χρήστες.

Μετά την επιτυχή σύνδεση, οι χρήστες μεταφέρονται στον πίνακα ελέγχου (dashboard) της εφαρμογής, όπου έχουν πρόσβαση στις λειτουργίες που σχετίζονται με τον ρόλο τους. Οι διδάσκοντες μπορούν να δουν τα μαθήματα που διδάσκουν, να δημιουργήσουν νέες εργασίες και να διαχειριστούν τις υποβολές των φοιτητών. Οι φοιτητές, από την άλλη πλευρά, μπορούν να δουν τα μαθήματα στα οποία είναι εγγεγραμμένοι, να υποβάλουν εργασίες και να δουν τις βαθμολογίες τους.

Η εφαρμογή παρέχει επίσης λειτουργίες ανάκτησης κωδικού πρόσβασης. Οι χρήστες που ξεχούν τον κωδικό τους μπορούν να ζητήσουν έναν σύνδεσμο επαναφοράς κωδικού μέσω email. Αυτή η λειτουργία διασφαλίζει ότι οι χρήστες μπορούν να ανακτήσουν την πρόσβασή τους με ασφάλεια και ευκολία.

Η διαχείριση χρηστών και η αυθεντικοποίηση αποτελούν κρίσιμες λειτουργίες της εφαρμογής, διασφαλίζοντας την ασφάλεια και την αξιοπιστία της πλατφόρμας. Η χρήση του reCAPTCHA της Google και οι λειτουργίες ανάκτησης κωδικού πρόσβασης ενισχύουν περαιτέρω την ασφάλεια και την εμπειρία χρήστη.

4.1.2 Διαχείριση Μαθημάτων και Εγγραφές

Η διαχείριση μαθημάτων είναι μια από τις βασικές λειτουργίες της εφαρμογής. Οι διδάσκοντες έχουν τη δυνατότητα να δημιουργούν και να διαχειρίζονται μαθήματα μέσω της πλατφόρμας. Κατά τη δημιουργία ενός μαθήματος, οι διδάσκοντες εισάγουν πληροφορίες όπως ο τίτλος του μαθήματος, η περιγραφή και ένας μοναδικός κωδικός μαθήματος. Ο κωδικός αυτός χρησιμοποιείται από τους φοιτητές για την εγγραφή τους στο μάθημα.

Οι φοιτητές μπορούν να εγγραφούν σε μαθήματα εισάγοντας τον μοναδικό κωδικό μαθήματος σε μια ειδική σελίδα εγγραφής. Μόλις επιβεβαιωθεί ο κωδικός, οι φοιτητές προστίθενται στη λίστα των εγγεγραμμένων φοιτητών για το συγκεκριμένο μάθημα. Αυτή η διαδικασία εγγραφής διασφαλίζει ότι μόνο οι εξουσιοδοτημένοι φοιτητές μπορούν να εγγραφούν στα μαθήματα.

Οι διδάσκοντες έχουν πρόσβαση σε μια επισκόπηση όλων των μαθημάτων που έχουν δημιουργήσει. Από αυτήν την επισκόπηση, μπορούν να επεξεργαστούν τις λεπτομέρειες των μαθημάτων ή να διαγράψουν μαθήματα που δεν χρειάζονται πλέον. Επιπλέον, μπορούν να δουν τη λίστα των φοιτητών που είναι εγγεγραμμένοι σε κάθε μάθημα και να διαχειριστούν τις εγγραφές τους.

Η διαχείριση μαθημάτων και οι εγγραφές παρέχουν μια ευέλικτη και ασφαλή διαδικασία για τη δημιουργία και τη διαχείριση μαθημάτων. Οι διδάσκοντες μπορούν να οργανώσουν τα μαθήματά τους αποτελεσματικά, ενώ οι φοιτητές μπορούν να εγγραφούν εύκολα στα μαθήματα που τους ενδιαφέρουν.

4.1.3 Διαχείριση Εργασιών και Υποβολές

Η διαχείριση εργασιών είναι μια από τις κύριες λειτουργίες της εφαρμογής. Οι διδάσκοντες μπορούν να δημιουργούν εργασίες για τα μαθήματά τους, ορίζοντας λεπτομέρειες όπως ο τίτλος της εργασίας, η περιγραφή, η ημερομηνία λήξης, ο μέγιστος αριθμός αρχείων που μπορούν να υποβληθούν και οι επιτρεπόμενοι τύποι αρχείων. Οι εργασίες εμφανίζονται στους φοιτητές όταν επισκέπτονται τη σελίδα του μαθήματος, επιτρέποντάς τους να δουν όλες τις απαραίτητες πληροφορίες.

Οι φοιτητές μπορούν να υποβάλουν τις εργασίες τους απευθείας μέσω της πλατφόρμας. Κατά την υποβολή μιας εργασίας, οι φοιτητές ανεβάζουν τα αρχεία τους και η πλατφόρμα ελέγχει αν τα αρχεία πληρούν τις προδιαγραφές που έχουν οριστεί από τον διδάσκοντα. Οι υποβολές αποθηκεύονται με ασφάλεια και οι φοιτητές μπορούν να δουν τα αρχεία που έχουν υποβάλει, καθώς και να τα διαγράψουν αν χρειαστεί να κάνουν αλλαγές πριν από την προθεσμία.

Οι διδάσκοντες έχουν τη δυνατότητα να δουν όλες τις υποβολές των φοιτητών για κάθε εργασία. Μπορούν να ανοίξουν τα αρχεία, να τα ελέγξουν και να βαθμολογήσουν τις υποβολές. Η βαθμολογία μπορεί να γίνει με βάση την ποιότητα της εργασίας, την πληρότητα και άλλους παράγοντες που ο διδάσκων κρίνει σημαντικούς. Οι διδάσκοντες μπορούν επίσης να προσθέσουν σχόλια και παρατηρήσεις για κάθε υποβολή, παρέχοντας ανατροφοδότηση στους φοιτητές.

Η εφαρμογή διασφαλίζει ότι οι φοιτητές δεν μπορούν να υποβάλουν αρχεία μετά την ημερομηνία λήξης της εργασίας, επιτρέποντας μόνο τις έγκαιρες υποβολές. Αυτή η λειτουργία διασφαλίζει την τήρηση των προθεσμιών και την ομαλή διεξαγωγή των μαθημάτων.

Η διαχείριση εργασιών και υποβολών παρέχει ένα ολοκληρωμένο και ευέλικτο σύστημα για τη δημιουργία, την υποβολή και την αξιολόγηση των εργασιών, διευκολύνοντας την αλληλεπίδραση μεταξύ διδασκόντων και φοιτητών.

4.1.4 Αξιολόγηση και Ανατροφοδότηση

Η αξιολόγηση των εργασιών και η παροχή ανατροφοδότησης είναι κρίσιμες λειτουργίες της εφαρμογής. Οι διδάσκοντες μπορούν να βαθμολογήσουν τις υποβολές των φοιτητών απευθείας μέσω της πλατφόρμας. Κατά την αξιολόγηση, οι διδάσκοντες μπορούν να δουν τα αρχεία που έχουν υποβάλει οι φοιτητές, να προσθέσουν βαθμολογία και να γράψουν σχόλια.

Η βαθμολογία μπορεί να βασίζεται σε διάφορα κριτήρια, όπως η πληρότητα της εργασίας, η ακρίβεια, η καινοτομία και η συμμόρφωση με τις οδηγίες. Οι διδάσκοντες μπορούν να παρέχουν λεπτομερή ανατροφοδότηση, βοηθώντας τους φοιτητές να κατανοήσουν τα δυνατά και αδύναμα σημεία τους και να βελτιωθούν στις μελλοντικές εργασίες.

Οι φοιτητές μπορούν να δουν τις βαθμολογίες και τα σχόλια των διδασκόντων στις σελίδες των εργασιών τους. Αυτή η ανατροφοδότηση είναι σημαντική για την ακαδημαϊκή τους πρόοδο και την κατανόηση των απαιτήσεων του μαθήματος. Οι φοιτητές μπορούν επίσης να επικοινωνήσουν με τους διδάσκοντες για να ζητήσουν περαιτέρω διευκρινίσεις ή βοήθεια.

Η εφαρμογή επιτρέπει στους διδάσκοντες να ενημερώνουν τις βαθμολογίες και τα σχόλια, παρέχοντας ευελιξία στην αξιολόγηση. Επιπλέον, οι διδάσκοντες μπορούν να δουν την πρόοδο των φοιτητών τους συνολικά, επιτρέποντάς τους να εντοπίσουν τυχόν προβλήματα και να προσφέρουν επιπλέον υποστήριξη όπου χρειάζεται.

Η διαφάνεια και η άμεση ανατροφοδότηση βελτιώνουν τη μαθησιακή εμπειρία και ενισχύουν τη δέσμευση των φοιτητών. Μέσω της εφαρμογής, οι φοιτητές λαμβάνουν την απαραίτητη υποστήριξη και καθοδήγηση για την ακαδημαϊκή τους επιτυχία.

Η αξιολόγηση και η ανατροφοδότηση είναι αναπόσπαστο μέρος της εκπαιδευτικής διαδικασίας που υποστηρίζεται από την εφαρμογή. Οι διδάσκοντες μπορούν να παρέχουν έγκαιρη και ουσιαστική ανατροφοδότηση, βοηθώντας τους φοιτητές να βελτιωθούν και να πετύχουν τους ακαδημαϊκούς τους στόχους.

4.2 Εισαγωγή – Η διαδικασία

Η σελίδα εγγραφής επιτρέπει στους νέους χρήστες να δημιουργήσουν έναν λογαριασμό στο σύστημα. Οι χρήστες πρέπει να συμπληρώσουν διάφορα πεδία με προσωπικές πληροφορίες, όπως όνομα, επώνυμο, διεύθυνση email και κωδικό πρόσβασης. Η φόρμα εγγραφής περιλαμβάνει επίσης επιλογές για τον ρόλο του χρήστη, δηλαδή αν είναι φοιτητής ή διδάσκων. Μόλις ολοκληρωθεί η εγγραφή, οι χρήστες θα μπορούν να συνδεθούν και να αρχίσουν να χρησιμοποιούν τις λειτουργίες του συστήματος..

Καλωσορίσατε στο Σύστημα Διαχείρισης Εργασιών

Το σύστημα αυτό σας επιτρέπει να διαχειρίζεστε τις εργασίες μαθημάτων με ευκολία. Οι διδάσκοντες μπορούν να δημιουργήσουν και να βαθμολογήσουν εργασίες, ενώ οι φοιτητές μπορούν να υποβάλλουν τις εργασίες τους ηλεκτρονικά.

Επιλέξτε σύνδεση ή εγγραφή για να συνεχίσετε.

Εικόνα 4.1: Το σύστημα – η πρώτη σελίδα για είσοδο ή εγγραφή

Η σελίδα εγγραφής (Εικόνα 4.2) επιτρέπει στους νέους χρήστες να δημιουργήσουν έναν λογαριασμό στο σύστημα. Οι χρήστες καλούνται να εισάγουν τα προσωπικά τους στοιχεία, όπως όνομα, επίθετο, διεύθυνση ηλεκτρονικού ταχυδρομείου και κωδικό πρόσβασης. Επιπλέον, πρέπει να επιλέξουν τον ρόλο τους (φοιτητής ή διδάσκων) για να προσαρμοστεί η εμπειρία χρήσης ανάλογα με τις ανάγκες τους. Η διαδικασία εγγραφής είναι απλή και γρήγορη, εξασφαλίζοντας ότι οι χρήστες μπορούν να ξεκινήσουν να χρησιμοποιούν το σύστημα χωρίς καθυστερήσεις.

Task Management Login

Εγγραφή

Όνομα

Επώνυμο

Email

marios.tzolis@gmail.com

Κωδικός

....

Επιβεβαίωση Κωδικού

Ρόλος

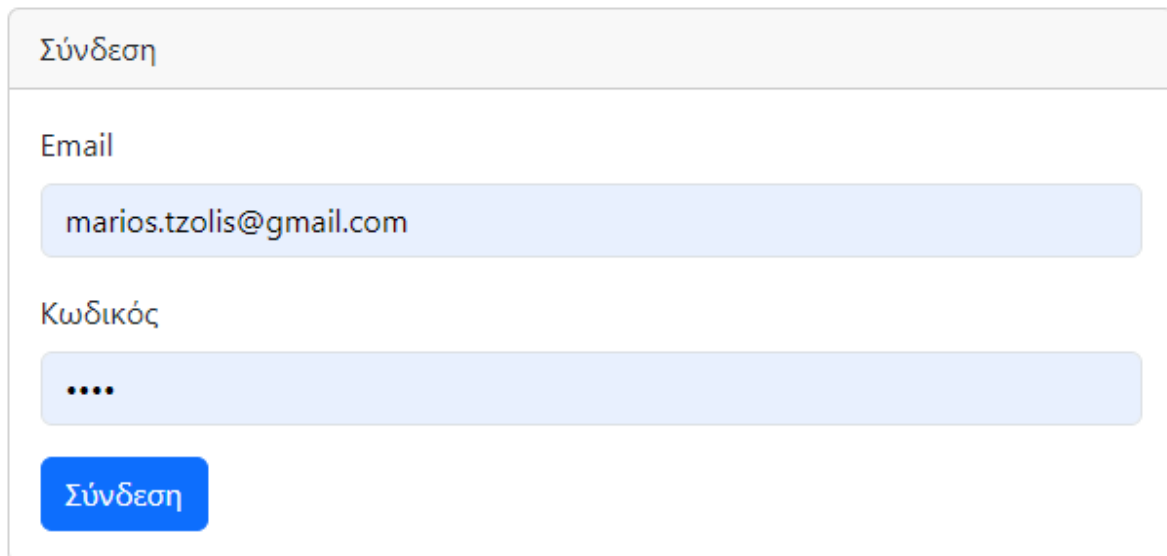
Φοιτητής

Εγγραφή

Εικόνα 4.2: Σελίδα Εγγραφής

Η σελίδα σύνδεσης (Εικόνα 4.3) επιτρέπει στους εγγεγραμμένους χρήστες να εισέλθουν στο σύστημα χρησιμοποιώντας τη διεύθυνση ηλεκτρονικού ταχυδρομείου και τον κωδικό πρόσβασής τους. Αυτή η

σελίδα είναι απλή και σαφής, με πεδία για την εισαγωγή των στοιχείων σύνδεσης και κουμπί για την επιβεβαίωση της εισόδου. Σε περίπτωση που ο χρήστης έχει ξεχάσει τον κωδικό πρόσβασής του, παρέχεται επίσης η επιλογή επαναφοράς κωδικού μέσω email.



Σύνδεση

Email

marios.tzolis@gmail.com

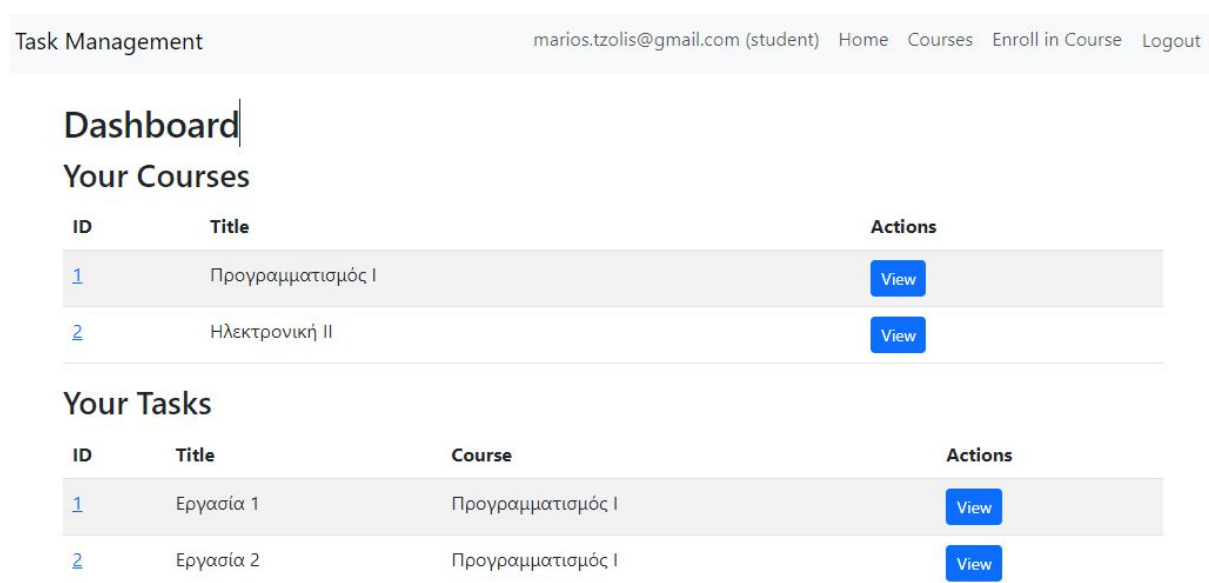
Κωδικός

••••

Σύνδεση

Εικόνα 4.3: Σελίδα για σύνδεση του χρήστη

Η κεντρική σελίδα dashboard για τους φοιτητές (Εικόνα 4.4) προσφέρει μια συνοπτική εικόνα των μαθημάτων και των εργασιών τους. Σε αυτή τη σελίδα, οι φοιτητές μπορούν να δουν όλα τα μαθήματα στα οποία είναι εγγεγραμμένοι και τις σχετικές εργασίες. Η σελίδα αυτή παρέχει πληροφορίες όπως τις προθεσμίες υποβολής και την κατάσταση κάθε εργασίας. Η φιλική προς τον χρήστη διεπαφή βοηθά τους φοιτητές να παρακολουθούν τις ακαδημαϊκές τους υποχρεώσεις εύκολα και αποτελεσματικά.



Task Management

marios.tzolis@gmail.com (student) Home Courses Enroll in Course Logout

Dashboard

Your Courses

ID	Title	Actions
1	Προγραμματισμός I	View
2	Ηλεκτρονική II	View

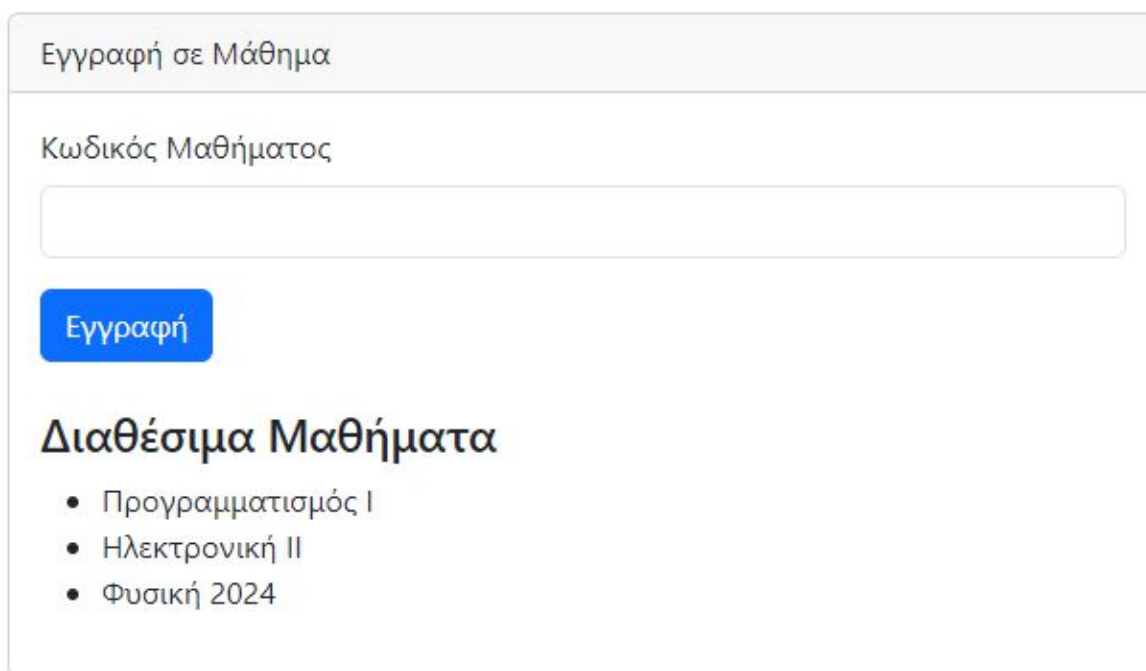
Your Tasks

ID	Title	Course	Actions
1	Εργασία 1	Προγραμματισμός I	View
2	Εργασία 2	Προγραμματισμός I	View

Εικόνα 4.4: Κεντρική σελίδα dashboard για παρακολούθηση μαθημάτων και εργασιών: Ρόλος Φοιτητής

Η εγγραφή του φοιτητή σε κάποιο μάθημα (Εικόνα 4.5) γίνεται μέσω της εισαγωγής ενός μοναδικού κωδικού μαθήματος που παρέχεται από τον διδάσκοντα. Αυτή η διαδικασία διασφαλίζει ότι μόνο οι εξουσιοδοτημένοι φοιτητές μπορούν να εγγραφούν στα μαθήματα. Η σελίδα εγγραφής επιτρέπει στους φοιτητές να εισάγουν τον κωδικό και να προσθέσουν το μάθημα στο προφίλ τους, εξασφαλίζοντας την πρόσβαση στις σχετικές εργασίες και υλικό μαθήματος.

Η επιλογή ενός μαθήματος από την κεντρική σελίδα (Εικόνα 4.6) επιτρέπει στους φοιτητές να δουν όλες τις διαθέσιμες εργασίες για το συγκεκριμένο μάθημα. Η σελίδα αυτή εμφανίζει τις εργασίες που βρίσκονται εντός προθεσμίας υποβολής, παρέχοντας πληροφορίες όπως την ημερομηνία δημιουργίας και την προθεσμία υποβολής. Οι φοιτητές μπορούν να επιλέξουν κάθε εργασία για να δουν περισσότερες λεπτομέρειες και να ξεκινήσουν τη διαδικασία υποβολής των εργασιών τους.



Εγγραφή σε Μάθημα

Κωδικός Μαθήματος

Εγγραφή

Διαθέσιμα Μαθήματα

- Προγραμματισμός I
- Ηλεκτρονική II
- Φυσική 2024

Εικόνα 4.5: Εγγραφή του φοιτητή σε κάποιο μάθημα αφού έχει στη διάθεση του τον κατάλληλο κωδικό μαθήματος που του έχει δοθεί από τον διδάσκοντα

[Dashboard](#) / [Course Details](#)

Course Details

Προγραμματισμός I

Προγραμματισμός I

Tasks

ID	Title	Due Date	Actions
1	Εργασία 1	24-05-2024 21:38	View
2	Εργασία 2	17-05-2024 21:38	View

Εικόνα 4.6: Επιλογή ενός μαθήματος από την κεντρική σελίδα για προβολή των διαθέσιμων και εντός προθεσμιών εργασιών του μαθήματος. Ρόλος Φοιτητής

[Dashboard](#) / [Course](#) / [Task Details](#)

Task Details

Εργασία 1

Εργασία 1 για Προγραμματισμό I

Due Date: 24-05-2024 21:38

Your Submissions

ID	File	Submission Date	Actions
1	View File	17-05-2024 19:01	Delete
2	View File	17-05-2024 19:07	Delete

Submit Your Work

File

No file chosen

Your Grade

6.00

Εικόνα 4.7: Προβολή των αρχείων, της βαθμολογίας και των χαρακτηριστικών για μια εργασία ενός μαθήματος. Ρόλος Φοιτητής

Η σελίδα προβολής των αρχείων και της βαθμολογίας για μια εργασία (Εικόνα 4.7) επιτρέπει στους φοιτητές να δουν τα αρχεία που έχουν υποβάλει, τη βαθμολογία που έχουν λάβει και τα σχόλια του διδάσκοντα. Οι φοιτητές μπορούν επίσης να δουν λεπτομέρειες όπως την ημερομηνία υποβολής και

την προθεσμία. Αυτή η σελίδα παρέχει μια ολοκληρωμένη εικόνα της προόδου τους και τους επιτρέπει να παρακολουθούν τις αξιολογήσεις των εργασιών τους.

Όταν μια εργασία έχει περάσει την προθεσμία υποβολής, οι φοιτητές βλέπουν ένα μήνυμα ενημέρωσης (Εικόνα 4.8) που τους ειδοποιεί ότι η υποβολή είναι πλέον εκτός προθεσμίας. Αυτή η λειτουργία διασφαλίζει ότι οι φοιτητές γνωρίζουν τις προθεσμίες και μπορούν να διαχειριστούν καλύτερα τον χρόνο τους για μελλοντικές υποβολές. Το μήνυμα αυτό είναι σαφές και εμφανίζεται σε εμφανές σημείο στη σελίδα της εργασίας.

Η σελίδα σύνδεσης για τον διδάσκοντα (Εικόνα 4.9) είναι παρόμοια με τη σελίδα σύνδεσης των φοιτητών, αλλά προορίζεται ειδικά για τους διδάσκοντες. Οι διδάσκοντες χρησιμοποιούν τα διαπιστευτήριά τους για να εισέλθουν στο σύστημα και να αποκτήσουν πρόσβαση στις λειτουργίες που είναι διαθέσιμες για αυτούς, όπως η δημιουργία και η διαχείριση μαθημάτων και εργασιών.

The screenshot shows a web interface for task management. At the top, there is a navigation bar with 'Task Management' on the left and user information 'marios.tzolis@gmail.com (student)' along with links for 'Home', 'Courses', 'Enroll in Course', and 'Logout' on the right. Below the navigation bar, there is a breadcrumb trail: 'Dashboard / Course / Task Details'. The main heading is 'Task Details', followed by 'Εργασία 2' (Task 2) and 'Εργασία 2 για Προγραμματισμό Ι' (Task 2 for Programming I). The 'Due Date' is listed as '17-05-2024 21:38'. A red message states 'The deadline for this task has passed.' Below this, there is a section for 'Your Submissions' with a table that has columns for 'ID', 'File', 'Submission Date', and 'Actions'. A red message 'The deadline for this task has passed.' is also present under the submissions section. Finally, there is a section for 'Your Grade' which shows 'Not Graded'.

Εικόνα 4.8: Προβολή μηνύματος για εκτός προθεσμίας για μια εργασία ενός μαθήματος. Ρόλος Φοιτητής

Σύνδεση

Email

Κωδικός

[Σύνδεση](#)

Εικόνα 4.9: Σύνδεση του διδάσκοντα

Το dashboard για τον διδάσκοντα (Εικόνα 4.10) προσφέρει μια κεντρική σελίδα όπου οι διδάσκοντες μπορούν να δουν όλα τα μαθήματα που διαχειρίζονται, καθώς και τις σχετικές εργασίες. Η σελίδα αυτή παρέχει επιλογές για την προβολή, την επεξεργασία και τη διαγραφή των μαθημάτων και των εργασιών, διευκολύνοντας τη διαχείριση του εκπαιδευτικού υλικού και των ακαδημαϊκών υποχρεώσεων των φοιτητών.

Task Management
ktsiak@ihu.gr (teacher) Home Courses Logout

Dashboard

Add Course

Your Courses

ID	Title	Actions
1	Προγραμματισμός I	View Edit Delete
2	Ηλεκτρονική II	View Edit Delete
3	Φυσική 2024	View Edit Delete

Your Tasks

ID	Title	Course	Actions
1	Εργασία 1	Προγραμματισμός I	View Edit Delete
2	Εργασία 2	Προγραμματισμός I	View Edit Delete

Εικόνα 4.10: Προβολή του dashboard για τον διδάσκοντα. Διαθέσιμα τα μαθήματα του και οι εργασίες τους με κάθε επιλογή, όπως προβολή, επεξεργασία και διαγραφή. Ρόλος Διδάσκοντας

Edit Course

Title
Προγραμματισμός Ι

Description
Προγραμματισμός Ι

Unique Code
PROA1

Update Course

Εικόνα 4.11: Επεξεργασία ενός μαθήματος από τον διδάσκοντα

Η σελίδα επεξεργασίας ενός μαθήματος (Εικόνα 4.11) επιτρέπει στους διδάσκοντες να τροποποιήσουν τις λεπτομέρειες ενός μαθήματος, όπως τον τίτλο, την περιγραφή και τις προθεσμίες των εργασιών. Οι διδάσκοντες μπορούν να ενημερώνουν το περιεχόμενο και τις απαιτήσεις του μαθήματος για να διασφαλίσουν ότι είναι πάντα επίκαιρες και ανταποκρίνονται στις ανάγκες των φοιτητών.

Edit Task

Course
Προγραμματισμός Ι

Title
Εργασία 1

Description
Εργασία 1 για Προγραμματισμό Ι

Due Date
05/24/2024 09:38 PM

Max Files
4

Allowed File Types
pdf

Update Task

Εικόνα 4.12: Επεξεργασία μιας εργασίας ενός μαθήματος από τον διδάσκοντα. Καθορισμός των απαραίτητων πεδίων.

[Dashboard](#) / [Course](#) / Task Details

Task Details

Εργασία 1

Εργασία 1 για Προγραμματισμό 1

Due Date: 24-05-2024 21:38

Grade Submissions

Μάριος Τζώλης	6.00	Grade
Νίκος Πέτρου	5.00	Grade

Submissions

Student	ID	File	Submission Date	Actions
Μάριος Τζώλης	1	View File	17-05-2024 19:01	Delete
Μάριος Τζώλης	2	View File	17-05-2024 19:07	Delete
Νίκος Πέτρου	3	View File	18-05-2024 08:20	Delete
Νίκος Πέτρου	8	View File	18-05-2024 09:19	Delete
Νίκος Πέτρου	9	View File	18-05-2024 12:13	Delete

[Edit Task](#) [Delete Task](#)

Εικόνα 4.13: Προβολή και επεξεργασία των αρχείων και της βαθμολογίας μιας εργασίας για όλους τους φοιτητές που συμμετέχουν σε αυτή.

Η σελίδα επεξεργασίας μιας εργασίας (Εικόνα 4.12) επιτρέπει στους διδάσκοντες να ορίσουν ή να τροποποιήσουν τα πεδία μιας εργασίας, όπως τον τίτλο, την περιγραφή, τον αριθμό των επιτρεπόμενων αρχείων και τις προθεσμίες. Αυτή η λειτουργία διευκολύνει τους διδάσκοντες να διαμορφώσουν τις εργασίες ανάλογα με τις ανάγκες και τις απαιτήσεις του μαθήματος, διασφαλίζοντας την ποιότητα και τη σαφήνεια των ακαδημαϊκών υποχρεώσεων.

Η σελίδα προβολής και επεξεργασίας των αρχείων και της βαθμολογίας μιας εργασίας (Εικόνα 4.13) επιτρέπει στους διδάσκοντες να δουν όλα τα αρχεία που έχουν υποβληθεί από τους φοιτητές για μια συγκεκριμένη εργασία. Οι διδάσκοντες μπορούν να βαθμολογήσουν τις εργασίες, να προσθέσουν σχόλια και να διαχειριστούν τα αρχεία, διευκολύνοντας τη διαδικασία αξιολόγησης και ανατροφοδότησης. Αυτή η σελίδα παρέχει μια ολοκληρωμένη εικόνα της προόδου των φοιτητών και διευκολύνει την αλληλεπίδραση μεταξύ διδασκόντων και φοιτητών.

4.3 Η Βάση

4.3.1 Δομή και Σχέσεις της Βάσης Δεδομένων για το Σύστημα Διαχείρισης Εργασιών Μαθήματος

Η βάση δεδομένων αποτελεί τη θεμέλια λίθο κάθε εφαρμογής διαχείρισης δεδομένων, καθώς αποθηκεύει και οργανώνει όλες τις πληροφορίες που απαιτούνται για τη λειτουργία της εφαρμογής. Στο σύστημα διαχείρισης εργασιών μαθήματος, η βάση δεδομένων είναι σχεδιασμένη με τέτοιο τρόπο ώστε να υποστηρίζει τις λειτουργίες διαχείρισης χρηστών, μαθημάτων, εργασιών και αξιολόγησης. Σε αυτό το κεφάλαιο, θα περιγράψουμε τη δομή των πινάκων της βάσης δεδομένων, τις σχέσεις μεταξύ αυτών των πινάκων και τον τρόπο με τον οποίο αλληλεπιδρούν για να παρέχουν μια ολοκληρωμένη λύση.

Πίνακας user

Ο πίνακας user αποθηκεύει πληροφορίες για όλους τους χρήστες της εφαρμογής, συμπεριλαμβανομένων των διδασκόντων, των φοιτητών και των διαχειριστών. Ο πίνακας περιλαμβάνει τα ακόλουθα πεδία:

id: Μοναδικό αναγνωριστικό για κάθε χρήστη (AUTO_INCREMENT).

firstname: Το όνομα του χρήστη.

lastname: Το επώνυμο του χρήστη.

email: Η διεύθυνση email του χρήστη (πρέπει να είναι μοναδική).

password: Ο κωδικός πρόσβασης του χρήστη (αποθηκεύεται με hash).

role: Ο ρόλος του χρήστη (student, teacher, admin).

active: Ένδειξη αν ο χρήστης είναι ενεργός (1) ή όχι (0).

created_at: Ημερομηνία και ώρα δημιουργίας της εγγραφής.

updated_at: Ημερομηνία και ώρα τελευταίας ενημέρωσης της εγγραφής.

Πίνακας course

Ο πίνακας course αποθηκεύει πληροφορίες για κάθε μάθημα που δημιουργείται από τους διδάσκοντες.

Ο πίνακας περιλαμβάνει τα ακόλουθα πεδία:

id: Μοναδικό αναγνωριστικό για κάθε μάθημα (AUTO_INCREMENT).

title: Ο τίτλος του μαθήματος.

description: Μια περιγραφή του μαθήματος.

unique_code: Ένας μοναδικός κωδικός για το μάθημα, που χρησιμοποιείται για την εγγραφή φοιτητών.

teacher_id: Το αναγνωριστικό του διδάσκοντος που δημιούργησε το μάθημα (foreign key προς τον πίνακα user).

active: Ένδειξη αν το μάθημα είναι ενεργό (1) ή όχι (0).

created_at: Ημερομηνία και ώρα δημιουργίας της εγγραφής.

updated_at: Ημερομηνία και ώρα τελευταίας ενημέρωσης της εγγραφής.

Πίνακας coursetask

Ο πίνακας coursetask αποθηκεύει πληροφορίες για κάθε εργασία που ανήκει σε ένα μάθημα. Ο πίνακας περιλαμβάνει τα ακόλουθα πεδία:

id: Μοναδικό αναγνωριστικό για κάθε εργασία (AUTO_INCREMENT).

course_id: Το αναγνωριστικό του μαθήματος στο οποίο ανήκει η εργασία (foreign key προς τον πίνακα course).

title: Ο τίτλος της εργασίας.

description: Μια περιγραφή της εργασίας.

due_date: Η ημερομηνία λήξης της εργασίας.

max_files: Ο μέγιστος αριθμός αρχείων που μπορούν να υποβληθούν για την εργασία.

allowed_file_types: Οι επιτρεπόμενοι τύποι αρχείων για την εργασία.

active: Ένδειξη αν η εργασία είναι ενεργή (1) ή όχι (0).

created_at: Ημερομηνία και ώρα δημιουργίας της εγγραφής.

updated_at: Ημερομηνία και ώρα τελευταίας ενημέρωσης της εγγραφής.

Πίνακας courseattend

Ο πίνακας `courseattend` αποθηκεύει πληροφορίες για τη συμμετοχή των φοιτητών σε μαθήματα. Ο πίνακας περιλαμβάνει τα ακόλουθα πεδία:

`id`: Μοναδικό αναγνωριστικό για κάθε εγγραφή (AUTO_INCREMENT).

`course_id`: Το αναγνωριστικό του μαθήματος (foreign key προς τον πίνακα `course`).

`student_id`: Το αναγνωριστικό του φοιτητή (foreign key προς τον πίνακα `user`).

`created_at`: Ημερομηνία και ώρα δημιουργίας της εγγραφής.

`updated_at`: Ημερομηνία και ώρα τελευταίας ενημέρωσης της εγγραφής.

Πίνακας `coursetaskattend`

Ο πίνακας `coursetaskattend` αποθηκεύει πληροφορίες για τις υποβολές εργασιών των φοιτητών και τη βαθμολογία τους. Ο πίνακας περιλαμβάνει τα ακόλουθα πεδία:

`id`: Μοναδικό αναγνωριστικό για κάθε εγγραφή (AUTO_INCREMENT).

`task_id`: Το αναγνωριστικό της εργασίας (foreign key προς τον πίνακα `coursetask`).

`student_id`: Το αναγνωριστικό του φοιτητή (foreign key προς τον πίνακα `user`).

`course_id`: Το αναγνωριστικό του μαθήματος (foreign key προς τον πίνακα `course`).

`grade`: Η βαθμολογία που έχει δοθεί στη συγκεκριμένη εργασία.

`created_at`: Ημερομηνία και ώρα δημιουργίας της εγγραφής.

`updated_at`: Ημερομηνία και ώρα τελευταίας ενημέρωσης της εγγραφής.

Πίνακας `files`

Ο πίνακας `files` αποθηκεύει πληροφορίες για τα αρχεία που υποβάλλονται από τους φοιτητές για τις εργασίες. Ο πίνακας περιλαμβάνει τα ακόλουθα πεδία:

`id`: Μοναδικό αναγνωριστικό για κάθε αρχείο (AUTO_INCREMENT).

`task_id`: Το αναγνωριστικό της εργασίας (foreign key προς τον πίνακα `coursetask`).

`student_id`: Το αναγνωριστικό του φοιτητή που υπέβαλε το αρχείο (foreign key προς τον πίνακα `user`).

`file_path`: Η διαδρομή του αρχείου που αποθηκεύτηκε στο σύστημα.

created_at: Ημερομηνία και ώρα δημιουργίας της εγγραφής.

updated_at: Ημερομηνία και ώρα τελευταίας ενημέρωσης της εγγραφής.

Σχέσεις μεταξύ των Πινάκων

Οι σχέσεις μεταξύ των πινάκων της βάσης δεδομένων είναι θεμελιώδεις για την οργάνωση και την ακεραιότητα των δεδομένων. Οι κύριες σχέσεις μεταξύ των πινάκων περιλαμβάνουν:

Χρήστες και Μαθήματα: Ένας διδάσκων (teacher) μπορεί να δημιουργήσει πολλά μαθήματα, αλλά κάθε μάθημα ανήκει σε έναν μόνο διδάσκοντα. Αυτή η σχέση είναι ένα προς πολλά (one-to-many) και υλοποιείται με το πεδίο teacher_id στον πίνακα course.

Μαθήματα και Εργασίες: Κάθε μάθημα μπορεί να έχει πολλές εργασίες, αλλά κάθε εργασία ανήκει σε ένα μόνο μάθημα. Αυτή η σχέση είναι επίσης ένα προς πολλά (one-to-many) και υλοποιείται με το πεδίο course_id στον πίνακα coursetask.

Φοιτητές και Μαθήματα: Οι φοιτητές μπορούν να εγγραφούν σε πολλά μαθήματα και κάθε μάθημα μπορεί να έχει πολλούς φοιτητές. Αυτή η σχέση είναι πολλά προς πολλά (many-to-many) και υλοποιείται με τον πίνακα courseattend, που περιέχει τα πεδία course_id και student_id.

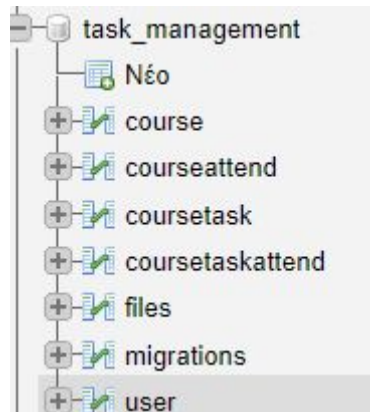
Εργασίες και Υποβολές: Κάθε εργασία μπορεί να έχει πολλές υποβολές από διαφορετικούς φοιτητές, αλλά κάθε υποβολή ανήκει σε μία μόνο εργασία. Αυτή η σχέση είναι ένα προς πολλά (one-to-many) και υλοποιείται με το πεδίο task_id στον πίνακα files.

Φοιτητές και Υποβολές: Κάθε φοιτητής μπορεί να υποβάλει πολλά αρχεία για διάφορες εργασίες, αλλά κάθε υποβολή αρχείου ανήκει σε έναν μόνο φοιτητή. Αυτή η σχέση είναι ένα προς πολλά (one-to-many) και υλοποιείται με το πεδίο student_id στον πίνακα files.

Φοιτητές και Βαθμολογίες: Κάθε φοιτητής μπορεί να λάβει πολλές βαθμολογίες για διάφορες εργασίες, αλλά κάθε βαθμολογία σχετίζεται με μια συγκεκριμένη εργασία και φοιτητή. Αυτή η σχέση είναι ένα προς πολλά (one-to-many) και υλοποιείται με τα πεδία task_id και student_id στον πίνακα coursetaskattend.

Αυτές οι σχέσεις διασφαλίζουν ότι τα δεδομένα αποθηκεύονται με συνέπεια και επιτρέπουν την αποτελεσματική ανάκτηση πληροφοριών από τη βάση δεδομένων. Η οργάνωση και η δομή της βάσης δεδομένων είναι θεμελιώδη για την ομαλή λειτουργία της εφαρμογής και τη διασφάλιση της ακεραιότητας των δεδομένων.

4.3.2 Δομή και περιεχόμενα των πινάκων



Εικόνα 4.14: Η βάση με τους πίνακες

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα
<input type="checkbox"/>	1	id 🔑	int(10)		UNSIGNED	Όχι	Καμία	AUTO_INCREMENT
<input type="checkbox"/>	2	firstname	varchar(255)	utf8mb4_general_ci		Όχι	Καμία	
<input type="checkbox"/>	3	lastname	varchar(255)	utf8mb4_general_ci		Όχι	Καμία	
<input type="checkbox"/>	4	name	varchar(255)	utf8mb4_general_ci		Όχι	Καμία	
<input type="checkbox"/>	5	email 🔑	varchar(255)	utf8mb4_general_ci		Όχι	Καμία	
<input type="checkbox"/>	6	password	varchar(255)	utf8mb4_general_ci		Όχι	Καμία	
<input type="checkbox"/>	7	role	enum('student', 'teacher', 'admin')	utf8mb4_general_ci		Όχι	Καμία	
<input type="checkbox"/>	8	active	tinyint(3)		UNSIGNED	Ναι	1	
<input type="checkbox"/>	9	created_at	timestamp			Όχι	current_timestamp()	
<input type="checkbox"/>	10	updated_at	timestamp			Όχι	current_timestamp()	ON UPDATE CURRENT_TIMESTAMP()

Εικόνα 4.15: Η δομή του Πίνακα user

id	firstname	lastname	name	email	password	role	active	created_at	updated_at
1	admin	admin	admin	admin@gmail.com	1234	admin	1	2024-05-17 14:59:05	2024-05-17 15:00:22
2	Μάριος	Τζώλης	Τζώλης Μάριος	marios.tzolis@gmail.com	1234	student	1	2024-05-17 14:59:06	2024-05-17 15:00:10
3	Κυριάκος	Τσιακμάκης	Τσιακμάκης Κυριάκος	ksiak@ihu.gr	1234	teacher	1	2024-05-17 14:59:06	2024-05-20 22:26:15
5	Νίκος	Πέτρου	Πέτρου Νίκος	nikos@gmail.com	1234	student	1	2024-05-18 10:14:44	2024-05-18 10:14:44
6	Μαρία	Νάκου	Νάκου Μαρία	maria@gmail.com	1234	student	1	2024-05-18 10:14:44	2024-05-18 10:14:44
7	Κώστας	Πατάκης	Πατάκης Κώστας	kostas@ihu.gr	1234	student	1	2024-05-18 11:07:43	2024-05-18 14:08:25

Εικόνα 4.16: Τα περιεχόμενα του Πίνακα user

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα
1	id 🔑	int(10)		UNSIGNED	Όχι	Καμία		AUTO_INCREMENT
2	teacher_id	int(10)		UNSIGNED	Όχι	Καμία		
3	title	varchar(255)	utf8mb4_general_ci		Όχι	Καμία		
4	description	text	utf8mb4_general_ci		Ναι	NULL		
5	unique_code 🗝	varchar(50)	utf8mb4_general_ci		Όχι	Καμία		
6	active	tinyint(3)		UNSIGNED	Ναι	1		
7	created_at	timestamp			Όχι	current_timestamp()		
8	updated_at	timestamp			Όχι	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Εικόνα 4.17: Η δομή του Πίνακα course

id	teacher_id	title	description	unique_code	active	created_at	updated_at
1	3	Προγραμματισμός I	Προγραμματισμός I	PROA1	1	2024-05-17 20:47:41	2024-05-17 20:47:41
2	3	Ηλεκτρονική II	Ηλεκτρονική II	ELECB1	1	2024-05-17 20:47:41	2024-05-17 20:47:41
3	3	Φυσική 2024	Φυσική 2024	PH123	1	2024-05-17 18:24:37	2024-05-17 18:24:46

Εικόνα 4.18: Τα περιεχόμενα του Πίνακα course

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα
1	id 🔑	int(10)		UNSIGNED	Όχι	Καμία		AUTO_INCREMENT
2	student_id	int(10)		UNSIGNED	Όχι	Καμία		
3	course_id	int(10)		UNSIGNED	Όχι	Καμία		
4	active	tinyint(3)		UNSIGNED	Ναι	1		
5	created_at	timestamp			Όχι	current_timestamp()		
6	updated_at	timestamp			Όχι	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Εικόνα 4.19: Η δομή του Πίνακα courseattend

id	student_id	course_id	active	created_at	updated_at
1	2	1	1	2024-05-17 20:48:51	2024-05-17 20:48:51
2	2	2	1	2024-05-17 20:48:51	2024-05-17 20:48:51
3	5	3	1	2024-05-18 08:01:27	2024-05-18 08:01:27
4	5	1	1	2024-05-18 08:14:39	2024-05-18 08:14:39

Εικόνα 4.20: Τα περιεχόμενα του Πίνακα courseattend

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα
1	id	int(10)		UNSIGNED	Όχι	Καμία		AUTO_INCREMENT
2	course_id	int(10)		UNSIGNED	Όχι	Καμία		
3	title	varchar(255)	utf8mb4_general_ci		Όχι	Καμία		
4	description	text	utf8mb4_general_ci		Ναι	NULL		
5	due_date	datetime			Όχι	Καμία		
6	max_files	tinyint(3)		UNSIGNED	Όχι	Καμία		
7	allowed_file_types	varchar(255)	utf8mb4_general_ci		Όχι	Καμία		
8	active	tinyint(3)		UNSIGNED	Ναι	1		
9	created_at	timestamp			Όχι	current_timestamp()		
10	updated_at	timestamp			Όχι	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Εικόνα 4.21: Η δομή του Πίνακα coursetask

id	course_id	title	description	due_date	max_files	allowed_file_types	active	created_at	updated_at
1	1	Εργασία 1	Εργασία 1 για Προγραμματισμό I	2024-05-24 21:38:00	4	pdf	1	2024-05-17 18:38:28	2024-05-17 22:07:19
2	1	Εργασία 2	Εργασία 2 για Προγραμματισμό I	2024-05-17 21:38:00	3	pdf	1	2024-05-17 18:38:28	2024-05-17 22:07:19

Εικόνα 4.22: Τα περιεχόμενα του Πίνακα coursetask

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα
1	id	int(10)		UNSIGNED	Όχι	Καμία		AUTO_INCREMENT
2	student_id	int(10)		UNSIGNED	Όχι	Καμία		
3	task_id	int(10)		UNSIGNED	Όχι	Καμία		
4	course_id	int(10)		UNSIGNED	Όχι	Καμία		
5	grade	decimal(5,2)			Ναι	NULL		
6	initial_submission_date	timestamp			Όχι	current_timestamp()		
7	resubmission_date	timestamp			Όχι	current_timestamp()		
8	active	tinyint(3)		UNSIGNED	Ναι	1		
9	created_at	timestamp			Όχι	current_timestamp()		
10	updated_at	timestamp			Όχι	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

Εικόνα 4.23: Η δομή του Πίνακα coursetaskattend

id	student_id	task_id	course_id	grade	initial_submission_date	resubmission_date	active	created_at	updated_at
1	2	1	1	6.00	2024-05-18 10:12:00	2024-05-18 10:12:00	1	2024-05-18 10:12:00	2024-05-18 07:56:33
2	5	2	1	0.00	2024-05-18 10:12:00	2024-05-18 10:12:00	1	2024-05-18 10:12:00	2024-05-18 07:56:33
3	5	1	1	5.00	2024-05-18 12:19:52	2024-05-18 12:19:52	1	2024-05-18 09:19:52	2024-05-18 09:20:22

Εικόνα 4.24: Τα περιεχόμενα του Πίνακα coursetaskattend

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα
1	id 🔑	bigint(20)		UNSIGNED	Όχι	Καμία		AUTO_INCREMENT
2	task_id	bigint(20)		UNSIGNED	Όχι	Καμία		
3	student_id	bigint(20)		UNSIGNED	Όχι	Καμία		
4	file_path	varchar(255)	utf8mb4_unicode_ci		Όχι	Καμία		
5	created_at	timestamp			Ναι	NULL		
6	updated_at	timestamp			Ναι	NULL		

Εικόνα 4.25: Η δομή του Πίνακα files

id	task_id	student_id	file_path	created_at	updated_at
1	1	2	submissions/iEWISg6xAAtZB9Nha3PtMLq5n0ft4Xoe1CXXFb...	2024-05-17 19:01:02	2024-05-17 19:01:02
2	1	2	public/submissions/kzyCNx8h1OdvU3XVbCN2xLweYYUfcP4...	2024-05-17 19:07:28	2024-05-17 19:07:28
3	1	5	public/submissions/sxmtKTnskAml0t8btwuljlqNzzeiov7...	2024-05-18 08:20:49	2024-05-18 08:20:49
8	1	5	public/submissions/Nw1hDjEQ4vL4BmKoyhHsDEW3WBDyPFf...	2024-05-18 09:19:52	2024-05-18 09:19:52
9	1	5	public/submissions/eRgrsNh523Ycl9W5xm89UugPaaM3HaU...	2024-05-18 12:13:45	2024-05-18 12:13:45

Εικόνα 4.26: Τα περιεχόμενα του Πίνακα files

4.3.3 Η βάση με το Laravel

Η Laravel είναι ένα δημοφιλές PHP framework που διευκολύνει την ανάπτυξη εφαρμογών web, προσφέροντας ένα ευρύ φάσμα εργαλείων για τη διαχείριση βάσεων δεδομένων. Σε αυτό το κεφάλαιο, θα εξηγήσουμε πώς μπορούμε να χειριζόμαστε τη βάση δεδομένων και τους πίνακες χρησιμοποιώντας τη Laravel, με παραδείγματα από την εφαρμογή διαχείρισης εργασιών μαθημάτων που έχουμε αναπτύξει.

Δημιουργία και Διαχείριση Πινάκων με Migrations

Τα migrations είναι μια ισχυρή δυνατότητα της Laravel που επιτρέπει τη δημιουργία, την αλλαγή και τη διαχείριση των πινάκων της βάσης δεδομένων με προγραμματιστικό τρόπο. Τα migrations είναι βασικά αρχεία PHP που περιέχουν κώδικα για τη διαμόρφωση των πινάκων.

Παράδειγμα: Δημιουργία του Πίνακα users

```

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateUsersTable extends Migration
{
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id(); // Μοναδικό αναγνωριστικό (primary key)

            $table->string('firstname'); // Όνομα

            $table->string('lastname'); // Επώνυμο

            $table->string('email')->unique(); // Email (μοναδικό)

            $table->string('password'); // Κωδικός (hashed)

            $table->enum('role', ['student', 'teacher', 'admin']); // Ρόλος χρήστη

            $table->boolean('active')->default(1); // Κατάσταση ενεργού χρήστη

            $table->timestamps(); // created_at και updated_at
        });
    }

    public function down()
    {
        Schema::dropIfExists('users');
    }
}

```

Σε αυτό το migration, δημιουργούμε τον πίνακα users με τα πεδία firstname, lastname, email, password, role και active. Χρησιμοποιούμε το Schema::create για να δημιουργήσουμε τον πίνακα και το Blueprint για να ορίσουμε τα πεδία του πίνακα.

Διαχείριση Δεδομένων με το Eloquent ORM

Η Laravel περιλαμβάνει ένα ισχυρό ORM (Object-Relational Mapping) που ονομάζεται Eloquent. Το Eloquent επιτρέπει την εύκολη αλληλεπίδραση με τη βάση δεδομένων μέσω μοντέλων.

Παράδειγμα: Μοντέλο User

```
<?php

namespace App\Models;

use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;

class User extends Authenticatable
{
    use Notifiable;

    protected $fillable = [
        'firstname', 'lastname', 'email', 'password', 'role', 'active',
    ];

    protected $hidden = [
        'password', 'remember_token',
    ];
}
```

```
protected $casts = [  
    'email_verified_at' => 'datetime',  
];  
}
```

Το μοντέλο User αντιστοιχεί στον πίνακα users. Χρησιμοποιούμε την ιδιότητα \$fillable για να ορίσουμε ποια πεδία μπορούν να εκχωρηθούν μαζικά (mass assignable), και την ιδιότητα \$hidden για να αποκρύψουμε πεδία κατά την μετατροπή του μοντέλου σε array ή JSON.

Δημιουργία, Ενημέρωση και Διαγραφή Δεδομένων

Παράδειγμα: Δημιουργία Χρήστη

```
use App\Models\User;  
use Illuminate\Support\Facades\Hash;  
  
$user = User::create([  
    'firstname' => 'John',  
    'lastname' => 'Doe',  
    'email' => 'john.doe@example.com',  
    'password' => Hash::make('password123'), // Χρήση hash για τον κωδικό  
    'role' => 'student',  
    'active' => 1,  
]);
```

Χρησιμοποιούμε τη μέθοδο create του μοντέλου User για να δημιουργήσουμε έναν νέο χρήστη. Το Laravel αναλαμβάνει τη δημιουργία της κατάλληλης SQL εντολής για την εισαγωγή των δεδομένων στον πίνακα users.

Παράδειγμα: Ενημέρωση Χρήστη

```
$user = User::find(1); // Εύρεση χρήστη με ID 1
$user->firstname = 'Jane';
$user->lastname = 'Smith';
$user->save(); // Αποθήκευση αλλαγών
```

Με τη χρήση του Eloquent, μπορούμε να βρούμε έναν χρήστη με το `find` και να ενημερώσουμε τα πεδία του. Η μέθοδος `save` αποθηκεύει τις αλλαγές στη βάση δεδομένων.

Παράδειγμα: Διαγραφή Χρήστη

```
$user = User::find(1); // Εύρεση χρήστη με ID 1
$user->delete(); // Διαγραφή χρήστη
```

Η μέθοδος `delete` διαγράφει τον χρήστη από τη βάση δεδομένων.

Σχέσεις μεταξύ Μοντέλων

Οι σχέσεις μεταξύ των πινάκων μπορούν να εκφραστούν με αντίστοιχες σχέσεις μεταξύ των μοντέλων στο Eloquent.

Παράδειγμα: Σχέση Ένα-Προς-Πολλά (One-to-Many)

Η σχέση ανάμεσα στους πίνακες `course` και `coursetask` είναι μία σχέση ένα-προς-πολλά, καθώς ένα μάθημα μπορεί να έχει πολλές εργασίες.

```
// Model Course
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Course extends Model
{
```

```

use HasFactory;

protected $fillable = [
    'title', 'description', 'unique_code', 'teacher_id', 'active',
];

public function tasks()
{
    return $this->hasMany(Coursetask::class);
}
}

// Model Coursetask
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Coursetask extends Model
{
    use HasFactory;

    protected $fillable = [
        'course_id', 'title', 'description', 'due_date', 'max_files', 'allowed_file_types', 'active',
    ];

    public function course()
    {

```

```
return $this->belongsTo(Course::class);  
}  
}
```

Η μέθοδος `tasks` στο μοντέλο `Course` ορίζει τη σχέση ένα-προς-πολλά με το μοντέλο `Coursetask`, ενώ η μέθοδος `course` στο μοντέλο `Coursetask` ορίζει τη σχέση πολλοί-προς-ένα με το μοντέλο `Course`.

4.4 Ασφάλεια στο σύστημα και στα δεδομένα

Η ασφάλεια είναι ένας από τους πλέον σημαντικούς παράγοντες σε κάθε σύστημα πληροφορικής. Στο σύστημα διαχείρισης εργασιών μαθημάτων που έχουμε αναπτύξει, έχουμε λάβει μια σειρά από μέτρα για να διασφαλίσουμε την προστασία των δεδομένων και την ασφάλεια των χρηστών. Σε αυτό το κεφάλαιο, θα αναλύσουμε τις διάφορες πτυχές της ασφάλειας που έχουμε υιοθετήσει.

Ασφάλεια με Κωδικούς Πρόσβασης

Οι κωδικοί πρόσβασης είναι η πρώτη γραμμή άμυνας σε κάθε σύστημα αυθεντικοποίησης. Στο σύστημα μας, οι κωδικοί πρόσβασης των χρηστών αποθηκεύονται με τη χρήση `hash`, διασφαλίζοντας ότι οι πραγματικοί κωδικοί δεν αποθηκεύονται ποτέ στη βάση δεδομένων. Η διαδικασία `hash` διασφαλίζει ότι ακόμη και αν η βάση δεδομένων παραβιαστεί, οι κωδικοί πρόσβασης δεν θα είναι άμεσα διαθέσιμοι στους επιτιθέμενους.

Παράδειγμα Χρήσης Hash

Για την αποθήκευση των κωδικών πρόσβασης, χρησιμοποιούμε τη μέθοδο `Hash::make` της `Laravel`:

```
use Illuminate\Support\Facades\Hash;  
  
$user = User::create([  
    'firstname' => 'John',  
    'lastname' => 'Doe',  
    'email' => 'john.doe@example.com',
```

```
'password' => Hash::make('password123'), // Χρήση hash για τον κωδικό
'role' => 'student',
'active' => 1,
]);
```

Με αυτόν τον τρόπο, ο κωδικός password123 αποθηκεύεται στη βάση δεδομένων ως ένα κρυπτογραφημένο hash, το οποίο δεν μπορεί να αντιστραφεί εύκολα σε μορφή κειμένου.

Ασφάλεια με Sessions για την Αυθεντικοποίηση

Η χρήση των sessions είναι κρίσιμη για την ασφάλεια της αυθεντικοποίησης και της επικοινωνίας μεταξύ του χρήστη και του συστήματος. Όταν ένας χρήστης συνδέεται στο σύστημα, δημιουργείται ένα μοναδικό session ID, το οποίο αποθηκεύεται στον browser του χρήστη και στον server. Αυτό το session ID χρησιμοποιείται για να ταυτοποιήσει τον χρήστη σε κάθε επόμενο αίτημα.

Παράδειγμα Διαχείρισης Session

Κατά την επιτυχή σύνδεση ενός χρήστη, οι πληροφορίες του αποθηκεύονται στη session:

```
if ($user && Hash::check($request->password, $user->password)) {
    Session::put('user_id', $user->id);
    Session::put('user_role', $user->role);
    Session::put('user_email', $user->email);

    return redirect()->route('dashboard');
}
```

Η χρήση των sessions διασφαλίζει ότι μόνο οι πιστοποιημένοι χρήστες έχουν πρόσβαση σε προστατευμένες περιοχές του συστήματος.

Hashing και Salting

Το hashing είναι η διαδικασία μετατροπής ενός δεδομένου σε μια σταθερού μεγέθους κωδικοποιημένη μορφή, ενώ το salting προσθέτει επιπλέον τυχαία δεδομένα στη διαδικασία hashing για να αυξήσει την ασφάλεια. Στο σύστημά μας, το hashing χρησιμοποιείται κυρίως για την αποθήκευση των κωδικών πρόσβασης, ενώ η Laravel εφαρμόζει αυτομάτως salting κατά τη δημιουργία των hashes.

Παράδειγμα Χρήσης Hash και Salt

Η Laravel χρησιμοποιεί το bcrypt αλγόριθμο για το hashing των κωδικών πρόσβασης, ο οποίος περιλαμβάνει ενσωματωμένο salting:

```
$password = Hash::make('password123'); // Δημιουργία hash με salt
```

Αυτό διασφαλίζει ότι ακόμα και αν δύο χρήστες έχουν τον ίδιο κωδικό πρόσβασης, τα αποθηκευμένα hashes τους θα είναι διαφορετικά.

Προστασία από SQL Injection

Το SQL Injection είναι μια από τις πιο κοινές μορφές επίθεσης σε βάσεις δεδομένων, όπου ένας επιτιθέμενος εισάγει κακόβουλο SQL κώδικα σε μια φόρμα εισόδου δεδομένων για να παραβιάσει την ασφάλεια της βάσης δεδομένων. Στο σύστημά μας, χρησιμοποιούμε τις δυνατότητες του Eloquent ORM και τις προετοιμασμένες δηλώσεις (prepared statements) της Laravel για την προστασία από SQL Injection.

Παράδειγμα Προστασίας από SQL Injection

Όταν χρησιμοποιούμε το Eloquent για την αλληλεπίδραση με τη βάση δεδομένων, η Laravel χειρίζεται αυτόματα την ασφάλεια των ερωτημάτων:

```
$user = User::where('email', $request->email)->first();
```

Ακόμη και αν ο χρήστης εισάγει κακόβουλο SQL κώδικα στο πεδίο email, η Laravel θα το χειριστεί με ασφάλεια, αποτρέποντας την εκτέλεση κακόβουλων εντολών.

Όταν χρησιμοποιείτε το `DB::table('user')` με τις μεθόδους που προσφέρει η Laravel, έχετε προστασία από SQL injection. Οι μέθοδοι της Laravel για τη διαχείριση της βάσης δεδομένων, όπως το Eloquent και το Query Builder, χρησιμοποιούν προετοιμασμένες δηλώσεις (prepared statements) που προστατεύουν από SQL injection.

Ας δούμε ένα παράδειγμα:

Παράδειγμα Χρήσης του Query Builder με Προστασία από SQL Injection

```
use Illuminate\Support\Facades\DB;

$user = DB::table('user')->where('email', $request->email)->first();
```

Ακόμα και αν ο χρήστης εισάγει κακόβουλο SQL κώδικα στο πεδίο email, η Laravel θα χειριστεί το αίτημα με ασφάλεια, αποτρέποντας την εκτέλεση κακόβουλων εντολών. Η Laravel μετατρέπει αυτόματα τις εισόδους του χρήστη σε παραμέτρους στις προετοιμασμένες δηλώσεις, διασφαλίζοντας ότι δεν εκτελούνται κακόβουλες εντολές SQL.

Παράδειγμα Χρήσης του Query Builder για Εισαγωγή Δεδομένων

Όταν χρησιμοποιείτε το Query Builder για εισαγωγή δεδομένων, η Laravel χειρίζεται αυτόματα την ασφάλεια των ερωτημάτων:

```
use Illuminate\Support\Facades\DB;

DB::table('user')->insert([

    'firstname' => $request->firstname,

    'lastname' => $request->lastname,

    'email' => $request->email,

    'password' => Hash::make($request->password),

    'role' => 'student',

    'active' => 1,
```

```
]);
```

Προστασία από SQL Injection με Χειροκίνητα Ερωτήματα

Αν χρειαστεί να εκτελέσετε πιο σύνθετα ερωτήματα με το Query Builder, η Laravel παρέχει τρόπους για να το κάνετε με ασφάλεια:

```
use Illuminate\Support\Facades\DB;
```

```
$results = DB::select('select * from user where email = ?', [$request->email]);
```

Χρησιμοποιώντας ερωτήματα με συνδεδεμένες παραμέτρους (bound parameters), η Laravel διασφαλίζει ότι οι είσοδοι του χρήστη δεν μπορούν να προκαλέσουν SQL injection.

Είτε χρησιμοποιείτε το Query Builder είτε το Eloquent ORM της Laravel, η πλατφόρμα σας προσφέρει ισχυρή προστασία από SQL injection μέσω της χρήσης προετοιμασμένων δηλώσεων. Αυτή η πρακτική διασφαλίζει ότι οι εισόδοι των χρηστών διαχειρίζονται με ασφάλεια, προστατεύοντας τη βάση δεδομένων από κακόβουλες επιθέσεις. Η χρήση αυτών των εργαλείων καθιστά τη Laravel μια ασφαλή επιλογή για την ανάπτυξη εφαρμογών web.

Προστασία από CSRF (Cross-Site Request Forgery)

Το CSRF είναι μια μορφή επίθεσης όπου ένας κακόβουλος χρήστης εκτελεί μη εξουσιοδοτημένες ενέργειες σε ένα web application εκ μέρους ενός αυθεντικοποιημένου χρήστη. Η Laravel προστατεύει από τέτοιου είδους επιθέσεις με την ενσωμάτωση CSRF tokens σε όλες τις φόρμες.

Παράδειγμα Προστασίας από CSRF

Σε κάθε φόρμα, η Laravel προσθέτει αυτόματα ένα κρυφό πεδίο με το CSRF token:

```
<form method="POST" action="/login">
    @csrf
    <!-- Πεδίο Email -->
    <input type="email" name="email" required>
    <!-- Πεδίο Κωδικού -->
```

```
<input type="password" name="password" required>
<!-- Υποβολή Φόρμας -->
<button type="submit">Σύνδεση</button>
</form>
```

Το CSRF token επαληθεύεται από τη Laravel σε κάθε αίτημα POST, διασφαλίζοντας ότι τα αιτήματα υποβάλλονται από εξουσιοδοτημένες πηγές.

Η ασφάλεια είναι αναπόσπαστο μέρος της ανάπτυξης οποιουδήποτε συστήματος διαχείρισης δεδομένων. Στο σύστημα διαχείρισης εργασιών μαθημάτων, έχουμε λάβει μια σειρά από μέτρα για να διασφαλίσουμε την ασφάλεια των δεδομένων και την προστασία των χρηστών. Χρησιμοποιώντας κωδικούς πρόσβασης με hash, sessions για την αυθεντικοποίηση, προστασία από SQL Injection και CSRF, και εφαρμόζοντας βέλτιστες πρακτικές ασφάλειας, διασφαλίζουμε ότι το σύστημα είναι ασφαλές και αξιόπιστο. Αυτές οι πρακτικές συμβάλλουν στη δημιουργία ενός ασφαλούς περιβάλλοντος για τους διδάσκοντες και τους φοιτητές, προστατεύοντας τα προσωπικά τους δεδομένα και διασφαλίζοντας την ακεραιότητα των ακαδημαϊκών τους πληροφοριών.

Κεφάλαιο 5ο: Συμπεράσματα και προτάσεις βελτίωσης

Η ανάπτυξη ενός συστήματος διαχείρισης εργασιών μαθημάτων με τη χρήση του Laravel αποδεικνύεται ως μια αποτελεσματική και αξιόπιστη λύση που ανταποκρίνεται στις ανάγκες τόσο των διδασκόντων όσο και των φοιτητών. Παρουσιάστηκε και αναλύθηκε μια λεπτομερής μελέτη της σχεδίασης της βάσης δεδομένων, η οποία περιλάμβανε την οργάνωση των πινάκων και των σχέσεων τους για την αποθήκευση και διαχείριση των δεδομένων των χρηστών, των μαθημάτων και των εργασιών. Η ενσωμάτωση σύγχρονων πρακτικών ασφαλείας, όπως η χρήση hash για τους κωδικούς πρόσβασης, η προστασία από SQL injection και η χρήση CSRF tokens, διασφαλίζει την ακεραιότητα και την εμπιστευτικότητα των δεδομένων, καθιστώντας το σύστημα ασφαλές και αξιόπιστο. Η χρησιμοποίηση ισχυρών εργαλείων διαχείρισης δεδομένων, όπως το Eloquent ORM και το Query Builder της Laravel, επιτρέπει την αποδοτική και ασφαλή αλληλεπίδραση με τη βάση δεδομένων, προσφέροντας ευελιξία και επεκτασιμότητα. Παρουσιάστηκε επίσης η χρήση του Bootstrap για τη δημιουργία μιας φιλικής προς τον χρήστη διεπαφής, η οποία διευκολύνει την πλοήγηση και τη χρήση του συστήματος τόσο από τους διδάσκοντες όσο και από τους φοιτητές. Με αυτές τις προσεγγίσεις, υλοποιήθηκε ένα σύστημα που όχι μόνο καλύπτει τις λειτουργικές απαιτήσεις αλλά και διασφαλίζει την προστασία και την ακεραιότητα των δεδομένων των χρηστών του, παρέχοντας μια ολοκληρωμένη και ασφαλή εμπειρία χρήσης.

Βάση Δεδομένων και Δομές Πινάκων

Η βάση δεδομένων του συστήματος διαχείρισης εργασιών μαθημάτων είναι σχεδιασμένη με προσοχή για να υποστηρίζει τη λειτουργικότητα και την ακεραιότητα των δεδομένων. Οι κύριοι πίνακες όπως users, courses, coursetasks, courseattend και files οργανώνουν τα δεδομένα με τρόπο που διευκολύνει την αποθήκευση, την ανάκτηση και την επεξεργασία τους. Οι σχέσεις μεταξύ των πινάκων, όπως οι σχέσεις ένα-προς-πολλά και πολλά-προς-πολλά, επιτρέπουν την αποτελεσματική διαχείριση των δεδομένων και τη σωστή αλληλεπίδραση μεταξύ των οντοτήτων του συστήματος.

Ασφάλεια

Η ασφάλεια είναι ζωτικής σημασίας για την προστασία των δεδομένων και των χρηστών του συστήματος. Μέσω της χρήσης ασφαλών κωδικών πρόσβασης που αποθηκεύονται με hash, της προστασίας από SQL injection με προετοιμασμένες δηλώσεις, και της χρήσης CSRF tokens για την αποτροπή επιθέσεων Cross-Site Request Forgery, το σύστημα διασφαλίζει την ακεραιότητα και την εμπιστευτικότητα των δεδομένων. Η χρήση sessions για την αυθεντικοποίηση διασφαλίζει ότι μόνο

εξουσιοδοτημένοι χρήστες έχουν πρόσβαση στις ευαίσθητες πληροφορίες και λειτουργίες του συστήματος.

Διαχείριση Χρηστών και Αυθεντικοποίηση

Το σύστημα διαχείρισης χρηστών παρέχει μια ευέλικτη και ασφαλή πλατφόρμα για την εγγραφή, τη σύνδεση και τη διαχείριση των χρηστών. Οι χρήστες μπορούν να εγγραφούν ως διδάσκοντες ή φοιτητές, και να συνδεθούν στο σύστημα για να έχουν πρόσβαση στις αντίστοιχες λειτουργίες. Η αυθεντικοποίηση με τη χρήση sessions και η δυνατότητα ανάκτησης κωδικού πρόσβασης διασφαλίζουν την απρόσκοπτη και ασφαλή πρόσβαση στο σύστημα.

Διαχείριση Μαθημάτων και Εργασιών

Η δυνατότητα δημιουργίας, επεξεργασίας και διαγραφής μαθημάτων και εργασιών από τους διδάσκοντες επιτρέπει την ευέλικτη διαχείριση της εκπαιδευτικής διαδικασίας. Οι φοιτητές μπορούν να εγγραφούν σε μαθήματα χρησιμοποιώντας μοναδικούς κωδικούς, να υποβάλλουν εργασίες και να λαμβάνουν βαθμολογίες και ανατροφοδότηση από τους διδάσκοντες. Η πλατφόρμα προσφέρει ένα εύχρηστο περιβάλλον εργασίας που διευκολύνει την αλληλεπίδραση μεταξύ διδασκόντων και φοιτητών.

Εργαλεία και Τεχνολογίες

Η χρήση του Laravel, ενός ισχυρού PHP framework, διευκολύνει την ανάπτυξη και τη συντήρηση του συστήματος. Το Eloquent ORM και το Query Builder παρέχουν ισχυρά εργαλεία για τη διαχείριση των δεδομένων, ενώ οι migrations επιτρέπουν τη διαχείριση της βάσης δεδομένων με προγραμματιστικό τρόπο. Επιπλέον, η ενσωμάτωση του Bootstrap προσφέρει μια φιλική προς τον χρήστη διεπαφή, καθιστώντας την πλοήγηση και τη χρήση του συστήματος ευχάριστη και αποτελεσματική.

Η ανάπτυξη ενός συστήματος διαχείρισης εργασιών μαθημάτων με τη χρήση του Laravel αποτελεί μια επιτυχημένη προσέγγιση που συνδυάζει την ασφάλεια, την ευελιξία και την αποδοτικότητα. Η προσεκτική σχεδίαση της βάσης δεδομένων, η εφαρμογή προηγμένων πρακτικών ασφαλείας και η χρήση σύγχρονων εργαλείων ανάπτυξης εξασφαλίζουν ότι το σύστημα ανταποκρίνεται στις ανάγκες των χρηστών και προστατεύει τα δεδομένα τους. Με αυτές τις πρακτικές, οι διδάσκοντες μπορούν να διαχειριστούν αποτελεσματικά τα μαθήματα και τις εργασίες τους, ενώ οι φοιτητές μπορούν να υποβάλλουν εργασίες και να λαμβάνουν την ανατροφοδότηση που χρειάζονται για την ακαδημαϊκή τους πρόοδο.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Moodle

[1] Dougiamas, M., & Taylor, P. C. (2003). Moodle: Using Learning Communities to Create an Open Source Course Management System. In EdMedia+ Innovate Learning (pp. 171-178). Association for the Advancement of Computing in Education (AACE).

[2] Rice, W. (2015). Moodle E-Learning Course Development: A complete guide to successful learning using Moodle 3. Packt Publishing Ltd.

[3] Moodle. (n.d.). Retrieved from <https://moodle.org/>

Blackboard

[4] Coates, H., James, R., & Baldwin, G. (2005). A Critical Examination of the Effects of Learning Management Systems on University Teaching and Learning. *Tertiary Education and Management*, 11(1), 19-36.

[5] Bradford, P., Porciello, M., Balkon, N., & Backus, D. (2007). The Blackboard Learning System: The Be All and End All in Educational Instruction? *Journal of Educational Technology Systems*, 35(3), 301-314.

[6] Blackboard. (n.d.). Retrieved from <https://www.blackboard.com/>

PHP

[7] Lerdorf, R., & Tatroe, K. (2002). *Programming PHP*. O'Reilly Media, Inc.

[8] Suraski, Z., & Gutmans, A. (2004). *PHP 5 Power Programming*. Prentice Hall Professional.

[9] PHP. (n.d.). Retrieved from <https://www.php.net/>

Laravel

[10] Stauffer, M. (2019). *Laravel: Up & Running: A Framework for Building Modern PHP Apps*. O'Reilly Media, Inc.

[11] Farina, F. (2020). *Learning Laravel: The Easiest Way*. Leanpub.

[12] Laravel. (n.d.). Retrieved from <https://laravel.com/>

Bootstrap

[13] Spurlock, J. (2013). *Bootstrap: Responsive Web Development*. O'Reilly Media, Inc.

[14] Cochran, J. (2014). *Bootstrap in 24 Hours, Sams Teach Yourself*. Sams Publishing.

[15] Bootstrap. (n.d.). Retrieved from <https://getbootstrap.com/>

MySQL

[16] DuBois, P. (2013). *MySQL Cookbook: Solutions for Database Developers and Administrators*. O'Reilly Media, Inc.

[17] Schwartz, B., Zaitsev, P., & Tkachenko, V. (2012). *High Performance MySQL: Optimization, Backups, and Replication*. O'Reilly Media, Inc.

[18] MySQL. (n.d.). Retrieved from <https://www.mysql.com/>

ΠΑΡΑΡΤΗΜΑ Α

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

UserController

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Session;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Mail;
use Illuminate\Support\Str;

class UserController extends Controller
{
    public function welcome()
    {
        return view('welcome');
    }

    public function showLoginForm()
    {
        return view('auth.login');
    }

    public function login(Request $request)
    {
        $request->validate([
            'email' => 'required|email',
            'password' => 'required|string',
            // 'g-recaptcha-response' => 'required|captcha',
        ]);

        $user = DB::table('user')->where('email', $request->email)->first();

        //if ($user && Hash::check($request->password, $user->password)) {
        if ($user && ($request->password=== $user->password)) {
            Session::put('user_id', $user->id);
            Session::put('user_role', $user->role);
            Session::put('user_email', $user->email);

            return redirect()->route('dashboard');
        }
    }
}
```

```

    return back()->withErrors(['email' => 'Invalid email or password.']);
}

public function showRegisterForm()
{
    return view('auth.register');
}

public function register(Request $request)
{
    $request->validate([
        'firstname' => 'required|string|max:255',
        'lastname' => 'required|string|max:255',
        'email' => 'required|email|unique:user,email',
        'password' => 'required|string|confirmed|min:4',
        'role' => 'required|string|in:student,teacher',
        // 'g-recaptcha-response' => 'required|captcha',
    ]);

    DB::table('user')->insert([
        'firstname' => $request->firstname,
        'lastname' => $request->lastname,
        'name' => $request->lastname.' '.$request->firstname,
        'email' => $request->email,
        'password' => Hash::make($request->password),
        'role' => $request->role,
        'created_at' => now(),
        'updated_at' => now(),
    ]);

    return redirect()->route('login')->with('success', 'Registration successful. Please login.');
```

```

}

public function showForgotPasswordForm()
{
    return view('auth.forgot-password');
}

public function sendResetLinkEmail(Request $request)
{
    $request->validate([
        'email' => 'required|email',
        // 'g-recaptcha-response' => 'required|captcha',
    ]);

    $user = DB::table('user')->where('email', $request->email)->first();

```

```

if ($user) {
    $token = Str::random(60);
    DB::table('password_resets')->insert([
        'email' => $request->email,
        'token' => $token,
        'created_at' => now(),
    ]);

    Mail::send('auth.emails.password', ['token' => $token], function ($message) use ($request) {
        $message->to($request->email);
        $message->subject('Password Reset Link');
    });

    return back()->with('success', 'Password reset link has been sent to your email.');
```

```

}

return back()->withErrors(['email' => 'No account found with that email address.']);
}

public function logout(Request $request)
{
    Session::flush();
    return redirect()->route('login');
}

public function dashboard()
{
    $user_id = Session::get('user_id');
    $user_role = Session::get('user_role');

    if ($user_role == 'teacher') {
        $courses = DB::table('course')->where('teacher_id', $user_id)->get();
        $tasks = DB::table('coursetask')
            ->join('course', 'coursetask.course_id', '=', 'course.id')
            ->where('course.teacher_id', $user_id)
            ->select('coursetask.*', 'course.title as course_title')
            ->get();
    } else {
        $courses = DB::table('course')
            ->join('courseattend', 'course.id', '=', 'courseattend.course_id')
            ->where('courseattend.student_id', $user_id)
            ->select('course.*', 'course.id as id')
            ->orderBy('id')
            ->get();
        $tasks = DB::table('coursetask')
            ->join('course', 'coursetask.course_id', '=', 'course.id')
            ->join('courseattend', 'course.id', '=', 'courseattend.course_id')
            ->where('courseattend.student_id', $user_id)

```

```

        ->select('coursetask.*', 'course.title as course_title')
        ->get();
    }

    return view('dashboard', compact('courses', 'tasks'));
}

public function showEnrollmentForm()
{
    $courses = DB::table('course')->get();
    return view('enroll', compact('courses'));
}

public function enroll(Request $request)
{
    $request->validate([
        'unique_code' => 'required|string|exists:course,unique_code',
    ]);

    $course = DB::table('course')->where('unique_code', $request->unique_code)->first();
    $user_id = Session::get('user_id');

    DB::table('courseattend')->insert([
        'course_id' => $course->id,
        'student_id' => $user_id,
        'created_at' => now(),
        'updated_at' => now(),
    ]);

    return redirect()->route('dashboard');
}
}

```

TaskController

```

<?php
namespace App\Http\Controllers;

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

class TaskController extends Controller
{
    public function index()
    {
        $user_id = Session::get('user_id');
    }
}

```

```

$tasks = DB::table('coursetask')
->join('course', 'coursetask.course_id', '=', 'course.id')
->where('course.teacher_id', $user_id)
->select('coursetask.*', 'course.title as course_title')
->get();

return view('tasks.index', compact('tasks'));
}

public function show($id)
{
    $task = DB::table('coursetask')->where('id', $id)->first();
    $user_role = Session::get('user_role');
    $user_id = Session::get('user_id');
    $submissions = [];
    $grades = [];

    if ($user_role == 'teacher') {
        $submissions = DB::table('files')
            ->join('user', 'files.student_id', '=', 'user.id')
            ->where('task_id', $id)
            ->select('files.*', 'user.firstname', 'user.lastname')
            ->get();
        $grades = DB::table('coursetaskattend')
            ->join('user', 'coursetaskattend.student_id', '=', 'user.id')
            ->where('task_id', $id)
            ->select('coursetaskattend.*', 'user.firstname', 'user.lastname')
            ->get();
    } elseif ($user_role == 'student') {
        $submissions = DB::table('files')
            ->join('user', 'files.student_id', '=', 'user.id')
            ->where('task_id', $id)
            ->where('student_id', $user_id)
            ->select('files.*', 'user.firstname', 'user.lastname')
            ->get();
        $grades = DB::table('coursetaskattend')
            ->where('task_id', $id)
            ->where('student_id', $user_id)
            ->get();
    }

    return view('tasks.show', compact('task', 'submissions', 'grades', 'user_role'));
}

public function create(Request $request)
{
    $user_id = Session::get('user_id');
    $courses = DB::table('course')->where('teacher_id', $user_id)->get();

```

```

$course_id = $request->query('course_id');
return view('tasks.create', compact('courses', 'course_id'));
}

public function store(Request $request)
{
    $request->validate([
        'course_id' => 'required|exists:course,id',
        'title' => 'required|string|max:255',
        'description' => 'nullable|string',
        'due_date' => 'required|date',
        'max_files' => 'required|integer|min:1',
        'allowed_file_types' => 'required|string|max:255',
    ]);

    DB::table('coursetask')->insert([
        'course_id' => $request->course_id,
        'title' => $request->title,
        'description' => $request->description,
        'due_date' => $request->due_date,
        'max_files' => $request->max_files,
        'allowed_file_types' => $request->allowed_file_types,
        'created_at' => now(),
        'updated_at' => now(),
    ]);

    return redirect()->route('courses.show', $request->course_id);
}

public function edit($id)
{
    $task = DB::table('coursetask')->where('id', $id)->first();
    $user_id = Session::get('user_id');
    $courses = DB::table('course')->where('teacher_id', $user_id)->get();
    return view('tasks.edit', compact('task', 'courses'));
}

public function update(Request $request, $id)
{
    $request->validate([
        'course_id' => 'required|exists:course,id',
        'title' => 'required|string|max:255',
        'description' => 'nullable|string',
        'due_date' => 'required|date',
        'max_files' => 'required|integer|min:1',
        'allowed_file_types' => 'required|string|max:255',
    ]);
}

```

```

DB::table('coursetask')->where('id', $id)->update([
    'course_id' => $request->course_id,
    'title' => $request->title,
    'description' => $request->description,
    'due_date' => $request->due_date,
    'max_files' => $request->max_files,
    'allowed_file_types' => $request->allowed_file_types,
    'updated_at' => now(),
]);

return redirect()->route('courses.show', $request->course_id);
}

public function destroy($id)
{
    $task = DB::table('coursetask')->where('id', $id)->first();
    DB::table('coursetask')->where('id', $id)->delete();
    return redirect()->route('courses.show', $task->course_id);
}

public function submit(Request $request, $id)
{
    $task = DB::table('coursetask')->where('id', $id)->first();
    $user_id = Session::get('user_id');
    $request->validate([
        'file' => 'required|file|mimes:' . $task->allowed_file_types,
    ]);

    $file = $request->file('file');
    $filePath = $file->store('public/submissions');

    DB::table('files')->insert([
        'task_id' => $id,
        'student_id' => $user_id,
        'file_path' => $filePath,
        'created_at' => now(),
        'updated_at' => now(),
    ]);

    // Δημιουργία καταχώρησης στο coursetaskattend κατά την πρώτη υποβολή
    $attendance = DB::table('coursetaskattend')
        ->where('task_id', $id)
        ->where('student_id', $user_id)
        ->first();

    if (!$attendance) {
        DB::table('coursetaskattend')->insert([
            'task_id' => $id,

```

```

        'student_id' => $user_id,
        'course_id' => $task->course_id,
        'created_at' => now(),
        'updated_at' => now(),
    ]);
}

return redirect()->route('tasks.show', $id);
}

public function delete_submission($id)
{
    $submission = DB::table('files')->where('id', $id)->first();
    Storage::delete($submission->file_path);
    DB::table('files')->where('id', $id)->delete();

    return redirect()->route('tasks.show', $submission->task_id);
}

public function grade(Request $request, $id)
{
    $request->validate([
        'grade' => 'required|numeric|min:0|max:100',
        'student_id' => 'required|exists:user,id'
    ]);

    DB::table('coursetaskattend')
        ->where('task_id', $id)
        ->where('student_id', $request->student_id)
        ->update([
            'grade' => $request->grade,
            'updated_at' => now(),
        ]);

    return back();
}
}

```

Web.php

```

<?php
use App\Http\Controllers\CourseController;
use App\Http\Controllers\TaskController;
use App\Http\Controllers\UserController;
use Illuminate\Support\Facades\Route;

Route::get('/', [UserController::class, 'welcome']->name('welcome'));
Route::get('/login', [UserController::class, 'showLoginForm']->name('login'));
Route::post('/login', [UserController::class, 'login']);

```

```

Route::post('/logout', [UserController::class, 'logout'])->name('logout');
Route::get('/dashboard', [UserController::class, 'dashboard'])->name('dashboard');
Route::get('/register', [UserController::class, 'showRegisterForm'])->name('register');
Route::post('/register', [UserController::class, 'register']);
Route::get('/forgot-password', [UserController::class, 'showForgotPasswordForm'])->name('password.request');
Route::post('/forgot-password', [UserController::class, 'sendResetLinkEmail'])->name('password.email');
Route::get('/enroll', [UserController::class, 'showEnrollmentForm'])->name('enroll.form');
Route::post('/enroll', [UserController::class, 'enroll'])->name('enroll');

// Routes for managing courses
Route::get('/courses', [CourseController::class, 'index'])->name('courses.index');
Route::get('/courses/create', [CourseController::class, 'create'])->name('courses.create');
Route::post('/courses', [CourseController::class, 'store'])->name('courses.store');
Route::get('/courses/{id}', [CourseController::class, 'show'])->name('courses.show');
Route::get('/courses/{id}/edit', [CourseController::class, 'edit'])->name('courses.edit');
Route::put('/courses/{id}', [CourseController::class, 'update'])->name('courses.update');
Route::delete('/courses/{id}', [CourseController::class, 'destroy'])->name('courses.destroy');

// Routes for managing tasks
Route::get('/tasks', [TaskController::class, 'index'])->name('tasks.index');
Route::get('/tasks/create', [TaskController::class, 'create'])->name('tasks.create');
Route::post('/tasks', [TaskController::class, 'store'])->name('tasks.store');
Route::get('/tasks/{id}', [TaskController::class, 'show'])->name('tasks.show');
Route::get('/tasks/{id}/edit', [TaskController::class, 'edit'])->name('tasks.edit');
Route::put('/tasks/{id}', [TaskController::class, 'update'])->name('tasks.update');
Route::delete('/tasks/{id}', [TaskController::class, 'destroy'])->name('tasks.destroy');
Route::post('/tasks/{id}/submit', [TaskController::class, 'submit'])->name('tasks.submit');
Route::delete('/tasks/{submission_id}/delete', [TaskController::class, 'delete_submission'])->name('tasks.delete_submission');
Route::post('/tasks/{id}/grade', [TaskController::class, 'grade'])->name('tasks.grade');

```