

ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΤΜΗΜΑΤΟΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ ΔΙΑΧΕΙΡΙΣΗΣ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΜΑΘΗΜΑΤΩΝ

Διπλωματική Εργασία

του

Τσιατμά Ιωάννη

Θεσσαλονίκη, Νοέμβριος 2021

ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ ΔΙΑΧΕΙΡΙΣΗΣ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΜΑΘΗΜΑΤΩΝ

Τσιατμάς Ιωάννης

Πτυχίο Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας, 1997

Διπλωματική Εργασία

υποβαλλόμενη για τη μερική εκπλήρωση των απαιτήσεων του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΤΙΤΛΟΥ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ
ΠΛΗΡΟΦΟΡΙΚΗ

Επιβλέπων Καθηγητής
Χατζηγεωργίου Αλέξανδρος

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την / /

Χατζηγεωργίου Αλέξανδρος

Κασκάλης Θεόδωρος

Αμπατζόγλου Απόστολος

Τσιατμάς Ιωάννης

.....

Περίληψη

Η ραγδαία ανάπτυξη της τεχνολογίας έχει ως αποτέλεσμα την εισαγωγή νέων μεθόδων διδασκαλίας στην εκπαίδευση. Οι δυνατότητες των νέων τεχνολογιών, να κατασκευάζουν αλληλεπιδραστικές εφαρμογές και γενικότερα να αναπτύσσουν περιβάλλοντα μάθησης, τις καθιστά ιδιαίτερα χρήσιμα εργαλεία στα χέρια των εκπαιδευτικών όλων των βαθμίδων, τόσο στη δημόσια, όσο και στην ιδιωτική εκπαίδευση.

Η παρούσα διπλωματική εργασία αποσκοπεί στη δημιουργία μιας διαδικτυακής εφαρμογής διαχείρισης ηλεκτρονικών μαθημάτων, η οποία παρέχει σε εκπαιδευτές και εκπαιδευόμενους ένα δυναμικό περιβάλλον αλληλεπίδρασης μέσω ιστού, με στόχο την οργάνωση, αποθήκευση και παρουσίαση ψηφιακού εκπαιδευτικού υλικού. Πιο συγκεκριμένα, παρέχει τη δυνατότητα σε χρήστες με ρόλο εκπαιδευτή να δημιουργήσουν μαθήματα, να τα οργανώσουν σε ενότητες και να τα εμπλουτίσουν με το κατάλληλο υλικό (κείμενα, παρουσιάσεις, εικόνες, διαγράμματα, κλπ), ενώ ταυτόχρονα, δίνει τη δυνατότητα σε χρήστες με ρόλο εκπαιδευόμενου να αξιοποιήσουν το υλικό αυτό στα πλαίσια της εκπαιδευτικής διαδικασίας. Επίσης, παρέχει τη δυνατότητα στους εκπαιδευτές και εκπαιδευόμενους να επικοινωνούν μεταξύ τους, ανταλλάσσοντας μηνύματα και αρχεία.

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκαν τεχνολογίες αιχμής, βασισμένες στις γλώσσες προγραμματισμού PHP, HTML, CSS και JavaScript. Ειδικότερα, για την ανάπτυξη του back-end έχει χρησιμοποιηθεί η MySQL ως σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων και το Laravel, ένα δωρεάν, ανοιχτού κώδικα PHP framework. Για την ανάπτυξη του front-end έχει χρησιμοποιηθεί το Bootstrap, ένα ανοιχτού κώδικα web design framework. Όλα τα frameworks και οι πρόσθετες βιβλιοθήκες που χρησιμοποιήθηκαν, επιτάχυναν και διευκόλυναν σε πολλά σημεία την ανάπτυξη της εφαρμογής.

Λέξεις Κλειδιά: e-learning, Συστήματα Διαχείρισης Μάθησης, LMS, Moodle, Ε-τάξη, Διαδικτυακή εφαρμογή, MVC, Laravel, PHP, HTML, CSS, JavaScript, MySQL, Bootstrap

Abstract

The introduction of new teaching methods in education is the result of the rapid development of technology. The possibilities of new technologies, to build interactive applications and in general to develop learning environments, make them particularly useful tools in the hands of teachers of all levels, both in public and private education.

This master thesis aims to create an online e-learning management application, which provides educators and learners with a dynamic web-based interaction environment, with the aim of organizing, storing and presenting digital educational material. More specifically, it enables users with an instructor role to create lessons, organize them into sections and enrich them with the appropriate material (texts, presentations, pictures, diagrams, etc.), while at the same time, enables users with a role trainee to use this material in the educational process. It also enables trainers and learners to communicate with each other, exchanging messages and files.

State-of-the-art technologies based on PHP, HTML, CSS and JavaScript were used to develop the application. In particular, for the development of the back-end, MySQL has been used as a relational database management system and Laravel, a free, open source PHP framework. Bootstrap, an open source web design framework, has been used to develop the front-end. All the frameworks and additional libraries used, accelerated and facilitated in many places the development of the application.

Keywords: e-learnng, Learning Management Systems, LMS, Moodle, E-class, Web application, MVC, Laravel, PHP, HTML, CSS, JavaScript, MySQL, Bootstrap

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω την γυναίκα μου κ. Κράντου Αλεξία για την αμέριστη συμπαράσταση, χωρίς αυτήν δε θα τα είχα καταφέρει. Επίσης το κίνητρο μου, τα παιδιά μου, Κωνσταντίνο-Ιωάννη και Αντώνιο-Βίκτωρ.

Θα ήθελα επίσης να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Χατζηγεωργίου Αλέξανδρο για την εμπιστοσύνη που μου έδειξε με την ανάθεση της παρούσας διπλωματικής εργασίας. Οι γνώσεις που μου μετέδωσε από τα μαθήματά και η καθοδήγησή του σε όλη τη διάρκεια εκπόνησης της εργασίας υπήρξαν ιδιαίτερα πολύτιμες και καθοριστικές.

Περιεχόμενα

1. Εισαγωγή	13
1.1 Πρόβλημα – Σημαντικότητα του θέματος	13
1.2 Σκοπός – Στόχοι	14
1.3 Διάρθρωση της μελέτης	14
2. Ηλεκτρονική Μάθηση – Θεωρητικό υπόβαθρο	16
2.1 Ηλεκτρονική Μάθηση	16
2.2 Κατηγορίες Ηλεκτρονικής Μάθησης	17
2.3 Πλεονεκτήματα – Μειονεκτήματα της Ηλεκτρονικής Μάθησης	18
2.4 Συστήματα Διαχείρισης Εκπαίδευσης (Learning Management System, LMS)	20
2.5 Πλατφόρμες LMS	22
2.5.1 Moodle	22
2.5.2 Υπηρεσίες του Moodle	23
2.5.3 E-Class	25
2.6 Ηλεκτρονική Μάθηση και εκπαιδευτικό υλικό	25
3 Τεχνολογίες που χρησιμοποιήθηκαν	27
3.1 HTML	27
3.1.1 Εισαγωγικά στοιχεία της HTML 5	29
3.1.2 Χαρακτηριστικά της HTML5	30
3.1.3 Νέα στοιχεία – ετικέτες της HTML5	31
3.2 Bootstrap	32
3.2.1 Ιστορικά στοιχεία	33
3.2.2 Χαρακτηριστικά	33
3.2.3 Δομή και λειτουργία	34
3.2.4 Χρήση	35
3.3 Laravel	38
3.3.1 Χαρακτηριστικά – πλεονεκτήματα του Laravel	38
3.3.2 Το μοντέλο MVC	40
3.3.3 Routing	42
3.3.4 Middleware	44
3.3.5 Controllers	44
3.3.6 Views	46
3.3.7 Blade Templates	47

3.3.8 Βάσεις Δεδομένων	51
3.3.9 Migrations	52
3.3.10 Query Builder	54
3.3.11 Eloquent ORM	55
3.4 MySQL	61
4 Συγκρίνοντας το Laravel με άλλα PHP Frameworks	63
4.1 Χρήση της PHP	63
4.2 Γιατί να χρησιμοποιήσουμε ένα PHP Framework	64
4.3 Κριτήρια για την επιλογή του σωστού Framework	66
4.4 Περιγραφή και βασικά χαρακτηριστικά του Laravel	67
4.5 Περιγραφή και βασικά χαρακτηριστικά του Symfony	69
4.6 Περιγραφή και βασικά χαρακτηριστικά του CodeIgniter	71
4.7 Συγκρίνοντας το Laravel με το Symfony	73
4.8 Συγκρίνοντας το Laravel με το CodeIgniter	75
4.9 Επιλογή PHP Framework σύμφωνα με τη δημοτικότητα και τις τάσεις της αγοράς	77
5 Περιγραφή και Δομή της εφαρμογής	80
5.1 Περιγραφή	80
5.2 Επιπλέον πακέτα (packages) που χρησιμοποιήθηκαν στο Laravel	81
5.3 Βάση δεδομένων	83
5.4 Models	87
5.5 Routing	88
5.6 Controllers	90
5.7 Views	93
6. Εικονογραφημένος οδηγός της εφαρμογής	97
6.1 Λειτουργίες που αφορούν όλους τους χρήστες της εφαρμογής	97
6.2 Λειτουργίες που αφορούν χρήστες με ρόλο Διαχειριστή	101
6.3 Λειτουργίες που αφορούν χρήστες με ρόλο Εκπαιδευτή	105
6.4 Λειτουργίες που αφορούν χρήστες με ρόλο Εκπαιδευόμενου	108
6.5 Λοιπές λειτουργίες της εφαρμογής	110
6.5.1 Σύστημα επικοινωνίας Εκπαιδευτών και Εκπαιδευόμενων	110
6.5.2 Εγχειρίδια χρήσης	113
7. Συμπεράσματα – Επεκτάσεις	114
7.1 Συμπεράσματα	114

7.2 Επεκτάσεις	115
Βιβλιογραφία	117
Ιστοσελίδες	118

Κατάλογος Εικόνων

Εικόνα 1: Αρχική σελίδα Moodle	22
Εικόνα 2 : Ο πρώτος web browser του Τιμ Μπέρνερς Λι	28
Εικόνα 3 : Το λογότυπο της HTML5	29
Εικόνα 4 : Το λογότυπο του Bootstrap	31
Εικόνα 5 : Το λογότυπο του Laravel	37
Εικόνα 6 : Τα μέρη του μοντέλου MVC	40
Εικόνα 7: Διαδρομή που εκτελεί τη μέθοδο ενός Controller	41
Εικόνα 8: HTTP μέθοδοι που υποστηρίζει το Laravel	42
Εικόνα 9: Διαδρομή με παράμετρο που εκτελεί τη μέθοδο ενός Controller	42
Εικόνα 10: Ομάδα διαδρομών (Route Group)	42
Εικόνα 11: Παράδειγμα ελεγκτή (Controller)	44
Εικόνα 12: Παράδειγμα διαδρομής (route)	45
Εικόνα 13: Παράδειγμα προβολής (view)	46
Εικόνα 14: Παράδειγμα διαδρομής με κλήση προβολής	46
Εικόνα 15: Παράδειγμα χρήσης μεταβλητής σε Blade αρχείο	47
Εικόνα 16: Παράδειγμα δήλωσης «if»	47
Εικόνα 17: Παραδείγματα βρόχων (loops)	48
Εικόνα 18: Παράδειγμα εντολής @include	48
Εικόνα 19: Παράδειγμα κύριου Blade Template	49
Εικόνα 20: Παράδειγμα θυγατρικού Blade Template	50
Εικόνα 21: Παράδειγμα Migration	51
Εικόνα 22: Παράδειγμα Migration αρχείου	52
Εικόνα 23: Παράδειγμα Query Builder ερωτήματος	53
Εικόνα 24: Παράδειγμα Query Builder ερωτήματος	54
Εικόνα 25: Παράδειγμα Eloquent μοντέλου (Model Course)	55
Εικόνα 26: Παράδειγμα ανάκτησης όλων των εγγραφών από τον πίνακα courses, χρησιμοποιώντας το μοντέλο Course	56
Εικόνα 27: Παράδειγμα ανάκτησης όλων των μαθημάτων (courses) που είναι ορατά (η στήλη status έχει την τιμή 1)	56
Εικόνα 28: Παράδειγμα διαγραφής μαθήματος, χρησιμοποιώντας τη μέθοδο delete()	57

Εικόνα 29: Παράδειγμα προσθήκης νέου μαθήματος, χρησιμοποιώντας τη μέθοδο create()	57
Εικόνα 30: Παράδειγμα ενημέρωσης μαθήματος, χρησιμοποιώντας τη μέθοδο update()	57
Εικόνα 31: Παράδειγμα σχέσης ένα προς πολλά – η συνάρτηση hasMany()	58
Εικόνα 32: Παράδειγμα ανάκτησης δεδομένων από συσχετιζόμενα μοντέλα	59
Εικόνα 33: Παράδειγμα σχέσης ένα προς πολλά – η συνάρτηση belongsTo()	60
Εικόνα 34: Παράδειγμα μοντέλου πελάτη-εξυπηρετητή	60
Εικόνα 35: Κατάταξη των δημοφιλέστερων DBMS	61
Εικόνα 36 : Χρήση της PHP στις διαδικτυακές εφαρμογές	62
Εικόνα 37 : Θέση της PHP σε σχέση με άλλες γλώσσες προγραμματισμού	63
Εικόνα 38 : Το PHP πλαίσιο Laravel	66
Εικόνα 39 : Το PHP πλαίσιο Symfony	68
Εικόνα 40 : Το PHP πλαίσιο CodeIgniter	70
Εικόνα 41 : Google Trends [2019]	77
Εικόνα 42 : Stack Overflow Trends [2019]	77
Εικόνα 43: Διάγραμμα οντοτήτων - συσχετίσεων της βάσης δεδομένων	83
Εικόνα 44: Η δομή του migration αρχείου του πίνακα «courses»	84
Εικόνα 45: Τα πεδία του πίνακα courses	85
Εικόνα 46: Το μοντέλο Lesson	86
Εικόνα 47: Οι ομάδες διαδρομών της εφαρμογής	87
Εικόνα 48: Οι μέθοδοι, οι διαδρομές και οι ελεγκτές της ομάδας διαδρομών για τους Students	89
Εικόνα 49: Ομάδα διαδρομών μόνο για πιστοποιημένους χρήστες με ρόλο Student	89
Εικόνα 50: Οι ελεγκτές (controllers) της εφαρμογής	90
Εικόνα 51: Οι συναρτήσεις του ελεγκτή EnrollmentController	91
Εικόνα 52: Οργάνωση των αρχείων προβολών (views) της εφαρμογής	93
Εικόνα 53: Το αρχείο mainlayout.blade.php, που περιέχει την κύρια διάταξη της Εφαρμογής	94
Εικόνα 54: Παράδειγμα αρχείου προβολής της εφαρμογής	95
Εικόνα 55: Δημιουργία λογαριασμού	96
Εικόνα 56: Εγγραφή νέου χρήστη	97
Εικόνα 57: Σύνδεση χρήστη	98

Εικόνα 58: Αλλαγή προφίλ από το πτυσσόμενο μενού	98
Εικόνα 59: Αλλαγή προφίλ από το κάθετο αριστερό μενού	98
Εικόνα 60: Αλλαγή προσωπικών στοιχείων προφίλ	99
Εικόνα 61: Αλλαγή στοιχείων επικοινωνίας προφίλ	99
Εικόνα 62: Αλλαγή κωδικού χρήστη	100
Εικόνα 63: Διαχείριση χρηστών	101
Εικόνα 64: Λειτουργίες διαχείρισης χρηστών	101
Εικόνα 65: Στατιστικά στοιχεία	102
Εικόνα 66: Εγγεγραμμένοι χρήστες	102
Εικόνα 67: Εγγεγραμμένοι χρήστες ανά μήνα	103
Εικόνα 68: Μαθήματα που έχουν δημιουργηθεί	103
Εικόνα 69: Μεταφορτωμένα αρχεία	104
Εικόνα 70: Δημιουργία μαθήματος	105
Εικόνα 71: Αποθήκευση μαθήματος	105
Εικόνα 72: Προσθήκη ενότητας	106
Εικόνα 73: Ανέβασμα αρχείου	106
Εικόνα 74: Προσθήκη συνδέσμου	107
Εικόνα 75: Εγγραφή σε μάθημα	108
Εικόνα 76: Μήνυμα επιτυχούς εγγραφής σε μάθημα	108
Εικόνα 77: Απενεργοποίηση μαθήματος, μετά από επιτυχή εγγραφή	108
Εικόνα 78: Αποστολή νέου μηνύματος εκπαιδευτή	109
Εικόνα 79: Φόρμα αποστολής μηνύματος εκπαιδευτή	109
Εικόνα 80: Αποστολή νέου μηνύματος εκπαιδευόμενου	110
Εικόνα 81: Φόρμα αποστολής μηνύματος εκπαιδευόμενου	110
Εικόνα 82: Απάντηση μηνύματος	111
Εικόνα 83: Προβολή μηνύματος	111
Εικόνα 84: Απάντηση μηνύματος	112
Εικόνα 85: Εγχειρίδιο χρήσης εκπαιδευτή	112

Κατάλογος Πινάκων

Πίνακας 1 : Μερικές βασικές ετικέτες της HTML	27
Πίνακας 2 : Βασικά τεχνικά χαρακτηριστικά του Laravel	66
Πίνακας 3 : Βασικά τεχνικά χαρακτηριστικά του Symfony	68
Πίνακας 4 : Βασικά τεχνικά χαρακτηριστικά του CodeIgniter	70
Πίνακας 5 : Σύγκριση του Laravel με το Symfony	72
Πίνακας 6 : Σύγκριση του Laravel με το CodeIngiter	75
Πίνακας 7 : Github Stars [2021]	78
Πίνακας 8: Επιπλέον πακέτα Laravel, που χρησιμοποιήθηκαν στην εφαρμογή	82

1. Εισαγωγή

1.1 Πρόβλημα – Σημαντικότητα του θέματος

Τα τελευταία χρόνια η συμβολή του διαδικτύου στην ηλεκτρονική, εξ αποστάσεως εκπαίδευση ήταν καταλυτική. Οι πλατφόρμες ηλεκτρονικής μάθησης είναι η βασική τεχνολογική υποδομή λογισμικού για τα περιβάλλοντα ηλεκτρονικής μάθησης και εξ' αποστάσεως εκπαίδευσης. Είναι σε γενικές γραμμές διαδικτυακά λογισμικά, τα οποία μέσα από τη διεπαφή χρήσης του φυλλομετρητή ανταλλάσσουν πληροφορίες με μια βάση δεδομένων, η οποία περιέχει πληροφορίες για χρήστες, μαθήματα και περιεχόμενο. Χρησιμοποιούνται σε όλες τις βαθμίδες εκπαίδευσης (πρωτοβάθμια, δευτεροβάθμια και τριτοβάθμια), προσφέροντας αρκετά πλεονεκτήματα σε εκπαιδευτές και εκπαιδευόμενους, σε σχέση με την παραδοσιακή διαδικασία μάθησης. Εμπεριέχουν διάφορα εργαλεία, αναφορικά με την οργάνωση και τη διανομή του μαθησιακού υλικού, τη διαχείριση των μαθημάτων, την αξιολόγηση των μαθητών, καθώς και εργαλεία επικοινωνίας και συνεργασίας. Επίσης, διακριτοί ρόλοι χρηστών (εκπαιδευτής, εκπαιδευόμενος, διαχειριστής), ευκολία ανάρτησης και πρόσβασης σε ηλεκτρονικό υλικό, πρόσβαση μέσω εγγραφής/σύνδεσης χρήστη, τυποποιημένη διεπαφή χρήστη και δυνατότητα πρόσβασης από κινητές συσκευές είναι μερικά από τα τεχνικά χαρακτηριστικά που διαθέτουν.

Και στην Ελλάδα τα τελευταία χρόνια, παρατηρείται μια σημαντική αύξηση των διαθέσιμων εργαλείων ηλεκτρονικής μάθησης, τόσο από δημόσιους, όσο και από ιδιωτικούς φορείς. Χαρακτηριστικό παράδειγμα είναι η εκπαιδευτική πλατφόρμα η-τάξη (e-class). Η πλατφόρμα e-class είναι ένα ολοκληρωμένο Σύστημα Διαχείρισης Ηλεκτρονικών Μαθημάτων. Βασίζεται στη φιλοσοφία του λογισμικού ανοικτού κώδικα και βασική επιδίωξη της πλατφόρμας είναι η ενσωμάτωση των νέων τεχνολογιών και η εποικοδομητική χρήση του διαδικτύου στην εκπαιδευτική διαδικασία.

Βλέπουμε λοιπόν ότι όλο και περισσότερες διαδικτυακές εφαρμογές αναπτύσσονται με σκοπό τη μετάδοση και διαχείριση της μάθησης, παρέχοντας στους χρήστες τους, όχι μόνο ευκολία στην πρόσβαση σε μεγάλο όγκο πληροφοριών, αλλά και μια πληθώρα νέων διαδραστικών εκπαιδευτικών εργαλείων.

1.2 Σκοπός – Στόχοι

Η εργασία αυτή έχει ως σκοπό τη δημιουργία μιας διαδικτυακής εφαρμογής, που θα χρησιμοποιηθεί ως εργαλείο για τη διαχείριση ηλεκτρονικών μαθημάτων, προσφέροντας σε καθηγητές και μαθητές, τόσο της πρωτοβάθμιας, όσο και της δευτεροβάθμιας βαθμίδας εκπαίδευσης, ένα φιλικό και αλληλεπιδραστικό περιβάλλον μάθησης.

Πιο συγκεκριμένα, η εφαρμογή θα υποστηρίζει τρεις ρόλους χρηστών: τον ρόλο του διαχειριστή (Admin), τον ρόλο του εκπαιδευτή (Teacher) και τον ρόλο του εκπαιδευόμενου (Student). Τα ζητούμενα χαρακτηριστικά της εφαρμογής είναι:

- Η εγγραφή / σύνδεση χρήστη
- Η δημιουργία και διαχείριση προφίλ χρήστη
- Η δημιουργία ηλεκτρονικών μαθημάτων
- Η οργάνωση του ψηφιακού εκπαιδευτικού υλικού σε ενότητες
- Η εύκολη ανάρτηση και πρόσβαση στο ψηφιακό εκπαιδευτικό υλικό
- Η επικοινωνία καθηγητών και μαθητών μέσω μηνυμάτων και αρχείων
- Η εγγραφή / απεγγραφή μαθητή σε / από μάθημα
- Η παροχή στατιστικών στοιχείων για τους χρήστες και το εκπαιδευτικό υλικό

Τέλος, αξιοποιώντας όλες τις σύγχρονες τεχνολογίες που υπάρχουν σήμερα και χρησιμοποιώντας τα κατάλληλα back-end και front-end frameworks, θα σχεδιάσουμε και θα υλοποιήσουμε, μια δυναμική εφαρμογή ιστού με μοντέρνο design και προσαρμοστική (responsive) διεπαφή χρήστη, προσβάσιμη από οποιαδήποτε συσκευή (ηλεκτρονικό υπολογιστή, tablet και smartphone).

1.3 Διάρθρωση της μελέτης

Η παρούσα διπλωματική είναι οργανωμένη ως εξής:

Στο κεφάλαιο 2 θα παρουσιαστεί το θεωρητικό υπόβαθρο της εργασίας. Συγκεκριμένα, θα ασχοληθούμε με το τι είναι ηλεκτρονική μάθηση, ποιες είναι οι κατηγορίες στην ηλεκτρονική μάθηση και ποια πλεονεκτήματα και μειονεκτήματα παρέχουν. Επίσης, θα ασχοληθούμε με τα Συστήματα Διαχείρισης Εκπαίδευσης (LMS) και θα παρουσιάσουμε κάποιες από τις γνωστότερες πλατφόρμες LMS.

Στο κεφάλαιο 3, θα παρουσιαστούν οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Αυτό κρίνεται απαραίτητο για την καλύτερη κατανόηση της αρχιτεκτονικής και του τρόπου που δημιουργήθηκε η εφαρμογή. Συγκεκριμένα, θα αναφερθούμε και θα αναλύσουμε τα χαρακτηριστικά και τις δυνατότητες, τόσο των frameworks, όσο και των γλωσσών προγραμματισμού που χρησιμοποιήθηκαν.

Στο κεφάλαιο 4, θα παρουσιαστεί μια συνοπτική συγκριτική μελέτη πάνω στα τρία πιο δημοφιλή php frameworks που χρησιμοποιούν οι προγραμματιστές διαδικτυακών εφαρμογών σήμερα. Συγκεκριμένα, θα αναφερθούμε στο Laravel, το Symfony και το CodeIgniter.

Στο κεφάλαιο 5, θα παρουσιαστεί μια αναλυτική περιγραφή της εφαρμογής και των λειτουργιών που υποστηρίζει. Επίσης, θα παρουσιαστούν συνοπτικά τα βασικά δομικά στοιχεία που χρησιμοποιήθηκαν για την ανάπτυξη της.

Τέλος, στο κεφάλαιο 6, θα παραθέσουμε έναν εικονογραφημένο οδηγό με τις σημαντικότερες λειτουργίες της εφαρμογής, ο οποίος θα περιέχει και τις αντίστοιχες εικόνες (screenshots).

2. Ηλεκτρονική Μάθηση – Θεωρητικό υπόβαθρο

2.1 Ηλεκτρονική Μάθηση

Η ηλεκτρονική μάθηση e-learning είναι συνυφασμένη με την εξ αποστάσεως εκπαίδευση. Σύμφωνα με τη Wikipedia (Wikipedia - Ηλεκτρονική Μάθηση 2018), «η ηλεκτρονική μάθηση είναι η διαδικασία με την οποία κάποιος μαθαίνει με τη χρήση τεχνολογίας και Ηλεκτρονικών Υπολογιστών». Η ηλεκτρονική μάθηση μπορεί να γίνει με χρήση Η/Υ χωρίς να βασίζεται στο διαδίκτυο, Computer Based Instruction CBI, αλλά και με τη χρήση εργαλείων του διαδικτύου Web Based Training WBT. Στην έκθεση του Ευρωπαϊκού Δικτύου εξ Αποστάσεως Εκπαιδύσεως (European Distance Education Network, UNESCO 2001) γίνεται δεκτός ο ορισμός του e-learning τον οποίο εισηγούνται οι (Waller & Wilson, 2001): «E-Learning είναι η αποτελεσματική μαθησιακή διαδικασία η οποία προάγεται από τον συνδυασμό ψηφιακού διδακτικού περιεχομένου μαζί με (μαθησιακή) υποστήριξη και υπηρεσίες».

Έτσι ο εκπαιδευτικός και ο μαθητής μπορεί να επικοινωνήσει και να αναπτύξει μία αλληλεπίδραση με το περιεχόμενο διδασκαλίας χρησιμοποιώντας εργαλεία του διαδικτύου όπως email, teleconference, message instant κα. Η (Ζερδέβα, 2007) παρουσιάζει το e-learning ως μια διαδικασία μαθήσεως η οποία εμπεριέχει την WBT (η οποία επίσης περιέχει την CBI) και περιβάλλεται από την εξ αποστάσεως εκπαίδευση. Στην ηλεκτρονική μάθηση δε περιλαμβάνει μόνο η χρήση Η/Υ άλλα και άλλων τεχνολογιών όπως ραδιόφωνο, Video, υποστηρίζει όλες τις μορφές δεδομένων όπως κείμενο, εικόνα, animation, ήχος κ.α. και όλα αυτά συνδυάζονται σε ηλεκτρονικές εφαρμογές. Θα πρέπει να τονίσουμε ότι η ηλεκτρονική μάθηση ξεκίνησε με την έννοια της πολυμεσικής μάθησης, multimedia learning.

Την τελευταία δεκαετία με την εμφάνιση των έξυπνων τηλεφώνων (smart phone) αναπτύχθηκε μία άλλη μορφή μάθησης, το M-Learning (Mobile Learning). Ο συνδυασμός σύγχρονης τεχνολογίας κινητών τηλεφώνων με τη χρήση διαδικτύου υψηλών ταχυτήτων είναι συνυφασμένη με την έννοια της εξ αποστάσεως εκπαίδευσης, ιδιαίτερα σε απομακρυσμένες περιοχές. Επίσης στηρίζει μεγάλο φάσμα επιστημών όπως ιατρική, μηχανική, φυσική κ.α.

2.2 Κατηγορίες Ηλεκτρονικής Μάθησης

Η ηλεκτρονική μάθηση αποτελείται από τις παρακάτω κατηγορίες:

- Σύγχρονη Εκπαίδευση
- Ασύγχρονη Εκπαίδευση

Ο όρος Σύγχρονη Εκπαίδευση χρησιμοποιείται για να περιγράψει διαδικασίες μάθησης που λαμβάνουν χώρο την ίδια χρονική στιγμή, άλλα όχι απαραίτητα στον ίδιο χώρο (Moore, K., G. & Kearsley, G 2005). Στην σύγχρονη εκπαίδευση έχουμε την ίδια χρονική στιγμή αλληλεπίδραση του εκπαιδευτικού με τους μαθητές, ανταλλάσσουν περιεχόμενο, απόψεις, ερωτήσεις, χρησιμοποιώντας τα ηλεκτρονικά μέσα και τις εφαρμογές. Γίνεται κατανοητό ότι η όλη διαδικασία στηρίζεται σε υπηρεσίες διαδικτύου.

Αντίθετα η Ασύγχρονη Εκπαίδευση δεν απαιτεί την ταυτόχρονη παρουσία εκπαιδευτικού και μαθητή. Ο μαθητής έχει τη δυνατότητα να επικοινωνεί με τον εκπαιδευτικό οποιαδήποτε χρονική στιγμή, να έχει πρόσβαση στο υλικό και να μπορεί να διαμοιράζεται τις δικές του ασκήσεις και το υλικό του με τους μαθητές και τους εκπαιδευτικούς. Από τη μεριά του ο εκπαιδευτικός μπορεί χάρη σε αυτή την ιδιαιτερότητα να έχει ευελιξία στο διαμοιρασμό του υλικού του, άλλα και στην επικοινωνία με τον μαθητή, πχ έχει περισσότερο χρόνο να δημιουργήσει το υλικό του, να το εμπλουτίσει και να το χρησιμοποιήσει όποτε εκείνος κρίνει απαραίτητο και να απευθυνθεί στο μαθητή εξατομικευμένα. Από την άλλη όμως, η χρονική καθυστέρηση που υπάρχει στην επικοινωνία με τον μαθητή, ή έλλειψη αμεσότητας, δημιουργεί προβλήματα στη διαδικασία της μάθησης και της αλληλεπίδρασης.

Και τα δύο είδη είναι μέρος της εξ αποστάσεως διαδικτυακής εκπαίδευσης (eLearning) ή αλλιώς γνωστή με τον όρο "Τηλεκπαίδευση". Αυτό που συμβαίνει στο eLearning είναι ότι ο εκπαιδευτής και οι εκπαιδευόμενοι βρίσκονται σε διαφορετικούς χώρους και η έννοια της "τάξης" δημιουργείται εικονικά – π.χ. από τον υπολογιστή.

Με αυτόν τον τρόπο η διδασκαλία μπορεί να είναι με ασύγχρονη συνεργασία (asynchronous collaboration), με σύγχρονη συνεργασία (synchronous collaboration) ή με εξατομικευμένο ρυθμό (self-based), (Moore and Kearsley, 1996, Massicotte, 1997)

2.3 Πλεονεκτήματα – Μειονεκτήματα της Ηλεκτρονικής Μάθησης

Η Ηλεκτρονική Μάθηση παρέχει μία σειρά από σημαντικά πλεονεκτήματα στην διαδικασία της μάθησης (Cowan, 1995), όπως:

- Διαθέσιμο εκπαιδευτικό υλικό εφόσον υπάρχει αδιάλειπτη σύνδεση με το διαδίκτυο.
- Η διαδικασία της εκπαίδευσης δε σταματάει από φυσικούς παράγοντες όπως φυσικές καταστροφές μεταβολές του καιρού ή οποιαδήποτε άλλη ανωμαλία η οποία ανήκει στο χώρο του απρόβλεπτου πχ (επιδημία). Όλα τα παραπάνω δεν θα πρέπει να επηρεάζουν τη σύνδεση με το διαδίκτυο.
- Δεν χρειάζεται να υπάρχουν αίθουσες διδασκαλίας, γραφεία και οτιδήποτε άλλο υπάρχει σε ένα οργανωμένο χώρο διδασκαλίας, αρκεί μονό η σύνδεση στο διαδίκτυο.
- Το ότι το μάθημα είναι σε ηλεκτρονική μορφή επιτρέπει να περιέχει ήχο, video, εικόνα, διαδραστικές ασκήσεις, να είναι πλούσιο και ποικίλο σε περιεχόμενο.
- Εξαιτίας του γεγονότος ότι είναι σε ηλεκτρονική μορφή η διαδικασία της μάθησης είναι πιο αποτελεσματική να πετύχει τους εκπαιδευτικούς της στόχους. Τα παιδιά μπορούν να αφομοιώσουν καλύτερα το εκπαιδευτικό υλικό αφού χρησιμοποιούν περισσότερες αισθήσεις δηλαδή περισσότερες αντιπροσωπικές αναπαραστάσεις.
- Ο τρόπος μετάδοσης μπορεί να είναι είτε σύγχρονος, την ίδια χρονική στιγμή εξασφαλίζοντας έτσι και την άμεση επικοινωνία μαθητή - εκπαιδευτικού είτε ασύγχρονος, με περισσότερή ευελιξία στο μαθητή και στον εκπαιδευτικό για το διαμοιρασμό του υλικού.
- Η ηλεκτρονική μάθηση επιτρέπει τη γρήγορη ανανέωση του υλικού και ανταποκρίνεται περισσότερο στα νέα επιστημονικά δεδομένα.
- Αλληλεπίδραση με το μαθητή είναι σε πολύ μεγάλο βαθμό αναπτυγμένη
- Υπάρχει τμηματοποίηση και οργάνωση του υλικού. Αυτό επιτρέπει να δημιουργηθούν Βάσεις Δεδομένων με υλικό και με κοινή χρήση.

Ωστόσο η Ηλεκτρονική μάθηση δεν περιέχει μόνο πλεονεκτήματα αλλά έχει και μειονεκτήματα, όπως:

- Δεν υπάρχει η αίθουσα διδασκαλίας. Έτσι ο μαθητής χάνει την αίσθηση του χώρου και της θετικής ενέργειας που δίνει στη διδασκαλία. Ο μαθητής στο δικό του χώρο χαλαρώνει, δεν έχει τις αισθήσεις του σε εγρήγορση και μπορεί αυτό να τον αποπροσανατολίζει από τη διαδικασία της μάθησης.
- Υπάρχει σοβαρό πρόβλημα με τις πιστοποιήσεις και βεβαιώσεις σπουδών ή τα πτυχία που δίδονται από φορείς που παρέχουν Ηλεκτρονική Μάθηση. Πολλοί κρατικοί φορείς και οργανισμοί είτε στο ιδιωτικό τομέα, είτε στο δημόσιο δεν αναγνωρίζουν τέτοιου είδους πτυχία με αποτέλεσμα οι μαθητές να μη μπορούν να αναγνωρίσουν τυχόν επαγγελματικά τους δικαιώματα.
- Το οικονομικό κόστος είναι πολύ μεγάλο όταν δημιουργείς Ηλεκτρονική μάθηση. Τόσο από πλευράς εκπαιδευτικού οργανισμού, όπου πρέπει να δημιουργήσει εφαρμογές, εκπαιδευτικές πλατφόρμες που μπορούν να σηκώσουν το υλικό άλλα και να είναι λειτουργικό στον μαθητή. Επιφέρει μεγάλο οικονομικό κόστος αφού οι εφαρμογές χρειάζονται μεγάλη υπολογιστική ισχύ, επίσης χρειάζεται και εξειδικευμένο υπαλληλικό προσωπικό για να στηρίζει αυτές τις εφαρμογές. Το οικονομικό κόστος γίνεται ακόμη μεγαλύτερο όταν η Ηλεκτρονική Μάθηση παρέχεται στη Σύγχρονη Εκπαίδευση (on-time).
- Από τη πλευρά του εκπαιδευτικού είναι πάλι ακριβή, γιατί τα ηλεκτρονικά μαθήματα έχουν κόστος και οικονομικό και χρόνου.

Από όλα τα παραπάνω δε θα πρέπει να ξεχνάμε το βασικό αντικείμενο που είναι η μάθηση. Μάθηση, όμως είναι η διαδικασία με την οποία ο άνθρωπος προσλαμβάνει γνώσεις, διαδικασία νοητική, που δε θα μπορούσε να χαρακτηριστεί ηλεκτρονική, ή να προσλάβει άλλους τεχνολογικούς προσδιορισμούς, όσο και αν τα τεχνολογικά μέσα βοηθούν αυτή την διαδικασία. Αυτή η εξωτερική (έξω από τον άνθρωπο) υποστήριξη της μαθήσεως όταν γίνεται με οργανωμένο τρόπο είναι η εκπαίδευση.

Το e-learning, όπως κυρίως ορίζεται στην διεθνή βιβλιογραφία, είναι ένας συστηματικός και οργανωμένος τρόπος εξ αποστάσεως εκπαίδευσης, βασιζόμενος σε πολύ μεγάλο βαθμό στην χρήση των ΤΠΕ. Γι' αυτό θα μπορούσε ίσως να χρησιμοποιηθεί ο όρος «ηλεκτρονική τηλεκπαίδευση» αντί του «ηλεκτρονική μάθηση», ο οποίος τείνει να επικρατήσει στην ελληνική βιβλιογραφία.

2.4 Συστήματα Διαχείρισης Εκπαίδευσης (Learning Management System, LMS)

Για να στηριχθεί η Ηλεκτρονική μάθηση έπρεπε να βρεθούν τρόποι που να στηρίζουν ολόκληρη την εκπαιδευτική κοινότητα (εκπαιδευτικούς, μαθητές, υλικό). Γρήγορα μεγάλοι εκπαιδευτικοί οργανισμοί όπως Πανεπιστήμια, συνειδητοποίησαν ότι έπρεπε να οργανωθούν πάνω σε αυτό τον τομέα. Γι' αυτό έπρεπε να δημιουργήσουν έναν μηχανισμό που θα ανταποκρινόταν στο εγχείρημα της Ηλεκτρονικής Μάθησης. Στην ουσία έπρεπε μέσα από τις υπηρεσίες του διαδικτύου να παρέχουν αν όχι όλες, τις περισσότερες εκπαιδευτικές υπηρεσίες τους. Τη λύση ήρθαν να τη δώσουν τα Συστήματα Εκπαιδευτικής Διαχείρισης.

Το Σύστημα Διαχείρισης Μάθησης (Learning Management System, LMS) είναι συστήματα διαχείρισης που στηρίζονται στο Περιβάλλον του Διαδικτύου, παρέχοντας στους εκπαιδευτικούς και στους μαθητές ένα ολοκληρωμένο περιβάλλον διδασκαλίας και διαχείρισης της μάθησης με μία πληθώρα εφαρμογών και εργαλείων άμεσα αξιοποιήσιμα από την εκπαιδευτική κοινότητα. Το LMS αποτελεί μέρος της Ηλεκτρονικής Μάθησης e-learning και της εξ' αποστάσεως εκπαίδευσης. Σύμφωνα με τους (Δημητριάδης, Καραγιαννίδης, Πομπόρτσας, Τσιάτσος, 2007), είναι σύστημα διαχείρισης που βασίζεται σε τεχνολογίες διαδικτύου για να υποστηρίξουν δράσεις τηλεεκπαίδευσης με έξυπνο παιδαγωγικό ορθό τρόπο.

Τα συστήματα αυτά παρέχουν ένα ολοκληρωμένο εκπαιδευτικό περιβάλλον. Ο μαθητής μπορεί:

- Να δει το εκπαιδευτικό υλικό σε διάφορες μορφές (ήχο, εικόνα, κείμενο).
- Να ανεβάσει (upload) το υλικό του, τις ασκήσεις ή να κατεβάσει (download) εκπαιδευτικό υλικό. Μάλιστα η ξαναχρησιμοποίηση του υλικού είναι πολύ σημαντική σύμφωνα με τους (Montrieux, Hannelore, Vanderlinde, Ruben, Schellens, Tammy. Marez, Lieven De 2015), γιατί δίνει την δυνατότητα στους εμπλεκόμενους να ξαναγυρίζουν σε αυτό όταν το χρειάζονται.
- Να στείλει ηλεκτρονικά μηνύματα στον καθηγητή του και να πάρει απαντήσεις είτε άμεσα, είτε έμμεσα.

- Να έρθει σε επαφή με τους συμμαθητές του και να ανταλλάξει μέσα από Forums, chat ή από δωμάτια επικοινωνίας που παρέχει το LMS απόψεις και ιδέες.
- Να δει την πορεία της εκπαίδευσης του στο μάθημα εργασίας, βαθμολογία, πρόοδο, να έχει ηλεκτρονικό πορτοφόλιο.

Από τη μεριά του εκπαιδευτικού:

- Μπορεί να οργανώσει το μάθημα του σε ενότητες
- Να έχει πλήρη εικόνα του μαθητή, και των εργασιών του
- Να περνάει βαθμολογίες
- Να επικοινωνεί άμεσα η όχι με τους μαθητές
- Να ανεβάζει το υλικό του και να το αποθηκεύει
- Και να παρεμβαίνει στους μαθητές του όποτε κρίνει απαραίτητο.

Από τη μεριά του εκπαιδευτικού οργανισμού, όλες οι υπηρεσίες του οργανισμού, γραμματειακή υποστήριξη, υλικοτεχνική υποστήριξη, οικονομικές και κοινωνικές λειτουργίες εκπληρώνονται μέσα από συστήματα LMS. Παράλληλα προσδίδουν κύρος στον οργανισμό και εμφανές διαδικτυακό ίχνος και προσβασιμότητα στην εκπαιδευτική αγορά.

Σύμφωνα με τον (Long, Phillip, D. 2004) υπάρχουν τα εξής πλεονεκτήματα στα LMS

- Διαλειτουργικότητα
- Προσβασιμότητα
- Επαναχρησιμοποίηση
- Συντήρηση
- Προσαρμοστικότητα
- Ανθεκτικότητα

Εκτός όμως από πλεονεκτήματα έχουμε και μειονεκτήματα τα σημαντικότερα από αυτά είναι:

- Οι εκπαιδευτικοί πρέπει να είναι εξοικειωμένοι με τις νέες τεχνολογίες και να προσαρμόσουν τα μαθήματά τους ηλεκτρονικά

- Το κόστος για τη δημιουργία LMS είναι πολλές φορές καθοριστικό για το επίπεδο της παρεχόμενης υπηρεσίας.
- Υπάρχουν έρευνες που λένε ότι, η ηλεκτρονική μάθηση και η χρήση LMS αυξάνουν το χρόνο εργασίας στους εκπαιδευτικούς, χωρίς να υπάρχει ανάλογη αύξηση των απολαβών τους. Στους δε μαθητές μπορεί να τους αποπροσανατολίζει από τη διαδικασία της μάθησης.
- Υπάρχουν προβλήματα στη συντήρηση και υποστήριξη των LMS συστημάτων, όπως απαρχαιωμένο εκπαιδευτικό υλικό που πρέπει να διαγραφεί.

Τα συστήματα LMS αναπτύσσονται είτε από αυτόνομους προγραμματιστές developers είτε και από Συστήματα Διαχείρισης Περιεχομένου Content Management System CMS όπως το Wordpress, Joomla, Drupal. Πολλοί μπερδεύουν τα συστήματα LMS με CMS ωστόσο η διαφορά είναι εμφανής αφού στα συστήματα CMS μπορούν να δημιουργήσουν LMS και όχι το αντίθετο.

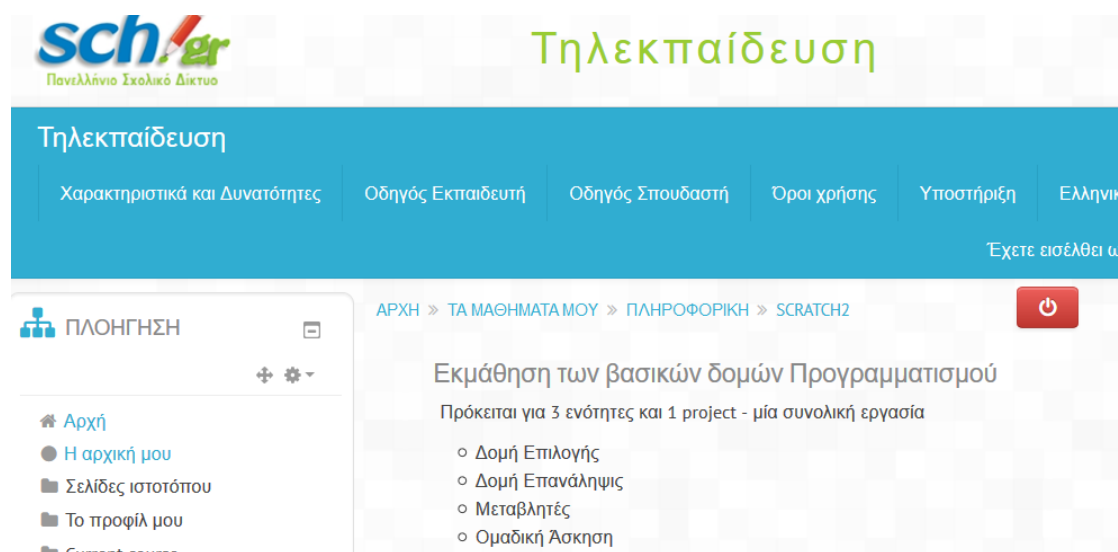
Πρόσφατα όμως έχει προστεθεί και ο όρος LCMS (LMS+CMS). Πρόκειται για ένα περιβάλλον όπου οι developers μπορούν να δημιουργούν, να αποθηκεύουν, να επαναχρησιμοποιούν, να διαχειρίζονται και να διανείμουν μαθησιακό περιεχόμενο από μια κεντρική δεξαμενή, συνήθως μια βάση δεδομένων, δηλαδή ένα περιβάλλον LMS για τη δημιουργία LMS. Σε ένα τέτοιο σύστημα, ο developer εισάγει και αποθηκεύει πόρους που θα χρησιμοποιηθούν για τη δημιουργία ενός αντικειμένου μάθησης και ταυτόχρονα δημιουργεί και αποθηκεύει το περιεχόμενο της μάθησης, το ελέγχει και το τροποποιεί κατάλληλα. Έτσι το LMS έχει πρόσβαση στο CMS και ανακτά το περιεχόμενο της μάθησης. Το LCMS συνδυάζει την ισχύ του CMS με την αρτιότητα του LMS. Παρακάτω θα γίνει μία σύντομη αναφορά στα πιο βασικά συστήματα LMS.

2.5 Πλατφόρμες LMS

2.5.1 Moodle

Προέρχεται από τα αρχικά των λέξεων Modular Object Oriented Development Learning Environment, είναι γραμμένο σε php κώδικά και εμφανίστηκε πρώτη φορά το 2002. Δημιουργήθηκε από τον Αυστραλό Martin Dougiamas και είναι ελεύθερο

λογισμικό διαχείρισης Μαθημάτων παρέχοντας ένα ολοκληρωμένο εκπαιδευτικό περιβάλλον που στηρίζεται στο πρότυπο SCORM (Sharable Content Object Reference Mode) Το Moodle τρέχει σε όλα τα λειτουργικά συστήματα και είναι mobile app σε όλες τις τεχνολογίες κινητών. Την εφαρμογή μπορεί κανείς να κατεβάσει από όλα τα ηλεκτρονικά καταστήματα κινητών, Google Play, App Store. Για να τρέξει το moodle χρειάζεται να υποστηρίζεται από ένα Server πχ Apache Server και υποστηρίζει όλες τις βάσεις δεδομένων, κυρίως την MySql (Τσιάτσος Θ, 2015).



Εικόνα 1: Αρχική σελίδα Moodle

2.5.2 Υπηρεσίες του Moodle

Το moodle διακρίνεται από δύο οπτικές γωνίες: η μία είναι του εκπαιδευτικού και η άλλη του Μαθητή. Έτσι με βάση την οπτική γωνία του εκπαιδευτικού, διακρίνουμε τις λειτουργίες που επιτελεί από δύο πλευρές: του υλικού και των δραστηριοτήτων.

Με βάση το υλικό:

- Εισαγωγή Αρχείων οποιασδήποτε μορφής
- Εισαγωγή βαθμολογίας
- Εισαγωγή ενοτήτων, ετικετών
- Δημιουργία Φακέλων
- Έλεγχος Προόδου
- Εισαγωγή πηγών, μεταδεδομένων

- Δημιουργία Συμβάντων
- Δημιουργία εργασιών με χρονικό περιορισμό
- Εισαγωγή περιεχομένου από άλλες πλατφόρμες

Με βάση τις δραστηριότητες:

- Δημιουργία πολλαπλών ασκήσεων με διαδραστικό περιεχόμενο πχ quiz, πολλαπλές ασκήσεις κ.α.
- Δημιουργία Blogs, forums, chat
- Δημιουργία Έρευνας
- Δημιουργία Βάσης Δεδομένων
- Δημιουργία online κειμένου
- Δημιουργία ομάδας

Με βάση την οπτική γωνία του Μαθητή:

- Ο μαθητής έχει προσωπικό προφίλ
- Προσωπικό πορτοφόλιο ασκήσεων, εργασιών
- Διαχείριση ηλεκτρονικής αλληλογραφίας
- Επικοινωνία με τα άλλα μέλη της τάξης και με τον εκπαιδευτικό

Ο μαθητής έχει κωδικό για να μπορεί να μπαίνει στη σελίδα της τάξης του και από κει μπορεί να διαχειρίζεται τις λειτουργίες που του επιτρέπονται (δικαιώματα χρηστών). Από την άλλη ο εκπαιδευτικός είναι αυτός που έχει τη διαχείριση της τάξης και δίνει τα δικαιώματα. Υπάρχουν τα εξής δικαιώματα ή ρόλοι χρηστών:

Admin: πλήρης δικαιώματα μπορεί να ορίσει δικαιώματα σε άλλους χρήστες όπως και να τους δημιουργήσει.

Teachers: Μπορούν να δημιουργήσουν μαθήματα, να αποδεχτούν μαθητές και να ανεβάσουν υλικό.

Students: πρόκειται για τους μαθητές που ανεβάζουν υλικό, λύνουν τις ασκήσεις τους, επικοινωνούν με τους εκπαιδευτικούς και έχουν το δικό τους πορτοφόλιο.

2.5.3 E-Class

Δημιουργήθηκε από το GUN Greek Universal Network, Ελληνικό Ακαδημαϊκό Δίκτυο και εμφανίστηκε πρώτη φορά το 2003. Είναι ευρέως διαδεδομένο στην Μέση Εκπαίδευση στην Ελλάδα αλλά και πολλά Ακαδημαϊκά ιδρύματα το έχουν υιοθετήσει λόγω της ευχρηστικότητά της. Στηρίζεται σε περιβάλλον διαδικτύου και στο πρότυπο SCORM.

Το E-Class είναι παρόμοιο με το Moodle και οι υπηρεσίες που παρέχονται είναι λίγο πολύ κοινές. Με το E-Class ο εκπαιδευτικός μπορεί να δημιουργήσει ηλεκτρονικά μαθήματα με δυνατότητα Login χωρίς κωδικό (ανοικτά μαθήματα), διαχείριση χρηστών, απόδοση δικαιωμάτων, διατήρηση και δημιουργία αντιγράφων ασφαλείας για το περιεχόμενο. Επίσης υποστήριξη πολυμεσικού υλικού, προσθήκη πηγών, γλωσσάρι, εισαγωγή βιβλιογραφίας κ.α.

Ακόμη, οι χρήστες έχουν τη δυνατότητα να χρησιμοποιούν πλειάδα εργαλείων και εφαρμογών, υπάρχουν εργαλεία ενημερώσεων και ειδοποιήσεων, τηλεργασία, επικοινωνία σε πραγματικό χρόνο, δημιουργία ομάδων και συνεργατική γραφή, βαθμολόγηση, απουσιολόγιο, υποβολή εργασιών, στατιστικά, δημιουργία ασκήσεων ανοικτού και κλειστού τύπου και άλλα πολλά.

Λόγω του εύχρηστου περιβάλλον και της απλοϊκότητας των εφαρμογών προτιμάται σε σχέση με τη πλατφόρμα Moodle. Ωστόσο η πλατφόρμα Moodle κυρίως σε εργαλεία ειδοποιήσεων και πραγματικού χρόνου παρέχει αρκετές εφαρμογές όπως forums, Chat, Blogs.

2.6 Ηλεκτρονική Μάθηση και εκπαιδευτικό υλικό

Απαραίτητο στοιχείο της ηλεκτρονικής μάθησης είναι η δημιουργία ηλεκτρονικών μαθημάτων που να στηρίζεται στη χρήση όλων των ηλεκτρονικών μορφών πληροφορίας (ήχος, Video, animation, εικόνα, κείμενο). Τα ηλεκτρονικά μαθήματα θα πρέπει να εμπεριέχουν στοιχεία αλληλεπίδραση και διαδραστικές ασκήσεις για να κρατούν το ενδιαφέρον των μαθητών. Πολλοί εκπαιδευτικοί οργανισμοί τόσο ιδιωτικοί όσο και δημόσιοι έχουν αναπτύξει εφαρμογές και εργαλεία.

Ένα χρήσιμο εργαλείο που έχει κατακλύσει το διαδίκτυο είναι τα MOOCs Massive Open Online Courses, Μαζικών Ανοικτών Διαδικτυακών Μαθημάτων. Πρόκειται για μαθήματα ανοικτά στο διαδίκτυο που στηρίζονται στη χρήση Video, γνωστά ως και βιντεομαθήματα. Κυρίως πρόκειται για tutorials τα οποία είναι διαβαθμισμένα με σαφείς εκπαιδευτικούς όρους. Το περιβάλλον των MOOCs παρέχει εργαλεία όπου οι μαθητές μπορούν να απαντήσουν σε ερωτήσεις, να διεξαγάγουν online test και να συμμετέχουν σε ασκήσεις. Επίσης σε ορισμένες περιπτώσεις το περιβάλλον επιτρέπει και την αλληλεπίδραση με τον εκπαιδευτή. Με την άνοδο των κοινωνικών μέσων δικτύωσης τα μαθήματα MOOCs σε αρκετές περιπτώσεις ενσωματώθηκαν σε εκπαιδευτικές σελίδες που έχουν δημιουργηθεί στα κοινωνικά μέσα δικτύωσης. Ορισμένα εκπαιδευτικά μαθήματα, σεμινάρια που στηρίζονται σε περιβάλλον Moocs παρέχουν και ένα είδος πιστοποίησης στους εκπαιδευόμενους.

Υπάρχουν πολλές εταιρείες που δημιουργούν MOOCs. Οι πιο γνωστές είναι η Coursera, Edx, Udacity. Ένα από τα πλεονεκτήματα των MOOCs είναι ότι ο μαθητής μπορεί να αυτορρυθμίζει τα μαθήματα με βάση το δικό του επίπεδο και η παρακολούθηση μπορεί να γίνει στο δικό του χρόνο. Ωστόσο δε παύει να είναι απρόσωπη, αφού απουσιάζει ο ενεργός ρόλος του δασκάλου - καθοδηγητού και το περιβάλλον της τάξης.

Εν κατακλείδι η ηλεκτρονική μάθηση ανοίγει νέους ορίζοντες στην διαδικασία της μάθησης. Ο όρος Τεχνολογίες Πληροφορικής στην Εκπαίδευση ΤΠΕ, ολοένα και εδραιώνεται στους εκπαιδευτικούς φορείς. Γίνεται έτσι αναγκαίο η προσαρμογή όλων (εκπαιδευτικών, μαθητών, φορέων) στα νέα δεδομένα της ηλεκτρονικής μάθησης, διαφορετικά η ίδια η μάθηση θα φαίνεται απαρχαιωμένη με το πέρασμα του χρόνου.

3 Τεχνολογίες που χρησιμοποιήθηκαν

3.1 HTML

Η HTML είναι η κύρια γλώσσα για την δημιουργία ιστοσελίδων. Το όνομά της είναι το ακρωνύμιο των λέξεων HyperText Markup Language, δηλαδή Γλώσσα Σήμανσης Υπερκειμένου και βασίζεται στη γλώσσα SGML (Standard Generalized Markup Language), που είναι ένα πολύ μεγαλύτερο σύστημα επεξεργασίας εγγράφων και δόμησης ιστοσελίδων του Παγκόσμιου Ιστού (WWW). Δεν θεωρείται γλώσσα προγραμματισμού, καθώς δεν μπορεί να δημιουργήσει δυναμικές λειτουργίες, αλλά μια περιγραφική γλώσσα αναπαράστασης κειμένου, πινάκων, εικόνων κλπ.

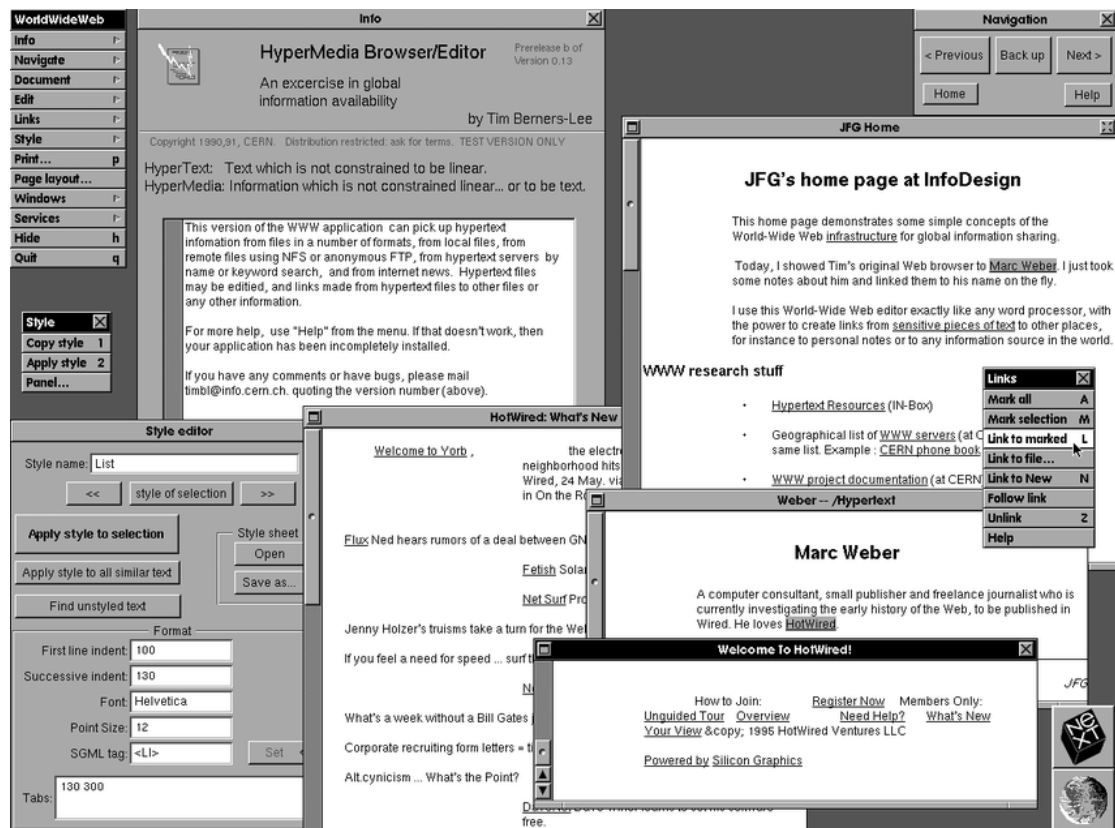
Η HTML γράφεται υπό μορφή στοιχείων HTML (HTML elements), τα οποία αποτελούνται από ετικέτες (tags). Κάθε ετικέτα περικλείεται από τα σύμβολα «μεγαλύτερο από» ">" και «μικρότερο από» "<". Συνήθως, λειτουργούν ανά ζεύγη (για παράδειγμα <h1> </h1>), με την πρώτη να ονομάζεται ετικέτα έναρξης (start tag) και τη δεύτερη ετικέτα λήξης (end tag). Το πρόγραμμα περιήγησης ιστοσελίδων (web browser), διαβάζει αρχεία HTML και δεν εμφανίζει τις ετικέτες που αυτά περιέχουν, αλλά τις χρησιμοποιεί ώστε να παρουσιάσει το περιεχόμενο τους σε σελίδες που μπορεί κανείς να διαβάσει και να ακούσει (wikipedia.org - HTML 2021).

Χρησιμοποιώντας τις κατάλληλες ετικέτες της HTML, δημιουργούμε ιστοσελίδες στις οποίες μπορούμε να ενσωματώσουμε κεφαλίδες, κείμενο, λίστες, πίνακες, εικόνες, βίντεο και υπερσυνδέσμους. Μερικές βασικές ετικέτες παρουσιάζονται στον πίνακα 1.

Ετικέτα	Περιγραφή
<head>	Ορίζει πληροφορίες σχετικά με το έγγραφο.
<title>	Ορίζει τον τίτλο του εγγράφου.
<link>	Ορίζει τη σχέση μεταξύ ενός εγγράφου και μιας εξωτερικής πηγής.
<body>	Ορίζει το κυρίως σώμα του εγγράφου.

<script>	Ορίζει ένα script.
<style>	Ορίζει πληροφορίες σχετικά με το στυλ του εγγράφου.
<a>	Ορίζει έναν υπερσύνδεσμο.
<h1> έως <h6>	Ορίζει επικεφαλίδες.
<div>	Ορίζει μια ενότητα σε ένα έγγραφο.
<p>	Ορίζει μια παράγραφο.
	Ορίζει μια εικόνα.
<table>	Ορίζει έναν πίνακα.
	Ορίζει μία αριθμητική (ταξινομημένη) λίστα.
	Ορίζει μία μη αριθμητική (μη ταξινομημένη) λίστα.
	Ορίζει ένα στοιχείο λίστας.

Πίνακας 1 : Μερικές βασικές ετικέτες της HTML



Εικόνα 2 : Ο πρώτος web browser του Τιμ Μπέρνερς Λι.

3.1.1 Εισαγωγικά στοιχεία της HTML 5

Σύμφωνα με τη Wikipedia (wikipedia.org - HTML 2021) « το 1980, ο φυσικός Τιμ Μπέρνερς Λι, ο οποίος εργαζόταν στο CERN, επινόησε το ENQUIRE, ένα σύστημα χρήσης και διαμοιρασμού εγγράφων για τους ερευνητές του CERN, και κατασκεύασε ένα πρωτότυπό του ». Αργότερα, στα τέλη του 1990, έφτιαξε την πρώτη προδιαγραφή της HTML και έγραψε τον browser (Εικόνα 2) και το λογισμικό εξυπηρητή.

Η HTML είναι μια πολύ εξελισσόμενη γλώσσα σήμανσης. Τα τελευταία χρόνια πολλές νέες εκδόσεις έχουν κυκλοφορήσει, επιτρέποντας στους δημιουργούς ιστοσελίδων να φτιάχνουν ιστοσελίδες με πολύ πιο εύκολο, όμορφο και αποδοτικό τρόπο. Το 1995 δημοσιεύτηκε η HTML 2.0, η οποία περιείχε όλες τις δυνατότητες της HTML 1.0 μαζί με πρόσθετα στοιχεία, καθιερώνοντας την HTML ως μια τυπική γλώσσα σήμανσης για τον σχεδιασμό και τη δημιουργία ιστοσελίδων. Το 1999 δημοσιεύτηκε η HTML 4.01 η οποία χρησιμοποιήθηκε ευρέως και ήταν μια πολύ σημαντική και επιτυχημένη έκδοση της HTML (w3schools.com – HTML History 2021)

Η HTML5 (Εικόνα 3) αποτελεί την πέμπτη και πιο πρόσφατη έκδοση της HTML και δημοσιεύτηκε το 2014 από το W3C (World Wide Web Consortium). Η HTML 5 προσθέτει νέα χαρακτηριστικά, δίνοντας ιδιαίτερο βάρος στη σημασιολογία των ετικετών, καθώς και νέες δυνατότητες που περιορίζουν την ανάγκη χρήσης πρόσθετων (plug-ins) στα προγράμματα πλοήγησης.



Εικόνα 3 : Το λογότυπο της HTML5

3.1.2 Χαρακτηριστικά της HTML5

Η HTML 5 παρέχει πολλά νέα χαρακτηριστικά, κάποια από τα οποία παρουσιάζονται συνοπτικά παρακάτω (Τσιάτσος Θ, 2015):

- Προσθήκη νέων ετικετών (για παράδειγμα `<header>`, `<footer>`, `<nav>` κλπ.) που κάνουν τη δημιουργία και τη διαχείριση των ιστοσελίδων ακόμη πιο εύκολες.
- Καταργείται η χρήση κάποιων μη αποδεκτών στοιχείων της HTML 4.01 (για παράδειγμα η ετικέτα `<center>`), τα οποία μπορούν να αντικατασταθούν από CSS.
- Δυνατότητα σχεδιασμού γραφικών με χρήση JavaScript και της νέας ετικέτας `<canvas>`.
- Αναπαραγωγή βίντεο και ήχου χωρίς να χρειάζεται η εγκατάσταση πρόσθετων (plug-ins) με τις νέες ετικέτες `<video>` και `<audio>`.
- Προσθήκη νέων στοιχείων στις HTML φόρμες, όπως `<calendar>`, `<date>` , `<time>` κλπ.
- Η σύνταξη της HTML5 δε βασίζεται πλέον στην SGML. Έχει σχεδιαστεί να είναι συμβατή προς τα πίσω με τις παλαιότερες εκδόσεις της HTML.
- Ορίζονται scripting application programming interfaces (APIs), τα οποία μπορούν να χρησιμοποιηθούν με τη JavaScript. Κάποια από τα νέα APIs

περιλαμβάνουν Offline Web Applications, Drag-and-drop, Web Storage, MIME type, Web Messaging κλπ.

- Διαχείριση λαθών: η HTML5 είναι σχεδιασμένη έτσι ώστε οι παλαιότεροι περιηγητές να μπορούν πλέον να αγνοούν με ασφάλεια νέες HTML5 δομές. Επίσης περιλαμβάνονται λεπτομερείς κανόνες με στόχο διαφορετικά συμβατοί περιηγητές να παράγουν το ίδιο αποτέλεσμα στην περίπτωση που η σύνταξη είναι λάθος.

3.1.3 Νέα στοιχεία – ετικέτες της HTML5

Έχουν προστεθεί πολλές νέες ετικέτες, κάποιες από τις οποίες είναι οι παρακάτω (w3schools.com – HTML5 New Elements 2021):

- **Στοιχεία για τη δομή του εγγράφου**

<section>	Ορίζει μια ενότητα, ένα τμήμα του εγγράφου.
<article>	Ορίζει ένα αυτόνομο τμήμα περιεχομένου, όπως το κείμενο μιας ανάρτησης σε ιστολόγιο.
<header>	Ορίζει εισαγωγικές πληροφορίες και στοιχεία πλοήγησης.
<footer>	Ορίζει πληροφορίες στο τέλος του εγγράφου.
<nav>	Ορίζει συνδέσμους πλοήγησης στο έγγραφο.
<aside>	Ορίζει περιεχόμενο ανεξάρτητο από το περιεχόμενο της ιστοσελίδας.
<main>	Ορίζει το κυρίως περιεχόμενο του εγγράφου.
<dialog>	Ορίζει ένα πλαίσιο ή παράθυρο διαλόγου.

- **Γραφικά**

<canvas>	για σχεδιασμό γραφικών με JavaScript
<svg>	για σχεδιασμό γραφικών με SVG

- **Στοιχεία για φόρμες**

<datalist>	Ορίζει προκαθορισμένες επιλογές για έλεγχο της
------------	--

εισόδου του χρήστη.

<output> Ορίζει το αποτέλεσμα ενός υπολογισμού.

Ήχος & Βίντεο

<audio> Ορίζει ως περιεχόμενο ήχο ή μουσική.

<embed> Ορίζει «containers» για συμπερίληψη εξωτερικών εφαρμογών, όπως πρόσθετα.

<source> Ορίζει πηγές για ήχο και βίντεο.

<video> Ορίζει ως περιεχόμενο βίντεο ή ταινία.

3.2 Bootstrap

Σύμφωνα με τη Wikipedia (wikipedia.org - Bootstrap 2021) «Το Bootstrap είναι μια συλλογή εργαλείων ανοιχτού κώδικα (ελεύθερο λογισμικό) για τη δημιουργία ιστοσελίδων και διαδικτυακών εφαρμογών. Περιέχει HTML και CSS για τις μορφές τυπογραφίας, κουμπιά πλοήγησης και άλλων στοιχείων του περιβάλλοντος, καθώς και προαιρετικές επεκτάσεις JavaScript».



Εικόνα 4 : Το λογότυπο του Bootstrap

Το Bootstrap (Εικόνα 4) είναι αυτή τη στιγμή ένα από τα πιο δημοφιλή front-end πλαίσια (framework), που χρησιμοποιούνται στην κατασκευή ιστοσελίδων. Είναι μια τεράστια συλλογή από εύχρηστα, επαναχρησιμοποιήσιμα κομμάτια κώδικα "γραμμένα" σε HTML, CSS και JavaScript. Είναι επίσης ένα πλαίσιο ανάπτυξης που επιτρέπει στους προγραμματιστές και τους σχεδιαστές να δημιουργούν γρήγορα και αποτελεσματικά δυναμικούς ιστότοπους.

3.2.1 Ιστορικά στοιχεία

Το Bootstrap αναπτύχθηκε από τον Mark Otto και τον Jacob Thornton στο Twitter ως ένα εσωτερικό εργαλείο ανάπτυξης, στα μέσα του 2010, και ήταν γνωστό ως Twitter Blueprint. Τον Αύγουστο του 2011 κυκλοφόρησε ως λογισμικό ανοιχτού κώδικα και τον Φεβρουάριο του 2012, ήταν το πιο δημοφιλές έργο ανάπτυξης στο GitHub. Από τότε έχουμε πάνω από 20 νέες κυκλοφορίες, συμπεριλαμβανομένων 3 βασικών εκδόσεων, την έκδοση 2, 3 και 4. Με το Bootstrap 2 προστέθηκε η λειτουργία του responsive design, δηλαδή η δυνατότητα η διάταξη των ιστοσελίδων να προσαρμόζεται δυναμικά, λαμβάνοντας υπόψη τα χαρακτηριστικά της συσκευής που χρησιμοποιείται (PC, tablet, κινητό τηλέφωνο). Τέλος, με την έκδοση 4, μερικές από τις σημαντικές αλλαγές που έγιναν ήταν: η μετάβαση στο SASS φύλλο στιλ και στο flexbox CSS πρότυπο διάταξης (getbootstrap.com – History 2021).

3.2.2 Χαρακτηριστικά

Το Bootstrap διατίθεται δωρεάν για όλους. Σύμφωνα με τη Sitesbay (sitesbay.com - Features of Bootstrap 2021) μερικά από τα βασικά χαρακτηριστικά του είναι τα ακόλουθα:

- Εύκολο στη χρήση. Οποιοσδήποτε έχει βασικές γνώσεις HTML και CSS, μπορεί να το χρησιμοποιήσει.
- Προσέγγιση Mobile-first. Από την έκδοση 3, η προσέγγιση αυτή είναι μέρος του πυρήνα του Bootstrap.
- Απλή ενσωμάτωση. Το Bootstrap μπορεί να ενσωματωθεί εύκολα μαζί με άλλες πλατφόρμες και πλαίσια, σε υπάρχοντες ιστότοπους.
- Προσαρμοστικότητα. Το Bootstrap μπορεί να προσαρμοστεί σύμφωνα με τα σχέδια του έργου σας.
- Συμβατότητα. Το Bootstrap είναι συμβατό με όλα τα σύγχρονα προγράμματα περιήγησης (Chrome, Firefox, Internet Explorer, Safari και Opera).
- Responsive διάταξη. Το Bootstrap βασίζεται σε πλέγματα 12 στηλών, διατάξεις και στοιχεία. Είτε χρειάζεστε ένα responsive πλέγμα είτε ένα σταθερό πλέγμα, είναι θέμα μόνο μερικών αλλαγών.

- Προσθήκες JavaScript. Στοιχεία όπως το αναπτυσσόμενο μενού για παράδειγμα, γίνονται διαδραστικά με τις πολυάριθμες προσθήκες JavaScript που περιλαμβάνονται στο πακέτο Bootstrap.
- Εκτενής λίστα επαναχρησιμοποιήσιμων συστατικών. Αναπτυσσόμενα μενού, ομάδες κουμπιών, γραμμές πλοήγησης, breadcrumbs, ετικέτες, μηνύματα ειδοποιήσεων, μπάρες προόδου και πολλά άλλα.
- Βασικό στυλ για τα περισσότερα στοιχεία HTML. Τα στοιχεία HTML για τα οποία παρέχονται στυλ είναι. επικεφαλίδες, παράγραφοι, πίνακες, φόρμες, κουμπιά, εικόνες, εικονίδια κλπ.
- Καλή τεκμηρίωση. Το Bootstrap παρέχει μεγάλη τεκμηρίωση με επίδειξη παραδειγμάτων που διευκολύνουν ακόμη και τους αρχάριους προγραμματιστές.

3.2.3 Δομή και λειτουργία

Για να καταλάβουμε πώς λειτουργεί το Bootstrap, θα πρέπει να κατανοήσουμε την αρθρωτή δομή που χρησιμοποιεί καθώς και τα πιο σημαντικά του στοιχεία, που είναι τα αρχεία CSS και JavaScript.

Σύμφωνα με τη Wikipedia (wikipedia.org - Bootstrap 2021) «Το Bootstrap είναι σπονδυλωτό και αποτελείται ουσιαστικά από μια σειρά στυλ (stylsheets) που εφαρμόζουν τα διάφορα συστατικά του πακέτου εργαλείων. Ένα στυλ που ονομάζεται bootstrap.less περιλαμβάνει τα συστατικά stylesheets. Οι προγραμματιστές μπορούν να προσαρμόσουν το αρχείο Bootstrap, επιλέγοντας τα στοιχεία που θέλουν να χρησιμοποιήσουν στο έργο τους».

Αρχεία CSS

Αποτελούνται από καθολικές ρυθμίσεις και χρησιμοποιούνται για τον καθορισμό, χρησιμοποιώντας πολλές κλάσεις, και την εμφάνιση των στοιχείων HTML, όπως λίστες, κείμενα, πίνακες, στοιχεία φόρμας και εικόνες. «Προσαρμογές είναι δυνατές σε περιορισμένη έκταση μέσω ενός κεντρικού στυλ διαμόρφωσης. Η χρήση γλώσσας στυλ επιτρέπει τη χρήση για μεταβλητές, λειτουργίες και φορείς (operators), ένθετους επιλογείς, γνωστά και ως μείγματα mixin.» (wikipedia.org - Bootstrap 2021).

Στα αρχεία CSS, περιέχονται επίσης αρκετές κλάσεις, οι οποίες επιτρέπουν τη δημιουργία ταχύτερων διατάξεων και μορφών απόκρισης. Τέλος το Bootstrap επιτρέπει την προσαρμογή και την επέκταση των ήδη υπαρχόντων κλάσεων για την δημιουργία νέων προσαρμοσμένων κλάσεων. Οι κλάσεις αυτές έχουν να κάνουν με στοιχεία HTML, τα οποία χρησιμοποιούνται πολύ συχνά στους σημερινούς ιστότοπους. Αυτά τα στοιχεία περιλαμβάνουν επέκταση στοιχείων, κουμπιών και προηγμένες λειτουργίες όπως ομαδοποίηση, σελιδοποίηση και μενού πλοήγησης, μπάρες προόδου ή μηνύματα.

Αρχεία JavaScript

Το Bootstrap έρχεται με πολλά συστατικά JavaScript με τη μορφή πρόσθετων (plugins) από την βιβλιοθήκη jQuery. Παρέχεται πρόσθετη διεπαφή χρήστη με στοιχεία όπως παράθυρα διαλόγου, επεξηγήσεις, και καρουσέλ. Μπορούν επίσης να επεκτείνουν τη λειτουργικότητα ορισμένων υφιστάμενων στοιχείων, όπως για παράδειγμα τα πεδία εισαγωγής στις φόρμες. Τα προαναφερθέντα στοιχεία δεν μπορούν μόνο να δημιουργηθούν αλλά και να προσαρμοστούν και να επεκταθούν, χρησιμοποιώντας κώδικα JavaScript, αλλά και πολλά χαρακτηριστικά που βασίζονται σε HTML. Αυτή είναι μια καλή διευκόλυνση, ειδικά για τους λιγότερο έμπειρους προγραμματιστές και ο αριθμός των αλλαγών που γίνονται με αυτόν τον τρόπο είναι πολύ μεγάλος.

3.2.4 Χρήση

Όπως αναφέραμε και πιο πάνω, το Bootstrap είναι ένα CSS-JavaScript πλαίσιο (framework) που διευκολύνει τους προγραμματιστές στην ανάπτυξη της διεπαφής χρήστη με τη χρήση προκατασκευασμένων responsive κλάσεων και άλλων βοηθητικών βιβλιοθηκών. Σύμφωνα με τη Bootstrapdash (bootstrapdash.com - How to Use Bootstrap With HTML 2021), για να συμπεριλάβουμε το Bootstrap σε ένα HTML έγγραφο, μπορούμε να χρησιμοποιήσουμε μία από τις τρεις παρακάτω μεθόδους:

1. Χρήση Bootstrap CDN

CSS

Απλώς αντιγράφουμε τον παρακάτω σύνδεσμο φύλλου στυλ στο <head> του αρχείου HTML.

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhc
Wr7x9JvoRxT2MZw1T" crossorigin="anonymous">
```

JavaScript

Οι λειτουργίες JavaScript ορισμένων στοιχείων όπως αναπτυσσόμενα μενού, καρτέλες κλπ. εξαρτώνται από τις βιβλιοθήκες jquery και popper. Έτσι, μπορούμε να συμπεριλάβουμε τα jquery και popper.js με την ακόλουθη σειρά, ακριβώς πριν φορτωθεί το αρχείο JavaScript του bootstrap (bootstrap.min.js), για σωστή λειτουργία.

```
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous">
</script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJK6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrn
Qq4sF86dIHNDz0W1" crossorigin="anonymous">
</script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUUA YoIly6OrQ6VrjIEaFf/nJGzIxFDsf4
x0xIM+B07jRM" crossorigin="anonymous">
</script>
```

2. Λήψη των αρχείων τοπικά

Αντί να χρησιμοποιήσουμε CDN, μπορούμε να κατεβάσουμε τα αρχεία τοπικά στο φάκελο του έργου μας από τη διεύθυνση <https://getbootstrap.com/docs/4.3/getting-started/download/>

Μόλις κατεβάσουμε τα αρχεία, μπορούμε να συμπεριλάβουμε το αρχείο bootstrap.min.css στο <head> και το bootstrap.min.js στο <body>. Ακόμα κι αν χρησιμοποιείτε η δεύτερη μέθοδος (λήψη αρχείων bootstrap τοπικά), πρέπει να

συμπεριλάβουμε τα `jquery.min.js` και `popper.min.js` πριν από τη φόρτωση του `bootstrap.min.js`

3. Χρήση διαχειριστών πακέτων

Το Bootstrap μπορεί εύκολα να χρησιμοποιηθεί σε οποιοδήποτε έργο χρησιμοποιώντας διαχειριστές πακέτων όπως «`npm`», «`yarn`» κλπ. Δεδομένου ότι το `npm` είναι ο πιο δημοφιλής διαχειριστής πακέτων που χρησιμοποιείται από τους προγραμματιστές, προχωράμε με την εντολή `npm` για την εγκατάσταση του bootstrap. Πληκτρολογούμε την παρακάτω εντολή μέσα στο φάκελο του έργου μας.

```
npm install bootstrap
```

Αυτή η εντολή θα κατεβάσει ένα τοπικό αντίγραφο αρχείων bootstrap μέσα στο φάκελο του έργου μας. Στη συνέχεια, μπορούμε να συμπεριλάβουμε το αρχείο `bootstrap.min.css` στο `<head>` και το `bootstrap.min.js` στο `<body>`.

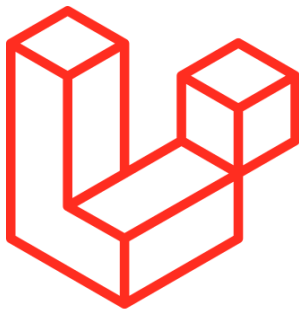
Ένα παράδειγμα αρχείου HTML μετά τη συμπερίληψη bootstrap CSS και JavaScript θα μοιάζει με το παρακάτω. Στο παράδειγμα αυτό έχουμε χρησιμοποιήσει τη μέθοδο CDN. Μπορούμε να τροποποιήσουμε την παράμετρο `href` για φύλλα στυλ και `src` για JavaScript με την κατάλληλη διαδρομή, εάν επιλέξουμε οποιαδήποτε άλλη μέθοδο.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
    href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
    integrity="sha384-gOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQU
    OhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
  <title>Bootstrap example</title>
</head>
<body>
```

```
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo
" crossorigin="anonymous"> </script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSjQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZ
wrnQq4sF86dIHNDz0W1" crossorigin="anonymous"> </script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzI
xFDs4x0xIM+B07jRM" crossorigin="anonymous"> </script>
</body>
</html>
```

3.3 Laravel

Το Laravel (Εικόνα 5) είναι ένα δωρεάν, ανοικτού κώδικα PHP πλαίσιο (framework) που δημιουργήθηκε από τον Taylor Otwell το 2011, με κύριο στόχο την ανάπτυξη διαδικτυακών εφαρμογών που χρησιμοποιούν το αρχιτεκτονικό μοτίβο MVC (wikipedia.org - Laravel 2021).



Εικόνα 5 : Το λογότυπο του Laravel

3.3.1 Χαρακτηριστικά – πλεονεκτήματα του Laravel

Σύμφωνα με τη Tutorialandexample (tutorialandexample.com - Advantages of Laravel 2019), το Laravel προσφέρει στο σχεδιασμό και στην ανάπτυξη μιας διαδικτυακής εφαρμογής τα παρακάτω πλεονεκτήματα:

1. Ισχυρό σύστημα ελέγχου ταυτότητας χρήστη (Authentication system)

Το PHP πλαίσιο του Laravel αναπτύχθηκε με σκοπό να βοηθήσει τους προγραμματιστές. Επιτρέπει στους προγραμματιστές να δημιουργήσουν διάφορες λειτουργίες που χρησιμοποιούνται σε εφαρμογές ιστού, πολύ πιο γρήγορα και αποτελεσματικά. Μία από αυτές τις λειτουργίες είναι η ανάπτυξη του συστήματος ελέγχου ταυτότητας χρήστη, η οποία περιέχει διάφορες τεκμηριωμένες επιλογές για την προσαρμογή της συμπεριφοράς των υπηρεσιών ελέγχου ταυτότητας.

2. MVC Αρχιτεκτονική

Η αρχιτεκτονική Model-View-Controller (η οποία θα εξηγηθεί παρακάτω) μας παρέχει ενσωματωμένη λειτουργικότητα με την οποία οι προγραμματιστές μπορούν να αξιοποιήσουν με τον καλύτερο τρόπο την ανάπτυξη ιστοσελίδων.

3. Μετεγκατάσταση βάσης δεδομένων (Database Migration)

Η μετεγκατάσταση βάσης δεδομένων επιτρέπει στους προγραμματιστές να αναιρέσουν γρήγορα τις αλλαγές που έγιναν στη βάση δεδομένων. Επίσης διασφαλίζει ότι οι βάσεις δεδομένων δημιουργούνται και συντηρούνται σωστά χωρίς να χρειάζεται να δημιουργηθούν εκ νέου.

4. Ανοιχτού κώδικα ισχυρή κοινότητα

Το Laravel διαθέτει μια μεγάλη κοινότητα προγραμματιστών και εταιρειών ανάπτυξης που παρέχουν πάντα και συνεχώς υποστήριξη για να κάνουν το Laravel πιο ευέλικτο και κλιμακούμενο.

5. Βελτιωμένη απόδοση

Το Laravel έρχεται με τα διάφορα εργαλεία που μπορούν να βοηθήσουν τους προγραμματιστές να βελτιώσουν και να ενισχύσουν την απόδοση της διαδικτυακής τους εφαρμογής, τόσο κατά τον σχεδιασμό, όσο και κατά την ανάπτυξη της.

6. Αντικειμενοστραφείς βιβλιοθήκες

Οι βιβλιοθήκες είναι προεγκατεστημένες. Μια από αυτές είναι για παράδειγμα η βιβλιοθήκη ελέγχου ταυτότητας χρήστη. Αυτές οι βιβλιοθήκες είναι εύχρηστες και

εύκολες στην χρήση τους, επιτρέποντας στους προγραμματιστές να αναπτύξουν μια αποκριτική (responsive) και αρθρωτή (modular) εφαρμογή Ιστού.

7. Blade πρότυπα

Το Laravel επιτρέπει στους προγραμματιστές να χρησιμοποιούν προκαθορισμένα πρότυπα όσον αφορά την εμφάνιση των ιστοσελίδων της εφαρμογής. Αυτά τα πρότυπα είναι δυναμικά, γεγονός που επιτρέπει στους προγραμματιστές να γράφουν κώδικα αποτελεσματικά και γρήγορα χωρίς να επηρεάζεται η απόδοση της εφαρμογής.

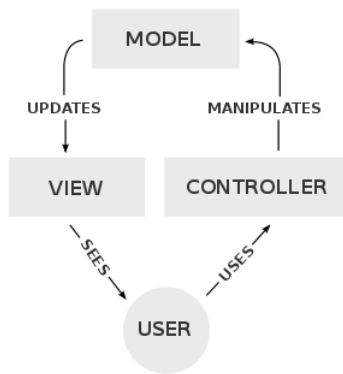
8. Ασφάλεια

Το Laravel παρέχει ασφάλεια υψηλής ποιότητας. Είναι ένα ασφαλές πλαίσιο που δεν επιτρέπει καμία δραστηριότητα ή απειλή ασφάλειας να εισέλθει σε μια εφαρμογή, πράγμα που σημαίνει ότι ο κωδικός της εφαρμογής μας είναι ασφαλής.

3.3.2 Το μοντέλο MVC

Το MVC (Model-Views-Controller) είναι ένα μοτίβο σχεδίασης στην ανάπτυξη λογισμικού που χωρίζει τη λογική του προγράμματος σε 3 διασυνδεδεμένα στοιχεία. τα Models, τα Views και τους Controllers. Η εσωτερική δομή του Laravel ακολουθεί αυτό το μοτίβο σχεδίασης.

Σύμφωνα με τη Wikipedia (wikipedia.org Model-view-controller 2020) «Το κύριο μέρος του μοντέλου είναι το αντικείμενο Model το οποίο διαχειρίζεται την ανάκτηση/αποθήκευση των δεδομένων στο σύστημα. Το αντικείμενο View χρησιμοποιείται μόνο για να παρουσιάζεται η πληροφορία στον χρήστη (π.χ. με γραφικό τρόπο). Το τρίτο μέρος είναι ο Controller ο οποίος δέχεται την είσοδο και στέλνει εντολές στα αντικείμενα Model και View». Επομένως, σύμφωνα και με την παρακάτω εικόνα (Εικόνα 6) στο μοντέλο MVC έχουμε:



Εικόνα 6 : Τα μέρη του μοντέλου MVC

Models

Το αντικείμενο model παρέχει όλες τις μεθόδους και τα χαρακτηριστικά που είναι απαραίτητα ώστε να έχουμε την αλληλεπίδραση με την βάση δεδομένων της εφαρμογής. Μπορεί να απαντήσει σε αιτήματα για ανάκτηση δεδομένων ή να απαντήσει σε οδηγίες για να αλλάξει η κατάσταση των δεδομένων που είναι αποθηκευμένα στην βάση δεδομένων της εφαρμογής. Επίσης το model ενημερώνει τα views και τους controllers όταν υπάρχει αλλαγή στα δεδομένα της εφαρμογής. Εν ολίγοις, είναι η αναπαράσταση και η διαχείριση των δεδομένων της εφαρμογής.

Views

Το αντικείμενο view παρέχει με αποτελεσματικό τρόπο το στοιχείο της διεπαφής χρήστη της εφαρμογής. Αναπαριστά με γραφικό τρόπο (με μια φόρμα ή έναν πίνακα για παράδειγμα) την πληροφορία που περιέχει το model.

Controllers

Το αντικείμενο controller λαμβάνει την είσοδο του χρήστη και πραγματοποιεί κλήσεις σε αντικείμενα model και view για την εκτέλεση των κατάλληλων ενεργειών. Έτσι ο controller μπορεί να στείλει εντολές στο model και να ενημερώνει την κατάσταση του ή μπορεί να στείλει εντολές ώστε να γίνει η αντίστοιχη αναπαράσταση των δεδομένων του model μέσω του View.

Ας δούμε ένα απλό παράδειγμα. Εάν μεταβούμε σε έναν ιστότοπο ηλεκτρονικού εμπορίου, οι διαφορετικές σελίδες που βλέπουμε παρέχονται από το view. Όταν κάνουμε κλικ σε ένα συγκεκριμένο προϊόν για να δούμε περισσότερα, ο controller επεξεργάζεται την ενέργεια του χρήστη. Αυτό μπορεί να περιλαμβάνει τη λήψη

δεδομένων από μια βάση δεδομένων χρησιμοποιώντας το model. Τα δεδομένα στη συνέχεια ομαδοποιούνται ,ταξινομούνται και εμφανίζονται στον χρήστη από το view.

3.3.3 Routing

Σύμφωνα με τη Laravel (laravel.com - Basic Routing 2021), το Laravel παρέχει μια πολύ απλή μέθοδο καθορισμού διαδρομών (routes), χωρίς να απαιτείται η χρήση περίπλοκων αρχείων διαμόρφωσης δρομολόγησης. Οι περισσότερες και οι πιο βασικές διαδρομές δέχονται ένα URL και το όνομα μιας συνάρτησης, η οποία έχει ορισθεί μέσα σε κάποιον ελεγκτή (Controller). Όλα τα routes της εφαρμογής ορίζονται στο αρχείο routes/web.php το οποίο φορτώνεται από τη κλάση App\Providers\RouteServiceProvider.

Επίσης μπορούμε να καθορίσουμε ένα όνομα για μία διαδρομή, τοποθετώντας το στο τέλος του ορισμού της διαδρομής. Το όνομά αυτό θα πρέπει να είναι μοναδικό, δεν μπορούν δηλαδή δύο διαφορετικές διαδρομές να έχουν το ίδιο όνομα (laravel.com - Named Routes 2021).

Για παράδειγμα, στην παρακάτω εικόνα (Εικόνα 7), βλέπουμε τη διαδρομή με όνομα dashboard που εκτελεί τη μέθοδο get από τη συνάρτηση index του Controller DashboardController.

```
Route::get('/dashboard', 'App\Http\Controllers\DashboardController@index')->name('dashboard');
```

Εικόνα 7: Διαδρομή που εκτελεί τη μέθοδο ενός Controller

Εκτός από τη μέθοδο get, το Laravel υποστηρίζει (Εικόνα 8) τις παρακάτω HTTP μεθόδους (laravel.com - Available Router Methods 2021):

```
Route::get($uri, $callback);  
Route::post($uri, $callback);  
Route::put($uri, $callback);  
Route::patch($uri, $callback);  
Route::delete($uri, $callback);  
Route::options($uri, $callback);
```

Εικόνα 8: HTTP μέθοδοι που υποστηρίζει το Laravel

Υπάρχουν φορές που χρειάζεται να απομονωθεί ένα τμήμα του URI μέσα στο route. Αυτό μπορεί να γίνει καθορίζοντας παραμέτρους για τη διαδρομή (laravel.com Route Parameters 2021). Για παράδειγμα, στην παρακάτω εικόνα (Εικόνα 9), βλέπουμε τη διαδρομή με όνομα `studentslessons.show` που εκτελεί τη μέθοδο `get` από τη συνάρτηση `show` του Controller `StudentsLessonsController`. Εδώ έχει χρησιμοποιηθεί η παράμετρος `{lessonId}`, ώστε να απομονωθεί το ID μιας ενότητας .

```
Route::get('/dashboard/students/courses/lessons/{lessonId}'  
          ,[StudentsLessonsController::class, 'show']->name('studentslessons.show');
```

Εικόνα 9: Διαδρομή με παράμετρο που εκτελεί τη μέθοδο ενός Controller

Τέλος, οι ομάδες διαδρομών (Route Groups) μας επιτρέπουν να μοιραζόμαστε χαρακτηριστικά διαδρομής, όπως για παράδειγμα το `middleware`, σε μεγάλο αριθμό διαδρομών χωρίς να χρειάζεται να ορίσουμε αυτά τα χαρακτηριστικά σε κάθε μεμονωμένη διαδρομή. Για να εκχωρήσουμε το `middleware` σε όλες τις διαδρομές μιας ομάδας, μπορούμε να χρησιμοποιήσουμε τη μέθοδο του `middleware` πριν ορίσουμε την ομάδα (laravel.com - Route Groups 2021).

Για παράδειγμα, στην παρακάτω εικόνα (Εικόνα 10) βλέπουμε μια ομάδα διαδρομών, στις οποίες εκτελούνται οι μέθοδοι `get` και `put`. Επίσης, βλέπουμε ότι οι διαδρομές αυτές θα εκτελεστούν μόνο για πιστοποιημένους (authenticated) χρήστες που έχουν στην εφαρμογή ρόλο διαχειριστή (admin).

```
Route::group(['middleware' => ['auth', 'role:admin']], function() {  
    Route::get('/dashboard/usersmanagement', 'App\Http\Controllers\Admin\UsersManagementController@index')  
        ->name('dashboard.usersmanagement');  
    Route::get('/dashboard/users/{user}/editpassword', [UsersManagementController::class, 'editpassword'])  
        ->name('users.editpassword');  
    Route::put('/dashboard/{user}', [UsersManagementController::class, 'updatepassword'])  
        ->name('users.updatepassword');  
    Route::resource('/dashboard/users',UsersManagementController::class);  
    Route::put('usersmanagement', [\App\Http\Controllers\Admin\UsersManagementController::class, 'store'])  
        ->name('usersmanagement.store');  
    Route::get('/dashboard/usersstats', [AdminStatsController::class, 'index']->name('usersstats.adminstats');  
});
```

Εικόνα 10: Ομάδα διαδρομών (Route Group)

3.3.4 Middleware

Σύμφωνα με τη Laravel (laravel.com - Middleware - Introduction 2021), «το Middleware παρέχει έναν βολικό μηχανισμό για τον έλεγχο και το φιλτράρισμα αιτημάτων HTTP που εισέρχονται στην εφαρμογή μας». Για παράδειγμα, το Laravel μέσω ενός middleware ελέγχει αν ένας χρήστης είναι πιστοποιημένος (authenticated) στην εφαρμογή ή όχι. Αν είναι του επιτρέπει την πρόσβαση σε συγκεκριμένα τμήματα της εφαρμογής. Διαφορετικά τον ανακατευθύνει στη φόρμα σύνδεσης (login) ή εγγραφής (register) αν δεν είναι ήδη εγγεγραμμένος.

Επιπρόσθετα, το middleware μπορεί να χρησιμοποιηθεί για την εκτέλεση ποικίλων εργασιών εκτός από τον έλεγχο ταυτότητας. Για παράδειγμα, ένα middleware καταγραφής μπορεί να καταγράψει όλα τα εισερχόμενα αιτήματα σε μια εφαρμογή. Μπορεί κανείς να βρει όλα τα middleware στο φάκελο App/Http/Middleware μεταξύ των οποίων υπάρχουν middleware για συντήρηση (maintenance), αυθεντικοποίηση (authentication), προστασία (CSRF protection) κλπ.

3.3.5 Controllers

Αντί να ορίσουμε όλη τη λογική χειρισμού των αιτημάτων της εφαρμογής μέσα από συναρτήσεις στα αρχεία διαδρομής (routes), μπορούμε να οργανώσουμε αυτήν τη συμπεριφορά χρησιμοποιώντας κλάσεις ελεγκτή (controller). Οι ελεγκτές (controllers) μπορούν να ομαδοποιήσουν τη σχετική λογική διαχείρισης αιτημάτων σε μια κλάση. Από προεπιλογή, οι ελεγκτές αποθηκεύονται στον κατάλογο app/Http/Controllers (laravel.com - Controllers - Introduction 2021).

```

<?php

namespace App\Http\Controllers;

use App\Http\Controllers\Controller;
use App\Models\User;

class UserController extends Controller
{
    /**
     * Show the profile for a given user.
     *
     * @param int $id
     * @return \Illuminate\View\View
     */
    public function show($id)
    {
        return view('user.profile', [
            'user' => User::findOrFail($id)
        ]);
    }
}

```

Εικόνα 11: Παράδειγμα ελεγκτή (Controller)

Ας ρίξουμε μια ματιά σε ένα παράδειγμα ελεγκτή (laravel.com - Controllers - Basic Controllers 2021). Σύμφωνα με την παραπάνω εικόνα (Εικόνα 11), έχουμε ορίσει την κλάση `UserController` η οποία επεκτείνει (extends) τη βασική κατηγορία ελεγκτή που περιλαμβάνεται στο Laravel στον κατάλογο `App\Http\Controllers\Controller`. Με την κλάση `UserController` μπορούμε να χειριστούμε όλες τις εισερχόμενες αιτήσεις (requests) που σχετίζονται με τους χρήστες της εφαρμογής. Στην κλάση αυτή μπορούμε να συμπεριλάβουμε συναρτήσεις (μεθόδους) που έχουν να κάνουν με την εμφάνιση, την δημιουργία, την ενημέρωση και την διαγραφή χρηστών. Στο συγκεκριμένο παράδειγμα χρησιμοποιούμε την συνάρτηση `show`. Όπως φαίνεται και στην εικόνα, η συνάρτηση `show` δέχεται ως είσοδο το `id` ενός συγκεκριμένου χρήστη, βρίσκει με την συνάρτηση `findOrFail` την εγγραφή αυτού του χρήστη στην βάση δεδομένων και επιστρέφει τη σελίδα του προφίλ του. Επίσης, μπορούμε να ορίσουμε μια διαδρομή (route) για αυτήν την συνάρτηση `show` του ελεγκτή `UserController`.

```
use App\Http\Controllers\UserController;

Route::get('/user/{id}', [UserController::class, 'show']);
```

Εικόνα 12: Παράδειγμα διαδρομής (route)

Στην εικόνα 12, βλέπουμε ότι όταν ένα εισερχόμενο αίτημα ταιριάζει με το καθορισμένο URI, θα κληθεί η μέθοδος (συνάρτηση) show του ελεγκτή App\Http\Controllers\UserController και η παράμετρος ({id}) θα μεταφερθεί στη μέθοδο.

Τέλος, σύμφωνα με τη Laravel (laravel.com - Controllers - Basic Controllers 2021), « οι ελεγκτές δεν απαιτείται να επεκτείνουν την βασική κλάση Controller. Ωστόσο, αν δεν γίνει αυτό, δεν θα έχουμε πρόσβαση σε λειτουργίες, όπως το middleware και χρήσιμες μεθόδους εξουσιοδότησης (authorize methods) ».

3.3.6 Views

Οι διαδρομές (routes) και οι ελεγκτές (controllers) επιστρέφουν ολόκληρες συμβολοσειρές HTML εγγράφων. Αυτό όμως, δεν είναι και πολύ πρακτικό. Το Laravel μας παρέχει έναν βολικό τρόπο για να τοποθετούμε την εμφάνιση της εφαρμογής (όλο το HTML μέρος) σε ξεχωριστά αρχεία. Ο τρόπος αυτός είναι οι προβολές (Views). Οι προβολές περιέχουν τον κώδικα html που απαιτείται από την εφαρμογή μας και είναι μια μέθοδος στο Laravel που διαχωρίζει τη λογική του ελεγκτή από τη λογική της παρουσίασης. Οι προβολές βρίσκονται στο φάκελο resources στη διαδρομή resources/views. (laravel.com - Views - Introduction 2021).

Ας δούμε ένα απλό παράδειγμα (laravel.com Views - Introduction 2021).

Για να δημιουργήσουμε μία προβολή (Εικόνα 13) πρέπει να τοποθετήσουμε ένα αρχείο με επέκταση .blade.php στον κατάλογο resources/views της εφαρμογής μας. Η επέκταση .blade.php ενημερώνει το Laravel ότι το αρχείο περιέχει πρότυπο Blade.

```
<!-- View stored in resources/views/greeting.blade.php -->

<html>
  <body>
    <h1>Hello, {{ $name }}</h1>
  </body>
</html>
```

Εικόνα 13: Παράδειγμα προβολής (view)

Αφού δημιουργήσουμε την προβολή, μπορούμε να την καλέσουμε από μία διαδρομή ή έναν ελεγκτή χρησιμοποιώντας τη καθολική μέθοδο `view()`.

```
Route::get('/', function () {
    return view('greeting', ['name' => 'James']);
});
```

Εικόνα 14: Παράδειγμα διαδρομής με κλήση προβολής

Όπως φαίνεται στον παραπάνω κώδικα (Εικόνα 14), η μέθοδος `view()` περιέχει δύο ορίσματα. Το πρώτο όρισμα είναι το όνομα του αρχείου της προβολής και το δεύτερο όρισμα είναι ένας πίνακας που μεταβιβάζεται σε αυτό το αρχείο. Στον πίνακα του παραδείγματος περνάμε τη μεταβλητή `name` με τιμή `'James'`. Το αποτέλεσμα είναι να εμφανιστεί στην οθόνη `'Hello James'`.

3.3.7 Blade Templates

Το Blade είναι ένας ισχυρός μηχανισμός προτύπου (template) που χρησιμοποιεί το Laravel. Σε ένα Blade Template μπορεί κανείς να συμπεριλάβει πέρα από καθαρή PHP και κώδικα από άλλες γλώσσες προγραμματισμού, όπως HTML, CSS, JavaScript αλλά και τεχνολογίες όπως Bootstrap και jQuery. Ο μηχανισμός αυτός είναι εύκολος στη χρήση και καθιστά το γράψιμο του κώδικα πολύ απλή υπόθεση. Τα αρχεία προτύπου Blade χρησιμοποιούν την επέκταση `.blade.php` και συνήθως αποθηκεύονται στον κατάλογο `resources/views`. (laravel.com - Blade Templates - Introduction 2021).

Χρησιμοποιώντας Blade Templates, μπορούμε να συμπεριλάβουμε σε μια Blade προβολή δεδομένα που μεταβιβάζονται από μια διαδρομή ή ένα ελεγκτή. Μπορούμε, για παράδειγμα (laravel.com - Blade Templates - Displaying Data 2021) να εμφανίσουμε δεδομένα χρησιμοποιώντας μια μεταβλητή μέσα σε διπλά άγκιστρα «{{ }}» (Εικόνα 15).

```
Hello, {{ $name }}.
```

Εικόνα 15: Παράδειγμα χρήσης μεταβλητής σε Blade αρχείο

Το Blade, εκτός από την εμφάνιση δεδομένων παρέχει επίσης βολικές συντομεύσεις για κοινές δομές ελέγχου PHP, όπως δηλώσεις «if» και βρόχους. Αυτές οι συντομεύσεις έχουν παραπλήσια σύνταξη με τις δομές ελέγχου της PHP, βοηθώντας τους προγραμματιστές να γράφουν κώδικα πιο εύκολα και γρήγορα. Στις παρακάτω εικόνες (Εικόνες 16 και 17) παραθέτουμε κάποια παραδείγματα (laravel.com - Blade Templates - If Statements 2021) και (laravel.com - Blade Templates - Loops 2021).

```
@if (count($records) === 1)
    I have one record!
@endif
@elseif (count($records) > 1)
    I have multiple records!
@else
    I don't have any records!
@endif
```

Εικόνα 16: Παράδειγμα δήλωσης «if»


```

@for ($i = 0; $i < 10; $i++)
    The current value is {{ $i }}
@endfor

@foreach ($users as $user)
    <p>This is user {{ $user->id }}</p>
@endforeach

@forelse ($users as $user)
    <li>{{ $user->name }}</li>
@empty
    <p>No users</p>
@endforelse

@while (true)
    <p>I'm looping forever.</p>
@endwhile

```

Εικόνα 17: Παραδείγματα βρόχων (loops)

Επίσης, « η εντολή `@include` μας επιτρέπει να συμπεριλάβουμε μια προβολή Blade μέσα σε μια άλλη προβολή. Όλες οι μεταβλητές που είναι διαθέσιμες στη γονική προβολή θα διατίθενται και στη θυγατρική προβολή » (laravel.com - Blade Templates - Including Subviews 2021).

```

<div>
    @include('shared.errors')

    <form>
        <!-- Form Contents -->
    </form>
</div>

```

Εικόνα 18: Παράδειγμα εντολής `@include`

Το Blade επίσης, επιτρέπει στους προγραμματιστές να χρησιμοποιούν προκαθορισμένα πρότυπα. Το κύριο πλεονέκτημα της χρήσης τους είναι ότι μπορούμε να δημιουργήσουμε ένα κύριο πρότυπο, το οποίο μπορεί να επεκταθεί (να επαναχρησιμοποιηθεί) σε άλλα αρχεία. Αυτά τα πρότυπα είναι δυναμικά, γεγονός που μας επιτρέπει να γράφουμε λιγότερο κώδικα, πιο αποτελεσματικά και γρήγορα.

```

<!-- resources/views/layouts/app.blade.php -->

<html>
  <head>
    <title>App Name - @yield('title')</title>
  </head>
  <body>
    @section('sidebar')
      This is the master sidebar.
    @show

    <div class="container">
      @yield('content')
    </div>
  </body>
</html>

```

Εικόνα 19: Παράδειγμα κύριου Blade Template

Δεδομένου ότι οι περισσότερες εφαρμογές ιστού διατηρούν την ίδια διάταξη (layout) σε διάφορες σελίδες, είναι βολικό να ορίσουμε αυτήν τη κύρια διάταξη ως μία προβολή (αρχείο) Blade. Το αρχείο αυτό συνήθως περιέχει το header, το footer και το κεντρικό menu της εφαρμογής και αποθηκεύεται στον κατάλογο `/resources/views/layouts`. Στο παράδειγμά της παραπάνω εικόνας (laravel.com - Blade Templates - Defining A Layout 2021) αυτή η διάταξη είναι το αρχείο `app.blade.php`. Όπως φαίνεται στην εικόνα 19 το αρχείο αυτό περιέχει βασικές ετικέτες HTML, αλλά και τις εντολές `@section` και `@yield`. Η εντολή `@section`, ορίζει μια ενότητα περιεχομένου, ενώ η εντολή `@yield` χρησιμοποιείται για την εμφάνιση του περιεχομένου μιας δεδομένης ενότητας.

Αφού ορίσουμε την κύρια διάταξη, μπορούμε να ορίσουμε και θυγατρικές σελίδες που κληρονομούν αυτή τη διάταξη (laravel.com Blade Templates - Extending A Layout 2021). Στο παράδειγμα που ακολουθεί (Εικόνα 20), η θυγατρική σελίδα είναι το αρχείο `child.blade.php`

```
<!-- resources/views/child.blade.php -->

@extends('layouts.app')

@section('title', 'Page Title')

@section('sidebar')
    @parent

    <p>This is appended to the master sidebar.</p>
@endsection

@section('content')
    <p>This is my body content.</p>
@endsection
```

Εικόνα 20: Παράδειγμα θυγατρικού Blade Template

Όπως φαίνεται στο παραπάνω παράδειγμα, χρησιμοποιείται η εντολή `@extends` για να καθορίσουμε ποια κύρια διάταξη θα κληρονομήσει η θυγατρική προβολή. Αυτή είναι στο αρχείο `app.blade.php` που είναι αποθηκευμένο στον υποκατάλογο `layouts`.

3.3.8 Βάσεις Δεδομένων

Σύμφωνα με τη Laravel (laravel.com - Database: Getting Started 2021), το Laravel καθιστά την αλληλεπίδραση με βάσεις δεδομένων εξαιρετικά απλή και εύκολη υπόθεση, υποστηρίζοντας αρκετά συστήματα διαχείρισης βάσεων δεδομένων και χρησιμοποιώντας είτε καθαρή SQL, είτε τα εργαλεία Query Builder και Eloquent ORM. Έως και σήμερα, το Laravel υποστηρίζει τέσσερις βάσεις δεδομένων:

- MySQL
- PostgreSQL
- SQLite
- SQL Server

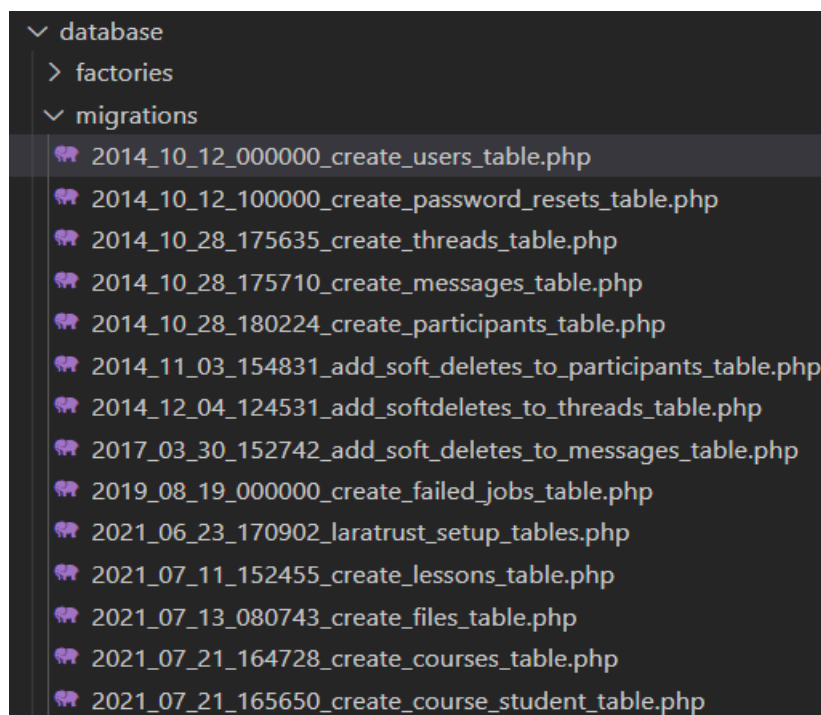
3.3.9 Migrations

Στο Laravel το Migration, παρέχει έναν εύκολο τρόπο για την κοινή χρήση του σχήματος (schema) της βάσης δεδομένων της εφαρμογής, δίνοντας τη δυνατότητα σε όλους τους προγραμματιστές που δουλεύουν επάνω στο ίδιο σχήμα να έχουν πρόσβαση στην ίδια έκδοση του σχήματος. Επίσης κάνει τη δημιουργία και την τροποποίηση του σχήματος της βάσης δεδομένων, με την χρήση απλών εντολών μέσω κονσόλας, πολύ πιο εύκολη. Είναι πολύ χρήσιμο ειδικά όταν έχουμε πολλούς πίνακες με πολλές στήλες, καθώς θα μειώσει το χρόνο εργασίας κατά πολύ σε σχέση με τη δημιουργία των πινάκων με μη αυτόματο τρόπο (laravel.com - Database: Migrations 2021).

Για να δημιουργήσουμε ένα Migration (laravel.com - Database: Migrations - Generating Migrations 2021), χρησιμοποιούμε την artisan εντολή:

```
php artisan make:migration create_users_table
```

Artisan, είναι ένα πανίσχυρο εργαλείο γραμμής εντολών μέσω κονσόλας, που περιέχεται στο Laravel και users είναι το όνομα του πίνακα που θα δημιουργηθεί στην βάση δεδομένων της εφαρμογής.



Εικόνα 21: Παράδειγμα Migration

Όλα τα Migrations αρχεία που δημιουργούμε χρησιμοποιώντας την εντολή: `php artisan make`, βρίσκονται στον κατάλογο `database/migrations`. Έτσι, όπως φαίνεται και στην εικόνα 21, αφού εκτελέσουμε την παραπάνω εντολή, θα δημιουργηθεί ένα αρχείο PHP με το όνομα που καθορίσαμε. Στο παράδειγμά μας το όνομα του αρχείου είναι: `2014_10_12_000000_create_users_table.php`.

Επίσης, το αρχείο θα δημιουργηθεί με κάποια προκαθορισμένη κλάση και συναρτήσεις όπως φαίνεται στην παρακάτω εικόνα (Εικόνα 22):

```
<?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();
            $table->string('name');
            $table->string('email')->unique();
            $table->string('town')->nullable();
            $table->string('region')->nullable();
            $table->string('address')->nullable();
            $table->string('postcode')->nullable();
            $table->string('phone1')->nullable();
            $table->string('phone2')->nullable();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }

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

Εικόνα 22: Παράδειγμα Migration αρχείου

Στην παραπάνω εικόνα, βλέπουμε την κλάση `CreateUsersTable` η οποία επεκτείνει την γενική κλάση `Migration`. Επίσης, μέσα στην κλάση υπάρχουν δύο συναρτήσεις: η `up()` και η `down()`. Η `up` καλείται όταν εκτελούμε το `Migration` για να δημιουργήσει τον πίνακα `users` και η συνάρτηση `down` καλείται όταν θέλουμε να τον διαγράψουμε. Μέσα στις συναρτήσεις, χρησιμοποιούνται οι μέθοδοι `create` και `dropIfExists`. Στην μέθοδο `create`, το πρώτο όρισμα είναι το όνομα του πίνακα που θα δημιουργηθεί (`users`) και το δεύτερο όρισμα είναι μια συνάρτηση με ένα αντικείμενο `blueprint`, που οι παράμετροί του είναι οι στήλες του πίνακα.

Τέλος, για να εκτελέσουμε όλα τα `Migrations` που έχουμε δημιουργήσει ([laravel.com - Database: Migrations - Running Migrations 2021](#)), πρέπει να εκτελέσουμε την `artisan` εντολή: **`php artisan migrate`**

3.3.10 Query Builder

Σύμφωνα με τη `Laravel` ([laravel.com - Database: Query Builder - Introduction 2021](#)), «το εργαλείο `Query Builder` του `Laravel`, παρέχει μια βολική και εύχρηστη διεπαφή για τη δημιουργία και εκτέλεση ερωτημάτων βάσης δεδομένων. Μπορεί να χρησιμοποιηθεί για την εκτέλεση των περισσότερων λειτουργιών βάσης δεδομένων στην εφαρμογή σας και λειτουργεί τέλεια με όλα τα υποστηριζόμενα συστήματα βάσεων δεδομένων του `Laravel`». Ας δούμε ένα παράδειγμα ([laravel.com - Database: Query Builder - Running Database Queries 2021](#)):

```
class UserController extends Controller
{
    /**
     * Show a list of all of the application's users.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $users = DB::table('users')->get();

        return view('user.index', ['users' => $users]);
    }
}
```

Εικόνα 23: Παράδειγμα `Query Builder` ερωτήματος

Στο παράδειγμα της εικόνας 23, χρησιμοποιούμε τη μέθοδο `table` που παρέχεται από το DB facade για να δημιουργήσουμε ένα απλό ερώτημα στον πίνακα `users`. Με αυτό το ερώτημα θέλουμε να ανακτήσουμε όλες τις εγγραφές του πίνακα. Η ανάκτηση επιτυγχάνεται με την μέθοδο `get()`.

Επιπρόσθετα, για την ανάκτηση μιας μεμονωμένης εγγραφής, μπορούμε να τροποποιήσουμε τον παραπάνω κώδικα προσθέτοντας μια δήλωση `where`:

```
$user = DB::table('users') -> where('name', 'John') -> first();
```

Στον παραπάνω κώδικα προσπαθούμε να ανακτήσουμε μια συγκεκριμένη εγγραφή (γραμμή) που έχει την τιμή `John` στη στήλη `name` του πίνακα `users`. Η μέθοδος `first()` θα επιστρέψει το πρώτο εύρημα. Αν θέλουμε όμως, από την συγκεκριμένη εγγραφή, μόνο το πεδίο `id` του χρήστη `John`, πρέπει να χρησιμοποιήσουμε την μέθοδο `pluck()`:

```
$user_id = DB::table('users') -> where('name', 'John') -> pluck('id');
```

Επίσης, μπορούμε να χρησιμοποιήσουμε την δήλωση `select`, για να ανακτήσουμε τιμές από περισσότερες από μία στήλες στον πίνακα:

```
$users = DB::table('users') -> select('name', 'email') -> get();
```

Τέλος, ο Query builder με την μέθοδο `raw()`, μας παρέχει τη δυνατότητα να γράψουμε ερωτήματα χρησιμοποιώντας καθαρή SQL. Μπορούμε ακόμα να καλέσουμε και μεθόδους για την ανάκτηση συγκεντρωτικών τιμών όπως `count()`, `max()`, `min()`, `avg()`, και `sum()` (Εικόνα 24).

```
$users = DB::table('users')
    ->select(DB::raw('count(*) as user_count, status'))
    ->where('status', '<>', 1)
    ->groupBy('status')
    ->get();
```

Εικόνα 24: Παράδειγμα Query Builder ερωτήματος

3.3.11 Eloquent ORM

Σύμφωνα με τη Wikipedia ([wikipedia.org](https://en.wikipedia.org/wiki/Object-relational_mapping) - Object-relational mapping 2021), το ORM ή Object Relational Mapper (Αντικειμενοστραφής Χαρτογράφηση) είναι μια

τεχνική πρόσβασης σε αντικείμενα χωρίς να χρειάζεται να λάβουμε υπόψη πώς αυτά τα αντικείμενα σχετίζονται με την πηγή τους.

Το ORM που περιλαμβάνεται στο Laravel ονομάζεται Eloquent και μας επιτρέπει να εργαζόμαστε με τα αντικείμενα και τις σχέσεις της βάσης δεδομένων, χρησιμοποιώντας πολύ εύκολη σύνταξη. Στο Laravel, κάθε πίνακας της βάσης δεδομένων έχει ένα αντίστοιχο "Μοντέλο" (Model). Το Eloquent ORM παρέχει μια απλή υλοποίηση του Active Record που σημαίνει ότι κάθε μοντέλο που δημιουργούμε αντιστοιχεί σε έναν πίνακα στη βάση δεδομένων μας. Εκτός από την ανάκτηση εγγραφών από τους πίνακες της βάσης δεδομένων, το Eloquent μας επιτρέπει να εισάγουμε, να ενημερώνουμε και να διαγράφουμε εγγραφές επίσης (laravel.com - Eloquent: Getting Started - Introduction 2021).

```
<?php

namespace App\Models;

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

class Course extends Model
{
    use HasFactory;

    protected $fillable = [
        'name',
        'description',
        'user_id',
        'status',
    ];

    public function lessons(){
        return $this->hasMany(Lesson::class);
    }
}
```

Εικόνα 25: Παράδειγμα Eloquent μοντέλου (Model Course)

Για την καλύτερη κατανόηση όσων αναφέρθηκαν παραπάνω, ας δούμε ένα παράδειγμα. Για να δημιουργήσουμε ένα Model (μοντέλο), χρησιμοποιούμε την artisan εντολή: **php artisan make:model Course** όπου, Course είναι το όνομα του

μοντέλου. Θα δημιουργηθεί έτσι ένα αρχείο με όνομα Course.php, το οποίο θα τοποθετηθεί στον κατάλογο app/Models.

Όπως φαίνεται και στην εικόνα 25, στο αρχείο Course.php υπάρχει η προκατασκευασμένη κλάση Course η οποία επεκτείνει την βασική κλάση Model. Επίσης, το όνομα της κλάσης, θα χρησιμοποιηθεί αυτόματα (στον πληθυντικό αριθμό) ως όνομα για τον πίνακα της βάσης δεδομένων στον οποίο θα αντιστοιχεί το μοντέλο αυτό. Στο παράδειγμά μας, το όνομα του πίνακα θα είναι το courses, εκτός και αν ορίσουμε ρητά κάποιο άλλο όνομα. Αυτό θα μπορούσε να γίνει, μέσα στην κλάση Course, με την εντολή: **protected \$table = 'my_courses'** για παράδειγμα.

Αφού δημιουργήσουμε το μοντέλο, μπορούμε να προβούμε στην ανάκτηση δεδομένων από τη βάση δεδομένων. Χρησιμοποιώντας την μέθοδο all(), μπορούμε να ανακτήσουμε όλες τις εγγραφές του πίνακα courses, όπως φαίνεται και στην εικόνα 26:

```
use App\Models\Course;

foreach (Course::all() as $course) {
    echo $course->name;
}
```

Εικόνα 26: Παράδειγμα ανάκτησης όλων των εγγραφών από τον πίνακα courses, χρησιμοποιώντας το μοντέλο Course

Ωστόσο, δεδομένου ότι κάθε μοντέλο Eloquent χρησιμεύει και ως εργαλείο δημιουργίας ερωτημάτων, μπορούμε να προσθέσουμε επιπλέον περιορισμούς στα ερωτήματα. Μπορούμε να χρησιμοποιήσουμε τη δήλωση where και στη συνέχεια με τη μέθοδο get(), να ανακτήσουμε τα αποτελέσματα (Εικόνα 27).

```
$courses = Course::where('status', 1)
    ->orderBy('name')
    ->get();
```

Εικόνα 27: Παράδειγμα ανάκτησης όλων των μαθημάτων (courses) που είναι ορατά (η στήλη status έχει την τιμή 1)

Όπως αναφέραμε και πιο πάνω, το Eloquent, εκτός από την ανάκτηση εγγραφών, μας επιτρέπει να εισάγουμε (create), να ενημερώνουμε (update) και να διαγράφουμε (delete / soft delete) εγγραφές. Στις παρακάτω εικόνες (Εικόνα 28, Εικόνα 29 και Εικόνα 30) παραθέτουμε αντίστοιχα παραδείγματα.

```
public function delete($id){  
  
    $user = Auth:: user();  
    $course=$user->courses()->find($id);  
    $course->delete();  
  
    return redirect('/dashboard/courses')  
        ->with('message','Το μάθημα διαγράφηκε επιτυχώς');  
}
```

Εικόνα 28: Παράδειγμα διαγραφής μαθήματος, χρησιμοποιώντας τη μέθοδο delete()

```
public function store(Request $request){  
  
    Course::create([  
        'user_id' => Auth::id(),  
        'status' => '1',  
        'name' => $request->name,  
        'description' => $request->description,  
    ]);  
  
    return redirect('/dashboard/courses')  
        ->with('message','Το μάθημα δημιουργήθηκε επιτυχώς');  
}
```

Εικόνα 29: Παράδειγμα προσθήκης νέου μαθήματος, χρησιμοποιώντας τη μέθοδο create()

```
public function update(Request $request ,$id){  
  
    $user = Auth:: user();  
    $course=$user->courses()->find($id);  
    $course->update($request->all());  
  
    return redirect('/dashboard/courses')  
        ->with('message','Το μάθημα ενημερώθηκε επιτυχώς');  
}
```

Εικόνα 30: Παράδειγμα ενημέρωσης μαθήματος, χρησιμοποιώντας τη μέθοδο update()

Στις περισσότερες εφαρμογές ιστού, οι πίνακες των βάσεων δεδομένων σχετίζονται μεταξύ τους. Για παράδειγμα, ένας χρήστης (με ρόλο μαθητή) μπορεί να είναι εγγεγραμμένος σε πολλά μαθήματα ή ένα μάθημα μπορεί να σχετίζεται με πολλούς χρήστες-μαθητές. Μέσα στα πλεονεκτήματα που παρέχει το Eloquent βρίσκονται και οι συσχετίσεις μεταξύ των διαφόρων μοντέλων που χρησιμοποιούνται σε μια εφαρμογή. Σύμφωνα με τη Laravel (laravel.com - Eloquent: Relationships - Introduction 2021), «το Eloquent διευκολύνει τη διαχείριση και τη συνεργασία των πινάκων υποστηρίζοντας μια ποικιλία κοινών σχέσεων (Relationships)». Οι σημαντικότερες είναι οι παρακάτω:

- Ένα προς ένα (One To One)
- Ένα προς πολλά (One To Many)
- Πολλά προς πολλά (Many To Many)

Ας δούμε ένα παράδειγμα για τη σχέση ένα προς πολλά. «Μια σχέση ενός προς πολλά χρησιμοποιείται για τον καθορισμό σχέσεων όπου ένα μόνο μοντέλο είναι ο γονέας ενός ή περισσότερων μοντέλων παιδιών» (laravel.com - Eloquent: Relationships - One To Many 2021). Για παράδειγμα, ένα μάθημα (course) μπορεί να έχει πολλές ενότητες (lessons). Στο Eloquent, η σχέση ένα προς πολλά, όπως και όλες οι άλλες σχέσεις, ορίζεται με τον καθορισμό μιας συνάρτησης μέσα στην κλάση του μοντέλου:

```
<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Course extends Model
{
    public function lessons(){
        return $this->hasMany(Lesson::class);
    }
}
```

Εικόνα 31: Παράδειγμα σχέσης ένα προς πολλά – η συνάρτηση `hasMany()`

Όπως φαίνεται στην εικόνα 31, η μέθοδος `hasMany()` της συνάρτησης `lessons()`, ορίζει ότι η κλάση `Course` (μάθημα) έχει πολλές κλάσεις `Lesson` (ενότητες).

Αφού οριστεί η μέθοδος `hasMany()`, μπορούμε να έχουμε πρόσβαση στις ιδιότητες (για παράδειγμα τους τίτλους) μιας συλλογής από ενότητες (`lessons`) ενός συγκεκριμένου μαθήματος (`course`) με δυναμικό τρόπο, όπως φαίνεται και στην εικόνα 32:

```
use App\Models\Course;

$lessons = Course::find(1)->lessons;

foreach ($lessons as $lesson) {
    echo $lesson->title;
}
```

Εικόνα 32: Παράδειγμα ανάκτησης δεδομένων από συσχετιζόμενα μοντέλα

Τέλος, μπορούμε να ορίσουμε και την αντίστροφη διαδικασία. Δηλαδή, μπορούμε να ορίσουμε μια σχέση που θα επιτρέπει σε μια ενότητα (`lesson`) να έχει πρόσβαση στο "μητρικό" του μάθημα (`course`) με δυναμικό τρόπο. Σύμφωνα με τη Laravel ([laravel.com - Eloquent: Relationships - One To Many \(Inverse\) / Belongs To 2021](https://laravel.com/docs/eloquent-relationships#one-to-many-inverse)), «για να ορίσουμε το αντίστροφο μιας σχέσης `hasMany`, πρέπει να ορίσουμε μια μέθοδο `belongsTo()` στο θυγατρικό μοντέλο» Συγκεκριμένα, όπως φαίνεται και στη εικόνα 33, ορίζουμε τη μέθοδο `belongsTo()` στη συνάρτηση `course()` μέσα στην κλάση `Lesson` του αντίστοιχου μοντέλου.

```
<?php

namespace App\Models;

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

class Lesson extends Model
{
    use HasFactory;

    protected $fillable = [
        'title',
        'body',
        'course_id',
    ];

    public function course(){
        return $this->belongsTo(Course::class);
    }

    public function files(){
        return $this->hasMany(File::class);
    }
}
```

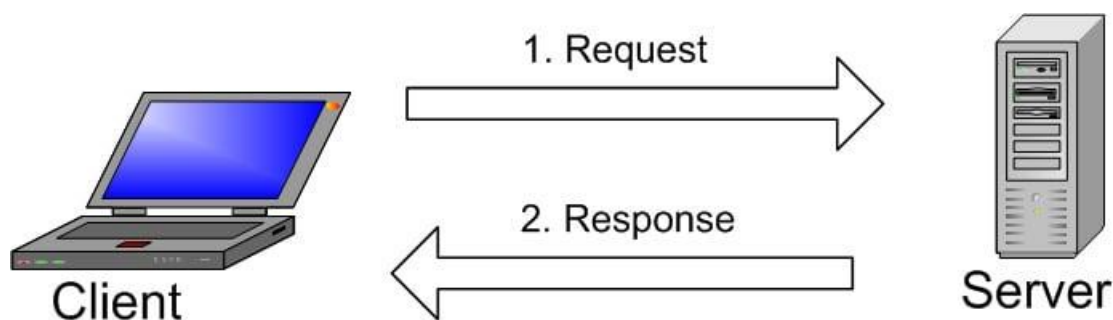
Εικόνα 33: Παράδειγμα σχέσης ένα προς πολλά – η συνάρτηση belongsTo()

3.4 MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS), που εκτελείται ("τρέχει") σε έναν εξυπηρετητή (server), παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων (wikipedia.org – MySQL 2021). Επομένως, δομικό στοιχείο του συστήματος της MySQL είναι ο αντίστοιχος εξυπηρετητής (server), ο οποίος ρυθμίζει την πρόσβαση των χρηστών στα δεδομένα υποστηρίζοντας (Γσιάτσος Θ, 2015):

- ασφάλεια και ελεγχόμενη πρόσβαση,
- ταυτόχρονη πρόσβαση πολλών χρηστών
- διαφορετικό επίπεδο δικαιωμάτων
- αξιοπιστία
- επεκτασιμότητα
- ταχύτητα

Η MySQL χρησιμοποιεί τη γλώσσα SQL (Structured Query Language). Σύμφωνα με τη Wikipedia (wikipedia.org - SQL 2021), η SQL «περιλαμβάνει δυνατότητες ανάκτησης και ενημέρωσης δεδομένων, δημιουργίας και τροποποίησης σχημάτων και σχεσιακών πινάκων, αλλά και ελέγχου πρόσβασης στα δεδομένα». Ανάλογα με το περιβάλλον προγραμματισμού που χρησιμοποιούμε, μπορούμε να εισάγουμε απευθείας SQL εντολές, ή να τις ενσωματώσουμε σε άλλη γλώσσα προγραμματισμού (όπως για παράδειγμα την PHP).



Εικόνα 34: Παράδειγμα μοντέλου πελάτη-εξυπηρετητή

Όπως φαίνεται και στην εικόνα 34, η MySQL, ακολουθώντας το μοντέλο πελάτη-εξυπηρετητή, δημιουργεί μια βάση δεδομένων για αποθήκευση και χειρισμό των δεδομένων, καθορίζοντας και τη σχέση του κάθε πίνακα της εφαρμογής. Οι πελάτες, μέσω της MySQL, μπορούν να υποβάλουν αιτήματα πληκτρολογώντας συγκεκριμένες δηλώσεις SQL και η εφαρμογή διακομιστή απαντάει με τις ζητούμενες πληροφορίες (hostinger.com - what-is-mysql Richard B. 2021).

Η MySQL είναι Ελεύθερο Λογισμικό Ανοικτού Κώδικα (open-source) και ο κώδικας της είναι διαθέσιμος μέσω της GNU (General Public License). Ωστόσο, υπάρχουν και άδειες για εμπορική χρήση. Χρηματοδοτείται από τη σουηδική εταιρεία MySQL AB, η οποία σήμερα ανήκει στην Oracle. Δεν είναι το μοναδικό DBMS στην αγορά, αλλά είναι ένα από τα πιο δημοφιλή. Χρησιμοποιείται σε κάποιες από τις πιο διαδεδομένες διαδικτυακές πλατφόρμες, όπως YouTube, Wikipedia, Google, Facebook κλπ. Επίσης είναι συμβατό με τα περισσότερα λειτουργικά συστήματα, όπως: Linux, Unix και Window (wikipedia.org – MySQL 2021).

380 systems in ranking, October 2021

Rank			DBMS	Database Model	Score		
Oct 2021	Sep 2021	Oct 2020			Oct 2021	Sep 2021	Oct 2020
1.	1.	1.	Oracle +	Relational, Multi-model ⓘ	1270.35	-1.19	-98.42
2.	2.	2.	MySQL +	Relational, Multi-model ⓘ	1219.77	+7.24	-36.61
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model ⓘ	970.61	-0.24	-72.51
4.	4.	4.	PostgreSQL +	Relational, Multi-model ⓘ	586.97	+9.47	+44.57
5.	5.	5.	MongoDB +	Document, Multi-model ⓘ	493.55	-2.95	+45.53
6.	6.	↑ 8.	Redis +	Key-value, Multi-model ⓘ	171.35	-0.59	+18.07
7.	7.	↓ 6.	IBM Db2	Relational, Multi-model ⓘ	165.96	-0.60	+4.06
8.	8.	↓ 7.	Elasticsearch	Search engine, Multi-model ⓘ	158.25	-1.98	+4.41
9.	9.	9.	SQLite +	Relational	129.37	+0.72	+3.95
10.	10.	10.	Cassandra +	Wide column	119.28	+0.29	+0.18

Εικόνα 35: Κατάταξη των δημοφιλέστερων DBMS

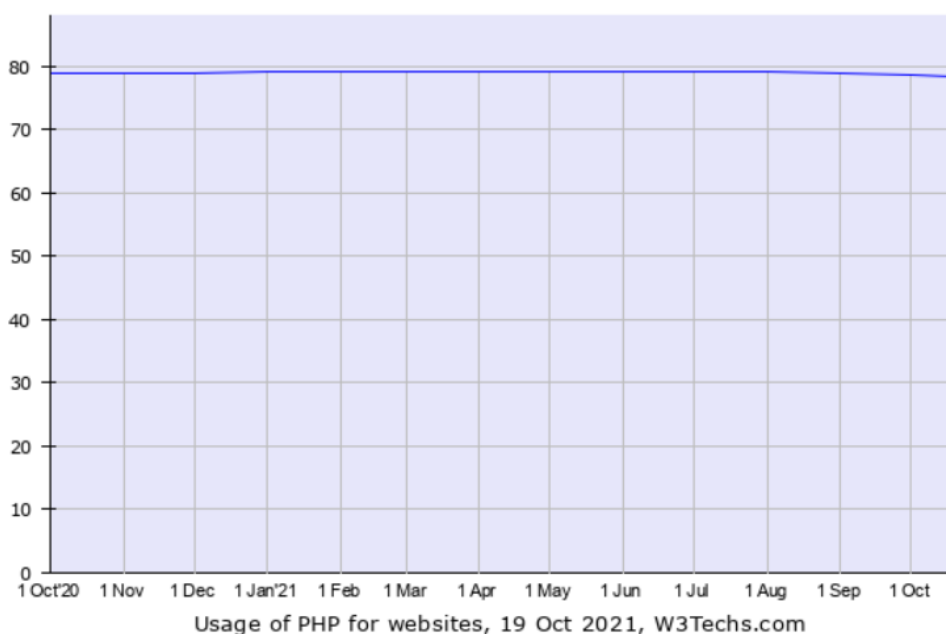
Σύμφωνα με τη DB-ENGINES (db-engines.com - DB-Engines Ranking 2021), όπως φαίνεται και στην εικόνα 35, η MySQL κατέχει τη δεύτερη θέση, πίσω από την Oracle, στην κατάταξη των δημοφιλέστερων DBMS.

4 Συγκρίνοντας το Laravel με άλλα PHP Frameworks

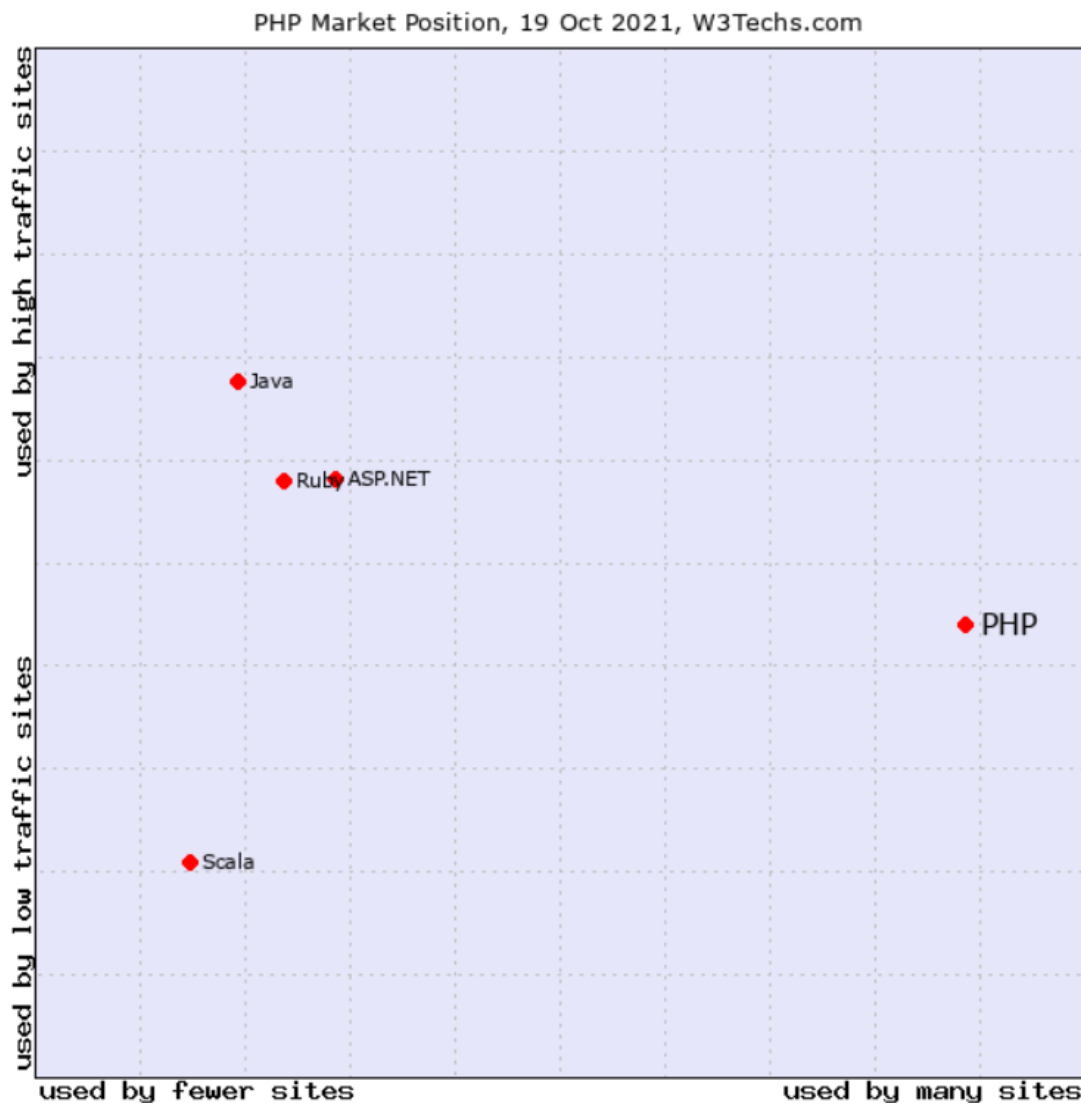
4.1 Χρήση της PHP

Σύμφωνα με τη Wikipedia (Wikipedia.org – PHP 2021), η PHP είναι η πιο εκτενώς χρησιμοποιούμενη γλώσσα προγραμματισμού στο διαδίκτυο. Είναι μια γλώσσα δέσμης ενεργειών ανοιχτού κώδικα από την πλευρά του διακομιστή (π.χ. Apache server). Διάσημες εφαρμογές κάνουν εκτενή χρήση της PHP, όπως το Facebook και η Wikipedia καθώς και πολύ γνωστά Σύστημα Διαχείρισης Περιεχομένου (CMS), όπως το WordPress και το Drupal.

Η PHP είναι μια εξαιρετικά δημοφιλής γλώσσα προγραμματισμού. Η ιστορία της ξεκινά από το 1994, φτάνοντας στις μέρες μας, να τη χρησιμοποιούν σχεδόν το 80% των ιστότοπων παγκοσμίως (Εικόνα 36). Επίσης στην Εικόνα 37, βλέπουμε τη θέση της PHP στην αγορά ως προς τη δημοτικότητα και την επισκεψιμότητα σε σύγκριση με τις πιο δημοφιλείς γλώσσες προγραμματισμού από την πλευρά του διακομιστή (w3techs.com - Usage statistics of PHP for websites 2021).



Εικόνα 36 : Χρήση της PHP στις διαδικτυακές εφαρμογές



Εικόνα 37 : Θέση της PHP σε σχέση με άλλες γλώσσες προγραμματισμού

4.2 Γιατί να χρησιμοποιήσουμε ένα PHP Framework

Η ανάπτυξη λογισμικού γίνεται όλο και πιο περίπλοκη, απαιτώντας τη χρήση ενοτήτων, προτύπων και εργαλείων για την παραγωγή αξιόπιστων εφαρμογών σε εύλογο χρονικό διάστημα.

Ένα πλαίσιο PHP παρέχει τη βασική δομή και πολλά άλλα στοιχεία και εργαλεία για τη δημιουργία εφαρμογών ιστού. Επιπλέον, επαναχρησιμοποιεί κομμάτια κώδικα για να αποφύγει επαναλαμβανόμενες εργασίες. Τελικά όμως, υπάρχουν πολλά περισσότερα οφέλη εκτός από την επιτάχυνση της διαδικασίας ανάπτυξης. Συνοπτικά θα μπορούσαμε να αναφέρουμε τα παρακάτω (hostinger.com - Why You Should Use PHP Frameworks 2021):

1) Εξοικονομεί χρόνο ανάπτυξης

Πολλοί παράγοντες στην ανάπτυξη ενός έργου βοηθούν τους προγραμματιστές να μειώσουν το χρόνο ανάπτυξης του. Τα πλαίσια PHP έχουν ενσωματωμένες λειτουργίες που επιτρέπουν στους προγραμματιστές να επιλέγουν και να τροποποιούν έτοιμο κώδικα (μέσω γεννητριών κώδικα) αντί να γράφουν τον δικό τους. Οι βιβλιοθήκες PHP προσφέρουν επίσης κοινά στοιχεία εφαρμογών ιστού, όπως επικύρωση φόρμας και λειτουργίες CRUD, έτσι ώστε να μην χρειάζεται να δημιουργηθούν από την αρχή.

Επιπλέον, τα περισσότερα πλαίσια PHP χρησιμοποιούν την αρχιτεκτονική MVC. Αυτό επιτρέπει στους προγραμματιστές να γράφουν ξεχωριστό κώδικα για τα τρία διαφορετικά συστατικά του μοντέλου (Model-View-Controller). Έτσι υποστηρίζεται παράλληλη ανάπτυξη, πράγμα που σημαίνει ότι ένας προγραμματιστής μπορεί να εργάζεται με τα Models, καθώς ένας άλλος προγραμματιστής θα εργάζεται στους Controllers, εξορθολογίζοντας και επιταχύνοντας τη διαδικασία παραγωγής του έργου.

2) Καλές πρακτικές κωδικοποίησης

Τα πλαίσια PHP επιτρέπουν στους προγραμματιστές να γράφουν κώδικα με τακτοποιημένο και οργανωμένο τρόπο. Χωρίζουν τον κώδικα σε τμήματα τα οποία αποθηκεύουν σε καταλόγους με βάση τις λειτουργίες του. Αυτό έχει σαν αποτέλεσμα η παραγωγή, η συντήρηση και επέκταση του κώδικα να γίνεται πιο γρήγορα και εύκολα.

3) Καλύτερη Συνεργασία

Συνήθως η ανάπτυξη διαδικτυακών εφαρμογών είναι ομαδική δουλειά. Πολλοί προγραμματιστές πρέπει να συνεργαστούν για ένα καλύτερο και πιο γρήγορο αποτέλεσμα. Η χρήση ενός πλαισίου PHP καθορίζει σαφείς οδηγίες που έχουν να κάνουν με την τεκμηρίωση και την χρήση προτύπων σχεδίασης και ανάπτυξης ενός έργου. Ακόμα κι αν προστεθεί στην ομάδα ένας νέος προγραμματιστής, ο οποίος δεν είναι εξοικειωμένος με το πλαίσιο, είναι σε θέση να ενσωματωθεί στην ομάδα γρήγορα και να συνεργαστεί για την παραγωγή του έργου.

4) Καλύτερη ασφάλεια

Στις εφαρμογές ιστού υπάρχουν πολυάριθμες αδυναμίες όσον αφορά την ασφάλεια τους. Καθημερινά, επιθέσεις χάκερ προσπαθούν να εκμεταλλευτούν κενά στην ασφάλεια τους, χρησιμοποιώντας απειλές: SQL injection, cross-site request forgery (CSRF) κλπ. Τα πλαίσια PHP χρησιμοποιούν καλές πρακτικές κωδικοποίησης για την ελαχιστοποίηση των προαναφερθέντων κινδύνων.

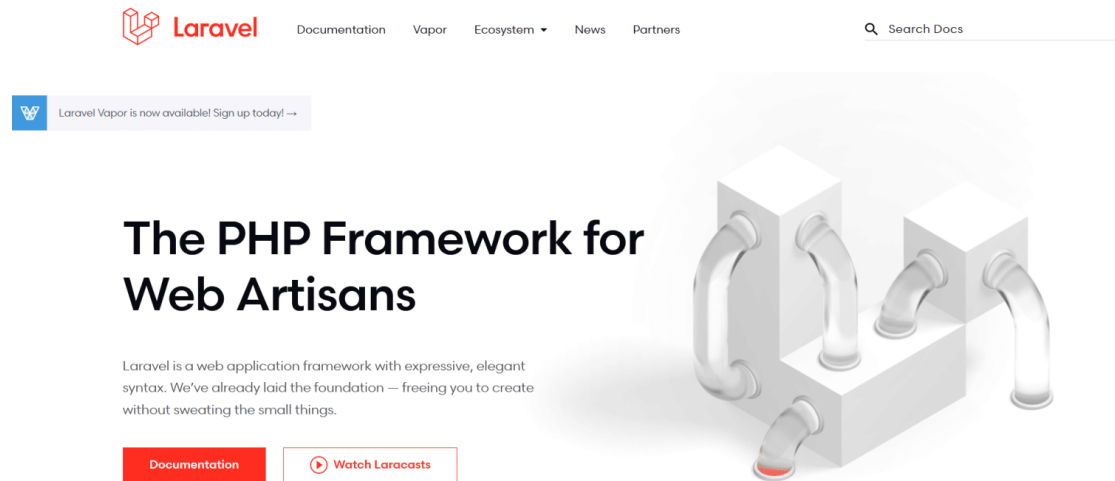
4.3 Κριτήρια για την επιλογή του σωστού Framework

Δεν είναι εύκολο να επιλέξουμε το κατάλληλο Framework για την ανάπτυξη μιας διαδικτυακής εφαρμογής. Στην αγορά υπάρχει πληθώρα Frameworks. Η Wikipedia (wikipedia.org - Category:PHP frameworks 2019) απαριθμεί 40 πλαίσια PHP, αλλά μερικά από αυτά περιγράφονται καλύτερα ως συστήματα διαχείρισης περιεχομένου (CMS) και σίγουρα υπάρχουν πολλά περισσότερα. Το κάθε ένα από αυτά έχει τα δικά του χαρακτηριστικά, πλεονεκτήματα και μειονεκτήματα. Συνήθως η επιλογή βασίζεται στον τύπο της εφαρμογής ιστού που θέλουμε να αναπτύξουμε και στις ανάγκες των προγραμματιστών. Σύμφωνα με το Symfony (symfony.com - Ten criteria for choosing the correct framework 2021), στην προσπάθειά μας να επιλέξουμε το κατάλληλο PHP Framework, πρέπει να λάβουμε υπόψη μας τα παρακάτω κριτήρια:

- Τη δημοτικότητα του πλαισίου και το μέγεθος της κοινότητας που το υποστηρίζει.
- Τη φιλοσοφία με την οποία έχει φτιαχτεί το πλαίσιο.
- Τη βιωσιμότητα του πλαισίου. Αυτό διευκολύνει και απλοποιεί, τόσο τη συντήρηση, όσο και την αναβάθμιση των εφαρμογών.
- Υποστήριξη: ευκολία εύρεσης απαντήσεων και λήψη βοήθειας.
- Τεχνική - Πρότυπα σχεδίασης.
- Ασφάλεια
- Τεκμηρίωση. Ένα καλά τεκμηριωμένο Framework είναι και πιο εύκολο στη χρήση και πιο αναβαθμίσιμο.
- Άδειες χρήσης.
- Διαθεσιμότητα πόρων στην αγορά.

- Δοκιμάστε το! Δοκιμάζοντας το, θα μπορέσουμε να αποφασίσουμε και να βεβαιωθούμε ότι είμαστε απόλυτα άνετοι με το Framework.

4.4 Περιγραφή και βασικά χαρακτηριστικά του Laravel



Εικόνα 38 : Το PHP πλαίσιο Laravel

Όπως αναφέραμε στο Κεφάλαιο 3 (3.3 Laravel), το Laravel είναι ένα δωρεάν, ανοικτού κώδικα PHP πλαίσιο (framework) που δημιουργήθηκε από τον Taylor Otwell, με κύριο στόχο την ανάπτυξη διαδικτυακών εφαρμογών που χρησιμοποιούν το μοντέλο MVC. Στον παρακάτω πίνακα (Πίνακας 2), βλέπουμε μερικά βασικά τεχνικά χαρακτηριστικά του:

Έτος έναρξης	2011
Τρέχουσα έκδοση	8
Είναι γραμμένο σε	PHP
Δημιουργός	Taylor Otwell
Άδεια	MIT License
Απαιτήσεις διακομιστή	PHP >= 7.2.5 (ή χρήση του Laravel Homestead)
Βάσεις Δεδομένων που υποστηρίζει	MySQL, PostgreSQL, SQLite, SQL Server

Πίνακας 2 : Βασικά τεχνικά χαρακτηριστικά του Laravel

Σύμφωνα με τη Brotherton (kinsta.com – The Most Popular PHP Frameworks to Use in 2021: Pros of Laravel 2021), τα βασικότερα χαρακτηριστικά - πλεονεκτήματα του Laravel είναι τα παρακάτω:

1) Είναι εύκολο να ξεκινήσει κανείς με το Laravel, γιατί υπάρχει το Homestead. Το Homestead είναι ένα προσυσκευασμένο, εικονικό περιβάλλον ανάπτυξης που παρέχεται από το πλαίσιο. Έτσι, για την ανάπτυξη μιας εφαρμογής τοπικά (local), δεν χρειάζεται η εγκατάσταση της PHP, του διακομιστή ιστού (π.χ. Apache server) ή οποιουδήποτε άλλου λογισμικού διακομιστή (π.χ. MySQL server).

2) Χρησιμοποιεί μια μηχανή προτύπων (templating engine) που ονομάζεται Blade. Ένα από τα πολλά πλεονέκτημα του Blade, είναι ότι μπορούμε να χρησιμοποιήσουμε καθαρή PHP, κάτι που δεν μπορούμε να κάνουμε με τα άλλα Frameworks που χρησιμοποιούν αντίστοιχες μηχανές προτύπων.

3) Παρέχει μια σειρά από χαρακτηριστικά και μεθόδους ασφαλείας, που καλύπτουν:

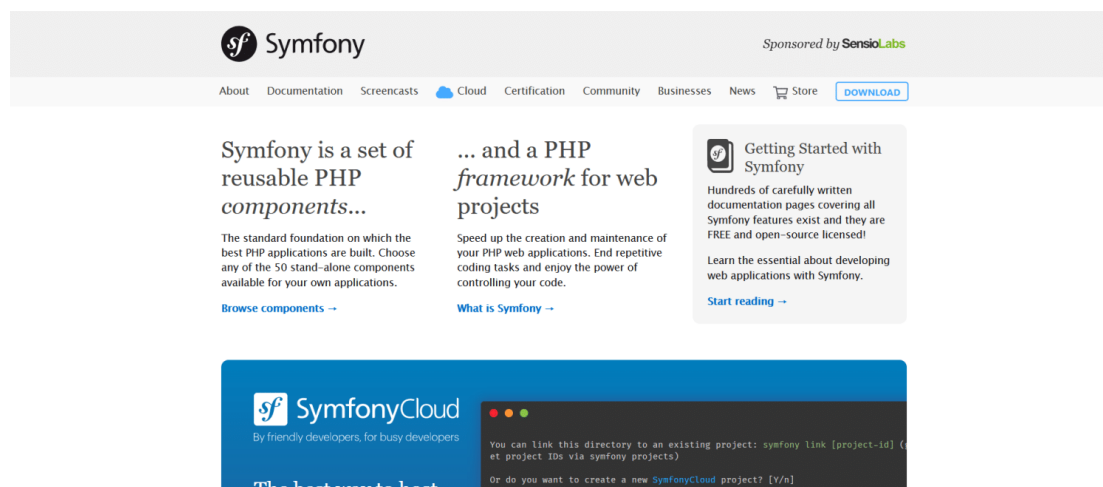
- Αυθεντικοποίηση
- Εξουσιοδότηση
- Επαλήθευση email
- Κρυπτογράφηση
- Hashing
- Επαναφορά κωδικού

4) Προσφέρει αντικειμενοστραφής χαρτογράφηση (ORM) με το εργαλείο Eloquent. Έτσι, επιτρέπει στους προγραμματιστές να γράφουν ερωτήματα στη βάση δεδομένων της εφαρμογής με σύνταξη PHP, χωρίς να είναι απαραίτητο να χρησιμοποιήσουν καθαρή SQL. Επίσης, για τη δημιουργία και εκτέλεση ερωτημάτων βάσης δεδομένων διαθέτει και το εργαλείο Query Builder.

5) Διαθέτει το εργαλείο γραμμής εντολών Artisan Console. Επιταχύνεται έτσι η ανάπτυξη της εφαρμογής, επιτρέποντας στους προγραμματιστές να αυτοματοποιούν επαναλαμβανόμενες εργασίες και να δημιουργούν γρήγορα σκελετό κώδικα.

6) Διαθέτει μια μεγάλη κοινότητα προγραμματιστών, όπως: Laracasts, Laravel.io, The Laravel subreddit και άλλα πολλά.

4.5 Περιγραφή και βασικά χαρακτηριστικά του Symfony



Εικόνα 39 : Το PHP πλαίσιο Symfony

Το Symfony είναι ένα PHP πλαίσιο (framework) καθώς και ένα σύνολο επαναχρησιμοποιήσιμων PHP στοιχείων/βιβλιοθηκών. Δημοσιεύτηκε ως λογισμικό ανοιχτού κώδικα το 2005 και κυκλοφόρησε με άδεια του MIT (wikipedia.org – Symfony 2021). Στον παρακάτω πίνακα (Πίνακας 3), βλέπουμε μερικά βασικά τεχνικά χαρακτηριστικά του:

Έτος έναρξης	2005
Τελευταία έκδοση	5.1.4
Είναι γραμμένο σε	PHP
Δημιουργός	Fabien Potencier
Άδεια	MIT License
Απαιτήσεις διακομιστή	PHP >= 7.2.5
Βάσεις Δεδομένων που υποστηρίζει	Drizzle, MySQL, Oracle, PostgreSQL, SAP Sybase SQL Anywhere, SQLite, SQL Server

Πίνακας 3 : Βασικά τεχνικά χαρακτηριστικά του Symfony

Σύμφωνα με τη Brotherton (kinsta.com – The Most Popular PHP Frameworks to Use in 2021: Pros of Symfony 2021), τα βασικότερα χαρακτηριστικά - πλεονεκτήματα του Symfony είναι τα παρακάτω:

1) Είναι μια εξαιρετική επιλογή για ιστότοπους και εφαρμογές ιστού που πρέπει να είναι κλιμακούμενες. Είναι ένα πολύ ευέλικτο πλαίσιο. Το αρθρωτό σύστημα επαναχρησιμοποιήσιμων στοιχείων που περιέχει, δεν περιορίζει τους προγραμματιστές στο περιβάλλον του πλαισίου, αλλά τους επιτρέπει να επιλέξουν και

να χρησιμοποιήσουν στην εφαρμογή τους μόνο τα στοιχεία που χρειάζονται. Περιλαμβάνει πολλά πρόσθετα, φύλλα στυλ, αρχεία PHP, εικόνες, JavaScript και πολλές άλλες δυνατότητες. Έτσι, οι προγραμματιστές δεν χρειάζεται να εγκαταστήσουν ολόκληρο το πλαίσιο, αλλά μόνο τα συγκεκριμένα στοιχεία.

2) Υποστηρίζει τις περισσότερες βάσεις δεδομένων, σε σχέση με τα υπόλοιπα δημοφιλή PHP πλαίσια:

- Drizzle
- MySQL
- Oracle
- PostgreSQL
- SAP Sybase SQL Anywhere
- SQLite
- SQL Server

3) Βασίζεται στο πρότυπο MVC (Model, View, Controller) και προσφέρει μια ποικιλία εργαλείων γραμμής εντολών (εργαλεία CLI), τα οποία είναι χρήσιμα στην επιτάχυνση της διαδικασίας ανάπτυξης της εφαρμογής.

4) Προσφέρει αντικειμενοστραφής χαρτογράφηση (ORM) παρέχοντας το εργαλείο Doctrine.

5) Ο εντοπισμός σφαλμάτων στις εφαρμογές που γράφονται με το Symfony είναι απλός με το ενσωματωμένο εργαλείο που διαθέτει.

6) Χρησιμοποιεί τη μηχανή προτύπων (templating engine) Twig, η οποία είναι εύκολη στην εκμάθηση, γρήγορη και ασφαλής.

7) Διαθέτει μια τεράστια κοινότητα με πάνω από 600.000 προγραμματιστές να συμμετέχουν ενεργά. Οι προγραμματιστές που χρησιμοποιούν το Symfony μπορούν να λάβουν βοήθεια μέσω πολλών καναλιών: Symfony full documentation, Sensio Labs University, SymfonyCasts, Symfony Certification και άλλα.

4.6 Περιγραφή και βασικά χαρακτηριστικά του CodeIgniter



Εικόνα 40 : Το PHP πλαίσιο CodeIgniter

Το CodeIgniter είναι ένα ανοιχτού κώδικα, ταχείας ανάπτυξης PHP πλαίσιο (framework), το οποίο χρησιμοποιείται για τη δημιουργία δυναμικών ιστοσελίδων και εφαρμογών ιστού. Αρχικά αναπτύχθηκε από την εταιρεία EllisLab, αλλά αργότερα υιοθετήθηκε από το British Columbia Institute of Technology (wikipedia.org – CodeIgniter 2021). Σύμφωνα με τη CodeIgniter (codeigniter.com - CodeIgniter Rocks 2021), είναι «ένα ισχυρό PHP πλαίσιο με πολύ μικρό αποτύπωμα, σχεδιασμένο για προγραμματιστές που χρειάζονται μια απλή και κομψή εργαλειοθήκη για τη δημιουργία εφαρμογών web με πλήρεις δυνατότητες». Στον παρακάτω πίνακα (Πίνακας 4), βλέπουμε μερικά βασικά τεχνικά χαρακτηριστικά του:

Έτος έναρξης	2006
Τελευταία έκδοση	4.1.4
Είναι γραμμένο σε	PHP
Δημιουργός	EllisLab
Άδεια	MIT License
Απαιτήσεις διακομιστή	PHP >= 7.2
Βάσεις Δεδομένων που υποστηρίζει	MySQL, PostgreSQL, SQLite3

Πίνακας 4 : Βασικά τεχνικά χαρακτηριστικά του CodeIgniter

Σύμφωνα με τη Brotherton (kinsta.com – The Most Popular PHP Frameworks to Use in 2021: Pros of CodeIgniter 2021), τα βασικότερα χαρακτηριστικά - πλεονεκτήματα του CodeIgniter είναι τα παρακάτω:

- 1) Είναι πραγματικά γρήγορο και ελαφρύ. Είναι ένα από τα πιο γρήγορα PHP πλαίσια που κυκλοφορούν στην αγορά. Υποστηρίζει πολλαπλές μεθόδους προσωρινής αποθήκευσης (caching), επιταχύνοντας τις εφαρμογές ιστού.
- 2) Είναι ένα πλαίσιο με μικρό αποτύπωμα (μέγεθος λήψης περίπου 1,2 MB, συμπεριλαμβανομένης της τεκμηρίωσης). Οι προγραμματιστές μπορούν να προσθέσουν ακριβώς τα στοιχεία που χρειάζονται.
- 3) Ενθαρρύνει την ανάπτυξη εφαρμογών που βασίζονται στο πρότυπο MVC (Model, View, Controller), αλλά μπορεί επίσης να τροποποιηθεί για να χρησιμοποιηθεί το πρότυπο HMVC (Hierarchical Model View Controller).
- 4) Είναι εύκολο στην εκμάθηση του, ειδικά για αρχάριους προγραμματιστές, σε σύγκριση με άλλα πλαίσια. Επίσης είναι εύκολο στην εγκατάσταση του και είναι αρκετά επεκτάσιμο, αφού συνοδεύεται από μια πλήρη γκάμα βιβλιοθηκών που συνήθως απαιτούνται για εργασίες ανάπτυξης εφαρμογών ιστού.
- 5) Περιλαμβάνει ενσωματωμένα εργαλεία τα οποία παρέχουν προστασία έναντι επιθέσεων CSRF και XSS.

4.7 Συγκρίνοντας το Laravel με το Symfony

	LARAVEL	SYMFONY
MODULARITY AND SCALING	Χρησιμοποιεί εφαρμογές που βασίζονται στο πρότυπο MVC, με έναν αριθμό προκατασκευασμένων εξαρτήσεων. Αυτό το καθιστά ελαφρώς λιγότερο ευέλικτο, αλλά πιο φιλικό προς το χρήστη.	Χρησιμοποιεί επαναχρησιμοποιήσιμα στοιχεία, εξασφαλίζοντας πιο στιβαρή αρθρωτότητα. Ο κώδικας είναι καλύτερα οργανωμένος.
TEMPLATING ENGINE	Η προεπιλεγμένη μηχανή προτύπων είναι το Blade, επιτρέποντας την επαναχρησιμοποίηση κώδικα.	Χρησιμοποιεί το Twig ως την προεπιλεγμένη μηχανή προτύπων.
DATABASE SUPPORT	Χρησιμοποιεί αντικειμενοστραφής χαρτογράφηση (ORM) για πρόσβαση σε δεδομένα μέσω του Eloquent. Υποστηρίζει τις ακόλουθες βάσεις δεδομένων: MySQL, PostgreSQL, SQLite και SQL Server.	Χρησιμοποιεί αντικειμενοστραφής χαρτογράφηση (ORM) για πρόσβαση σε δεδομένα μέσω του Doctrine. Υποστηρίζει τις ακόλουθες βάσεις δεδομένων: Drizzle, MySQL, Oracle, PostgreSQL, SAP Sybase SQL Anywhere, SQLite, SQL Server.
DATABASE MIGRATIONS	Το Migration είναι χειροκίνητο, αλλά δεν απαιτείται ο καθορισμός των πεδίων ενός πίνακα της βάσης δεδομένων.	Το Migration είναι αυτόματο, απαιτώντας μόνο απλούς ορισμούς για τα πεδία ενός πίνακα της βάσης δεδομένων, μέσα στο αντίστοιχο μοντέλο.
DATA MODELING	Απαιτεί ουσιαστική γνώση SQL.	Δεν απαιτεί σημαντική γνώση SQL.

Πίνακας 5 : Σύγκριση του Laravel με το Symfony

Στο σημείο αυτό, θα επιχειρήσουμε μια σύγκριση των PHP πλαισίων Laravel και Symfony. Στον Πίνακα 5, βλέπουμε τις διαφορές στα πιο βασικά χαρακτηριστικά μεταξύ των δύο πλαισίων. Τα χαρακτηριστικά αυτά έχουν να κάνουν με τις εξής κατηγορίες: Modularity, Scaling, Templating Engine, Database Support, Migration

και Data Modeling (asperbrothers.com - Laravel vs Symfony in 2021 – Which PHP Framework Choose For Your Project 2019).

Το Laravel και το Symphony είναι δύο δημοφιλή και αξιόπιστα πλαίσια. Και τα δύο προσφέρουν ένα πλούσιο σύνολο χαρακτηριστικών και είναι ιδανικά για ένα ευρύ φάσμα ανάπτυξης εφαρμογών ιστού. Όπως αναφέραμε και προηγουμένως, το κρίσιμο είναι να εξετάσουμε ποια εφαρμογή θέλουμε να αναπτύξουμε, ποια χαρακτηριστικά πρέπει να έχει και ποιον στόχο θέλουμε να πετύχουμε.

Το Laravel φαίνεται να είναι ιδανικό για προγραμματιστές οι οποίοι στοχεύουν στην γρήγορη ανάπτυξη μιας εφαρμογής. Εστιάζει περισσότερο στην απλότητα και την απόδοση ανάπτυξης, ακόμα και για προγραμματιστές που δεν είναι προχωρημένοι.

Όταν θέλουμε για παράδειγμα, να αναπτύξουμε εφαρμογές ιστού με εξουσιοδότηση χρήστη, το Laravel είναι ένα πλαίσιο που μπορεί να μας βοηθήσει πολύ. Χρησιμοποιώντας τα ενσωματωμένα εργαλεία που προσφέρει, όπως το Eloquent ORM, το Migration, καθώς και το εργαλείο γραμμής εντολών Artisan, μπορούμε να πετύχουμε το στόχο μας εύκολα και γρήγορα, αποφεύγοντας τον πολύ περίπλοκο και πολύ μεγάλο κώδικα.

Το Symphony από την άλλη, είναι ένα πλαίσιο το οποίο απευθύνεται σε προγραμματιστές οι οποίοι σχεδιάζουν να δημιουργήσουν εξειδικευμένες και μεγάλης κλίμακας εφαρμογές ιστού. Είναι μια λύση για μακροπρόθεσμα και πολύπλοκα έργα. Επίσης, στοχεύει σε προχωρημένους προγραμματιστές γιατί είναι λίγο πιο δύσκολο στην εκμάθηση και στην χρήση του, ειδικά στην αρχή.

Τέλος, το Symphony είναι ιδανικό όταν θέλουμε να εξατομικεύουμε την εφαρμογή μας. Όπως είπαμε και πιο πάνω, το πλαίσιο αποτελείται από επαναχρησιμοποιούμενα πακέτα. Νέες εκδόσεις του Symphony, (για παράδειγμα το Symphony 4) μας παρέχουν μόνο τα πιο απαραίτητα και μας επιτρέπουν να προσθέσουμε μόνο εκείνα τα στοιχεία που πραγματικά θα απαιτηθούν. Έτσι, δεν χρειάζεται να αλλάξουμε ολόκληρο το έργο.

4.8 Συγκρίνοντας το Laravel με το CodeIgniter

	Laravel	CodeIgniter
Database Model	Αντικειμενοστραφές Μοντέλο	Σχεσιακό Αντικειμενοστραφές Μοντέλο
Built-in Module	Διαθέτει διάφορες ενσωματωμένες μονάδες (modules), που βοηθούν στη διαδικασία ανάπτυξης	Δεν διαθέτει ενσωματωμένες μονάδες (modules).
Popularity	Είναι το πιο δημοφιλές PHP πλαίσιο.	Πολλοί προγραμματιστές το προτιμούν επειδή προσφέρει ευκολία στη χρήση του.
Structure	Ακολουθεί το πρότυπο MVC.	Ενθαρρύνει την ανάπτυξη εφαρμογών που βασίζονται στο πρότυπο MVC, αλλά μπορεί επίσης να τροποποιηθεί για να χρησιμοποιηθεί το πρότυπο HMVC.
Database Support	Βάσεις Δεδομένων που υποστηρίζει: MySQL, PostgreSQL, SQLite, SQL Server.	Βάσεις Δεδομένων που υποστηρίζει: MySQL, PostgreSQL, SQLite3
Templates	Χρησιμοποιεί έναν ισχυρό μηχανισμό προτύπων που ονομάζεται Blade. Το Blade επιτρέπει στους προγραμματιστές να βελτιώνουν και να χειρίζονται τις προβολές (views) της εφαρμογής.	Δεν συνοδεύεται από ενσωματωμένη μηχανή προτύπων. Οι προγραμματιστές πρέπει να χρησιμοποιήσουν εξωτερικά εργαλεία προτύπων.
Documentation	Προσφέρει επίσημη τεκμηρίωσή, η οποία παρέχει υποστήριξη στον προγραμματιστή κατά τη διαδικασία ανάπτυξης.	Προσφέρει μικρή, αλλά επαρκή τεκμηρίωση. Ο επίσημος ιστότοπός της διαθέτει επίσης έναν οδηγό που μπορούν να χρησιμοποιήσουν οι προγραμματιστές.

Testing	Η ενσωματωμένη λειτουργία δοκιμής μονάδας επιτρέπει τους προγραμματιστές να δοκιμάζουν τον κώδικα διεξοδικά και συνεχώς.	Δεν διαθέτει ενσωματωμένα εργαλεία δοκιμής. Επομένως, οι προγραμματιστές πρέπει να χρησιμοποιούν πρόσθετα εργαλεία.
Authentication	Παρέχει ενσωματωμένες λειτουργίες, που επιτρέπουν στους προγραμματιστές να εφαρμόζουν εύκολα τον έλεγχο ταυτότητας και εξουσιοδότησης χρηστών.	Οι προγραμματιστές, για να υλοποιήσουν λειτουργίες ελέγχου ταυτότητας και εξουσιοδότησης χρηστών, πρέπει να γράφουν προσαρμοσμένες επεκτάσεις CodeIgniter.
Beginners	Δεν είναι εύκολο για τους αρχάριους προγραμματιστές να μάθουν και να χρησιμοποιήσουν το πλαίσιο, λόγω των πολλών δυνατοτήτων που προσφέρει.	Είναι πραγματικά εύκολο για τους αρχάριους προγραμματιστές να μάθουν και να χρησιμοποιήσουν το CodeIgniter, γεγονός που το καθιστά το κορυφαίο PHP πλαίσιο για αρχάριους.

Πίνακας 6 : Σύγκριση του Laravel με το CodeIgniter

Εδώ, θα επιχειρήσουμε μια σύγκριση των PHP πλαισίων Laravel και CodeIgniter. Στον Πίνακα 6, βλέπουμε τις διαφορές στα πιο βασικά χαρακτηριστικά μεταξύ των δύο πλαισίων. Τα χαρακτηριστικά αυτά έχουν να κάνουν με τις εξής κατηγορίες: Database Model, Built-in Module, Popularity, Structure, Database Support, **Templates**, Documentation, Testing, Authentication και Beginners (mavencluster.com - Laravel vs Codeigniter-Which is Better PHP Framework 2021).

Και τα δύο αυτά πλαίσια έχουν σημαντικά και μοναδικά χαρακτηριστικά.

Το Laravel (όπως είδαμε και πιο πάνω) διαθέτει μια προκαθορισμένη αρχιτεκτονική (MVC) που διευκολύνει την ανάπτυξη της εφαρμογής. Προσφέρει ένα σύστημα ελέγχου εκδόσεων που βοηθά στην απλοποιημένη διαχείριση και μετεγκατάσταση του πλαισίου. Υποστηρίζει το εργαλείο Eloquent ORM και πλατφόρμες DBMS όπως MySQL, PostgreSQL και SQL Server. Διαθέτει εξαιρετική τεκμηρίωση. Χρησιμοποιεί αρθρωτά πακέτα και προσφέρει χαρακτηριστικά όπως τα πρότυπα

Blade, για τη δημιουργία μιας κομψής και προηγμένης εφαρμογής ιστού. Τέλος, διαθέτει μια τεράστια κοινότητα ειδικών της PHP και του Laravel.

Συνολικά, οι προγραμματιστές PHP μπορούν να επωφεληθούν από μια σειρά από πρόσθετες δυνατότητες επιλέγοντας το Laravel. Αλλά οι αρχάριοι προγραμματιστές πρέπει να αφιερώσουν επιπλέον χρόνο και προσπάθεια για να μάθουν και να χρησιμοποιήσουν αποτελεσματικά το πλαίσιο.

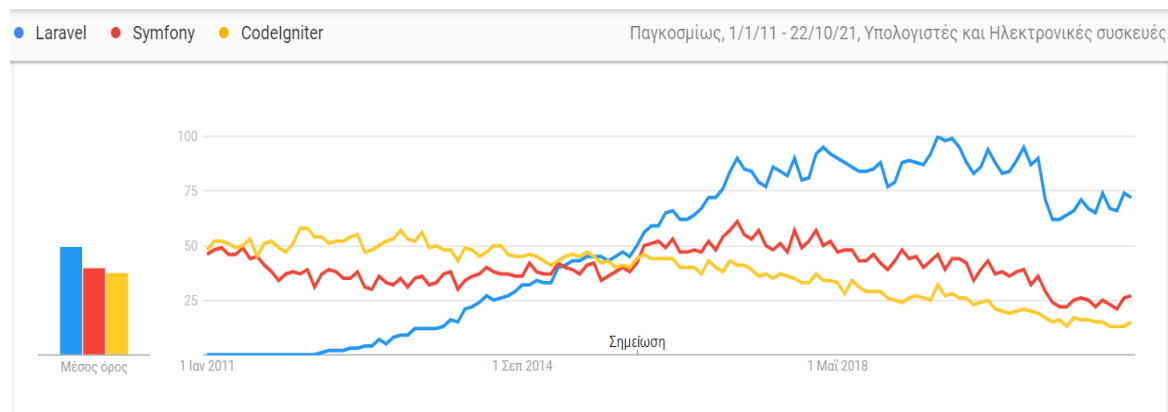
Το CodeIgniter διαθέτει πληθώρα πρόσθετων (plug-ins), έτοιμων προς χρήση βιβλιοθηκών και διάφορων άλλων πόρων. Διευκολύνει το έργο της ανάπτυξης εφαρμογών ιστού παρέχοντας διάφορες προηγμένες δυνατότητες στον προγραμματιστή. Παρέχει λειτουργίες για δυναμική και αποκριτική (responsive) ανάπτυξη εφαρμογών. Ταυτόχρονα, το CodeIgniter είναι εύκολο τόσο στην κατανόηση, όσο και στην χρήση του, διευκολύνοντας τους αρχάριους προγραμματιστές.

Επομένως, εάν θέλουμε να δημιουργήσουμε μια ελαφριά web εφαρμογή, η οποία να διαθέτει εντυπωσιακά χαρακτηριστικά με κομψή διεπαφή χρήστη και βελτιωμένη ασφάλεια, τότε θα πρέπει να επιλέξουμε το CodeIgniter.

4.9 Επιλογή PHP Framework σύμφωνα με τη δημοτικότητα και τις τάσεις της αγοράς

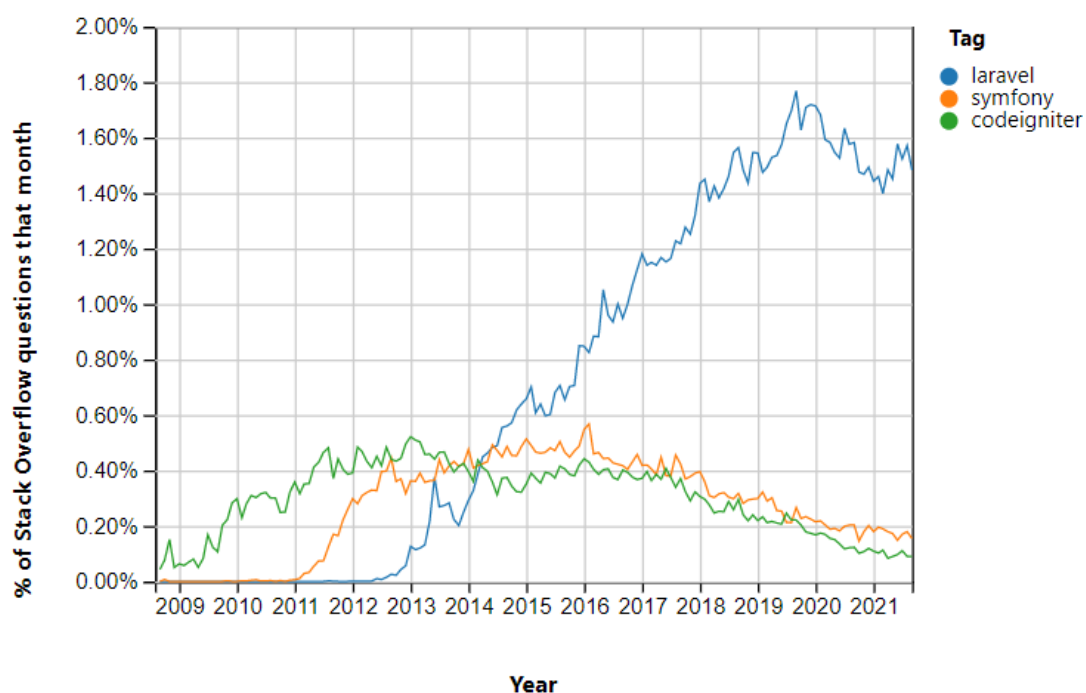
Προκειμένου να διερευνήσουμε ποιο από τα τρία PHP Framework που εξετάζουμε είναι το πιο δημοφιλές στην αγορά, θα χρησιμοποιήσουμε τρεις αξιόπιστες πηγές: Google Trends, Stack Overflow Trends και Github Stars.

Σύμφωνα με τη Google Trends (trends.google.com – Σύγκριση 2021), όπως φαίνεται στην εικόνα 41, το Laravel είναι ξεκάθαρα μπροστά στη λίστα με τα δημοφιλέστερα PHP Frameworks το 2021, ενώ το Symfony και το CodeIgniter βρίσκονται πολύ κοντά το ένα με το άλλο, με ένα μικρό προβάδισμα του πρώτου.



Εικόνα 41 : Google Trends [2019]

Και πάλι, σύμφωνα με τις τάσεις Stack Overflow (insights.stackoverflow.com - Stack Overflow Trends 2021), το Laravel είναι πολύ μπροστά στη λίστα με τα δημοφιλέστερα PHP πλαίσια το 2021 (Εικόνα 42). Επίσης, βλέπουμε ότι το Laravel από το 2012 και έπειτα, έχει σημειώσει πολύ μεγάλη ανάπτυξη, ενώ τα άλλα δύο πλαίσια, είτε έχουν παραμείνει σχετικά στάσιμα, είτε έχουν μειωθεί σε δημοτικότητα.



Εικόνα 42 : Stack Overflow Trends [2019]

Ο έλεγχος του αριθμού των Github Stars (github.com - Github-Ranking/Top100 2021) είναι ένας από τους καλύτερους τρόπους ελέγχου της δημοτικότητας μιας συγκεκριμένης τεχνολογίας.

Όπως φαίνεται ξεκάθαρα από τον παρακάτω πίνακα (Πίνακας 7), το Laravel προηγείται σαφώς (66806 αστέρια) στη λίστα με τα δημοφιλέστερα PHP Frameworks το 2021, ενώ τα Symfony και CodeIgniter βρίσκονται αρκετά πίσω με λιγότερο από τα μισά αστέρια (25819 και 18167 αντίστοιχα). Παρά το γεγονός ότι το Laravel εμφανίστηκε 5-6 χρόνια αργότερα, σε σύγκριση με τα άλλα δύο πλαίσια, έχει καταφέρει να κερδίσει πολλά περισσότερα αστέρια Github.

Github Ranking

Project Name	Stars	Date of First Commit	Description
laravel	66806	Thursday, 9 June 2011	A PHP framework for web artisans.
symfony	25819	Monday, 4 January 2010	The Symfony PHP framework
CodeIgniter	18167	Friday, 25 August 2006	Open Source PHP Framework (originally from EllisLab)

Πίνακας 7 : Github Stars [2021]

5 Περιγραφή και Δομή της εφαρμογής

5.1 Περιγραφή

Η εφαρμογή που αναπτύξαμε, είναι ένα σύστημα διαχείρισης ηλεκτρονικών μαθημάτων μέσω διαδικτύου.

Πιο συγκεκριμένα, είναι ένα σύστημα διαχείρισης περιεχομένου, το οποίο υποστηρίζει τρεις ρόλους χρηστών: τον ρόλο του διαχειριστή (Admin), τον ρόλο του εκπαιδευτή (Teacher) και τον ρόλο του εκπαιδευόμενου (Student). Μέσα από τις λειτουργίες που υλοποιούνται στην εφαρμογή, προσφέρονται οι παρακάτω δυνατότητες:

1) Δυνατότητες που έχουν να κάνουν με όλους τους χρήστες της εφαρμογής. Ένας χρήστης μπορεί:

- Να εγγραφεί (να δημιουργήσει λογαριασμό) στην εφαρμογή, πληκτρολογώντας στοιχεία όπως: ονοματεπώνυμο, email, κωδικό και επιβεβαίωση κωδικού.
- Να συνδέεται στην εφαρμογή, πληκτρολογώντας σωστά το email και τον προσωπικό του κωδικό.
- Να διατηρεί ένα προφίλ, το οποίο μπορεί και να διαχειρίζεται. Μπορεί δηλαδή, να ενημερώνει στοιχεία όπως: ονοματεπώνυμο, email, πόλη, διεύθυνση, τηλέφωνο κλπ.
- Να αλλάζει τον προσωπικό του κωδικό, όσες φορές το επιθυμεί.
- Κάθε χρήστης που εγγράφεται (δημιουργεί λογαριασμό) στην εφαρμογή, έχει «εξ' ορισμού» ρόλο Student. Μόνο ο διαχειριστής (Admin) της εφαρμογής μπορεί να αλλάξει τον ρόλο ενός χρήστη.

2) Δυνατότητες που έχουν να κάνουν με χρήστες που έχουν ρόλο Admin. Ένας Admin μπορεί:

- Να διαχειρίζεται τους λογαριασμούς των χρηστών της εφαρμογής. Συγκεκριμένα μπορεί να προσθέτει νέους χρήστες, να αλλάζει το ρόλο ενός εγγεγραμμένου χρήστη, να αλλάζει τον κωδικό ενός εγγεγραμμένου χρήστη και να διαγράφει τον λογαριασμό ενός εγγεγραμμένου χρήστη.

- Να ενημερώνεται για τα στατιστικά της εφαρμογής. Τα στατιστικά αυτά έχουν να κάνουν με: εγγεγραμμένους χρήστες ανά ρόλο, εγγεγραμμένους χρήστες ανά μήνα, μαθήματα που έχουν δημιουργηθεί ανά καθηγητή, αρχεία που έχουν μεταφορτωθεί στην πλατφόρμα και άλλα.

3) Δυνατότητες που έχουν να κάνουν με χρήστες που έχουν ρόλο Teacher. Ένας Teacher μπορεί:

- Να δημιουργεί όσα μαθήματα επιθυμεί.
- Να διαχειρίζεται τα μαθήματα που έχει δημιουργήσει. Συγκεκριμένα, μπορεί να ενημερώνει, να διαγράφει και να κάνει ορατό / μη ορατό ένα μάθημα.
- Να δημιουργεί και να διαχειρίζεται όσες ενότητες θέλει, μέσα σε ένα μάθημα.
- Να "ανεβάζει" και να διαχειρίζεται το εκπαιδευτικό υλικό (αρχεία και συνδέσμους) του κάθε μαθήματος.
- Να απεγγράφει έναν ή και περισσότερους μαθητές από τα μαθήματά του.
- Να επικοινωνεί με τους μαθητές που είναι εγγεγραμμένοι στα μαθήματά του, μέσω αποστολής μηνυμάτων και αρχείων.

4) Δυνατότητες που έχουν να κάνουν με χρήστες που έχουν ρόλο Student. Ένας Student μπορεί:

- να εγγράφεται σε όσα μαθήματα θέλει (του ίδιου ή και διαφορετικών εκπαιδευτών).
- Να έχει πρόσβαση στο εκπαιδευτικό υλικό που περιέχουν τα μαθήματα.
- Να "κατεβάζει" το εκπαιδευτικό υλικό των μαθημάτων.
- Να επικοινωνεί με τους καθηγητές των μαθημάτων που είναι εγγεγραμμένος, μέσω αποστολής μηνυμάτων και αρχείων.

5.2 Επιπλέον πακέτα (packages) που χρησιμοποιήθηκαν στο Laravel

Το Laravel έρχεται με μια πληθώρα προ-εγκαταστημένων πακέτων, αλλά δίνει και τη δυνατότητα στους προγραμματιστές να "κατεβάσουν" και να εγκαταστήσουν όσα πακέτα επιθυμούν. Υπάρχει ένας πολύ μεγάλος αριθμός πακέτων Laravel, που είναι διαθέσιμα στο διαδίκτυο (για παράδειγμα το αποθετήριο Packagist.org).

Τα επιπλέον πακέτα, που εγκαταστάθηκαν και χρησιμοποιήθηκαν στην εφαρμογή, παρουσιάζονται στον πίνακα 8.

Laravel Breeze	Το Laravel Breeze παρέχει μια απλή υλοποίηση όλων των δυνατοτήτων ελέγχου ταυτότητας του Laravel, όπως σύνδεση, εγγραφή, επαναφορά κωδικού πρόσβασης, επαλήθευση email και επιβεβαίωση κωδικού πρόσβασης. Χρησιμοποιεί (από προεπιλογή) τα φύλλα στυλ Tailwind CSS, αντί για το Bootstrap. Υποστηρίζει επίσης, ομάδες, πολλαπλούς χρήστες, ρόλους χρηστών, ιδιοκτησίες αντικειμένων και είναι συμβατό με το σύστημα πολιτικών του Laravel (laravel.com - Starter Kits: Laravel Breeze 2021).
Laravel-Breadcrumbs	Το Laravel Breadcrumbs είναι ένα πακέτο του Dave James Miller που κάνει την προσθήκη της πλοήγησης breadcrumb σε μια εφαρμογή απλή και εύκολη. Το πακέτο είναι εξαιρετικά προσαρμόσιμο και έχει πλήρη τεκμηρίωση και οδηγίες εγκατάστασης (packagist.org - Laravel Breadcrumbs 2021).
Bootstrap	Το Bootstrap είναι ένα πλαίσιο για την υλοποίηση της γραφικής διασύνδεσης χρήστη. Περιλαμβάνει έτοιμα εργαλεία και βοηθητικές κλάσεις "γραμμένα" σε HTML, CSS και JavaScript.
Laravel Messenger	Το Laravel Messenger είναι ένα πακέτο, που δημιουργήθηκε από τον Chris Gmyr και μας επιτρέπει να προσθέσουμε ένα πλήρες σύστημα ανταλλαγής μηνυμάτων χρήστη στην εφαρμογή μας. Μερικές από τις δυνατότητες που προσφέρει είναι: αποστολή πολλαπλών μηνυμάτων, λήψη μηνυμάτων όλων των χρηστών ή συγκεκριμένου χρήστη, εύκολη διαχείριση του αριθμού των μη αναγνωσμένων μηνυμάτων και άλλα πολλά (github.com - Laravel Messenger 2021).
Pusher	Το Pusher είναι μια φιλοξενούμενη υπηρεσία που καθιστά εξαιρετικά εύκολη την προσθήκη δεδομένων σε πραγματικό

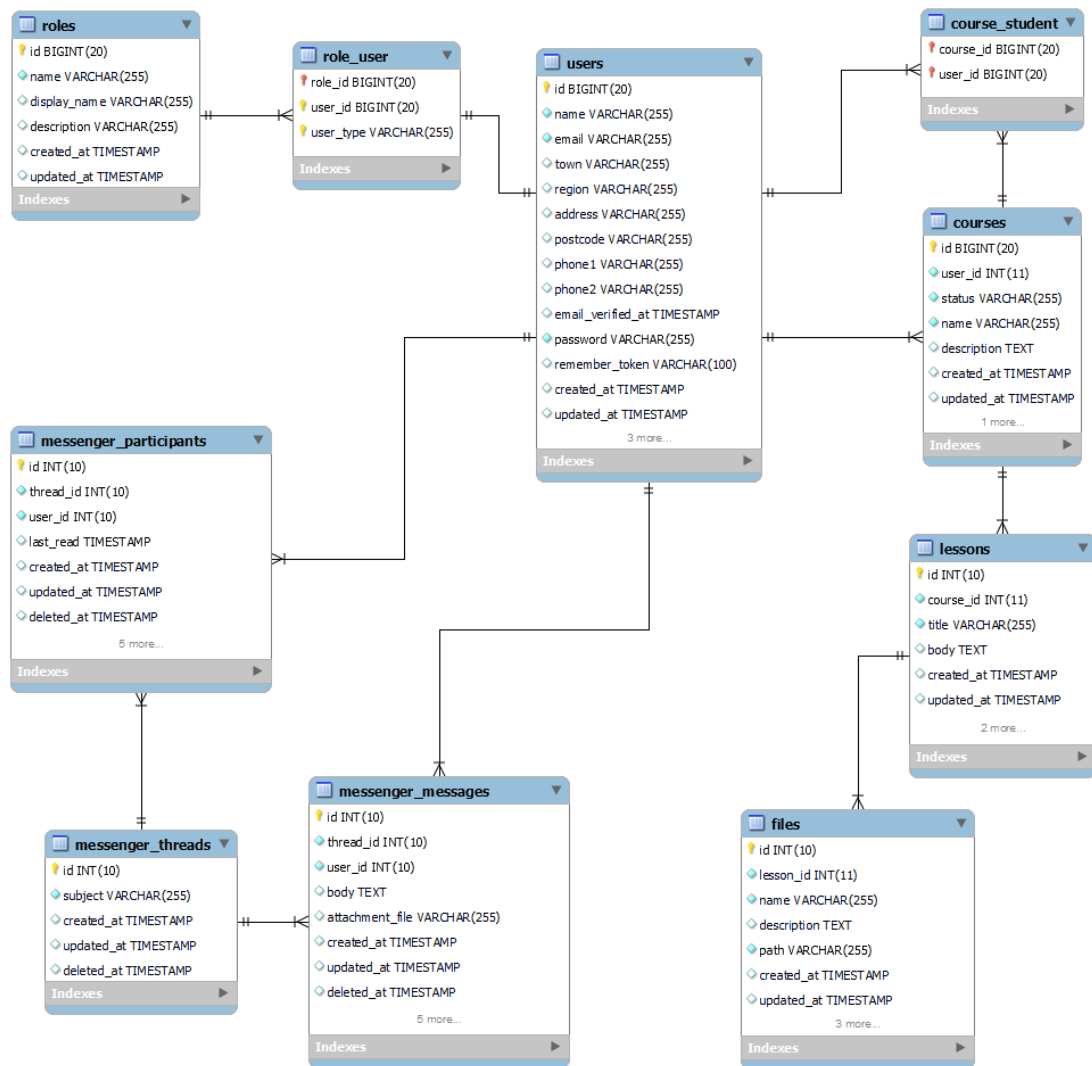
	<p>χρόνο (real-time) σε εφαρμογές ιστού. Λειτουργεί ως ενδιάμεσο στρώμα (layer) μεταξύ των διακομιστών και των εφαρμογών. Διατηρεί μόνιμες συνδέσεις με τις εφαρμογές - μέσω WebSocket - έτσι ώστε μόλις οι διακομιστές έχουν νέα δεδομένα που θέλουν να προωθήσουν στις εφαρμογές, να μπορούν να το κάνουν άμεσα. Προσφέρει βιβλιοθήκες για ενσωμάτωση σε όλα τα πλαίσια PHP και όχι μόνο (pusher-community.github.io - What is Pusher 2021).</p>
--	--

Πίνακας 8: Επιπλέον πακέτα Laravel, που χρησιμοποιήθηκαν στην εφαρμογή

5.3 Βάση δεδομένων

Στην εφαρμογή, όπως φαίνεται και από το διάγραμμα οντοτήτων - συσχετίσεων της εικόνας 43, έχουν δημιουργηθεί οι παρακάτω πίνακες:

- users
- roles
- role_user
- courses
- lessons
- files
- course_student
- messenger_threads
- messenger_messages
- messenger_participants



Εικόνα 43: Διάγραμμα οντοτήτων - συσχετίσεων της βάσης δεδομένων

Στους πίνακες users, roles και στον ενδιάμεσο πίνακα role_user αποθηκεύονται στοιχεία που έχουν να κάνουν με τις λειτουργίες της αυθεντικοποίησης (authentication) και της εξουσιοδότησης (authorization) της εφαρμογής. Επίσης, αποθηκεύονται τα στοιχεία των προφίλ των χρηστών. Στους πίνακες courses, lessons, και files αποθηκεύονται στοιχεία που έχουν να κάνουν με τα μαθήματα, τις ενότητες μαθήματος και τα αρχεία και τους συνδέσμους που δημιουργούν και διαχειρίζονται οι χρήστες της εφαρμογής. Στους πίνακες messenger_threads, messenger_messages και messenger_participants αποθηκεύονται τα στοιχεία που είναι απαραίτητα για την επικοινωνία των χρηστών, δηλαδή τα στοιχεία των μηνυμάτων και των αρχείων που ανταλλάσσουν οι χρήστες της εφαρμογής. Τέλος, στον ενδιάμεσο πίνακα

course_student αποθηκεύονται τα στοιχεία που είναι απαραίτητα για την εγγραφή και απεγγραφή των μαθητών από τα μαθήματα της εφαρμογής.

Το Laravel, μέσα από το ενσωματωμένο εργαλείο Migration, μας παρέχει τη δυνατότητα να δημιουργήσουμε και να διαχειριστούμε τους πίνακες της βάσης δεδομένων, εύκολα και γρήγορα, χρησιμοποιώντας απλές εντολές μέσω κονσόλας (Artisan CLI). Ας δούμε ενδεικτικά τον πίνακα courses.

Πληκτρολογώντας την εντολή: `php artisan make:migration create_courses_table`, δημιουργήθηκε το php αρχείο «2021_07_21_164728_create_courses_table.php».

```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateCoursesTable extends Migration
{
    public function up()
    {
        Schema::create('courses', function (Blueprint $table) {
            $table->id();
            $table->integer('user_id');
            $table->string('status');
            $table->string('name');
            $table->text('description')->nullable();
            $table->timestamps();
        });
    }

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

Εικόνα 44: Η δομή του migration αρχείου του πίνακα «courses»

Στην εικόνα 44, βλέπουμε ότι στην κλάση CreateCoursesTable του αρχείου, χρησιμοποιώντας την μέθοδο create(), καθορίσαμε για τον πίνακα courses τα πεδία: id, user_id, status, name, και description. Επίσης με την συνάρτηση nullable(), δηλώσαμε ότι το πεδίο description μπορεί να μείνει κενό (null).

Στη συνέχεια, πληκτρολογώντας την εντολή: `php artisan migrate`, δημιουργήθηκε και προστέθηκε στη βάση δεδομένων της εφαρμογής μας ο πίνακας `courses` (Εικόνα 45).

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή
1	id 	bigint(20)		UNSIGNED	Όχι	Καμία
2	user_id	int(11)			Όχι	Καμία
3	status	varchar(255)	utf8mb4_unicode_ci		Όχι	Καμία
4	name	varchar(255)	utf8mb4_unicode_ci		Όχι	Καμία
5	description	text	utf8mb4_unicode_ci		Ναι	NULL
6	created_at	timestamp			Ναι	NULL
7	updated_at	timestamp			Ναι	NULL

Εικόνα 45: Τα πεδία του πίνακα `courses`

Στην εικόνα 45, βλέπουμε ότι το πεδίο `id` είναι το πρωτεύων κλειδί του πίνακα και είναι μοναδικό για κάθε μάθημα. Το πεδίο `user_id` είναι το `id` του χρήστη (με ρόλο `Teacher`) που δημιουργεί το μάθημα. Το πεδίο `status` παίρνει την τιμή "1", όταν ένα μάθημα είναι ορατό και την τιμή "0", όταν δεν είναι ορατό. Τα πεδία `name` και `description` χρησιμοποιούνται για τον τίτλο και την περιγραφή του μαθήματος αντίστοιχα. Τέλος, τα πεδία `created_at` και `updated_at` έχουν προστεθεί αυτόματα στον πίνακα και είναι τύπου `timestamp`. Αναφέρονται στη χρονική στιγμή (ημερομηνία και ώρα) που κάποιος χρήστης (με ρόλο `Teacher`) προσθέσει ένα μάθημα στην εφαρμογή και στη χρονική στιγμή που έγινε η τελευταία ανανέωση των στοιχείων αυτού του μαθήματος αντίστοιχα.

5.4 Models

Το Laravel υλοποιεί την τεχνική της Αντικειμενοστραφούς Χαρτογράφησης (ORM) μέσα από το εξαιρετικό εργαλείο που διαθέτει, το Eloquent. Κάθε μοντέλο που δημιουργήσαμε για την εφαρμογή μας, αντιστοιχεί και σε έναν πίνακα στη βάση δεδομένων. Επίσης, στα μοντέλα δηλώθηκαν και οι σχέσεις μεταξύ των πινάκων που αναπαριστούν. Τα μοντέλα που υλοποιήθηκαν στην εφαρμογή αφορούν βασικές οντότητες της βάσης δεδομένων και είναι τα παρακάτω:

- User (αντιστοιχεί στον πίνακα users)
- Role (αντιστοιχεί στον πίνακα roles)
- Course (αντιστοιχεί στον πίνακα courses)
- Lesson (αντιστοιχεί στον πίνακα lessons)
- File (αντιστοιχεί στον πίνακα files)

Ας δούμε ενδεικτικά το μοντέλο Lessons.

```
<?php

namespace App\Models;

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

class Lesson extends Model
{
    use HasFactory;
    protected $fillable = [
        'title',
        'body',
        'course_id',
    ];

    public function course(){
        return $this->belongsTo(Course::class);
    }

    public function files(){
        return $this->hasMany(File::class);
    }
}
```

Εικόνα 46: Το μοντέλο Lesson

Στην εικόνα 46, παρατηρούμε ότι στο συγκεκριμένο μοντέλο δηλώσαμε τα πεδία του πίνακα lessons που μπορούν να συμπληρωθούν μαζικά. Αυτό γίνεται με την ιδιότητα \$fillable. Επίσης, βλέπουμε ότι δηλώσαμε και τις σχέσεις μεταξύ των πινάκων lessons και courses, αλλά και μεταξύ των πινάκων lessons και files. Συγκεκριμένα, με τη μέθοδο belongsTo() της συνάρτησης course(), δηλώσαμε ότι ο πίνακας courses σχετίζεται με τον πίνακα lessons με σχέση ένα προς πολλά (ένα μάθημα έχει πολλές ενότητες). Με τη μέθοδο hasMany() της συνάρτησης files(), δηλώσαμε ότι ο πίνακας lessons σχετίζεται με τον πίνακα files με σχέση ένα προς πολλά επίσης (μια ενότητα περιέχει πολλά αρχεία και συνδέσμους).

5.5 Routing

Όλες οι διαδρομές (routes) της εφαρμογής ορίζονται στο αρχείο web.php, το οποίο είναι αποθηκευμένο στον κατάλογο routes. Τα routes είναι οι διαδρομές από τις οποίες κάθε χρήστης της εφαρμογής αποκτά πρόσβαση, μέσω HTTP requests, στα views ή και στους controllers.

```
// routes for all users
> Route::group(['middleware' => ['auth']], function() { ...
});

// routes for admin
> Route::group(['middleware' => ['auth', 'role:admin']], function() { ...
});

// routes for teachers
> Route::group(['middleware' => ['auth', 'role:teacher']], function() { ...
});

// routes for students
> Route::group(['middleware' => ['auth', 'role:student']], function() { ...
});

// routes for teachers & students - messaging system
> Route::group(['prefix' => 'messages', 'middleware' => ['auth', 'role:student|teacher']], function () { ...
});
```

Εικόνα 47: Οι ομάδες διαδρομών της εφαρμογής

Το Laravel, παρέχει τη δυνατότητα να ομαδοποιήσουμε τις διαδρομές της εφαρμογής, ώστε να δηλωθούν κοινά χαρακτηριστικά για αυτές. Στις ομάδες αυτές χρησιμοποιήσαμε τον μηχανισμό Middleware, ώστε μόνο πιστοποιημένοι (authenticated) χρήστες και χρήστες με συγκεκριμένο ρόλο στην εφαρμογή, να

μπορούν να χρησιμοποιούν τις διαδρομές αυτών των ομάδων. Το Middleware, ουσιαστικά αποτελεί ένα ενδιάμεσο στάδιο προτού ένα request φτάσει στον controller που προορίζεται. Έτσι, αν για παράδειγμα ένας χρήστης δεν είναι πιστοποιημένος, τον ανακατευθύνουμε (redirect) στην σελίδα σύνδεσης λογαριασμού (login page).

Όπως φαίνεται στην εικόνα 47, στην εφαρμογή έχουμε δημιουργήσει πέντε ομάδες διαδρομών. Συγκεκριμένα, στις διαδρομές της πρώτης ομάδας έχουν πρόσβαση μόνο πιστοποιημένοι χρήστες, όλων των ρόλων (Admin, Teacher & Student). Στην δεύτερη ομάδα δηλώσαμε διαδρομές που αφορούν σελίδες (views), στις οποίες έχουν πρόσβαση μόνο πιστοποιημένοι χρήστες με ρόλο Admin. Στις διαδρομές της τρίτης και της τέταρτης ομάδας έχουν πρόσβαση μόνο πιστοποιημένοι χρήστες με ρόλο Teacher και Student αντίστοιχα. Τέλος, στην πέμπτη ομάδα υπάρχουν διαδρομές που σχετίζονται με το σύστημα επικοινωνίας (messaging system) της εφαρμογής. Σύμφωνα με τις λειτουργίες της εφαρμογής, μόνο πιστοποιημένοι χρήστες με ρόλο Teacher ή Student, μπορούν να επικοινωνήσουν μεταξύ τους στέλνοντας γραπτά μηνύματα ή αρχεία.

Στην εικόνα 49, βλέπουμε αναλυτικά όλες τις διαδρομές, τις μεθόδους HTTP (get, put), τους ελεγκτές και τις συναρτήσεις των ελεγκτών (index, store, show) της ομάδας διαδρομών που αφορά τους Students της εφαρμογής μας. Εδώ να επισημάνουμε ότι κάποιες από τις διαδρομές είναι παραμετροποιημένες (`{courseId}`, `{lessonId}`), ώστε να απομονωθεί το id ενός μαθήματος ή μιας ενότητας, δηλαδή να έχουμε πρόσβαση σε συγκεκριμένη εγγραφή της βάσης δεδομένων.

Τέλος, στην εικόνα 48, παρουσιάζοντα συνοπτικά οι μέθοδοι HTTP, οι διαδρομές και οι συναρτήσεις των ελεγκτών της παραπάνω ομάδας.

```
C:\xampp\htdocs\multiauth>php artisan route:list --compact --path=student
```

Method	URI	Action
GET HEAD	dashboard/students/courses	App\Http\Controllers\StudentsCoursesController@index
GET HEAD	dashboard/students/courses/lessons/{lessonId}	App\Http\Controllers\StudentsLessonsController@show
GET HEAD	dashboard/students/courses/lessons/{lessonId}/files	App\Http\Controllers\StudentsFilesController@index
GET HEAD	dashboard/students/courses/{courses}	App\Http\Controllers\StudentsCoursesController@show
GET HEAD	dashboard/students/enrollment	App\Http\Controllers\EnrollmentController@index
PUT	dashboard/students/enrollment/{courseId}	App\Http\Controllers\EnrollmentController@store

Εικόνα 48: Οι μέθοδοι, οι διαδρομές και οι ελεγκτές της ομάδας διαδρομών για τους Students

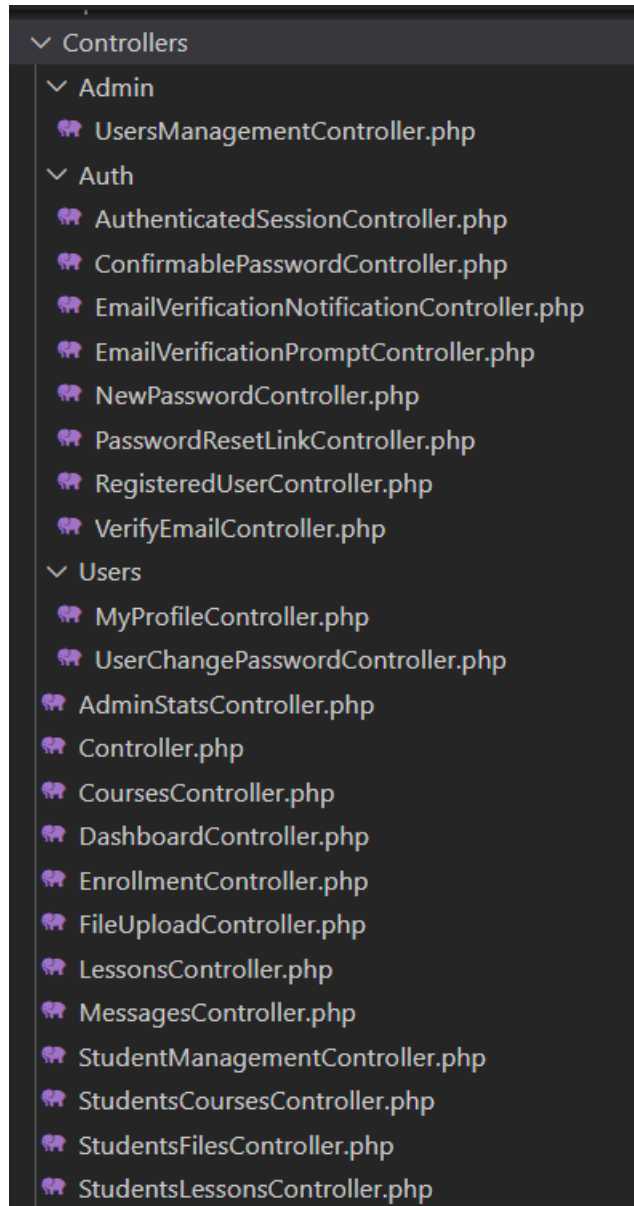
```
// routes for students
Route::group(['middleware' => ['auth', 'role:student']], function() {
    // enroll
    Route::get('/dashboard/students/enrollment', 'App\Http\Controllers\EnrollmentController@index')
        ->name('enrollment.index');
    Route::put('/dashboard/students/enrollment/{courseId}', [\App\Http\Controllers\EnrollmentController::class, 'store'])
        ->name('enrollment.store');
    // courses
    Route::get('/dashboard/students/courses', [StudentsCoursesController::class, 'index'])
        ->name('studentscourses.index');
    Route::get('/dashboard/students/courses/{courses}', [StudentsCoursesController::class, 'show'])
        ->name('studentscourses.show');
    // lessons
    Route::get('/dashboard/students/courses/lessons/{lessonId}', [StudentsLessonsController::class, 'show'])
        ->name('studentslessons.show');
    // files
    Route::get('/dashboard/students/courses/lessons/{lessonId}/files', [StudentsFilesController::class, 'index'])
        ->name('studentsfiles.index');
});
```

Εικόνα 49: Ομάδα διαδρομών μόνο για πιστοποιημένους χρήστες με ρόλο Student

5.6 Controllers

Το Laravel χρησιμοποιεί τους ελεγκτές (controllers) για την διαχείριση των προβολών (views), των διαδρομών (routes), και των αιτημάτων (requests) της εφαρμογής. Επίσης, οι controllers είναι αυτοί που αναλαμβάνουν την επικοινωνία με τη βάση δεδομένων. Συγκεκριμένα, αναλαμβάνουν τον έλεγχο, την επεξεργασία, την ανάκτηση και την αποθήκευση των στοιχείων της εφαρμογής μας στη βάση δεδομένων.

Από προεπιλογή, όλοι οι ελεγκτές που δημιουργήθηκαν αποθηκεύτηκαν στον κατάλογο `app/Http/Controllers`. Οι ελεγκτές που χρησιμοποιήθηκαν στην εφαρμογή μας παρουσιάζονται στην παρακάτω εικόνα (Εικόνα 50):



Εικόνα 50: Οι ελεγκτές (controllers) της εφαρμογής

Να επισημάνουμε εδώ ότι, οι ελεγκτές που είναι αποθηκευμένοι στον κατάλογο `Auth`, παρέχονται έτοιμοι από το `Laravel`.

Ένας από τους controllers που υλοποιήθηκαν στην εφαρμογή είναι ο `EnrollmentController` (Εικόνα 000), ο οποίος αναλαμβάνει μια πολύ σημαντική

λειτουργία της εφαρμογής. Αυτή είναι η εγγραφή συγκεκριμένου μαθητή σε μάθημα που επιλέγει. Οι λειτουργίες που επιτελεί, είναι:

- Εμφάνιση όλων των διαθέσιμων προς εγγραφή μαθημάτων
- Εγγραφή μαθητή σε μάθημα

```
class EnrollmentController extends Controller
{
    public function index()
    {
        $user_id = Auth::user()->id;

        $registeredCourses = DB::table('course_student')
            ->where('course_student.user_id', $user_id)
            ->distinct()
            ->get();

        $availableCourses = DB::table('courses')
            ->join('users', 'users.id', '=', 'courses.user_id')
            ->select('users.name AS teacher_name', 'courses.id',
                'courses.status', 'courses.name AS course_name')
            ->where('courses.status', '=', '1')
            ->paginate(5);

        return view('enrollment.index',
            compact('availableCourses', 'registeredCourses'))
            ->with('i', (request()->input('page', 1) - 1) * 5);
    }

    public function store(Request $request, $courseId)
    {
        $user_id = $request->user()->id;

        DB::table('course_student')->insert([
            'course_id' => $courseId,
            'user_id' => $user_id
        ]);

        return redirect()->route('enrollment.index')
            ->with('message', 'Η εγγραφή στο μάθημα έγινε επιτυχώς');
    }
}
```

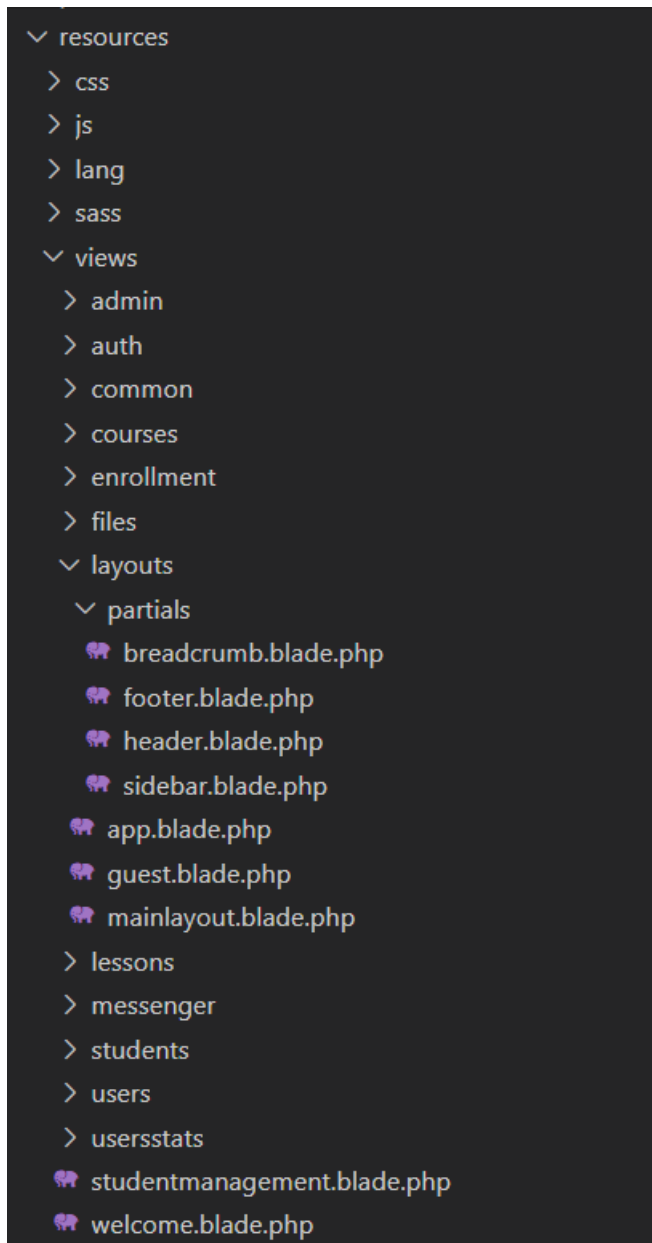
Εικόνα 51: Οι συναρτήσεις του ελεγκτή EnrollmentController

Όπως βλέπουμε στην εικόνα 51, η συνάρτηση `index()` αναλαμβάνει να ανακτήσει και να "στείλει" στην σελίδα `index.blade.php` (μέθοδος `view`) τα στοιχεία όλων των ορατών μαθημάτων που έχουν δημιουργηθεί από όλους τους καθηγητές της εφαρμογής. Συγκεκριμένα, χρησιμοποιώντας το εργαλείο Query Builder του Laravel, δημιουργήσαμε δύο ερωτήματα προς τη βάση δεδομένων. Με το πρώτο ερώτημα ανακτούμε όλα τα μαθήματα στα οποία είναι ήδη εγγεγραμμένος ο μαθητής (πίνακας `course_student`). Αυτό το ερώτημα είναι απαραίτητο, γιατί στη σελίδα εμφανίζονται αγνά (απενεργοποιημένα) τα μαθήματα που είναι ήδη εγγεγραμμένος. Με το δεύτερο ερώτημα ανακτούμε όλα τα ορατά μαθήματα (`status=1`) που έχουν δημιουργηθεί από όλους τους καθηγητές, καθώς και τα ονόματα των καθηγητών αυτών. Εδώ να επισημάνουμε ότι, τα μη ορατά μαθήματα της εφαρμογής δεν εμφανίζονται στους μαθητές, ακόμα και αν είναι εγγεγραμμένοι σε αυτά.

Τέλος, η συνάρτηση `store()`, αναλαμβάνει να εισάγει (μέθοδος `insert`) μία νέα εγγραφή με το `id` του συγκεκριμένου μαθητή και το `id` του μαθήματος που επιλέγει να εγγραφεί (η συνάρτηση το δέχεται ως είσοδο), στον πίνακα `course_student`. Ο πίνακας `course_student` είναι ένας ενδιάμεσος πίνακας, που συνδέει τους πίνακες `users` και `courses` στην βάση δεδομένων της εφαρμογής και επιτελεί κεντρικό ρόλο στην λειτουργία της εγγραφής μαθητή σε μάθημα. Μόλις αποθηκευτεί η εγγραφή, ο μαθητής ανακατευθύνεται στην ίδια σελίδα και η εφαρμογή μας εμφανίζει το μήνυμα "Η εγγραφή στο μάθημα έγινε επιτυχώς".

5.7 Views

Όπως αναφέραμε και στο κεφάλαιο 3, το Laravel χρησιμοποιεί το Blade ως μηχανή προτύπων για τη δημιουργία προβολών (`views`) σε μια διαδικτυακή εφαρμογή. Όλες οι προβολές (αρχεία `blade.php`) της εφαρμογής οργανώθηκαν σε υποκαταλόγους, ανάλογα με τους ρόλους των χρηστών και τις λειτουργίες που επιτελούν. Οι υποκατάλογοι αυτοί αποθηκευτήκαν στον κατάλογο `resources/views` (Εικόνα 52).



Εικόνα 52: Οργάνωση των αρχείων προβολών (views) της εφαρμογής

Δεδομένου ότι οι περισσότερες εφαρμογές ιστού διατηρούν την ίδια διάταξη (layout) σε όλες τις σελίδες ή ομάδες σελίδων, ορίσαμε για όλες τις σελίδες της εφαρμογής μας, μια κύρια διάταξη προβολής. Αυτή είναι στο αρχείο `mainlayout.blade.php`, που είναι αποθηκευμένο στην τοποθεσία `resources/views/layout` (εικόνα 52).

Οι υπόλοιπες προβολές της εφαρμογής ή περιέχονται ή επεκτείνουν την κύρια διάταξη προβολής. Συγκεκριμένα, στην εικόνα 53 βλέπουμε ότι, χρησιμοποιώντας την εντολή `@include()`, συμπεριλαμβάνουμε στην κύρια διάταξη τα κομμάτια `header`,

sidebar και footer. Αυτά αποτελούν ξεχωριστά αρχεία blade.php, που είναι αποθηκευμένα στον υποκατάλογο partials, όπως φαίνεται και στην εικόνα 52.

```
<body class="hold-transition sidebar-mini">
<!-- Site wrapper -->
<div class="wrapper">
  <!-- Navbar -->
  @include('layouts.partials.header')
  <!-- /.navbar -->

  <!-- Main Sidebar Container -->
  @include('layouts.partials.sidebar')
  <!-- Content Wrapper. Contains page content -->
  <div class="content-wrapper">
    <!-- Content Header (Page header) -->

    <!-- Main content -->
    <section class="content">
      <!-- Default box -->
      @yield('content')
    </section>
    <!-- /.content -->
  </div>
  @include('layouts.partials.footer')
  <!-- /.content-wrapper -->
  <!-- up button -->
  <button onclick="topFunction()" id="myBtn" title="Κορυφή">
    <i class="fa fa-arrow-up" aria-hidden="true"></i>
  </button>
</div>
<!-- ./wrapper -->
```

Εικόνα 53: Το αρχείο mainlayout.blade.php, που περιέχει την κύρια διάταξη της εφαρμογής

Υπάρχουν όμως και προβολές (οι περισσότερες στην εφαρμογή), που επεκτείνουν την κύρια διάταξη. Αυτό το πετύχαμε χρησιμοποιώντας την εντολή @ extends, όπως φαίνεται και στην εικόνα 54.

Τέλος, όπως έχουμε ήδη αναφέρει, για την ανάπτυξη της διεπαφής χρήστη (UI) της εφαρμογής μας έχουν χρησιμοποιηθεί πολλά από τα πρότυπα και τις κλάσεις που προσφέρει το framework Bootstrap. Στην εφαρμογή μας, όπως βλέπουμε και από την εικόνα 54, χρησιμοποιήσαμε πάρα πολλές κλάσεις όπως card, card-body, container,

table table-bordered, thead-light, alert alert-success, nav-link, nav-icon, btn btn-success, form-inline, form-group, και άλλες πολλές, τις οποίες παρέχει και διαθέτει το Bootstrap. Δημιουργήσαμε έτσι ένα καλαίσθητο και αποκριτικό (responsive) αποτέλεσμα, χωρίς να σπαταλήσουμε χρόνο, γράφοντας από το μηδέν αρχεία CSS και JavaScript.

```

@extends('layouts.mainlayout')

@section('content')
<section class="content-header">
  <div class="container-fluid">
    <div class="row mb-2">
      <div class="col-sm-12">
        <ol class="breadcrumb float-sm-right">
          <li class="breadcrumb-item">
            {{ Breadcrumbs::render('studentscourses.index') }}
          </li>
        </ol>
      </div>
    </div>
  </section>
  <div class="card">
    <div class="card-header">
      <h3 class="card-title">Μαθήματα</h3>
      <div class="card-tools"></div>
    </div>
    <div class="card-body">
      @if (session('status'))
      <div class="alert alert-success" role="alert">
        {{ session('status') }}
      </div>
      @endif
      @include('common.alert')
      <table class="table">
        <thead class="thead-light">
          <tr>
            <th width="50px">Α/Α</th>
            <th>Τίτλος</th>
            <th>Περιγραφή</th>
          </tr>
        </thead>
        <tbody>
          @foreach($courses as $course)
          <tr>
            <td>{{ ++$i }}</td>
            <td>
              <a class="text-info" href="{{ route('studentscourses.show', $course->id) }}">{{ $course->name }}</a>
            </td>
            <td>{{ $course->description }}</td>
          </tr>
          @endforeach
        </tbody>
      </table>
      <div class="d-flex justify-content-center">
        {!! $courses->links() !!}
      </div>
    </div>
    <div class="card-footer"></div>
  </div>
@endsection

```

Εικόνα 54: Παράδειγμα αρχείου προβολής της εφαρμογής

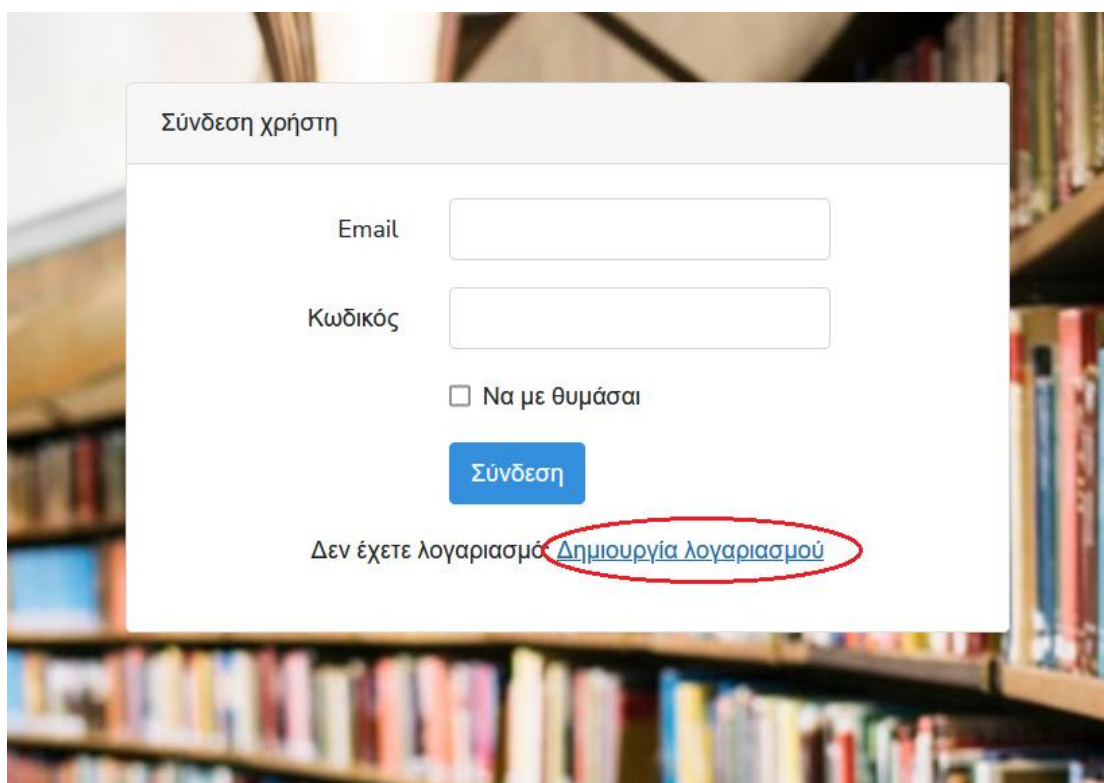
6. Εικονογραφημένος οδηγός της εφαρμογής

6.1 Λειτουργίες που αφορούν όλους τους χρήστες της εφαρμογής

1) Δημιουργία λογαριασμού.

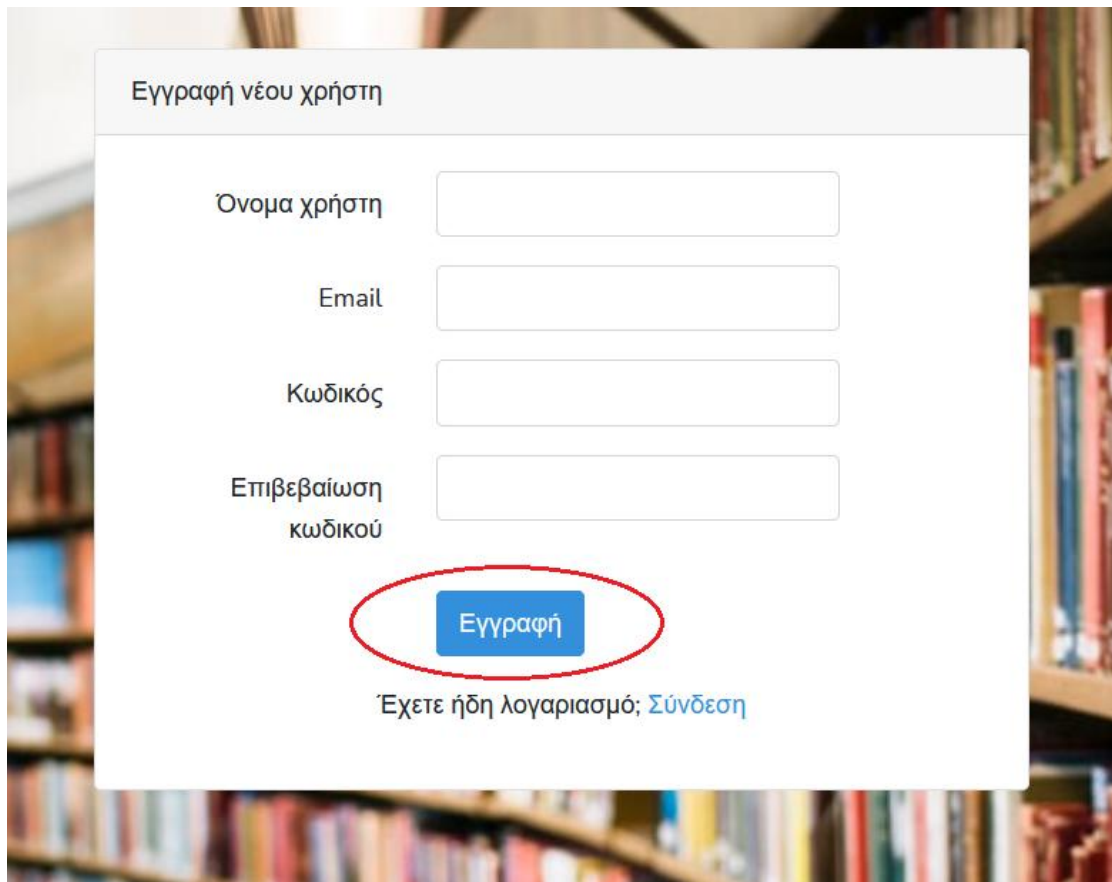
Κάθε χρήστης που εγγράφετε (δημιουργεί λογαριασμό) στην εφαρμογή, έχει «εξ' ορισμού» ρόλο εκπαιδευόμενου (Student). Για να δημιουργήσετε έναν λογαριασμό νέου χρήστη, θα πρέπει να κάνετε τα εξής:

Κάντε κλικ στην αρχική σελίδα, στο «Δημιουργία λογαριασμού» (Εικόνα 55).



Εικόνα 55: Δημιουργία λογαριασμού

Θα ανοίξει η φόρμα «Εγγραφή νέου χρήστη». Συμπληρώστε εκεί τα πεδία Όνομα χρήστη, Email, Κωδικός και Επιβεβαίωση κωδικού και πατήστε «Εγγραφή» (Εικόνα 56).

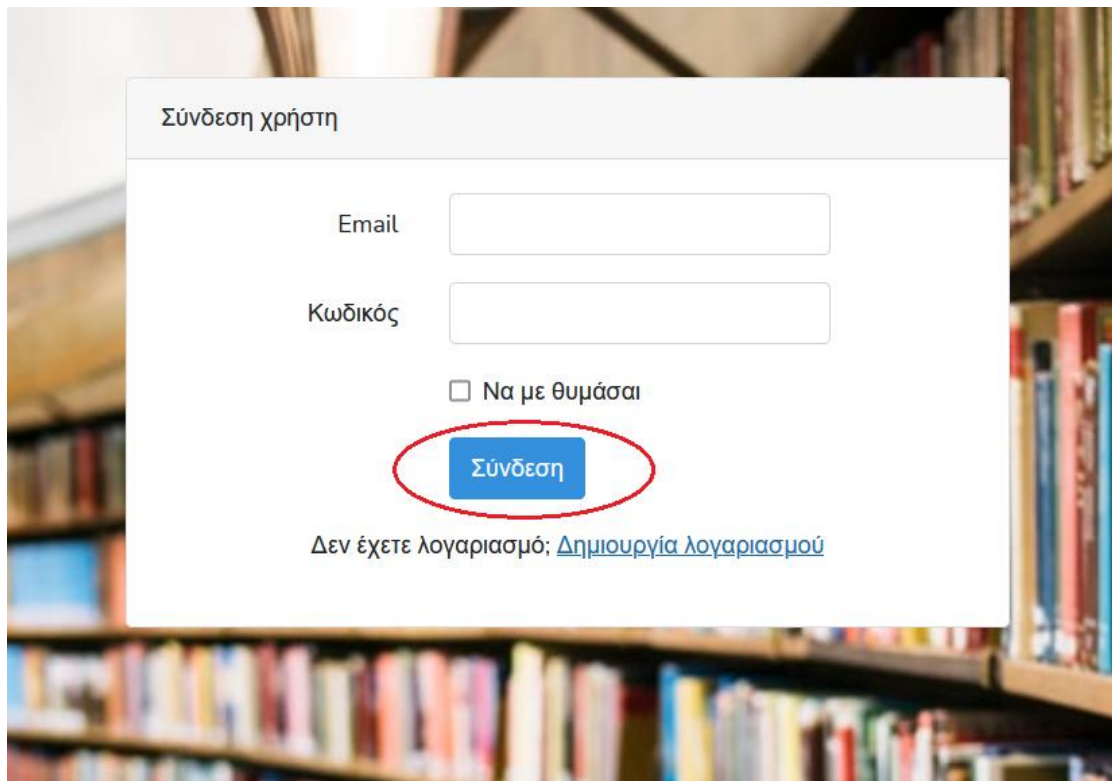


Εικόνα 56: Εγγραφή νέου χρήστη

Θα έχετε δημιουργήσει έτσι, έναν νέο χρήστη ο οποίος θα έχει ρόλο εκπαιδευόμενου. Μόνο ο διαχειριστής (Admin) της εφαρμογής μπορεί να αλλάξει τον ρόλο ενός χρήστη.

2) Σύνδεση χρήστη

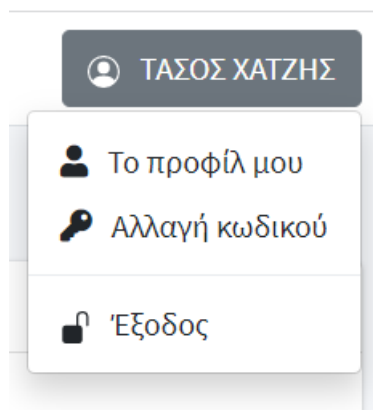
Αφού αποκτήσετε λογαριασμό στην πλατφόρμα, θα πρέπει για να συνδεθείτε στον λογαριασμό σας, να πληκτρολογείτε στην αρχική σελίδα το Email και τον προσωπικό σας Κωδικός και να πατάτε «Σύνδεση» (Εικόνα 57)



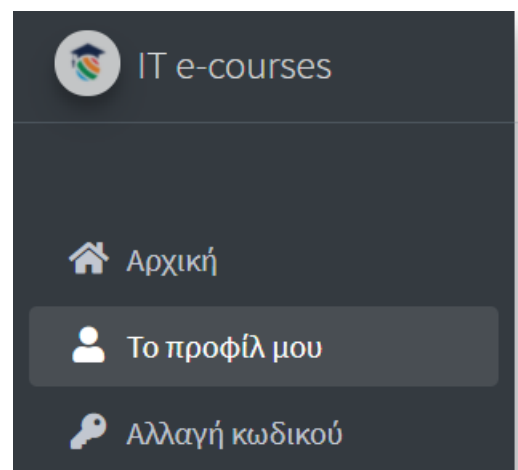
Εικόνα 57: Σύνδεση χρήστη

Με τον τρόπο αυτό θα έχετε πρόσβαση στις αντίστοιχες λειτουργίες της πλατφόρμας, ανάλογα με τον ρόλο σας.

3) Αλλαγή προφίλ χρήστη



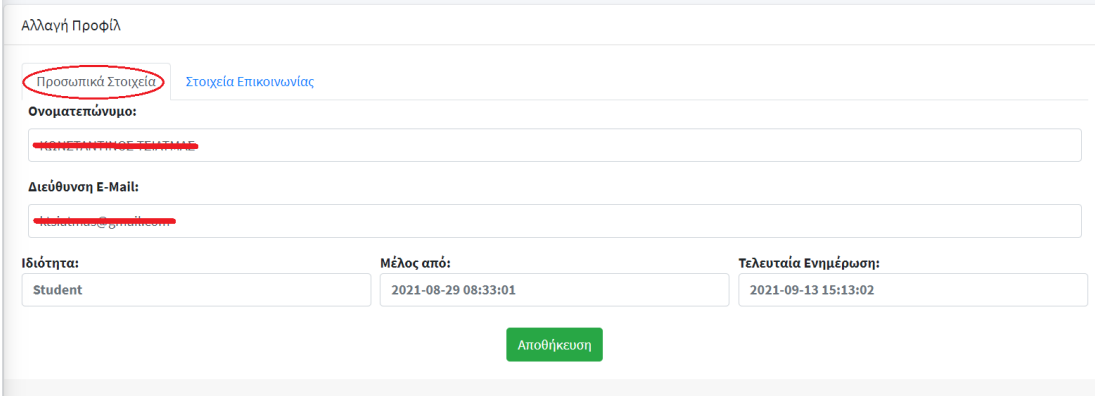
Εικόνα 58: Αλλαγή προφίλ από το πτυσσόμενο μενού



Εικόνα 59: Αλλαγή προφίλ από το κάθετο αριστερό μενού

Ενώ είστε συνδεδεμένος στον λογαριασμό σας, επιλέξτε «Το προφίλ μου» στο πτυσσόμενο μενού πάνω δεξιά (Εικόνα 58) ή στην αριστερή στήλη του κάθετου μενού επιλογών (Εικόνα 59) και αλλάξτε τα προσωπικά στοιχεία και τα στοιχεία επικοινωνίας του προφίλ σας.

Κάντε κλικ στην καρτέλλα «Προσωπικά Στοιχεία» για να αλλάξετε στοιχεία όπως το όνομα χρήστη (Ονοματεπώνυμο) και τη διεύθυνσή email (τα στοιχεία αυτά είναι υποχρεωτικά). Όταν ολοκληρώσετε τις αλλαγές κάντε κλικ στο κουμπί με την ένδειξη «Αποθήκευση» (Εικόνα 60).



Αλλαγή Προφίλ

Προσωπικά Στοιχεία Στοιχεία Επικοινωνίας

Όνοματεπώνυμο:
[Redacted]

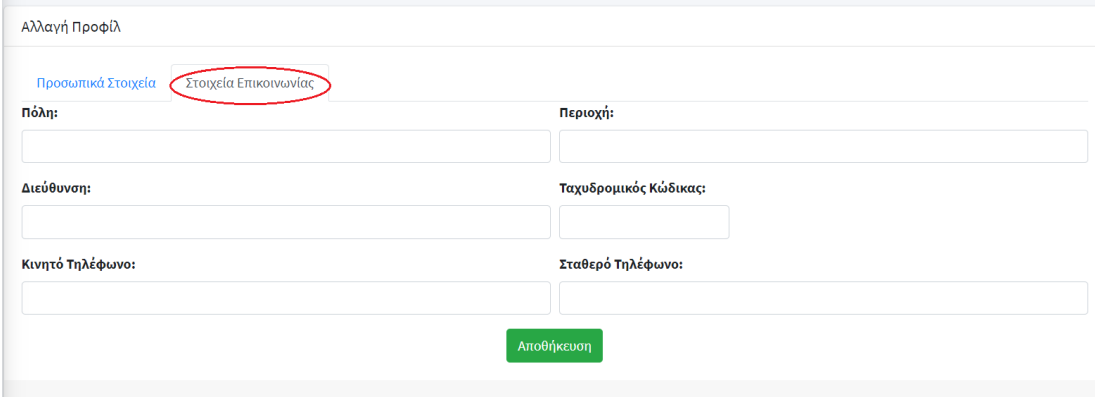
Διεύθυνση E-Mail:
[Redacted]

Ιδιότητα: Student Μέλος από: 2021-08-29 08:33:01 Τελευταία Ενημέρωση: 2021-09-13 15:13:02

Αποθήκευση

Εικόνα 60: Αλλαγή προσωπικών στοιχείων προφίλ

Κάντε κλικ στην καρτέλα «Στοιχεία Επικοινωνίας» για να αλλάξετε στοιχεία όπως Πόλη, Περιοχή, Διεύθυνση, Ταχυδρομικός Κώδικας, Κινητό Τηλέφωνο και Σταθερό Τηλέφωνο (τα στοιχεία αυτά είναι προαιρετικά). Όταν ολοκληρώσετε τις αλλαγές κάντε κλικ στο κουμπί με την ένδειξη «Αποθήκευση» (Εικόνα 61).



Αλλαγή Προφίλ

Προσωπικά Στοιχεία Στοιχεία Επικοινωνίας

Πόλη: Περιοχή:

Διεύθυνση: Ταχυδρομικός Κώδικας:

Κινητό Τηλέφωνο: Σταθερό Τηλέφωνο:

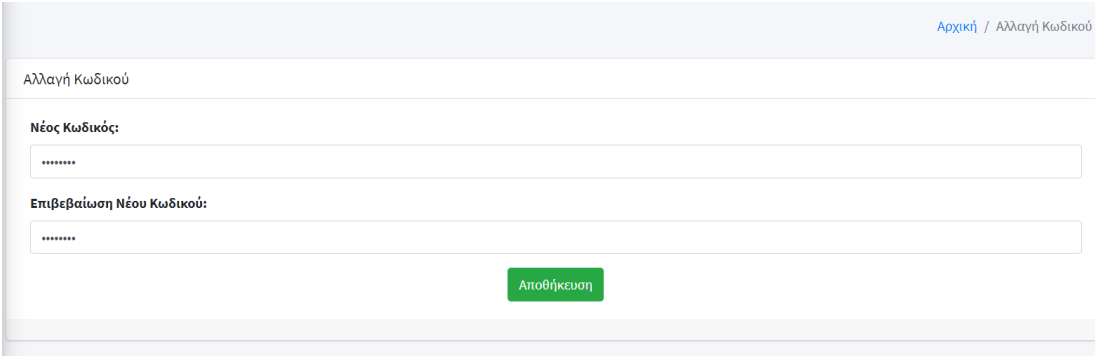
Αποθήκευση

Εικόνα 61: Αλλαγή στοιχείων επικοινωνίας προφίλ

4) Αλλαγή κωδικού χρήστη

Ο κωδικός (password) αποτελεί το στοιχείο εκείνο το οποίο χρησιμοποιείτε προκειμένου να αποκτήσετε πρόσβαση στην πλατφόρμα. Εάν επιθυμείτε να το αλλάξετε τότε μπορείτε να ακολουθήσετε τα παρακάτω βήματα (Εικόνα 62).

1. Επιλέξτε «Αλλαγή κωδικού» στην αριστερή στήλη του κάθετου μενού επιλογών (Εικόνα 59) ή στο πτυσσόμενο μενού πάνω δεξιά (Εικόνα 58).
2. Εισάγετε τον νέο κωδικό καθώς και την επιβεβαίωση του.
3. Η διαδικασία ολοκληρώνεται κάνοντας κλικ στο κουμπί με την ένδειξη «Αποθήκευση».



Εικόνα 62: Αλλαγή κωδικού χρήστη

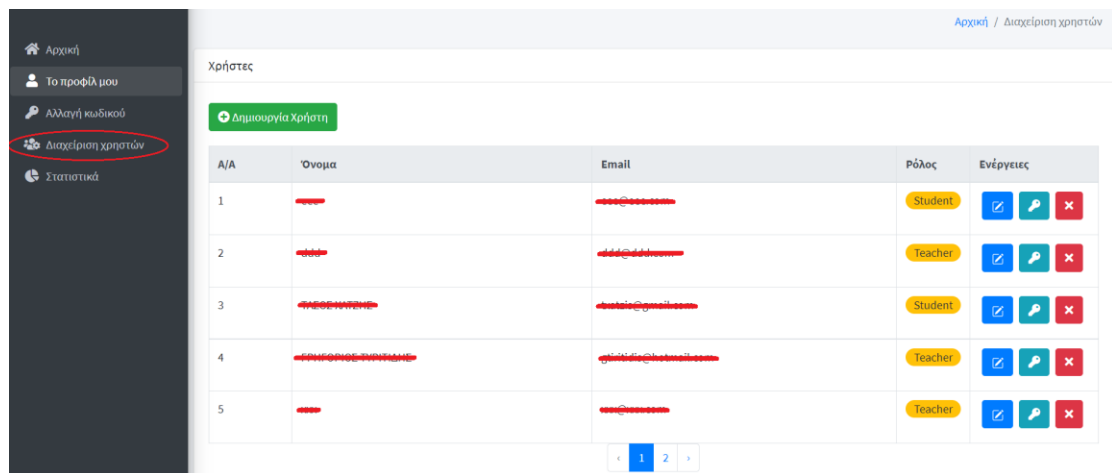
5) Έξοδο από την εφαρμογή

Κάντε κλικ στην επιλογή «Έξοδος» στο πτυσσόμενο μενού πάνω δεξιά (Εικόνα 58).

6.2 Λειτουργίες που αφορούν χρήστες με ρόλο Διαχειριστή

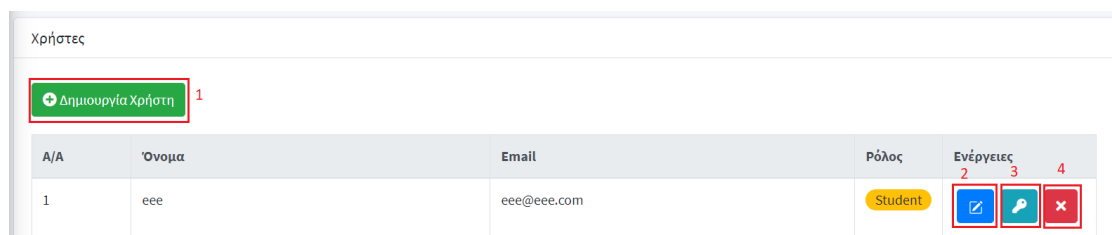
1) Διαχείριση λογαριασμών χρηστών

Επιλέξτε «Διαχείριση χρηστών» στην αριστερή στήλη, στο κάθετο μενού επιλογών, για να διαχειριστείτε τους εγγεγραμμένους χρήστες της πλατφόρμας (Εικόνα 63)



Εικόνα 63: Διαχείριση χρηστών

Συγκεκριμένα, κάνοντας κλικ στο αντίστοιχο κουμπί, μπορείτε να κάνετε τα εξής (Εικόνα 64):

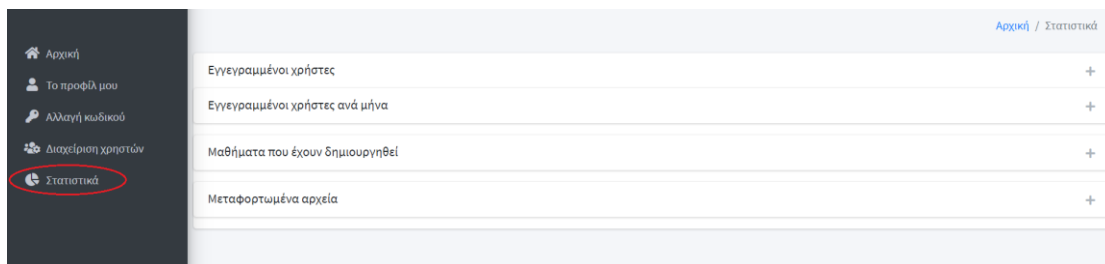


Εικόνα 64: Λειτουργίες διαχείρισης χρηστών

- Να προσθέσετε έναν νέο χρήστη.
- Να αλλάξετε τον ρόλο ενός εγγεγραμμένου χρήστη.
- Να αλλάξετε τον κωδικό ενός εγγεγραμμένου χρήστη.
- Να διαγράψετε τον λογαριασμό ενός εγγεγραμμένου χρήστη.

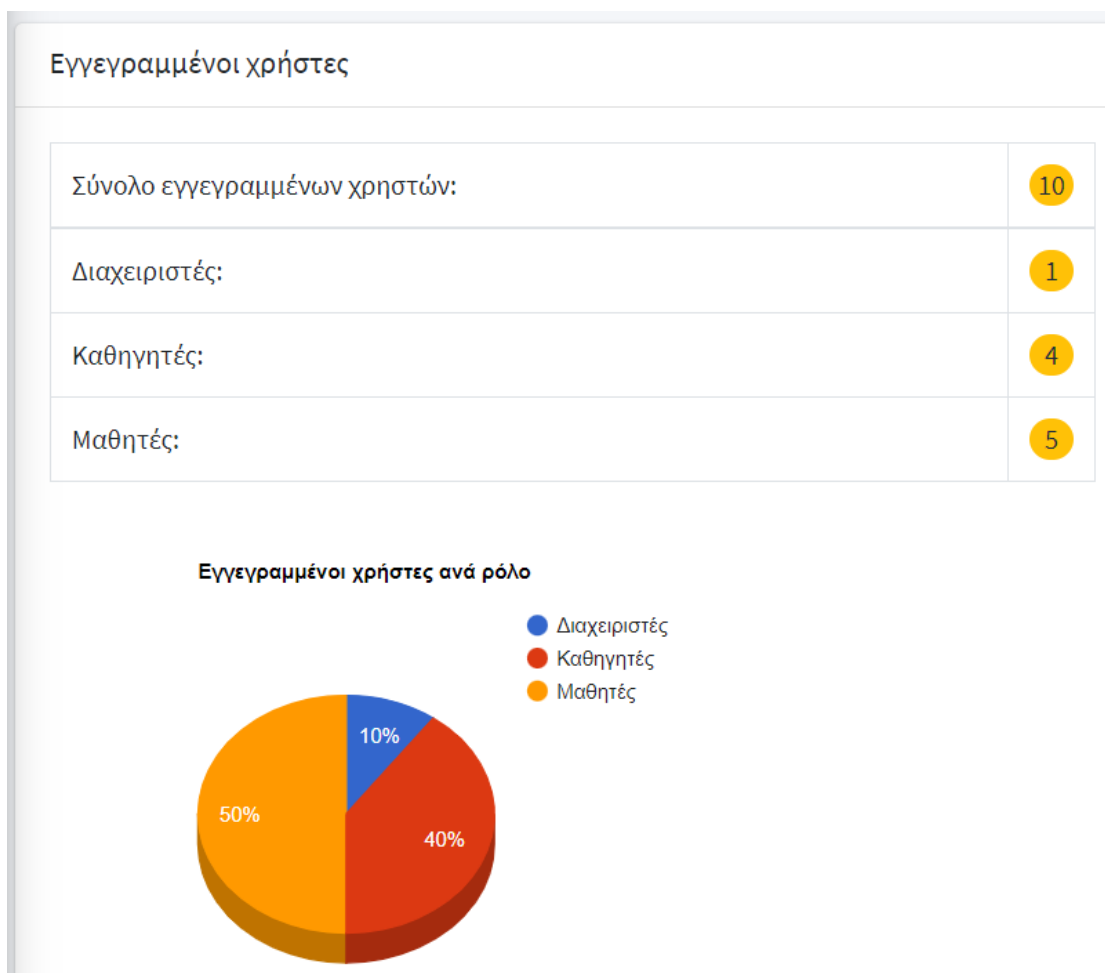
2) Στατιστικά

Επιλέξτε «Στατιστικά» στην αριστερή στήλη, στο κάθετο μενού επιλογών, προκειμένου να εμφανίσετε τα στατιστικά στοιχεία της πλατφόρμας (Εικόνα 65). Εκεί μπορεί να δείτε (και σε γραφική αναπαράσταση) τα παρακάτω στατιστικά στοιχεία:



Εικόνα 65: Στατιστικά στοιχεία

- Εγγεγραμμένοι χρήστες (Εικόνα 66).



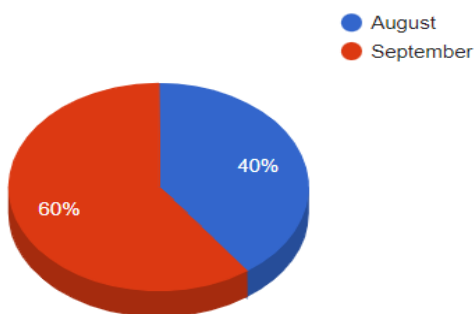
Εικόνα 66: Εγγεγραμμένοι χρήστες

- Εγγεγραμμένοι χρήστες ανά μήνα (Εικόνα 67).

Εγγεγραμμένοι χρήστες ανά μήνα

Τον μήνα August ,οι εγγεγραμμένοι χρήστες είναι:	4
Τον μήνα September ,οι εγγεγραμμένοι χρήστες είναι:	6

Εγγεγραμμένοι χρήστες ανά μήνα



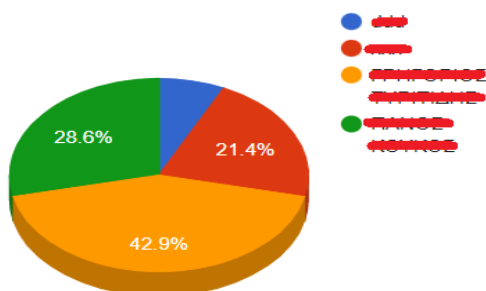
Εικόνα 67: Εγγεγραμμένοι χρήστες ανά μήνα

- Μαθήματα που έχουν δημιουργηθεί (συνολικά και ανά καθηγητή) (Εικόνα 68).

Μαθήματα που έχουν δημιουργηθεί

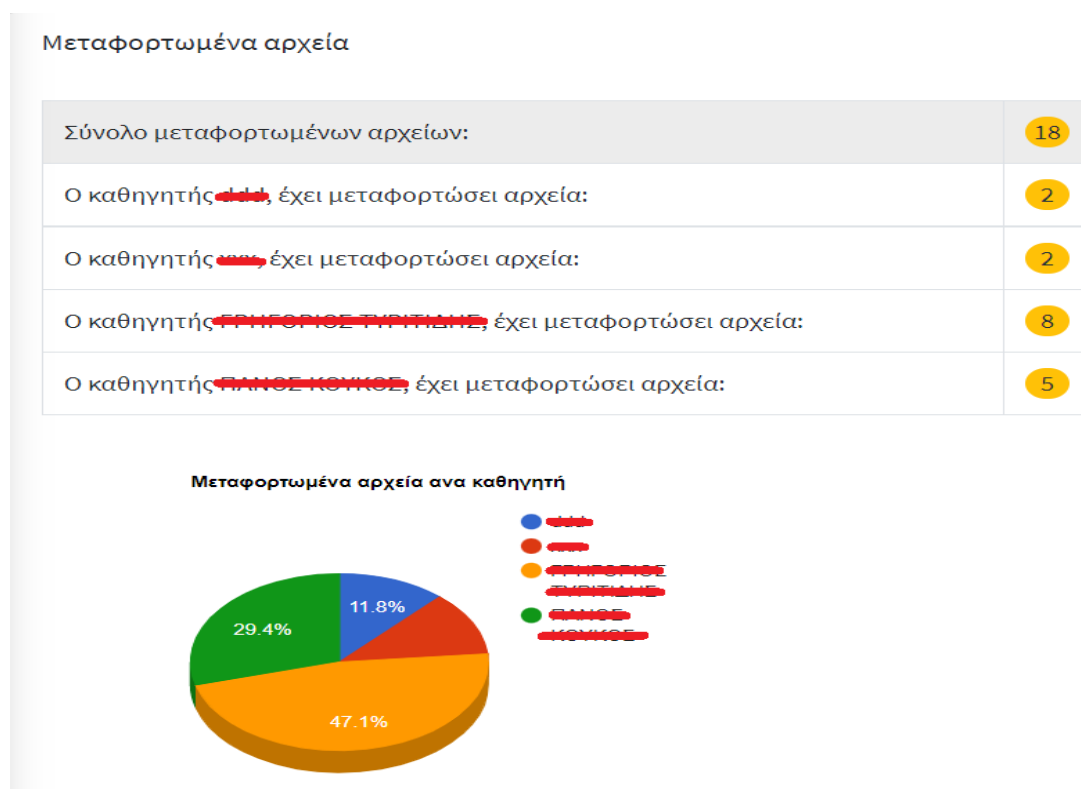
Σύνολο μαθημάτων:	14
Ο καθηγητής ΑΔΑΔ , έχει δημιουργήσει μαθήματα:	1
Ο καθηγητής ΑΝΩΣ έχει δημιουργήσει μαθήματα:	3
Ο καθηγητής ΕΡΜΕΦΟΡΙΟΣ ΤΥΡΤΙΔΗΣ , έχει δημιουργήσει μαθήματα:	6
Ο καθηγητής ΠΑΝΟΣ ΚΟΥΚΟΣ , έχει δημιουργήσει μαθήματα:	4

Μαθήματα ανά καθηγητή



Εικόνα 68: Μαθήματα που έχουν δημιουργηθεί

- Μεταφορτωμένα αρχεία (συνολικά και ανά καθηγητή) (Εικόνα 69).



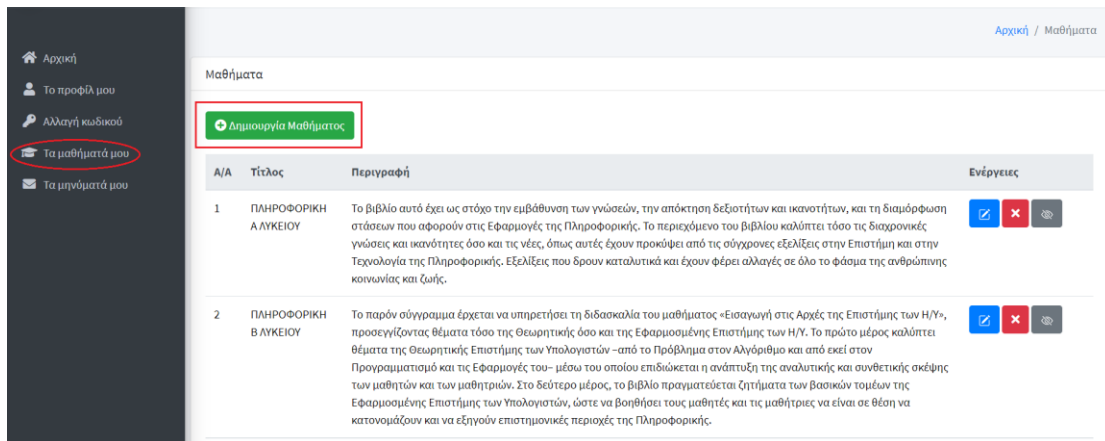
Εικόνα 69: Μεταφορτωμένα αρχεία

6.3 Λειτουργίες που αφορούν χρήστες με ρόλο Εκπαιδευτή

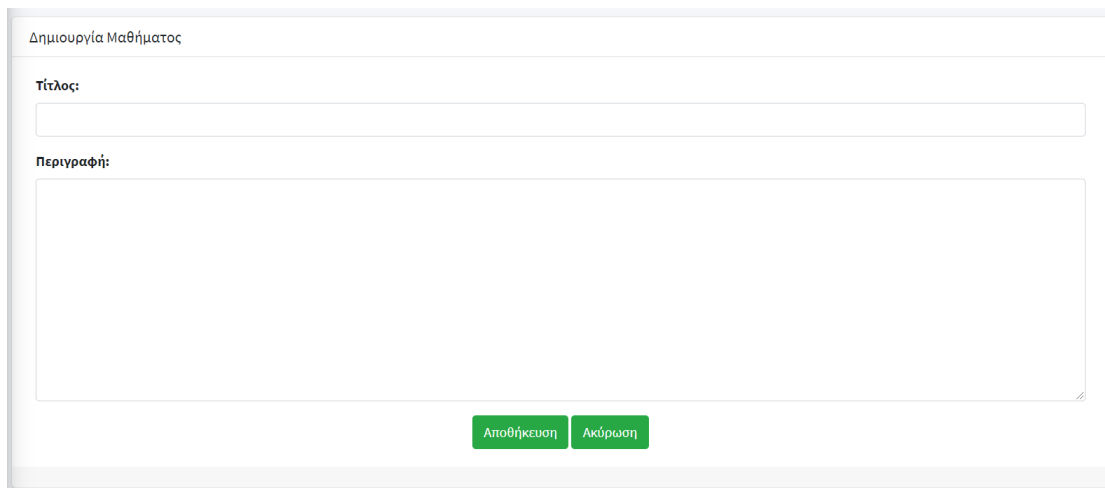
1) Δημιουργία μαθήματος

Η Δημιουργία Μαθήματος είναι η πιο σημαντική ενέργεια του χρήστη – Εκπαιδευτή στην πλατφόρμα. Ειδικότερα, για να δημιουργήσετε ένα νέο μάθημα, επιλέξτε «Τα μαθήματά μου» στην αριστερή στήλη στο κάθετο μενού επιλογών και στη συνέχεια πατήστε το κουμπί «Δημιουργία Μαθήματος» πάνω αριστερά (Εικόνα 70).

Στην φόρμα «Δημιουργία Μαθήματος» (Εικόνα 71), συμπληρώστε τα στοιχεία του νέου μαθήματος. Πληκτρολογήστε έναν "Τίτλο" για το μάθημα (είναι υποχρεωτικό) και μια "Περιγραφή" (είναι προαιρετικό). Τέλος κάντε κλικ στο κουμπί «Αποθήκευση» για να δημιουργηθεί το μάθημά σας. Μόλις ολοκληρωθεί η δημιουργία του, μπορείτε να μπειτε σε αυτό και να προσθέσετε ενότητες. Να σημειωθεί ότι κάθε εκπαιδευτής μπορεί να δημιουργεί στην πλατφόρμα όσα μαθήματα επιθυμεί χωρίς να υπάρχει ανώτατο όριο.



Εικόνα 70: Δημιουργία μαθήματος

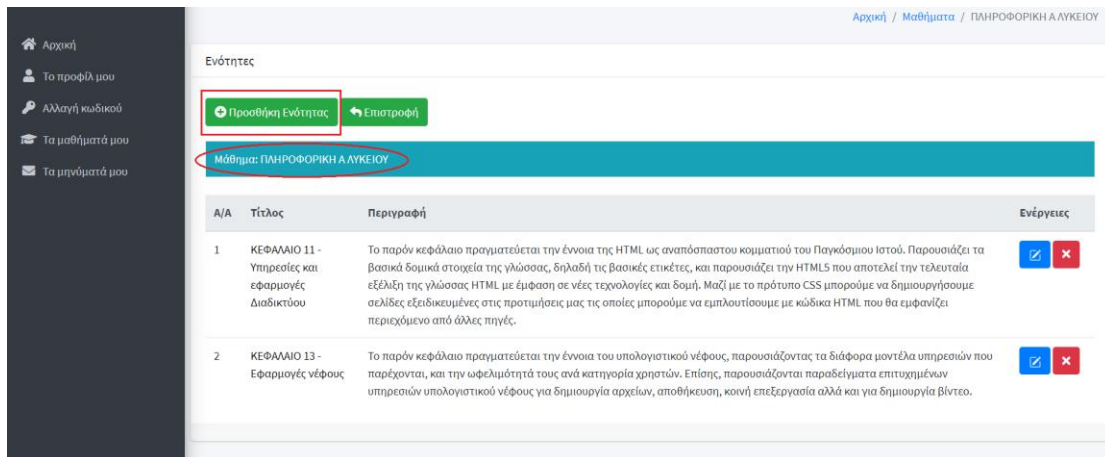


Εικόνα 71: Αποθήκευση μαθήματος

2) Δημιουργία ενότητας μαθήματος

Αφού μπειτε στο μάθημα για το οποίο θέλετε να δημιουργήσετε ενότητες, πατήστε το κουμπί «Προσθήκη Ενότητας» πάνω αριστερά (Εικόνα 72).

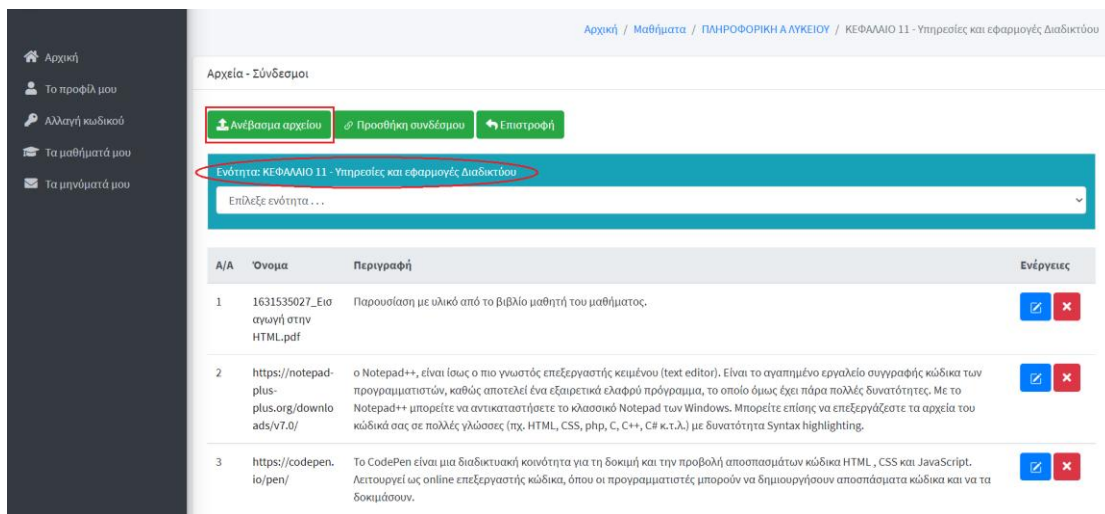
Στην φόρμα «Προσθήκη Ενότητας» συμπληρώστε τα στοιχεία της νέας ενότητας. Πληκτρολογήστε έναν "Τίτλο" για την ενότητα (είναι υποχρεωτικό) και μια "Περιγραφή" (είναι προαιρετικό). Τέλος κάντε κλικ στο κουμπί «Αποθήκευση» για να δημιουργηθεί η ενότητα. Μόλις ολοκληρωθεί η δημιουργία της, μπορείτε να μπειτε σε αυτή και να ανεβάσετε και να προσθέσετε όσα αρχεία και συνδέσμους επιθυμείτε.



Εικόνα 72: Προσθήκη ενότητας

3) Ανέβασμα αρχείου

Αφού μπειτε στην ενότητα για την οποία θέλετε να ανεβάσετε αρχεία, πατήστε το κουμπί «Ανέβασμα αρχείου» πάνω αριστερά (Εικόνα 73).

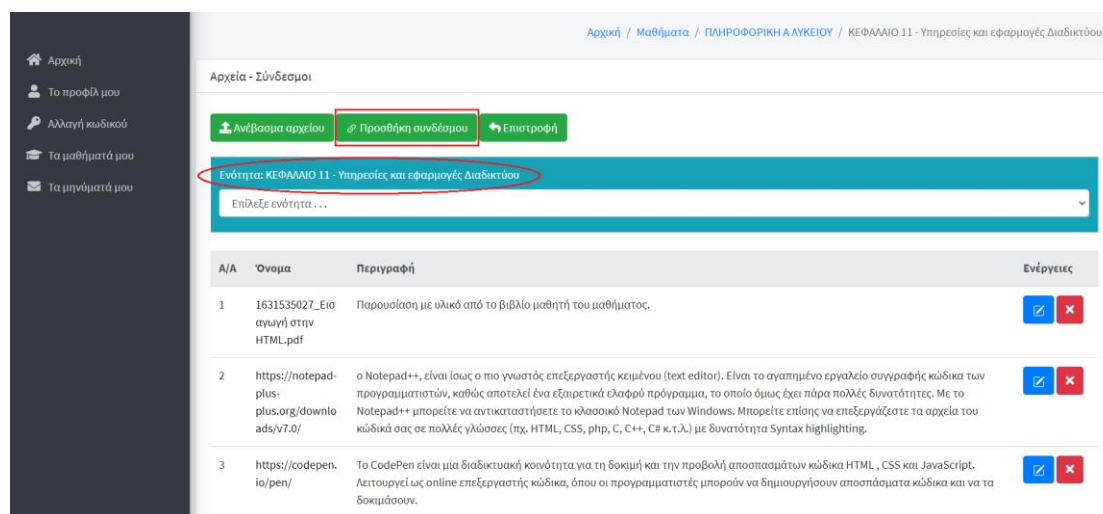


Εικόνα 73: Ανέβασμα αρχείου

Στην φόρμα «Ανέβασμα αρχείου» κάντε κλικ στο "Choose file..." ή στο "Browse" για να επιλέξετε το αρχείο το οποίο θέλετε να ανεβάσετε (είναι υποχρεωτικό) και πληκτρολογήστε μία Περιγραφή (είναι προαιρετικό). Για να ολοκληρωθεί η διαδικασία κάντε κλικ στο κουμπί «Ανέβασμα».

4) Προσθήκη συνδέσμου

Αφού μπείτε στην ενότητα για την οποία θέλετε να προσθέσετε συνδέσμους, πατήστε το κουμπί «Προσθήκη συνδέσμου» πάνω στο κέντρο (Εικόνα 74).



Εικόνα 74: Προσθήκη συνδέσμου

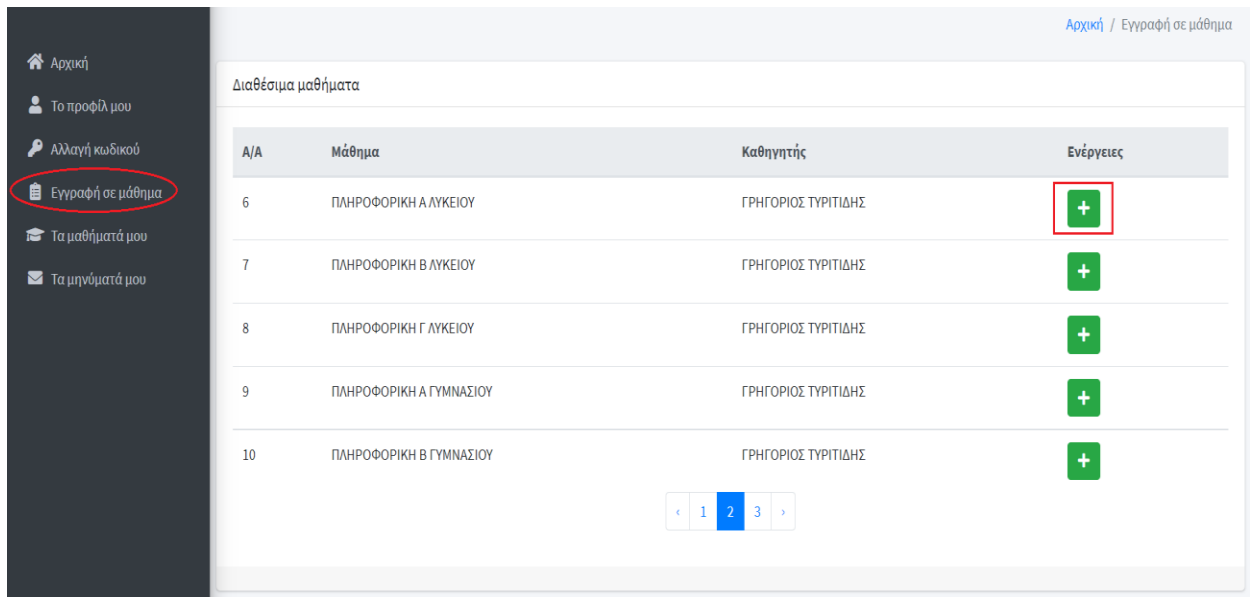
Στην φόρμα «Προσθήκη συνδέσμου» πληκτρολογήστε έναν έγκυρο URL σύνδεσμο (είναι υποχρεωτικό) και μια Περιγραφή (είναι προαιρετικό) και πατήστε στο κουμπί «Προσθήκη» για να ολοκληρωθεί η διαδικασία.

6.4 Λειτουργίες που αφορούν χρήστες με ρόλο Εκπαιδευόμενου

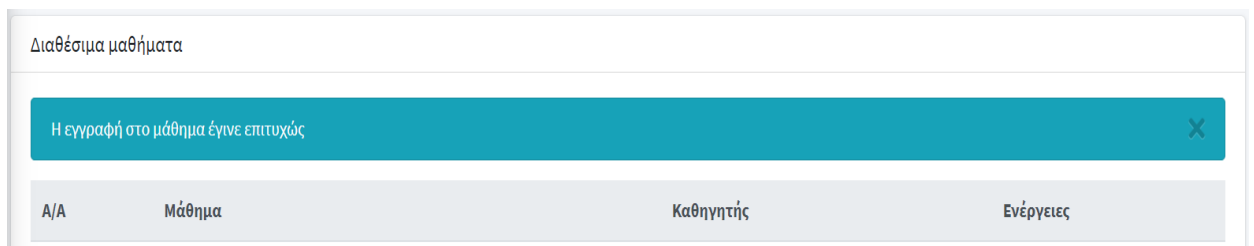
1) Εγγραφή σε μάθημα

Επιλέγοντας «Εγγραφή σε μάθημα» στην αριστερή στήλη στο κάθετο μενού επιλογών, θα μεταφερθείτε στη λίστα μαθημάτων στην οποία μπορείτε να επιλέξετε ποια από τα διαθέσιμα μαθήματα θέλετε να συμπεριλάβετε στην προσωπική σας λίστα παρακολούθησης. Στα διαθέσιμα μαθήματα βλέπετε τον τίτλο του μαθήματος και το όνομα του καθηγητή που έχει δημιουργήσει το μάθημα (Εικόνα 75).

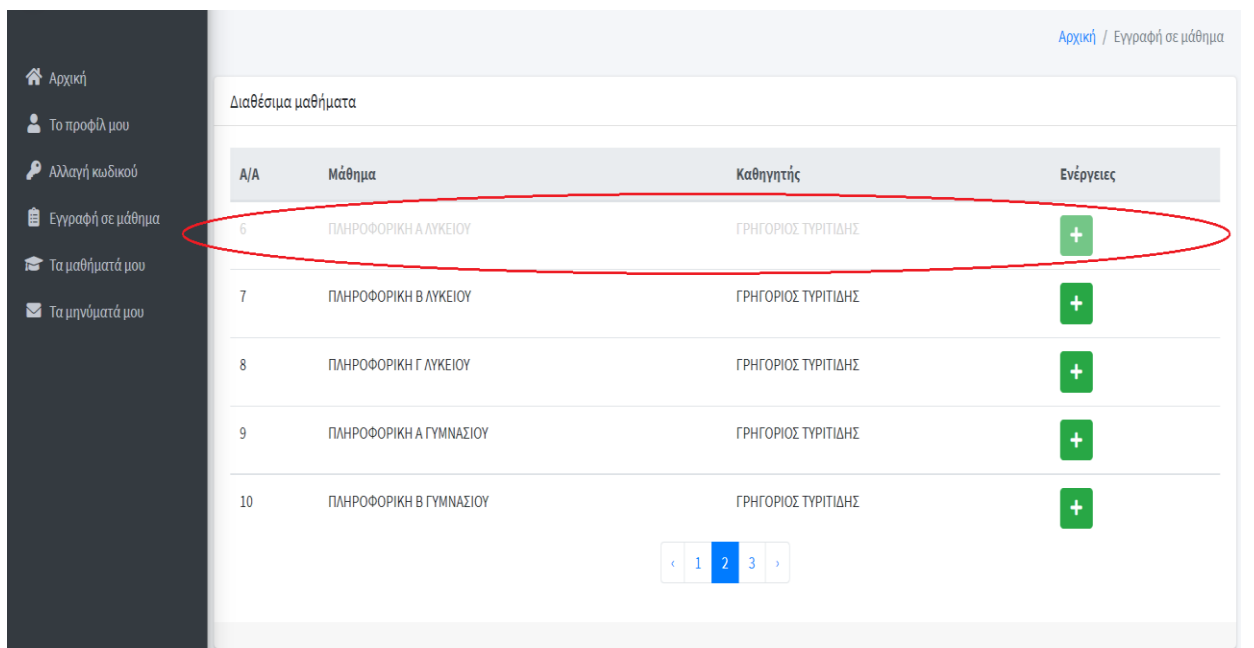
Για να εγγραφείτε σε ένα συγκεκριμένο μάθημα, πρέπει να κάνετε κλικ στο κουμπί με ένδειξη « + ». Αμέσως μετά θα εμφανιστεί ένα μήνυμα επιβεβαίωσης (Εικόνα 76) για την εγγραφή σας στο μάθημα και το συγκεκριμένο μάθημα θα απενεργοποιηθεί (θα φαίνεται αχνά) από την λίστα των διαθέσιμων μαθημάτων (Εικόνα 77).



Εικόνα 75: Εγγραφή σε μάθημα



Εικόνα 76: Μήνυμα επιτυχούς εγγραφής σε μάθημα



Εικόνα 77: Απενεργοποίηση μαθήματος, μετά από επιτυχή εγγραφή

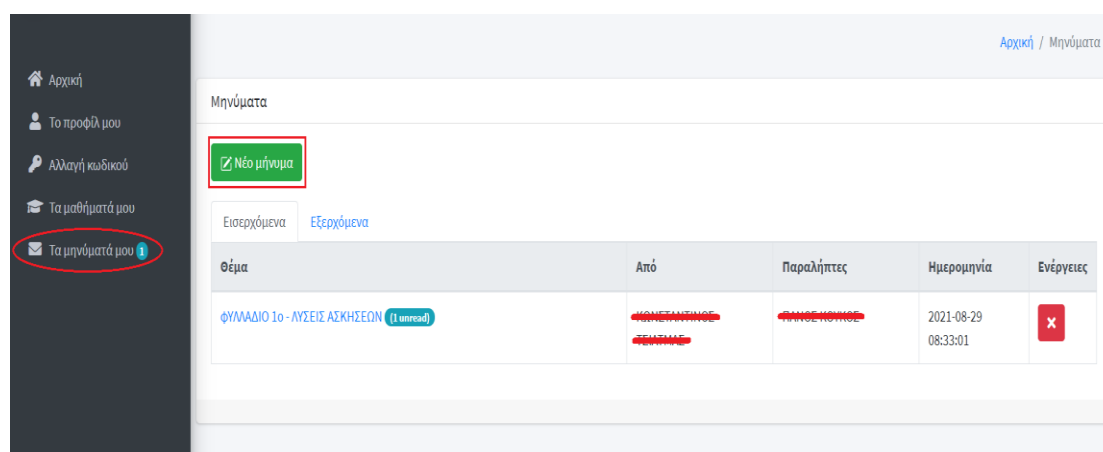
6.5 Λοιπές λειτουργίες της εφαρμογής

6.5.1 Σύστημα επικοινωνίας Εκπαιδευτών και Εκπαιδευόμενων

Το υποσύστημα “Τα μηνύματά μου” παρέχει στον εκπαιδευτή και στον εκπαιδευόμενο τις παρακάτω δυνατότητες:

1) Αποστολή μηνυμάτων σε εγγεγραμμένους μαθητές ενός μαθήματος

Έχοντας συνδεθεί ως καθηγητής στην εφαρμογή, για να στείλετε ένα νέο μήνυμα σε έναν ή περισσότερους εγγεγραμμένους μαθητές ενός μαθήματος σας, επιλέξτε «Τα μηνύματά μου» στην αριστερή στήλη στο κάθετο μενού επιλογών, και πατήστε στο κουμπί «Νέο μήνυμα» (Εικόνα 78).



Εικόνα 78: Αποστολή νέου μηνύματος εκπαιδευτή

Νέο μήνυμα

Μάθημα:
ΠΛΗΡΟΦΟΡΙΚΗ Α ΛΥΚΕΙΟΥ

Προς:
~~ΚΩΝΣΤΑΝΤΙΝΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ~~
~~ΚΩΝΣΤΑΝΤΙΝΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ~~
~~ΚΩΝΣΤΑΝΤΙΝΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ~~

Θέμα:

Μήνυμα:

Συνημμένο αρχείο:
Choose file... Browse

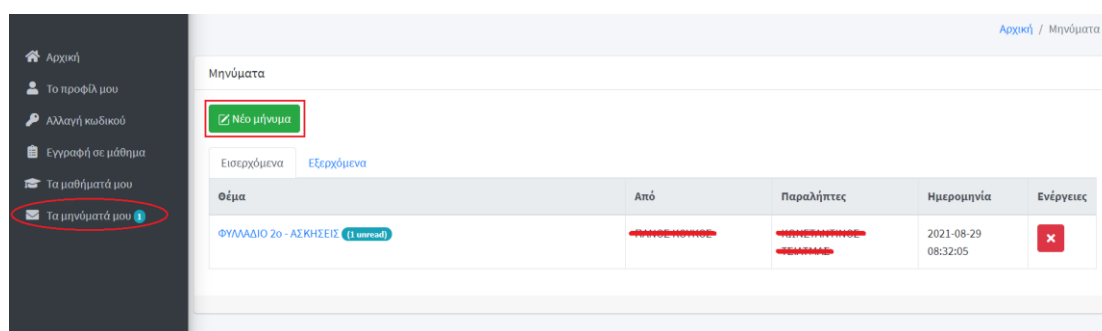
Αποστολή Ακύρωση

Εικόνα 79: Φόρμα αποστολής μηνύματος εκπαιδευτή

Στην φόρμα «Νέο μήνυμα» (Εικόνα 79), επιλέξτε το μάθημα στο οποίο θα γίνει η αποστολή του μηνύματος, τους παραλήπτες (έναν ή περισσότερους), το θέμα του μηνύματος (είναι υποχρεωτικό), το κείμενο του μηνύματος , το συνημμένο αρχείο (εάν επιθυμείτε) και πατήστε στο κουμπί «Αποστολή» για να αποσταλεί το μήνυμα.

2) Αποστολή μηνυμάτων στον καθηγητή ενός μαθήματος

Έχοντας συνδεθεί ως μαθητής στην εφαρμογή, για να στείλετε ένα νέο μήνυμα στον καθηγητή ενός μαθήματος σας, επιλέξτε «Τα μηνύματά μου» στην αριστερή στήλη στο κάθετο μενού επιλογών, και πατήστε στο κουμπί «Νέο μήνυμα» (Εικόνα 80).



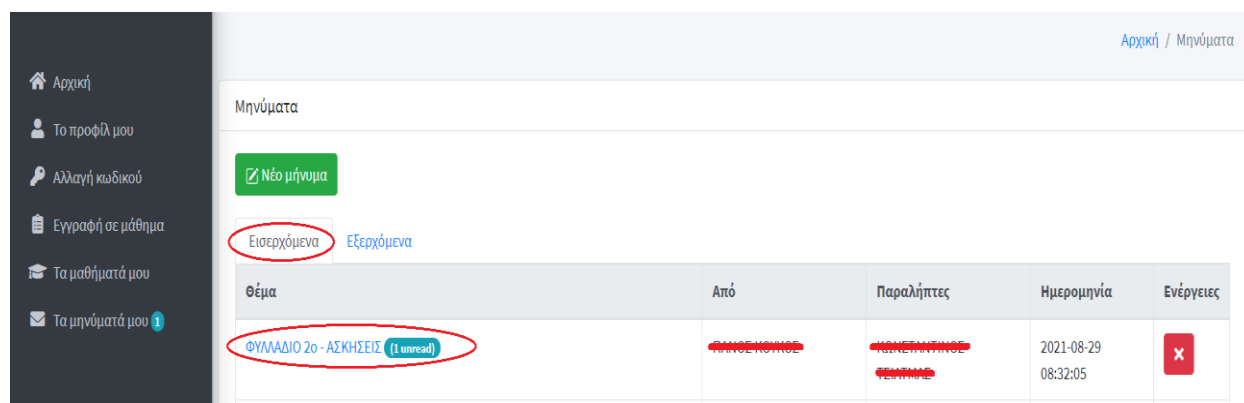
Εικόνα 80: Αποστολή νέου μηνύματος εκπαιδευόμενου

Στην φόρμα «Νέο μήνυμα» (Εικόνα 81), επιλέξτε το μάθημα στο οποίο θα γίνει η αποστολή του μηνύματος, τον παραλήπτη (που θα είναι εξ' ορισμού ο καθηγητής του μαθήματος), το θέμα του μηνύματος (είναι υποχρεωτικό), το κείμενο του μηνύματος , το συνημμένο αρχείο (εάν επιθυμείτε) και πατήστε στο κουμπί «Αποστολή» για να αποσταλεί το μήνυμα.

Εικόνα 81: Φόρμα αποστολής μηνύματος εκπαιδευόμενου

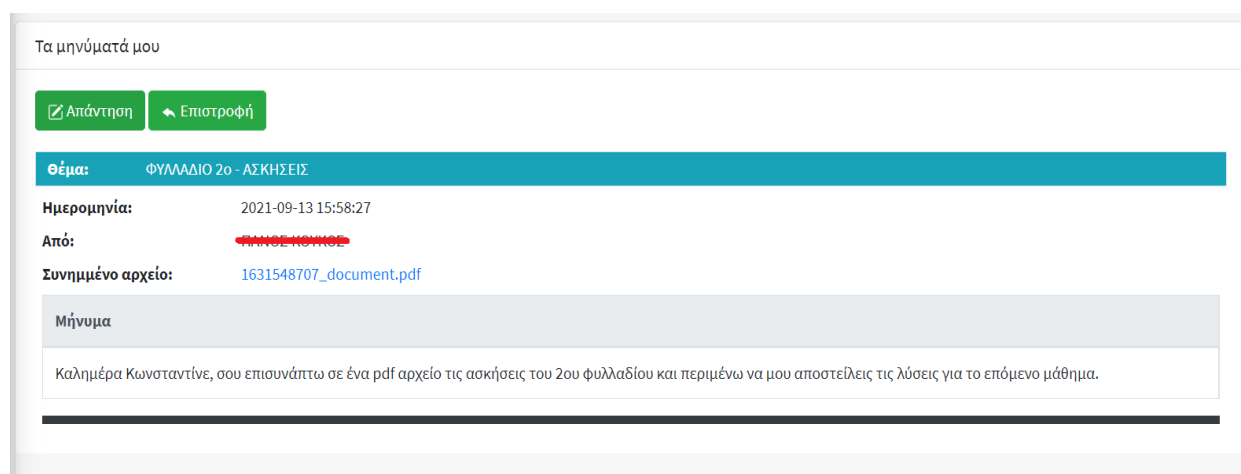
3) Απάντηση σε συγκεκριμένο μήνυμα

Για να απαντήσετε σε ένα μήνυμα που σας έχει σταλεί, επιλέξτε στην καρτέλα "Εισερχόμενα" και κάντε κλικ πάνω στο θέμα του συγκεκριμένου μηνύματος (Εικόνα 82).



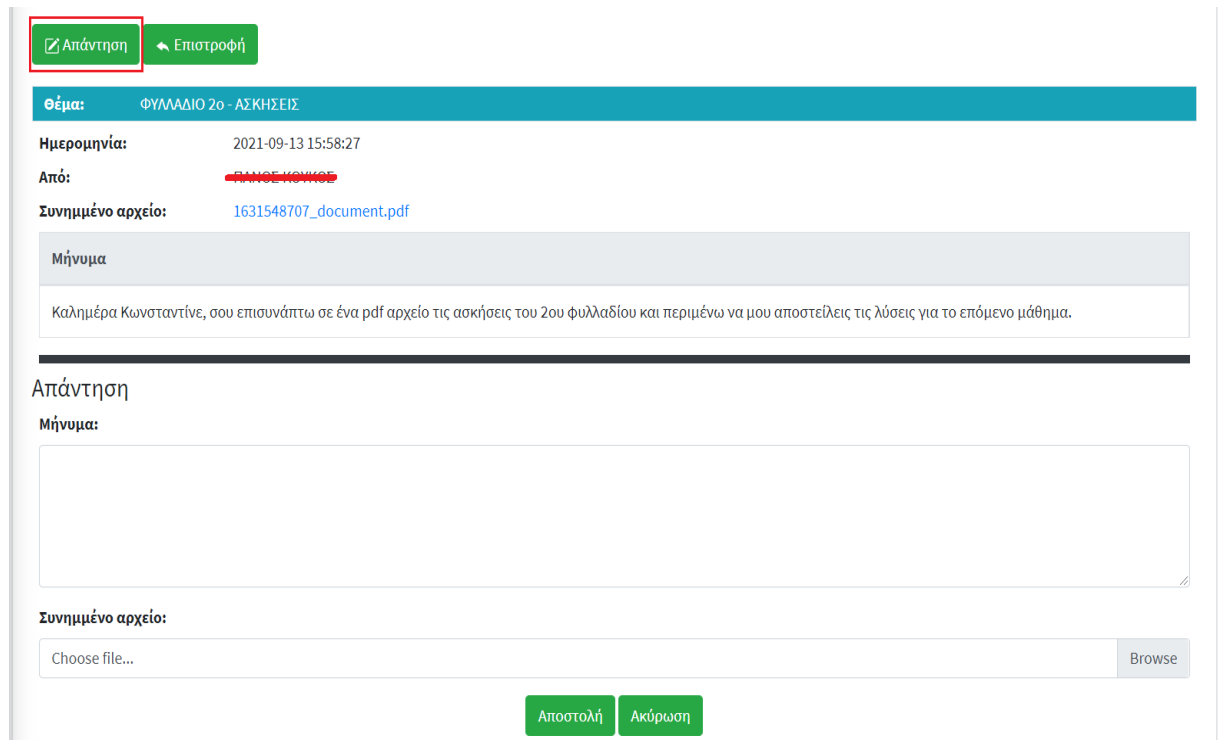
Εικόνα 82: Απάντηση μηνύματος

Στην φόρμα «Τα μηνυμάτά μου», βλέπετε το θέμα, την ημερομηνία αποστολής, τον αποστολέα, το συνημμένο αρχείο (εάν υπάρχει) και το κείμενο του μηνύματος για το συγκεκριμένο μήνυμα (Εικόνα 83).



Εικόνα 83: Προβολή μηνύματος

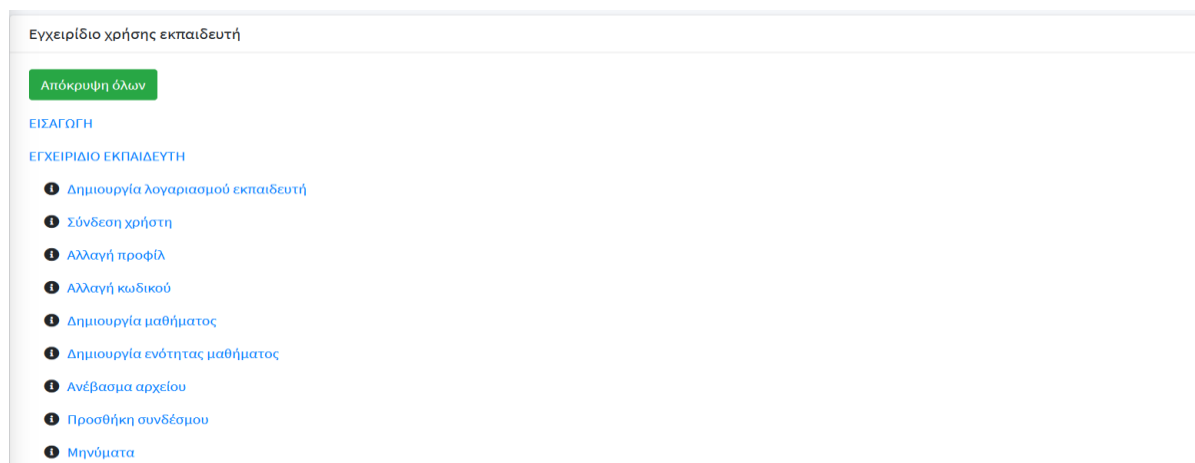
Για να απαντήσετε κάντε κλικ στο κουμπί «Απάντηση», πληκτρολογήστε το κείμενο του μηνύματος, το συνημμένο αρχείο (εάν επιθυμείτε) και πατήστε στο κουμπί «Αποστολή» (Εικόνα 84).



Εικόνα 84: Απάντηση μηνύματος

6.5.2 Εγχειρίδια χρήσης

Τέλος, η εφαρμογή παρέχει στους εγγεγραμμένους χρήστες της, ανάλογα με το ρόλο που επιτελούν, εγχειρίδια χρήσης. Έτσι, για παράδειγμα, βλέπουμε στην παρακάτω εικόνα (Εικόνα 85) τη σελίδα με το εγχειρίδιο χρήσης, χωρισμένο ανά θέματα βοήθειας, ενός χρήστη με ρόλο εκπαιδευτή.



Εικόνα 85: Εγχειρίδιο χρήσης εκπαιδευτή

7. Συμπεράσματα – Επεκτάσεις

7.1 Συμπεράσματα

Αρχικός στόχος της εργασίας ήταν η ανάπτυξη μιας εφαρμογής διαχείρισης ηλεκτρονικών μαθημάτων μέσω ιστού. Μέσα από τις λειτουργίες της εφαρμογής αυτής, χρήστες με ρόλους εκπαιδευτή και εκπαιδευόμενου, θα έπρεπε να μπορούν να αλληλεπιδρούν με στόχο την αποθήκευση, οργάνωση και παρουσίαση ηλεκτρονικών μαθημάτων.

Πιο συγκεκριμένα, κάθε εκπαιδευτής της εφαρμογής, θα έπρεπε να μπορεί να δημιουργεί ηλεκτρονικά μαθήματα και να οργανώνει το ψηφιακό εκπαιδευτικό υλικό (κείμενα, παρουσιάσεις, εικόνες, διαγράμματα, κλπ) των μαθημάτων σε ενότητες. Επίσης, κάθε εκπαιδευόμενος της εφαρμογής, θα έπρεπε να μπορεί να έχει πρόσβαση στο ψηφιακό υλικό καθώς και δυνατότητες επικοινωνίας με τους εκπαιδευτές. Επιπρόσθετα, σημαντικό χαρακτηριστικό της εφαρμογής θα ήταν, όλες αυτές οι δυνατότητες, να μπορούν να υλοποιούνται εύκολα και γρήγορα, μέσα από ένα φιλικό περιβάλλον διεπαφής χρήστη.

Η ιδέα της χρήσης ενός PHP framework βοήθησε πολύ. Η επιλογή του Laravel, το οποίο με την MVC αρχιτεκτονική που ακολουθεί, έδωσε τη δυνατότητα για καλύτερη οργάνωση των αρχείων, ευκολότερη και ταχύτερη ανάπτυξη, επαναχρησιμοποίηση τμημάτων κώδικα, μεγαλύτερη ευκολία στον σχεδιασμό την ανάπτυξη και την συντήρηση της βάσης δεδομένων (μέσα από τα εξαιρετικά εργαλεία Eloquent ORM και Migration) και άλλα πολλά, αποδείχθηκε εξαιρετική.

Επίσης, αξιοποιώντας όλες τις σύγχρονες τεχνολογίες που υπάρχουν στο Web design σήμερα και μέσα από τη χρήση του framework Bootstrap, καταφέραμε να δημιουργήσουμε μια εφαρμογή, με κομψή και μοντέρνα διεπαφή χρήστη (UI), καθώς και με προσαρμοστική (responsive) διάταξη (layout), ώστε να μπορεί να προσαρμόζεται στις οθόνες διαφορετικών συσκευών, συμπεριλαμβανομένων ηλεκτρονικών υπολογιστών, tablets και smartphones.

Κλείνοντας, η διαδικτυακή εφαρμογή που υλοποιήσαμε για την παρούσα διπλωματική εργασία, πιστεύουμε ότι μπορεί να αποτελέσει ένα ψηφιακό

εκπαιδευτικό εργαλείο μάθησης, που θα φανεί χρήσιμο τόσο στους καθηγητές όσο και στους μαθητές της πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης.

7.2 Επεκτάσεις

Όσων αφορά τις πιθανές μελλοντικές επεκτάσεις της εφαρμογής, αρχικά μια πρόταση θα ήταν η χρήση ενσωματωμένου κειμενογράφου. Για τη μορφοποίηση των κειμένων που εισάγει ένας χρήστης της εφαρμογής, καλό θα ήταν να μπορεί να χρησιμοποιεί τη μπάρα εργαλείων ενός ενσωματωμένου κειμενογράφου, ώστε να:

- επιλέγει τη γραμματοσειρά, το μέγεθος, το χρώμα των γραμμμάτων και το χρώμα του φόντου για το κείμενο που θα πληκτρολογεί
- επιλέγει τα γράμματα να εμφανίζονται έντονα, πλάγια ή υπογραμμισμένα
- ορίζει τη στοίχιση του κειμένου και να επιλέγει αν το κείμενο θα εμφανίζεται σε μορφή λίστας
- εισάγει ένα σύνδεσμο ή μια εικόνα στο κείμενο

Μια άλλη επέκταση, η οποία θα μπορούσε να πραγματοποιηθεί στην εφαρμογή, έχει να κάνει με τη λειτουργία της απεγγραφής ενός μαθητή από ένα μάθημα. Σύμφωνα με τις δυνατότητες της εφαρμογής, ένας καθηγητής μπορεί να απεγγράψει από ένα μάθημα μόνο έναν μαθητή κάθε φορά. Θα ήταν πιο λειτουργικό να μπορεί να απεγγράψει ταυτόχρονα περισσότερους ή και όλους τους μαθητές που είναι εγγεγραμμένοι σε ένα μάθημα του.

Μια άλλη πρόταση έχει να κάνει με τα στατιστικά. Θα μπορούσαμε να προσθέσουμε στατιστικά μάθησης στην εφαρμογή. Τα στατιστικά μάθησης αποτελούν μια μετρική η οποία αφορά την συλλογή και ανάλυση δεδομένων τα οποία στοχεύουν στην βελτιστοποίηση της μάθησης από την πλευρά των μαθητών. Από την πλευρά των καθηγητών αποτελούν ένα πολύ σημαντικό εργαλείο συλλογής, αναφοράς και ανάλυσης των δεδομένων του μαθήματος με σκοπό την βελτίωσή του. Έτσι θα μπορούσαμε να προσθέσουμε στις λειτουργίες του καθηγητή τη δυνατότητα να μπορεί να παρακολουθεί, συνολικά για ένα μάθημα ή και ανά μαθητή, στατιστικά στοιχεία που έχουν να κάνουν με:

- τον αριθμό αποστολής αρχείων εργασιών
- συνδέσεις στο μάθημα

- διάρκεια σύνδεσης στο μάθημα
- "χτυπήματα" στα αρχεία και στους συνδέσμους του εκπαιδευτικού υλικού του μαθήματος

Κλείνοντας, μια τελευταία επέκταση για την εφαρμογή μας θα μπορούσε να είναι, η δυνατότητα ενός καθηγητή να κρατάει αντίγραφα ασφαλείας για τα μαθήματά του. Η διασφάλιση ολόκληρου του περιεχομένου ενός μαθήματος, σε περιπτώσεις ακούσιας διαγραφής ή καταστροφής, είναι μια πάρα πολύ σημαντική και χρήσιμη λειτουργία.

Βιβλιογραφία

- Δημητριάδης Σ., Καραγιαννίδης Χ., Πομπόρτσης Α., & Τσιάτσος Θ. (2007). *Ευέλικτη μάθηση με χρήση τεχνο-λογιών πληροφορίας και επικοινωνιών*. Εκδόσεις Τζιόλας. ISBN: 960-418-142-4.
- Ζερδέβα, Σ., (2007). *Σχεδιασμός Έξυπνου Συστήματος Ηλεκτρονικής Μάθησης (E-learning) στο Τομέα της Φυσικής της πυρηνικής Ιατρικής*. Διπλωματική εργασία ΜΔΕ στην Ιατρική Φυσικής, Πανεπιστήμιο Πατρών. Ανάκτηση 4-4-2012, από <http://nemertes.lis.upatras.gr/jspui/handle/10889/609>
- Θρασύβουλος-Κωνσταντίνος Τσιάτσος, (2015) Εκπαιδευτικά περιβάλλοντα Διαδικτύου.
- EDEN – UNESCO, (2001). *Higher Education Open and Distance Learning Knowledge Base for Decision Makers. First Interim Report*. Budapest
- Cowan J., (1995), «The Advantages and Disadvantages of Distance Education», *Distance Education for Language Teachers*, 1995, pp. 14-20
- Long, Phillip D. (2004). *Εγκυκλοπαίδεια της Κατανεμημένης Μάθησης*. Thousand Oaks: SAGE Publications, Inc. σελ. 291-293. doi : 10.4135 / 9781412950596.n99
- Montrieux, Hannelore; Vanderlinde, Ruben; Schellens, Tammy. Marez, Lieven De (2015-12-7). “Διδασκαλία και εκμάθηση με την κινητή τεχνολογία: Ποιοτική διερευνητική μελέτη για την εισαγωγή συσκευών δισκίων στη δευτεροβάθμια εκπαίδευση”.
- Moore, M.G. & Kearsley, G. (2005) *Distance education. A systems view*. Belmont, CA:Wadsworth
- Moore, M.G. & Kearsley, G. (2005) *Distance education. A systems view*. Belmont, CA:Wadsworth.

Ιστοσελίδες

Wikipedia, Ηλεκτρονική μάθηση, wikipedia.org. Διαθέσιμο: https://el.wikipedia.org/wiki/Ηλεκτρονική_μάθηση , (2021).

Wikipedia, HTML, wikipedia.org. Διαθέσιμο: <https://el.wikipedia.org/wiki/HTML>, (2019).

w3schools, HTML History, w3schools.in. Διαθέσιμο: <https://www.w3schools.in/html-tutorial/history/> , (2021).

w3schools, HTML5 New Elements, w3schools-fa.ir. Διαθέσιμο: http://w3schools-fa.ir/html/html5_new_elements.html, (2021).

Wikipedia, Bootstrap, wikipedia.org. Διαθέσιμο: <https://el.wikipedia.org/wiki/Bootstrap> , (2020).

getbootstrap, About, getbootstrap.com. Διαθέσιμο: <https://getbootstrap.com/docs/4.6/about/overview/> , (2021).

Sitesbay, Features of Bootstrap, sitesbay.com. Διαθέσιμο: <https://www.sitesbay.com/bootstrap/bootstrap-features-of-bootstrap>, (2021).

Safwana, How to Use Bootstrap With HTML, bootstrapdash.com. Διαθέσιμο: <https://www.bootstrapdash.com/use-bootstrap-with-html/> , (2021).

Wikipedia, Laravel, wikipedia.org. Διαθέσιμο: <https://en.wikipedia.org/wiki/Laravel>, (2021).

Tutorialandexample, Laravel Tutorial, tutorialandexample.com. Διαθέσιμο: <https://www.tutorialandexample.com/laravel-tutorial/>, (2019).

Wikipedia, Model-view-controller, wikipedia.org. Διαθέσιμο: <https://el.wikipedia.org/wiki/Model-view-controller>, (2021).

Laravel, Routing, laravel.com. Διαθέσιμο: <https://laravel.com/docs/8.x/routing>, (2021).

Laravel, Middleware, laravel.com. Διαθέσιμο: <https://laravel.com/docs/8.x/middleware>, (2021).

Laravel, Controllers, laravel.com. Διαθέσιμο: <https://laravel.com/docs/8.x/controllers>, (2021).

Laravel, Views, laravel.com. Διαθέσιμο: <https://laravel.com/docs/8.x/views>, (2021).

Laravel, Blade Templates, laravel.com. Διαθέσιμο: <https://laravel.com/docs/8.x/blade>, (2021).

Laravel, Database: Migrations, laravel.com. Διαθέσιμο: <https://laravel.com/docs/8.x/migrations>, (2021).

Laravel, Database: Getting Started, laravel.com. Διαθέσιμο: <https://laravel.com/docs/8.x/database>, (2021).

Laravel, Database: Query Builder, laravel.com. Διαθέσιμο: <https://laravel.com/docs/8.x/queries>, (2021).

Laravel, Eloquent: Getting Started, laravel.com. Διαθέσιμο: <https://laravel.com/docs/8.x/eloquent>, (2021).

Wikipedia, Object–relational mapping, wikipedia.org. Διαθέσιμο: https://en.wikipedia.org/wiki/Object–relational_mapping, (2021).

Laravel, Eloquent: Relationships, laravel.com. Διαθέσιμο: <https://laravel.com/docs/8.x/eloquent-relationships>, (2021).

Wikipedia, MySQL, wikipedia.org. Διαθέσιμο: <https://el.wikipedia.org/wiki/MySQL>, (2020).

Wikipedia, SQL, wikipedia.org. Διαθέσιμο: <https://el.wikipedia.org/wiki/SQL>, (2021).

Richard, B., What is MySQL: MySQL Explained For Beginners, hostinger.com. Διαθέσιμο: <https://www.hostinger.com/tutorials/what-is-mysql>, (2021).

DB-ENGINES, DB-Engines Ranking, db-engines.com. Διαθέσιμο: <https://db-engines.com/en/ranking>, (2021).

Wikipedia, PHP, wikipedia.org. Διαθέσιμο: <https://el.wikipedia.org/wiki/PHP>, (2021).

Will, M., Why You Should Use PHP Frameworks, hostinger.com. Διαθέσιμο: <https://www.hostinger.com/tutorials/best-php-framework>, (2021).

W3Techs, Usage statistics of PHP for websites, w3techs.com. Διαθέσιμο: <https://w3techs.com/technologies/details/pl-php>, (2021).

Wikipedia, Category: PHP frameworks, wikipedia.org. Διαθέσιμο: https://en.wikipedia.org/wiki/Category:PHP_frameworks, (2019).

Symfony, Ten criteria for choosing the correct framework, symfony.com. Διαθέσιμο: <https://symfony.com/ten-criteria>, (2021).

Brotherton, C., The Most Popular PHP Frameworks to Use in 2021: Pros of Laravel, kinsta.com. Διαθέσιμο: <https://kinsta.com/blog/php-frameworks/>, (2021).

Wikipedia, Symfony, wikipedia.org. Διαθέσιμο: <https://en.wikipedia.org/wiki/Symfony>, (2021).

Brotherton, C., The Most Popular PHP Frameworks to Use in 2021: Pros of Symfony, kinsta.com. Διαθέσιμο: <https://kinsta.com/blog/php-frameworks/>, (2021).

Wikipedia, CodeIgniter, wikipedia.org. Διαθέσιμο: <https://en.wikipedia.org/wiki/CodeIgniter>, (2021).

CodeIgniter, CodeIgniter Rocks, codeigniter.com. Διαθέσιμο: <https://codeigniter.com/>, (2021).

Brotherton, C., The Most Popular PHP Frameworks to Use in 2021: Pros of CodeIgniter, kinsta.com. Διαθέσιμο: <https://kinsta.com/blog/php-frameworks/>, (2021).

ASPERBROTHERS, Laravel vs Symfony in 2021 – Which PHP Framework Choose For Your Project, asperbrothers.com. Διαθέσιμο: <https://asperbrothers.com/blog/laravel-vs-symfony/>, (2019).

Faheem, H., Laravel vs Codeigniter-Which is Better PHP Framework, mavencluster.com. Διαθέσιμο: <https://www.mavencluster.com/blog/laravel-vs-codeigniter-php-framework/>, (2021).

Github, Github-Ranking, github.com. Διαθέσιμο: <https://github.com/EvanLi/Github-Ranking/blob/master/Top100/PHP.md>, (2021).

Stack Overflow, Stack Overflow Trends, insights.stackoverflow.com. Διαθέσιμο: <https://insights.stackoverflow.com/trends?tags=laravel%2Csymfony%2Ccodeigniter>, (2021).

Google Trends, Σύγκριση, trends.google.com. Διαθέσιμο: <https://trends.google.com/trends/explore?cat=5&date=2011-01-01%202021-10-22&q=%2Fm%2F0jwy148,%2Fm%2F09cjcl,%2Fm%2F02qgdkj#TIMESERIES>, (2021).

Laravel, Starter Kits: Laravel Breeze, laravel.com. Διαθέσιμο: <https://laravel.com/docs/8.x/starter-kits>, (2021).

Packagist, Laravel Breadcrumbs, packagist.org. Διαθέσιμο: <https://packagist.org/packages/diglactic/laravel-breadcrumbs>, (2021).

GitHub, Laravel Messenger, github.com. Διαθέσιμο: <https://github.com/cmgyr/laravel-messenger>, (2021).

Pusher-community.github, What is Pusher, pusher-community.github.io. Διαθέσιμο: <https://pusher-community.github.io/real-time-laravel/introduction/what-is-pusher.html>, (2021).