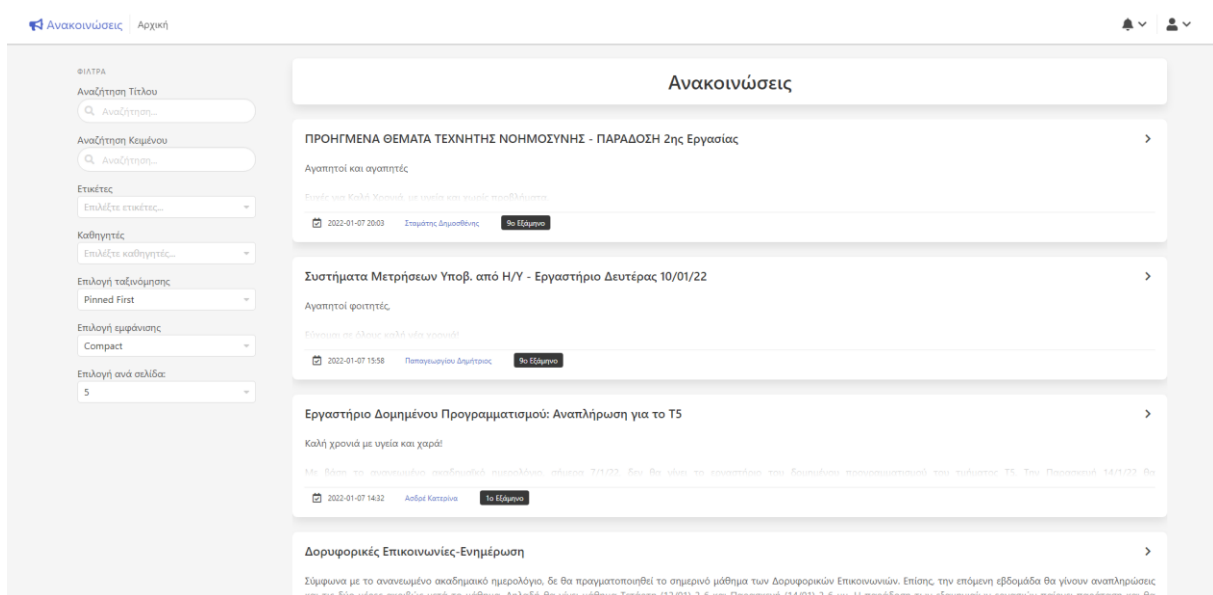


ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
«ΒΕΛΤΙΩΣΗ ΣΥΣΤΗΜΑΤΟΣ ΑΝΑΚΟΙΝΩΣΕΩΝ
ABOARD»



Των φοιτητών
Στίνη Γεώργιου
Παπαδόπουλου Παντελεήμονα
Αρ. Μητρώου: 174890 - 174891

Επιβλέπων
Σιδηρόπουλος Αντώνιος
Βαθμίδα: Επίκουρος Καθηγητής

22 Ιουνίου 2022

Τίτλος Δ.Ε.: ΒΕΛΤΙΩΣΗ ΣΥΣΤΗΜΑΤΟΣ ΑΝΑΚΟΙΝΩΣΕΩΝ ABOARD

Κωδικός Δ.Ε.: 21342

Ονοματεπώνυμο φοιτητών: Στίνης Γεώργιος – Παπαδόπουλος Παντελεήμων

Ονοματεπώνυμο εισηγητή Σιδηρόπουλος Αντώνιος

Ημερομηνία ανάληψης Δ.Ε.: 13-10-2021

Ημερομηνία περάτωσης Δ.Ε. 22-06-2022

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

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

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

*«Την παρούσα πτυχιακή την αφιερώνουμε στους ανθρώπους που μας στάθηκαν σε
όλα αυτά τα χρόνια των σπουδών μας»*

Πρόλογος

Λόγο της προηγούμενης εμπειρίας που είχαμε στον τομέα της ανάπτυξης διαδικτυακών εφαρμογών αλλά και της προηγούμενης μας συνεργασίας, αποφασίσαμε να αναλάβουμε να βελτιώσουμε την ήδη υπάρχουσα εφαρμογή, η οποία αφορούσε το ανανεωμένο σύστημα ανακοινώσεων του τμήματος. Επιλέξαμε την παρούσα πτυχιακή καθώς χρησιμοποιούνταν τεχνολογίες γνώριμες και για τους δύο, αλλά και επειδή αφορούσε έναν τομέα τον οποίο θέλαμε να ακολουθήσουμε επαγγελματικά στο μέλλον. Κατά την διάρκεια υλοποίησης της εργαζόμασταν παράλληλα σε εταιρείες με κύρια αρμοδιότητα την ανάπτυξη διαδικτυακών εφαρμογών. Η πτυχιακή αυτή μας βοήθησε να βελτιωθούμε στον τομέα της συνεργασίας και να εμβαθύνουμε τις γνώσεις μας επάνω στον τομέα των διαδικτυακών εφαρμογών, κάτι το οποίο μας φάνηκε ιδιαίτερα χρήσιμο και στον επαγγελματικό μας τομέα. Επιπλέον, λόγο του ενδεχόμενου να χρησιμοποιηθεί η παρούσα πτυχιακή από όλους τους καθηγητές, αλλά και τους φοιτητές, μας έδωσε ένα κίνητρο να ασχοληθούμε με περισσότερη λεπτομέρεια και επαγγελματισμό.

Περίληψη

Στόχος της πτυχιακής εργασίας είναι η βελτίωση της ιστοσελίδας Aboard, όπου αναπτύχθηκε σε προηγούμενη πτυχιακή, και η οποία θα αποτελέσει το μελλοντικό σύστημα ανακοινώσεων του τμήματος, από το οποίο οι φοιτητές θα μπορούν να ενημερώνονται για οποιαδήποτε πληροφορία αφορά την φοίτηση τους. Ο φοιτητής μετά την βελτίωση μπορεί να αναζητήσει με μεγαλύτερη λεπτομέρεια συγκεκριμένες ανακοινώσεις μέσα στην ιστοσελίδα χάρη στα πολλαπλά φίλτρα, τη δυνατότητα ταξινόμησης και αλλαγής τρόπου εμφάνισης των ανακοινώσεων, που προστέθηκαν κατά την διάρκεια υλοποίησης της. Έπειτα μπορεί να εγγράφεται σε διάφορες ετικέτες που αφορούν συγκεκριμένα μαθήματα ώστε να έρχονται ειδοποιήσεις μόνο για τα μαθήματα που έχει κάνει εγγραφή και πιθανώς παρακολουθεί. Παράλληλα έγιναν πολλαπλές βελτιώσεις στο κομμάτι του Front-End όπως η ανανέωση του navigation-pagination, η βελτίωση εμφάνισης εκδηλώσεων και σημαντικών ανακοινώσεων, κ.α. Τέλος βελτιστοποιήθηκε σε μεγάλο βαθμό και το API της εφαρμογής έτσι ώστε να υπάρχει μεγαλύτερη δυνατότητα επεκτασιμότητας, βελτίωσης και συντήρησης στο μέλλον.

« Aboard Announcement System Improvement »

« Stinis Georgios – Papadopoulos Pantelis »

Abstract

The aim of the thesis is to improve the aboard website, where it was developed in a previous thesis, and which will be used in the future as the announcement system of our department, from which students will be able to be informed about any information related to their study. After the improvement, the student can search, in great detail, specific announcements within the website thanks to the multiple filters, the ability to sort and change the appearance of the announcements, which were added during its implementation. He can enroll in various course-specific labels to get notifications only about the courses he has enrolled in and are likely to attend. At the same time, multiple improvements were made in the Front-End, such as the renewal of navigation-pagination, the improvement of the appearance of events and important announcements, etc. Finally, the API of the application has been greatly optimized so that there is more scalability, improvement and maintenance in the future.

Ευχαριστίες

Αρχικά, θα θέλαμε να ευχαριστήσουμε τις οικογένειες μας για την ηθική τους υποστήριξη, καθώς και την εμπιστοσύνη που μας έδειξαν όλα αυτά τα χρόνια των σπουδών μας. Επίσης θα θέλαμε να ευχαριστήσουμε τον υπεύθυνο καθηγητή μας, κ. Αντώνη Σιδηρόπουλο για την πολύτιμη βοήθεια του, την καθοδήγησή του και τις συμβουλές που μοιράστηκε μαζί μας κατά την εκπόνηση της πτυχιακής εργασίας.

Περιεχόμενα

Πρόλογος.....	v
Περίληψη.....	vi
Abstract	vii
Ευχαριστίες	viii
Περιεχόμενα	ix
Κατάλογος Σχημάτων	xi
Συντομογραφίες.....	xiii
Κεφάλαιο 1ο: Ανάλυση Απαιτήσεων	14
1.1 Περιγραφή υπάρχοντος συστήματος.....	14
1.2 Ανάλυση Απαιτήσεων	14
Κεφάλαιο 2ο: Τεχνολογίες που χρησιμοποιήθηκαν	16
2.1 Εισαγωγή.....	16
2.2 Τεχνολογίες Front-End.....	16
2.2.1 HTML/CSS/Javascript	16
2.2.2 Bulma CSS Framework.....	17
2.2.3 npm.....	18
2.2.4 Vue.js.....	18
2.3 Τεχνολογίες Back-End	22
2.3.1 PHP.....	23
2.3.2 Laravel.....	24
2.3.3 Composer.....	28
2.3.4 API	29
2.3.5 JSON	31
2.3.6 OAuth2.0	32
2.3.7 MySQL.....	33
2.3.8 Postman	35
2.4 Επίλογος.....	36
Κεφάλαιο 3ο: Οδηγοί Εγκατάστασης και Λειτουργίας της Εφαρμογής	37
3.1 Εισαγωγή.....	37
3.2 Τοπική Εγκατάσταση (Windows)	37
3.3 Εγκατάσταση στον server.....	38
3.4 Οδηγίες λειτουργίας	39

3.5	Οδηγός εγκατάστασης OAuth2.0	39
3.6	Επίλογος.....	41
Κεφάλαιο 4ο: Σχεδίαση του συστήματος.....		42
4.1	Εισαγωγή.....	42
4.2	Database Schema.....	42
4.3	Σχέσεις μεταξύ πινάκων	43
4.4	API της εφαρμογής.....	44
4.4.1	Μέθοδοι GET	45
4.4.2	Μέθοδοι POST	51
4.4.3	Μέθοδοι PUT	53
4.4.4	Μέθοδοι DELETE.....	54
Κεφάλαιο 5ο: Παρουσίαση της εφαρμογής.....		55
5.1	Εισαγωγή.....	55
5.2	Περιγραφή των φίλτρων.....	55
5.3	Σύνδεση χρήστη μέσω του τμήματος.....	63
5.4	Δυνατότητα εμφάνισης όλων των ανακοινώσεων	64
5.5	Δυνατότητες συνδεδεμένου χρήστη	64
5.6	Δυνατότητες συγγραφέα.....	68
5.7	Δυνατότητες διαχειριστή.....	69
5.8	Επίλογος.....	72
Κεφάλαιο 6ο: Συμπεράσματα ή/και προτάσεις βελτίωσης		73
6.1	Εισαγωγή.....	73
6.2	Μελλοντικές προτάσεις βελτίωσης	73
6.3	Συμπεράσματα.....	74
6.4	Επίλογος.....	75
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		76

Κατάλογος Σχημάτων

Σχήμα 2.1: Λογότυπα HTML/CSS/JS.....	16
Σχήμα 2.2: HTML Released Year.....	17
Σχήμα 2.3: Λογότυπο Bulma.....	17
Σχήμα 2.4: Λογότυπο npm.....	18
Σχήμα 2.5: Λογότυπο Vue.js.....	18
Σχήμα 2.6: Κύκλος ζωής ενός Vue Instance.....	20
Σχήμα 2.7: Παράδειγμα ενός Vue Component.....	21
Σχήμα 2.8: Παράδειγμα εισαγωγής prop σε Vue Component.....	21
Σχήμα 2.9: Παράδειγμα χρήσης Vue DevTools.....	22
Σχήμα 2.10: Λογότυπο PHP.....	23
Σχήμα 2.11: Λογότυπο Laravel.....	24
Σχήμα 2.12: Αρχιτεκτονική MVC στην Laravel.....	27
Σχήμα 2.13: Παράδειγμα Laravel DebugBar.....	28
Σχήμα 2.14: Λογότυπο Composer.....	29
Σχήμα 2.15: Λογότυπο API.....	29
Σχήμα 2.16: Λειτουργία του REST API.....	31
Σχήμα 2.17: Παράδειγμα JSON.....	32
Σχήμα 2.18: Λογότυπο OAuth2.0.....	32
Σχήμα 2.19: Λογότυπο MySQL.....	33
Σχήμα 2.20: Λογότυπο MySQL.....	34
Σχήμα 2.21: Λογότυπο Postman.....	35
Σχήμα 4.1: Παράδειγμα του σχήματος της βάσης.....	42
Σχήμα 4.2: Σχέσεις μεταξύ των πινάκων.....	43
Σχήμα 5.1: Αναζήτηση ανακοινώσεων βάση τίτλου.....	55
Σχήμα 5.2: Αναζήτηση ανακοινώσεων βάση κειμένου.....	56
Σχήμα 5.3: Φιλτράρισμα ανακοινώσεων βάση πολλαπλών ετικετών.....	57
Σχήμα 5.4: Φιλτράρισμα ανακοινώσεων βάση πολλαπλών κατηγοριών.....	57
Σχήμα 5.5: Ταξινόμηση ανακοινώσεων.....	58
Σχήμα 5.6: Εμφάνιση κειμένου ανακοινώσεων.....	59
Σχήμα 5.7: Εμφάνιση μόνο του τίτλου ανακοινώσεων.....	59
Σχήμα 5.8: Επιλογή αριθμού ανακοινώσεων ανά σελίδα.....	60
Σχήμα 5.9: Δυνατότητα τοποθέτησης φίλτρων στον σύνδεσμο.....	61
Σχήμα 5.10: Δυνατότητα επιλογής φιλτραρίσματος μέσω της ανακοίνωσης.....	62
Σχήμα 5.11: Συνδυασμός φίλτρων.....	62
Σχήμα 5.12: Αποθήκευση φίλτρων εμφάνισης στο local storage.....	63
Σχήμα 5.13: Ανανεωμένο σύστημα σελιδοποίησης.....	63
Σχήμα 5.14: Αυθεντικοποίηση μέσω του τμήματος.....	64
Σχήμα 5.15: Εμφάνιση των τελευταίων ειδοποιήσεων ενός συνδεδεμένου χρήστη.....	65
Σχήμα 5.16: Εμφάνιση όλων των ειδοποιήσεων.....	65
Σχήμα 5.17: Δυνατότητα εγγραφής σε πολλαπλές ετικέτες.....	66
Σχήμα 5.18: Δυνατότητα υποβολής ενός προβλήματος.....	66
Σχήμα 5.19: Documentation για το API της εφαρμογής.....	67
Σχήμα 5.20: Documentation για τους πίνακες της βάσης.....	67
Σχήμα 5.21: Δημιουργία ανακοίνωσης 1/2.....	68

Σχήμα 5.22: Δημιουργία ανακοίνωσης 2/2	69
Σχήμα 5.23: Δυνατότητα διαγραφής και επεξεργασίας για τον συγγραφέα των ανακοινώσεων του ...	69
Σχήμα 5.24: Ανέβασμα ανακοίνωσης ως άλλος χρήστης	70
Σχήμα 5.25: Διαχείριση ετικετών.....	71
Σχήμα 5.26: Προσθήκη/Επεξεργασία ετικέτας	71
Σχήμα 5.27: Εμφάνιση προβλημάτων	72

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΙΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
API	Application Programming Interface
URL	Uniform Resource Locator
JS	JavaScript
HTML	HyperText Markup Language
XHTML	Extensible HyperText Markup Language
CSS	Cascading Style Sheets
npm	Node Package Manager
CLI	Command-Line Interface
DOM	Document Object Model
CPU	Central Processing Unit
PHP	Hypertext Preprocessor
CSRF	Cross-Site Request Forgery
ORM	Object–relational mapper
MVC	Model View Controller
JSON	JavaScript Object Notation
IIS	Internet Information Services
IDE	Integrated Development Environment
OAuth	Open Authorization
XML	Extensible Markup Language
XML-RPC	XML-Remote Procedure Call
UDDI	Universal Description, Discovery and Integration
SOAP	Simple Object Access Protocol
HTTP	HyperText Transfer Protocol
SMTP	Simple Mail Transfer Protocol
REST	Representational State Transfer
JWT	JSON Web Token
SQL	Structured Query Language
FPM	FastCGI Process Manager

Κεφάλαιο 1ο: Ανάλυση Απαιτήσεων

1.1 Περιγραφή υπάρχοντος συστήματος

Στόχος της παρούσας εργασίας ήταν να βελτιωθεί η υπάρχουσα εφαρμογή που είχε δημιουργηθεί από προηγούμενο συμφοιτητή μας. Στην υπάρχουσα εφαρμογή υπήρχαν ήδη αρκετές λειτουργίες για τους χρήστες. Ο χρήστης μπορούσε να συνδεθεί χρησιμοποιώντας το σύστημα LDAP του τμήματος. Υπήρχε η δυνατότητα δημιουργίας/επεξεργασίας/διαγραφής μίας ανακοίνωσης από τον διαχειριστή και τον συγγραφέα της εκάστοτε ανακοίνωσης. Ο χρήστης είχε την δυνατότητα να εγγράφεται σε πολλαπλές ετικέτες. Επίσης υπήρχε η επιλογή να οριστεί μία ανακοίνωση ως εκδήλωση ή και ως σημαντική. Ο χρήστης μπορούσε να αναζητήσει ανακοινώσεις με βάσει κάποιων κριτηρίων και να γίνεται εμφάνιση μόνο των αποτελεσμάτων που είναι σε θέση να λάβει αναλόγως αν είναι συνδεδεμένος ή όχι. Τέλος υπήρχε η δυνατότητα εμφάνισης όλων των ανακοινώσεων, κατά την είσοδο από το IP range του ΔΠΠΑΕ, χωρίς να είναι απαραίτητη η σύνδεση στην εφαρμογή.

1.2 Ανάλυση Απαιτήσεων

Παρακάτω αναλύονται οι απαιτήσεις που υπήρχαν κατά τη διάρκεια υλοποίησης της εφαρμογής. Αναφέρονται ονομαστικά και γίνεται μία σύντομη ανάπτυξη της κάθε απαίτησης.

1. Οργάνωση των αρχείων της εφαρμογής:
Διαγραφή αρχείων και φακέλων που δεν χρησιμοποιούνταν. Οργάνωση του κώδικα με επεξήγηση για κάθε κομμάτι.

Back-end:

2. Υποστήριξη OAuth2.0 για τη σύνδεση με το σύστημα αυθεντικοποίησης του τμήματος:
Έγινε υλοποίηση του OAuth2.0 πρωτοκόλλου έτσι ώστε ο χρήστης να συνδέεται μέσα από το σύστημα αυθεντικοποίησης που παρέχει το τμήμα, και χρησιμοποιείται από όλες τις ιστοσελίδες του τμήματος.
3. Ενοποίηση των end-points αναζήτησης:
Πλέον όλο το κομμάτι της αναζήτησης γίνεται μέσα από ένα end-point.
4. Αλλαγή αυθεντικοποίησης από auth('api') σε auth('web') σε όλα τα routes:
Έγινε αλλαγή αυθεντικοποίησης σε auth('web') για την υποστήριξη του πρωτοκόλλου του OAuth2.0 στο api.
5. Δημιουργία mail-list:
Με αυτό τον τρόπο γίνεται με ευκολία η μαζική αποστολή mail σε περίπτωση δημιουργίας μίας ανακοίνωσης.
6. Εγγραφή καινούριου χρήστη σε όλες τις ανακοινώσεις
Όταν ένας χρήστης συνδέεται για πρώτη φορά στην ιστοσελίδα εγγράφεται σε όλες τις ανακοινώσεις
7. Δυνατότητα επιλογής εγγεγραμμένων χρηστών σε μία ετικέτα:
Δημιουργήθηκε ένα end-point για την λήψη των εγγεγραμμένων χρηστών σε μία ετικέτα, με πρόσβαση μόνο μέσω του server.

Front-end:

8. Βελτίωση του τρόπου/χώρου εμφάνισης των Σημαντικών ανακοινώσεων (pinned):
Οι σημαντικές ανακοινώσεις εμφανίζονται μαζί με όλες τις ανακοινώσεις. Ο χρήστης έχει την δυνατότητα να τις εμφανίσει πρώτες, ακριβώς πάνω από τις ανακοινώσεις, ή να τις εντάξει χρονολογικά.
9. Βελτίωση του pagination:
Πλέον ο χρήστης έχει την δυνατότητα να αλλάζει σελίδες επιλέγοντας είτε την επόμενη/προηγούμενη καθώς και να πάει απευθείας στην πρώτη ή στην τελευταία σελίδα.
10. Επιλογή τρόπου εμφάνισης σε compact και list:
Για τον τρόπο εμφάνισης των ανακοινώσεων πλέον έχουμε δύο επιλογές. Η πρώτη επιλογή είναι αυτή που υπήρχε ήδη, δηλαδή να εμφανίζεται και το κείμενο μαζί με τον τίτλο σε κάθε ανακοίνωση. Η δεύτερη επιλογή είναι να εμφανίζονται οι ανακοινώσεις σαν λίστα, μόνο με τον τίτλο και τις πληροφορίες της κάθε ανακοίνωσης. Η επιλογή αυτή αποθηκεύεται τοπικά στον φυλλομετρητή.
11. Δυνατότητα επιλογής ταξινόμησης:
Οι επιλογές ταξινόμησης είναι τρεις. Η πρώτη είναι η προκαθορισμένη και εμφανίζει τις σημαντικές ανακοινώσεις πρώτες. Η δεύτερη εμφανίζει πρώτα τις νεότερες ανακοινώσεις. Και η τρίτη εμφανίζει τις παλαιότερες.
12. Δυνατότητα επιλογής ανακοινώσεων ανά σελίδα:
Ο χρήστης μπορεί να επιλέγει πόσες ανακοινώσεις θέλει να βλέπει σε κάθε σελίδα. Η επιλογή αυτή επίσης αποθηκεύεται τοπικά στον φυλλομετρητή.
13. Βελτίωση αναζήτησης με πολλαπλά κριτήρια:
Πλέον η αναζήτηση περιλαμβάνει τέσσερα κριτήρια. Το πρώτο και δεύτερο κριτήριο είναι η αναζήτηση μέσω του τίτλου και του περιεχομένου αντίστοιχα. Το τρίτο κριτήριο αφορά την αναζήτηση με τις ετικέτες, όπου γίνεται μέσω επιλογής. Κάθε ετικέτα ως επιλογή εμφανίζει και τον αριθμό των ανακοινώσεων που περιέχει. Το τέταρτο και τελευταίο κριτήριο είναι η αναζήτηση μέσω του καθηγητή, όπου επίσης γίνεται μέσω επιλογής. Όπως και στις ετικέτες εμφανίζεται και ο αριθμός ανακοινώσεων που έχει αναρτήσει ο κάθε καθηγητής. Τέλος όλα αυτά τα κριτήρια μπορούν να συνδυαστούν για μεγαλύτερη ακρίβεια στην αναζήτηση.
14. Προσθήκη φίλτρων στο url:
Με αυτόν τον τρόπο ένας χρήστης μπορεί να στείλει την σελίδα, τοποθετώντας τα δικά του φίλτρα, σε κάποιον άλλο χρήστη. Ο χρήστης που έλαβε τον σύνδεσμο μπορεί να δει τις ανακοινώσεις με βάση τα φίλτρα που υπάρχουν ορισμένα στο url.
15. Τοποθέτηση συνημμένων μέσα στον compact τρόπο εμφάνισης:
Με αυτόν τον τρόπο ο χρήστης μπορεί να κατεβάσει το συνημμένο αρχείο ή να το ανοίξει χωρίς να χρειαστεί να μπει στην ανακοίνωση.

Κεφάλαιο 2ο: Τεχνολογίες που χρησιμοποιήθηκαν

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει μία ανάλυση στις τεχνολογίες, τα εργαλεία και τις βιβλιοθήκες που χρησιμοποιήθηκαν για την ανάπτυξη της διαδικτυακής εφαρμογής.

Επιλέξαμε να βελτιώσουμε την εφαρμογή χρησιμοποιώντας τις τεχνολογίες που ήδη είχαν χρησιμοποιηθεί, προσθέτοντας ορισμένα εργαλεία και βιβλιοθήκες που ήταν απαραίτητα για την υλοποίηση. Η επιλογή αυτή έγινε καθώς ορισμένες από τις τεχνολογίες ήταν γνώριμες, εύκολα κατανοητές και αρκετά εύκολες για την συντήρηση της υπάρχουσας εφαρμογής. Οι τεχνολογίες αυτές έχουν μία πολύ μεγάλη κοινότητα με αποτέλεσμα να υπάρχει μεγάλη υποστήριξη για οποιοδήποτε πρόβλημα θα προέκυπτε.

2.2 Τεχνολογίες Front-End

Πλέον, υπάρχουν πολλά διαφορετικά frameworks τα οποία μπορούν να προσφέρουν τις απαραίτητες λειτουργίες και ταυτόχρονα να είναι εύκολα στην εκμάθηση και στην συντήρηση της εφαρμογής καθώς και στην επέκτασή της.

Για το front-end επιλέχθηκε το framework της Vue.js καθώς είναι ένα από τα πιο διαδεδομένα κατά την περίοδο συγγραφής και ήταν μια ευκαιρία να γνωρίσουμε μια νέα τεχνολογία.

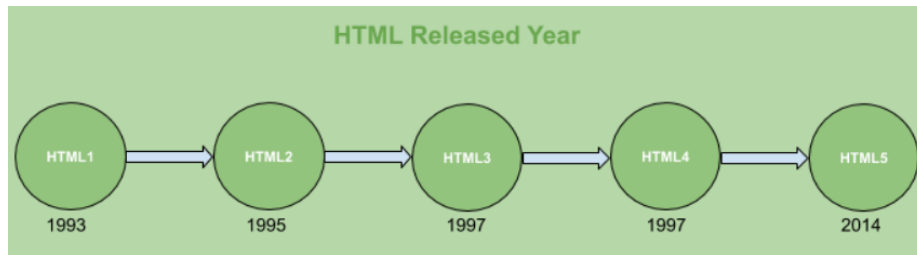
2.2.1 HTML/CSS/Javascript



Σχήμα 2.1: Λογότυπα HTML/CSS/JS

Η HTML (HyperText Markup Language) χρησιμοποιείται για τον σχεδιασμό ιστοσελίδων κάνοντας χρήση μιας γλώσσας σήμανσης. Η HTML είναι ο συνδυασμός υπερκειμένου και γλώσσας σήμανσης. Το υπερκείμενο ορίζει μια σύνδεση μεταξύ των ιστοσελίδων. Η γλώσσα σήμανσης χρησιμοποιείται για να ορίσει ένα έγγραφο κειμένου μέσα σε ετικέτες οι οποίες ορίζουν την δομή των ιστοσελίδων. Αυτή η γλώσσα χρησιμοποιείται για να σχολιάσει κείμενο, έτσι ώστε να γίνεται κατανοητή από ένα μηχάνημα και να μπορέσει να χειριστεί το κείμενο ανάλογα. Οι περισσότερες γλώσσες σήμανσης γίνονται εύκολα κατανοητές από τον άνθρωπο.

Η HTML χρησιμοποιείται από το πρόγραμμα περιήγησης για να χειρίζεται εικόνες, κείμενο και άλλου περιεχομένου, προκειμένου να το εμφανίσει στην απαιτούμενη μορφή. Δημιουργήθηκε από τον Tim Berners-Lee το 1991.



Σχήμα 2.2: HTML Released Year

Το Cascading Style Sheet, συχνά αναφέρεται ως CSS, είναι μία απλή γλώσσα σχεδιασμού με κύριο σκοπό να κάνει εμφανίσιμες τις ιστοσελίδες με απλό και κατανοητό τρόπο. Το CSS χειρίζεται το εμφανισιακό και το αισθητικό κομμάτι της ιστοσελίδας. Χρησιμοποιώντας το CSS, υπάρχει η δυνατότητα να ελέγχονται τα χρώματα του κειμένου, το στυλ των γραμματοσειρών, την απόσταση μεταξύ των παραγράφων, το μέγεθος και τη διάταξη των στηλών, να τεθεί εικόνα φόντου, να παραλλάξει η οθόνη για διαφορετικές συσκευές και μεγέθη οθόνης καθώς και να επιλεγεί μια ποικιλία εφέ.

Το CSS είναι εύκολο στην εκμάθηση και στην κατανόηση, και παρέχει ισχυρό έλεγχο στην παρουσίαση ενός εγγράφου HTML. Συνηθέστερα, το CSS συνδυάζεται με γλώσσες σήμανσης όπως HTML ή XHTML.

Η Javascript είναι μια αντικειμενοστρεφής γλώσσα σεναρίου σε πολλές πλατφόρμες που χρησιμοποιείται για να κάνει τις ιστοσελίδες διαδραστικές (π.χ. να έχει πολύπλοκα κινούμενα σχέδια, κουμπιά με δυνατότητα κλικ, αναδυόμενα μενού κ.λπ.). Υπάρχουν επίσης και πιο προηγμένες εκδόσεις Javascript από την πλευρά του διακομιστή, όπως είναι το Node.js, που επιτρέπει την προσθήκη περισσότερης λειτουργικότητας σε έναν ιστότοπο.

Η Javascript από την πλευρά του πελάτη επεκτείνει τη βασική γλώσσα παρέχοντας αντικείμενα για τον έλεγχο ενός προγράμματος περιήγησης και του DOM. Για παράδειγμα, οι επεκτάσεις από την πλευρά του πελάτη επιτρέπουν σε μια εφαρμογή να τοποθετεί στοιχεία σε μια φόρμα HTML και να ανταποκρίνεται σε συμβάντα χρήστη είναι η εισαγωγή φόρμας και η πλοήγηση στη σελίδα.

2.2.2 Bulma CSS Framework

Το CSS δίνει στους προγραμματιστές περισσότερο έλεγχο στη διάταξη, την τυπογραφία και την οπτική αισθητική μιας ιστοσελίδας. Στην πραγματικότητα, το CSS έχει γίνει τόσο διαδεδομένο στον σχεδιασμό ιστοσελίδων, που σχεδόν κάθε μία από τις περίπου πεντέμισι δισεκατομμύρια ιστοσελίδες στον Παγκόσμιο Ιστό ενσωματώνει CSS. Προφανώς, κάνει την δουλειά για την οποία δημιουργήθηκε. Αυτό όμως δεν σημαίνει ότι δεν μπορεί να βελτιωθεί.



Σχήμα 2.3: Λογότυπο Bulma

Εκεί έρχονται να βοηθήσουν τα CSS Frameworks. Το Bulma CSS Framework εξαλείφει μερικές από τις ταλαιπωρίες του CSS, παρέχοντας τυποποιημένα κριτήρια, έννοιες και πρακτικές για την αντιμετώπιση κοινών ζητημάτων που εντοπίζονται στην ανάπτυξη ιστοσελίδων. Τι είναι όμως το Bulma

Framework; Το Bulma είναι μια δωρεάν λύση CSS η οποία βασίζεται στη διάταξη Flexbox. Με το Bulma, η εκτεταμένη γκάμα των ενσωματωμένων χαρακτηριστικών σημαίνει γρηγορότερη υλοποίηση της εφαρμογής και λιγότερος κώδικας CSS.

Επίσης είναι πλήρως ανοιχτού κώδικα, και δεν υπάρχει όριο στο πόσο μπορεί να επεκταθεί η λειτουργικότητά του. Γιατί όμως να το χρησιμοποιήσει κάποιος;

Παρακάτω αναφέρονται μερικά από τα πλεονεκτήματά του:

- Εύκολο στην εκμάθηση
- Είναι modular καθώς επιλέγετε τι θα εισάγετε
- Περιέχει μόνο CSS, χωρίς την ύπαρξη της Javascript
- Είναι responsive για κάθε τύπο συσκευών
- Παρέχει ολοκληρωμένη τεκμηρίωση
- Παρέχει όλα τα στοιχεία που χρειάζεται ένας ιστότοπος
- Είναι συμβατό με όλους τους φυλλομετρητές
- Βελτιώνεται συνεχώς

2.2.3 npm



Σχήμα 2.4: Λογότυπο npm

Το npm (Node Package Manager) είναι το μεγαλύτερο μητρώο λογισμικού στον κόσμο. Οι προγραμματιστές ανοικτού κώδικα από όλο τον κόσμο χρησιμοποιούν το npm για να μοιράζονται και να δανείζονται πακέτα, και πολλοί οργανισμοί χρησιμοποιούν το npm για να διαχειρίζονται την ιδιωτική ανάπτυξη εφαρμογών.

Αποτελείται από τρία διακριτά στοιχεία:

1. Τον ιστότοπο - όπου υπάρχει η δυνατότητα ανακάλυψης πακέτων, ρύθμισης του προφίλ και διαχείρισης άλλων πτυχών της εμπειρίας με το npm
2. Τη διεπαφή γραμμής εντολών (CLI) - όπου εκτελείται από ένα τερματικό και έτσι αλληλοεπιδρούν οι περισσότεροι προγραμματιστές με το npm
3. Το μητρώο - το οποίο είναι μια μεγάλη δημόσια βάση δεδομένων λογισμικού JS και των μετα-πληροφοριών που το περιβάλλουν

2.2.4 Vue.js



Σχήμα 2.5: Λογότυπο Vue.js

Το Vue.js είναι ένα προοδευτικό Javascript framework ανοιχτού κώδικα που χρησιμοποιείται για την ανάπτυξη διαδραστικών διεπαφών ιστού. Πλέον είναι ένα από τα πιο δημοφιλέστερα frameworks που

χρησιμοποιούνται για την απλοποίηση της ανάπτυξης ιστοσελίδων. Εστιάζει στο επίπεδο της εμφάνισης. Ενσωματώνεται εύκολα σε μεγάλα project χωρίς πολλά προβλήματα.

Η εγκατάσταση για το Vue.js είναι αρκετά απλή. Οποιοσδήποτε προγραμματιστής μπορεί εύκολα να κατανοήσει και να δημιουργήσει διαδραστικές διεπαφές ιστού σε σύντομο χρονικό διάστημα. Δημιουργήθηκε από τον Evan You, πρώην υπάλληλος της Google. Η πρώτη του έκδοση κυκλοφόρησε τον Φεβρουάριο του 2014. Η πιο πρόσφατη σταθερή έκδοση κατά τη συγγραφή του παρόντος είναι η v3.2.23.

2.2.4.1 Κύρια χαρακτηριστικά

Virtual DOM

Το framework της Vue.js κάνει χρήση του εικονικού DOM, το οποίο επίσης χρησιμοποιείται και από άλλα frameworks όπως είναι η React, το Ember, κτλ. Οι αλλαγές δεν γίνονται απευθείας στο DOM, αντ' αυτού δημιουργείται ένα αντίγραφο του DOM το οποίο παρουσιάζεται ως μια μορφή δομών δεδομένων Javascript. Οι αλλαγές γίνονται εκεί πέρα και συγκρίνονται στην συνέχεια με την αρχική δομή δεδομένων. Τέλος οι αλλαγές ενημερώνονται στο αρχικό DOM, το οποίο βλέπει και ο χρήστης. Αυτό βοηθάει στο να γίνουν πιο γρήγορα οι αλλαγές, καταναλώνει λιγότερο χρόνο στη CPU και βοηθά στην βελτιστοποίηση.

Data Binding

Το χαρακτηριστικό της δέσμευσης δεδομένων (data binding) βοηθάει στο να χειριστεί ή να εκχωρήσει τιμές στα χαρακτηριστικά της HTML, να αλλάξει το στυλ και να εκχωρήσει κλάσεις με τη βοήθεια του v-bind που είναι διαθέσιμο με το Vue.js.

Event Handling

Με την οδηγία v-on τα διάφορα στοιχεία της HTML περιμένουν συγκεκριμένα events για να γίνει trigger ένα κομμάτι κώδικα. Κάποια παραδείγματα είναι το Click του ποντικιού, το πάτημα ενός πλήκτρου, κτλ.

Computed Properties

Ένα από τα σημαντικότερα χαρακτηριστικά του Vue.js. Δίνει τη δυνατότητα στον χρήστη να ακούει τις αλλαγές που έγιναν στη διεπαφή του χρήστη και εκτελεί του απαραίτητους υπολογισμούς. Δεν χρειάζεται περαιτέρω κώδικας για να λειτουργήσει.

Templates

Το Vue.js προσφέρει πρότυπα βασισμένα σε HTML τα οποία συνδέουν το DOM με τα δεδομένα του Vue Instance. Το Vue μεταγλωττίζει τα πρότυπα σε συναρτήσεις απόδοσης (render functions) του εικονικού DOM.

Directives

Το Vue.js δίνει την δυνατότητα χρήσης ενσωματωμένων οδηγιών, όπως είναι το v-if, v-else, v-show, v-on, v-bind και το v-model, τα οποία χρησιμοποιούνται για να εκτελέσουν διάφορες ενέργειες στο frontend.

Watchers

Ο παρατηρητής (Watcher) κάνει ακριβώς αυτό που λέει, παρατηρεί για τυχόν αλλαγές στα δεδομένα.

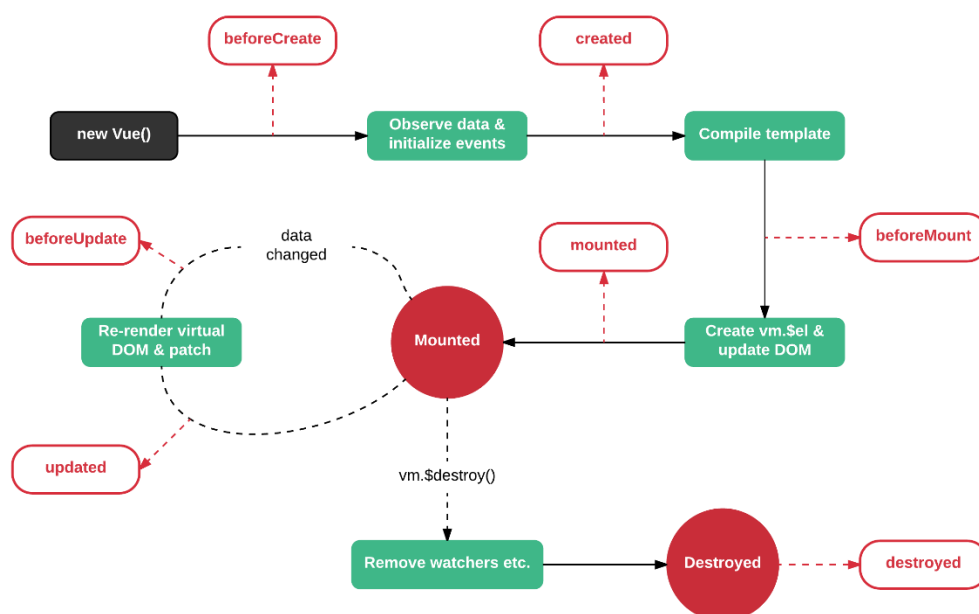
Routing

Η πλοήγηση μεταξύ των σελίδων γίνεται με την βοήθεια του vue-router.

Instance Lifecycle Hooks

Κάθε Vue instance περνάει από διάφορα στάδια όταν δημιουργείται. Τα στάδια είναι τα εξής:

1. beforeCreate - Καλείται συγχρονισμένα αμέσως μετά την προετοιμασία του στιγμιότυπου (instance), πριν από την παρατήρηση δεδομένων και τη ρύθμιση συμβάντος/παρατηρητή.
2. created - Καλείται συγχρονισμένα μετά τη δημιουργία του στιγμιότυπου. Έχουν ρυθμιστεί τα εξής: παρατήρηση δεδομένων, computed ιδιότητες, μέθοδοι, επιστροφές κλήσης παρακολούθησης/συμβάντος. Ωστόσο η φάση τοποθέτησης δεν έχει ξεκινήσει και η ιδιότητα \$el δεν είναι διαθέσιμη.
3. beforeMount - Καλείται πριν ξεκινήσει το mounting: η συνάρτηση απόδοσης πρόκειται να κληθεί για πρώτη φορά.
4. mounted - Καλείται αφού το στιγμιότυπο έχει γίνει mounted.
5. beforeUpdate - Καλείται όταν αλλάζουν τα δεδομένα, προτού αλλάξει το DOM.
6. updated - Η κλήση μετά από μια αλλαγή δεδομένων προκαλεί την εκ νέου απόδοση και επιδιόρθωση του εικονικού DOM.
7. beforeDestroy - Καλείται ακριβώς πριν την καταστροφή ενός στιγμιότυπου Vue. Σε αυτό το στάδιο το στιγμιότυπο ακόμα είναι πλήρως λειτουργικό.
8. destroyed - Καλείται μετά την καταστροφή ενός στιγμιότυπου Vue.



Σχήμα 2.6: Κύκλος ζωής ενός Vue Instance

Components

Ένα σημαντικό χαρακτηριστικό του Vue είναι η δυνατότητα χρήσης components. Τα components είναι επαναχρησιμοποιήσιμα στιγμιότυπα Vue με προσαρμοσμένα στοιχεία HTML. Μπορούν να επαναχρησιμοποιηθούν όσες φορές χρειαστεί ή και να χρησιμοποιηθούν σε άλλο component, καθιστώντας το θυγατρικό στοιχείο. Δεδομένα, computed properties, παρατηρητές, και μέθοδοι μπορούν να χρησιμοποιηθούν μέσα σε Vue component.

```
<template>
  <a>
  </a>
</template>
```

```
<script>
export default {
  props: {
    id: {
      type: Number,
      required: true
    }
  },
  data: function() {
    return {};
  }
};
</script>
```

Σχήμα 2.7: Παράδειγμα ενός Vue Component

Props

Στο Vue.js, οι ιδιότητες (props) είναι ένας τρόπος να περάσετε δεδομένα από ένα γονικό στοιχείο στα θυγατρικά του.

Μία εφαρμογή αποτελούμενη από στοιχεία έχει μία δομή δεδομένων σαν δέντρο. Αποτελείται από:

1. γονείς
2. παιδιά
3. προγόνους
4. απογόνους

Τα δεδομένα ρέουν από το ριζικό στοιχείο, αυτό που βρίσκεται στην κορυφή, και μεταδίδονται μέσω των ιδιοτήτων στα παιδιά.

```
<template>
  <my-component v-bind:msg="hello world"></my-component>
</template>
```

Σχήμα 2.8: Παράδειγμα εισαγωγής prop σε Vue Component

Όμως, όταν υπάρχει πρόσβαση στις ιδιότητες μέσα από ένα στοιχείο, δεν μπορούν να αλλάξουν τα δεδομένα (όπως δεν μπορούν να αλλάξουν τα γονίδια που δώθηκαν από τους γονείς στα παιδιά). Παρόλο που είναι εφικτό να αλλάξουν τα δεδομένα μέσα σε ένα στοιχείο, είναι μία κακή τακτική. Καταλήγει να αλλάζει και η τιμή του γονιού, το οποίο μπορεί να προκαλέσει σύγχυση.

Client-Side Storage

Η αποθήκευση από την πλευρά του πελάτη είναι ένας εξαιρετικός τρόπος να αυξηθεί η απόδοση σε μια εφαρμογή. Αποθηκεύοντας δεδομένα στο ίδιο το πρόγραμμα περιήγησης, παραλείπεται η λήψη πληροφοριών από τον διακομιστή κάθε φορά που ο χρήστης τις χρειάζεται. Ενώ είναι ιδιαίτερα χρήσιμο όταν ο χρήστης βρίσκεται εκτός σύνδεσης, ακόμη και οι διαδικτυακοί χρήστες θα επωφεληθούν από τη χρήση τοπικής αποθήκευσης.

Η αποθήκευση από την πλευρά του πελάτη μπορεί να γίνει με τη χρήση Cookie, Local Storage, IndexedDB και WebSQL.

LocalStorage

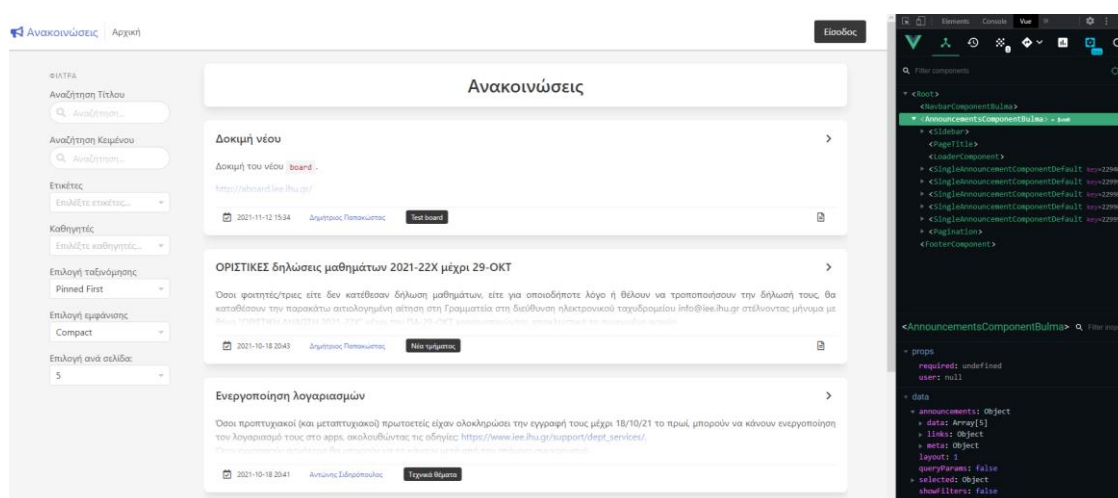
Στην εργασία αυτή έγινε χρήση του Local Storage, ο απλούστερος μηχανισμός αποθήκευσης. Χρησιμοποιεί ένα σύστημα κλειδιού/τιμής για την αποθήκευση δεδομένων. Αποθηκεύει απλές τιμές ή

διαφορετικά για σύνθετα δεδομένα γίνεται χρήση JSON. Είναι κατάλληλος για προτιμήσεις χρηστών, όπου και χρησιμοποιήθηκαν στην συγκεκριμένη εργασία.

2.2.4.2 Vue DevTools

Τα DevTools είναι ένα σύνολο βοηθητικών εργαλείων που προορίζονται να βοηθήσουν τους προγραμματιστές στη διαδικασία της δοκιμής, καθώς και στον εντοπισμό σφαλμάτων στον κώδικά τους. Αυτά τα εργαλεία διαφέρουν από τα IDE και άλλα προγράμματα δημιουργίας ιστοτόπων. Δεν προορίζονται για τη δημιουργία μιας ιστοσελίδας, αντίθετα αυτά τα εργαλεία βοηθούν στη δοκιμή μιας ιστοσελίδας.

Υπάρχουν διάφορα είδη DevTools (πχ. Chrome DevTools) τα οποία μπορούν να χρησιμοποιηθούν για τους προγραμματιστές ιστού. Για τους προγραμματιστές του Vue.js, υπάρχει το Vue DevTools, το οποίο βοηθά στον εντοπισμό σφαλμάτων στην εφαρμογή μας.



Σχήμα 2.9: Παράδειγμα χρήσης Vue DevTools

Για να χρησιμοποιηθεί το Vue DevTools, πρέπει η εφαρμογή να βρίσκεται σε Development Mode και να εγκατασταθεί το πρόγραμμα. Η εγκατάσταση μπορεί να γίνει είτε μέσω ενός φυλλομετρητή είτε κατεβάζοντας την standalone electron εφαρμογή. Το Vue DevTools παρέχει έλεγχο των component, του state και των event.

Στην καρτέλα Components δείχνει όλα τα components που χρησιμοποιούνται σε μια σελίδα, μαζί με τις ιδιότητες των δεδομένων και τις τιμές των prop.

Στην καρτέλα Events εμφανίζει τα συμβάντα που καταγράφονται. Η επιλογή ενός συμβάντος εμφανίζει τις πληροφορίες του. Υπάρχει η δυνατότητα φιλτραρίσματος και παύσης για να σταματήσει η καταγραφή των συμβάντων πατώντας το κουμπί εγγραφής.

Τέλος στην καρτέλα Vuex εμφανίζει τις μεταλλάξεις που έχουν συμβεί. Επιλέγοντας ένα θα εμφανιστούν πληροφορίες σχετικά με αυτήν τη μετάλλαξη. Όταν επιλεγεί ένα, υπάρχει η δυνατότητα επαναφοράς της κατάστασης στην οποία έγινε αυτή η ενέργεια.

2.3 Τεχνολογίες Back-End

Το Back-end αποτελεί το αντικείμενο με το οποίο επικοινωνεί η εφαρμογή με την βάση και τον σέρβερ. Επίσης συγκρατεί τα δεδομένα μας σε μία βάση ώστε να χρησιμοποιηθούν περαιτέρω. Όπως αναφέρουμε και παραπάνω αναλάβαμε να βελτιώσουμε την ιστοσελίδα aboard χρησιμοποιώντας τις τεχνολογίες που έχουν ήδη χρησιμοποιηθεί για την δημιουργία του.

Οι τεχνολογίες αυτές είναι οι εξής: Η χρήση της γλώσσας PHP σε συνδυασμό με ενός από τα καλύτερα και πιο διαδεδομένα framework, της Laravel.

2.3.1 PHP

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία ιστοσελίδων με δυναμικό περιεχόμενο, είναι μία ανοικτού κώδικα γλώσσα γενικού σκοπού και αποτελεί μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού για τη δημιουργία διαδικτυακών εφαρμογών καθώς ένα μεγάλο μέρος του παγκόσμιου ιστού είναι γραμμένο σε PHP.



Σχήμα 2.10: Λογότυπο PHP

Η PHP αποτελείται από τα εξής χαρακτηριστικά:

1. Περιλαμβάνει πολλαπλές βιβλιοθήκες
2. Έχει μεγάλη προγραμματιστική κοινότητα
3. Διαθέτει μεγάλη ιστορία
4. Παράγει δυναμικό περιεχόμενο
5. Έχει έλλειψη αυστηρού ελέγχου κώδικα

Χρησιμοποιείται κυρίως για εφαρμογές από πλευράς του διακομιστή αλλά και για command-line scripting. Υποστηρίζει τα πιο διαδεδομένα λειτουργικά συστήματα που υπάρχουν (windows, linux, macOS) καθώς και τους πιο διαδεδομένους διακομιστές ιστού που υπάρχουν (Apache web server, IIS, Nginx, lighttpd) γι' αυτό το λόγο κάποιες από τις μεγαλύτερες εταιρείες όπως το Facebook χρησιμοποιούν PHP.

2.3.1.1 PHP-FPM

Το PHP-FPM (FastCGI Process Manager) είναι μια δημοφιλής εναλλακτική εφαρμογή FastCGI η οποία αυξάνει δραματικά την απόδοση ενός περιβάλλοντος PHP. Το FPM περιλαμβάνει πολλές δυνατότητες που μπορούν να αποδειχθούν ωφέλιμες για ιστοτόπους που λαμβάνουν συχνά επισκεψιμότητα σε μεγάλο όγκο. Οι δυνατότητες αυτές χωρίζονται ως εξής:

- Δυνατότητα έναρξης workers χρησιμοποιώντας διάφορα uid/gid/chroot/environment και του php.ini, που αντικαθιστά την ασφαλή λειτουργία που μπορεί να περιμένουν οι χρήστες
- Διαχείριση σε βάθος για απλή εκκίνηση και διακοπή
- Καταγραφή stdout and stderr
- Δυνατότητα επανεκκίνησης εκτατής ανάγκης στην περίπτωση που καταστραφεί κατά λάθος η cache
- Η υποστήριξη για το upload είναι γρηγορότερη
- Βασισμένο σε αρχεία διαμόρφωσης php.ini
- Διαμόρφωση μεταβλητής αργής καταγραφής για τον εντοπισμό συναρτήσεων που χρειάζονται περισσότερο χρόνο από το συνηθισμένο για να εκτελεστούν

- FastCGI βελτιώσεις με ειδική λειτουργία για διακοπή και λήψη δεδομένων κατά την ολοκλήρωση μακρών διαδικασιών (π.χ. επεξεργασία στατιστικών στοιχείων)
- Διατίθενται βασικά στατιστικά στοιχεία, παρόμοια με τη μονάδα mod-status στο Apache

2.3.1.2 Nginx και PHP-FPM: ο ιδανικός συνδυασμός

Το Nginx είναι ο ιδανικός συνδυασμός με το PHP-FPM, καθώς είναι ένας σταθερός διακομιστής ιστού που αναγνωρίζεται για την εντυπωσιακή του απόδοση και τη χαμηλή κατανάλωση πόρων.

Διαθέτει μια ασύγχρονη δομή που είναι εξαιρετικά επεκτάσιμη, σύμφωνα με τα γεγονότα. Επιπλέον, η απόδοση της κατανάλωσης μνήμης είναι σημαντικά καλύτερη όταν χρησιμοποιείτε ο Nginx και το PHP-FPM μαζί.

Η PHP εκτελείται ως απομονωμένη υπηρεσία όταν χρησιμοποιείτε το PHP-FPM. Η χρήση αυτής της έκδοσης PHP ως διερμηνέα γλώσσας σημαίνει ότι τα αιτήματα θα υποβάλλονται σε επεξεργασία μέσω μιας υποδοχής TCP/IP και ο διακομιστής Nginx διαχειρίζεται μόνο αιτήματα HTTP, ενώ το PHP-FPM ερμηνεύει τον κώδικα PHP. Η αξιοποίηση δύο ξεχωριστών υπηρεσιών είναι ζωτικής σημασίας για την αύξηση της αποτελεσματικότητας.

2.3.2 Laravel

Η Laravel είναι ένα PHP framework ανοιχτού κώδικα, το οποίο είναι ισχυρό και εύκολο στην κατανόηση και είναι βασισμένο στο Symfony. Ακολουθεί το μοτίβο σχεδίασης model-view-controller. Επαναχρησιμοποιεί τα υπάρχοντα components διαφορετικών frameworks που βοηθούν στην ανάπτυξη διαδικτυακών εφαρμογών.

Η Laravel προσφέρει ένα πλούσιο σύνολο λειτουργιών που ενσωματώνει τα βασικά χαρακτηριστικά των πλαισίων PHP όπως είναι το CodeIgniter, το Yii και άλλες γλώσσες προγραμματισμού όπως η Ruby on Rails. Επίσης παρέχει και ένα πολύ πλούσιο σύνολο δυνατοτήτων που θα ενισχύσουν την ταχύτητα της ανάπτυξης ιστού.



Σχήμα 2.11: Λογότυπο Laravel

Όπως αναφέραμε και πιο πάνω η Laravel είναι βασισμένη στο Symfony. Το Symfony είναι ένα σετ από components και βιβλιοθήκες που έχει ως στόχο την ταχύτερη δημιουργία και συντήρηση διαδικτυακών εφαρμογών. Προωθεί την επαναχρησιμοποίηση κώδικα, δίνει πολλές ελευθερίες στον προγραμματιστή (π.χ. οργάνωση των φακέλων και βιβλιοθηκών) και επιτρέπει την πλήρη παραμετροποίηση του project.

2.3.2.1 Κύκλος ζωής της Laravel

Auto Loader

Το αρχικό σημείο για όλα τα αιτήματα σε μία εφαρμογή Laravel είναι το αρχείο **index.php** που βρίσκεται στον φάκελο **public**. Όλα τα αιτήματα κατευθύνονται σε αυτό το αρχείο σύμφωνα με το configuration του διακομιστή που χρησιμοποιείται. Το αρχείο αυτό δεν περιλαμβάνει πολύ κώδικα, είναι απλώς ένα σημείο εκκίνησης για τη φόρτωση του υπόλοιπου framework.

Φορτώνει τα αρχεία αυτόματης φόρτωσης που δημιουργούνται από τον composer.

Στη συνέχεια ανακτά ένα στιγμιότυπο της εφαρμογής από το σενάριο **bootstrap/app.php**.

Η πρώτη ενέργεια που έκανε η ίδια η Laravel είναι να δημιουργήσει ένα στιγμιότυπο της εφαρμογής.

Kernel

Το επόμενο βήμα θα συμβεί στο τμήμα του πυρήνα της εφαρμογής. Το εισερχόμενο αίτημα αποστέλλεται είτε στον πυρήνα HTTP (HTTP Kernel) είτε στον πυρήνα της κονσόλας (Console Kernel), ανάλογα με τον τύπο του αιτήματος που εισέρχεται στην εφαρμογή. Οι δύο αυτοί πυρήνες χρησιμεύουν ως η κεντρική τοποθεσία από την οποία ρέουν όλα τα αιτήματα.

Ο πυρήνας HTTP, ο οποίος τοποθετείται στο **app/Http/Kernel.php**, απλώς λαμβάνει ένα αίτημα και επιστρέφει μία απάντηση. Οι Bootstrappers, που ορίζονται από την κλάση Kernel, διαμορφώνουν τον χειρισμό σφαλμάτων, διαμορφώνουν τα logs, εντοπίζουν το περιβάλλον της εφαρμογής και εκτελούν εργασίες που πρέπει να γίνουν πριν την διαχείριση του αιτήματος. Συνήθως, αυτές οι κλάσεις χειρίζονται εσωτερικές ρυθμίσεις της Laravel για τις οποίες δεν χρειάζεται να εμπλακεί ο χρήστης.

Ο πυρήνας HTTP θα καθορίσει τη λίστα των middleware που διαβιβάζονται πριν από την διαχείριση από την εφαρμογή.

Service Providers

Το επόμενο βήμα του πυρήνα είναι να φορτώσει τους παρόχους υπηρεσιών ως μέρος της ενέργειας bootstrapping. Οι πάροχοι που χρειάζονται για την εφαρμογή τοποθετούνται στο αρχείο διαμόρφωσης **config/app.php**.

Καθώς καλείται η μέθοδος εγγραφής, όλοι οι πάροχοι θα εγγραφούν. Μόλις εγγραφούν όλοι οι πάροχοι, θα κληθεί η μέθοδος εκκίνησης.

Dispatch Request

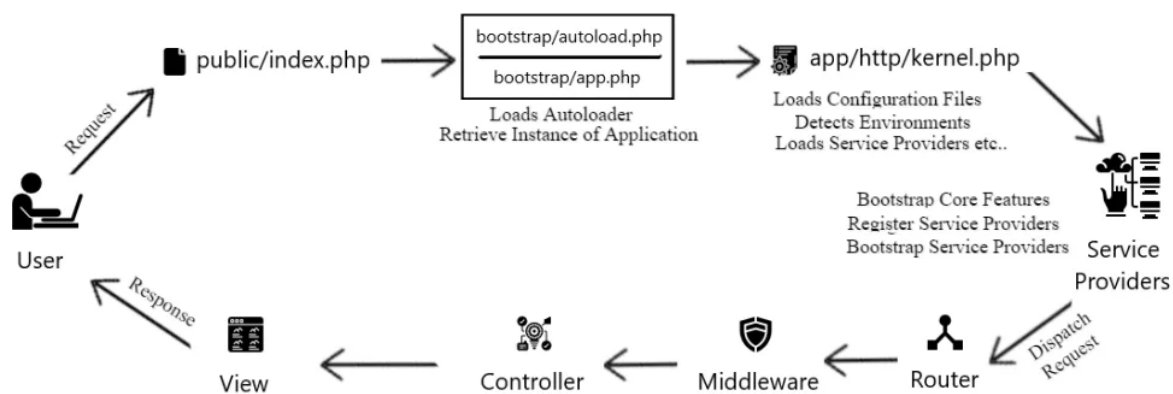
Μόλις ολοκληρωθεί η εκκίνηση της εφαρμογής και εγγραφούν και εκκινηθούν όλοι οι πάροχοι υπηρεσιών, το αίτημα παραδίδεται στον δρομολογητή για αποστολή. Ο δρομολογητής θα αποστείλει το αίτημα σε μία διαδρομή ή σε έναν ελεγκτή (controller), καθώς και θα εκτελέσει οποιοδήποτε middleware της συγκεκριμένης διαδρομής.

Router

Τώρα το αίτημα θα αποσταλεί από τον δρομολογητή και θα καταλήξει με τις προβολές (Views) όπως φαίνονται παρακάτω:

1. Ο δρομολογητής θα κατευθύνει το αίτημα HTTP σε έναν ελεγκτή ή θα επιστρέψει μία προβολή ή θα απαντήσει απευθείας παραλείποντας τον ελεγκτή. Αυτές οι διαδρομές θα τοποθετηθούν στο **app/routes.php**.
2. Ο Controller εκτελεί συγκεκριμένες ενέργειες και στέλνει δεδομένα σε μια προβολή.
3. Η προβολή μορφοποιεί κατάλληλα τα δεδομένα, παρέχοντας το HTTP Response.

Όλα τα παραπάνω βήματα εξηγούνται στην παρακάτω διαγραμματική προβολή.



Σχήμα 2.12: Κύκλος ζωής ενός request στην Laravel

2.3.2.2 Πλεονεκτήματα της Laravel

Η Laravel προσφέρει τα ακόλουθα πλεονεκτήματα:

- Η διαδικτυακή εφαρμογή γίνεται πιο επεκτάσιμη.
- Εξοικονομείται σημαντικός χρόνος στη σχεδίαση της διαδικτυακής εφαρμογής, αφού η Laravel επαναχρησιμοποιεί τα components από άλλα frameworks για την ανάπτυξη διαδικτυακών εφαρμογών.
- Περιλαμβάνει namespaces και διεπαφές, επομένως βοηθά στην οργάνωση και διαχείριση των πόρων.
- Παρέχεται προστασία CSRF
- Υποστήριξη API

Composer

Το Composer είναι ένα εργαλείο που περιλαμβάνει όλα τα dependencies και τις βιβλιοθήκες. Οι βιβλιοθήκες τρίτων μπορούν να εγκατασταθούν εύκολα με τη βοήθειά του.

Όλα τα dependencies σημειώνονται στο αρχείο composer.json που τοποθετείται στον φάκελο προέλευσης.

Artisan

Η διεπαφή γραμμής εντολών που χρησιμοποιείται στη Laravel ονομάζεται Artisan. Παρέχει ένα σύνολο εντολών στον προγραμματιστή που τον βοηθά στη δημιουργία της εφαρμογής. Για παράδειγμα μπορεί να ελέγχει τα database migrations, το middleware καθώς και τις περισσότερες λειτουργίες της εφαρμογής.

2.3.2.3 Χαρακτηριστικά της Laravel

Η Laravel προσφέρει τα ακόλουθα βασικά χαρακτηριστικά που την καθιστούν ιδανική επιλογή για το σχεδιασμό διαδικτυακών εφαρμογών.

Modularity

Παρέχει 20 ενσωματωμένες βιβλιοθήκες και ενότητες που βοηθούν στη βελτίωση της εφαρμογής. Κάθε λειτουργική μονάδα είναι ενσωματωμένη με το Composer Dependency Manager που διευκολύνει τις ενημερώσεις.

Testability

Περιλαμβάνει χαρακτηριστικά και βοηθούς που βοηθούν στον έλεγχο σε διάφορες περιπτώσεις δοκιμών. Αυτή η δυνατότητα βοηθά στη διατήρηση του κώδικα σύμφωνα με τις απαιτήσεις.

Routing

Παρέχει μια ευέλικτη προσέγγιση στον χρήστη για να ορίσει διαδρομές στην διαδικτυακή εφαρμογή. Η δρομολόγηση βοηθά στην καλύτερη κλιμάκωση της εφαρμογής και αυξάνει την απόδοσή της. Οι διαδρομές αυτές μπορούν να προστατευτούν αναλόγως με την κρίση του προγραμματιστή.

Query Builder και ORM

Ενσωματώνει ένα πρόγραμμα δημιουργίας query το οποίο βοηθά στην αναζήτηση βάσεων δεδομένων χρησιμοποιώντας μεθόδους αλυσίδας. Παρέχει υλοποίηση ORM (Object Relational Mapper) και ActiveRecord που ονομάζεται Eloquent. Το Eloquent ORM αποτελεί τον μηχανισμό που αντιστοιχεί τα μοντέλα της εφαρμογής σε πίνακες στη βάση.

Schema Builder

Το Schema Builder διατηρεί τους ορισμούς και το σχήμα της βάσης δεδομένων σε κώδικα PHP. Διατηρεί επίσης μια παρακολούθηση αλλαγών σε σχέση με τις μετεγκαταστάσεις της βάσης δεδομένων.

Template Engine

Χρησιμοποιεί το Blade Template Engine, μία ελαφριά γλώσσα προτύπου που χρησιμοποιείται για το σχεδιασμό ιεραρχικών μπλοκ και διατάξεων με προκαθορισμένα μπλοκ που περιλαμβάνουν δυναμικό περιεχόμενο.

E-mail

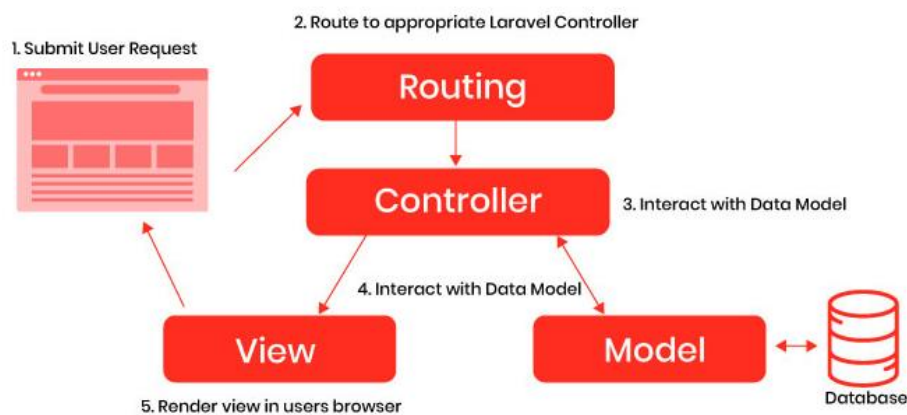
Περιλαμβάνει μια κλάση αλληλογραφίας που βοηθάει στην αποστολή αλληλογραφίας με πλούσιο περιεχόμενο και συνημμένα από την διαδικτυακή εφαρμογή.

Authentication

Η ταυτοποίηση του χρήστη είναι μια κοινή δυνατότητα σε διαδικτυακές εφαρμογές. Η Laravel διευκολύνει σχεδιασμό της ταυτοποίησης καθώς περιλαμβάνει λειτουργίες όπως εγγραφή, ανάκτηση κωδικού πρόσβασης και αποστολή υπενθυμίσεων κωδικού πρόσβασης.

2.3.2.4 Laravel και MVC

Όπως αναφέραμε και παραπάνω η Laravel υλοποιεί το μοτίβο MVC. Στη Laravel, το Model αναπαρίσταται ως μία κλάση της PHP η οποία έχει ιδιότητες που αντιστοιχίζονται με τα ονόματα των στηλών στη βάση. Για να δημιουργήσουμε ένα καινούριο μοντέλο, χρειάζεται να τρέξουμε την εντολή ``php artisan make:model Announcements``. Με την εκτέλεση της εντολής δημιουργείται στον φάκελο app μια κλάση με όνομα Announcements.



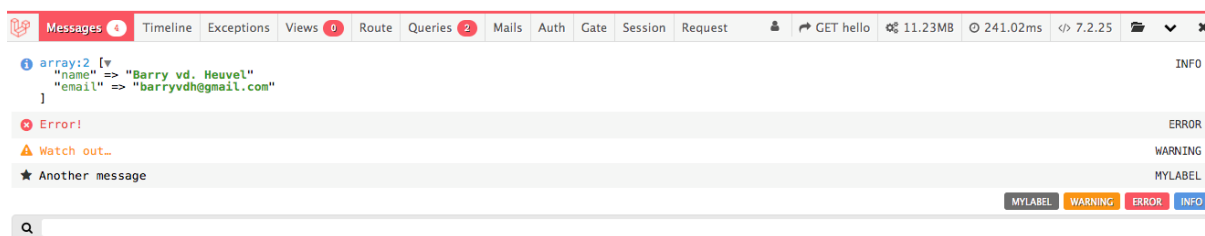
Σχήμα 2.13: Αρχιτεκτονική MVC στην Laravel

Προχωρώντας στο κομμάτι του Controller, η Laravel δίνει την δυνατότητα του Resource Controller. Κατά τη δημιουργία του Resource Controller δημιουργούνται οι τυπικές διαδρομές CRUD (Create, Read, Update, Delete) σε έναν controller με μία εντολή. Για να γίνει αυτό θα πρέπει να τρέξουμε την εντολή `php artisan make: controller AnnouncementController -resource`. Με την εκτέλεση της εντολής δημιουργείται στον φάκελο `app/Http/Controllers` ένα αρχείο με όνομα `AnnouncementController`.

Τέλος για το κομμάτι του View, η Laravel χρησιμοποιεί το Blade Template όπως προαναφέραμε. Η κλήση κάποιου view, γίνεται είτε μέσω των διαδρομών είτε μέσω των controllers. Τα views αποθηκεύονται στο φάκελο `resources/views`.

2.3.2.5 Laravel DebugBar

Το Laravel DebugBar είναι ένα πακέτο που σας επιτρέπει να παρακολουθείτε γρήγορα και εύκολα την εφαρμογή σας στο Framework της Laravel κατά τη διάρκεια της ανάπτυξης της. Με απλή εγκατάσταση και ισχυρά εργαλεία, το πακέτο του DebugBar αποτελεί ακρογωνιαίο λίθο για την ανάπτυξη μιας εφαρμογής.



Σχήμα 2.14: Παράδειγμα Laravel DebugBar

Ένα από τα σημαντικότερα εργαλεία του είναι οι collectors. Οι collectors στην ουσία είναι κάποια tabs στο interface που βοηθούν τον προγραμματιστή να συλλέξει πληροφορίες για πολλές διαφορετικές πτυχές της εφαρμογής για debugging. Μερικά από αυτά είναι τα εξής:

1. QueryCollector: εμφάνιση όλων των queries που έτρεξαν
2. RouteCollector: εμφάνιση πληροφοριών σχετικά με το τρέχων route
3. ViewCollector: εμφάνιση των τρεχόντων views
4. EventsCollector: εμφάνιση όλων των events
5. LaravelCollector: περιέχει την έκδοση και το περιβάλλον της Laravel
6. SymfonyRequestCollector: αντικαθιστά το RequestController με περισσότερες πληροφορίες σχετικά με το request/response
7. LogsCollector: εμφάνιση των πιο πρόσφατων καταχωρήσεων (logs)
8. FilesCollector: εμφάνιση των αρχείων που περιλαμβάνονται/απαιτούνται από την PHP
9. ConfigCollector: εμφανίζει τις τιμές από τα αρχεία διαμόρφωσης (config)
10. CacheCollector: εμφανίζει ό,τι έχει αποθηκευμένο στην cache

Για την εγκατάσταση αυτού του πακέτου θα πρέπει μέσω του Composer που αναφέρουμε παρακάτω να εκτελέσουμε την εξής εντολή: `composer require barryvdh/laravel-debugbar --dev`. Επίσης η εφαρμογή μας θα πρέπει να έχει ενεργοποιημένο το `APP_DEBUG` στο αρχείο `.env`.

2.3.3 Composer

Ο Composer είναι ένας διαχειριστής πακέτων σε επίπεδο εφαρμογής για την PHP. Επιτρέπει στον προγραμματιστή να θέσει σε λειτουργία βιβλιοθήκες τις οποίες χρησιμοποιεί και διαχειρίζεται. Τα dependencies είναι διαχειρίσιμα σε επίπεδο πρότζεκτ και αποθηκεύονται σε έναν φάκελο (συνήθως `vendor`). Επιτρέπει επίσης τη δήλωση συγκεκριμένων εκδόσεων των βιβλιοθηκών και όχι απαραίτητα την τελευταία έκδοση για αποφυγή ασυμβατότητας με το εκάστοτε πρότζεκτ.



Σχήμα 2.15: Λογότυπο Composer

Για την εγκατάστασή του απαιτείται μόνο ένα εκτελέσιμο PHP, κάτι που επιτρέπει την εγκατάστασή του σε όσα λειτουργικά συστήματα υποστηρίζουν PHP. Όπως προαναφέραμε η διαχείριση των dependencies γίνεται μέσω ενός αρχείου με όνομα `composer.json`, το οποίο περιέχει τις βιβλιοθήκες με τις εκδόσεις τους. Υποστηρίζει κάποια από τα πιο διαδεδομένα framework που υπάρχουν όπως: το Laravel, Symfony, CodeIgniter, CakePHP κ.α.

Χρήσιμες εντολές:

- **install:** εγκαθιστά τις βιβλιοθήκες τις επιλογής μας
- **remove:** απεγκαθιστά μία βιβλιοθήκη και την αφαιρεί από το `composer.json`
- **require:** προσθέτει τη βιβλιοθήκη στο `composer.json` και την εγκαθιστά
- **update:** αναβαθμίζει όλες τις βιβλιοθήκες σύμφωνα με το `composer.json` και τους εκάστοτε περιορισμούς

2.3.4 API

Η Διεπαφή Προγραμματισμού Εφαρμογών (API), είναι η διεπαφή των προγραμματιστικών διαδικασιών που παρέχει ένα λειτουργικό σύστημα, μία βιβλιοθήκη ή μία εφαρμογή προκειμένου να επιτρέπει να γίνονται προς αυτά αιτήσεις από άλλα προγράμματα ή και ανταλλαγή δεδομένων. Βασικός σκοπός ενός API είναι να ορίζει και να διατυπώνει το σύνολο των λειτουργιών-υπηρεσιών που μπορεί να παρέχει μια βιβλιοθήκη ή ένα λειτουργικό σύστημα σε άλλα προγράμματα.



Σχήμα 2.16: Λογότυπο API

Τύποι API με βάση την προσβασιμότητα

1. Ανοιχτά API (Open APIs): Είναι ανοιχτά προς το κοινό, γι' αυτό είναι και γνωστά ως Δημόσια APIs, και δεν υπάρχουν περιορισμοί πρόσβασης.
2. API συνεργατών (Partner APIs): Οι προγραμματιστές χρειάζονται δικαιώματα ή άδειες για να έχουν πρόσβαση σε αυτόν τον τύπο API, επειδή δεν είναι διαθέσιμα προς το κοινό.

3. Εσωτερικά API (Internal APIs): Είναι συνήθως σχεδιασμένα για εσωτερική χρήση σε έναν οργανισμό και χρησιμοποιούνται μόνο μεταξύ των διαφόρων εσωτερικών ομάδων. Γνωστά και ως Ιδιωτικά (Private) APIs.
4. Σύνθετα API (Composite APIs): Συνδυάζουν διαφορετικά μεταξύ τους APIs για την επιτάχυνση της διαδικασίας εκτέλεσης και τη βελτίωση της απόδοσης των ακροατών στις διεπαφές Ιστού.

Υπηρεσίες Ιστού (Web Services)

Οι υπηρεσίες ιστού είναι ένα μέσο με το οποίο μπορεί να επικοινωνήσει η εφαρμογή του πελάτη με τον διακομιστή. Αποτελούν μια ενότητα που έχει σχεδιαστεί με κύριο σκοπό να εκτελεί ένα συγκεκριμένο σύνολο εργασιών.

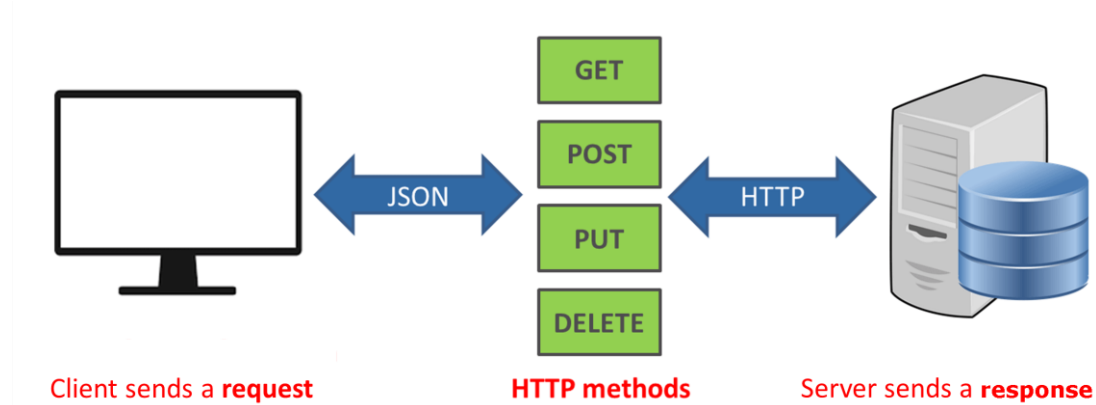
Τύποι Υπηρεσιών Ιστού

Μερικοί από τους κεντρικούς τύπους υπηρεσιών ιστού είναι το XML-RPC, UDDI, SOAP και REST.

1. XML-RPC (Remote Procedure Call): Το XML-RPC είναι το πιο βασικό πρωτόκολλο XML για την ανταλλαγή δεδομένων μεταξύ μιας μεγάλης ποικιλίας συσκευών σε ένα δίκτυο. Χρησιμοποιεί HTTP (Hypertext Transfer Protocol) για γρήγορα και εύκολη μεταφορά δεδομένων και επικοινωνίας άλλων πληροφοριών από πελάτη σε διακομιστή.
2. UDDI (Universal Description, Discovery and Integration): Το UDDI είναι ένα πρότυπο βασισμένο σε XML με σκοπό τη λεπτομέρεια, τη δημοσίευση και την ανακάλυψη των υπηρεσιών ιστού. Είναι στην ουσία ένα διαδικτυακό μητρώο για επιχειρήσεις σε όλο τον κόσμο.
3. SOAP (Simple Object Access Protocol): Το SOAP είναι ένα πρωτόκολλο υπηρεσιών ιστού που σε XML με σκοπό την ανταλλαγή δεδομένων και εγγράφων μέσω HTTP ή SMTP (Simple Mail Transfer Protocol). Επιτρέπει σε ανεξάρτητες διαδικασίες που λειτουργούν σε διαφορετικά συστήματα να επικοινωνούν χρησιμοποιώντας XML.
4. REST (Representational State Transfer): Τέλος το REST, του οποίου γίνεται χρήση στην παρούσα εφαρμογή, παρέχει επικοινωνία και συνδεσιμότητα μεταξύ συσκευών και του διαδικτύου για εργασίες βασισμένες σε API. Οι περισσότερες υπηρεσίες RESTful χρησιμοποιούν το HTTP ως πρωτόκολλο υποστήριξης.

2.3.4.1 REST

Το Representational State Transfer η αλλιώς REST είναι μια αρχιτεκτονική που δημιουργήθηκε για να καθοδηγήσει το σχεδιασμό και την ανάπτυξη της αρχιτεκτονικής για τον Παγκόσμιο Ιστό. Το REST όπως και όλα τα αρχιτεκτονικά στυλ λογισμικών ορίζει ένα σύνολο περιορισμών για το πώς πρέπει να συμπεριφέρεται ο Ιστός. Το αρχιτεκτονικό στυλ του REST δίνει έμφαση στην επεκτασιμότητα των αλληλεπιδράσεων μεταξύ των στοιχείων, στις ομοιόμορφες διεπαφές, στην ανεξάρτητη ανάπτυξη στοιχείων και στη δημιουργία μιας πολύ-επίπεδης αρχιτεκτονικής για τη διευκόλυνση της αποθήκευσης των στοιχείων στην μνήμη cache, για τη μείωση του χρόνου απόκρισης που αντιλαμβάνεται ο χρήστης, την ενίσχυση της ασφάλειας και την ενθύλακωση παλαιών συστημάτων.



Σχήμα 2.17: Λειτουργία του REST API

Το REST έχει χρησιμοποιηθεί σε όλη τη βιομηχανία λογισμικού και είναι ένα ευρέως αποδεκτό σύνολο κατευθυντήριων γραμμών για τη δημιουργία αξιόπιστων API εφαρμογών. Ένα web API που υπακούει στους περιορισμούς REST περιγράφεται ανεπίσημα ως RESTful. Τα RESTful web APIs βασίζονται συνήθως σε μεθόδους HTTP για πρόσβαση σε πόρους μέσω παραμέτρων οι οποίες είναι κωδικοποιημένες με URL, αλλά και στη χρήση JSON ή XML για τη μεταφορά δεδομένων.

Τα "Web resources" ορίστηκαν για πρώτη φορά στον Παγκόσμιο Ιστό ως έγγραφα ή αρχεία που προσδιορίζονται από τις διευθύνσεις URL τους. Σήμερα, ο ορισμός είναι πολύ πιο γενικός και αφηρημένος και περιλαμβάνει κάθε πράγμα, οντότητα ή ενέργεια που μπορεί να αναγνωριστεί, να ονομαστεί, να απευθυνθεί, να χειριστεί ή να εκτελεστεί με οποιονδήποτε τρόπο στον Ιστό. Σε μια υπηρεσία RESTful Web, τα αιτήματα που γίνονται στο URI ενός πόρου προκαλούν μια απάντηση με ένα ωφέλιμο φορτίο μορφοποιημένο σε HTML, XML, JSON ή κάποια άλλη μορφή. Για παράδειγμα, η απάντηση μπορεί να επιβεβαιώσει ότι η κατάσταση του πόρου έχει αλλάξει. Η απάντηση μπορεί επίσης να περιλαμβάνει συνδέσμους υπερκειμένου προς σχετικούς πόρους. Το πιο κοινό πρωτόκολλο για αυτά τα αιτήματα και απαντήσεις είναι το HTTP. Παρέχει λειτουργίες (μέθοδοι HTTP) όπως GET, POST, PUT και DELETE. Χρησιμοποιώντας ένα πρωτόκολλο χωρίς κατάσταση και τυπικές λειτουργίες, τα συστήματα RESTful στοχεύουν σε γρήγορη απόδοση, αξιοπιστία και δυνατότητα ανάπτυξης με την επαναχρησιμοποίηση στοιχείων που μπορούν να διαχειρίζονται και να ενημερώνονται χωρίς να επηρεάζεται το σύστημα στο σύνολό του, ακόμη και όταν εκτελείται.

Ο στόχος του REST είναι η αύξηση της απόδοσης, της επεκτασιμότητας, της απλότητας, της δυνατότητας τροποποίησης, της ορατότητας, της φορητότητας και της αξιοπιστίας. Αυτό επιτυγχάνεται ακολουθώντας τις αρχές REST, όπως η αρχιτεκτονική πελάτη-διακομιστή, η statelessness, η προσωρινή αποθήκευση, η χρήση ενός πολύ-επίπεδου συστήματος, η υποστήριξη για κώδικα κατά παραγγελία και η χρήση ομοιόμορφης διεπαφής. Αυτές οι αρχές πρέπει να τηρούνται για να ταξινομηθεί το σύστημα.

2.3.5 JSON

Το JSON (Javascript Object Notation) είναι ένα ελαφρύ πρότυπο μεταφοράς δεδομένων, είναι εύκολο στην ανάγνωση για τους προγραμματιστές και στην ανάλυση για τους υπολογιστές. Το JSON στην ουσία είναι μία μορφή κειμένου που δεν σχετίζεται με κάποια γλώσσα, παρόλα αυτά χρησιμοποιεί συμβάσεις που είναι πανομοιότυπες σε όλες τις γλώσσες προγραμματισμού. Το JSON έχει δύο τρόπους απεικόνισης:

1. Ζευγάρια ονόματος/τιμής: σε διαφορετικές γλώσσες αυτός ο τύπος μεταφράζεται ως object, record, struct, dictionary, hash table, keyed list ή associative array.

2. Ταξινομημένη λίστα τιμών: σε διαφορετικές γλώσσες αυτός ο τύπος μεταφράζεται ως array, vector, list ή sequence.

```
{
  "data": {
    "id": 2,
    "title": "1ο Εξάμηνο",
    "parent_id": 163,
    "is_public": false,
    "maillist_name": null
  }
}
```

Σχήμα 2.18: Παράδειγμα JSON

2.3.6 OAuth2.0

Το OAuth2.0 (Open Authorization), είναι ένα πρότυπο που έχει σχεδιαστεί για να επιτρέπει σε έναν ιστότοπο ή μια εφαρμογή να έχει πρόσβαση σε πόρους που φιλοξενούνται από άλλες εφαρμογές ιστού για λογαριασμό ενός χρήστη. Αντικατέστησε το OAuth1.0 το 2012 και είναι πλέον το de facto βιομηχανικό πρότυπο για την ηλεκτρονική εξουσιοδότηση. Το OAuth2.0 παρέχει πρόσβαση με συναίνεση και περιορίζει τις ενέργειες που μπορεί να εκτελέσει η εφαρμογή-πελάτης σε πόρους για λογαριασμό του χρήστη, χωρίς ποτέ να μοιράζεται τα διαπιστευτήρια του χρήστη.

Αρχές του OAuth2.0

Το OAuth2.0 είναι πρωτόκολλο εξουσιοδότησης και όχι πρωτόκολλο ελέγχου ταυτότητας. Έχει σχεδιαστεί κυρίως ως μέσο παροχής πρόσβασης σε ένα σύνολο πόρων, για παράδειγμα, απομακρυσμένα API ή δεδομένα χρήστη.



Σχήμα 2.19: Λογότυπο OAuth2.0

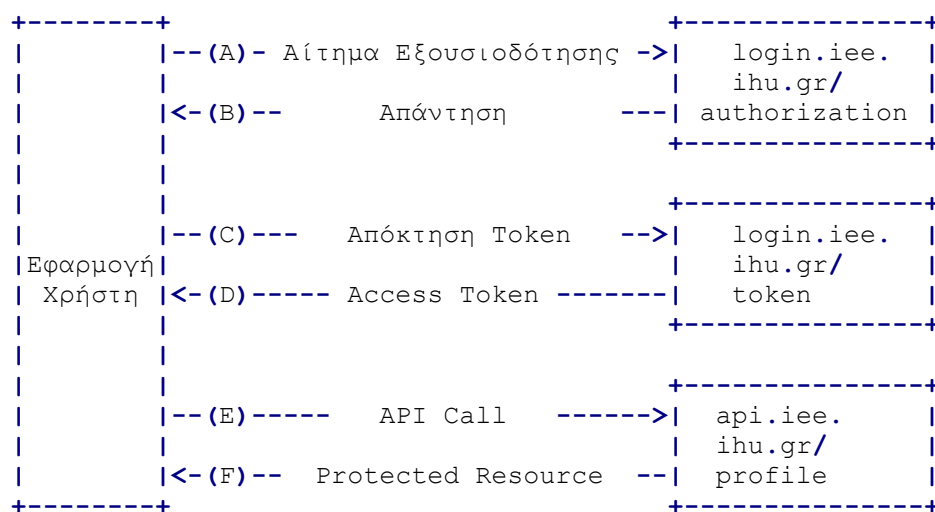
Χρησιμοποιεί Access Tokens. Ένα Access Token είναι ένα κομμάτι δεδομένων που αντιπροσωπεύει την εξουσιοδότηση πρόσβασης σε πόρους για λογαριασμό του τελικού χρήστη. Δεν ορίζει συγκεκριμένη μορφή για τα Tokens. Ωστόσο, σε ορισμένα περιβάλλοντα, χρησιμοποιείται συχνά η μορφή του JSON Web Token ή αλλιώς JWT. Αυτό επιτρέπει στους εκδότες διακριτικών να συμπεριλάβουν δεδομένα στο ίδιο το διακριτικό. Επίσης, για λόγους ασφαλείας, τα Access Tokens ενδέχεται να έχουν ημερομηνία λήξης.

Ρόλοι OAuth2.0

Τα βασικά στοιχεία ενός συστήματος OAuth2.0 είναι τα εξής:

1. **Resource Owner:** Ο χρήστης ή το σύστημα που κατέχει τους προστατευμένους πόρους και μπορεί να παραχωρήσει πρόσβαση σε αυτούς.
2. **Client:** Ο πελάτης είναι το σύστημα που απαιτεί πρόσβαση στους προστατευμένους πόρους. Για να αποκτήσει πρόσβαση σε πόρους, ο πελάτης πρέπει να διατηρεί το κατάλληλο διακριτικό πρόσβασης.
3. **Authorization Server:** Αυτός ο διακομιστής λαμβάνει αιτήματα από τον Πελάτη για διακριτικά πρόσβασης και τα εκδίδει μετά από επιτυχή έλεγχο ταυτότητας και συναίνεση από τον Κάτοχο πόρων. Ο διακομιστής εξουσιοδότησης εκθέτει δύο τελικά σημεία: το τελικό σημείο εξουσιοδότησης, το οποίο χειρίζεται τον αλληλεπιδραστικό έλεγχο ταυτότητας και τη συναίνεση του χρήστη και το τελικό σημείο Token, το οποίο εμπλέκεται στην αλληλεπίδραση μεταξύ μηχανής.
4. **Resource Server:** Διακομιστής που προστατεύει τους πόρους του χρήστη και λαμβάνει αιτήματα πρόσβασης από τον πελάτη. Αποδέχεται και επικυρώνει ένα Access Token από τον πελάτη και επιστρέφει τους κατάλληλους πόρους.

Ροή OAuth2.0



2.3.7 MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που έχει περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Έλαβε το όνομα της από την κόρη του συνιδρυτή της, Michael Widenius. Το πρόγραμμα τρέχει έναν εξυπηρετητή δίνοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων.



Σχήμα 2.20: Λογότυπο MySQL

Ο κώδικας του εγχειρίματος είναι ελεύθερος και διατίθεται μέσω της GNU General Public License (Γενική Άδεια Δημόσιας Χρήσης), καθώς και μέσω ορισμένων ιδιόκτητων συμφωνιών. Η MySQL αρχικά άνηκε και χρηματοδοτούνταν από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB, η οποία αγοράστηκε από την Sun. Το 2010, όταν η Oracle εξαγόρασε τη Sun, ο Widenius διοχέτευσε το έργο ανοιχτού κώδικα MySQL για να δημιουργήσει την MariaDB, η οποία χρησιμοποιείται και στην παρούσα εφαρμογή.

2.3.7.1 MariaDB

Η MariaDB είναι μια βάση δεδομένων SQL κατάλληλη για μικρές έως μεσαίες ιστοσελίδες. Ξεκίνησε ως fork της MySQL από τον ιδιοκτήτη της όταν αυτή πουλήθηκε στην Oracle. Η MariaDB έχει διατηρήσει υψηλή συμβατότητα με τη MySQL και, ως εκ τούτου, μπορεί να χρησιμοποιηθεί ως drop-in replacement αυτής. Οι κοινές εφαρμογές της περιλαμβάνουν εφαρμογές web που βασίζονται σε διάφορες γλώσσες προγραμματισμού που απαιτούν back-end αποθήκευσης DB.



Σχήμα 2.21: Λογότυπο MySQL

Η MariaDB χρησιμοποιείται συνήθως με 2 διαφορετικούς μηχανισμούς αποθήκευσης. Ο ένας ονομάζεται MyISAM, δεν υποστηρίζει συναλλαγές και αποθηκεύει κάθε πίνακα σε ένα σύνολο 3 αρχείων. Ο δεύτερος λέγεται InnoDB και υποστηρίζει συναλλαγές. Αυτός ο μηχανισμός αποθήκευσης αποθηκεύει όλα τα δεδομένα σε ένα ενιαίο σύνολο από bytes ή χρησιμοποιεί ένα σύνολο bytes ανά κατάλογο βάσεων δεδομένων. Η MariaDB έχει ένα μεγάλο πλεονέκτημα, καθώς είναι δωρεάν είναι συνήθως διαθέσιμη σε πακέτα κοινής φιλοξενίας (shared hosting) και μπορεί εύκολα να εγκατασταθεί σε περιβάλλον Linux, Unix ή Windows.

2.3.8 Postman



Σχήμα 2.22: Λογότυπο Postman

Το Postman είναι μια εφαρμογή που χρησιμοποιείται για δοκιμές API. Είναι ένας HTTP Client που δοκιμάζει αιτήματα HTTP, χρησιμοποιώντας μια γραφική διεπαφή χρήστη, μέσω της οποίας λαμβάνουμε διαφορετικούς τύπους απαντήσεων που πρέπει στη συνέχεια να επικυρωθούν. Το Postman προσφέρει πολλές μεθόδους αλληλεπίδρασης τελικού σημείου. Τα παρακάτω είναι μερικά από τα πιο χρησιμοποιούμενα, συμπεριλαμβανομένων των λειτουργιών τους:

- GET: Λαμβάνει πληροφορίες
- POST: Προσθέτει πληροφορίες
- PUT: Αντικαθιστά πληροφορίες
- PATCH: Ανανεώνει κάποιες συγκεκριμένες πληροφορίες
- DELETE: Σβήνει πληροφορίες

Κατά τη δοκιμή των API με το Postman, συνήθως λαμβάνουμε διαφορετικούς κωδικούς απόκρισης. Μερικοί από τους πιο κοινούς περιλαμβάνουν:

- 100 Series: Προσωρινές απαντήσεις
- 200 Series: Απαντήσεις όπου ο πελάτης αποδέχεται το αίτημα και ο διακομιστής με επιτυχία το επεξεργάζεται
- 300 Series: Απαντήσεις που σχετίζονται με την ανακατεύθυνση URL
- 400 Series: Απαντήσεις σφαλμάτων στην πλευρά του πελάτη
- 500 Series: Απαντήσεις σφαλμάτων στην πλευρά του διακομιστή

Το Postman έπειτα δίνει τη δυνατότητα ομαδοποίησης διαφορετικών αιτημάτων. Αυτή η δυνατότητα είναι γνωστή και ως «συλλογές» και βοηθά στην οργάνωση των δοκιμών.

Αυτές οι συλλογές είναι φάκελοι που αποθηκεύονται τα αιτήματα και μπορούν να δοθούν με όποιον τρόπο προτιμά η ομάδα. Δίνεται επίσης η δυνατότητα εισαγωγής-εξαγωγής τους από την εφαρμογή.

2.4 Επίλογος

Στο παραπάνω κεφάλαιο αναλύθηκαν οι τεχνολογίες που χρησιμοποιούνται στην παρούσα εφαρμογή. Αναλύθηκαν τόσο οι τεχνολογίες που αναπτύχθηκαν στο Front-End όσο και στο Back-End.

Κεφάλαιο 3ο: Οδηγοί Εγκατάστασης και Λειτουργίας της Εφαρμογής

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει μία ανάλυση στον τρόπο εγκατάστασης, στον τρόπο λειτουργίας της εφαρμογής καθώς και στον τρόπο υλοποίηση του OAuth2.0 πρωτόκολλου.

3.2 Τοπική Εγκατάσταση (Windows)

Για την εγκατάσταση της εφαρμογής στο τοπικό μας μηχάνημα θα χρειαστούν τα εξής βήματα:

1. Αρχικά εγκαθιστούμε το πακέτο που θα περιλαμβάνει την PHP και την MySQL, όπως είναι το (Xampp, Wampp κ.α.). Επίσης θα χρειαστούμε το npm, το Node.js και τον composer.
2. Για να εγκαταστήσουμε την εφαρμογή θα πρέπει να κλωνοποιήσουμε το repository που περιέχει το πρότζεκτ στην επιφάνεια εργασίας μας.
3. Έπειτα τρέχουμε την εντολή **composer install**, για να εγκαταστήσουμε τα πακέτα που απαιτεί η εφαρμογή μας μέσα στο αρχείο composer.json. Σε περίπτωση που η εντολή δεν ολοκληρωθεί με επιτυχία, τρέχουμε **composer clearcache** και στη συνέχεια ξανά τρέχουμε το **composer install**.
4. Αφού έχουμε εγκαταστήσει τα πακέτα μας, αντιγράφουμε το παράδειγμα .env.example στο αρχείο .env. Για να γίνει αυτό πρέπει να τρέξουμε την εντολή **copy .env.example .env** ή **cp .env.example .env** για το git bash.
5. Δημιουργούμε μία βάση δεδομένων δίνοντας της απλά ένα όνομα μέσα από το πρόγραμμα που χρησιμοποιούμε (πχ. SQLyog, MySQL, HeidiSQL, phpMyAdmin κ.α.).
6. Στη συνέχεια τοποθετούμε το όνομα που δώσαμε στη βάση μέσα στο .env στη μεταβλητή DB_DATABASE και τοποθετούμε τα στοιχεία που έχουμε δώσει στη βάση μας ως username και password (ως προεπιλογή συνήθως είναι root για το username και αφήνουμε κενό το password). Τέλος ορίζουμε ως APP_URL το <http://127.0.0.1:8000/>.
7. Μετά δημιουργούμε το κλειδί της εφαρμογής με την εντολή **php artisan:key generate**.
8. Αφού δημιουργήσουμε το κλειδί, ήρθε η ώρα να φτιάξουμε τους πίνακες. Για να γίνει αυτό πρέπει να τρέξουμε την εντολή **php artisan migrate**.
9. Όταν ολοκληρωθεί η δημιουργία των πινάκων είμαστε έτοιμοι να εγκαταστήσουμε την Vue στο τοπικό μας μηχάνημα. Για να γίνει αυτό τρέχουμε την εντολή **npm install**, η οποία μας εγκαθιστά τα απαραίτητα αρχεία που απαιτεί η εφαρμογή μας μέσα στο αρχείο package.json.
10. Τέλος για να ξεκινήσουμε την εφαρμογή στο Front-end σε Development Mode τρέχουμε την εντολή **npm run dev**, η οποία επιτρέπει την λειτουργία του debugging.
11. Αφού έχουμε τρέξει όλα αυτά τα βήματα η εφαρμογή μας είναι έτοιμη. Για να επισκεφθούμε την ιστοσελίδα θα πρέπει να ανοίξουμε τον server, αυτό γίνεται με την εντολή **php artisan serve**.
12. Σε περίπτωση που θέλουμε να κάνουμε login τον χρήστη, επειδή η εφαρμογή συνδέεται με το πρωτόκολλο OAuth2.0, θα χρειαστεί να δημιουργήσουμε την εφαρμογή μας στο <https://login.tee.ihu.gr>:
 - a. Αφού συνδεθούμε στον παραπάνω σύνδεσμο, πατάμε οι εφαρμογές μου και δημιουργία εφαρμογής.
 - b. Συμπληρώνουμε τα απαραίτητα πεδία. Στην αρχική σελίδα εφαρμογής βάζουμε: <http://127.0.0.1:8000/announcements> ή <https://127.0.0.1:8000/announcements>.
 - c. και στο URL επιστροφής: <http://127.0.0.1:8000/sign-in/redirect> ή <https://127.0.0.1:8000/sign-in/redirect>.
 - d. Τέλος επιλέγουμε μόνο το πλήρη στοιχεία χρήστη για προβολή. Αφού δημιουργήσουμε την εφαρμογή μεταφέρουμε το **Client ID**, **Secret** και **Url Επιστροφής** στα πεδία που βρίσκονται στο .env.

3.3 Εγκατάσταση στον server

Για να εγκαταστήσουμε στον server της σχολής την εφαρμογή για development, καθώς η εφαρμογή για production είναι ήδη εγκατεστημένη, χρειάζονται τα εξής:

1. Συνδεόμαστε στον server μέσω ενός SSH Client (πχ. PuTTY)
2. Σε περίπτωση που δεν έχουμε τον λογαριασμό του aboard αλλά είμαστε superuser τρέχουμε την εντολή `'sudo su - aboard'`
3. Μόλις συνδεθούμε με τον χρήστη aboard δημιουργούμε έναν φάκελο με το όνομα που θέλουμε στον root φάκελο του χρήστη.
4. Έπειτα κλωνοποιούμε το repository του project μας στον φάκελο που δημιουργήσαμε.
5. Ο server τρέχει μέσω του nginx. Διαμορφώνουμε τον nginx με τον εξής τρόπο:
 - a. Μεταφερόμαστε στο εξής path `'cd /etc/nginx/sites-enabled/'`.
 - b. Δημιουργούμε ένα αρχείο με το όνομα που θέλουμε.
 - c. Αφού δημιουργήσουμε το αρχείο, αντιγράφουμε το configuration που μας δίνει η Laravel μέσω της ιστοσελίδας: <https://laravel.com/docs/6.x/deployment>.
 - d. Αλλάζουμε το server_name με το όνομα που θέλουμε να έχει η ιστοσελίδα μας.
 - e. Αλλάζουμε το root βάζοντας το path που βρίσκεται ο public φάκελος μέσα στο project μας (πχ. /home/aboard/tboard.iee.ihu.gr/public).
6. Αφού έχουμε διαμορφώσει τον nginx πηγαίνουμε στον φάκελο του project και τρέχουμε την εντολή `'composer install'`, για να εγκαταστήσουμε τα πακέτα που απαιτεί η εφαρμογή μας μέσα στο αρχείο composer.json. Σε περίπτωση που η εντολή δεν ολοκληρωθεί με επιτυχία, τρέχουμε `'composer clearcache'` και στη συνέχεια ξανά τρέχουμε το `'composer install'`.
7. Αντιγράφουμε το .env.example στο αρχείο .env με την εντολή `'cp .env.example .env'`.
8. Δημιουργούμε την βάση με την mysql σε περίπτωση που δεν υπάρχει.
9. Στη συνέχεια τοποθετούμε το όνομα που δώσαμε στη βάση μέσα στο .env στη μεταβλητή DB_DATABASE και τοποθετούμε τα στοιχεία που έχουμε δώσει στη βάση μας ως username και password. Τέλος ορίζουμε ως APP_URL το url της ιστοσελίδας μας (πχ. http://tboard.iee.ihu.gr).
10. Μετά δημιουργούμε το κλειδί της εφαρμογής με την εντολή `'php artisan:key generate'`.
11. Αυτό το βήμα είναι προαιρετικό, σε περίπτωση που έχουμε ήδη τους πίνακες σε μία βάση δεν απαιτείται η εκτέλεση της παρακάτω εντολής. Αφού δημιουργήσουμε το κλειδί, ήρθε η ώρα να φτιάξουμε τους πίνακες. Για να γίνει αυτό πρέπει να τρέξουμε την εντολή `'php artisan migrate'`.
12. Όταν ολοκληρωθεί η δημιουργία των πινάκων είμαστε έτοιμοι να εγκαταστήσουμε την Vue στο τοπικό μας μηχάνημα. Τρέχουμε την εντολή `'npm install'`, η οποία μας εγκαθιστά τα απαραίτητα αρχεία που απαιτεί η εφαρμογή μας μέσα στο αρχείο package.json.
13. Τέλος για να ξεκινήσουμε την εφαρμογή στο Front-end σε Production Mode τρέχουμε την εντολή `'npm run prod'`.
14. Για να επισκεφθούμε την ιστοσελίδα μας πηγαίνουμε στο αντίστοιχο url που έχουμε ορίσει.
15. Σε περίπτωση που θέλουμε να κάνουμε login τον χρήστη, επειδή η εφαρμογή συνδέεται με το πρωτόκολλο OAuth2.0, θα χρειαστεί να φτιάξουμε την εφαρμογή μας στο <https://login.iee.ihu.gr> :
 - a. Αφού συνδεθούμε στον παραπάνω σύνδεσμο, πατάμε οι εφαρμογές μου και δημιουργία εφαρμογής.
 - b. Συμπληρώνουμε τα απαραίτητα πεδία. Στην αρχική σελίδα εφαρμογής βάζουμε: <http://XXXXXXXXXX/announcements> ή <https://XXXXXXXXXX/announcements>.
 - c. και στο URL επιστροφής: <http://XXXXXXXXXX/sign-in/redirect> ή <https://XXXXXXXXXX/sign-in/redirect>.
 - d. Τέλος επιλέγουμε μόνο το πλήρη στοιχεία χρήστη για προβολή. Αφού δημιουργήσουμε την εφαρμογή μεταφέρουμε το **Client ID**, **Secret** και **Url Επιστροφής** στα πεδία που βρίσκονται στο .env.

3.4 Οδηγίες λειτουργίας

Αφού έχουμε ολοκληρώσει την εγκατάσταση της εφαρμογής μας, είτε τοπικά είτε στον server, ήρθε η ώρα να δούμε πώς να ενημερώνουμε την εφαρμογή με τις αλλαγές που έχουμε κάνει.

Σε περίπτωση που θέλουμε να πάρουμε τις αλλαγές από το github και να ενημερώσουμε την εφαρμογή μας τοπικά ή και στον server κάνουμε τα εξής βήματα:

1. Τρέχουμε την εντολή `git pull origin <your branch>` για να πάρουμε τις αλλαγές που έχουμε κάνει στο github.
2. Στην συνέχεια σε περίπτωση που έχουμε εγκαταστήσει κάποιο πακέτο στην Laravel τρέχουμε την εντολή `composer install`.
3. Το ίδιο ισχύει και για την Vue.js. Τρέχουμε την εντολή `npm install`.
4. Αφού έχουμε κάνει αυτά τα βήματα, για να ενημερώσουμε την εφαρμογή μας έχουμε τρεις τρόπους:
 - a. Για να τρέξει η εφαρμογή σε Production Mode τρέχουμε `npm run prod`
 - b. Για να τρέξει σε Development Mode μπορούμε να τρέξουμε, είτε `npm run dev` είτε `npm run watch`. Η τελευταία εντολή τρέχει μέχρι να την σταματήσουμε και ελέγχει για αλλαγές που κάνουμε στα αρχεία της Vue.js, αυτό έχει σαν αποτέλεσμα να κάνει build μόνο τα αρχεία που αλλάζουμε και όχι όλο το πρότζεκτ από την αρχή.

3.5 Οδηγός εγκατάστασης OAuth2.0

1. Σαν 1^ο βήμα πρέπει να δημιουργήσουμε την εφαρμογή μας στο login.iew.ihu.gr συμπληρώνοντας τα εξής:
 - a. Αρχική σελίδα εφαρμογής (πχ. <https://aboard.iew.ihu.gr>)
 - b. URL επιστροφής στην εφαρμογή (πχ. <https://aboard.iew.ihu.gr/sign-in/redirect>). Αυτό είναι το URL το οποίο θα παίρνει τις παραμέτρους από το login.iew.ihu.gr και θα μας επιστρέφει στη σελίδα που θέλουμε.
 - c. Στην πρόσβαση της εφαρμογής επιλέγουμε μόνο τα πλήρη στοιχεία χρήστη.
2. Αφού δημιουργήσουμε την εφαρμογή μας στο login.iew.ihu.gr, τοποθετούμε στα config/services.php ένα όνομα με το οποίο θα καλούμε τα πεδία που θα βάλουμε (πχ. iew). Τα πεδία που πρέπει να βάλουμε είναι τα εξής:
 - a. `'client_id' => env('CLIENT_ID'),`
 - b. `'redirect' => env('REDIRECT_URI'),`
 - c. `'response_type' => env('RESPONSE_TYPE'),`
 - d. `'client_secret' => env('CLIENT_SECRET')`
3. Έπειτα πρέπει να τοποθετήσουμε στο αρχείο .env τα παραπάνω πεδία ως εξής:
 - a. `CLIENT_ID='το client_id που παίρνουμε από το login.iew.ihu.gr'`
 - b. `RESPONSE_TYPE='code'`
 - c. `REDIRECT_URI='εδώ τοποθετούμε το Callback URL'`
 - d. `CLIENT_SECRET='εδώ παίρνουμε το secret από το login.iew.ihu.gr'`
4. Στη συνέχεια εγκαθιστούμε το πακέτο του socialite:
 - a. `composer require laravel/socialite`
 - b. `composer install`
5. Αφού εγκαταστήσουμε το πακέτο, δημιουργούμε τον δικό μας πάροχο στο Auth\Social\Two\{Your_Name}Provider.php
6. Στον Provider πρέπει να εισάγουμε τα εξής αρχεία του Socialite:
 - a. `use Laravel\Socialite\Two\AbstractProvider;`
 - b. `use Laravel\Socialite\Two\ProviderInterface;`
 - c. `use Laravel\Socialite\Two\User;`
7. Στη συνέχεια κάνουμε extend το AbstractProvider και implement το ProviderInterface για να έχουμε πρόσβαση στις μεθόδους και να τις υλοποιήσουμε όπως εμείς χρειαζόμαστε.
8. Οι μέθοδοι που χρειάζεται να υλοποιήσουμε είναι οι εξής:
 - a. `getAuthUrl($state)` \Rightarrow επιστρέφουμε το `$this->buildAuthUrlFromBase('https://login.iew.ihu.gr/authorization', $state)`

- b. getTokenUrl() => κάνουμε return το token URL - 'https://login.iee.ihu.gr/token'
- c. getUserByToken(\$token) => παίρνουμε τον user σύμφωνα με το token που μας έδωσε ο πάροχος με τον εξής τρόπο:

```
$userUrl = 'https://api.iee.ihu.gr/profile?access_token=' . $token;

$response = $this->getHttpClient()->get(
    $userUrl, ['headers' => [
        'Accept' => 'application/json',
    ]],
    1);

$user = json_decode($response->getBody(), true);

return $user;
```

- d. mapUserToObject(array \$user) => παίρνουμε τον user σαν array όπως μας τον επιστρέφει ο πάροχος και επιλέγουμε τα πεδία που θέλουμε να χρησιμοποιήσουμε. Έπειτα επιστρέφουμε τον user σαν Socialite User Instance.

```
$group = $user['eduPersonAffiliation'];
$name_gr = $group === "staff" ?
    $user['sn;lang-el'] . " " . $user['givenName;lang-el'] :
    Str::upper($user['sn;lang-el'] . " " . $user['givenName;lang-el']);
$name_eng = !empty(Str::title($user['cn'])) ? Str::title($user['cn']) :
    Str::ascii($user['cn;lang-el']);
$is_author = $group === "staff";
$email = $user['mail'];

return (new User)->setRaw($user)->map([
    'uid' => $user['uid'],
    'name' => $name_gr,
    'name_eng' => $name_eng,
    'email' => $email,
    'is_author' => $is_author
]);
```

9. Αφού τελειώσουμε με τον Provider μας, πρέπει να τον προσθέσουμε στους providers τους Socialite. Για να γίνει αυτό πρέπει να δημιουργήσουμε το αρχείο app/Providers/SocialitePlusServiceProvider και να κάνουμε extend το SocialiteServiceProvider και να υλοποιήσουμε την εξής μέθοδο:

```
public function register() {
    // Register our own custom Socialite Provider
    $this->app->singleton('Laravel\Socialite\Contracts\Factory', function
($app) {
        $socialiteManager = new SocialiteManager($app);

        $socialiteManager->extend('iee', function() use ($socialiteManager) {
            $config = $this->app['config']['services.iee'];

            return $socialiteManager->buildProvider(
                'App\Auth\Social\Two\IeeProvider', $config
            );
        });
        return $socialiteManager;
    });
}
```

- a. Κάνουμε extend στον Socialite Manager τον Provider μας βάζοντας το όνομα που επιθυμούμε και θέλουμε να τον καλούμε, και στην συνέχεια βάζουμε το όνομα που έχουμε στο services.php. Στο buildProvider βάζουμε το config που φτιάξαμε παραπάνω.
10. Μετά πρέπει να βάλουμε το SocialitePlusServiceProvider στο config/app.php μέσα στους παρόχους.
 11. Στο routes/web.php βάζουμε τα δύο καινούρια routes ένα για το sign-in και ένα για το callback.

```
Route::get('/sign-in', 'Auth\AuthController@signIn')->name('login');
Route::get('/sign-in/redirect', 'Auth\AuthController@redirect');
```


12. Στο sign in το μόνο που κάνουμε είναι να επιστρέψουμε:
`Socialite::driver('{your_provider_name}')->redirect()`.
13. Αφού γίνει το login επιτυχώς στο login.iew.edu.gr μεταφερόμαστε στο Callback URL. Μέσα στην μέθοδο `redirect` παίρνουμε τον user μας από το Socialite και κάνουμε login και στη σελίδα μας παίρνοντας τον χρήστη από την βάση μας και κάνοντας προσπάθεια για login μέσω του Laravel Auth. Αφού ολοκληρωθούν τα βήματα αυτά κάνουμε `redirect` στο URL που θέλουμε να μεταφερθεί ο χρήστης.

3.6 Επίλογος

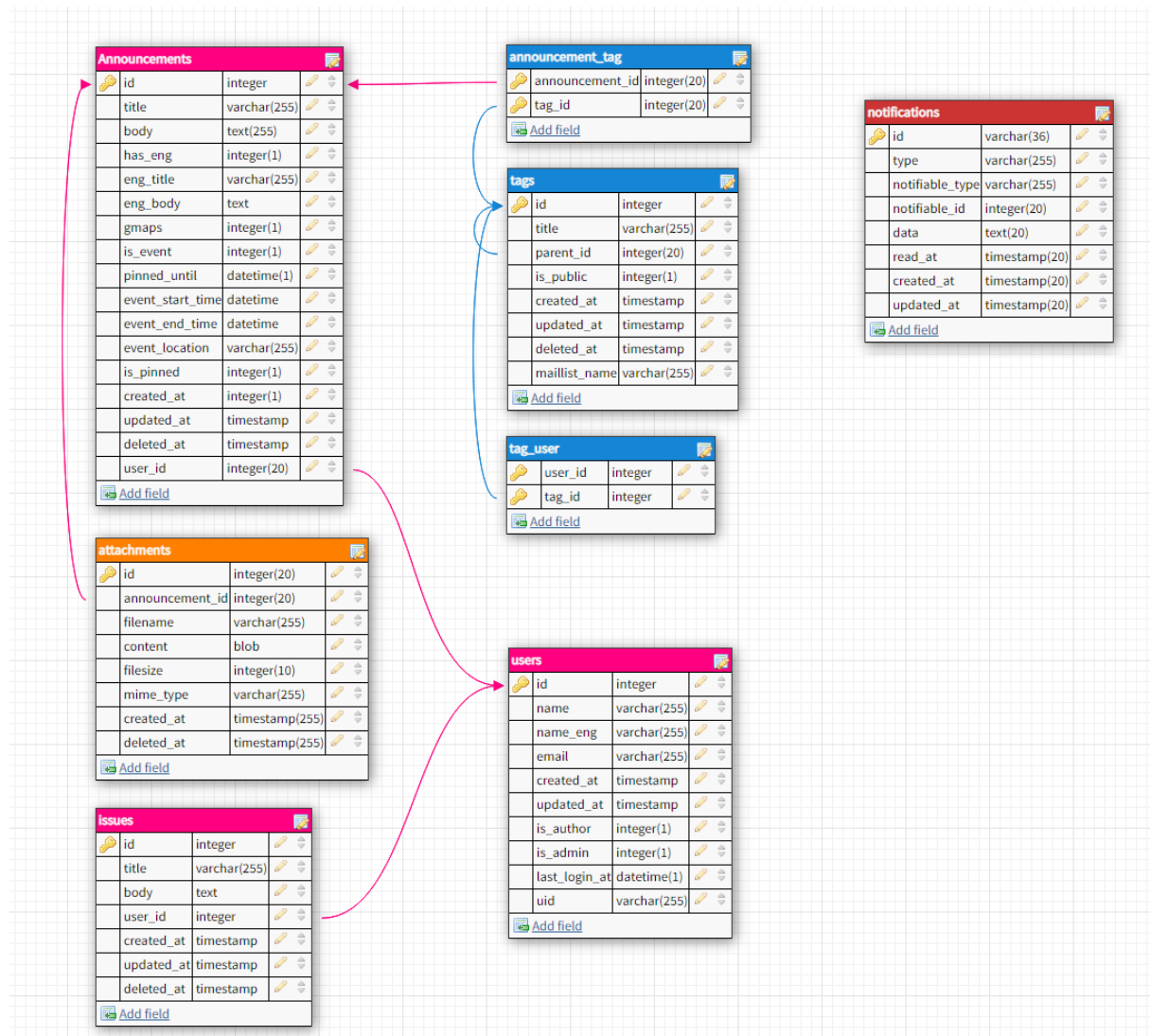
Σε αυτό το κεφάλαιο έγινε η ανάλυση στον τρόπο εγκατάστασης της εφαρμογής μας, είτε τοπικά είτε και στον server της σχολής μας. Στη συνέχεια αναλύθηκε και ο τρόπος με τον οποίο ενημερώνεται η ιστοσελίδα με τις αλλαγές που κάνουμε. Τέλος ακολούθησε και ένας οδηγός εγκατάστασης του OAuth2.0 πρωτόκολλου με τη χρήση του πακέτου Socialite της Laravel.

Κεφάλαιο 4ο: Σχεδίαση του συστήματος

4.1 Εισαγωγή

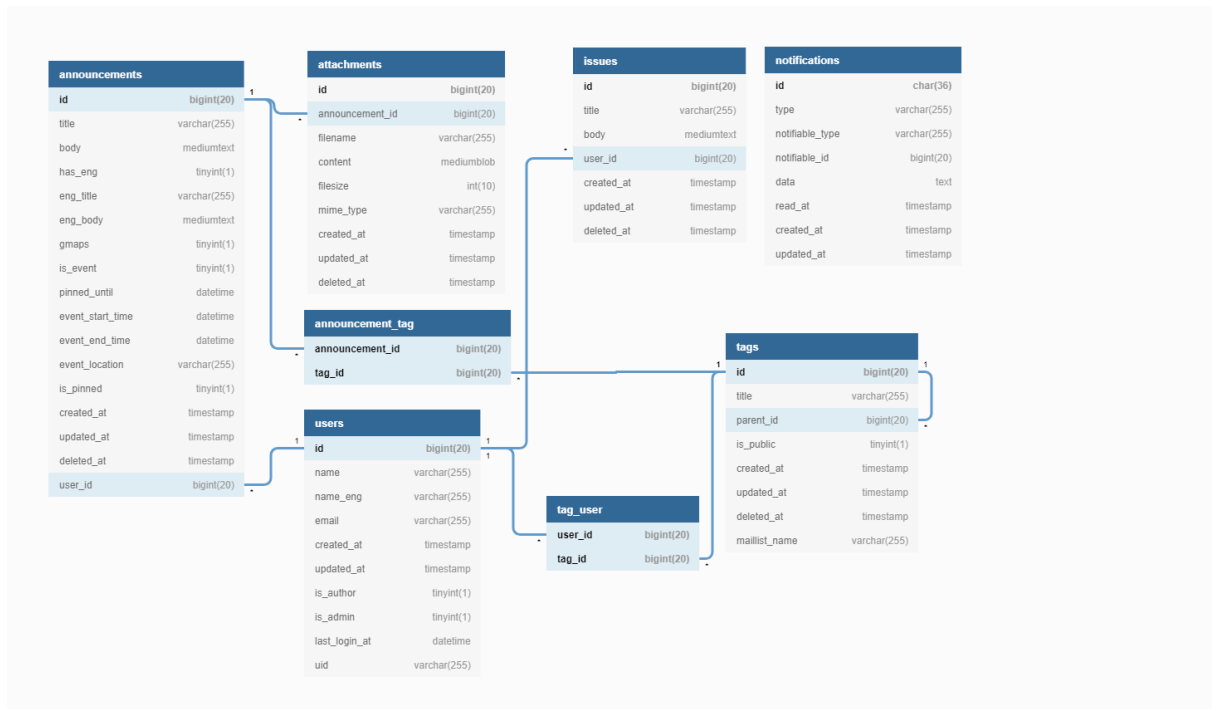
Η βάση δεδομένων μιας εφαρμογής είναι εάν από τα σημαντικότερα δομικά στοιχεία και καθορίζει σε μεγάλο βαθμό την σωστή λειτουργία της αλλά και την επεκτασιμότητα της στο μέλλον. Στο παρακάτω κεφάλαιο θα αναλύσουμε τους πίνακες της βάσης καθώς και την σχέση μεταξύ τους.

4.2 Database Schema



Σχήμα 4.1: Παράδειγμα του σχήματος της βάσης

4.3 Σχέσεις μεταξύ πινάκων



Σχήμα 4.2: Σχέσεις μεταξύ των πινάκων

Ανάλυση σχέσεων

Στην παραπάνω εικόνα βλέπουμε τις σχέσεις μεταξύ των πινάκων. Οι κύριες σχέσεις που υπάρχουν είναι οι εξής 6:

1. Σχέση Ανακοίνωσης – Επισύναψης: Μία ανακοίνωσή μπορεί να περιέχει πολλά συνημμένα, ενώ το κάθε συνημμένο μπορεί να βρίσκεται σε μία μόνο ανακοίνωση. Η σχέση αυτή χαρακτηρίζεται ως ένα προς πολλά (One to Many) και συνδέεται μέσω του id (announcements) και του announcement_id (attachments).
2. Σχέση Ανακοίνωσης – Χρήστη: Ένας χρήστης μπορεί να δημιουργήσει πολλές ανακοινώσεις εφόσον είναι συγγραφέας, ενώ κάθε ανακοίνωση ανήκει σε έναν μόνο χρήστη. Και εδώ η σχέση είναι ένα προς πολλά και συνδέεται μέσω του id (users) και του user_id (announcements).
3. Σχέση Ανακοίνωσης – Ετικέτας: Μία ανακοίνωση μπορεί να περιέχει πολλές ετικέτες, παράλληλα μία ετικέτα μπορεί να περιέχεται σε πολλές ανακοινώσεις. Αυτή η σχέση χαρακτηρίζεται ως πολλά προς πολλά (Many to Many) και για να λειτουργήσει απαιτείται ένας ενδιάμεσος πίνακας (announcement_tag), ο οποίος συνδέεται μέσω του id (announcements) και του announcement_id (announcement_tag) για τις ανακοινώσεις και μέσω του id (tags) και του tag_id (announcement_tag) για τις ετικέτες.
4. Σχέση Χρήστη – Ετικέτας: Ένας χρήστης μπορεί να εγγραφεί σε πολλές ετικέτες, επίσης μία ετικέτα μπορεί να είναι εγγεγραμμένη από πολλούς χρήστες. Και αυτή η σχέση είναι πολλά προς πολλά και απαιτεί τον ενδιάμεσο πίνακα, ο οποίος συνδέεται μέσω του id (users) και του user_id (tag_user) για τον χρήστη και μέσω του id (tags) και του tag_id (tag_user) για τις ετικέτες.
5. Σχέση Ετικέτας – Ετικέτας: Μία γονική ετικέτα μπορεί να περιέχει πολλές ετικέτες ενώ μία ετικέτα παιδιού μπορεί να περιέχεται σε μία μόνο γονική ετικέτα. Η σχέση αυτή χαρακτηρίζεται ως ένα προς πολλά και συνδέεται μέσω του id (tags) και του parent_id (tags).
6. Σχέση Χρήστη - Προβλήματος: Ένας χρήστης μπορεί να δημιουργήσει πολλά issues και αντίστοιχα ένα issue ανήκει σε έναν μόνο χρήστη. Και αυτή η σχέση είναι ένα προς πολλά και συνδέεται μέσω του id (users) και του user_id (issues).

4.4 API της εφαρμογής

Στον παρακάτω πίνακα παρουσιάζεται το API της εφαρμογής. Στη συνέχεια γίνεται αναλυτικότερη περιγραφή.

Endpoint	Τύπος request	Περιγραφή
/api/announcements?params	GET	Ανακτά όλες τις ανακοινώσεις με την χρήση των params ως φίλτρων
/api/announcements/{id}	GET	Ανακτά μία ανακοίνωση
/api/announcements/{an_id}/attachments/{at_id}	GET	Ανακτά ένα συνημμένο μίας ανακοίνωσης
/api/tags	GET	Ανακτά όλες τις ετικέτες
/api/filtertags?params	GET	Ανακτά όλες τις ετικέτες με χρήση των params και έχοντας το πλήθος ανακοινώσεων
/api/tags/{id}	GET	Ανακτά μία ετικέτα
/api/subscribetags	GET	Ανακτά τις ετικέτες με τα παιδιά τους
/api/auth/logout	GET	Αποσυνδέει τον χρήστη
/api/auth/user	GET	Ανακτά τον χρήστη
/api/auth/user/notifications	GET	Ανακτά τις ειδοποιήσεις
/api/auth/user/notifications/read	GET	Διαβάζει και ανακτά τις ειδοποιήσεις
/api/auth/subscriptions	GET	Ανακτά τις εγγραφές του χρήστη
/api/announcements	POST	Δημιουργεί μία ανακοίνωση
/api/tags	POST	Δημιουργεί μία ετικέτα
/api/auth/subscribe	POST	Εγγράφει τον χρήστη σε ετικέτες
/api/announcements/{id}	PUT	Επεξεργάζεται μία ανακοίνωση
/api/tags/{id}	PUT	Επεξεργάζεται μία ετικέτα
/api/announcements/{id}	DELETE	Διαγράφει μία ανακοίνωση
/api/tags/{id}	DELETE	Διαγράφει μία ετικέτα

4.4.1 Μέθοδοι GET

/api/announcements?params

Το /api/announcements χωρίς την μεταβλητή params ανακτά όλες τις ανακοινώσεις σελιδοποιημένες κατά 10 και ταξινομημένες ως προς τις σημαντικότερες ανακοινώσεις και έπειτα τις νεότερες.

Το /api/announcements?params ανακτά φιλτραρισμένες ανακοινώσεις χρησιμοποιώντας πολλαπλές παραμέτρους. Μερικά παραδείγματα παραμέτρων είναι οι users, τα tags, το perPage, το sortId, το title και body, τέλος το updatedAt και το updatedBefore.

Παράδειγμα response:

```
{
  "data": [
    {
      "id": 5,
      "title": "Ανακοίνωση",
      "eng_title": null,
      "body": "<p>Δημόσια ανακοίνωση</p>",
      "eng_body": null,
      "has_eng": 0,
      "created_at": "2020-04-13 20:37",
      "updated_at": "2020-04-17 18:57",
      "is_pinned": 0,
      "pinned_until": null,
      "is_event": 1,
      "event_start_time": "2020-04-29 00:00",
      "event_end_time": "2020-04-30 00:00",
      "event_location": "ΔΙΠΑΕ Σίνδος",
      "gmaps": 1,
      "tags": [
        {
          "id": 19,
          "title": "Άλλα δημόσια νέα",
          "parent_id": 1,
          "is_public": 1
        }
      ],
      "attachments": [],
      "author": {
        "name": "Nikolaos Christos Nikolaidis",
        "id": 2
      }
    }
  ],
  "links": {
    "first": "https://aboard.iee.ihu.gr/api/announcements?page=1",
    "last": "https://aboard.iee.ihu.gr/api/announcements?page=1",
    "prev": null,
    "next": null
  },
  "meta": {
    "current_page": 1,
    "from": 1,
    "last_page": 1,
    "path": "https://aboard.iee.ihu.gr/api/announcements",
    "per_page": 10,
    "to": 1,
    "total": 1
  }
}
```

/api/announcements/{id}

Ανακτά μία συγκεκριμένη ανακοίνωση δίνοντας το id της.

Παράδειγμα response:

```
{
  "data": {
    "id": 5,
    "title": "Ανακοίνωση",
    "eng_title": null,
    "body": "<p>Δημόσια ανακοίνωση</p>",
    "eng_body": null,
    "has_eng": 0,
    "created_at": "2020-04-13 20:37",
    "updated_at": "2020-04-17 18:57",
    "is_pinned": 0,
    "pinned_until": null,
    "is_event": 1,
    "event_start_time": "2020-04-29 00:00",
    "event_end_time": "2020-04-30 00:00",
    "event_location": "ΔΙΠΑΕ Σίνδος",
    "gmaps": 1,
    "tags": [
      {
        "id": 19,
        "title": "Άλλα δημόσια νέα",
        "parent_id": 1,
        "is_public": 1
      }
    ],
    "attachments": [],
    "author": {
      "name": "Nikolaos Christos Nikolaidis",
      "id": 2
    }
  }
}
```

/api/announcements/{an_id}/attachments/{at_id}

Ανακτά το συνημμένο με το {an_id} της ανακοίνωσης με το αναγνωριστικό του {at_id}

Παράδειγμα response:

```
{
  "data": "Attachment's file"
}
```

/api/tags

Ανακτά όλες τις ετικέτες.

Παράδειγμα response:

```
{
  "data": [
    {
      "id": 3,
      "title": "Εκδηλώσεις",
      "parent_id": 1,
      "is_public": 1
    },
    {
      "id": 2,
      "title": "Εξάμηνο Α",
      "parent_id": 1,
      "is_public": 0
    },
    {
      "id": 1,
      "title": "Όλες οι ανακοινώσεις",
      "parent_id": null,
      "is_public": 0
    }
  ]
}
```

/api/filtertags?params

Το /api/filtertags ανακτά όλες τις ετικέτες μετρώντας το πλήθος των ανακοινώσεων για όλες τις ανακοινώσεις τους χωρίς παραμέτρους.

Το /api/filtertags? ανακτά όλες τις ετικέτες μετρώντας το πλήθος των ανακοινώσεων για φιλτραρισμένες ανακοινώσεις. Όπως και στην ανάκτηση των ανακοινώσεων έτσι και εδώ τα παραδείγματα παραμέτρων είναι τα ίδια.

Παράδειγμα response:

```
[
  {
    "id": 163,
    "title": "1ο Έτος",
    "parent_id": 1,
    "is_public": false,
    "created_at": "2021-03-30 16:46:26",
    "updated_at": "null",
    "deleted_at": null,
    "announcements_count": 785,
    "children_recursive": [
      {
        "id": 163,
        "title": "1ο Εξάμηνο",
        "parent_id": 163,
        "is_public": false,
        "created_at": "2020-05-11 21:15:15",
        "updated_at": "2020-05-11 21:15:15",
        "deleted_at": null,
        "announcements_count": 34,
        "children_recursive": []
      }
    ]
  },
  ]
```

/api/tags/{id}

Ανακτά πληροφορίες της εκάστοτε ετικέτας [tag] με αναγνωριστικό το id.

Παράδειγμα response:

```
{
  "data": {
    "id": 2,
    "title": "Εξάμηνο Α",
    "parent_id": 1,
    "is_public": 0
  }
}
```

/api/subscribetags

Ανακτά τις ετικέτες με τα παιδιά τους για την σελίδα των εγγραφών.

Παράδειγμα response:

```
[
  {
    "id": 163,
    "title": "1ο Έτος",
    "parent_id": 1,
    "is_public": false,
    "created_at": "2021-03-30 16:46:26",
    "updated_at": "null",
    "deleted_at": null,
    "children_recursive": [
      {
        "id": 163,
        "title": "1ο Εξάμηνο",
        "parent_id": 163,
        "is_public": false,
        "created_at": "2020-05-11 21:15:15",
        "updated_at": "2020-05-11 21:15:15",
        "deleted_at": null,
        "children_recursive": []
      }
    ]
  },
  {
    "id": 160,
    "title": "Test board",
    "parent_id": 1,
    "is_public": true,
    "created_at": "2021-03-30 16:45:13",
    "updated_at": "2021-03-30 16:45:13",
    "deleted_at": null,
    "children_recursive": []
  }
]
```

/api/auth/logout

Αποσυνδέει τον χρήστη από την εφαρμογή.

Παράδειγμα response:

```
{
  "message": "Logout success"
}
```


/api/auth/user

Γυρνάει πληροφορίες για τον χρήστη.

Παράδειγμα response:

```
{
  "data": {
    "id": 21,
    "name": "Nikolaos-Christos Nikolaidis",
    "email": "it113763@it.teithe.gr",
    "uid": "it113763",
    "is_admin": 0,
    "is_author": 0,
    "subscriptions": [
      {
        "id": 4,
        "title": "Εξάμηνο Γ",
        "parent_id": 1,
        "is_public": 0
      },
      {
        "id": 6,
        "title": "Εξάμηνο Ε",
        "parent_id": 1,
        "is_public": 0
      }
    ]
  },
  "last_interaction_time": "2020-05-27 23:45:38",
  "last_login_at": "2020-05-27 23:45:25"
}
```

/api/auth/user/notifications και /api/auth/user/notifications/read

Στην πρώτη περίπτωση γυρνάει τις ειδοποιήσεις του συνδεδεμένου χρήστη. Στην δεύτερη περίπτωση διαβάζει τις νέες ειδοποιήσεις και γυρνάει τις ειδοποιήσεις του συνδεδεμένου χρήστη.

Παράδειγμα response:

```
{
  "data": [
    {
      "id": "c0783866-91ba-4b4a-bf02-16a1421d5d08",
      "data": {
        "type": "user.login",
        "id": 2,
        "user": "Nikolaos Christos Nikolaidis"
      },
      "created_at": "2020-04-21 22:14"
    }
  ],
  "links": {
    "first": "https://aboard.iee.ihu.gr/api/auth/user/notifications?page=1",
    "last": "https://aboard.iee.ihu.gr/api/auth/user/notifications?page=7",
    "prev": null,
    "next": "https://aboard.iee.ihu.gr/api/auth/user/notifications?page=2"
  },
  "meta": {
    "current_page": 1,
    "from": 1,
    "last_page": 7,
    "path": "https://aboard.iee.ihu.gr/api/auth/user/notifications",
    "per_page": 10,
    "to": 10,
    "total": 69
  }
}
```

/api/auth/subscriptions

Ανακτά τις εγγραφές των χρηστών.

Παράδειγμα response:

```
[
  {
    "id": 11,
    "title": "Πρακτική άσκηση",
    "parent_id": 1,
    "is_public": false,
    "created_at": "2020-05-11 21:15:15",
    "updated_at": "2020-05-11 21:15:15",
    "deleted_at": null,
    "pivot": {
      "user_id": 173,
      "tag_id": 11
    }
  },
  {
    "id": 12,
    "title": "Πτυχιακές Εργασίες",
    "parent_id": 1,
    "is_public": false,
    "created_at": "2020-05-11 21:15:15",
    "updated_at": "2020-10-04 20:15:53",
    "deleted_at": null,
    "pivot": {
      "user_id": 173,
      "tag_id": 12
    }
  }
]
```

/api/auth/authors?params

Το /api/auth/authors ανακτά τη λίστα συγγραφέων μετρώντας το πλήθος ανακοινώσεων που έχει δημιουργήσει ο καθένας για όλες τις ανακοινώσεις χωρίς παραμέτρους.

Το /api/auth/authors?params ανακτά τη λίστα συγγραφέων μετρώντας το πλήθος ανακοινώσεων που έχει δημιουργήσει ο καθένας για φιλτραρισμένες ανακοινώσεις. Όπως και στην ανάκτηση των ανακοινώσεων έτσι και εδώ τα παραδείγματα παραμέτρων είναι τα ίδια.

Παράδειγμα response:

```
[
  {
    "id": 2,
    "name": "Nikolaos Christos Nikolaidis",
    "announcement_count": 5
  },
  {
    "id": 3,
    "name": "George Stinis",
    "announcement_count": 3
  },
  {
    "id": 4,
    "name": "Pantelis Papadopoulos",
    "announcement_count": 6
  }
]
```

4.4.2 Μέθοδοι POST

/api/announcements

Δημιουργεί μια νέα ανακοίνωση.

Παράδειγμα request Content-Type (text/plain):

```
"title" = Test title
"body" = <p>Test announcement</p>
"is_pinned" = true
"pinned_until" = "2020-05-31 00:00"
"is_event" = true
"event_location" = TEITHE
"event_start_time" = "2020-05-28 00:00"
"event_end_time" = "2020-05-31 00:00"
"gmaps" = true
"tags" = [108]
"has_eng" = true
"eng_title" = Test title (English)
"eng_body" = <p>Test announcement (English)</p>
```

Παράδειγμα response:

```
{
  "data": {
    "id": 5,
    "title": "Test title",
    "eng_title": "Test title (English)",
    "body": "<p>Test announcement</p>",
    "eng_body": "<p>Test announcement (English)</p>",
    "has_eng": true,
    "created_at": "2020-05-28 22:06",
    "updated_at": "2020-05-28 22:06",
    "is_pinned": true,
    "pinned_until": "2020-05-31 00:00",
    "is_event": true,
    "event_start_time": "2020-05-28 00:00",
    "event_end_time": "2020-05-31 00:00",
    "event_location": "TEITHE",
    "gmaps": true,
    "tags": [
      {
        "id": 10,
        "title": "Εξάμηνο Θ",
        "parent_id": 1,
        "is_public": 0
      }
    ]
  },
  "attachments": [
    {
      "id": 4,
      "announcement_id": 5,
      "filename": "test.txt",
      "content": "",
      "filesize": 0,
      "mime_type": "inode/x-empty"
    }
  ],
  "author": {
    "name": "Nikolaos-Christos Nikolaidis",
    "id": 21
  }
}
```

/api/tags

Δημιουργεί ένα καινούριο tag.

Παράδειγμα request:

```
'Authorization: Bearer ACCESS_TOKEN'
"title" = "Test",
"parent_id" = 2,
"is_public" = false
```

Παράδειγμα response:

```
{
  "title": "Test",
  "is_public": false,
  "parent_id": 2,
  "updated_at": "2020-05-28 21:38:38",
  "created_at": "2020-05-28 21:38:38",
  "id": 109
}
```

/api/auth/subscribe

Κάνει εγγραφή τον χρήστη σε συγκεκριμένα tags.

Παράδειγμα request:

```
'Authorization: Bearer ACCESS_TOKEN'
"tags" = [4,6,108,107]
```

Παράδειγμα response:

```
{
  "id": 21,
  "subscriptions": [
    {
      "id": 4,
      "title": "Εξάμηνο Γ",
      "parent_id": 1,
      "is_public": 0,
      "created_at": "2020-05-11 21:15:15",
      "updated_at": "2020-05-11 21:15:15",
      "deleted_at": null,
      "pivot": {
        "user_id": 21,
        "tag_id": 4
      }
    },
    {
      "id": 6,
      "title": "Εξάμηνο Ε",
      "parent_id": 1,
      "is_public": 0,
      "created_at": "2020-05-11 21:15:15",
      "updated_at": "2020-05-11 21:15:15",
      "deleted_at": null,
      "pivot": {
        "user_id": 21,
        "tag_id": 6
      }
    }
  ]
}
```

4.4.3 Μέθοδοι PUT

/api/announcements/{id}

Ενημερώνει την ανακοίνωση με το αναγνωριστικό του {id}.

Παράδειγμα response:

```
{
  "data": {
    "id": 5,
    "title": "Test title (edit)",
    "eng_title": null,
    "body": "<p>Test announcement (edit)</p>",
    "eng_body": null,
    "has_eng": false,
    "created_at": "2020-05-28 22:06",
    "updated_at": "2020-05-28 22:22",
    "is_pinned": false,
    "pinned_until": null,
    "is_event": false,
    "event_start_time": null,
    "event_end_time": null,
    "event_location": null,
    "gmaps": null,
    "tags": [
      {
        "id": 10,
        "title": "Εξάμηνο Θ",
        "parent_id": 1,
        "is_public": 0
      },
      {
        "id": 108,
        "title": "1975 Τεχνολογία Πολυμέσων",
        "parent_id": 10,
        "is_public": 0
      }
    ],
    "attachments": [],
    "author": {
      "name": "Nikolaos-Christos Nikolaidis",
      "id": 21
    }
  }
}
```

/api/tags/{id}

Ανανεώνει τα δεδομένα του tag με το αναγνωριστικό του {id}.

Παράδειγμα response:

```
{
  "data": {
    "id": 109,
    "title": "Test edit",
    "parent_id": 10,
    "is_public": 1
  }
}
```

4.4.4 Μέθοδοι DELETE

/api/announcements/{id}

Σβήνει την ανακοίνωση με το αναγνωριστικό του {id}.

Παράδειγμα response:

```
{
  "data": {
    "all": "announcements",
    "not shown": "to keep this page as short as possible"
  }
}
```

/api/tags/{id}

Σβήνει την ετικέτα με το αναγνωριστικό του {id}.

Παράδειγμα response:

```
{
  "data": {
    "all": "tags",
    "not shown": "to keep this page as short as possible"
  }
}
```

Κεφάλαιο 5ο: Παρουσίαση της εφαρμογής

5.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει μία παρουσίαση της εφαρμογής από την οπτική ενός μη συνδεδεμένου χρήστη, ενός απλού χρήστη, ενός συγγραφέα και τέλος ενός διαχειριστή.

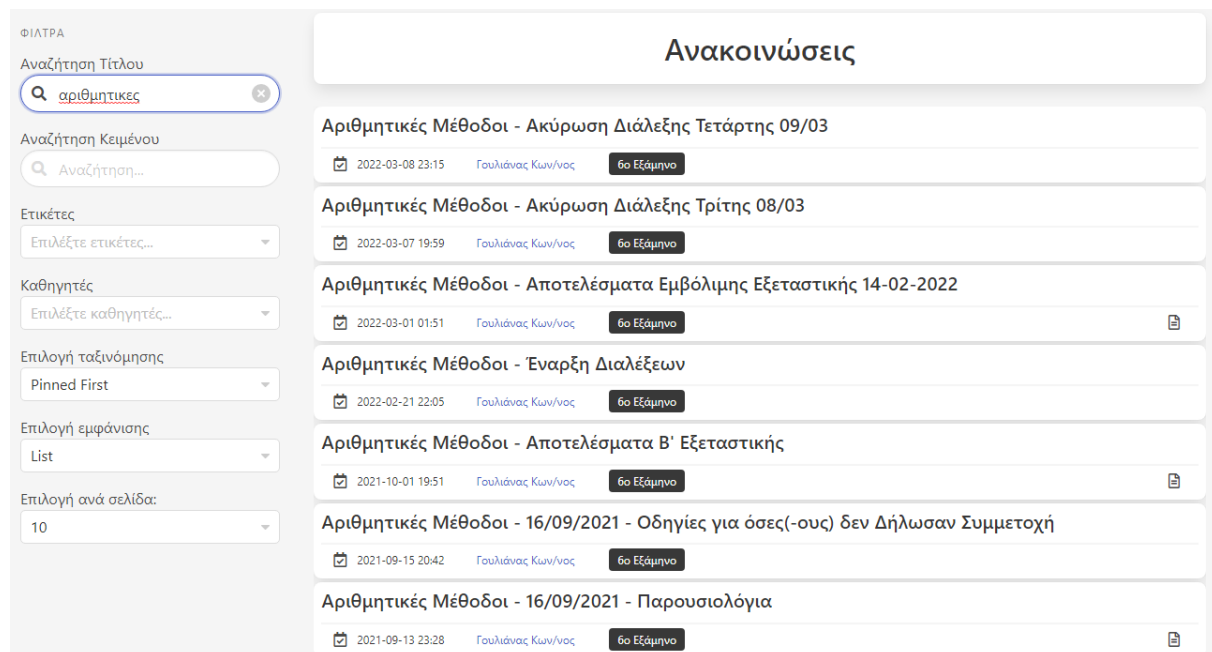
5.2 Περιγραφή των φίλτρων

Το πρώτο πράγμα που παρατηρεί κάποιος χρήστης είναι η ύπαρξη των φίλτρων στα αριστερά το οποίο επιτρέπει στον χρήστη να αναζητήσει τις ανακοινώσεις που θέλει συνδυάζοντας τα κατάλληλα φίλτρα. Οι επιλογές του χρήστη στο φιλτράρισμα και στην αναζήτηση χωρίζονται στα εξής:

Αναζήτηση βάση τίτλου

Στην προηγούμενη υλοποίηση της εφαρμογής η αναζήτηση γινόταν με έναν πιο δύσχρηστο τρόπο. Για να αναζητήσει ο χρήστης μία ανακοίνωση έπρεπε να εισάγει κάποια tags, τα οποία έπρεπε να τα γράφει ένα-ένα και για το καθένα έπρεπε να πατάει το πλήκτρο Enter για να το εισάγει. Στη συνέχεια αφού είχες τοποθετήσει όλα τα tags με τα οποία ήθελες να αναζητήσεις είχες την δυνατότητα να πατήσεις το κουμπί της αναζήτησης και να σε μεταφέρει σε μία καινούρια σελίδα με τα αποτελέσματα.

Πλέον στην σελίδα η αναζήτηση με βάση τον τίτλο γίνεται απλά γράφοντας τον επιθυμητό τίτλο, είτε με κεφαλαία είτε με μικρά, και απλά πατώντας το πλήκτρο Enter. Τα δεδομένα ανανεώνονται χωρίς να χρειάζεται να ξανά φορτώσει την σελίδα ο φυλλομετρητής.



Σχήμα 5.1: Αναζήτηση ανακοινώσεων βάση τίτλου

Η ανανέωση αυτή έχει ως αποτέλεσμα μία πιο φιλική προς τον χρήστη διαδικασία αναζήτησης καθώς και μείωση του χρόνου που απαιτείται για την ανάκτηση των αποτελεσμάτων.

Αναζήτηση βάση κειμένου

Προστέθηκε επίσης η δυνατότητα αναζήτησης με βάση το περιεχόμενο που έχει η κάθε ανακοίνωση. Η διαδικασία παραμένει ίδια με την αναζήτηση με βάση τον τίτλο.

Σχήμα 5.2: Αναζήτηση ανακοινώσεων βάση κειμένου

Φιλτράρισμα βάση ετικέτας

Προηγουμένως για να αναζητήσει κάποιος χρησιμοποιώντας τις ετικέτες έπρεπε να μεταφερθεί στη σελίδα των ετικετών και να επιλέξει μία ετικέτα με την οποία θα αναζητούσε τις ανακοινώσεις. Το πρόβλημα που είχε αυτή η υλοποίηση ήταν ότι ο χρήστης δεν μπορούσε να επιλέξει περισσότερες από μία ετικέτες.

Στην καινούρια υλοποίηση ο χρήστης έχει την δυνατότητα να επιλέξει όσες ετικέτες θέλει, καθώς επίσης και να αναζητήσει την ετικέτα την οποία ψάχνει. Επιπλέον προστέθηκε και η δυνατότητα να βλέπει ο χρήστης τον αριθμό των ανακοινώσεων που περιέχει η κάθε ετικέτα.

ΦΙΛΤΡΑ

Αναζήτηση Τίτλου
Αναζήτηση...

Αναζήτηση Κειμένου
Αναζήτηση...

Ετικέτες

- [50] Ερασμός
- [434] 1ο Εξάμηνο
- [69] Άλλα δημόσια νέα
- [276] Νέα τμήματος
- [6] 1ο Έτος
 - [468] 2ο Εξάμηνο
 - [3] 1205 Αντικειμενοστ...
- [0] 3ο Έτος
 - [443] 6ο Εξάμηνο
 - [0] 1602 Ενσωματωμέν...
- [1] 5ο Έτος
 - [517] 9ο Εξάμηνο
 - [0] 1949 Κατανομή...
- [276] Νέα τμήματος

Ανακοινώσεις

Νέα έκδοση προγράμματος μαθημάτων E2021-22
2022-03-08 11:44 Αντωνίου Ευστάθιος Νέα τμήματος

Δομημένος Προγραμματισμός - Αποτελέσματα Α' Εξαεστικής
2022-03-07 03:30 Γουλάνης Κων/νος 1ο Εξάμηνο

Συστήματα Διαχείρισης Βάσεων Δεδομένων: Κατανομή σε Εργαστηριακά Τμήματα
2022-03-04 16:47 Αρβανίτου Ελβίρα Μαρία 1ο Εξάμηνο

Ερευνητικό Εργαστήριο imselab – Παρουσιάσεις θεμάτων διπλωματικών εργασιών
2022-02-28 20:10 Κεραμόπουλος Ευκλείδης Νέα τμήματος

Βιντεο-διαλέξεις του Εισαγωγής στα Λειτουργικά Συστήματα
2022-02-25 10:54 Σιδηρόπουλος Αντώνης Άλλα δημόσια νέα

Ενημέρωση για εκπόνηση διπλωματικής/πτυχιακής (Εαρινό εξάμηνο)
2022-02-23 14:32 Περιοχής Χατζημπίος Άλλα δημόσια νέα

Εισαγωγή στην Επιστήμη των Υπολογιστών τελική βαθμολογία
2022-02-22 22:18 Ηλιοδής Χρήστος 1ο Εξάμηνο

Σχήμα 5.3: Φιλτράρισμα ανακοινώσεων βάση πολλαπλών ετικετών

Αυτό είχε σαν αποτέλεσμα να διαγραφεί η σελίδα που υπήρχε με τις ετικέτες και να γίνεται η αναζήτηση απευθείας, χωρίς να χρειάζεται να περιμένει ο χρήστης για να φορτώσει η σελίδα.

Φιλτράρισμα βάση καθηγητή

Για το φιλτράρισμα με βάση τους καθηγητές υπήρχε παρόμοια υλοποίηση με τις ετικέτες. Ο χρήστης έπρεπε πάλι να μεταφερθεί στην σελίδα με τους καθηγητές και να επιλέξει τον καθηγητή με τον οποίο ήθελε να κάνει την αναζήτηση.

Πλέον ο χρήστης, ακριβώς όπως με τις ετικέτες, μπορεί να αναζητήσει με βάση τους καθηγητές τους οποίους επιθυμεί. Η λειτουργία είναι ίδια με αυτή των ετικετών και υπάρχει και εδώ ο αριθμός των ανακοινώσεων που έχει αναρτήσει ο κάθε καθηγητής.

ΦΙΛΤΡΑ

Αναζήτηση Τίτλου
Αναζήτηση...

Αναζήτηση Κειμένου
Αναζήτηση...

Ετικέτες
Επιλέξτε ετικέτες...

Καθηγητές

- [87] Αδαμίδης Παναγιώτης
- [105] Αντωνίου Ευστάθιος
- [79] Ασδρέ Κατερίνα
- [90] Αβδελίδης Κωνσταντίνος
- [1] Αγγέλου Κωνσταντίνος
- [23] Αθανάσιος Ιωσήφιδης
- [22] Αλεξούδη Θεονότα
- [25] Αμπατζόγλου Αποστόλης
- [105] Αντωνίου Ευστάθιος
- [79] Ασδρέ Κατερίνα
- [0] Βαγιωνάς Χρήστος
- [127] Βασιλίδης Βίτσας
- [14] Βάσιος Βασίλειος

Ανακοινώσεις

Νέα έκδοση προγράμματος μαθημάτων E2021-22
2022-03-08 11:44 Αντωνίου Ευστάθιος Νέα τμήματος

Εργαστήρια Αντικειμενοστρεφούς Προγραμματισμού: Απαλλαγή από ΠΑΡΑΚΟΛΟΥΘΗΣΗ
2022-03-04 15:46 Ασδρέ Κατερίνα 2ο Εξάμηνο

Εργαστήρια Αντικειμενοστρεφούς Προγραμματισμού-Παράταση επιλογής τμήματος
2022-02-25 22:41 Ασδρέ Κατερίνα 2ο Εξάμηνο

Εργαστήρια Αντικειμενοστρεφούς Προγραμματισμού
2022-02-21 10:48 Ασδρέ Κατερίνα 2ο Εξάμηνο

Αντικειμενοστρεφής Προγραμματισμός - Κανόνες εργαστηρίων
2022-02-20 20:49 Αδαμίδης Παναγιώτης 2ο Εξάμηνο

Μαθηματικά II: Έναρξη μαθημάτων
2022-02-20 11:40 Αντωνίου Ευστάθιος 2ο Εξάμηνο

Αντικειμενοστρεφής Προγραμματισμός - Έναρξη μαθημάτων
2022-02-19 11:39 Αδαμίδης Παναγιώτης 2ο Εξάμηνο

Σχήμα 5.4: Φιλτράρισμα ανακοινώσεων βάση πολλαπλών καθηγητών

Και εδώ η σελίδα που υπήρχε με τους καθηγητές διαγράφηκε καθώς μειώθηκε αρκετά η ευκολία αναζήτησης τους.

Ταξινόμηση

Η ταξινόμηση προστέθηκε για να επιτρέψει στον χρήστη να επιλέγει ποιες ανακοινώσεις θέλει να βλέπει. Υπάρχει η δυνατότητα να βλέπει πρώτα τις σημαντικές ανακοινώσεις και από κάτω τις νεότερες. Αν ο χρήστης δεν επιθυμεί να βλέπει τις σημαντικές ανακοινώσεις ως πρώτες μπορεί να επιλέξει να βλέπει όλες τις νεότερες ανακοινώσεις. Τέλος υπάρχει και η επιλογή να βλέπει τις παλαιότερες ανακοινώσεις πρώτες.

The screenshot shows a web interface for announcements. On the left is a sidebar with filters: 'ΦΙΛΤΡΑ' (Filters), 'Αναζήτηση Τίτλου' (Title Search) with a search bar containing 'αριθμητικές', 'Αναζήτηση Κειμένου' (Text Search) with a search bar containing 'τετάρτη', 'Ετικέτες' (Tags) with a dropdown 'Επιλέξτε ετικέτες...', 'Καθηγητές' (Teachers) with a dropdown 'Επιλέξτε καθηγητές...', 'Επιλογή ταξινόμησης' (Sorting) with a dropdown menu showing 'Oldest', 'Pinned First', 'Newest', and 'Oldest' (selected), and 'Επιλογή ανά σελίδα' (Items per page) with a dropdown '10'. The main content area is titled 'Ανακοινώσεις' and contains three announcements:

- Αριθμητικές Μέθοδοι - ΠΡΟΑΙΡΕΤΙΚΗ ΕΡΓΑΣΙΑ - ΣΥΝΑΝΤΗΣΗ 4/5**: Έσες ΠΡΟΑΙΡΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ στο μάθημα Αριθμητικές Μέθοδοι είχαν υποβληθεί μέχρι χθες, παρουσιάστηκαν και εξετάστηκαν σήμερα 14:00-17:00. Αν κάποιοι εξακολουθούν να δουλεύουν σε κάποια εργασία, μπορούν να μου τη στείλουν μέχρι αύριο Τετάρτη 3/5/2020, 11:59 και να την παρουσιάσουν την Πέμπτη 3/5/2020. Διαφορετικά, την Πέμπτη θα επιλυθούν απορίες και θα μιλήσουμε για την εξεταστέα ύλη και το είδος των θεμάτων της γραπτής εξέτασης στο μάθημα. (2020-06-02 23:09, Γουλιάνας Κων/νος, 6ο Εξάμηνο)
- ΠΡΟΣΟΧΗ - Αριθμητικές Μέθοδοι - Έλεγχος Δήλωσης Μαθήματος**: Επειδή τα θέματα για όσες και όσους χρωστάνε ή παίρνουν για πρώτη φορά το μάθημα Αριθμητικές Μέθοδοι, θα είναι περισσότερα (άρα θα απαιτήσουν και περισσότερο χρόνο) από τα θέματα για όσες και όσους χρωστάνε ένα ΜΟΝΟ μέρος (Θεωρία ή Εργαστήριο) του μαθήματος Αριθμητική Ανάλυση και Προγραμματισμός Επιστημονικών Εφαρμογών, θα σας (2020-06-23 01:13, Γουλιάνας Κων/νος, 6ο Εξάμηνο)
- Αριθμητικές Μέθοδοι - Ακύρωση Διάλεξης Τετάρτης 09/03**: Η Διάλεξη της Τετάρτης 09/03 για το μάθημα Αριθμητικές Μέθοδοι δεν θα πραγματοποιηθεί λόγω προβλήματος του διδασκόντος. (2022-03-08 23:15, Γουλιάνας Κων/νος, 6ο Εξάμηνο)

Σχήμα 5.5: Ταξινόμηση ανακοινώσεων

Αλλαγή εμφάνισης

Στο κομμάτι της εμφάνισης υπήρχε μόνο μία επιλογή. Η επιλογή αυτή έδειχνε ένα κομμάτι τις ανακοίνωσης και ο χρήστης μπορούσε να επιλέξει να την ανοίξει χωρίς να αλλάξει την σελίδα. Αυτός ο τρόπος υιοθετήθηκε και στην καινούρια υλοποίηση προσθέτοντας την δυνατότητα να μπορεί ο χρήστης να κατεβάσει ή και να προβάλει τα συνημμένα της εκάστοτε ανακοίνωσης.

Κεφάλαιο 5

The screenshot shows a web interface for 'Ανακοινώσεις' (Announcements). On the left is a sidebar with filters: 'ΦΙΛΤΡΑ' (Filters), 'Αναζήτηση Τίτλου' (Title Search) with a search bar containing 'δοκ', 'Αναζήτηση Κειμένου' (Text Search) with a search bar containing 'Αναζήτηση...', 'Ετικέτες' (Tags) with a dropdown 'Επιλέξτε ετικέτες...', 'Καθηγητές' (Teachers) with a dropdown 'Επιλέξτε καθηγητές...', 'Επιλογή ταξινόμησης' (Sorting) with a dropdown 'Pinned First', 'Επιλογή εμφάνισης' (Display) with a dropdown 'Compact', and 'Επιλογή ανά σελίδα:' (Items per page) with a dropdown '10'. The main content area has a title 'Ανακοινώσεις' and a list of announcements. The first announcement is 'Δοκιμαστική Ανακοίνωση' (Mock Announcement) by 'ΣΙΔΗΡΟΠΟΥΛΟΣ demo ANTONΗΣ' dated '2022-03-09 10:33'. It has tags '1ο Εξάμηνο', '1102 Δομημένος Προγραμματισμός', '1105 Κυκλώματα Συνεχούς Ρεύματος', and '1ο Έτος'. Below it is a 'Δοκιμή' (Mock) section with the title 'Τίτλος' (Title) by 'Ευστάθιος Αντωνίου' dated '2021-09-30 12:56', with tags '1ο Εξάμηνο', '1101 Μαθηματικά I', and '1ο Έτος'. The third section is 'Δοκιμή νέου' (Mock new) by 'Δημήτριος Παπακώστας' dated '2021-09-29 20:15', with a tag 'Test board' and a link 'http://aboard.iee.ihu.gr/'.

Σχήμα 5.6: Εμφάνιση κειμένου ανακοινώσεων

Επιπλέον προστέθηκε και ένας ακόμα τρόπος εμφάνισης, της λίστας. Η λίστα περιέχει μόνο τον τίτλο της ανακοίνωσης και κάποιες πληροφορίες όπως π.χ. πότε ανέβηκε η ανακοίνωση, ποιος την ανέβασε, σε ποιες ετικέτες εμπεριέχεται και αν υπάρχουν συνημμένα.

The screenshot shows the same 'Ανακοινώσεις' (Announcements) page, but with a different display mode. The sidebar filters are the same. The main content area shows a list of announcements with only the title and some basic information. The first announcement is 'Δοκιμαστική Ανακοίνωση' (Mock Announcement) by 'ΣΙΔΗΡΟΠΟΥΛΟΣ demo ANTONΗΣ' dated '2022-03-09 10:33'. It has tags '1ο Εξάμηνο', '1102 Δομημένος Προγραμματισμός', '1105 Κυκλώματα Συνεχούς Ρεύματος', and '1ο Έτος'. Below it is a 'Δοκιμή' (Mock) section with the title 'Τίτλος' (Title) by 'Ευστάθιος Αντωνίου' dated '2021-09-30 12:56', with tags '1ο Εξάμηνο', '1101 Μαθηματικά I', and '1ο Έτος'. The third section is 'Δοκιμή νέου' (Mock new) by 'Δημήτριος Παπακώστας' dated '2021-09-29 20:15', with a tag 'Test board' and a link 'http://aboard.iee.ihu.gr/'.

Σχήμα 5.7: Εμφάνιση μόνο του τίτλου ανακοινώσεων

Και στους δύο τρόπους εμφάνισης στις σημαντικές ανακοινώσεις, επειδή τοποθετήθηκαν μαζί με όλες τις υπόλοιπες, τοποθετήθηκε ένα εικονίδιο για να τις ξεχωρίζει.

Αλλαγή ανακοινώσεων ανά σελίδα

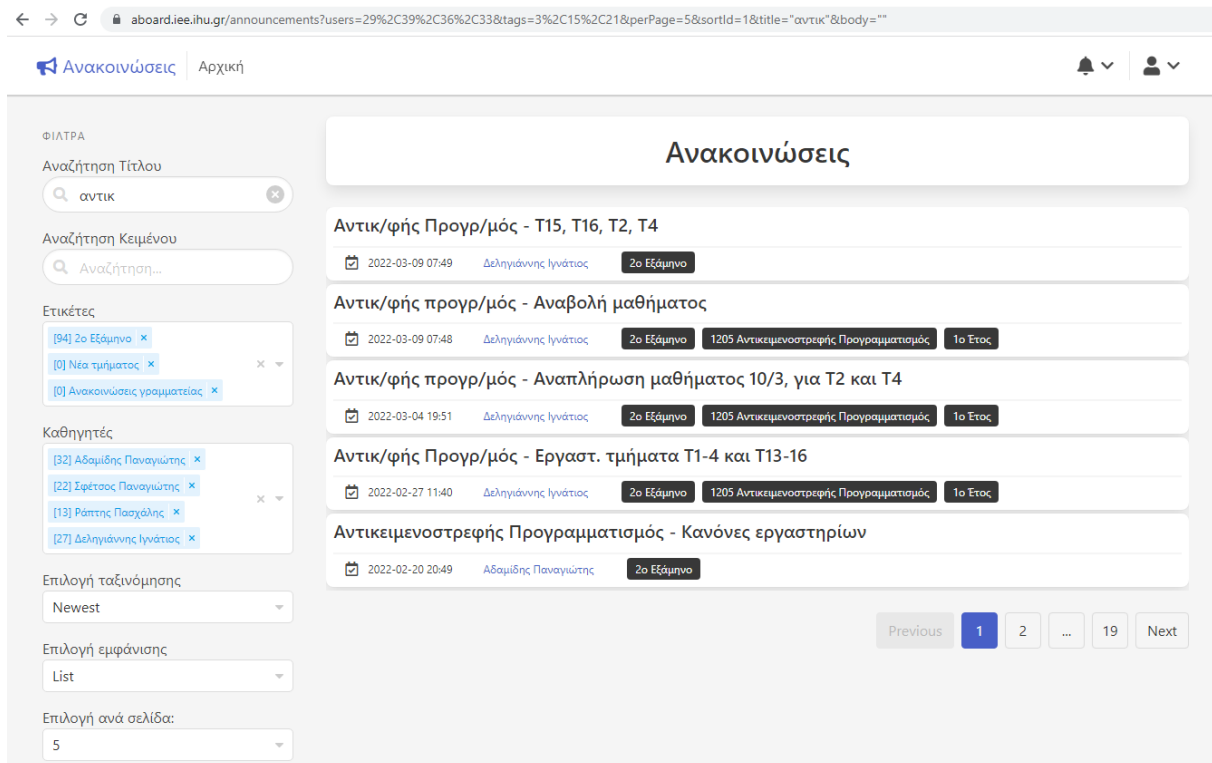
Τέλος προστέθηκε και η δυνατότητα ο χρήστης να μπορεί να επιλέξει πόσες ανακοινώσεις θέλει να βλέπει σε κάθε σελίδα.

The screenshot shows a web application interface for announcements. On the left is a sidebar with filters: 'ΦΙΛΤΡΑ' (Filters), 'Αναζήτηση Τίτλου' (Title Search), 'Αναζήτηση Κειμένου' (Text Search), 'Ετικέτες' (Tags), 'Καθηγητές' (Instructors), 'Επιλογή ταξινόμησης' (Sorting), 'Επιλογή εμφάνισης' (Display), and 'Επιλογή ανά σελίδα' (Items per page). The 'Επιλογή ανά σελίδα' dropdown is currently set to 5. The main area is titled 'Ανακοινώσεις' (Announcements) and displays a list of announcements. Each announcement includes a title, a date, a user, and a status. The announcements are: 1. 'Αντικ/φής Προγρ/μός - T15, T16, T2, T4' (2022-03-09 07:49, Δελιγιάννης Ιγνάτιος, 2ο Εξάμηνο). 2. 'Αντικ/φής προγρ/μός - Αναβολή μαθήματος' (2022-03-09 07:48, Δελιγιάννης Ιγνάτιος, 2ο Εξάμηνο, 1205 Αντικειμενοστρεφής Προγραμματισμός, 1ο Έτος). 3. 'Αρχές και Μέθοδοι Μηχανικής Μάθησης - Ακύρωση Διάλεξης Τετάρτης 09/03' (2022-03-08 23:18, Γουλιάνος Κων/νος, 8ο Εξάμηνο). 4. 'Αριθμητικές Μέθοδοι - Ακύρωση Διάλεξης Τετάρτης 09/03' (2022-03-08 23:15, Γουλιάνος Κων/νος, 6ο Εξάμηνο). 5. 'ΠΙΘΑΝΕΣ ΠΤΥΧΙΑΚΕΣ (ΣΑΛΑΜΠΑΣΗΣ)' (2022-03-08 21:00, Σαλαμπάσης Μιχαήλ, Πτυχιακή/Διπλωματική Εργασία). At the bottom right, there is a pagination control with 'Previous', '1', '2', '...', '997', and 'Next' buttons.

Σχήμα 5.8: Επιλογή αριθμού ανακοινώσεων ανά σελίδα

Τοποθέτηση φίλτρων στο url

Στην προηγούμενη ιστοσελίδα καθώς δεν υπήρχαν τα φίλτρα δεν υπήρχε και ο λόγος να αποθηκεύονται κάπου. Στην καινούρια υλοποίηση όμως ήταν κάτι το οποίο θα μπορούσε να γίνει και θα ήταν αρκετά εύχρηστο, καθώς ο χρήστης αφού επιλέξει τα φίλτρα που επιθυμεί μπορεί να στείλει την σελίδα στην οποία βρίσκεται σε κάποιον άλλο χρήστη στέλνοντας μαζί και τα φίλτρα που είχε τοποθετήσει. Με αυτόν τον τρόπο η σελίδα μόλις φορτώσει θα τοποθετηθούν τα φίλτρα τα οποία βρίσκονταν μέσα στον σύνδεσμο, επιτρέποντας τον χρήστη να βρει ευκολότερα την ανακοίνωση που έψαχνε. Επίσης επιτρέπει στον χρήστη να επιστρέψει στις ανακοινώσεις από οποιαδήποτε σελίδα χωρίς να έχει χάσει τα καταχωρημένα φίλτρα.



Σχήμα 5.9: Δυνατότητα τοποθέτησης φίλτρων στον σύνδεσμο

Κατά την επιλογή των φίλτρων η σελίδα δεν ανανεώνεται, καλεί μόνο το `api` για την ενημέρωση των δεδομένων. Στην JS υπήρχαν δύο τρόποι να τοποθετήσουμε τα φίλτρα στο `url`. Ο ένας τρόπος ήταν να ανανεώνεται ο σύνδεσμος ανανεώνοντας και την σελίδα μαζί. Ο άλλος τρόπος ήταν να ανανεώνεται ο σύνδεσμος χωρίς να απαιτείται ανανέωση από την σελίδα. Λόγω της υλοποίησης των φίλτρων ήταν πιο πρακτικό να ανανεώνουμε μόνο τον σύνδεσμο. Αυτό γίνεται με την βοήθεια της ιδιότητας ``window.history.pushState()``.

Αφού τοποθετηθούν τα φίλτρα στο σύνδεσμο πρέπει να μπορούμε όταν τον στείλουμε κάπου αλλού να γίνεται αρχικοποίηση των τιμών. Αυτό γίνεται ελέγχοντας αν υπάρχουν παράμετροι μέσα στο σύνδεσμο και να τις θέσει στα δεδομένα που έχει το αντίστοιχο `component` με την βοήθεια του ``URLSearchParams()``.

Επιλογή συγγραφέα και ετικέτας από την ανακοίνωση

Στην προηγούμενη υλοποίηση, όταν ο χρήστης επιθυμούσε να δει τις ανακοινώσεις βάσει μίας ετικέτας ή ενός συγγραφέα, μπορούσε να πατήσει στο ανάλογο πεδίο που εμφανιζόταν στις πληροφορίες κάτω από κάθε ανακοίνωση. Λόγω της αλλαγής του τρόπου φιλτραρίσματος των ανακοινώσεων, έπρεπε να γίνει μετατροπή αυτής της λειτουργίας για να χρησιμοποιηθεί με την καινούρια υλοποίηση. Πλέον ο χρήστης όταν επιλέγει τον συγγραφέα της ανακοίνωσης ή μίας από τις ετικέτες της, ενημερώνεται η διεύθυνση `url`, χωρίς να ανανεώνεται η σελίδα, με το επιλεγμένο πεδίο και θέτει τις ανάλογες τιμές στα φίλτρα.

The screenshot shows the 'Ανακοινώσεις' (Announcements) page. On the left, there is a sidebar with filters: 'ΦΙΛΤΡΑ' (Filters), 'Αναζήτηση Τίτλου' (Title Search), 'Αναζήτηση Κειμένου' (Text Search), 'Ετικέτες' (Tags) with '[11] Πτυχιακή/Διπλωματική Εργασία', 'Καθηγητές' (Teachers) with '[11] Κεραμόπουλος Ευκλείδης', 'Επιλογή ταξινόμησης' (Sorting) set to 'Pinned First', 'Επιλογή εμφάνισης' (Display) set to 'List', and 'Επιλογή ανά σελίδα:' (Items per page) set to '10'. The main content area lists several announcements, including 'Πρόγραμμα Εξέτασης ΔΕ/ΠΕ - Φεβρουάριος 2022', 'Παρουσιάσεις ΔΕ/ΠΕ - Φεβρουάριος 2022', 'Πρακτικό Επιτροπής ΒΠΕ - ΙΟΥΛ. 2021', 'Υποβολή υποψηφιότητας Βέλτιστης Διπλωματικών/Πτυχιακής Εργασίας', 'Πρόγραμμα παρουσίασης ΔΕ/ΠΕ Ιούλιος 2021', 'Παρουσίαση ΔΕ/ΠΕ – Εαρινό εξάμηνο 2021', and 'Αναθέσεις ΔΕ/ΠΕ Εαρινού Εξαμήνου 2021'. Each announcement includes a date, a teacher's name, and a category tag.

Σχήμα 5.10: Δυνατότητα επιλογής φιλτραρίσματος μέσω της ανακοίνωσης

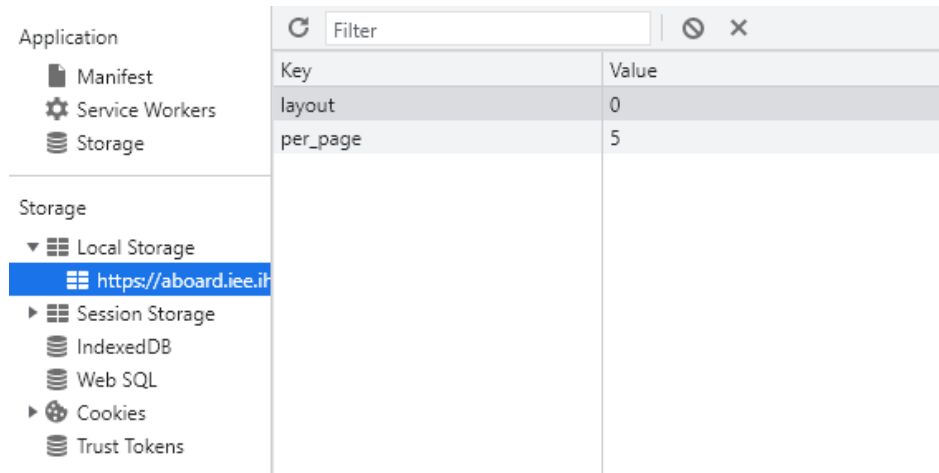
Λοιπές λειτουργίες

Ο χρήστης πλέον έχει την δυνατότητα να κάνει συνδυασμό των φίλτρων του. Επειδή πλέον η αναζήτηση και το φιλτράρισμα γίνονται κάτω από το ίδιο αρι δίνεται η δυνατότητα στο χρήστη να ψάξει τις ανακοινώσεις που επιθυμεί χρησιμοποιώντας τον τίτλο, το περιεχόμενο, τις ετικέτες της ανακοίνωσης, του συγγραφείς της, καθώς και να αλλάξει τον τρόπο με τον οποίο βλέπει τα αποτελέσματα. Αυτό έχει σαν αποτέλεσμα μία πιο άμεση εμφάνιση αποτελεσμάτων και πιο αποτελεσματική αναζήτηση.

The screenshot shows the 'Ανακοινώσεις' (Announcements) page with multiple filters applied. The sidebar filters include: 'ΦΙΛΤΡΑ' (Filters), 'Αναζήτηση Τίτλου' (Title Search) with 'εν', 'Αναζήτηση Κειμένου' (Text Search) with 'χε', 'Ετικέτες' (Tags) with '[1] Πτυχιακή/Διπλωματική Εργασία', '[41] Νέα τμήματος', '[0] Παρουσιάσεις πτυχικών', '[0] ΠΜΣ Ευφυείς Τεχνολογίες Διαδικτύου', and '[10] Πρακτική άσκηση', 'Καθηγητές' (Teachers) with '[2] Βασιλης Βίτσας', '[30] Γουλιάνος Κων/νος', '[12] Δημήτριος Παπακώστας', '[0] Δημήτριος Αμανατιάδης', '[3] Ιορδάνης Κιοσκερίδης', '[4] Γιακουμής Άγγελος', and '[1] Αντωνίου Ευστάθιος', 'Επιλογή ταξινόμησης' (Sorting) set to 'Newest', 'Επιλογή εμφάνισης' (Display) set to 'List', and 'Επιλογή ανά σελίδα:' (Items per page) set to '5'. The main content area lists several announcements, including 'ΠΡΟΣΚΛΗΣΗ ΕΚΔΗΛΩΣΗΣ ΕΝΔΙΑΦΕΡΟΝΤΟΣ ΠΡΑΚΤΙΚΗΣ ΑΣΚΗΣΗΣ ΕΑΡΙΝΟΥ ΕΞΑΜΗΝΟΥ 2020', 'Έναρξη εαρινού εξαμήνου 2019-2020 και Δηλώσεις μαθημάτων', 'ΗΜΕΡΕΣ - ΩΡΕΣ ΕΝΙΣΧΥΤΙΚΗΣ ΔΙΔΑΣΚΑΛΙΑΣ ΣΤΑ ΜΑΘΗΜΑΤΑ ΤΟΥ ΠΡΩΤΟΥ ΕΞΑΜΗΝΟΥ', 'Πρώτο σεμινάριο επιμόρφωσης των μεντόρων του δικτύου ΟΜΠΡΕΛΑ, 20/11/2019, 15:00', and 'Πρόγραμμα Εξέτασης Διπλωματικών Εργασιών, ΠΜΣ-Εφαρμοσμένα Ηλεκτρονικά Συστήματα'. At the bottom, there is a pagination bar with 'Previous', '1', '...', '6', '7', '8', '...', '11', and 'Next'.

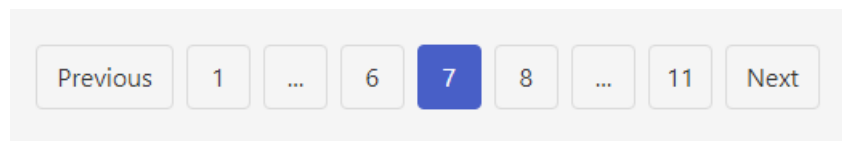
Σχήμα 5.11: Συνδυασμός φίλτρων

Την επιλογή ανακοινώσεων ανά σελίδα και της εμφάνισης αποφασίσαμε να την αποθηκεύουμε στον χρήστη μέσω του Local Storage, καθώς αν έπρεπε να επιλέγει ο χρήστης κάθε φορά το πώς ήθελε να βλέπει τα δεδομένα δεν θα ήταν αρκετά εύχρηστο σαν λειτουργία. Με αυτόν τον τρόπο ο χρήστης κάθε φορά που μπαίνει στην ιστοσελίδα, ανεξαρτήτως αν είναι συνδεδεμένος ή όχι, θα βλέπει τις ανακοινώσεις με τον τρόπο που έχει επιλέξει.



Σχήμα 5.12: Αποθήκευση φίλτρων εμφάνισης στο local storage

Επιπλέον έχει αλλάξει και ο τρόπος με τον οποίο γινόταν το pagination στην σελίδα. Πλέον ο χρήστης μπορεί, εκτός από την επόμενη και προηγούμενη σελίδα, να μεταφερθεί απευθείας στην πρώτη και στην τελευταία σελίδα.

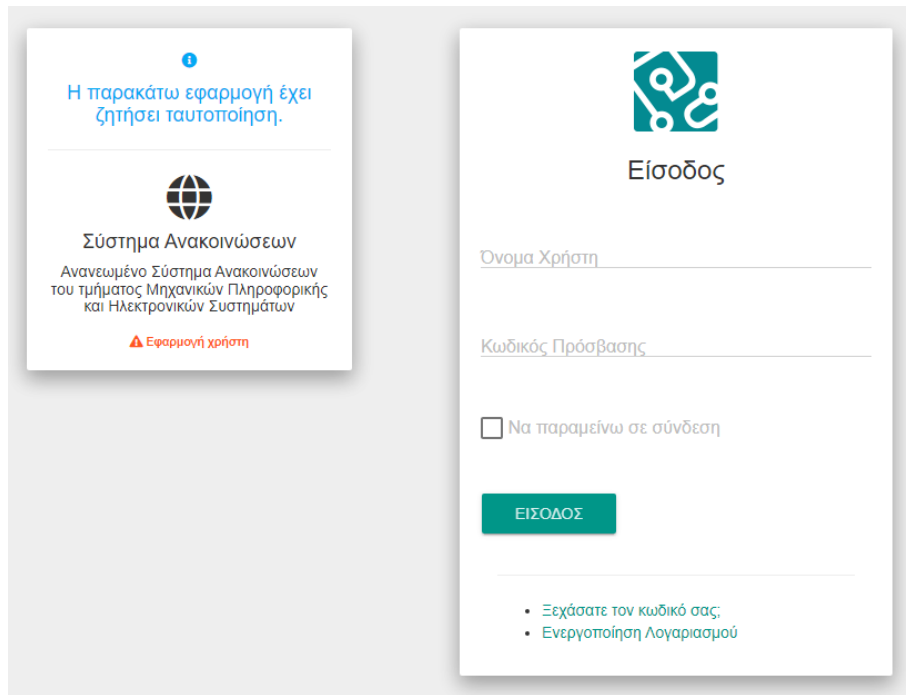


Σχήμα 5.13: Ανανεωμένο σύστημα σελιδοποίησης

5.3 Σύνδεση χρήστη μέσω του τμήματος

Στην προηγούμενη ιστοσελίδα είχε γίνει μία προσπάθεια υλοποίησης του OAuth2.0 πρωτοκόλλου που χρησιμοποιεί το τμήμα μας. Τελικά όμως χρησιμοποιήθηκε μία άλλη μέθοδος σύνδεσης μέσω του LDAP διακομιστή, μέσα στον οποίο ήταν αποθηκευμένοι οι χρήστες. Αυτό είχε σαν αποτέλεσμα η σύνδεση να μην είναι τόσο ασφαλής και δημιουργούσε προβλήματα όταν έληγε το access token του χρήστη.

Σε αυτή την υλοποίηση καταφέραμε να λειτουργήσουμε το πρωτόκολλο OAuth2.0 για να επιτευχθεί ένας πιο ασφαλής τρόπος σύνδεσης στην εφαρμογή. Για να γίνει αυτό χρειάστηκε να γίνει η εγκατάσταση ενός πακέτου της Laravel, το Socialite. Το πρόβλημα που συναντήσαμε με το πακέτο αυτό ήταν ότι υποστηρίζει αυθεντικοποίηση μόνο για συγκεκριμένους παρόχους (πχ, Facebook, GitHub, Twitter, LinkedIn, Google, GitLab, και το BitBucket).



Σχήμα 5.14: Αυθεντικοποίηση μέσω του τμήματος

Στο Κεφάλαιο 3 αναλύθηκε ο τρόπος εγκατάστασης του OAuth2.0 μέσω του πακέτου Socialite, υλοποιώντας τις απαραίτητες μεθόδους και τοποθετώντας τα απαραίτητα στοιχεία που χρειάζεται το πακέτο για να επιτευχθεί η αυθεντικοποίηση μέσω του πρωτόκολλου OAuth.

5.4 Δυνατότητα εμφάνισης όλων των ανακοινώσεων

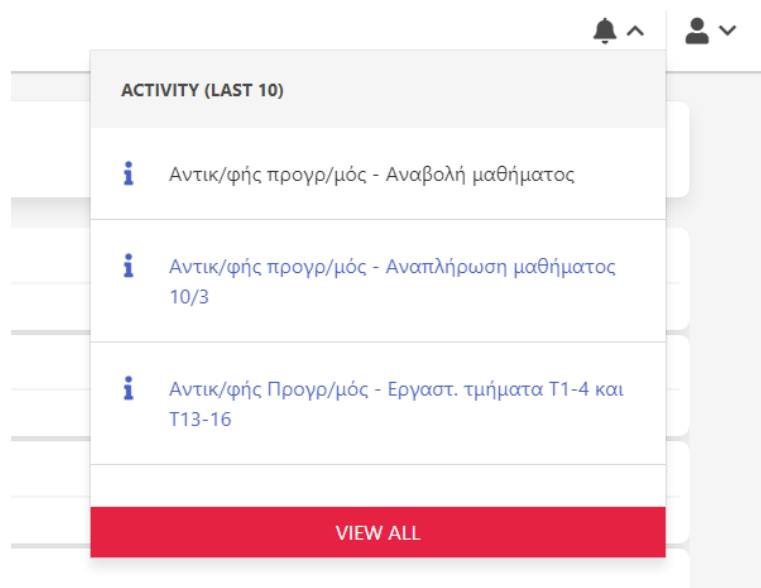
Ένας μη συνδεδεμένος χρήστης έχει την δυνατότητα να δει όλες τις ανεβασμένες ανακοινώσεις, ανεξάρτητα αν είναι δημόσιες ή όχι, μέσα από το δίκτυο του τμήματος. Αυτό μπορεί να γίνει είτε μέσω απομακρυσμένης σύνδεσης στο δίκτυο του τμήματος, είτε μέσω απευθείας σύνδεσης στο δίκτυο του τμήματος.

5.5 Δυνατότητες συνδεδεμένου χρήστη

Με την σύνδεση του ο χρήστης μπορεί πλέον να βλέπει όλες τις ανακοινώσεις που ανεβάζει το τμήμα. Το φιλτράρισμα πλέον αφορά όλες τις ανακοινώσεις και όχι μόνο τις δημόσιες.

Ειδοποιήσεις

Πλέον ο χρήστης μπορεί να λαμβάνει ειδοποίηση σχετικά με την κάθε ανακοίνωση που έχει δημιουργηθεί. Η ειδοποίηση αυτή εμφανίζεται στο καμπανάκι που βρίσκεται στο πάνω μέρος της ιστοσελίδας. Πατώντας το καμπανάκι εμφανίζονται οι τελευταίες δέκα ειδοποιήσεις που έχουν δημιουργηθεί και σχετίζονται με τις εγγεγραμμένες ετικέτες του χρήστη. Πατώντας πάνω σε μία ειδοποίηση μεταφερόμαστε μέσα στην ανακοίνωση.



Σχήμα 5.15: Εμφάνιση των τελευταίων ειδοποιήσεων ενός συνδεδεμένου χρήστη

Προτιμήσεις χρήστη

Δίπλα από το καμπανάκι υπάρχει το προφίλ. Πατώντας στο εικονίδιο ο χρήστης βλέπει τα στοιχεία του και έχει την δυνατότητα να πάει στις προτιμήσεις χρήστη ή να αποσυνδεθεί.

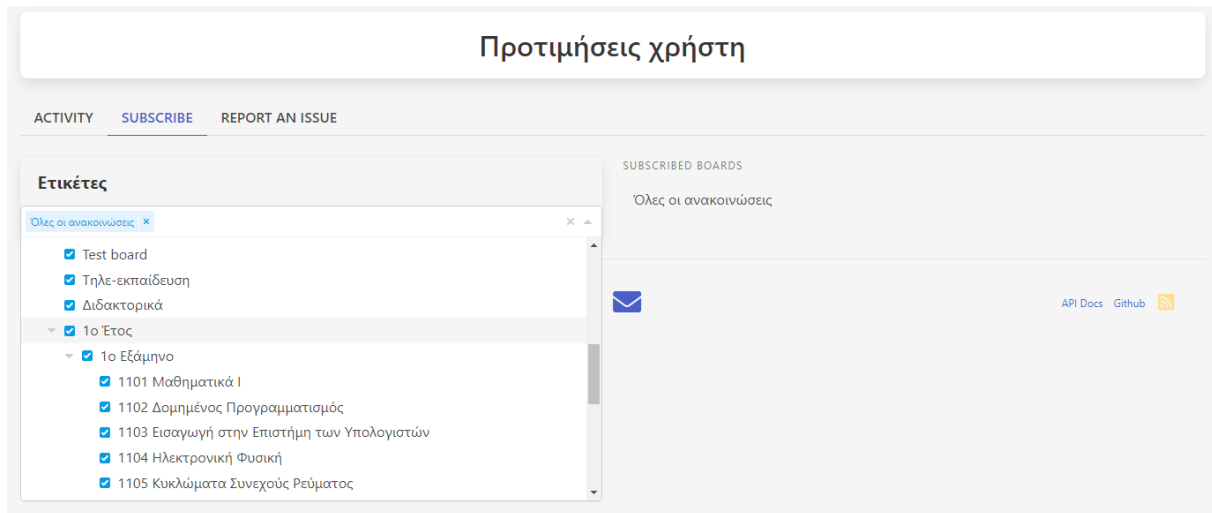
Αφού πατήσει τις προτιμήσεις, μεταφέρεται σε μία άλλη σελίδα όπου περιέχει τρεις καρτέλες: Activity, Subscribe, Report an Issue.

Στην καρτέλα Activity, μπορεί να δει όλες τις δραστηριότητες που έχουν έρθει ως ειδοποιήσεις. Εμφανίζεται ο τύπος της ειδοποίησης, το link με τον τίτλο της, τον χρήστη που την δημιούργησε και την ημερομηνία δημιουργίας.

Προτιμήσεις χρήστη			
ACTIVITY	SUBSCRIBE	REPORT AN ISSUE	
TYPE	LINK	USER	DATE
announcement.created	Αντικ/φής προγρ/μός - Αναβολή μαθήματος	Δεληγιάννης Ιγνάτιος	2022-03-09 07:48
announcement.created	Αντικ/φής προγρ/μός - Αναπλήρωση μαθήματος 10/3	Δεληγιάννης Ιγνάτιος	2022-03-04 19:39
announcement.created	Αντικ/φής Προγρ/μός - Εργαστ. τμήματα T1-4 και T13-16	Δεληγιάννης Ιγνάτιος	2022-02-27 11:05

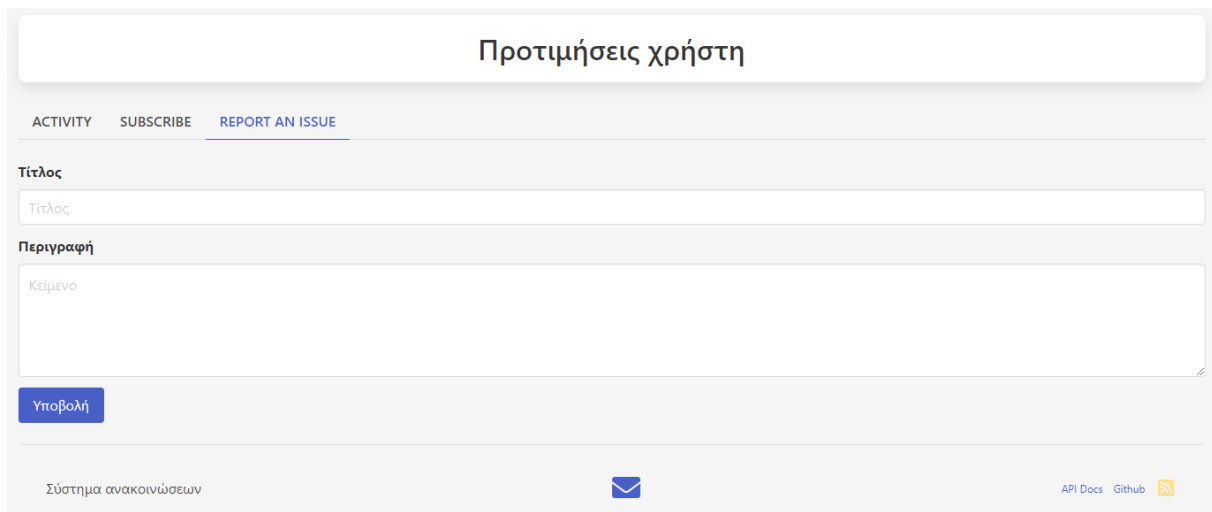
Σχήμα 5.16: Εμφάνιση όλων των ειδοποιήσεων

Στην καρτέλα Subscribe, μπορεί να επιλέξει τις ετικέτες στις οποίες θέλει να εγγραφεί για να λαμβάνει ειδοποιήσεις μέσα στην εφαρμογή αλλά και μέσω του ηλεκτρονικού του email. Η εγγραφή του χρήστη γίνεται μέσω ενός select box μέσα στο οποίο υπάρχουν οι ετικέτες σε μορφή δέντρου. Επιλέγοντας μία ετικέτα ο χρήστης εγγράφεται αυτόματα και σε όλα τα παιδιά που έχει η ετικέτα. Επίσης υπάρχει και η δυνατότητα να αναζητήσει την ετικέτα που επιθυμεί, καθώς και να διαγράψει όλες τις ετικέτες με ένα κλικ ή να διαγράψει συγκεκριμένη ετικέτα με τα παιδιά της.



Σχήμα 5.17: Δυνατότητα εγγραφής σε πολλαπλές ετικέτες

Τέλος στην καρτέλα Report an Issue, ο χρήστης μπορεί να στείλει στον διαχειριστή για οποιοδήποτε πρόβλημα υπάρχει με την εφαρμογή.



Σχήμα 5.18: Δυνατότητα υποβολής ενός προβλήματος

API Documentation

Στο Documentation ο χρήστης μπορεί να δει τα routes για το API της ιστοσελίδας εμφανίζοντας τον τύπο και το route μαζί με τις παραμέτρους που δέχεται. Επιπλέον υπάρχει και ένα παράδειγμα με την απάντηση του διακομιστή. Η κλειδαριά στο πλάι δείχνει αν το route είναι δημόσιο ή απαιτεί σύνδεση του χρήστη, ενώ το εικονίδιο με την ασπίδα δείχνει ότι μόνο ο admin έχει πρόσβαση σε εκείνο το route.

API Documentation		
Announcements		
>	GET /api/announcements?users[]=userId1&users[]=userId2&tags[]=tagId1&tags[]=tagId2&perPage=5&sortId=0&q=%22words+words%22	🔒
>	GET /api/announcements/{id}	🔒
>	POST /api/announcements	🔒
>	PUT /api/announcements/{id}	🔒
>	DELETE /api/announcements/{id}	🔒
Attachments		
>	GET /api/announcements/{id}/attachments/{id}	🔒
Tags		
>	GET /api/tags	🔒
>	GET /api/filtertags?users[]=userId1&users[]=userId2&q=%22words+words%22	🔒
>	GET /api/tags/{id}	🔒
>	POST /api/tags	🔒 👤

Σχήμα 5.19: Documentation για το API της εφαρμογής

Εκτός από το API υπάρχουν και οι πίνακες της ιστοσελίδας εμφανίζοντας τα πεδία που περιέχουν, καθώς και τον τύπο των πεδίων.

Models	
Primary Models	
Announcement	>
Tag	>
Attachment	>
Notification	>
User	>
Relation Models	
Announcement_Tag	▼
<pre>{ announcement_id bigint(20) p_key f_key tag_id bigint(20) p_key f_key }</pre>	
Tag_User	>

Σχήμα 5.20: Documentation για τους πίνακες της βάσης

5.6 Δυνατότητες συγγραφέα

Ο συγγραφέας, πέραν από τις δυνατότητες που αναφέραμε παραπάνω, έχει την δυνατότητα να δημιουργήσει ανακοινώσεις και να επεξεργαστεί ή να διαγράψει ανακοινώσεις που έχει δημιουργήσει ο ίδιος.

Δημιουργία ανακοίνωσης

Επιλέγοντας προσθήκη ανακοίνωσης ο συγγραφέας μπορεί να ορίσει τον τίτλο και το κείμενο της ανακοίνωσης στα ελληνικά αλλά και στα αγγλικά επιλέγοντας το checkbox που βρίσκεται από κάτω.

Προσθήκη ανακοίνωσης

Τίτλος Ανακοίνωσης

Τίτλος

Κείμενο Ανακοίνωσης

B I H1 H2 H3

☒ Προσθήκη ανακοίνωσης στα αγγλικά

Τίτλος Ανακοίνωσης Στα Αγγλικά

Τίτλος

Κείμενο Ανακοίνωσης Στα Αγγλικά

B I H1 H2 H3

Σχήμα 5.21: Δημιουργία ανακοίνωσης 1/2

Παρακάτω υπάρχει η επιλογή ετικετών με την οποία ορίζονται οι ετικέτες που θα περιέχει η ανακοίνωση. Οι γονείς των ετικετών εισάγονται αυτόματα κατά την επιλογή μίας ετικέτας.

Επίσης υπάρχει και η δυνατότητα προσθήκης της ανακοίνωσης στις σημαντικές, έτσι ώστε να εμφανίζεται πριν από τις κανονικές ανακοινώσεις για το χρονικό διάστημα που έχει επιλεγεί.

Επιπλέον υπάρχει και η επιλογή να προστεθούν στην ανακοίνωση συνημμένα αρχεία. Η λειτουργία αυτή δέχεται διάφορους τύπους αρχείων (πχ. Word, Pdf, PowerPoint, Excel, κ.α.).

Τέλος υπάρχει και η δυνατότητα προσθήκης εκδήλωσης, όπου μπορεί ο συγγραφέας να επιλέξει την τοποθεσία της εκδήλωσης, να προσθέσει το link στο Google Maps και να ορίσει την έναρξη και την λήξη της εκδήλωσης.

Επιλογή Ετικετών

Αναζήτηση...

- ☒ 1101 Μαθηματικά I
- ☐ 1102 Δομημένος Προγραμματισμός
- ☒ 1103 Εισαγωγή στην Επιστήμη των Υπολογιστών
- ☐ 1104 Ηλεκτρονική Φυσική
- ☐ 1105 Κυκλώματα Συνεχούς Ρεύματος
- ☐ 1201 Μαθηματικά II
- ☐ 1202 Μετρήσεις και Κυκλώματα Εναλλασσόμενου Ρεύματος

☒ Προσθήκη ανακοίνωσης στις σημαντικές

Ανέβασμα Αρχείων

Επιλογή αρχείου...

☒ Προσθήκη εκδήλωσης

Τοποθεσία Εκδήλωσης

ΔΙΠΑΕ

☐ Προσθήκη link στο Google Maps

Έναρξη Εκδήλωσης

March 10, 2022 3:00 PM

Λήξη Εκδήλωσης

March 10, 2022 6:00 PM

Επιλεγμένες Ετικέτες (Auto-Inserted)

1101 Μαθηματικά I 1ο Εξάμηνο 1ο Έτος Όλες οι ανακοινώσεις

1103 Εισαγωγή στην Επιστήμη των Υπολογιστών

Εμφάνιση Μέχρι

March 17, 2022 12:00 AM

Λίστα Αρχείων

RECTANG_sketch4.pdf × B5 (a).xls ×

Αποθήκευση

Σχήμα 5.22: Δημιουργία ανακοίνωσης 2/2

Επεξεργασία ανακοίνωσης

Ο συγγραφέας επιλέγοντας μία ανακοίνωση που έχει δημιουργήσει ο ίδιος, έχει την δυνατότητα να την επεξεργαστεί όπως ακριβώς γίνεται και η δημιουργία που αναφέραμε προηγουμένως με τη διαφορά ότι στην επεξεργασία συμπληρώνονται αυτόματα οι πληροφορίες της υπάρχουσας ανακοίνωσης.

Διαγραφή ανακοίνωσης

Ο συγγραφέας επιλέγοντας μία ανακοίνωση που έχει δημιουργήσει ο ίδιος, έχει την δυνατότητα να την διαγράψει.

Δοκιμαστική Ανακοίνωση

Δοκιμαστική Ανακοίνωση Κείμενο

1ο Εξάμηνο 1102 Δομημένος Προγραμματισμός 1105 Κυκλώματα Συνεχούς Ρεύματος 1ο Έτος

2022-03-09 10:33 · ΣΙΔΗΡΟΠΟΥΛΟΣ demo ANTQNHZ

< Αρχική Επεξεργασία Διαγραφή

Σχήμα 5.23: Δυνατότητα διαγραφής και επεξεργασίας για τον συγγραφέα των ανακοινώσεων του

5.7 Δυνατότητες διαχειριστή

Ο διαχειριστής, πέραν από τις δυνατότητες του απλού χρήστη και του συγγραφέα, έχει την δυνατότητα να ανεβάζει ανακοινώσεις ως άλλος χρήστης, να επεξεργάζεται και να διαγράφει ανακοινώσεις που

έχουν ανεβάσει άλλοι συγγραφείς, να διαχειρίζεται τις ετικέτες και να βλέπει τα Issues που έχουν δημιουργηθεί.

Δημιουργία ανακοίνωσης

Ο διαχειριστής έχει όλες τις δυνατότητες που έχει ο συγγραφέας για την δημιουργία της ανακοίνωσης. Αυτό που προστίθεται είναι η δυνατότητα να ανεβάσει την ανακοίνωση χρησιμοποιώντας το όνομα ενός άλλου συγγραφέα.























Σχήμα 5.24: Ανέβασμα ανακοίνωσης ως άλλος χρήστης

Επεξεργασία και διαγραφή ανακοίνωσης

Επιλέγοντας οποιαδήποτε ανεβασμένη ανακοίνωση ο διαχειριστής έχει την δυνατότητα να την επεξεργαστεί ή και να την διαγράψει. Στην περίπτωση της επεξεργασίας η διαδικασία παραμένει ίδια με την δημιουργία.

Διαχείριση ετικετών

Στην αρχική σελίδα και πάνω από όλες τις ανακοινώσεις βρίσκεται το κουμπί Admin Area. Πατώντας το κουμπί ο διαχειριστής έχει την επιλογή να δημιουργήσει, να επεξεργαστεί ή και να διαγράψει μία ετικέτα.

Διαχείριση ετικετών	
Προσθήκη Ετικέτας	
Όνομα	Διαχείριση
1101 Μαθηματικά I	 
1102 Δομημένος Προγραμματισμός	 
1103 Εισαγωγή στην Επιστήμη των Υπολογιστών	 
1104 Ηλεκτρονική Φυσική	 
1105 Κυκλώματα Συνεχούς Ρεύματος	 
1201 Μαθηματικά II	 
1202 Μετρήσεις και Κυκλώματα Εναλλασσόμενου Ρεύματος	 
1203 Τεχνική Συγγραφή, Παρουσίαση και Ορολογία Ξένης Γλώσσας	 
1204 Σχεδίαση Ψηφιακών Συστημάτων	 
1205 Αντικειμενοστρεφής Προγραμματισμός	 
1301 Θεωρία Πιθανοτήτων και Στατιστική	 
Exit	

Σχήμα 5.25: Διαχείριση ετικετών

Δημιουργώντας μία ετικέτα ο χρήστης επιλέγει το όνομα της ετικέτας, το όνομα για την mail λίστα, αν η ετικέτα θα είναι δημόσια ή όχι και τέλος αν η ετικέτα θα ανήκει σε μία γονική ετικέτα.

Προσθήκη ετικέτας	
Όνομα ετικέτας	
<input type="text" value="Όνομα ετικέτας"/>	
Όνομα λίστας mail	
<input type="text" value="Όνομα λίστας mail"/>	
Δημόσιο: <input type="checkbox"/> Yes	
Γονική ετικέτα	
<input type="text" value=""/>	
Αποθήκευση	Ακύρωση

Σχήμα 5.26: Προσθήκη/Επεξεργασία ετικέτας




Αντίστοιχα κατά την επεξεργασία μίας ετικέτας υπάρχει η δυνατότητα αλλαγής κάποιων στοιχείων της ετικέτας.

Επίσης υπάρχει και η δυνατότητα διαγραφής σε περίπτωση που κριθεί απαραίτητο.

Προβολή προβλημάτων

Στην προβολή προβλημάτων εμφανίζονται όλα τα προβλήματα που έχουν αναφερθεί από τους χρήστες της εφαρμογής με σκοπό την άμεση επίλυσή τους.

Προβολή προβλημάτων

Τίτλος	Περιγραφή	Διαγραφή
2	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec posuere sagittis hendrerit. Proin accumsan sodales leo. Fusce auctor quam libero, non congue velit euismod quis. In congue placerat odio nec porttitor. Nunc non sem tincidunt, volutpat libero eu, mollis ipsum. Donec convallis odio quis eros laoreet mattis. Integer sem risus, consectetur viverra placerat semper, elementum ac metus. Nulla sit amet purus orci. Proin consectetur lobortis metus, et posuere mauris efficitur vitae. Phasellus fermentum tellus congue pharetra lobortis. Donec in nibh at ligula semper suscipit sed ac sem. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae;	
3	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec posuere sagittis hendr...	
4	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec posuere sagittis hendr...	

Exit

Σχήμα 5.27: Εμφάνιση προβλημάτων

5.8 Επίλογος

Σε αυτό το κεφάλαιο έγινε μία παρουσίαση της εφαρμογής αναλύοντας τις καινούριες αλλά και τις υπάρχουσες λειτουργίες του project. Η παρουσίαση αφορούσε την οπτική όλων των χρηστών (πχ. μη συνδεδεμένου και συνδεδεμένου χρήστη, συγγραφέα και διαχειριστή).

Κεφάλαιο 6ο: Συμπεράσματα ή/και προτάσεις βελτίωσης

6.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα παρατεθούν κάποιες προτάσεις βελτίωσης για το project καθώς και κάποια συμπεράσματα ως προς την υλοποίηση της πτυχιακής και της εφαρμογής αντίστοιχα.

6.2 Μελλοντικές προτάσεις βελτίωσης

Στο project αυτό κληθήκαμε να βελτιώσουμε μία αρκετά καλή υλοποίηση που είχε γίνει από τον συνάδελφό μας τον Νικολαΐδη Νικόλα-Χρήστο. Οι απαιτήσεις της εφαρμογής αφορούσαν προτάσεις που είχαν γίνει από τους καθηγητές του τμήματος, αλλά και από τον Νίκο-Χρήστο Νικολαΐδη. Στόχος μας ήταν να τις υλοποιήσουμε όσο το δυνατόν πιο αποτελεσματικά.

Όπως είναι λογικό μία πτυχιακή εργασία, η οποία έχει ως κύριο σκοπό να χρησιμοποιηθεί από ολόκληρο το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων, είναι δύσκολο να ολοκληρωθεί σε μία εργασία, καθώς υπάρχει η ανάγκη για συνεχόμενη βελτίωση και συντήρηση. Παρόλα αυτά οι απαιτήσεις βελτίωσης της εφαρμογής στα πλαίσια της πτυχιακής ήταν συγκεκριμένες. Παρακάτω θα γίνει μία σχετική ανάλυση αναφορικά με τις προτάσεις που πιστεύουμε ότι θα βελτίωναν ακόμη περισσότερο την εφαρμογή.

Χρήση Vuex

Σε μία εφαρμογή τέτοιου μεγέθους η διαχείριση των states κρίνεται απαραίτητη. Χωρίς την χρήση του Vuex η μεταφορά δεδομένων μεταξύ component μπορεί να γίνει υπερβολικά περίπλοκη, και σε μερικές περιπτώσεις αδύνατη. Χρησιμοποιώντας το Vuex δίνεται η δυνατότητα στον προγραμματιστή να μπορεί να αποθηκεύει τις καταστάσεις δεδομένων κάπου κεντρικά και να μπορεί να τις διαχειριστεί μέσα από όλα τα components. Με αυτόν τον τρόπο μπορεί να παρακολουθεί για αλλαγές που γίνονται στην τιμή μιας μεταβλητής και να ενημερώνει όλα τα components που την χρησιμοποιούν με την νέα τιμή χωρίς να χρειαστεί να το στείλει σαν prop μέσα σε όλα τα components που την χρησιμοποιούν. Επίσης υπάρχει η δυνατότητα σε κάθε αλλαγή της τιμής να γίνονται ορισμένες λειτουργίες που μπορεί να θέσει ο εκάστοτε προγραμματιστής.

Διαχωρισμός Front-End από το Back-End

Μία ενδιαφέρουσα πρόταση και αρκετά χρήσιμη για την συντηρησιμότητα της εφαρμογής είναι ο πλήρης διαχωρισμός του Front-End από το Back-End. Αυτό έχει σαν αποτέλεσμα να είναι πιο εύκολο να επεκταθεί η εφαρμογή σε διαφορετικές πλατφόρμες (π.χ. iOS, Android, Desktop App) χωρίς να χρειάζονται περαιτέρω ρυθμίσεις. Η υπάρχουσα εφαρμογή περιέχει τα αρχεία του front-end μέσα στην δομή της Laravel, αυτό παρόλο που έχει σαν αποτέλεσμα να γίνεται πιο εύκολη η υλοποίηση μίας ολοκληρωμένης εφαρμογής, αρχίζει να δυσκολεύει την κατάσταση όταν πρόκειται να χρησιμοποιηθεί κατά βάση το API της εφαρμογής π.χ. για μία Android εφαρμογή. Επίσης συστήνεται σαν καλύτερη πρακτική να χωρίζεται το Front-End από το Back-End όταν οι προγραμματιστές δεν προορίζονται για Full Stack καθώς μπορούν να δουλεύουν ταυτόχρονα χωρίς να επηρεάζεται η εφαρμογή από τις αλλαγές που γίνονται στον κάθε τομέα της εφαρμογής και χωρίς να χρειάζεται να γίνεται συνέχεια συγχρονισμός των αρχείων.

Καλύτερη διαχείριση Protected Routes

Λόγο της υλοποίησης του OAuth 2.0 μέσω της βιβλιοθήκης Socialite, ο αυθεντικοποιημένος χρήστης παραμένει συνδεδεμένος μέσω χρήσης συνεδριών (sessions). Αυτό παρόλο που είναι αρκετά αποδοτικό

για χρήση στο διαδίκτυο, έχει σαν αποτέλεσμα την δυσκολία χρήσης του API σε διαφορετική πλατφόρμα, καθώς η συνεδρία αφορά κατά κύριο λόγο διαδικτυακές εφαρμογές. Αυτό που προτείνεται να γίνει είναι η χρήση των token καθώς αποθηκεύεται στον εκάστοτε client αντί για το session που αποθηκεύεται στον server και αυτό μπορεί να γίνει πρόβλημα όταν υπάρχει μεγάλος αριθμός χρηστών την ίδια στιγμή.

Μετατροπή από MPA σε SPA

Ως προκαθορισμένο τρόπο οι εφαρμογές της Laravel χρησιμοποιούν την αρχιτεκτονική Multi Page Application (MPA). Αυτό έχει σαν αποτέλεσμα η ιστοσελίδα να πρέπει να ξανά κάνει render όλα τα δεδομένα σε περίπτωση που αλλάξουμε διαδρομή. Χρησιμοποιώντας την αρχιτεκτονική Single Page Application (SPA) η ιστοσελίδα φορτώνει όλα τα δεδομένα που χρειάζεται από το πρώτο render. Αυτό έχει σαν αποτέλεσμα να μην απαιτείται re-render για κάθε αλλαγή διαδρομής. Η ιστοσελίδα με αυτόν τον τρόπο χειρίζεται μόνο τα requests από και προς την βάση.

Υλοποίηση Integration και Unit Testing

Μία πολύ καλή πρακτική για την υλοποίηση μίας εφαρμογής είναι η δημιουργία των Tests. Με τα Tests μπορείς να βεβαιωθείς ότι οι λειτουργίες της εφαρμογής εκτελούνται φέρνοντας τα επιθυμητά αποτελέσματα. Με αυτόν τον τρόπο όταν γίνεται μία αλλαγή στην εφαρμογή μπορείς εύκολα να ελέγξεις αν η αλλαγή αυτή προκάλεσε κάποιο λειτουργικό πρόβλημα. Τα Tests μπορούν να χωριστούν στα Integration Tests, που αφορούν την δοκιμή πολλαπλών λειτουργιών (π.χ. ένα ολόκληρο component), και στα Unit Tests, που αφορούν την δοκιμή μίας συγκεκριμένης λειτουργίας (π.χ. μίας μεθόδου). Η Laravel υποστηρίζει την συγγραφή και των δύο ειδών tests μέσω του phpunit.

Βελτιστοποίησης της εφαρμογής

Η βελτιστοποίηση μπορεί να βοηθήσει στην καλύτερη απόδοση της εφαρμογής. Υπάρχουν πολλοί τρόποι για να επιτευχθεί αυτό, μερικά παραδείγματα είναι τα εξής:

- Βελτιστοποίηση των ερωτημάτων: η Laravel παρόλο που παρέχει λειτουργίες που βοηθούν τον χρήστη για την δημιουργία περίπλοκων ερωτημάτων, χρησιμοποιεί μεθόδους που μπορούν να καθυστερήσουν την εφαρμογή.
- Βελτιστοποίηση της βάσης: μέσω των ευρετηρίων η SQL μπορεί να ψάξει γρήγορα και εύκολα ορισμένες εγγραφές που αναζητούνται συχνότερα.

Η βελτιστοποίηση γενικότερα μπορεί να ωφελήσει σε μεγάλο βαθμό εφαρμογές που έχουν πολλά δεδομένα ή που χρησιμοποιούν περίπλοκα ερωτήματα.

6.3 Συμπεράσματα

Στη συγκεκριμένη εργασία δημιουργήθηκε μία διαδικτυακή εφαρμογή η οποία έχει ως σκοπό να αντικαταστήσει το ήδη υπάρχον σύστημα ανακοινώσεων που υπάρχει στο apps. Η παρούσα εργασία είχε ξεκινήσει να υλοποιείται σε προηγούμενη πτυχιακή, και στη συνέχεια αναλάβαμε να υλοποιήσουμε και να βελτιώσουμε ορισμένες λειτουργίες που κρίθηκαν απαραίτητες. Η εφαρμογή όπως αναφέραμε και παραπάνω μπορεί να αναβαθμιστεί σε πολλούς τομείς. Παρόλα αυτά στην εφαρμογή έγιναν αρκετές αλλαγές, τόσο στο κομμάτι του Front-End όσο και στο κομμάτι του Back-End. Η αλλαγές αυτές βελτίωσαν ορισμένα μέρη της εφαρμογής και την καθιστούν πιο εύχρηστη από πριν.

Κατά τη διάρκεια της υλοποίησης εξοικειωθήκαμε με τεχνολογίες που θα μας βοηθήσουν στην μελλοντική μας πορεία, βελτιωθήκαμε στην συγγραφή κώδικα, και το κυριότερο, μάθαμε να συνεργαζόμαστε.

Πέραν όμως της εξοικείωσης με τις τεχνολογίες, οι οποίες εξελίσσονται διαρκώς, θεωρούμε σημαντικό κέρδος την τριβή με δομικά στοιχεία για μία διαδικτυακή εφαρμογή όπως είναι για παράδειγμα:

- Την υλοποίηση του OAuth 2.0 πρωτόκολλου
- Την επίλυση προβλημάτων
- Την ανάπτυξη και βελτίωση ενός REST API
- Τη σωστή δόμηση ενός επεκτάσιμου προγράμματος ακολουθώντας γνωστά σχεδιαστικά πρότυπα
- Και τέλος την επεξήγηση του κώδικα και την δημιουργία Documentation στο κομμάτι του API.

6.4 Επίλογος

Στο τελευταίο κεφάλαιο έγινε μία προσπάθεια να αναλυθούν προτάσεις οι οποίες θα μπορούσαν να βελτιώσουν ακόμα περισσότερο την υπάρχουσα εφαρμογή. Επίσης αναλύθηκαν τα συμπεράσματα των συγγραφέων ως προς την υλοποίηση της εργασίας αυτής.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] «Javascript Introduction,» MDN, [Ηλεκτρονικό]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction>. [Πρόσβαση 30 Νοεμβρίου 2021].
- [2] «HTML Introduction,» GeeksforGeeks, [Ηλεκτρονικό]. Available: <https://www.geeksforgeeks.org/html-introduction/>. [Πρόσβαση 25 Νοεμβρίου 2021].
- [3] «What is CSS?,» TutorialsPoint, [Ηλεκτρονικό]. Available: https://www.tutorialspoint.com/css/what_is_css.htm. [Πρόσβαση 26 Νοεμβρίου 2021].
- [4] «The Vue Instance,» Vue.js, [Ηλεκτρονικό]. Available: <https://v2.vuejs.org/v2/guide/instance.html>. [Πρόσβαση 6 Δεκεμβρίου 2021].
- [5] «About npm,» npm, [Ηλεκτρονικό]. Available: <https://docs.npmjs.com/about-npm>. [Πρόσβαση 3 Δεκεμβρίου 2021].
- [6] «VueJS - Components,» TutorialsPoint, [Ηλεκτρονικό]. Available: https://www.tutorialspoint.com/vuejs/vuejs_components.htm. [Πρόσβαση 6 Δεκεμβρίου 2021].
- [7] A. Esther, «What are Vue Components?,» Educative, [Ηλεκτρονικό]. Available: <https://www.educative.io/edpresso/what-are-vue-components>. [Πρόσβαση 7 Δεκεμβρίου 2021].
- [8] «Props Versus Data in Vue: The Subtle Differences You Need to Know,» MichaelThiessen, [Ηλεκτρονικό]. Available: <https://michaelnthiessen.com/vue-props-vs-data/>. [Πρόσβαση 14 Δεκεμβρίου 2021].
- [9] «Client-Side Storage,» Vue.js, [Ηλεκτρονικό]. Available: <https://v2.vuejs.org/v2/cookbook/client-side-storage.html?redirect=true>. [Πρόσβαση 20 Δεκεμβρίου 2021].
- [10] Luke, Thomson και Welling, PHP and MySQL Web Development, Ergodebooks, 2013.
- [11] J. Clark, «What is a DevTool?,» Back4App, [Ηλεκτρονικό]. Available: <https://blog.back4app.com/what-is-a-devtool/>. [Πρόσβαση 4 Ιανουαρίου 2022].
- [12] «What is JSON and what is it used for?,» [Ηλεκτρονικό]. Available: <https://stackoverflow.com/questions/383692/what-is-json-and-what-is-it-used-for>. [Πρόσβαση 10 Ιανουαρίου 2022].
- [13] «What is Composer?,» RIP Tutorial, [Ηλεκτρονικό]. Available: <https://riptutorial.com/php/example/3396/what-is-composer->. [Πρόσβαση 21 Δεκεμβρίου 2021].
- [14] «What Are Web Services? Easy-to-Learn Concepts with Examples,» Cleo, [Ηλεκτρονικό]. Available: <https://www.cleo.com/blog/knowledge-base-web-services>. [Πρόσβαση 4 Ιανουαρίου 2022].

- [15] «What are Web Services?,» TutorialsPoint, [Ηλεκτρονικό]. Available: https://www.tutorialspoint.com/webservices/what_are_web_services.htm. [Πρόσβαση 3 Ιανουαρίου 2022].
- [16] «Laravel - Overview,» TutorialsPoint, [Ηλεκτρονικό]. Available: https://www.tutorialspoint.com/laravel/laravel_overview.htm. [Πρόσβαση 20 Δεκεμβρίου 2021].
- [17] P. Mbanugo, «Getting Familiar with Vue Devtools,» ProgressTelerik, [Ηλεκτρονικό]. Available: <https://www.telerik.com/blogs/getting-familiar-with-vue-devtools>. [Πρόσβαση 3 Ιανουαρίου 2022].
- [18] «Bulma CSS: What It Is and Why It's a Framework That Developers Love,» DevMountain, [Ηλεκτρονικό]. Available: <https://devmountain.com/blog/why-bulma-css-could-be-your-new-favorite-framework/>. [Πρόσβαση 2 Δεκεμβρίου 2021].
- [19] «How Laravel implements MVC and how to use it effectively,» [Ηλεκτρονικό]. Available: <https://blog.pusher.com/laravel-mvc-use/>. [Πρόσβαση 20 Ιανουαρίου 2022].
- [20] «What is MariaDB,» [Ηλεκτρονικό]. Available: <https://www.mariadbtutorial.com/getting-started/what-is-mariadb/>. [Πρόσβαση 1 Φεβρουαρίου 2022].
- [21] «What is Postman API Test,» Encora, [Ηλεκτρονικό]. Available: <https://www.encora.com/insights/what-is-postman-api-test>. [Πρόσβαση 6 Φεβρουαρίου 2022].
- [22] «What is OAuth 2.0?,» [Ηλεκτρονικό]. Available: <https://auth0.com/intro-to-iam/what-is-oauth-2/>. [Πρόσβαση 4 Φεβρουαρίου 2022].
- [23] «What is MySQL? Everything You Need to Know,» [Ηλεκτρονικό]. Available: <https://www.talend.com/resources/what-is-mysql/>. [Πρόσβαση 22 Ιανουαρίου 2022].
- [24] «Laravel Debugbar,» [Ηλεκτρονικό]. Available: <https://github.com/barryvdh/laravel-debugbar>. [Πρόσβαση 20 Φεβρουαρίου 2022].
- [25] P. Patel, «Laravel Request Lifecycle,» Dev.to, [Ηλεκτρονικό]. Available: <https://dev.to/patelparixit07/laravel-request-lifecycle-195e>. [Πρόσβαση 11 Μαΐου 2022].
- [26] A. Tejani, «Laravel Request Lifecycle,» Medium, [Ηλεκτρονικό]. Available: <https://medium.com/@ankitatejani84/laravel-request-lifecycle-7c2145aa1257>. [Πρόσβαση 10 Μαΐου 2022].