



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

Υλοποίηση και αξιολόγηση δικτυακού τόπου
εργαστηρίου Τηλεματικής και Κατανεμημένων
Συστημάτων σε Drupal και WordPress

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΜΠΟΥΛΑΦΕΝΤΗ ΧΡΗΣΤΟΥ

Επιβλέπων:

Μπούρας Χρήστος

Καθηγητής

Μέλη Τριμελούς

Εξεταστικής Επιτροπής :

Γαροφαλάκης Ιωάννης

Καθηγητής

Παπαϊωάννου Εύη

Επίκουρη Καθηγήτρια

Πάτρα, Ιούνιος 2026



Πανεπιστήμιο Πατρών
Πολυτεχνική Σχολή
Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής

Copyright ©–All rights reserved Μπουλαφέντης Χρήστος, 2026.

Με την επιφύλαξη παντός δικαιώματος.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Το περιεχόμενο αυτής της εργασίας δεν απηχεί απαραίτητα τις απόψεις του Τμήματος, του Επιβλέποντα, ή της επιτροπής που την ενέκρινε.

Πρόλογος

Η παρούσα πτυχιακή εργασία εκπονήθηκε στο πλαίσιο των σπουδών στο Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής του Πανεπιστημίου Πατρών. Αντικείμενό της αποτελεί η μεταφορά — ή, ορθότερα, η πλήρης επανακατασκευή — του ιστοτόπου του εργαστηρίου Τηλεματικής και Κατανεμημένων Συστημάτων από το Drupal 7 (D7) στο Drupal 10 (D10), με κίνητρο κυρίως το γεγονός ότι το D7 έφτασε στο τέλος του κύκλου ζωής του (End-of-Life — EOL) τον Ιανουάριο του 2025 και δεν λαμβάνει πλέον ενημερώσεις ασφαλείας. Παράλληλα, η εργασία περιλαμβάνει συστηματική σύγκριση της νέας υλοποίησης σε Drupal 10 με την υπάρχουσα υλοποίηση του ίδιου εργαστηρίου σε WordPress, τόσο ως προς την απόδοση, την προσβασιμότητα και την ασφάλεια όσο και ως προς τη διαδικασία ανάπτυξης και την εμπειρία τελικού χρήστη.

Η εργασία αυτή αποτέλεσε ευκαιρία για να αποκτηθεί σε βάθος γνώση της λειτουργίας ενός σύγχρονου Συστήματος Διαχείρισης Περιεχομένου (Content Management System — CMS), και ειδικότερα του Drupal 10. Πέρα από την τεχνική υλοποίηση, η διαδικασία ανέδειξε και σημαντικά ζητήματα που αφορούν τη μακροπρόθεσμη συντηρησιμότητα ιστοτόπων ακαδημαϊκών εργαστηρίων: την ανάγκη για τακτικές ενημερώσεις λογισμικού, τη σημασία του σωστού αρχικού σχεδιασμού της δομής, και την επιλογή κατάλληλων εργαλείων για κάθε περίπτωση.

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή κ. Μπούρα Χρήστο για την καθοδήγηση και την υποστήριξή του καθ' όλη τη διάρκεια της εκπόνησης της εργασίας. Ξεχωριστές ευχαριστίες οφείλω στον Δρ. Βασίλειο Κόγκινο για την πολύτιμη βοήθεια και τις εποικοδομητικές παρατηρήσεις του σε όλη την πορεία της εργασίας. Ευχαριστώ επίσης τα υπόλοιπα μέλη της τριμελούς εξεταστικής επιτροπής, καθηγητή κ. Γαροφαλάκη Ιωάννη και επίκουρη καθηγήτρια κα. Παπαϊωάννου Εύη, για τον χρόνο που αφιέρωσαν στην αξιολόγηση της παρούσας εργασίας. Τέλος, ευχαριστώ το εργαστήριο Τηλεματικής για την παροχή πρόσβασης στον server και των απαραίτητων δικαιωμάτων διαχείρισης.

Πάτρα, Ιούνιος 2026

Μπουλαφέντης Χρήστος

Περίληψη

Η παρούσα εργασία πραγματεύεται την υλοποίηση και αξιολόγηση του ιστοτόπου του εργαστηρίου Τηλεματικής και Κατανεμημένων Συστημάτων του Πανεπιστημίου Πατρών στο Drupal 10 (D10). Αφετηρία της εργασίας αποτέλεσε η ανάγκη μετάβασης από το Drupal 7 (D7), το οποίο έφτασε στο τέλος του κύκλου ζωής του (End-of-Life — EOL) τον Ιανουάριο του 2025 και δεν λαμβάνει πλέον ενημερώσεις ασφαλείας, καθιστώντας τη συνέχιση της λειτουργίας της παλιάς σελίδας ανεπίτρεπτη από πλευράς ασφάλειας.

Αρχικά παρουσιάζεται το Drupal στο πλαίσιο των Συστημάτων Διαχείρισης Περιεχομένου (Content Management Systems — CMS), με σύγκριση έναντι των δημοφιλέστερων εναλλακτικών (WordPress, Joomla, TYPO3), και αναλύονται οι θεμελιώδεις αρχιτεκτονικές διαφορές μεταξύ D7 και D10. Στη συνέχεια περιγράφεται η διαδικασία κατασκευής ιστοτόπου με D10, καλύπτοντας βασικά θέματα όπως content types, fields, theming με Twig templates, και το module Views. Αναλύονται επίσης οι λόγοι για τους οποίους η μετάβαση από D7 σε D10 δεν αποτελεί απλή αναβάθμιση αλλά ουσιαστικά πλήρη επανακατασκευή, και τεκμηριώνεται η σχετική απόφαση που λήφθηκε.

Στο κύριο μέρος της εργασίας παρουσιάζεται αναλυτικά η υλοποίηση του νέου ιστοτόπου: ο σχεδιασμός οκτώ custom content types (Members, News, Project, Publication, Research Areas, Partner, Link, Basic Page), η χρήση του module Views για την εμφάνιση του περιεχομένου, οι αισθητικές επιλογές με βάση το Business theme και την προσαρμογή μέσω CSS Asset Injection, η υλοποίηση responsive design με Bootstrap και το Responsive Image module του Drupal 10, καθώς και τα contributed modules που χρησιμοποιήθηκαν. Ακολουθεί οπτική σύγκριση της παλιάς και της νέας σελίδας για τις κύριες ενότητες, συμπεριλαμβανομένης σύγκρισης mobile εμφάνισης. Τέλος, πραγματοποιείται συστηματική αξιολόγηση και σύγκριση με την τρέχουσα WordPress υλοποίηση του εργαστηρίου, τόσο ποσοτικά (με χρήση των εργαλείων Google PageSpeed Insights, GTmetrix και SecurityHeaders.com) όσο και ποιοτικά (σύγκριση διαδικασίας ανάπτυξης και UI), και παρουσιάζονται τα συμπεράσματα και οι προτάσεις για μελλοντικές βελτιώσεις.

Λέξεις Κλειδιά

Drupal 10, Content Management System, CMS, ιστότοπος εργαστηρίου, αναβάθμιση, Views, Twig, theming, responsive design, ασφάλεια ιστότοπων, αξιολόγηση απόδοσης, WordPress.

Abstract

This thesis presents the implementation and evaluation of the website of the Telematics and Distributed Systems Laboratory of the University of Patras using Drupal 10 (D10). The primary motivation was the need to migrate from Drupal 7 (D7), which reached its End-of-Life (EOL) in January 2025 and no longer receives security updates, making the continued operation of the old website a significant security risk.

The thesis begins with an overview of Drupal in the context of Content Management Systems (CMS), comparing it with the most popular alternatives (WordPress, Joomla, TYPO3), and analyzing the fundamental architectural differences between D7 and D10. The process of building a website with D10 is then described, covering key topics such as content types, fields, theming with Twig templates, the Views module, and configuration management. The reasons why migrating from D7 to D10 is not a straightforward upgrade but rather a full rebuild are thoroughly analyzed, and the decision to pursue a complete reconstruction is justified.

The main part of the thesis presents the implementation of the new website in detail: the design of eight custom content types (Members, News, Project, Publication, Research Areas, Partner, Link, Basic Page), the use of the Views module for content display, the aesthetic choices based on the Business theme with CSS Asset Injection customization, responsive design using Bootstrap and the Drupal 10 Responsive Image module, and the contributed modules employed. A visual comparison of the old and new website is provided for all major sections, including mobile screenshots comparing the Drupal 10 and WordPress implementations. Finally, a systematic evaluation and comparison of the Drupal 10 and WordPress sites is performed using Google PageSpeed Insights, GTmetrix, and SecurityHeaders.com, alongside a qualitative comparison of the development experience and UI differences between the two platforms.

Keywords

Drupal 10, Content Management System, CMS, laboratory website, migration, Views, Twig, theming, responsive design, web security, performance evaluation, WordPress.

Περιεχόμενα

Πρόλογος	i
Περίληψη	iii
Abstract	v
Κατάλογος Εικόνων	xii
Κατάλογος Πινάκων	xiii
1 Εισαγωγή	1
1.1 Η σημασία των ιστότοπων για ακαδημαϊκά εργαστήρια	1
1.2 Το πρόβλημα του End-of-Life λογισμικού	2
1.3 Σκοπός και στόχοι της εργασίας	2
1.4 Μεθοδολογία	3
1.5 Δομή του κειμένου της εργασίας	3
2 Θεωρητικό Υπόβαθρο	5
2.1 Συστήματα Διαχείρισης Περιεχομένου (CMS)	5
2.2 Ιστορία και φιλοσοφία του Drupal	5
2.3 Πώς λειτουργεί το Drupal	6
2.3.1 Βασικές έννοιες	6
2.3.2 Αρχιτεκτονική	6
2.4 Σύγκριση Drupal με άλλα CMS	6
2.5 Drupal 7 έναντι Drupal 10	8
2.5.1 Αρχιτεκτονικές και τεχνικές διαφορές	8
2.5.2 Ασφάλεια και το τέλος ζωής του Drupal 7	9
3 Κατασκευή Ιστότοπου με Drupal 10	11
3.1 Απαιτήσεις συστήματος και εγκατάσταση	11
3.2 Το Admin interface	11
3.3 Content Types και Fields	12
3.4 Το σύστημα Theming	13

3.4.1	Twig Templates	13
3.4.2	Base Themes	13
3.4.3	Asset Injection	13
3.5	Modules — Core και Contributed	14
3.5.1	To Module Views	14
3.6	Configuration Management	15
4	Δυσκολίες αναβάθμισης από Drupal 7 σε Drupal 10	17
4.1	Γιατί δεν είναι απλή η αναβάθμιση	17
4.2	Η διαφορά αρχιτεκτονικής	17
4.3	Τα modules	18
4.4	Τα themes	18
4.5	Το Migrate API και οι περιορισμοί του	18
4.6	Άλλα πρακτικά ζητήματα	19
4.7	Η απόφαση για πλήρη ανακατασκευή	19
5	Υλοποίηση του νέου ιστότοπου	21
5.1	Επισκόπηση του ιστότοπου	21
5.2	Τεχνολογίες που χρησιμοποιήθηκαν	21
5.3	Δομή περιεχομένου	21
5.4	Αισθητικές επιλογές και Theme	23
5.5	Modules που χρησιμοποιήθηκαν	23
5.6	Προσαρμοστικός σχεδιασμός (Responsive Design)	23
5.6.1	Bootstrap και Responsive Grid	25
5.6.2	Στυλιστικές Προσαρμογές μέσω CSS Asset Injection	25
5.6.3	Responsive Image Module	26
5.7	Σύγκριση Drupal 7 και Drupal 10	26
5.7.1	Αρχική Σελίδα	27
5.7.2	Σελίδα Μέλους	28
5.7.3	Σελίδα Δημοσιεύσεων	29
5.7.4	Σελίδα Ερευνητικών Περιοχών	30
5.7.5	Σελίδα Links	31
5.7.6	Σελίδα Νέων	32
5.7.7	Σελίδα Projects	33
5.7.8	Σελίδα παραδείγματος δημοσίευσης	33
5.7.9	Σελίδα παραδείγματος προγράμματος	35
6	Αξιολόγηση και σύγκριση των δύο ιστότοπων	37
6.1	Μεθοδολογία αξιολόγησης	37
6.2	Εργαλεία Μέτρησης και Μετρικές	37
6.2.1	Google PageSpeed Insights	37
6.2.2	GTmetrix	38

6.2.3	SecurityHeaders.com	39
6.3	Αποτελέσματα Google PageSpeed Insights	39
6.4	Αποτελέσματα GTmetrix	42
6.5	Αποτελέσματα SecurityHeaders.com	44
6.6	Συνολική αξιολόγηση	44
6.7	Σύγκριση Διαδικασίας Ανάπτυξης: Drupal 10 έναντι WordPress	46
6.7.1	Δημιουργία Περιεχομένου	46
6.7.2	Διαχείριση Εμφάνισης	46
6.7.3	Εμφάνιση Λιστών Περιεχομένου	46
6.7.4	Responsive Images: Drupal 10 έναντι WordPress	47
6.7.5	Σύγκριση Mobile Εμφάνισης	49
6.7.6	Συνολική Αξιολόγηση Διαδικασίας Ανάπτυξης	52
6.8	Ποιοτική Σύγκριση των Δύο Ιστότοπων	53
6.8.1	Αρχική Σελίδα	53
6.8.2	Σελίδα Μελών	54
6.8.3	Σελίδα Δημοσιεύσεων	55
6.8.4	Σελίδα Νέων	55
6.8.5	Σελίδα Projects	57
6.8.6	Μεμονωμένη Σελίδα Δημοσίευσης	58
6.9	Ερμηνεία των Αποτελεσμάτων Απόδοσης	58
7	Συμπεράσματα	61
7.1	Κύρια ευρήματα	61
7.2	Μελλοντική εργασία	61
7.3	Τελικές παρατηρήσεις	64

Κατάλογος Εικόνων

5.1	Τα fields του content type Members στο Admin interface	22
5.2	Λίστα των ενεργοποιημένων Views στο Admin interface	22
5.3	Λίστα περιεχομένου στο Admin interface του Drupal 10	23
5.4	Λίστα CSS Injectors στο Admin interface	24
5.5	Παράδειγμα CSS κώδικα μέσα σε έναν injector	24
5.6	Η σελίδα διαχείρισης modules (Extend) στο Admin interface	25
5.7	Αρχική σελίδα — Drupal 7	27
5.8	Αρχική σελίδα — Drupal 10	27
5.9	Σελίδα μέλους — Drupal 7	28
5.10	Σελίδα μέλους — Drupal 10	28
5.11	Σελίδα δημοσιεύσεων — Drupal 7	29
5.12	Σελίδα δημοσιεύσεων — Drupal 10	29
5.13	Σελίδα ερευνητικών περιοχών — Drupal 7	30
5.14	Σελίδα ερευνητικών περιοχών — Drupal 10	30
5.15	Σελίδα links — Drupal 7	31
5.16	Σελίδα links — Drupal 10	31
5.17	Σελίδα νέων — Drupal 7	32
5.18	Σελίδα νέων — Drupal 10	32
5.19	Σελίδα projects — Drupal 7	33
5.20	Σελίδα projects — Drupal 10	33
5.21	Σελίδα παραδείγματος δημοσίευσης — Drupal 7	34
5.22	Σελίδα παραδείγματος δημοσίευσης — Drupal 10	34
5.23	Σελίδα παραδείγματος προγράμματος — Drupal 7	35
5.24	Σελίδα παραδείγματος προγράμματος — Drupal 10	35
6.1	Αρχική σελίδα — WordPress (mobile)	49
6.2	Αρχική σελίδα — Drupal 10 (mobile)	49
6.3	Σελίδα μελών — WordPress (mobile)	50
6.4	Σελίδα μελών — Drupal 10 (mobile)	50
6.5	Σελίδα δημοσιεύσεων — WordPress (mobile)	51
6.6	Σελίδα δημοσιεύσεων — Drupal 10 (mobile)	51
6.7	Σελίδα project — WordPress (mobile)	52

6.8	Σελίδα project — Drupal 10 (mobile)	52
6.9	Αρχική σελίδα — WordPress	53
6.10	Αρχική σελίδα — Drupal 10	54
6.11	Σελίδα μελών — WordPress	54
6.12	Σελίδα μελών — Drupal 10	55
6.13	Σελίδα δημοσιεύσεων — WordPress	56
6.14	Σελίδα δημοσιεύσεων — Drupal 10	56
6.15	Σελίδα νέων — WordPress	57
6.16	Σελίδα νέων — Drupal 10	57

Κατάλογος Πινάκων

2.1	Σύγκριση δημοφιλών CMS	8
6.1	Βαθμολογίες Google PageSpeed Insights ανά σελίδα — Desktop	40
6.2	Βαθμολογίες Google PageSpeed Insights ανά σελίδα — Mobile	40
6.3	Μετρικές ταχύτητας Google PageSpeed Insights (αρχική σελίδα) — Desktop	40
6.4	Μετρικές ταχύτητας Google PageSpeed Insights (αρχική σελίδα) — Mobile	41
6.5	Αποτελέσματα GTmetrix ανά σελίδα	42
6.6	Σύγκριση HTTP Security Headers	44

Κεφάλαιο 1

Εισαγωγή

Η παρούσα διπλωματική εργασία ασχολείται με τη μεταφορά του ιστότοπου του εργαστηρίου Τηλεματικής του Πανεπιστημίου Πατρών από την έκδοση 7 του Drupal στην έκδοση 10. Η σελίδα βρίσκεται στη διεύθυνση <https://telematics.upatras.gr/> και λειτουργούσε εδώ και αρκετά χρόνια ως το βασικό σημείο αναφοράς για το εργαστήριο.

Το κύριο κίνητρο για την αναβάθμιση ήταν ότι το Drupal 7 έφτασε στο τέλος του κύκλου ζωής του (end-of-life, EOL) και δεν λαμβάνει πλέον ενημερώσεις ασφαλείας — κάτι που σημαίνει ότι κάθε μέρα που περνούσε, η σελίδα γινόταν πιο εκτεθειμένη σε πιθανές επιθέσεις. Πέρα από το ζήτημα της ασφάλειας, το Drupal 10 έχει σαφώς καλύτερη απόδοση, ένα πιο σύγχρονο σύστημα για θέματα (theming) και γενικά μια πολύ πιο ευχάριστη εμπειρία ανάπτυξης σε σχέση με το Drupal 7 (D7) .

Η προσέγγιση που επιλέχθηκε ήταν η πλήρης επανακατασκευή της σελίδας από μηδενική βάση, χωρίς να αξιοποιηθεί κανένα κομμάτι από τον παλιό κώδικα. Αυτή η απόφαση πάρθηκε κυρίως επειδή το Drupal 7 και το Drupal 10 είναι ουσιαστικά δύο διαφορετικά συστήματα, και η απευθείας μετεγκατάσταση κώδικα ή ρυθμίσεων δεν ήταν εφικτή. Έγινε πάντως προσπάθεια να διατηρηθεί η αισθητική της αρχικής σελίδας όσο το δυνατόν πιο πιστά.

1.1 Η σημασία των ιστότοπων για ακαδημαϊκά εργαστήρια

Στο σύγχρονο ακαδημαϊκό περιβάλλον, ο ιστότοπος ενός ερευνητικού εργαστηρίου δεν αποτελεί απλώς μια «ψηφιακή επαγγελματική κάρτα». Είναι το κύριο σημείο επαφής του εργαστηρίου με την ευρύτερη επιστημονική κοινότητα, με υποψήφιους συνεργάτες, χρηματοδότες και φοιτητές. Μέσα από αυτήν παρουσιάζονται οι ερευνητικές δραστηριότητες, τα μέλη της ομάδας, οι δημοσιεύσεις και τα ενεργά ερευνητικά προγράμματα. Η ποιότητα και η λειτουργικότητα του ιστότοπου επηρεάζει άμεσα την εικόνα που σχηματίζουν τρίτοι για το επίπεδο και τη δυναμικότητα του εργαστηρίου.

Για τον λόγο αυτό, η συντήρηση και η έγκαιρη αναβάθμιση της τεχνολογικής υποδομής ενός ακαδημαϊκού ιστότοπου δεν είναι μια απλή τεχνική υποχρέωση, αλλά μια στρατηγική απόφαση που επηρεάζει τη μακροπρόθεσμη βιωσιμότητα και αξιοπιστία της παρουσίας του

εργαστηρίου στο διαδίκτυο. Ένας ιστότοπος που δεν συντηρείται εκθέτει το εργαστήριο σε κινδύνους ασφαλείας, παρουσιάζει απαρχαιωμένη εμφάνιση και δυσκολεύει τη διαχείριση του περιεχομένου από τα μέλη του.

Στην περίπτωση του εργαστηρίου Τηλεματικής και Κατανεμημένων Συστημάτων, η ανάγκη αναβάθμισης ήταν ιδιαίτερα επιτακτική: η υπάρχουσα υλοποίηση σε Drupal 7 είχε φτάσει στο τέλος του κύκλου ζωής της, γεγονός που σήμαινε ότι κάθε νέα ευπάθεια που ανακαλύπτεται παραμένει χωρίς διόρθωση για απροσδιόριστο χρονικό διάστημα.

1.2 Το πρόβλημα του End-of-Life λογισμικού

Το Drupal 7 κυκλοφόρησε το 2011 και για πάνω από μία δεκαετία αποτελούσε σταθερή επιλογή για χιλιάδες ιστότοπους παγκοσμίως. Ωστόσο, κάθε λογισμικό έχει κύκλο ζωής: κάποια στιγμή η κοινότητα ανάπτυξης σταματά να εκδίδει ενημερώσεις και στρέφεται αποκλειστικά σε νεότερες εκδόσεις. Για το Drupal 7, η ημερομηνία λήξης του κύκλου ζωής ορίστηκε τελικά στις 5 Ιανουαρίου 2025, έπειτα από διαδοχικές παρατάσεις λόγω της μεγάλης βάσης χρηστών [1].

Το End-of-Life (EOL) ενός CMS σημαίνει στην πράξη ότι όποια ευπάθεια ασφαλείας ανακαλυφθεί στον κώδικα δεν πρόκειται να διορθωθεί ποτέ με επίσημο patch. Οι επιτιθέμενοι το γνωρίζουν αυτό και εκμεταλλεύονται ακριβώς τις «γνωστές-και-αδιόρθωτες» ευπάθειες, συχνά με αυτοματοποιημένα εργαλεία που σαρώνουν χιλιάδες ιστότοπους ταυτόχρονα αναζητώντας ευάλωτες εκδόσεις [2, 3]. Ακόμα και αν ένας ιστότοπος δεν αποτελεί υψηλής αξίας στόχο, μπορεί να χρησιμοποιηθεί ως πλατφόρμα για επιθέσεις σε τρίτους ή για διανομή κακόβουλου λογισμικού, με σοβαρές νομικές και ηθικές επιπτώσεις για τον οργανισμό που τον φιλοξενεί [2].

Η κατάσταση αυτή καθιστούσε τη συνέχιση λειτουργίας της παλιάς σελίδας του εργαστηρίου όχι απλώς ανεπιθύμητη, αλλά ουσιαστικά ανεπίτρεπτη από πλευράς ορθής διαχείρισης τεχνολογικών συστημάτων.

1.3 Σκοπός και στόχοι της εργασίας

Η παρούσα εργασία έχει δύο κύριους στόχους. Ο πρώτος είναι η **υλοποίηση** του νέου ιστότοπου του εργαστηρίου σε Drupal 10, με πλήρη επανασχεδιασμό της δομής περιεχομένου και διατήρηση της αισθητικής της αρχικής σελίδας όπου αυτό ήταν εφικτό. Ο δεύτερος στόχος είναι η **αξιολόγηση** της νέας υλοποίησης και η συστηματική σύγκρισή της με την υπάρχουσα WordPress υλοποίηση του ιστότοπου του ίδιου εργαστηρίου, τόσο ποσοτικά (μετρήσεις απόδοσης, ασφάλειας, προσβασιμότητας) όσο και ποιοτικά (εμπειρία ανάπτυξης, οργάνωση περιεχομένου, εμπειρία χρήστη).

Ειδικότερα, η εργασία επιχειρεί να απαντήσει στα εξής ερωτήματα: Ποιες είναι οι βασικές αρχιτεκτονικές διαφορές μεταξύ D7 και D10 και γιατί η μετάβαση δεν συνιστά απλή αναβάθμιση; Πώς υλοποιείται ένας ιστότοπος εργαστηρίου με τις δυνατότητες του Drupal 10; Και, τελικά, σε ποια σημεία υπερτερεί το Drupal 10 έναντι του WordPress για την περίπτωση ενός ακαδημαϊκού εργαστηρίου, και σε ποια υπολείπεται;

1.4 Μεθοδολογία

Για την επίτευξη των παραπάνω στόχων ακολουθήθηκε η εξής μεθοδολογία: αρχικά μελετήθηκε το Drupal 10 ως σύστημα, οι βασικές του έννοιες και η αρχιτεκτονική του, καθώς και οι λόγοι για τους οποίους η μετάβαση από D7 δεν είναι απλή υπόθεση. Στη συνέχεια σχεδιάστηκε και υλοποιήθηκε ο νέος ιστότοπος, με έμφαση στη σωστή δομή των content types, στη χρήση του module Views για την εμφάνιση του περιεχομένου και στην προσαρμογή της εμφάνισης μέσω CSS Asset Injection.

Για την αξιολόγηση και σύγκριση των δύο υλοποιήσεων χρησιμοποιήθηκαν τρία καθιερωμένα εργαλεία: το Google PageSpeed Insights για απόδοση, προσβασιμότητα και SEO, το GTmetrix για λεπτομερείς μετρικές φόρτωσης, και το SecurityHeaders.com για την αξιολόγηση των HTTP security headers. Οι μετρήσεις πραγματοποιήθηκαν σε πέντε αντιπροσωπευτικές σελίδες κάθε υλοποίησης και όχι μόνο στην αρχική, ώστε τα αποτελέσματα να αντικατοπτρίζουν τη συνολική συμπεριφορά κάθε CMS και όχι ιδιαιτερότητες μιας μόνο σελίδας. Σημαντικό μεθοδολογικό πλεονέκτημα αποτελεί το γεγονός ότι και οι δύο ιστότοποι φιλοξενούνται στον ίδιο server του Πανεπιστημίου Πατρών, εξαλείφοντας έτσι τις διαφορές υποδομής ως παράγοντα επηρεασμού των αποτελεσμάτων.

1.5 Δομή του κειμένου της εργασίας

Η εργασία οργανώνεται ως εξής: στο δεύτερο κεφάλαιο παρουσιάζεται το θεωρητικό υπόβαθρο: το Drupal ως σύστημα διαχείρισης περιεχομένου, η ιστορία του, ο τρόπος που λειτουργεί και η σύγκρισή του με άλλα CMS. Στο τρίτο κεφάλαιο περιγράφεται η διαδικασία κατασκευής μιας σελίδας με Drupal 10 και τα πράγματα που πρέπει να έχει κανείς υπόψη του. Το τέταρτο κεφάλαιο καλύπτει τις δυσκολίες που παρουσιάζει μια αναβάθμιση από D7 σε D10 και γιατί δεν πρόκειται για απλή υπόθεση. Στο πέμπτο κεφάλαιο παρουσιάζεται η υλοποίηση της νέας σελίδας. Στο έκτο κεφάλαιο γίνεται συστηματική αξιολόγηση και σύγκριση της νέας σελίδας σε Drupal 10 με την αντίστοιχη σε WordPress, χρησιμοποιώντας εργαλεία μέτρησης απόδοσης, προσβασιμότητας και ασφάλειας. Τέλος, στο έβδομο κεφάλαιο παρουσιάζονται τα συμπεράσματα και οι προτάσεις για μελλοντική εργασία.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

2.1 Συστήματα Διαχείρισης Περιεχομένου (CMS)

Πριν μιλήσουμε για το Drupal συγκεκριμένα, αξίζει να σταθούμε λίγο στα CMS γενικότερα. Ένα Content Management System (CMS), δηλαδή σύστημα διαχείρισης περιεχομένου, είναι λογισμικό που επιτρέπει στους χρήστες να δημιουργούν, να επεξεργάζονται και να οργανώνουν ψηφιακό περιεχόμενο σε ένα website, συνήθως χωρίς να χρειάζεται να γράφουν κώδικα από μηδέν για κάθε μικρή αλλαγή. Η ανάπτυξη των CMS διευκόλυνε σημαντικά τη δημιουργία και συντήρηση ιστότοπων από άτομα χωρίς εξειδικευμένες τεχνικές γνώσεις, συμβάλλοντας έτσι στην ευρεία διάδοση του παγκόσμιου ιστού.

Τα πιο γνωστά CMS σήμερα είναι το WordPress, το Joomla, το Drupal και το TYPO3 [4, 5]. Το καθένα έχει τα δυνατά και αδύνατα σημεία του, αλλά όλα βασίζονται στην ίδια βασική φιλοσοφία: ξεχωρίζουν το περιεχόμενο (content) από την παρουσίαση (presentation), ώστε να μπορεί κάποιος να διαχειρίζεται τη σελίδα χωρίς να χρειάζεται να είναι προγραμματιστής. Η επιλογή του κατάλληλου CMS εξαρτάται από τις ανάγκες κάθε έργου, τις τεχνικές δυνατότητες της ομάδας και τις μακροπρόθεσμες απαιτήσεις συντήρησης.

2.2 Ιστορία και φιλοσοφία του Drupal

Το Drupal δημιουργήθηκε από τον Dries Buytaert το 2001 [6]. Ξεκίνησε ως ένα απλό forum για μια ομάδα φοιτητών, αλλά γρήγορα εξελίχθηκε σε ένα open-source project με μεγάλη κοινότητα. Η πρώτη επίσημη έκδοση ανοιχτού κώδικα κυκλοφόρησε το 2001 και από τότε το Drupal έχει γίνει ένα από τα πιο διαδεδομένα CMS παγκοσμίως, ειδικά σε εταιρικά περιβάλλοντα και κυβερνητικές σελίδες.

Η φιλοσοφία του Drupal είναι να είναι πολύ ευέλικτο και επεκτάσιμο. Δεν απευθύνεται τόσο στον μέσο χρήστη που θέλει να στήσει ένα απλό blog — γι' αυτό υπάρχει το WordPress — αλλά σε πιο σύνθετες εφαρμογές όπου χρειάζεται μεγάλος βαθμός προσαρμογής. Αυτό βέβαια έχει και αντίτιμο: μεγαλύτερη καμπύλη εκμάθησης, κάτι που διαπιστώσαμε και κατά την υλοποίηση της παρούσας εργασίας.

2.3 Πώς λειτουργεί το Drupal

2.3.1 Βασικές έννοιες

Το Drupal οργανώνει το περιεχόμενο γύρω από μερικές βασικές έννοιες που είναι σημαντικό να κατανοήσει κανείς πριν ξεκινήσει οποιαδήποτε εργασία. Κάθε κομμάτι περιεχομένου αποθηκεύεται ως ένας κόμβος (node), που μπορεί να είναι ένα άρθρο, μια σελίδα, μια ανακοίνωση κ.λπ. Κάθε κόμβος ανήκει σε έναν συγκεκριμένο **τύπο περιεχομένου** (content type), ο οποίος ορίζει τη δομή του: για παράδειγμα, ένα content type «Ανακοίνωση» μπορεί να έχει πεδία για τίτλο, κείμενο, ημερομηνία και εικόνα, ενώ ένα «Μέλος» θα έχει εντελώς διαφορετικά πεδία.

Το Drupal βασίζεται σε ένα σύστημα **αρθρωμάτων** (modules). Υπάρχει ο πυρήνας (core), δηλαδή τα βασικά modules που έρχονται με κάθε εγκατάσταση, και τα contributed modules που φτιάχνει η κοινότητα. Σχεδόν κάθε λειτουργία στο Drupal, από τη διαχείριση χρηστών ως τη δημιουργία φορμών, γίνεται μέσω modules. Η εμφάνιση της σελίδας ελέγχεται από τα **θέματα εμφάνισης** (themes), τα οποία καθορίζουν πώς εμφανίζεται το περιεχόμενο στον browser. Στο Drupal 10 τα themes χρησιμοποιούν το Twig ως template engine, κάτι που αλλάζει αρκετά τα πράγματα σε σχέση με το Drupal 7.

Ένα από τα πιο χαρακτηριστικά στοιχεία του Drupal είναι το σύστημα **hooks**. Τα hooks επιτρέπουν σε ένα module να «παρεμβαίνει» στη λογική ενός άλλου module ή του core, χωρίς να τροποποιεί απευθείας τον κώδικά του. Αυτό κάνει το σύστημα πολύ επεκτάσιμο και είναι από τα βασικά χαρακτηριστικά που διαφοροποιούν το Drupal από άλλα CMS [6].

2.3.2 Αρχιτεκτονική

Το Drupal ακολουθεί μια αρχιτεκτονική τριών επιπέδων: τη βάση δεδομένων (συνήθως MySQL ή PostgreSQL), το Drupal core που χειρίζεται την επιχειρησιακή λογική, και το theming layer που παράγει τον κώδικα HyperText Markup Language (HTML) που βλέπει τελικά ο χρήστης. Αυτός ο διαχωρισμός επιτρέπει στους developers να τροποποιούν την εμφάνιση χωρίς να επηρεάζουν τη λογική, και αντίστροφα, κάτι που κάνει το σύστημα πιο εύκολο να συντηρείται μακροπρόθεσμα.

Από το Drupal 8 και μετά, το Drupal ενσωμάτωσε πολλά στοιχεία από το Symfony framework [7] — μια πολύ μεγάλη αλλαγή σε σχέση με το Drupal 7, που είχε σχεδόν εξ ολοκλήρου δικό του σύστημα. Πλέον το Drupal χρησιμοποιεί dependency injection, Symfony components, και γενικά ακολουθεί πιο σύγχρονες πρακτικές object-oriented programming [8]. Αυτή η αλλαγή έκανε το Drupal πιο ελκυστικό για έμπειρους PHP developers, καθώς εκμεταλλεύονται γνώσεις που ήδη κατέχουν από άλλα frameworks.

2.4 Σύγκριση Drupal με άλλα CMS

Όπως αναφέρθηκε παραπάνω, το Drupal δεν είναι το μοναδικό CMS στην αγορά. Αξίζει να δούμε πού στέκεται σε σχέση με τα πιο δημοφιλή από αυτά, γιατί αυτό βοηθάει να καταλάβει

κανείς γιατί επιλέχθηκε το Drupal για έναν ιστότοπο σαν αυτόν του εργαστηρίου.

WordPress

Το WordPress [9] είναι αναμφισβήτητο το πιο δημοφιλές CMS στον κόσμο — σύμφωνα με στατιστικά στοιχεία, τρέχει περίπου το 40–45% όλων των ιστότοπων παγκοσμίως [10]. Ξεκίνησε ως blogging platform το 2003 και με τα χρόνια εξελίχθηκε σε ένα γενικής χρήσης CMS. Η μεγάλη του δύναμη είναι η ευκολία χρήσης και το τεράστιο οικοσύστημα από **πρόσθετα** (plugins) και themes — σχεδόν οτιδήποτε θέλει κανείς να κάνει, υπάρχει ήδη ένα plugin γι' αυτό.

Το WordPress όμως έχει και αδυναμίες. Έχει φήμη για προβλήματα ασφαλείας, κυρίως λόγω του μεγάλου αριθμού plugins τρίτων που δεν συντηρούνται πάντα σωστά [11]. Επίσης, για πολύπλοκες εφαρμογές με ιδιαίτερες απαιτήσεις δομής δεδομένων, το WordPress αρχίζει να δείχνει τα όριά του — απλώς δεν είναι φτιαγμένο για αυτό. Σε αντίθεση με το Drupal, που υπερέχει σε δυνατότητες μοντελοποίησης περιεχομένου (content modeling) και ενδείκνυται για σύνθετους ιστότοπους με ιδιαίτερες απαιτήσεις, το WordPress δυσκολεύεται στη δημιουργία σύνθετων σχέσεων μεταξύ διαφορετικών τύπων περιεχομένου — απαιτήσεις συνηθισμένες σε ακαδημαϊκά ή εταιρικά περιβάλλοντα [12].

Joomla

Το Joomla [13] κυκλοφόρησε το 2005 και για χρόνια ήταν το δεύτερο πιο δημοφιλές CMS. Βρίσκεται κάπου ανάμεσα στο WordPress και το Drupal από άποψη πολυπλοκότητας — πιο ευέλικτο από το WordPress, αλλά πιο προσιτό από το Drupal. Έχει καλό σύστημα διαχείρισης χρηστών και υποστηρίζει πολύγλωσσες σελίδες χωρίς επιπλέον extensions.

Το πρόβλημα με το Joomla είναι ότι τα τελευταία χρόνια έχει χάσει έδαφος. Η κοινότητά του έχει συρρικνωθεί σε σχέση με WordPress και Drupal, κάτι που σημαίνει λιγότερα διαθέσιμα extensions και πιο αργές ενημερώσεις. Αυτό αντικατοπτρίζεται και στα στατιστικά χρήσης: σύμφωνα με δεδομένα του W3Techs [10], το μερίδιο αγοράς του Joomla έχει μειωθεί σταδιακά τα τελευταία χρόνια, ενώ το WordPress και το Drupal διατηρούν πιο σταθερή παρουσία.

TYPO3

Το TYPO3 [14] είναι λιγότερο γνωστό στο ευρύ κοινό αλλά έχει σημαντική παρουσία στην Ευρώπη, ειδικά σε γερμανόφωνες χώρες. Απευθύνεται κυρίως σε μεγάλους οργανισμούς και επιχειρήσεις. Είναι πολύ δυνατό και ευέλικτο, με εξαιρετικό σύστημα διαχείρισης χρηστών και δικαιωμάτων, αλλά η καμπύλη εκμάθησής του είναι ακόμα πιο απότομη από του Drupal.

Συγκριτικός πίνακας

Ο παρακάτω Πίνακας 2.1 συνοψίζει τις κύριες διαφορές μεταξύ των τεσσάρων CMS [15, 16, 12, 17, 10]:

Πίνακας 2.1: Σύγκριση δημοφιλών CMS

Χαρακτηριστικό	Drupal	WordPress	Joomla	TYPO3
Ευκολία χρήσης	Δύσκολο	Εύκολο	Μέτριο	Πολύ δύσκολο
Ευελιξία	Πολύ υψηλή	Μέτρια	Υψηλή	Πολύ υψηλή
Απόδοση	Υψηλή	Μέτρια	Μέτρια	Υψηλή
Ασφάλεια	Υψηλή	Μέτρια	Μέτρια	Υψηλή
Κοινότητα	Μεγάλη	Τεράστια	Μέτρια	Μεσαία
Κατάλληλο για	Enterprise	Blog/SMB	Γενική χρήση	Enterprise

Από τον πίνακα φαίνεται καθαρά ότι το Drupal δεν είναι πάντα η καλύτερη επιλογή [12, 17]. Για ένα απλό blog ή μια μικρή επιχείρηση, το WordPress είναι προτιμότερη λύση, καθώς προσφέρει ευκολία χρήσης, ταχεία ανάπτυξη και χαμηλό κόστος συντήρησης, χωρίς την πολυπλοκότητα ενός συστήματος όπως το Drupal. Το Drupal όμως είναι η κατάλληλη επιλογή όταν απαιτείται μεγάλη ευελιξία στη δομή των δεδομένων [12, 8], υψηλές απαιτήσεις ασφαλείας [12, 15], και μια σελίδα που θα πρέπει να συντηρείται και να εξελίσσεται μακροπρόθεσμα — ακριβώς δηλαδή όπως η σελίδα ενός πανεπιστημιακού εργαστηρίου.

2.5 Drupal 7 έναντι Drupal 10

Η διαφορά ανάμεσα στο Drupal 7 και στο Drupal 10 δεν είναι απλώς αριθμητική. Πρόκειται ουσιαστικά για δύο διαφορετικά συστήματα που τυχαίνει να μοιράζονται το ίδιο όνομα. Αυτό δεν είναι υπερβολή: η μετάβαση από D7 σε D10 απαιτεί πρακτικά πλήρη επανασχεδιασμό του site, καθώς καμία γραμμή κώδικα ή ρύθμιση δεν μπορεί να μεταφερθεί αυτούσια μεταξύ των δύο εκδόσεων.

2.5.1 Αρχιτεκτονικές και τεχνικές διαφορές

Το Drupal 7 βασιζόταν σε μια αρχιτεκτονική σχεδιασμένη το 2011. Έχει procedural PHP: Hypertext Preprocessor (PHP) κώδικα σε μεγάλο βαθμό, χρησιμοποιεί το δικό του template σύστημα (PHPTemplate), και η διαχείριση εξαρτήσεων γίνεται χειροκίνητα. Δεν υπάρχει Composer, δεν υπάρχει namespace — τίποτα από τα σύγχρονα εργαλεία PHP. Αυτό σημαίνει ότι η εγκατάσταση modules και ενημερώσεων γινόταν με μη αυτόματες διαδικασίες, επιρρεπείς σε λάθη και χρονοβόρες.

Το Drupal 10 από την άλλη απαιτεί PHP 8.1 ή νεότερο, χρησιμοποιεί Composer για διαχείριση εξαρτήσεων, το theming γίνεται με Twig templates, και ο κώδικας είναι object-oriented με namespaces και interfaces. Αξίζει επίσης να σημειωθεί ότι το Drupal 10 απαιτεί τουλάχιστον MySQL 5.7.8, ενώ το D7 δούλευε και με πολύ παλαιότερες εκδόσεις.

Μια ακόμα σημαντική διαφορά είναι το configuration management. Στο D7 οι ρυθμίσεις αποθηκεύονταν κυρίως στη βάση δεδομένων. Στο D10 υπάρχει ένα πλήρες σύστημα configuration management, όπου οι ρυθμίσεις μπορούν να εξαχθούν ως αρχεία YAML (YAML Ain't

Markup Language) και να τεθούν υπό έλεγχο εκδόσεων (version control, π.χ. με Git) — δηλαδή σε ένα σύστημα που καταγράφει κάθε αλλαγή, επιτρέπει την επαναφορά σε προηγούμενη κατάσταση και τη μεταφορά ρυθμίσεων μεταξύ περιβαλλόντων (π.χ. από το τοπικό στο production). Έτσι η διαχείριση των ρυθμίσεων γίνεται πιο αξιόπιστη και ελεγχόμενη, σε αντίθεση με το D7, όπου οι ρυθμίσεις ήταν ουσιαστικά ‘κλειδωμένες’ μέσα στη βάση δεδομένων.

2.5.2 Ασφάλεια και το τέλος ζωής του Drupal 7

Ένα από τα πιο κρίσιμα θέματα στη σύγκριση D7 και D10 είναι η ασφάλεια. Και δεν είναι τυχαίο ότι αυτό ήταν και το βασικό κίνητρο για την αναβάθμιση του ιστότοπου του εργαστηρίου. Η ασφάλεια ενός CMS δεν εξαρτάται μόνο από τον κώδικά του, αλλά και από το αν υπάρχει ενεργή κοινότητα που το συντηρεί και εκδίδει ενημερώσεις όταν βρεθούν ευπάθειες [3].

Τι σημαίνει End-of-Life στην πράξη

Το Drupal 7 έφτασε επίσημα στο τέλος του κύκλου ζωής του (end-of-life, EOL) στις 5 Ιανουαρίου 2025, αφού είχε δοθεί παράταση αρκετές φορές λόγω της μεγάλης βάρσης χρηστών [1]. Το EOL σημαίνει ότι η ομάδα ασφαλείας του Drupal (Drupal Security Team) σταματά να εκδίδει patches για ευπάθειες που ανακαλύπτονται στο D7 core και στα contributed modules.

Αυτό είναι πρόβλημα γιατί οι ευπάθειες δεν σταματούν να ανακαλύπτονται μόλις ένα λογισμικό φτάσει στο EOL — αντίθετα, οι επιτιθέμενοι ξέρουν ότι δεν θα υπάρξει ποτέ patch και εκμεταλλεύονται αυτό. Ένα site που τρέχει D7 μετά το EOL είναι ουσιαστικά ένα site με γνωστές, μη διορθωμένες ευπάθειες. Στην πράξη, αυτό σημαίνει ότι ακόμα και αν το site δεν αποτελεί υψηλής αξίας στόχο, μπορεί να χρησιμοποιηθεί ως βάση για επιθέσεις σε τρίτους ή για διανομή κακόβουλου λογισμικού.

Τύποι ευπαθειών

Τα CMS όπως το Drupal είναι συχνός στόχος επιθέσεων γιατί τρέχουν σε πολλά sites ταυτόχρονα [3, 18] — αν βρεθεί μια ευπάθεια, μπορεί να εκμεταλλευτεί αυτόματα σε χιλιάδες sites με το ίδιο script. Οι πιο συνηθισμένοι τύποι ευπαθειών που αφορούν CMS περιλαμβάνουν το **Structured Query Language (SQL) Injection**, όπου ο επιτιθέμενος εισάγει κακόβουλο SQL κώδικα μέσω φορμών ή URLs αποκτώντας πρόσβαση στη βάση δεδομένων, το **Cross-Site Scripting (XSS)**, που αφορά εισαγωγή κακόβουλου JavaScript σε περιεχόμενο το οποίο μετά εκτελείται στον browser άλλων χρηστών, και το **Remote Code Execution (RCE)**, που είναι η πιο σοβαρή κατηγορία καθώς επιτρέπει στον επιτιθέμενο να εκτελέσει αυθαίρετο κώδικα στον server. Επιπλέον, το **Cross-Site Request Forgery (CSRF)** εξαπατά τον χρήστη ώστε να εκτελέσει ακούσιες ενέργειες στη σελίδα, ενώ το **Privilege Escalation** επιτρέπει απόκτηση υψηλότερων δικαιωμάτων από αυτά που δικαιούται ο χρήστης [2].

Το Drupal έχει ιστορικό με αρκετά σοβαρές ευπάθειες. Χαρακτηριστικό παράδειγμα είναι το **Drupalgeddon** (SA-CORE-2014-005) [19], μια κρίσιμη SQL injection ευπάθεια που

επηρέασε το Drupal 7 και έγινε αντικείμενο μαζικής εκμετάλλευσης από επιτιθέμενους μέσα σε λίγες ώρες από την ανακοίνωσή της. Παρόμοιες ευπάθειες εμφανίστηκαν και αργότερα (Drupalgeddon 2 και 3 το 2018) [20], δείχνοντας πόσο γρήγορα μπορεί να γίνει εκμετάλλευση μόλις μια ευπάθεια γίνει γνωστή.

Η ομάδα ασφαλείας του Drupal

Το Drupal έχει μια οργανωμένη ομάδα ασφαλείας (Drupal Security Team) που είναι υπεύθυνη για την ανακάλυψη και αντιμετώπιση ευπαθειών [6]. Όταν βρεθεί κάποια ευπάθεια, εκδίδεται ένα δελτίο ασφαλείας (Security Advisory, SA) με λεπτομέρειες για το πρόβλημα και το αντίστοιχο patch. Αυτό το σύστημα λειτουργεί καλά για εκδόσεις που βρίσκονται σε ενεργή συντήρηση — για το D7 μετά το EOL όμως, αυτή η προστασία έχει πλέον εκλείψει. Αξίζει να σημειωθεί ότι η ομάδα ασφαλείας του Drupal θεωρείται από τις πιο οργανωμένες στον χώρο των CMS, με σαφείς διαδικασίες αναφοράς και αντιμετώπισης ευπαθειών [3].

Πώς βελτιώνεται η κατάσταση στο Drupal 10

Το Drupal 10 δεν είναι από μόνο του πιο ασφαλές με την έννοια ότι δεν έχει ευπάθειες — κάθε λογισμικό μπορεί να έχει. Η διαφορά είναι ότι το D10 βρίσκεται σε ενεργή συντήρηση, οπότε οι ευπάθειες που ανακαλύπτονται διορθώνονται γρήγορα με security releases. Επιπλέον, η χρήση του Composer κάνει την εφαρμογή αυτών των updates πολύ πιο εύκολη — αντί να κατεβάζουμε και να αντικαθιστούμε αρχεία χειροκίνητα όπως στο D7, αρκεί ένα `composer update` για να ληφθούν τα τελευταία patches. Τέλος, το D10 χρησιμοποιεί PHP 8.x που έχει καλύτερα built-in χαρακτηριστικά ασφαλείας σε σχέση με τις παλιές εκδόσεις PHP που έτρεχε το D7.

Συνοπτικά, η μετάβαση στο D10 δεν εγγυάται μηδενικές ευπάθειες, αλλά εγγυάται ότι η ομάδα ασφαλείας και η κοινότητα του Drupal εξακολουθούν να τις διορθώνουν όταν βρεθούν — κάτι που για το D7 μετά το EOL δεν ισχύει πλέον. Από πρακτική σκοπιά, αυτό είναι η ουσιαστική διαφορά ασφάλειας μεταξύ των δύο εκδόσεων [1].

Κεφάλαιο 3

Κατασκευή Ιστότοπου με Drupal 10

3.1 Απαιτήσεις συστήματος και εγκατάσταση

Για να στηθεί ένα Drupal 10 site χρειάζονται κάποιες βασικές προϋποθέσεις: ένας web server (Apache ή Nginx), PHP 8.1 ή νεότερο, και μια βάση δεδομένων (MySQL, MariaDB ή PostgreSQL). Το πιο σημαντικό εργαλείο για ένα σύγχρονο Drupal site είναι αναμφίβολα το Composer.

Το **Composer** είναι ο package manager της PHP [21]. Στο Drupal 10, κάθε module, κάθε theme και το ίδιο το Drupal core διαχειρίζεται μέσω αυτού. Αυτό κάνει τις ενημερώσεις πολύ πιο εύκολες σε σχέση με το D7, όπου έπρεπε να κατεβάζουμε και να αντικαθιστούμε αρχεία χειροκίνητα — μια διαδικασία που ήταν επίπονη και επιρρεπής σε λάθη.

Η εγκατάσταση ξεκινά συνήθως με:

```
1 composer create-project drupal/recommended-project my_site
2 cd my_site
3 composer require drush/drush
```

Το Drush [22] είναι ένα command-line εργαλείο για το Drupal που επιταχύνει σημαντικά πολλές εργασίες, όπως cache clear, database updates, module enable/disable κ.λπ. Είναι ιδιαίτερα χρήσιμο σε περιβάλλοντα παραγωγής όπου η χρήση του Περιβάλλοντος διαχείρισης (Admin interface) μέσω browser είναι αδύνατη ή αναποτελεσματική, και αποτελεί βασικό εργαλείο κάθε Drupal developer.

3.2 Το Admin interface

Ένα από τα πρώτα πράγματα που παρατηρεί κανείς όταν στήσει ένα Drupal 10 site είναι το Admin interface — το περιβάλλον μέσα από το οποίο γίνεται η διαχείριση του drupal ιστοτόπου. Στην περίπτωση της συγκεκριμένης εργασίας, το Admin interface ήταν το κύριο εργαλείο δουλειάς, αφού όλη η ανάπτυξη έγινε αποκλειστικά μέσα από αυτό.

Το Admin interface του Drupal 10 είναι προσβάσιμο από το `/admin` Uniform Resource Locator (URL) και οργανώνεται γύρω από μια γραμμή εργαλείων (toolbar) στην κορυφή της σελίδας. Η ενότητα **Content** (`/admin/content`) είναι το σημείο από το οποίο διαχειριζόμαστε το περιεχόμενο του Drupal ιστότοπου: εμφανίζεται μια λίστα με όλα τα nodes που έχουν δημιουργηθεί, με δυνατότητα φιλτραρίσματος ανά content type, κατάσταση (published/unpublished) και γλώσσα.

Η ενότητα **Structure** (`/admin/structure`) είναι ίσως η πιο σημαντική για τον προγραμματιστή, καθώς περιέχει εργαλεία για τη δημιουργία και επεξεργασία των content types και των fields τους, τη διαχείριση των navigation menus και των Views, καθώς και την κατηγοριοποίηση περιεχομένου μέσω του Taxonomy. Η ενότητα **Appearance** (`/admin/appearance`) διαχειρίζεται τα themes: από εδώ μπορούμε να εγκαταστήσουμε νέα themes, να ορίσουμε το default theme και να ρυθμίσουμε τις επιλογές του.

Η ενότητα **Extend** (`/admin/modules`) είναι η σελίδα διαχείρισης modules. Εδώ εμφανίζονται όλα τα διαθέσιμα modules οργανωμένα σε κατηγορίες και μπορούν να ενεργοποιηθούν ή να απενεργοποιηθούν. Σημαντική λεπτομέρεια: η εγκατάσταση νέων modules γίνεται μέσω Composer και όχι από εδώ — το Extend απλώς ενεργοποιεί modules που έχουν ήδη κατέβει. Η ενότητα **Configuration** (`/admin/config`) περιέχει όλες τις ρυθμίσεις του site, μεταξύ των οποίων βρίσκεται και το Asset Injector που χρησιμοποιήσαμε για το custom CSS της σελίδας. Τέλος, η ενότητα **Reports** (`/admin/reports`) παρέχει πληροφορίες για την κατάσταση του site και τυχόν προβλήματα που εντοπίζει το σύστημα, ενώ η **People** (`/admin/people`) διαχειρίζεται τους χρήστες και τους ρόλους τους.

3.3 Content Types και Fields

Ένα από τα πρώτα πράγματα που γίνονται κατά την κατασκευή ενός Drupal site είναι ο ορισμός των content types — δηλαδή απαντάμε στο ερώτημα: τι είδη περιεχομένου χρειάζεται η σελίδα; Για ένα εργαστηριακό site όπως αυτό που υλοποιήσαμε, τα τυπικά είδη περιεχομένου περιλαμβάνουν μια **Basic Page** για στατικές σελίδες όπως «Επικοινωνία» ή «Σχετικά με εμάς», στοιχεία τύπου **Μέλος Εργαστηρίου** για τα προφίλ των μελών, **Δημοσίευση** για τις επιστημονικές δημοσιεύσεις, **Ερευνητικό Πρόγραμμα** για τα projects και **Νέα** για ανακοινώσεις.

Κάθε content type παίρνει τα fields που χρειάζεται. Το Drupal 10 παρέχει διάφορους τύπους fields: text, image, file, date, reference (για σχέσεις μεταξύ κόμβων), link κ.λπ. Αυτό το σύστημα είναι πολύ δυνατό γιατί μπορεί να κατασκευαστεί σχεδόν οποιαδήποτε δομή δεδομένων απαιτείται, χωρίς να χρειαστεί να γραφεί κώδικας.

3.4 Το σύστημα Theming

3.4.1 Twig Templates

Στο Drupal 10, η εμφάνιση ελέγχεται από Twig templates [23]. Το Twig είναι ένα template engine που χωρίζει καθαρά την παρουσίαση από τη λογική — μια σημαντική βελτίωση σε σχέση με το D7, όπου τα templates ήταν απλά PHP αρχεία.

Ένα βασικό Twig template μοιάζει κάπως έτσι:

```
1 {# node--article.html.twig #}
2 <article{{ attributes }}>
3   <h1>{{ label }}</h1>
4   <div class="content">
5     {{ content }}
6   </div>
7   {% if content.field_image %}
8     <div class="article-image">
9       {{ content.field_image }}
10    </div>
11  {% endif %}
12 </article>
```

3.4.2 Base Themes

Κατά την κατασκευή ενός custom theme, συνήθως δεν ξεκινάμε από το μηδέν. Η συνήθης πρακτική είναι να βασιστεί κανείς σε ένα base theme που παρέχει κάποια βασική δομή. Τα πιο δημοφιλή base themes για D10 είναι το **Olivero** (που είναι το default theme του D10), το **Bootstrap**, το **Barrio** (Bootstrap-based), και το **Gin** για το Admin interface.

Το sub-theming γίνεται δηλώνοντας στο `.info.yml` αρχείο του theme ποιο είναι το base theme:

```
1 name: 'My Custom Theme'
2 type: theme
3 base theme: olivero
4 version: '10.x'
```

3.4.3 Asset Injection

Ένας τρόπος για να προστεθεί custom CSS σε ένα Drupal site χωρίς να χρειαστεί εργασία απευθείας με αρχεία κώδικα είναι μέσω του module **Asset Injector**. Το module αυτό παρέχει ένα interface μέσα από το admin όπου μπορούν να δημιουργηθούν CSS (ή JavaScript) rules και να οριστεί σε ποιες σελίδες θα εφαρμόζονται, χρησιμοποιώντας συνθήκες βασισμένες στα μοτίβα του URL.

Η προσέγγιση αυτή έχει ένα σημαντικό πλεονέκτημα έναντι της τροποποίησης απευθείας των αρχείων του theme: οι αλλαγές αποθηκεύονται στη βάση δεδομένων και δεν χάνονται κατά την ενημέρωση του theme. Ταυτόχρονα, επιτρέπει λεπτομερή έλεγχο — μπορούμε

να ορίσουμε CSS που εφαρμόζεται μόνο σε συγκεκριμένες σελίδες (π.χ. μόνο στη σελίδα /Publications*) χωρίς να επηρεάζει τις υπόλοιπες. Στην υλοποίησή μας δημιουργήσαμε 12 ξεχωριστούς injectors, έναν για κάθε κύρια ενότητα του site.

3.5 Modules — Core και Contributed

Το Drupal 10 έρχεται με αρκετά core modules που καλύπτουν βασικές ανάγκες. Η προσθήκη contributed modules γίνεται πολύ απλά με Composer:

```
1 composer require drupal/pathauto
2 drush en pathauto
```

3.5.1 Το Module Views

Οι **Προβολές** (Views) είναι από τα πιο δυνατά και συχνά χρησιμοποιούμενα modules του Drupal, και από το Drupal 8 και μετά αποτελούν μέρος του core [24, 25]. Στην ουσία, επιτρέπουν στον χρήστη να δημιουργεί ερωτήματα στη βάση δεδομένων και να εμφανίζει τα αποτελέσματα με οποιοδήποτε τρόπο θέλει, χωρίς να χρειάζεται να γράψει SQL ή PHP.

Displays

Κάθε View μπορεί να έχει πολλαπλά **displays**: **Page display** εμφανίζει το View σε μια συγκεκριμένη URL, **Block display** εμφανίζει το View ως block που μπορεί να τοποθετηθεί σε οποιοδήποτε region της σελίδας, και **Feed display** παράγει RSS feed. Ένα View μπορεί να έχει ταυτόχρονα πολλαπλά displays που μοιράζονται τις ίδιες ρυθμίσεις — για παράδειγμα, η λίστα δημοσιεύσεων μπορεί να εμφανίζεται ταυτόχρονα ως πλήρης σελίδα και ως μικρό block με τις τελευταίες εγγραφές στην πλαϊνή στήλη.

Filters

Τα **filters** επιτρέπουν να περιορίσουμε ποιο περιεχόμενο εμφανίζεται. Για παράδειγμα, σε ένα View που δείχνει projects μπορούμε να προσθέσουμε filter ώστε να εμφανίζονται μόνο τα ενεργά. Τα filters μπορούν να είναι **exposed**, δηλαδή να εμφανίζονται στον χρήστη ως φόρμα αναζήτησης ή φίλτρου — αυτό χρησιμοποιήσαμε για την εμφάνιση δημοσιεύσεων ανά έτος, όπου ο χρήστης επιλέγει το έτος από dropdown μενού.

Contextual Filters

Τα **contextual filters** παίρνουν την τιμή τους δυναμικά — συνήθως από το URL της σελίδας. Έτσι με ένα μόνο View καλύπτονται πολλαπλές σελίδες. Για παράδειγμα, ένα View για τις δημοσιεύσεις ενός μέλους μπορεί να χρησιμοποιεί ως contextual filter το ID του μέλους από το URL, εμφανίζοντας αυτόματα τις σωστές δημοσιεύσεις για κάθε μέλος χωρίς να χρειαστεί ξεχωριστό View για τον καθένα.

3.6 Configuration Management

Ένα από τα πιο χρήσιμα χαρακτηριστικά του Drupal 8+ είναι το configuration management system. Επιτρέπει την εξαγωγή όλων των ρυθμίσεων (content types, views, fields κλπ.) σε YAML αρχεία:

```
1 # Export configuration
2 drush cex
3
4 # Import configuration
5 drush cim
```

Η πρακτική αξία αυτού του συστήματος είναι σημαντική. Πρώτον, επιτρέπει την αποθήκευση των ρυθμίσεων σε version control (π.χ. Git), κάτι που σημαίνει ότι κάθε αλλαγή στη δομή του site παρακολουθείται και μπορεί να αναιρεθεί. Δεύτερον, διευκολύνει τη μεταφορά ρυθμίσεων μεταξύ περιβαλλόντων: αν αναπτύσσουμε σε τοπικό περιβάλλον και θέλουμε να μεταφέρουμε τις αλλαγές στο production server, αρκεί να εξαγάγουμε τις ρυθμίσεις, να τις ανεβάσουμε μέσω Git και να τις εισαγάγουμε στον server. Αυτή η ροή εργασίας είναι πολύ πιο αξιόπιστη από την εναλλακτική του D7, όπου οι ρυθμίσεις ήταν αποκλειστικά στη βάση δεδομένων και η μεταφορά τους απαιτούσε εξαγωγή και εισαγωγή βάσης ή χειροκίνητη αναπαραγωγή των αλλαγών [8].

Κεφάλαιο 4

Δυσκολίες αναβάθμισης από Drupal 7 σε Drupal 10

4.1 Γιατί δεν είναι απλή η αναβάθμιση

Πολλοί χρήστες του Drupal 7 που ακούν για ‘αναβάθμιση’ σε D10 υποθέτουν ότι πρόκειται για κάτι που μπορεί να γίνει με λίγα βήματα, σαν την ενημέρωση ενός πρόσθετου (plugin). Αυτή η παρανόηση είναι αρκετά συνηθισμένη — και αρκετά επικίνδυνη. Η πραγματικότητα είναι πολύ διαφορετική: η αναβάθμιση από D7 σε D10 είναι μια σύνθετη διαδικασία για αρκετούς λόγους.

Πρώτα απ’ όλα, δεν υπάρχει απευθείας μονοπάτι αναβάθμισης. Θεωρητικά υπάρχει το Migrate Application Programming Interface (API) που μπορεί να βοηθήσει, αλλά στην πράξη τα πράγματα είναι πολύ πιο περίπλοκα από ό,τι ακούγεται. Αντίθετα με άλλα CMS όπως το WordPress, που επιτρέπει σχετικά εύκολη αναβάθμιση μεταξύ εκδόσεων, το Drupal απαιτεί ουσιαστικά νέα εγκατάσταση και χειροκίνητη μεταφορά του περιεχομένου.

4.2 Η διαφορά αρχιτεκτονικής

Το D7 και το D10 είναι ουσιαστικά διαφορετικά συστήματα. Ο κώδικας που έχει γραφτεί για D7 — custom modules, themes — δεν λειτουργεί σε D10. Για παράδειγμα, ένα απλό hook στο D7 έμοιαζε έτσι:

```
1 // Drupal 7
2 function mymodule_node_view($node, $view_mode, $langcode) {
3   if ($node->type == 'article') {
4     $node->content['extra_info'] = array(
5       '#markup' => 'Some extra info',
6     );
7   }
8 }
```

Στο D10, το ίδιο πράγμα γίνεται με εντελώς διαφορετικό τρόπο, χρησιμοποιώντας event subscribers ή hooks με διαφορετική σύνταξη. Η διαφορά δεν είναι απλώς συντακτική —

αντικατοπτρίζει μια θεμελιωδώς διαφορετική φιλοσοφία ανάπτυξης, βασισμένη στις αρχές του object-oriented programming και του Symfony framework [7].

4.3 Τα modules

Ένα από τα μεγαλύτερα προβλήματα είναι τα contributed modules. Πολλά modules που υπήρχαν για D7 δεν υπάρχουν για D10, ή έχουν αλλάξει τελείως. Κάποια modules που ήταν ξεχωριστά στο D7 έχουν ενσωματωθεί στον D10 core. Άλλα απλώς έχουν εγκαταλειφθεί από τους developers τους.

Χαρακτηριστικό παράδειγμα είναι το module **Views**: στο D7 ήταν contributed module που έπρεπε να εγκατασταθεί ξεχωριστά, ενώ στο D10 αποτελεί μέρος του core. Αντίθετα, το module **CCK** (Content Construction Kit) που ήταν ο βασικός τρόπος ορισμού fields στο D7, δεν υπάρχει στο D10 — η λειτουργικότητά του έχει ενσωματωθεί στον core ως Field API. Αυτό σημαίνει ότι ακόμα και αν το περιεχόμενο μεταφερθεί επιτυχώς, η λειτουργικότητα που παρείχαν τα παλιά modules πρέπει να ανακατασκευαστεί με διαφορετικά μέσα στο D10.

4.4 Τα themes

Ένα custom theme για D7 χρησιμοποιεί PHP templates (`.tpl.php`), ενώ ένα D10 theme χρησιμοποιεί Twig (`.html.twig`). Δεν είναι απλώς θέμα επέκτασης αρχείου — η σύνταξη είναι εντελώς διαφορετική. Στο D7 τα templates ήταν ουσιαστικά PHP αρχεία που μπορούσαν να περιέχουν οποιαδήποτε λογική, κάτι που ήταν επικίνδυνο από άποψη ασφάλειας. Το Twig του D10 λειτουργεί εκ σχεδιασμού σε απομονωμένο περιβάλλον (sandbox) — δεν επιτρέπει εκτέλεση αυθαίρετου PHP κώδικα, κάνοντας τα πρότυπα (templates) πολύ πιο ασφαλή [23].

Για παράδειγμα, το access στα variables στο D7:

```
1 <!-- Drupal 7 .tpl.php -->
2 <?php print $title; ?>
3 <?php print render($content); ?>
```

Ενώ στο D10 Twig:

```
1 {# Drupal 10 Twig #}
2 {{ title }}
3 {{ content }}
```

4.5 Το Migrate API και οι περιορισμοί του

Το Drupal παρέχει το Migrate API [26], ένα σύνολο modules που θεωρητικά μπορούν να μεταφέρουν δεδομένα από D7 σε D10. Στην πράξη όμως έχει αρκετούς περιορισμούς, κυρίως όταν υπάρχουν custom modules στο D7 που αποθηκεύουν δεδομένα με δικό τους τρόπο, ή όταν η δομή των δεδομένων στο D7 δεν αντιστοιχεί εύκολα στη νέα δομή.

Στη συγκεκριμένη περίπτωση, το παλιό site του εργαστηρίου χρησιμοποιούσε custom content types για τις δημοσιεύσεις και τα projects, με πεδία που δεν αντιστοιχούσαν άμεσα στα νέα. Δηλαδή, μέρος του περιεχομένου ήταν αποθηκευμένο σε μορφή που θα απαιτούσε και χειροκίνητη επεξεργασία. Ακόμα και αν το Migrate API κατάφερνε να μεταφέρει τα δεδομένα, θα έπρεπε στη συνέχεια να γίνει εκτεταμένος καθαρισμός και αναδιάρθρωσή τους, κάτι που θα αναιρούσε σε μεγάλο βαθμό το όφελος της αυτοματοποιημένης μεταφοράς. Τελικά κρίθηκε ότι η χειροκίνητη εισαγωγή του περιεχομένου στο νέο site ήταν πιο αποδοτική και ασφαλής επιλογή.

4.6 Άλλα πρακτικά ζητήματα

Εκτός από τα τεχνικά θέματα, υπάρχουν και μερικά πρακτικά ζητήματα που αξίζει να αναφερθούν. Πρώτον, ο χρόνος: μια σωστή αναβάθμιση απαιτεί πολύ περισσότερο χρόνο από ό,τι εκτιμά κανείς αρχικά, και η υποτίμηση αυτή είναι από τους πιο συνηθισμένους λόγους που τέτοια project καθυστερούν ή ολοκληρώνονται με συμβιβασμούς. Δεύτερον, είναι απαραίτητο να υπάρχει ένα μεταβατικό (staging) περιβάλλον για να δοκιμαστεί ο νέος ιστότοπος πριν βγει σε δημόσια χρήση (production), γιατί πολλά προβλήματα εμφανίζονται μόνο σε πραγματικές συνθήκες. Τρίτον, αν αλλάξουν τα URLs, μπορεί να επηρεαστεί αρνητικά η κατάταξη στα search engines, οπότε πρέπει να υπάρχουν redirects από τα παλιά URLs στα νέα.

4.7 Η απόφαση για πλήρη ανακατασκευή

Λαμβάνοντας υπόψη όλα τα παραπάνω, αποφασίστηκε να γίνει πλήρης ανακατασκευή του ιστότοπου από μηδέν. Η απόφαση αυτή βασίστηκε σε τρεις κυρίως παράγοντες: πρώτον, το παλιό site είχε σχετικά μικρό όγκο περιεχομένου, δεύτερον, η πλήρης ανακατασκευή δίνει την ευκαιρία να σχεδιαστεί σωστά η δομή από την αρχή, και τρίτον, ο κώδικας που θα προκύψει είναι πολύ πιο καθαρός και εύκολος να συντηρηθεί μελλοντικά.

Η απόφαση αυτή, αν και αρχικά φαίνεται πιο χρονοβόρα, αποδείχθηκε στην πράξη η ορθότερη. Ξεκινώντας από μηδενική βάση είχαμε την ελευθερία να επανασχεδιάσουμε τα content types ακολουθώντας τις συνιστώμενες πρακτικές ανάπτυξης του Drupal 10 [8, 25], να επιλέξουμε τα κατάλληλα modules χωρίς περιορισμούς συμβατότητας και να δημιουργήσουμε ένα site που ευθυγραμμίζεται με τη σύγχρονη αρχιτεκτονική του Drupal 10. Αν είχαμε επιλέξει τη διαδρομή της μεταφοράς μέσω Migrate API, θα αντιμετωπίζαμε αναπόφευκτα προβλήματα συμβατότητας και θα καταλήγαμε σε ένα site με «κληρονομημένες» δομές από το D7 που θα δυσκόλευαν τη μελλοντική συντήρηση.

Κεφάλαιο 5

Υλοποίηση του νέου ιστότοπου

5.1 Επισκόπηση του ιστότοπου

Ο ιστότοπος <https://telematics.upatras.gr/>, υλοποιημένος σε Wordpress, είναι ο ιστότοπος του εργαστηρίου Τηλεματικής και Κατανεμημένων Συστημάτων του Τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής του Πανεπιστημίου Πατρών. Το κοινό-στόχος της σελίδας είναι κυρίως ακαδημαϊκοί, ερευνητές, υποψήφιοι συνεργάτες και φοιτητές που ενδιαφέρονται για τις δραστηριότητες του εργαστηρίου. Τον ιστότοπο υλοποιεί σε Drupal 10 μπορείτε να επισκεφτείτε στο <https://telematics1.upatras.gr/>.

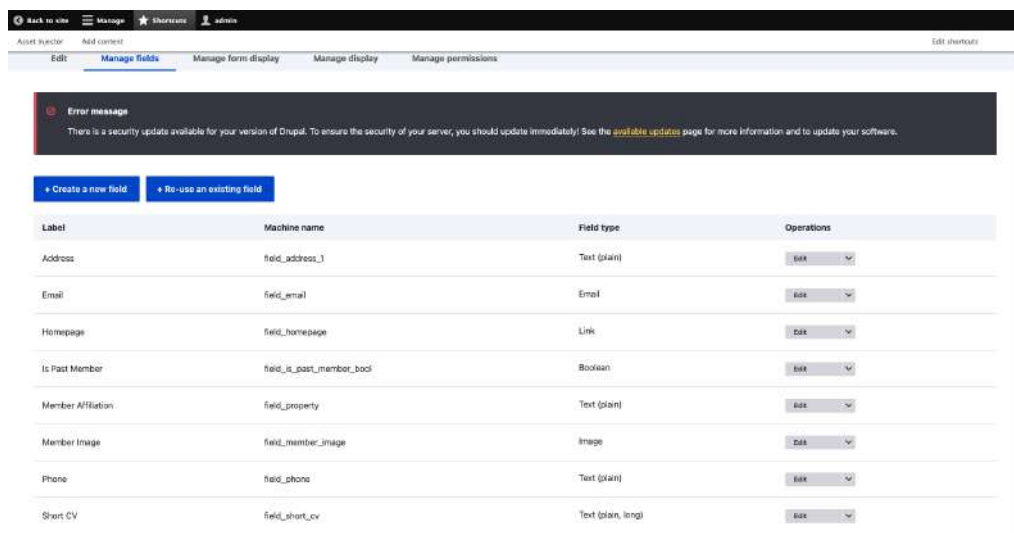
5.2 Τεχνολογίες που χρησιμοποιήθηκαν

Για την ανάπτυξη του νέου ιστότοπου χρησιμοποιήθηκαν το **Drupal 10.1.2** ως βασικό CMS, ο **Apache** ως web server, η **MySQL 8.0.37** (Percona Server) ως βάση δεδομένων, η **PHP 8.1.34** ως γλώσσα προγραμματισμού, και το **Bootstrap** μέσω κοινοτικών θεμάτων (contributed theme) για δυναμική διάταξη. Η επιλογή του Percona Server αντί της τυπικής MySQL έγινε για λόγους συμβατότητας με την υπάρχουσα υποδομή του εργαστηρίου, απλοποιώντας τη διαχείριση.

5.3 Δομή περιεχομένου

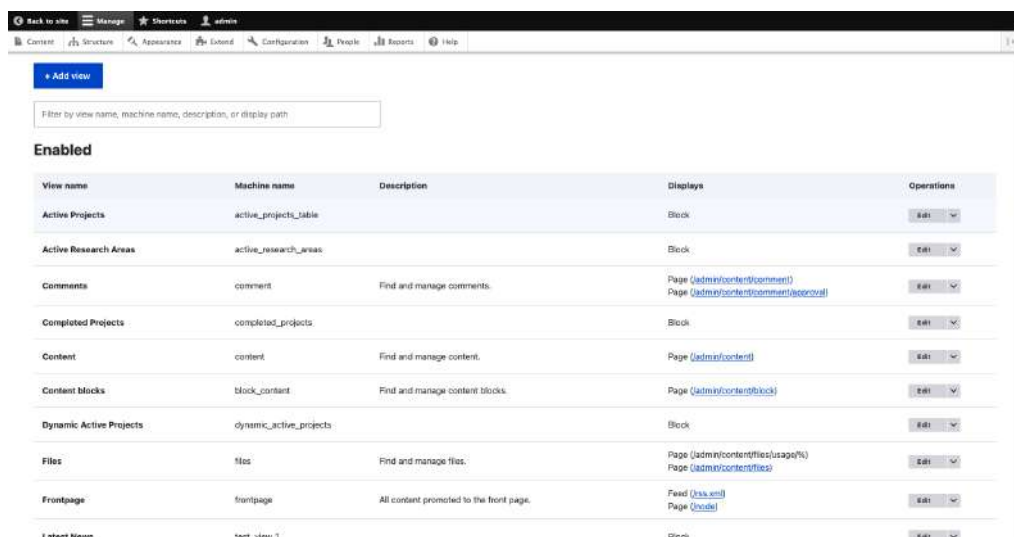
Για τον νέο ιστότοπο σχεδιάστηκαν συνολικά οκτώ προσαρμοσμένα content types, καθένα με τα κατάλληλα fields για τις ανάγκες της. Το **Basic Page** χρησιμοποιείται για στατικές σελίδες γενικής χρήσης. Το **Link** αντιστοιχεί στα αντικείμενα που εμφανίζονται στη σελίδα «Links» του site. Το **Members** καλύπτει τα προφίλ των μελών του εργαστηρίου, με πεδία για όνομα, θέση, email, φωτογραφία και σύντομη βιογραφία. Το **News** χρησιμοποιείται για τα νέα και τις ανακοινώσεις του εργαστηρίου. Το **Partner** αφορά τους συνεργάτες και χρηματοδότες, ενώ το **Project** καλύπτει τα ερευνητικά projects, τόσο ενεργά όσο και ολοκληρωμένα. Το **Publication** χρησιμοποιείται για τις επιστημονικές δημοσιεύσεις και το **Research Areas** για τους τομείς έρευνας του εργαστηρίου.

Συνολικά δημιουργήθηκαν 8 custom content types, καθένα με τα κατάλληλα fields για τις ανάγκες του ιστότοπου το εργαστηρίου. Ως παράδειγμα, το content type **Members** περιέχει τα παρακάτω fields: **Address** (Text plain), **Email**, **Homepage** (Link), **Is Past Member** (Boolean), **Member Affiliation** (Text plain), **Member Image** (Image), **Phone** (Text plain), και **Short CV** (Text plain, long), όπως φαίνεται στην Εικόνα 5.1. Η λίστα των ενεργοποιημένων Views και η λίστα περιεχομένου στο Admin interface φαίνονται αντίστοιχα στις Εικόνες 5.2 και 5.3.



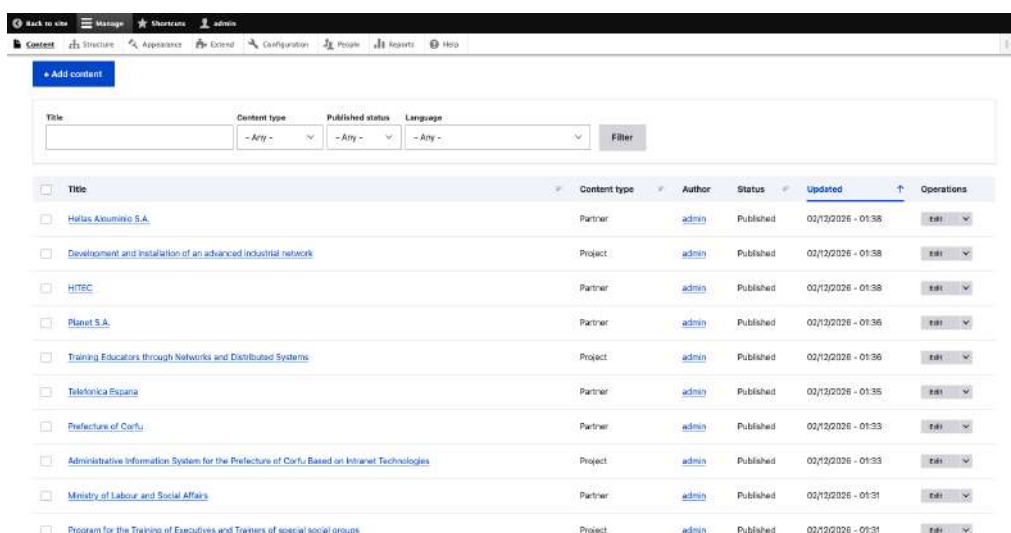
Label	Machine name	Field type	Operations
Address	field_address_1	Text (plain)	Edit
Email	field_email	Email	Edit
Homepage	field_homepage	Link	Edit
Is Past Member	field_is_past_member_bool	Boolean	Edit
Member Affiliation	field_property	Text (plain)	Edit
Member Image	field_member_image	Image	Edit
Phone	field_phone	Text (plain)	Edit
Short CV	field_short_cv	Text (plain, long)	Edit

Εικόνα 5.1: Τα fields του content type Members στο Admin interface



View name	Machine name	Description	Displays	Operations
Active Projects	active_projects_table		Block	Edit
Active Research Areas	active_research_areas		Block	Edit
Comments	comment	Find and manage comments.	Page (admin/content/comment) Page (admin/content/comment/accept)	Edit
Completed Projects	completed_projects		Block	Edit
Content	content	Find and manage content.	Page (admin/content)	Edit
Content blocks	block_content	Find and manage content blocks.	Page (admin/content/block)	Edit
Dynamic Active Projects	dynamic_active_projects		Block	Edit
Files	files	Find and manage files.	Page (admin/content/files/usage/%) Page (admin/content/files)	Edit
Frontpage	frontpage	All content promoted to the front page.	Feed (rss.xml) Page (/node)	Edit
Latest News	test_view_1		Block	Edit

Εικόνα 5.2: Λίστα των ενεργοποιημένων Views στο Admin interface



Title	Content type	Author	Status	Updated	Operations
Hellas Assuminko S.A.	Partner	admin	Published	02/12/2026 - 01:38	Edit
Development and installation of an advanced industrial network	Project	admin	Published	02/12/2026 - 01:38	Edit
HITEC	Partner	admin	Published	02/12/2026 - 01:38	Edit
Planet S.A.	Partner	admin	Published	02/12/2026 - 01:36	Edit
Training Educators through Networks and Distributed Systems	Project	admin	Published	02/12/2026 - 01:36	Edit
Telefonica Espana	Partner	admin	Published	02/12/2026 - 01:35	Edit
Prefecture of Corfu	Partner	admin	Published	02/12/2026 - 01:33	Edit
Administrative Information System for the Prefecture of Corfu Based on Internet Technologies	Project	admin	Published	02/12/2026 - 01:33	Edit
Ministry of Labour and Social Affairs	Partner	admin	Published	02/12/2026 - 01:31	Edit
Program for the Training of Executives and Trainers of special social groups	Project	admin	Published	02/12/2026 - 01:31	Edit

Εικόνα 5.3: Λίστα περιεχομένου στο Admin interface του Drupal 10

5.4 Αισθητικές επιλογές και Theme

Για τη βάση του theme επιλέχθηκε το **Business** template, ένα έτοιμο Drupal 10 starter theme κατάλληλο για επαγγελματικές και ακαδημαϊκές σελίδες. Η επιλογή αυτή επέτρεψε να ξεκινήσουμε από μια σταθερή βάση με σωστή προσαρμοστική (responsive) συμπεριφορά και καλή συμβατότητα με τα modules που χρησιμοποιούμε, χωρίς να χρειαστεί να γράψουμε theme από το μηδέν.

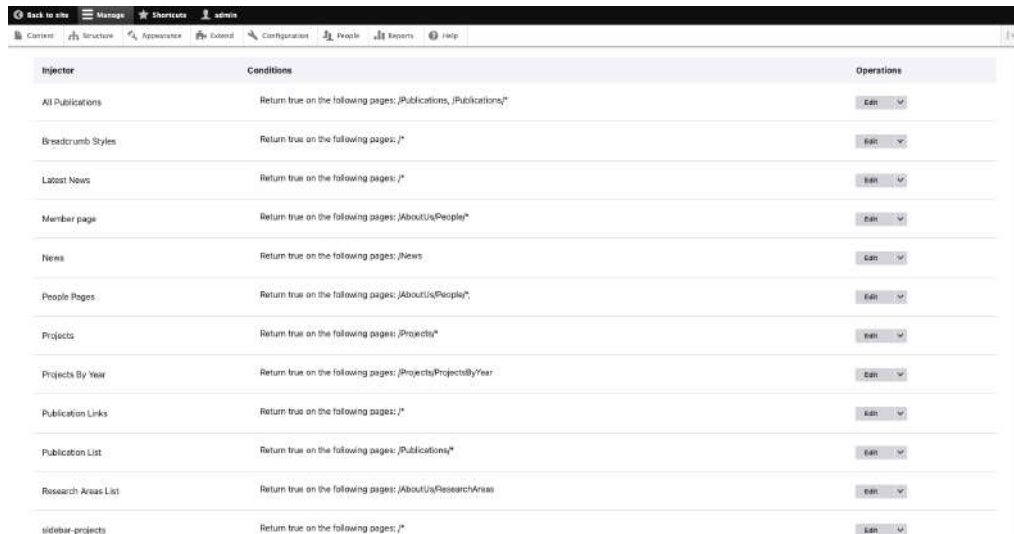
Για τη στυλιστική προσαρμογή της σελίδας χρησιμοποιήθηκε το module **Asset Injector** και συγκεκριμένα το CSS Injector που παρέχει. Στην Εικόνα 5.4 φαίνεται η λίστα των CSS Injectors που δημιουργήθηκαν, και στην Εικόνα 5.5 ένα παράδειγμα CSS κώδικα μέσα σε έναν injector.

5.5 Modules που χρησιμοποιήθηκαν

Εκτός από τα βασικά core modules του Drupal, χρησιμοποιήθηκαν αρκετά contributed modules. Για τη διαχείριση περιεχομένου και navigation: **Pathauto**, **Token**, και **Easy Breadcrumb**. Για το display και το theming: **Asset Injector**, **Display Suite**, και **Views** (core). Η σελίδα διαχείρισης modules (Extend) με τα ενεργοποιημένα modules φαίνεται στην Εικόνα 5.6.

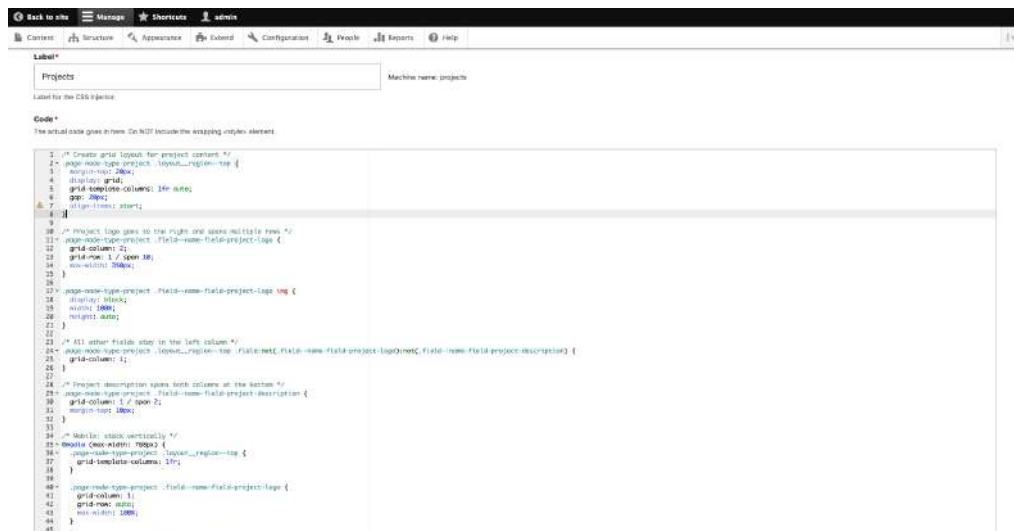
5.6 Προσαρμοστικός σχεδιασμός (Responsive Design)

Ένα από τα βασικά απαιτούμενα χαρακτηριστικά κάθε σύγχρονης ιστοσελίδας είναι η **responsive** συμπεριφορά — η δυνατότητα να προσαρμόζεται αυτόματα σε οθόνες διαφορετικών μεγεθών, από desktop μέχρι κινητά τηλέφωνα [27]. Αυτό είναι ιδιαίτερα σημαντικό καθώς



Injector	Conditions	Operations
All Publications	Return true on the following pages: /Publications, /Publications/*	Edit
Breadcrumb Styles	Return true on the following pages: /*	Edit
Latest News	Return true on the following pages: /*	Edit
Member page	Return true on the following pages: /AboutUs/People/*	Edit
News	Return true on the following pages: /News	Edit
People Pages	Return true on the following pages: /AboutUs/People/*	Edit
Projects	Return true on the following pages: /Projects/*	Edit
Projects By Year	Return true on the following pages: /Projects/ProjectsByYear	Edit
Publication Links	Return true on the following pages: /*	Edit
Publication List	Return true on the following pages: /Publications/*	Edit
Research Areas List	Return true on the following pages: /AboutUs/ResearchAreas	Edit
sidebar-projects	Return true on the following pages: /*	Edit

Εικόνα 5.4: Λίστα CSS Injectors στο Admin interface



Label*

Projects

Machine name: projects

Label for the CSS object:

Code*

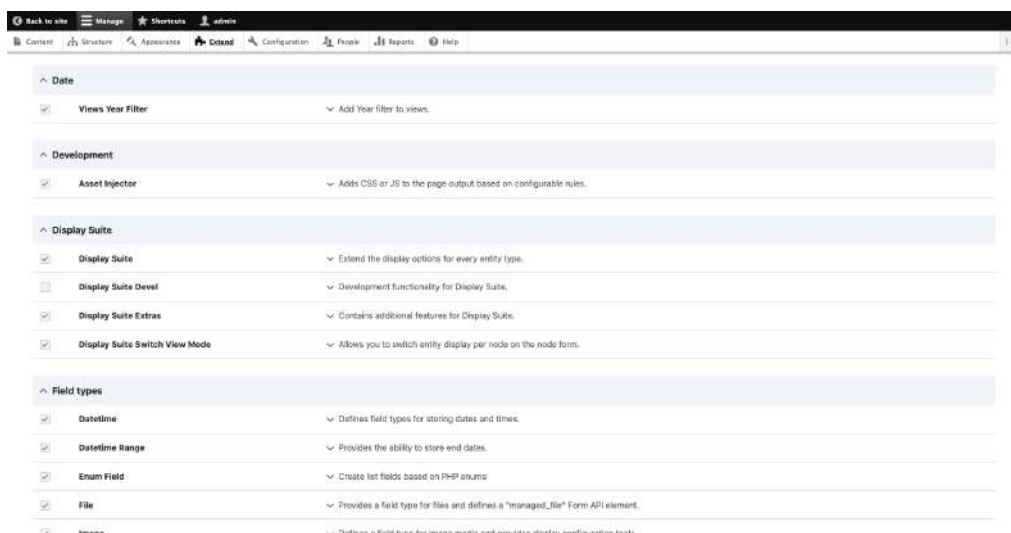
The actual code given to them. Don't forget to include the mapping styles element.

```

1 /* Create grid layout for project content */
2 /* page name type projects - layout...region-top {
3   background: #fff;
4   display: grid;
5   grid-template-columns: 1fr 1fr;
6   gap: 20px;
7   padding: 10px;
8 }
9
10 /* Project logo goes to the right and spans multiple rows */
11 .page-name-type-projects .field-name-field-project-logo {
12   grid-column: 2;
13   grid-row: 1 / span 10;
14   width: 250px;
15 }
16
17 /* page-name-type-projects .field-name-field-project-logo img {
18   display: block;
19   width: 100%;
20   height: auto;
21 }
22
23 /* All other fields stay on the left column */
24 .page-name-type-projects .layout__region-top .field-name-field-project-logo, .field-name-field-project-description {
25   grid-column: 1;
26 }
27
28 /* Project description spans both columns at the bottom */
29 .page-name-type-projects .field-name-field-project-description {
30   grid-column: 1 / span 2;
31   height: 100px;
32 }
33
34 /* sidebar: make vertically */
35 .page-name-type-projects {
36   .page-name-type-projects .layout__region-top {
37     grid-template-columns: 1fr;
38   }
39
40   .page-name-type-projects .field-name-field-project-logo {
41     grid-column: 1;
42     grid-row: auto;
43     width: 100%;
44   }
45
46 }

```

Εικόνα 5.5: Παράδειγμα CSS κώδικα μέσα σε έναν injector



Εικόνα 5.6: Η σελίδα διαχείρισης modules (Extend) στο Admin interface

σήμερα σημαντικό ποσοστό των επισκεπτών μιας ιστοσελίδας προέρχεται από κινητές συσκευές.

5.6.1 Bootstrap και Responsive Grid

Η responsive συμπεριφορά της νέας σελίδας βασίζεται στο **Bootstrap framework**, το οποίο ενσωματώνεται μέσω του Business theme. Το Bootstrap χρησιμοποιεί ένα σύστημα 12 στηλών (grid system) που επιτρέπει τον ορισμό διαφορετικής διάταξης για desktop, tablet και mobile. Στον ιστότοπο μας αυτό φαίνεται στη διάταξη του header και του sidebar: σε desktop το κύριο περιεχόμενο και η πλαϊνή στήλη εμφανίζονται δίπλα-δίπλα, ενώ σε mobile στοιβάζονται κατακόρυφα αυτόματα. Το μενού πλοήγησης (navigation menu) μετατρέπεται επίσης αυτόματα σε hamburger menu σε μικρές οθόνες μέσω του Bootstrap navbar component, χωρίς επιπλέον κώδικα.

5.6.2 Στυλιστικές Προσαρμογές μέσω CSS Asset Injection

Εκτός από τη βασική responsive συμπεριφορά που παρέχει το Bootstrap, χρησιμοποιήθηκε το CSS Asset Injection για γενικές στυλιστικές προσαρμογές — χρώματα, αποστάσεις, γραμματοσειρές, διάταξη στοιχείων. Οι προσαρμογές αυτές εφαρμόζονται ενιαία σε όλες τις συσκευές χωρίς media queries (κανόνες CSS που εφαρμόζουν διαφορετικά στυλ ανάλογα με το μέγεθος της οθόνης), αφού η προσαρμογή στο μέγεθος οθόνης αναλαμβάνεται αποκλειστικά από το Bootstrap grid. Η απόφαση αυτή απλοποιεί τη συντήρηση του CSS — υπάρχει ένα ενιαίο σύνολο κανόνων αντί για παράλληλα breakpoints.

5.6.3 Responsive Image Module

Το Drupal 10 παρέχει το **Responsive Image module** ως μέρος του core [6]. Η προσέγγιση είναι πιο χειροκίνητη αλλά δίνει πλήρη έλεγχο: ο developer ορίζει τα breakpoints σε ένα αρχείο `.breakpoints.yml` στο theme:

```

1 mytheme.mobile:
2   label: mobile
3   mediaQuery: '(max-width: 767px)'
4   weight: 0
5   multipliers:
6     - 1x
7     - 2x
8 mytheme.desktop:
9   label: desktop
10  mediaQuery: '(min-width: 1024px)'
11  weight: 2
12  multipliers:
13    - 1x

```

Στη συνέχεια δημιουργούνται **Responsive Image Styles** που αντιστοιχούν κάθε breakpoint με ένα image style (π.χ. μικρό crop για mobile, μεγάλο για desktop). Το αποτέλεσμα είναι HTML5 `<picture>` element με πλήρη art direction [25]:

```

1 <picture>
2   <source srcset="image-small.jpg"
3     media="(max-width: 767px)">
4   <source srcset="image-large.jpg"
5     media="(min-width: 1024px)">
6   
7 </picture>

```

Στη δική μας υλοποίηση ενεργοποιήσαμε το Responsive Image module και ρυθμίσαμε τα δύο έτοιμα styles που παρέχει το core: **Narrow** για εικόνες προφίλ μελών και λογότυπα συνεργατών, και **Wide** για εικόνες projects και νέων. Αντιστοιχίσαμε τα styles αυτά στα image fields των αντίστοιχων content types μέσω του Manage Display, με αποτέλεσμα οι εικόνες να σερβίρονται πλέον στο κατάλληλο μέγεθος ανά συσκευή. Αυτή η αλλαγή αναμένεται να βελτιώσει σημαντικά το LCP σε mobile συσκευές, που αποτελούσε το κύριο αδύνατο σημείο στις μετρήσεις απόδοσης [28, 29].

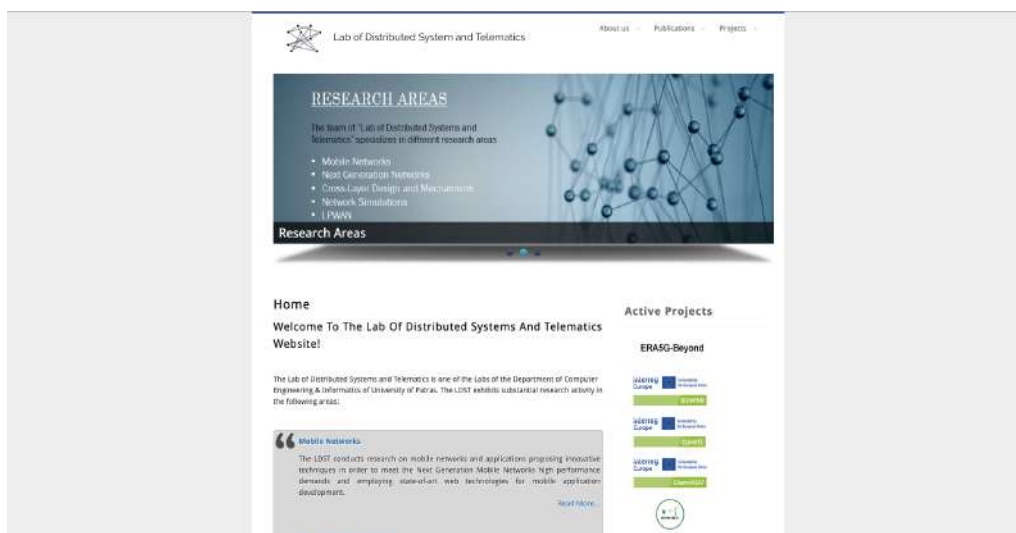
5.7 Σύγκριση Drupal 7 και Drupal 10

Στην ενότητα αυτή παρουσιάζεται οπτική σύγκριση μεταξύ της παλιάς σελίδας σε D7 και της νέας σε D10 για τις κύριες ενότητες του site. Έχει γίνει προσπάθεια διατήρησης της αισθητικής της Drupal 7 version για χάρη διευκόλυνσης των χρηστών. Παρ' όλα αυτά, έχει γίνει προσπάθεια βελτίωσης και εκσυγχρονισμού της αισθητικής.

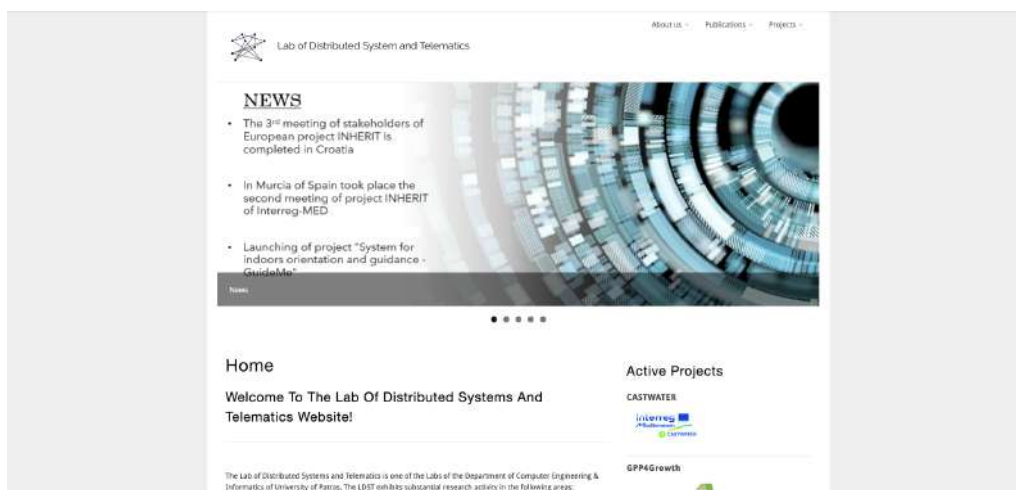
Κατά τη σχεδίαση του νέου ιστότοπου, η βασική φιλοσοφία ήταν η απλοποίηση και ο εκμοντερνισμός της εμφάνισης σε σχέση με την παλιά έκδοση. Η παλιά σελίδα χαρακτηριζόταν από

πυκνή διάταξη, με πολλά πλαίσια και πίνακες που γέμιζαν τον διαθέσιμο χώρο και δυσκόλευαν την ανάγνωση. Στη νέα έκδοση επιλέξαμε λιτότερη προσέγγιση: καλύτερες αποστάσεις μεταξύ των στοιχείων, μειωμένος αριθμός πλαισίων, και γενικά πιο «αέρινη» παρουσίαση που κάνει το περιεχόμενο πιο εύκολα σαρώσιμο από τον επισκέπτη. Οι αλλαγές αυτές ακολουθούν τα σύγχρονα πρότυπα σχεδιασμού ιστοσελίδων, όπου η απλότητα και η ευανάγνωστη διάταξη προτιμώνται έναντι της πυκνότητας πληροφορίας.

5.7.1 Αρχική Σελίδα



Εικόνα 5.7: Αρχική σελίδα — Drupal 7

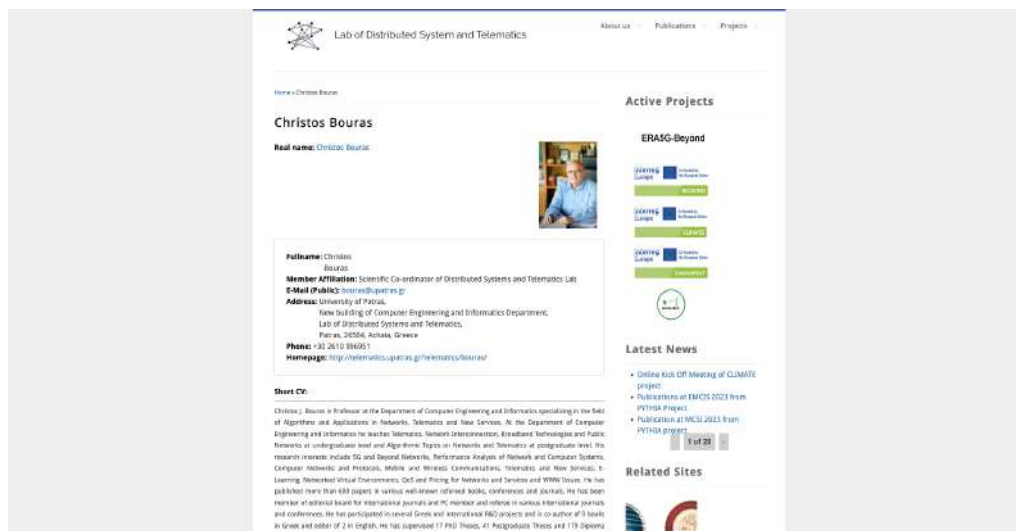


Εικόνα 5.8: Αρχική σελίδα — Drupal 10

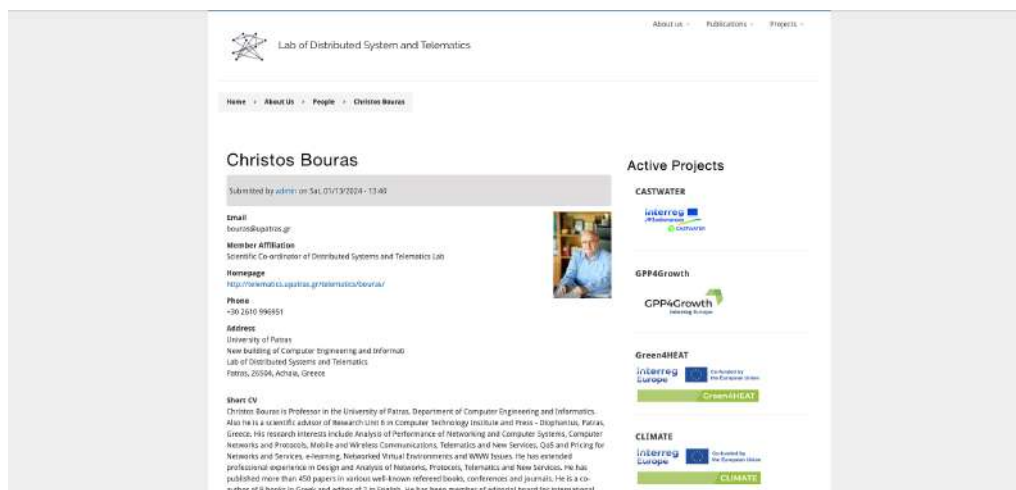
Η αρχική σελίδα διατηρήθηκε σε μεγάλο βαθμό ίδια, όπως φαίνεται στις Εικόνες 5.7 και 5.8. Οι κύριες δομικές ενότητες — ο slider ειδήσεων, η περιγραφή των ερευνητικών περιοχών και

η λίστα ενεργών projects — παραμένουν στις ίδιες θέσεις. Η πιο αισθητή αλλαγή είναι η αναδιοργάνωση του slider: στη νέα έκδοση τα νέα εμφανίζονται με πιο αραιό layout και σαφέστερη ιεράρχηση πληροφορίας.

5.7.2 Σελίδα Μέλους



Εικόνα 5.9: Σελίδα μέλους — Drupal 7

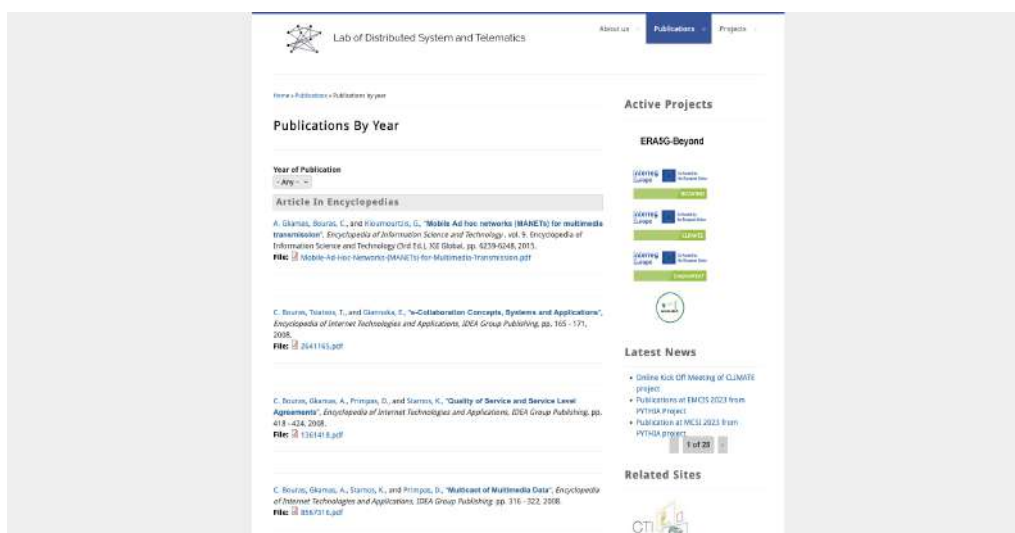


Εικόνα 5.10: Σελίδα μέλους — Drupal 10

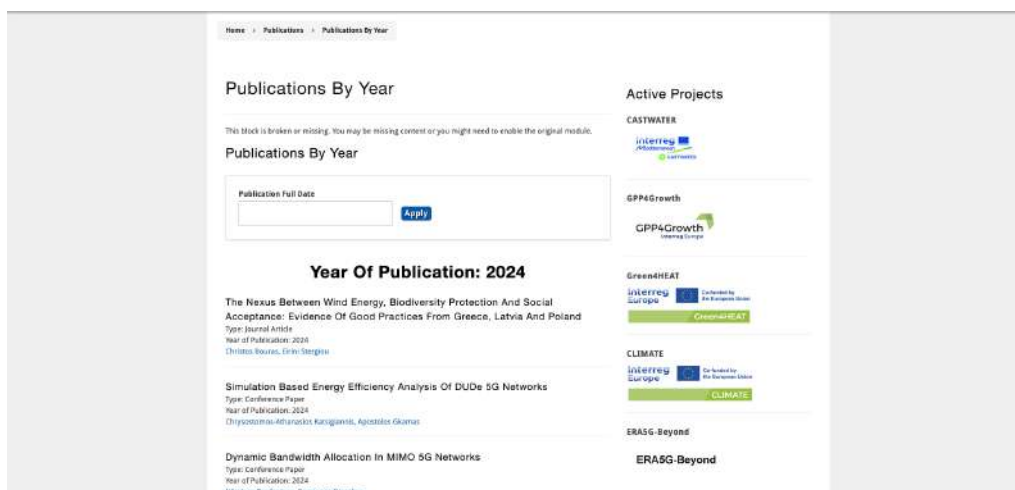
Η σελίδα μέλους παρέμεινε δομικά ίδια ως προς τα βασικά στοιχεία επικοινωνίας και το Short CV. Η πιο σημαντική διαφορά αφορά την παρουσίαση των δημοσιεύσεων του μέλους: στην παλιά έκδοση (Εικόνα 5.9) κάθε δημοσίευση εμφανίζεται με πλήρη βιβλιογραφικά στοιχεία inline (συγγραφείς, περιοδικό, σελίδες, σύνδεσμος PDF), δημιουργώντας πολύ πυκνή λίστα.

Στη νέα έκδοση (Εικόνα 5.10) εμφανίζονται μόνο οι τίτλοι ομαδοποιημένοι ανά κατηγορία, με τις λεπτομέρειες κάθε δημοσίευσης να είναι προσβάσιμες στη σελίδα της. Αυτό κάνει τη σελίδα μέλους πολύ πιο εύκολα σαρώσιμη, ιδιαίτερα για μέλη με μεγάλο αριθμό δημοσιεύσεων.

5.7.3 Σελίδα Δημοσιεύσεων



Εικόνα 5.11: Σελίδα δημοσιεύσεων — Drupal 7

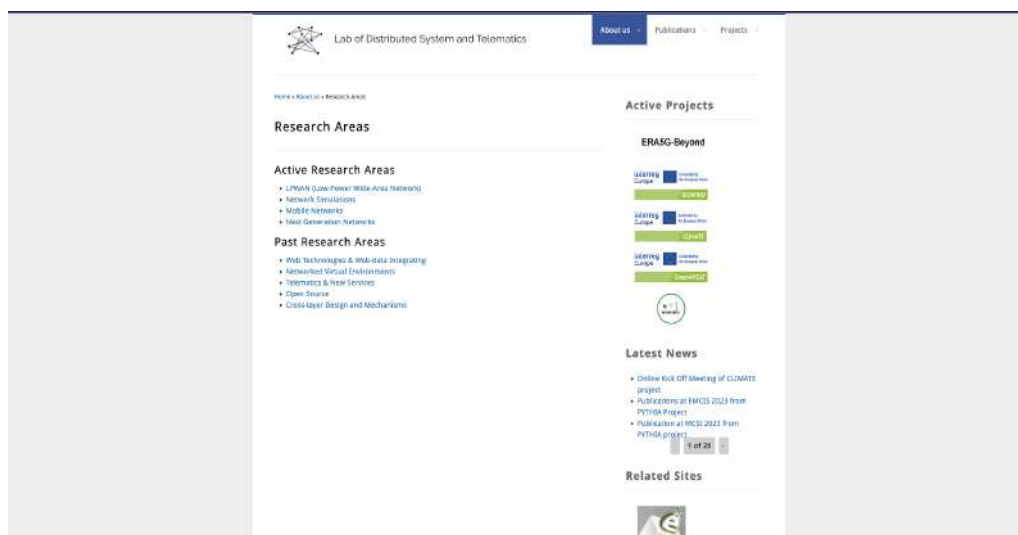


Εικόνα 5.12: Σελίδα δημοσιεύσεων — Drupal 10

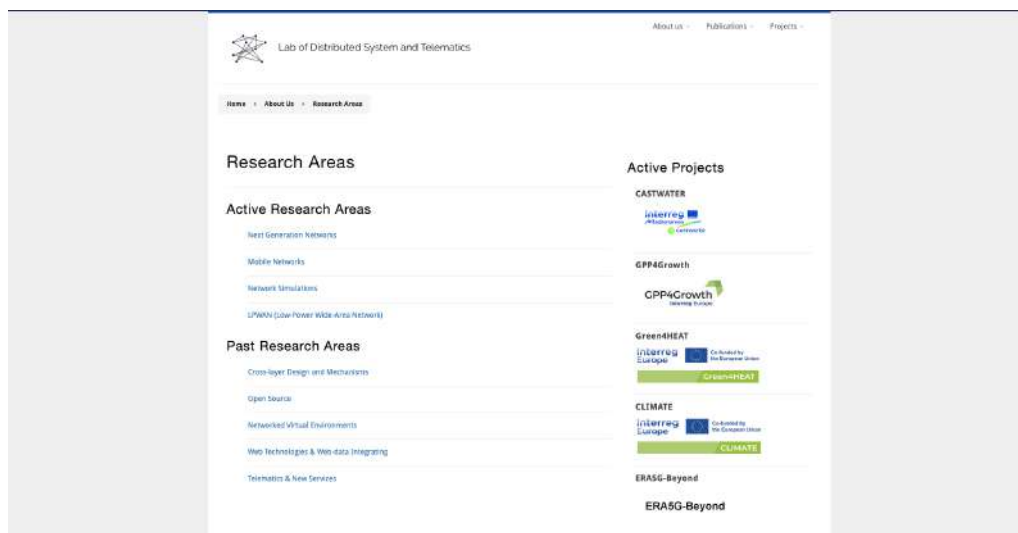
Η ενότητα δημοσιεύσεων αποτελείται από τρεις σελίδες: λίστα ανά κατηγορία, ανά έτος και ανά συγγραφέα. Και οι τρεις έχουν την ίδια λογική παρουσίασης, οπότε οι διαφορές που περιγράφονται εδώ αφορούν και τις τρεις. Η βασική διαφορά είναι ότι στην παλιά έκδοση (Εικόνα 5.11) κάθε εγγραφή εμφανίζει πλήρη βιβλιογραφικά στοιχεία inline (τύπος, συγγραφείς,

στοιχεία περιοδικού, σύνδεσμος PDF), δημιουργώντας πυκνή και δύσκολα σαρώσιμη λίστα. Στη νέα έκδοση (Εικόνα 5.12) κάθε εγγραφή εμφανίζει μόνο τίτλο, τύπο, έτος και συγγραφείς, με τις λεπτομέρειες να είναι προσβάσιμες στη σελίδα της κάθε δημοσίευσης. Αισθητικά, οι επικεφαλίδες ομαδοποίησης εμφανίζονται πλέον ως σαφώς οριοθετημένες ενότητες με κεντραρισμένο τίτλο, αντί για απλό διαχωριστικό κείμενο που χρησιμοποιούσε η παλιά έκδοση.

5.7.4 Σελίδα Ερευνητικών Περιοχών



Εικόνα 5.13: Σελίδα ερευνητικών περιοχών — Drupal 7

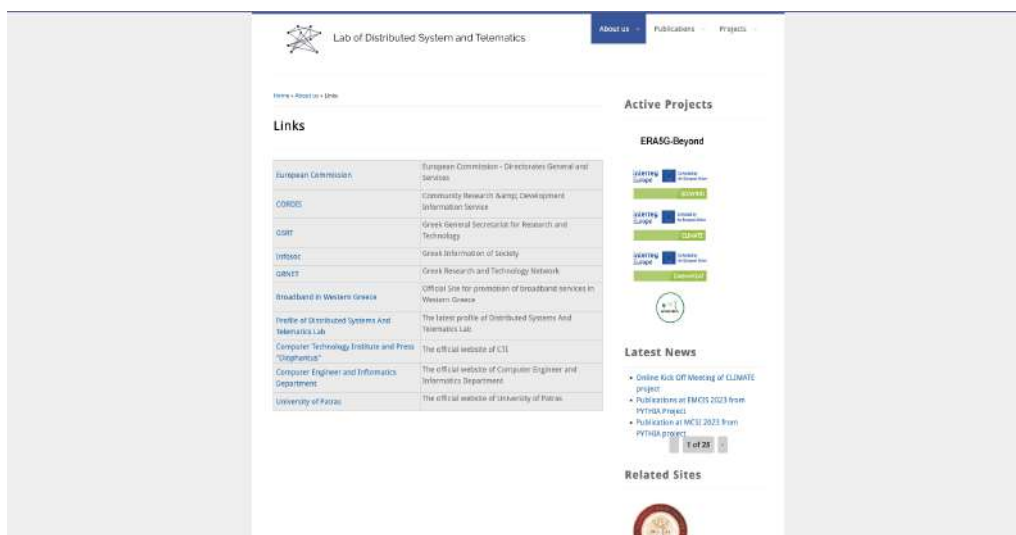


Εικόνα 5.14: Σελίδα ερευνητικών περιοχών — Drupal 10

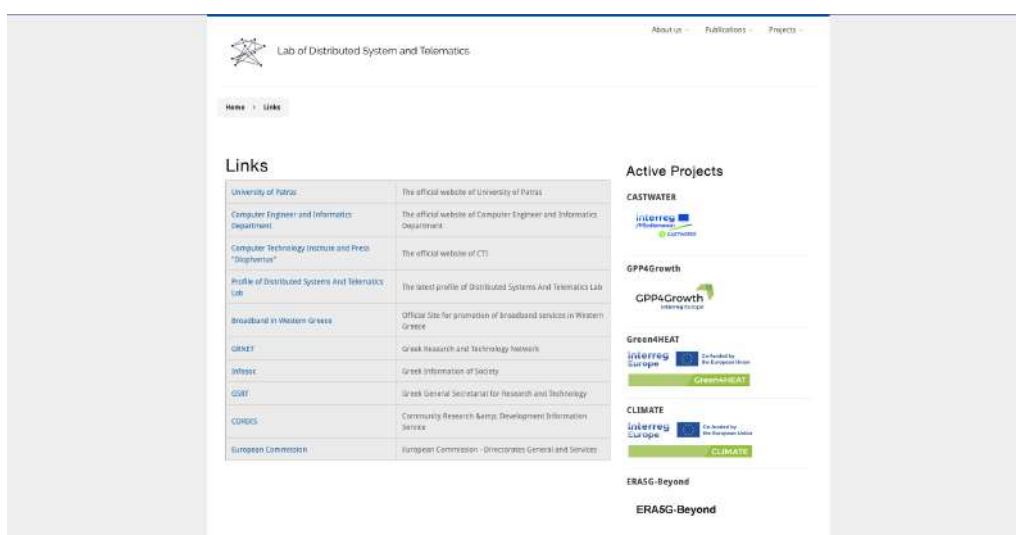
Η σελίδα ερευνητικών περιοχών διατήρησε τον ίδιο διαχωρισμό σε ενεργές και παρελθοντι-

κές περιοχές. Αισθητικά, στην παλιά έκδοση (Εικόνα 5.13) οι τίτλοι εμφανίζονταν ως απλές κουκκίδες με μικρό μέγεθος γραμματοσειράς, ενώ στη νέα (Εικόνα 5.14) κάθε τίτλος αποκτά πιο εμφανές styling με μεγαλύτερο κείμενο και χρωματική διαφοροποίηση, κάνοντας τη λίστα πιο ευδιάκριτη.

5.7.5 Σελίδα Links



Εικόνα 5.15: Σελίδα links — Drupal 7

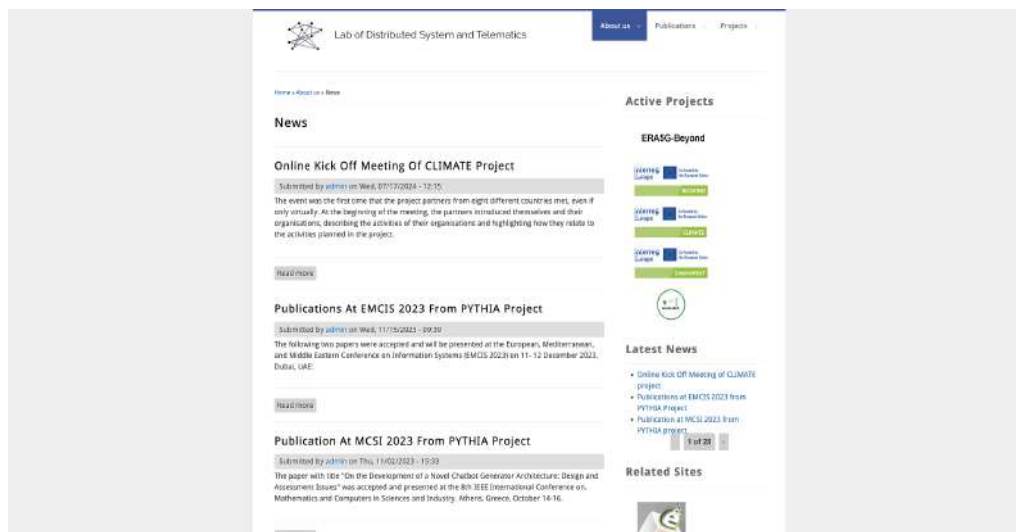


Εικόνα 5.16: Σελίδα links — Drupal 10

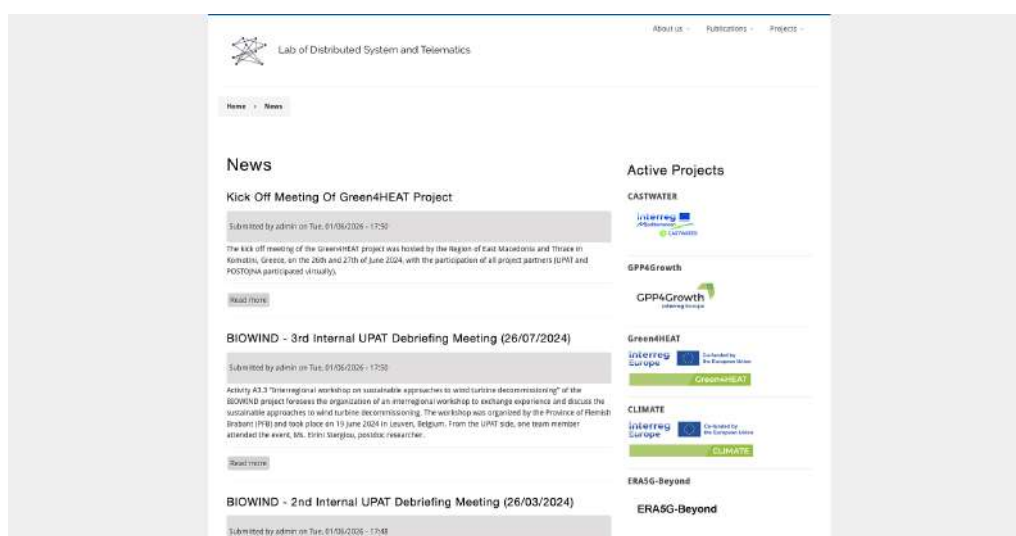
Η σελίδα Links μεταφέρθηκε χωρίς αλλαγές, με τον ίδιο πίνακα δύο στηλών να διατηρείται αναλλοίωτος, όπως φαίνεται στα Εικόνες 5.15 και 5.16. Αυτή ήταν ενσυνείδητη επιλογή,

καθώς η δομή του πίνακα εξυπηρετεί απόλυτα τον σκοπό του ιστότοπου: γρήγορη σάρωση και εύρεση συνδέσμων με σύντομη περιγραφή, χωρίς περιττά οπτικά στοιχεία.

5.7.6 Σελίδα Νέων



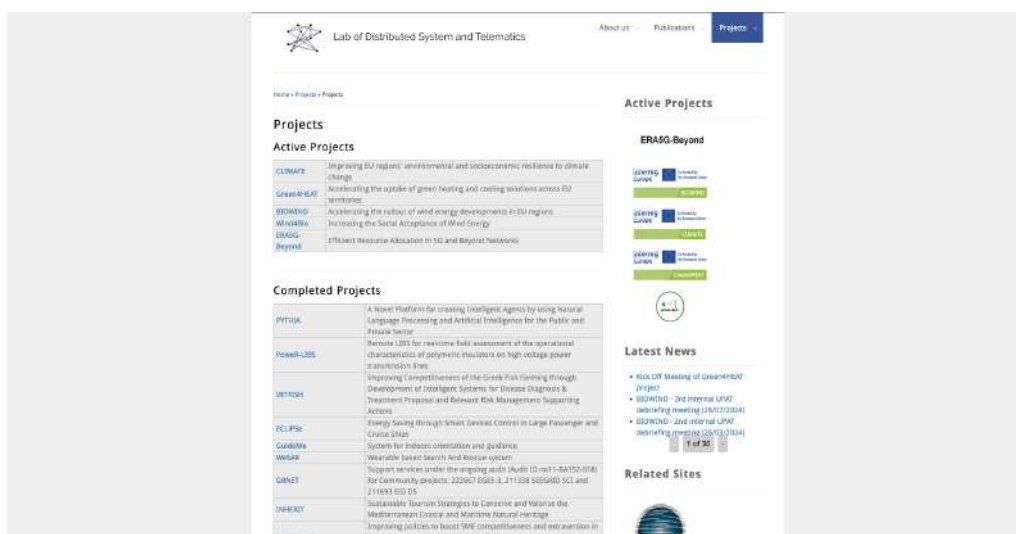
Εικόνα 5.17: Σελίδα νέων — Drupal 7



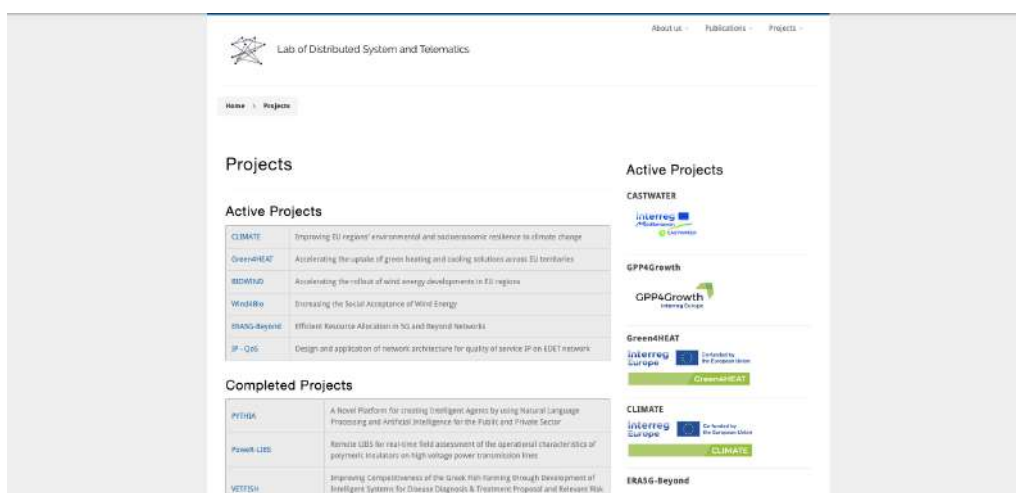
Εικόνα 5.18: Σελίδα νέων — Drupal 10

Η σελίδα νέων διατήρησε την ίδια δομή, με χρονολογική λίστα ανακοινώσεων, όπως φαίνεται στις Εικόνες 5.17 και 5.18. Η κύρια αλλαγή αφορά τον τρόπο εμφάνισης κάθε εγγραφής: στη νέα έκδοση ο τίτλος, η ημερομηνία και το απόσπασμα εμφανίζονται με σαφέστερη οπτική ιεράρχηση, ενώ το «Read more» διατηρείται ως σύνδεσμος για την πλήρη ανάγνωση κάθε νέου.

5.7.7 Σελίδα Projects



Εικόνα 5.19: Σελίδα projects — Drupal 7



Εικόνα 5.20: Σελίδα projects — Drupal 10

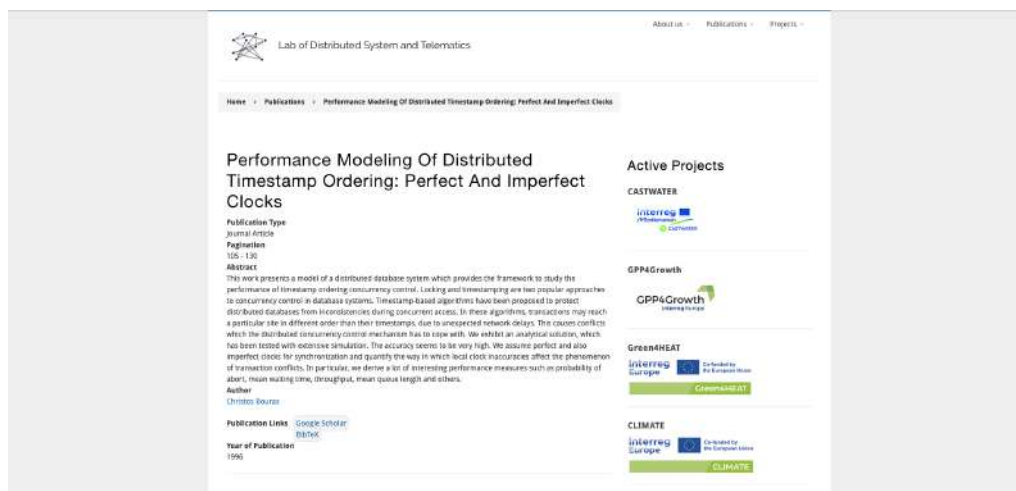
Η ενότητα Projects αποτελείται από δύο σελίδες: την απλή λίστα και τη λίστα με φίλτρο επιλογής έτους. Και οι δύο διατηρούν τον ίδιο διαχωρισμό ενεργών και ολοκληρωμένων projects. Η μόνη αισθητική διαφορά, όπως φαίνεται στα Εικόνες 5.19 και 5.20, είναι οι βελτιωμένες αποστάσεις στη νέα έκδοση.

5.7.8 Σελίδα παραδείγματος δημοσίευσης

Η σελίδα μεμονωμένης δημοσίευσης είναι από τις πιο αλλαγμένες αισθητικά. Στην παλιά έκδοση (Εικόνα 5.21) εμφανίζονται όλα τα πεδία με πυκνή διάταξη, ενώ στη νέα (Εικόνα 5.22)



Εικόνα 5.21: Σελίδα παραδείγματος δημοσίευσης — Drupal 7



Εικόνα 5.22: Σελίδα παραδείγματος δημοσίευσης — Drupal 10

η παρουσίαση είναι πιο αραιή και οργανωμένη, με σαφή διαχωρισμό μεταξύ των επιμέρους πληροφοριών.

5.7.9 Σελίδα παραδείγματος προγράμματος



Εικόνα 5.23: Σελίδα παραδείγματος προγράμματος — Drupal 7



Εικόνα 5.24: Σελίδα παραδείγματος προγράμματος — Drupal 10

Η σελίδα μεμονωμένου project διατήρησε το ίδιο σύνολο πεδίων. Η κυριότερη δομική διαφορά, όπως φαίνεται στα Εικόνες 5.23 και 5.24, αφορά την τοποθέτηση των λογοτύπων χρηματοδότησης: στην παλιά έκδοση εμφανίζονταν στο κέντρο της σελίδας ανάμεσα στα πεδία, ενώ στη νέα τοποθετούνται δίπλα στον τίτλο στην κορυφή, δίνοντας άμεση οπτική αναγνώριση του χρηματοδότη.

Κεφάλαιο 6

Αξιολόγηση και σύγκριση των δύο ιστότοπων

Στο κεφάλαιο αυτό παρουσιάζεται η συστηματική αξιολόγηση και σύγκριση του νέου ιστότοπου του εργαστηρίου σε Drupal 10 (<https://telematics1.upatras.gr/>) με τον υπάρχοντα ιστότοπο σε WordPress (<https://telematics.upatras.gr/>), η οποία βρίσκεται σε λειτουργία κατά τη συγγραφή της παρούσας εργασίας. Η σύγκριση αυτή αποτελεί το κεντρικό αξιολογητικό μέρος της εργασίας και εξετάζει τρεις βασικές διαστάσεις: την απόδοση φόρτωσης, την προσβασιμότητα και το SEO, και την ασφάλεια σε επίπεδο HTTP headers.

6.1 Μεθοδολογία αξιολόγησης

Για την αξιολόγηση χρησιμοποιήθηκαν τρία καθιερωμένα εργαλεία μέτρησης απόδοσης ιστότοπων: το Google PageSpeed Insights, το GTmetrix και το SecurityHeaders.com. Όλες οι μετρήσεις πραγματοποιήθηκαν την ίδια ημέρα (4 Μαΐου 2026) ώστε να διασφαλιστεί η συγκρισιμότητα των αποτελεσμάτων. Κάθε μέτρηση επαναλήφθηκε πολλαπλές φορές και τα αποτελέσματα που παρουσιάζονται αντιστοιχούν στον μέσο όρο των μετρήσεων αυτών, με σκοπό την εξάλειψη τυχαίων αποκλίσεων που μπορεί να οφείλονται σε φορτίο δικτύου ή server εκείνη τη στιγμή.

Σημειώνεται ότι οι δύο ιστότοποι φιλοξενούνται στον ίδιο server του Πανεπιστημίου Πατρών, γεγονός που αποτελεί σημαντικό μεθοδολογικό πλεονέκτημα: οι διαφορές που παρατηρούνται αντικατοπτρίζουν αποκλειστικά τις επιλογές του εκάστοτε CMS και όχι διαφορές στην υποδομή ή τη γεωγραφική θέση του server.

6.2 Εργαλεία Μέτρησης και Μετρικές

6.2.1 Google PageSpeed Insights

Το Google PageSpeed Insights (PSI) είναι εργαλείο της Google που αναλύει τη φόρτωση μιας ιστοσελίδας χρησιμοποιώντας το open-source εργαλείο Lighthouse [30]. Παράγει αποτε-

λέσματα τόσο για desktop όσο και για mobile περιβάλλον, καλύπτοντας έτσι το σύνολο των χρηστών. Αξιολογεί τέσσερις κατηγορίες, η κάθε μία με βαθμολογία από 0 έως 100.

Η κατηγορία **Performance** εκτιμά την ταχύτητα φόρτωσης της σελίδας και αποτελεί σύνθετη βαθμολογία που βασίζεται σε επιμέρους μετρικές. Η κατηγορία **Accessibility** αξιολογεί κατά πόσο η σελίδα είναι προσβάσιμη σε χρήστες με αναπηρίες, ελέγχοντας στοιχεία όπως κείμενα alt σε εικόνες, αντίθεση χρωμάτων και δομή επικεφαλίδων. Η κατηγορία **Best Practices** ελέγχει εάν η σελίδα ακολουθεί σύγχρονες τεχνικές πρακτικές ανάπτυξης, όπως χρήση HTTPS και ενημερωμένες βιβλιοθήκες. Τέλος, η κατηγορία **SEO** (Search Engine Optimization) αξιολογεί κατά πόσο η σελίδα είναι βελτιστοποιημένη για τις μηχανές αναζήτησης.

Οι βασικές μετρικές απόδοσης που αναφέρονται στα αποτελέσματα είναι οι εξής. Το **Largest Contentful Paint (LCP)** μετράει πόσος χρόνος απαιτείται μέχρι να φορτωθεί το μεγαλύτερο ορατό στοιχείο της σελίδας (συνήθως μια μεγάλη εικόνα ή ένα μεγάλο κομμάτι κειμένου). Αποτελεί το κύριο μέτρο της αντιληπτής ταχύτητας φόρτωσης από τον χρήστη. Τιμές κάτω από 2.5 δευτερόλεπτα θεωρούνται καλές [29]. Το **Cumulative Layout Shift (CLS)** μετράει πόσο «πηδάει» το περιεχόμενο της σελίδας κατά τη φόρτωσή της. Υψηλές τιμές CLS δυσκολεύουν τον χρήστη, ο οποίος μπορεί να κάνει κλικ σε λάθος στοιχείο επειδή η διάταξη άλλαξε ξαφνικά. Τιμές κάτω από 0.1 θεωρούνται καλές [29]. Το **Speed Index** μετράει πόσο γρήγορα συμπληρώνεται οπτικά η σελίδα κατά τη φόρτωση. Διαφέρει από το LCP στο ότι λαμβάνει υπόψη ολόκληρη τη διαδικασία εμφάνισης και όχι μόνο ένα στοιχείο.

6.2.2 GTmetrix

Το GTmetrix [31] είναι εργαλείο μέτρησης απόδοσης που χρησιμοποιεί επίσης το Lighthouse ως βάση, αλλά προσθέτει επιπλέον μετρικές και εκτελεί τη μέτρηση από πραγματικό server σε συγκεκριμένη γεωγραφική τοποθεσία. Στην παρούσα εργασία οι μετρήσεις εκτελέστηκαν από server στο Λονδίνο. Παράγει συνολική βαθμολογία (Grade από A έως F) καθώς και επιμέρους βαθμολογίες Performance και Structure.

Οι βασικές μετρικές του GTmetrix που χρησιμοποιούνται στη σύγκριση είναι οι εξής. Το **Fully Loaded Time** είναι ο συνολικός χρόνος που απαιτείται για την πλήρη φόρτωση όλων των στοιχείων της σελίδας, συμπεριλαμβανομένων εικόνων, σεναρίων και CSS. Αποτελεί τη σημαντικότερη μετρική για τη συνολική εμπειρία χρήστη. Το **Time to First Byte (TTFB)** μετράει πόσος χρόνος μεσολαβεί μεταξύ της αποστολής ενός αιτήματος από τον browser και της λήψης του πρώτου byte δεδομένων από τον server. Αποτελεί ουσιαστικά μέτρο της ανταπόκρισης του server και δεν σχετίζεται με το μέγεθος ή τη δομή της σελίδας [32]. Ο **αριθμός HTTP αιτημάτων** αντιστοιχεί στο πλήθος των αιτημάτων που εκτελεί ο browser για να φορτώσει τη σελίδα (εικόνες, αρχεία CSS, αρχεία JavaScript κ.λπ.). Μεγαλύτερος αριθμός αιτημάτων σημαίνει μεγαλύτερη πολυπλοκότητα και συνήθως αργότερη φόρτωση. Τέλος, το **Total Blocking Time (TBT)** μετράει πόσος χρόνος κατά τη φόρτωση αναλώνεται σε εργασίες που εμποδίζουν τον browser να ανταποκριθεί στις ενέργειες του χρήστη. Υψηλές τιμές TBT κάνουν τη σελίδα να φαίνεται «παγωμένη» [29].

6.2.3 SecurityHeaders.com

To SecurityHeaders.com [33] ελέγχει τα HTTP security headers που επιστρέφει ο server μαζί με κάθε σελίδα. Τα security headers είναι οδηγίες που στέλνει ο server στον browser και καθορίζουν πώς να χειριστεί συγκεκριμένες πτυχές ασφαλείας [34]. Η βαθμολογία κυμαίνεται από F (χειρότερο) έως A+ (άριστο). Οι βασικότεροι headers που εξετάζονται είναι οι εξής:

Το **Strict-Transport-Security (HSTS)** αναγκάζει τον browser να συνδέεται πάντα μέσω HTTPS, εμποδίζοντας υποβαθμίσεις σε HTTP. Η **Content-Security-Policy (CSP)** καθορίζει από ποιες πηγές επιτρέπεται η φόρτωση περιεχομένου, αποτρέποντας επιθέσεις Cross-Site Scripting (XSS). Το **X-Frame-Options** εμποδίζει την εμφάνιση της σελίδας μέσα σε iframe άλλης ιστοσελίδας — δηλαδή την ενσωμάτωσή της σε ένα στοιχείο HTML που επιτρέπει σε μια ιστοσελίδα να εμφανίζει μια άλλη στο εσωτερικό της. Έτσι προστατεύει από επιθέσεις clickjacking, μια τεχνική κατά την οποία ο επιτιθέμενος «κρύβει» τη νόμιμη σελίδα μέσα σε ένα αόρατο iframe, ώστε ο χρήστης να εκτελεί ανεπιθύμητες ενέργειες νομίζοντας ότι αλληλεπιδρά με κάτι άλλο. Το **X-Content-Type-Options** εμποδίζει τον browser να «μαντεύει» τον τύπο αρχείου αγνοώντας τον δηλωμένο τύπο, αποτρέποντας συγκεκριμένες κατηγορίες επιθέσεων. Η **Referrer-Policy** ελέγχει ποιες πληροφορίες αποστέλλονται στον server προορισμού όταν ο χρήστης ακολουθεί έναν σύνδεσμο. Τέλος, η **Permissions-Policy** καθορίζει ποια χαρακτηριστικά του browser (π.χ. κάμερα, μικρόφωνο, γεωτοποθεσία) επιτρέπεται να χρησιμοποιεί η σελίδα.

6.3 Αποτελέσματα Google PageSpeed Insights

Για να αποκτηθεί μια πιο ολοκληρωμένη εικόνα της απόδοσης, η αξιολόγηση με το Google PageSpeed Insights δεν περιορίστηκε στην αρχική σελίδα αλλά επεκτάθηκε σε **πέντε αντιπροσωπευτικές σελίδες** κάθε ιστότοπου: την αρχική, δύο σελίδες με δυναμικές λίστες περιεχομένου (Publications, News), τη σελίδα ενός project (CLIMATE) και τη σελίδα προφίλ ενός μέλους. Η επιλογή αυτή επιτρέπει τη διάκριση αν τυχόν διαφορές οφείλονται στην αρχιτεκτονική κάθε CMS ή σε τοπικές ιδιαιτερότητες της αρχικής σελίδας.

Οι Πίνακες 6.1 και 6.2 παρουσιάζουν τις βαθμολογίες των τεσσάρων κατηγοριών αξιολόγησης του PageSpeed Insights (Performance, Accessibility, Best Practices, SEO) ανά σελίδα, σε desktop και mobile αντίστοιχα. Κάθε κατηγορία βαθμολογείται από 0 έως 100, με τιμές άνω του 90 να θεωρούνται άριστες και τιμές κάτω του 50 να απαιτούν βελτίωση [29].

Συμπληρωματικά, οι Πίνακες 6.3 και 6.4 παρουσιάζουν τις τιμές των βασικών μετρικών ταχύτητας (Core Web Vitals) για την αρχική σελίδα κάθε ιστότοπου, που χρησιμοποιήθηκε ως λεπτομερές σημείο αναφοράς. Οι τιμές αυτές αποτελούν τους βασικούς δείκτες αξιολόγησης της εμπειρίας χρήστη, με σαφή όρια για το τι θεωρείται καλό, μέτριο ή προβληματικό [29].

Από τα αποτελέσματα προκύπτουν αρκετές ενδιαφέρουσες παρατηρήσεις. Στην **αρχική σελίδα**, οι δύο ιστοσελίδες εμφανίζονται ουσιαστικά ισοδύναμες στο desktop Performance (71 έναντι 72), με οριακή υπεροχή του WordPress. Όταν όμως η μέτρηση επεκτείνεται στις υπόλοιπες σελίδες, η εικόνα αλλάζει ριζικά: το Drupal 10 σημειώνει σταθερά βαθμολογίες

Πίνακας 6.1: Βαθμολογίες Google PageSpeed Insights ανά σελίδα — Desktop

Σελίδα	CMS	Perf	A11y	BP	SEO
Αρχική	Drupal 10	71	75	95	77
Αρχική	WordPress	72	84	100	85
Publications	Drupal 10	96	87	100	85
Publications	WordPress	60	81	100	77
News	Drupal 10	96	86	100	83
News	WordPress	68	89	100	83
Project (CLIMATE)	Drupal 10	97	88	100	85
Project (CLIMATE)	WordPress	74	86	100	85
Προφίλ μέλους	Drupal 10	97	84	100	85
Προφίλ μέλους	WordPress	71	75	100	77

Πίνακας 6.2: Βαθμολογίες Google PageSpeed Insights ανά σελίδα — Mobile

Σελίδα	CMS	Perf	A11y	BP	SEO
Αρχική	Drupal 10	67	79	92	77
Αρχική	WordPress	55	82	96	85
Publications	Drupal 10	78	90	100	85
Publications	WordPress	54	72	96	77
News	Drupal 10	80	90	100	83
News	WordPress	55	89	96	83
Project (CLIMATE)	Drupal 10	83	91	96	85
Project (CLIMATE)	WordPress	56	84	96	85
Προφίλ μέλους	Drupal 10	78	88	100	85
Προφίλ μέλους	WordPress	55	74	96	77

Πίνακας 6.3: Μετρικές ταχύτητας Google PageSpeed Insights (αρχική σελίδα) — Desktop

Μετρική	Drupal 10	WordPress
LCP	2.4s	2.4s
CLS	0.26	0.006
Speed Index	1.4s	3.8s

Πίνακας 6.4: Μετρικές ταχύτητας Google PageSpeed Insights (αρχική σελίδα) — Mobile

Μετρική	Drupal 10	WordPress
LCP	11.7s	12.2s
CLS	0.00	0.00
Speed Index	3.7s	12.9s

96–97 στο desktop Performance σε όλες τις σελίδες πέραν της αρχικής, ενώ το WordPress κυμαίνεται μεταξύ 60–74. Αντίστοιχα στο mobile Performance, το Drupal 10 ανεβαίνει σε 78–83 στις άλλες σελίδες, ενώ το WordPress παραμένει σταθερά γύρω στο 54–56.

Αυτή η διαφοροποίηση δείχνει ότι η αρχική σελίδα του Drupal 10 ευθύνεται για τη χαμηλή γενική εντύπωση του ιστότοπου στο PageSpeed Insights: ο slider νέων και η πολυπλοκότητα του Business theme εισάγουν επιπλέον φόρτο απεικόνισης (rendering work) που δεν εμφανίζεται στις άλλες σελίδες. Στις υπόλοιπες σελίδες, που χτίζονται με το Views module ή είναι μεμονωμένο κόμβο (single-node), το Drupal 10 αξιοποιεί πλήρως την αρχιτεκτονική του και ξεπερνά καθαρά το WordPress. Αντίθετα, το WordPress εμφανίζει συγκρίσιμη — και χαμηλότερη — απόδοση σε όλους τους τύπους σελίδας, καθώς χρησιμοποιεί παντού τον ίδιο μηχανισμό Elementor και plugins.

Στις κινητές συσκευές η συνολική εμπειρία της αρχικής σελίδας παραμένει προβληματική και για τα δύο CMS: το LCP ξεπερνά τα 11 δευτερόλεπτα και στις δύο περιπτώσεις (Πίνακας 6.4), τιμές πολύ πέρα από το αποδεκτό όριο των 2.5 δευτερολέπτων. Αυτό υποδεικνύει ότι ανεξάρτητα από CMS, οι εικόνες της αρχικής σελίδας απαιτούν βελτιστοποίηση για κινητές συσκευές.

Αξιοσημείωτη παραμένει η διαφορά στο CLS (Cumulative Layout Shift) της αρχικής σελίδας: το Drupal 10 εμφανίζει τιμή 0.26, που θεωρείται προβληματική (needs improvement), ενώ το WordPress έχει τιμή 0.006, που θεωρείται άριστη. Το πρόβλημα εντοπίζεται κυρίως στον slider της αρχικής σελίδας και μπορεί να αντιμετωπιστεί με τον ορισμό ρητών διαστάσεων στις εικόνες — ένα ακόμα στοιχείο που επιβεβαιώνει την ιδιαιτερότητα της αρχικής σελίδας του Drupal 10.

Όσον αφορά τις κατηγορίες Accessibility, Best Practices και SEO, η αρχική επισκόπηση με βάση μόνο την αρχική σελίδα είχε αφήσει την εντύπωση γενικής υπεροχής του WordPress, ωστόσο η διευρυμένη ανάλυση διαφοροποιεί την εικόνα. Στο Best Practices το Drupal 10 λαμβάνει συστηματικά 100 σε όλες τις εσωτερικές σελίδες desktop (96–100 σε mobile), συγκρίσιμα με το WordPress. Στο SEO, η διαφορά που είχε παρατηρηθεί στην αρχική σελίδα (77 έναντι 85) εξισορροπείται στις εσωτερικές σελίδες (83–85 και για τα δύο CMS). Στο Accessibility, τα αποτελέσματα είναι μικτά: το Drupal 10 κυρίως υπερέχει σε mobile στις εσωτερικές σελίδες (88–91 έναντι 72–89 του WordPress), ενώ το WordPress εμφανίζει ελαφρώς υψηλότερες τιμές στην αρχική σελίδα.

Σημαντικό στοιχείο που αναδεικνύεται από όλες τις μετρήσεις είναι η **σταθερότητα** του Drupal 10: οι βαθμολογίες Performance desktop σε όλες τις εσωτερικές σελίδες κυμαίνονται

μόλις από 96 έως 97, και στο mobile από 78 έως 83 — εξαιρετικά συμπαγές εύρος. Το WordPress εμφανίζει επίσης σταθερές αλλά χαμηλότερες επιδόσεις mobile (54–56), ενώ στο desktop παρουσιάζει μεγαλύτερη διακύμανση (60–74).

Συμπερασματικά, η διευρυμένη αξιολόγηση αναδεικνύει ότι **το Drupal 10 υπερέρχει σαφώς στο σύνολο των σελίδων**, με τη μόνη εξαίρεση να είναι η αρχική σελίδα, όπου ο slider και η δομή του Business theme περιορίζουν την απόδοση. Αυτό υποδηλώνει ότι τυχόν μελλοντική βελτιστοποίηση της σελίδας θα πρέπει να εστιάσει **στην αρχική σελίδα ειδικά**, αντί για γενικευμένες αλλαγές στο theme.

6.4 Αποτελέσματα GTmetrix

Ο Πίνακας 6.5 παρουσιάζει τα αποτελέσματα του GTmetrix για τους δύο ιστότοπους, σε πέντε αντιπροσωπευτικές σελίδες κάθε υλοποίησης (αρχική, Publications, News, ένα project και ένα προφίλ μέλους). Οι μετρήσεις εκτελέστηκαν από server στο Λονδίνο, καθώς αποτελεί τον πλησιέστερο διαθέσιμο κόμβο του GTmetrix στο γεωγραφικό κοινό του εργαστηρίου και δίνει έτσι πιο αντιπροσωπευτικές τιμές για μετρικές που εξαρτώνται από τη δικτυακή καθυστέρηση (π.χ. TTFB). Και οι δύο ιστότοποι μετρήθηκαν από την ίδια τοποθεσία, ώστε η σύγκριση να είναι δίκαιη. Επιλέχθηκαν επτά βασικές μετρικές που χαρακτηρίζουν τόσο τη συνολική απόδοση όσο και τη συμπεριφορά του CMS σε επίπεδο server και client.

Πίνακας 6.5: Αποτελέσματα GTmetrix ανά σελίδα

Σελίδα	CMS	Grade	Perf	TTFB	LCP	CLS	FL	Req
Αρχική	Drupal 10	B	84%	348ms	1.33s	0.23	1.33s	24
Αρχική	WordPress	C	69%	1.75s	2.75s	0.01	3.15s	56
Publications	Drupal 10	A	99%	339ms	735ms	0.02	917ms	16
Publications	WordPress	C	68%	1.8s	2.9s	0.01	2.9s	45
News	Drupal 10	A	98%	417ms	891ms	0.01	1.0s	16
News	WordPress	D	54%	1.9s	3.2s	0.01	3.4s	41
Project (CLIMATE)	Drupal 10	A	99%	328ms	687ms	0.05	820ms	17
Project (CLIMATE)	WordPress	C	68%	1.7s	2.9s	0	2.9s	46
Προφίλ μέλους	Drupal 10	A	99%	328ms	831ms	0.01	1.1s	17
Προφίλ μέλους	WordPress	C	69%	1.8s	2.8s	0	2.8s	44

Τα αποτελέσματα του GTmetrix επιβεβαιώνουν τη γενική εικόνα του PageSpeed Insights — το Drupal 10 υπερτερεί συστηματικά του WordPress — αλλά αναδεικνύουν ακόμη μεγαλύτερες διαφορές στις μετρικές που αφορούν την απόκριση του server, και κυρίως στο TTFB, δηλαδή τον χρόνο που χρειάζεται ο server για να στείλει το πρώτο byte. Η διαφορά εκεί φτάνει περίπου τις πέντε φορές και αντανακλά το βαρύτερο φορτίο επεξεργασίας του WordPress σε κάθε αίτημα: σε κάθε φόρτωση το WordPress εκτελεί όλα τα ενεργά plugins — κώδικα PHP που τρέχει στον server και πραγματοποιεί ερωτήματα στη βάση δεδομένων, επιβραδύνοντας την απόκρισή του [35]. Στη συγκεκριμένη υλοποίηση, η επιβάρυνση αυτή εντείνεται από το

Elementor, το οποίο παράγει και αποδίδει τη διάταξη κάθε σελίδας κατά την εκτέλεση. Το Drupal 10 λαμβάνει σταθερά **Grade A** σε όλες τις εσωτερικές σελίδες, με Performance Score 98–99%, ενώ η αρχική σελίδα του παραμένει η εξάιρεση με Grade B (84%). Το WordPress κυμαίνεται μεταξύ Grade C (68–69%) και Grade D (54%, στη σελίδα News).

Η πιο αποκαλυπτική μετρική παραμένει το **TTFB**: το Drupal 10 ανταποκρίνεται σε 328–417ms σε όλες τις σελίδες, ενώ το WordPress χρειάζεται 1.7–1.9s — μια διαφορά **περίπου 5 φορές** που παραμένει σταθερή ανεξάρτητα από τον τύπο σελίδας. Αυτό επιβεβαιώνει ότι η καθυστέρηση οφείλεται στην αρχιτεκτονική του CMS και όχι στο συγκεκριμένο περιεχόμενο: το WordPress εκτελεί πολύ περισσότερες λειτουργίες PHP σε κάθε αίτημα, κυρίως λόγω του Elementor page builder και των ενεργοποιημένων plugins. Στο ίδιο συμπέρασμα οδηγεί ο αριθμός HTTP αιτημάτων: το Drupal 10 χρησιμοποιεί σταθερά 16–17 αιτήματα στις εσωτερικές σελίδες έναντι 41–46 του WordPress, αναλογία περίπου 1:2.7 που μένει σταθερή από σελίδα σε σελίδα.

Ιδιαίτερα διαφωτιστική είναι η σύγκριση του **μεγέθους σελίδας** όταν εξετάζεται πέρα από την αρχική. Στην αρχική σελίδα, τα δύο CMS εμφάνιζαν σχεδόν ίδιο συνολικό μέγεθος (2.33MB έναντι 2.35MB), εύρημα που είχε αρχικά οδηγήσει στο συμπέρασμα ότι η διαφορά απόδοσης οφείλεται αποκλειστικά στον τρόπο οργάνωσης των δεδομένων. Οι νέες μετρήσεις όμως δείχνουν ότι αυτή η ισοβαθμία ίσχυε μόνο για την αρχική σελίδα: στις εσωτερικές σελίδες, το Drupal 10 σερβίρει σταθερά 610–682KB, ενώ το WordPress σερβίρει 2.00–2.16MB ανεξαρτήτως περιεχομένου. Η διαφορά είναι **περίπου τριπλάσια** και αντικατοπτρίζει το γεγονός ότι το WordPress φορτώνει τα ίδια assets (Elementor CSS/JS, plugin styles) σε κάθε σελίδα ανεξάρτητα από το αν τα χρειάζεται, ενώ το Drupal 10 φορτώνει επιλεκτικά μόνο όσα αφορούν την κάθε σελίδα μέσω του μηχανισμού asset injection.

Το πρόβλημα **CLS** του Drupal 10 (0.23 στην αρχική σελίδα) αποδεικνύεται **εντελώς εντοπισμένο στην αρχική σελίδα**: στις άλλες σελίδες οι τιμές κυμαίνονται από 0 έως 0.05, εντός του ορίου του «καλού» (0.1). Επομένως η αρχική παρατήρηση ότι το WordPress υπερέχει σε οπτική σταθερότητα ίσχυε μόνο λόγω του slider της αρχικής σελίδας του Drupal 10 και δεν αποτελεί γενικότερο χαρακτηριστικό της υλοποίησης.

Ξεχωριστή αναφορά αξίζει στη **σελίδα News του WordPress**, η οποία λαμβάνει τη χαμηλότερη βαθμολογία ολόκληρου του συνόλου (Grade D, Performance 54%). Το Total Blocking Time φτάνει τα 282ms — τιμή περίπου 10 φορές υψηλότερη από τις άλλες σελίδες του WordPress (1–16ms) — και υποδηλώνει ότι κάποιο plugin εκτελεί εκτεταμένη JavaScript εργασία κατά τη φόρτωση της λίστας νέων, καθυστερώντας την αλληλεπίδραση χρήστη.

Συνολικά, η διευρυμένη ανάλυση του GTmetrix επιβεβαιώνει και ενισχύει τα ευρήματα του PageSpeed Insights: το Drupal 10 υπερέχει συστηματικά σε όλες τις τεχνικές μετρικές απόδοσης, με μοναδική εξάιρεση την αρχική σελίδα όπου ο slider προκαλεί τοπικά προβλήματα CLS και Performance. Στις υπόλοιπες σελίδες, η αρχιτεκτονική διαφορά μεταξύ Views + asset injection (Drupal 10) και Elementor + plugins (WordPress) μεταφράζεται σε **περίπου 5 φορές ταχύτερο TTFB**, **περίπου 3 φορές μικρότερες σελίδες** και **περίπου 2.7 φορές λιγότερα HTTP αιτήματα**.

6.5 Αποτελέσματα SecurityHeaders.com

Ο Πίνακας 6.6 παρουσιάζει τα αποτελέσματα της ανάλυσης HTTP security headers και για τους δύο ιστότοπους. Η βαθμολογία κυμαίνεται από F (χειρότερο) έως A+ (άριστο) και αντικατοπτρίζει την παρουσία ή απουσία κάθε header [34].

Πίνακας 6.6: Σύγκριση HTTP Security Headers

Security Header	Drupal 10	WordPress
Συνολική βαθμολογία	D	D
Strict-Transport-Security	Όχι	Όχι
Content-Security-Policy	Όχι	Όχι
X-Frame-Options	Ναι	Όχι
X-Content-Type-Options	Ναι	Όχι
Referrer-Policy	Όχι	Όχι
Permissions-Policy	Όχι	Μερικώς

Και οι δύο ιστότοποι λαμβάνουν βαθμολογία D, που υποδηλώνει ελλιπή διαμόρφωση security headers. Ωστόσο, υπάρχουν σημαντικές ποιοτικές διαφορές μεταξύ τους. Το Drupal 10 ορίζει σωστά δύο headers (X-Frame-Options: SAMEORIGIN και X-Content-Type-Options: nosniff), τα οποία παρέχονται αυτόματα από τον πυρήνα του Drupal 10 χωρίς επιπλέον διαμόρφωση. Αντίθετα, το WordPress δεν ορίζει κανένα από αυτά τα δύο headers εξ ορισμού. Ένα επιπλέον εύρημα είναι ότι η επικεφαλίδα X-Generator του Drupal 10 αποκαλύπτει ότι η σελίδα τρέχει Drupal 10 (X-Generator: Drupal 10 (<https://www.drupal.org>)), κάτι που θα μπορούσε να αφαιρεθεί για λόγους ασφαλείας ώστε να μην αποκαλύπτεται το χρησιμοποιούμενο CMS σε πιθανούς επιτιθέμενους.

Σημαντική παρατήρηση είναι ότι και οι δύο ιστότοποι φιλοξενούνται στον ίδιο server του Πανεπιστημίου Πατρών, ο οποίος επιτυγχάνει βαθμολογία A στον έλεγχο SSL/TLS του Qualys SSL Labs [36], χρησιμοποιώντας σύγχρονα πρωτόκολλα (TLS 1.2 και TLS 1.3) με RSA 2048-bit πιστοποιητικά. Αυτό σημαίνει ότι η ασφάλεια σε επίπεδο μεταφοράς δεδομένων είναι ισοδύναμη και για τους δύο ιστότοπους, και οι διαφορές που παρατηρούνται στον έλεγχο security headers αντικατοπτρίζουν αποκλειστικά επιλογές εφαρμογής του εκάστοτε CMS.

6.6 Συνολική αξιολόγηση

Συνοψίζοντας τα αποτελέσματα της αξιολόγησης, μπορούμε να καταλήξουμε σε μια ολοκληρωμένη εικόνα της απόδοσης των δύο ιστότοπων.

Στο πεδίο της **ταχύτητας και απόδοσης**, το Drupal 10 υπερέχει σαφώς και με υψηλή συνέπεια σε όλους τους τύπους σελίδας. Στις εσωτερικές σελίδες (Publications, News, projects, προφίλ μελών) λαμβάνει σταθερά Grade A στο GTmetrix με Performance Score 98–99%, ενώ το WordPress κυμαίνεται μεταξύ Grade C και Grade D. Το TTFB είναι περίπου

5 φορές ταχύτερο (328–417ms έναντι 1.7–1.9s), ο χρόνος πλήρους φόρτωσης 3–4 φορές μικρότερος και ο αριθμός HTTP αιτημάτων περίπου τριπλάσια χαμηλότερος. Το συνολικό μέγεθος σελίδας στις εσωτερικές σελίδες είναι επίσης περίπου τριπλάσια μικρότερο στο Drupal 10 (610–682KB έναντι 2.00–2.16MB), διαφορά που δεν παρατηρείται στην αρχική σελίδα όπου τα δύο CMS εμφανίζουν παραπλήσιο μέγεθος λόγω της παρουσίας του slider. Τα πλεονεκτήματα αυτά οφείλονται στην απλούστερη αρχιτεκτονική του Drupal 10 και στον μηχανισμό asset injection, που φορτώνει επιλεκτικά τα assets κάθε σελίδας, σε αντίθεση με το οικοσύστημα plugins και page builder του WordPress [28].

Στο πεδίο της **προσβασιμότητας, βέλτιστων πρακτικών και SEO**, τα δύο CMS εμφανίζονται σε γενικές γραμμές ισορροπημένα. Στο Best Practices και το SEO οι βαθμολογίες συγκλίνουν στις εσωτερικές σελίδες (και τα δύο CMS σημειώνουν 100% στο Best Practices desktop και 83–85% στο SEO). Στο Accessibility τα αποτελέσματα είναι μικτά: το Drupal 10 σημειώνει υψηλότερες τιμές στις εσωτερικές σελίδες mobile, ενώ το WordPress υπερέχει ελαφρώς στην αρχική σελίδα.

Στο πεδίο της **οπτικής σταθερότητας (CLS)**, η αρχική σελίδα του Drupal 10 εμφανίζει προβληματική τιμή (0.23 στο GTmetrix, 0.26 στο PageSpeed Insights desktop), όμως στις υπόλοιπες σελίδες οι τιμές βρίσκονται μέσα στο όριο του «καλού» (0–0.05). Το ζήτημα είναι επομένως **εντοπισμένο στον slider της αρχικής σελίδας** και μπορεί να αντιμετωπιστεί τοπικά με ορισμό ρητών διαστάσεων στις εικόνες του slider.

Στο πεδίο της **ασφάλειας εφαρμογής**, το Drupal 10 υπερέχει ελαφρά χάρη στους δύο security headers που ορίζει αυτόματα ο πυρήνας του. Ωστόσο, η συνολική βαθμολογία D και για τους δύο ιστότοπους υποδηλώνει ότι υπάρχουν περαιτέρω βελτιώσεις που μπορούν να γίνουν σε επίπεδο διαμόρφωσης server.

Συνολικά, το Drupal 10 υπερέχει σχεδόν σε κάθε πτυχή απόδοσης, με τη μόνη συστηματική αδυναμία να εντοπίζεται στην αρχική σελίδα όπου ο slider προκαλεί τοπικά προβλήματα CLS και επιβραδύνει τη φόρτωση. Η μελλοντική βελτιστοποίηση μπορεί επομένως να στοχεύσει αποτελεσματικά στην αρχική σελίδα ειδικά, αντί σε γενικευμένες αλλαγές στο theme.

Επιστημαίνεται ότι η αξιολόγηση που παρουσιάστηκε βασίζεται αποκλειστικά σε αντικειμενικές, αυτοματοποιημένες τεχνικές μετρικές (απόδοση, ασφάλεια, SEO) μέσω τυποποιημένων εργαλείων, και όχι σε δεδομένα από πραγματικούς χρήστες. Η συλλογή ανατροφοδότησης από επισκέπτες (π.χ. μέσω ερωτηματολογίων ικανοποίησης ή δοκιμών ευχρηστίας) δεν εντάσσεται στο πλαίσιο της παρούσας εργασίας, η οποία εστιάζει στην τεχνική σύγκριση των δύο υλοποιήσεων. Επιπλέον, μια τέτοια αξιολόγηση δεν είναι προς το παρόν εφικτή, καθώς ο νέος ιστότοπος σε Drupal 10 δεν έχει ακόμη τεθεί σε πλήρη λειτουργία· η δημόσια προσβάσιμη έκδοση του ιστότοπου του εργαστηρίου παραμένει αυτή σε WordPress. Συνεπώς, δεν υπάρχει ακόμη βάση πραγματικών χρηστών στη νέα υλοποίηση από την οποία θα μπορούσαν να αντληθούν τέτοια δεδομένα. Η αξιολόγηση με βάση τους χρήστες, όπως δοκιμές ευχρηστίας, ερωτηματολόγια ικανοποίησης και παρακολούθηση πραγματικών χρηστών, αποτελεί φυσική επέκταση της εργασίας μετά τη μετάβαση του νέου ιστότοπου σε παραγωγή και προτείνεται ως μελλοντική εργασία.

6.7 Σύγκριση Διαδικασίας Ανάπτυξης: Drupal 10 έναντι WordPress

Πέρα από τις μετρικές απόδοσης, μια εξίσου σημαντική διάσταση σύγκρισης είναι ο τρόπος με τον οποίο κάθε CMS προσεγγίζει την ανάπτυξη και τη διαχείριση περιεχομένου. Οι δύο πλατφόρμες έχουν θεμελιωδώς διαφορετική φιλοσοφία, που αντικατοπτρίζεται τόσο στο Admin interface όσο και στον τρόπο οργάνωσης του περιεχομένου [12].

6.7.1 Δημιουργία Περιεχομένου

Στο WordPress, η δημιουργία περιεχομένου γίνεται κυρίως μέσω του **Gutenberg editor**, ενός block-based επεξεργαστή που επιτρέπει στον χρήστη να συνθέτει σελίδες σύροντας και αποθέτοντας (drag and drop) διάφορα μπλοκ (κείμενο, εικόνες, κουμπιά, στήλες κ.λπ.). Η προσέγγιση αυτή είναι πολύ φιλική προς τον μη τεχνικό χρήστη, καθώς το αποτέλεσμα φαίνεται άμεσα στην οθόνη (What You See Is What You Get — WYSIWYG). Στην υλοποίηση που εξετάζουμε, το WordPress site χρησιμοποιεί επιπλέον το Elementor page builder, που επεκτείνει περαιτέρω αυτές τις δυνατότητες με έτοιμα templates και προχωρημένες επιλογές διάταξης [37].

Στο Drupal 10, η δημιουργία περιεχομένου βασίζεται στην έννοια των **content types** και **fields**. Κάθε content type (π.χ. Publication, Member, Project) έχει συγκεκριμένα πεδία που ορίζει ο administrator, και ο χρήστης συμπληρώνει δομημένες φόρμες. Αυτή η προσέγγιση είναι λιγότερο διαισθητική για αρχάριους, αλλά εξασφαλίζει **συνέπεια** στη δομή του περιεχομένου: είναι αδύνατον να δημιουργηθεί μια δημοσίευση χωρίς τα απαραίτητα πεδία (τίτλος, συγγραφείς, έτος κ.λπ.) [8].

6.7.2 Διαχείριση Εμφάνισης

Η διαχείριση της εμφάνισης αποτελεί ίσως τη μεγαλύτερη διαφορά μεταξύ των δύο συστημάτων. Στο WordPress με Elementor, ο χρήστης σχεδιάζει οπτικά κάθε σελίδα ξεχωριστά — επιλέγει χρώματα, γραμματοσειρές, διατάξεις απευθείας από το interface. Αυτό δίνει μεγάλη ευελιξία αλλά μπορεί να οδηγήσει σε ασυνέπεια: διαφορετικές σελίδες μπορεί να έχουν διαφορετική εμφάνιση αν δεν υπάρχει πειθαρχία από τον editor.

Στο Drupal 10, η εμφάνιση ελέγχεται κεντρικά από το theme και τα CSS αρχεία. Οι αλλαγές που γίνονται στο theme επηρεάζουν αυτόματα όλες τις σελίδες. Στην υλοποίησή μας, χρησιμοποιήσαμε CSS Asset Injection για να προσαρμόσουμε την εμφάνιση χωρίς να τροποποιήσουμε απευθείας τα αρχεία του theme, κάτι που διευκολύνει τις μελλοντικές ενημερώσεις. Το αποτέλεσμα είναι μια **ομοιόμορφη εμφάνιση** σε όλες τις σελίδες, η οποία συντηρείται κεντρικά.

6.7.3 Εμφάνιση Λιστών Περιεχομένου

Ένα από τα πιο χαρακτηριστικά παραδείγματα της διαφοράς φιλοσοφίας είναι ο τρόπος εμφάνισης λιστών περιεχομένου, όπως η λίστα δημοσιεύσεων ή η λίστα μελών. Στο WordPress,

αυτό συνήθως γίνεται είτε με plugin (π.χ. Custom Post Type UI + Advanced Custom Fields) είτε με Elementor widgets. Το αποτέλεσμα είναι λειτουργικό αλλά εξαρτάται από τη σωστή διαμόρφωση πολλών διαφορετικών plugins που πρέπει να συνεργάζονται μεταξύ τους [35].

Στο Drupal 10, η εμφάνιση λιστών γίνεται μέσω του **Views module**, το οποίο αποτελεί μέρος του core. Το Views επιτρέπει τη δημιουργία σύνθετων ερωτημάτων στη βάση δεδομένων με οπτικό τρόπο, χωρίς κώδικα, και την εμφάνιση των αποτελεσμάτων με πλήρη έλεγχο της διάταξης. Για παράδειγμα, η σελίδα δημοσιεύσεων ανά έτος, ανά κατηγορία και ανά συγγραφέα υλοποιήθηκαν με τρία διαφορετικά Views — χωρίς να χρειαστεί κανένα επιπλέον plugin [25].

6.7.4 Responsive Images: Drupal 10 έναντι WordPress

Μια από τις πιο σημαντικές τεχνικές πτυχές του responsive design είναι η διαχείριση των εικόνων — το να παρουσιάζεται η εικόνα στην κατάλληλη ανάλυση ανάλογα με το μέγεθος οθόνης, αποφεύγοντας την περιττή κατανάλωση bandwidth σε κινητές συσκευές. Τα δύο CMS αντιμετωπίζουν αυτό το πρόβλημα με διαφορετική φιλοσοφία [29].

WordPress: Αυτόματα Responsive Images

Το WordPress υποστηρίζει native responsive images από την έκδοση 4.4 [9]. Κάθε εικόνα που ανεβαίνει στη βιβλιοθήκη πολυμέσων παράγει αυτόματα πολλαπλές εκδόσεις διαφορετικών διαστάσεων. Το WordPress στη συνέχεια προσθέτει αυτόματα τα attributes **srcset** και **sizes** στο markup των εικόνων, επιτρέποντας στον browser να επιλέξει την κατάλληλη έκδοση:

```
1 
```

Αυτή η διαδικασία γίνεται αυτόματα χωρίς καμία ρύθμιση, κάτι που την κάνει πολύ εύκολη στη χρήση. Ωστόσο, αν χρειαστεί πλήρης έλεγχος των breakpoints (π.χ. να εμφανίζεται εντελώς διαφορετική εικόνα σε κινητές συσκευές), ο προγραμματιστής πρέπει να παρέμβει μέσω φίλτρων στο `functions.php` [37].

Drupal 10: Χειροκίνητη Ρύθμιση με Πλήρη Έλεγχο

Το Drupal 10 παρέχει το **Responsive Image module** ως μέρος του core [6]. Η προσέγγιση είναι πιο χειροκίνητη αλλά δίνει πλήρη έλεγχο: ο developer ορίζει τα breakpoints σε ένα αρχείο `.breakpoints.yml` στο theme:

```
1 mytheme.mobile:
2   label: mobile
3   mediaQuery: '(max-width: 767px)'
4   weight: 0
5   multipliers:
6     - 1x
```

```
7   - 2x
8 mytheme.desktop:
9   label: desktop
10  mediaQuery: '(min-width: 1024px)'
11  weight: 2
12  multipliers:
13    - 1x
```

Στη συνέχεια δημιουργούνται **Responsive Image Styles** που αντιστοιχούν κάθε break-point με ένα image style (π.χ. μικρό crop για mobile, μεγάλο για desktop). Το αποτέλεσμα είναι HTML5 <picture> element με πλήρη art direction [25]:

```
1 <picture>
2   <source srcset="image-small.jpg"
3     media="(max-width: 767px)">
4   <source srcset="image-large.jpg"
5     media="(min-width: 1024px)">
6   
7 </picture>
```

Σύγκριση και Επιλογές στην Υλοποίησή μας

Συνοψίζοντας, το WordPress προσφέρει responsive images έτοιμα προς χρήση, χωρίς καμία απαίτηση ρύθμισης από τον χρήστη, κατάλληλο για την πλειοψηφία των χρήσεων. Το Drupal 10 απαιτεί αρχική ρύθμιση αλλά δίνει **πλήρη έλεγχο** ανά breakpoint και ανά image field — σημαντικό για sites με σύνθετες εικαστικές απαιτήσεις.

Στη δική μας υλοποίηση ενεργοποιήσαμε το Responsive Image module και ρυθμίσαμε τα δύο έτοιμα styles που παρέχει το core: **Narrow** για εικόνες προφίλ μελών και λογότυπα συνεργατών, και **Wide** για εικόνες projects και νέων. Αντιστοιχίσαμε τα styles αυτά στα image fields των αντίστοιχων content types μέσω του Manage Display, με αποτέλεσμα οι εικόνες να σερβίρονται πλέον στο κατάλληλο μέγεθος ανά συσκευή. Αυτή η αλλαγή αναμένεται να βελτιώσει σημαντικά το LCP σε mobile συσκευές, που αποτελούσε το κύριο αδύνατο σημείο στις μετρήσεις απόδοσης [28, 29].

Η mobile απόδοση αποτελεί σημείο που χρειάζεται βελτίωση και στις δύο υλοποιήσεις. Το Largest Contentful Paint (LCP) ξεπερνά τα 11 δευτερόλεπτα και στις δύο σελίδες, τιμή που βρίσκεται πολύ πάνω από το αποδεκτό όριο των 2.5 δευτερολέπτων [29]. Το κύριο αίτιο είναι οι εικόνες που δεν έχουν βελτιστοποιηθεί για mobile: δεν χρησιμοποιούνται προσαρμοστικές εικόνες μέσω του χαρακτηριστικού srcset, ούτε καθυστερημένη φόρτωση (lazy loading) για τις εικόνες εκτός της ορατής περιοχής της οθόνης. Αυτή η βελτίωση συγκαταλέγεται στις προτεραιότητες μελλοντικής εργασίας και αναμένεται να βελτιώσει σημαντικά τη βαθμολογία mobile Performance [28].

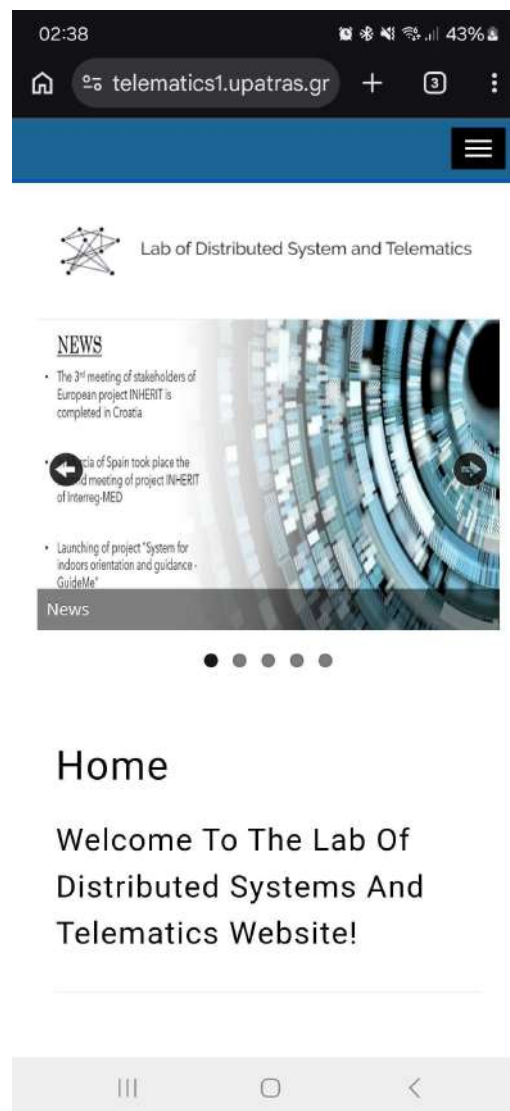
6.7.5 Σύγκριση Mobile Εμφάνισης

Στις παρακάτω εικόνες παρουσιάζεται η εμφάνιση των κύριων σελίδων σε κινητή συσκευή για τις δύο υλοποιήσεις. Οι εικόνες αριστερά αντιστοιχούν στο WordPress (telematics.upatras.gr) και οι εικόνες δεξιά στο Drupal 10 (telematics1.upatras.gr).

Αρχική Σελίδα



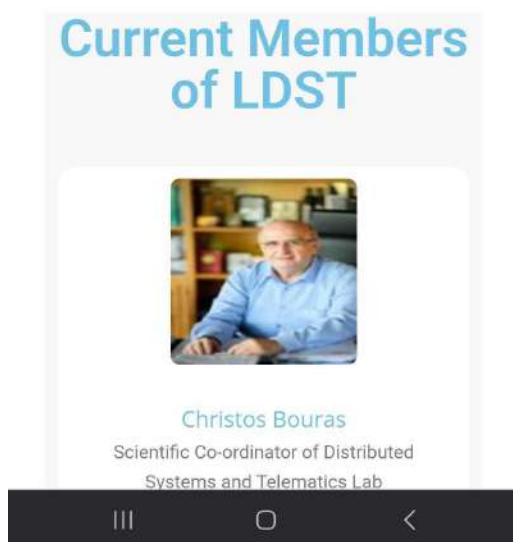
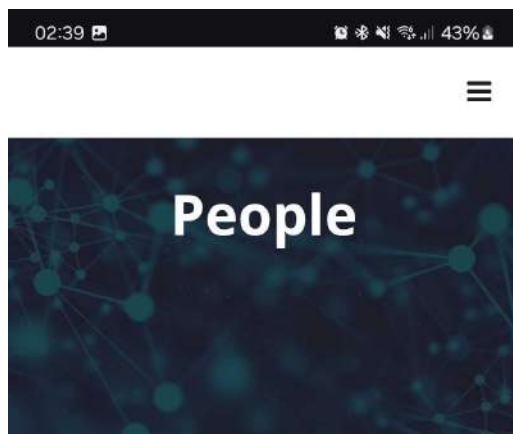
Εικόνα 6.1: Αρχική σελίδα — WordPress (mobile)



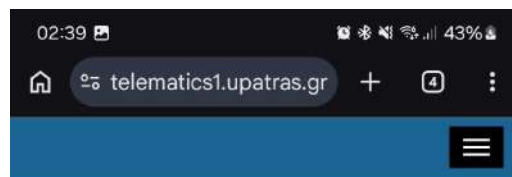
Εικόνα 6.2: Αρχική σελίδα — Drupal 10 (mobile)

Όπως φαίνεται στις Εικόνες 6.1 και 6.2, και οι δύο υλοποιήσεις εμφανίζουν hamburger menu στην κορυφή. Η WordPress έκδοση χρησιμοποιεί hero section που καταλαμβάνει ολόκληρη την οθόνη, ενώ η Drupal 10 έκδοση προχωράει αμέσως στο περιεχόμενο με τον slider νέων.

Σελίδα Μελών



Εικόνα 6.3: Σελίδα μελών — WordPress (mobile)



Lab of Distributed System and Telematics

Home > About Us > People

People

Members Of LDST

Title	
Vassiliki Mantadaki	vmantadaki@upatras.gr
Apostolos Gkamas	gkamas@gmail.com
Charalampos Chatzigeorgiou	cchatzigeorgiou@ceid.upatras.gr



Εικόνα 6.4: Σελίδα μελών — Drupal 10 (mobile)

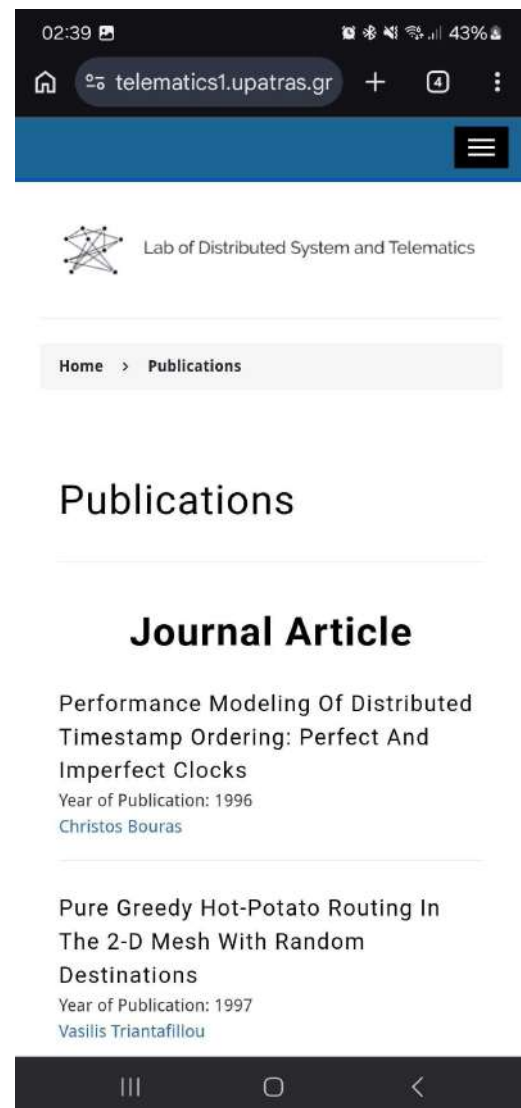
Στις Εικόνες 6.3 και 6.4 φαίνεται καθαρά η διαφορά φιλοσοφίας: το WordPress εμφανίζει τα μέλη σε grid με φωτογραφία και ονοματεπώνυμο, ενώ το Drupal 10 χρησιμοποιεί πίνακα με ονοματεπώνυμο και email. Και οι δύο προσαρμόζονται σωστά στο πλάτος της οθόνης.

Σελίδα Δημοσιεύσεων

Στις Εικόνες 6.5 και 6.6 φαίνεται ότι το WordPress εμφανίζει πλήρη βιβλιογραφικά στοιχεία inline με φίλτρα αναζήτησης, ενώ το Drupal 10 παρουσιάζει πιο λιτή λίστα με τίτλο, έτος και συγγραφείς. Η Drupal 10 προσέγγιση είναι πιο εύκολα σαρώσιμη σε mobile.



Εικόνα 6.5: Σελίδα δημοσιεύσεων — WordPress (mobile)



Εικόνα 6.6: Σελίδα δημοσιεύσεων — Drupal 10 (mobile)

Σελίδα Project



Εικόνα 6.7: Σελίδα project — WordPress (mobile)



Εικόνα 6.8: Σελίδα project — Drupal 10 (mobile)

Στις Εικόνες 6.7 και 6.8 παρουσιάζεται η σελίδα του project CLIMATE και στις δύο υλοποιήσεις. Το λογότυπο χρηματοδότησης εμφανίζεται σωστά και στις δύο περιπτώσεις. Η Drupal 10 έκδοση τοποθετεί τα λογότυπα δίπλα στον τίτλο, ενώ η WordPress έκδοση τα εμφανίζει στο κέντρο της σελίδας ανάμεσα στα πεδία.

6.7.6 Συνολική Αξιολόγηση Διαδικασίας Ανάπτυξης

Το WordPress με Elementor τείνει να ταιριάζει καλύτερα σε χρήστες που θέλουν να στήσουν γρήγορα μια σελίδα, ακόμα και με ελάχιστη τεχνική εμπλοκή και χωρίς συγγραφή κώδικα, παραμένοντας προσβάσιμο ακόμη και σε όσους δεν διαθέτουν εκτεταμένη τεχνική

γνώση, καθώς και σε sites όπου η οπτική ευελιξία αποτελεί προτεραιότητα [38, 12]. Το Drupal 10 είναι καταλληλότερο για sites με σύνθετη δομή δεδομένων, όπου η συνέπεια και η συντηρησιμότητα του περιεχομένου είναι πιο σημαντικές από την οπτική ευελιξία [12, 17]. Για ένα ακαδημαϊκό εργαστήριο με καλά ορισμένους τύπους περιεχομένου (δημοσιεύσεις, μέλη, projects), το Drupal 10 αποδεικνύεται η πιο κατάλληλη επιλογή.

6.8 Ποιοτική Σύγκριση των Δύο Ιστότοπων

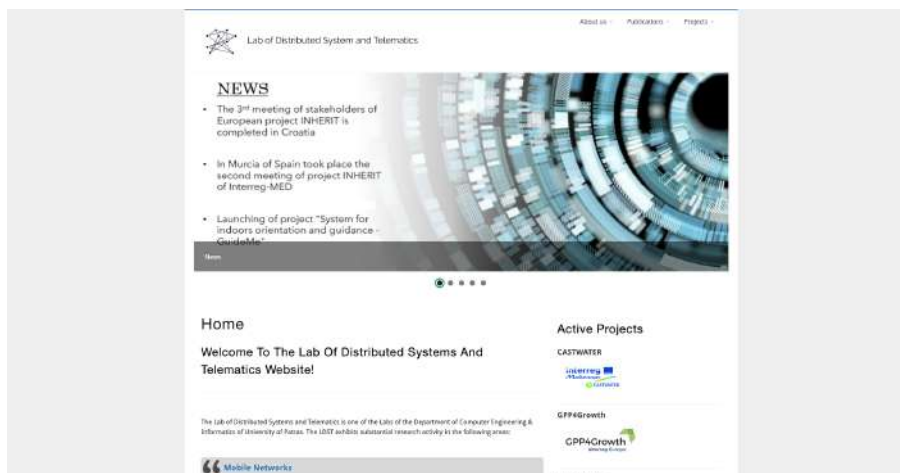
Πέρα από τις ποσοτικές μετρικές απόδοσης, η σύγκριση των δύο ιστότοπων αποκαλύπτει σημαντικές διαφορές στη φιλοσοφία οργάνωσης και παρουσίασης του περιεχομένου, οι οποίες έχουν άμεσες επιπτώσεις στην εμπειρία του τελικού χρήστη. Στις επόμενες ενότητες εξετάζουμε τις κυριότερες από αυτές τις διαφορές μέσα από τέσσερα αντιπροσωπευτικά παραδείγματα.

6.8.1 Αρχική Σελίδα

Η πρώτη και πιο άμεση διαφορά μεταξύ των δύο sites εντοπίζεται στην αρχική σελίδα, όπως φαίνεται στα Εικόνες 6.9 και 6.10. Η WordPress έκδοση υποδέχεται τον επισκέπτη με ένα μεγάλο hero τμήμα που καταλαμβάνει ολόκληρο το ορατό τμήμα της οθόνης, με animation δικτύου στο παρασκήνιο και τον τίτλο του εργαστηρίου στο κέντρο. Αυτή η προσέγγιση δημιουργεί ισχυρή πρώτη εντύπωση και επικοινωνεί άμεσα τον επαγγελματικό χαρακτήρα του εργαστηρίου, κάτι που είναι ιδιαίτερα σημαντικό για έναν νέο επισκέπτη — π.χ. έναν υποψήφιο συνεργάτη ή χρηματοδότη — που αξιολογεί το εργαστήριο για πρώτη φορά [39]. Η Drupal 10 έκδοση από την άλλη πλευρά παραλείπει αυτό το οπτικό εισαγωγικό στάδιο και προχωράει κατευθείαν στο περιεχόμενο, εξυπηρετώντας καλύτερα τον επαναλαμβανόμενο επισκέπτη που γνωρίζει ήδη το εργαστήριο και αναζητά συγκεκριμένη πληροφορία χωρίς να θέλει να «χάσει χρόνο» με εισαγωγικά στοιχεία.



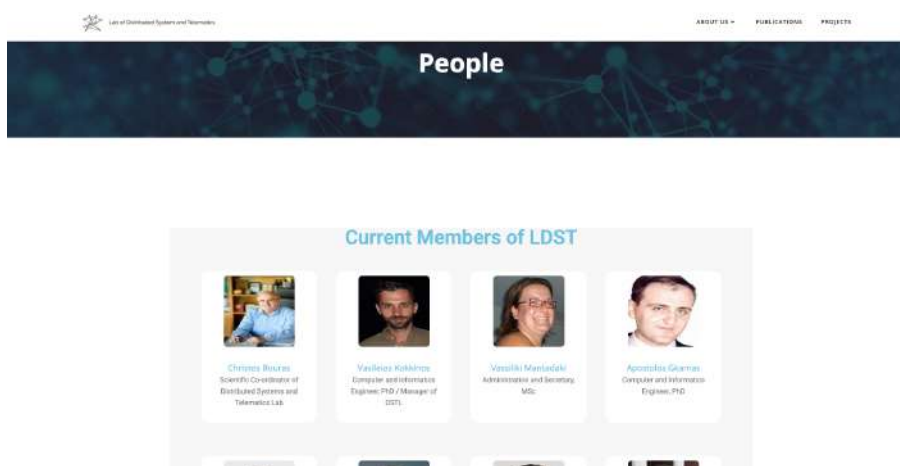
Εικόνα 6.9: Αρχική σελίδα — WordPress



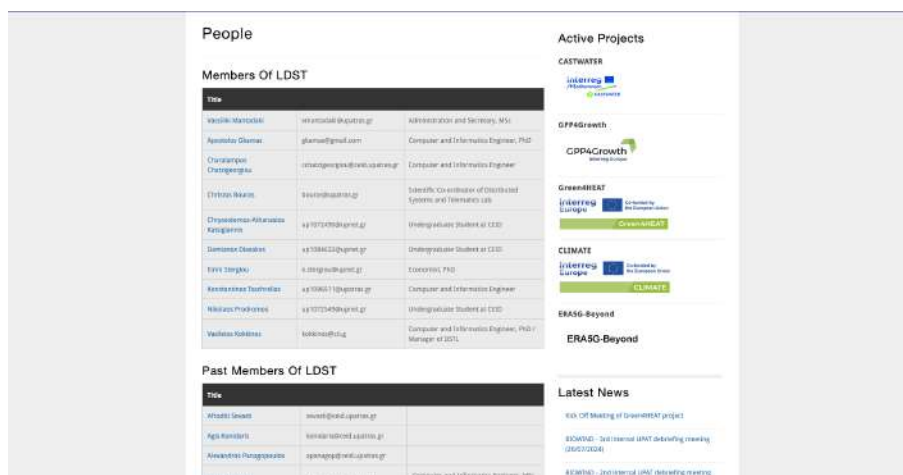
Εικόνα 6.10: Αρχική σελίδα — Drupal 10

6.8.2 Σελίδα Μελών

Η σελίδα «People» αποτελεί χαρακτηριστικό παράδειγμα του πώς η ίδια πληροφορία μπορεί να εξυπηρετεί διαφορετικούς τύπους χρηστών ανάλογα με τον τρόπο παρουσίασής της, όπως φαίνεται στις Εικόνες 6.11 και 6.12. Στη WordPress έκδοση τα μέλη εμφανίζονται σε grid με φωτογραφία, όνομα και τίτλο, δημιουργώντας μια παρουσίαση που διευκολύνει την οπτική αναγνώριση προσώπων — χρήσιμη για έναν επισκέπτη που συνάντησε κάποιο μέλος σε συνέδριο και θέλει να το αναγνωρίσει ή να θυμηθεί το όνομά του. Στη Drupal 10 έκδοση, τα μέλη εμφανίζονται σε πίνακα με όνομα, e-mail και τίτλο θέσης, προσφέροντας άμεση πρόσβαση στα στοιχεία επικοινωνίας χωρίς επιπλέον κλικ — αυτό εξυπηρετεί τον χρήστη που γνωρίζει ήδη ποιον αναζητά και θέλει απλώς να βρει γρήγορα ένα e-mail ή τίτλο. Κάθε προσέγγιση υπηρετεί διαφορετικό σενάριο χρήσης, και η επιλογή μεταξύ τους εξαρτάται από το ποιο κοινό-στόχο εξυπηρετεί κυρίως η ιστοσελίδα [40].



Εικόνα 6.11: Σελίδα μελών — WordPress



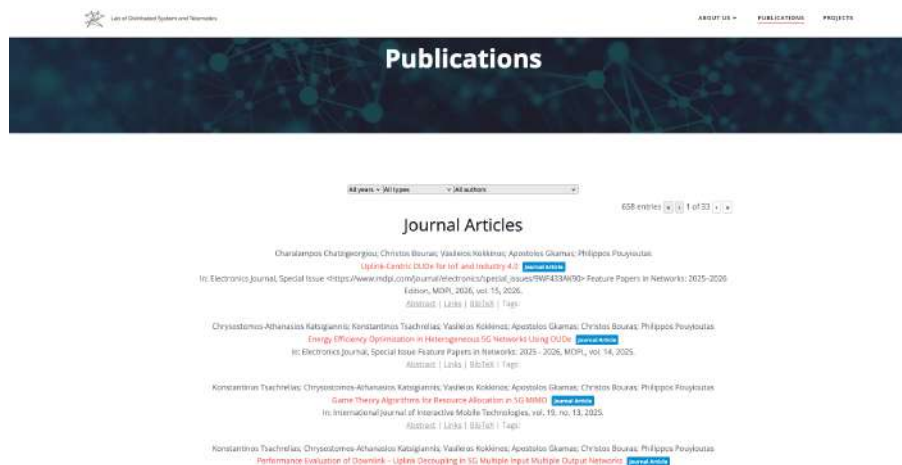
Εικόνα 6.12: Σελίδα μελών — Drupal 10

6.8.3 Σελίδα Δημοσιεύσεων

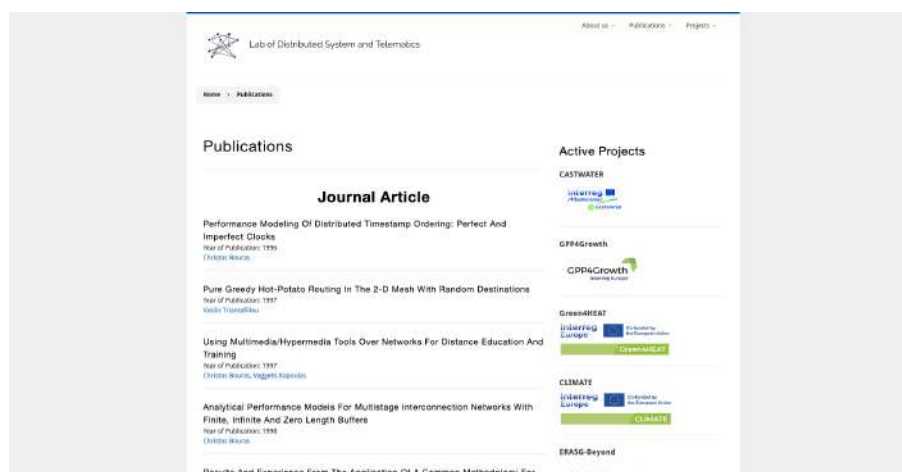
Η ενότητα δημοσιεύσεων αναδεικνύει ίσως την πιο ουσιαστική διαφορά φιλοσοφίας μεταξύ των δύο υλοποιήσεων, όπως φαίνεται στις Εικόνες 6.13 και 6.14. Στη WordPress έκδοση, κάθε εγγραφή εμφανίζει inline τον τίτλο, τους συγγραφείς, το περιοδικό, abstract, BibTeX και tags, οργανωμένα σε σελίδες των 20 εγγραφών. Αυτή η προσέγγιση εξυπηρετεί έναν ερευνητή που θέλει να αξιολογήσει γρήγορα μια δημοσίευση ή να αντιγράψει τα βιβλιογραφικά στοιχεία χωρίς να χρειαστεί να ανοίξει νέα σελίδα. Στη Drupal 10 έκδοση, η λίστα εμφανίζει μόνο τον τίτλο, τον τύπο, το έτος και τους συγγραφείς, με κάθε δημοσίευση να διαθέτει ξεχωριστή σελίδα με permalink. Αυτό προσφέρει ένα σημαντικό πλεονέκτημα: κάθε δημοσίευση μπορεί να κοινοποιηθεί με μοναδικό URL, κάτι ιδιαίτερα χρήσιμο για έναν ερευνητή που θέλει να αναφερθεί σε συγκεκριμένη εργασία του σε e-mail ή σε βιογραφικό. Η απουσία αυτής της δυνατότητας από τη WordPress υλοποίηση αποτελεί λειτουργική παράλειψη για ένα ακαδημαϊκό εργαστήριο [41].

6.8.4 Σελίδα Νέων

Η σελίδα νέων αποτελεί ένα ακόμα χαρακτηριστικό παράδειγμα της διαφορετικής φιλοσοφίας σχεδίασης των δύο υλοποιήσεων, όπως φαίνεται στις Εικόνες 6.15 και 6.16. Η WordPress έκδοση χρησιμοποιεί masonry layout, όπου τα νέα τοποθετούνται σε κάρτες διαφορετικού ύψους που εναλλάσσονται οπτικά στον χώρο της σελίδας. Αν και αυτή η παρουσίαση είναι οπτικά ελκυστική, δημιουργεί πρακτικά προβλήματα αναγνωσιμότητας: ο χρήστης δυσκολεύεται να αντιληφθεί ποιο νέο είναι πιο πρόσφατο, καθώς η χρονολογική σειρά δεν είναι προφανής, ενώ το κείμενο κάθε νέου εμφανίζεται κομμένο χωρίς να φαίνεται ολόκληρο. Αντίθετα, η Drupal 10 έκδοση παρουσιάζει τα νέα σε απλή χρονολογική λίστα, όπου κάθε εγγραφή εμφανίζει τον τίτλο, την ημερομηνία και ένα σύντομο απόσπασμα. Αυτή η προσέγγιση, αν και λιγότερο εντυπωσιακή οπτικά, εξυπηρετεί σαφώς καλύτερα τον στόχο μιας σελίδας νέων: ο επισκέπτης

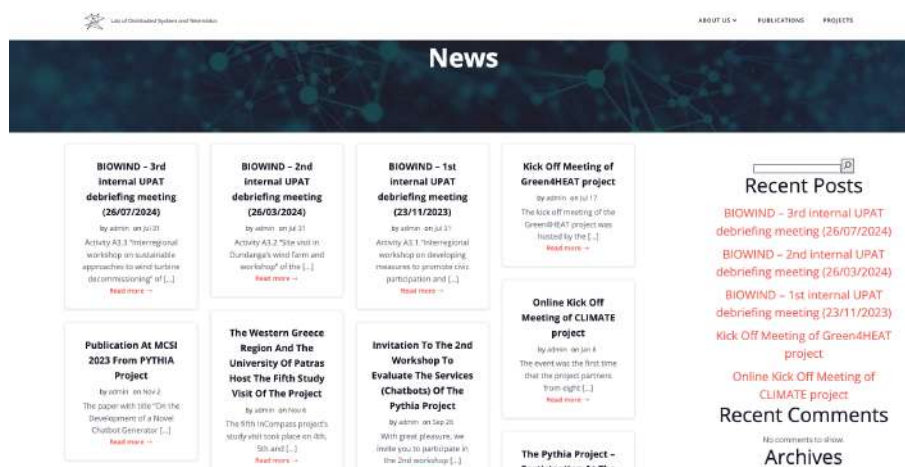


Εικόνα 6.13: Σελίδα δημοσιεύσεων — WordPress

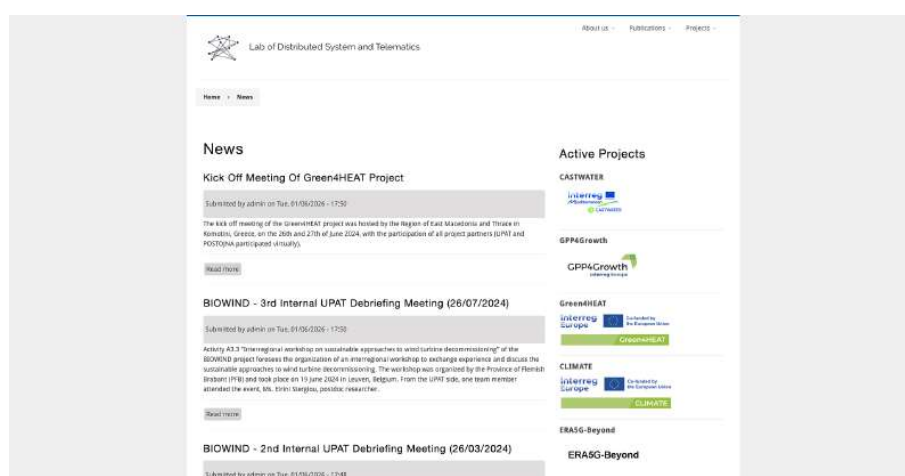


Εικόνα 6.14: Σελίδα δημοσιεύσεων — Drupal 10

μπορεί να σαρώσει γρήγορα τη λίστα, να αντιληφθεί αμέσως ποια είναι τα πιο πρόσφατα νέα και να αποφασίσει αν τον ενδιαφέρει να διαβάσει περισσότερα με ένα κλικ.



Εικόνα 6.15: Σελίδα νέων — WordPress



Εικόνα 6.16: Σελίδα νέων — Drupal 10

6.8.5 Σελίδα Projects

Η σελίδα projects αποτελεί ένα χαρακτηριστικό παράδειγμα διαφορετικής οργάνωσης της ίδιας πληροφορίας. Στη WordPress υλοποίηση τα projects εμφανίζονται σε απλή λίστα χωρίς διαχωρισμό μεταξύ ενεργών και ολοκληρωμένων, ενώ η πλήρης περιγραφή κάθε project εμφανίζεται inline. Στο Drupal 10 τα projects διαχωρίζονται ρητά σε «Active Projects» και «Completed Projects», με κάθε εγγραφή να εμφανίζει μόνο τον τίτλο και μια σύντομη περιγραφή. Αυτός ο διαχωρισμός εξυπηρετεί τον επισκέπτη που θέλει να δει αμέσως ποια projects βρίσκονται σε εξέλιξη, κάτι που είναι από τις πιο συχνές ανάγκες ενός επισκέπτη ακαδημαϊκού

εργαστηρίου [41].

6.8.6 Μεμονωμένη Σελίδα Δημοσίευσης

Η σελίδα μεμονωμένης δημοσίευσης αναδεικνύει μια σημαντική λειτουργική διαφορά μεταξύ των δύο υλοποιήσεων. Στη WordPress έκδοση δεν υπάρχει ξεχωριστή permalink σελίδα για κάθε δημοσίευση — όλα τα βιβλιογραφικά στοιχεία εμφανίζονται inline στη λίστα. Αυτό σημαίνει ότι δεν είναι δυνατόν να κοινοποιηθεί μια συγκεκριμένη δημοσίευση με μοναδικό URL. Στο Drupal 10 κάθε δημοσίευση έχει τη δική της σελίδα με μοναδικό URL, πράγμα που επιτρέπει την κοινοποίηση σε email ή βιογραφικό. Αυτή η δυνατότητα είναι ιδιαίτερα σημαντική για ένα ακαδημαϊκό εργαστήριο που συχνά χρειάζεται να αναφέρεται σε συγκεκριμένες εργασίες [41].

6.9 Ερμηνεία των Αποτελεσμάτων Απόδοσης

Τα αποτελέσματα των μετρήσεων απόδοσης αποκτούν μεγαλύτερη ερμηνευτική αξία όταν εξετάζονται υπό το πρίσμα της τεχνικής αρχιτεκτονικής των δύο sites. Ο ιστότοπος WordPress χρησιμοποιεί το page builder Elementor (έκδοση 4.0.7), όπως αποκαλύπτεται από τον πηγαίο κώδικα της σελίδας. Το Elementor φορτώνει πολλαπλά αρχεία CSS και JavaScript ανεξάρτητα από το αν χρησιμοποιούνται σε κάθε σελίδα, γεγονός που εξηγεί τον υπερδιπλάσιο αριθμό HTTP αιτημάτων (56 έναντι 24). Το υψηλό TTFB του WordPress (1.75s έναντι 348ms, περίπου πενταπλάσιο) δεν οφείλεται σε αυτά τα αρχεία, καθώς το TTFB μετρά τον χρόνο απόκρισης του server πριν καν ξεκινήσει η φόρτωσή τους, αλλά στο βαρύτερο φορτίο επεξεργασίας ανά αίτημα: σε κάθε κλήση το WordPress εκτελεί τη μηχανή απόδοσης του Elementor και τα ενεργά plugins, με αντίστοιχο κόστος σε κώδικα PHP και ερωτήματα στη βάση δεδομένων [32, 37, 35]. Αντίθετα, το Drupal 10 επωφελείται από το ενεργό ενσωματωμένο page caching και τον αποδοτικό render pipeline του, που του επιτρέπουν να απαντά στα αιτήματα ανώνυμων επισκεπτών πολύ ταχύτερα.

Η υπεροχή του Drupal 10 στο mobile Performance (67 έναντι 55) είναι ιδιαίτερα σημαντική, δεδομένου ότι το κινητό web αντιπροσωπεύει πλέον την πλειονότητα της διαδικτυακής κίνησης παγκοσμίως [27, 42]. Το Elementor επηρεάζει δυσανάλογα τις mobile συσκευές, οι οποίες διαθέτουν λιγότερους υπολογιστικούς πόρους για την επεξεργασία του επιπλέον JavaScript. Στο desktop η διαφορά είναι αμελητέα (71 έναντι 72), κάτι που υποδηλώνει ότι για έναν τυπικό επισκέπτη από υπολογιστή η εμπειρία φόρτωσης είναι πρακτικά ισοδύναμη.

Η μόνη μετρική στην οποία το WordPress υπερέρχει σαφώς είναι το Cumulative Layout Shift (CLS) σε desktop (0.006 έναντι 0.26), που σημαίνει ότι κατά τη φόρτωση τα στοιχεία δεν μετακινούνται ορατά. Στο Drupal 10 η μετατόπιση οφείλεται κυρίως στον slider της αρχικής σελίδας και μπορεί να αντιμετωπιστεί με τον ορισμό ρητών διαστάσεων στις εικόνες μέσω CSS [43].

Οι διαφορές στις μετρικές απόδοσης αποκτούν συγκεκριμένη σημασία όταν μεταφραστούν σε εμπειρία χρήστη. Ένα TTFB της τάξης των 1.75s σημαίνει ότι ο browser του επισκέπτη

περιμένει σχεδόν δύο δευτερόλεπτα χωρίς να λάβει κανένα δεδομένο από τον server — το ίδιο το περιεχόμενο δεν έχει καν αρχίσει να φορτώνει. Έρευνες έχουν δείξει ότι καθυστερήσεις άνω του ενός δευτερολέπτου διακόπτουν την αίσθηση άμεσης ανταπόκρισης για τον χρήστη [44], ενώ καθυστερήσεις άνω των τριών δευτερολέπτων οδηγούν σε εγκατάλειψη της σελίδας από σημαντικό ποσοστό επισκεπτών [45]. Στην περίπτωση του Drupal 10, το TTFB των 348ms τοποθετεί την ιστοσελίδα εντός των αποδεκτών ορίων, παρέχοντας εμπειρία που γίνεται αντιληπτή ως άμεση.

Κεφάλαιο 7

Συμπεράσματα

Η παρούσα εργασία ασχολήθηκε με τη μεταφορά και αξιολόγηση του ιστότοπου του εργαστηρίου Τηλεματικής από Drupal 7 σε Drupal 10. Στα πλαίσιά της εξετάστηκε το Drupal ως σύστημα, οι αρχιτεκτονικές διαφορές μεταξύ των δύο εκδόσεων, οι δυσκολίες της μετάβασης, η διαδικασία υλοποίησης του νέου site και, τέλος, η σύγκρισή του με την υπάρχουσα WordPress υλοποίηση μέσω ποσοτικών και ποιοτικών μετρήσεων.

7.1 Κύρια ευρήματα

Το βασικό συμπέρασμα αφορά τη φύση της ίδιας της μετάβασης: η «αναβάθμιση» από D7 σε D10 δεν συνιστά αναβάθμιση με την παραδοσιακή έννοια, αλλά ουσιαστικά πλήρη ανακατασκευή, και η απόφαση για πλήρη ανακατασκευή αντί για αξιοποίηση του Migrate API αποδείχθηκε ορθή, δεδομένου του σχετικά μικρού όγκου περιεχομένου και της ανάγκης για καθαρό, συντηρήσιμο κώδικα.

Από πλευράς απόδοσης, η σύγκριση με την WordPress υλοποίηση ανέδειξε σαφή πλεονεκτήματα του Drupal 10: χρόνος πλήρους φόρτωσης 2.4 φορές μικρότερος, TTFB 5 φορές καλύτερο και αριθμός HTTP αιτημάτων σχεδόν στο μισό. Αντίθετα, οι βαθμολογίες SEO και προσβασιμότητας υπολείπονται, κυρίως λόγω απουσίας meta descriptions και alt κειμένων — προβλήματα που μπορούν να διορθωθούν εύκολα.

7.2 Μελλοντική εργασία

Η υλοποίηση που παρουσιάστηκε αποτελεί σταθερή βάση για περαιτέρω βελτιώσεις, οι οποίες οργανώνονται σε πέντε κατηγορίες: βελτιώσεις SEO (Search Engine Optimization), προσβασιμότητας και ασφάλειας, λειτουργικές επεκτάσεις, τεχνικές βελτιστοποιήσεις και ζητήματα μακροπρόθεσμης συντηρησιμότητας. Ορισμένες από αυτές δεν υλοποιήθηκαν στο πλαίσιο της παρούσας εργασίας για συγκεκριμένους λόγους: είτε απαιτούν πρόσβαση στη διαμόρφωση του server (η οποία δεν ήταν διαθέσιμη, καθώς η διαχείριση περιοριζόταν στο Admin interface του Drupal), είτε εξαρτώνται από την εκκρεμή αλλαγή του οριστικού ονόματος τομέα (domain) του ιστότοπου, είτε αποτελούν αυτοτελή έργα με σημαντικό φόρτο ανάπτυξης

ή μετάφρασης περιεχομένου.

Βελτιώσεις SEO, προσβασιμότητας και ασφάλειας

Σε επίπεδο SEO, το περιεχόμενο μπορεί να εμπλουτιστεί με δομημένα μεταδεδομένα. Η εγκατάσταση και ρύθμιση του module **Metatag** [6] επιτρέπει τον ορισμό meta περιγραφών ανά τύπο περιεχομένου, ενώ το Simple XML Sitemap module παράγει XML sitemap που διευκολύνει την ευρετηρίαση από τις μηχανές αναζήτησης. Για ακαδημαϊκό περιεχόμενο, η προσθήκη δομημένων δεδομένων (structured data) με το πρότυπο Schema.org (ιδίως των τύπων **ScholarlyArticle** και **Person**) βελτιώνει την εμφάνιση των αποτελεσμάτων στη Google [29]. Η εφαρμογή τους απαιτεί την εγκατάσταση των αντίστοιχων modules και τον συστηματικό ορισμό μεταδεδομένων για κάθε τύπο περιεχομένου.

Σε επίπεδο προσβασιμότητας, εκκρεμεί η συστηματική συμπλήρωση εναλλακτικών κειμένων (alt text) στις εικόνες του site, που ωφελεί ταυτόχρονα τους χρήστες υποστηρικτικών τεχνολογιών και το SEO. Η εργασία αυτή γίνεται από το Admin interface, αλλά ο φόρτος της είναι ανάλογος του όγκου του υπάρχοντος περιεχομένου.

Σε επίπεδο ασφάλειας, η προσθήκη HTTP Security Headers, όπως Strict-Transport-Security, Content-Security-Policy και Referrer-Policy, αναμένεται να αναβαθμίσει τη βαθμολογία ασφαλείας χωρίς αλλαγές στον κώδικα της εφαρμογής [34]. Η ρύθμιση αυτή γίνεται σε επίπεδο διαμόρφωσης του Apache server (αρχείο **.htaccess** ή virtual host), στην οποία δεν υπήρχε πρόσβαση στο πλαίσιο της εργασίας. Επιπλέον, ρυθμίσεις όπως το Strict-Transport-Security συνδέονται με το οριστικό domain και έχει νόημα να οριστικοποιηθούν μετά την αλλαγή του. Στον ίδιο άξονα, η απόκρυψη της επικεφαλίδας **X-Generator**, που αποκαλύπτει το χρησιμοποιούμενο CMS, περιορίζει την έκθεση πληροφοριών και επιτυγχάνεται επίσης σε επίπεδο διαμόρφωσης server.

Λειτουργικές επεκτάσεις

Μια σημαντική επέκταση είναι η προσθήκη δεύτερης γλώσσας στο site (το υπάρχον περιεχόμενο είναι στα αγγλικά). Το Drupal 10 υποστηρίζει πολύγλωσση λειτουργία εκ του πυρήνα μέσω των modules Language και Content Translation, που αποτελούν μέρος του core. Η τεχνική υποδομή είναι επομένως άμεσα διαθέσιμη· το κύριο κόστος της επέκτασης δεν είναι τεχνικό, αλλά αφορά τη μετάφραση του συνόλου του περιεχομένου (δημοσιεύσεις, projects, σελίδες, στοιχεία μενού) ανά κόμβο, που για τον όγκο περιεχομένου του εργαστηρίου αποτελεί σημαντικό έργο. Ολοκληρωμένη, θα αύξανε την εμβέλεια του εργαστηρίου στη διεθνή ακαδημαϊκή κοινότητα [6].

Εξίσου χρήσιμη θα ήταν η αυτοματοποίηση της εισαγωγής δημοσιεύσεων, η οποία σήμερα γίνεται χειροκίνητα και είναι χρονοβόρα και επιρρεπής σε λάθη. Μια λύση είναι η ανάπτυξη custom module που να επιτρέπει εισαγωγή από αρχείο BibTeX, ή η αξιοποίηση του DOI (Digital Object Identifier) για αυτόματη άντληση μεταδεδομένων μέσω του API του CrossRef. Στον ίδιο άξονα εντάσσεται η σύνδεση με υπηρεσίες όπως το Google Scholar και το ORCID (Open Researcher and Contributor ID) για αυτόματο συγχρονισμό των δημοσιεύσεων κάθε

μέλους. Πρόκειται για αυτοτελές έργο ανάπτυξης λογισμικού με ενσωμάτωση εξωτερικών υπηρεσιών, που υπερβαίνει το πλαίσιο της παρούσας εργασίας.

Τέλος, η δημιουργία RSS feeds για τις ενότητες Νέα και Δημοσιεύσεις, μέσω του Feed display του Views, θα επέτρεπε στους ενδιαφερόμενους να παρακολουθούν τις ενημερώσεις του εργαστηρίου μέσω feed reader, χωρίς να επισκέπτονται τακτικά τη σελίδα.

Τεχνικές βελτιστοποιήσεις

Σε επίπεδο απόδοσης, η παρούσα υλοποίηση αξιοποιεί ήδη αρκετούς από τους μηχανισμούς βελτιστοποίησης του Drupal 10. Το ενσωματωμένο page caching για ανώνυμους επισκέπτες (μέσω των core modules Internal Page Cache και Dynamic Page Cache) είναι ενεργό και αποθηκεύει τις εξόδους απόδοσης (render) αντί να τις υπολογίζει εκ νέου σε κάθε αίτημα, συνεισφέροντας στους χαμηλούς χρόνους απόκρισης που καταγράφηκαν [6]. Η λειτουργία του επιβεβαιώθηκε εμπειρικά μέσω της επικεφαλίδας **X-Drupal-Cache: HIT** σε αιτήματα ανώνυμων επισκεπτών. Επιπλέον, ρυθμίστηκε η προσωρινή αποθήκευση στον περιηγητή και στους διαμεσολαβητές (browser/proxy caching) μέσω της επικεφαλίδας **Cache-Control max-age**, ώστε οι επαναλαμβανόμενες επισκέψεις και οι ενδιαμέσοι διαμεσολαβητές (proxies) να μην επιβαρύνουν εκ νέου τον διακομιστή (server).

Παρομοίως, το Drupal 10 εφαρμόζει ήδη lazy loading (**loading="lazy"**) στις εικόνες του ιστότοπου (site), όπως επιβεβαιώθηκε στον πηγαίο κώδικα διαφόρων σελίδων· εξαιρούνται, ορθώς, οι εικόνες που βρίσκονται εντός της αρχικής ορατής περιοχής της οθόνης (viewport), στις οποίες η τεχνική δεν προσφέρει όφελος. Το κυριότερο περιθώριο βελτίωσης για το mobile LCP, που αποτελεί το ασθενέστερο σημείο της παρούσας υλοποίησης, εντοπίζεται επομένως όχι στον τρόπο φόρτωσης αλλά στη βελτιστοποίηση των ίδιων των εικόνων: συμπίεση, χρήση σύγχρονων μορφών (π.χ. WebP) και απόδοση κατάλληλων μεγεθών ανά συσκευή μέσω responsive images [28, 29].

Περαιτέρω βελτίωση θα προσέφερε η χρήση δικτύου διάδοσης περιεχομένου (CDN-Content Delivery Network), ενός γεωγραφικά κατανεμημένου δικτύου διακομιστών (servers) που σερβίρει στατικά αρχεία (εικόνες, CSS, JavaScript) από την πλησιέστερη στον χρήστη τοποθεσία, μειώνοντας τους χρόνους φόρτωσης για επισκέπτες από διαφορετικές γεωγραφικές περιοχές. Η αξιοποίησή του εξαρτάται από την υποδομή και το οριστικό domain του ιστότοπου [29].

Τέλος, η ρύθμιση αυτοματοποιημένων αντιγράφων ασφαλείας μέσω του module Backup and Migrate, τόσο της βάσης δεδομένων όσο και των αρχείων του ιστότοπου, σε εξωτερικό αποθηκευτικό χώρο (π.χ. remote SFTP ή υπηρεσία αποθήκευσης νέφους, cloud storage), αποτελεί βασική απαίτηση για κάθε ιστότοπο σε παραγωγή και εξασφαλίζει γρήγορη αποκατάσταση σε περίπτωση βλάβης ή επίθεσης. Η ρύθμισή τους προϋποθέτει πρόσβαση στο παραγωγικό περιβάλλον και στον εξωτερικό αποθηκευτικό χώρο.

Μακροπρόθεσμη συντηρησιμότητα

Ένα ζήτημα που συχνά παραβλέπεται στους ακαδημαϊκούς ιστότοπους είναι η **τεκμηρίωση** για τον επόμενο διαχειριστή. Η παρούσα εργασία τεκμηριώνει την αρχική υλοποίηση, αλλά χρειάζεται επιπλέον ένας πρακτικός **οδηγός διαχείρισης** που να καλύπτει τις συχνότερες εργασίες: πώς προστίθεται νέο μέλος, πώς εισάγεται δημοσίευση, πώς δημιουργείται νέο project. Χωρίς τέτοια τεκμηρίωση, κάθε αλλαγή διαχειριστή συνεπάγεται χαμπύλη εκμάθησης που μπορεί να οδηγήσει σε λάθη ή σε εγκατάλειψη της ενεργού συντήρησης του site. Πρόκειται για αυτοτελές παραδοτέο, πέραν της παρούσας τεκμηρίωσης.

Στον ίδιο άξονα εντάσσεται η καθιέρωση διαδικασίας αυτοματοποιημένων ενημερώσεων για το Drupal core και τα contributed modules, είτε μέσω του module **Automatic Updates**, είτε με προγραμματισμένο cron job που εκτελεί **composer update** και ειδοποιεί τον διαχειριστή για διαθέσιμες ενημερώσεις ασφαλείας. Αυτό αντιμετωπίζει ακριβώς το πρόβλημα που οδήγησε στην παρούσα εργασία: την παραμονή σε μια ξεπερασμένη έκδοση CMS λόγω έλλειψης συστηματικής διαδικασίας ενημέρωσης [1].

7.3 Τελικές παρατηρήσεις

Η εκπόνηση αυτής της εργασίας ανέδειξε με σαφήνεια ότι η συντήρηση και αναβάθμιση λογισμικού αποτελεί κρίσιμη πτυχή της λειτουργίας ενός ακαδημαϊκού εργαστηρίου. Η αδράνεια απέναντι σε τεχνολογικές αλλαγές — όπως η συνέχιση χρήσης ενός CMS μετά το EOL του — δεν συνιστά εξοικονόμηση πόρων αλλά συσσώρευση τεχνικού χρέους που καθίσταται δυσανάλογα δαπανηρό να αντιμετωπιστεί στο μέλλον. Το Drupal 10 είναι εξαιρετική επιλογή για σύνθετες εφαρμογές με αυστηρές απαιτήσεις δομής δεδομένων και ασφάλειας, αλλά η υψηλή χαμπύλη εκμάθησής του απαιτεί προσεκτικό αρχικό σχεδιασμό και τεχνική υποστήριξη για τη μακροπρόθεσμη συντήρησή του.

Βιβλιογραφία

- [1] Drupal.org, *Drupal 7 End of Life*, <https://www.drupal.org/psa-2023-11-01>, 2023.
- [2] OWASP Foundation, *OWASP Top Ten – The Ten Most Critical Web Application Security Risks*, <https://owasp.org/www-project-top-ten/>, Accessed: 2025.
- [3] C.A. Conțu, E.C. Popovici, O. Fratu, M.G. Berceanu, *Security Issues in Most Popular Content Management Systems*, in: 2016 International Conference on Communications (COMM), IEEE, pp. 277–280, 2016.
- [4] S.K. Patel, V.R. Rathod, S. Parikh, *Joomla, Drupal and WordPress — A Statistical Comparison of Open Source CMS*, in: 3rd International Conference on Trendz in Information Sciences & Computing (TISC 2011), IEEE, pp. 182–187, 2011.
- [5] S.K. Patel, F. Suthar, S. Patel, J. Prajapati, *A Comparison of Top-Rated Open-Source CMS — Joomla, Drupal, and WordPress for E-Commerce Website*, in: Computer Vision and Robotics, Algorithms for Intelligent Systems, Springer, Singapore, 2023.
- [6] Drupal.org, *Drupal Documentation*, <https://www.drupal.org/documentation>, Accessed: 2025.
- [7] Symfony, *Symfony Documentation*, <https://symfony.com/doc/current/index.html>, Accessed: 2025.
- [8] M. Glaman, K. Quillen, *Drupal 10 Development Cookbook: Practical Recipes to Harness the Power of Drupal for Building Digital Experiences and Dynamic Websites*, Packt Publishing, 2023.
- [9] WordPress, *WordPress – Blog Tool, Publishing Platform, and CMS*, <https://wordpress.org/>, Accessed: 2025.
- [10] W3Techs, *Usage statistics of content management systems*, https://w3techs.com/technologies/overview/content_management, Accessed: 2025.
- [11] J. Ruohonen, *A Demand-Side Viewpoint to Software Vulnerabilities in WordPress Plugins*, in: Proceedings of the 23rd International Conference on Evaluation and Assessment in Software Engineering (EASE '19), ACM, pp. 222–228, 2019.

- [12] A.A. Prasad, N.A. Sharma, A. Kumar, *A Comparative Analysis of Joomla, Drupal, WordPress and ASP.NET: Exploring Features, Performance, and Suitability*, in: 2023 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), IEEE, pp. 1–7, 2023.
- [13] Joomla, *Joomla! – The CMS Trusted By Millions*, <https://www.joomla.org/>, Accessed: 2025.
- [14] TYPO3 Association, *TYPO3 – The Professional, Flexible Content Management System*, <https://typo3.org/>, Accessed: 2025.
- [15] T. Patel, S. Mittal, N.K. Awadhiya, *A Review on Content Management Systems of Web Development*, International Journal of Computer Science Trends and Technology (IJCST), Vol. 7, Issue 2, pp. 101–104, 2019.
- [16] A. Kumar, A. Narayan, V. Sharma, A.A. Prasad, M. Sami, *Decoding the Web CMS Landscape: A Comparative Study of Popular Web Content Management Systems*, International Journal of Computers and Their Applications (IJCA), Vol. 31, No. 4, pp. 281–292, 2024.
- [17] A. Mirdha, A. Jain, K. Shah, *Comparative Analysis of Open Source Content Management Systems*, in: 2014 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), IEEE, pp. 1–4, 2014.
- [18] M. Alqadhi, A. Alabduljabbar, K. Thomas, S. Salem, D. Nyang, D. Mohaisen, *Do Content Management Systems Impact the Security of Free Content Websites? A Correlation Analysis*, arXiv preprint arXiv:2210.12083, 2022.
- [19] Drupal.org, *SA-CORE-2014-005 – Drupal core – SQL injection*, <https://www.drupal.org/SA-CORE-2014-005>, 2014.
- [20] Drupal.org, *SA-CORE-2018-002 – Drupal core – Remote Code Execution*, <https://www.drupal.org/sa-core-2018-002>, 2018.
- [21] Composer, *Composer – A Dependency Manager for PHP*, <https://getcomposer.org/doc/>, Accessed: 2025.
- [22] Drush, *Drush Documentation*, <https://www.drush.org/latest/>, Accessed: 2025.
- [23] Twig, *Twig – The flexible, fast, and secure template engine for PHP*, <https://twig.symfony.com/doc/3.x/>, Accessed: 2025.
- [24] Drupal.org, *Views module documentation*, <https://www.drupal.org/docs/8/core/modules/views>, Accessed: 2025.
- [25] A. Bergstein, *Drupal 10 Masterclass: Build Responsive Drupal Applications to Deliver Custom and Extensible Digital Experiences to Users*, Packt Publishing, 2023.

- [26] Drupal.org, *Migrate API*, <https://www.drupal.org/docs/drupal-apis/migrate-api>, Accessed: 2025.
- [27] Statcounter Global Stats, *Mobile vs Desktop vs Tablet Market Share Worldwide*, <https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet>, Accessed: 2026.
- [28] J. Manhas, *A Study of Factors Affecting Websites Page Loading Speed for Efficient Web Performance*, International Journal of Computer Sciences and Engineering, vol. 1, no. 3, pp. 32–35, 2013.
- [29] Google, *Web Vitals — Essential metrics for a healthy site*, <https://web.dev/vitals/>, Accessed: 2026.
- [30] Google, *Lighthouse — Automated auditing, performance metrics, and best practices*, <https://developer.chrome.com/docs/lighthouse/>, Accessed: 2026.
- [31] GTmetrix, *GTmetrix — Website Performance Testing and Monitoring*, <https://gtmetrix.com/>, Accessed: 2026.
- [32] Google, *Time to First Byte (TTFB)*, <https://web.dev/ttfb/>, Accessed: 2026.
- [33] Scott Helme, *Security Headers*, <https://securityheaders.com/>, Accessed: 2026.
- [34] U. Kishnani, S. Das, *Securing the Web: Analysis of HTTP Security Headers in Popular Global Websites*, in: International Conference on Information Systems Security (ICISS 2024), Springer, pp. 87–106, 2024.
- [35] N. Surber, *The WordPress Plugin Problem*, WPBeginner, <https://www.wpbeginner.com/wp-tutorials/how-wordpress-plugins-affect-your-sites-load-time/>, Accessed: 2026.
- [36] Qualys, *SSL Server Test*, <https://www.ssllabs.com/ssltest/>, Accessed: 2026.
- [37] M. Jaquith, *WordPress Performance: The Definitive Guide*, Kinsta Blog, <https://kinsta.com/learn/speed-up-wordpress/>, Accessed: 2026.
- [38] J. Cabot, *WordPress: A Content Management System to Democratize Publishing*, IEEE Software, vol. 35, no. 3, pp. 89–92, 2018.
- [39] G. Lindgaard, G. Fernandes, C. Dudek, J. Brown, *Attention web designers: You have 50 milliseconds to make a good first impression!*, Behaviour & Information Technology, vol. 25, no. 2, pp. 115–126, Taylor & Francis, 2006.
- [40] J. Nielsen, *F-Shaped Pattern of Reading on the Web*, Nielsen Norman Group, <https://www.nngroup.com/articles/f-shaped-pattern-reading-web-content/>, 2006.
- [41] J. Nielsen, H. Loranger, *Prioritizing Web Usability*, New Riders Press, 2006.

-
- [42] HTTP Archive, *Web Almanac 2024: Performance*, HTTP Archive, <https://almanac.httparchive.org/en/2024/performance>, 2024.
- [43] Google, *Optimize Cumulative Layout Shift*, <https://web.dev/articles/optimize-cls>, Accessed: 2026.
- [44] J. Nielsen, *Response Times: The 3 Important Limits*, Nielsen Norman Group, <https://www.nngroup.com/articles/response-times-3-important-limits/>, 1993.
- [45] SOASTA/Akamai, *The State of Online Retail Performance*, Akamai Technologies, <https://www.akamai.com/newsroom/press-release/akamai-releases-spring-2017-state-of-online-retail-performance-report>, 2017.