

# Σχεδίαση & ανάπτυξη πιλοτικού επιστημονικού κόμβου για διάχυση ερευνητικού έργου

---



Σχολή Επιστήμης της Πληροφορίας και Πληροφορικής

Τμήμα Πληροφορικής

Ιόνιο Πανεπιστήμιο

**Στοιχεία Φοιτητή:**

Παπαϊωάννου Ελευθέριος Π2016016

27/09/2024

**Επιβλέπων καθηγητής:**

Θεόδωρος Ανδρόνικος

**Επιτροπή:**

Θεμιστοκλής Έξαρχος

Μιχαήλ Στεφανιδάκης



## Περίληψη

Η παρούσα εργασία επικεντρώνεται στην ανάπτυξη ενός ιστοτόπου για ερευνητική ομάδα στην οποία τα μέλη της ομάδας θα μπορούν να μοιραστούν νέα που τους ενδιαφέρουν, να ανεβάσουν τα έργα τους και τα επαγγελματικά τους βιόματα. Το έργο περιλαμβάνει τη δημιουργία σελίδων για τη δημοσίευση ειδήσεων, ερευνητικών άρθρων και έργων, δημοσιεύσεων και πληροφοριών σχετικά με τα μέλη της ερευνητικής ομάδας. Επιπλέον, έχει ενσωματωθεί υποστήριξη για πολυγλωσσικότητα (Astro i18n, n.d.). Αρχικά, περιγράφεται η διαδικασία επιλογής τεχνολογιών, όπως το Astro για τη βελτίωση της απόδοσης και εύκολης χρήσης δημιουργίας περιεχομένου, τη χρήση React για διαδραστικά στοιχεία. Εν συνεχεία, αναλύεται η ανάπτυξη του ιστοτόπου με έμφαση στη δομή των σελίδων, την οργάνωση του περιεχομένου με χρήση Markdown και Front Matter, και τη δημιουργία προτύπων για την εύκολη προσθήκη περιεχομένου από τους συνεργάτες της ομάδας. Τέλος, συζητούνται οι προκλήσεις που προέκυψαν κατά τη διάρκεια της ανάπτυξης, οι λύσεις που εφαρμόστηκαν, και οι μελλοντικές δυνατότητες βελτίωσης.

## Abstract

This paper focuses on the development of a website for a research team, where the "authors" can share relevant news, upload their projects, and present their professional experiences. The project includes the creation of pages for publishing news, research articles and projects, publications, and information about the research team members. Additionally, multilingual support has been integrated (Astro i18n, n.d.). The paper begins by describing the process of selecting technologies, such as Astro for performance improvement and ease of content creation, and React for interactive elements. The website's development is then analyzed, with emphasis on page structure, content organization using Markdown and Front Matter, and the creation of templates for easily adding content by team members. Finally, the challenges encountered during development, the solutions implemented, and potential future improvements are discussed.



## Πίνακας Περιεχομένων

|                                                                                         |           |
|-----------------------------------------------------------------------------------------|-----------|
| <b>1. Εισαγωγή.....</b>                                                                 | <b>7</b>  |
| Σκοπός του έργου.....                                                                   | 7         |
| Έμφαση της έρευνας.....                                                                 | 7         |
| <b>2. Θεωρητικό Πλαίσιο.....</b>                                                        | <b>9</b>  |
| 2.1 Astro Framework.....                                                                | 9         |
| 2.2 Υποστηρικτικά Εργαλεία και Τεχνολογίες.....                                         | 14        |
| 2.3 Πλεονεκτήματα και Μειονεκτήματα.....                                                | 16        |
| <b>3. Ανάπτυξη του Έργου.....</b>                                                       | <b>19</b> |
| 3.1. Επιλογή του Astro Framework.....                                                   | 19        |
| 3.2. Εξέταση CMS Λύσεων για Διαχείριση Περιεχομένου.....                                | 19        |
| 3.3. Ορισμός Περιεχομένου και Δημιουργία Σχήματος.....                                  | 20        |
| 3.4. Ενσωμάτωση Πρόσθετων Εργαλείων: Tailwind CSS, React, και Shaden UI.....            | 23        |
| 3.5. Routing και χρήση Layouts στο Astro.....                                           | 24        |
| 3.6. Ενσωμάτωση Light/Dark Mode.....                                                    | 27        |
| 3.7. Δημιουργία Σελίδων και Χρήση των Content Collections του Astro.....                | 32        |
| 3.8. Δημιουργία Interactive Header για Κινητές Συσκευές.....                            | 34        |
| 3.9. Δημιουργία Διαδραστικών Components για Φιλτράρισμα και Αναζήτηση Περιεχομένου..... | 34        |
| 3.10. Ενσωμάτωση Υποστήριξης Πολυγλωσσικότητας (i18n).....                              | 36        |
| <b>4. Ρύθμιση Έργου και Διαχείριση Περιεχομένου.....</b>                                | <b>39</b> |
| 4.1 Εγκατάσταση.....                                                                    | 39        |
| 4.2 Προτεινόμενα Εργαλεία και Plugins.....                                              | 39        |
| 4.3 Εντολές (Commands).....                                                             | 41        |
| 4.4 Δομή του Έργου.....                                                                 | 42        |
| 4.5 Διαχείριση Περιεχομένου.....                                                        | 44        |
| 4.6 Διαχείριση Μεταφράσεων.....                                                         | 53        |
| <b>5. Προκλήσεις και Λύσεις.....</b>                                                    | <b>57</b> |
| 5.1 Διαχείριση Περιεχομένου από Μη Τεχνικούς Χρήστες.....                               | 57        |
| 5.2 Πρόκληση: Δημιουργία Responsive UI.....                                             | 58        |
| 5.3 Πρόκληση: Διαχείριση Πολυγλωσσικότητας και Μεταφράσεων.....                         | 58        |
| 5.4 Πρόκληση: Deployment και CI/CD.....                                                 | 58        |
| <b>6. Συμπεράσματα.....</b>                                                             | <b>61</b> |
| <b>7. Βιβλιογραφία.....</b>                                                             | <b>63</b> |



# 1. Εισαγωγή

## Σκοπός του έργου

Ο βασικός σκοπός του παρόντος έργου είναι η δημιουργία ενός ιστοτόπου για μια ερευνητική ομάδα, η οποία επιθυμεί να μοιράζεται τα νέα της, τις δημοσιεύσεις της, καθώς και τα ερευνητικά της έργα με το ακαδημαϊκό κοινό της αλλά και με άλλους ενδιαφερόμενους ανά τον κόσμο. Η ανάγκη για έναν οργανωμένο και λειτουργικό ιστότοπο προέκυψε από τη συσσώρευση μεγάλου όγκου πληροφοριών που σχετίζονται με την ερευνητική δραστηριότητα της ομάδας.

Η διαχείριση του περιεχομένου του ιστοτόπου γίνεται με τη χρήση αρχείων Markdown, τα οποία επιτρέπουν στα μέλη της ομάδας να προσθέτουν και να ενημερώνουν άρθρα, δημοσιεύσεις και έργα χωρίς να απαιτείται εκτεταμένη γνώση προγραμματισμού. Επιπλέον, τα κείμενα που αφορούν τις μεταφράσεις αποθηκεύονται σε ένα configuration file, το οποίο μπορεί εύκολα να επεξεργαστεί για να προστεθούν νέες γλώσσες ή να τροποποιηθεί υπάρχον περιεχόμενο.

Το έργο σχεδιάστηκε με σκοπό την προσφορά μιας φιλικής προς τον χρήστη πλατφόρμας, που να παρέχει ταυτόχρονα την ευελιξία για τη διαχείριση μεγάλου όγκου δεδομένων και πληροφοριών. Μέσα από την εύκολη και γρήγορη ενημέρωση του ιστοτόπου, τα μέλη της ομάδας μπορούν να παραμένουν συνεχή και συνεπείς με τις τελευταίες τους δημοσιεύσεις, δημιουργώντας έτσι ένα ζωντανό ψηφιακό αρχείο της ερευνητικής τους δραστηριότητας.

## Έμφαση της έρευνας

Η έρευνα και η ανάπτυξη του συγκεκριμένου έργου επικεντρώθηκε στη δημιουργία ενός ιστοτόπου με χρήση του Astro framework (Astro, n.d.), το οποίο επιτρέπει την ανάπτυξη στατικών ιστοσελίδων με υψηλή απόδοση. Η επιλογή της τεχνολογίας αυτής έγινε με γνώμονα τις ανάγκες της ερευνητικής ομάδας για ένα γρήγορο, ασφαλές και εύκολα διαχειρίσιμο περιβάλλον. Η ευκολία στη διαχείριση περιεχομένου ήταν βασικό κριτήριο, καθώς τα μέλη της ομάδας πρέπει να μπορούν να προσθέτουν νέα άρθρα, δημοσιεύσεις και ερευνητικά έργα χωρίς ιδιαίτερες τεχνικές γνώσεις.

Η τεχνολογική λύση που επιλέχθηκε υποστηρίζει επίσης την πολυγλωσσικότητα (i18n), κάτι το οποίο κρίθηκε απαραίτητο για την παροχή περιεχομένου σε πολλαπλές γλώσσες, διευκολύνοντας έτσι την πρόσβαση των αποτελεσμάτων της ερευνητικής δραστηριότητας σε διεθνές κοινό. Μέσω της υποστήριξης i18n, ο ιστότοπος είναι έτοιμος να φιλοξενήσει περιεχόμενο στα ελληνικά και στα αγγλικά, καλύπτοντας τις ανάγκες επικοινωνίας της ομάδας σε διαφορετικές γλώσσες.

Η έρευνα κατά τη διάρκεια της ανάπτυξης του έργου επικεντρώθηκε επίσης στην επιλογή των κατάλληλων εργαλείων για τη δημιουργία δυναμικού και ευέλικτου περιεχομένου. Οι τεχνολογίες Markdown και Front Matter επιλέχθηκαν ως κύρια μέσα για την οργάνωση των άρθρων και των δεδομένων της ερευνητικής ομάδας, παρέχοντας την απαραίτητη ευελιξία για την εύκολη εισαγωγή και ενημέρωση περιεχομένου.



## 2. Θεωρητικό Πλαίσιο

### 2.1 Astro Framework

Το Astro είναι ένα framework (Astro, n.d.) που παρέχει δυνατότητες για στατικό site generation (SSG) που έχει σχεδιαστεί με έμφαση στην απόδοση και την ευκολία χρήσης. Η κύρια λειτουργία του Astro είναι να δημιουργεί στατικές σελίδες, οι οποίες αποδίδονται στον χρήστη γρήγορα και αποτελεσματικά. Οι στατικές σελίδες παρέχουν ορισμένα κρίσιμα πλεονεκτήματα για ιστοτόπους, όπως βελτιωμένη απόδοση, αυξημένη ασφάλεια και καλύτερη υποστήριξη SEO. Ο στατικός χαρακτήρας του Astro το καθιστά ιδανικό για έργα όπως η κατασκευή ενός ιστοτόπου για ερευνητική ομάδα, όπου η πλειοψηφία του περιεχομένου μπορεί να ανανεώνεται χωρίς την ανάγκη συνεχών δυναμικών ενημερώσεων σε πραγματικό χρόνο.

Ένας σημαντικός λόγος για την επιλογή του Astro ήταν η δυνατότητα να συνδυάσει στατικές σελίδες με διαδραστικά στοιχεία μέσω της υποστήριξης **Island Architecture**. Σε αντίθεση με παραδοσιακές μεθόδους όπου όλο το περιεχόμενο και οι διαδραστικές λειτουργίες φορτώνονται μαζί, το Astro επιτρέπει τη φόρτωση μόνο των απαραίτητων τμημάτων JavaScript για κάθε συγκεκριμένο στοιχείο της σελίδας. Αυτό σημαίνει ότι το περιεχόμενο παραμένει ελαφρύ, και η σελίδα φορτώνει γρήγορα, ενώ διαδραστικά στοιχεία όπως φόρμες ή μενού μπορούν να προστεθούν με τη χρήση React ή άλλων JavaScript frameworks, χωρίς να επιβαρύνουν το συνολικό φορτίο.

Επιπλέον, το Astro προσφέρει μεγάλη ευελιξία στην επιλογή εργαλείων και τεχνολογιών που μπορούν να χρησιμοποιηθούν για την κατασκευή του ιστοτόπου. Μια από τις βασικές καινοτομίες του είναι η υποστήριξη **multiple frontend frameworks**. Αυτό επιτρέπει στον προγραμματιστή να χρησιμοποιήσει διάφορες βιβλιοθήκες, όπως **React**, **Vue**, **Svelte**, ή ακόμα και **Preact**, για να προσθέσει διαδραστικά στοιχεία στον ιστότοπο, διατηρώντας ταυτόχρονα τα πλεονεκτήματα της στατικής απόδοσης. Αυτή η δυνατότητα ευνοεί έργα που απαιτούν την ενσωμάτωση διαφορετικών τύπων frontend τεχνολογιών, χωρίς να περιορίζεται ο προγραμματιστής σε μία μόνο προσέγγιση.

Στο συγκεκριμένο έργο, το Astro επιλέχθηκε για την ικανότητά του να διαχειρίζεται αποτελεσματικά μεγάλες ποσότητες περιεχομένου, καθώς και για την **ευκολία προσθήκης νέων σελίδων και περιεχομένου μέσω Markdown**. Οι σελίδες για τις ειδήσεις, τα έργα και τις δημοσιεύσεις δημιουργούνται με Markdown, το οποίο παρέχει έναν απλό αλλά αποτελεσματικό τρόπο διαχείρισης κειμένων και μεταδεδομένων, καθιστώντας εύκολη την ενημέρωση και προσθήκη νέου υλικού από τα μέλη της ερευνητικής ομάδας.

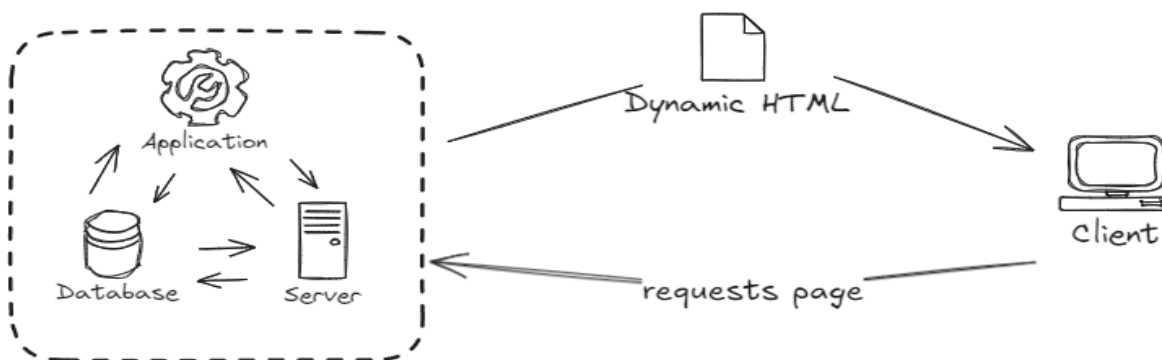
Ένα άλλο σημαντικό χαρακτηριστικό του Astro είναι ότι προσφέρει εξαιρετική υποστήριξη για **πολυγλωσσικές εφαρμογές** μέσω της ενσωμάτωσης του **i18n**. Στο πλαίσιο της ανάπτυξης του ιστοτόπου της ερευνητικής ομάδας, η υποστήριξη πολλαπλών γλωσσών κρίθηκε απαραίτητη, καθώς το περιεχόμενο του ιστότοπου πρέπει να είναι διαθέσιμο τόσο στα ελληνικά όσο και στα αγγλικά, διευρύνοντας την προσβασιμότητα και το κοινό στο οποίο απευθύνεται.

## Πώς Λειτουργεί το Static Site Generation (SSG) με το Astro

Πριν αναλύσουμε το πώς λειτουργεί το Astro ως Static Site Generator (SSG), είναι σημαντικό να εξετάσουμε το παραδοσιακό μοντέλο ανάπτυξης ιστοτόπων.

### Server Side Rendering (SSR)

Στο κλασικό μοντέλο, οι περισσότερες ιστοσελίδες χρησιμοποιούσαν δυναμικά συστήματα διαχείρισης περιεχομένου (CMS) όπως το WordPress. Αυτές οι πλατφόρμες βασίζονταν σε server-side rendering (SSR), όπου κάθε φορά που ένας χρήστης ζητούσε μια σελίδα, ο διακομιστής εκτελούσε εντολές για να συλλέξει περιεχόμενο από τη βάση δεδομένων, να το επεξεργαστεί μέσω ενός template engine, και να αποδώσει τη σελίδα σε HTML για τον τελικό χρήστη. Αυτή η διαδικασία συνέβαινε σε πραγματικό χρόνο για κάθε αίτημα του χρήστη, το οποίο σήμαινε ότι οι σελίδες αποδίδονταν δυναμικά. (patterns, n.d.)



Σχήμα 1: Διάγραμμα Server Side Rendering (SSG)

### Πλεονεκτήματα του Παραδοσιακού Μοντέλου

- Δυναμικό Περιεχόμενο σε Πραγματικό Χρόνο: Το δυναμικό rendering επέτρεπε σελίδες με δεδομένα που αλλάζουν συχνά, όπως φόρμες, ζωντανά στατιστικά ή social media feeds, να ενημερώνονται αυτόματα σε κάθε επίσκεψη.
- Ευελιξία: Οι δυναμικές σελίδες προσέφεραν ευελιξία, καθώς ο ιστότοπος μπορούσε να αλλάξει ανάλογα με το αίτημα του χρήστη (π.χ. διαλογή βάσει φίλτρων, αναζήτηση).

### Μειονεκτήματα του Παραδοσιακού Μοντέλου

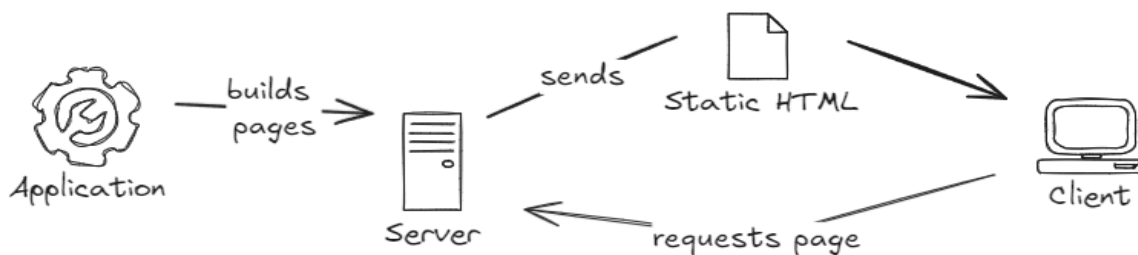
- Απόδοση: Οι ιστοσελίδες αυτές συχνά υστερούσαν σε απόδοση, καθώς κάθε σελίδα έπρεπε να δημιουργηθεί δυναμικά σε πραγματικό χρόνο, καθυστερώντας τη φόρτωση, ειδικά σε περιπτώσεις μεγάλου φόρτου επισκέψεων.

- Ασφάλεια: Η ύπαρξη βάσεων δεδομένων και server-side rendering αυξάνει τις πιθανότητες για επιθέσεις, όπως SQL injection ή hacking σε επίπεδο server.
- Κόστος και Συντήρηση: Η συντήρηση των δυναμικών ιστοτόπων ήταν πιο ακριβή και περίπλοκη, καθώς χρειαζόταν εξειδικευμένη υποστήριξη για τη διαχείριση των διακομιστών και των βάσεων δεδομένων.

#### Static Site Generation (SSG) και Astro

Το Static Site Generation (SSG) προσεγγίζει την ανάπτυξη ιστοσελίδων διαφορετικά, προσφέροντας στατικές σελίδες που προ-αποδίδονται κατά τη διαδικασία της κατασκευής του ιστοτόπου. Με αυτή τη μέθοδο, όλες οι σελίδες του ιστοτόπου δημιουργούνται κατά τη διάρκεια του build process και εξάγονται ως στατικά αρχεία HTML, CSS και JavaScript. Αυτά τα αρχεία φιλοξενούνται σε έναν static server και αποδίδονται άμεσα στον χρήστη, χωρίς την ανάγκη για δυναμικό rendering από τον διακομιστή.

Το Astro είναι ένα μοντέρνο Static Site Generator που ακολουθεί αυτή τη λογική και προσφέρει επιπλέον δυνατότητες όπως τη χρήση Island Architecture για τη φόρτωση JavaScript μόνο όταν είναι απαραίτητο. Το Astro κατά τη διάρκεια του build process παίρνει τις σελίδες Markdown και Front Matter δεδομένα και δημιουργεί στατικές HTML σελίδες που μπορούν να φορτωθούν άμεσα από τον χρήστη. (patterns, n.d.)



Σχήμα 2: Διάγραμμα Static Site Generation (SSG)

## Πλεονεκτήματα του Static Site Generation

- **Εξαιρετική Απόδοση:** Οι σελίδες είναι ήδη προ-αποδομένες και σε στατική μορφή, επιτρέποντας ταχύτερη φόρτωση. Οι σελίδες εξυπηρετούνται απευθείας από CDN ή απλό web server, εξαλείφοντας την ανάγκη για δυναμική επεξεργασία.
- **Ασφάλεια:** Η απουσία βάσεων δεδομένων και δυναμικού server-side rendering μειώνει σημαντικά τα σημεία που μπορούν να εκμεταλλευτούν οι επιτιθέμενοι, καθιστώντας τον ιστότοπο πιο ασφαλή.
- **Κλιμάκωση:** Ένας στατικός ιστότοπος μπορεί να κλιμακωθεί πολύ εύκολα και να διανεμηθεί μέσω δικτύων παράδοσης περιεχομένου (CDN), προσφέροντας ταχύτατη απόδοση ακόμη και με υψηλό φόρτο επισκέψεων.
- **Απλή Συντήρηση:** Δεν απαιτείται διαρκής συντήρηση server ή βάση δεδομένων, μειώνοντας το κόστος και την πολυπλοκότητα της υποδομής.

## Μειονεκτήματα του Static Site Generation

- **Περιορισμένη Δυναμικότητα:** Εφόσον οι σελίδες δημιουργούνται κατά το build process, οι ιστοσελίδες αυτές δεν μπορούν να αλλάξουν δυναμικά σε πραγματικό χρόνο, εκτός αν χρησιμοποιηθούν API ή άλλες τεχνικές για τη διαχείριση δεδομένων που αλλάζουν συχνά.
- **Μεγάλες Ιστοσελίδες:** Η παραγωγή στατικών αρχείων μπορεί να αυξήσει τον χρόνο build για ιστοσελίδες με χιλιάδες σελίδες, καθώς όλα τα αρχεία πρέπει να δημιουργηθούν εκ των προτέρων.

## Astro και Island Architecture

Ένα από τα μεγαλύτερα πλεονεκτήματα του Astro είναι η καινοτομία του με τη χρήση του **Island Architecture**, το οποίο επιτρέπει τη στατική απόδοση σελίδων με τη δυνατότητα προσθήκης JavaScript μόνο στα διαδραστικά στοιχεία που χρειάζονται, αντί να φορτώνεται JavaScript για ολόκληρη τη σελίδα. Αυτό σημαίνει ότι το μεγαλύτερο μέρος της σελίδας αποδίδεται γρήγορα, ενώ τα διαδραστικά στοιχεία (όπως μενού ή φόρμες) φορτώνονται μόνο όταν είναι απαραίτητο, βελτιώνοντας έτσι την απόδοση. (patterns, n.d.)

Αυτή η προσέγγιση λύνει ένα από τα πιο συχνά προβλήματα που αντιμετωπίζουν οι προγραμματιστές με σύγχρονες web εφαρμογές: την απόδοση. Σε παραδοσιακές σελίδες που κατασκευάζονται με JavaScript frameworks (όπως React ή Vue), συνήθως όλη η σελίδα γίνεται render μέσω JavaScript, ακόμα και αν κάποια στοιχεία της σελίδας δεν χρειάζονται διαδραστικότητα. Αυτό μπορεί να οδηγήσει σε αργή φόρτωση και περιττή χρήση πόρων, καθώς το σύστημα αποδίδει JavaScript για ολόκληρη τη σελίδα, ακόμη και για στατικά στοιχεία.

Η αρχιτεκτονική **Island Architecture** επιλύει αυτό το πρόβλημα με έναν έξυπνο και αποτελεσματικό τρόπο. Στο Astro, οι σελίδες αντιμετωπίζονται κυρίως ως **στατικές HTML**, που αποδίδονται πολύ

γρήγορα από τον browser. Όταν χρειάζεται, όμως, κάποια συγκεκριμένα τμήματα της σελίδας (όπως φόρμες, μενού, ή δυναμικά στοιχεία) μπορούν να αποδοθούν με JavaScript, χωρίς να επηρεάζεται το υπόλοιπο της σελίδας.

Αυτά τα διαδραστικά τμήματα της σελίδας ονομάζονται **"νησιά"** (islands). Μόνο αυτά τα "νησιά" φορτώνουν JavaScript, και μόνο όταν είναι πραγματικά αναγκαία. Έτσι, το Astro αποφεύγει την άσκοπη φόρτωση μεγάλων ποσοτήτων JavaScript και επιτρέπει στις σελίδες να παραμένουν ελαφριές και γρήγορες, ενώ ταυτόχρονα παρέχει τη δυνατότητα προσθήκης διαδραστικών στοιχείων.

#### Πλεονεκτήματα της Island Architecture:

- **Βελτιωμένη Απόδοση:** Με τη φόρτωση JavaScript μόνο για τα διαδραστικά μέρη της σελίδας, μειώνεται το συνολικό βάρος της σελίδας και αυξάνεται η ταχύτητα απόκρισης, ειδικά σε κινητές συσκευές ή σε περιοχές με αργή σύνδεση στο διαδίκτυο.
- **Αποδοτική Χρήση Πόρων:** Αντί να χρησιμοποιεί JavaScript για να αποδώσει όλη τη σελίδα, το Astro χρησιμοποιεί HTML για την απόδοση στατικού περιεχομένου και προσθέτει JavaScript μόνο όπου είναι αναγκαίο, κάνοντας αποδοτικότερη χρήση των πόρων του browser.
- **Ευελιξία:** Το Astro επιτρέπει στον προγραμματιστή να χρησιμοποιήσει διάφορες βιβλιοθήκες JavaScript (React, Vue, Svelte, κλπ.) για τα "νησιά", χωρίς να επιβάλλει έναν συγκεκριμένο τρόπο διαχείρισης της διαδραστικότητας.

#### Πώς δουλεύει στην πράξη:

1. **Κατά το build:** Όταν γίνεται το build της ιστοσελίδας, το Astro αποδίδει τις σελίδες ως στατικό HTML περιεχόμενο.
2. **Προσθήκη JavaScript:** Στα τμήματα της σελίδας που έχουν οριστεί ως "νησιά" (π.χ., μια φόρμα ή ένα διαδραστικό μενού), η JavaScript φορτώνεται μόνο όταν χρειάζεται, και όχι για όλη τη σελίδα.
3. **Διαδραστικότητα:** Μόνο αυτά τα διαδραστικά στοιχεία επηρεάζονται από τη JavaScript, διατηρώντας έτσι την απόδοση της υπόλοιπης σελίδας εξαιρετικά γρήγορη.

Ένα παράδειγμα χρήσης αυτής της αρχιτεκτονικής είναι μια σελίδα που περιέχει άρθρα και μια δυναμική αναζήτηση. Αντί να φορτώνεται όλη η σελίδα με JavaScript, τα άρθρα μπορούν να αποδοθούν ως απλό HTML και μόνο η αναζήτηση να είναι διαδραστική, χρησιμοποιώντας JavaScript στο "νησί" της αναζήτησης.

## 2.2 Υποστηρικτικά Εργαλεία και Τεχνολογίες

Για να επιτευχθεί η ανάπτυξη ενός αποδοτικού και εύχρηστου ιστοτόπου, η επιλογή των κατάλληλων εργαλείων και τεχνολογιών ήταν κρίσιμη. Η σωστή συνεργασία των εργαλείων αυτών βελτίωσε όχι μόνο την ταχύτητα ανάπτυξης, αλλά και τη δυνατότητα συντήρησης και ενημέρωσης του περιεχομένου από την ερευνητική ομάδα. Ακολουθούν τα κύρια υποστηρικτικά εργαλεία και τεχνολογίες που χρησιμοποιήθηκαν:

### 1. Markdown και Front Matter

Το **Markdown** είναι μια απλή και ελαφριά γλώσσα σήμανσης που επιτρέπει την εύκολη δημιουργία και διαχείριση περιεχομένου, χωρίς την ανάγκη χρήσης HTML. Η επιλογή του Markdown για τη διαχείριση του περιεχομένου στον ιστότοπο ήταν ιδανική, καθώς επιτρέπει στα μέλη της ερευνητικής ομάδας να προσθέτουν ή να ενημερώνουν άρθρα, δημοσιεύσεις και ερευνητικά έργα με ελάχιστη τεχνική γνώση. Κάθε αρχείο Markdown συνοδεύεται από **Front Matter**, το οποίο χρησιμοποιείται για την προσθήκη μεταδεδομένων όπως τίτλος, συγγραφέας, ημερομηνία δημοσίευσης και κατηγορίες. (Markdown Guide, n.d.)

Το **Front Matter** είναι ένα σύνολο μεταδεδομένων που τοποθετείται στην αρχή των αρχείων Markdown και βοηθά στην οργάνωση και τη διαχείριση του περιεχομένου. Στο έργο αυτό, το Front Matter χρησιμοποιήθηκε για να αποθηκεύσει βασικές πληροφορίες για κάθε σελίδα ή άρθρο, όπως ο τίτλος, η ημερομηνία δημοσίευσης, οι συγγραφείς, και οι ετικέτες (tags). Αυτή η δομημένη πληροφορία επιτρέπει την εύκολη αναζήτηση και ταξινόμηση περιεχομένου.

**Πλεονεκτήματα του Markdown:**

- **Απλότητα:** Το Markdown είναι εύκολο να μάθει και να χρησιμοποιηθεί, καθιστώντας το κατάλληλο για μη προγραμματιστές.
- **Δομή Περιεχομένου:** Χρησιμοποιώντας Front Matter, το περιεχόμενο μπορεί να οργανωθεί και να κατηγοριοποιηθεί εύκολα, επιτρέποντας γρήγορη αναζήτηση και απόδοση στον ιστότοπο.
- **Ευελιξία:** Παρέχει ευελιξία στη μορφοποίηση του κειμένου, χωρίς την πολυπλοκότητα του HTML.

### 2. React

Παρά το γεγονός ότι το Astro παράγει στατικό HTML περιεχόμενο, το **React** χρησιμοποιήθηκε για την ενσωμάτωση διαδραστικών στοιχείων στον ιστότοπο. Η δυνατότητα ενσωμάτωσης React μέσω της **Island Architecture** του Astro επιτρέπει την προσθήκη διαδραστικών στοιχείων, όπως φόρμες, dropdown μενού, και άλλες λειτουργίες που απαιτούν τη χρήση JavaScript, χωρίς να επιβαρύνεται η συνολική απόδοση της σελίδας. (React, n.d.)

#### Χρήση του React στο έργο:

- **Διαδραστικά Στοιχεία:** Ορισμένα τμήματα της ιστοσελίδας απαιτούν διαδραστικότητα (π.χ. φόρμες, μενού, αναζητήσεις), και το React επιτρέπει την υλοποίηση αυτών των λειτουργιών με απλό και αποδοτικό τρόπο.
- **Απομόνωση Λειτουργιών:** Με την Island Architecture, μόνο τα μέρη της σελίδας που χρειάζονται διαδραστικότητα φορτώνουν JavaScript, επιτρέποντας στον υπόλοιπο ιστότοπο να παραμένει στατικός και γρήγορος.

### 3. Tailwind CSS

Για την εμφάνιση και τη μορφοποίηση του ιστοτόπου, χρησιμοποιήθηκε το **Tailwind CSS**, ένα utility-first πλαίσιο για τον σχεδιασμό CSS. Το Tailwind CSS προσφέρει έτοιμες κλάσεις, οι οποίες επιτρέπουν τη γρήγορη και αποδοτική διαμόρφωση της εμφάνισης του ιστοτόπου, χωρίς να απαιτείται η συγγραφή προσαρμοσμένων CSS από το μηδέν. (Tailwind CSS, n.d.)

#### Πλεονεκτήματα του Tailwind CSS:

- **Ταχύτητα Ανάπτυξης:** Με την χρήση έτοιμων κλάσεων, η μορφοποίηση των στοιχείων γίνεται γρήγορα, χωρίς να χρειάζεται η δημιουργία νέων CSS κανόνων.
- **Συνεπής Σχεδιασμός:** Το Tailwind επιτρέπει τη χρήση ενός συνεπούς σχεδιαστικού συστήματος σε όλο τον ιστότοπο, διασφαλίζοντας ομοιόμορφη εμφάνιση σε όλες τις σελίδες.
- **Προσαρμοστικότητα:** Με τις έτοιμες responsive κλάσεις, η δημιουργία ιστοτόπου φιλικού προς κινητές συσκευές γίνεται εύκολα και χωρίς επιπλέον προσπάθεια.

### 4. Shaden UI

Το **Shaden UI** χρησιμοποιήθηκε ως βιβλιοθήκη έτοιμων UI components για την κατασκευή του ιστοτόπου. Αυτή η βιβλιοθήκη παρέχει ένα σύνολο προσαρμόσιμων στοιχείων, τα οποία μπορούν εύκολα να ενσωματωθούν σε ένα project, προσφέροντας ταυτόχρονα ένα συνεπές και επαγγελματικό design.

Ένα από τα βασικά πλεονεκτήματα του **Shaden UI** είναι ότι τα components που προσφέρει είναι **προσβάσιμα** (accessible). Αυτό σημαίνει ότι συμμορφώνονται με τις βέλτιστες πρακτικές προσβασιμότητας (WCAG guidelines), επιτρέποντας την εύκολη χρήση των στοιχείων από χρήστες με αναπηρίες, όπως χρήστες που χρησιμοποιούν screen readers ή πλοηγούνται μέσω πληκτρολογίου.

Το **Shaden** επικεντρώνεται στη δυνατότητα γρήγορης δημιουργίας **UI components** που είναι σχεδιασμένα με γνώμονα την επαναχρησιμοποίηση και την προσαρμοστικότητα. Σε αυτό το έργο, χρησιμοποιήθηκε για να απλοποιήσει την κατασκευή διαδραστικών στοιχείων όπως μενού, καρτέλες, και πλοηγήσεις (navigations), επιτρέποντας γρήγορη ανάπτυξη με ένα συνεκτικό και αισθητικά ευχάριστο αποτέλεσμα. (shaden, n.d.)

#### Πλεονεκτήματα του Shaden UI:

- **Προσαρμοστικότητα:** Παρότι προσφέρει έτοιμα components, τα στοιχεία του Shaden UI είναι

ιδιαίτερα προσαρμοσίμα, επιτρέποντας εύκολες αλλαγές στην εμφάνιση και τη λειτουργικότητα.

- **Συνεπής Σχεδιαστική Γλώσσα:** Το Shaden παρέχει ένα ενιαίο design system, διασφαλίζοντας ότι όλα τα UI elements έχουν κοινή αισθητική και αλληλεπίδραση, βελτιώνοντας την εμπειρία χρήστη.
- **Προσβασιμότητα:** Τα components του Shaden UI είναι κατασκευασμένα με γνώμονα την προσβασιμότητα, εξασφαλίζοντας ότι τα HTML στοιχεία συμμορφώνονται με τα διεθνή πρότυπα, όπως το WCAG. Αυτό καθιστά τον ιστότοπο εύχρηστο και φιλικό για όλους τους χρήστες, συμπεριλαμβανομένων των χρηστών με αναπηρίες.
- **Γρήγορη Ανάπτυξη:** Η χρήση έτοιμων components επιταχύνει την ανάπτυξη της διεπαφής χρήστη, επιτρέποντας στους προγραμματιστές να εστιάσουν στη λειτουργικότητα και τη διαδραστικότητα.

Το Shaden UI συνεργάζεται άριστα με τα υπόλοιπα εργαλεία που χρησιμοποιήθηκαν (όπως το Tailwind CSS), επιτρέποντας την εύκολη και γρήγορη προσαρμογή των UI components σε διαφορετικές ανάγκες και απαιτήσεις του ιστότοπου.

## 5. i18n (Internationalization)

Ένα από τα σημαντικότερα χαρακτηριστικά που ενσωματώθηκαν στον ιστότοπο είναι η υποστήριξη για **πολυγλωσσικότητα** μέσω του **i18n**. Η δυνατότητα αυτή επιτρέπει την εμφάνιση του περιεχομένου σε διαφορετικές γλώσσες, καθιστώντας τον ιστότοπο προσβάσιμο σε ευρύτερο κοινό. Οι μεταφράσεις αποθηκεύονται σε configuration files, τα οποία μπορούν να επεξεργαστούν από τα μέλη της ομάδας. (MDN, 2024) (Astro i18n, n.d.)

### Λειτουργία του i18n:

- **Αποθήκευση Μεταφράσεων:** Οι μεταφράσεις για τις διάφορες γλώσσες αποθηκεύονται σε ξεχωριστά αρχεία, κάνοντας την προσθήκη νέων γλωσσών ή την ενημέρωση των υφιστάμενων κειμένων εύκολη και γρήγορη.
- **Διαχείριση Πολλαπλών Γλωσσών:** Το i18n επιτρέπει την εμφάνιση διαφορετικών εκδοχών του περιεχομένου σε πολλές γλώσσες, εξασφαλίζοντας ότι ο ιστότοπος μπορεί να προσεγγίσει διεθνές κοινό.

## 2.3 Πλεονεκτήματα και Μειονεκτήματα

### Πλεονεκτήματα

#### 1. Υψηλή Απόδοση

Η χρήση του **Astro Framework** για τη δημιουργία στατικών σελίδων εξασφαλίζει γρήγορη απόδοση του περιεχομένου, καθώς οι σελίδες δημιουργούνται και προ-αποδίδονται κατά τη διαδικασία του build, αποφεύγοντας το runtime rendering. Αυτό επιτρέπει στον ιστότοπο να φορτώνει γρήγορα, ειδικά όταν

εξυπηρετείται από CDN, μειώνοντας τους χρόνους απόκρισης και βελτιώνοντας τη συνολική εμπειρία χρήστη.

## 2. Ασφάλεια

Η στατική φύση του Astro μειώνει την επιφάνεια επίθεσης, καθώς δεν υπάρχουν δυναμικές συνδέσεις με βάσεις δεδομένων ή server-side κώδικα που να επεξεργάζεται αιτήματα σε πραγματικό χρόνο. Έτσι, οι κίνδυνοι από επιθέσεις όπως SQL injection ή server hacking είναι ελάχιστοι, κάνοντας τον ιστότοπο ασφαλέστερο.

## 3. Ευκολία Διαχείρισης Περιεχομένου

Χάρη στη χρήση **Markdown** και **Front Matter**, η προσθήκη νέου περιεχομένου (άρθρα, δημοσιεύσεις, έργα) είναι απλή και προσιτή στα μέλη της ερευνητικής ομάδας, χωρίς να απαιτούνται τεχνικές γνώσεις. Τα αρχεία Markdown επιτρέπουν εύκολη δημιουργία και ενημέρωση περιεχομένου, ενώ το Front Matter επιτρέπει την αποθήκευση δομημένων μεταδεδομένων που κάνουν το περιεχόμενο οργανωμένο και εύκολα προσβάσιμο.

## 4. Διαδραστικότητα μέσω React

Η δυνατότητα χρήσης του **React** για διαδραστικά στοιχεία μέσω της **Island Architecture** του Astro επιτρέπει τη δημιουργία δυναμικών, φιλικών προς τον χρήστη εμπειριών. Τα διαδραστικά τμήματα της σελίδας (όπως φόρμες και μενού) φορτώνονται μόνο όταν χρειάζεται, βελτιώνοντας έτσι την απόδοση του ιστότοπου.

## 5. Προσαρμοστικός Σχεδιασμός μέσω Tailwind CSS

Η χρήση του **Tailwind CSS** για την ανάπτυξη του σχεδιασμού της ιστοσελίδας επιτρέπει την ταχύτατη δημιουργία UI με έτοιμες κλάσεις CSS. Η ευελιξία και η προσαρμοστικότητα του Tailwind διευκολύνουν τη δημιουργία responsive και consistent σχεδίων, βελτιώνοντας την εμπειρία του χρήστη σε κάθε συσκευή.

## 6. Προσβασιμότητα μέσω Shaden UI

Η χρήση του **Shaden UI** εξασφαλίζει ότι όλα τα components είναι προσβάσιμα και συμβατά με τα πρότυπα WCAG (W3C, n.d.), επιτρέποντας σε χρήστες με αναπηρίες να πλοηγούνται και να χρησιμοποιούν τον ιστότοπο χωρίς προβλήματα. Αυτό διασφαλίζει ότι ο ιστότοπος είναι φιλικός προς όλους, ανεξαρτήτως των φυσικών ή τεχνολογικών περιορισμών που μπορεί να αντιμετωπίζουν οι χρήστες.

## 7. Υποστήριξη Πολυγλωσσικότητας (i18n)

Το **i18n** επιτρέπει την εύκολη διαχείριση και μετάφραση περιεχομένου σε πολλαπλές γλώσσες. Αυτό καθιστά τον ιστότοπο προσβάσιμο σε ένα διεθνές κοινό, ενώ τα configuration files επιτρέπουν στα μέλη της ομάδας να προσθέτουν νέες μεταφράσεις χωρίς να απαιτείται αλλαγή του κώδικα.

### Μειονεκτήματα

#### 1. Περιορισμένη Δυναμικότητα

Αν και το **Astro** προσφέρει εξαιρετική απόδοση ως Static Site Generator, ενδέχεται να μην είναι κατάλληλο για περιπτώσεις όπου απαιτείται δυναμικό περιεχόμενο σε πραγματικό χρόνο (π.χ., ζωντανές ενημερώσεις ή περιεχόμενο που αλλάζει συνεχώς). Για τέτοιες περιπτώσεις, χρειάζονται επιπλέον λύσεις, όπως η χρήση εξωτερικών APIs ή serverless functions.

#### 2. Χρόνος Build για Πολύ Μεγάλες Ιστοσελίδες

Για ιστότοπους με χιλιάδες σελίδες, ο χρόνος build μπορεί να αυξηθεί σημαντικά, καθώς όλες οι σελίδες πρέπει να δημιουργηθούν κατά τη διαδικασία του build. Αν και αυτό δεν αποτελεί πρόβλημα για μικρούς ή μεσαίου μεγέθους ιστότοπους, σε πολύ μεγάλες εφαρμογές μπορεί να χρειαστεί επιπλέον βελτιστοποίηση.

#### 3. Απαιτεί Τεχνική Κατάρτιση για Προχωρημένες Προσαρμογές

Παρόλο που η διαχείριση περιεχομένου μέσω Markdown είναι απλή, οποιεσδήποτε σημαντικές αλλαγές στη λειτουργικότητα του ιστότοπου ή προσθήκη νέων δυνατοτήτων απαιτούν γνώση προγραμματισμού και εξειδίκευση στις τεχνολογίες που χρησιμοποιούνται, όπως το React ή το Tailwind CSS.

#### 4. Εξάρτηση από Configuration Files για Μεταφράσεις

Ενώ το **i18n** απλοποιεί τη διαχείριση πολλαπλών γλωσσών, η αποθήκευση των μεταφράσεων σε configuration files μπορεί να οδηγήσει σε δυσκολίες συντήρησης, εάν τα αρχεία αυτά γίνουν μεγάλα και περίπλοκα. Σε τέτοιες περιπτώσεις, ίσως χρειαστεί ένα σύστημα πιο εξελιγμένης διαχείρισης μεταφράσεων.

### 3. Ανάπτυξη του Έργου

#### 3.1. Επιλογή του Astro Framework

Η επιλογή του Astro Framework ήταν μια από τις πρώτες και πιο κρίσιμες αποφάσεις για την ανάπτυξη του ιστοτόπου της ερευνητικής ομάδας. Το Astro ξεχωρίζει ως ένα στατικό site generator (SSG) που προσφέρει εξαιρετικά γρήγορη απόδοση και απλότητα στη διαχείριση του περιεχομένου. Ένα από τα κύρια πλεονεκτήματα του Astro είναι ότι παρέχει τη δυνατότητα δημιουργίας στατικών σελίδων, οι οποίες μπορούν να εξυπηρετηθούν απευθείας από CDN, καθιστώντας το ιδανικό για ιστότοπους με μεγάλο όγκο περιεχομένου, όπως αυτός της ερευνητικής ομάδας.

Το Astro επιτρέπει επίσης τη χρήση διαφορετικών frontend frameworks, όπως React, Vue, ή Svelte, μέσα στο ίδιο έργο, γεγονός που προσφέρει μεγάλη ευελιξία στην προσθήκη διαδραστικών στοιχείων όπου είναι απαραίτητο. Η αρχιτεκτονική Island Architecture (patterns, n.d.) του Astro είναι επίσης ιδιαίτερα σημαντική, καθώς επιτρέπει τη φόρτωση JavaScript μόνο στα στοιχεία της σελίδας που το χρειάζονται, διατηρώντας έτσι την απόδοση της υπόλοιπης σελίδας εξαιρετικά γρήγορη.

Ένας άλλος βασικός παράγοντας για την επιλογή του Astro ήταν η υποστήριξη για πολυγλωσσικότητα (i18n). Η ενσωμάτωση του i18n επιτρέπει την εύκολη διαχείριση μεταφράσεων, κάτι που ήταν απαραίτητο για τον ιστότοπο, καθώς το περιεχόμενο θα είναι διαθέσιμο τόσο στα ελληνικά όσο και στα αγγλικά.

#### 3.2. Εξέταση CMS Λύσεων για Διαχείριση Περιεχομένου

Αφού έγινε η επιλογή του Astro, το επόμενο βήμα ήταν να εξετάσω λύσεις που θα μπορούσαν να υποστηρίξουν την εύκολη διαχείριση του περιεχομένου. Καθώς το Astro υποστηρίζει τη χρήση Markdown για τη δημιουργία στατικών σελίδων, αναζητήθηκαν CMS λύσεις που θα μπορούσαν να συνδυαστούν με το Astro και να προσφέρουν ένα περιβάλλον διαχείρισης περιεχομένου πιο προσβάσιμο στα μέλη της ερευνητικής ομάδας.

Αρκετές λύσεις CMS αξιολογήθηκαν κατά τη διάρκεια αυτής της διαδικασίας:

- **Keystatic:** Αυτό το CMS φάνηκε ιδιαίτερα ελκυστικό, καθώς ενσωματώνεται άριστα με το Astro και προσφέρει ένα φιλικό UI για τη δημιουργία περιεχομένου σε μορφή Markdown. Το Keystatic επιτρέπει την εύκολη διαχείριση **content collections**, καθιστώντας το ιδανικό για τη διαχείριση πολλαπλών κατηγοριών περιεχομένου όπως νέα, πρότζεκτ και δημοσιεύσεις. Ωστόσο, η **έλλειψη υποστήριξης για internalization** το καθιστούν λιγότερο κατάλληλο για τις ανάγκες του έργου, δεδομένου ότι η πολυγλωσσικότητα ήταν απαραίτητη. (Keystatic, n.d.)
- **Directus:** Το Directus προσέφερε εξαιρετική ευελιξία στη διαμόρφωση της δομής του περιεχομένου και υποστήριζε πλήρως το **internalization**. Παρείχε επίσης ένα εξαιρετικά ισχυρό σύστημα διαχείρισης δεδομένων, επιτρέποντας στους χρήστες να ορίσουν προσαρμοσμένες σχέσεις μεταξύ των δεδομένων. Ωστόσο, το Directus είχε ένα **μεγάλο learning curve** και απαιτούσε σημαντική προσαρμογή, κάτι που το καθιστούν πιο περίπλοκο από τις ανάγκες του

έργου. (Directus, n.d.)

- **Payload CMS:** Το **Payload CMS** προσέφερε πλήρη υποστήριξη για internalization και είχε όλες τις δυνατότητες που απαιτούνταν για τη διαχείριση περιεχομένου. Παρόλα αυτά, το Payload CMS, όπως και το Directus, είχε ένα μεγάλο learning curve, καθιστώντας το δύσκολο να ενσωματωθεί χωρίς να απαιτηθεί σημαντική επένδυση χρόνου και μάθησης. (Payload, n.d.)
- **Front Matter CMS:** Το **Front Matter CMS** ήταν μια από τις πιο υποσχόμενες λύσεις που βρέθηκαν κατά τη διάρκεια της αναζήτησης. Το συγκεκριμένο CMS συνδυάζει την ευελιξία του Markdown με ένα UI που ενσωματώνεται απευθείας στο **VSCode code editor**, επιτρέποντας στους χρήστες να διαχειρίζονται το περιεχόμενο με ευκολία. Το Front Matter CMS επιτρέπει τη διαμόρφωση της δομής του περιεχομένου μέσω configuration files και παρέχει plugin για διαχείριση περιεχομένου μέσα στον editor μέσω ενός UI. Αυτό το καθιστά ιδανικό για το project. Ωστόσο, η ανακάλυψή του έγινε αργά στην ανάπτυξη, και έτσι δεν υπήρχε αρκετός χρόνος για την πλήρη ενσωμάτωσή του. (Struyf, 2024)

### 3.3. Ορισμός Περιεχομένου και Δημιουργία Σχήματος

Κατά την ανάπτυξη του έργου, ορίστηκαν διάφορες κατηγορίες περιεχομένου, και δημιουργήθηκαν **schemas** για κάθε τύπο περιεχομένου. Τα schemas αυτά διασφαλίζουν την οργανωμένη διαχείριση του περιεχομένου και τη σαφή δομή για την αποθήκευση και εμφάνισή του στον ιστότοπο. Ακολουθούν οι βασικές κατηγορίες περιεχομένου και τα πεδία τους, όπως έχουν οριστεί στο project.

#### 1. Συντάκτες (Authors)

Το σχήμα για τους συγγραφείς ή τα μέλη της ερευνητικής ομάδας περιλαμβάνει τα εξής πεδία:

- **Email (email):** Το email του συντάκτη.
- **Όνομα (firstname):** Το όνομα του συντάκτη.
- **Επίθετο (lastname):** Το επώνυμο του συντάκτη.
- **Avata (avatar):** Μια προαιρετική εικόνα προφίλ του συντάκτη.

Οι συντάκτες σχετίζονται με άρθρα, έργα, εκπαίδευση και εργασιακή εμπειρία.

## 2. Άρθρα Νέων (News)

Τα άρθρα ειδήσεων δημοσιεύονται με τις παρακάτω πληροφορίες:

- **Τίτλος (title):** Ο τίτλος του άρθρου.
- **Σύνοψη (summary):** Μια σύντομη περίληψη του άρθρου.
- **Κατάσταση (status):** Ορίζει αν το άρθρο είναι σε προσχέδιο ή δημοσιευμένο.
- **Ημερομηνία Δημοσίευσης (publishedAt):** Η ημερομηνία κατά την οποία δημοσιεύεται το άρθρο.
- **Εξωτερική Πηγή (source):** Μια πηγή που συνδέεται με το άρθρο, περιλαμβάνοντας όνομα και URL.
- **Ετικέτες (tags):** Προαιρετική λίστα με κατηγορίες ή λέξεις-κλειδιά που σχετίζονται με το άρθρο.
- **Εξώφυλλο (cover):** Προαιρετική εικόνα εξωφύλλου.

## 3. Έργα (Projects)

Τα ερευνητικά έργα της ομάδας περιλαμβάνουν τα παρακάτω πεδία:

- **Τίτλος (title):** Ο τίτλος του έργου.
- **Σύνοψη (summary):** Μια σύντομη περιγραφή του έργου.
- **Κατάσταση (status):** Η κατάσταση του έργου (προσχέδιο ή δημοσιευμένο).
- **Συγγραφείς (authors):** Λίστα με τα μέλη της ομάδας που συμμετέχουν στο έργο.
- **Εξωτερικοί Συγγραφείς (externalAuthors):** Προαιρετικά εξωτερικά άτομα ή ομάδες που συμμετείχαν στο έργο.
- **Αποθετήριο (repo):** URL προς τον κώδικα του έργου (αν υπάρχει).
- **Demo (demo):** URL προς την παρουσίαση ή demo του έργου.
- **Ετικέτες (tags):** Προαιρετικές κατηγορίες ή λέξεις-κλειδιά.

## 4. Δημοσιεύσεις (Publications)

Οι δημοσιεύσεις περιλαμβάνουν ακαδημαϊκές εργασίες και άρθρα της ομάδας, με τα παρακάτω πεδία:

- **Τίτλος (title):** Ο τίτλος της δημοσίευσης.
- **Σύνοψη (summary):** Σύντομη περιγραφή της δημοσίευσης.
- **Κατάσταση (status):** Αν η δημοσίευση είναι προσχέδιο ή δημοσιευμένη.
- **Ημερομηνία Δημοσίευσης (publishedAt):** Η ημερομηνία δημοσίευσης της εργασίας.
- **Συγγραφείς (authors):** Λίστα με τα μέλη της ομάδας που συμμετείχαν στη δημοσίευση.
- **Εξωτερικοί Συγγραφείς (externalAuthors):** Προαιρετικά εξωτερικοί συγγραφείς ή συνεργάτες.
- **Παράθεση (Cite):** Παράθεση της δημοσίευσης (σε μορφή BibTeX ή άλλη).
- **Σύνδεσμοι (Links):** Προαιρετικοί σύνδεσμοι προς εξωτερικές πηγές ή πληροφορίες.
- **Ετικέτες (tags):** Προαιρετικές κατηγορίες ή λέξεις-κλειδιά.

## 5. Εκπαίδευση (Education) και Εργασιακή Εμπειρία (Work Experience)

Αυτά τα schemas συνδέονται με τα μέλη της ομάδας και περιλαμβάνουν:

- **Τίτλος (title):** Ο τίτλος του πτυχίου ή της θέσης.
- **Ινστιτούτο/Εταιρεία (institute/company):** Το όνομα του εκπαιδευτικού ιδρύματος ή της εταιρείας.
- **Συγγραφέας (author):** Το μέλος της ομάδας που συμμετείχε στην εκπαίδευση ή την εμπειρία.
- **Ημερομηνία Έναρξης (startDate):** Η ημερομηνία έναρξης.
- **Ημερομηνία Λήξης (endDate):** Η ημερομηνία λήξης.

#### 6. Νομικές Σελίδες (Legal Pages)

Οι νομικές σελίδες περιλαμβάνουν περιεχόμενο που σχετίζεται με νομικές πληροφορίες ή πολιτικές της ερευνητικής ομάδας. Αυτές μπορεί να περιλαμβάνουν τη σελίδα όρων χρήσης, την πολιτική απορρήτου ή άλλες σχετικές πληροφορίες.

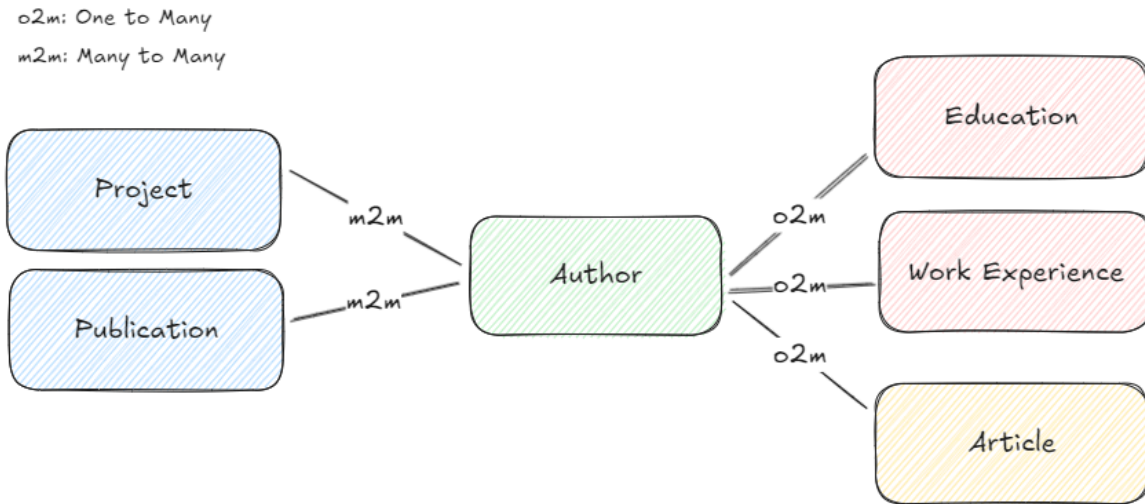
Το σχήμα για τις νομικές σελίδες περιλαμβάνει τα εξής πεδία:

- **Τίτλος (Title):** Ο τίτλος της νομικής σελίδας.
- **Ημερομηνία (Date):** Η ημερομηνία που ισχύει η συγκεκριμένη νομική πολιτική ή έγγραφο.

#### Σχέσεις Μεταξύ Περιεχομένου

Οι σχέσεις μεταξύ των διαφορετικών τύπων περιεχομένου είναι σημαντικές για τη διασύνδεση δεδομένων, όπως:

- Οι **συγγραφείς** σχετίζονται με **έργα**, **δημοσιεύσεις**, και **άρθρα νέων**, όπως επίσης με **εκπαίδευση** και **εργασιακή εμπειρία**.
- Τα **έργα** μπορούν να έχουν συγγραφείς και εξωτερικούς συνεργάτες.



Σχήμα 3: Διάγραμμα Entity Relationship (ER)

### 3.4. Ενσωμάτωση Πρόσθετων Εργαλείων: Tailwind CSS, React, και Shaden UI

Αφού οριστικοποιήθηκε η δομή του περιεχομένου και τα schemas, το επόμενο βήμα ήταν η ενσωμάτωση των εργαλείων για τον σχεδιασμό και τη διαδραστικότητα του ιστότοπου. Το έργο χρειαζόταν ένα ισχυρό εργαλείο για τον σχεδιασμό των UI components, καθώς και τη δυνατότητα προσθήκης διαδραστικών στοιχείων.

#### 1. Ενσωμάτωση Tailwind CSS

Το **Tailwind CSS** ήταν η πρώτη προσθήκη στο project. Το Tailwind επιλέχθηκε λόγω της ευελιξίας και της ταχύτητας που προσφέρει στη δημιουργία και προσαρμογή του design. Με το **utility-first** σύστημα του Tailwind, ήταν εύκολο να δημιουργήσω responsive σελίδες και να διατηρήσω μια συνεπή σχεδιαστική γλώσσα σε ολόκληρο τον ιστότοπο.

Στο άστρο είναι πολύ εύκολη η ενσωμάτωση της tailwind. Αρκεί να εκτελέσεις μια απλή εντολή:

```
npx astro add tailwind
```

Αυτό είχε ως αποτέλεσμα να κατεβάσει κάποια πακέτα που χρειάζονται να τρέξει το tailwind και τη δημιουργία του `tailwind.config.mjs` αρχείου στο οποίο μπορείς να κάνεις configure settings σχετικά με το tailwind όπως να δηλώσεις τη χρωματική παλέτα ή να προσθέσει κενούρια χρώματα,

animations και utilities.

Επίσης δήλωσε αυτόματα το framework ως plugin που θα χρησιμοποιηθεί από το πρότζεκτ στο `astro.config.mjs` configuration file

## 2. Ενσωμάτωση React για Διαδραστικότητα

Αν και το Astro παρέχει στατική απόδοση για τη βελτιστοποίηση της ταχύτητας, η ενσωμάτωση του **React** προσέφερε τη δυνατότητα προσθήκης διαδραστικών στοιχείων στον ιστότοπο. Το React ενσωματώθηκε μέσω της αρχιτεκτονικής **Island Architecture** του Astro, επιτρέποντας τη φόρτωση μόνο των απαραίτητων JavaScript στοιχείων, διατηρώντας τον υπόλοιπο ιστότοπο γρήγορο.

Το άστρο παρέχει επίσης εύκολη ενσωμάτωση με το **React** framework εκτελώντας μια εντολή.

```
npx astro add react
```

Αυτή η εντολή κατέβασε τα απαραίτητα πακέτα και εργαλεία για την ενσωμάτωση του framework και δήλωσε την react ως plugin στο configuration file του framework.

## 3. Ενσωμάτωση Shaden UI

Μετά την ενσωμάτωση του Tailwind CSS και του React, το επόμενο βήμα ήταν η χρήση του **Shaden UI** για τη δημιουργία **UI components** που προσφέρουν διαδραστικότητα και προσβασιμότητα. Το **Shaden UI** προσφέρει έτοιμα components που είναι σχεδιασμένα με έμφαση στην προσβασιμότητα και μπορούν να προσαρμοστούν εύκολα για να ταιριάζουν στις ανάγκες του έργου.

Η χρήση του **Shaden UI** σε συνδυασμό με το **Tailwind CSS** επιτάχυνε την ανάπτυξη των UI components, διατηρώντας παράλληλα τον ιστότοπο φιλικό προς τους χρήστες και προσβάσιμο για όλους.

Για την προσθήκη του component library ακολουθηθηκαν οι οδηγίες από το documentation. Αυτό είχε ως αποτέλεσμα την προσθήκη κάποιων configurations και styles από το shaden.

### 3.5. Routing και χρήση Layouts στο Astro

Ένα από τα βασικά χαρακτηριστικά του **Astro** είναι η απλότητα με την οποία μπορείς να δημιουργήσεις σελίδες και να διαχειριστείς τη δομή τους. Το Astro ακολουθεί την προσέγγιση του **file-based routing**, όπου κάθε αρχείο στον φάκελο `src/pages` αντιστοιχεί σε μια ξεχωριστή σελίδα του ιστότοπου.

#### Δημιουργία Σελίδων στο Astro

Κάθε σελίδα στο Astro δημιουργείται μέσα στον φάκελο `src/pages` και η ονομασία του αρχείου καθορίζει τη διαδρομή (route) του ιστότοπου. Για παράδειγμα:

- Ένα αρχείο με το όνομα `index.astro` μέσα στον φάκελο `src/pages` θα αντιστοιχεί στην

αρχική σελίδα του ιστότοπου.

- Ένα αρχείο `news.astro` θα δημιουργήσει μια σελίδα στη διαδρομή `/news` που θα εμφανίζει τα άρθρα ειδήσεων.

Κάθε σελίδα μπορεί να περιέχει:

- **Static content:** Το περιεχόμενο της σελίδας σε HTML ή Markdown.
- **Dynamic content:** Δυναμικά στοιχεία που μπορούν να φορτωθούν με React ή άλλα JavaScript frameworks.

### Χρήση Layouts για Κοινή Δομή Σελίδων

Για να διασφαλίσω ότι όλες οι σελίδες έχουν μια κοινή δομή και να μειώσω τον επαναλαμβανόμενο κώδικα, χρησιμοποίησα **layouts**. Τα layouts στο Astro είναι ειδικά αρχεία που επιτρέπουν την επαναχρησιμοποίηση δομών και στοιχείων σε πολλές σελίδες. Το Base Layout που δημιουργήθηκε παρέχει μια κοινή δομή για όλες τις σελίδες του ιστότοπου.

Το layout περιλαμβάνει:

- **Header:** Τοποθετείται στην κορυφή της σελίδας και περιέχει το μενού πλοήγησης και τυχόν πληροφορίες branding.
- **Footer:** Στο κάτω μέρος κάθε σελίδας, περιέχει πληροφορίες όπως τα δικαιώματα πνευματικής ιδιοκτησίας και συνδέσμους σε νομικές σελίδες.
- **Main Content:** Το layout ορίζει ένα δυναμικό μέρος για το περιεχόμενο κάθε σελίδας, χρησιμοποιώντας το `slot`, το οποίο αντικαθίσταται από το μοναδικό περιεχόμενο της σελίδας.

Για παράδειγμα, ένα βασικό layout θα μπορούσε να μοιάζει με το εξής:

```
---
// BaseLayout.astro
---
<html>
  <head>
    <title>Research Portal</title>
  </head>
  <body>
    <header>
      <!-- Navigation and Logo go here -->
    </header>

    <main>
      <slot /> <!-- Unique page content goes here -->
    </main>

    <footer>
      <!-- Footer content goes here -->
    </footer>
  </body>
</html>
```

Με αυτόν τον τρόπο, κάθε σελίδα μπορεί να "κληρονομήσει" το layout και να προσθέσει το μοναδικό της περιεχόμενο, χωρίς να χρειάζεται να αντιγράφεις τον ίδιο κώδικα για header και footer σε κάθε σελίδα.

### Σύνδεση Σελίδων με Layouts

Κάθε σελίδα μπορεί να χρησιμοποιήσει το base layout με τον εξής τρόπο:

```
---
// news.astro
import BaseLayout from '../layouts/BaseLayout.astro';
---
<BaseLayout>
  <h1>Latest News</h1>
  <p>Here are the latest updates from our research team...</p>
</BaseLayout>
```

Σε αυτή την περίπτωση, η σελίδα `news.astro` χρησιμοποιεί το **Base Layout** που δημιουργήθηκε, και το μοναδικό περιεχόμενο της σελίδας (τίτλος και περιγραφή) τοποθετείται στο `slot` του layout.

### Ευελιξία στη Δημιουργία Σελίδων

Η προσέγγιση αυτή επιτρέπει την εύκολη και γρήγορη δημιουργία νέων σελίδων στον ιστότοπο, διατηρώντας παράλληλα τη συνοχή στη σχεδίαση και τη λειτουργικότητα. Κάθε σελίδα μπορεί να έχει το δικό της μοναδικό περιεχόμενο, ενώ τα κοινά στοιχεία (header, footer) παραμένουν σταθερά και εύκολα διαχειρίσιμα.

### 3.6. Ενσωμάτωση Light/Dark Mode

Μετά την ενσωμάτωση του βασικού layout, το επόμενο βήμα ήταν η δημιουργία της λειτουργίας **light/dark mode**. Για να επιτευχθεί μια ομαλή εμπειρία για τους χρήστες, αναπτύχθηκε ένα **custom script** που χειρίζεται την αλλαγή του θέματος ανάμεσα σε light και dark mode, εξασφαλίζοντας ότι η προτίμηση του χρήστη αποθηκεύεται και εφαρμόζεται σωστά. Το script που αναπτύχθηκε λειτουργεί με τον εξής τρόπο:

## Φόρτωση Θέματος από το LocalStorage ή Προτίμηση Χρήστη

Το script ελέγχει πρώτα αν υπάρχει αποθηκευμένη προτίμηση θέματος στο **localStorage**. Αν υπάρχει, εφαρμόζει την αποθηκευμένη προτίμηση. Εάν δεν υπάρχει αποθηκευμένη προτίμηση, το script ελέγχει το **system preference** του χρήστη μέσω της **matchMedia** για να ορίσει αν το θέμα θα είναι **light** ή **dark**.

```
const getThemePreference = () => {  
  if (localStorage.getItem("theme")) {  
    return localStorage.getItem("theme");  
  }  
  return window.matchMedia("(prefers-color-scheme: dark)").matches  
    ? "dark"  
    : "light";  
};
```

## Εφαρμογή του Θέματος

Το script προσθέτει ή αφαιρεί την κλάση **"dark"** στο στοιχείο **html** του εγγράφου, ανάλογα με την επιλογή του χρήστη, ώστε να εφαρμοστεί το κατάλληλο θέμα.

```
const applyTheme = (document, theme) => {  
  const isDark = theme === "dark";  
  document.documentElement.classList[isDark ? "add" : "remove"]("dark");  
};
```

## Παρακολούθηση Αλλαγών Θέματος

Για να διασφαλιστεί ότι οποιαδήποτε αλλαγή στην προτίμηση του χρήστη αποθηκεύεται αυτόματα, το script δημιουργεί έναν **observer** που παρακολουθεί τις αλλαγές στην κλάση "dark" στο στοιχείο `html`. Όταν αλλάζει το θέμα, ο observer αποθηκεύει την προτίμηση στο **localStorage**.

```
const setupObserver = (document) => {  
  const observer = new MutationObserver(() => {  
    const isDark = document.documentElement.classList.contains("dark");  
    localStorage.setItem("theme", isDark ? "dark" : "light");  
  });  
  
  observer.observe(document.documentElement, {  
    attributes: true,  
    attributeFilter: ["class"],  
  });  
};
```

## Φόρτωση του Θέματος Πριν από την Απόδοση της Σελίδας

Το script τοποθετήθηκε στις **head properties** της σελίδας ώστε να φορτώνεται πριν το υπόλοιπο περιεχόμενο, αποτρέποντας το φαινόμενο του "**flashing**", όπου η σελίδα εμφανίζεται σε διαφορετικό θέμα για λίγα δευτερόλεπτα πριν εφαρμοστεί η προτίμηση του χρήστη.

```
const loadTheme = (document) => {  
  const theme = getThemePreference();  
  setupObserver(document);  
  applyTheme(document, theme);  
};  
  
loadTheme(document);
```

## Ενσωμάτωση με το Astro

Για την πλήρη ενσωμάτωση στο **Astro**, το script ακούει στο event `astro:before-swap` ώστε να εφαρμόσει το θέμα στο νέο document κατά τη διάρκεια της πλοήγησης.

```
document.addEventListener("astro:before-swap", (ev) => {  
  loadTheme(ev.newDocument);  
});
```

Με αυτή την υλοποίηση, το θέμα του ιστότοπου προσαρμόζεται με βάση την προτίμηση του χρήστη, χωρίς να προκαλείται καθυστέρηση στην αλλαγή του θέματος κατά την πλοήγηση.

Τέλος δημιουργήθηκε ένα επαναχρησιμοποιούμενο interactive component με χρήση react ώστε να μπορεί ο χρήστης να εναλλάσσει το θέμα στη σελίδα και να αποθηκεύετε στον browser η προτίμηση του.

### Δημιουργία Component για Εναλλαγή Θέματος (ThemeToggle)

Το **ThemeToggle** component αναπτύχθηκε για να επιτρέπει στους χρήστες να αλλάζουν μεταξύ **light** και **dark mode** στον ιστότοπο. Παρακάτω περιγράφονται τα κύρια σημεία του κώδικα, χωρισμένα σε τμήματα για καλύτερη κατανόηση.

#### 1. Αρχικοποίηση της Κατάστασης του Θέματος

Το πρώτο μέρος του κώδικα ελέγχει αν το `document.documentElement` περιέχει την κλάση `"dark"` και αρχικοποιεί την κατάσταση `theme` ανάλογα:

```
const [theme, setTheme] = useState(() =>  
  document.documentElement.classList.contains("dark") ? "dark" : "light",  
);
```

Αυτό επιτρέπει στον ιστότοπο να ξεκινά είτε με το **dark mode** είτε με το **light mode** ανάλογα με την τρέχουσα κατάσταση του HTML στοιχείου. Το **useState** hook χρησιμοποιείται για να παρακολουθεί την κατάσταση του θέματος.

## 2. Δυναμική Επιλογή Εικονιδίων

Το component επιλέγει δυναμικά το κατάλληλο εικονίδιο (ήλιος ή φεγγάρι) ανάλογα με την τρέχουσα κατάσταση του θέματος:

```
const ThemeIcon = theme === "dark" ? SunIcon : MoonIcon;
```

Όταν το θέμα είναι "dark", εμφανίζεται το **SunIcon** για να υποδηλώσει ότι ο χρήστης μπορεί να αλλάξει σε **light mode**. Όταν είναι "light", εμφανίζεται το **MoonIcon**.

## 3. Εναλλαγή Θέματος

Η λειτουργία **toggleTheme** χρησιμοποιείται για να εναλλάσσει το θέμα όταν ο χρήστης πατά το κουμπί. Η κλάση "dark" προστίθεται ή αφαιρείται από το `document.documentElement`, ανάλογα με την τρέχουσα κατάσταση:

```
const toggleTheme = () => {  
  const newTheme = theme === "dark" ? "light" : "dark";  
  document.documentElement.classList.toggle("dark", newTheme === "dark");  
};
```

Αυτή η λειτουργία διαχειρίζεται την αλλαγή του θέματος άμεσα, χωρίς να απαιτείται ανανέωση της σελίδας.

#### 4. Παρακολούθηση Αλλαγών στο DOM

Το component χρησιμοποιεί έναν **MutationObserver** για να παρακολουθεί τις αλλαγές στην κλάση "dark" στο HTML στοιχείο. Όταν ανιχνεύεται αλλαγή, η κατάσταση theme ενημερώνεται αναλόγως:

```
useEffect(() => {
  const observer = new MutationObserver(() => {
    const isDark = document.documentElement.classList.contains("dark");
    setTheme(isDark ? "dark" : "light");
  });

  observer.observe(document.documentElement, {
    attributes: true,
    attributeFilter: ["class"],
  });

  return () => observer.disconnect();
}, []);
```

Αυτό εξασφαλίζει ότι το **ThemeToggle** component παραμένει συγχρονισμένο με την τρέχουσα κατάσταση του θέματος, ακόμα και αν αλλάξει από εξωτερικές πηγές.

#### 5. Απόδοση του Component

Τέλος, το **ThemeToggle** αποδίδει ένα κουμπί που εμφανίζει το εικονίδιο του θέματος και καλεί τη λειτουργία εναλλαγής όταν πατηθεί:

```
return (
  <Button onClick={toggleTheme} {...rest}>
    <ThemeIcon {...iconProps} />
  </Button>
);
```

Το κουμπί ενεργοποιεί την αλλαγή του θέματος, ενώ το εικονίδιο αλλάζει ανάλογα με την τρέχουσα κατάσταση.

### 3.7. Δημιουργία Σελίδων και Χρήση των Content Collections του Astro

Μετά την υλοποίηση της λειτουργίας **light/dark mode** και των βασικών λειτουργιών της διεπαφής χρήστη, το επόμενο βήμα ήταν να δημιουργηθούν οι σελίδες που θα παρουσιάζουν το περιεχόμενο του

ιστότοπου. Ο ιστότοπος απαιτούσε σελίδες για την εμφάνιση των νέων, των έργων, των δημοσιεύσεων, και των προφίλ μελών της ερευνητικής ομάδας. Για να διαχειριστώ τα δεδομένα αυτών των σελίδων, χρησιμοποίησα τις **content collections** του **Astro**, οι οποίες παρέχουν έναν απλό και οργανωμένο τρόπο αποθήκευσης και ανάκτησης περιεχομένου.

### Δημιουργία Σελίδων

Για κάθε κατηγορία περιεχομένου (νέα, έργα, δημοσιεύσεις, μέλη ομάδας), δημιουργήθηκαν ξεχωριστές σελίδες μέσα στον φάκελο `src/pages` του έργου. Αυτές οι σελίδες αντιστοιχούν σε διαδρομές (routes) του ιστότοπου και φορτώνουν το περιεχόμενο μέσω των **content collections**.

#### Σελίδες Νέων:

- Η σελίδα `/news/index.astro` ανακτά και εμφανίζει τα άρθρα ειδήσεων από τα **content collections**. Τα άρθρα ειδήσεων είναι αποθηκευμένα σε αρχεία Markdown και περιέχουν πληροφορίες όπως ο τίτλος, η σύνοψη, η ημερομηνία δημοσίευσης και οι ετικέτες.
- Η σελίδα `/news/[slug].astro` εμφανίζει τις πληροφορίες και το περιεχόμενο ενός άρθρου.

#### Σελίδα Έργων:

- Η σελίδα `/projects/index.astro` φορτώνει τα δεδομένα από τη συλλογή έργων (projects) και τα παρουσιάζει με έναν οργανωμένο τρόπο. Κάθε έργο περιλαμβάνει τον τίτλο, τη σύνοψη, τα μέλη που συμμετέχουν και τους εξωτερικούς συνεργάτες (αν υπάρχουν).
- Η σελίδα `/projects/[slug].astro` εμφανίζει τις πληροφορίες και το περιεχόμενο ενός έργου..

#### Σελίδα Δημοσιεύσεων:

- Η σελίδα `/publications/index.astro` εμφανίζει ακαδημαϊκές δημοσιεύσεις της ερευνητικής ομάδας, αντλώντας δεδομένα από τη συλλογή δημοσιεύσεων (publications). Οι πληροφορίες περιλαμβάνουν τον τίτλο, τη σύνοψη, τους συγγραφείς και την ημερομηνία δημοσίευσης.
- Η σελίδα `/publications/[slug].astro` εμφανίζει τις πληροφορίες και το περιεχόμενο μιας ακαδημαϊκής δημοσίευσης.

#### Σελίδα Μελών Ομάδας:

- Η σελίδα `/team/index.astro` παρουσιάζει τα προφίλ των μελών της ομάδας, με πληροφορίες όπως το όνομα, την ειδικότητα, και τα έργα ή τις δημοσιεύσεις στις οποίες συμμετέχουν. Τα δεδομένα για τα μέλη της ομάδας προέρχονται από τη συλλογή authors.
- Η σελίδα `/team/[slug].astro` παρουσιάζει περιεχόμενο και πληροφορίες για έναν συγκεκριμένο μέλος της ομάδας.

## Χρήση των Content Collections του Astro

Το **Astro** προσφέρει τις **content collections**, οι οποίες επιτρέπουν την αποθήκευση δομημένου περιεχομένου σε μορφή Markdown ή JSON και την εύκολη ανάκτηση αυτών των δεδομένων στις σελίδες του ιστότοπου. Οι content collections χρησιμοποιήθηκαν για να διαχειριστούν διαφορετικές κατηγορίες περιεχομένου όπως νέα, έργα και δημοσιεύσεις.

## Πλεονεκτήματα Χρήσης των Content Collections

Η χρήση των **content collections** του Astro προσφέρει πολλαπλά πλεονεκτήματα:

- **Απλότητα:** Τα αρχεία Markdown είναι εύκολα στη δημιουργία και διαχείριση, επιτρέποντας στα μέλη της ερευνητικής ομάδας να προσθέτουν νέο περιεχόμενο χωρίς την ανάγκη χρήσης βάσεων δεδομένων.
- **Οργάνωση Δεδομένων:** Με τη χρήση των schemas, τα δεδομένα παραμένουν οργανωμένα και σωστά δομημένα, μειώνοντας τα λάθη στη διαχείριση περιεχομένου.
- **Ευελιξία:** Τα δεδομένα μπορούν εύκολα να φορτωθούν και να εμφανιστούν με διάφορους τρόπους στις σελίδες, παρέχοντας ευελιξία στη μορφοποίηση και στην παρουσίαση.

## 3.8. Δημιουργία Interactive Header για Κινητές Συσκευές

Ένα από τα σημαντικά στοιχεία της εμπειρίας χρήστη σε έναν ιστότοπο είναι το **header**, που περιλαμβάνει το μενού πλοήγησης. Καθώς οι χρήστες έχουν πρόσβαση στον ιστότοπο από διάφορες συσκευές, ήταν απαραίτητο να δημιουργηθεί ένα **responsive header** που να προσαρμόζεται ανάλογα με το μέγεθος της οθόνης. Για τον λόγο αυτό, υλοποιήθηκε ένα **interactive header** ως React component, το οποίο προσαρμόζεται για κινητές συσκευές και περιλαμβάνει ένα **sidebar menu**.

## 3.9. Δημιουργία Διαδραστικών Components για Φιλτράρισμα και Αναζήτηση Περιεχομένου

Για να βελτιωθεί η εμπειρία των χρηστών και να διευκολυνθεί η αναζήτηση περιεχομένου, δημιουργήθηκαν διαδραστικά components που επιτρέπουν το **φιλτράρισμα** και την **αναζήτηση** περιεχομένων. Τα φίλτρα και η λειτουργία αναζήτησης εφαρμόστηκαν στις σελίδες **νέων**, **έργων** και **δημοσιεύσεων**, ώστε οι χρήστες να μπορούν εύκολα να εντοπίζουν την πληροφορία που αναζητούν.

### Λειτουργίες Αναζήτησης και Φιλτραρίσματος

#### 1. Αναζήτηση με Βάση τον Τίτλο:

Οι χρήστες μπορούν να εισάγουν λέξεις-κλειδιά ή φράσεις για να αναζητήσουν συγκεκριμένα άρθρα, έργα ή δημοσιεύσεις με βάση τον τίτλο τους. Η αναζήτηση γίνεται σε πραγματικό χρόνο και εμφανίζει μόνο τα αποτελέσματα που ταιριάζουν με τον όρο αναζήτησης.

#### 2. Φιλτράρισμα με Βάση τη Χρονική Περίοδο:

Το φιλτράρισμα με βάση τη χρονική περίοδο επιτρέπει στους χρήστες να περιορίσουν τα αποτελέσματα

σε συγκεκριμένες χρονικές περιόδους, όπως:

1. Σήμερα
2. Αυτή τη βδομάδα
3. Αυτό το μήνα
4. Αυτή τη χρονιά

Αυτή η λειτουργία διευκολύνει τους χρήστες να βρουν γρήγορα πρόσφατα άρθρα, έργα ή δημοσιεύσεις.

### 3. Φιλτράρισμα με Βάση τον Συγγραφέα:

Οι χρήστες έχουν τη δυνατότητα να φιλτράρουν τα αποτελέσματα με βάση τον συγγραφέα. Αυτό επιτρέπει την εύκολη προβολή άρθρων, έργων ή δημοσιεύσεων από συγκεκριμένα μέλη της ομάδας.

### 4. Φιλτράρισμα με Βάση τις Ετικέτες (Tags):

Το φιλτράρισμα με βάση τις ετικέτες δίνει τη δυνατότητα στους χρήστες να περιορίσουν τα αποτελέσματα σε συγκεκριμένες κατηγορίες ή θέματα που έχουν προ-καθοριστεί μέσω των ετικετών που έχουν αποδοθεί στα άρθρα, έργα ή δημοσιεύσεις.

### Ενσωμάτωση με Χρήση React και Context για Διαχείριση Δεδομένων

Για να πραγματοποιηθεί interactive φιλτραρισμα των περιεχομένων στο Astro, χρησιμοποιήθηκε **React context** για κάθε τύπο περιεχομένου (νέα, έργα, δημοσιεύσεις). Αυτή η προσέγγιση επιτρέπει την αρχική φόρτωση όλων των δεδομένων από τα content collections, τα οποία αποθηκεύονται στο context, προσφέροντας έτσι τη δυνατότητα **client-side** φιλτραρίσματος.

Τα δεδομένα των άρθρων, έργων και δημοσιεύσεων φορτώνονται από τα **content collections** του Astro κατά την αρχική φόρτωση της σελίδας. Αυτά τα δεδομένα περνιούνται στο αντίστοιχο context (π.χ., `NewsContext`, `ProjectsContext`, `PublicationsContext`), κάνοντας τα διαθέσιμα για επεξεργασία και εμφάνιση στις σελίδες.

Με τα δεδομένα αποθηκευμένα στο context, η αναζήτηση και το φιλτράρισμα πραγματοποιούνται **client-side**, χωρίς την ανάγκη επαναφόρτωσης της σελίδας ή επιπλέον αιτήματα προς τον server. Αυτή η προσέγγιση βελτιστοποιεί την απόδοση, επιτρέποντας άμεση απόκριση στις ενέργειες του χρήστη και βελτιώνοντας τη συνολική εμπειρία πλοήγησης.

Τα διαδραστικά φίλτρα, όπως η αναζήτηση με βάση τον τίτλο, το φιλτράρισμα ανά χρονική περίοδο (σήμερα, αυτή τη βδομάδα, αυτό το μήνα, αυτό το χρόνο), συγγραφέα ή ετικέτες, εφαρμόζονται στα δεδομένα που αποθηκεύονται στο context. Αυτό επιτρέπει τη δυναμική ανανέωση των αποτελεσμάτων σε πραγματικό χρόνο με βάση τα κριτήρια του χρήστη, χωρίς την ανάγκη επαναφόρτωσης της σελίδας.

### 3.10. Ενσωμάτωση Υποστήριξης Πολυγλωσσικότητας (i18n)

Για την υποστήριξη πολλαπλών γλωσσών στον ιστότοπο, υλοποιήθηκε η λειτουργία **internalization (i18n)**. Αυτή η προσθήκη επέτρεψε την εμφάνιση του περιεχομένου σε διαφορετικές γλώσσες ανάλογα με τις προτιμήσεις του χρήστη, καθώς και τη δυναμική φόρτωση και εμφάνιση του περιεχομένου με βάση τη γλώσσα.

#### Ρύθμιση Παραμέτρων i18n

Αρχικά, δημιουργήθηκε ένα αρχείο ρυθμίσεων για τον ορισμό των διαθέσιμων γλωσσών του ιστότοπου. Σε αυτό το αρχείο καθορίστηκαν οι υποστηριζόμενες γλώσσες, καθώς και η προεπιλεγμένη γλώσσα:

```
// i18n/config.ts
export const languages = {
  en: "English",
  el: "Ελληνικά",
} as const;

export const defaultLang = "en";
export const showDefaultLang = true;
```

#### Ενημέρωση του Astro Configuration

Οι ρυθμίσεις αυτές ενσωματώθηκαν στο αρχείο παραμετροποίησης του **Astro** (`astro.config.mjs`), ώστε να γίνεται δυναμική φόρτωση του περιεχομένου βάσει της γλώσσας και να προστίθεται η γλώσσα στις διευθύνσεις URL:

```
// astro.config.mjs
import { defaultLang, languages } from "../src/i18n/config.ts";

export default defineConfig({
  i18n: {
    defaultLocale: defaultLang,
    locales: Object.keys(languages),
    routing: {
      prefixDefaultLocale: true,
      redirectToDefaultLocale: true,
    },
  },
});
```

#### Αναδιάρθρωση των Σελίδων και Content Collections

Η δομή των σελίδων και των content collections τροποποιήθηκε ώστε να περιλαμβάνει τη γλώσσα στο path. Με αυτόν τον τρόπο, κάθε σελίδα και το περιεχόμενο που σχετίζεται με αυτήν προσαρμόζεται ανάλογα με τη γλώσσα του χρήστη:

- **Παλιά Δομή:** `/pages/news/index.astro`
- **Νέα Δομή:** `/pages/[lang]/news/index.astro`

### Βοηθητικές Μέθοδοι για τη Διαχείριση της Γλώσσας

Για την καλύτερη διαχείριση της γλώσσας στον ιστότοπο, αναπτύχθηκαν βοηθητικές μέθοδοι. Αυτές οι μέθοδοι περιλαμβάνουν τη δυνατότητα εξαγωγής της γλώσσας από το URL, τη μετάφραση των διαδρομών (URIs), και την πρόσβαση στις μεταφράσεις:

```
export function useTranslations(lang: keyof typeof translations) {  
  return function t(key: keyof (typeof translations)[typeof defaultLang]) {  
    return translations[lang][key] || translations[defaultLang][key];  
  };  
}
```

### Υλοποίηση των Μεταφράσεων

Δημιουργήθηκε ένα αρχείο που περιέχει όλες τις μεταφράσεις των κειμένων του ιστότοπου, οργανωμένες ανά σελίδα και component. Οι μεταφράσεις αυτές είναι διαθέσιμες μέσω της utility μεθόδου `useTranslations`, η οποία χρησιμοποιεί την τρέχουσα γλώσσα για να επιστρέφει τη σωστή μετάφραση του κειμένου:

```
// /i18n/translations.ts  
export const translations = {  
  en: {  
    "page.home.title": "We share news and we do research in Computer Science",  
    "nav-link.home": "Home",  
    // άλλες μεταφράσεις...  
  },  
  el: {  
    "page.home.title": "Μοιραζόμαστε νέα και κάνουμε έρευνα στην Επιστήμη Υπολογιστών",  
    "nav-link.home": "Αρχική",  
    // άλλες μεταφράσεις...  
  },  
};
```

Με αυτή τη δομή, το περιεχόμενο προσαρμόζεται αυτόματα στη γλώσσα του χρήστη, διασφαλίζοντας ότι η εμπειρία πλοήγησης είναι πλήρως μεταφρασμένη.

## 4. Ρύθμιση Έργου και Διαχείριση Περιεχομένου

### 4.1 Εγκατάσταση

Για να τρέξει το project τοπικά, απαιτούνται ορισμένα εργαλεία. Ακολουθούν οι προαπαιτήσεις και τα βήματα εγκατάστασης.

#### Προαπαιτούμενα

Πριν ξεκινήσετε, βεβαιωθείτε ότι έχετε εγκατεστημένα τα παρακάτω:

- [Node.js](#) (έκδοση 18 ή νεότερη)
- **npm** (περιλαμβάνεται με την εγκατάσταση του Node.js)

#### Βήματα Εγκατάστασης

1. Αντιγραφή του αποθετηρίου:

```
git clone https://github.com/lefpap/research-portal.git  
cd <repository-directory>
```

2. Εγκατάσταση των dependencies

Για την εγκατάσταση όλων των απαραίτητων εξαρτήσεων, τρέξτε την εντολή:

```
npm install
```

3. Εκκίνηση του development server

Για να ξεκινήσετε τον server ανάπτυξης και να δείτε το website σε τοπικό περιβάλλον:

```
npm run dev
```

Ο ιστότοπος θα είναι διαθέσιμος στη διεύθυνση <http://localhost:4321>. Οποιαδήποτε αλλαγή στα αρχεία θα ανανεώνει αυτόματα τον browser.

### 4.2 Προτεινόμενα Εργαλεία και Plugins

#### Εργαλεία Ανάπτυξης

Για ευκολότερη ανάπτυξη του έργου, προτείνεται η χρήση του **Visual Studio Code (VSCode)** και η εγκατάσταση των παρακάτω επεκτάσεων:

- [Astro VSCode Extension](#) για υποστήριξη συντακτικού του Astro.
- [Tailwind CSS IntelliSense](#) για αυτόματες προτάσεις και συμπλήρωση κώδικα στο Tailwind CSS.
- [Prettier](#) για την αυτόματη μορφοποίηση του κώδικα κατά την αποθήκευση.

Για να ρυθμίσετε το Prettier να μορφοποιεί αυτόματα τα αρχεία `.astro` κατά την αποθήκευση, προσθέστε τις παρακάτω ρυθμίσεις στο **settings.json** του VSCode:

```
{
  "editor.defaultFormatter": "esbenp.prettier-vscode",
  "editor.formatOnSave": true,
  "prettier.documentSelectors": ["**/*.astro"],
  "[astro]": {
    "editor.defaultFormatter": "esbenp.prettier-vscode"
  }
}
```

Επιπλέον, βεβαιωθείτε ότι το **.prettierrc** αρχείο στο root directory περιλαμβάνει:

```
{
  "plugins": ["prettier-plugin-astro", "prettier-plugin-tailwindcss"],
  "overrides": [
    {
      "files": "*.astro",
      "options": {
        "parser": "astro"
      }
    }
  ]
}
```

### 4.3 Εντολές (Commands)

Για τη διαχείριση του project, μπορείς να χρησιμοποιήσεις τις παρακάτω εντολές μέσω του τερματικού, εκτελώντας τες από το root directory του έργου:

| Command                              | Action                                                                                      |
|--------------------------------------|---------------------------------------------------------------------------------------------|
| <code>npm install</code>             | Εγκατάσταση των απαραίτητων εξαρτήσεων                                                      |
| <code>npm run dev</code>             | Εκκίνηση του development server στη διεύθυνση <code>localhost:4321</code>                   |
| <code>npm run build</code>           | Δημιουργία της παραγωγικής έκδοσης του ιστότοπου στον φάκελο <code>./dist/</code>           |
| <code>npm run preview</code>         | Προεπισκόπηση της παραγωγικής έκδοσης τοπικά, πριν από την ανάπτυξη                         |
| <code>npm run astro ...</code>       | Εκτέλεση εντολών μέσω του Astro CLI, όπως <code>astro add</code> , <code>astro check</code> |
| <code>npm run astro -- --help</code> | Εμφάνιση βοήθειας σχετικά με τις εντολές του Astro CLI                                      |

Αυτές οι εντολές είναι απαραίτητες για την εκτέλεση και διαχείριση του project, είτε σε περιβάλλον ανάπτυξης είτε σε παραγωγικό περιβάλλον.

## 4.4 Δομή του Έργου

Η δομή του project οργανώνεται με σαφή και εύκολο τρόπο ώστε να υποστηρίζει την ανάπτυξη και τη διαχείριση του περιεχομένου. Ακολουθεί η ανάλυση των κύριων φακέλων και αρχείων:

```
├── public
│   ├── favicon.svg
│   ├── icons
│   └── js
├── src
│   ├── components
│   ├── content
│   ├── context
│   ├── env.d.ts
│   ├── hooks
│   ├── i18n
│   ├── layouts
│   ├── lib
│   ├── pages
│   └── styles
├── tailwind.config.mjs
├── tsconfig.json
├── astro.config.mjs
├── remark-reading-time.mjs
├── components.json
├── package-lock.json
└── package.json
```

### Root Directory

- **README.md**: Περιέχει οδηγίες για το project, όπως περιγραφή, βήματα εγκατάστασης και οδηγίες χρήσης.
- **astro.config.mjs**: Το κύριο αρχείο παραμετροποίησης του Astro, που καθορίζει ρυθμίσεις και επιλογές για το static site generator.
- **tailwind.config.mjs**: Αρχείο παραμετροποίησης του **Tailwind CSS**, που καθορίζει προσαρμογές και ρυθμίσεις για τα στυλ.
- **tsconfig.json**: Αρχείο παραμετροποίησης του **TypeScript**, που ορίζει επιλογές για τον compiler και τις ρυθμίσεις του έργου.
- **package.json**: Ορίζει τις εξαρτήσεις και τα scripts του έργου.
- **components.json**: Σχετίζεται με το **Shadcn** και χρησιμοποιείται για τον ορισμό και τη διαχείριση

των UI components.

### Public Directory

- **public/**: Περιέχει στατικά αρχεία που εξυπηρετούνται απευθείας από τον server χωρίς επεξεργασία, όπως εικόνες και αρχεία JavaScript.
  - **favicon.svg**: Το favicon που εμφανίζεται στην καρτέλα του browser.
  - **icons/**: Περιέχει εικόνες και εικονίδια που χρησιμοποιούνται στον ιστότοπο.
  - **js/**: Περιέχει JavaScript αρχεία που χρησιμοποιούνται απευθείας από τον client.

### Source Directory (src)

- **src/**: Ο κύριος φάκελος του έργου, όπου γίνεται όλη η ανάπτυξη και ο προγραμματισμός.
  - **components/**: Περιέχει επαναχρησιμοποιούμενα UI components.
  - **content/**: Περιέχει τη δήλωση των **content collections** του Astro και των αντίστοιχων σχημάτων τους. Κάθε collection ορίζεται με το δικό του σχήμα για την επαλήθευση και οργάνωση των δεδομένων. Για παράδειγμα, υπάρχουν collections για **νέα**, **έργα**, **δημοσιεύσεις**, και **συγγραφείς**:

```
import { defineCollection } from "astro:content";

const newsCollection = defineCollection({
  schema: z.object({
    title: z.string(),
    summary: z.string(),
    publishedAt: z.date(),
    tags: z.array(z.string()),
  }),
});
```

Αφού φτιαχτεί το collection και το σχήμα του, πρέπει να δηλωθεί με ένα όνομα σε ένα object που ονομάζεται **collections**.

```
export const collections = {  
  news: newsCollection,  
};
```

Έτσι δηλώνεται στο Astro ότι υπάρχει ένα collection με το όνομα *news* στο οποίο μπορούν να φτιαχτούν markdown αρχεία που αντιπροσωπεύουν το σχήμα των news στον φάκελο `/content/news/` για να φτιάξεις δυναμικό περιεχόμενο.

- **context/**: Παρέχει context providers για διαχείριση κατάστασης (state) και δεδομένων.
- **i18n/**: Περιέχει τα αρχεία ρυθμίσεων και μεταφράσεων για την υποστήριξη πολυγλωσσικότητας (i18n) στον ιστότοπο. Περιλαμβάνει τα εξής:
  - **config.ts**: Ορισμός των υποστηριζόμενων γλωσσών και των προεπιλεγμένων ρυθμίσεων γλώσσας.
  - **translations.ts**: Περιέχει όλες τις μεταφράσεις για τα κείμενα και τα μηνύματα που εμφανίζονται στον ιστότοπο.
  - **utils.ts**: Βοηθητικές μέθοδοι για τη διαχείριση των γλωσσών, όπως η εξαγωγή της τρέχουσας γλώσσας από το URL και η μετάφραση των διαδρομών (URIs).
- **layouts/**: Περιέχει layout components που χρησιμοποιούνται για την παροχή συνεπούς δομής και μορφοποίησης στις σελίδες.
- **lib/**: Περιέχει utility functions και βιβλιοθήκες που χρησιμοποιούνται ευρέως στο project.
- **pages/**: Ο φάκελος αυτός περιέχει τα αρχεία σελίδων που αντιστοιχούν στις διαδρομές (routes) του ιστότοπου.
- **styles/**: Περιέχει CSS ή άλλα αρχεία στυλ για το project.

## 4.5 Διαχείριση Περιεχομένου

Η διαχείριση του περιεχομένου στον ιστότοπο είναι κρίσιμη για την παροχή ενημερωμένων και σχετικών πληροφοριών στους χρήστες. Σε αυτό το κεφάλαιο, θα εξετάσουμε πώς μπορείτε να χρησιμοποιήσετε το **Markdown** και το **Front Matter** για τη δημιουργία περιεχομένου, πώς να προσθέσετε νέο περιεχόμενο στον ιστότοπο και πώς να διαχειριστείτε την πολυγλωσσικότητα (i18n).

### 4.5.1 Χρήση του Markdown και Front Matter

Το **Markdown** είναι μια ελαφριά γλώσσα σήμανσης που επιτρέπει τη γρήγορη και εύκολη δημιουργία πλούσιου κειμένου χρησιμοποιώντας απλό κείμενο. Τα αρχεία περιεχομένου για τον ιστότοπο αποθηκεύονται στον φάκελο `src/content/` και γράφονται σε μορφή Markdown.

Το **Front Matter** είναι ένα μπλοκ από **YAML** ή **JSON** που τοποθετείται στην αρχή ενός αρχείου Markdown και περιέχει μεταδεδομένα για το περιεχόμενο. Τα μεταδεδομένα αυτά περιλαμβάνουν πληροφορίες όπως ο τίτλος, η ημερομηνία, ο συγγραφέας, οι ετικέτες και άλλα. Το Front Matter

περικλείεται μεταξύ των οριοθετών `---` ή `+++`.

```
---
title: "Δείγμα Άρθρου"
date: "2022-01-01"
author: "John Doe"
tags:
  - Astro
  - Markdown
---
```

Μετά το Front Matter, μπορείτε να γράψετε το περιεχόμενο του άρθρου σας χρησιμοποιώντας τη σύνταξη του Markdown.

## # Δείγμα Άρθρου

Αυτό είναι ένα δείγμα άρθρου γραμμένο σε Markdown.

### ## Υποενότητα

Εδώ είναι μια λίστα αντικειμένων:

- Αντικείμενο 1
- Αντικείμενο 2
- Αντικείμενο 3

#### 4.5.2 Προσθήκη Περιεχομένου

Ο ιστότοπος οργανώνει το περιεχόμενό του μέσω των **Content Collections**, που ορίζονται στα αρχεία `src/content/config.ts` και `src/content/schema.ts`. Υπάρχουν επτά τύποι περιεχομένου:

1. **Authors (Συγγραφείς):** Αντιπροσωπεύει μια συλλογή από συγγραφείς που μπορούν να συσχετιστούν με άρθρα ή δημοσιεύσεις.
2. **News (Νέα):** Αντιπροσωπεύει μια συλλογή από άρθρα ειδήσεων.
3. **Projects (Έργα):** Αντιπροσωπεύει μια συλλογή από έργα ή αντικείμενα χαρτοφυλακίου.
4. **Publications (Δημοσιεύσεις):** Αντιπροσωπεύει μια συλλογή από δημοσιεύσεις ή ακαδημαϊκές εργασίες.
5. **Education (Εκπαίδευση):** Αντιπροσωπεύει μια συλλογή από εκπαιδεύσεις ενός συγγραφέα.
6. **Work Experience (Εργασιακή Εμπειρία):** Αντιπροσωπεύει μια συλλογή από επαγγελματικές εμπειρίες ενός συγγραφέα.
7. **Legal (Νομικές Σελίδες):** Αντιπροσωπεύει μια συλλογή από νομικές σελίδες (Πολιτική

Απορρήτου, Όροι Χρήσης, κτλ.)

Λόγω της υποστήριξης πολυγλωσσικότητας (i18n), κάθε συλλογή περιεχομένου (**Content Collection**) οργανώνεται πλέον με υποφακέλους που αντιστοιχούν σε κάθε υποστηριζόμενη γλώσσα. Αυτό επιτρέπει την αποθήκευση και διαχείριση του περιεχομένου ξεχωριστά για κάθε γλώσσα.

#### Δομή φακέλων:

- **Συγγραφείς (Authors):**
  - `src/content/authors/en/` για τα στοιχεία συγγραφέα στα αγγλικά.
  - `src/content/authors/el/` για τα στοιχεία συγγραφέα στα ελληνικά.
- **Νέα (News):**
  - `src/content/news/en/` για τα αγγλικά άρθρα.
  - `src/content/news/el/` για τα ελληνικά άρθρα.
- **Έργα (Projects):**
  - `src/content/projects/en/` για τα αγγλικά έργα.
  - `src/content/projects/el/` για τα ελληνικά έργα.
- **Δημοσιεύσεις (Publications):**
  - `src/content/publications/en/` για τις αγγλικές δημοσιεύσεις.
  - `src/content/publications/el/` για τις ελληνικές δημοσιεύσεις.
- **Εκπαίδευση (Education):**
  - `src/content/education/en/` για τις εκπαιδεύσεις του συγγραφέα στα αγγλικά.
  - `src/content/education/el/` για τις εκπαιδεύσεις του συγγραφέα στα ελληνικά.
- **Εργασιακή Εμπειρία (Work Experience):**
  - `src/content/work-experience/en/` για τις αγγλικές εργασιακές εμπειρίες ενός συγγραφέα.
  - `src/content/work-experience/el/` για τις ελληνικές εργασιακές εμπειρίες ενός συγγραφέα.
- **Νομικές Σελίδες (Legal):**
  - `src/content/legal/en/` για τις αγγλικές νομικές σελίδες.
  - `src/content/legal/el/` για τις ελληνικές νομικές σελίδες.

#### Παράδειγμα:

- Αγγλικό άρθρο: `src/content/news/en/some-article.md`
- Ελληνικό άρθρο: `src/content/news/el/some-article.md`

Χρησιμοποιώντας το ίδιο όνομα αρχείου (`some-article.md`) και στους δύο φακέλους γλώσσας (`en` και `el`), διασφαλίζεται ότι οι μεταφράσεις συνδέονται σωστά μεταξύ τους και ότι ο ιστότοπος μπορεί να παρέχει το αντίστοιχο περιεχόμενο ανάλογα με τη γλώσσα που έχει επιλέξει ο χρήστης.

#### 4.5.2 Προσθήκη Εικόνων

Για το περιεχόμενο που μπορεί να περιέχει εικόνες, υπάρχει ένας φάκελος `_images` στο ανώτατο επίπεδο της κάθε συλλογής. Οι εικόνες που σχετίζονται με τα άρθρα, έργα ή δημοσιεύσεις αποθηκεύονται εκεί, διευκολύνοντας την οργάνωση και την πρόσβαση σε αυτές.

**Δομή φακέλων για εικόνες:**

- **Συγγραφείς (Authors):**
  - `src/content/authors/_images/`
- **Νέα (News):**
  - `src/content/news/_images/`

**Παράδειγμα:**

- Εικόνα εξωφύλλου άρθρου: `src/content/news/_images/some-cover.png`

#### 4.5.3 Προσθήκη Περιεχομένου

Για να προσθέσετε νέο περιεχόμενο στον ιστότοπο, ακολουθήστε τα παρακάτω βήματα για κάθε συλλογή. Θα δώσουμε παραδείγματα για τις συλλογές **Νέα (News)**, **Έργα (Projects)**, **Δημοσιεύσεις (Publications)** και **Συγγραφείς (Authors)**.

**Προσθήκη Συγγραφέα (Author):**

1. Εντοπίστε τη συλλογή και γλώσσα:
  - `src/content/authors/en/`
  - `src/content/authors/el/`
2. Δημιουργήστε ένα νέο αρχείο Markdown:
  - Παράδειγμα: `src/content/authors/el/john-doe.md`
3. Προσθέστε την εικόνα του συγγραφέα στον φάκελο
  - Παράδειγμα: `src/content/authors/_images/john-doe.jpg`
4. Προσθέστε το Front Matter και το περιεχόμενο:

```
---  
email: "john.doe@email.com"  
firstname: "John"  
lastname: "Doe"  
avatar: "../_images/john-doe.jpg"  
---
```

Software engineer with a passion for creating innovative solutions. With a strong background in programming languages such as Python and JavaScript, he excels at developing robust and scalable applications. Bob's attention to detail and problem-solving skills make him a valuable asset to any development team. In his free time, he enjoys exploring new technologies and contributing to open-source projects. Connect with Bob on LinkedIn to learn more about his work in the software engineering field.

#### Προσθήκη Άρθρου Νέων (News Article):

1. Εντοπίστε τη σωστή συλλογή και γλώσσα:
  - ο Για ένα άρθρο στα αγγλικά: `src/content/news/en/`
  - ο Για ένα άρθρο στα ελληνικά: `src/content/news/el/`
2. Δημιουργήστε ένα νέο αρχείο Markdown:
  - ο Παράδειγμα: `src/content/news/en/new-technology-trends.md`
3. Προσθέστε την εικόνα εξωφύλλου του άρθρου στον φάκελο
  - ο Παράδειγμα: `src/content/news/_images/new-technology-trends.png`
4. Προσθέστε το Front Matter και το περιεχόμενο:

```
---
title: "New Technology Trends in 2023"
summary: "An overview of the latest technology trends for the
upcoming year."
publishedAt: "2023-10-01"
tags:
  - Technology
  - Trends
author: "john-doe"
---

# Introduction

The technology sector is evolving rapidly, with new trends emerging
every year...

![[Cover Image]](/src/content/news/_images/technology-trends-cover.png)
```

### Προσθήκη Έργου (Project):

1. Εντοπίστε τη σωστή συλλογή και γλώσσα:
  - ο Αγγλικά: `src/content/projects/en/`
  - ο Ελληνικά: `src/content/projects/el/`
2. Δημιουργήστε ένα νέο αρχείο Markdown:
  - ο Παράδειγμα: `src/content/projects/en/ai-research-project.md`
3. Προσθέστε το Front Matter και το περιεχόμενο:

```
---
title: AI Research Project
summary: A project focused on developing advanced AI algorithms.
status: draft
authors:
  - john-doe
externalAuthors:
  - fullname: Jane Smith
    url: "https://example.com/jane-smith"
tags:
  - AI
  - Research
repo: "https://github.com/username/ai-research-project"
demo: "https://ai-research.example.com"
---

## Project Overview

The AI Research Project aims to push the boundaries of artificial intelligence by exploring new algorithms...

![[Project Image]](/src/content/projects/_images/ai-project-image.png)
```

### Προσθήκη Έρευνας (Publication):

1. Εντοπίστε τη σωστή συλλογή και γλώσσα:
  - ο Αγγλικά: `src/content/publications/en/`
  - ο Ελληνικά: `src/content/publications/el/`
2. Δημιουργήστε ένα νέο αρχείο Markdown:
  - ο Παράδειγμα:

src/content/publications/en/advanced-machine-learning.md

3. Προσθέστε το Front Matter και το περιεχόμενο:

```
---
title: Advanced Machine Learning Techniques
summary: An in-depth study of advanced machine learning algorithms.
Status: "published"
publishedAt: "2023-08-20"
authors:
  - john-doe
externalAuthors:
  - fullname: Jane Smith
    url: "https://example.com/jane-smith"
tags:
  - Machine Learning
  - Research
cite: |
  @article{doe2023advanced,
    title={Advanced Machine Learning Techniques},
    author={Doe, John and Smith, Jane},
    journal={Journal of AI Research},
    year={2023},
    volume={10},
    pages={100-120},
  }
links:
  - name: doi
    url: https://doi.org/10.1234/abcd
---

## Abstract

In this paper, we explore advanced techniques in machine learning
that have the potential to revolutionize the field...
```

**Προσθήκη Εκπαίδευσης (Education):**

Η συλλογή Education έχει διαφορετική οργάνωση. Για κάθε συγγραφέα, υπάρχει ένας φάκελος που περιέχει όλα τα αρχεία εκπαίδευσης που σχετίζονται με αυτόν. Αυτή η δομή διευκολύνει τη διαχείριση του εκπαιδευτικού ιστορικού κάθε συγγραφέα.

1. Εντοπίστε τη συλλογή εκπαίδευσης και τον συγγραφέα:
  - Η βασική διαδρομή είναι `src/content/education/`.
  - Μέσα σε αυτόν, υπάρχουν υποφακέλοι για κάθε γλώσσα (`en`, `el`).
  - Μέσα στον φάκελο της γλώσσας, δημιουργήστε έναν φάκελο με το όνομα του συγγραφέα.

Παράδειγμα:

- Για τον συγγραφέα **John Doe** στα αγγλικά:  
`src/content/education/en/john-doe/`
2. Δημιουργήστε ένα νέο αρχείο Markdown:
    - Τοποθετήστε το αρχείο στον φάκελο του συγγραφέα.
    - Παράδειγμα:  
`src/content/education/en/john-doe/computer-science-degree.md`
  3. Προσθέστε το Front Matter και το περιεχόμενο:

```
--  
author: en/john-doe  
degree: "Bachelor of Science in Computer Science"  
institution: "University of Technology"  
start: 2015  
end: 2019  
---  
  
During my studies at the University of Technology, I focused on  
software development and machine learning... data structures, and  
software engineering."
```

**Προσθήκη Επαγγελματικής Εμπειρίας (Work Experience):**

Η συλλογή Work Experience οργανώνεται παρόμοια με την Education.

Παράδειγμα περιεχομένου:

```
---  
title: "Software Engineer"  
company: "TechCorp"  
start: 2020  
end: null # null ή αφήστε κενό αν εργάζεστε ακόμη εκεί  
description: "Developing scalable web applications using modern  
technologies."
```

---

At TechCorp, I have participated in various projects that required...

### Προσθήκη Νομικών Κειμένων (Legal):

1. Εντοπίστε τη συλλογή νομικών κειμένων:
  - `src/content/legal/`
2. Δημιουργήστε ένα νέο αρχείο Markdown στον κατάλληλο φάκελο γλώσσας:
  - Αγγλικά: `src/content/legal/en/privacy-policy.md`
  - Ελληνικά: `src/content/legal/el/privacy-policy.md`
3. Προσθέστε το Front Matter και το περιεχόμενο:

```
---  
title: "Πολιτική Απορρήτου"  
lastUpdated: "2023-09-15"  
---
```

Δεσμευόμαστε να προστατεύουμε τα προσωπικά σας δεδομένα και το δικαίωμά σας στην ιδιωτικότητα...

### Σημαντικές Σημειώσεις:

- **Ονομασία αρχείων (slug):**

Για να λειτουργεί σωστά η αντιστοίχιση των μεταφράσεων μεταξύ των διαφορετικών γλωσσών, **τα αρχεία περιεχομένου πρέπει να έχουν το ίδιο όνομα (slug)** σε όλους τους υποφακέλους γλώσσας.

Παράδειγμα:

- Αγγλικό άρθρο: `src/content/news/en/some-article.md`
- Ελληνικό άρθρο: `src/content/news/el/some-article.md`

Χρησιμοποιώντας το ίδιο όνομα αρχείου (`some-article.md`) και στους δύο φακέλους γλώσσας (`en` και `el`), διασφαλίζεται ότι οι μεταφράσεις συνδέονται σωστά μεταξύ τους και ότι ο ιστότοπος μπορεί να παρέχει το αντίστοιχο περιεχόμενο ανάλογα με τη γλώσσα που έχει επιλέξει ο χρήστης.

- **Αποθήκευση Εικόνων:**

Για τα περιεχόμενα που μπορούν να περιέχουν εικόνες, χρησιμοποιήστε τον φάκελο `_images` στο ανώτατο επίπεδο της συλλογής. Αυτό βοηθά στην οργάνωση των εικόνων και διευκολύνει

την πρόσβαση σε αυτές από τα αρχεία περιεχομένου.

#### Παράδειγμα:

- Εικόνα εξωφύλλου άρθρου: `src/content/news/_images/some-cover.png`

#### Διαδρομές Εικόνων:

Όταν αναφέρεστε σε εικόνες στο περιεχόμενό σας, χρησιμοποιήστε τη σχετική διαδρομή που ξεκινά με `/src/...` για να εξασφαλίσετε ότι οι εικόνες φορτώνονται σωστά.

```
! [Περιγραφή εικόνας] (/src/content/news/_images/my-article-image.png)
```

- **Συνέπεια στα Μεταδεδομένα:**

Βεβαιωθείτε ότι τα πεδία στο Front Matter συμβαδίζουν με το σχήμα της συλλογής (schema). Αν κάποια πεδία είναι υποχρεωτικά, πρέπει να τα συμπεριλάβετε.

## 4.6 Διαχείριση Μεταφράσεων

Η διαχείριση των μεταφράσεων στον ιστότοπο είναι κρίσιμη για την υποστήριξη της πολυγλωσσικότητας και τη βελτίωση της εμπειρίας των χρηστών από διαφορετικές γλωσσικές κοινότητες. Σε αυτή την ενότητα, θα εξετάσουμε πώς μπορείτε να διαχειριστείτε τις μεταφράσεις των στατικών κειμένων και των μηνυμάτων του ιστότοπου, καθώς και πώς να προσθέσετε ή να ενημερώσετε μεταφράσεις.

#### Αρχείο Μεταφράσεων (translations.ts)

Οι μεταφράσεις στον ιστότοπο διαχειρίζονται μέσω του αρχείου `src/i18n/translations.ts`. Αυτό το αρχείο περιέχει τα κείμενα του ιστότοπου, οργανωμένα ανά κατηγορία και γλώσσα. Η δομή του αρχείου επιτρέπει την εύκολη προσθήκη, ενημέρωση και διαχείριση των μεταφράσεων.

Δομή του αρχείου `translations.ts`:

1. Μεταφράσεις Σελίδων:

```
const PAGE_TRANSLATIONS = {
  en: {
    /* Home Page EN */
    "page.home.title": "We share news and we do research in Computer Science",
    "page.home.subtitle":
      "Stay up to date with the latest news and research in computer
```

```

science.",
  // ... άλλες μεταφράσεις σελίδων στα Αγγλικά
},
el: {
  /* Home Page EL */
  "page.home.title":
    "Μοιραζόμαστε νέα και κάνουμε έρευνα στην Επιστήμη Υπολογιστών",
  "page.home.subtitle":
    "Μείνετε ενήμεροι με τα τελευταία νέα και την έρευνα στην επιστήμη των
υπολογιστών.",
  // ... άλλες μεταφράσεις σελίδων στα Ελληνικά
},
};

```

## 2. Μεταφράσεις των Navigation Links:

```

const NAV_LINK_TRANSLATIONS = {
  en: {
    "nav-link.home": "Home",
    "nav-link.team": "Team",
    "nav-link.news": "News",
    // ... άλλοι σύνδεσμοι πλοήγησης στα Αγγλικά
  },
  el: {
    "nav-link.home": "Αρχική",
    "nav-link.team": "Ομάδα",
    "nav-link.news": "Νέα",
    // ... άλλοι σύνδεσμοι πλοήγησης στα Ελληνικά
  },
};

```

## 3. Μεταφράσεις των Component:

```

const COMPONENT_TRANSLATIONS = {
  en: {
    /* News Components EN */
    "component.news-display.empty": "No news to display",
    "component.news-filters.search.placeholder": "Search news...",
    // ... άλλες μεταφράσεις components στα Αγγλικά
  },
};

```

```

el: {
  /* News Components EL */
  "component.news-display.empty": "Δεν υπάρχουν νέα",
  "component.news-filters.search.placeholder": "Αναζητήστε νέα...",
  // ... άλλες μεταφράσεις components στα Ελληνικά
},
};

```

Στο τέλος του αρχείου, οι μεταφράσεις από τις τρεις κατηγορίες συνδυάζονται για κάθε γλώσσα, χρησιμοποιώντας τον **spread operator** (...). Αυτό δημιουργεί ένα ενιαίο αντικείμενο `translations` για κάθε γλώσσα, το οποίο μπορεί να χρησιμοποιηθεί εύκολα στον κώδικα.

```

export const translations = {
  en: {
    ...PAGE_TRANSLATIONS.en,
    ...NAV_LINK_TRANSLATIONS.en,
    ...COMPONENT_TRANSLATIONS.en,
  },
  el: {
    ...PAGE_TRANSLATIONS.el,
    ...NAV_LINK_TRANSLATIONS.el,
    ...COMPONENT_TRANSLATIONS.el,
  },
} as const;

```

Έπειτα με τη χρήση του `useTranslations` utility που περιγράψαμε στην ενότητα 3.10, μπορεί να χρησιμοποιηθεί για την πρόσβαση στις μεταφράσεις της τρέχουσα γλώσσας.

Παράδειγμα:

```

import { useTranslations } from "@i18n/utils";

const t = useTranslations(lang);

function HomePage() {
  return (
    <div>
      <h1>{t("page.home.title")}</h1>
      <p>{t("page.home.subtitle")}</p>
      <button>{t("page.home.button.team")}</button>
    </div>
  );
}

```

```

    </div>
  );
}

```

### Προσθήκη Νέων Μεταφράσεων

Για να προσθέσετε νέες μεταφράσεις, ακολουθήστε τα παρακάτω βήματα:

1. Εντοπίστε το κλειδί της μετάφρασης:
  - ο Επιλέξτε ένα μοναδικό κλειδί που θα αντιπροσωπεύει το κείμενο που θέλετε να μεταφράσετε. Συνήθως, χρησιμοποιείται μια συμβολοσειρά που περιγράφει τη θέση και τον σκοπό του κειμένου, όπως `"page.home.button.text.submit"` ή `"page.home.title"`.
2. Προσθέστε το κλειδί στο αρχείο μεταφράσεων:
  - ο Στο `translations.ts`, προσθέστε το νέο κλειδί στις αντίστοιχες γλώσσες.
3. Παράδειγμα Προσθήκης Νέας Μετάφρασης για ένα Κουμπί σε Μια Σελίδα:

```

// Στο PAGE_TRANSLATIONS
en: {
  // ... υπάρχουσες μεταφράσεις
  "page.home.button.learn-more": "Learn More",
},
el: {
  // ... υπάρχουσες μεταφράσεις
  "page.home.button.learn-more": "Μάθετε Περισσότερα",
},

```

4. Χρήση της Νέας Μετάφρασης στον Κώδικα

```

function HomePage() {
  return (
    <button>{t("page.home.button.learn-more")}</button>
  );
}

```

## 5. Προκλήσεις και Λύσεις

Κατά τη διάρκεια της ανάπτυξης του έργου, αντιμετωπίστηκαν διάφορες προκλήσεις που αφορούσαν τη διαχείριση του περιεχομένου, τη σχεδίαση του UI, την υποστήριξη πολυγλωσσικότητας και την ανάπτυξη του ιστότοπου. Σε αυτό το κεφάλαιο, θα παρουσιαστούν οι κύριες προκλήσεις και οι λύσεις που εφαρμόστηκαν για την αντιμετώπισή τους.

### 5.1 Διαχείριση Περιεχομένου από Μη Τεχνικούς Χρήστες

#### Πρόβλημα:

Η εύρεση ενός CMS (Content Management System) που να επιτρέπει σε μη τεχνικούς χρήστες να διαχειρίζονται εύκολα το περιεχόμενο και τις αλλαγές στον ιστότοπο ήταν μια σημαντική πρόκληση. Η ανάγκη ήταν να υπάρχει ένα απλό και φιλικό προς τον χρήστη εργαλείο που να διευκολύνει την ενημέρωση του περιεχομένου χωρίς να απαιτεί γνώσεις προγραμματισμού.

#### Λύση:

Χρησιμοποιήθηκαν τα **Content Collections** του Astro, τα οποία επιτρέπουν την οργάνωση και διαχείριση του περιεχομένου μέσω αρχείων Markdown με Front Matter. Αυτή η προσέγγιση είναι απλή και δεν προσθέτει πολυπλοκότητα στο έργο. Επιπλέον, σχεδιάστηκε με γνώμονα αργότερα να συνδυαστεί με το **FrontMatter CMS**, που προσφέρει ένα γραφικό περιβάλλον για τη διαχείριση του περιεχομένου. Με αυτόν τον τρόπο, οι μη τεχνικοί χρήστες θα μπορούν να επεξεργάζονται και να ενημερώνουν το περιεχόμενο μέσω ενός φιλικού προς τον χρήστη UI, χωρίς να χρειάζεται να επεξεργάζονται απευθείας τα αρχεία του έργου.

## 5.2 Πρόκληση: Δημιουργία Responsive UI

### Πρόβλημα:

Η ανάπτυξη ενός responsive UI που να εμφανίζεται σωστά σε όλες τις συσκευές και αναλύσεις ήταν μια πρόκληση. Έπρεπε να διασφαλιστεί ότι ο ιστότοπος είναι αισθητικά ευχάριστος και λειτουργικός τόσο σε επιτραπέζιους υπολογιστές όσο και σε κινητές συσκευές.

### Λύση:

Χρησιμοποιήθηκαν τα **utilities του Tailwind CSS** για τη δημιουργία responsive σχεδίασης. Το Tailwind προσφέρει ευέλικτες κλάσεις που διευκολύνουν την εφαρμογή στυλ ανάλογα με το μέγεθος της οθόνης. Επιπλέον, αξιοποιήθηκαν τα **components του shaden**, που είναι προσαρμόσιμα και συμβάλλουν στη δημιουργία συνεπών και responsive UI. Αυτή η συνδυασμένη προσέγγιση επέτρεψε την ταχεία ανάπτυξη ενός UI που προσαρμόζεται ομαλά σε διαφορετικές συσκευές και αναλύσεις.

## 5.3 Πρόκληση: Διαχείριση Πολυγλωσσικότητας και Μεταφράσεων

### Πρόβλημα:

Η εύρεση ενός εύχρηστου τρόπου για τη διαχείριση των μεταφράσεων αποτέλεσε πρόκληση. Αν και το `i18n` στο Astro παρέχει ευκολίες, η υλοποίησή του οδήγησε σε μεγαλύτερη πολυπλοκότητα στην οργάνωση του περιεχομένου και των σελίδων. Επιπλέον, οι μεταφράσεις των κειμένων διαχειρίζονται μέσω ενός αρχείου, γεγονός που μπορεί να μην είναι τόσο απλό για τους μη τεχνικούς χρήστες.

### Λύση:

Η καλύτερη λύση που εφαρμόστηκε ήταν η χρήση του **`i18n` του Astro**. Παρά την πολυπλοκότητα που προσθέτει, παρέχει ένα ισχυρό πλαίσιο για την υποστήριξη πολλαπλών γλωσσών. Η οργάνωση των αρχείων περιεχομένου σε υποφακέλους ανά γλώσσα και η χρήση των μεταφράσεων μέσω του αρχείου `translations.ts` επέτρεψαν την αποτελεσματική διαχείριση της πολυγλωσσικότητας. Αν και υπάρχει περιθώριο βελτίωσης, όπως η εξερεύνηση πιο φιλικών προς τον χρήστη εργαλείων ή μεθόδων για τη διαχείριση των μεταφράσεων, η τρέχουσα προσέγγιση καλύπτει τις ανάγκες του έργου.

## 5.4 Πρόκληση: Deployment και CI/CD

### Πρόβλημα:

Η φιλοξενία του ιστότοπου στους servers του πανεπιστημίου παρουσίασε μια πρόκληση, καθώς δεν υπήρχε σαφής εικόνα των δυνατοτήτων και των περιορισμών του συστήματος φιλοξενίας. Χωρίς πληροφορίες σχετικά με το ποιες τεχνολογίες ή υπηρεσίες υποστηρίζονται στους πανεπιστημιακούς servers, έπρεπε να βρεθεί ένας τρόπος να διασφαλιστεί ότι ο ιστότοπος θα μπορεί να αναπτυχθεί και να λειτουργεί σωστά σε αυτό το περιβάλλον.

## Λύση:

Επιλέχθηκε η χρήση της αρχιτεκτονικής **Static Site Generation (SSG)** για τον ιστότοπο, η οποία παράγει στατικά αρχεία HTML, CSS και JavaScript. Αυτή η προσέγγιση έχει το πλεονέκτημα ότι τα στατικά αρχεία είναι ελαφριά, ασφαλή και μπορούν να φιλοξενηθούν εύκολα σε οποιοδήποτε web server, χωρίς την ανάγκη για ειδικές ρυθμίσεις ή υποστήριξη server-side τεχνολογιών.

Το σχέδιο για να απλοποιηθεί η διαδικασία του deployment και να αυτοματοποιηθεί η συνεχής ολοκλήρωση και παράδοση (CI/CD), είναι να δημιουργηθεί ένα **GitHub Workflow**. Αυτό το workflow θα ενεργοποιείται κάθε φορά που γίνεται **push** ή **merge** στο κύριο branch (π.χ., **main** ή **master**) του αποθετηρίου. Το workflow αυτοματοποιεί τη διαδικασία ως εξής:

1. **Δημιουργία της παραγωγικής έκδοσης του ιστότοπου:**
  - Εκτελεί την εντολή `npm run build` για να δημιουργήσει τα στατικά αρχεία στον φάκελο `dist/`.
2. **Μεταφορά των αρχείων στον server:**
  - Χρησιμοποιεί εργαλεία όπως το `scp` ή `rsync` για να μεταφέρει τα παραγόμενα αρχεία στους servers του πανεπιστημίου μέσω ασφαλούς σύνδεσης (SSH).
3. **Ενημέρωση των servers:**
  - Αν χρειάζεται, εκτελεί εντολές ή script στον server για την ανανέωση της προσωρινής μνήμης (cache) ή την επανεκκίνηση υπηρεσιών.

Με αυτόν τον τρόπο, κάθε αλλαγή που γίνεται στον κώδικα και γίνεται merge στο κύριο branch αναπτύσσεται αυτόματα στον παραγωγικό ιστότοπο, εξασφαλίζοντας ότι ο ιστότοπος είναι πάντα ενημερωμένος.

## Πλεονεκτήματα αυτής της προσέγγισης:

- **Απλότητα:** Η χρήση στατικών αρχείων καθιστά το deployment απλό, καθώς δεν απαιτούνται πολύπλοκες ρυθμίσεις server ή εγκατάσταση εξειδικευμένου λογισμικού.
- **Αυτοματοποίηση:** Η χρήση του **GitHub Actions** για το CI/CD εξαλείφει την ανάγκη για χειροκίνητο deployment, μειώνοντας τα σφάλματα και εξοικονομώντας χρόνο.
- **Ευελιξία:** Αν στο μέλλον αλλάξουν οι απαιτήσεις ή οι δυνατότητες των servers του πανεπιστημίου, η αρχιτεκτονική SSG επιτρέπει την εύκολη μεταφορά του ιστότοπου σε άλλες πλατφόρμες φιλοξενίας.



## 6. Συμπεράσματα

Το έργο αυτό είχε ως βασικό στόχο την ανάπτυξη ενός σύγχρονου, πολυγλωσσικού και responsive ιστότοπου για την παρουσίαση νέων, έργων και δημοσιεύσεων σε διάφορους ερευνητικούς τομείς. Κατά τη διάρκεια της ανάπτυξης, αντιμετωπίστηκαν και ξεπεράστηκαν διάφορες προκλήσεις, όπως η διαχείριση περιεχομένου από μη τεχνικούς χρήστες, η δημιουργία ενός responsive UI, η υποστήριξη πολυγλωσσικότητας και η αυτοματοποίηση της διαδικασίας ανάπτυξης.

Με τη χρήση του **Astro** για τη στατική δημιουργία σελίδων, του **Tailwind CSS** για την ευέλικτη και responsive σχεδίαση, και των **content collections** για την οργάνωση του περιεχομένου, έγινε εφικτή η δημιουργία ενός ιστότοπου που είναι εύκολος στη διαχείριση και προσφέρει μια άριστη εμπειρία χρήστη. Η ενσωμάτωση του **i18n** επέτρεψε την υποστήριξη πολλαπλών γλωσσών, καθιστώντας τον ιστότοπο προσβάσιμο σε ένα ευρύτερο κοινό.

Η υιοθέτηση της αρχιτεκτονικής **Static Site Generation (SSG)** και η και το πλάνο για αυτοματοποίηση της διαδικασίας ανάπτυξης μέσω ενός **GitHub Workflow** εξασφαλίζουν ότι ο ιστότοπος μπορεί να φιλοξενηθεί εύκολα στους servers του πανεπιστημίου και ότι οι ενημερώσεις μπορούν να γίνονται γρήγορα και αποτελεσματικά.

Η ανάπτυξη αυτού του έργου παρείχε πολύτιμη εμπειρία στην αντιμετώπιση πραγματικών προβλημάτων και στην εφαρμογή σύγχρονων τεχνολογιών και πρακτικών. Το τελικό αποτέλεσμα είναι ένας ιστότοπος που όχι μόνο καλύπτει τις τρέχουσες ανάγκες αλλά έχει και τη δυνατότητα να επεκταθεί και να προσαρμοστεί σε μελλοντικές απαιτήσεις.



## 7. Βιβλιογραφία

- Astro. (n.d.). *Astro Docs*. Getting Started | Docs. Retrieved September 20, 2024, from <https://docs.astro.build/en/getting-started/>
- Astro i18n. (n.d.). *Internationalization (i18n) Routing* | Docs. Astro Docs. Retrieved September 11, 2024, from <https://docs.astro.build/en/guides/internationalization/>
- Directus. (n.d.). *Directus Docs*. Directus Docs. Retrieved September 20, 2024, from <https://docs.directus.io/>
- Keystatic. (n.d.). *Introduction - Docs*. Keystatic. Retrieved September 20, 2024, from <https://keystatic.com/docs/introduction>
- Markdown Guide. (n.d.). *Markdown Guide*. Markdown Guide. Retrieved September 15, 2024, from <https://www.markdownguide.org/>
- MDN. (2024, September 1). *i18n - Mozilla* | MDN. MDN Web Docs. Retrieved September 15, 2024, from <https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/i18n>
- patterns. (n.d.). *Islands Architecture*. Patterns.dev. Retrieved September 15, 2024, from <https://www.patterns.dev/vanilla/islands-architecture/>
- patterns. (n.d.). *Server-side Rendering*. Patterns.dev. Retrieved September 15, 2024, from <https://www.patterns.dev/react/server-side-rendering/>
- patterns. (n.d.). *Static Rendering*. Patterns.dev. Retrieved September 15, 2024, from <https://www.patterns.dev/react/static-rendering>
- Payload. (n.d.). *What is Payload? | Documentation* | Payload. Payload CMS. Retrieved September 20, 2024, from <https://payloadcms.com/docs/getting-started/what-is-payload>
- React. (n.d.). *Quick Start – React*. React. Retrieved September 20, 2024, from <https://react.dev/learn>

- React. (n.d.). *React Reference Overview – React*. React. Retrieved September 15, 2024, from <https://react.dev/reference/react>
- shadcn. (n.d.). *Introduction*. shadcn/ui. Retrieved September 15, 2024, from <https://ui.shadcn.com/docs>
- Struyf, E. (2024, June 20). *Documentation | Front Matter*. Front Matter CMS. Retrieved September 20, 2024, from <https://frontmatter.codes/docs>
- Tailwind CSS. (n.d.). *Tailwind CSS*. Tailwind CSS - Rapidly build modern websites without ever leaving your HTML. Retrieved September 15, 2024, from <https://tailwindcss.com/>
- Typescript. (n.d.). *Documentation - TypeScript for JavaScript Programmers*. TypeScript. Retrieved September 15, 2024, from <https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html>
- W3C. (n.d.). *WCAG 2 Overview | Web Accessibility Initiative (WAI)*. W3C. Retrieved September 20, 2024, from <https://www.w3.org/WAI/standards-guidelines/wcag/>