Γεια σου! Πολύ ενδιαφέρουσα και μοντέρνα η προσέγγισή σου για την ανάπτυξη του web app. Το "vibe coding" ή "agent-driven development" είναι το μέλλον της δημιουργίας λογισμικού, και το project σου είναι ιδανικό για να το εφαρμόσεις.

Ανέλυσα το πλάνο που ανέβασες και κατανόησα πλήρως τον στόχο σου: να δημιουργήσεις μια διαδικασία όπου ένας LLM agent αναλαμβάνει όλη τη βαριά δουλειά—από την οργάνωση των αρχείων μέχρι τη δημιουργία του τελικού website—με εσένα να έχεις τον ρόλο του project manager που δίνει τις οδηγίες.

Ας δούμε αναλυτικά τις τεχνολογίες και τη μεθοδολογία που σου προτείνω.

**1. Το Τεχνολογικό Stack του Web App**

Για το τελικό προϊόν (τη στατική ιστοσελίδα), ο στόχος είναι να επιλέξουμε τεχνολογίες που είναι εύκολες για έναν LLM agent να τις γράψει, να τις συντηρήσει και να παράγουν ένα γρήγορο, μοντέρνο αποτέλεσμα.

|  |  |  |
| --- | --- | --- |
| **Κατηγορία** | **Τεχνολογία** | **Γιατί είναι ιδανική για το project σου** |
| **Frontend Framework** | **Next.js (React)** | Είναι το κορυφαίο framework για τη δημιουργία στατικών ιστοσελίδων (Static Site Generation - SSG). Ο agent μπορεί εύκολα να δημιουργήσει components για κάθε ενότητα (π.χ., γκαλερί εικόνων, video player) και το Next.js θα τα συνθέσει σε μια βελτιστοποιημένη, στατική σελίδα. |
| **Styling** | **Tailwind CSS** | Αυτή είναι η πιο κρίσιμη επιλογή για "vibe coding". Το Tailwind CSS επιτρέπει στον agent να εφαρμόζει στυλ απευθείας στο HTML με utility classes. Είναι πολύ πιο εύκολο να δώσεις οδηγίες όπως "Κάνε αυτό το κουτί σκούρο γκρι με στρογγυλεμένες γωνίες" και ο agent να γράψει 2-3 κλάσεις. |
| **Πηγή Περιεχομένου** | **Markdown / MDX** | Για να διαχωρίσεις το περιεχόμενο από τον κώδικα, ο agent μπορεί να μετατρέπει τα αρχικά σου αρχεία (Word, κείμενα) σε αρχεία Markdown (.md) ή MDX (.mdx). Το MDX είναι ακόμα πιο ισχυρό, γιατί σου επιτρέπει να εισάγεις React components μέσα στο κείμενό σου. |
| **Deployment** | **Vercel / Netlify** | Και οι δύο πλατφόρμες προσφέρουν άψογη ενσωμάτωση με το GitHub. Μόλις ο agent κάνει push τον κώδικα στο repository, η Vercel ή η Netlify αυτόματα θα κάνει build και θα δημοσιεύσει την ιστοσελίδα σου, χωρίς να χρειαστεί καμία δική σου ενέργεια. |

**Σύνοψη Stack:** **Next.js + Tailwind CSS + MDX**, με deployment στη **Vercel**. Αυτός ο συνδυασμός είναι ο "χρυσός κανόνας" για σύγχρονες, στατικές ιστοσελίδες και είναι εξαιρετικά φιλικός προς τους LLM agents.

**2. Ο LLM Agent και η Ενσωμάτωσή του**

Εδώ είναι η καρδιά του project σου. Χρειάζεσαι ένα περιβάλλον που να δίνει στον LLM "χέρια και μάτια" για να αλληλεπιδρά με τον υπολογιστή σου.

**Εργαλεία και Πλατφόρμες**

1. **Open Interpreter (Προτεινόμενο για τις εργασίες αρχείων)**
   * **Τι είναι:** Είναι ένα open-source εργαλείο που σου επιτρέπει να εκτελείς LLMs τοπικά, δίνοντάς τους πρόσβαση στο terminal (CLI) του υπολογιστή σου.
   * **Πώς θα το χρησιμοποιήσεις:** Θα το εγκαταστήσεις στον υπολογιστή σου και θα του δίνεις εντολές σε φυσική γλώσσα, όπως ακριβώς περιγράφεις στο **ΒΗΜΑ 1**. Για παράδειγμα:

"Κοίτα τον φάκελο ~/Documents/Epirus\_Project\_X. Πάρε όλες τις εικόνες .jpg και .png και βάλε τις σε έναν νέο φάκελο images. Πάρε τα αρχεία .mp4 και βάλε τα στον φάκελο videos. Δημιούργησε ένα αρχείο summary.mdx και γράψε μέσα μια περίληψη του περιεχομένου του info.docx."

* + **Ασφάλεια:** Το Open Interpreter σου ζητάει πάντα επιβεβαίωση πριν εκτελέσει οποιαδήποτε εντολή στο σύστημά σου, κάνοντάς το ασφαλές στη χρήση.

1. **Cursor ή VS Code με GitHub Copilot (Για τη συγγραφή κώδικα)**
   * **Τι είναι:**
     + **Cursor:** Ένας code editor φτιαγμένος από την αρχή για AI development. Έχει ενσωματωμένο chat που "γνωρίζει" όλη τη βάση του κώδικά σου και μπορεί να κάνει αλλαγές σε πολλαπλά αρχεία ταυτόχρονα.
     + **VS Code + GitHub Copilot Extension:** Το κλασικό VS Code με την πανίσχυρη επέκταση του Copilot. Σου δίνει εκπληκτικές δυνατότητες auto-completion και ένα chat panel για να ζητάς αλλαγές.
   * **Πώς θα τα χρησιμοποιήσεις:** Αφού το Open Interpreter οργανώσει τα αρχεία (ΒΗΜΑ 1), θα ανοίξεις τον φάκελο του project στον Cursor ή στο VS Code και θα αρχίσεις να δίνεις εντολές για το **ΒΗΜΑ 2**. Για παράδειγμα:

"Φτιάξε ένα νέο React component με όνομα ProjectGallery.tsx που να παίρνει μια λίστα από image URLs και να τις εμφανίζει σε ένα grid."

"Στην κεντρική σελίδα, διάβασε το αρχείο summary.mdx και εμφάνισε το περιεχομένό του. Ακριβώς από κάτω, χρησιμοποίησε το ProjectGallery component για να δείξεις τις εικόνες από τον φάκελο images."

**3. Συνολική Ροή Εργασίας (Workflow)**

Ας συνδυάσουμε τα παραπάνω σε ένα πρακτικό σενάριο:

1. **Προετοιμασία:**
   * Εγκαθιστάς το Node.js, το VS Code (με Copilot) ή το Cursor, και το Open Interpreter.
   * Δημιουργείς ένα νέο, κενό repository στο GitHub.
2. **Βήμα 0: Δημιουργία Project (μέσω Agent)**
   * Ανοίγεις το terminal και λες στο Open Interpreter:

"Create a new Next.js project with TypeScript and Tailwind CSS in a folder named 'epirus-projects-showcase'. Initialize a git repository and connect it to my GitHub repo [link]."

1. **Βήμα 1: Οργάνωση Περιεχομένου (μέσω Agent)**
   * Τοποθετείς τα "χύμα" αρχεία ενός έργου σε έναν φάκελο.
   * Λες στο Open Interpreter:

"Inside the 'epirus-projects-showcase/public/' directory, create a new folder for 'project-a'. Look at my files in '~/Downloads/raw\_project\_a', organize them into images, videos, and documents subfolders inside 'public/project-a'. Create a main content file index.mdx."

1. **Βήμα 2: Ανάπτυξη του UI (μέσω Agent)**
   * Ανοίγεις τον φάκελο epirus-projects-showcase στον Cursor/VS Code.
   * Ξεκινάς το "vibe coding", δίνοντας οδηγίες στο chat:

"Build the main page to read all project folders from /public. For each project, display a card with its title."

"When a user clicks on a project card, navigate to a dynamic page for that project. This page should render the content from the project's index.mdx file and display its images and videos."

"The navigation menu on top should have links that scroll to the 'About', 'Projects', and 'Contact' sections of the single page."

1. **Βήμα 3: Deployment (Αυτόματο)**
   * Συνδέεις το GitHub repository σου με τη Vercel.
   * Κάθε φορά που λες στον agent "Commit and push the changes to GitHub", η Vercel θα αναλαμβάνει αυτόματα να κάνει deploy την τελευταία έκδοση του site σου.