**Στρατηγική Υλοποίησης Web App για την Περιφέρεια Ηπείρου με LLM Agent**

Αυτό το έγγραφο περιγράφει τη βέλτιστη στρατηγική για την ανάπτυξη μιας στατικής ιστοσελίδας παρουσίασης πολυμεσικών αρχείων, χρησιμοποιώντας ένα μεγάλο γλωσσικό μοντέλο (LLM) ως αυτόνομο "developer agent". Ο στόχος είναι η μηδενική συγγραφή κώδικα από τον developer.

**1. Η Έννοια του "Agent" και η Σωστή Προσέγγιση**

Αντί να αντιμετωπίζεις το LLM σαν ένα απλό εργαλείο παραγωγής κώδικα, πρέπει να το δεις ως έναν **βοηθό-εκτελεστή (assistant-executor)**. Εσύ θα είσαι ο **"Project Manager"** που θα δίνει σαφείς, διαδοχικές εντολές, και το LLM θα εκτελεί. Η διαδικασία δεν θα είναι μία μοναδική εντολή, αλλά μια **συζήτηση καθοδήγησης**.

Ο "agent" δεν θα μπορεί (προς το παρόν) να αποκτήσει απευθείας πρόσβαση στο τοπικό σου σύστημα αρχείων (CLI) για λόγους ασφαλείας. Η λύση είναι να του ζητήσεις να γράψει **scripts** (π.χ., σε Python) που *εσύ* θα εκτελείς τοπικά για να κάνεις τις απαραίτητες ενέργειες στο σύστημα αρχείων.

**2. Προτεινόμενο Technology Stack (Βέλτιστο για LLM)**

Η απλότητα είναι το κλειδί. Ένα LLM μπορεί να διαχειριστεί πιο εύκολα τεχνολογίες που δεν απαιτούν περίπλοκο setup, compilers ή build steps.

* **Frontend:** **HTML, CSS, και JavaScript (Vanilla JS)**. Ζήτα από το LLM να δημιουργήσει ένα **ενιαίο αρχείο index.html** που θα περιέχει τα πάντα. Αυτό είναι το πιο εύκολο για διαχείριση.
* **Styling:** **Tailwind CSS μέσω CDN**. Αυτή είναι η καλύτερη επιλογή. Δεν χρειάζεται εγκατάσταση. Το LLM μπορεί να γράψει τις κλάσεις του Tailwind απευθείας μέσα στο HTML, δημιουργώντας μοντέρνο και responsive design πολύ γρήγορα.
* **Hosting Αρχείων (Input):** **Public GitHub Repository**. Αυτή είναι η ιδανική λύση. Δημιούργησε ένα public repository για κάθε έργο, ανέβασε εκεί τα "χύμα" πολυμεσικά αρχεία. Έτσι, το LLM μπορεί θεωρητικά να "δει" τη δομή και το περιεχόμενο αν του δώσεις συνδέσμους.
* **Deployment (Output):** **GitHub Pages** ή **Netlify**. Και οι δύο πλατφόρμες προσφέρουν δωρεάν hosting για στατικές σελίδες και συνδέονται απευθείας με το GitHub repository σου, κάνοντας την ανάρτηση του τελικού site υπόθεση ενός git push.

**3. Λεπτομερής Ροή Εργασίας Βήμα προς Βήμα**

Αυτή είναι η ακριβής διαδικασία που θα ακολουθήσεις, δίνοντας εντολές (prompts) στο LLM.

**Φάση 1: Προετοιμασία**

1. Δημιούργησε ένα νέο, public GitHub repository για το συγκεκριμένο έργο (π.χ., ergο-ioanninon-odos-x).
2. Ανέβασε όλα τα πολυμεσικά αρχεία (εικόνες, βίντεο, PDF, κείμενα) σε έναν φάκελο input-media μέσα στο repository.

**Φάση 2: ΒΗΜΑ 1 - Οργάνωση Περιεχομένου (με script)**

**Prompt προς το LLM:**

"Γράψε ένα script σε Python που κάνει τα εξής:

1. Παίρνει ως όρισμα μια διαδρομή φακέλου (source folder).
2. Δημιουργεί έναν νέο φάκελο με όνομα dist (destination).
3. Μέσα στο dist, δημιουργεί υποφακέλους: images, videos, documents.
4. Σαρώνει όλα τα αρχεία στον source folder.
5. Μετακινεί κάθε αρχείο στον κατάλληλο υποφάκελο του dist ανάλογα με την κατάληξή του (π.χ., .jpg, .png -> images, .mp4 -> videos, .pdf, .txt -> documents).
6. Τέλος, δημιουργεί ένα αρχείο manifest.json μέσα στο dist. Αυτό το JSON θα περιέχει μια λίστα με τις σχετικές διαδρομές όλων των αρχείων που οργάνωσε. Παράδειγμα manifest.json:

{  
  "images": ["images/photo1.jpg", "images/photo2.png"],  
  "videos": ["videos/clip1.mp4"],  
  "documents": ["documents/plan.pdf"]  
}  
```"

**Εκτέλεση από εσένα:**

* Αποθηκεύεις τον κώδικα που σου έδωσε το LLM ως organize.py.
* Τρέχεις το script από το terminal σου, δίνοντας ως όρισμα τον φάκελο input-media.
* Αποτέλεσμα: Θα έχεις έναν νέο, καθαρό φάκελο dist με τα πάντα οργανωμένα και το manifest.json.

**Φάση 3: ΒΗΜΑ 2 - Δημιουργία του Web App (το κυρίως μέρος)**

**Prompt προς το LLM (το πιο σημαντικό):**

"Θέλω να δημιουργήσεις μια ολοκληρωμένη, στατική ιστοσελίδα σε ένα μόνο αρχείο index.html.

**Τεχνικές Προδιαγραφές:**

* Χρησιμοποίησε HTML5.
* Ενσωμάτωσε το Tailwind CSS χρησιμοποιώντας το πιο πρόσφατο CDN link μέσα στο <head>.
* Το design να είναι μοντέρνο, καθαρό, και επαγγελματικό, κατάλληλο για κυβερνητικό οργανισμό. Χρησιμοποίησε αποχρώσεις του μπλε και του λευκού.
* Η σελίδα θα είναι single-page, με navigation bar που λειτουργεί με anchor links.

Περιεχόμενο και Δομή:

Η δομή και το περιεχόμενο της σελίδας θα βασιστούν στο παρακάτω manifest.json. Πάρε τα δεδομένα από αυτό το JSON και δημιούργησε τις αντίστοιχες ενότητες (sections) στη σελίδα.

**(Εδώ κάνεις επικόλληση το περιεχόμενο του manifest.json που δημιούργησες στο προηγούμενο βήμα)**

**Απαιτούμενες Ενότητες:**

1. **Header:** Ένα σταθερό (sticky) navigation bar στην κορυφή με το λογότυπο της Περιφέρειας Ηπείρου (χρησιμοποίησε ένα placeholder) και links προς τις ενότητες: "Εικόνες", "Βίντεο", "Έγγραφα".
2. **Hero Section:** Μια μεγάλη, εντυπωσιακή εικόνα (χρησιμοποίησε την πρώτη από τη λίστα images) με τον τίτλο του έργου.
3. **Ενότητα Εικόνων (id="images"):** Δημιούργησε μια responsive photo gallery (grid) που θα εμφανίζει όλες τις εικόνες από τη λίστα images του JSON. Όταν ο χρήστης κάνει κλικ σε μια εικόνα, να ανοίγει σε lightbox (χρησιμοποίησε απλό JavaScript για αυτό).
4. **Ενότητα Βίντεο (id="videos"):** Για κάθε βίντεο στη λίστα videos, ενσωμάτωσέ το χρησιμοποιώντας το HTML <video> tag με controls.
5. **Ενότητα Εγγράφων (id="documents"):** Για κάθε έγγραφο στη λίστα documents, δημιούργησε ένα όμορφο card component που θα δείχνει το όνομα του αρχείου και ένα κουμπί "Download" που θα οδηγεί στο αρχείο.
6. **Footer:** Ένα απλό footer με πληροφορίες πνευματικών δικαιωμάτων.

Γράψε τον πλήρη κώδικα για το αρχείο index.html."

**Εκτέλεση από εσένα:**

* Αντιγράφεις τον κώδικα που θα σου δώσει το LLM.
* Τον αποθηκεύεις ως index.html μέσα στον φάκελο dist.
* Ανοίγεις το αρχείο σε έναν browser για να δεις το αποτέλεσμα.

**Φάση 4: Deployment**

1. Κάνε commit και push ολόκληρο τον φάκελο dist (μαζί με το index.html και τους υποφακέλους πολυμέσων) στο GitHub repository σου.
2. Στις ρυθμίσεις του repository, πήγαινε στην καρτέλα "Pages".
3. Επίλεξε το main branch και τον φάκελο /dist ως πηγή (source).
4. Το site σου είναι πλέον online!

**4. Χρήση APIs - Είναι απαραίτητα;**

**Όχι.** Για τον σκοπό που περιγράφεις (στατική παρουσίαση αρχείων), **δεν χρειάζεσαι κανένα εξωτερικό API**. Η προσέγγιση με το manifest.json που παράγεται τοπικά είναι πολύ πιο απλή, γρήγορη και απόλυτα ευθυγραμμισμένη με τον στόχο της δημιουργίας μιας στατικής σελίδας από ένα LLM. Η χρήση APIs θα προσέθετε πολυπλοκότητα (authentication, async calls) που δεν είναι απαραίτητη εδώ.

Ακολουθώντας αυτή τη δομημένη προσέγγιση, μπορείς να αξιοποιήσεις πλήρως τις δυνατότητες ενός LLM για να αυτοματοποιήσεις όλη τη διαδικασία ανάπτυξης, διατηρώντας εσύ τον στρατηγικό έλεγχο.