



AI Website Prompt Library and Release Workbook

Prompt chains, review rubrics, debugging procedures, and release
controls for reliable AI-assisted web projects

How to use the prompt chains

A single giant prompt rarely produces a maintainable website. This workbook breaks the work into controlled stages: discovery, architecture, implementation, review, repair, and release. Each stage has required inputs, a reusable prompt, an expected output, and a decision gate.

Use the chains in order. Do not continue merely because the AI produced code. Continue only when the output passes the stated review.

Core operating rules

1. Give the assistant the exact project tree and complete affected files.
2. Ask it to identify assumptions and risks before writing code.
3. Request one coherent change group.
4. Require complete replacement files or a precise patch, never an unclear mixture.
5. Compare the result with the previous state.
6. Test locally and on the deployed site.
7. Save a known-good commit before the next chain.

Prompt placeholder rule. Replace every value in brackets. A prompt containing `[INSERT FILE HERE]` is incomplete and should not be sent.

Contents

- | | |
|-----------------------------------|--|
| 1. Project brief worksheet | 13. Regression-control chain |
| 2. Context packet builder | 14. Release-readiness chain |
| 3. Planning and page-map chain | 15. Maintenance and refactoring chain |
| 4. File-architecture chain | 16. Complete worked example |
| 5. Semantic HTML chain | 17. Evaluation rubrics |
| 6. Responsive CSS chain | 18. Product landing-page chain |
| 7. JavaScript behavior chain | 19. Measurement and funnel chain |
| 8. Bilingual parity chain | 20. Migration and rollback chain |
| 9. Accessibility audit chain | 21. Incident-response chain |
| 10. SEO and content-quality chain | 22. Prompt anti-pattern repair library |
| 11. Security and privacy chain | 23. Change log and test record |
| 12. Debugging and 404 chain | |

1. Project brief worksheet

Complete this before asking for code.

Field	Your answer
Project name	
Primary audience	
Main visitor problem	
Primary action	
Required pages	
Required languages	
Hosting platform	
Existing domain	
Existing files	
Essential features	
Features that must not be added	
Visual direction	
Accessibility requirements	
Privacy constraints	
Maintenance owner	

Outcome statement

Fill in this sentence:

This website helps **[audience]** accomplish **[outcome]** by providing **[content or service]**. The main action is **[action]**, and success means **[measurable result]**.

Scope boundary

Write three lists:

Must ship: items required for the first public release.

Can wait: useful ideas that should not delay launch.

Will not build: unsafe, irrelevant, expensive, or distracting features.

2. Context packet builder

AI needs a current snapshot, not a memory of earlier messages. Assemble this packet each time the project changes significantly.

```
PROJECT PURPOSE
[One paragraph]

HOSTING
[GitHub Pages project site, custom domain, or other host]

FILE TREE
[Paste output of: find . -maxdepth 4 -type f | sort]

AFFECTED FILES
[Paste complete current contents]

REQUESTED CHANGE
[One coherent change group]

INVARIANTS
[Features, IDs, URLs, design tokens, languages, analytics consent, or other requirements that must remain]

KNOWN PROBLEMS
[Current bugs and exact error messages]

OUTPUT CONTRACT
[Complete replacement files, no omissions, followed by a test list]
```

Context quality gate

Do not send the packet until:

- ☐ The file tree reflects the repository's current state.
- ☐ No file is truncated with "rest unchanged."
- ☐ Secrets and personal data are removed.
- ☐ The requested change is testable.
- ☐ Existing behavior that must remain is explicitly listed.

3. Planning and page-map chain

Prompt A: challenge the idea

Act as a product-focused web architect. Do not write code yet.

Project brief:

[PASTE BRIEF]

Identify:

1. The primary user journey.
2. Missing information that blocks a responsible design.
3. Pages that are necessary at launch.
4. Pages or features that should be postponed.
5. Trust, accessibility, privacy, and maintenance risks.
6. A measurable definition of launch success.

Return a concise decision document and ask only questions that materially affect architecture.

Review gate

Reject the output if it adds features unrelated to the visitor's goal, assumes a backend where none exists, or treats branding as more important than content and navigation.

Prompt B: create the page map

Using the approved brief and answers below, create a page map for a small static website.

[PASTE APPROVED DECISION DOCUMENT]

[PASTE ANSWERS]

For every page provide:

- Purpose.
- Primary audience question.
- Required sections in reading order.
- Primary action.
- Internal links to and from the page.
- English and Greek content requirements.
- Acceptance criteria.

Do not write HTML yet.