# Frontend Development Ideation & Storyboarding — Deep Dive

Generated: 2025-10-21 17:13:04 +0000

Format: Creative Developer Ideation Style — Detailed mockups, ASCII diagrams, component specs, interaction tables, and storyboard frames.

## A. Project Ideation (Expanded)

This document is a comprehensive creative-technical ideation and storyboard for the Frontend Development Module. It is written to guide teams (students and mentors) through in-depth UX thinking, component design, prototyping, interaction design, accessibility planning, testing, and reflection. The tone is practical and developer-friendly.

### A1. Creative Vision

Vision Statement:  
Create a frontend learning experience that is joyful to build and clear to use — where every visual decision communicates intent, and every interaction teaches a web development concept (layout, responsiveness, accessibility, or event-driven behavior).

#### A2. Guiding Principles

- Empathy-first: start designs by specifying user needs and constraints.

- Progressive complexity: begin simple, layer features progressively.

- Visual hierarchy: prioritize information using spacing, size, and contrast.

- Accessibility by design: WCAG-compliant choices from the start.

- Component reusability: design components to be modular and testable.

### A3. Target Personas (detailed)

Persona 1 — 'Visual Vicky' (Visual & Creative Learner)

- Age: 16-18  
- Goals: Produce a beautiful portfolio piece; learn to convert Figma to HTML/CSS  
- Pain points: Overwhelmed by code syntax; frightened of breakpoints  
- Motivations: Aesthetic control, portfolio presentation, peer recognition  
- Success metric: A pixel-accurate page that responds gracefully on mobile

Persona 2 — 'Tech Tariq' (Logical & Technical Learner)

- Age: 16-18  
- Goals: Implement robust, maintainable UI with JS behaviors  
- Pain points: Design choices feel subjective; accessibility is not intuitive  
- Motivations: Clean code, bug-free interactivity, learning best practices  
- Success metric: Componentized, accessible UI with unit-testable JS behaviors

### A4. Empathy Map (summary)

Says: "I want my site to look professional."  
Thinks: "How do I make this responsive across devices?"  
Does: Sketches in Figma, searches templates, tests in DevTools  
Feels: Nervous about code, excited about creative control

### A5. Brainstorming & Ideation Artifacts

Use these artifacts during ideation workshops. Encourage rapid sketching and low-fidelity prototypes first, then iterate:  
- 3 rapid wireframe sketches per team (mobile, tablet, desktop)  
- 'How Might We' questions to spark feature ideas (HMW make navigation feel lightweight?)  
- A prioritized feature list (MVP vs. Nice-to-have)

Sample 'How Might We' prompts:

- HMW enable students to test responsive breakpoints quickly?

- HMW teach semantic HTML through interactive examples?

- HMW integrate accessibility checks as part of the design workflow?

## B. Design Framework & Component Specifications

This section specifies components, recommended CSS patterns, accessibility attributes, and interaction behaviors.

### B1. Style Guide (recommended)

Colors (example palette):  
- Primary: #b8860b (gold)  
- Accent: #0b3b5a (deep blue)  
- Background: #ffffff / dark mode #0f1115  
- Text high contrast: #111111  
  
Typography:  
- Headline: Inter, 700  
- Body: Roboto or system sans, 400  
  
Spacing Scale: 4,8,16,24,32,48

Accessibility notes: Always provide 4.5:1 contrast for body text, 3:1 for large text. Include ARIA labels for interactive widgets.

### B2. Component Catalog (detailed)

Header / Nav: Logo, nav links, theme toggle, mobile hamburger. Behaviors: sticky header, reduced height on scroll.

Hero: Title, subheading, CTA with aria-label, optional hero image. Responsive: stack on narrow screens.

Card Grid: Project cards: image, title, tags, short description, hover state shows actions. Use <figure>/<figcaption>.

Modal / Lightbox: Accessible modal with focus trap, ESC to close, role=dialog.

Form (Contact): Labelled inputs, client-side validation, aria-invalid on errors, success toast.

### B3. Interaction Patterns & JS Contracts

Define small JS contracts (input -> output) for common interactions. Example:  
  
Theme Toggle  
Contract: click theme toggle -> set data-theme attribute on <html> -> persist choice to localStorage  
Edge cases: system theme change, invalid stored value  
  
Modal Open  
Contract: trigger element (data-modal-target) -> focus trap, aria-hidden toggle on main content, restore focus on close

### B4. Example HTML/CSS Snippets (teaching aids)

HTML (Hero):

<section class="hero">  
 <h1>Frontend Development Module</h1>  
 <p>Design. Code. Reflect.</p>  
 <button class="cta" aria-label="Start the module">Start</button>  
</section>

CSS (Responsive Grid):

.grid {  
 display: grid;  
 grid-template-columns: repeat(auto-fit, minmax(220px, 1fr));  
 gap: 16px;  
}

## C. Storyboarding — Expanded Frames

Below are multiple storyboard frames with ASCII-style mockups and descriptions. Each frame includes: purpose, key UI elements, interactions, accessibility notes, and testing checklist.

### Frame 1 — Landing / Onboarding

Purpose: Welcome the learner, set expectations, show quick start CTA.

Mockup:

+------------------------------------------------------------+  
| LOGO | Home | Lessons | Projects | About | Sign In |  
+------------------------------------------------------------+  
| HERO IMAGE / ILLUSTRATION |  
| Title: Frontend Module |  
 | Subtitle: Build a responsive portfolio |  
| [Start Lesson] [View Projects] |  
+------------------------------------------------------------+

Interactions: keyboard focus on CTAs, skip-to-content link, responsive image alt text.

Accessibility test: tab order, screen reader label for hero, color contrast check.

### Frame 2 — Lesson Index / Cards

Purpose: Present lessons as cards; quick preview and launch.

Mockup:

+----------------------------------------------------+  
| Filters: [All] [HTML] [CSS] [JS] Search: [\_\_\_\_] |  
+----------------------------------------------------+  
| [Card] [Card] [Card] |  
| Title Title Title |  
| short desc short desc short desc |  
+----------------------------------------------------+

Interactions: card hover shows 'Preview' and 'Start' buttons; keyboard accessible card actions.

Testing checklist: screen reader reads card title and button roles; mobile stacking behavior.

### Frame 3 — Code Playground (Split View)

Purpose: Live-edit HTML/CSS/JS and preview; teaches iterative development.

Mockup:

+----------------------+-----------------------------+  
| Editor (HTML/CSS/JS) | Live Preview |  
| [file.js] [style.css] | <section class="hero">... |  
| line 1 | |  
+----------------------+-----------------------------+

Interactions: auto-preview on save, console logs area, responsive width simulator.

Accessibility: ensure editor is keyboard usable and themes have sufficient contrast for code readability.

### Frame 4 — Project Showcase / Gallery

Purpose: Display student projects; allow peer comments and 'like' feedback.

Mockup:

+------------------------------------------------+  
| Project Card |  
| [Thumbnail] Title | Short description |  
| [View] [Comment] [Like] |  
+------------------------------------------------+

Interactions: comments open accessible modal, likes update asynchronously, view opens project in new tab.

Testing: ARIA roles for dialogs, network failure handling for likes/comments.

## D. Interaction Sequences & State Transition Tables

Detailed sequences for key interactions. Use these as acceptance criteria for implementation and testing.

### D1. Theme Toggle Sequence

Sequence steps:

1. User clicks theme toggle.  
2. JS toggles data-theme on <html>.  
3. Persist selection to localStorage.  
4. Apply CSS variables for colors.  
5. Announce change via aria-live region.

State Table:

State | data-theme | localStorage | UI  
------|-----------|--------------|----  
Light | 'light' | 'light' | Light colors  
Dark | 'dark' | 'dark' | Dark colors

### D2. Modal Dialog State

Sequence steps:

1. Open modal -> set aria-hidden on main -> trap focus -> show animation  
2. Close modal -> restore focus to trigger -> remove aria-hidden

Edge cases: ESC key, click outside, screen resize.

## E. Testing & Accessibility Checklist

- All images have descriptive alt text or aria-hidden if decorative.

- Color contrast meets WCAG 2.1 AA for body text.

- Interactive elements have focus styles and are keyboard operable.

- Forms provide inline validation with aria-invalid and helpful messages.

- Semantic HTML used (header, nav, main, article, footer).

- ARIA roles used only when necessary and properly implemented.

- Local storage interactions handled gracefully (invalid data fallback).

- Responsive breakpoints tested at common widths (320, 375, 768, 1024, 1440).

### E1. Automated QA & Linting Recommendations

- Use Lighthouse for performance/accessibility audits.  
- Use ESLint and stylelint for JS/CSS linting.  
- Add unit tests for JS behavior using Jest (where applicable).

## F. Assessment & Reflection Prompts

Rubric highlights (suggested):  
- Design fidelity (25%) — how closely the implementation matches approved wireframes.  
- Code quality (25%) — semantic structure, modular CSS, JS clarity.  
- Accessibility (20%) — ARIA usage, contrast, keyboard navigation.  
- Interactivity (15%) — robust, tested JS features.  
- Reflection & Documentation (15%) — rationale, screenshots, lessons learned.

Reflection prompts for students:

- What design trade-offs did you make and why?

- How did accessibility concerns affect your layout choices?

- Describe a bug you fixed and what you learned from debugging.

- How would you iterate on your design if given more time?

## G. Assets and Image Placeholders

When creating the final doc or site, include the following image assets in an 'assets/img' folder and reference them in documentation:  
- wireframe-home.png (low-fidelity sketch)  
- wireframe-lesson-cards.png  
- storyboard-frame-1.png ... storyboard-frame-6.png  
- style-palette.png

Below are textual placeholders indicating where images should be inserted in final deliverables.

[Image Placeholder] wireframe-home.png

[Image Placeholder] storyboard-frame-1.png

[Image Placeholder] style-palette.png

## H. Appendix — Sample One-Hour Lesson Plan (Design to Code)

0-10 min: Inspiration & Goals (show 3 live examples, discuss what makes them good)  
10-20 min: Rapid wireframe (teams sketch mobile/desktop)  
20-35 min: Figma to HTML planning (select grid, typography, image sizes)  
35-50 min: Implementation sprint (build header + hero + one card)  
50-60 min: Share, Feedback, Reflection (use rubric quick score)

Materials: starter repo, code playground, Figma template, accessibility checklist, rubric.