DESIGN:

URL: https://mehrbawan.github.io/swe632_hw3/

Scenarios

Scenario 1:
A frontend developer needs a responsive design for a login page, including username and password entry, a “remember” and “forget password” option, and more at the start of their design process. Instead of writing the HTML, CSS, and JavaScript from scratch, they describe the requirements to the UI Code Assistant on the Generate page, which generates the code instantly.

Scenario 2:
A UX designer in need of design review uploads a PNG mockup of their site’s completed landing page to the UI Code Assistant on the Analyze page. The tool analyzes the layout, and suggests improvements (e.g., "Increase font size for better readability") to successfully better the design in terms of accessibility and usability.

Scenario 3:
A frontend developer needing to verify their page’s layout in the middle of the programming process pastes their HTML/CSS code for a checkout screen into the UI Code Assistant. The tool flags inefficiencies (e.g., "Use CSS Grid to optimize layout") and suggests optimizations.

Argument

UI development is time-consuming and error-prone, especially for developers without lots of frontend experience. Existing tools like Bootstrap provide building blocks and can aid in speeding up the process but lack personalized feedback for a given developer’s goals. The process can be further slowed down due to lengthy code and design reviews. An AI-driven tool like UI AI can optimize the frontend development process by speeding up small, repetitive tasks through code generation as well as instant code and design feedback on demand.

Prototype Plan

The prototype is implemented as a web app known as AI UI with the goals outlined above. The design has three main components to provide well-rounded frontend support:

- Code Generator: Text area for describing components + dropdown for framework (HTML/React/Vue). Outputs code snippets (as in generator.html).

- Design Analyzer: Image file upload for designs/URL input for live sites. Returns AI feedback (analyze.html).

- Code Review: Paste actual code to receive suggestions (review.html).

Socratic Questioning

Clarification Questions

- *What are the main functions of the UI Code Assistant?*
  → It generates components, analyzes existing code, and evaluates UI designs for improvement.
- *Is this tool intended for beginner or expert developers?*
  → Both. Beginners can benefit from the code generation feature for tasks they lack skills in and can learn best practices, while experts save time on repetitive tasks.

Assumption Questions

- *What assumptions am I making about the problems developers face?*
  → That developers spend too much time on boilerplate code and manually reviewing UI code, and perhaps that my own personal experiences in frontend dev are universal.
- *Am I assuming that automated suggestions are always beneficial?*
  → Yes, but to account for this the app should work in a way that users can easily accept or reject suggestions such as through redirection or regenerating reports.

Evidence Questions

- *How do I know developers need this tool?*
  → From complaints found on code forums like Stack Overflow and Reddit, as well as in-person testimony from students learning frontend development in colleges.
- *What existing tools support this idea?*
  → Tools like GitHub Copilot and ChatGPT being used for code-related issues suggest there's a growing market for AI-powered code assistance.

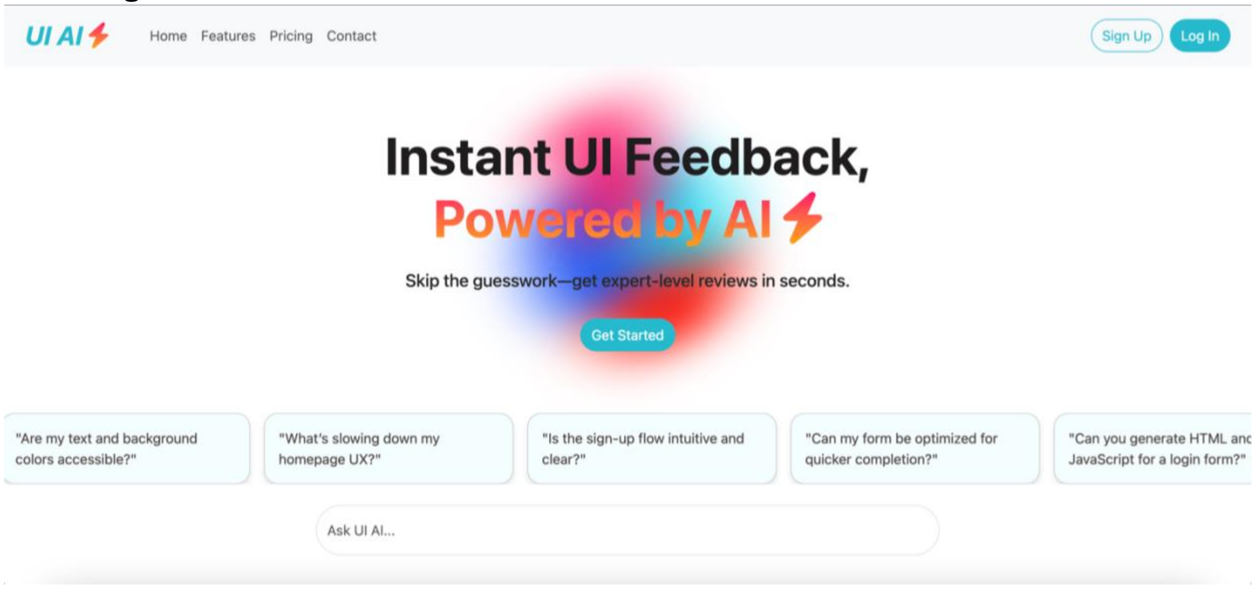
Implication and Consequence Questions

- *What happens if this tool fails to give accurate suggestions?*
  → Users may leave unsatisfied and lose trust in the tool which may lead to a drop in users.
- *What are the consequences of relying on the assistant too much?*
  → Developers may stop learning the fundamentals of frontend dev and become over-reliant on tools. To counter this, we can add educational tooltips and links to documentation or in-house tutorials.

IMPLEMENTATION:

<u>Walkthrough</u>

Front Page

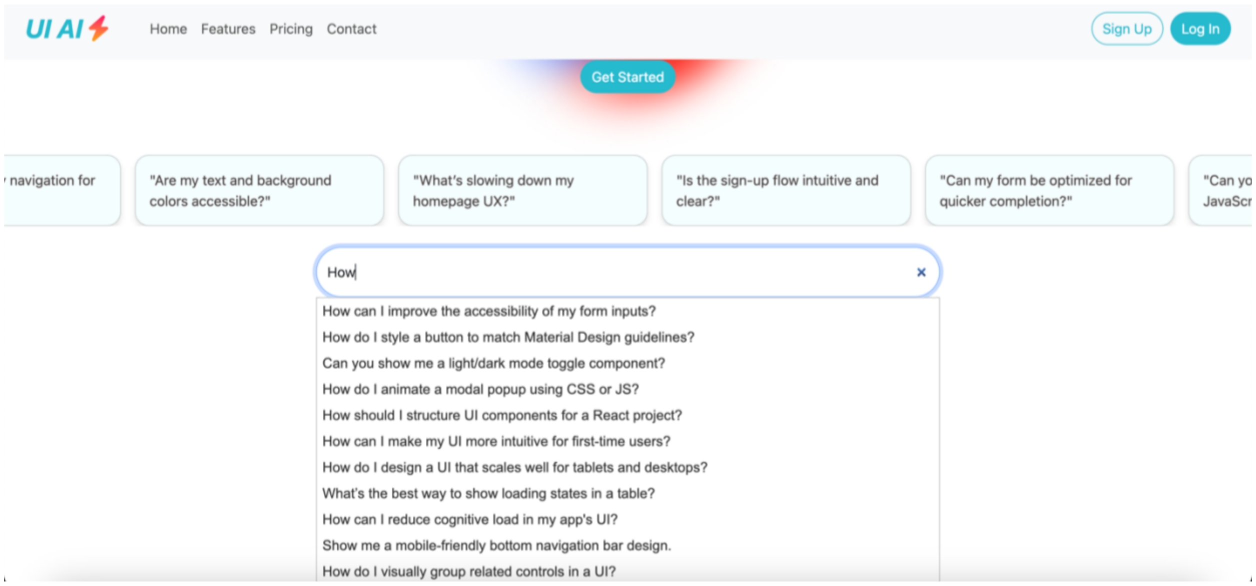


Instant UI Feedback,
Powered by AI

Skip the guesswork—get expert-level reviews in seconds.

"Are my text and background colors accessible?"
"What's slowing down my homepage UX?"
"Is the sign-up flow intuitive and clear?"
"Can my form be optimized for quicker completion?"
"Can you generate HTML and JavaScript for a login form?"

Regular front page layout. Theme is clean and modern inspired by other AI tools like ChatGPT, DeepSeek, and UXPilot.

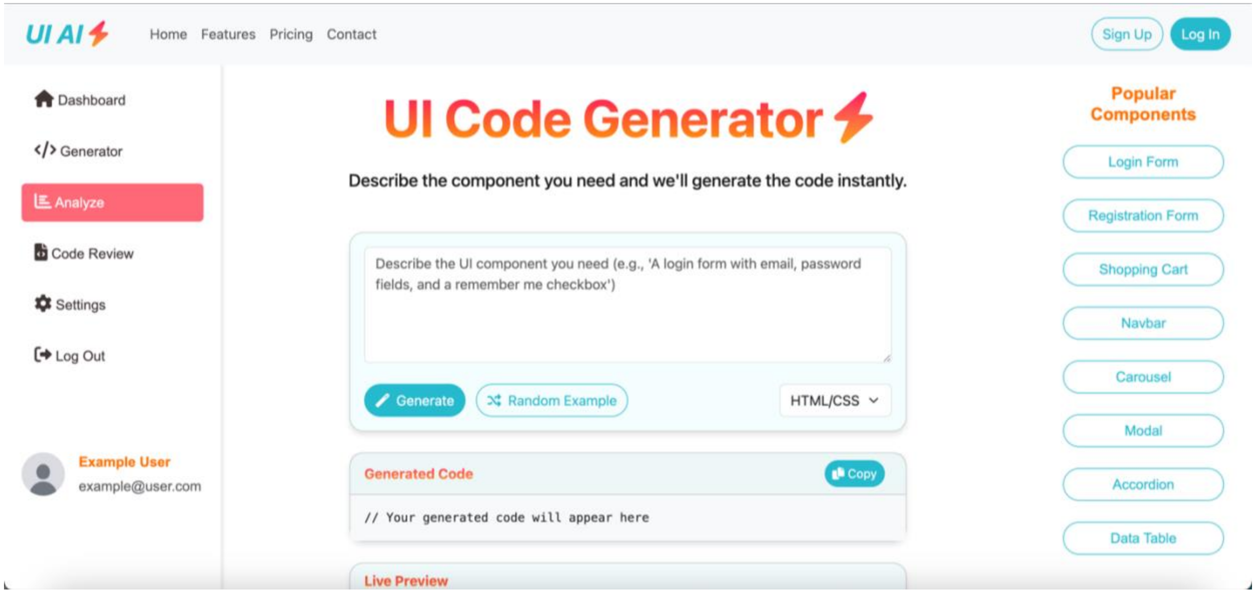


How

How can I improve the accessibility of my form inputs?
How do I style a button to match Material Design guidelines?
Can you show me a light/dark mode toggle component?
How do I animate a modal popup using CSS or JS?
How should I structure UI components for a React project?
How can I make my UI more intuitive for first-time users?
How do I design a UI that scales well for tablets and desktops?
What's the best way to show loading states in a table?
How can I reduce cognitive load in my app's UI?
Show me a mobile-friendly bottom navigation bar design.
How do I visually group related controls in a UI?

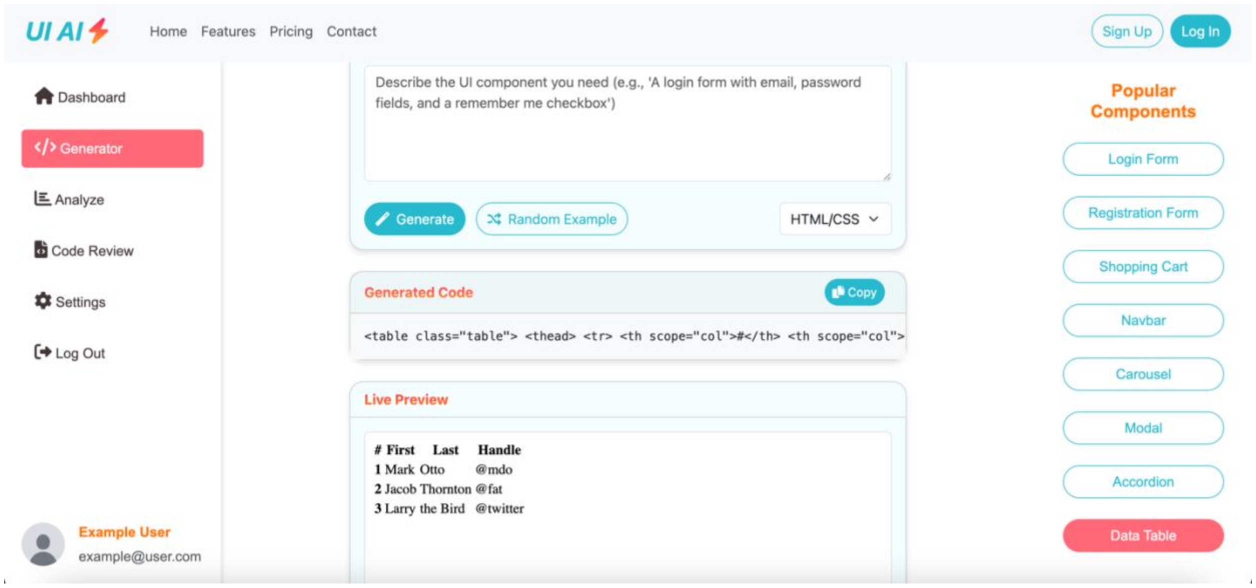
Quick UI search bar on front page has jQuery autocomplete for popular queries users may be asking.

UI Code Generator



```
// Your generated code will appear here
```

"Generate HTML/JavaScript code for standard components such as login, shopping cart, etc." Center panel for custom code generation, side buttons add code for popular components into the code box.

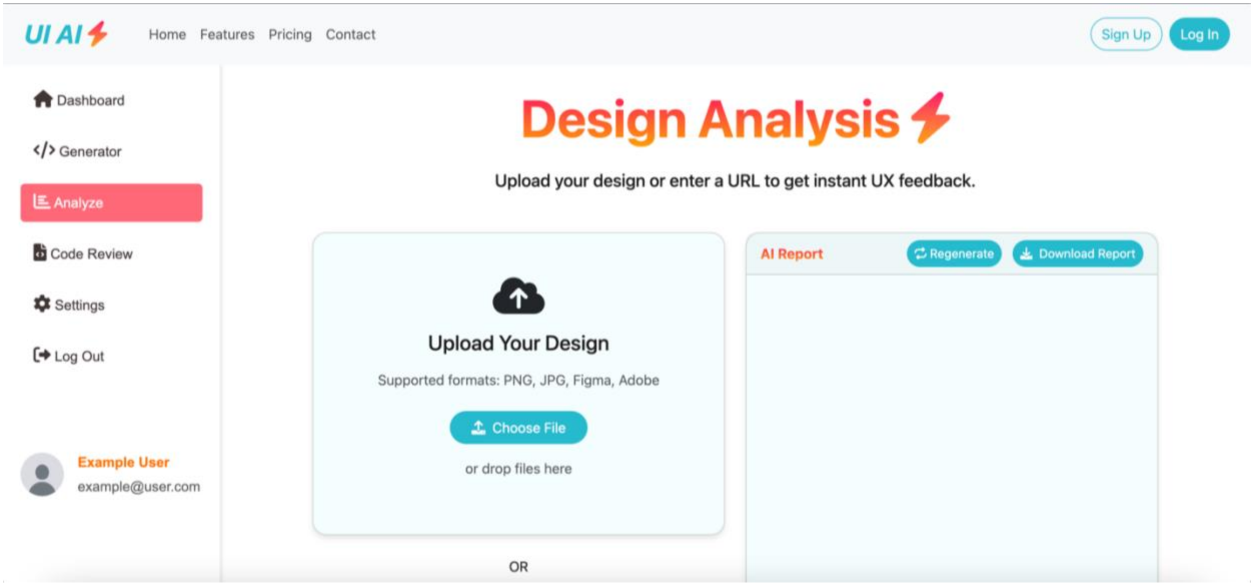


```
<table class="table"> <thead> <tr> <th scope="col">#</th> <th scope="col">
```

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

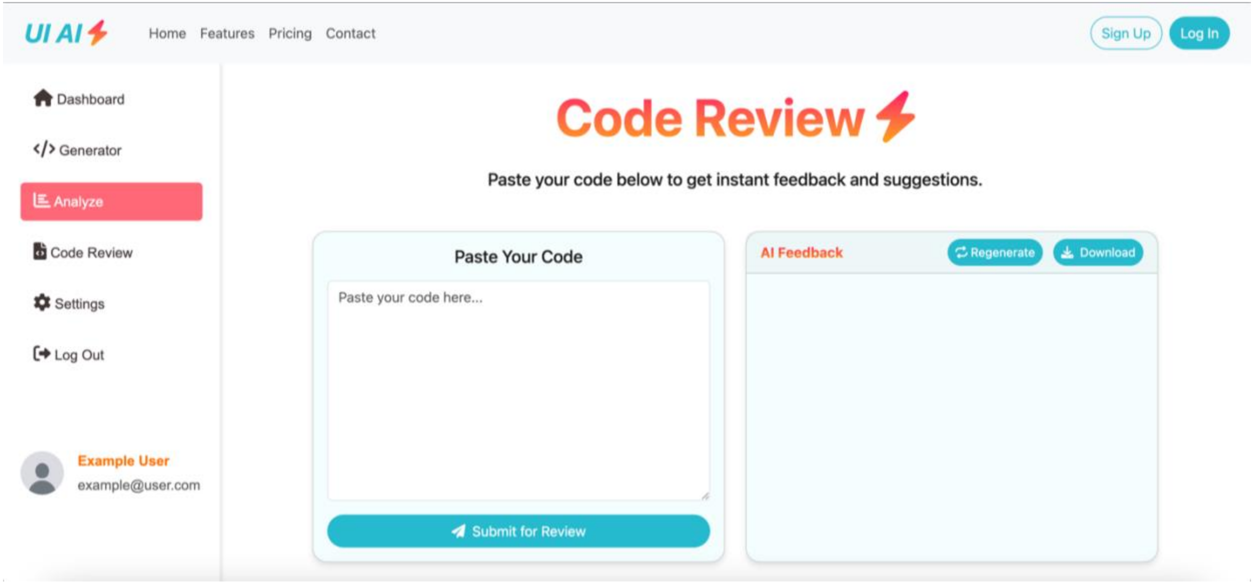
Ex. pressing "data table" will provide premade code and a preview (preview does not work for all items as they're bootstrap which I couldn't get to work in the preview lol oops)

Design Analysis



"Analyze your UI design and provide suggestions." Uploading picture or linking website and pressing generate custom AI report on the right (right now dummy AI text).

Code Review



"Analyze your UI code and determine any inefficiencies, improvements." Here instead of UI layout images, paste code to receive tips and reviews. Like above, not actually AI but should generate dummy AI review.

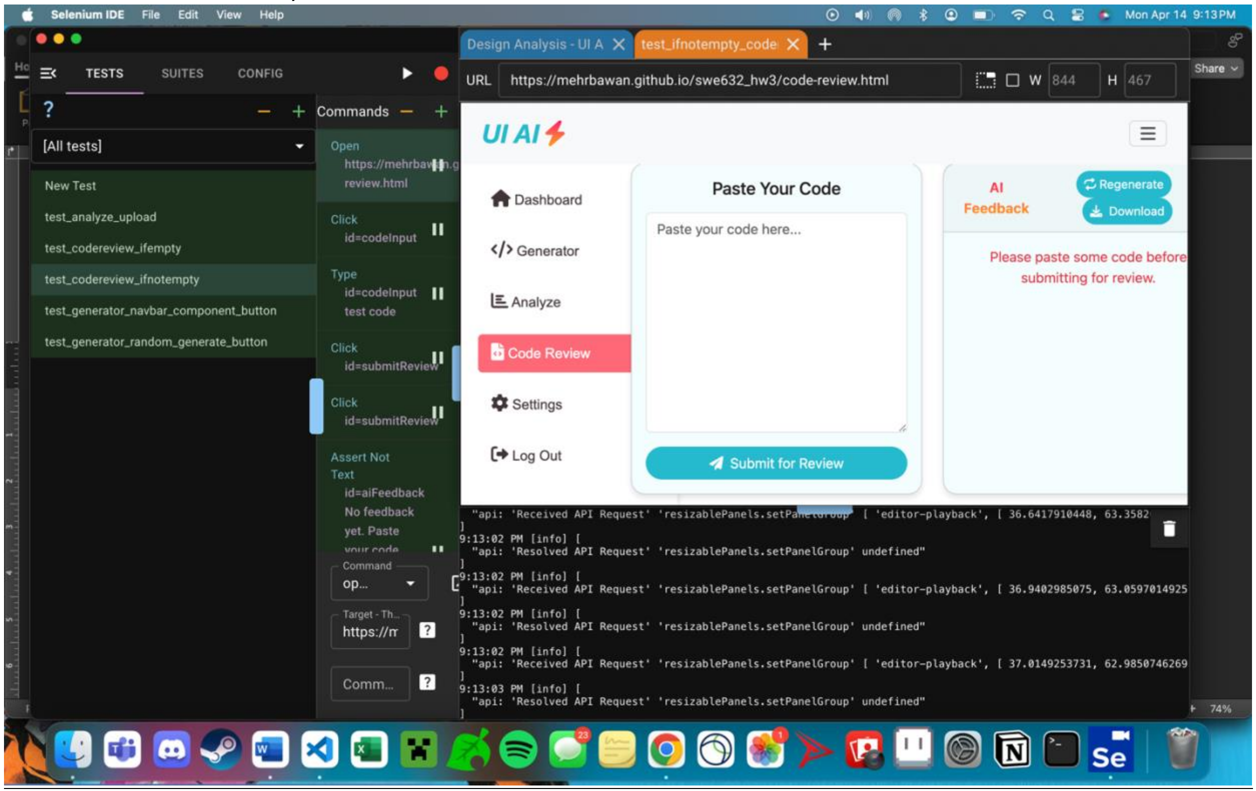
Used HTML, Bootstrap (buttons and CSS), Javascript (functionalities), and jQuery (autocomplete on front page search bar, etc.).

Testing

Selenium

Testing for each page (attached in UIAI tests folder):

- Code review page gives warning if submitted with empty query
- Code review page provides AI feedback if submitted WITH query
- Code design analyze page upload button works and gives feedback
- Code generator random example provides example
- Code generator right side premade component button works (tested with navbar)



[All tests]

New Test
test_analyze_upload
test_codereview_ifempty
test_codereview_ifnotempty
test_generator_navbar_component_button
test_generator_random_generate_button

Open https://mehrbawan.g… review.html
Click id=codeInput
Type id=codeInput test code
Click id=submitReview
Click id=submitReview
Assert Not Text id=aiFeedback No feedback yet. Paste your code

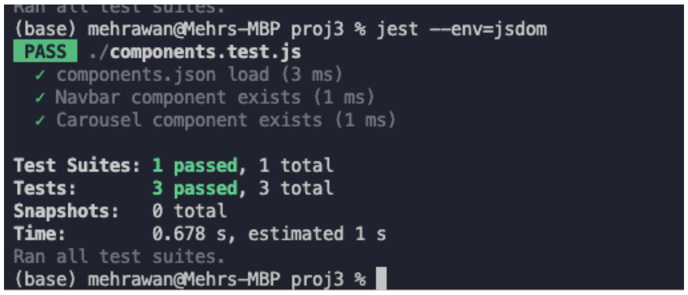
URL https://mehrbawan.github.io/swe632_hw3/code-review.html

Please paste some code before submitting for review.

```
  "api: 'Received API Request' 'resizablePanels.setPanelGroup' [ 'editor-playback', [ 36.6417910448, 63.3582
]
9:13:02 PM [info] [
  "api: 'Resolved API Request' 'resizablePanels.setPanelGroup' undefined"
]
9:13:02 PM [info] [
  "api: 'Received API Request' 'resizablePanels.setPanelGroup' [ 'editor-playback', [ 36.9402985075, 63.0597014925
]
9:13:02 PM [info] [
  "api: 'Resolved API Request' 'resizablePanels.setPanelGroup' undefined"
]
9:13:02 PM [info] [
  "api: 'Received API Request' 'resizablePanels.setPanelGroup' [ 'editor-playback', [ 37.0149253731, 62.9850746269
]
9:13:03 PM [info] [
  "api: 'Resolved API Request' 'resizablePanels.setPanelGroup' undefined"
]
```

Jest

Simple tests on json file data loading properly with Jest (attached in main folder)



```
(base) mehrawan@Mehrs-MBP proj3 % jest --env=jsdom
 PASS  ./components.test.js
  ✓ components.json load (3 ms)
  ✓ Navbar component exists (1 ms)
  ✓ Carousel component exists (1 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        0.678 s, estimated 1 s
Ran all test suites.
(base) mehrawan@Mehrs-MBP proj3 %
```

INTERFACE METRICS:

Learnability

Metric	Examples in Project
Gulfs of Execution	- Explicit sidebar on side always to show how to navigate. - Gulfs bridged by clear instructions, intuitive and familiar pages and setup. All buttons needed are on screen.
Affordance	- Buttons like "Generate Study Plan" and "Generate Weekly Schedule" are designed to look clickable/show selectability on hover. - Input fields are clear and method is evident: o Self-entry text fields are white, while boxes where the TOOL will put text are tinted to indicate no clickability
Signifier	- Appropriate icons for each sidebar heading gives idea of each page (ex. code symbol for the code generator, code doc symbol for reviewer, stat symbol for analysis, etc.) - Page you are on is highlighted in sidebar. - Popular iconography for AI is used to show that it is an AI-driven tool (magic wand, sparkles, and lightning bolts are all rapidly becoming signifiers for AI).
Gulfs of Evaluation	- Quick and concise error messages upon erroneous behavior (ex. when pressing “generate” without writing any text in any of the AI query boxes)

Interface Architecture

Component States

- Active/inactive navigation items (highlighted when selected)
- Form input states (empty, filled, submitted)
- Code generation state (before/after generation)
- Analysis report state (empty/loaded)

Events

- Click events for navigation buttons and form submissions
- Input events for search bars and text areas
- Load events when fetching component data from JSON files
- Hover events for interactive elements like buttons, cards (color change/glow)

Model-View-Controller

Model

- Data in JSON files (queries.json, components.json)
- Definitions in HTML code

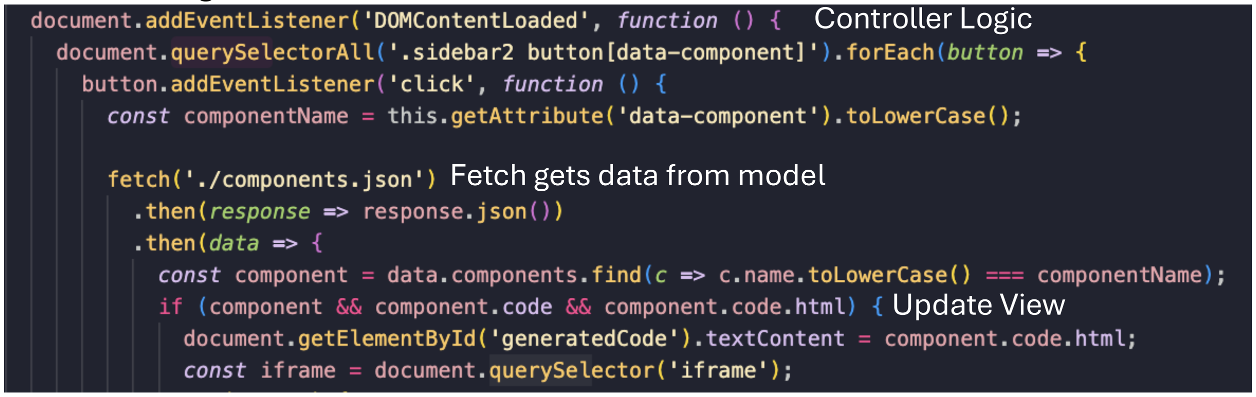
View

- HTML Page
- Components on page that read user actions

Controller

- JavaScript event handlers (script.js and scripts built into html files)
- AJAX calls to load JSON data
- Logic to update views based on user actions

Ex. code from generator.html



```
document.addEventListener('DOMContentLoaded', function () {
  document.querySelectorAll('.sidebar2 button[data-component]').forEach(button => {
    button.addEventListener('click', function () {
      const componentName = this.getAttribute('data-component').toLowerCase();

      fetch('./components.json')
        .then(response => response.json())
        .then(data => {
          const component = data.components.find(c => c.name.toLowerCase() === componentName);
          if (component && component.code && component.code.html) {
            document.getElementById('generatedCode').textContent = component.code.html;
            const iframe = document.querySelector('iframe');
```

Hierarchy

Overall Website Hierarchy

- Index Page
- Dashboard
- Generator
- Analyzer
- Code Reviewer

General Page Hierarchy

- Navbar
  - Home
  - Features
  - Pricing
  - Contact
  - Signup/Login
- Sidebar
  - Dashboard

    - Generator
    - Analyze
    - Code Review
    - Settings
    - Logout
    - Profile
- Main Content
    - Varies

Generator View Hierarchy

- Navbar (Above)
- Sidebar (Above)
- Main Content
    - Header
    - AI Query Container
        - Text Field
        - Generate Button
        - Random Button
        - Language Choice
    - Generated Code Container
        - Text Field
    - Live Preview Container
        - Text Field
- Component Sidebar
    - Login Form, Registration Form, etc. components

Analyze View Hierarchy

- Navbar (Above)
- Sidebar (Above)
- Main Content
    - Header
    - Upload Container
        - Choose File Button
        - Text Instructions
    - Live Web Upload
        - Website Link
        - Analyze Button
    - AI Report Container
        - Text Field
        - Regenerate Button
        - Download Button

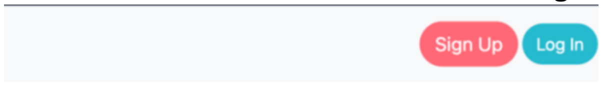
Code Review View Hierarchy

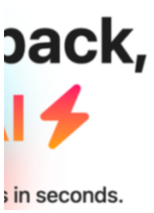
- Navbar (Above)
- Sidebar (Above)
- Main Content
  - Header
  - Code Container
    - Paste Text
    - Text Field
    - Submit Button
  - AI Feedback Container
    - Text Field
    - Regenerate Button
    - Download Button

Pointing

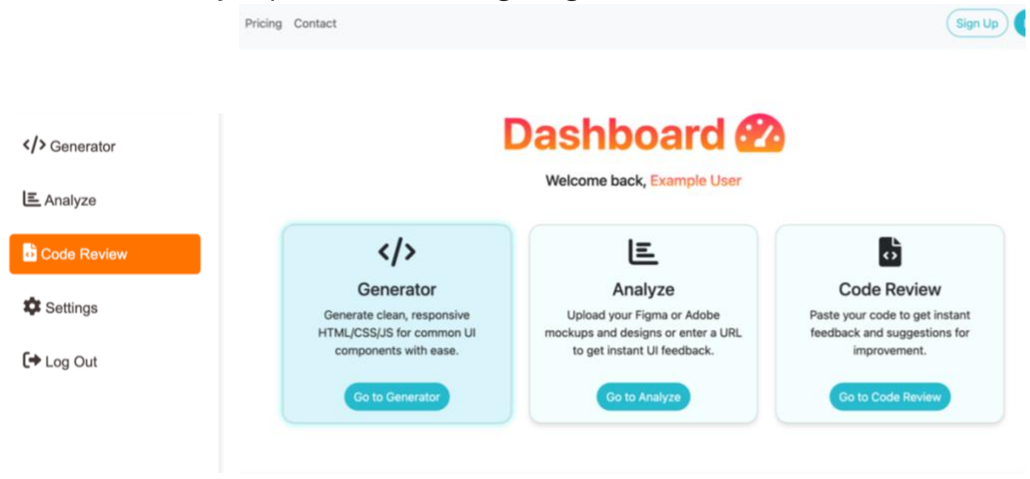
Pointing Features

- Button zoom on small buttons for easier clicking





  - (mouse pointer not visible in my computer's screenshots, but pointer *is* on signup button)
- Buttons are spaced Far apart enough to avoid misclicks, and button borders are highlighted if not already explicit to avoid targeting errors.


Pricing
Contact
Sign Up
Generator
Analyze
Code Review
Settings
Log Out
Dashboard
Welcome back, Example User
Generator
Generate clean, responsive HTML/CSS/JS for common UI components with ease.
Go to Generator
Analyze
Upload your Figma or Adobe mockups and designs or enter a URL to get instant UI feedback.
Go to Analyze
Code Review
Paste your code to get instant feedback and suggestions for improvement.
Go to Code Review

  - (again mouse pointer not visible; pointer hovering on generator above)