**As a tester, I am seeking to automate my company’s website to conduct both UI and API testing. To achieve this, I intend to create playwright test files based on user prompts generated using a large language model (LLM). This LLM can be hosted on a cloud platform such as Claude, OpenAI, or locally.**

**The prompts will be stored in a designated folder, and the AI will process each prompt to generate corresponding playwright files that can be executed using the playwright framework. Test execution reports will be generated using the allure framework.**

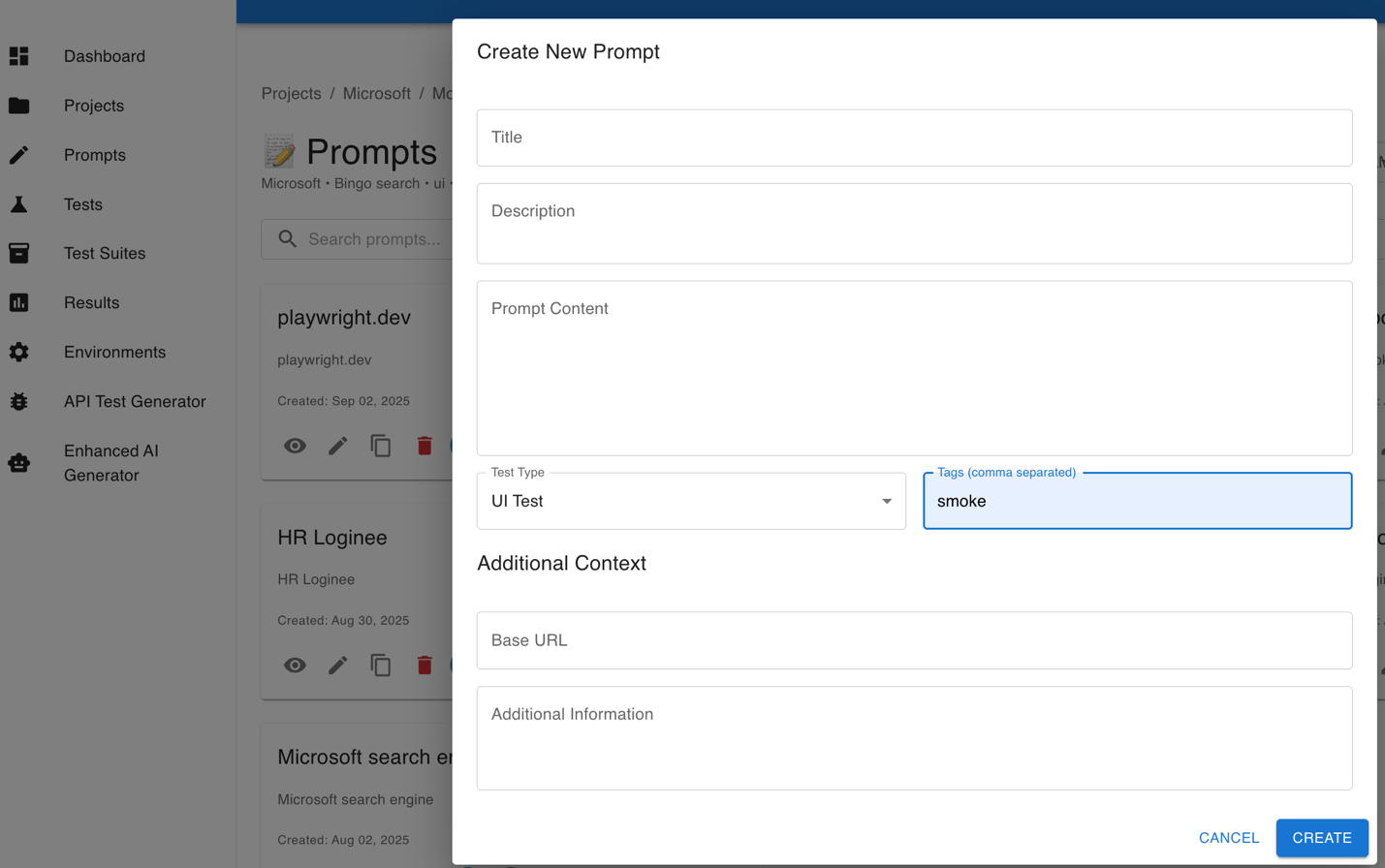
**The technology stack for this project will primarily consist of React and Node.js, with optional integration of LLM capabilities. If necessary, the framework UI should include a file explorer, similar to a tree structure, to facilitate prompt input.**

**Below is a proposed design for the framework flow:**

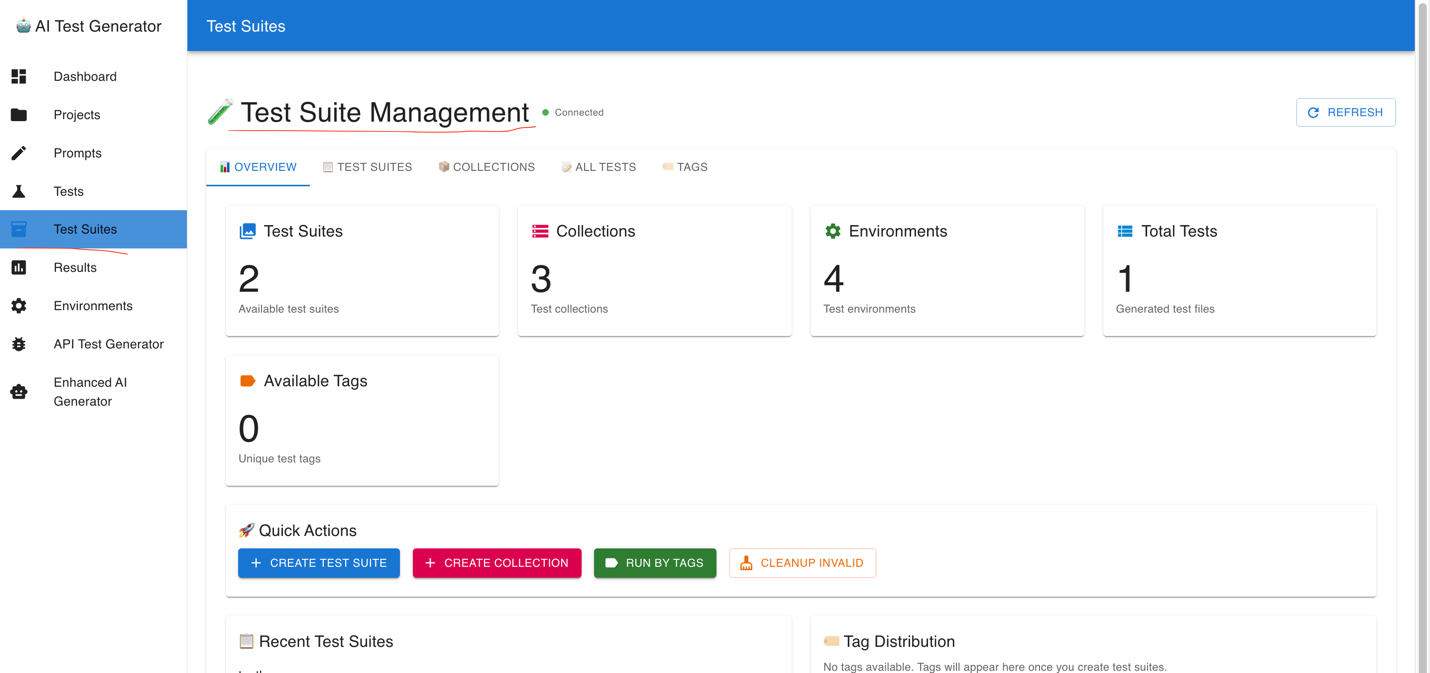
**2. Core Features**

1. **Web UI for Prompt Input**
   * **-** Text area for entering test prompts (e.g., *“Navigate to login, enter username/password, click login, validate dashboard”*).
   * - “Generate Test” button.
2. **Prompt Parser**
   * **-** Parse natural-language prompts into **structured steps/intents** (navigate, click, fill, assert).
   * - Detect UI flows (login, navigation, form filling) and API calls (GET/POST/etc.).
3. **Template-Based Code Generator**
   * **-** Deterministic templates for each step.
   * - Assemble steps into a comprehensive Playwright test spec.
   * - Add default waits, retries, and assertions.
4. **Generated Playwright Spec Output**
   * - Display generated .spec.ts (or .spec.js) in the web UI.
   * - Enable **download** or **copy-to-clipboard functionality**.
5. **LLM Response as Comments (Optional)**
   * **-** If integrated with an LLM, retain its raw response.
   * **6 Primary Spec**
   * - Always template-driven.
6. **Metadata & Reporting**
   1. **-** Store metadata per generated test (prompt text, parsed steps, file path, timestamp).

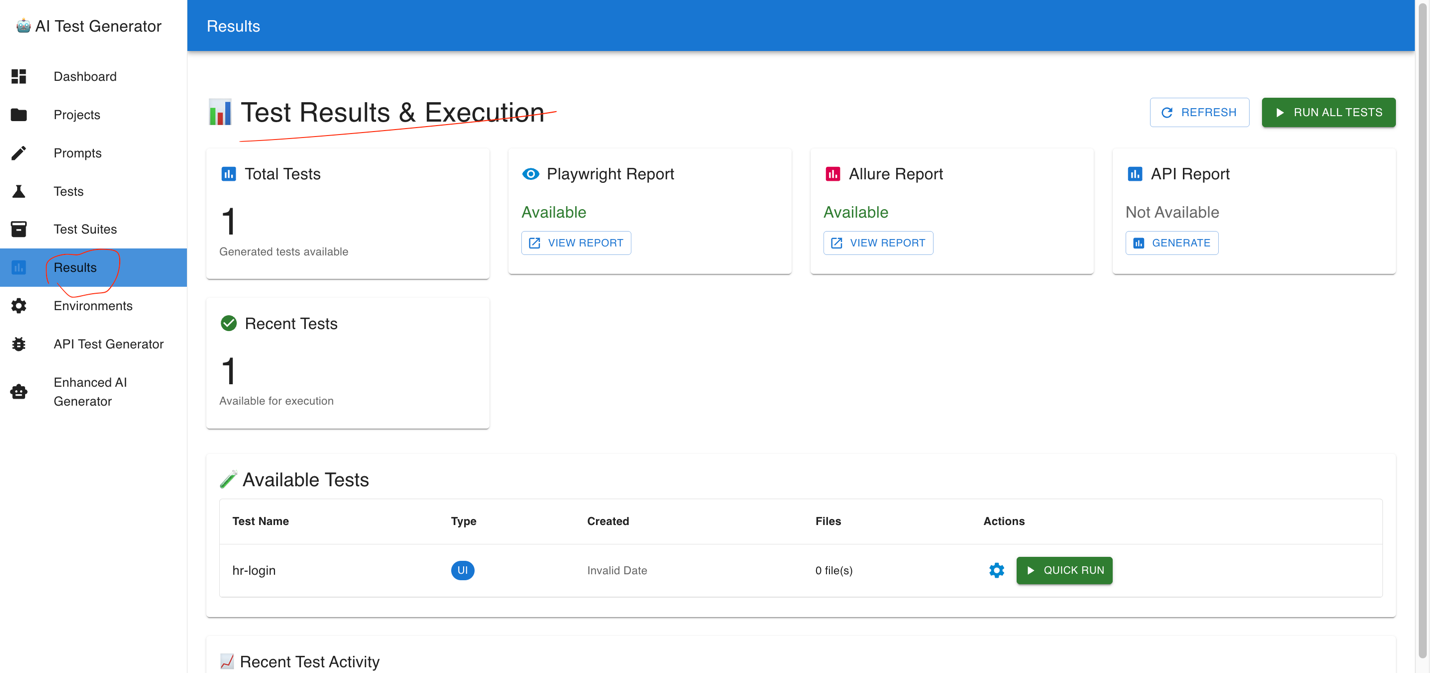
Prompt UI:



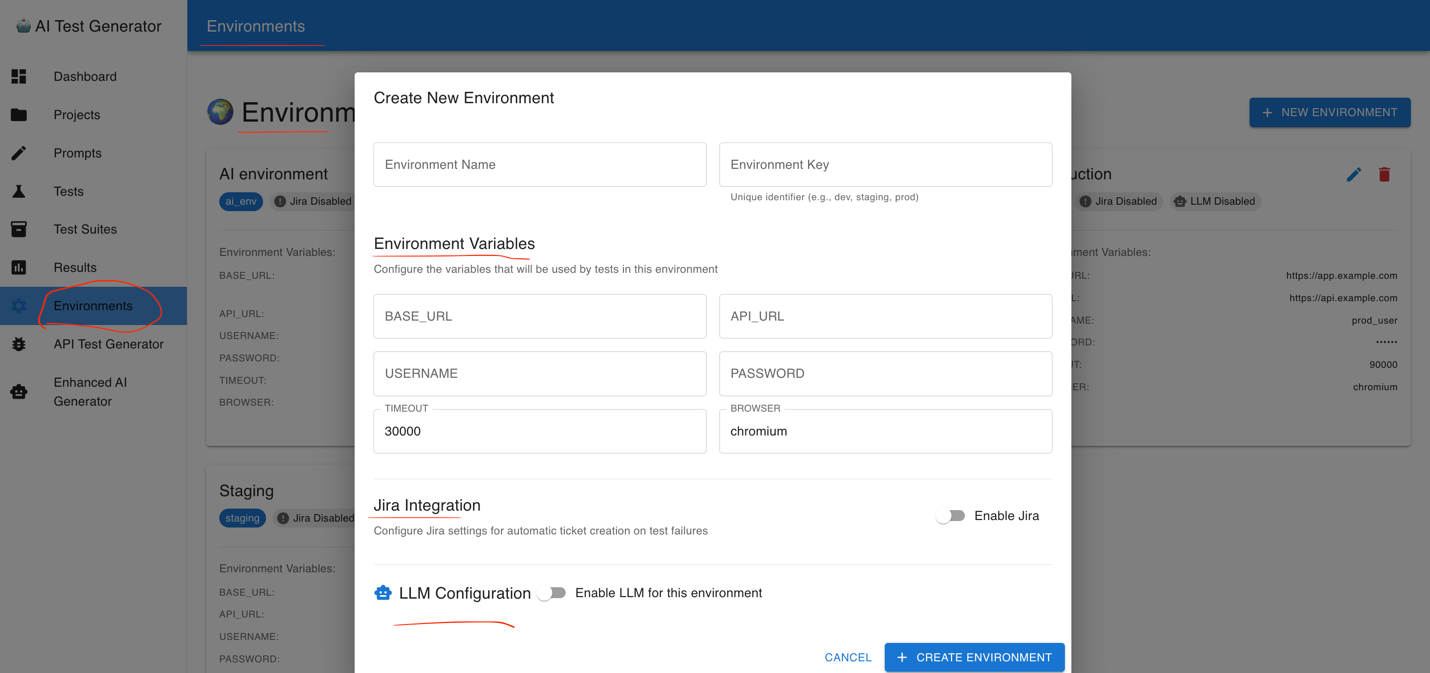
Test Suite: User Interface



Results Tab: Allure/HTML reporting is enabled in the generated specifications.



Environment Tab



API Test Generator

