**AI Marketplace: Web Development Sprint Plan (Initial Phase)**

**Version:** 1.0 **Date:** April 23, 2025 **Project Goal:** Develop the core front-end pages for the AI Marketplace, focusing on a minimalist, responsive design, and integrating essential user interactions (search, favorites, ratings viewing). This plan covers the initial build-out, including testing phases.

**Assumptions:**

1. Basic project structure (HTML/CSS/JS framework/boilerplate) is set up.
2. Version control (Git) is initialized and follows the branching strategy outlined in the SOPs.
3. Team members (initially 2) are familiar with the chosen tech stack and the Project Guidelines/SOPs.
4. Database connection (if needed for backend interactions planned within these sprints) is functional or mock APIs are available.
5. Task management tool (e.g., Trello, GitHub Projects) is in use.

**Sprint Length:** 2 Weeks

**Definition of Done (DoD):**

* **For a Task/User Story:**
  + Code implemented according to acceptance criteria.
  + Code adheres to Project Coding Standards (linting, formatting).
  + Relevant Unit Tests written and passing.
  + Integration Tests written and passing (if applicable).
  + Code reviewed by at least one other team member and approved.
  + Functionality manually tested on target browsers (Chrome, Firefox, Safari) and key screen sizes (mobile, tablet, desktop).
  + Code merged into the develop branch.
  + Task updated in the tracking tool.
* **For a Sprint:**
  + All committed stories/tasks meet their DoD.
  + Sprint goals achieved.
  + Working software demonstrated in Sprint Review.
  + Sprint Retrospective conducted.

**Sprint Breakdown & Timeline Estimate:**

**(Start Date: Approx. April 28, 2025 - Adjust as needed)**

**Sprint 1: Foundation & Homepage (Weeks 1-2)**

* **Goal:** Establish core site structure, responsive layout foundations, reusable navigation/footer components, and build the static Homepage.
* **Key Tasks:**
  + Set up basic routing.
  + Create reusable Header component (with logo, nav links placeholders, search icon placeholder).
  + Create reusable Footer component.
  + Implement foundational responsive CSS grid/layout system.
  + Build Homepage structure (HTML/CSS) - sections for featured tools, categories, intro text.
  + Apply initial minimalist styling (colors, typography).
  + Set up Unit Testing framework (e.g., Jest/Vitest) and write initial tests for Header/Footer.
* **Testing Focus:**
  + **Unit Testing:** For Header, Footer, any utility functions created.
  + **Manual Testing:** Basic responsiveness checks (dev tools, different browser widths), cross-browser checks (initial).
  + **Code Reviews:** Enforce standards from the start.

**Sprint 2: Search/Discovery Page & Tool Listing (Weeks 3-4)**

* **Goal:** Implement the Search/Discovery page layout, basic search input functionality (frontend), category filters UI, and display a list of tools using a reusable card component (initially with static/mock data).
* **Key Tasks:**
  + Build Search/Discovery page layout (HTML/CSS).
  + Implement Search Bar component (UI + basic state handling).
  + Implement Filter components (e.g., checkboxes, dropdowns for categories - UI only).
  + Create reusable Tool Card component (displaying name, category, short description, rating placeholder, favorite placeholder).
  + Populate Search/Discovery page with Tool Cards using mock data.
  + Refine responsive design for this page.
* **Testing Focus:**
  + **Unit Testing:** For Search Bar, Filter components, Tool Card.
  + **Manual Testing:** Functionality of search input UI, filter UI interactions, Tool Card display, responsiveness. Cross-browser checks.
  + **Code Reviews.**

**Sprint 3: Individual Tool Page & Ratings/Favorites UI (Weeks 5-6)**

* **Goal:** Develop the detailed Individual Tool page layout, display comprehensive tool information (using mock data), and implement the UI elements for viewing ratings/reviews and toggling favorites (state managed locally initially).
* **Key Tasks:**
  + Create routing for individual tool pages (e.g., /tool/:id).
  + Build Tool Page template (HTML/CSS) - sections for description, features, images/video, rating summary, reviews list.
  + Implement Rating display component (e.g., showing stars).
  + Implement Review display component (showing user review text).
  + Implement Favorite button component (UI toggle state).
  + Populate page with mock data based on a hypothetical tool ID.
  + Ensure responsiveness of the Tool Page.
* **Testing Focus:**
  + **Unit Testing:** For Rating display, Review display, Favorite button components.
  + **Manual Testing:** Layout and content display on Tool Page, Favorite button toggling, responsiveness. Cross-browser checks.
  + **Code Reviews.**

**Sprint 4: User Profile/Dashboard & Basic Functionality (Weeks 7-8)**

* **Goal:** Create the User Profile/Dashboard page structure, display a list of the user's favorited tools (mocked/local state), and include placeholders for user settings and analytics. Implement basic user auth UI flow placeholders if needed now.
* **Key Tasks:**
  + Build Dashboard page layout (HTML/CSS) - sections for favorites, settings, analytics.
  + Implement component to display a list of favorited tools (using Tool Card component, based on local state/mock data).
  + Create placeholder components/sections for User Settings and Analytics.
  + Implement basic Login/Signup/Logout page UI placeholders and routing (if auth is tackled now, otherwise defer).
  + Ensure responsiveness of the Dashboard.
* **Testing Focus:**
  + **Unit Testing:** For Dashboard components (e.g., Favorite List).
  + **Manual Testing:** Layout and display on Dashboard, interaction with favorite list placeholders, responsiveness. Cross-browser checks.
  + **Code Reviews.**

**Sprint 5: Backend Integration (API Connectivity) & UAT Prep (Weeks 9-10)**

* **Goal:** Connect frontend features (Search, Tool data, Favorites, Ratings) to backend APIs. Implement data fetching, error handling, and state management related to API interactions. Prepare test cases for UAT. Deploy to a Staging environment.
* **Key Tasks:**
  + Integrate API calls for fetching tools (Search/Discovery).
  + Integrate API calls for fetching individual tool details (Tool Page).
  + Integrate API calls for fetching/saving user favorites (Tool Card, Dashboard).
  + Integrate API calls for fetching ratings/reviews (Tool Page).
  + Implement loading states and error handling for API calls.
  + Refactor state management as needed (e.g., using Context API, Redux, Zustand if applicable).
  + Write **Integration Tests** for API service modules/hooks.
  + Write basic **E2E Tests** for critical flows (e.g., Search -> View Tool -> Favorite).
  + Prepare detailed UAT scenarios/test cases document.
  + Deploy the current develop branch state to a Staging environment.
* **Testing Focus:**
  + **Unit Testing:** Continue as needed.
  + **Integration Testing:** Verify frontend-backend communication for core features.
  + **E2E Testing:** Automate key user paths.
  + **Manual Testing:** Verify features work with live API data. Test loading/error states.
  + **Code Reviews.**

**Sprint 6: User Acceptance Testing (UAT) & Bug Fixing (Weeks 11-12)**

* **Goal:** Conduct UAT with stakeholders (initially you and your friend, potentially others later) using the Staging environment. Gather feedback and fix identified bugs.
* **Key Tasks:**
  + Facilitate UAT sessions based on prepared test cases.
  + Collect and document all feedback and bugs found during UAT (use task tracker).
  + Prioritize UAT bugs/feedback.
  + Fix validated bugs, following the standard Git workflow (fix branches, PRs).
  + Perform regression testing (manual and automated) to ensure fixes don't break existing functionality.
  + Re-deploy fixes to Staging for verification if needed.
* **Testing Focus:**
  + **UAT Execution:** Formal testing by non-developers against requirements.
  + **Regression Testing:** Ensure stability after fixes.
  + **Manual Testing:** Exploratory testing on Staging.

**Sprint 7: Final Polish, Optimization & Release Preparation (Weeks 13-14)**

* **Goal:** Address any remaining high-priority bugs, perform final UI/UX polishing based on feedback, conduct performance checks, update documentation, and prepare for a potential production release.
* **Key Tasks:**
  + Fix any remaining critical/high-priority bugs from UAT or final reviews.
  + Perform cross-browser and cross-device polishing.
  + Conduct accessibility checks (WCAG compliance).
  + Run performance analysis (e.g., Lighthouse) and implement optimizations (image compression, code splitting, etc.).
  + Final code cleanup and refactoring.
  + Update project README and any other necessary documentation.
  + Prepare deployment scripts/checklist for Production.
  + Final merge from develop to main. Tag the release.
* **Testing Focus:**
  + **Final Regression Testing:** Comprehensive check before release.
  + **Performance Testing.**
  + **Accessibility Testing.**
  + **Manual Testing:** Final review of all key flows and pages.