**Milestone 3**

Stage 1: Initial Prototype

Prompt

”I am a software engineering student working on a project to create a basic home service

website. The website should include the following pages: Home, Login, Signup, Service Requests, and an Admin Dashboard. It should have simple navigation elements, such as a top navigation bar and footer, and placeholders for service categories and descriptions. The layout should include clear call-to-action buttons like ’Request Service,’ ’Login,’ and ’Sign Up.’ Focus on a clean and user-friendly design. My requirements are attached in the provided file, and all of them need to be satisfied. Start by outlining the necessary files and their names. The project must be developed using Visual Studio Code, with MySQL as the database.”

Techniques Used

1. Clear and Actionable Instruction: The prompt explicitly states the task (”cre- ate a basic website”) and lists key elements, such as required pages and navigation structure.

2. Task Breakdown: Each core feature is detailed (e.g., Home, Login, Signup, Admin Dashboard) to provide clear implementation guidelines.

3. User-Centric Focus: The design emphasizes usability through clear call-to-action buttons and a user-friendly layout.

4. Practical Constraints: The prompt specifies the development environment (Vi- sual Studio Code) and database choice (MySQL), aligning with the project require- ments.

5. Testable Outcome: Deliverables like navigation, placeholders, and buttons are easily testable for functionality and completeness.

Stage 2: Adding Basic Interactivity

Prompt

”Introduce interactive features like a searchable list of service providers. Implement filters for location, service type, and availability. Ensure that the results dynamically update based on user input without reloading the page. Design a responsive interface to ensure usability on mobile and desktop.”

Techniques Used

1. Incremental Complexity: Adds interactivity by introducing dynamic updates

and search functionality.

1. Explicit Functional Requirements: Specifies filters and real-time updates to ensure clear functionality.
2. Responsive Design: Stresses design compatibility with multiple screen sizes.
3. UI/UX Elements: Introduces dropdowns and search bars for intuitive use.
4. Testable Outcome: The success can be tested by trying the filters and ensuring responsive behavior.

Stage 3: User Authentication System

Prompt

”Create a user authentication system with login, registration, and role-based access con- trol for clients, workers, and admins. Upon login, users should be directed to role-specific dashboards (e.g., clients to service request history, workers to task lists, and admins to user management). Ensure session management for login persistence.”

Techniques Used

1. Task Decomposition: Breaks down authentication into manageable tasks (login,

registration, role-based access).

1. Role-Specific Instructions: Clearly specifies what each user role should experi- ence post-login.
2. Technical Context: Highlights session management for persistence.
3. Logical Flow: Guides user experience from login to role-specific dashboards.

5. Testable Outcome: Easily testable by simulating login for each role and verifying dashboard access.

Stage 4: Service Request Workflow

Prompt

”Enable clients to create service requests by selecting a service type, specifying location and time, and leaving optional instructions. Allow workers to view and accept/reject service requests through their dashboard. Notify clients about the status of their requests in real-time.”

Techniques Used

1. Feature-Specific Clarity: Details the exact steps for service request creation and

management.

1. Role-Specific Design: Outlines separate workflows for clients and workers.
2. Real-Time Interaction: Emphasizes real-time notifications for immediate up- dates.
3. Actionable Design: Clear user actions like ’accept’ or ’reject’ for workers.
4. Testable Outcome: Verify the workflow by creating requests and monitoring

status changes.

Stage 5: Admin Panel

Prompt

”Design an admin panel to manage users, monitor service requests, and generate system reports. Include features like user role assignment, approval/rejection of new workers, and viewing platform analytics (e.g., completed tasks, user activity).”

Techniques Used

1. Task Prioritization: Focuses on critical admin functionalities (user management,

analytics).

1. Role-Specific Clarity: Highlights administrative oversight and control.
2. Data Visualization: Encourages use of charts and graphs for analytics.
3. Management Tools: Introduces approval workflows for worker onboarding.
4. Testable Outcome: Testable by verifying user role assignment and analytics.

Stage 6: Payment Integration

Prompt

”Integrate secure payment processing for service transactions. Allow clients to pay through various methods (e.g., credit card, digital wallets). Notify workers of successful payments. Store payment records securely for future reference.”

Techniques Used  
1. Secure Implementation: Highlights the importance of secure payment gateways.

2. User-Centric Design: Focuses on seamless payment options for clients.  
3. Real-Time Updates: Ensures workers are notified of payments instantly.  
4. Data Retention: Includes secure storage of payment history.  
5. Testable Outcome: Validate by testing payment flows with mock transactions.

Stage 7: Feedback and Rating System

Prompt

”Implement a feedback system allowing clients to rate workers and leave comments after task completion. Display aggregated ratings on worker profiles. Notify workers of new feedback.”

Techniques Used

1. User Engagement: Encourages client-worker interaction through ratings.
2. Data Aggregation: Summarizes ratings for quick assessment.
3. Notifications: Keeps workers informed of client opinions.
4. Transparency: Builds trust by showing ratings publicly.
5. Testable Outcome: Test by submitting feedback and checking its reflection on profiles.

Stage 8: Finalizing Design

Prompt

”Polish the website’s design with a cohesive color scheme, clean typography, and user- friendly animations. Add hover effects on buttons, improve spacing, and ensure all ele- ments align properly across pages. Test for accessibility, responsiveness, and cross-browser compatibility.”

Techniques Used

1. Aesthetic Refinement: Focuses on visual consistency and branding.
2. UX Enhancements: Encourages smooth interactions with animations and hover effects.
3. Accessibility: Promotes inclusive design practices.
4. Cross-Device Functionality: Ensures responsiveness on various devices.
5. Testable Outcome: Verify through design audits and cross-browser tests.

The source code for this project is available on GitHub:

https://github.com/alyelsaka/HomeService.git