Phase 7: Complex Computing Problem (CCP) Framework

1. Problem Complexity Analysis

Several design and technical challenges were encountered, such as:

- Ensuring responsive design across different screen sizes.

- Optimizing AI-generated images to maintain quality and background colors.

- Implementing the unique feature

2. Multiple Solution Approaches Considered

During the development of the website, multiple solutions were considered to address the challenges faced:

Responsive Design vs. Static Layouts: We used Flexbox and CSS Grid to accommodate various screen sizes and devices.

Bootstrap vs. Pure CSS: We chose pure CSS for styling over Bootstrap for more control over the design.

AI-Generated Image Handling: Several AI tools (such as ChatGPT/Gemini) were used for generating product images.

Unique Technical Feature Implementation: The decision to implement a basic product recommendation system using pre-defined categories added complexity to ensure that data interactions were seamless and user-friendly.

3. Technical Challenges Encountered

Several technical challenges arose during the development:

Git Version Control Conflicts: Working in pairs, we occasionally faced issues with version control when merging branches, which required careful conflict resolution.

Figma to Code Translation: Translating the pixel-perfect designs from Figma into HTML/CSS presented challenges, especially with the consistency of spacing and alignment.

4. Integration of Multiple Technologies

The integration of multiple technologies (HTML, CSS, JavaScript, Git, and AI-generated images) was one of the major complexities of the project:

AI-Generated Images: We utilized generative AI tools to produce high-quality images that matched the e-commerce theme. These images were integrated into the product display.

HTML/CSS: Using HTML semantic tags and CSS techniques, we ensured that the layout was responsive and consistent with the Figma designs.

JavaScript: The interactive features, such as the login, sign-up, and checkout forms, were implemented using JavaScript. This included form validation and dynamic updates to the cart as users made selections or changed quantities.

Git Version Control: Throughout the project, we maintained version control using Git to track changes and collaborate effectively across multiple pages and features.

5. Performance & Trade-Off Analysis

Balancing between performance and design fidelity was a key challenge:

Image Optimization: AI-generated images were optimized for web use by compressing. While this improved performance, it also required careful consideration of the balance between image quality and load speed.

Responsive Design vs. Development Time: Achieving perfect designs took more time due to the responsive layout requirements.

Feature Richness vs. Code Complexity: Some features were simplified to keep the code manageable. More complex features were initially planned but reduced to a basic implementation for easier implementation