PROJECT OVERVIEW AND TECHNICAL ARCHITECTURE

PROJECT REPORT: E-COMMERCE PLATFORM FOR BOUTIQUE RETAIL

GitHub Repository Link: https://github.com/ssarakaya/boutique-project

1. PROJECT OVERVIEW AND OBJECTIVES

This project represents the successful design and implementation of an end-to-end, boutique-style e-commerce platform developed as part of the Web Design and Programming final evaluation. The system is architected as a full-stack application, leveraging a decoupled structure that separates the user interface from the business logic. The primary objective was to build a highly responsive, data-driven application that fulfills modern retail requirements—such as dynamic product management, secure data handling, and an intuitive user experience—while utilizing the React framework and Spring Boot ecosystem.

2. TECHNICAL ARCHITECTURE AND SYSTEM DESIGN

The technical foundation of the project is built upon three distinct layers to ensure scalability and high performance.

Frontend Implementation: The client-side application is developed using the **React Framework**. To provide a professional aesthetic and consistent user interface, the **Material UI (MUI)** library was extensively integrated. Instead of basic CSS, MUI's advanced component system was used to build the navigation menus, product grids, and interactive buttons. The application utilizes a component-based architecture, which allows for reusable UI elements and cleaner code maintenance.

Backend Implementation: The server-side is powered by **Java Spring Boot**. It functions as a RESTful API provider, handling all data requests from the frontend. The backend manages the business logic, including product categorization and user review processing. Dependency management is handled through **Maven**, ensuring that all necessary libraries are automatically resolved and consistent across different development environments.

Database Infrastructure: A relational database management system is employed to store all persistent data. To ensure easy deployment and evaluation, the entire database schema and sample data have been exported into a single self-contained SQL file, located within the project's database directory.

3. NOVEL FEATURES AND PROJECT DIFFERENTIATION

What makes this project unique and separates it from standard e-commerce templates is the integration of several advanced "novel" features:

- **Infinite Scrolling Testimonial Band:** A standout feature of the home page is the custom-engineered testimonial section. Unlike static reviews, this component uses advanced CSS Keyframe animations to create a continuous, infinite loop of customer feedback. This not only adds a premium "boutique" feel but also demonstrates an advanced understanding of DOM manipulation and CSS transitions.
- **Asynchronous Content Strategy:** The home page is not static; it dynamically pulls the most significant products from the Spring Boot backend service. To enhance user experience, "Skeleton" loading states were implemented to provide visual feedback while data is being fetched, preventing layout shifts and ensuring a smooth transition.
- **Intelligent State Management:** The application efficiently manages the shopping cart and user preferences without the need for page reloads. This is achieved by using React's internal state management tools to synchronize data across multiple components, such as the product detail page and the checkout view.

4. CHALLENGES AND TECHNICAL SOLUTIONS

During the development process, several architectural challenges were overcome. One major hurdle was managing Cross-Origin Resource Sharing (CORS) between the React frontend (operating on port 3000) and the Spring Boot backend (operating on port 8081). This was resolved by implementing a global CORS configuration on the server side to allow secure and authorized data flow.

Another challenge involved ensuring that the product filtering system remained fast as the database grew. This was addressed by optimizing the database queries in the Spring Boot layer and implementing real-time filtering on the frontend, allowing users to navigate through categories and sub-categories with zero latency.

5. PROJECT STRUCTURE AND DEPLOYMENT

In accordance with the project requirements, the source code is organized into a strict hierarchy under the root `SourceCode` folder:

- **Frontend:** Contains the complete React project, including all dependencies, and is fully executable with the `npm start` command.
- **Backend:** A Maven-based Spring Boot project that can be run directly via IntelliJ IDE.
- **Database:** A dedicated folder containing the self-contained SQL file for database structure and data restoration.