Vietnam National University HCMC, International University

School of Computer Science and Engineering

Final Report

Group 09

Topic: SLATT ONLINE FASHION STORE

Course: Web Application Development

Instructor: Nguyen Trung Nghia, Nguyen Van Sinh

Member's list:

Number	Member' Name	Member' ID
1	Huỳnh Lâm Đăng Khoa (Leader)	ITCSIU21138
2	Nguyễn Ngọc Đình Trung	ITITIU20331
3	Lê Gia Bảo	ITITWE20020

Table of contents

List of figures

List of tables

CHAPTER 1: INTRODUCTION

- **Abstract**

  This report documents the entire process of planning, designing, developing, and deploying the simple e-commerce website for Slatt Fashion Store. Slatt Fashion is a dynamic online retailer targeting the youth market with trendy streetwear and contemporary fashion. The project aimed to create a fully functional, scalable web platform to serve as the primary digital sales channel. Key outcomes include an intuitive user interface, login, payment integration, a robust admin dashboard, and a platform with potential for future upgrades. This report details the systematic approach from analysis to implementation.

- **System Overview**

  The Slatt Fashion Store is a full-stack web application operating on a client-server model.

  - **Front-end (Client Side)**: It features dynamic components for product display, shopping cart, and user authentication, ensuring fast rendering and a smooth user interface without page reloads.

  - **Back-end (Server Side)**: Developed with Node.js and Express.js, creating a RESTful API that handles business logic, processes HTTP requests (e.g., user login, order submission), and communicates with the database.

  - **Database**: is used for its flexibility and scalability in managing diverse data types such as user profiles, product catalogs with multiple variants, and complex order histories.

- **Project Goals**

- Develop a secure, reliable, and visually appealing e-commerce website capable of effectively converting visitors into potential customers.
- Implement a complete product management system.
- Provide a secure and simple user registration, login, and profile management system.
- Integrate a fully functional shopping cart and checkout process with a payment gateway.
- Develop an admin dashboard to manage products, orders, and users.
- Ensure website security (data encryption).
- Create an intuitive and engaging user experience (UX) that aligns with streetwear aesthetics.

- **Techniques and tools used**

- To ensure the successful development and implementation of the Slatt Fashion Stote, a specific set of software, frameworks, and tools were employed. These were chosen to align with industry standards and the course requirements.
- Database Management System (DBMS): Microsoft SQL Server and My SQL workbench were selected as the relational database management system. It provides the robustness required to handle complex queries, enforce data integrity constraints, and manage concurrent user access effectively.
- Programming Language: The core logic of the application is developed using Java. Java was chosen for its platform independence, strong standard libraries, and object-oriented features which simplify the mapping of data objects to database tables.
- Web Interface Technologies: In addition to the core application logic, a web-based interface was developed using HTML (HyperText Markup Language) for structure, CSS (Cascading Style Sheets) for styling and layout, and JavaScript for client-side interactivity. This allows users to access key system functions via standard web browsers.

- Desktop GUI Framework: JavaFX was utilized to create the administrative desktop interface, providing a robust environment for complex management tasks and reporting.
- Development Environment: IntelliJ IDEA, VSCode, Eclipse,.... served as the Integrated Development Environment (IDE), offering advanced tools for code completion, debugging, and database connectivity testing.
- Database Connectivity: JDBC (Java Database Connectivity) driver was used to establish a secure and reliable connection between the Java application and the SQL Server database.
- Version Control: Git and GitHub were employed for version control, allowing the team to collaborate effectively, track changes to the source code and SQL scripts, and maintain a history of project iterations.

CHAPTER 2: TASK TIMELINE & DIVISION

1. Contribution Evaluating:

Project Contribution Assessment Framework (100%)

This framework outlines the criteria for assessing individual member contributions to the project, with a total possible score of 100%.

- **Timely Submission of Work (10%):**

Assesses adherence to agreed-upon deadlines for task and deliverable submissions.

- **Content Completion and Quality (60%):**

Evaluates the thoroughness, accuracy, depth, and overall quality of the assigned work and its contribution to the project's objectives.

- **Team Contribution and Collaboration (20%):**

Measures active participation in group discussions, offering constructive ideas, supporting other team members, and overall positive engagement in teamwork.

- **Presentation Delivery (10%):**

Recognizes active participation in delivering a portion of the final project presentation.

Scoring:

Each member's final contribution percentage will be calculated as the sum of the scores achieved across these five criteria. This system aims to provide a fair and transparent evaluation of individual efforts.

Table 1: Individual responsibility and contribution (%)

Member' Name	Member' ID	Evaluation in the respective	Total
Huỳnh Lâm Đăng Khoa	ITCSIU21138	Frontend	33.33%
Nguyễn Ngọc Đình Trung	ITITIU20331	Backend + Database	33.33%
Lê Gia Bảo	ITITWE20020	Backend + Report	33.33%

CHAPTER 3: USE CASE DIAGRAM AND ANALYSIS

1. Overview:

The Use Case Diagram illustrates the core functionalities of the Slatt Fashion Store e-commerce system and the interactions between the system and its primary actors: Customer, Admin, and System. The purpose of this diagram is to clearly define the functional scope of the application and to specify how each actor interacts with the system to achieve specific goals.

By modeling user interactions at a high level, the Use Case Diagram helps the development team gain a comprehensive understanding of user requirements, identify essential business processes, and establish clear system boundaries. This serves as a foundation for subsequent phases, including system design, database modeling, and implementation.

2. List Actor:

2.1. Customer:

The Customer represents end users who interact with the system to browse and purchase products. Customers are able to:

- Create an account and register personal information
- Log in to the system to access personalized features
- Browse and view available fashion products
- Add selected products to the shopping cart
- Place orders and track the status of their orders

These interactions reflect the primary business flow of the e-commerce platform, focusing on usability and a smooth shopping experience.

2.2. Admin:

The Admin actor represents authorized personnel responsible for managing the system's content and ensuring smooth operation of the online store. Admin functionalities include:

- Logging into the system with administrative privileges
- Managing product data by adding new products, updating existing product information, and deleting products when necessary
- Maintaining the accuracy and availability of products displayed to customers

The Admin role ensures that the product catalog remains up-to-date and that business operations are properly controlled.

2.3. System:

The System actor represents internal backend processes and services that operate automatically without direct user interaction. The system is responsible for:

- Authenticating users and verifying login credentials
- Checking product availability and stock levels
- Calculating order totals and processing order-related logic
- Storing, retrieving, and updating data in the MySQL database
- Ensuring secure and reliable communication between the frontend and backend components

The System actor encapsulates core business logic and data management, enabling seamless interaction between users and the application.

-

3. Design

3. 1. Use Case Diagram:

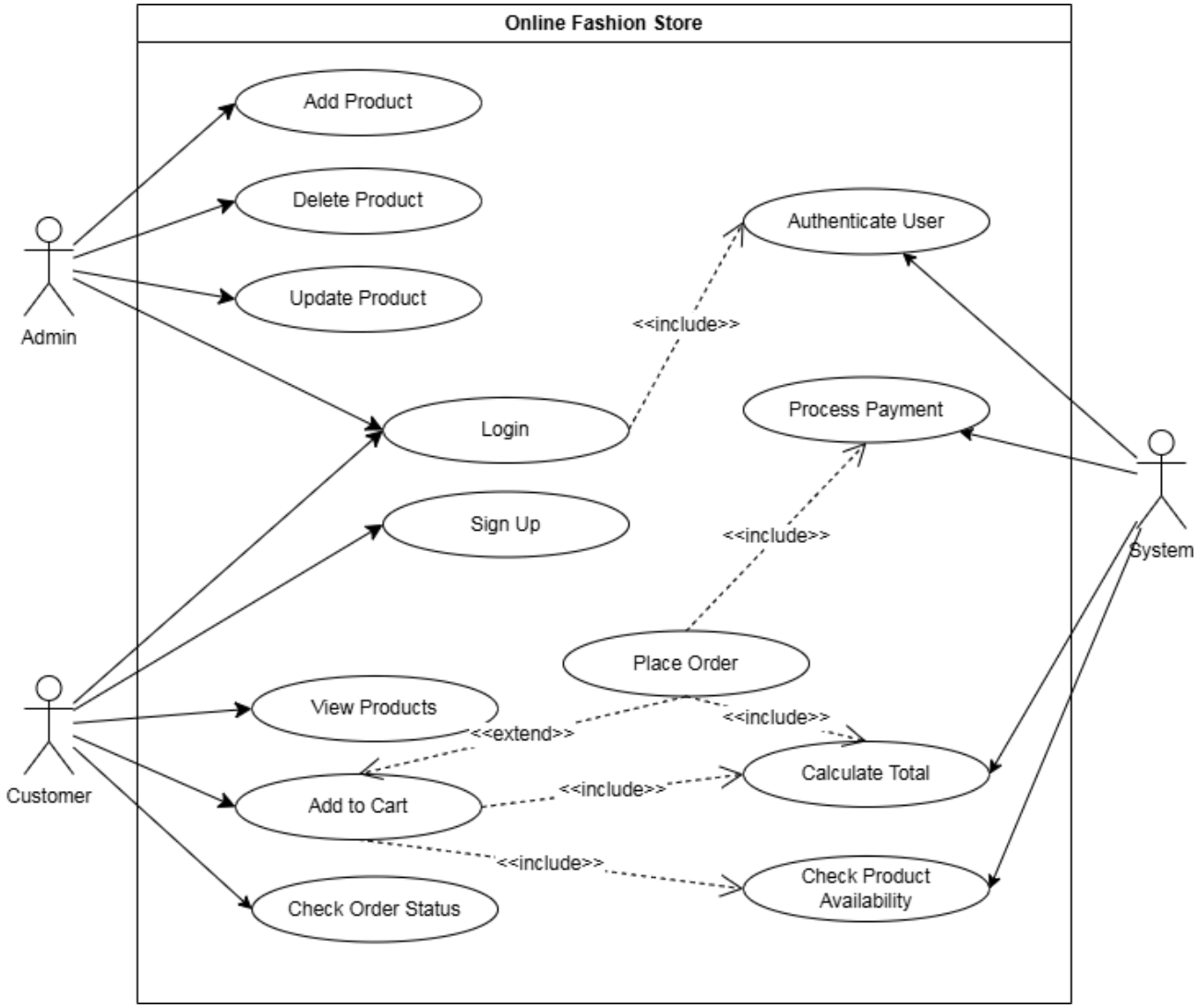

Figure 1: Use Case Diagram

The Online Fashion Store system is designed to support online shopping activities for customers and management operations for administrators. The system involves three main actors: Customer, Admin, and System, each playing a different role in the overall process.

Customers interact with the system to browse available products, manage their shopping cart, place orders, and track order status. They can view product information, add selected products to the cart, and proceed to checkout. During the ordering process, the system automatically performs essential operations such as checking product availability, calculating the total order cost, and processing payments. These system operations are

mandatory and are modeled as included use cases to ensure data consistency and transaction correctness.

Admin are responsible for managing the product catalog within the system. They can add new products, update existing product information, and delete products when necessary. To access these management functions, administrators must first log in to the system. Authentication and authorization mechanisms are enforced by the system to ensure that only authorized users can perform administrative actions.

The System actor represents internal services and automated processes that support business logic execution. These include user authentication, authorization control, product availability checking, total price calculation, and payment processing. The system does not initiate use cases independently but participates in included use cases to handle background processing and validation.

3.2. Use Case Login:

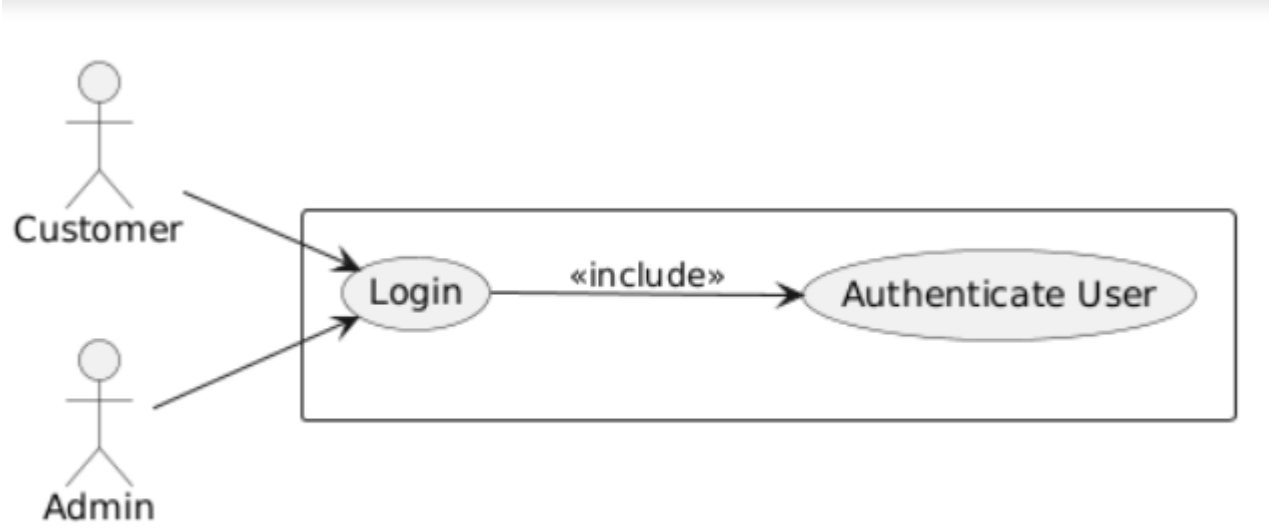

Figure 2: Log In

This use case describes the process by which a user (Customer or Admin) logs into the Online Fashion Store system. The user provides valid authentication credentials in order to gain access to system functionalities based on their role.

3.3. Use Case Add to Cart:

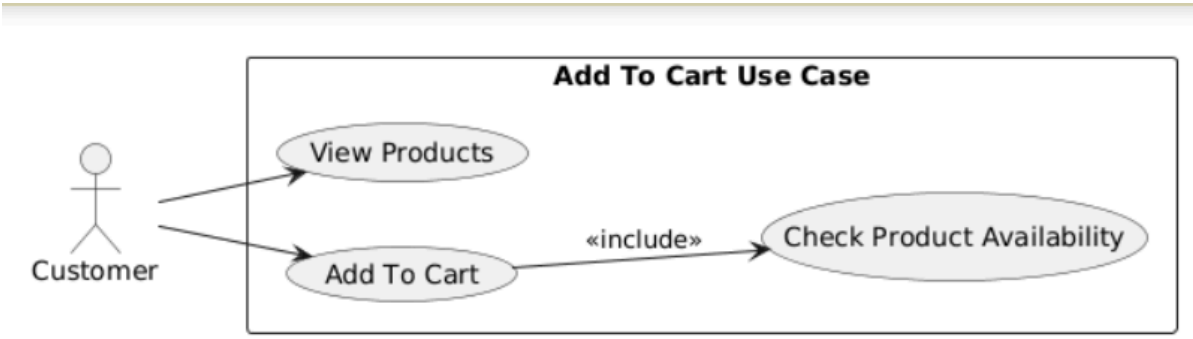

Figure 3: Add To Cart

This use case describes how a customer adds a selected product to their shopping cart. The system ensures that the product is available before adding it to the cart.

3.4. Use Case Place Order:

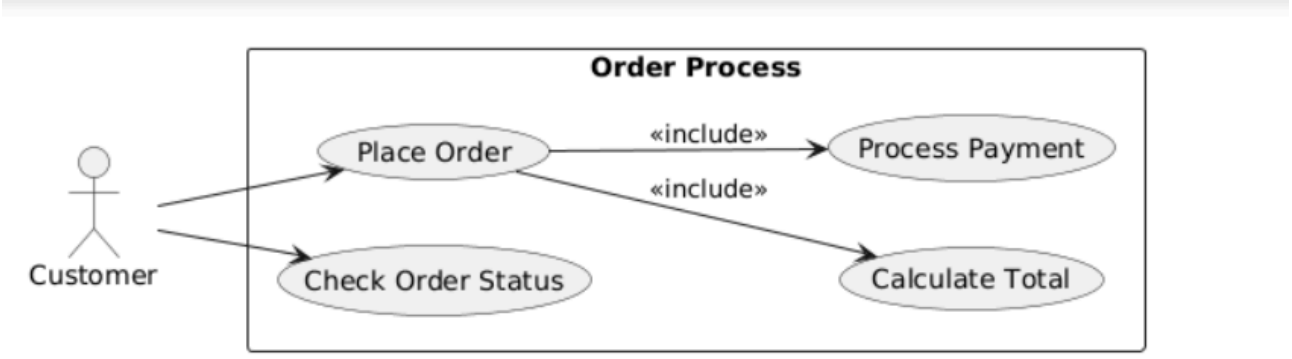

Figure 4: Place Order

This use case allows a customer to view the current status of their previously placed orders. The system retrieves and displays order-related information such as order status and details.

3.5. Register sequence diagram:

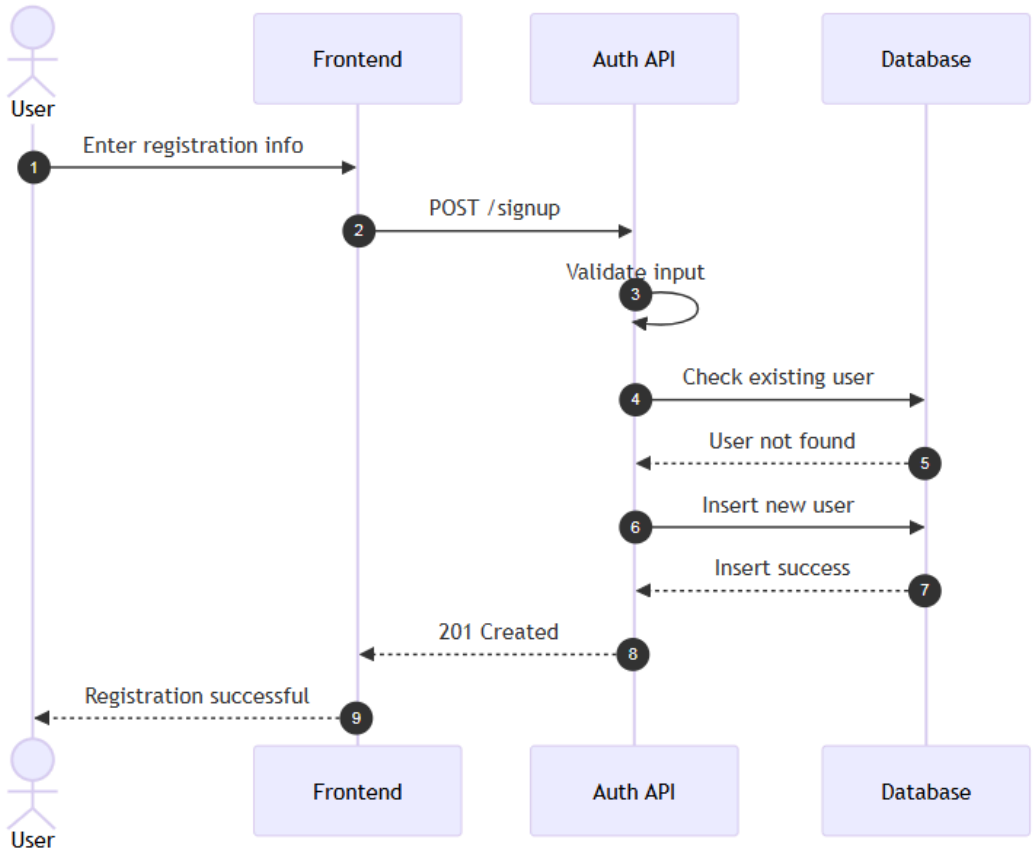

Figure 5: Sign Up

The Sign Up sequence diagram illustrates the process of user registration in the system. The user provides registration information through the frontend interface, which is then sent to the authentication API. The backend validates the input data and checks whether the user already exists in the database. If no duplicate account is found, a new user record is created and stored in the database. Finally, the system returns a successful response to the frontend, informing the user that the registration process has been completed successfully.

3.6. Login sequence diagram:

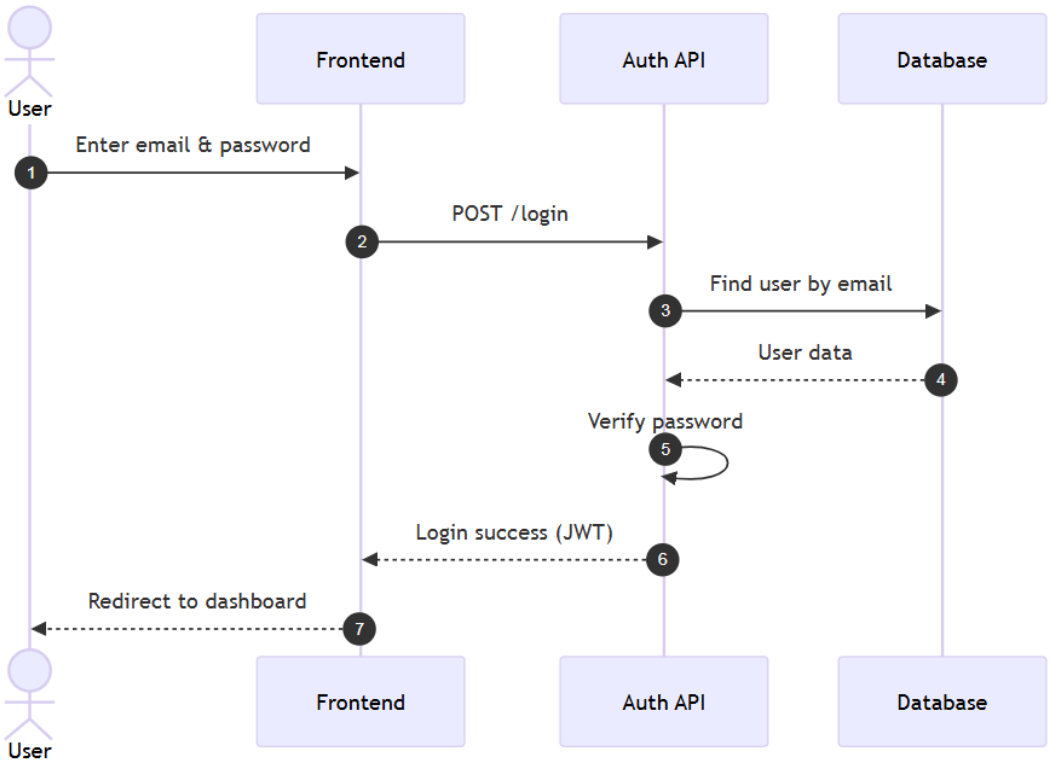

Figure 6: Log In

The Login sequence diagram describes how a user authenticates into the system. The user submits login credentials via the frontend interface. These credentials are forwarded to the authentication API, which retrieves the corresponding user data from the database. The backend verifies the password and authentication information. Upon successful verification, the system grants access by returning a success response, typically including an authentication token, and redirects the user to the appropriate dashboard.

3.7. Add Product sequence diagram:

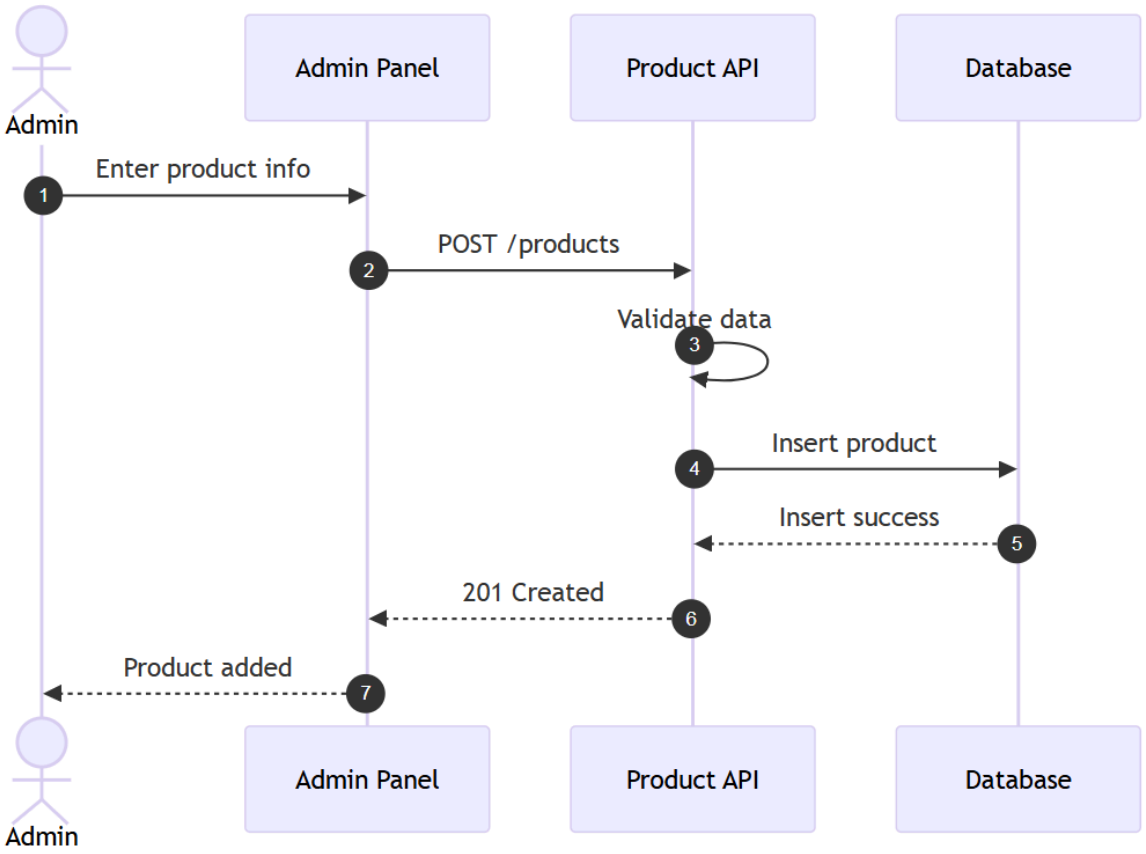

Figure 7: Add Product

The Add Product sequence diagram illustrates how an administrator adds a new product to the system. The admin enters product details through the admin panel, which sends the data to the product management API. The backend validates the input data and stores the new product information in the database. Once the insertion is successful, the system responds with a confirmation message, and the new product becomes available for customers to view.

3.8. Delete Product sequence diagram:

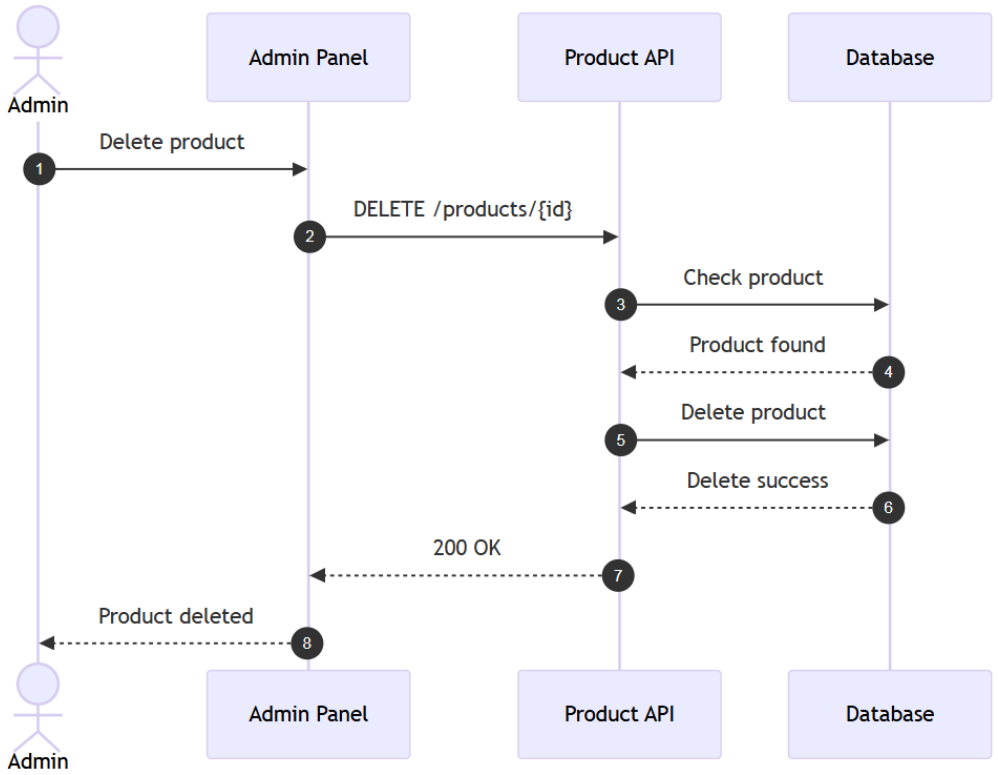

Figure 8: Delete Product

The Delete Product sequence diagram describes how an administrator removes a product from the system. After the admin initiates the delete action, the request is sent to the product API. The backend verifies the existence of the product before deleting it from the database. Upon successful deletion, the system confirms the operation and updates the frontend accordingly. In real-world applications, this operation can be extended to a soft delete mechanism to preserve data integrity.

3.9. Update Product sequence diagram:

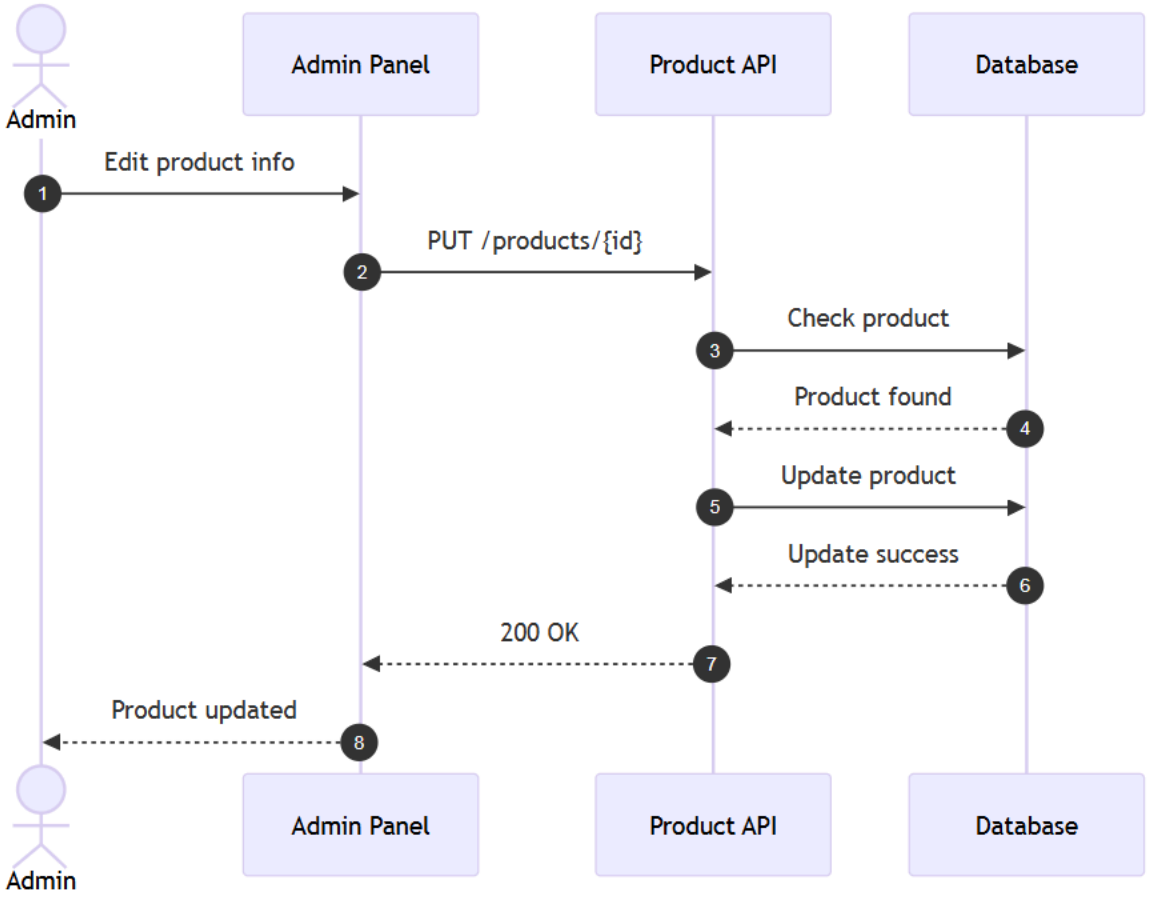

Figure 9: Update Product

The Update Product sequence diagram demonstrates the process of modifying existing product information. The administrator selects a product and updates its details through the admin interface. The product API first verifies the existence of the product in the database and then applies the updated information. After the database is successfully updated, the system notifies the admin that the product information has been updated.

3.10. Add to Cart sequence diagram:

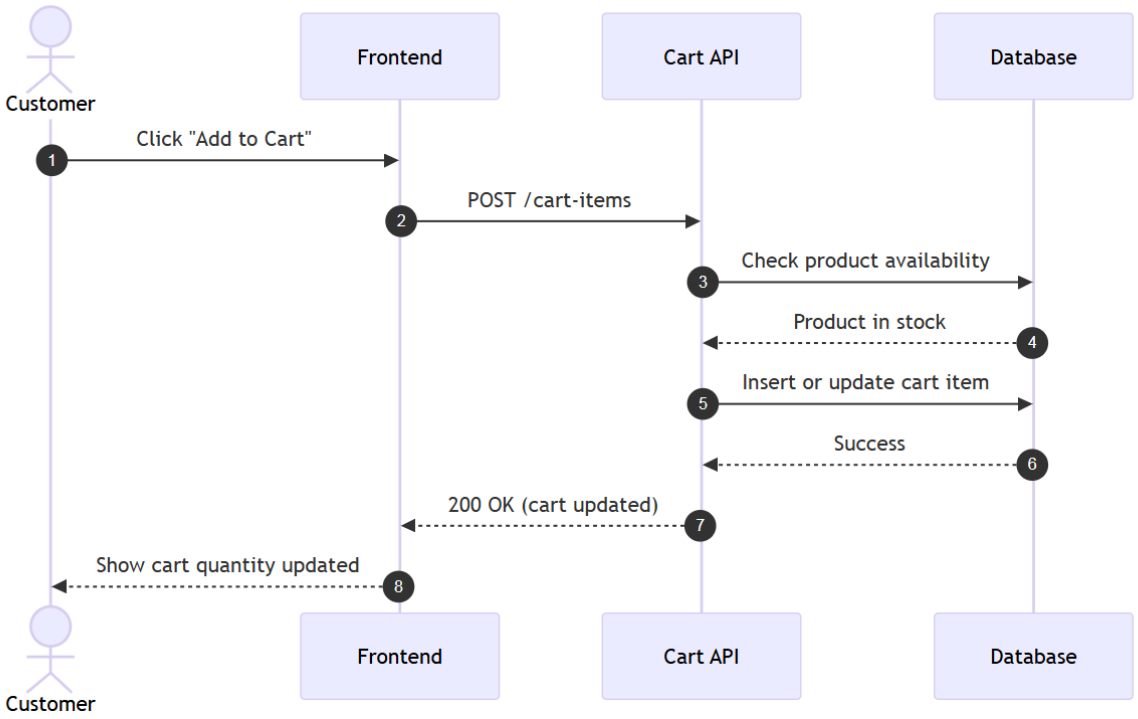

Figure 10: Add To Cart

The Add to Cart sequence diagram illustrates the interaction between the customer and the system when adding a product to the shopping cart. The customer selects a product and triggers the add-to-cart action from the frontend interface. The cart API checks product availability in the database to ensure sufficient stock. If the product is available, the system inserts a new cart item or updates the existing cart item quantity. The frontend then displays an updated cart status to the customer.

3.11. Place Order sequence diagram:

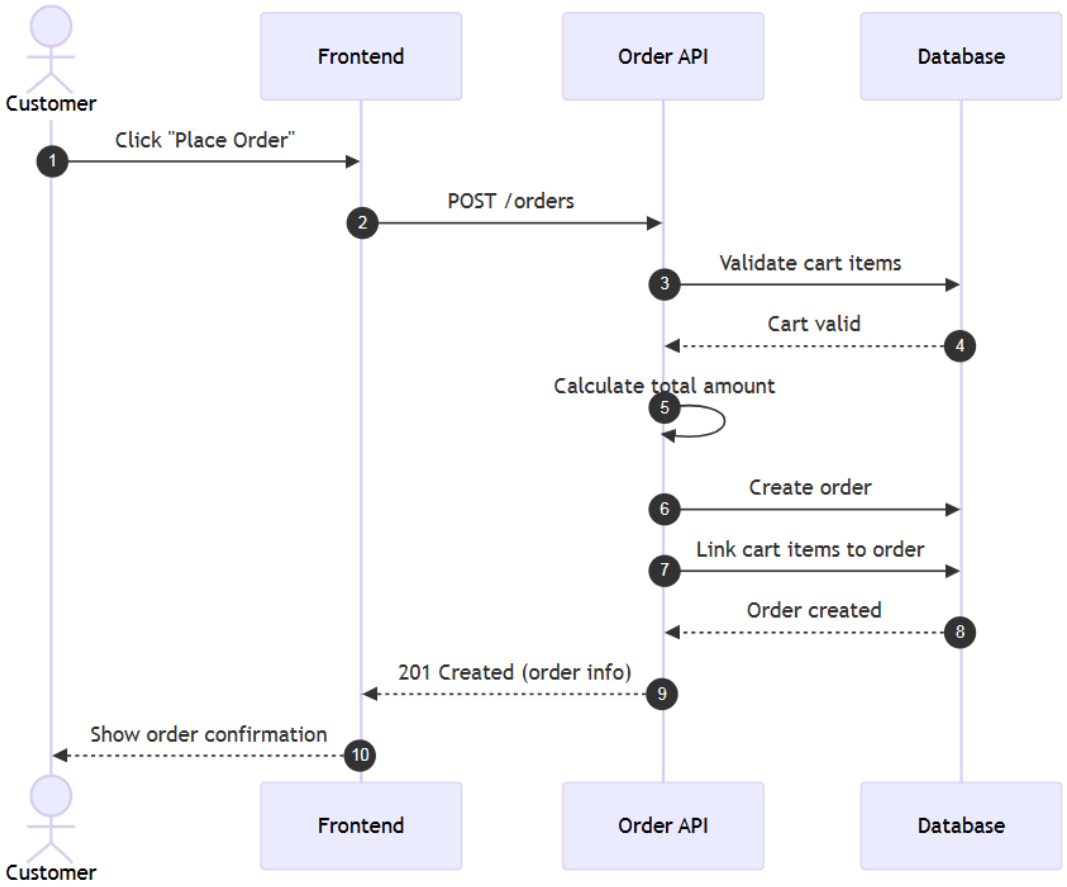

Figure 11: Place Order

The Place Order sequence diagram represents the checkout process in the system. When the customer confirms the order, the frontend sends a request to the order API. The backend validates the cart items, calculates the total order amount, and creates a new order record in the database. The associated cart items are linked to the order, and the system returns an order confirmation response. The customer is then notified that the order has been placed successfully.

CHAPTER 4: CONNECT DATA BACKEND AND FRONTEND

The user interface of the Slatt Fashion Store is designed as a web-based application developed using React.js and JavaScript. The interface focuses on providing a clean, intuitive, and user-friendly experience for both customers and administrators. Responsive design techniques are applied to ensure compatibility across multiple devices, including desktops, tablets, and smartphones. The system includes well-structured forms for user authentication, product management, cart operations, and order processing, enabling efficient data entry and interaction. This design approach improves usability, accessibility, and overall user satisfaction.

1. Connect Sever

The backend of the system is implemented using Node.js, with the main server configuration defined in the index.js file. The Node.js server is responsible for handling API requests from the frontend, managing business logic, and coordinating communication between the user interface and the database. By utilizing a RESTful API architecture, the server ensures modularity, scalability, and clear separation of concerns between frontend and backend components.

2. Connect Database

The system connects to a MySQL database through a dedicated connectToDatabase function. This function establishes a connection between the Node.js server and the database, ensuring that data operations such as user authentication, product management, cart handling, and order processing are executed reliably. When the connection is successful, the system operates normally and is ready to process requests. In the event of a connection failure, error information is logged to the console to assist in debugging and maintenance. The connectToDatabase() function is typically invoked during server startup to guarantee database availability before handling client requests.

3. GUI design

The graphical user interface (GUI) of the Slatt Fashion Store is designed with a strong focus on simplicity, usability, and modern aesthetics. The interface is developed as a web-based application using React.js, ensuring smooth interaction and efficient component-based rendering.

The layout follows a responsive design approach, allowing the website to adapt seamlessly across different devices, including desktops, tablets, and smartphones. Navigation elements such as the header menu, product listings, cart icon, and user account access are consistently positioned to provide a familiar and intuitive user experience.

The homepage features a large visual banner used to highlight brand identity, promotions, and featured products, encouraging user engagement. Forms for actions such as login, sign up, adding products to the cart, and checkout are designed to be user-friendly, with clear labels and validation feedback.

Overall, the GUI design aims to enhance the shopping experience by providing a clean interface, smooth navigation, and visually appealing presentation while maintaining ease of use for both customers and administrators.

4. Web demo

Homepage: The homepage serves as the main landing page of the Slatt Fashion Store, featuring a visually striking banner that reflects the brand identity. It highlights promotional content and featured products, along with a clear call-to-action button that encourages users to explore products and place orders. The navigation menu provides quick access to key sections such as Home, Products, About Us, and Contact Us, ensuring a smooth and intuitive user experience.

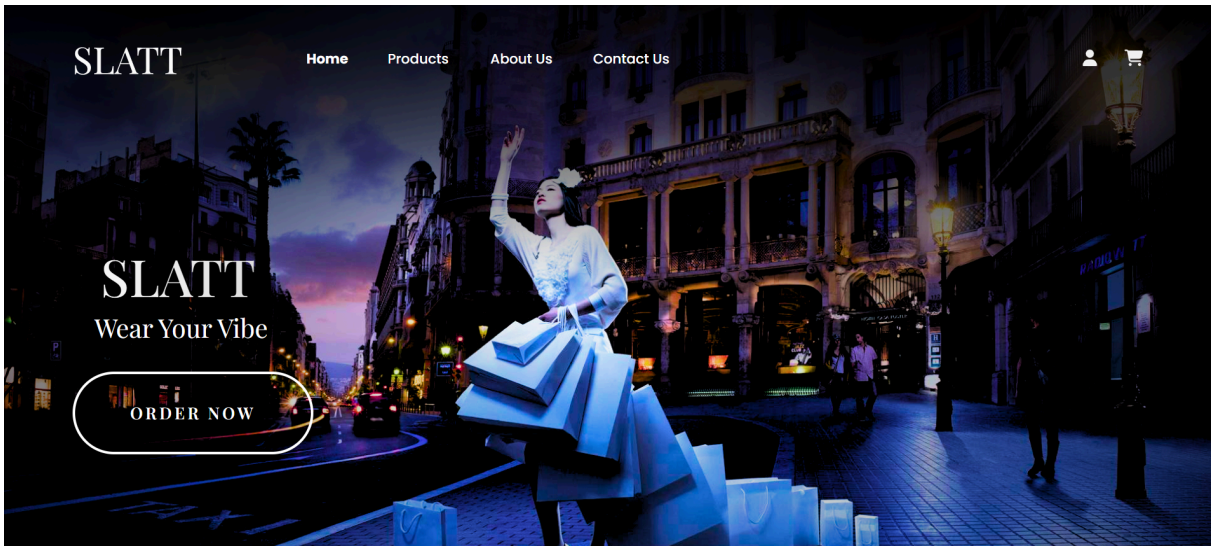

Figure 12. Homepage user interface

Product page: The product page features a sleek, modern UI with a nighttime cityscape background that enhances its stylish appeal. At the top, a navigation bar provides quick access to Home, Products, About Us, and Contact Us, alongside a search bar and a sorting dropdown labeled "Sort by Name (A-Z)." On the left, users can filter products by category. The main section displays a grid of product cards, each showing an image, name, and price. The layout is clean and intuitive, making it easy for users to browse and refine their shopping experience.

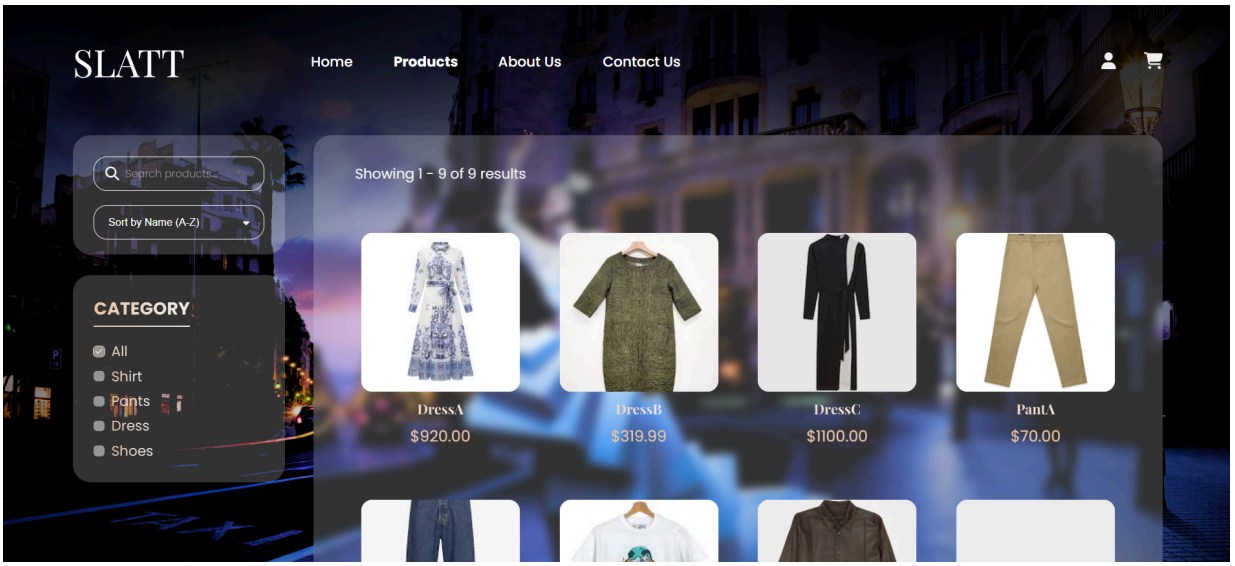

Figure 13. Product page

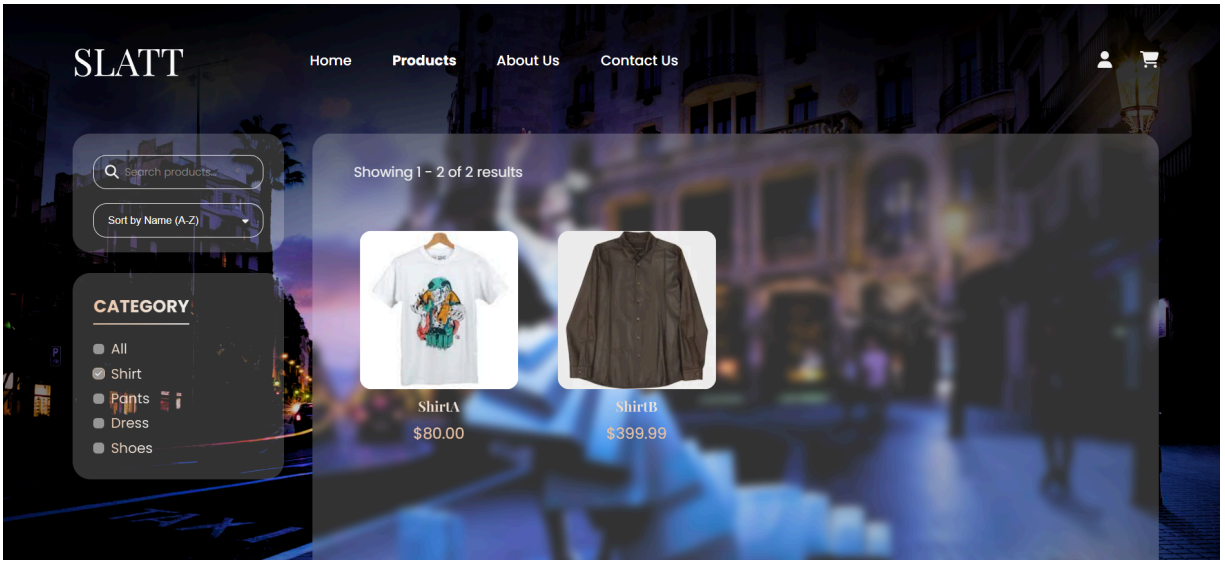

Figure 14. Filter with option "Shirt" in product page

Shopping cart page: The shopping cart page presents a clean and inviting interface with a blurred city street background that adds a touch of elegance. At the top, a navigation bar includes links to Home, Products, About Us, and Contact Us, along with icons for user account and shopping cart. The central section displays a friendly message accompanied by a prominent “Continue Shopping” button. The design suggests an interactive cart experience, where item quantities can be updated dynamically based on customer selections.

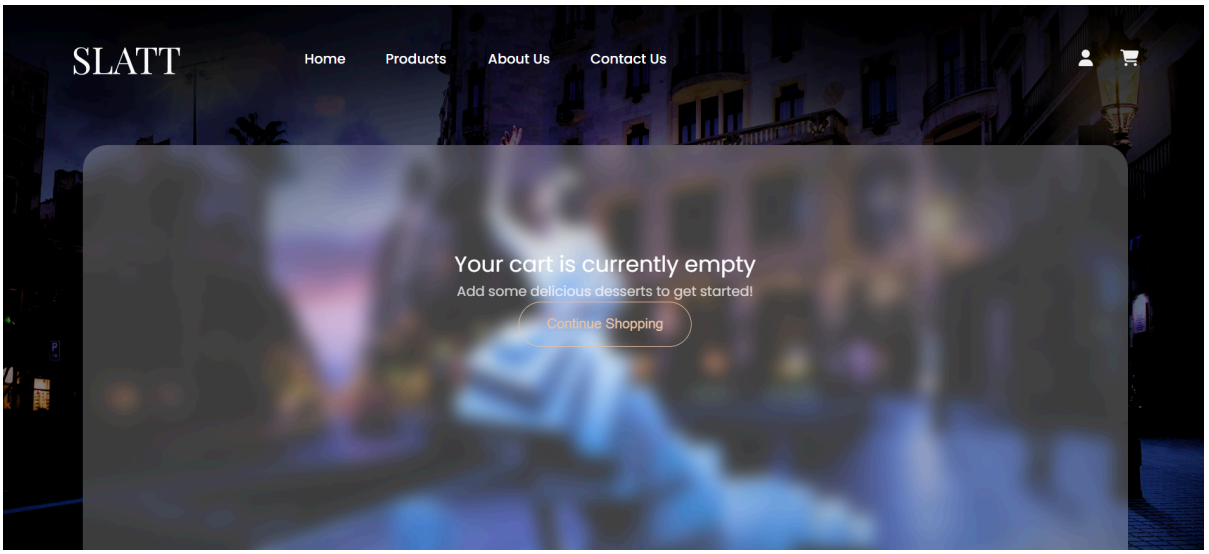

Figure 15. Shopping cart page when cart empty

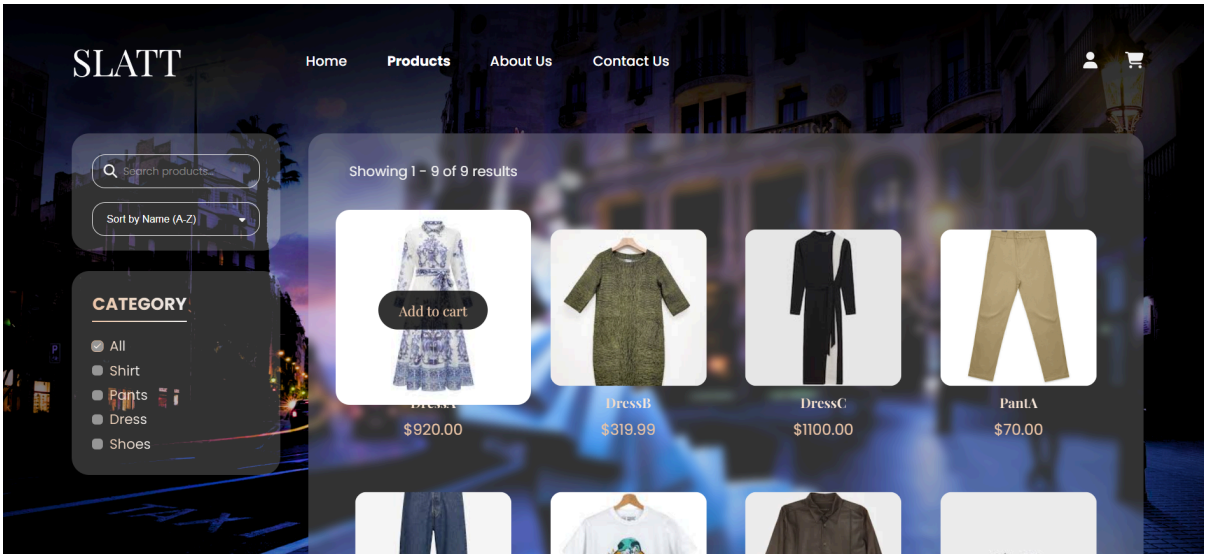

Figure 16. Add to cart button in shopping cart page

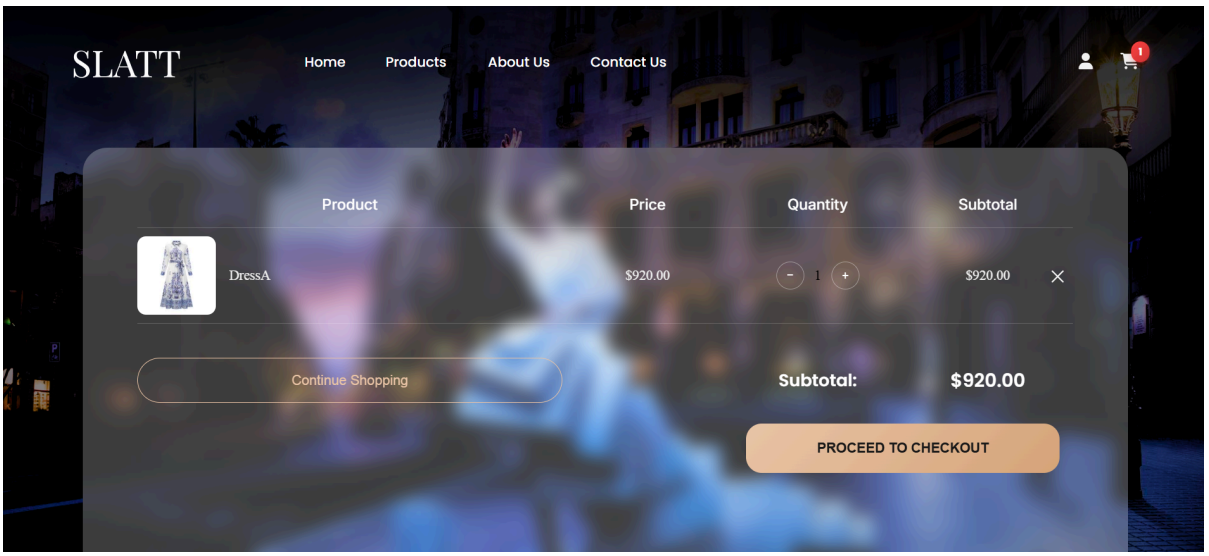

Figure 17. Shopping cart page when product is added

Checkout page: The checkout page showcases a refined, user-friendly interface set against a nighttime city street backdrop. At the top, a navigation bar offers links to key sections like Home, Products, About Us, and Contact Us, with a user icon displaying a notification badge. The central overlay presents a multi-step form that captures delivery details followed by a summary of the order. A clear breakdown of the item and subtotal is shown, with a prominent "PLACE ORDER" button at the bottom, guiding users through a smooth and intuitive checkout experience.

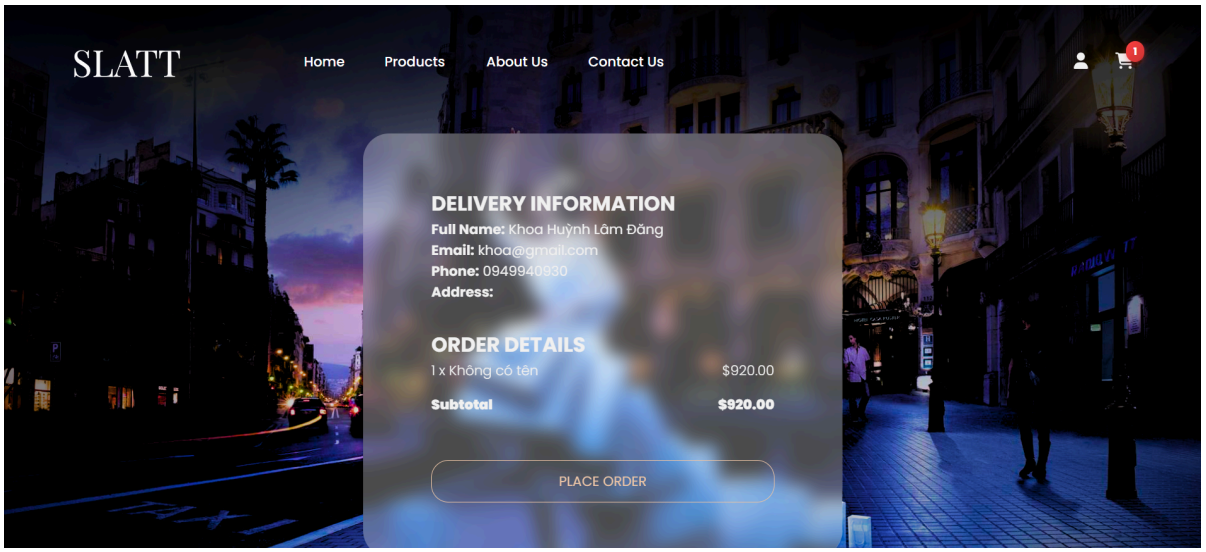

Figure 18. Checkout page

CHAPTER 5: CONCLUSION & FUTURE WORK

1. Conclusion

The Slatt Fashion Store e-commerce website project has been successfully designed, developed, and implemented in accordance with the initial objectives. The system provides a basic yet fully functional online shopping platform that ensures usability, security, and reliability for both customers and administrators. Through this project, key aspects of full-stack web development have been effectively applied, including database schema design, backend business logic implementation, and frontend user interaction. The project demonstrates a holistic development approach, showcasing the integration of system components into a coherent and user-friendly e-commerce solution.

2. Future Work

Although the current system fulfills the core functional requirements, several enhancements can be considered to further improve the Slatt Fashion Store. Future development may include the implementation of advanced features such as a product recommendation engine based on user behavior and purchase history. Social and community-oriented functionalities, including user-generated reviews with images, wishlists, sharing capabilities, and social media authentication, could be integrated to increase user engagement. Additionally, supporting multiple languages would enhance accessibility for international users. To improve the overall user experience, dedicated mobile applications for iOS and Android platforms could be developed. Finally, integrating an analytics dashboard would allow administrators to monitor user behavior, sales performance, and key business metrics, enabling more data-driven decision-making.

REFERENCES

1. MySQL Documentation (https://dev.mysql.com/doc/)

2. Normal Form (https://www.geeksforgeeks.org/normal-forms-in-dbms/)

3. Library connect database (Java JDBC API)

4. Designing UI with Next.js (https://nextjs.org)

5. MySQL Workbench Manual (https://dev.mysql.com/doc/workbench/en/)

6. Designing Backend with SpringBoot (https://spring.io/projects/spring-boot)

7. Mapping support in Backend (Mapping: Spring Data Relational)

8. Tutorial: Develop the website (Wondering How To Make A Website? Here's A Complete Tutorial - DreamHost)