

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

CHAPTER 1: INTRODUCTION	7
1. Abstract	7
2. System Overview	7
3. Project Goals	7
4. Techniques and tools used	8
CHAPTER 2: TASK TIMELINE & DIVISION	9
1. Contribution Evaluating:	9
2. Project timeline & task division:	10
CHAPTER 3: PROJECT ANALYSIS	11
1. Approach Analysis	11
1.1. Requirement Analysis	11
1.2. System Design	11
1.3. Development Methodology	11
1.4. Development and Maintenance	11
2. System Analysis	11
2.1. Database Design	11
2.2. Data Query	11
CHAPTER 4: CONNECT DATA BACKEND AND FRONTEND	12
1. Connect Sever	
2. Connect Database	
3. Java Script to promote the interact	
4. GUI design	
5. Web demo	
CHAPTER 5: CONCLUSION & FUTURE WORK	
1. Conclusion	
2. Future work	
REFERENCES	

List of figures

Figure 1:	5
Figure 2:	7
Figure 3:	7
Figure 4:	8
Figure 5:	8
Figure 6:	9
Figure 7:	9
Figure 8:	10

List of tables

Table 1: Individual responsibility and contribution (%).....

Table 2: Detailed timeline and task division.....

Table 3: Normal form achievement.....

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 Định Trung	ITITIU20331	Backend + Database	33.33%
Lê Gia Bảo	ITITWE20020	Backend + Report	33.33%

CHAPTER 3: PROJECT ANALYSIS

1. Approach Analysis:

1.1. Requirement Analysis

The system allows for account registration and login, user authentication, product categorization, shopping cart/checkout, payment processing, and an admin dashboard. Additionally, it includes advanced filters, a search function, and a wishlist.

1.2. System Design

The system follows a layered architecture: Presentation Layer (React), Application Layer (Express API), and Data Layer. This separation ensures modularity, accuracy, maintainability, and future scalability.

1.3. Development Methodology

1.4. Deployment and Maintenance:

2. Systems Analysis:

2.1. Database Design

User Collection: Stores credentials, shipping addresses.

Product Collection: Stores details, images, price, countInStock.

Order Collection: References User and contains an array of ordered items, shipping/payment status.

2.2. Database Query

Queries are executed via MySQL

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The main window has a 'Navigator' pane on the left showing 'SCHEMAS' (slatt_fashion) and 'Tables' (cart_items, orders, products, users). A central pane displays a query editor with the SQL command: 'SELECT * FROM slatt_fashion.products;'. Below the query is a 'Result Grid' showing a list of products with columns: id, name, description, category, imageURL, price, available, stock, createdAt, and updatedAt. The results show 10 rows of product data. At the bottom, there is an 'Output' pane titled 'Action Output' showing the history of actions taken: '1 12:06:09 SELECT * FROM slatt_fashion.products LIMIT 0, 1000', '2 12:06:19 DROP DATABASE 'student-management'', and '3 12:06:23 DROP DATABASE 'product_management''. The 'Message' column indicates the outcome of each action.

#	Time	Action	Message
1	12:06:09	SELECT * FROM slatt_fashion.products LIMIT 0, 1000	9 row(s) returned
2	12:06:19	DROP DATABASE 'student-management'	5 row(s) affected
3	12:06:23	DROP DATABASE 'product_management'	1 row(s) affected

Figure 1. Queries are executed via MySQL

CHAPTER 4: CONNECT DATA BACKEND AND FRONTEND

The user interface design focuses on creating a web-based interface using React.js and JavaScript. The design is responsive to support various devices (desktops, tablets, smartphones) and features user-friendly forms for data entry and management.

1. Connect Sever

It sets up a Node.js server using index.js. And, it configures the backend environment to handle API requests from the frontend.

2. Connect Database

This “connectToDatabase” function helps connect the Node.js server to the MongoDB database. and, it helps ensure that if the connection is successful, the system operates normally, and if there is an error, it reports to the console. Then we will usually call “connectToDatabase()” when the Node.js server starts.

3. Java Script to promote the interact

4. GUI design

5. Web demo

Homepage: Main banner, where announcements and featured product advertisements are displayed.

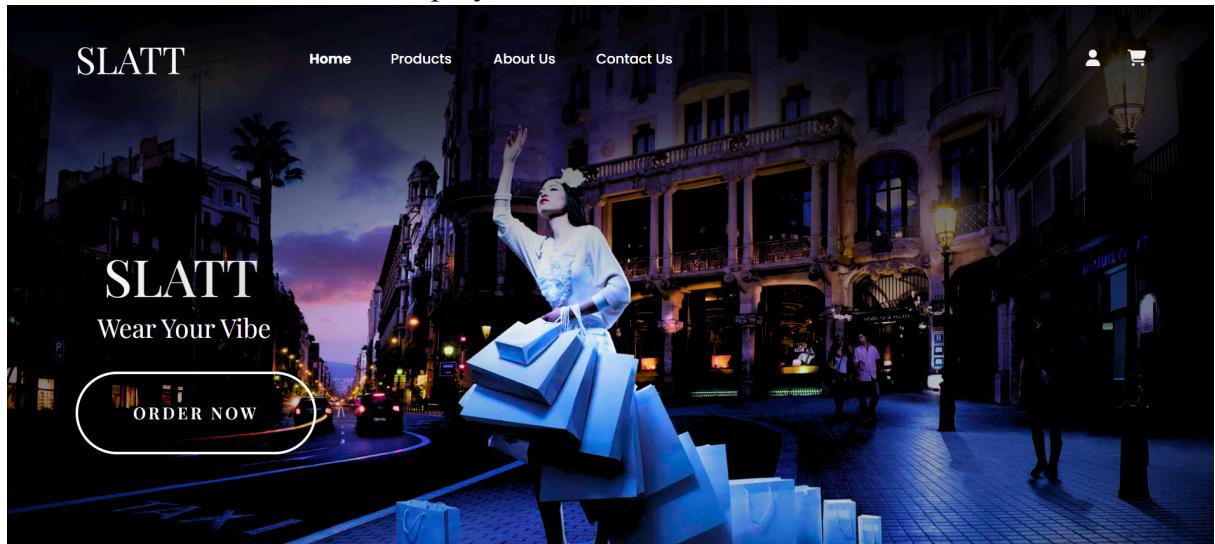


Figure 2. Homepage user interface

Product page: Filterable product grid.

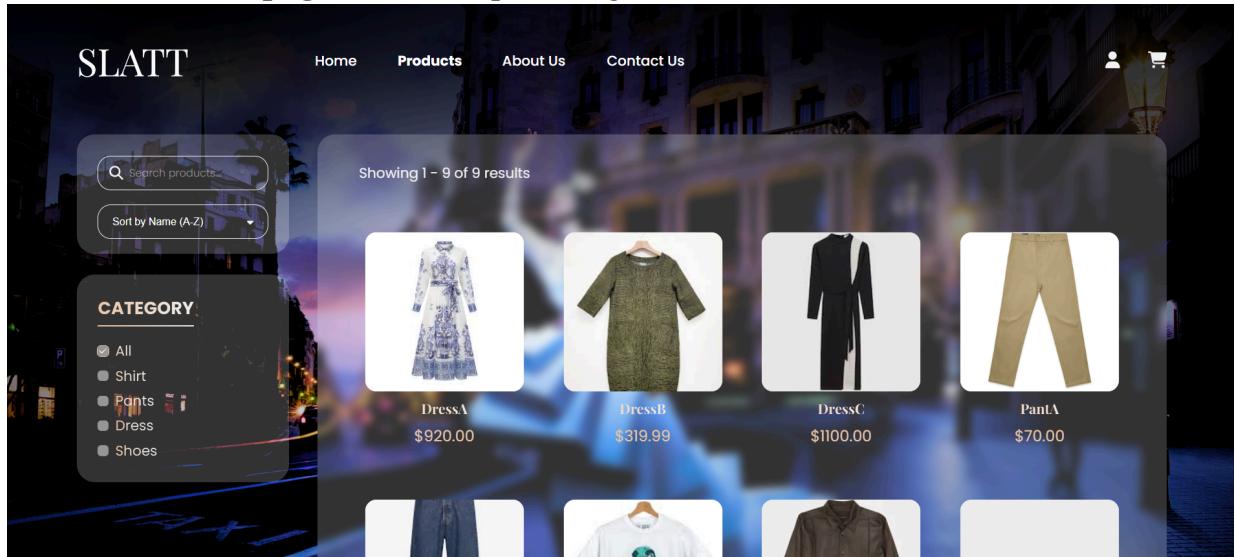


Figure 3. Product page

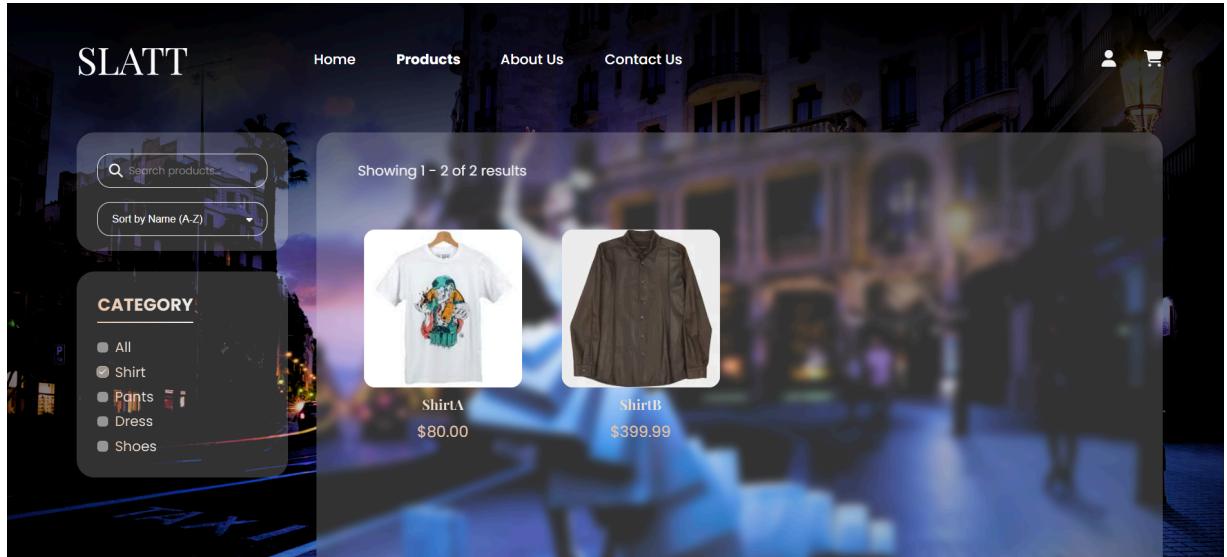


Figure 4. Filter with option “Shirt” in product page

Shopping cart page: Interactive shopping cart with updated quantities based on customer selections.

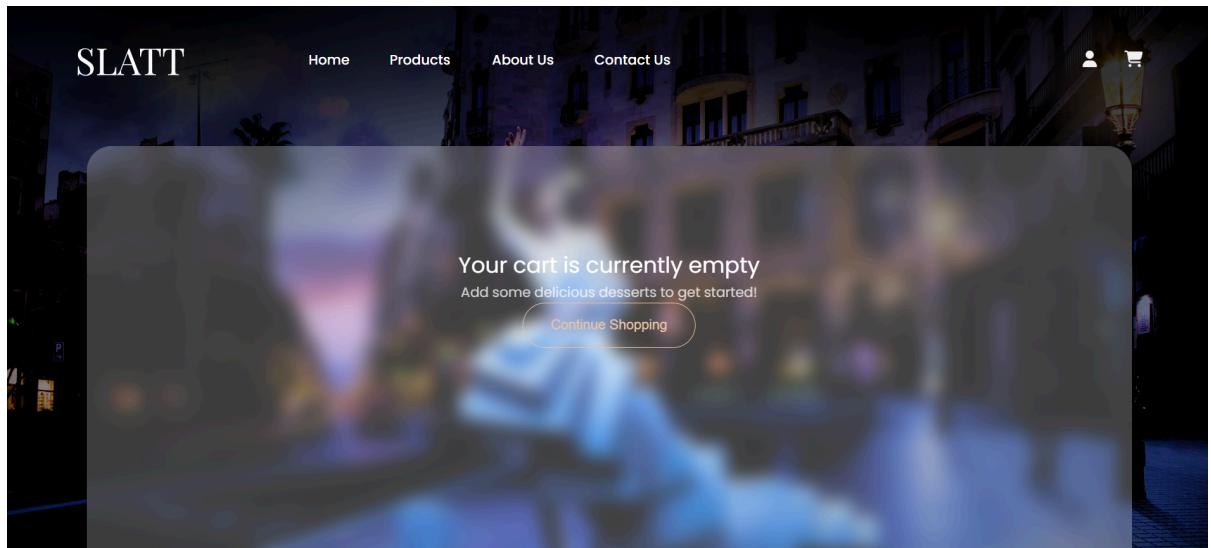


Figure 5. Shopping cart page when cart empty

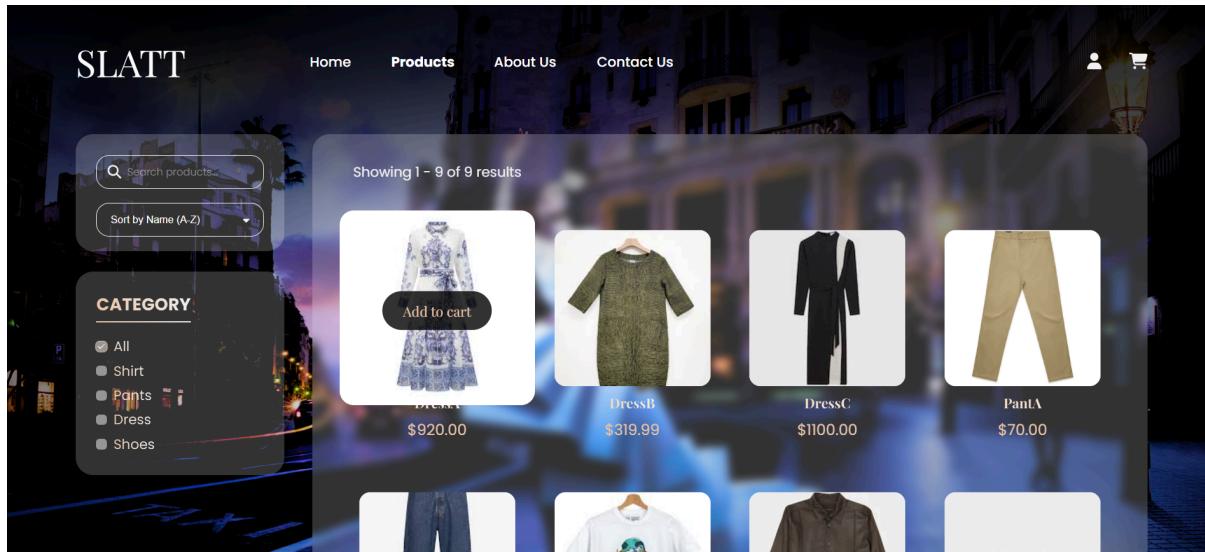


Figure 6. Add to cart button in shopping cart page

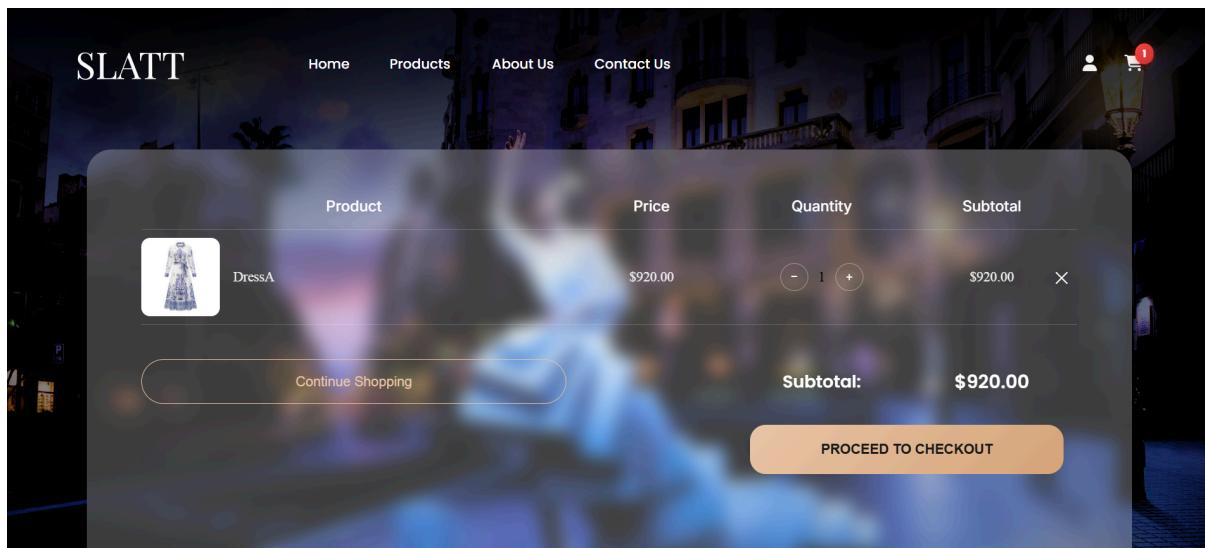


Figure 7. Shopping cart page when product is added

Checkout page: Multi-step form with integrated checkout demo.

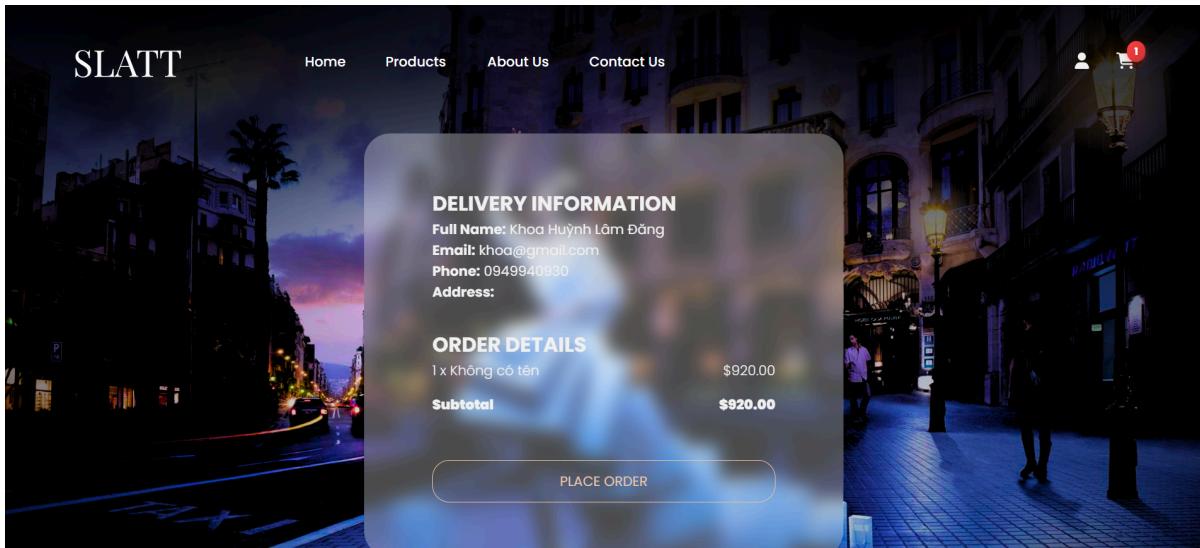


Figure 8. Checkout page

CHAPTER 5: CONCLUSION & FUTURE WORK

1. Conclusion

The simple e-commerce website project for the Slatt fashion store has been successfully developed and implemented. All initial objectives have been met: providing a basic, fully functional, secure, and user-friendly platform. This project has demonstrated the effective application of holistic development principles, from database design to user interface interaction.

2. Future Work

To enhance the Slatt Fashion Store, planned improvements include:

1. **Advanced Features:** Implement a product recommendation engine based on user behavior and purchase history.
2. **Social & Community:** Integrate user-generated content (reviews with photos), a wishlist/sharing function, and social media login.
3. **Multiple Language Support:** Enhancing accessibility for global users.
4. **Mobile Experience:** Develop via iOS and Android mobile engagement.
5. **Analytics Dashboard:** Integrate advanced analytics to track user behavior and sales metrics for data-driven decisions.

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](#))