



PREMIER UNIVERSITY  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

*A Project Report On*  
**E-Commerce Platform**

**Course Title:** Software Development

**Course Code:** CSE 364

Submitted To:  
Dhrubajyoti Das  
Assistant Professor  
Department of Computer Science and Engineering

Submitted By:  
**Md. Mostafa Jaman Taif** (2104010202213)  
**Saikat Barua** (2104010202196)  
**Jahirul Islam** (2104010202209)  
**Md Ismail Hossain** (2104010202210)

September, 2024

# TABLE OF CONTENTS

<b>TITLE PAGE</b>	<b>i</b>
<b>TABLE OF CONTENTS</b>	<b>ii</b>
<b>LIST OF FIGURES</b>	<b>iv</b>
<b>LIST OF TABLES</b>	<b>1</b>
<b>1 Introduction</b>	<b>2</b>
<b>2 Problem Statement</b>	<b>2</b>
<b>3 Objectives</b>	<b>2</b>
<b>4 Methodology</b>	<b>4</b>
4.1 Requirement identification . . . . .	4
4.1.1 Study of Existing System / Literature Review . . . . .	4
4.1.2 Requirement Analysis . . . . .	5
4.2 Feasibility Study . . . . .	6
4.2.1 Technical . . . . .	6
4.2.2 Operational Feasibility . . . . .	9
4.2.3 Economic Feasibility . . . . .	10
4.2.4 Schedule(Gantt chart showing the project timeline) . . . . .	11
4.3 High-Level Design of System . . . . .	12
4.3.1 Methodology of the proposed system . . . . .	12
4.4 Flow Charts/Working Mechanism of Proposed System . . . . .	15
<b>5 Expected Output</b>	<b>20</b>
<b>REFERENCES</b>	<b>26</b>

## List of Figures

4.1	Requirement identification process(Flow Chart) . . . . .	4
4.2	Current System Limitation . . . . .	5
4.3	System Integration Workflow . . . . .	7
4.4	Gantt Chart demonstrating schedule feasibility . . . . .	11
4.5	Data Flow Diagram for Proposed System . . . . .	15
4.6	Entity Relationship Diagram for Proposed System . . . . .	16
4.7	Use Case Diagram for Proposed System . . . . .	17
4.8	Class Diagram for Proposed System . . . . .	18
4.9	Sequence Diagram for Proposed System . . . . .	19
5.1	Interface of Happy Shopping . . . . .	21
5.2	Login . . . . .	21
5.3	Login As A User . . . . .	21
5.4	Add To Cart . . . . .	22
5.5	Show Cart . . . . .	22
5.6	Payment method . . . . .	22
5.7	Generated Bill . . . . .	23
5.8	Registration . . . . .	23
5.9	Categories Information . . . . .	24
5.10	Details . . . . .	24
5.11	All Orders . . . . .	24
5.12	Add New Product . . . . .	24

5.13 Searching Products . . . . .	25
5.14 Show Orders . . . . .	25

## **List of Tables**

4.1	Cost-Benefit Analysis of E-commerce Project . . . . .	10
-----	---	----

## **1. Introduction**

The rise of digital technology has transformed various sectors, and e-commerce is one of the most significant developments. This project focuses on building an e-commerce platform using Laravel and Bootstrap 4 to provide users with a seamless online shopping experience. The system is designed to handle product management, user interaction, payment processing, and order tracking, ensuring efficient and scalable operations.

## **2. Problem Statement**

Traditional retail systems face issues with inventory management, limited accessibility, and a lack of online presence. The need for a robust and scalable online platform that can handle transactions, manage products, and maintain security is vital for modern businesses. This project aims to develop a responsive and secure e-commerce platform tailored for both sellers and buyers.

## **3. Objectives**

### **1. Develop a Scalable E-Commerce Platform**

Create a system that allows users to browse, purchase, and manage products efficiently.

### **2. Ensure Secure Payment Integration**

Implement secure payment methods for users to make transactions seamlessly.

### **3. Provide a Responsive User Interface**

Ensure the platform is accessible on various devices, including mobile phones, tablets, and desktops.

### **4. Develop an Admin Panel for Product and User Management**

Create an admin panel where administrators can manage product listings, monitor sales, process orders, and manage user accounts.

**Measurable:** Track the number of products managed, orders processed, and user interactions handled via the admin panel.

**Time-bound:** The admin panel will be fully implemented and functional within 4 months.

#### 5. **Ensure Accessibility and User-Friendliness for All Users**

Design the system to be user-friendly and accessible to users with varying levels of technical expertise, ensuring wide adoption.

**Measurable:** Gather feedback through user surveys and track platform adoption rates.

**Time-bound:** User testing and feedback collection will occur 2 months after deployment.

## 4. Methodology

### 4.1. Requirement identification

The requirement identification process for the e-commerce platform involves understanding the needs of both the business and the users. It starts with studying existing systems, followed by identifying functional and non-functional requirements, and evaluating the technical feasibility of the project. Once the requirements are clear, they are documented for the next phase of development.

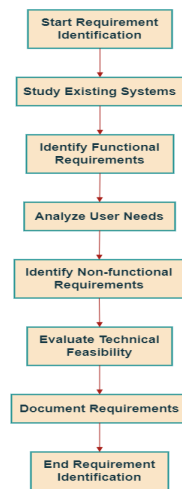


Figure 4.1: Requirement identification process(Flow Chart)

#### 4.1.1. Study of Existing System / Literature Review

In the e-commerce domain, various platforms like Shopify, WooCommerce, and Magento dominate the market, offering a wide range of functionalities such as product management, payment processing, and order tracking. [1] These platforms have proven efficient in scaling and handling large customer bases, but they also come with limitations. For instance, some platforms impose high transaction fees or lack customization options for small businesses. Additionally, open-source solutions like WooCommerce provide flexibility but require extensive technical expertise for setup and maintenance.

[2] Research into academic literature reveals a need for customizable, scalable, and secure e-commerce platforms tailored to the specific requirements of businesses. Studies highlight



the importance of features such as responsive design, [3] secure payment gateways, and efficient product management systems, which are crucial for modern e-commerce platforms to thrive in a competitive market. Therefore, the proposed e-commerce platform aims to address these challenges by offering a balance of flexibility, user-friendliness, and security.

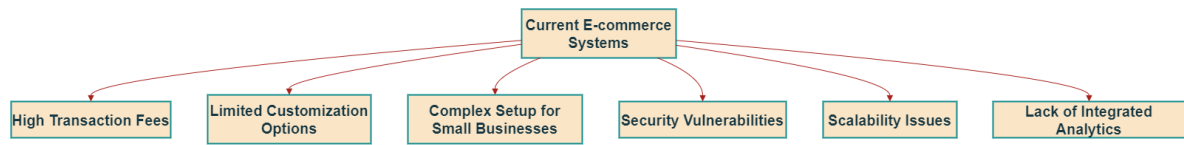


Figure 4.2: Current System Limitation

#### 4.1.2. Requirement Analysis

The requirement analysis phase focuses on identifying and defining the key technical, operational, and user needs for the e-commerce platform. This ensures that the platform meets business goals and user expectations while remaining technically feasible. The analysis is divided into three categories:

##### Technical Requirements

- **Scalability:** The platform must handle a growing number of users and transactions efficiently.
- **Security:** Secure payment gateways and data encryption are essential for protecting sensitive user information.

- **Responsive Design:** The platform should work seamlessly on mobile, tablet, and desktop devices.
- **Integration:** Ability to integrate third-party services, such as shipping and inventory management tools.

## **Operational Requirements**

- **User-Friendly Interface:** Both sellers and customers should find the platform easy to navigate, with minimal technical expertise required.
- **Admin Panel:** An admin interface should allow the business to manage products, orders, and user accounts, and provide real-time analytics.

## **User Requirements**

- **Product Management:** Sellers must be able to easily add, update, and delete products, including inventory tracking.
- **Secure Checkout:** Customers should have a seamless and secure payment experience, with multiple payment options.
- **Order Tracking:** Users need real-time updates on order status, including shipment tracking.

## **4.2. Feasibility Study**

The feasibility study evaluates the practicality of the E-commerce Platform project across technical, operational, and economic dimensions. This section also includes a cost-benefit analysis to determine the project's financial viability.

### **4.2.1. Technical**

Technical feasibility assesses whether the proposed technology and infrastructure can support the project's requirements for the e-commerce platform.

- **Technology Stack:** The project will use PHP, Laravel for the web framework, MySQL for the database, and JavaScript for front-end interactions. Payment gateways will be integrated via APIs like Stripe or PayPal.

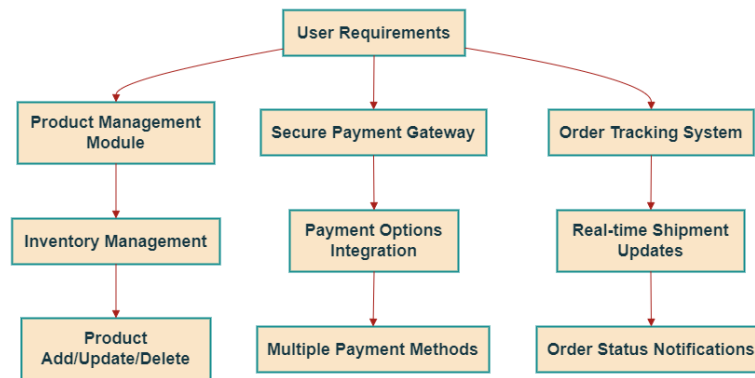


Figure 4.3: System Integration Workflow

- **System Integration:** Components like product management, payment processing, and user interface design are feasible with modern web technologies.
- **Infrastructure:** Cloud services like AWS or Azure will provide reliable hosting and ensure scalability.
- **Assumptions:**
  - Cloud hosting services are scalable and reliable.
  - APIs for payment processing are compatible with the platform.
  - Existing technology stack is sufficient for the platform’s requirements.

## Performance Metrics

### 1. Scalability

- Each user session requires 50MB of RAM.

- The platform will use 4 servers, each with 16GB of RAM. Server Capacity =  $(16,384 \text{ MB} - 50 \text{ MB}) \times 4 = 1,309 \text{ users}$  Server Capacity =  $(16,384 \text{ MB} - 50 \text{ MB}) \times 4 = 1,309 \text{ users}$

Conclusion: The system can handle approximately 1,309 concurrent users.

## 2. Data Storage Requirements

- Number of records: 200,000.
- Average size per record: 1KB. Database Size =  $200,000 \times 1 \text{ KB} = 200 \text{ MB}$

**Conclusion:** The database size of 200MB is manageable.

## 3. API Integration Costs

- Number of API calls per month: 100,000..
- Cost per API call: 0.01  
 $\text{Monthly API Cost} = 100,000 \times 0.01 = 1,000 \text{ USD}$

**Conclusion:** The monthly API cost of 1,000 is within the project budget.

**Conclusion:** The cost of \$500 per month is within the project budget and feasible for integration.

## 4. Technology and Tools

### 4.2.1.1 Database Size Calculation:

### 4.2.1.2 Available Tools and Technologies:

1. **Development Frameworks:** Laravel for PHP-based development.
2. **Database Management Systems:** MySQL or PostgreSQL for data management.
3. **Server Infrastructure:** Cloud hosting via AWS or Azure.
4. **APIs:** Stripe/PayPal for payments, and third-party logistics APIs for shipment tracking.

#### 4.2.1.3 Expertise Required:

1. **Development Skills:** Proficiency in PHP, Laravel, and JavaScript.
2. **Database Management:** SQL and database design.
3. **API Integration:** Payment and logistics API integration.
4. **Server Management:** Knowledge of cloud services like AWS or Azure.

#### 4.2.2. Operational Feasibility

Operational feasibility examines how easily the e-commerce platform can be used and maintained.

##### User Acceptance

Ensuring user-friendliness and ease of use is critical for success. Training sessions may be required for vendors and admins.

##### Assumption:

- Number of Users: 500 (initial vendors/customers).
- Training Sessions Required: 5.
- User Feedback Collection: 2 weeks.
- Cost per Feedback Collection Hour: 50.

##### Calculation:

Total Feedback Collection Cost = Feedback Sessions  $\times$  Duration of Each Session  $\times$  Cost per Hour

$$\text{Total Feedback Collection Cost} = 5 \times 20 \text{ hours} \times 50 = \$5,000$$

Conclusion: Feedback collection costs are estimated at 5,000, which will help ensure the platform meets user

##### Organizational Support

##### Assumption:

- Organizational Support Time Required: 4 weeks.
- Support Staff Hourly Rate: 100.

##### Calculation:

$$\text{Organizational Support Cost} = \text{Support Time} \times \text{Hourly Rate}$$

$$\text{Organizational Support Cost} = 4 \text{ weeks} \times 40 \text{ hours/week} \times 100 = \$16,000$$

Conclusion: Integration costs are estimated at 16,000

Effective planning will ensure seamless integration.

### **Compatibility with Existing Systems**

The system will need to integrate smoothly with existing order management, shipping, and inventory tools.

#### **Assumption:**

- Integration Time: 4 weeks.
- Integration Specialist Hourly Rate: 100.

#### **Calculation:**

$$\text{Integration Cost} = \text{Integration Time} \times \text{Hourly Rate}$$

$$\text{Integration Cost} = 4 \text{ weeks} \times 40 \text{ hours/week} \times 100 = \$16,000$$

Conclusion: The project is operationally feasible given adequate user training and support, along with smooth integration into existing systems.

### **4.2.3. Economic Feasibility**

Economic feasibility evaluates the cost-benefit of the e-commerce platform project. **Cost-Benefit Analysis**

### **Cost-Benefit Analysis**

Table 4.1: Cost-Benefit Analysis of E-commerce Project

<b>Item Description</b>	<b>Cost (\$)</b>	<b>Benefit (\$)</b>
Development Costs	250,000	-
Hardware Costs (Servers)	50,000	-
Training Costs	10,000	-
Maintenance Costs (Annual)	30,000	-
<b>Total Costs</b>	<b>340,000</b>	
Increased Efficiency		100,000 (Time savings)
Revenue Growth		150,000 (New user acquisition)
<b>Total Benefits</b>		<b>250,000</b>
<b>Annual Net Benefit</b>		<b>-90,000</b>
<b>Return on Investment (ROI)</b>		<b>-26.4%</b>

Summary:

- **Total Costs:** \$3,40,000 (Development + Maintenance)
- **Total Annual Benefits:** \$250,000 (Annual Savings + Increased Productivity)
- **Annual Net Benefit:** 90,000.**ROI:** – 26.4

#### 4.2.4. Schedule(Gantt chart showing the project timeline)

The Gantt chart is an essential tool in project management that visually represents the project's timeline, tasks, milestones, and their dependencies. In our e-commerce project, it outlines tasks such as "Requirement Analysis," "System Design," "Front-End Development," "Back-End Development," "Testing," and "Deployment."

Each task on the Gantt chart is depicted with its start and end points, and the progress of each task is tracked by shading the completed portions. This helps monitor which tasks are on track and which might be delayed. Significant milestones, such as "Completion of System Design" and "Platform Launch," are highlighted to emphasize key achievements in the project.

Dependencies between tasks are also illustrated. For example, "Testing" cannot begin until both "Front-End Development" and "Back-End Development" are completed, represented by arrows connecting these tasks. This ensures the project flows logically and allows for quick identification of any potential delays or issues.

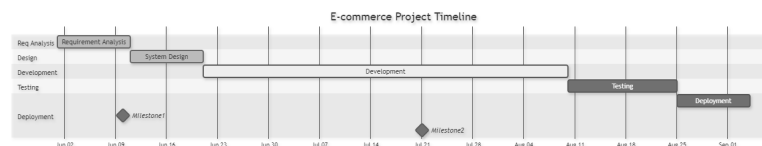


Figure 4.4: Gantt Chart demonstrating schedule feasibility

### **4.3. High-Level Design of System**

The e-commerce platform is designed with a modular and scalable architecture to efficiently manage various e-commerce functions. The system integrates multiple subsystems, each responsible for key features such as product management, inventory tracking, order processing, payment integration, and customer account management.

At the user interface (UI) layer, the system provides an intuitive and responsive web-based interface that is accessible to both customers and administrators. Customers can log in to browse products, manage their shopping cart, place orders, and view order history. Administrators have additional privileges to manage products, track inventory, process orders, and view sales reports. The interface also supports real-time product updates and inventory status, helping businesses keep track of stock levels.

The application logic layer, built on a Model-View-Controller (MVC) framework using PHP, serves as the system's core. This layer handles user requests, interactions with the database, and external API communications. Each major feature, such as product management, order processing, and customer accounts, is managed by a dedicated module. Additionally, this layer integrates with payment gateways to securely process transactions.

The database layer consists of relational databases that store and manage data related to products, orders, customers, and payments. Separate tables handle different types of information, ensuring data integrity and supporting efficient queries for real-time operations, reporting, and analytics.

To enhance functionality, the system incorporates external API integration for payment processing and shipping services. The architecture is designed to allow easy integration of future APIs, such as third-party marketing tools or product recommendation engines. Security is a key consideration, with robust authentication mechanisms ensuring that only authorized users (customers, administrators) can access sensitive information. Security protocols like SSL encryption and secure password hashing protect customer data and transactions.

An admin panel provides administrators with comprehensive control over the platform, including the ability to manage products, view order status, process refunds, and generate sales reports. The system is built for high availability and scalability, ensuring minimal downtime as the business grows. This architecture allows new features and modules to be added without disrupting current operations, making it adaptable to the evolving needs of the business.

#### **4.3.1. Methodology of the proposed system**

The development of the e-commerce platform will employ a hybrid methodology combining both Structured and Object-Oriented approaches. This dual methodology ensures a



comprehensive, scalable, and user-friendly solution for effective online retail management. Below is a detailed description of the techniques and approaches that will be used:

#### **4.3.1.1 Structured Approach**

##### **Requirements Analysis**

**Objective:** To gather and analyze the specific needs of the e-commerce platform, focusing on areas such as product management, inventory tracking, order processing, payment integration, and customer account management.

**Techniques:** Surveys, interviews, and analysis of existing e-commerce platforms to identify functional and non-functional requirements.

##### **System Design**

**Objective:** To create a detailed blueprint of the system that outlines its architecture, components, and interactions.

**Techniques:** Design specifications will be developed using structured design methods, including data flow diagrams (DFDs) and entity-relationship diagrams (ERDs) to clearly define data handling and system processes.

##### **Implementation**

**Objective:** To translate design specifications into a functional system.

**Techniques:** Coding will follow structured programming practices to ensure modularity and clarity. The system will be developed in phases, with each module (product management, orders, payments) being integrated and tested independently before full system integration.

##### **Testing**

**Objective:** To ensure that the system functions correctly and meets the specified requirements.

**Techniques:** Various testing methods, including unit testing, integration testing, and system testing, will be employed to identify and resolve any issues.

##### **Deployment**

**Objective:** To make the system available for use and ensure it operates in a real-world environment.

**Techniques:** The deployment phase includes user training, documentation, and support to ensure smooth adoption and operation.

#### **4.3.1.2 Object-Oriented Approach**

##### **Analysis**

**Objective:** To model the system using object-oriented techniques, focusing on defining the system's objects and their interactions.

**Techniques:** Use case diagrams, class diagrams, and sequence diagrams will be created to represent the system's functions and their relationships.

##### **Design**

**Objective:** To create detailed design models that represent the system's objects, their attributes, and their relationships.

**Techniques:** Object-oriented design principles, such as encapsulation, inheritance, and polymorphism, will be applied. Class diagrams and interaction diagrams will guide the development of system components.

##### **Implementation**

**Objective:** To develop the system using object-oriented programming (OOP) principles

**Techniques:** The code will be structured into classes and objects, promoting reusability and maintainability. Object-oriented programming languages and frameworks will be used to build the system.

##### **Testing and Refinement**

**Objective:** To ensure that each object and its interactions work correctly and contribute to the overall system functionality.

**Techniques:** Object-oriented testing methods, including class testing and integration testing of objects, will be employed. Continuous refinement will be carried out based on testing feedback.

##### **Conclusion**

The proposed system development methodology leverages both Structured and Object-Oriented approaches to ensure a robust, scalable, and user-friendly e-commerce platform. By integrating these methodologies, the system will effectively address the diverse needs of online retailers and improve overall operational efficiency. This hybrid approach ensures that the system is well-structured, modular, and adaptable to future changes or enhancements.

#### 4.4. Flow Charts/Working Mechanism of Proposed System

##### Data Flow Diagram

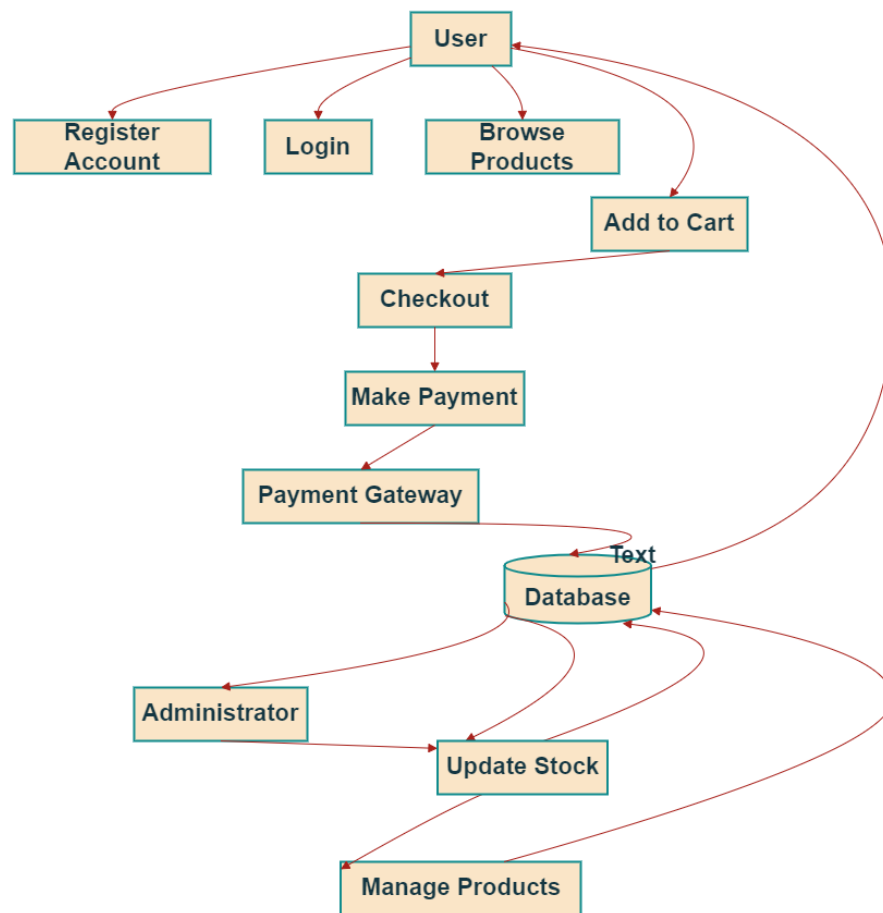


Figure 4.5: Data Flow Diagram for Proposed System

##### ER Diagram

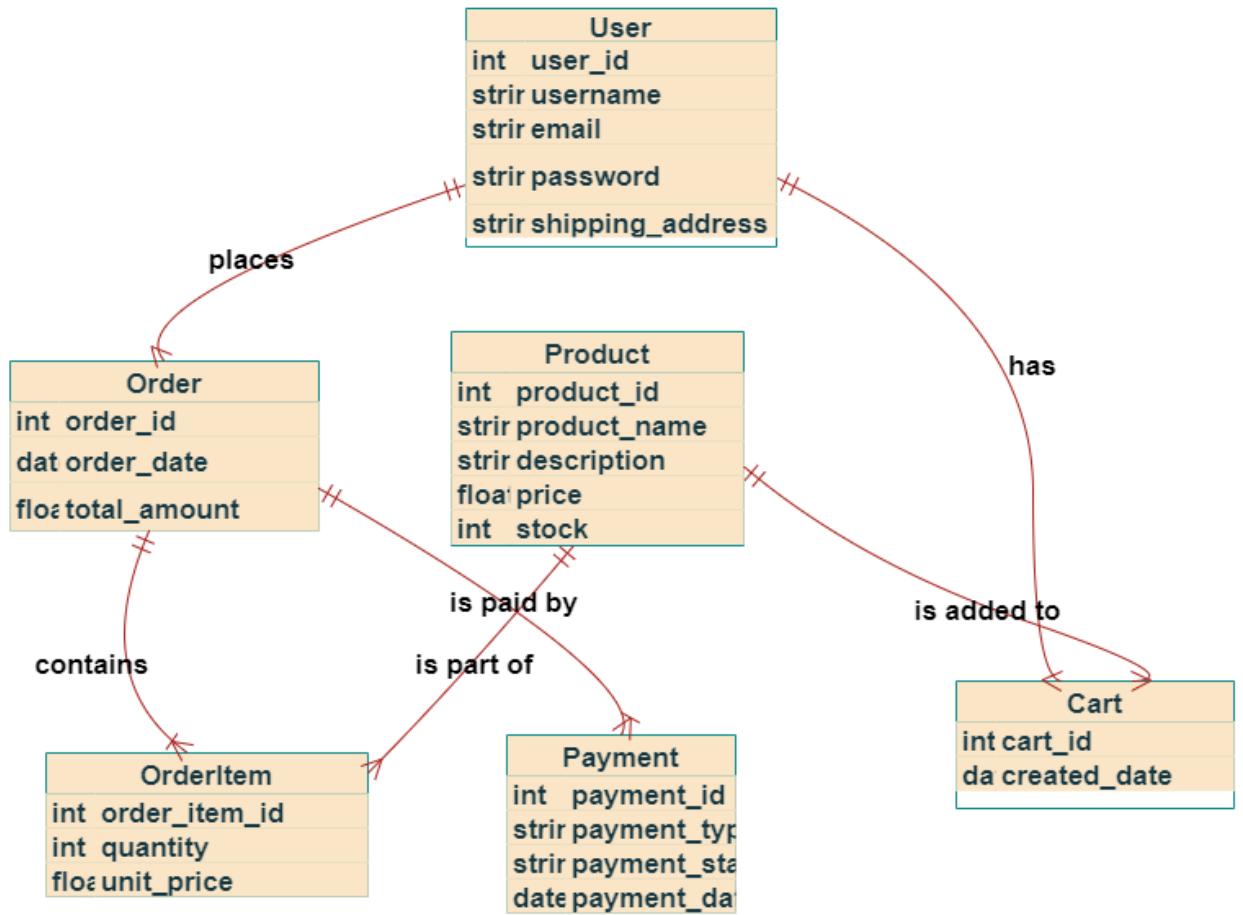


Figure 4.6: Entity Relationship Diagram for Proposed System

## Use Case Diagram

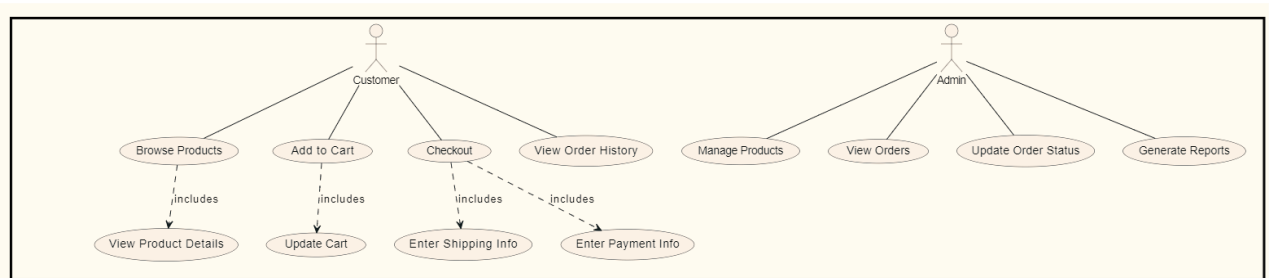


Figure 4.7: Use Case Diagram for Proposed System

## Class Diagram

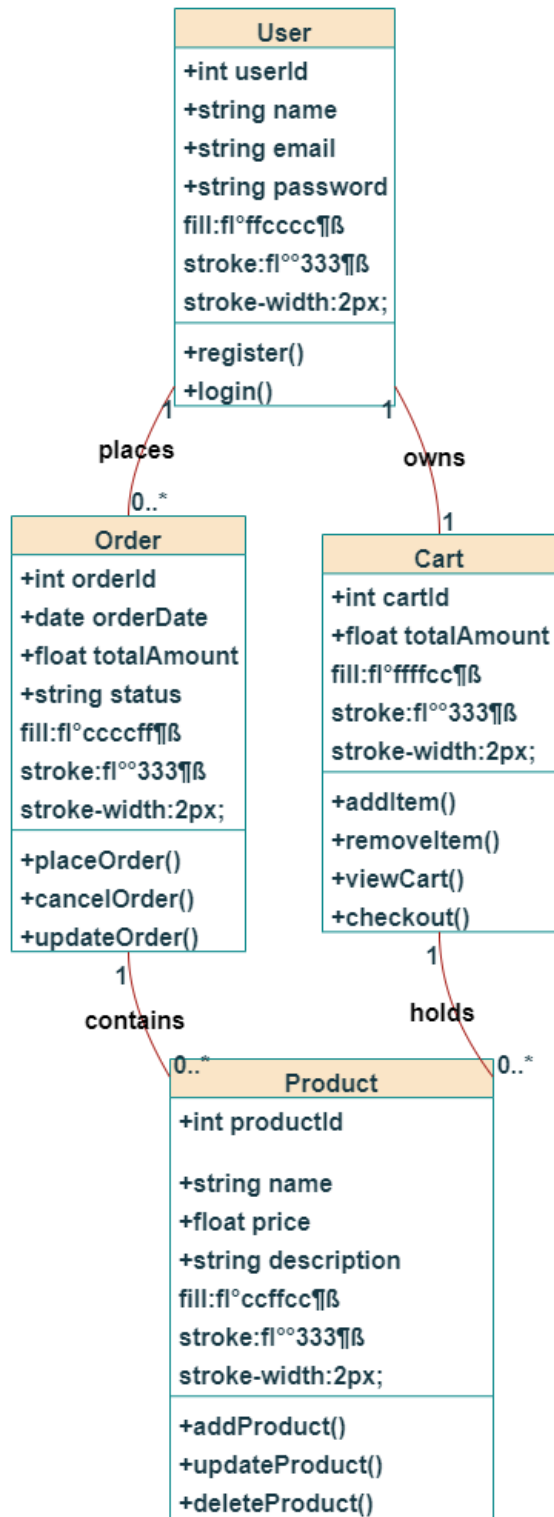


Figure 4.8: Class Diagram for Proposed System

## Sequence Diagram

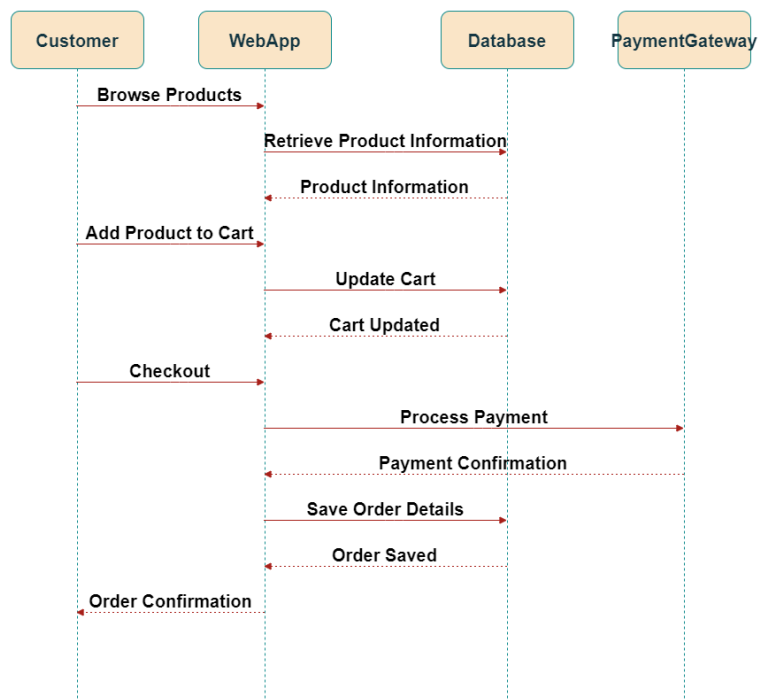


Figure 4.9: Sequence Diagram for Proposed System

## 5. Expected Output

1. **Efficient E-Commerce Operations:** The system will streamline critical e-commerce functions such as product management, order processing, inventory control, and customer relationship management (CRM). This will reduce manual effort and enhance overall operational efficiency.
2. **Data-Driven Insights:** Real-time data analytics, including sales trends, customer behavior, and inventory levels, will enable informed decision-making. This will lead to optimized marketing strategies, inventory management, and improved sales performance.
3. **Proactive Problem Resolution:** Predictive algorithms and data analytics will help identify potential issues such as inventory shortages, customer dissatisfaction, or fraudulent transactions before they escalate. This will minimize disruptions and enhance the customer experience.
4. **Improved Productivity:** Automation of routine tasks, such as order fulfillment and customer support, will boost productivity. The system will also provide tools for efficient resource management, leading to reduced operational costs and improved profitability.
5. **User-Friendly System:** The platform will feature an intuitive and responsive interface for both administrators and customers. This will make it easy to manage products, process orders, and access customer insights, ensuring a seamless user experience.

### Deliverables:

- **A Fully Functional E-Commerce Platform:** Integration of essential features such as product catalog management, shopping cart, order processing, payment gateways, and customer account management.
- **Detailed Documentation:** Comprehensive documentation covering system usage, configuration, and maintenance. This will also include guidelines for future scalability and enhancements.
- **A Web-Based Interface:** An accessible and user-friendly web interface that allows customers to browse products, make purchases, track orders, and manage their accounts effortlessly. Administrators will have a backend interface for managing inventory, processing orders, and analyzing data.

The project will deliver a powerful e-commerce platform that enhances the online shopping experience and streamlines business operations. It will improve decision-making and profitability while setting the stage for future e-commerce innovations.



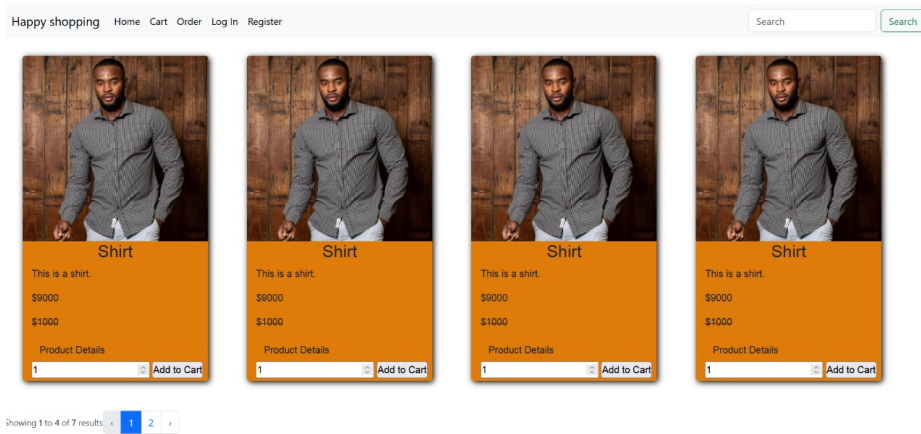


Figure 5.1: Interface of Happy Shopping

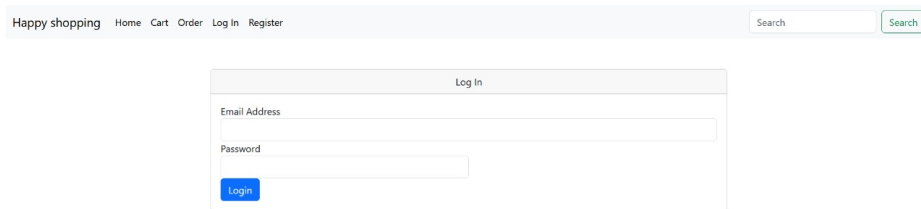


Figure 5.2: Login

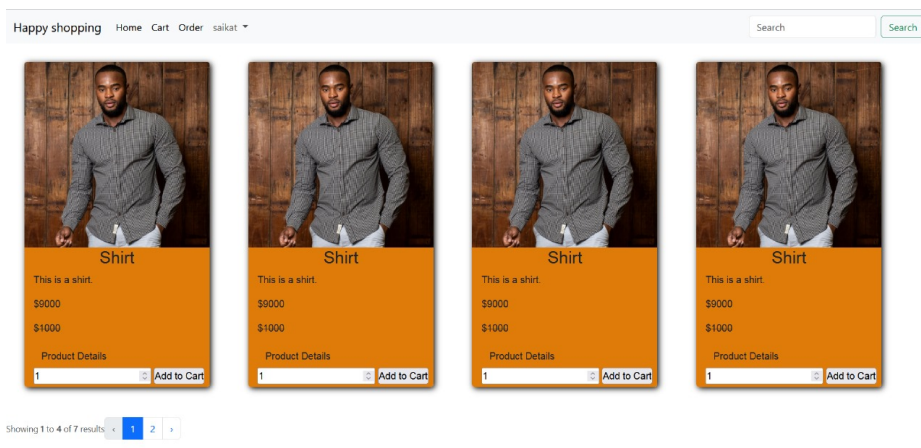


Figure 5.3: Login As A User

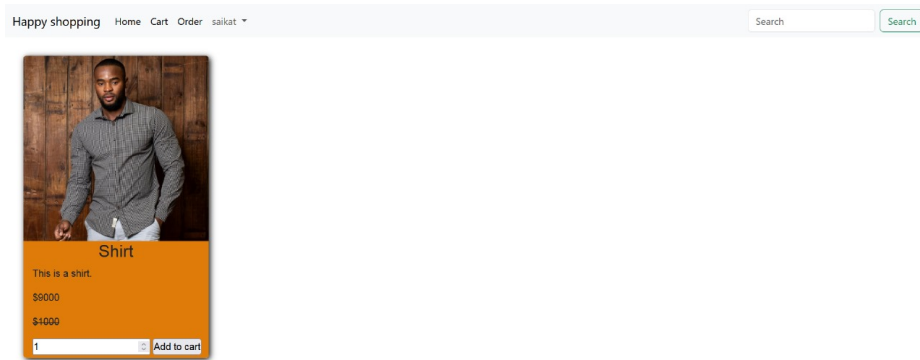


Figure 5.4: Add To Cart

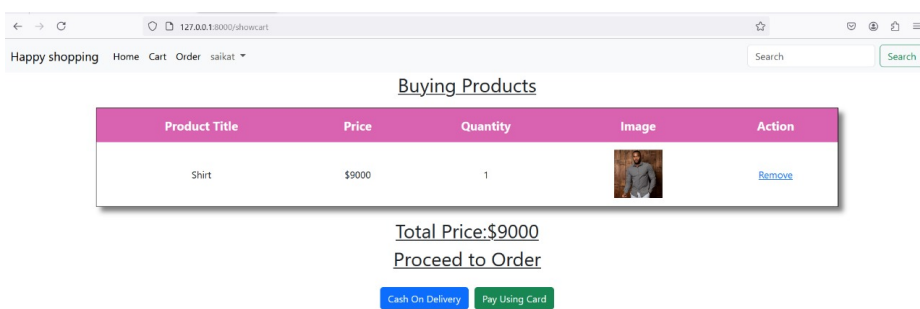


Figure 5.5: Show Cart

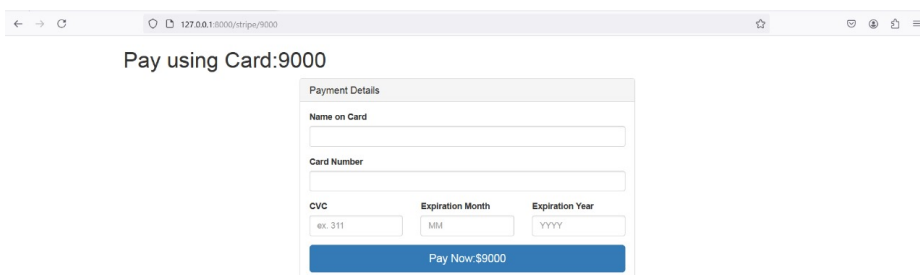


Figure 5.6: Payment method

INVOICE / BILL

CUSTOMER DETAILS

Name: Taif

Email: taif@gmail.com

Phone: 37397497

Address: chattogram

INVOICE INFO

Product Title	Quantity	Price
Shirt	1	9000

Payment Status:paid

Transaction ID:  
ch\_3PzEs906cFxbqTle1774fXE1

Total Amount Paid: 9000

Figure 5.7: Generated Bill

Happy shopping

Home

Cart

Order

Log In

Register

Search

Search

Register

Name

Email Address

Phone Number

Address

Password

Confirm Password

Register

Figure 5.8: Registration

Happy Shopping

BlogProductsAccount

Categories Information

Add Category

ID	Category Name	Action		
1	Shirt	<a href="#">View</a>	<a href="#">Update</a>	<a href="#">Delete</a>
2	Mobile	<a href="#">View</a>	<a href="#">Update</a>	<a href="#">Delete</a>

Figure 5.9: Categories Information

Happy Shopping

BlogProductsAccount

Details

Total Products:  
7

Total Orders:  
19

Total Customers:  
5

Total Revenue:  
216000

Order Delivered:  
3

Order Processing:  
16

Figure 5.10: Details

Happy Shopping

Blog

Products

Account

☰

All orders

search





Name	Email	phone	address	Product Title	Price	Quantity	Image	Payment Status	Delivery Status	Delivered	Bill
saikat	saikat@gmail.com	1567893784	Bandraban	Shirt	9000	1		paid	processing	<a href="#">Delivered</a>	<a href="#">Download Bill</a>
saikat	saikat@gmail.com	1567893784	Bandraban	Shirt	18000	2		paid	delivered	<a href="#">Delivered</a>	<a href="#">Download Bill</a>
saikat	saikat@gmail.com	1567893784	Bandraban	Shirt	9000	1		paid	delivered	<a href="#">Delivered</a>	<a href="#">Download Bill</a>
saikat	saikat@gmail.com	1567893784	Bandraban	Shirt	9000	1		paid	processing	<a href="#">Delivered</a>	<a href="#">Download Bill</a>

Figure 5.11: All Orders

Happy Shopping		Blog Products Account			
<div> <div>ADD NEW PRODUCT</div> <div> <div>Product Title</div> <div>Product Description</div> <div>Product Price</div> <div>Discount Price</div> <div>Product Quantity</div> <div>Product Catagroy</div> <div>Product Image</div> <div>Save</div> </div> </div>					

Figure 5.12: Add New Product

Happy Shopping
Blog
Products
Account

Category

Category

Save

Figure 5.13: Searching Products

Happy shopping
Home
Cart
Order
saikat

127.0.0.1:8000/showorder

Search

Search

### Buying Products






Product Title	Price	Quantity	Payment Status	Delivery Status	Image	Cancel Order	Bill
Shirt	\$9000	1	paid	processing		Not Allowed	<a href="#">Download Bill</a>
Shirt	\$18000	2	paid	delivered		Not Allowed	<a href="#">Download Bill</a>
Shirt	\$9000	1	paid	delivered		Not Allowed	<a href="#">Download Bill</a>
Shirt	\$9000	1	paid	processing		Not Allowed	<a href="#">Download Bill</a>
Shirt	\$9000	1	paid	processing		Not Allowed	<a href="#">Download Bill</a>

Figure 5.14: Show Orders

## References

- [1] P. K. Kannan and H. K. T. Yi, “E-commerce: Business, technology, society,” *IEEE Transactions on Engineering Management*, vol. 50, no. 3, pp. 256–265, Aug. 2003.
- [2] L. H. Kuo, Y. T. Chang, and C. C. Ko, “E-commerce security: A survey,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 8–16, Jan. 2006.
- [3] W. L. Berry and P. A. Johnson, “A framework for e-commerce application development,” *IEEE Transactions on Software Engineering*, vol. 27, no. 4, pp. 305–319, Apr. 2001.