**A**

**PROJECT REPORT**

**ON**

**DentPlus Dental Clinic**

**IN THE FULFILLMENT OF MASTER’S DEGREE**

**IN COMPUTER APPLICATIONS OF**

**SAVITRIBAI PHULE PUNE UNIVERSITY**

**SUBMITTED BY**

**Praful Makeshwar<>**

**GUIDED BY**

**Dr. Vandana Bagal**

****

**MASTER OF COMPUTER APPLICATION**

(UNDER FACULTY OF ENGINEERING)

**DEPARTMENT OF MCA**

**K.K. WAGH INSTITUTE OF ENGINEERING**

**EDUCATION & RESEARCH, NASHIK 422003**

(An Autonomous institute from A.Y. 2022-23)

**SAVITRIBAI PHULE PUNE UNIVERSITY**

**YEAR 2023-2024**

DISSERTATION APPROVAL SHEET

A Project titled

**DentPlus Dental Clinic**

Has been successfully completed by

**Praful Makeshwar<>**

At Department of MCA,

K.K. Wagh Institute of Engineering Education & Research, Nashik 422 003,

(An Autonomous Institute from A.Y. 2022-23)

Savitribai Phule Pune University

(Year 2022-2023)

Dr. Vandana Bagal Dr. V. C. Bagal

Guide I/c Head of Department

**Master of Computer Applications**

(Under Faculty of Engineering)

**Department of MCA**

**Internship Letter**

**Praful Makeshwar**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Title** | **Page No.** |
|  | Abstract |  |
| 1 | Introduction | 1-9 |
|  | * 1. Detailed Problem Definition   2. Current Market Survey   3. What Is Event Management System   4. Scope of project   5. Existing System   6. Feasibility Study |  |
| 2 | Requirement Analysis | 10-18 |
|  | 2.1 Project Overview  2.2 Functional Requirement  2.3 Non-Functional Requirement  2.4 Technical Requirements  2.5 Constraints  2.6 Assumptions  2.7 Risks  2.8 user Documentation Plan |  |
| 3. | Literature Survey | 19-25 |
| 4. | Methodology | 26-30 |
|  | 4.1 Software Utilized  4.2 Hardware Specifications  4.3 Programming Language  4.4 Development Platform  4.5 Project Component  4.6 Tools and techonologies  4.7 Project Work Flow |  |
| 5 | Modelling | 31-72 |
|  | 5.1 Use case diagram  5.2 Modules of StreamlineEvents Systems  5.3 Class Diagrams  5.4 ER Diagram  5.5 Context Diagram  5.6 Sequence Diagram for Administrator's Event Section Setting  5.7 Activity Diagram for Administrator's Event Section  5.8 Database Uses  5.9 ScreenShots |  |
| 6 | Results | 73-74 |
| 7 | Testing | 75-78 |
|  | 7.1 Test Cases |  |
| 8 | Conclusion | 79-80 |
| 9 | References | 81-82 |
| 10 | Acknowledgements | 83 |
| **Sr. No** | **Title** | **Page No.** |
|  | Abstract |  |
| 1 | Introduction | 1-9 |
|  | * 1. Detailed Problem Definition   2. Current Market Survey   3. What Is Event Management System   4. Scope of project   5. Existing System   6. Feasibility Study |  |
| 2 | Requirement Analysis | 10-18 |
|  | 2.1 Project Overview  2.2 Functional Requirement  2.3 Non-Functional Requirement  2.4 Technical Requirements  2.5 Constraints  2.6 Assumptions  2.7 Risks  2.8 user Documentation Plan |  |

**List of figures**

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **Title** | **Page no** |
| 1 | **System Architecture** | 16 |
| 2 | **Zero Level Data Flow Diagram** | 16 |
| 3 | **ER-Diagram** | 17 |
| 4 | **Use Case Diagram** | 18 |
| 5 | **Class Diagram** | 19 |
| 6 | **Module Hierarchy** | 20 |
| 7 | **Activity Diagram** | 21 |
| 8 | **Deployment Diagram** | 22 |

**List of tables**

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **Title** | **Page no** |
| 1 | **User Table** | 23 |
| 2 | **Admin Table** | 23 |
| 3 | **Product Table** | 24 |
| 4 | **Cart Table** | 24 |
| 5 | **Order Table** | 25 |
| 6 | **Invoice Table** | 25 |

**INTRODUCTION**

The dental clinic website project is designed to revolutionize the management of patient care and administrative processes within a dental clinic. Utilizing a comprehensive Entity-Relationship (ER) diagram as its foundation, this project outlines a robust system that seamlessly integrates patient information, enquiries, appointments, and service offerings. The aim is to create an efficient, user-friendly platform that enhances both patient experience and clinic operations.

At the heart of this system is the Admin entity, which provides centralized control over various operational aspects. Admins, equipped with attributes such as UserID, Email, Mobile, and Password, manage patient records, handle enquiries, schedule appointments, and oversee service details. This centralized management ensures streamlined operations and effective oversight.

The Patients entity is crucial for maintaining detailed patient profiles, including PatientID, Name, Email, Mobile, and Password. By consolidating patient information in one place, the system ensures that data is easily accessible and up-to-date, facilitating better patient care and communication.

Handling patient enquiries efficiently is managed through the Enquiries entity, which includes attributes like Enquiry ID, PatientID, Name, Email, Mobile, and Enquiry Time. This enables the clinic to track and respond to enquiries promptly, providing personalized and timely responses that enhance patient satisfaction.

The Appointments entity is integral to the scheduling process, with attributes such as AppointmentID, PatientID, Contact, Appointment Date, Gender, Appointment Time, and Status. This entity helps in organizing and managing appointments, ensuring that scheduling is error-free and patient flow is optimized.

Service offerings are managed through the Services entity, which details each service provided by the clinic. Attributes such as ServiceID, Service Name, Service Sub Title, and Description allow patients to easily understand and choose from the range of available treatments.

The interconnected relationships among these entities ensure that all components of the clinic's operations work in harmony. This integration supports seamless workflows, enhances data integrity, and promotes efficient management practices.

In summary, the dental clinic website project, underpinned by a meticulously designed ER diagram, aims to create a cohesive and efficient system for managing patient care and clinic operations. This project is poised to deliver significant improvements in service delivery, patient satisfaction, and operational efficiency, setting a new standard for dental clinic management.

1

**ABSTRACT**

The dental clinic website is a sophisticated platform designed to optimize the management of patient information, enquiries, appointments, and service offerings. Central to this system is a well-structured Entity-Relationship (ER) diagram that delineates key entities, their attributes, and interrelationships, ensuring seamless integration and data integrity.

The Admin entity oversees the entire clinic operation, managing patient records, enquiries, appointments, and services. Admins use attributes such as UserID, Email, Mobile, and Password to maintain secure and efficient control over all activities.

The Patients entity stores comprehensive patient profiles, including PatientID, Name, Email, Mobile, and Password. This ensures that all patient information is readily accessible and up-to-date, facilitating improved patient care and communication.

The Enquiries entity captures and tracks patient enquiries, linking them to specific patients via attributes like Enquiry ID, PatientID, Name, Email, Mobile, and Enquiry Time. This allows for prompt and personalized responses, enhancing patient satisfaction.

The Appointments entity streamlines appointment scheduling and management, with attributes such as AppointmentID, PatientID, Contact, Appointment Date, Gender, Appointment Time, and Status. This ensures organized scheduling and reduces the risk of errors, supporting efficient patient flow and clinic operations.

The Services entity details the clinic’s offerings, including ServiceID, Service Name, Service Sub Title, and Description. This helps patients understand available treatments and make informed decisions about their care.

The interrelationships among these entities ensure that all aspects of the clinic's operations are interconnected, promoting seamless workflows and high-quality patient care. This integrated system design supports the clinic's goal of delivering exceptional dental services with efficiency and precision.

2

**OBJECTIVES**

 **Enhance Administrative Efficiency**:

* Develop a centralized system for Admins to manage all aspects of clinic operations, including patient records, enquiries, appointments, and services.
* Utilize attributes such as UserID, Email, Mobile, and Password to ensure secure and efficient administration.

 **Streamline Patient Information Management**:

* Create comprehensive patient profiles within the **Patients** entity, encompassing attributes like PatientID, Name, Email, Mobile, and Password.
* Ensure that patient data is easily accessible and up-to-date to facilitate better patient care and communication.

 **Improve Enquiry Handling**:

* Implement a robust **Enquiries** entity to capture and track patient enquiries, linking them to specific patients via Enquiry ID, PatientID, Name, Email, Mobile, and Enquiry Time.
* Enable prompt and personalized responses to patient enquiries, enhancing overall patient satisfaction.

 **Optimize Appointment Scheduling**:

* Design an efficient **Appointments** entity to manage the scheduling process, with attributes such as AppointmentID, PatientID, Contact, Appointment Date, Gender, Appointment Time, and Status.
* Ensure organized scheduling, minimize the risk of double bookings or missed appointments, and support smooth patient flow.

 **Detail Service Offerings**:

* Develop a comprehensive **Services** entity to manage information about the clinic's offerings, including ServiceID, Service Name, Service Sub Title, and Description.
* Provide patients with clear and detailed information about available treatments to aid in informed decision-making.

 **Ensure Data Integrity and Security**:

* Maintain high standards of data integrity across all entities by ensuring accurate, consistent, and secure handling of patient and clinic information.
* Implement robust security measures to protect sensitive data, adhering to healthcare regulations and best practices.

 **Facilitate Seamless Integration and Interconnectivity**:

* Ensure that all entities are interrelated and integrated, promoting seamless workflows and efficient management practices.
* Foster an environment where data flows smoothly between entities, enhancing overall operational efficiency and patient care quality.

 **Enhance Overall Patient Experience**:

* Utilize the system to improve the overall patient experience by providing efficient, personalized, and responsive services.
* Ensure that patients have access to timely information and support, resulting in higher satisfaction and better healthcare outcomes.

3

|  |  |
| --- | --- |
| **Hardware Requirement** | **Specification** |
| **Processor** | Intel Core i5 or Higher Processor 1.80GHz |
| **Processor Speed** | 1 gigahertz (GHz) or faster processor |
| **Hard Disk** | 16 GB for 32-bit OS, 32 GB for 64-bit OS |
| **Main Memory (RAM)** | 1 GB for 32-bit OS, 2 GB for 64-bit OS |

**HARDWARE AND SOFTWARE REQUIREMENTS:**

**Hardware Requirements:**

**Software Requirements:**

|  |  |
| --- | --- |
| **Software Requirement** | **Specification** |
| **Language** | Dot Net APIs |
| **Server-Side Script** | HTML, CSS, JavaScript |
| **Deployment Environment** | Visual Studio Code |
| **Operating System** | Windows 11 |
| **Database** | SQL Server |

4

**Literature Survey**

**Introduction to E-commerce Platforms**

E-commerce platforms have revolutionized the way businesses operate, providing a digital marketplace where transactions can occur seamlessly. These platforms offer a wide array of functionalities including product listings, shopping cart capabilities, secure payment gateways, and customer feedback systems. The success of e-commerce platforms hinges on their ability to offer a user-friendly experience while ensuring robust security and scalability.

**Evolution of E-commerce**

The evolution of e-commerce has been marked by significant technological advancements. Early e-commerce systems were limited in functionality and often faced challenges related to security and scalability. However, with the advent of more sophisticated web development technologies, e-commerce platforms have become more robust and versatile. Key milestones in this evolution include the introduction of secure socket layer (SSL) encryption for secure transactions, the development of advanced database management systems, and the integration of real-time analytics for data-driven decision-making.

**Case Studies of Successful E-commerce Platforms**

Several successful e-commerce platforms provide valuable insights into best practices and strategies for achieving market success. For instance, Amazon's focus on customer satisfaction, extensive product range, and efficient logistics network has set a benchmark in the industry. Similarly, Alibaba's innovative use of big data analytics and artificial intelligence has enabled it to offer personalized shopping experiences to its users.

**Technological Frameworks and Tools**

The development of e-commerce platforms relies heavily on a robust technological framework. Commonly used tools and frameworks include ASP.NET for web application development, SQL Server for database management, and Bootstrap for responsive web design. These technologies enable developers to create scalable, secure, and user-friendly platforms that can handle high traffic volumes and complex transactions.

**Challenges in E-commerce**

Despite the advantages, e-commerce platforms face several challenges. Security concerns, such as data breaches and fraudulent activities, are prevalent issues that need continuous monitoring and improvement. Additionally, the rapid pace of technological change necessitates constant updates and iterations to keep the platform relevant and competitive. Scalability is another critical challenge, particularly for platforms experiencing rapid growth in user base and product listings.

**Consumer Behavior and Trends**

Understanding consumer behavior is crucial for the success of any e-commerce platform. Recent trends indicate a growing preference for mobile shopping, personalized recommendations, and seamless checkout experiences. Platforms that leverage artificial intelligence to analyze consumer behavior and tailor their offerings accordingly tend to perform better in terms of user engagement and sales conversions.

5

**Conclusion**

The literature on e-commerce platforms underscores the importance of technological innovation, user experience, and security in building successful online marketplaces. Pie-Cloth, by leveraging these insights and employing advanced web development technologies, aims to create a robust platform that caters to the needs of health-conscious consumers. By addressing the challenges and aligning with current consumer trends, Pie-Cloth is well-positioned to achieve significant growth and customer satisfaction in the competitive e-commerce landscape.

6

**Methodology**

**Waterfall**

The waterfall method, first designed by Winston W. Royce in 1970 for software development, is considered a more traditional, linear approach to managing a project. With the waterfall methodology, a project flows through a series of steps or phases. Generally, each must be completed before the next can begin.

**Stages of the waterfall model**

1. Requirements: In this first phase, you will work with stakeholders in order to clearly define the project's requirements.

2. Design: The critical design phase is when you will plan what the final product will look like and what steps your team needs to take to get there.

3. Implementation: This is where you can picture all your planning into action. For software projects, this is when programmers will write the actual code.

4. Verification: During verification, your team tests the product to ensure it meets the requirements laid out in the first phase.

5. Maintenance: After the project is complete, the development team responds to feedback and makes any necessary modifications.

7

**FEASIBILITY STUDY**

**1. TECHNICAL FEASIBILITY:**

Website Development: The technical infrastructure required for developing the e-commerce website, including front-end and back-end development, is feasible with existing technologies and frameworks such as remix.js, css and backend languages like Node.js

Database Management: Implementing a non-relational database management system (e.g.MongoDB) for storing user data, product information, and order details is technically feasible.

Payment Gateway Integration: Integration of secure payment gateways like Razorpay for online transactions is technically viable, leveraging APIs provided by payment service providers..

Scalability: The platform should be designed to scale efficiently to accommodate increasing user traffic and product inventory. Utilizing cloud services such as AWS or Azure for scalable hosting is technically feasible.

**2. OPERATIONAL FEASIBILITY:**

User Experience: Ensuring a user-friendly interface and intuitive navigation will enhance operational feasibility by reducing user learning curves and facilitating smooth interactions.

Inventory Management: Operational feasibility involves efficient inventory management processes to ensure accurate product availability, timely restocking, and order fulfillment.

Review Mechanism: Implementing a feedback mechanism allows for continuous improvement based on user input, enhancing operational effectiveness and user satisfaction.

**3. ECONOMIC FEASIBILITY:**

Development Costs: The initial investment in website development, including design, coding, and testing, should be economically feasible within the allocated budget.

Operating Costs: Ongoing operational expenses such as hosting fees, payment gateway charges, maintenance costs, and staffing for customer support should be economically viable.

Revenue Generation: The platform's revenue model, which may include product sales margins, transaction fees, or premium membership subscriptions, should be economically sustainable to ensure profitability.

8

**NEED OF SYSTEM**

The need for the Pie-Cloths e-commerce platform arises from the increasing consumer demand for healthier food options and the convenience of online shopping. In today's fast-paced world, many consumers are seeking nutritious and wholesome products that can easily be integrated into their busy lifestyles. Traditional brick-and-mortar stores often do not provide the variety and specialized products that health-conscious consumers are looking for, making it difficult for them to find the right products in one place.

Furthermore, the COVID-19 pandemic has accelerated the shift towards online shopping, with consumers increasingly preferring to shop from the safety and comfort of their homes. This shift has highlighted the need for a reliable and user-friendly online platform that can cater to the specific dietary needs and preferences of health-conscious consumers.

98

**ANALYSIS AND DESIGN**

**System Requirements (Functional and Non-functional)**

This section provides requirement overview of the system. Various functional modules that can implemented by the system will be -

**Functional Requirements:**

**1. Registration –**

If customer wants to order the product/item then he/she must be register. Unregister customer cannot buy the product.

**2. Login –**

Customer logins to the system by entering valid user id and password for order the products online.

**3. Changes to Cart -**

Changes to cart means the customer after login or registration can make order or cancel order of the item from the cart.

**4. Logout-**

After the payment or searching, the product the customer will logged out.

10

**NON-FUNCTIONAL REQUIREMENTS:**

**1. Security -**

The system use SSI. (Secured socket layer) in all transactions that include any confidential customer information. The system must automatically log out all customers after a period of inactivity.

The system should not leave any cookies on the customer's computer containing the user's password. The system's back-end servers shall only be accessible to authenticated

Administrators.

Sensitive data will be encrypted before sent over insecure connections like the internet

**2. Reliability -**

The system provides storage of all databases on redundant computers with automatic

Switch over. The reliability of the overall program depends on the reliability of the separate components.

The system has a backup of the database, which is continuously maintained and updated to reflect the most recent changes.

Thus, the overall stability of the system depends on the stability of container and its underlying operating system.

**3. Availability -**

The system should be available at all times, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs. In case of a hardware failure or database corruption, a replacement page will show. Also, in case of a hardware failure or database corruption, backups of the database should be retrieved from the server and saved by the administrator. Then the service will be restarted. It means 24 X 7 availability.

4. **Maintainability-**

A commercial database used for maintaining the database and the application server takes care of the site. In case of a failure, a re-initialization of the program will be done. In addition, the software design done with modularity in mind so that maintainability done efficiently.

11

**5. Portability:**

The application is HTML. And scripting language based. Therefore, the end-user part is fully portable and any system using any web browser should be able to use the features of the system, including any hardware platform that is available or will be available in the future. An end-user is using this system on any OS, it is either Windows or Linux. The system shall run on PC, Laptops, and PDA etc.

12

**MODULES DESCSRIPTION**

**User Management Module:**

This module handles user registration, login, and profile management functionalities. Users can create accounts, update their profiles, and reset passwords if needed. It also includes role-based access control to distinguish between regular users and administrators.

**Product Management Module:**

Product management allows administrators to add, edit, and delete products from the platform. Each product has attributes such as name, description, price, and image. Products are categorized based on fruit types for easy navigation.

**Shopping Cart Module:**

The shopping cart module enables users to add desired products to their cart while browsing. Users can view their cart contents, update quantities, and remove items as needed. It calculates the total order amount and facilitates seamless checkout.

**Checkout and Payment Module:**

This module facilitates secure online payments through integration with payment gateways like Razorpay. Users can choose their preferred payment method and complete transactions securely. It also supports cash on delivery (COD) for users who prefer offline payments.

**Order Management Module:**

The order management module handles order processing, tracking, and invoicing. Administrators can view all orders, update their status (e.g., processing, shipped), and generate invoices. Users can track their orders in real-time and download invoices for reference.

**Review Module:**

Users can provide feedback on products or the platform's performance through this module It includes a feedback form where users can submit comments, suggestions, or complaints. Additionally, users can make inquiries or contact support for assistance with orders or general queries.

**Admin Panel Module:**

The admin panel provides administrators with centralized control over the platform's operations. Admins can manage users, products, orders, and feedback submissions. It includes features such as user management, product management, order tracking, and analytics.

13

**Database Management Module:**

This module handles the storage and retrieval of data related to users, products, orders, and feedback. It ensures data integrity, security, and efficient querying to support the platform's functionalities. Database management includes tasks such as data backup, maintenance, and optimization.

**Security Module:**

Security measures are implemented throughout the platform to protect user data and transactions. It includes measures such as encryption, secure authentication, and protection against common web vulnerabilities.

Regular security audits and updates are conducted to ensure the platform's resilience against cyber threats.

14

**SCOPE OF SYSTEM**

The scope of the Pie-Clothing e-commerce platform encompasses various functionalities aimed at providing a seamless and enriching shopping experience for users while enabling efficient management and operation of the platform by administrators. Users can register by providing necessary details and create an account on the platform, with secure login and logout functionalities ensuring data protection, along with password recovery and reset options.

The product catalog management feature displays a wide range of clothing items such as shirts, trousers, dresses, jackets, and accessories, with detailed descriptions, images, pricing, and size information provided. Products are categorized based on types, making it easier for users to browse and find specific items.

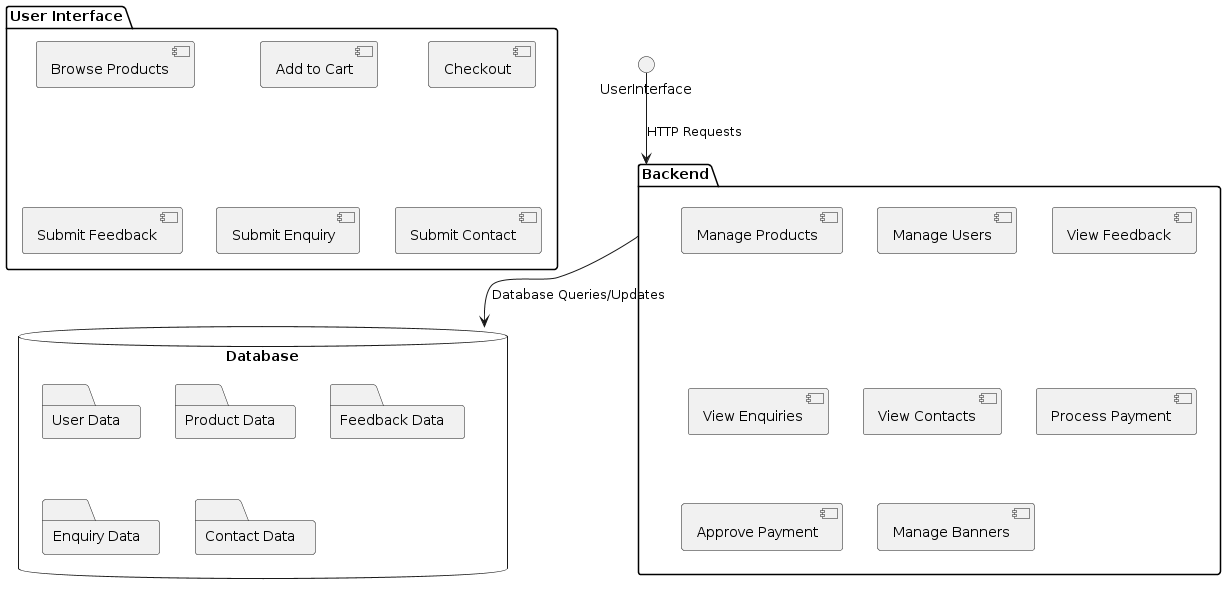
Users can add products to their shopping cart and modify quantities as needed, with the system calculating the total cost, including applicable taxes and discounts. The checkout process allows users to choose their preferred payment method, either through Razorpay for secure online transactions or cash on delivery. Users can view their order history and track the status of their current orders in real-time, with the platform generating and providing downloadable invoices for completed orders. Administrators can manage orders, update order statuses, and handle customer inquiries efficiently.

The platform is designed to handle increasing user traffic and expanding product inventory, with scalable infrastructure ensuring optimal performance as the platform grows. The focus is on providing a seamless and intuitive user interface, ensuring compatibility across different devices and browsers for a consistent user experience. The Pie-Clothing e-commerce platform aims to offer a comprehensive solution for buying and selling stylish clothing products, ensuring a high level of customer satisfaction

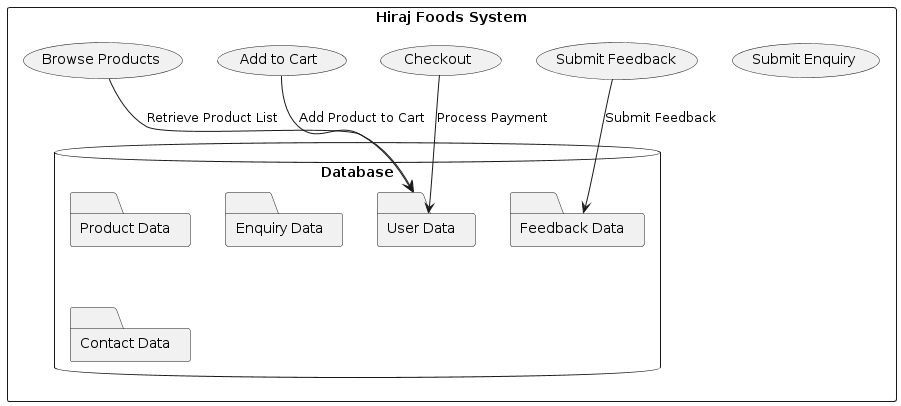
15

**SYSTEM DESIGN**

**System Architecture**



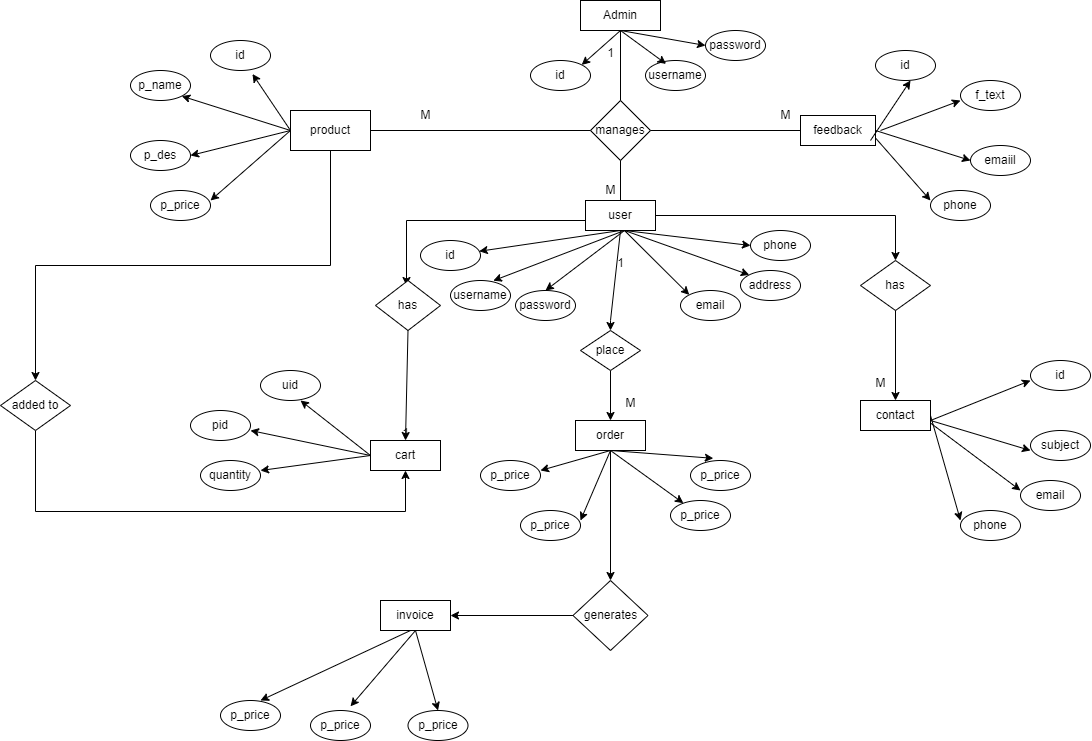
**ZERO LEVEL DATA FLOW DIAGRAM**



16

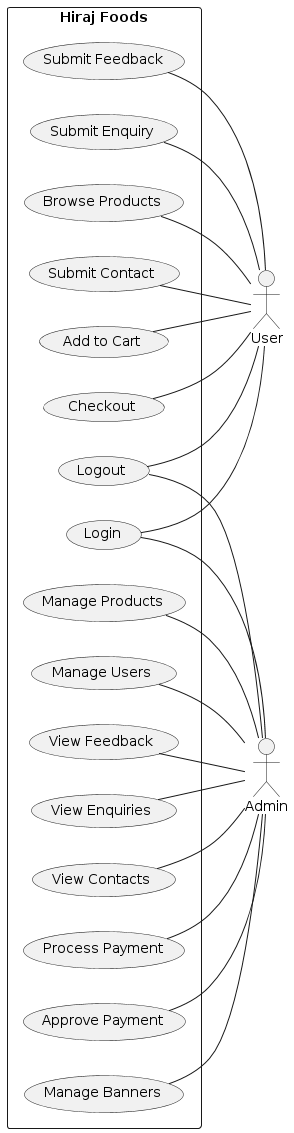
**UML DIAGRAMS**

**ER-DIAGRAM**



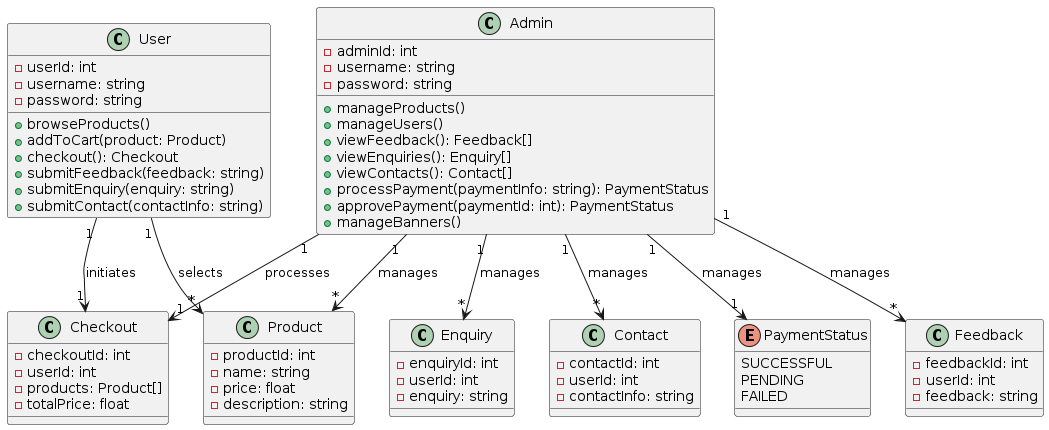
17

**USE CASE DIAGRAM**



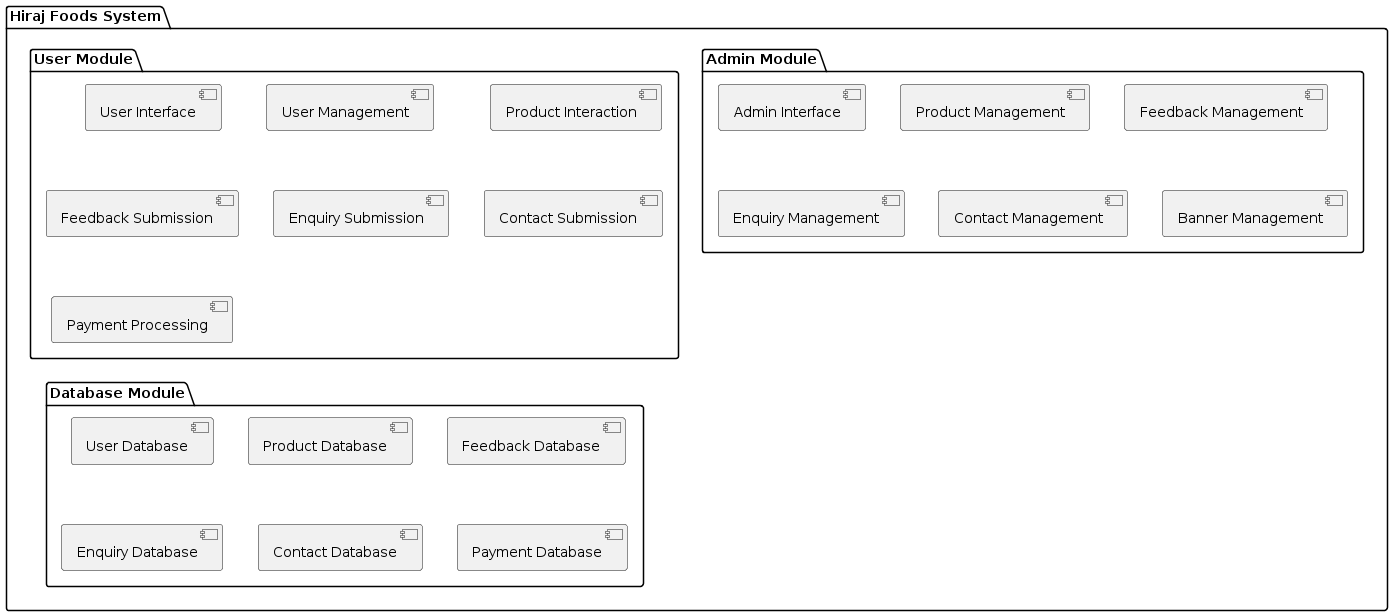
18

**CLASS DIAGRAM**

****

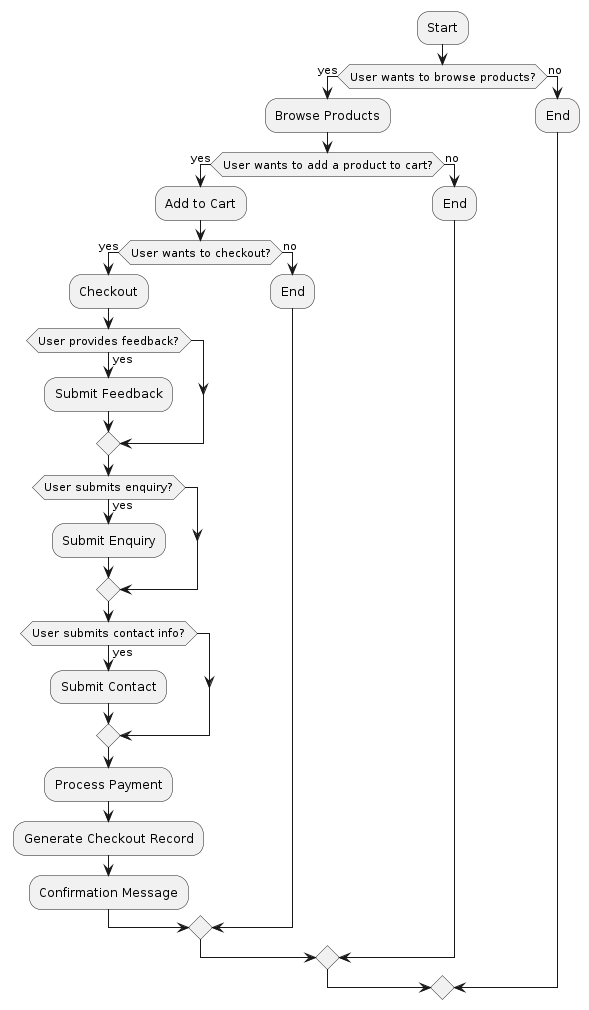
19

**MODULE HIERARCHY**

****

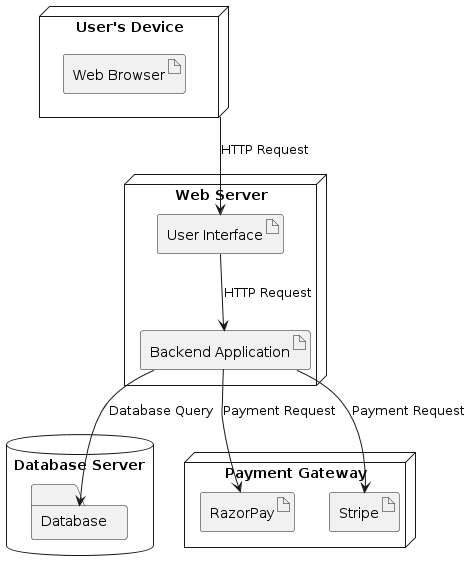
20

**ACTIVITY DIAGRAM:**

****

21

**DEPLOYMENT DIAGRAM**

****

22

**DATABASE DESIGN**

**Table name: User**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Datatype | Constraint | Description |
| UserID | OBJECT | primary key | unique user identifies |
| username | STRING | Not null | users username |
| Password | STRING | not null | User's password (hashed) |
| Email | STRING | not null | User's email address |
| Phone | STRING | not null | User Phone number |

**Table name: Admin**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Datatype | Constraint | Description |
| AdminID | OBJECT | primary key | unique user identifies |
| AdminName | STRING | Not null | users username |
| Password | STRING | not null | User's password (hashed) |
| Email | STRING | not null | User's email address |

23

**Product Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Datatype | Constraint | Description |
| ProductID | OBJECT | Primary Key | unique Product identifies |
| Name | STRING | Not Null | Product name |
| Description | STRING | Constraint | Product Description |
| Price | STRING | Not Null | Price |

**Cart Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Datatype | Constraint | Description |
| CartID | OBJECT | Primary Key | unique user identifies |
| UserID | OBJECT | Foreign Key (User.UserID) | Referenace of user id |
| ProductID | OBJECT | Foreign Key (Product.ProductID) | Referenace of product id |
| Quantity | OBJECT | Not null | Quantity of the product |

24

**Order Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Datatype | Constraint | Description |
| OrderID | OBJECT | Primary Key | unique order identifies |
| UserID | OBJECT | Foreign Key (User.UserID) | Referenace of user id |
| TotalAmount | INT | Not Null | Referenace of product id |
| Quantity | INT | Not null | Quantity of the product |

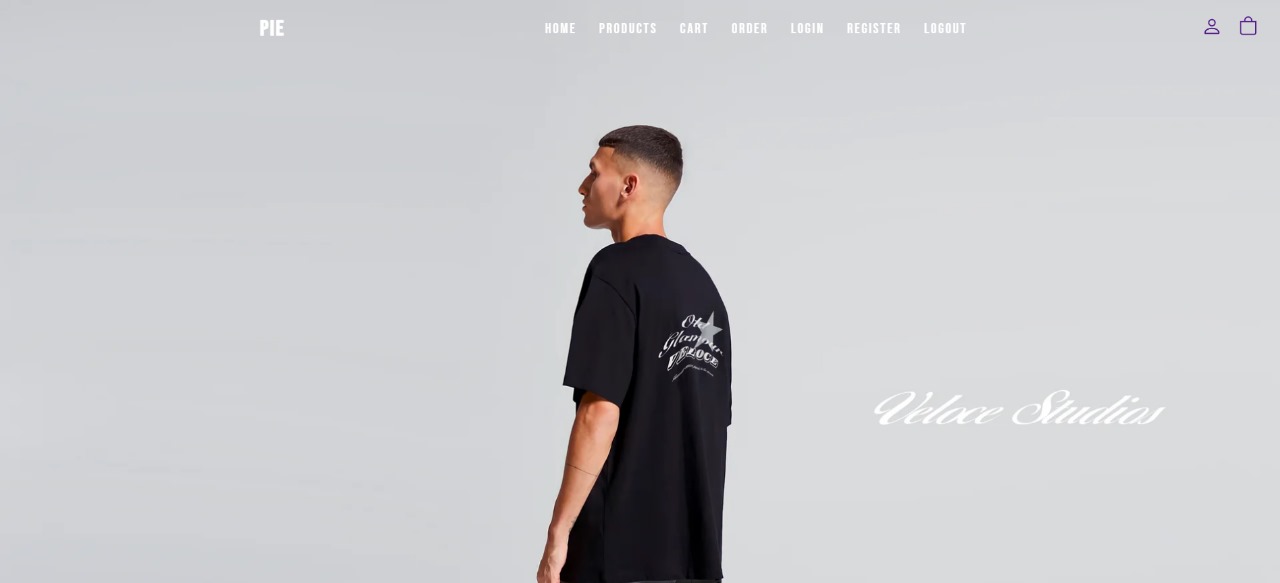
**Review Table:**

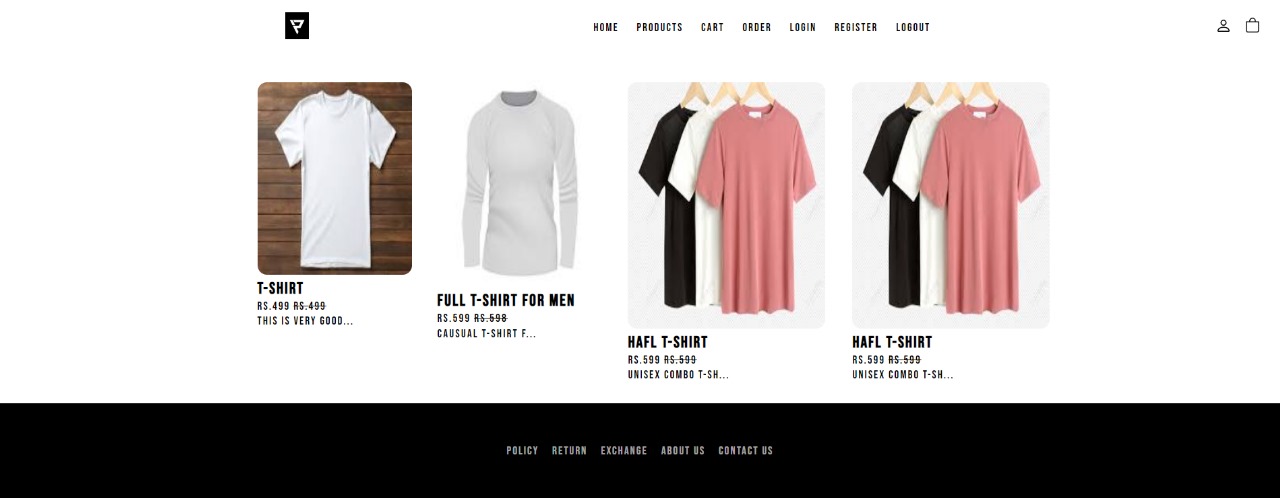
|  |  |  |  |
| --- | --- | --- | --- |
| Field | Datatype | Constraint | Description |
| ReviewId | OBJECT | Primary Key | unique identifies |
| ProductId | OBJECT | Foreign Key | Referenace of |
| Review | STRING | Not null | Review of Product |
| Rating | INT | Not null | Rating of Product |

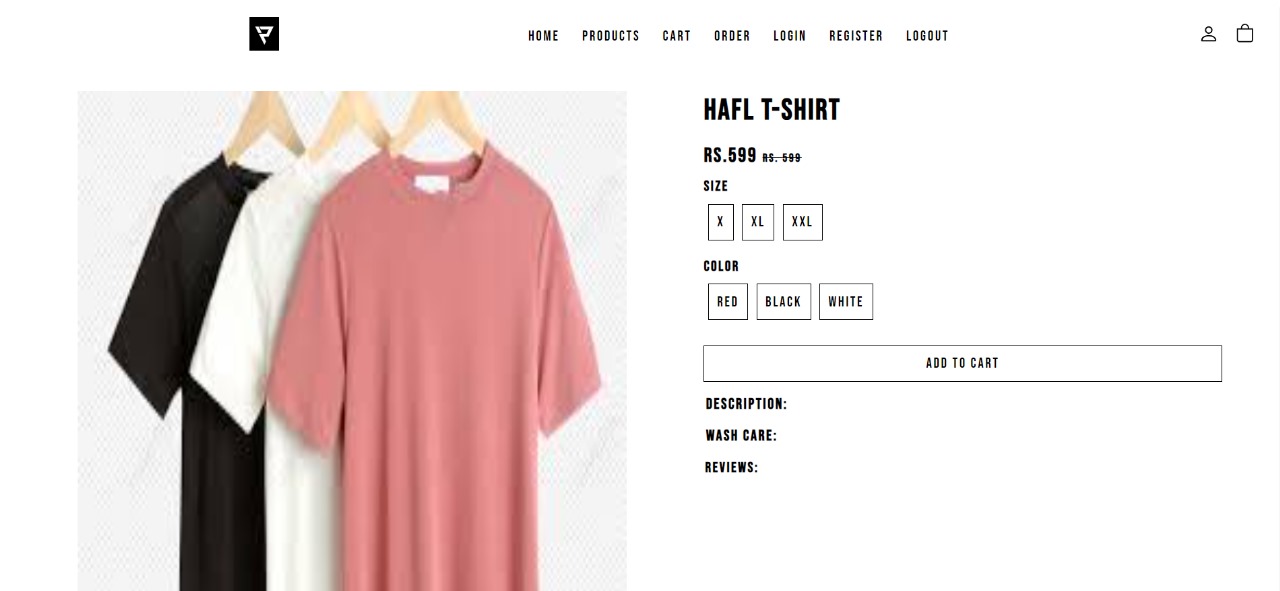
25

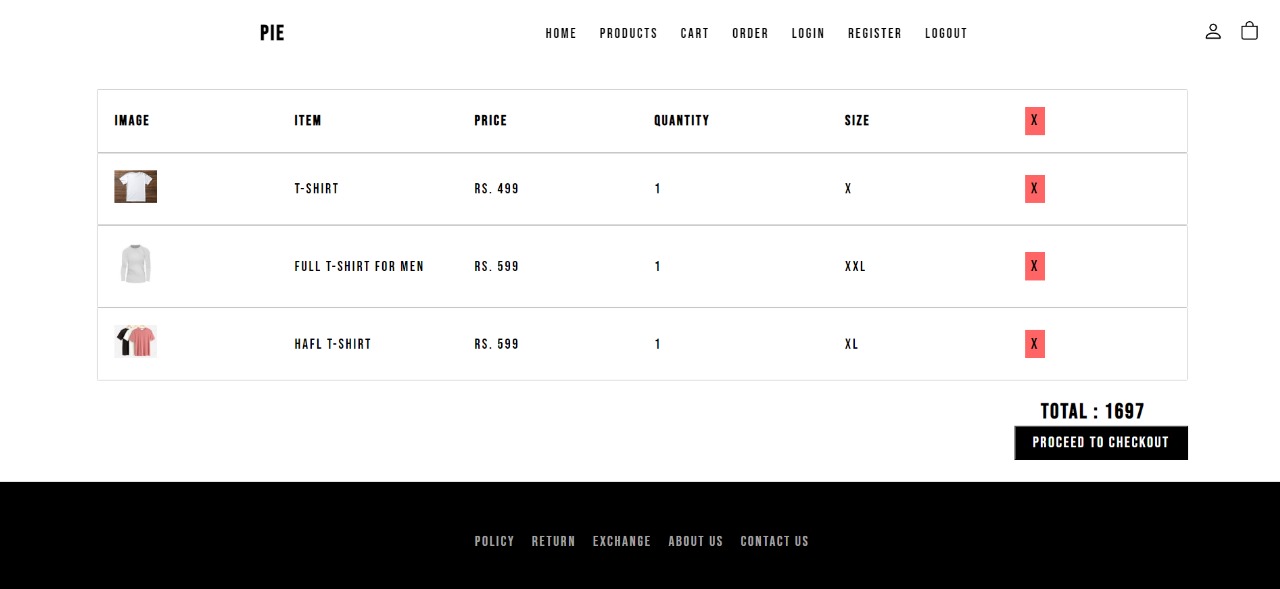
**Results**

**User Interface**

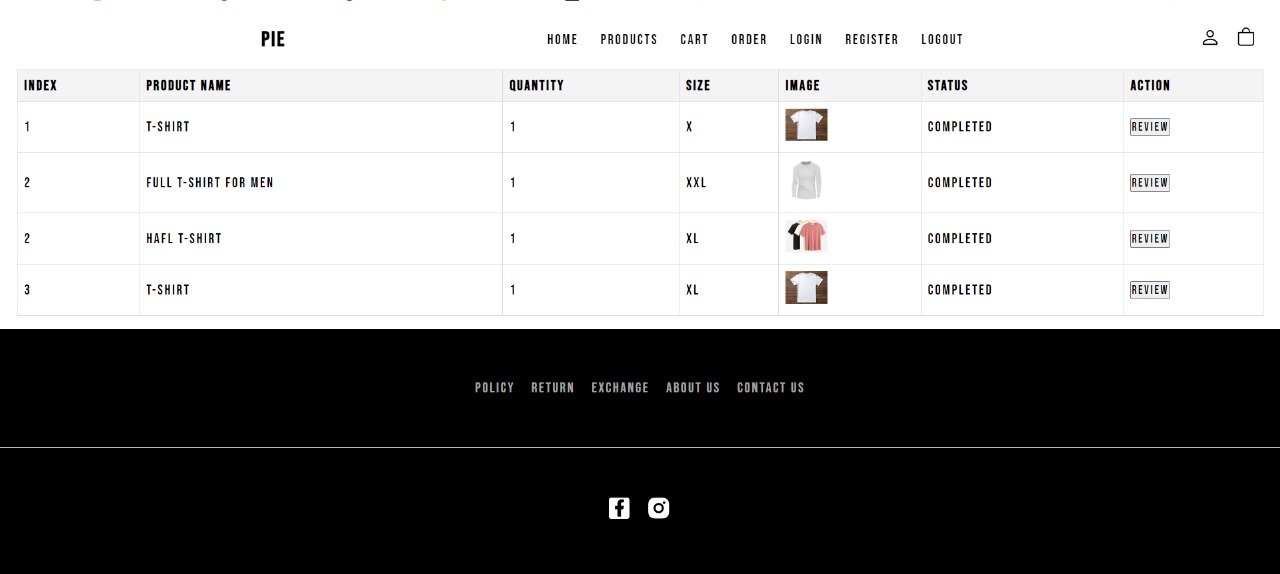






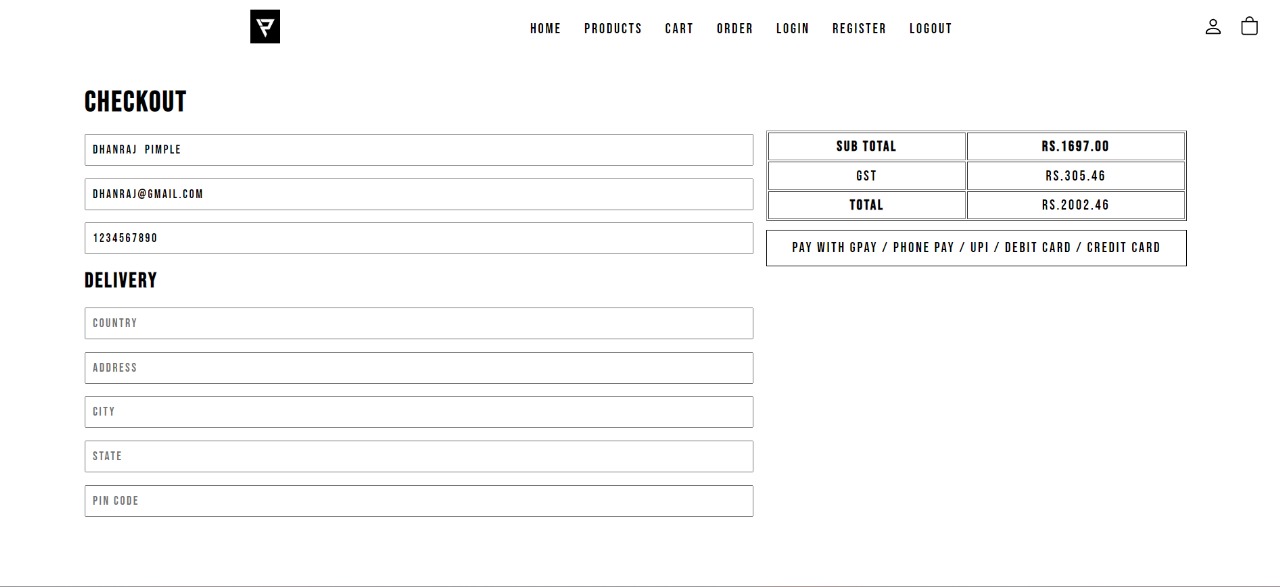


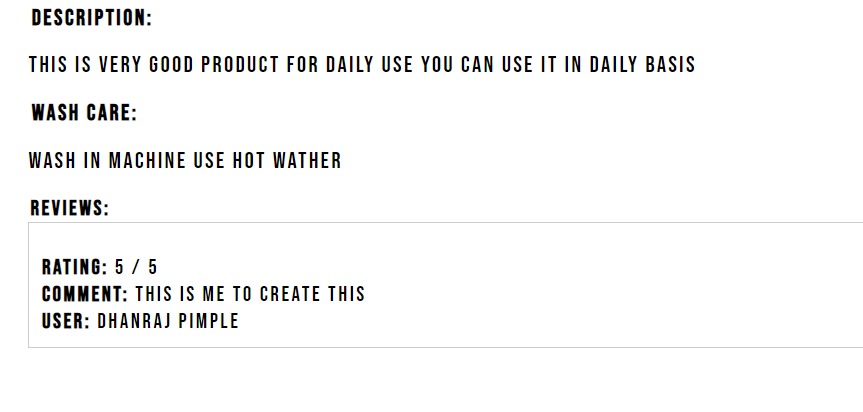
26





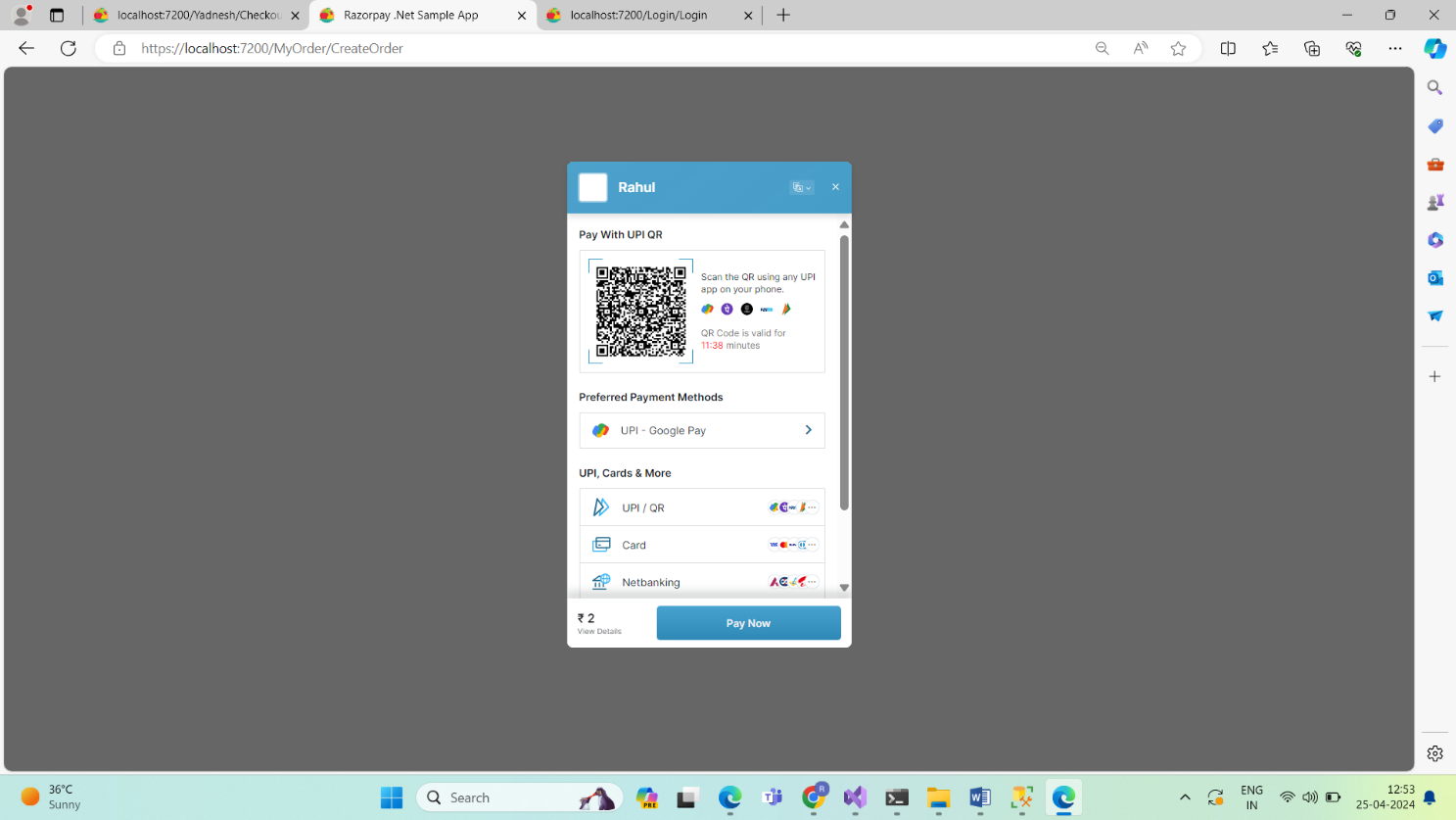
27





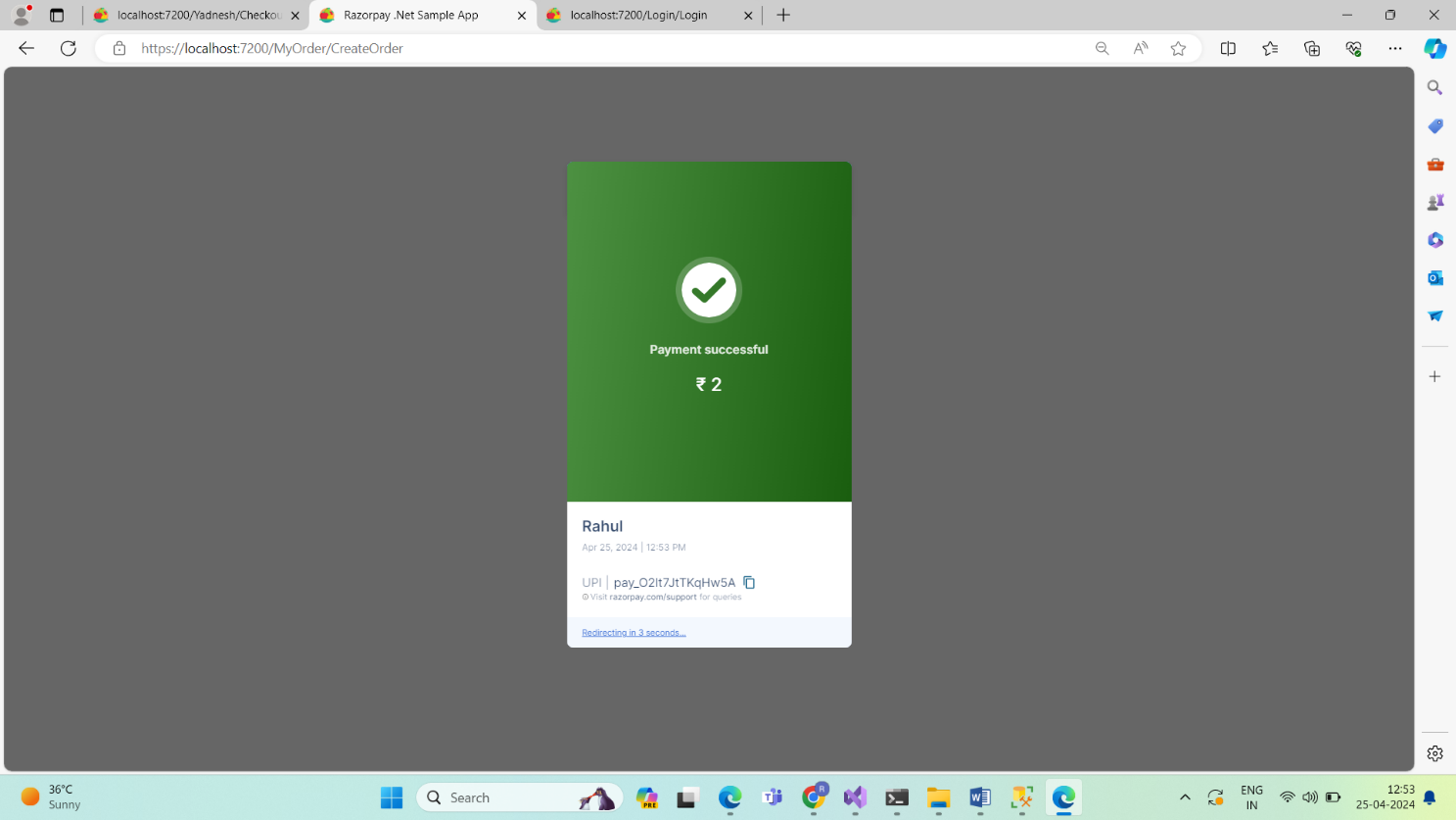
28

29



30

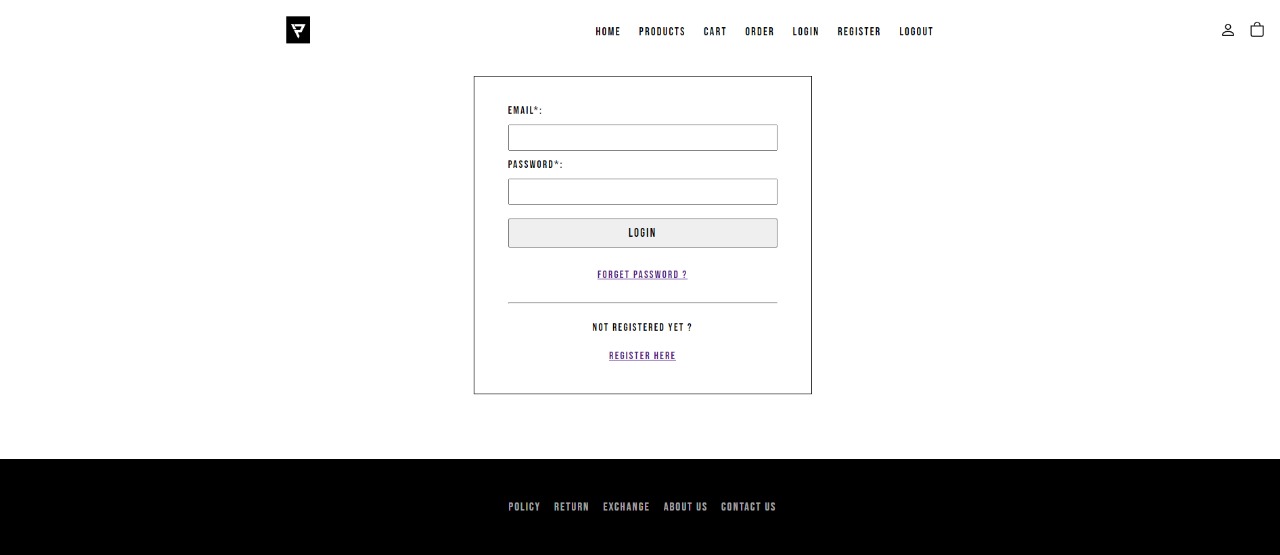
31

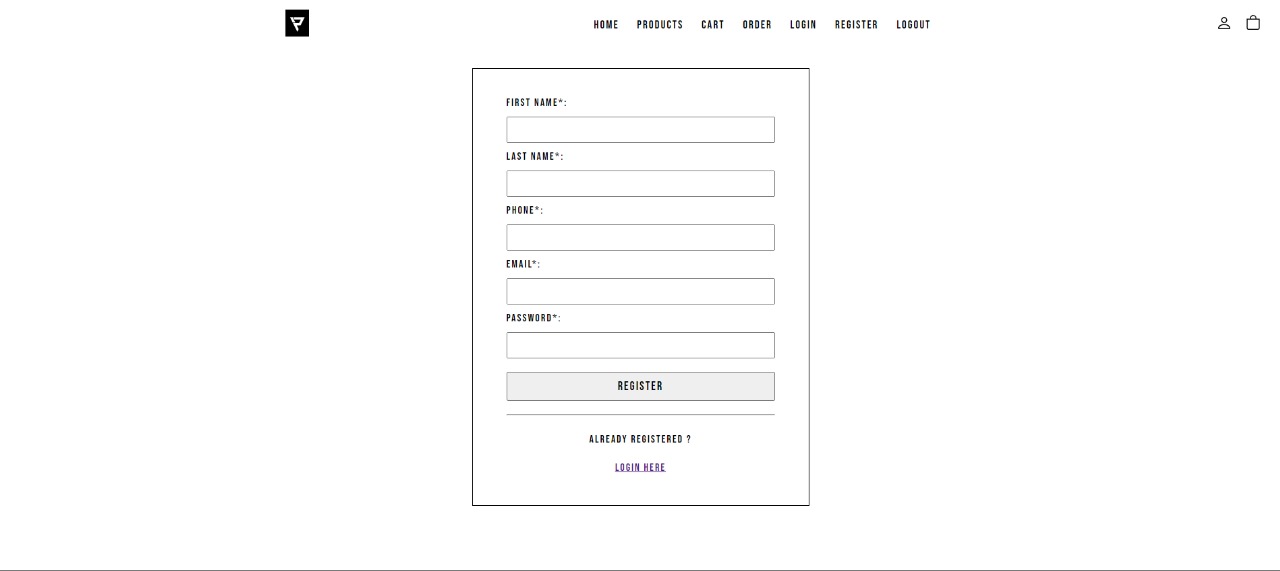


**Admin Dashboard:**

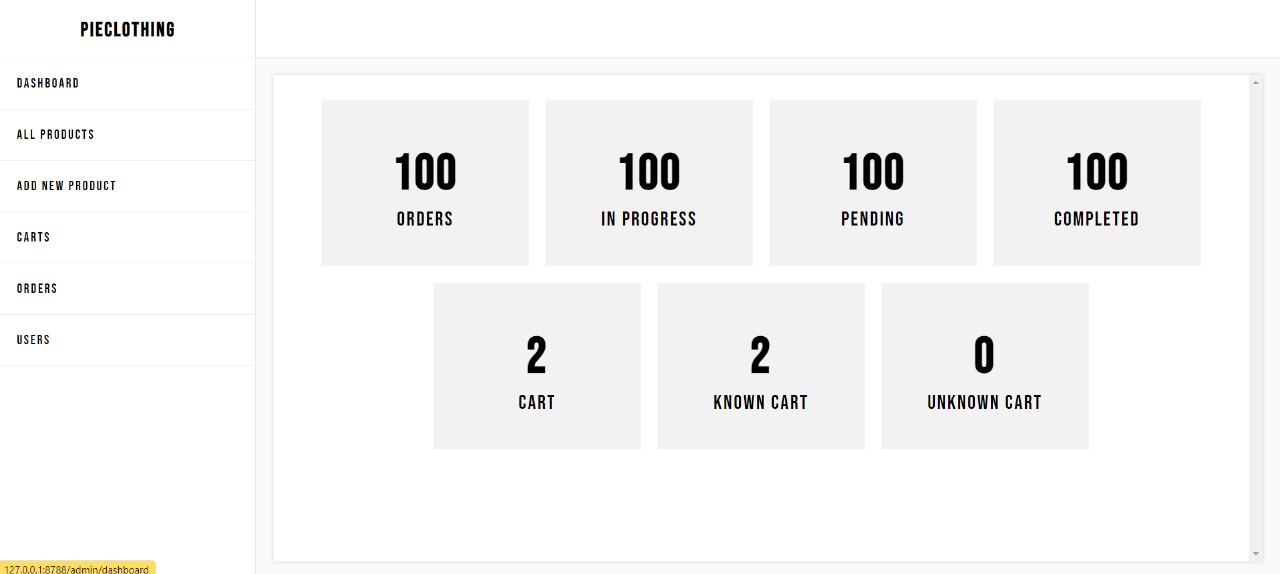
32

33

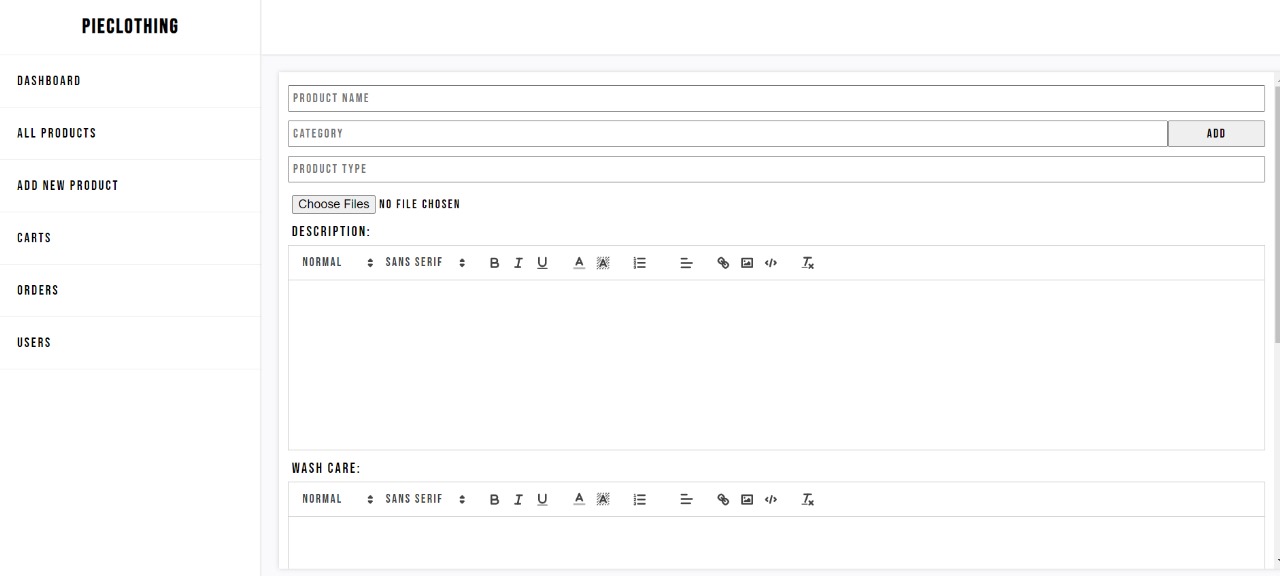


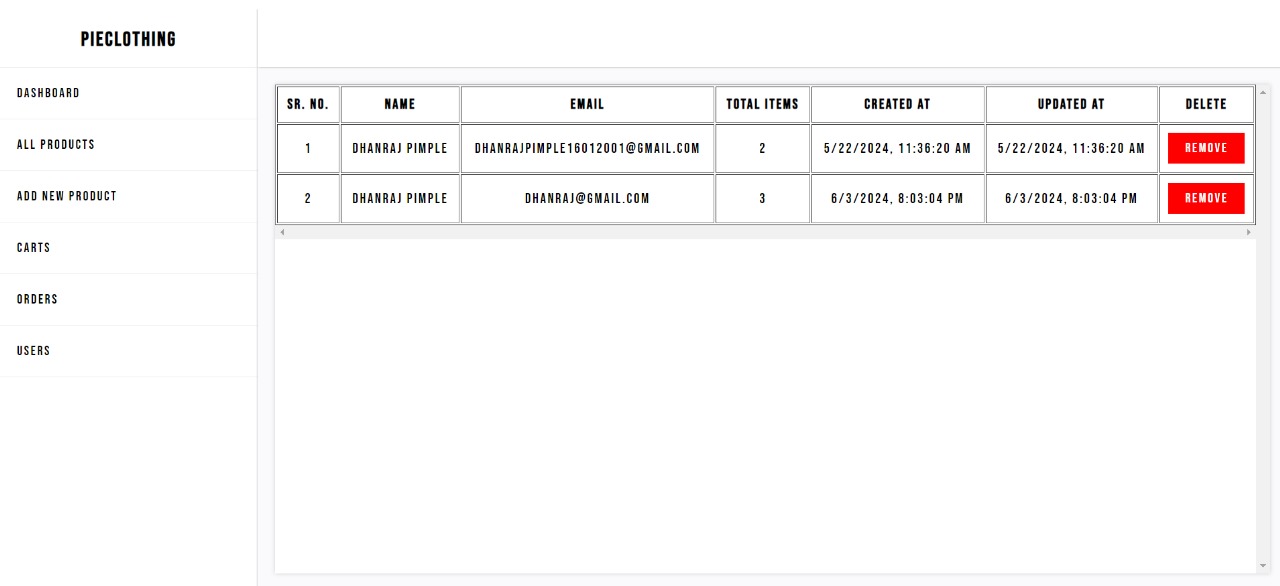
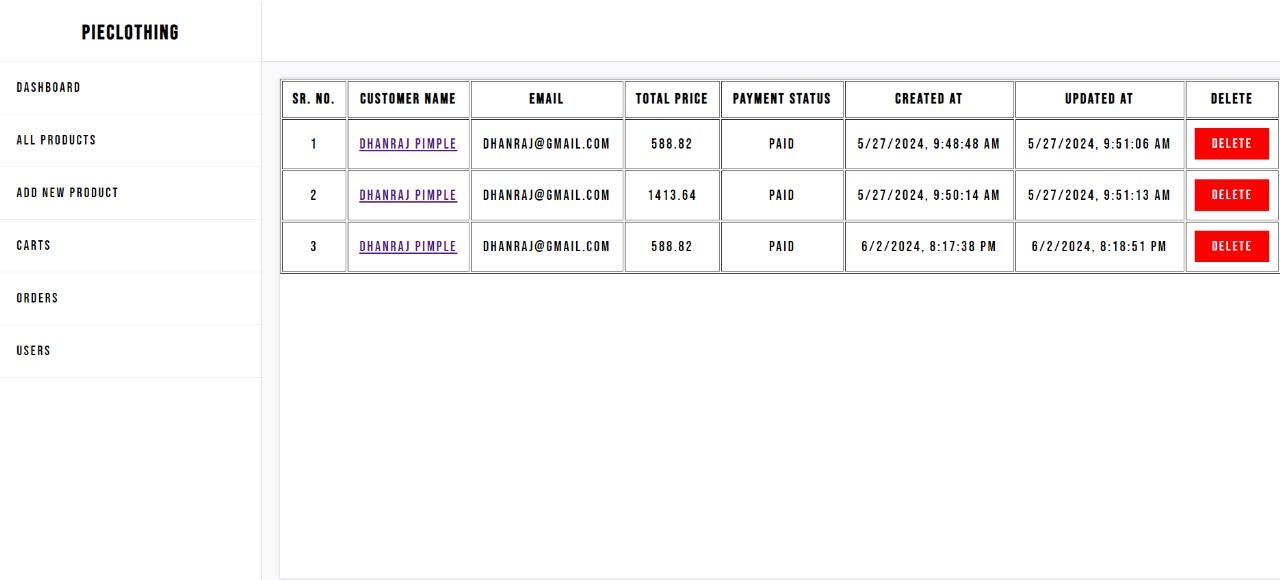


34



35



36





37

38

**CODING**

**User Model:**

const mongoose = require('mongoose');

const userSchema = new mongoose.Schema({

firstName: {

type: String,

required: true

},

lastName: {

type: String,

required: true

},

email: {

type: String,

required: true

},

phone: {

type: Number,

required:true,

},

AccountType: {

type: String,

enum:["user","admin"] // Assuming role is an array of strings

},

password: {

type: String,

required: true

},

extradetails: {

type: {},

},

created\_at: {

type: Date,

default: Date.now

},

updated\_at: {

type: Date,

default: Date.now

}

});

const UserModel = mongoose.model('User', userSchema);

module.exports = UserModel;

40

const mongoose = require("mongoose");

const orderSchema = new mongoose.Schema({

orderId: {

type: String,

required: true,

},

amountPaid: {

type: Number,

required: true,

default: 0,

},

amountRemaining: {

type: Number,

required: true,

default: 0,

},

name: {

type: String,

required: true,

},

email: {

type: String,

required: true,

},

phone: {

type: String,

required: true,

},

deliveryAdd: {

type: {

address: { type: String, required: true },

city: { type: String, required: true },

pincode: { type: String, required: true },

state: { type: String, required: true },

country: { type: String, required: true },

},

required: true,

},

status: {

type: String,

required: true,

enum: ["In Progress", "Completed"],

default: "In Progress"

},

cartId: {

type: String,

required: true,

},

products: [

{

productId: String,

gallery: [String],

},

],

uid: {

type: String,

required: true,

},

tempUid: {

type: String,

required: false,

},

created\_date: {

type: Date,

default: Date.now,

},

updated\_date: {

type: Date,

default: Date.now,

},

});

orderSchema.pre("save", function (next) {

this.updated\_date = Date.now();

next();

});

orderSchema.pre("findOneAndUpdate", function (next) {

this.\_update.updated\_date = Date.now();

next();

});

module.exports = mongoose.model("Order", orderSchema);

41

**Contact Model:**

const mongoose = require('mongoose');

const addressSchema = new mongoose.Schema({

uid: {

type: mongoose.Schema.Types.ObjectId,

ref: "User", },

street: {

type: String,

},

landmark: {

type: String,

required: true,

},

city: {

type: String,

required: true,

},

pin: {

type: Number,

required: true,

},

state: {

type: String,

required: true,

},});

const Address = mongoose.model('Address', addressSchema);

module.exports = Address;

42

**Enquiry Model:**

const mongoose = require("mongoose");

const cartSchema = new mongoose.Schema({

id: {

type: String,

// default: mongoose.Types.ObjectId(),

},

products: [],

uid: {

type: String,

},

tempUid: {

type: String,

},

created\_date: {

type: Date,

default: Date.now,

},

updated\_date: {

type: Date,

default: Date.now,

},

});

module.exports = mongoose.model("Cart", cartSchema);}

43

46

48

49

**Testing**

**Test Strategy**

* Automation Strategies: Utilized Selenium WebDriver and Spec flow for testing the online glossary shop's web and mobile applications. The automation scripts have been successfully executed and verified.
* Test Schedule: Tests are scheduled to run daily, with each test suite taking approximately 2 to 3 hours to complete.
* Resources: The test resources include detailed test data, user scenarios, and performance benchmarks relevant to the online glossary shop.

**Unit Test Plan**

* Product Catalog: Each module of the product catalog was individually unit tested to verify functionality such as item listing, filtering, and searching before integrating into the main application.
* User Accounts: User account functionalities, including registration, login, password recovery, and profile management, were unit tested and confirmed to work correctly.

**Acceptance Test**

**User Module Testing:**

* Test 1: Registered users were tested to ensure they can log in successfully.
* Test 2: After logging in, users were tested to verify they could browse through product categories and view lists of items by category.
* Test 3: Users were tested to confirm they could view detailed product information, including availability and pricing.
* Test 4: The shopping cart functionality was tested by adding items, updating quantities, and removing items to ensure correct operation.
* Test 5: The checkout process was tested, including selecting shipping options, entering payment information, and placing orders.

50

**Admin Module Testing:**

* Test 1: Admins were tested to ensure they could log in and access the admin dashboard.
* Test 2: Product management functionalities were tested, including adding new products, updating existing product details, and deleting products.
* Test 3: Order management was tested to verify admins could view and process incoming orders, update order statuses, and manage shipping information.
* Test 4: User management functionalities were tested, including viewing user profiles, managing user permissions, and handling user inquiries.
* Test 5: Inventory management was tested to ensure admins could track stock levels, receive stock alerts, and update inventory records.
* Test 6: Reports generation functionality was tested, ensuring admins could generate sales reports, customer reports, and inventory reports accurately

51

**Test Case:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Scenario** | **Description** | **Preconditions** | **Test Steps** | **Expected Result** | **Result** |
| TC-01 | Web application startup | Verify that the web application starts up correctly without errors. | The server and necessary services are set up and running. No configuration errors exist. | 1. Ensure the server hosting the web application is up and running.  2. Open a web browser.  3. Enter the URL of the web application  4. Press Enter to load the web application. | The web application loads successfully, displaying the homepage without any errors. | Pass |
| TC-02 | User Registration | Verify that a new user can register successfully by providing all required information and documents. | None | 1. Navigate to the registration page.  2. Enter valid user information.  3. Click on the "Register" button. | The user is registered successfully, and a verification email is sent. | Pass |
| TC-03 | User Login | Verify that a registered user can log in to the system. | The user must be registered and verified. | 1. Navigate to the login page.  2. Enter-valid email and password.  3. Click on the "Login" button. | The user is logged in and redirected to the dashboard. | Pass |
| TC-04 | Product Search | Search for a product | Verify that customers can search for products. | 1. Enter a product name or keyword in the search bar.  2. Click the search button. | Relevant products are displayed in the search results. | Pass |

**Valid Test Case for User:**

52

**Invalid Test Case for User:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Scenario** | **Description** | **Preconditions** | **Test Steps** | **Expected Result** | **Result** |
| TC-01 | Web Application Startup with Server Down | Verify that the web application does not start up when the server is down. | The server hosting the web application is not operational or has been deliberately shut down.. | 1. Ensure the server hosting the web application is shut down or not operational.  2. Open a web browser.  3. Enter the URL of the web application  4. Press Enter to load the web application. | The web application fails to load, and an error message (e.g., “Server not found” or “Unable to connect”) is displayed in the browser. | Fail |
| TC-02 | User Registration | New user registration with missing required information. | None | 1. Navigate to the registration page.  2. Enter invalid user information.  3. Click on the “Register” button. | Registration fails, and an error message is displayed indicating that all required fields must be filled. | Fail |
| TC-03 | User Login | Existing user login with invalid credentials | Verify that a user cannot log in with incorrect email or password. | 1. Navigate to the login page.  2. Enter-Invalid email and password.  3. Click on the “Login” button. | Login fails, and an error message is displayed indicating invalid credentials. | Fail |

53

**Valid Test Case for Admin:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Scenario** | **Description** | **Preconditions** | **Test Steps** | **Expected Result** | **Result** |
| TC-01 | Admin Login | Verify that a registered Admin can log in to the system. | The Admin must be registered and verified. | 1. Navigate to the login page.  2. Enter-valid email and password.  3. Click on the "Login" button. | The Admin is logged in and redirected to the dashboard. | Pass |
| TC-02 | Add a new product category | Verify that a shopkeeper can add a new product category. | Shopkeeper is logged in. | 1. Navigate to the product category management page.  2. Click on "Add New Category".  3. Enter category details and submit. | New category is displayed in the product categories list. | Pass |
| TC-03 | Submit customer feedback | Verify that customers can submit feedback on products. | Admin is logged in and has purchased a product. | Display the feedback | Feedback is successfully submitted and displayed on the product page | Pass |
| TC-04 | Display special offers | Verify that special offers are displayed correctly. | Special offers are configured in the system. | 1. Navigate to the homepage.  2. View the special offers section. | Special offers are displayed as configured. | Pass |

54

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Scenario** | **Description** | **Preconditions** | **Test Steps** | **Expected Result** | **Result** |
| TC-01 | Admin login with invalid credentials | Verify that an Admin cannot log in with incorrect email or password. | The Admin must be registered and verified. | 1. Navigate to the login page.  2. Enter-valid email and password.  3. Click on the "Login" button. | The Admin cannot log in, and an error message is displayed indicating invalid credentials.. | Fail |
| TC-02 | Add a new product category with missing details. | Verify that a shopkeeper cannot add a new product category if required details are missing. | Shopkeeper is logged in. | 1. Navigate to the product category management page.  2. Click on "Add New Category".  3. Enter category details and submit. | The new category is not added, and an error message is displayed indicating that all required fields must be completed. | Fail |
| TC-03 | Submit a customer support inquiry with missing details. | Verify that a customer cannot submit an inquiry if required details are missing. | Admin is logged in. | 1. Navigate to the customer support page.  2. Fill in the inquiry form with incomplete details (e.g., leave the message or contact information field empty).  3. Submit the inquiry. | Inquiry submission fails, and an error message is displayed indicating that all required fields must be completed. | Fail |

**Invalid Test Case For Admin:**

55

**FUTURE ENHANCEMENT**

* **Personalized** **Recommendations**: Implement machine learning algorithms to analyze user behavior and preferences, enabling the platform to offer personalized product recommendations. By understanding each user's preferences, the platform can suggest relevant products, enhancing the overall shopping experience and increasing sales.
* **Social** **Media** **Integration**: Integrate social media platforms such as Facebook, Instagram, and Twitter to enhance user engagement and reach. Enable users to share their purchases, reviews, and experiences with their social networks, fostering brand awareness and driving traffic to the platform.
* **Subscription** **Services**: Introduce subscription-based services for recurring purchases of favorite products. Allow users to subscribe to regular deliveries of their preferred fruit powders, offering convenience and ensuring customer loyalty.
* **Virtual** **Try**-**On**: Implement augmented reality (AR) technology to enable users to virtually try on products before making a purchase. This feature can simulate how different fruit powders would look and taste, helping users make informed decisions and reducing the likelihood of returns.
* **Enhanced** **Feedback** **Mechanisms**: Expand the feedback system to include more detailed surveys and ratings for products and services. Gather insights into user preferences, satisfaction levels, and areas for improvement, allowing the platform to continually refine its offerings and customer experience.
* **International** **Expansion**: Explore opportunities to expand the platform's presence into international markets, catering to a global audience of health-conscious consumers. Localize the platform by offering multilingual support and adapting product offerings to suit regional preferences and dietary requirements.
* **Partnerships** **with** **Influencers**: Collaborate with health and wellness influencers to promote Pie-clothing products to their followers. Leverage influencers' credibility and reach to increase brand visibility, attract new customers, and build trust within the target demographic.
* **Environmental** **Sustainability** **Initiatives**: Implement eco-friendly packaging solutions and sustainable sourcing practices to minimize the platform's environmental impact. Highlight these efforts to resonate with environmentally conscious consumers and differentiate the brand in the market.
* **Expanded** **Product** **Line**: Continuously innovate and expand the range of health-focused products offered on the platform. Introduce new fruit powders, superfood blends, and complementary health products to cater to evolving consumer preferences and emerging health trends.

56

**LIMITATIONS**

**Limited Product Range**: The platform's success heavily depends on the variety and quality of products offered. Limitations in sourcing or availability of healthy fruit-based products may restrict the platform's growth potential.

**Regional Availability**: Availability of products may be limited to certain regions due to logistical constraints, potentially excluding potential customers from areas where delivery is not feasible or cost-effective.

Dependency on Third-party Payment Gateways: Reliance on external payment gateways like Razorpay introduces a level of dependency and vulnerability to service interruptions, transaction fees, or changes in terms and conditions.

**Security Risks**: Despite implementing security measures, the platform may still be susceptible to cybersecurity threats such as data breaches, hacking attempts, or fraud, posing risks to user privacy and financial security.

**Scalability Challenges**: As the user base and product catalog grow, the platform may encounter scalability challenges in terms of server capacity, database performance, and processing power, leading to slower response times or system crashes during peak traffic periods.

**Technical Issues**: The platform may experience technical glitches, bugs, or compatibility issues across different devices and browsers, affecting user experience and satisfaction.

**Limited Customer Reach**: Marketing and promotional efforts may be limited by budget constraints, hindering the platform's ability to reach and attract a broader audience beyond existing customer networks.

**Competition**: The e-commerce market is highly competitive, with established players and new entrants vying for market share. Pie-Clothing may face challenges in standing out amidst competitors and capturing a significant share of the market.

**Regulatory Compliance:** Adhering to various regulations and compliance requirements related to e-commerce, data protection, taxation, and consumer rights adds complexity and overhead costs to the operation of the platform.

**Customer Retention**: Maintaining customer loyalty and repeat business may be challenging, especially in a competitive market where customers have numerous options. The platform must continuously focus on enhancing customer experience and offering incentives to encourage repeat purchases.

57

**CONCLUSION**

The development of the Pie-Clothing e-commerce platform represents a significant step towards providing consumers with convenient access to stylish and high-quality clothing. Through a thorough feasibility study, it has been determined that the proposed system is technically, operationally, and economically viable. Leveraging established web development technologies, the platform offers a user-friendly interface for seamless product exploration, purchasing, and feedback submission.

The objectives of the proposed system have been successfully outlined and addressed, aiming to create a comprehensive platform that caters to the diverse needs of both users and administrators. By expanding product offerings, enhancing user interaction, and empowering administrators with effective management tools, the platform is poised to deliver a superior e-commerce experience for all stakeholders involved.

The system's user base comprises various stakeholders, including end consumers, administrators, developers, and customer support representatives, each playing a crucial role in ensuring the platform's success. With a focus on optimizing performance, ensuring data security, and promoting scalability, the Pie-Clothing platform is positioned to meet the evolving needs of fashion-conscious consumers and drive growth for the brand.

In conclusion, the Pie-Clothing e-commerce platform represents a valuable investment in meeting the demands of the fashion market while fostering engagement, efficiency, and satisfaction among users and administrators alike. Through continuous iteration and improvement, the platform is poised to become a leading destination for stylish and high-quality clothing products, enhancing the lifestyle of consumers worldwide.

58

**BIBLIOGRAPHY**

**Remix Documentation**  
Remix. (n.d.). Remix Documentation. Retrieved from https://remix.run/docs

**CSS Documentation**  
W3Schools. (n.d.). CSS Tutorial. Retrieved from https://www.w3schools.com/css/

**TypeScript Documentation**TypeScript. (n.d.). TypeScript Documentation. Retrieved from https://www.typescriptlang.org/docs/

**Node.js Documentation**  
Node.js. (n.d.). Node.js Documentation. Retrieved from https://nodejs.org/en/docs/

**MongoDB Documentation**  
MongoDB. (n.d.). MongoDB Documentation. Retrieved from <https://docs.mongodb.com/>

59

**Acknowledgement**

With profound gratitude, we extend our heartfelt appreciation to all those whose contributions have been instrumental in realizing the completion of this project report.

Foremost among them is our esteemed project guide, **Mariyam Maniyar**, whose invaluable guidance, unwavering support, and motivational encouragement have served as the bedrock of this research endeavor. Her expertise, insightful feedback, and unwavering patience have played an indispensable role in shaping the trajectory and substance of this project.

We also wish to convey our sincere appreciation to the HOD, **Dr. Vandana Bagal**, for her invaluable contributions and scholarly insights, which have significantly enriched the content and scope of this work.

To all those whose wisdom, counsel, and encouragement have illuminated our path, we are deeply thankful.

Dhanraj Pimple

K. K. Wagh Institute of Engineering Education and Research

60

**Documentation page**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Report Documentation** | | | | | |
| Report Code: **MCA-Project 2023-2024** | | | | **Report Number:** | |
| Report Title: **Pie-Clothing** | | | | | |
| **Address (Details):**  **Department of MCA, K. K. Wagh Institute of Engineering Education & Research,**  **Hirabai Haridas Vidyanagari, Amrutdham, Nashik**  **Pin – 422 003, M.S. INDIA.** | | | | | |
| **Author1 [with Address, phone, E-mail]:**  **Address: Rane Nagar, Nashik**  **E-mail: dhanajpimple16@gamil.com**  **Roll: 44**  **Cell No: 914689.521** | | | | | |
| **Year:** 2023 – 2024  **Programme: MCA** | | | | | |
| ***Key Words*:**  **E-commerce, cloths, fashion, Secure checkout, User experience, e-shop, Customer satisfaction** | | | | | |
| Type of Report: FINAL | **Report Checked By**: Prof Mariyam Maniyar | Report Checked Date:  31-5-24 | **Guides Complete Name:** Prof Mariyam Maniyar | | Total Copies    **N+1** |

61