**A**

**PROJECT REPORT**

**ON**

### **CEREAL SPHERE**

**FOR**

### **Balaji Trading Company**

**IN PARTIAL FULFILLMENT OF**

**MASTER OF COMPUTER APPLICATION**

**BY**

**POOJA MAHENDRA KOTHAWADE**

**MCA –II SEM – IV DIV-C**

**(2023-2024)**

**ROLL NO: 22332**

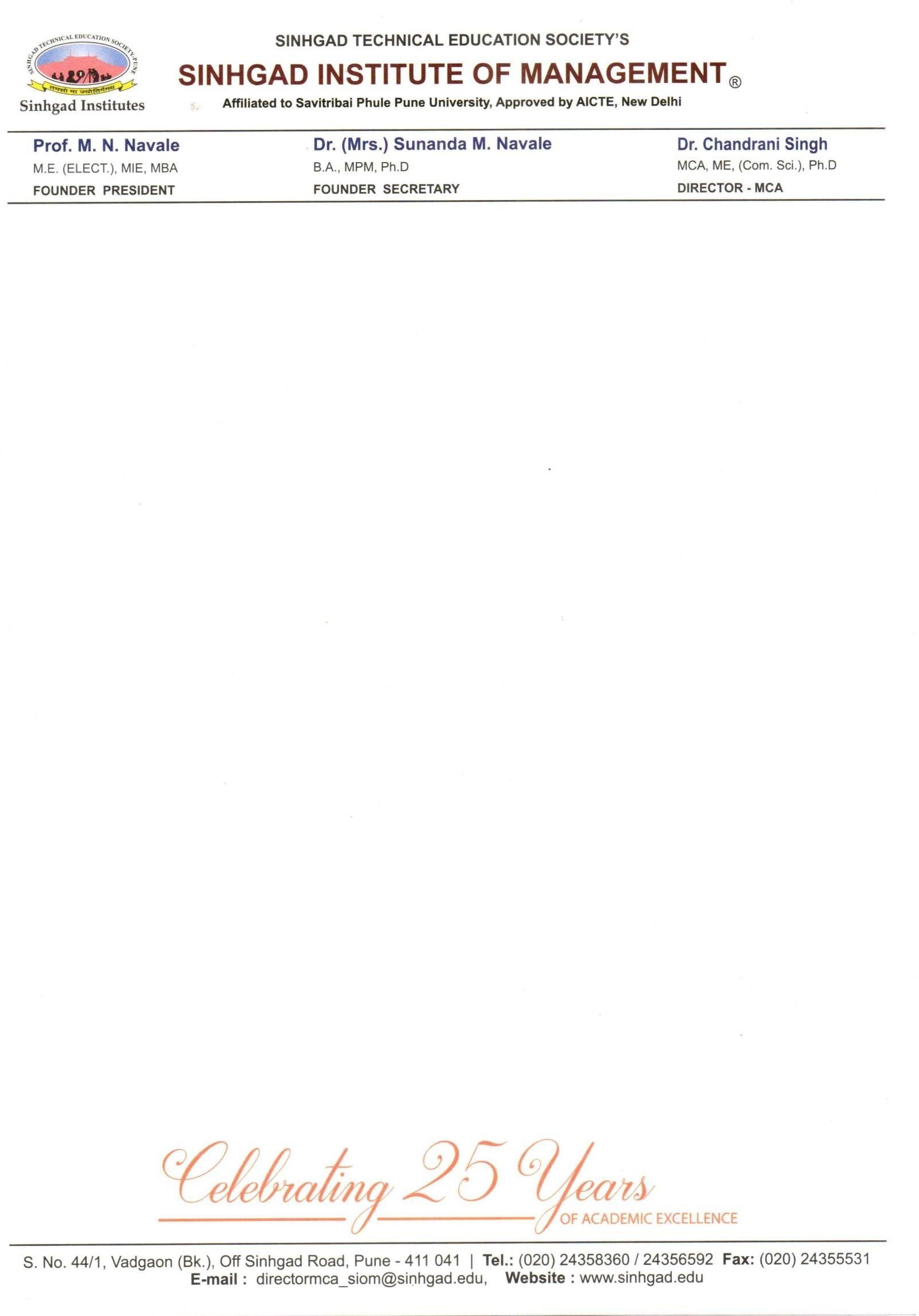
**Project Guide: Proff. Rupali Taware**

**SUBMITTED TO**

**SAVITRIBAI PHULE PUNE UNIVERSITY**

**SINHGAD INSTITUTE OF MANAGEMENT**

**PUNE-411 041**



# Date:

**CERTIFICATE**

This is to certify that **Ms. Pooja Kothawade**, has successfully completely his internship project work entitled **“ CEREAL SPHERE ”** in partial fulfilment of MCA – II SEM –IV Internship Project for the year **2023-2024**. He has worked under our guidance and direction.

# **Prof. Rupali Taware** **Dr. Chandrani Singh**

**Project Guide Director, SIOM-MCA**

**Examiner 1 Examiner 2**

**Date:**

**Place:**

**DECLARATION**

I, the undersigned hereby declare that the project titled **“Cereal sphere”** being submitted for the award of degree of Master of Computer Application by me to Sinhgad Institute of Management (SIOM) affiliated to Savitribai Phule Pune University is the result of an independent work carried out under the guidance of Prof. Milind Godse, is my original work. Further I declare that this project has not been submitted to this or any Institution for the award of any degree.

PLACE: PUNE **Pooja Kothawade**

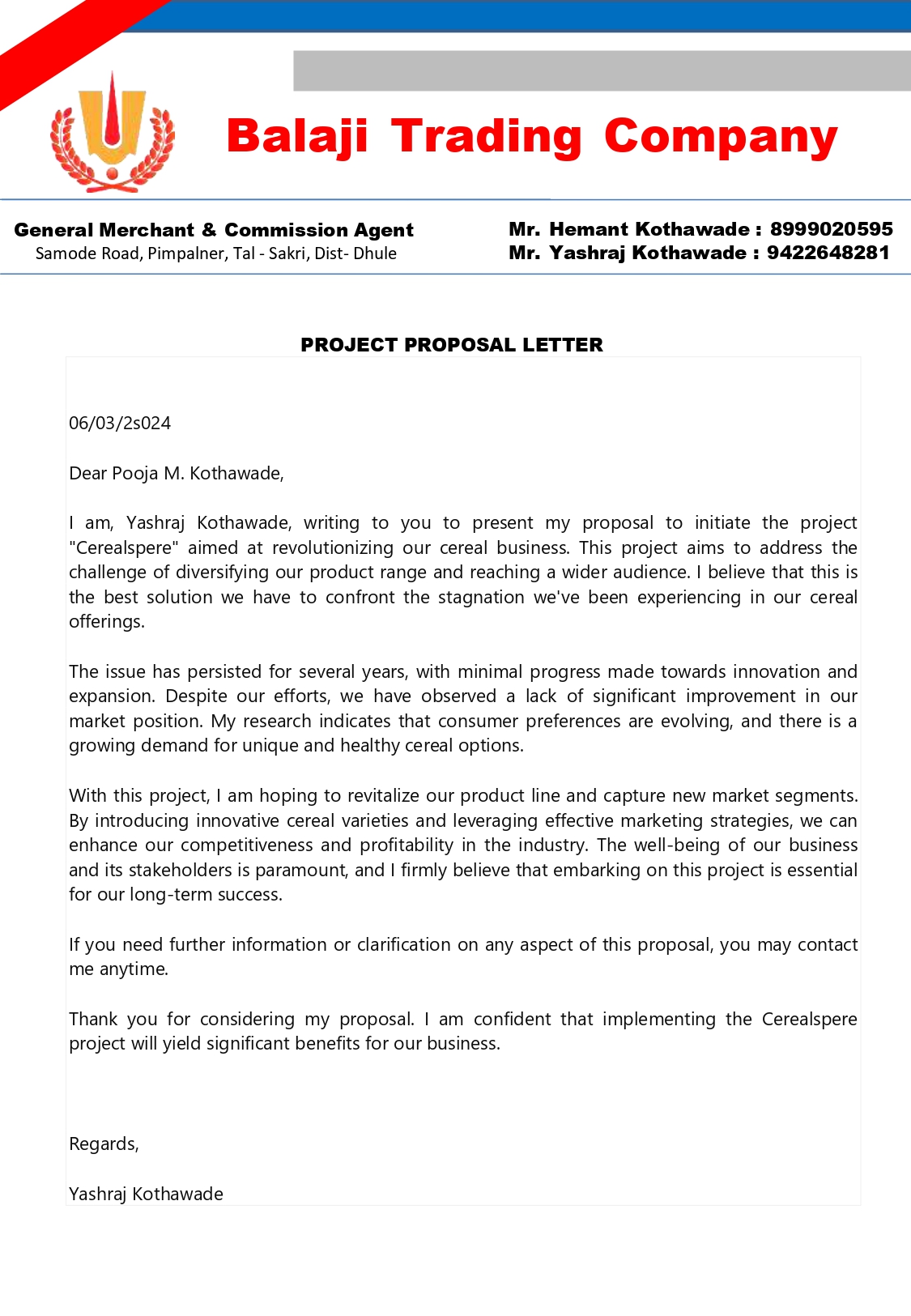
DATE: (Student)

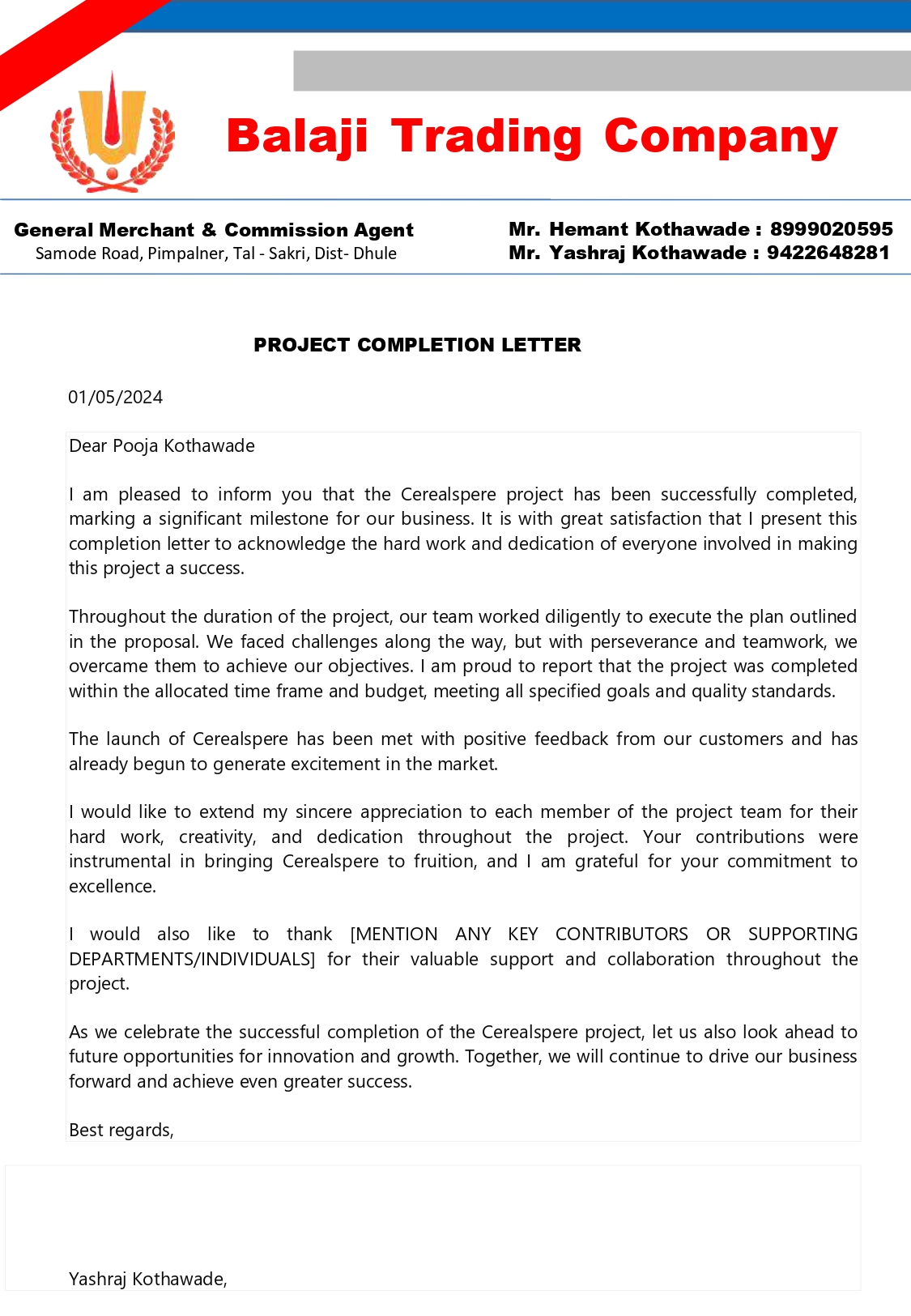
**ACKNOWLEDGEMENT**

The project developed for the MCA was not possible without the persons and organizations that helped me in completing this. I am deeply grateful to all whose enthusiasm and energy transformed my vision of this study into reality.

I take this opportunity to thank my guide **Prof. Rupali Taware** Ma’am and our Director Ma’am **Dr. Chandrani Singh**, for encouragement and guidance throughout the progress of this report.

**Pooja Kothawade**



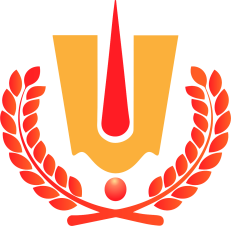


|  |  |  |
| --- | --- | --- |
| **Chapter No** |  | **Details** |
| **1** |  | **Introduction** |
|  | 1.1 | Company Profile / Institute Profile / Client Profile |
|  | 1.2 | Abstract |
|  | 1.3 | Existing System and Need for System |
|  | 1.4 | Scope of System |
|  | 1.5 | Operating Environment - Hardware and Software |
|  | 1.6 | Brief Description of Technology Used Operating systems used (Windows or Unix)  1.6.2 RDBMS/No Sql used to build database (MySQL/Oracle, Teradata, etc.) |
| **2** |  | **Proposed System** |
|  | 2.1 | Study of Similar Systems (If required research paper can be included) |
|  | 2.2 | Feasibility Study |
|  | 2.3 | Objectives of Proposed System |
|  | 2.4 | Users of System |
| **3** |  | **Analysis and Design** |
|  | 3.1 | System Requirements (Functional and Non-Functional requirements) |
|  | 3.2 | Entity Relationship Diagram (ERD) |
|  | 3.3 | Table Structure |
|  | 3.4 | Use Case Diagrams |
|  | 3.5 | Class Diagram |
|  | 3.6 | Activity Diagram |
|  | 3.7 | Sequence Diagram |
|  | 3.8 | Component Diagram |
|  | 3.9 | Data Dictionary |
| **4** |  | **User Manual** |
|  | 4.1 | Sample Input and Output Screen |
|  | 4.2 | Code snippets |
| **5** |  | **Testing** |
|  | 5.1 | Test Strategy |
|  | 5.2 | Unit Test Plan |
|  | 5.3 | Acceptance Test Plan |
|  | 5.4 | Test Case / Test Script |
|  | 5.5 | Defect report / Test Log |
| **6** |  | **Implementation and Software Specification Testing’s** |
| **7** |  | **Proposed Enhancements** |
| **8** |  | **Conclusion** |
| **9** |  | **Bibliography** |

**CHAPTER 1**

**INTRODUCTION**

* 1. **Company Profile**

 ****

**About Us: -**

Balaji Trading Company is a leading provider of high-quality grains and cereals, dedicated to serving a diverse range of industries with top-notch products and unparalleled service. Founded in 1995, Balaji Trading has grown to become a trusted name in the grain business, known for our commitment to quality, reliability, and customer satisfaction.

Our mission is to be the preferred partner for grains and cereals, offering premium products and exceptional service. We strive to maintain the highest standards of quality while fostering sustainable practices and contributing to the success of our clients.

We specialize in sourcing, processing, and distributing a wide variety of grains and cereals, including wheat, rice, corn, barley, and oats. Our comprehensive supply chain ensures that our products meet rigorous quality standards at every stage, from farm to distribution. We cater to clients across various industries, including food and beverage, retail, and hospitality.

* 1. **Abstract**

The CerealSphere project was developed to enhance the operations of Balaji Trading Company, a prominent business in the grains and cereals industry. This project aimed to create an integrated software platform that would streamline supply chain management, improve customer relations, optimize logistics, and provide comprehensive business analytics.The software will be developed as a website, boasting a user-friendly interface that streamlines the management process and eliminates redundancies. This intuitive interface will not only simplify operations but also enhance user experience, making it easier for customers to navigate through the store and place orders effortlessly.

The project employed the Agile methodology, facilitating continuous feedback and iterative development. The primary objectives were to improve efficiency, reduce costs, and enable data-driven decision-making across the company's operations.

The CerealSphere project represents a significant step forward for Balaji Trading Company, providing a solid foundation for future growth and customer satisfaction. By focusing on sustainable practices and customer-centric solutions, the project not only met but exceeded its initial objectives, setting a new standard in the grains and cereals industry.

Despite challenges such as data integration and user adoption, the project team successfully implemented the CerealSphere platform. The new system resulted in improved supply chain efficiency, enhanced customer service, reduced logistics costs, and greater business insights

* 1. **Existing System and Need of System**

Balaji Trading Company, a major player in the grains and cereals industry, previously operated with a mix of traditional methods and disparate software tools. The existing system had several key components, but it lacked integration, leading to inefficiencies and operational challenges.

Manual Processes: Many operations were manual, such as inventory tracking, order processing, and customer communication. This manual approach increased the risk of errors and slowed down the workflow.

Isolated Systems: The existing software tools used for different functions—like inventory management, logistics, and customer relationship management—were not integrated. This isolation made data sharing between departments cumbersome and prone to inconsistencies.

Lack of Real-Time Data: The lack of real-time data hindered the ability to make informed decisions. Managers often relied on outdated reports, leading to delays and missed opportunities.

Limited Analytics: The existing system offered minimal business analytics, making it challenging to gain insights into operational efficiency, customer trends, and market demands.

Inflexible Logistics: The transportation and logistics processes were rigid, leading to inefficiencies in routing and scheduling. This rigidity increased costs and extended delivery times.

Given these limitations, the need for an integrated system became apparent. The CerealSphere project aimed to address these issues and create a comprehensive platform to support Balaji Trading Company's growth and operational efficiency.

**Integrated Supply Chain Management**: A unified system was required to manage the entire supply chain, from grain sourcing to customer delivery. This integration would improve efficiency and reduce redundancy.

**Enhanced Customer Relationship Management**: A dedicated CRM module would streamline customer interactions, improve order tracking, and enable better customer service.

**Real-Time Data and Analytics:** The new system would provide real-time data and advanced analytics, allowing for data-driven decision-making and improved operational insights.

**Optimized Logistics:** The CerealSphere project needed to include a logistics optimization component to reduce transportation costs and delivery times, enabling more flexible routing and scheduling.

**Sustainability and Compliance:** With increasing focus on sustainability, the new system needed to support environmentally friendly practices and ensure compliance with industry regulations.

* 1. **Scope of System**

1. Supply Chain Management

**Inventory Tracking:** Implementation of a system to track grains and cereals from sourcing to distribution, including inventory levels, storage conditions, and quality control.

**Supplier Management:** Integration of supplier data to manage relationships, contracts, and performance.

**Order Processing:** Streamlining order management to ensure accurate and timely processing, reducing manual errors, and automating repetitive tasks.

2. Customer Relationship Management (CRM)

**Customer Database:** Development of a comprehensive database to manage customer information, preferences, and purchase history.

**Order Tracking:** Providing customers with real-time updates on order status, shipment tracking, and expected delivery times.

**Customer Communication:** Integration of communication tools for customer support and feedback.

3. Logistics Optimization

**Transportation Planning:** Implementation of a system to optimize transportation routes, schedules, and load planning, reducing costs and delivery times.

**Fleet Management:** Integration of fleet tracking and maintenance systems to ensure optimal performance and compliance with regulations.

**Warehouse Management:** Streamlining warehouse operations to improve storage efficiency and reduce handling times.

4. Business Analytics and Reporting

**Real-Time Data Analysis:** Integration of a business analytics platform to provide real-time insights into operations, sales trends, and customer behavior.

**Customized Reporting:** Development of customizable reports for various stakeholders, enabling data-driven decision-making.

**Performance Metrics:** Implementation of key performance indicators (KPIs) to track the success of various processes and identify areas for improvement.

5. User Training and Support

**Employee Training:** Providing comprehensive training for employees to ensure effective adoption of the new system.

**User Documentation:** Creating user-friendly manuals and documentation to assist with system use and troubleshooting.

**Technical Support:** Establishing a support system for ongoing maintenance and addressing technical issues.

6. Security and Compliance

**Data Security:** Ensuring robust security measures to protect sensitive data and maintain confidentiality.

**Compliance with Regulations:** Implementing processes to ensure compliance with industry standards and legal requirements.

**Disaster Recovery:** Establishing a disaster recovery plan to ensure business continuity in case of system failures or data loss.

7. Sustainability and Environmental Impact

**Sustainable Practices:** Incorporating environmentally friendly practices into supply chain and logistics operations.

**Waste Reduction:** Implementing processes to minimize waste and promote recycling.

* 1. **Operating Environment - Hardware and Software:**
* **Server Hardware Requirement:**
* Processor: Intel Core I3 or Above
* RAM: 4 GB or Above
* Storage: 512 GB or Above.
* **Server Software Requirement:**
* Operating System: Windows 07 or Above
* Web Browser: Internet Explorer, Google Chrome, etc.
* IDE: PyCharm Community or VS Code
* Front end: HTML, CSS, JS
* Framework: Django
* Backend: Python Shell
* Database: SQLite
* **Client Hardware Requirement:**
* Processor: Intel Core I3 or Above
* RAM: 4 GB or Above
* Storage: 512 GB or Above.
* **Client Software Requirement:**
* Operating System: Windows 07 or Above
* Web Browser: Internet Explorer, Google Chrome, etc
  1. **Brief Description of Technology Used:**

1. **PyCharm Community:**

PyCharm provides smart code completion, code inspections, on-the-fly error highlighting and quick-fixes, along with automated code refactoring’s and rich navigation capabilities. PyCharm’s smart code editor provides first-class support for Python, JavaScript, Coffee Script, TypeScript, CSS, popular template languages and more. Take advantage of language-aware code completion, error detection, and on-the-fly code fixes! Use smart search to jump to any class, file or symbol, or even any IDE action or tool window. It only takes one click to switch to the declaration, super method, test, usages, implementation, and more. Refactor your code the intelligent way, with safe Rename and Delete, Extract Method, Introduce Variable, Inline Variable or Method, and other refactoring’s. Language and framework-specific refactoring’s help you perform project-wide changes.

1. **Google Chrome:**

The Google Chrome Web browser is based on the open source Chromium project. Google released Chrome in 2008 and issues several updates a year. It is available for Windows, Mac OS X, Linux, Android and iOS operating systems. The Google Chrome browser takes a sandboxing-based approach to Web security. Each open website runs as its own process, which helps prevent malicious code on one page from affecting others (or the computer operating system at large). The browser also supports Web standards such as HTML5 and cascading style sheets (CSS).

1. **Django:**

Django was created in 2003 when web developers at the Lawrence Journal-World newspaper started using Python for their web development. After creating a number of websites, they started to factor out and reuse lots of common code and design patterns. That common code led to a generic web development framework that was open-sourced as the “Django” project in 2005. Since the original developers were surrounded by those newspaper writers, well-written documentation is a key part of Django. This means that there are excellent references to check out on the official Django documentation pages.

1. **The Django community**

The Django framework is extremely large, but the Django community is massive. The community has contributed a lot of third-party code for Django. No matter what we are trying to do, there is a good chance that we will find the solution for it on djangopackages.org. The website includes everything from authentication and authorization to full-on Django-powered content management systems, from e- commerce add-ons to integrations with Stripe.

1. **Django features**

Some features that make Django an ideal framework for web application development are as follows:

* Super-fast: Django development is extremely fast. Our ideas can take the shape of a product very quickly.
* Fully loaded: Django has dozens of projects that can be integrated to carry out common tasks such as user authentication, authorization, and content administration.
* Versatile: Django can be used for almost any kind of project, from CMSs to e- commerce apps to on-demand delivery platforms.
* Secure: Django also has support to prevent common security issues, including cross-site request forgery, cross-site scripting, SQL injection, and clickjacking.
* Scalable: Django websites can scale fast to meet high traffic demands.

1. **HTML/CSS**

HTML (the Hypertext Markup Language) and CSS (Cascading Style Sheets) are two of the core technologies for building Web pages. HTML provides the structure of the page, CSS the (visual and aural) layout, for a variety of devices. Along with graphics and scripting, HTML and CSS are the basis of building Web pages and Web Applications

1. **What is HTML?**

HTML is the language for describing the structure of Web pages. authors the means to: Publish online documents with headings, text, tables, lists, photos, etc. Retrieve online information via hypertext links, at the click of a button. Design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc. Include spread-sheets, video clips, sound clips, and other applications directly in their documents. With HTML, authors describe the structure of pages using markup.

1. **What is XHTML?**

XHTML is a variant of HTML that uses the syntax of XML, the Extensible Markup Language. XHTML has all the same elements (for paragraphs, etc.) as the HTML variant, but the syntax is slightly different. Because XHTML is an XML application, you can use other XML tools with it (such as XSLT, a language for transforming XML content).

1. **What is CSS?**

CSS is the language for describing the presentation of Web pages, including colours, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language.

**CHAPTER 2**

**PROPOSED SYSTEM**

* 1. **Feasibility Study**

Feasibility study aim to objectively rotationally uncover the strengths and weakness of the existing business purpose venture, opportunities and threats as presented by the environment, the resource required to carry through, and ultimately the prospectus for success in its simplest term, the two criteria to judge feasibility cost required and value to be attend. As such, well designed feasibility study should provide the historical background of the business of project, descriptions of the product or service, accounting statement, details of the operations and management, marketing research and policies financial data, legal requirements, text obligation.

* + - **ECONOMIC FEASIBILITY**

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/ benefit analysis, the procedure is to determine the benefits and the saving that are accepted from a candidate system and compare them with costs. If benefits outweigh cost, then the decision is made to design and implement the system. And entrepreneur must accurately weigh the cost versus benefits before taking an action.

* + - * Cost Based Study: It is important to identity cost and benefit factor which can be categories as follows:
* Development costs
* Operating costs
  + - **TECHNICAL FEASIBILITY**

The assessment is based on outline design of system requirements in term of input, processes, output, fields, programs and procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. in order to estimate whether the new system will perform adequately or not. Technical feasibility is carried out to determine whether the company has the capability in terms of software, hardware, personal and expertise, to handle the completion of the project.

* + - * A brief description of the business to assess more possible factors which could affect the study
      * The part of business being examined.
      * The human and economic factor.
      * The possible solution to the problems.
      * At this level, the concern is whether the proposal is both technically and legally feasible.
    - **BEHAVIORAL FEASIBILITY**

Proposed project of beneficiary only if they turned into information system that will meet organization operating requirement. Simply stated this test of feasibility asks if the system will work when it is developed and installed. Which is major barrier for implementation? Here is question that will help tested operation feasibility of project.

Is current business method acceptable to the user? If they are not, user may welcome the change that will about more operation and useful system.

* 1. **Objective of System**

The primary aim of the online shopping site is to streamline and automate the management of various aspects of the shopping experience, including transactions, payments, invoices, and customer interactions. Built exclusively for administrative use, the project restricts access to authorized personnel, ensuring data integrity and security.

By centralizing and digitizing key functions, the site effectively minimizes manual intervention, thereby enhancing efficiency and accuracy in operations. It serves as a comprehensive repository for tracking and recording all pertinent information related to payments, customers, orders, and invoices, providing administrators with real-time insights into the status and progress of transactions.

Through intuitive user interfaces and robust backend functionalities, the site empowers administrators to effortlessly navigate and manage a multitude of tasks, from processing orders and generating invoices to monitoring payment statuses and analyzing customer trends. This seamless integration of data and processes not only saves time and resources but also mitigates the risk of errors and discrepancies inherent in manual record-keeping.

Furthermore, the site facilitates seamless communication and collaboration between different departments, enabling stakeholders to access relevant information and updates in a timely manner. This fosters greater transparency and accountability across the organization, ultimately leading to improved decision- making and customer service.

* 1. **Purpose of This System**

The CerealSphere system is designed to transform and enhance the operations of Balaji Trading Company, a major player in the grains and cereals industry. The purpose of this system is to create a comprehensive, integrated platform that addresses key challenges, streamlines business processes, and improves overall efficiency. The system's ultimate aim is to provide a robust framework for growth, customer satisfaction, and sustainability. Here's a detailed breakdown of the purpose of the CerealSphere system:

1. Integration of Business Operations

The primary purpose of CerealSphere is to unify the various aspects of Balaji Trading Company's operations into a cohesive system. This integration eliminates data silos, reduces manual processes, and enhances communication across different departments, leading to a more efficient workflow.

2. Streamline Supply Chain and Logistics

CerealSphere is designed to streamline supply chain management and logistics. This includes optimizing inventory levels, improving supplier coordination, and reducing transportation costs through better logistics planning. By achieving these efficiencies, the system aims to enhance Balaji Trading Company's operational effectiveness.

3. Enhance Customer Relationship Management (CRM)

The system's purpose includes improving customer relationships by providing tools for effective communication, order tracking, and customer support. By centralizing customer data and offering personalized services, CerealSphere aims to enhance customer satisfaction and build stronger customer loyalty.

4. Provide Business Analytics and Insights

CerealSphere aims to empower Balaji Trading Company with real-time data and business analytics capabilities. This allows stakeholders to make data-driven decisions, identify trends, and optimize business processes. The system's purpose is to provide insights that support strategic planning and continuous improvement.

5. Ensure Security and Compliance

An essential purpose of CerealSphere is to ensure robust security and compliance with industry regulations. This includes protecting sensitive data, adhering to legal requirements, and maintaining a high level of data privacy. The system's focus on security is crucial for maintaining customer trust and business integrity.

6. Promote Sustainability and Social Responsibility

CerealSphere is designed with sustainability in mind. The system encourages sustainable practices, such as reducing waste, improving energy efficiency, and supporting environmentally friendly logistics. Additionally, the purpose of the system includes promoting social responsibility and community engagement, aligning with broader corporate values.

In summary, the purpose of the CerealSphere system is to create a comprehensive platform that improves operational efficiency, enhances customer satisfaction, and promotes sustainability. By achieving these objectives, the system supports Balaji Trading Company's growth and strengthens its competitive position in the grains and cereals industry. The system's focus on integration, customer-centric solutions, and data-driven insights provides a solid foundation for future success.

1. Top of Form

## **Features of CerealSphere Website: -**

* Login and registration facility.
* Shopping cart facility.
* Managing and ordering products.
* The transaction is done in on-line mode.
* Shows description of orders, payment and product.
* Increase the efficiency of managing shopping.
* Give feedback to the seller
* Check the weekly and monthly report of the profit
* Block or unblock the seller
* Manage stocks of the products
* Download Invoice
* Cancel Order
* Ask for Enquiry
* Order Status
* Payment Gateway
* Check all the order placed
  1. **Module Specification**
     + **Product Module:**

The main aim to develop this module is to provide detailed information of products. This module contains product description, details like price, quantity, photos of products. This module gives clear idea about product.

* + - **Payment Module:**

In this module user can do the payment of the purchased product himself by providing required information. Once Payment done then user get the mail of order accepted from the admin.

* + - **Customer Module:**

This module contains customer information like his details, login information. main objective of this module is to provide all functionality related to customer. customer can edit, update, delete his information. He can create his profile on site. Can add multiple addresses for his convenience.

* + - **Order Module:**

This module of site performs main task that is order summary once customer choose what he wants to buy then customer have to add that product to cart and then order will be processed. objective of this module is to maintain all orders and order details.

* + - **Admin Module:**

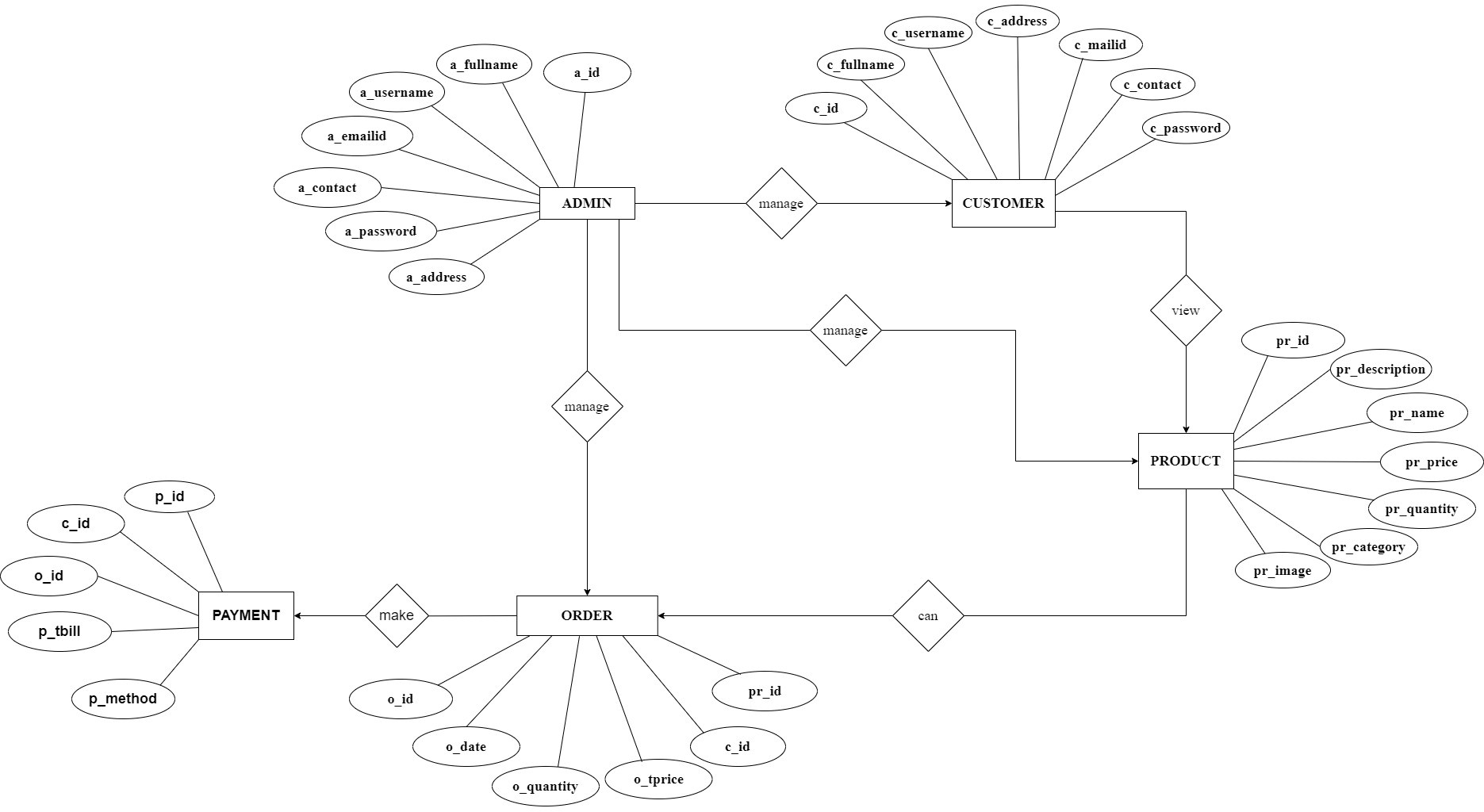
Admin can add update and delete the products according to their availability. Admin can manage the orders. He can change the status of ordered product.

* + - **Feedback Module:**

The feedback module in an e-commerce website. By following the instructions outlined here, customers can effectively submit feedback, read reviews, and engage in meaningful discussions. The feedback module serves as a valuable tool for businesses to gather insights, improve their products and services, and foster customer satisfaction.

**CHAPTER 3**

**ANALYSIS AND DESIGN**

* 1. **System Requirements (Functional & Non-Functional Requirements): -**
     + **Functional Requirement: -**
* Customer can login
* Customer can review on product
* Customer Give the feedback on provide service
* Admin can view Feedback
* Customer can view the delivery details
* Admin check the delivery status
* Customer select payment option
* Customer can payment for product
* Admin check payment of product
* Customer Buy product
* Customer view order
* Customer cancel order
* Admin can view order
* Customer Add product to cart
* Customer Remove product from cart
* Customer update the number of quantities of product in cart
* Admin add product
* View product
* Update product
* Delete product
* Admin can add category (e.g., Solar Panel, Solar Batteries etc.)
* Admin can delete category
* Admin can add sub-category (add parts of Solar Panel category wise)
* Customer can registration
* Customer can Login
* Customer can View Profile
* Customer can Update Profile
* Customer can Logout
* Admin can registration
* Admin can Login
* Admin can View profile
* Admin can Update profile
* Admin can View customer profile
* Admin can Update customer profile
* Admin can Delete Customer profile
* Admin can Logout
  + - **Non-Functional Requirement: -**
* Interoperability: Ability to integrate with other systems, tools, or technologies.
* Testability: Ability to be tested effectively, including unit testing, integration testing, and system testing.
* Maintainability: Ease of making changes or updates to the system without introducing errors or disrupting existing functionality.
* Scalability: Ability to handle increased workloads or users without negatively impacting performance or functionality.
* Extensibility: Ability to add new features or functionality to the system without requiring significant changes to the existing code.
* Usability: Ability of the system to be easily understood and used by its intended audience.
* Accessibility: Ability of the system to be used by people with disabilities, such as those with visual or hearing impairments.
* Portability: Ability to run the system on different platforms or environments without requiring significant modifications.
* Compatibility: Ability of the system to work with different hardware or software configurations.
* Compliance: Ability of the system to meet legal, regulatory, or industry standards and requirements.
* Performance: Non-functional requirements related to performance can include factors such as response time, throughput, and capacity. It's important to consider the performance requirements of the system in relation to the expected workload or usage patterns.
* Security: Non-functional requirements related to security can include authentication, authorization, confidentiality, and integrity. It's important to consider the security requirements of the system in relation to the sensitivity of the data being handled and potential risks or threats.
* Documentation: Non-functional requirements related to documentation can include the level of detail and completeness required for documentation, as well as standards for formatting and organization.
  1. **UML Diagrams**
     + **Entity Relationship Diagram**
  2. **Table specifications**

**ADMIN TABLE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.no | Field name | Field size | Data type | Description | Constraint |
| 1 | a\_id | 8 | Int | Admin id | Primary key |
| 2 | a\_username | 10 | Varchar | Admin username | Not null |
| 3 | a\_password | 12 | Varchar | Admin password | Not null |
| 4 | a\_name | 12 | Varchar | Admin name | Not null |
| 5 | a\_email | 27 | Varchar | Admin email | Not null |
| 6 | a\_contact | 11 | Int | Admin contact | Not null |

**CUSTOMER TABLE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.no | Field name | Field size | Data type | Description | Constraint |
| 1 | c\_id | 8 | Int | Customer id | Primary key |
| 2 | c\_username | 10 | Varchar | Customer username | Not null |
| 3 | c\_password | 12 | Varchar | Customer password | Not null |
| 4 | c\_name | 12 | Varchar | Customer name | Not null |
| 5 | c\_email | 27 | Varchar | Customer email | Not null |
| 6 | c\_contact | 11 | Int | Driver contact | Not null |
| 7 | c\_address | 10 | Varchar | Customer address | Not null |
| 8 | c\_photo | 250 | Image | Customer photo | Not null |

**PRODUCT TABLE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.no | Field name | Field size | Data type | Description | Constraint |
| 1 | p\_id | 8 | Int | Product id | Primary key |
| 2 | p\_name | 10 | Varchar | Product name | Not null |
| 3 | p\_desc | 10 | Varchar | Product Description | Not null |
| 4 | p\_category | 10 | Varchar | Product Category | Not null |
| 5 | p\_qty | 10 | Int | Product Quantity | Not null |
| 6 | p\_price | 10 | Int | Product Price | Not null |
| 7 | p\_maxprice | 10 | Int | Product Maxprice | Not null |
| 8 | p\_details | 10 | Varchar | Product Details | Not null |
| 9 | p\_brand | 10 | Varchar | Product brand | Not null |

**CART TABLE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.no | Field name | Field size | Data type | Description | Constraint |
| 1 | ct\_id | 8 | Int | Cart id | Primary key |
| 2 | c\_id | 8 | Int | Customer Id | Foreign Key |
| 3 | p\_id | 10 | Int | Product Id | Foreign Key |
| 4 | p\_details | 10 | Varchar | Product Details | Not null |
| 5 | p\_qty | 10 | Int | Product Quantity | Not null |

**FEEDBACK TABLE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.no | Field name | Field size | Data type | Description | Constraint |
| 1 | f\_id | 8 | Int | Feedback Id | Primary key |
| 2 | f\_desc | 8 | Varchar | Feedback Description | Not null |
| 3 | p\_details | 10 | Varchar | Product Details | Not null |
| 4 | c\_id | 10 | Int | Customer Id | Foreign Key |
| 5 | p\_id | 10 | Int | Product Id | Foreign Key |

**ORDER TABLE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.no | Field name | Field size | Data type | Description | Constraint |
| 1 | o\_id | 8 | Int | Order id | Primary key |
| 2 | c\_id | 10 | Int | Customer id | Foreign Key |
| 3 | p\_details | 10 | Varchar | Product details | Not null |
| 4 | p\_qty | 10 | Int | Product Quantity | Not null |
| 5 | t\_id | 10 | Int | Transaction Id | Not null |
| 6 | o\_date | 10 | Date | Order date | Not null |
| 7 | p\_price | 10 | Int | Product Price | Not null |

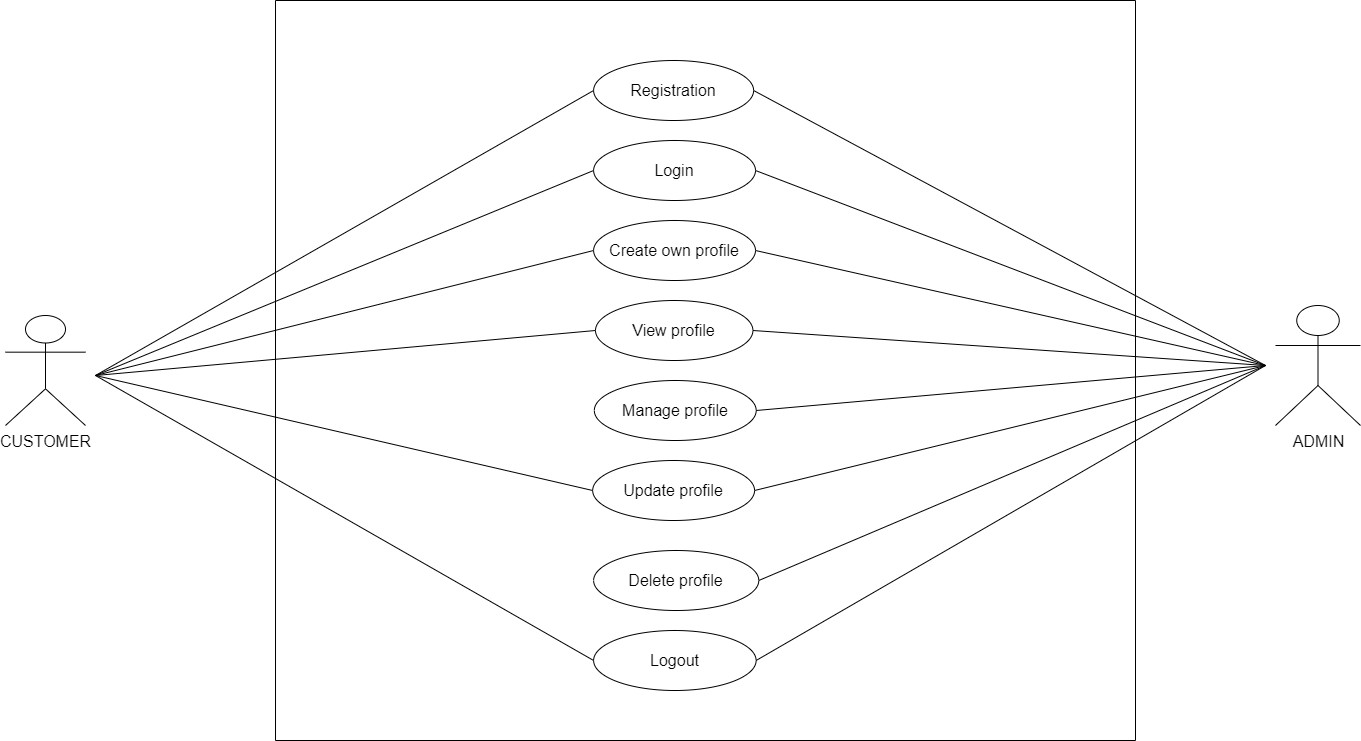
**PAYMENNT AND DELIVERY TABLE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.no | Field name | Field size | Data type | Description | Constraint |
| 1 | pay\_id | 8 | Int | Payment id | Primary key |
| 2 | p\_name | 8 | Varchar | Product name | Not null |
| 3 | p\_details | 10 | Varchar | Product details | Not null |
| 4 | p\_category | 10 | Varchar | Product category | Not null |
| 5 | p\_price | 10 | Int | Product price | Not null |
| 6 | p\_qty | 10 | Int | Product Quantity | Not null |
| 7 | pay\_transid | 10 | Int | Payment Transaction Id | Not null |
| 8 | pay\_mode | 20 | Varchar | Payment Mode | Not null |
| 9 | Payment\_status | 8 | Varchar | Payment Status | Not null |
| 10 | Delivery\_status | 8 | Varchar | Delivery Status | Not null |

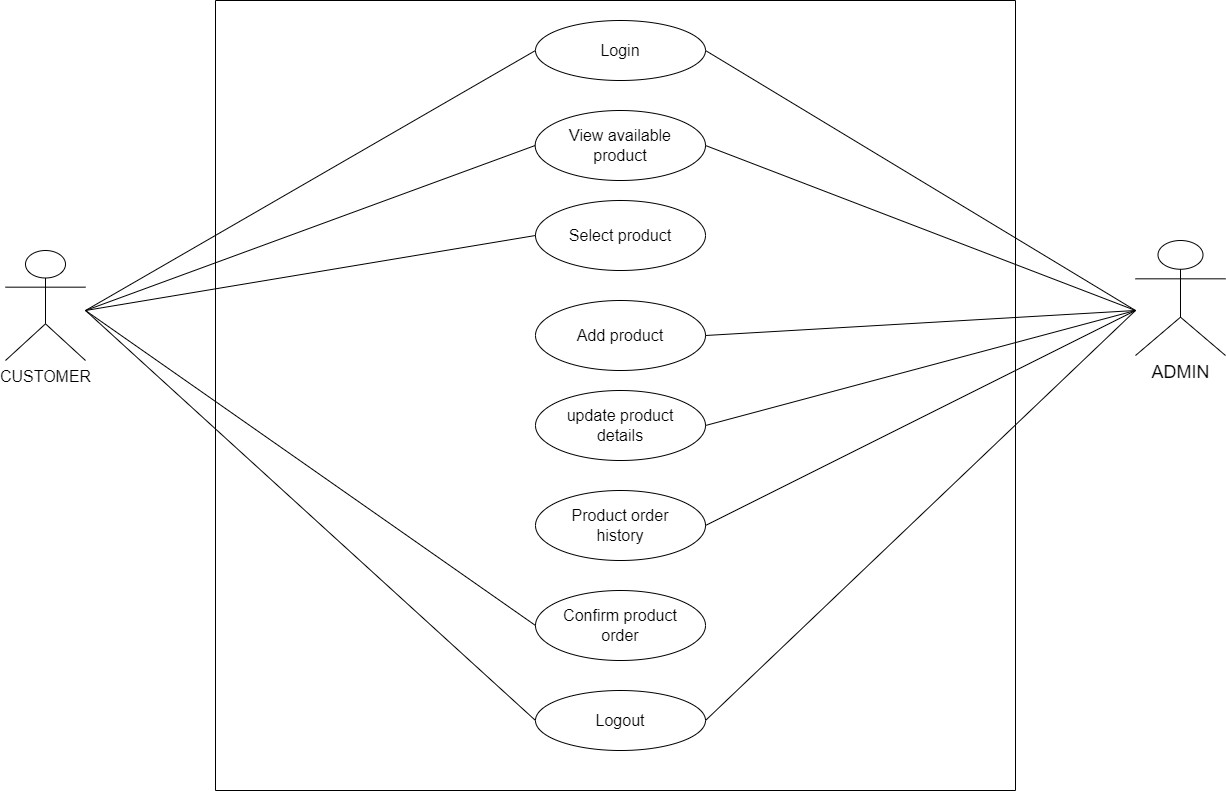
**Data Dictionary:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr.n o | Field Name | Field size | Data type | Description | Constrains | Table Name |
| 1 | a\_contact | 11 | Int | Admin contact | Not null | Admin |
| 2 | a\_email | 27 | Varchar | Admin email | Not null | Admin |
| 3 | a\_id | 8 | Int | Admin id | Primary key | Admin |
| 4 | a\_name | 12 | Varchar | Admin name | Not null | Admin |
| 5 | a\_password | 12 | Varchar | Admin password | Not null | Admin |
| 6 | a\_username | 10 | Varchar | Admin username | Not null | Admin |
| 7 | c\_address | 10 | Varchar | Customer address | Not null | Customer |
| 8 | c\_contact | 11 | Int | Driver contact | Not null | Customer |
| 9 | c\_email | 27 | Varchar | Customer email | Not null | Customer |
| 10 | c\_id | 8 | Int | Customer id | Primary key | Customer |
| 11 | c\_id | 8 | Int | Customer Id | Foreign Key | Cart |
| 12 | c\_id | 10 | Int | Customer id | Foreign Key | Order |
| 13 | c\_id | 10 | Int | Customer Id | Foreign Key | Feedback |
| 14 | c\_name | 12 | Varchar | Customer name | Not null | Customer |
| 15 | c\_password | 12 | Varchar | Customer password | Not null | Customer |
| 16 | c\_photo | 250 | Image | Customer photo | Not null | Customer |
| 17 | c\_username | 10 | Varchar | Customer username | Not null | Customer |
| 18 | ct\_id | 8 | Int | Cart id | Primary key | Cart |
| 19 | Delivery\_status | 8 | Varchar | Delivery Status | Not null | Payment & Delivery |
| 20 | f\_desc | 8 | Varchar | Feedback Description | Not null | Feedback |
| 21 | f\_id | 8 | Int | Feedback Id | Primary key | Feedback |
| 22 | o\_date | 10 | Date | Order date | Not null | Order |
| 23 | o\_id | 8 | Int | Order id | Primary key | Order |
| 24 | p\_brand | 10 | Varchar | Product brand | Not null | Product |
| 25 | p\_category | 10 | Varchar | Product Category | Not null | Product |
| 26 | p\_category | 10 | Varchar | Product category | Not null | Payment & Delivery |
| 27 | p\_desc | 10 | Varchar | Product Description | Not null | Product |
| 28 | p\_details | 10 | Varchar | Product Details | Not null | Product |
| 29 | p\_details | 10 | Varchar | Product Details | Not null | Cart |
| 30 | p\_details | 10 | Varchar | Product details | Not null | Order |
| 31 | p\_details | 10 | Varchar | Product details | Not null | Payment & Delivery |
| 32 | p\_details | 10 | Varchar | Product Details | Not null | Feedback |
| 33 | p\_id | 8 | Int | Product id | Primary key | Product |
| 34 | p\_id | 10 | Int | Product Id | Foreign Key | Cart |
| 35 | p\_id | 10 | Int | Product Id | Foreign Key | Feedback |
| 36 | p\_maxprice | 10 | Int | Product Maxprice | Not null | Product |
| 37 | p\_name | 10 | Varchar | Product name | Not null | Product |
| 38 | p\_name | 8 | Varchar | Product name | Not null | Payment & Delivery |
| 39 | p\_price | 10 | Int | Product Price | Not null | Product |
| 40 | p\_price | 10 | Int | Product Price | Not null | Order |
| 41 | p\_price | 10 | Int | Product price | Not null | Payment & Delivery |
| 42 | p\_qty | 10 | Int | Product Quantity | Not null | Product |
| 43 | p\_qty | 10 | Int | Product Quantity | Not null | Cart |
| 44 | p\_qty | 10 | Int | Product Quantity | Not null | Order |
| 45 | p\_qty | 10 | Int | Product Quantity | Not null | Payment & Delivery |
| 46 | pay\_id | 8 | Int | Payment id | Primary key | Payment & Delivery |
| 47 | pay\_mode | 20 | Varchar | Payment Mode | Not null | Payment & Delivery |
| 48 | pay\_transid | 10 | Int | Payment Transaction Id | Not null | Payment & Delivery |
| 49 | Payment\_status | 8 | Varchar | Payment Status | Not null | Payment & Delivery |
| 50 | t\_id | 10 | Int | Transaction Id | Not null | Order |

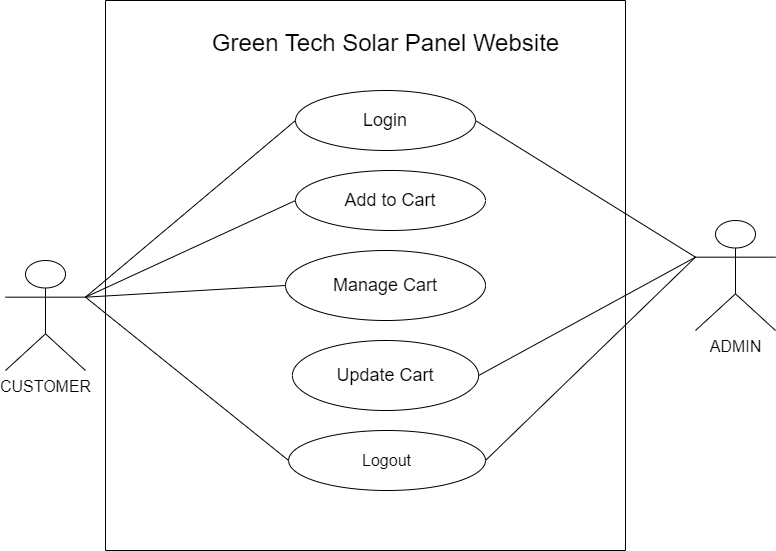
* 1. **Use Case Diagram**
     + **USE CASE OF PROFILE**



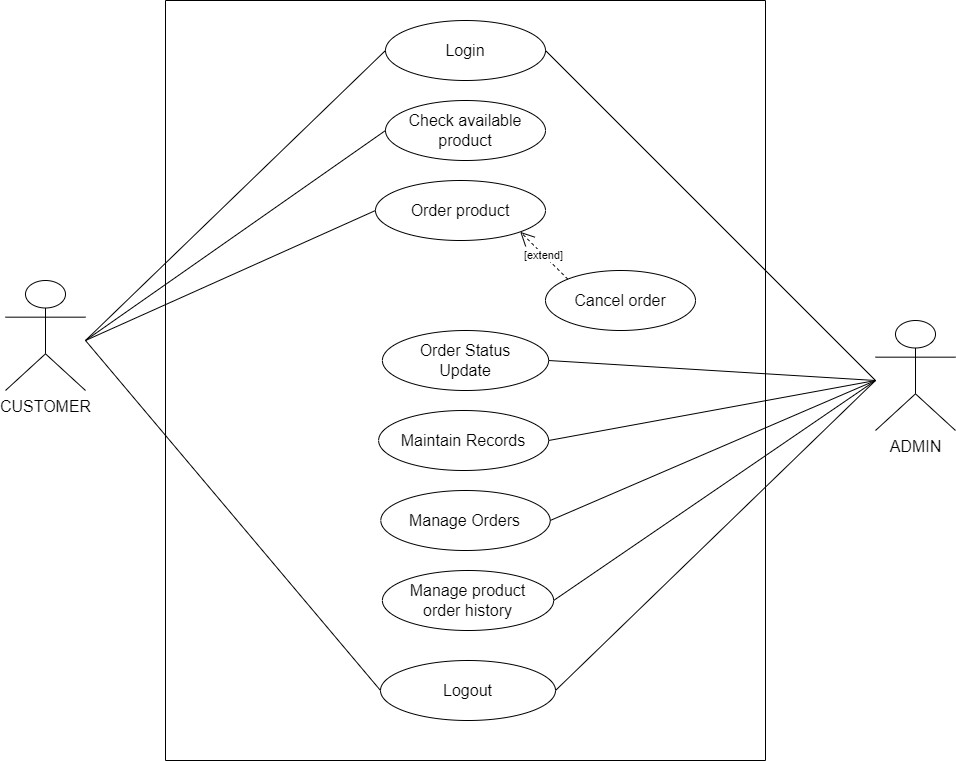
* + - **USE CASE OF PRODUCT MODULE**

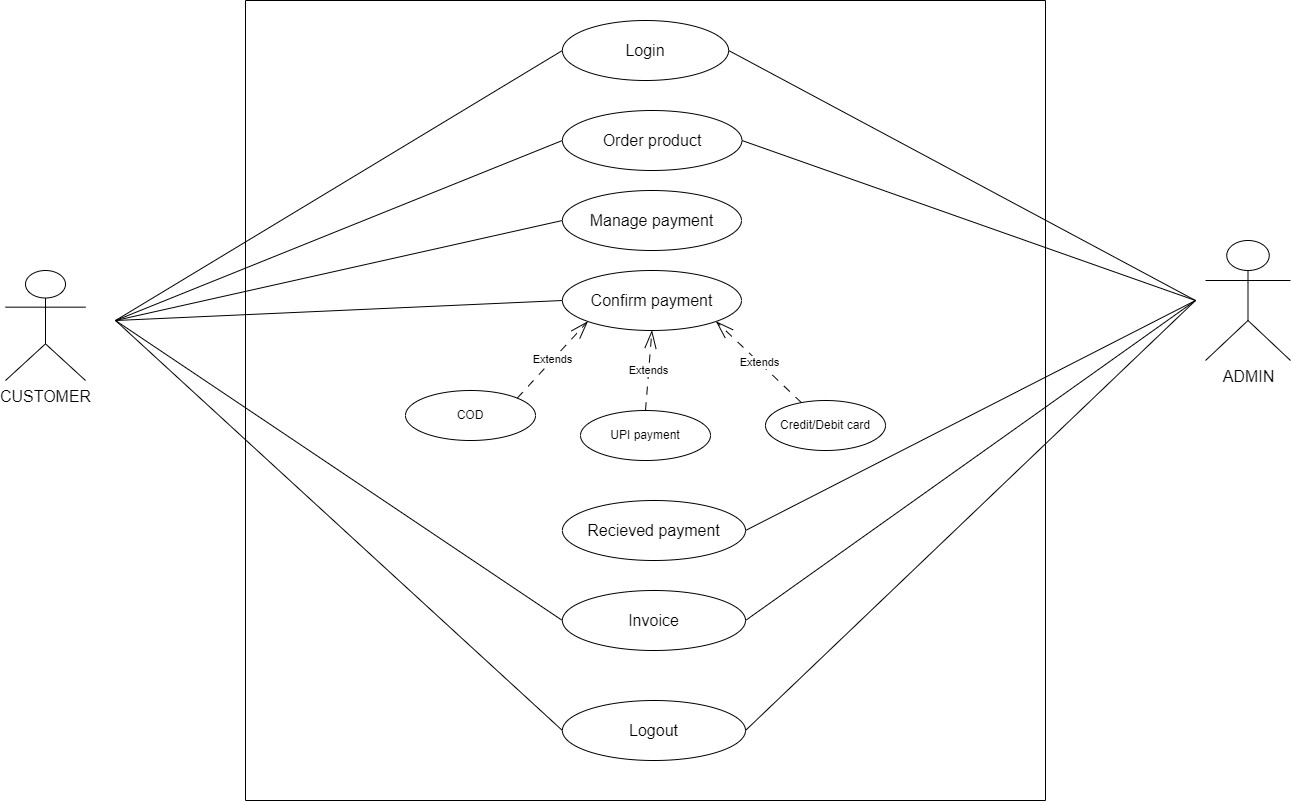


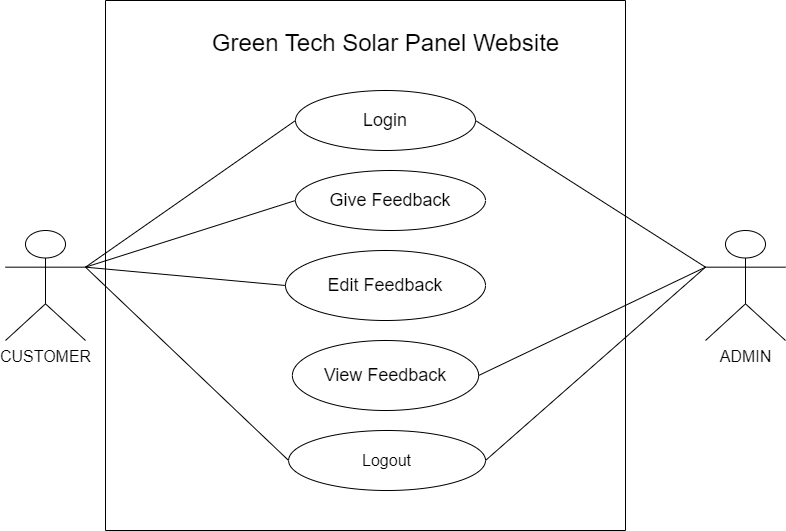
* + - **USE CASE OF CART MODULE**

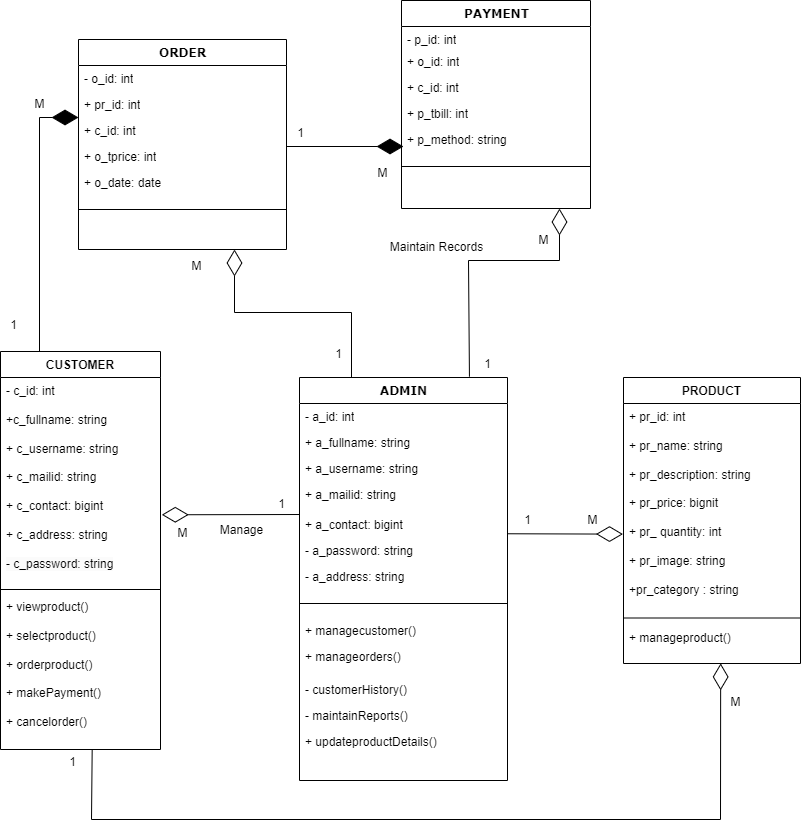


* + - **USE CASE OF ORDER MODULE**

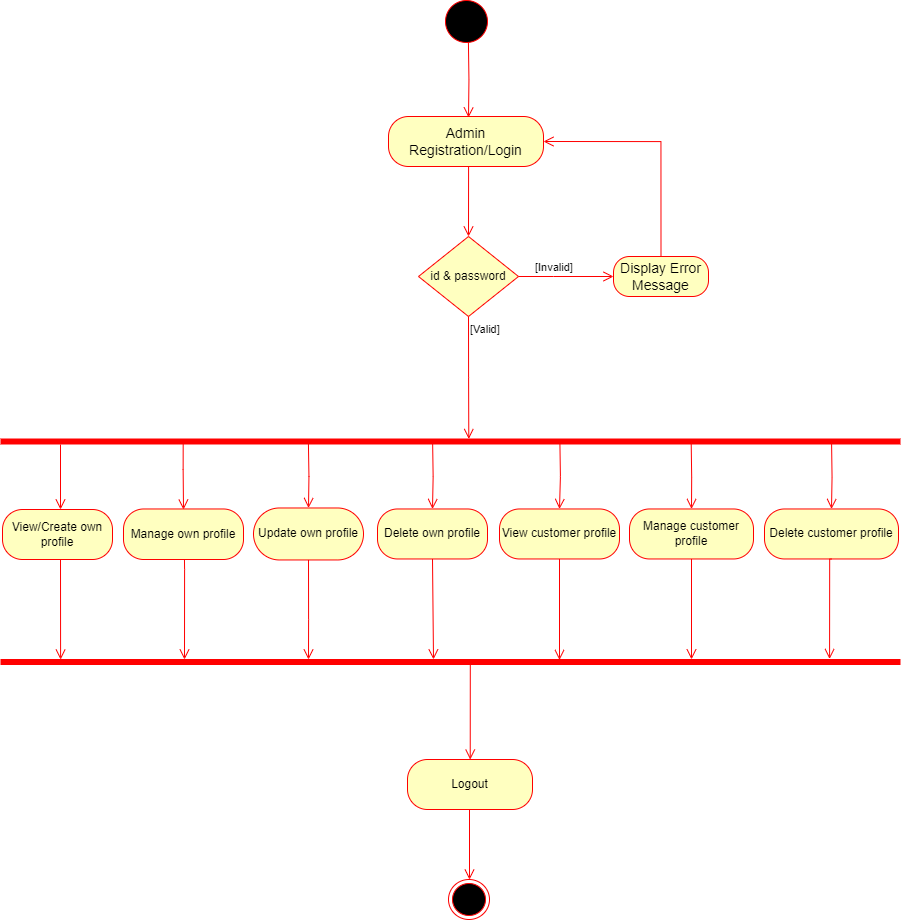


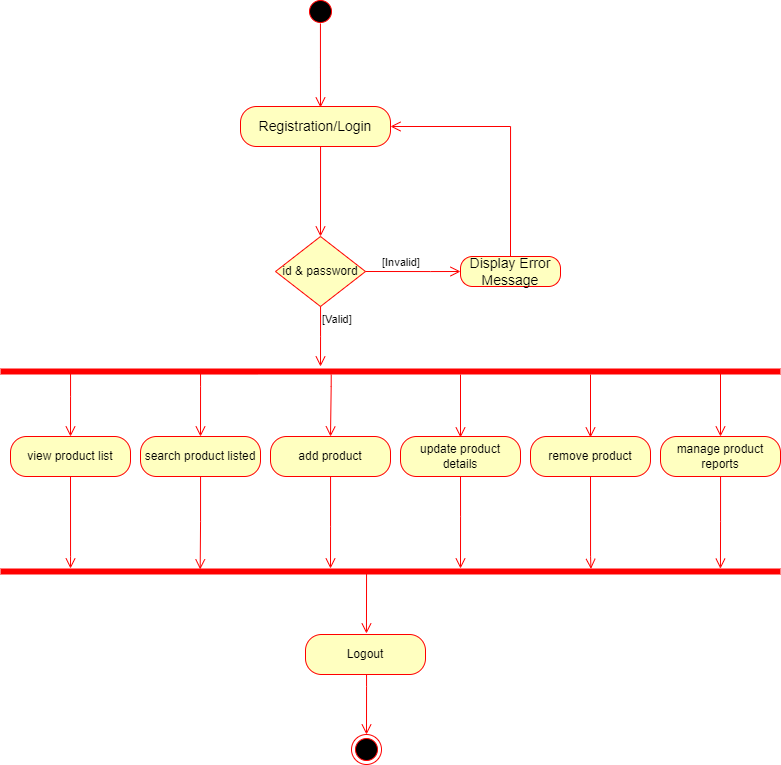
* + - **USE CASE OF PAYMENT & DELIVERY MODULE**
    - **USE CASE OF FEEDBACK MODULE**

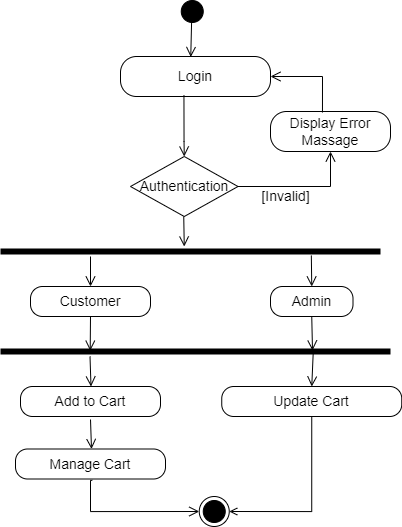


**3.5 CLASS DIAGRAM:**

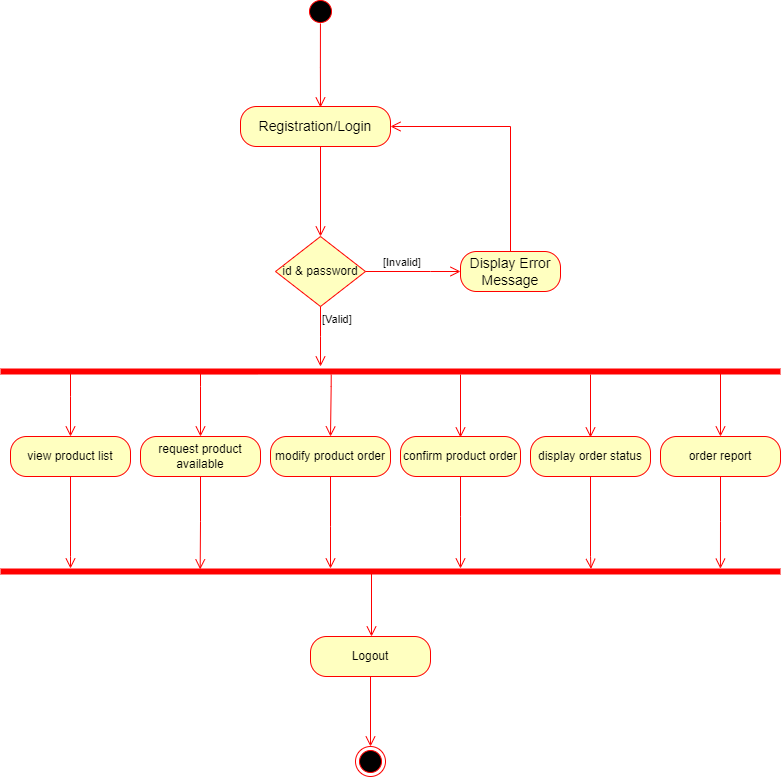
* 1. **ACTIVITY DIAGRAM:**
     1. **ACTIVITY DIAGRAM OF PROFILE**

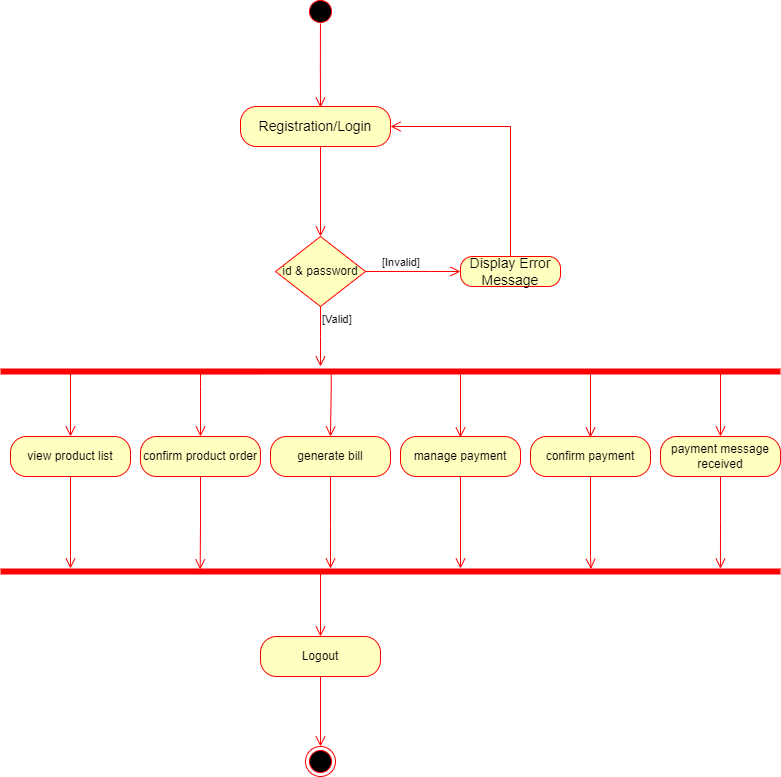


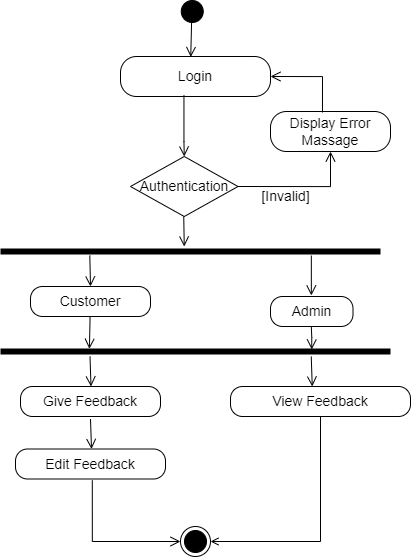
* + 1. **ACTIVITY DIAGRAM OF PRODUCT MODULE**
    2. **ACTIVITY DIAGRAM OF CART MODULE**



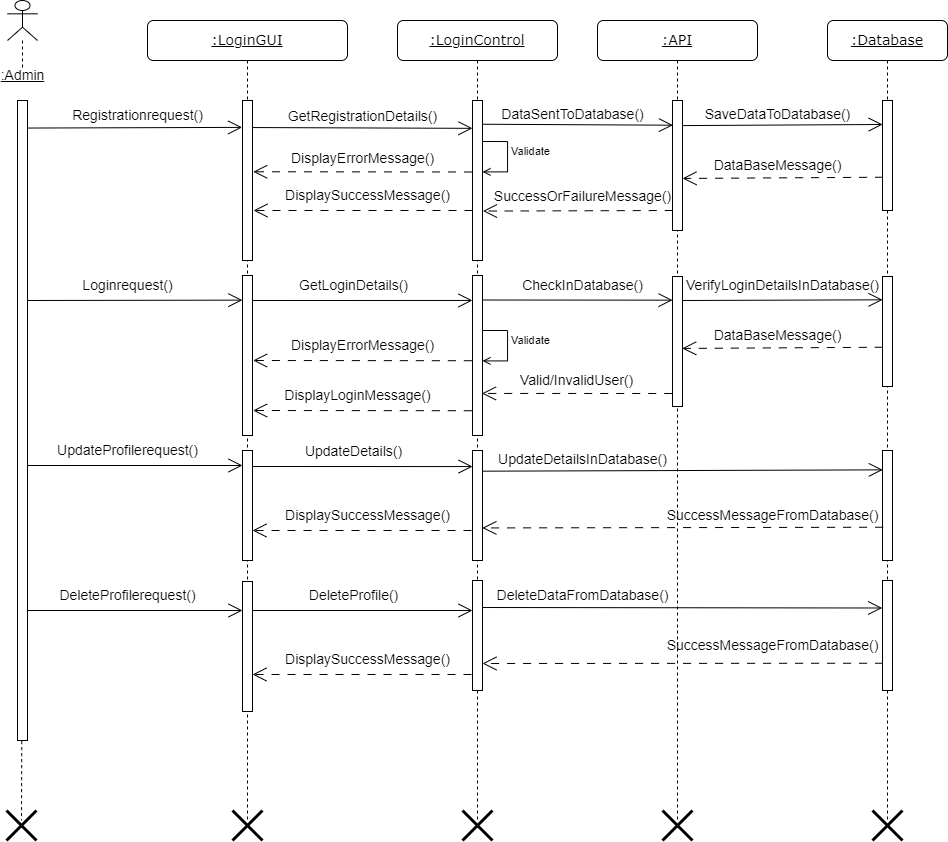
* + 1. **ACTIVITY DIAGRAM OF ORDER MODULE**

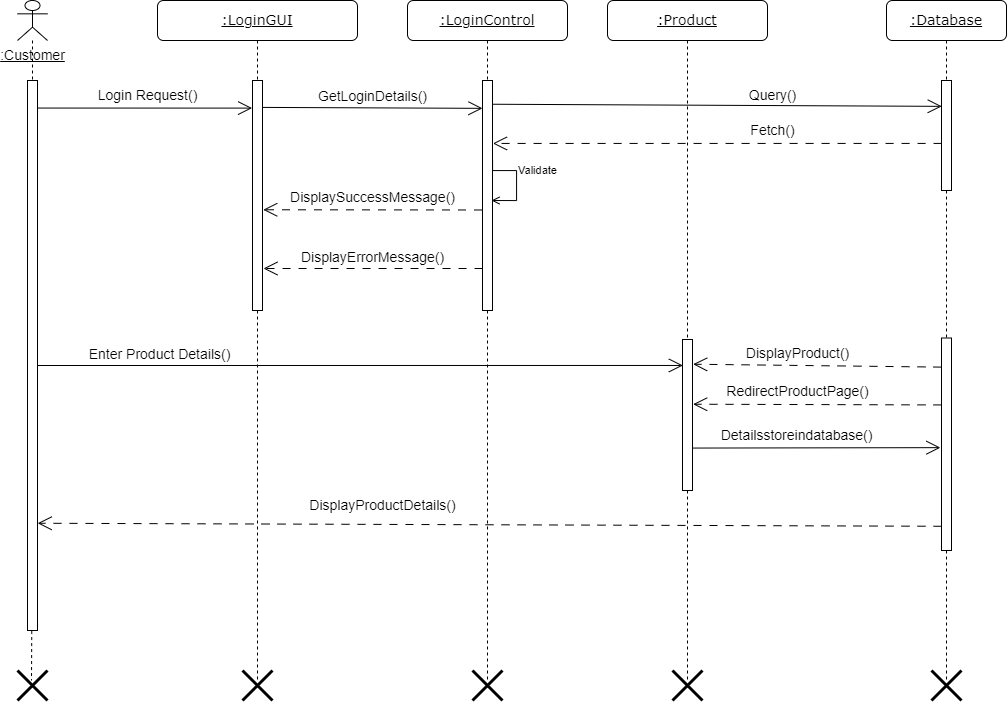


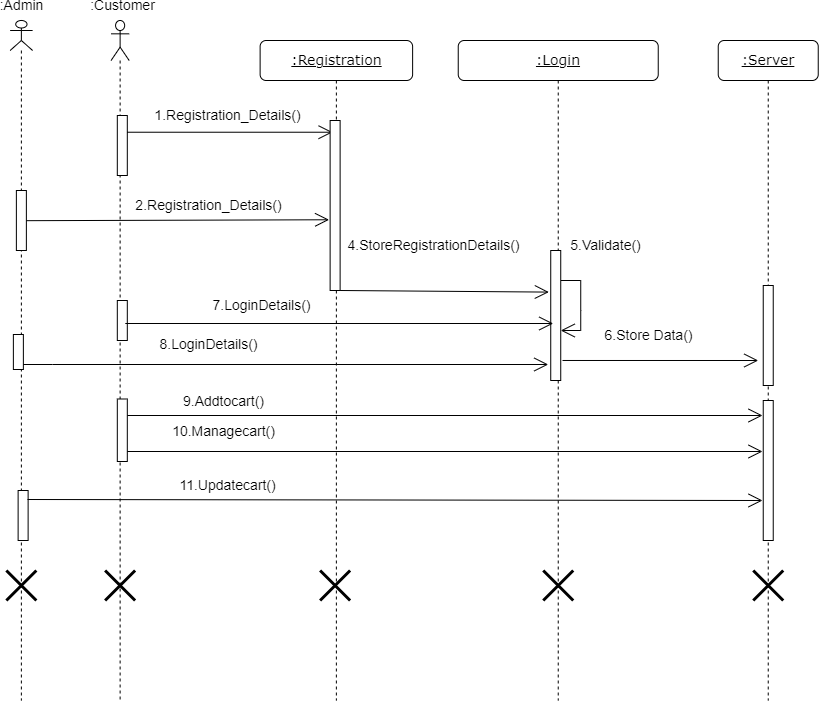
* + 1. **ACTIVITY DIAGRAM OF PAYMENT & DELIVERY MODULE**
    2. **ACTIVITY DIAGRAM OF FEEDBACK MODULE**

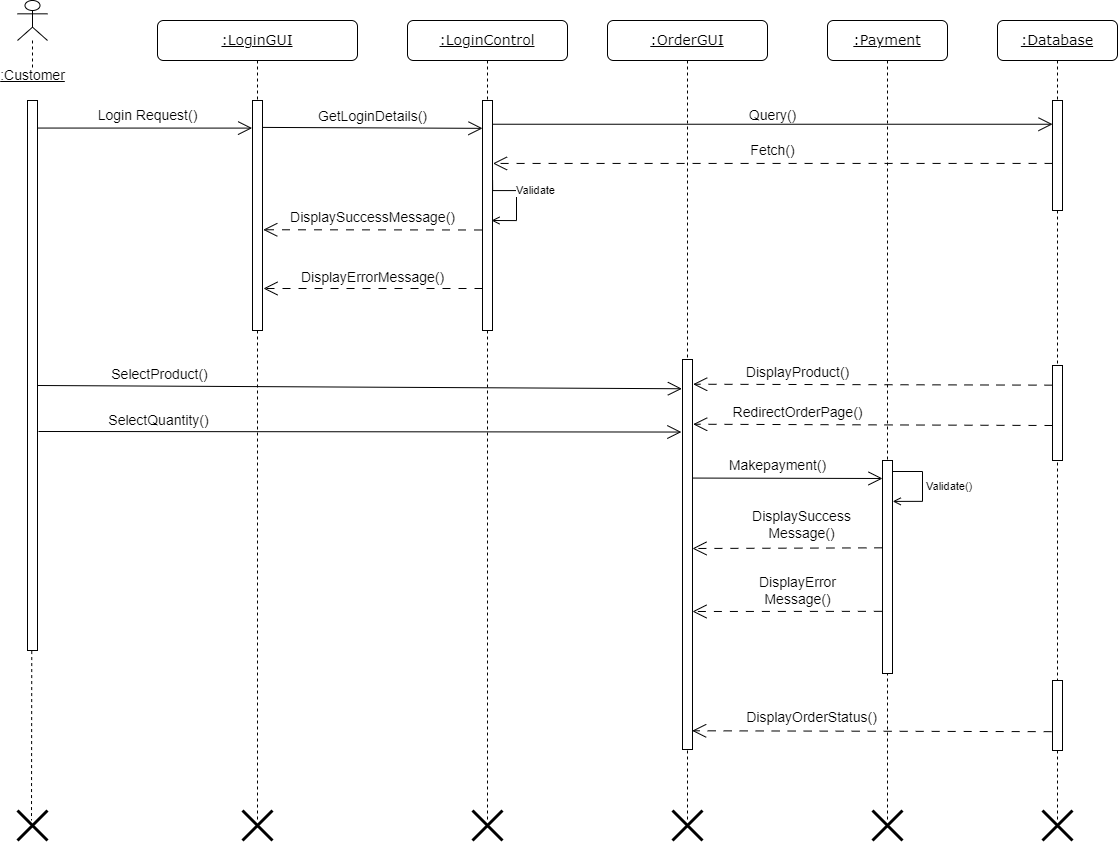
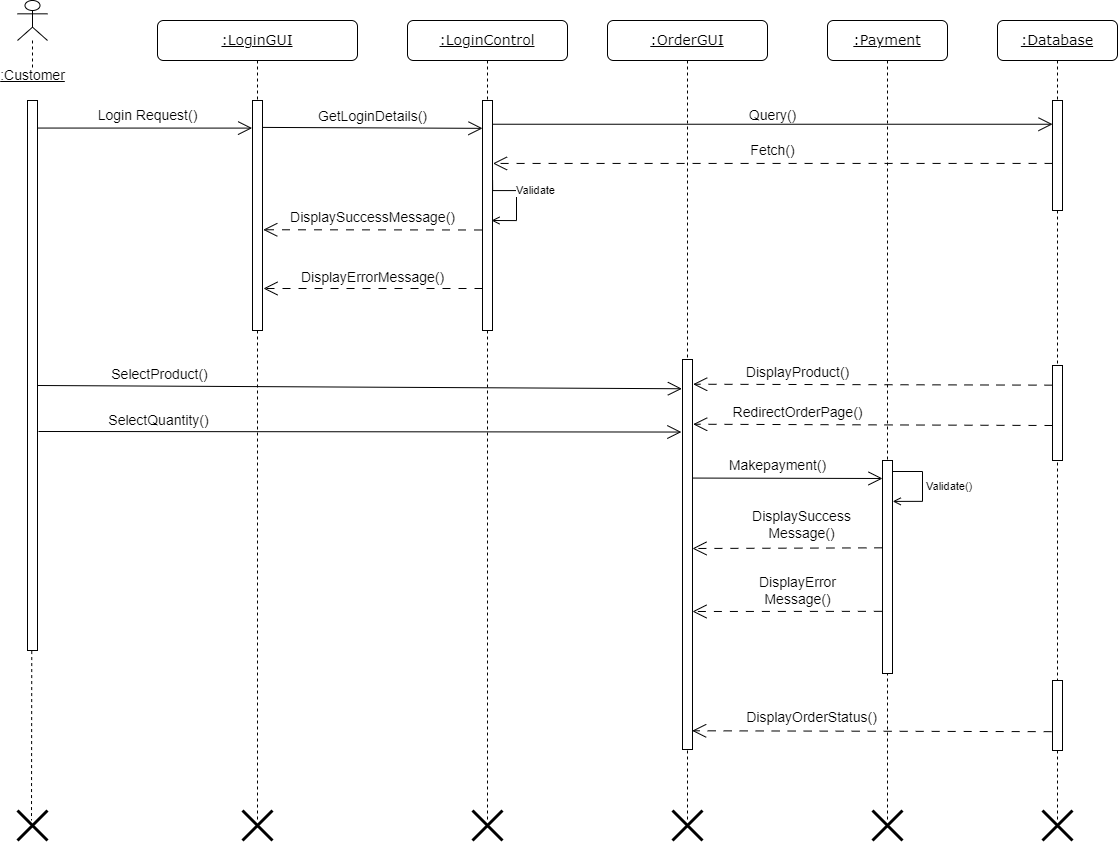


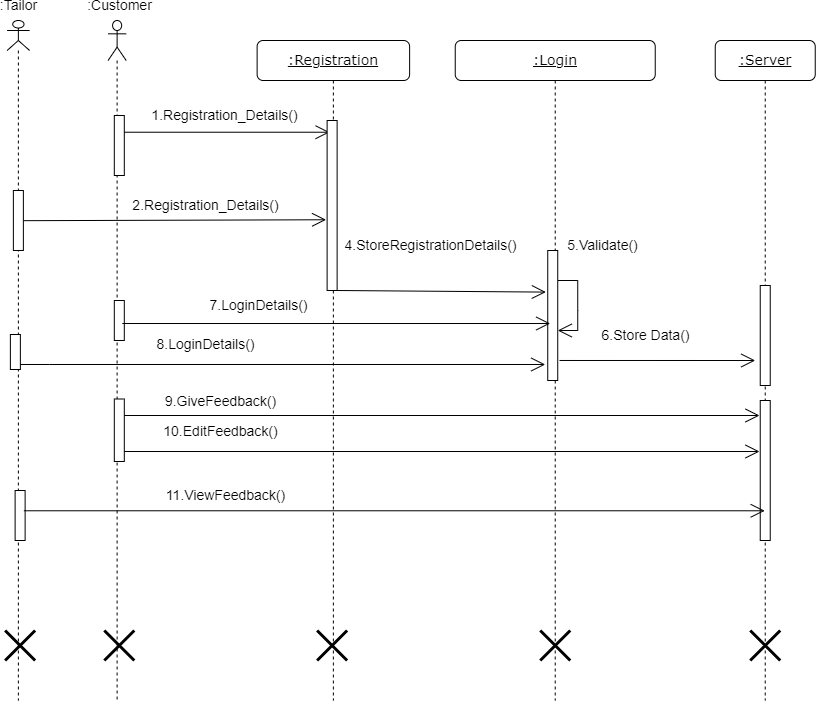
* 1. **SEQUENCE DIAGRAM:**
     1. **SEQUENCE DIAGRAM OF PROFILE**

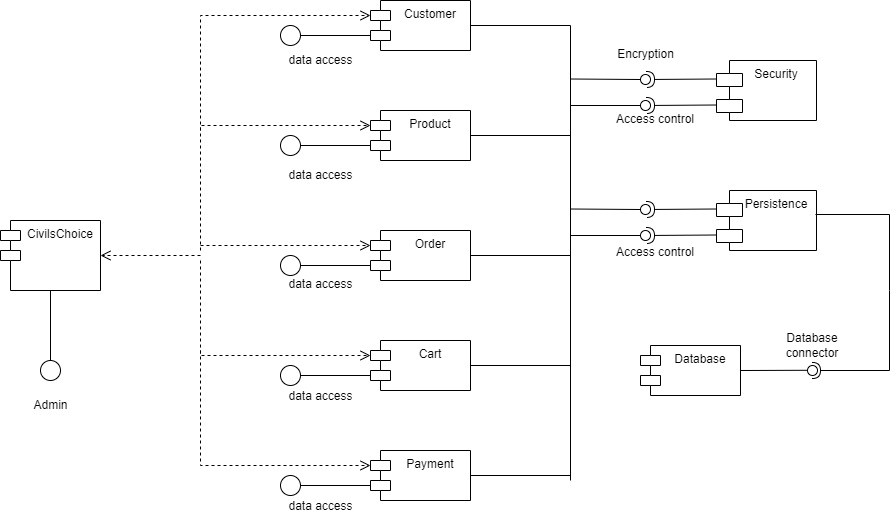


* + 1. **SEQUENCE DIAGRAM OF PRODUCT MODULE**
    2. **SEQUENCE DIAGRAM OF CART MODULE**

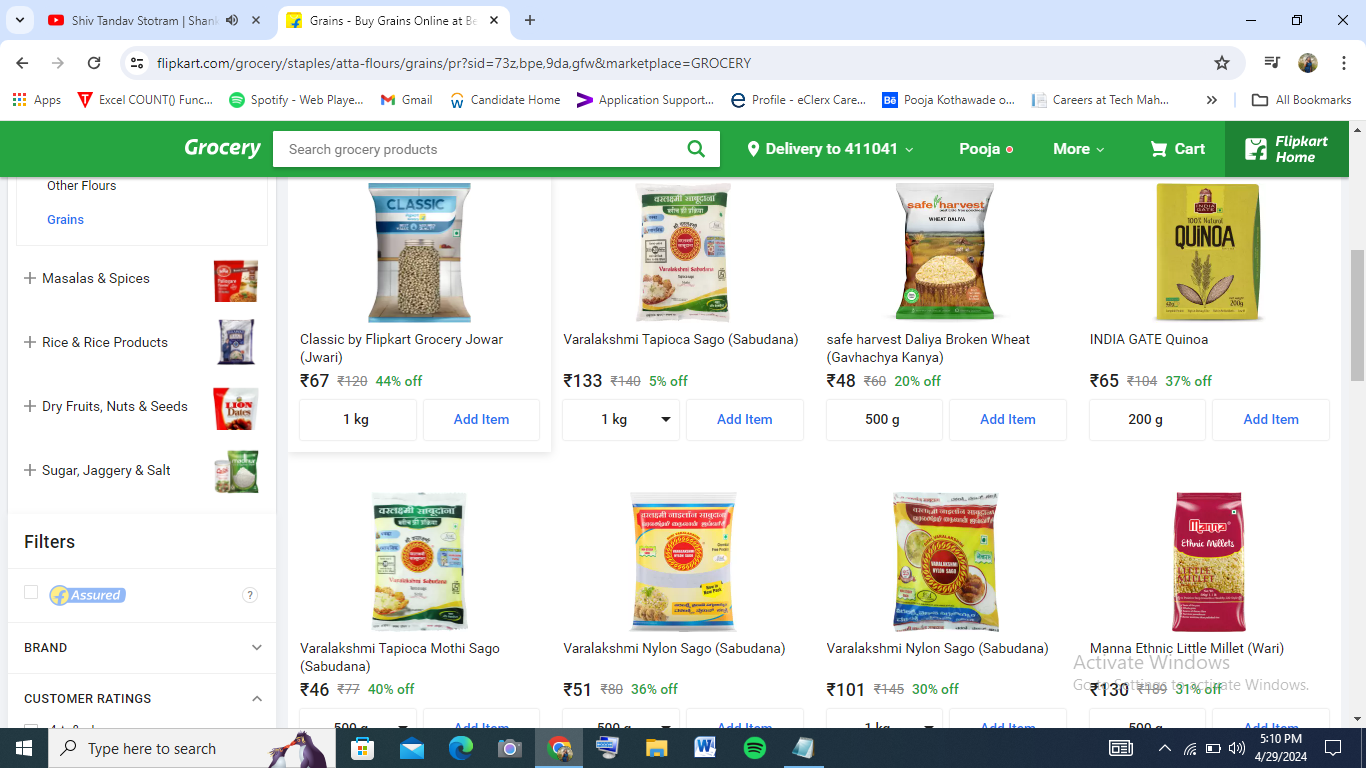


* + 1. **SEQUENCE DIAGRAM OF ORDER MODULE**
    2. **SEQUENCE DIAGRAM OF PAYMENT & DELIVERY MODULE**
    3. **SEQUENCE DIAGRAM OF FEEDBACK MODULE**

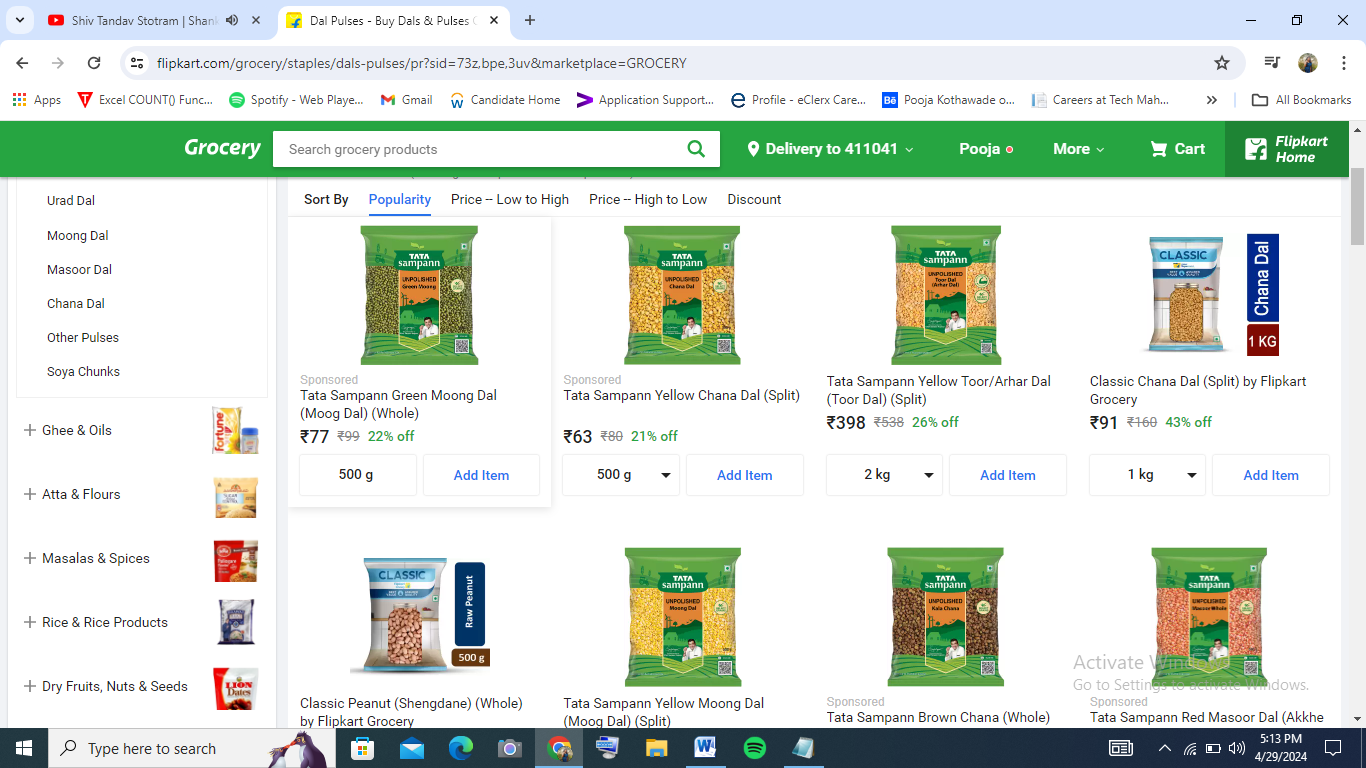


* 1. **COMPONENT DIAGRAM:**
  2. **User Input /Output screens with data**

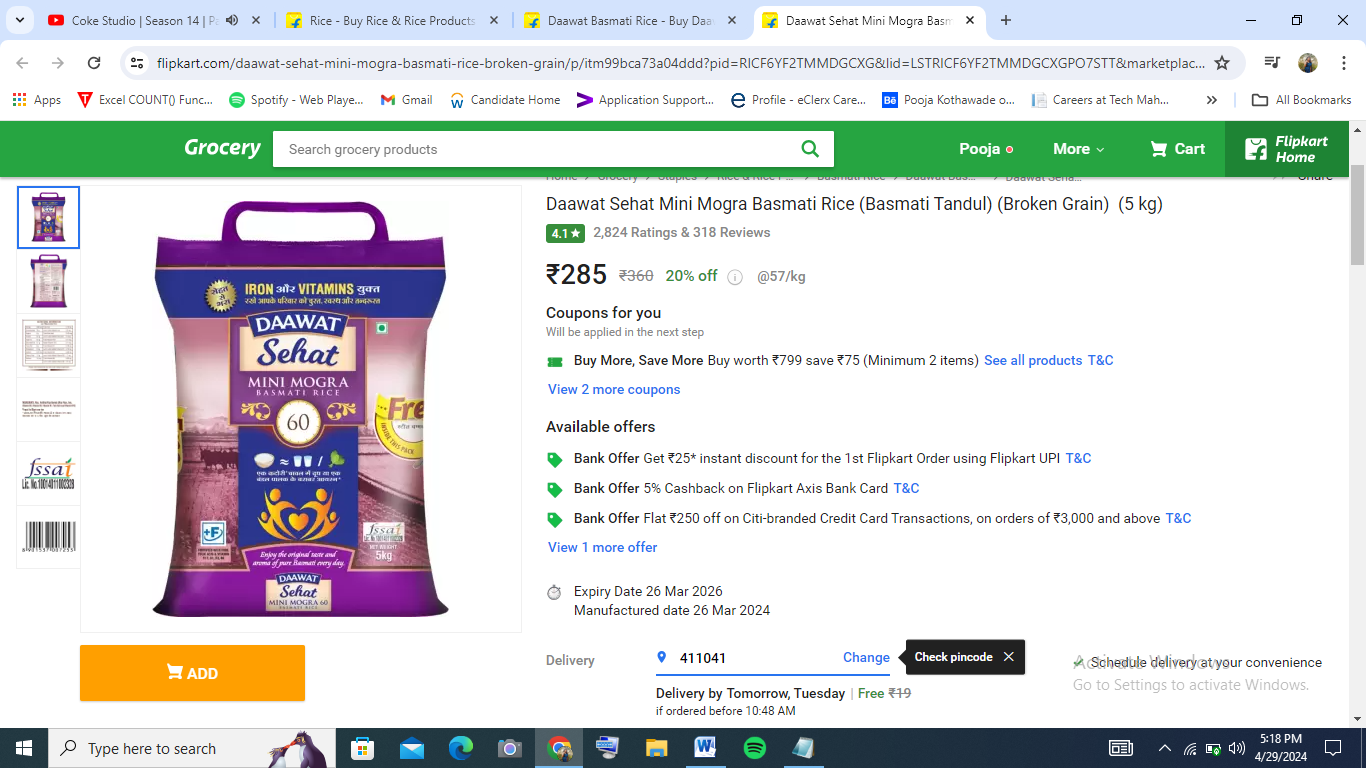
1. **Home Page:**



1. **Product Page:**



1. **Product Detail Page:**



1. **User Login Page:**
2. **User Sign-up Page:**
3. **Admin Login Page:**
4. **Contact Page:**
5. **Enquiry Page:**
6. **Admin Home Page:**
7. **View Users Page:**
8. **View Orders Page:**
9. **View Enquiry Page:**
10. **View Feedback Page:**
11. **Product Dashboard Page:**
12. **Admin Profile Page:**
13. **Add Product Page:**
14. **Edit Product Page:**
15. **Order Placed Page:**
16. **Add Address Page:**
17. **Existing Address Page:**
18. **Give Feedback Page:**
19. **Use Profile Page:**
20. **Shopping Cart Page:**
21. **Checkout Page:**
22. **Payment Page:**
23. **Payment Success Page:**

**CHAPTER 4**

* 1. **Source Code**
     1. **Manage.py: -**

from django.db import models

from django.contrib.auth.models import User

class Product(models.Model):

name = models.CharField(max\_length=100)

price = models.DecimalField(max\_digits=10, decimal\_places=2)

rating = models.IntegerField(choices=[(i, i) for i in range(1, 6)])

description = models.TextField()

category = models.CharField(max\_length=100)

quantity = models.CharField(max\_length=10, null=True)

product\_image = models.ImageField(upload\_to='product\_images/')

def \_\_str\_\_(self):

return self.name

class CustomerLogin(models.Model):

user = models.ForeignKey(User, on\_delete=models.CASCADE)

email = models.CharField(max\_length=100)

contact\_number = models.CharField(max\_length=15)

gender = models.CharField(max\_length=15, null=True)

type = models.CharField(max\_length=15, null=True)

def \_\_str\_\_(self):

return self.user.first\_name

class Enquiry(models.Model):

firstname = models.CharField(max\_length=100)

lastname = models.CharField(max\_length=100)

mobile = models.CharField(max\_length=15, null=True)

gender = models.CharField(max\_length=10, null=True)

mail = models.CharField(max\_length=100)

about = models.TextField()

creationdate = models.DateField()

def \_\_str\_\_(self):

return self.firstname

class AdminProfile(models.Model):

user = models.ForeignKey(User, on\_delete=models.CASCADE)

contact\_number = models.CharField(max\_length=15)

def \_\_str\_\_(self):

return self.user.first\_name

class CartItem(models.Model):

user = models.ForeignKey(User, on\_delete=models.CASCADE)

product = models.ForeignKey(Product, on\_delete=models.CASCADE)

quantity = models.PositiveIntegerField(default=1)

def \_\_str\_\_(self):

return self.user.first\_name

class UserAddres(models.Model):

user = models.ForeignKey(User, on\_delete=models.CASCADE)

name = models.CharField(max\_length=100)

address1 = models.CharField(max\_length=100)

address2 = models.CharField(max\_length=100)

city = models.CharField(max\_length=100)

state = models.CharField(max\_length=100)

contact = models.CharField(max\_length=100)

pincode = models.CharField(max\_length=100)

def \_\_str\_\_(self):

return self.user.first\_name

STATUS\_CHOICES = (

('Accepted','Accepted'),

('Packed','Packed'),

('On The Way','On The Way'),

('Delivered','Delivered'),

('Cancel','Cancel')

)

PAYMENT\_METHOD\_CHOICES = [

('COD', 'Cash on Delivery'),

('RAZORPAY', 'Pay with Razorpay'),

]

class OrderPlaced(models.Model):

user = models.ForeignKey(User, on\_delete=models.CASCADE)

customer = models.ForeignKey(CustomerLogin, on\_delete=models.CASCADE)

product = models.ForeignKey(Product, on\_delete=models.CASCADE)

quantity = models.IntegerField()

payment\_id = models.CharField(max\_length=100)

ordered\_date = models.DateTimeField(auto\_now\_add=True)

status = models.CharField(max\_length=50, choices=STATUS\_CHOICES, default='Pending')

def \_\_str\_\_(self):

return self.user.first\_name

class Feedback(models.Model):

firstname = models.CharField(max\_length=100)

lastname = models.CharField(max\_length=100)

mobile = models.CharField(max\_length=15, null=True)

gender = models.CharField(max\_length=10, null=True)

mail = models.CharField(max\_length=100)

about = models.TextField()

creationdate = models.DateField()

def \_\_str\_\_(self):

return self.firstname

* + 1. **Urls.py: -**

from django.contrib import admin

from django.urls import path

from core.views import \*

from django.conf import settings

from django.conf.urls.static import static

urlpatterns = [

path('admin/', admin.site.urls),

path('', index, name="index"),

path('admin\_index', admin\_index, name="admin\_index"),

path('admin\_login', admin\_login, name="admin\_login"),

path('admin\_profile', admin\_profile, name="admin\_profile"),

path('user\_profile', user\_profile, name="user\_profile"),

path('admin\_view\_user', admin\_view\_user, name="admin\_view\_user"),

path('delete\_user/<int:pid>', admin\_delete\_user, name="delete\_user"),

path('delete\_product/<int:pid>', admin\_delete\_product, name="delete\_product"),

path('admin\_edit\_product/<int:pid>', admin\_edit\_product, name="admin\_edit\_product"),

path('admin\_view\_enquiry', admin\_view\_enquiry, name="admin\_view\_enquiry"),

path('view\_feedback', view\_feedback, name="view\_feedback"),

path('admin\_view\_product', admin\_view\_product, name="admin\_view\_product"),

path('admin\_view\_order', admin\_view\_order, name="admin\_view\_order"),

path('admin\_add\_product', admin\_add\_product, name="admin\_add\_product"),

path('user\_login', user\_login, name="user\_login"),

path('new\_address', new\_address, name="new\_address"),

path('existing\_address',existing\_address,name="existing\_address"),

path('contact', contact, name="contact"),

path('user\_index', user\_index, name="user\_index"),

path('checkout/', checkout, name='checkout'),

path('user\_signup', user\_signup, name="user\_signup"),

path('Logout', Logout, name="Logout"),

path('swh', swh, name="swh"),

path('on', on, name="on"),

path('of', of, name="of"),

path('ss', ss, name="ss"),

path('sp', sp, name="sp"),

path('baseswh', baseswh, name="baseswh"),

path('baseon', baseon, name="baseon"),

path('baseof', baseof, name="baseof"),

path('basess', basess, name="basess"),

path('basesp', basesp, name="basesp"),

path('userswh', userswh, name="userswh"),

path('useron', useron, name="useron"),

path('userof', userof, name="userof"),

path('userss', userss, name="userss"),

path('usersp', usersp, name="usersp"),

path('product-detail/<int:pk>', ProductDetail.as\_view(), name='product-detail'),

path('admin\_product-detail/<int:pk>', AdminProductDetail.as\_view(), name='admin-product-detail'),

path('user\_product-detail/<int:pk>', UserProductDetail.as\_view(), name='user-product-detail'),

path('add\_to\_cart/<int:product\_id>/', add\_to\_cart, name='add\_to\_cart'),

path('remove\_from\_cart/<int:cart\_item\_id>/', remove\_from\_cart, name='remove\_from\_cart'),

path('plus\_cart/<int:cart\_item\_id>/', plus\_cart, name='plus\_cart'),

path('minus\_cart/<int:cart\_item\_id>/', minus\_cart, name='minus\_cart'),

path('view\_cart/', view\_cart, name='view\_cart'),

path('paymentdone', payment\_done, name='paymentdone'),

path('orders/', orders, name='orders'),

path('change\_status/<int:oid>', change\_status, name='change\_status'),

path('cancel\_order/<int:oid>', cancel\_order, name='cancel\_order'),

path('download\_invoice/<int:order\_id>/', download\_invoice, name='download\_invoice'),

path('add\_feedback', add\_feedback, name='add\_feedback'),

# path('generate\_pdf', generate\_pdf, name='generate\_pdf'),

] + static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

* + 1. **Views.py: -**

from datetime import date

from decimal import Decimal

from django.contrib import messages

from django.shortcuts import get\_object\_or\_404

from django.shortcuts import render, redirect

from django.contrib.auth import authenticate, login, logout

from .models import \*

from django.contrib.auth.models import User

from django.conf import settings

from django.views.decorators.csrf import csrf\_exempt

from django.contrib.auth.decorators import login\_required

from django.views import View

from razorpay import Client

from django.http import JsonResponse

import razorpay

from django.http import HttpResponse

from django.template.loader import render\_to\_string

from django.utils.text import slugify

from reportlab.pdfgen import canvas

from django.http import HttpResponse

def user\_index(request):

products = Product.objects.all()

category = request.GET.get('category')

if category:

products = products.filter(category=category)

d = {'products': products}

return render(request, 'user\_index.html', d)

def cancel\_order(request,oid):

if request.method == 'POST':

try:

order = OrderPlaced.objects.get(id=oid)

order.status = 'Cancelled'

order.save()

messages.success(request, 'Order cancelled successfully')

return redirect('orders')

except OrderPlaced.DoesNotExist:

messages.error(request, 'Cannot cancel delivered order')

return redirect('orders')

else:

return redirect('orders')

def admin\_edit\_product(request, pid):

if not request.user.is\_authenticated:

return redirect('admin\_login')

error = ""

pro = Product.objects.get(id=pid)

if request.method == "POST":

f = request.POST['pname']

ln = request.POST['price']

c = request.POST['rating']

g = request.POST['category']

e = request.POST['detail']

a = request.POST['quantity']

i = request.FILES['image']

pro.name = f

pro.price = ln

pro.rating = c

pro.category = g

pro.description = e

pro.quantity = a

pro.product\_image = i

try:

pro.save()

error = "no"

except:

error = "yes"

d = {'error': error, 'pro': pro}

return render(request, 'admin\_edit\_product.html', d)

def checkout(request):

totalitem = 0

user = request.user

add = UserAddres.objects.filter(user=user)

cart\_items = CartItem.objects.filter(user=user)

amount = Decimal(0.0)

totalamount = Decimal(0.0)

shipping\_amount = Decimal(70.0)

cart\_product = [p for p in CartItem.objects.all() if p.user == request.user]

for p in cart\_product:

tempamount = Decimal(p.quantity) \* p.product.price

amount += tempamount

totalamount = amount + shipping\_amount

if request.user.is\_authenticated:

totalitem = len(CartItem.objects.filter(user=request.user))

return render(request, 'checkout.html',{'add': add , 'totalamount':totalamount,'cart\_items':cart\_items,'totalitem':totalitem})

def payment\_done(request):

if request.method == 'GET':

payment\_id = request.GET.get('payment\_id')

if payment\_id:

# Initialize Razorpay client with your API key and secret

client = razorpay.Client(auth=("rzp\_test\_Kw423PWX9hLuGl", "R8GatWDvqy75pccVzbHR4LWD"))

try:

# Fetch payment details using the payment ID

payment = client.payment.fetch(payment\_id)

user = request.user

customer = CustomerLogin.objects.get(user=user)

cart = CartItem.objects.filter(user=user)

for c in cart:

OrderPlaced(user=user, customer=customer, product=c.product,

quantity=c.quantity, payment\_id=payment\_id, ordered\_date=date.today()).save()

c.delete()

return redirect("orders")

except Exception as e:

# Handle exceptions if any

return HttpResponse("Error processing payment: " + str(e))

else:

# Handle if payment ID is not provided

return HttpResponse("Payment ID is missing.")

elif request.method == 'POST':

# Handle Cash on Delivery payment

user = request.user

customer = CustomerLogin.objects.get(user=user)

cart = CartItem.objects.filter(user=user)

for c in cart:

OrderPlaced(user=user, customer=customer, product=c.product,

quantity=c.quantity, payment\_id="Cash on Delivery", ordered\_date=date.today()).save()

c.delete()

return redirect("orders")

else:

# Handle if request method is not GET or POST

return HttpResponse("Invalid request method.")

**CHAPTER 5**

**Security Testing of the Project**

Testing is vital for the success of any software. no system design is ever perfect. Testing is also carried in two phases. first phase is during the software engineering that is during the module creation. second phase is after the completion of software. this is system testing which verifies that the whole set of programs hanged together.

* 1. **White Box Testing:**

In this technique, the close examination of the logical parts through the software are tested by cases that exercise species sets of conditions or loops. all logical parts of the software checked once. errors that can be corrected using this technique are typographical errors, logical expressions which should be executed once may be getting executed more than once and error resulting by using wrong controls and loops.

* 1. **Black Box Testing:**

This method enables the software engineer to device sets of input techniques that fully exercise all functional requirements for a program. black box testing tests the input, the output and the external data. it checks whether the input data is correct and whether we are getting the desired output.

* 1. **Alpha Testing:**

Acceptance testing is also sometimes called alpha testing. Be spoke systems are developed for a single customer. The alpha testing proceeds until the system developer and the customer agree that the provided system is an acceptable implementation of the system requirements.

* 1. **Beta Testing:**

On the other hand, when a system is to be marked as a software product, another process called beta testing is often conducted. During beta testing, a system is delivered among a number of potential users who agree to use it. The customers then report problems to the developers. This provides the product for real use and detects errors which may not have been anticipated by the system developers.

* 1. **Unit Testing:**

Each module is considered independently. it focuses on each unit of software as implemented in the source code. it is white box testing.

* 1. **Integration Testing:**

Integration testing aims at constructing the program structure while at the same constructing tests to uncover errors associated with interfacing the modules. modules are integrated by using the top-down approach.

* 1. **Validation Testing:**

Validation testing was performed to ensure that all the functional and performance requirements are met.

* 1. **System Testing:**

It is executing programs to check logical changes made in it with intention of finding errors. a system is tested for online response, volume of transaction, recovery from failure etc. System testing is done to ensure that the system satisfies all the user requirements.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TC**  **.**  **No** | **Test Description**  **/ summary** | **Test Steps/ navigation**  **steps** | **Test data** | **Priority** | **Expected Result** | **Actual Result** | **Statu s** | **Remar k** |
| 1 | Verify login for correct Credentials | 1. Enter   Username and Enter Password  valid   1. Click on OK button | Username –  \*\*\*\*,  Password  –  Raju | P1 | Login  should be done successfully if username and password are correct | Login Successful | Pass |  |
| 2 | Verify username field | 1. Enter   username   1. Click on ok button | Username- Raju | P2 | Username should  accept characters | username is not characters | Fail | Raise Bug |
| 3 | Verify password field | Enter password  2.Click on ok button | Password- Raju | P2 | Password should  accept characters | Password is not characters | Fail | Raise Bug |
| 4 | Verify the login page, when the username and password both are blank. | 1. Click on login button. | NA |  | Login  should not be done successfully if username and password are blank | Login is not done successful y if username and password are blank. | Pass |  |
| 5 | Verify if the data in password field is not visible |  | NA | P1 | Data in password  field is not  visible | Data in password field is not visible | pass |  |
| 6 | Verify the time taken  to  login with a valid username and password. | 1. Enter valid username and password. 2. Click on login button | Username- Raju Password- Raju | P2 | Home page should be open in few second after login with a valid username and password | Home page  is to be open in few second after login with a valid username and password. | pass |  |

**Sign up Form**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TC.**  **No** | **Test**  **Description** | **Test**  **Steps** | **Test**  **data** | **Priority** | **Expected**  **Result** | **Actual**  **Result** | **Status** |
| 1. | Verify Registration for Correct Credentials | 1. Enter Username, email address and passwords 2. Click on Submit Button | Username:  Pooja  Email:  Pooja@ gmail.com  password:  pooja@ 123 | P1 | Registration should be done successfully if all the validation are correct | Registration Successful | Pass |
| 2. | Verify Username | 1. Enter Username 2. Click on Submit Button | Username:  Pooja | P2 | Username should have characters | Registration Successful | Pass |
| 3. | Verify Email | 1. Enter email address 2. Click on Submit Button | Email:  pooja@ gmail.com | P2 | Email should have characters | Registration Successful | Pass |
| 4. | Verify Password | 1. Enter passwords 2. Click on Submit Button | password:  pooja@ 123 | P2 | Password should have characters | Registration Successful | Pass |
| 5. | Verify if password is visible or not | 1. Click on submit button | NA | P1 | Data in password field are not visible | Data in password field are not visible | Pass |

**Feedback Form**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TC.**  **No** | **Test**  **Description** | **Test**  **Steps** | **Test**  **Data** | **Priority** | **Expected**  **Result** | **Actual**  **Result** | **Status** |
| 1. | Verify Feedback for Correct Credentials | 1. Enter Name, Mobile, City, State, Pin Code and Descriptions 2. Click on Submit Button | Name: \*\*\* Mobile: \*\* City: \*\*\*\* State: \*\*\* Pin code: \*  Description  : \*\*\* | P1 | Feedback should be done successfully if all the validation are correct | Feedback Submit Successful | Pass |
| 2. | Verify name | 1. Enter name 2. Click on Submit Button | Name: Raju | P2 | name should have characters | Feedback Submit Successful | Pass |
| 3. | Verify Mobile | 1. Enter Mobile Number 2. Click on Submit Button | Mobile:  \*\*\*\*\*\*\*\*\*  \* | P2 | Mobile should have integers | Feedback Submit Successful | Pass |
| 4. | Verify City | 1. Enter City name 2. Click on Submit Button | City: Pune | P2 | City should have characters | Feedback Submit Successful | Pass |
| 5. | Verify State | 1. Enter State name 2. Click on Submit Button | State: Maharashtr a | P2 | State should have characters | Feedback Submit Successful | Pass |
| 6. | Verify Pin code | 1. Enter Pin Code 2. Click on Submit Button | Pin Code:  \*\*\*\*\*\* | P2 | Pin Code should have integers | Feedback Submit Successful | Pass |
| 7. | Verify Description | 1. Enter Description 2. Click on Submit Button | Description  :  \*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\* | P2 | Description should have characters | Feedback Submit Successful | Pass |

**Address form**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TC.**  **No** | **Test**  **Description** | **Test**  **Steps** | **Test**  **Data** | **Priority** | **Expected**  **Result** | **Actual**  **Result** | **Status** |
| 1. | Verify Address for Correct Credentials | 1. Enter Name, Mobile, City, State, Pin Code and Locality 2. Click on Submit Button | Name: \*\*\* Mobile: \*\* City: \*\*\*\* State: \*\*\* Pin code: \*  Locality:  \*\*\* | P1 | Address should be done successfully if all the validation are correct | Address Submit Successful | Pass |
| 2. | Verify name | 1. Enter name 2. Click on Submit Button | Name: Raju | P2 | name should have characters | Address Submit Successful | Pass |
| 3. | Verify Mobile | 1. Enter Mobile Number 2. Click on Submit Button | Mobile:  \*\*\*\*\*\*\*\*\*  \* | P2 | Mobile should have integers | Address Submit Successful | Pass |
| 4. | Verify City | 1. Enter City name 2. Click on Submit Button | City: Pune | P2 | City should have characters | Address Submit Successful | Pass |
| 5. | Verify State | 1. Enter State name 2. Click on Submit Button | State: Maharashtr a | P2 | State should have characters | Address Submit Successful | Pass |
| 6. | Verify Pin code | 1. Enter Pin Code 2. Click on Submit Button | Pin Code:  \*\*\*\*\*\* | P2 | Pin Code should have integers | Address Submit Successful | Pass |
| 7. | Verify Locality | 1. Enter Locality 2. Click on Submit Button | Locality:  \*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\* | P2 | Locality should have characters | Address Submit Successful | Pass |

**CHAPTER 6**

**Limitations of Proposed System:**

You cannot buy product directly i.e. you have to go through the process where you are supposed to add product to your bag first and then you can buy the product. You cannot send notification on mobile number regarding product delivery.

**Implementation and Software Specification Testing’s**

* 1. **Detailed Design of Implementation**

This phase of the systems development life cycle refines hardware and software specifications, establishes programming plans, trains users and implements extensive testing procedures, to evaluate design and operating specifications and/or provide the basis for further modification.

* + 1. **Technical Design**

This activity builds upon specifications produced during new system design, adding detailed technical specifications and documentation.

* + 1. **Test Specifications and Planning**

This activity prepares detailed test specifications for individual modules and programs, job streams, subsystems, and for the system as a whole.

* + 1. **Programming and Testing**

This activity encompasses actual development, writing, and testing of program units or modules.

* + 1. **User Training**

This activity encompasses writing user procedure manuals, preparation of user training materials, conducting training programs, and testing procedures

* + 1. **Acceptance Test**

A final procedural review to demonstrate a system and secure user approval before a system becomes operational.

* + 1. **Installation Phase**

In this phase the new Computerized system is installed, the conversion to new procedures is fully implemented, and the potential of the new system is explored.

* + 1. **System Installation**

The process of starting the actual use of a system and training user personnel in its operation.

* + 1. **Review Phase**

This phase evaluates the successes and failures during a systems development project, and to measure the results of a new Computerized Tran system in terms of benefits and savings projected at the start of the project.

* + 1. **Development Recap**

A review of a project immediately after completion to find successes and potential problems in future work.

* + 1. **Post-Implementation Review**

A review, conducted after a new system has been in operation for some time, to evaluate actual system performance against original expectations and projections for cost-benefit improvements. Also identifies maintenance projects to enhance or improve the system

* 1. **THE STEPS IN THE SOFTWARE TESTING**

1. Preparation of the test cases.
2. Preparation of the possible test data with all the validation checks.
3. Complete code review of the module.
4. Actual testing done manually.
5. Modifications done for the errors found during testing.
6. Prepared the test result scripts.
   1. **The unit testing done included the testing of the following items:**
7. Functionality of the entire module/forms.
8. Validations for user input.
9. Checking of the Coding standards to be maintained during coding.
10. Testing the module with all the possible test data.
11. Testing of the functionality involving all type of calculations etc.
12. Commenting standard in the source files.

After completing the Unit testing of all the modules, the whole system is integrated with all its dependencies in that module. While System Integration, we integrated the modules one by one and tested the system at each step.

* 1. **The steps involved during System testing are as follows:**
* Integration of all the modules/forms in the system.
* Preparation of the test cases.
* Preparation of the possible test data with all the validation checks.
* Actual testing done manually.
* Recording of all the reproduced errors.
* Modifications done for the errors found during testing.
* Prepared the test result scripts after rectification of the errors
  1. **The System Testing done included the testing of the following items:**

1. Functionality of the entire system.
2. User Interface of the system.
3. Testing the dependent modules together with all the possible test data.
4. Verification and Validation testing.
5. Testing the reports with all its functionality.

After the completion of system testing, the next following phase was the Acceptance Testing. Clients at their end did this and accepted the system with appreciation. Thus, we reached the final phase of the project delivery.

* 1. **There is other six tests, which fall under special category. They are described below:**
* Peak Load Test: It determines whether the system will handle the volume of activities that occur when the system is at the peak of its processing demand. For example, test the system by activating all terminals at the same time.
* Storage Testing: It determines the capacity of the system to store transaction data on a discord in other files.
* Performance Time Testing: it determines the length of time system used by the system to process transaction data. This test is conducted prior to implementation to determine long it takes to get a response to an inquiry, make a backup copy of a file, or send a transmission and get a response.
* Recovery Testing: This testing determines the ability of user to recover data or re- start system after failure.
* Procedure Testing: It determines the clarity of documentation on operation and uses of system by having users do exactly what manuals request. For example, powering down system at the end of week or responding to paper-out light on printer.
* Human Factors Testing: It determines how users will use the system when processing data or preparing reports

**CHAPTER 7**

**PROPOSED ENHANCEMENT**

* 1. **Proposed Enhancement**

The proposed enhancement aims to improve the efficiency and user experience of the existing vehicle part selling system. By integrating various features and streamlining the process, this enhancement will enhance customer satisfaction and drive sales. The key features include a comprehensive search function, personalized recommendations, real-time inventory updates, and seamless integration with payment gateways.

* + 1. Comprehensive Search Function: Implement an advanced search function that allows customers to search for vehicle parts using multiple parameters such as make, model, year, part number, and description. This will help users quickly find the exact parts they need, reducing the time spent searching and increasing customer satisfaction.
    2. Personalized Recommendations: Leverage data analytics and machine learning algorithms to provide personalized recommendations to customers based on their previous searches, purchase history, and browsing behavior. By suggesting relevant parts or related accessories, the system can enhance cross-selling opportunities and increase customer engagement.
    3. Real-Time Inventory Updates: Integrate the system with the inventory management software to ensure real-time updates of available stock. This will prevent customers from ordering out-of-stock parts and reduce the need for manual stock checks. Additionally, implement a notification system that alerts customers when previously out-of-stock parts become available again.
    4. Easy-to-Use Interface: Design a user-friendly interface that is intuitive and visually appealing. Optimize the user experience by providing clear product descriptions, high-quality images, and customer reviews. Implement a seamless navigation system that allows users to easily browse through categories, filter search results, and compare different parts.
    5. Order Tracking and Shipment Updates: Enable customers to track their orders in real-time and provide regular updates regarding the shipment status. This feature will improve transparency and keep customers informed about the progress of their purchases, enhancing their overall experience with the system.
    6. Customer Support and Feedback: Implement a dedicated customer support system, including a live chat feature, to address customer inquiries, provide technical assistance, and handle complaints promptly. Additionally, encourage customers to leave feedback and ratings, which can be used to further improve the system and identify areas for enhancement.

**CHAPTER 8**

* 1. **Conclusion**

The CerealSphere project represents a significant advancement for Balaji Trading Company, addressing key operational challenges and setting the stage for improved efficiency, customer satisfaction, and sustainability. By implementing a comprehensive, integrated platform, the company has transformed its supply chain management, customer relationship management (CRM), logistics, and business analytics capabilities.

Operational Efficiency: The project successfully streamlined business processes, reducing manual tasks and enhancing coordination among different departments. This has led to a more efficient workflow and reduced costs.

Enhanced Customer Relationships: With a robust CRM system, Balaji Trading Company can now offer personalized customer experiences, improve order tracking, and respond more effectively to customer inquiries. This has contributed to increased customer satisfaction and loyalty.

Logistics Optimization: The CerealSphere system optimized logistics and transportation, resulting in shorter delivery times and reduced transportation costs. This achievement supports the company's commitment to operational excellence.

Data-Driven Insights: The integration of real-time business analytics has empowered stakeholders to make informed decisions, identify trends, and implement continuous improvements. This data-driven approach enhances the company's strategic planning and competitiveness.

Security and Compliance: The project's focus on robust security measures and compliance with industry regulations has ensured the protection of sensitive data and maintained the company's reputation for integrity.

Sustainability and Social Responsibility: CerealSphere's emphasis on sustainable practices, such as waste reduction and energy efficiency, aligns with Balaji Trading Company's commitment to environmental responsibility and community engagement.

There are several factors and variables that need to be considered and decided upon when starting an e-commerce business. Some of these include: types of e-commerce, marketing strategies, and countless more. If the correct methods and practices are followed, a business will prosper in an e-commerce setting with much success and profitability.

**CHAPTER 9**

**9.1. Bibliography:**

* Python: The Complete Reference by Martin C. Brown
* Python Data Analytics: With Pandas, NumPy, and Matplotlib 2nd ed. Edition by Fabio Nelli
* Core Python Programming by Wesley J. Chun Publisher: Prentice Hall
* Python Programming: A modular approach by Taneja Sheetal, Kumar Naveen
* Beginner's Guide to Python Programming: Learn Python 3 Fundamentals,
* Plotting and Tkinter GUI Development Easily by Serhan Yamacli
* Programming Python, O’reilly, by Mark Lutz
* Learning Python, O’reilly, Mark Lutz
* Head First Python, O’reilly, By Paul Barry

**9.2. Web References:**

* https://www.djangoproject.com/
* https://www.python.org/shell/
* https://www.sqlite.org/
* https://youtu.be/I6rR3Se72BU
* [www.stackoverflow.com](http://www.stackoverflow.com/)
* [www.w3schools.com](http://www.w3schools.com/)
* https://dinarys.com/blog/functional-requirements-for-ecommerce-site
* https://elogic.co/blog/how-to-write-an-ecommerce-Website-requirements...