

A
PROJECT REPORT ON
E-COMMERCE WEBSITE
FOR
YUKTI GROUP

SUBMITTED BY
RAHUL GURJAR (2038)

SUBMITTED TO
SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN PARTIAL FULFILLMENT OF DEGREE
MASTER OF COMPUTER APPLICATION SEM – IV

UNDER THE GUIDANCE OF
Dr. Rajesh kashyap



Through

Zeal Education Society's

**Zeal Institute of Business Administration,
Computer Application & Research
(ZIBACAR)**

Sr. No. 39, Narhe, Pune -411041, Phone No.:67206031

(Approved by A.I.C.T.E., New Delhi, Recognized by DTE, Govt. Maharashtra & Affiliated to S.P.P.U. Pune)

2022-2023



ZEAL EDUCATION SOCIETY'S

**ZEAL INSTITUTE OF BUSINESS ADMINISTRATION,
COMPUTER APPLICATION AND RESEARCH (ZIBACAR)**

NARHE | PUNE | INDIA



PUN CODE: IMMP013170

DTE CODE: 6152

AISHE CODE: C-41828

Certificate

This is to certify that the project report entitled, **E-Commerce Website** for '**Yukti Group**' being submitted here for the internal work of the degree of **MASTER OF COMPUTER APPLICATIONS (SEM-IV)** to Savitribai Phule Pune University, Pune is the result of the original project work completed by **Rahul Gurjar (2038)** under my supervision and guidance of **Dr. Rajesh kashyap** and to the best of my knowledge and belief, the work embodies in this Project has not formed earlier the basis for the award of any Degree of similar title or any other University or examining body.

Place: Pune

Date: 24/06/2023

Dr. Rajesh
Kashyap
Project Guide

Dr. Rupali Pawar
Project Coordinator

Prof. Kirti Samrit
Program Coordinator

Prof. Pandurang Patil
Director

Internal Examiner
External Examiner



CERTIFICATE

of Internship



PROUDLY PRESENTED TO

Rahul Gujjar

We are happy to certify that Mr. Rahul Gujjar has done an internship as a "Web Developer"
From 15 May 2023 to 20 July 2023. We appreciate his work and his contribution



Abhay pratap singh
SIGNATURE

DECLARATION BY STUDENT

To,

The Director,

ZIBACAR, Pune

I undersigned hereby declare that this project titled **E-Commerce Website** written and submitted by me to SSPU Pune, in partial fulfillment of the requirement of the award of the degree of **MASTER OF COMPUTER APPLICATIONS (MCA-II) SEM-IV** under the guidance of **Dr. Rajesh Kashyap** is my original project work.

I further declare that to the best of my knowledge and belief, this project has not been submitted to this or any other University or Institute for the award of any Degree.

Place: Pune

Date: 24/06/2023



Student Name

Rahul Gurjar

ACKNOWLEDGEMENT

I extend my sincere gratitude to Honorable **Shri. Sambhajirao Katkar**, President, ZES Pune And Head, ZES Management Program, **Prof. Pandurang Patil**, Director of ZIBACAR, and **Dr. Rajesh Kashyap** for allowing me to carry out the study and for his constant encouragement, valuable suggestion and guidance during their search work.

I extend my special thanks to **Dr. B.J Mohite**, **Dr. Madhavi Shamkuwar**, **Prof. Kirti Samrit**, **Dr. Rupali Kalekar** and **Dr. Rupali Pawar** their kind cooperation and inspiration.

I extend my special gratitude to my Dearest Mother and Father or family members and friends who encouraged and motivated me to complete the project report.

Place: Pune

Date: 24/06/2023



Student Name

Rahul Gurjar

INDEX

| Chapter No. | Details | Page No. |
|------------------------------|--|----------|
| 1 | INTRODUCTION | |
| | 1.1 Organization Profile | |
| | 1.2 Project Abstract | |
| | 1.3 Existing System and Need of System | |
| | 1.4 Scope of the System | |
| | 1.5 Operating Environment | |
| | 1.6 Brief Description of the technology used | |
| 2 | PROPOSED SYSTEM | |
| | 2.1 Study of Similar System | |
| | 2.2 Objectives of the Proposed System | |
| | 2.3 Users of the System | |
| 3 | ANALYSIS & DESIGN | |
| | 3.1 ERD | |
| | 3.2 Data Dictionary | |
| | 3.3 Table Structure | |
| | 3.4 Deployment Diagram | |
| | 3.5 Use Case Diagram | |
| | 3.6 Activity Diagram | |
| | 3.7 Class Diagram | |
| 3.8 Module Hierarchy Diagram | | |
| 4 | CODE SNIPPETS | |
| | 4.1 Screen Shot | |
| | 4.2 Coding | |
| 5 | TESTING STRATEGY | |
| | 5.1 Test Case | |
| 6 | LIMITATION OF THE PROPOSED SYSTEM | |
| 7 | PROPOSED ENHANCEMENT | |
| 8 | CONCLUSION | |
| 9 | BIBLIOGRAPHY | |

Chapter 1 Introduction

Introduction:

- This project is a web-based shopping system for an existing shop.
 - E-Commerce platform is the process where consumers directly buy goods or services from a seller in real-time, without an intermediary service, over the internet.
 - E-Commerce products buying through internet is growing not only in India but, globally we see an increasing number of people are going towards online shopping which is preferable now a days.
 - The main benefit of this website is that all the products can be purchased in one website as all the products are sold here so customers do not need to go to different stores to buy different products.
 - As you can get more Discounts and more shopping deals while purchasing online rather than in stores.
 - This website will be helpful to buy all the products in single click.
-
- **Organization/Client Name:** Yukti Group
 - **Location & Address:** Undri, Pune, Maharashtra 411052
 - **Timing:** Mon – Sat 9.00AM – 8.00PM

1.1 Client / Organization Profile:

📌 History :

- Yukti Group was built in 2015, It is a store Located in Pune, India. .
- This is a store in which we sell all e-commerce products such as include Fashion (Apparel, Footwear, Bags, Accessories, Jewelry, Eyewear, Cosmetics, and Beauty), Consumer Electronics, Kitchenware products, Sporting goods, Health & Wellness (Supplements, Nutraceuticals, Aromatherapy items, etc.) Supplements, Nutraceuticals, Aromatherapy products, etc. Pets, children's products (toys, clothes, etc.).
- This is known as the bestselling market amongst all other located in Undri, Pune.

📌 About Yukti Group:

- Yukti Group is an online website in which e-commerce products like: Fashion, Consumer Electronics, Kitchenware products, Sporting goods, Health & Wellness, Supplements, Nutraceuticals, Aromatherapy products, etc.
- The product details such as name of product, price, quantity, are displayed so have to option to choose your accordingly.
- The selected item will be added to cart than the product will be shown in your order and you can checkout out from order.

1.2 Abstract

Yukti Group is a web-based application intended for online retailers. This project is an attempt to provide the advantage of online shopping to customers of a real shop. The main objective of this application is to make it interactive and its ease of use. It would make searching, viewing and selection of product easier. It contains a sophisticated search engine for users to search for products specific to their needs. This search engine provides an easy and convenient way to search for products where a user can search for a product interactively and the search engine would refine the product available based on the user's inputs. The user can then view the complete specification of that particular product. The purpose of this Yukti group is to automate the existing manual system with the help of computerized equipment and full-fledged computer software, fulfilling their requirements so that their valuable data/information can be stored for a longer period with easy access and manipulation of the same. The required software and hardware are easily accessible and easy to work with. Yukti group, as described above, can lead to error-free, secure, reliable, and fast management systems. It can assist the user to concentrate on their other activities rather than concentrate on record keeping. Thus, it will help the organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant while being able to reach the information.

Yukti provides some features as following:

❖ Project Features

- Customer can see all e-commerce item in one click.
- Customers can buy products at low prices.
- Admin can add, update, delete & view all product details easily.
- The product which is added to cart can be easily ordered by visiting into my cart.
- Customers can get their all products by sitting at their home by just ordering it online.
- It provides 24X7 online customer service.

1.3 Existing System & Need for System:

- The present scenario for shopping is to visit the shops and market manually and then from the available product list one needs to choose the item he or she wants and then pay for the same item mainly in cash mode is done, as not every society is well educated and aware to use net banking etc.
- This system is not much user-friendly as one needs to go the market physically and then select items only from the available list. So mostly it is difficult to get the product as per our desire.
- Description about the product is less available and are mostly verbal only. For this type of shopping, one needs to have ample amount of free time. Also, not really good markets exist everywhere, so many times good markets become out of reach for certain people.
- In the proposed system customers need not go to the shops for purchasing the products. He/she can order the product he/she wishes to buy through the use of this system. The shop owner can be the admin of the system.

1.4 Scope of System

➤ User Module:

- Nowadays people are more dependent in online shopping, to make there work easier so the online platform is much in demand.
- The first step is that the customer has to register there customer's details.
- After the registration process is completed customer can login into yukti enterprises.
- After the registration process is successfully completed customer can view the products details
- Customer can also add there favorite item buy clicking on a heart which is given below the product and that product will be added to cart .
- Once the product added to cart customer has to fill address details the customer has to conform the order and it will delivered to them at there address.

➤ Admin Module:

- The overall System is under the control of admin.
- It is in the Admin's hand to add, delete, update the product details.
- All the order details is viewed by the admin as no customer has the authority tosee the other details
- The orders which are placed by the customer can only be edited by the adminsuch as they can cancel and update the details.

1.5 Operating Environment

➤ **Client side environment :**

- Processor : Intel Core I5 for better interface
- RAM : 4GB
- Hard Disk : 50GB
- Devices : Mobile, Tablet, Computer, Laptop etc.
- Browsers : Latest upgraded any browser like Brave, MS-Edge, Chrome, Safari etc.

➤ **Server side environment :**

- Processor : Intel Core I5 for better interface
- RAM : 4GB
- Hard Disk : 50GB
- Data Base Server : MongoDB

➤ **Development side environment :**

- Client Side: React(v18.2.0), Node(v18.10.0)
- Sever Side Language: Express (v4.18.1)
- Data Base Server: MySql
- Operating System: Windows 10 or above
- Editors: Visual Code, Sublime and Notepad
- Browser: Latest upgraded any browser like Brave, MS-Edge, Chrome, Safari, Internet Explorer, Opera etc.

1.1 Brief Description of Technology Used:

React:(Version 17)

React has been designed from the start for gradual adoption, and you can use as little or as much React as you need. Whether you want to get a taste of React, add some interactivity to a simple HTML page, or start a complex React-powered app, the links in this section will help you get started.

React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes. Declarative views make your code more predictable and easier to debug.

Prerequisites

We'll assume that you have some familiarity with HTML and JavaScript, but you should be able to follow along even if you're coming from a different programming language. We'll also assume that you're familiar with programming concepts like functions, objects, arrays, and to a lesser extent, classes. If you need to review JavaScript, we recommend reading this guide. Note that we're also using some features from ES6 — a recent version of JavaScript. In this tutorial, we're using arrow functions, classes, `let`, and `const` statements. You can use the Babel REPL to check what ES6 code compiles to.

Component-Based

Build encapsulated components that manage their own state, then compose them to make complex UIs. Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM.

A Simple Component

React components implement a `render ()` method that takes input data and returns what to display. This example uses an XML-like syntax called JSX. Input data that is passed into the component can be accessed by `render ()` via `this.props`. JSX is optional and not required to use React. Try the Babel REPL to see the raw JavaScript code produced by the JSX compilation step.

NodeJS:(Node 18.0.1)

Node.js (Node) is an open-source development platform for executing JavaScript code server side. Node is Useful for developing application that required a persistent connection from the browser to the server and is often used for real-time application such as chat, news feed and web push notification. There are several types of documentation available on this website:

A little bit of history

JavaScript is a programming language that was created at Netscape as a scripting tool to manipulate web pages inside their browser, Navigator. Part of the business model of Netscape was to sell Web Servers, which included an environment called Netscape Livewire that could create dynamic pages using server-side JavaScript. Unfortunately, Netscape Livewire wasn't very successful and server-side JavaScript wasn't popularized until recently, by the introduction of Node.js.

2009

- Node.js is born
- The first form of npm is created

2010

- Express is born
- Socket.io is born

2011

- npm hits version 1.0

- Larger companies start adopting Node.js: LinkedIn, Uber, etc.
- hapi is born

2011

- npm hits version 1.0
- Larger companies start adopting Node.js: LinkedIn, Uber, etc.
- hapi is born

2013

- First big blogging platform using Node.js: Ghost
- Koa is born

2014

- The Big Fork: io.js is a major fork of Node.js, with the goal of introducing ES6 support and moving faster

2015

- The Node.js Foundation is born
- IO.js is merged back into Node.js
- npm introduces private modules
- Node.js 4 (versions 1, 2 and 3 never previously released)

2016

- The left pad incident
- Yarn is born • Node.js 6

2017

- npm focuses more on security

Express is a minimal and flexible Node.js web application framework that provides a robust set of features to develop web and mobile applications. It facilitates the rapid development of Node based Web applications. Following are some of the core features of Express framework –

- Allows to set up middlewares to respond to HTTP Requests.
- Defines a routing table which is used to perform different actions based on HTTP Method and URL.
- Allows to dynamically render HTML Pages based on passing arguments to templates.
- It can be used to design single-page, multi-page and hybrid web applications.
- It allows to setup middlewares to respond to HTTP Requests.
- It defines a routing table which is used to perform different actions based on HTTP method and URL.
- It allows to dynamically render HTML Pages based on passing arguments to templates.

Express.js is a small framework that works on top of Node.js web server functionality to simplify its APIs and add helpful new features. It makes it easier to organize your application's functionality with middleware and routing. It adds helpful utilities to Node.js HTTP objects and facilitates the rendering of dynamic HTTP objects.

- Develops Node.js web applications quickly and easily.
- It's simple to set up and personalise.
- Allows you to define application routes using HTTP methods and URLs.
- Includes a number of middleware modules that can be used to execute additional requests and responses activities.
- Simple to interface with a variety of template engines, including Jade, Vash, and EJS.

2.1 Proposed System :

- The Proposed system will be based on web application, network based web application.
- This proposed system will help the customers to buy all the different products in one website.
- The system will categorize all products under main category and each category divide to another category to facilitate the dealing with the system.
- This system provides feedback mechanism.
- This is an online system that enables any customer to order his products from internet without going to any place.
- All the products are shown in my website so customers need not go to other website to buy the other products.
- The system also has email system mechanisms to send the email.

2.2 Feasibility Study :

As the name implies, a feasibility analysis is used to determine the viability of an idea, such as ensuring a project is legally and technically as well as economically justifiable. It tells us whether a project is worth the investment in some cases, a project may not be doable.

There can be many reasons for this, including requiring too many resources, which not only prevents those resources from performing other tasks but also may cost more than an organization would earn back by taking on a project that isn't profitable. A well-designed study should offer a historical background of the business or project, such as a description of the product or service, accounting statements, details of operations and management, marketing research and policies, financial data, legal requirements, and tax obligations. Generally, such studies precede technical development and project implementation.

Types of Feasibility Study.

Technical Feasibility:

Technical feasibility assesses the practicality and compatibility of implementing the e-commerce website from a technical standpoint. Consider the following factors:

- a. **Infrastructure:** Evaluate the required hardware, software, and network infrastructure to support the website. Assess if your existing resources are sufficient or if additional investments are needed.
- b. **Development and Maintenance:** Analyze the technical expertise and resources required for website development, customization, and ongoing maintenance. Consider factors like programming languages, content management systems, hosting options, and security measures.
- c. **Scalability:** Determine if the website infrastructure can handle increasing traffic, transactions, and data volumes. Assess if the platform can be easily scaled to accommodate future growth and evolving customer needs.
- d. **Integration:** Consider the need for integrating with other systems or third-party services, such as payment gateways, inventory management, shipping providers, or customer relationship management (CRM) systems.

2.3 Objective Of System :

- To offer multiple payment options to accommodate various customer preferences.
- To provide a system where administrator can add the order status manually.
- To provide excellent customer support through various channels like live chat, email, or phone.
- To provide a system where administrator can manage the stock.
- To provide detailed and transparent product pricing, including any additional fees or charges.
- To offer gift wrapping or customization options to enhance the customer gifting experience.
- To display daily/weekly/monthly sales.
- To display stock in shop.

2.4 Users of System

➤ User Requirements:

- Visitors new to the site should be able to register by themselves. Users will be differentiated by unique user identifiers.
- Users should be able to view a category wise of specified items available through the site.
- Customer can add product in the shopping cart.
- All selected items should be shipped to the user following purchase.
- Users should be able to view the status of items they have ordered.

➤ Administrator Requirements:

- Admin can see all the transaction happening on the system.
- Admin can add, delete, update the product details.
- Admin can generate the various report.
- Admin should be able to view all user transactions.
- Site administrators should be able to change the status of goods purchased by users after items have been shipped.

3.1 System Requirements (Functional and Non-Functional)

Functional Requirements:

- A Functional Requirements a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs.
- It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional Requirements are also called Functional specification.
- In software engineering and systems engineering, a Functional Requirement can range from the high-level abstract statement of the sender's necessity to detailed mathematical functional requirement specifications. Functional software requirements help you to capture the intended behavior of the system.

1. Admin :

- The system allows to admin to managing all products details.
- The system gives permission to admin to add new product and update existing product.
- The payment details are manages by admin.
- User can manage and update his own profile.
- Admin can generate the various report.

2. Customer :

- Users should be able to view a category wise of specified items available through the site.
- Customer can add product in the shopping cart.
- Users should be able to view the status of items they have ordered.
- User can manage and update his own profile.

3. Reports :

The system should allow only admin to generate reports. Admin can generate reports like:

➤ Order Details Reports:

It provides following type of reports :

- All order list reports.
- Date wise submitted / completed reports.
- Customer name wise reports.

➤ Payment Details Reports:

It provides following type of reports :

- All payment list reports.
- Pending payment reports.

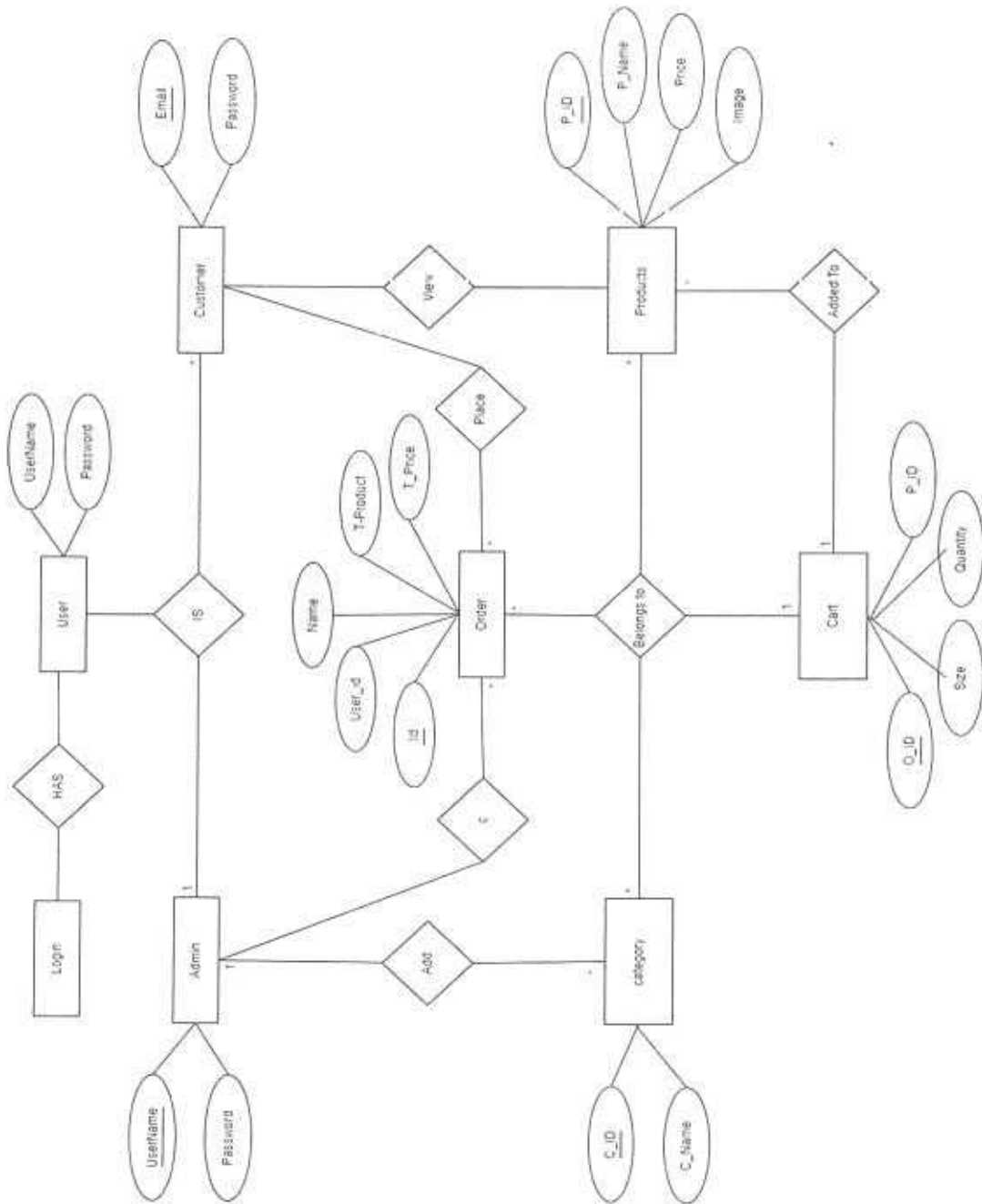
➤ Login :

For security, login module is developed for each user.

- The system automatically validates user and its type when he logged in.
- Only authenticated users have the right to login into system.
- The login panel shall contain fields to contain a user name and a field for password.
- System shall provide forget password facilities.

Non-Functional Requirements:

3.2 Entity Relationship Diagram For Online Book Shop:



3.3 Table Structure:

3.3.1 Data Dictionary :

| Field Name | Data Type | Constraints | Description |
|-------------|-------------|-------------|--|
| C_ID | Integer(4) | Not Null | A unique identifier assigned to each shopping cart in the e-commerce website. |
| Description | Varchar(50) | Not Null | A detailed description of the product, providing information about its features, specifications, and benefits. |
| Discount | Decimal | Not Null | Any discount or promotional offer applied to the order, reducing the total amount. |
| Email | Varchar(15) | Not Null | The email address associated with the user's account. |
| FullName | Varchar(10) | Not Null | The full name of the user. |
| O_Date | DateTime | Not Null | Store Date. |
| O_ID | Integer(4) | Primary Key | A unique identifier assigned to each order in the e-commerce website. |
| O_Number | Integer(5) | Not Null | Store Order Number. |
| P_ID | Integer(5) | Primary Key | A unique identifier assigned to each product in the e-commerce website. |
| P_Name | Varchar(15) | Not Null | The name or title of the product as displayed to the user. |
| P_Price | Decimal | Not Null | The price of the product, indicating the amount of money required to purchase it. |
| Password | Varchar(15) | Not Null | The encrypted password used to authenticate the user's |

E-Commerce Website

| | | | |
|---------------|-------------|-------------|---|
| | | | account. |
| Quantity | Integer(3) | Not Null | The quantity of the product added to the cart. |
| R_Amount | Decimal | Not Null | The amount refunded to the user for the order, including taxes, discounts, and other charges. |
| R_Date | DateTime | Not Null | The date and time when the refund was processed. |
| R_ID | Integer(5) | Primary Key | A unique identifier assigned to each refund. |
| R_Reason | Varchar(25) | Not Null | The reason provided for the refund, explaining the cause for the request. |
| R_Status | Varchar(10) | Not Null | The status of the refund (e.g., pending, approved, refunded). |
| Rating | Decimal | Not Null | The average rating or feedback score assigned by customers who have purchased and reviewed the product. |
| S_ZipCode | Integer(6) | Not Null | Zipcode of the shop. |
| S_Address | Varchar(30) | Not Null | The physical location or address of the shop. |
| S_description | Varchar(25) | Not Null | A description or summary of the shop, providing information about its offerings, specialties, or unique selling points. |
| S_Email | Varchar(15) | Not Null | Email of shop. |
| S_Id | Integer(5) | Primary Key | A unique identifier assigned to store in the e-commerce website |
| S_Name | Varchar(15) | Not Null | The name or title of the shop. |
| S_PhoneNumber | Integer(10) | Not Null | The contact information for the shop, such as phone number address. |

E-Commerce Website

| | | | |
|-----------------|-------------|-------------|--|
| Shipping_Fee | Decimal | Not Null | The fee or charge associated with shipping the order to the specified address. |
| Status | Varchar(15) | Not Null | The current status of the order (e.g., processing, shipped, delivered). |
| StockQuantity | Integer(3) | Not Null | The quantity of the product currently available in stock for purchase. |
| Tracking_Number | Integer(5) | Not Null | The tracking number assigned to the order for shipment tracking purposes. |
| U_ID | Integer(5) | Primary Key | A unique identifier assigned to each user in the e-commerce website. |
| U_Name | Varchar(20) | Not Null | The username chosen by the user for their account. |

3.3.2 Table Specification (Database) :

| Table Name: User | | |
|---------------------|---|--|
| Primary Key: | U_ID | |
| Description: | This Table contains all details of User | |
| Field Name | Data Type | Description |
| U_ID | Integer | A unique identifier assigned to each user in the e-commerce website. |
| U_Name | String | The username chosen by the user for their account. |
| Email | String | The email address associated with the user's account |
| Password | String | The encrypted password used to authenticate the user's account. |
| FullName | String | The full name of the user. |

| Table Name: Products | | |
|----------------------|---|--|
| Primary Key: | P_ID | |
| Description: | This Table contains all details of All Products | |
| Field Name | Data Type | Description |
| P_ID | Integer | A unique identifier assigned to each product in the e-commerce website |
| P_Name | String | The name or title of the product as displayed to the user. |
| Description | String | A detailed description of the product, providing information about its features, specifications, and benefits. |
| P_Price | Decimal | The price of the product, indicating the amount of money required to purchase it. |
| StockQuantity | Integer | The quantity of the product currently available in stock for purchase. |
| Rating | Decimal | The average rating or feedback score assigned by customers who have purchased and reviewed the product. |

| Table Name: Order | | |
|--------------------------|--|--|
| Primary Key: | O_ID | |
| Description: | This Table contains all details of All Order | |
| Field Name | Data Type | Description |
| O_ID | Integer | A unique identifier assigned to each order in the e-commerce website. |
| O_Number | Integer | Store Order Number. |
| O_Date | Date | Store Date. |
| Status | String | The current status of the order (e.g., processing, shipped, delivered). |
| Discount | Decimal | Any discount or promotional offer applied to the order, reducing the total amount. |
| Shipping_Fee | Decimal | The fee or charge associated with shipping the order to the specified address. |
| Tracking_Number | String | The tracking number assigned to the order for shipment tracking purposes. |

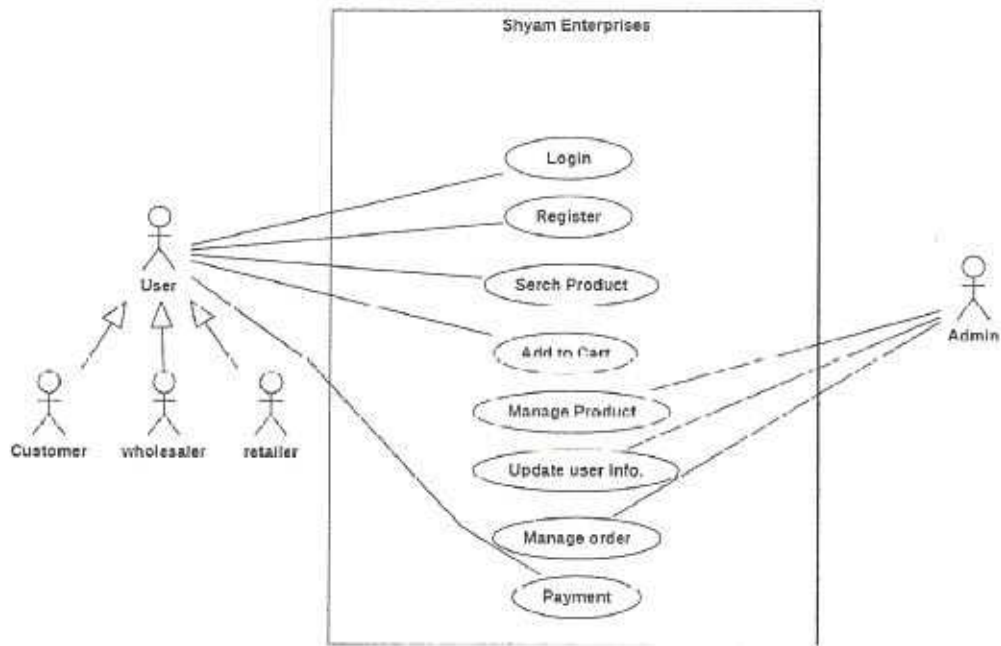
| Table Name: Shop | | |
|-------------------------|--|---|
| Primary Key: | S_ID | |
| Description: | This Table contains all details of shop. | |
| Field Name | Data Type | Description |
| S_Id | Integer | A unique identifier assigned to store in the e-commerce website |
| S_Name | String | The name or title of the shop. |
| S_description | String | A description or summary of the shop, providing information about its offerings, specialties, or unique selling points. |
| S_Address | String | The physical location or address of the shop. |
| S_PhoneNumber | Integer | The contact information for the shop, such as phone number address. |
| S_Email | String | Email of shop. |
| S_ZipCode | Integer | Zipcode of the shop. |

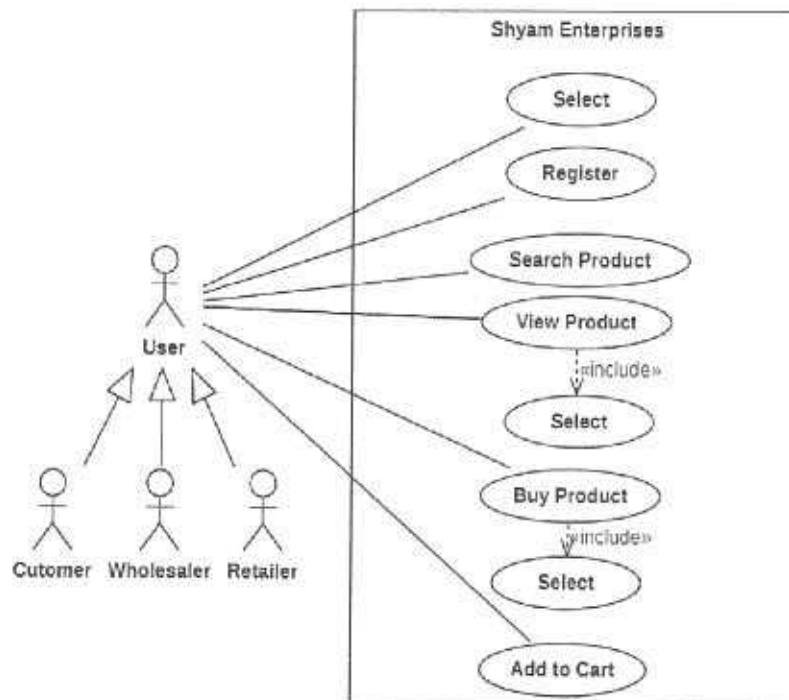
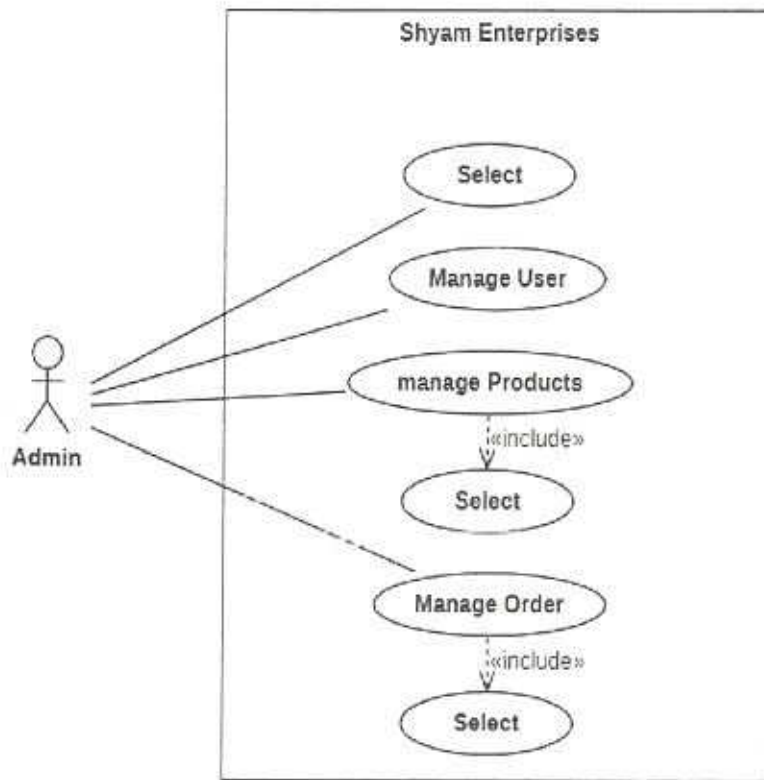
| Table Name: Bill | | |
|-------------------------|--|---|
| Primary Key: | Bill_ID | |
| Description: | This Table contains all details of bill. | |
| Field Name | Data Type | Description |
| Bill_ID | Integer | A unique identifier assigned to each bill in the e-commerce website. |
| O_ID | Integer | A unique identifier of the order associated with the bill. |
| U_ID | Integer | A unique identifier of the user who placed the order and is responsible for the bill. |
| BillDate | DateTime | The date and time when the bill was generated. |
| Total_Amount | Decimal | The total amount or cost of the bill, including taxes, discounts, and other charges. |
| Payment_Method | String | The method of payment used for the bill (e.g., credit card, PayPal, bank transfer). |
| Payment_Status | String | The status of the payment for the bill (e.g., pending, paid, refunded). |
| Payment_Date | DateTime | The date and time when the payment for the bill was processed. |
| Shipping_Address | String | The address to which the ordered products were shipped, as recorded in the bill. |
| Billing_Address | String | The billing address associated with the bill, which may be different from the shipping address. |
| Items | Array | An array of items or products included in the bill, containing details such as product ID, name, quantity, and price. |
| Discount | Decimal | Any discount or promotional offer applied to the bill, reducing the total amount. |
| Shipping_Fee | Decimal | The fee or charge associated with shipping the order, as recorded in the bill. |
| Total_Quantity | Integer | The total quantity of items or products included in the bill. |

| Table Name: Refund | | |
|---------------------------|--|---|
| Primary Key: | R_ID | |
| Description: | This Table contains all details of refund order. | |
| Field Name | Data Type | Description |
| R_ID | Integer | A unique identifier assigned to each refund in the e-commerce website. |
| O_ID | Integer | A unique identifier of the order associated with the refund. |
| U_ID | Integer | A unique identifier of the user who placed the order and is eligible for the refund. |
| R_Date | DateTime | The date and time when the refund was processed. |
| R_Amount | Decimal | The amount refunded to the user for the order, including taxes, discounts, and other charges. |
| R_Status | String | The status of the refund (e.g., pending, approved, refunded). |
| R_Reason | String | The reason provided for the refund, explaining the cause for the request. |

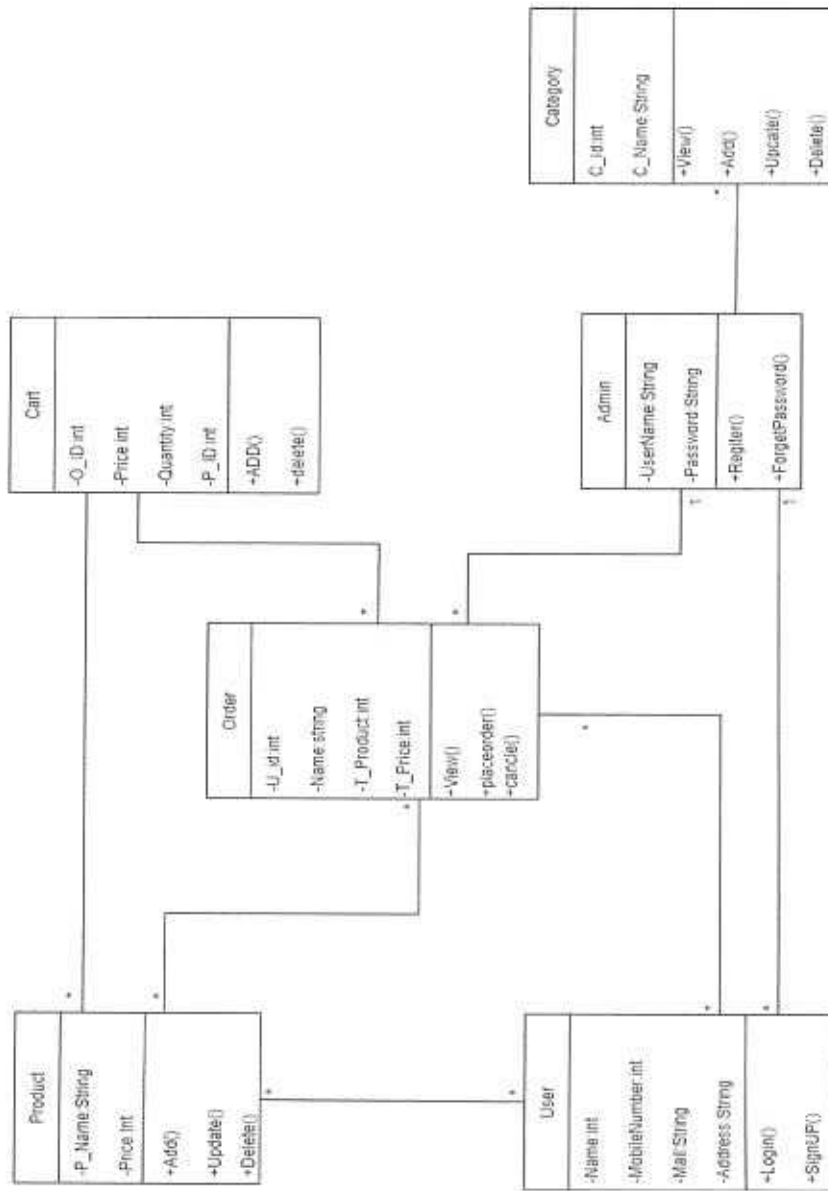
| Table Name: Cart | | |
|-------------------------|--|---|
| Primary Key: | C_ID | |
| Description: | This Table contains all details of All Product add into cart | |
| Field Name | Data Type | Description |
| C_ID | Integer | A unique identifier assigned to each shopping cart in the e-commerce website. |
| U_ID | Integer | A unique identifier of the user who owns or manages the shopping cart. |
| P_Name | String | The name or title of the product added to the cart. |
| Quantity | Integer | The quantity of the product added to the cart. |
| P_Price | Decimal | The price of the product added to the cart. |

3.4 Use case Diagram For E-Commerce Website:

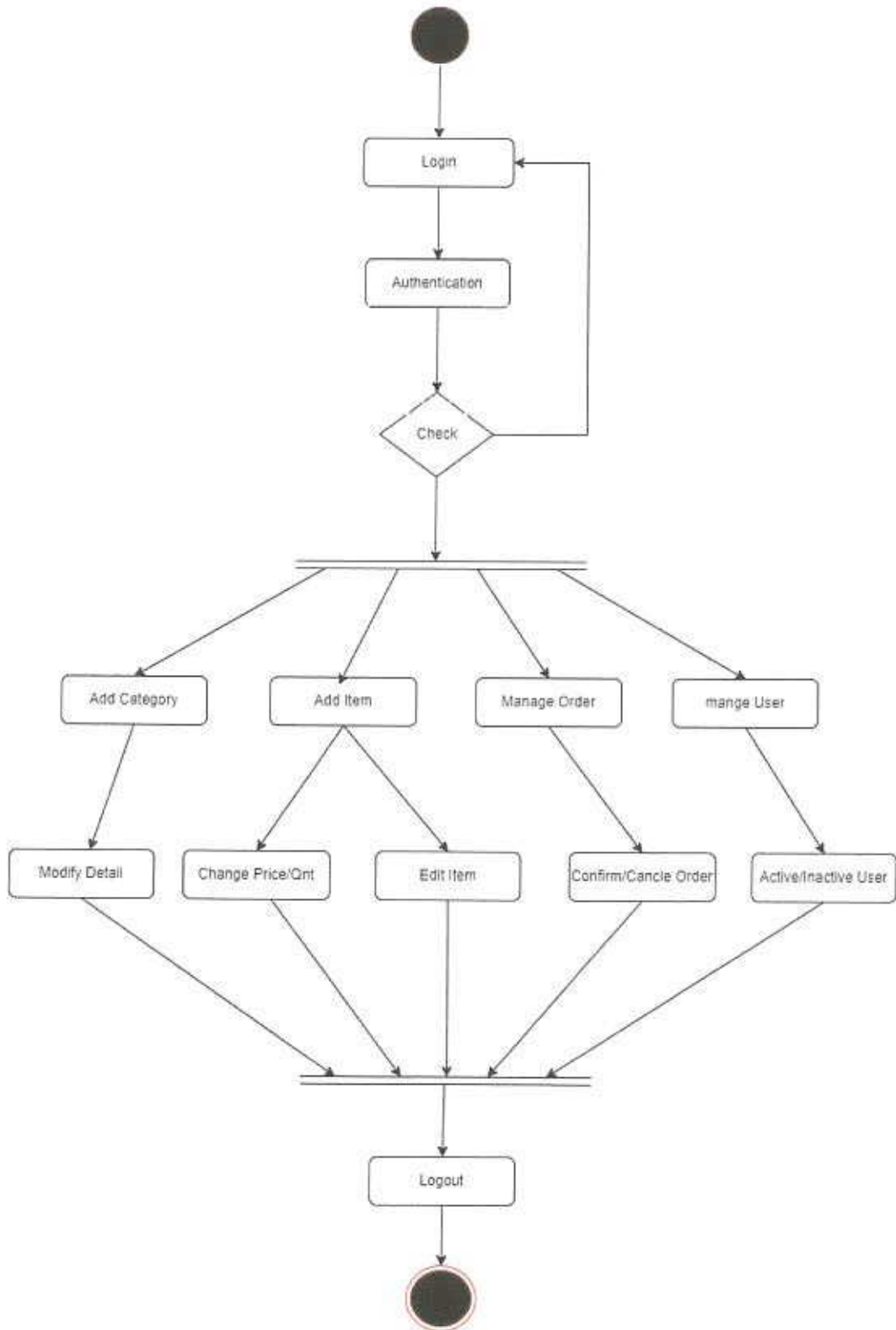




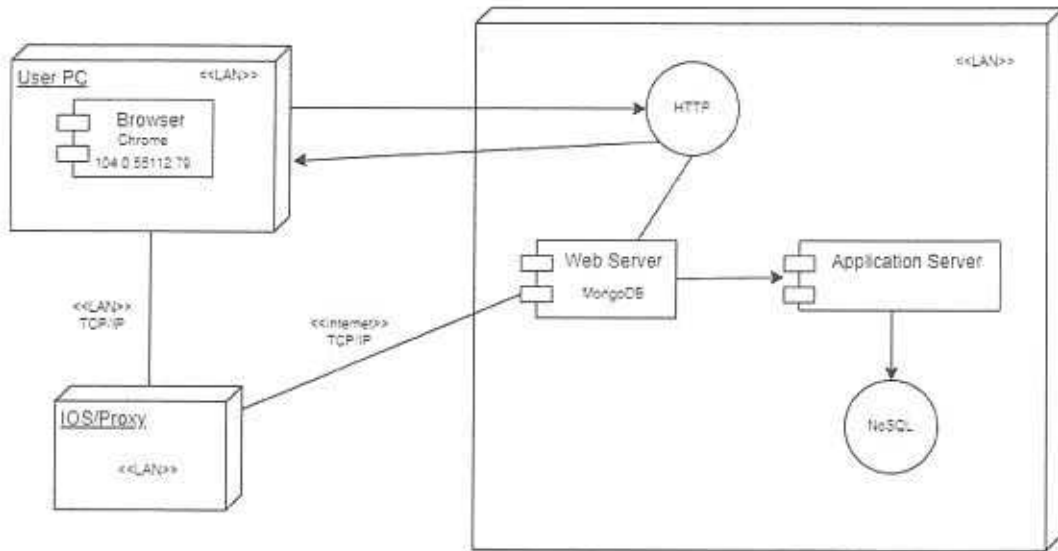
3.5 Class Diagram For E-Commerce Website:



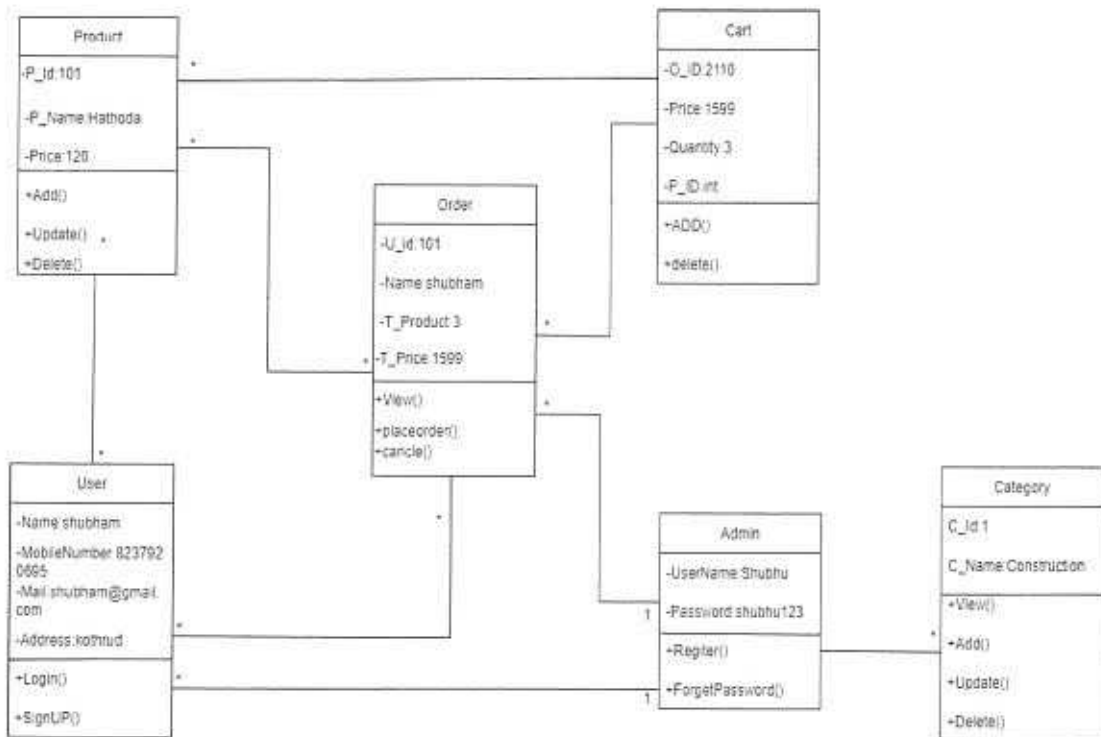
3.6 Activity Diagram For E-Commerce Website:

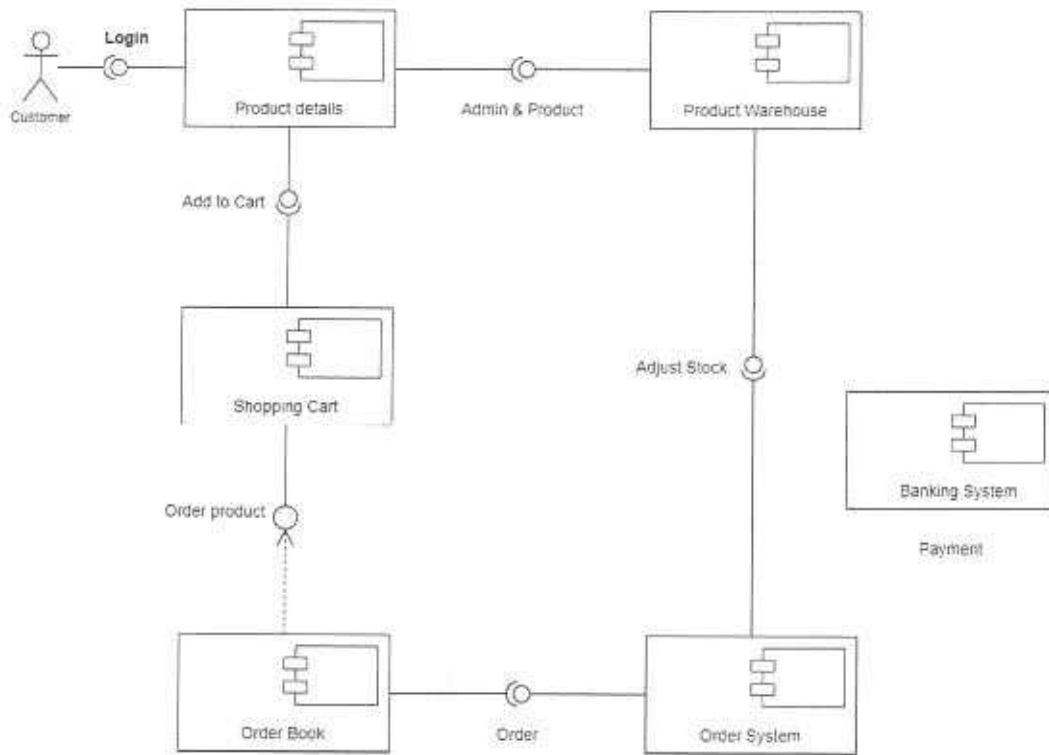


3.7 Deployment Diagram For E-Commerce Website:



3.8 Object Diagram For E-Commerce Website:

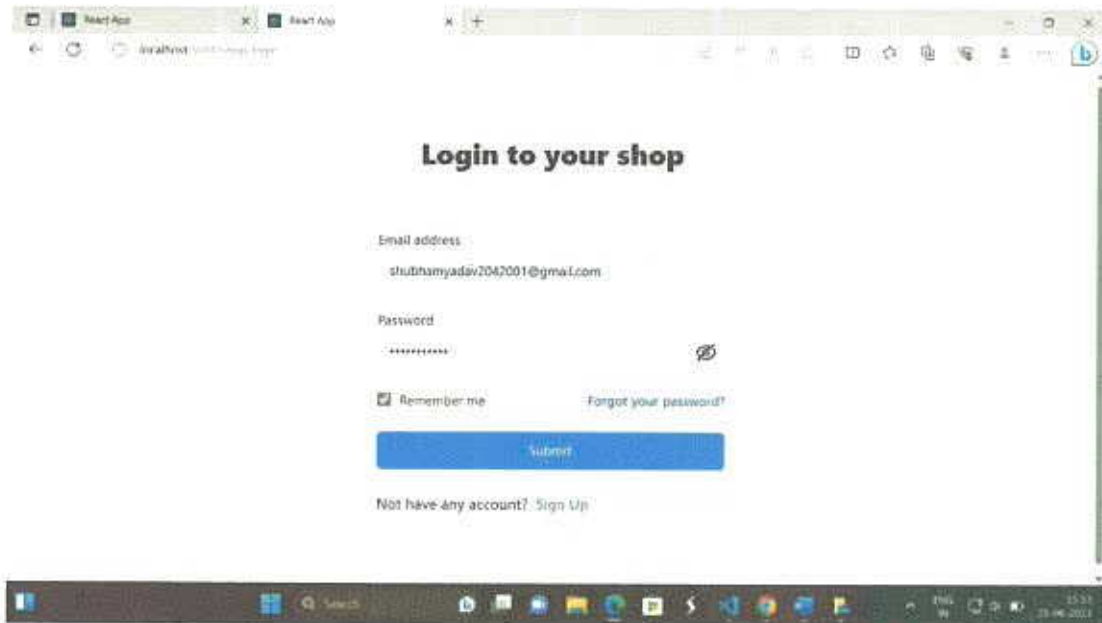


Component Diagram for Yukti Group :

3.9 Sample Input and Output Screens:

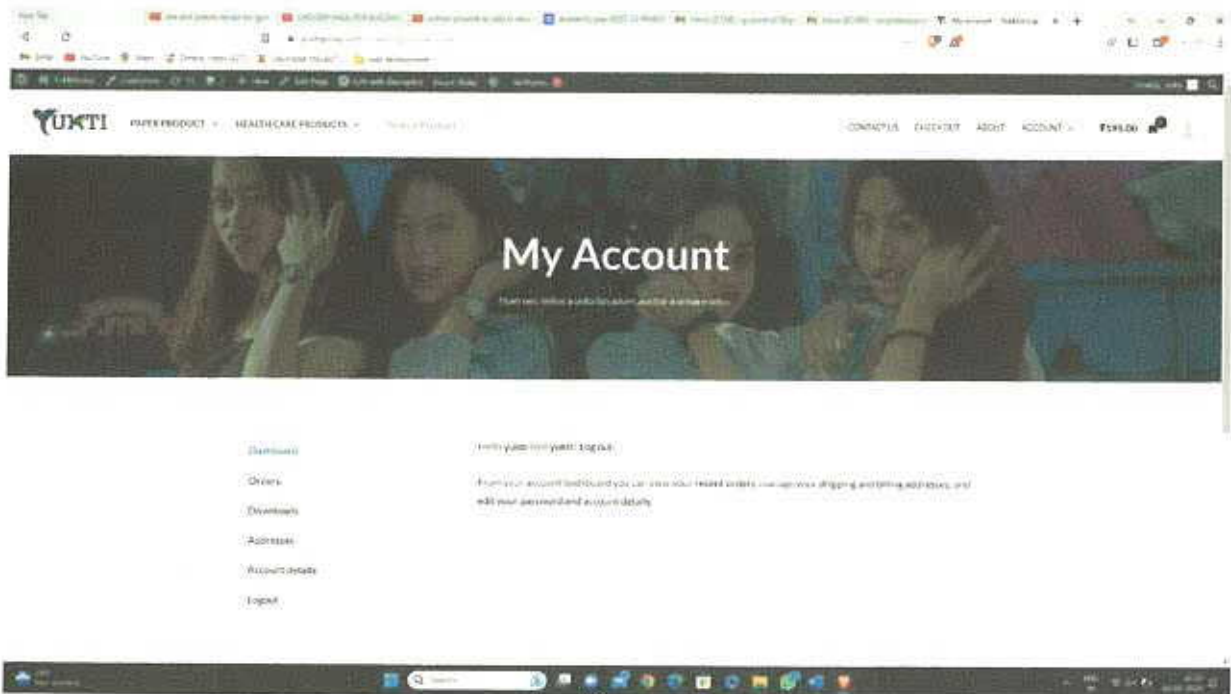
Admin Login:

This is admin login page where admin can login himself on system.



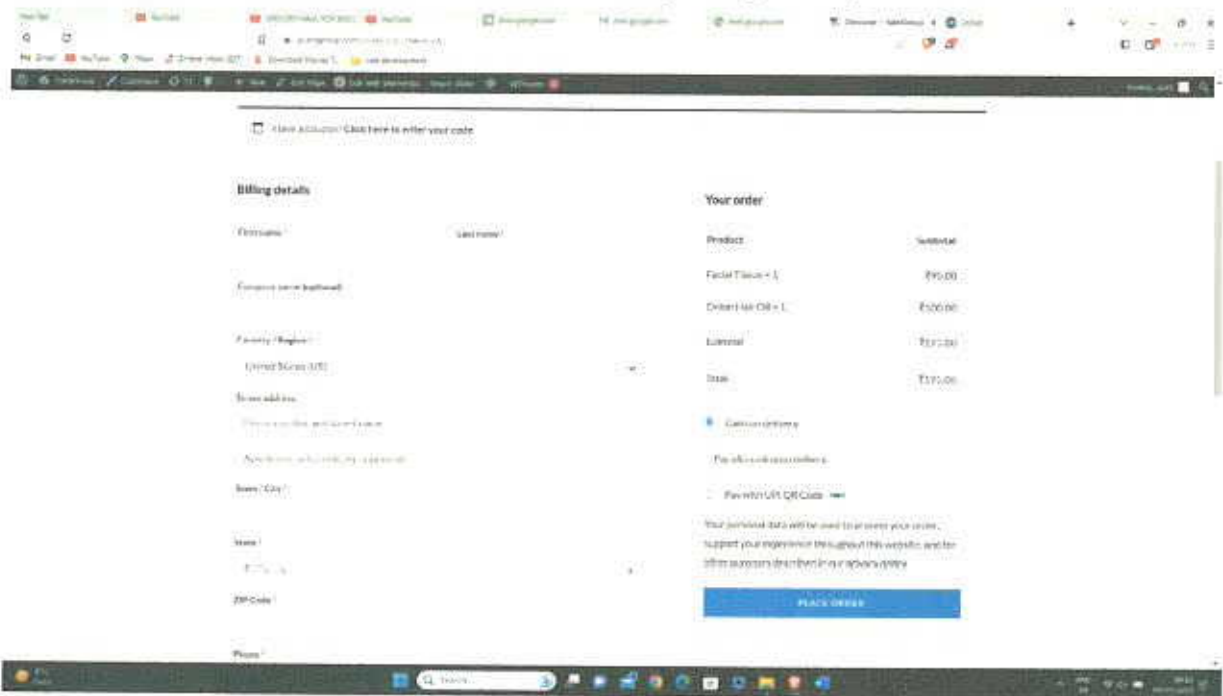
Admin Dashboard:

Admin can see the all records in dashboards.



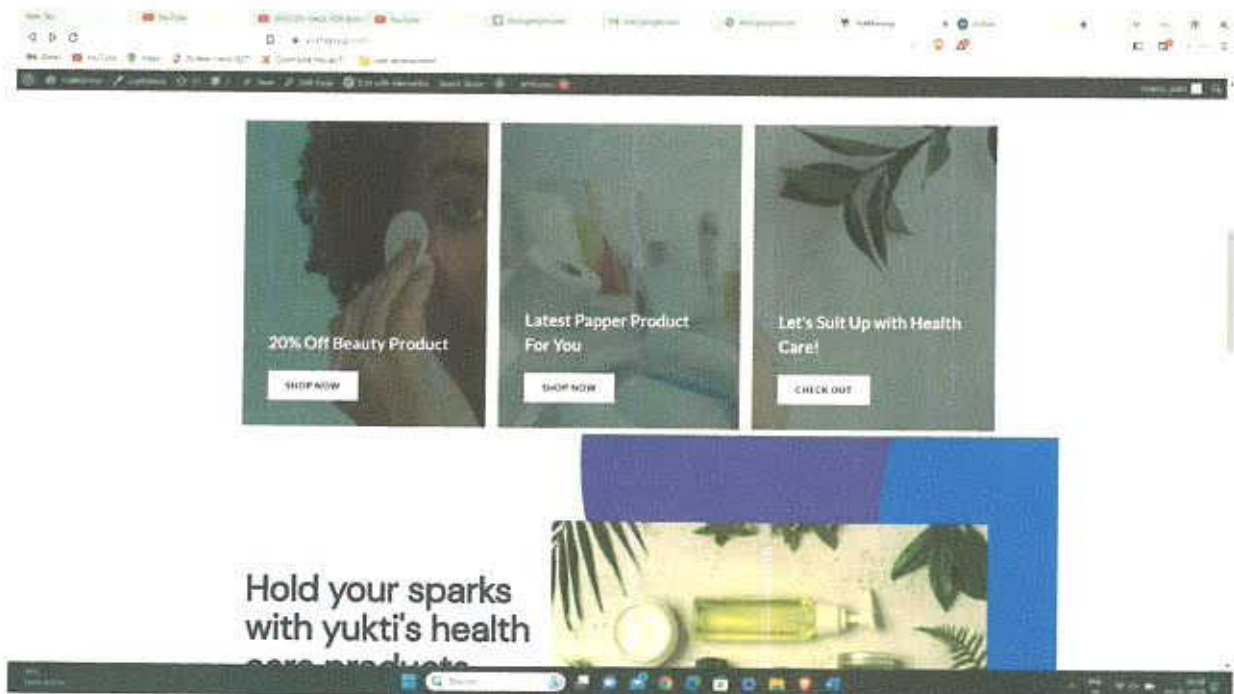
Placed Order:

Admin can see revived orders and amount that are pending or completed.



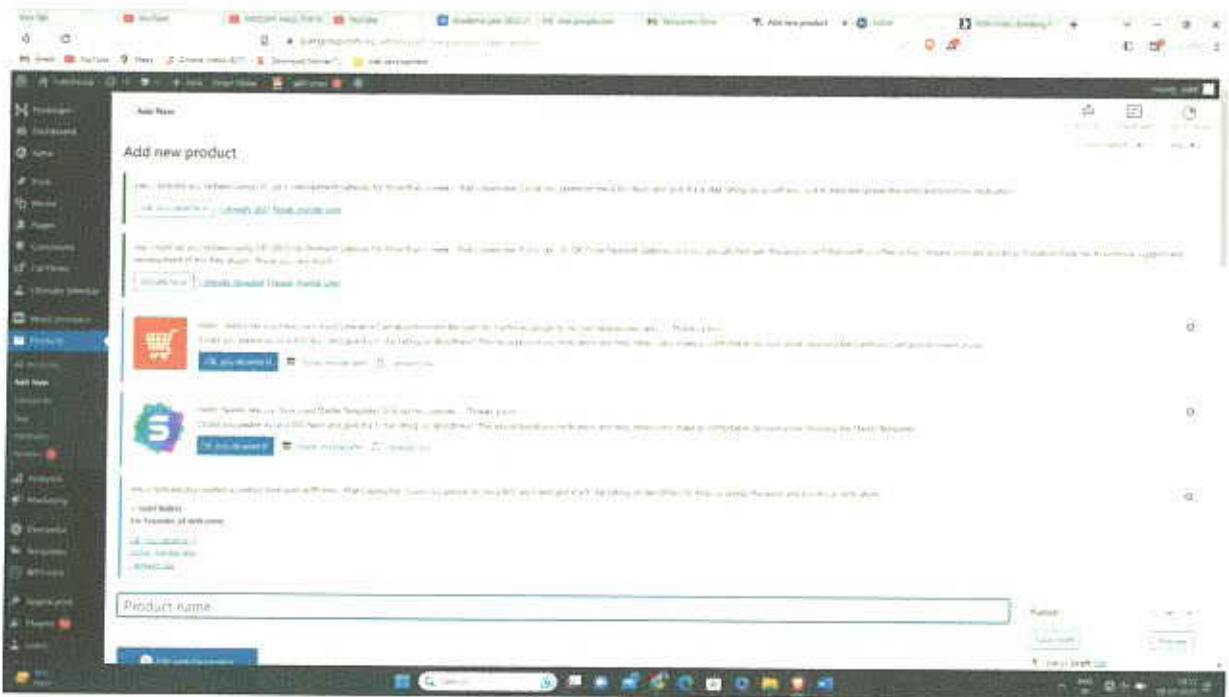
Product Details:

Admin can check and delete the products details as per requirement.



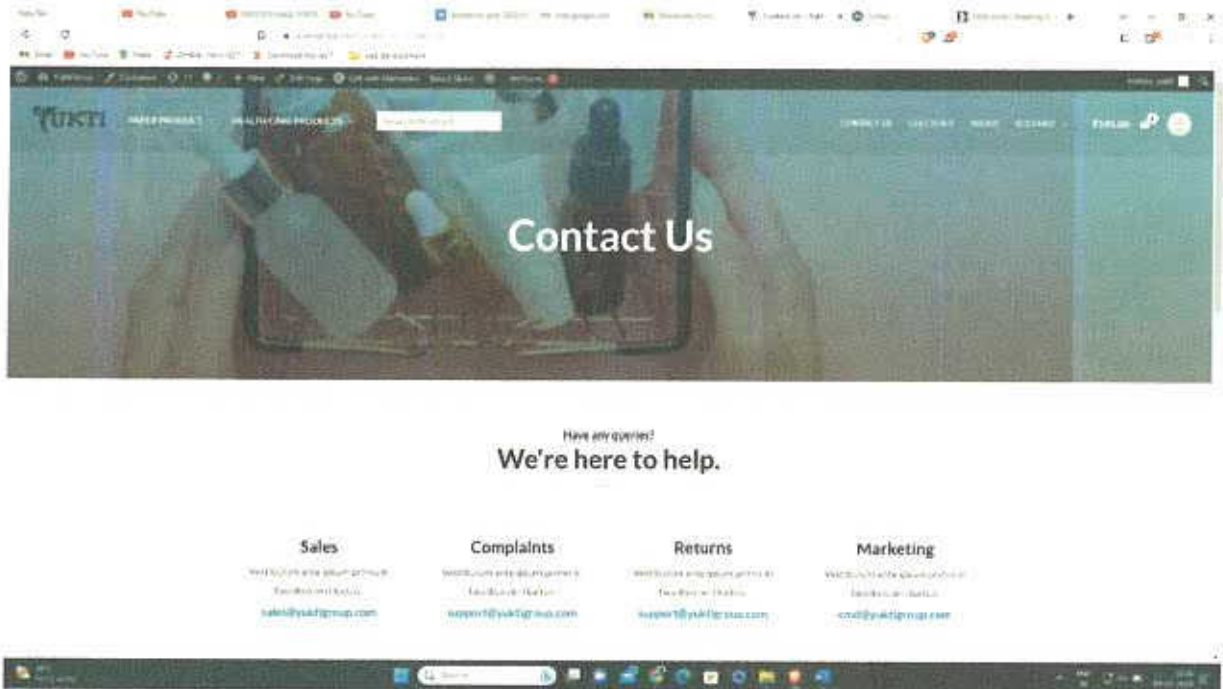
Add Product:

Admin can add products details.



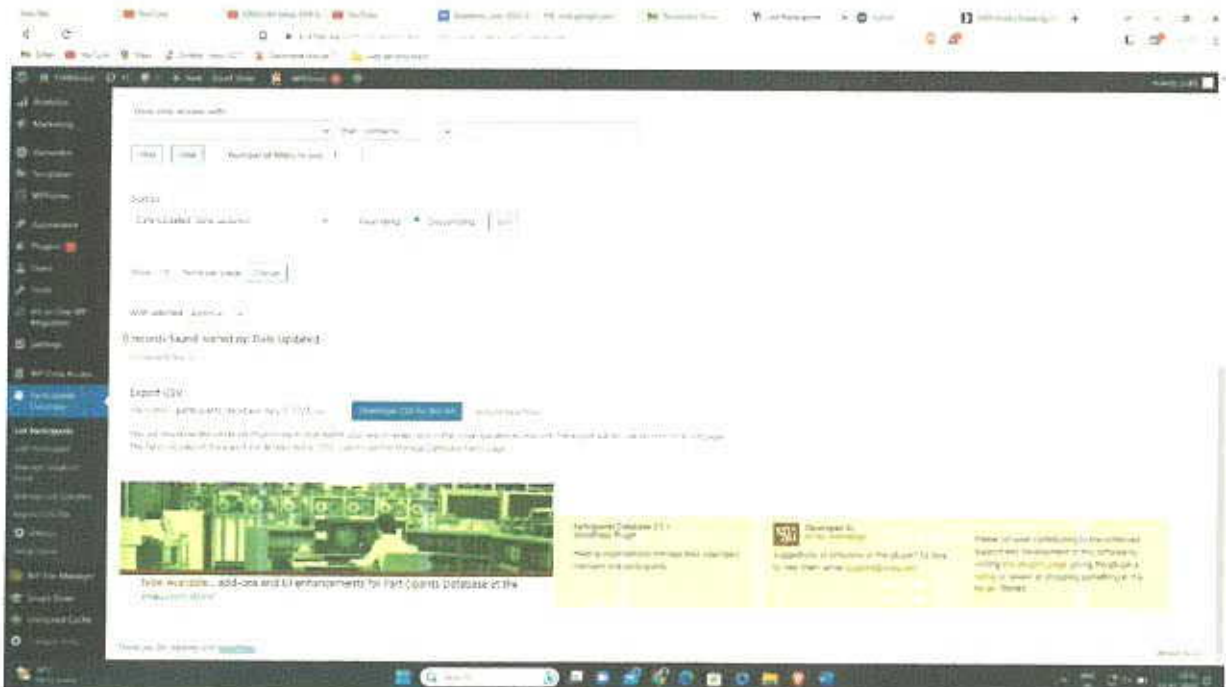
Messages:

Admin can see user to received messages.



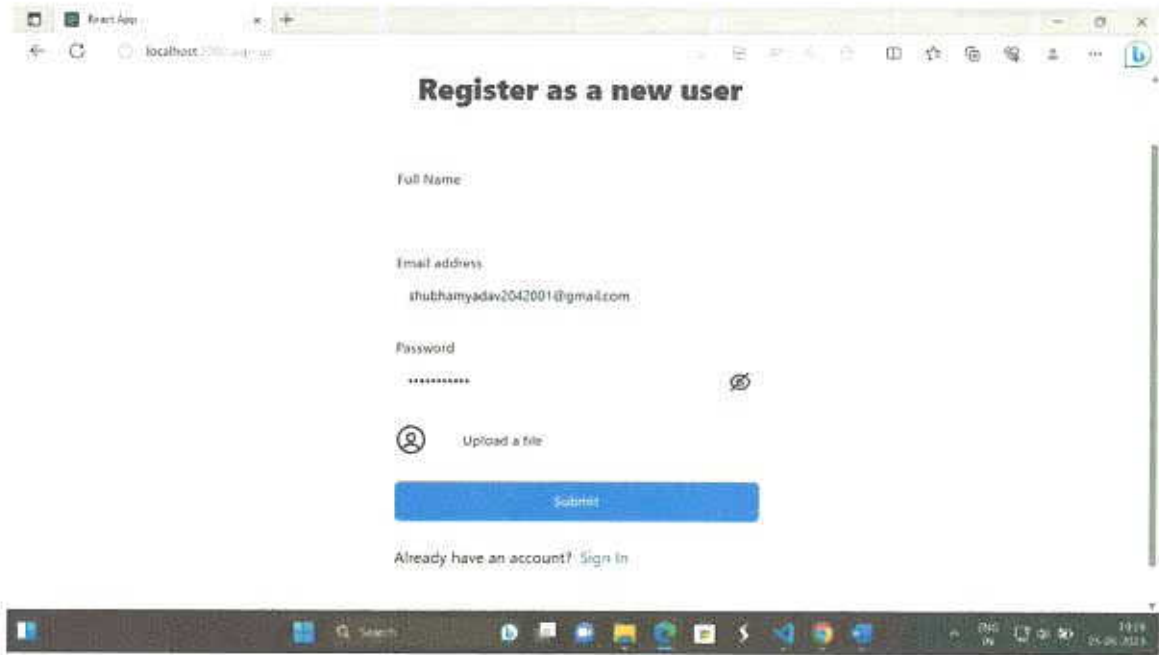
Discount Codes:

Admin can create coupon code and see all coupons.



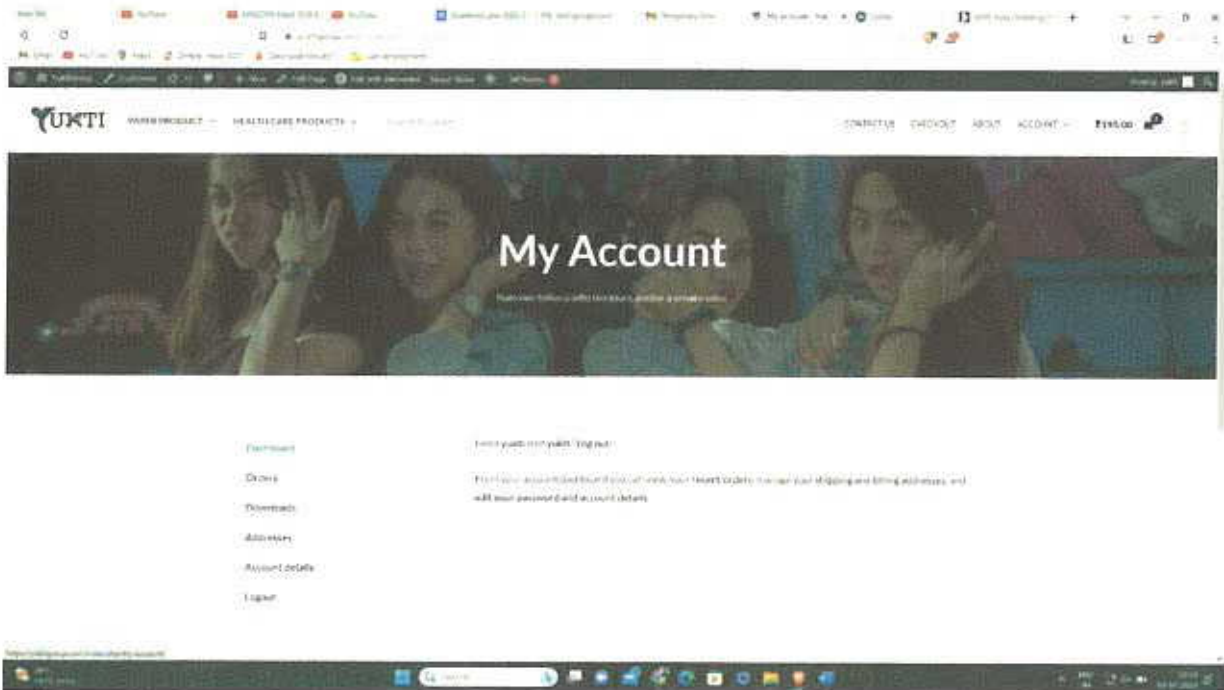
User register:

In this slide shows the registration of the customer. Customer can register account from here than login.



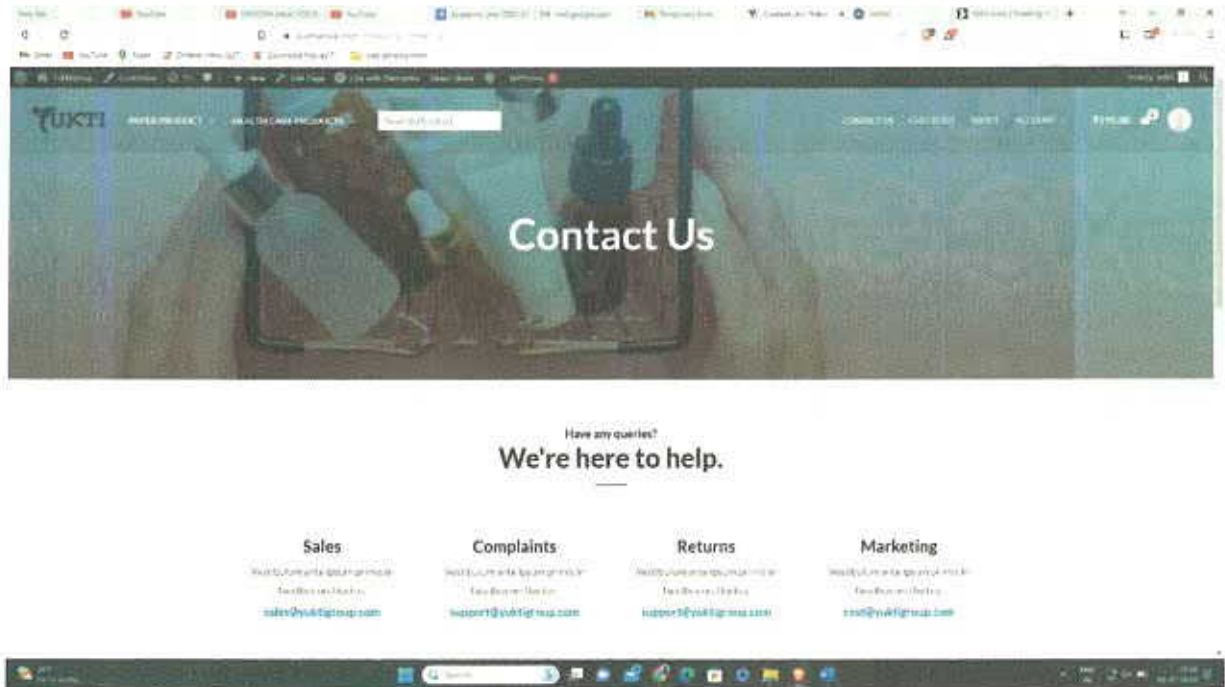
Profile:

User can see your profile.



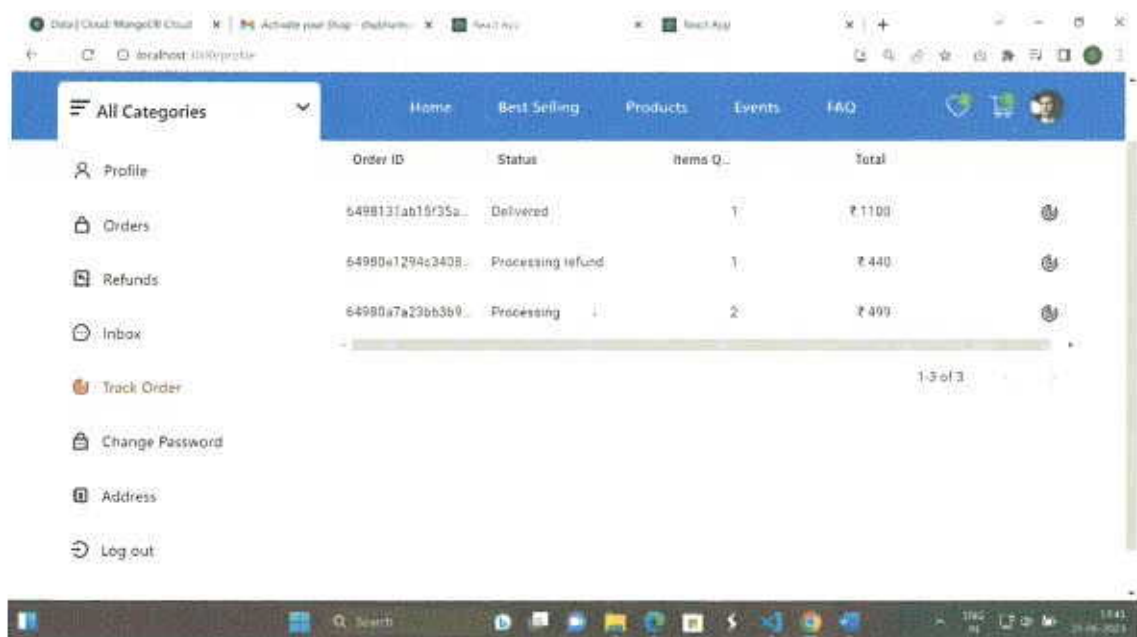
Messages:

The below attached page is to message the enterprise where user can send complaints, suggestions or queries.



Track Order:

Customer can track orders.



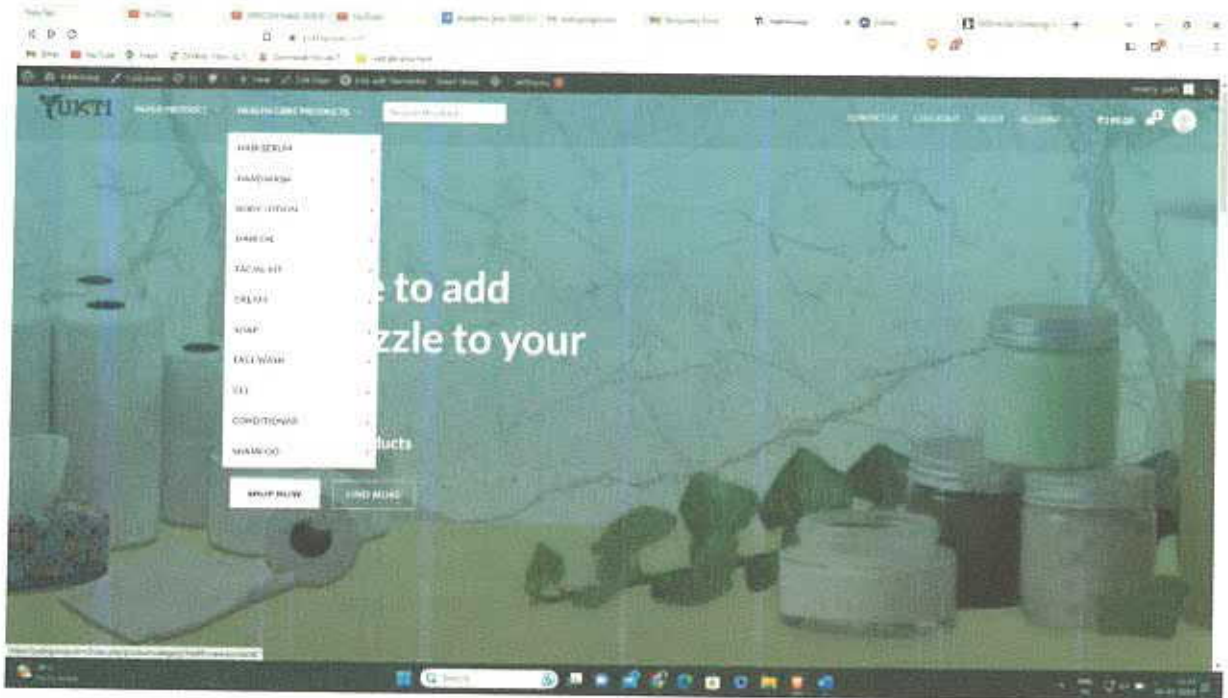
Home page:

This is home page of this website. Customer can see product category



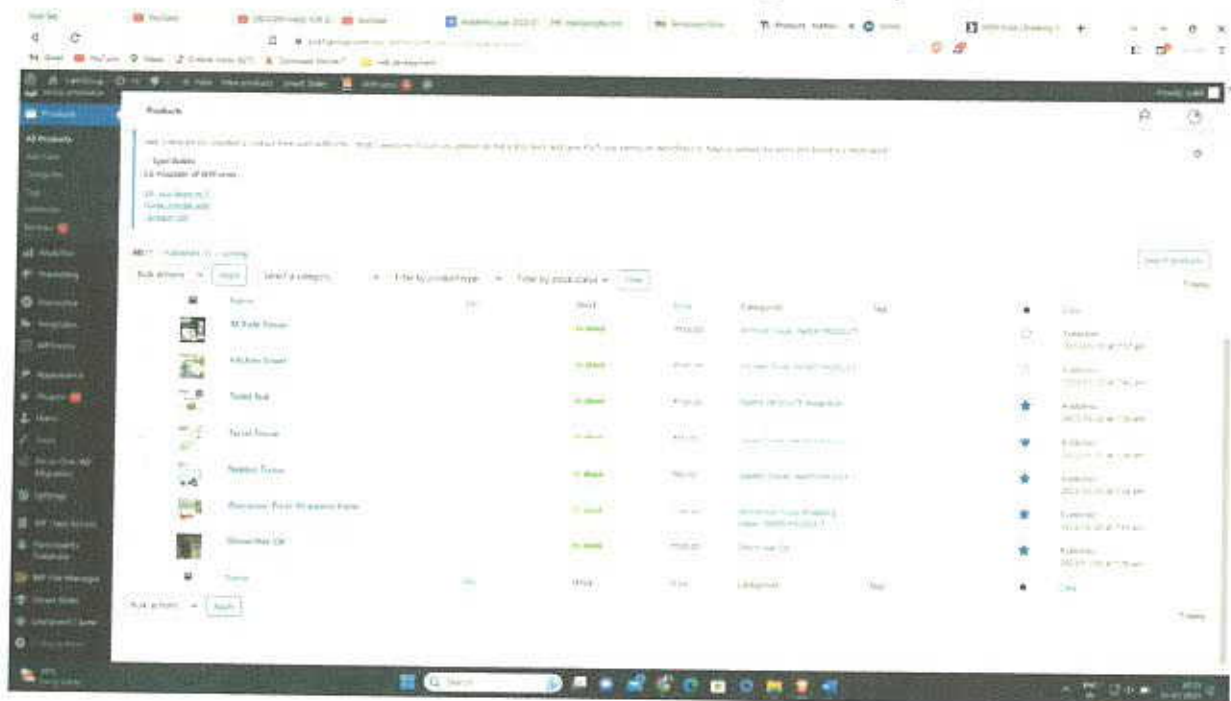
Categories:

User can see the all categories in home page.



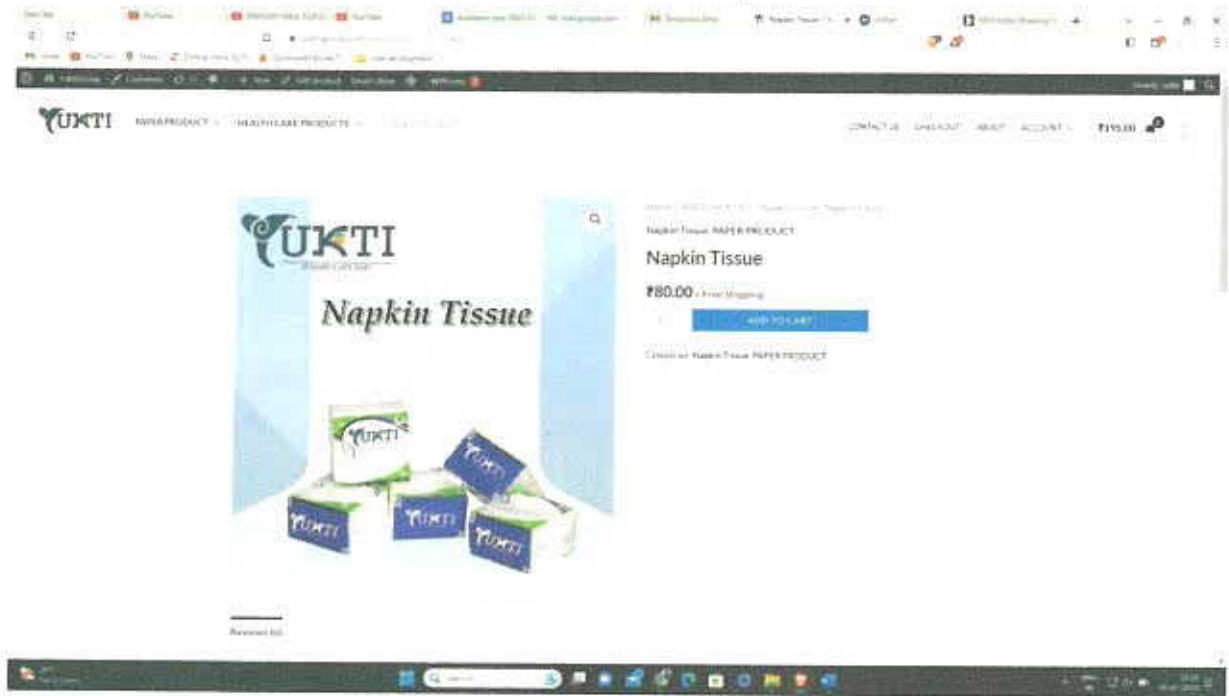
Products:

On this page, all products and their details are shown along with the prices.



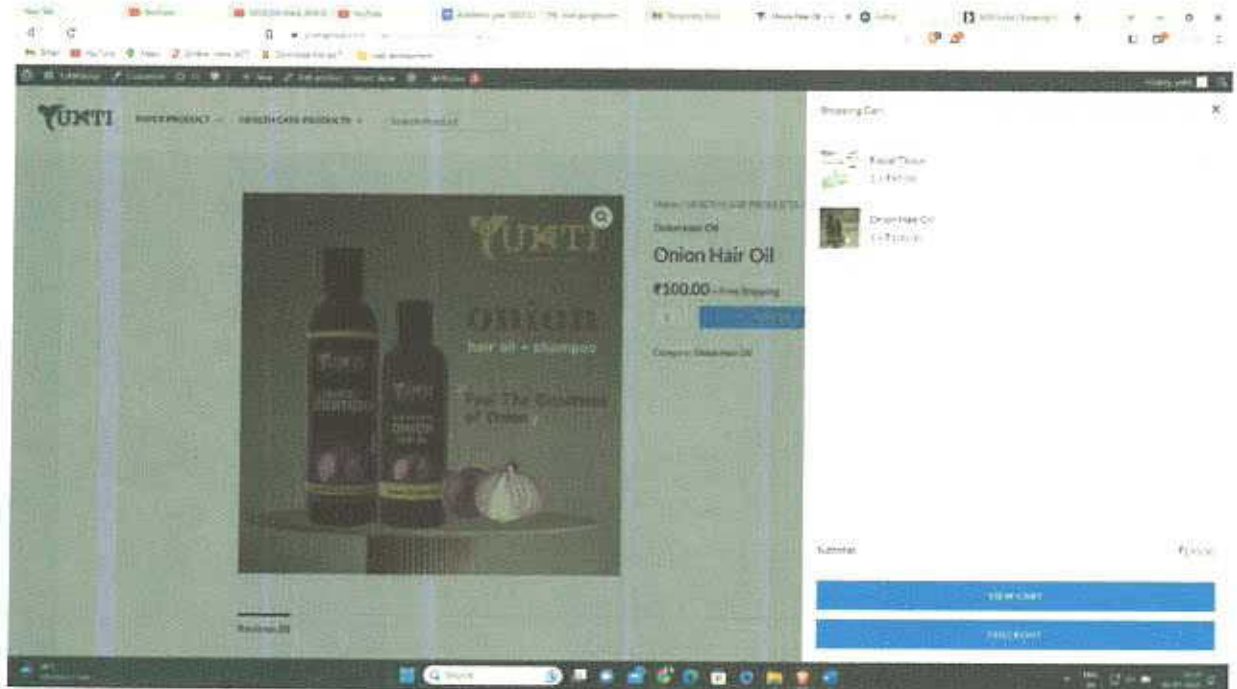
Events:

User can see and buy the special offers for particulate day.



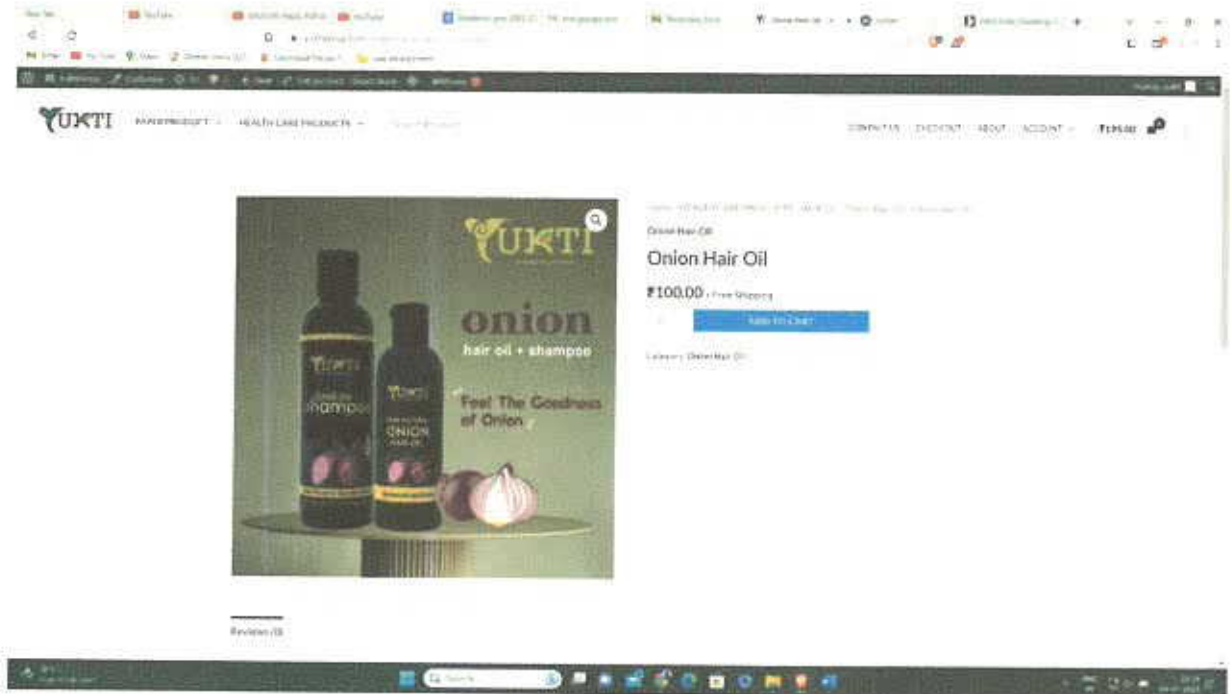
Add to Cart:

The below given page is to display shopping cart where the products added to cart can be seen.



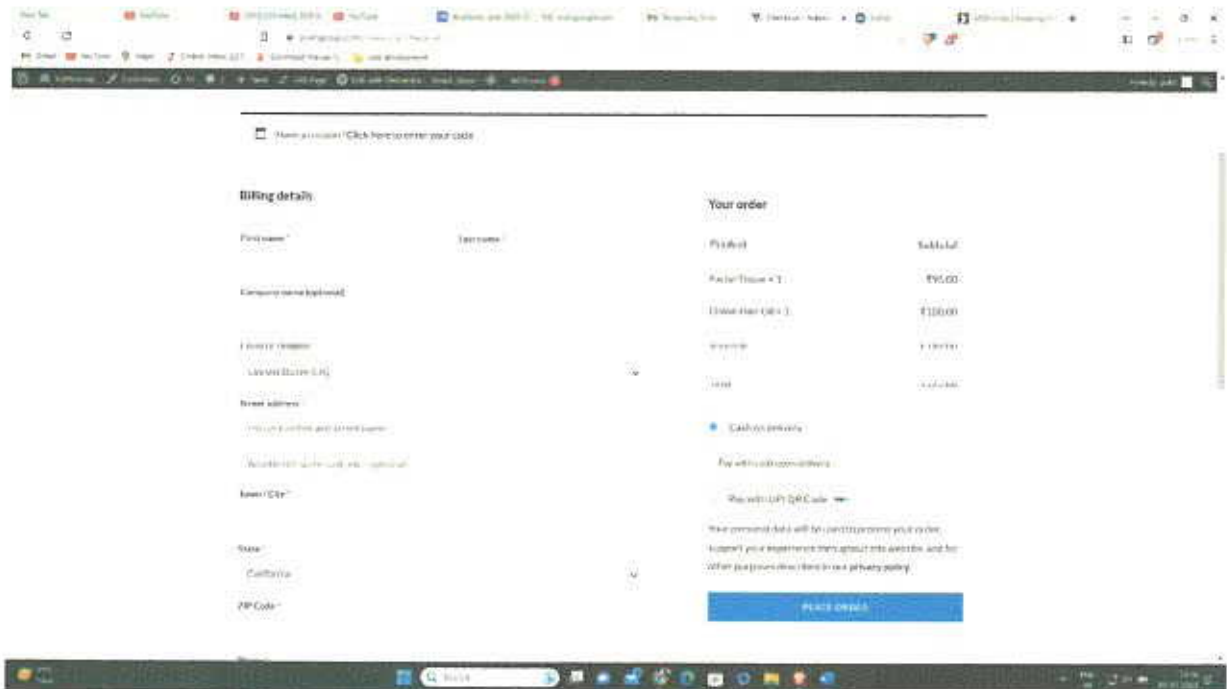
Quick View:

The below given page is for getting a quick view of product where user can see the product and decide to add to cart or buy product.



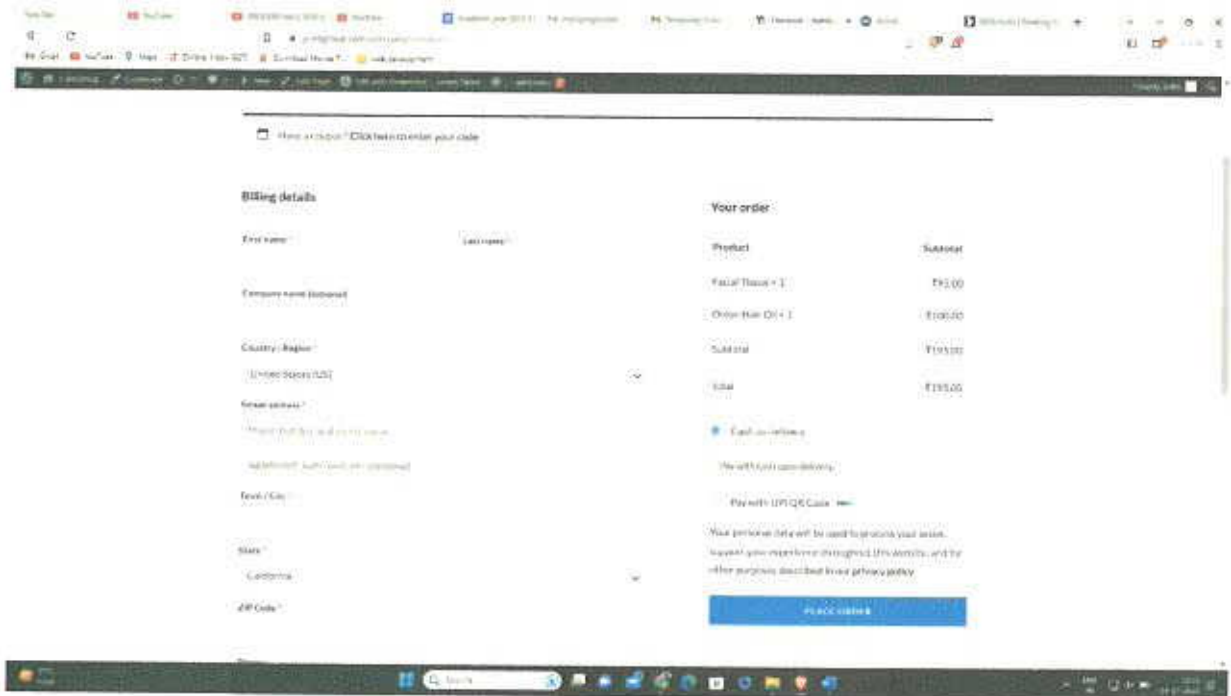
Place Order:

The below attached page is for placing orders which takes address and total of bill.



Payment:

User can pay the total amount via multiple payment method.



4.1 Code Snippets :

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name=
      "descri
      ption"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />

    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />

    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>

  </body>
</html>
```

5. Testing:

5.1 Testing Strategy:

A testing strategy is a plan for evaluating the quality and functionality of a software system. It outlines the approach to be taken for testing, the types of testing to be performed, the resources needed, and the schedule for testing. A good testing strategy helps ensure that the software meets its requirements, is free of defects, and is delivered on time. Key components of a testing strategy include:

1. **Objectives:** The objective of the testing should be to ensure that the e-commerce application is functioning correctly and meets the requirements set out by stakeholders.
2. **Test Types:** Types of testing that should be considered include functional testing, performance testing, security testing, compatibility testing, and usability testing.
3. **Test Environment:** The test environment should mirror the production environment as closely as possible.
4. **Test Data:** Test data should be created to accurately reflect a variety of real-world scenarios. This data should include products, customers, and payment methods.
5. **Test Procedures:** Detailed test plans and test cases should be developed that cover all the major functionality of the e-commerce application, including product search and display, product detail pages, shopping cart functionality, checkout process, and order confirmation.
6. **Test Scheduling:** The testing schedule should take into account the overall project timeline, as well as any deadlines for releasing new features or bug fixes.
7. **Test Resources:** A team of experienced testers should be put in place, along with tools such as automated testing software and load testing tools.
8. **Test Deliverables:** The output of testing should include detailed test results, bug reports, and test documentation.

5.2 Unit Test Plan:

A Unit Test Plan is a document that outlines the testing strategy for individual units or components of a software system. It provides a detailed description of the tests that will be performed, the testing environment, and the resources required. The main purpose of a Unit Test Plan is to ensure that each unit of the software system is thoroughly tested and meets the specified requirements before it is integrated into the overall system. The Unit Test Plan should include the following elements:

1. **Objectives:** The objectives of the unit testing should be to validate the functionality of individual components of the e-commerce application, such as product listings, shopping cart, and checkout process, to ensure they are working as expected.
2. **Test Types:** The types of unit testing that should be performed include functional testing, integration testing, and regression testing.
3. **Test Environment:** The test environment should include a development server and a test database that can be populated with test data.
4. **Test Data:** Test data should be created to accurately reflect a variety of real-world scenarios and should include products, customers, and payment methods.
5. **Test Procedures:** Detailed test plans and test cases should be developed for each unit of the e-commerce application, including product listings, shopping cart, checkout process, and order confirmation. The test plans should include test inputs, expected outputs, and the steps needed to perform the tests.
6. **Test Resources:** A team of experienced testers and automated testing software should be put in place to perform the unit testing.
7. **Test Deliverables:** The output of the unit testing should include detailed test results, bug reports, and test documentation.
8. **Test Status and Reporting:** The status of the unit testing should be regularly reported to stakeholders, including progress, issues encountered, and any changes to the testing schedule. Metrics such as test coverage, bug count, and test pass/fail rates should be used to measure progress.

5.3 Acceptance Test Plan

Test Plan Title: Acceptance Test Plan for E-commerce Website

1. Introduction:

The acceptance test plan outlines the approach and procedures for testing the e-commerce website to ensure its compliance with specified requirements and readiness for deployment. This plan focuses on validating the functionality, usability, performance, security, and integration aspects of the website.

2. Test Objectives:

- Validate the functionality of the e-commerce website, including product browsing, search, shopping cart management, checkout process, and order placement.
- Verify the usability and user-friendliness of the website, ensuring intuitive navigation, clear instructions, and responsive design.
- Evaluate the performance of the website, including response time, scalability, and concurrent user handling.
- Ensure the security of the website, including protection against unauthorized access, data breaches, and secure payment transactions.
- Validate the integration of the website with external systems such as payment gateways, inventory management systems, and shipping services.
- Verify compatibility with different browsers, devices, and operating systems.

3. Test Scope:

The acceptance testing will cover the following aspects of the e-commerce website:

- User registration and login
- Product browsing and search functionality
- Product details and reviews

4. Test Approach:

The acceptance testing will follow a combination of manual and automated testing approaches. It will involve the following steps:

- Test Case Identification: Identify test scenarios and create test cases to cover all functionalities.
- Test Data Preparation: Prepare test data, including sample products, user accounts, and orders.
- Test Execution: Execute test cases, following predefined test scripts and instructions.
- Defect Reporting: Document any defects or issues encountered during testing, including steps to reproduce and screenshots.
- Test Result Analysis: Analyze the test results to determine the website's compliance with specified requirements.
- Test Completion: Prepare a test summary report, including overall test coverage, results, and any outstanding issues.

5. Test Environment:

The test environment will include:

- E-commerce website deployed on appropriate web servers and databases
- Different browsers (e.g., Chrome, Firefox, Safari) and versions
- Various devices (e.g., desktop, laptop, mobile, tablet) and operating systems
- Test accounts with different user roles (e.g., admin, customer)

6. Test Procedures:

The following test procedures will be followed during the acceptance testing:

- Test Case Preparation: Identify and create test cases based on functional requirements.
- Test Data Setup: Prepare relevant test data, including sample products, user accounts, and orders.
- Test Execution: Execute test cases, recording observations and outcomes.
- Defect Reporting: Document any defects encountered, providing detailed information and prioritization.
- Defect Resolution: Work with the development team to address reported defects.
- Retesting: Verify that resolved defects have been fixed successfully.
- Test Completion: Prepare a test summary report, including test coverage, results, and any open issues.

7. Test Schedule:

The acceptance testing will be conducted according to the following schedule:

- Test Planning and Preparation: [Start Date] - [End Date]
- Test Execution: [Start Date] - [End Date]
- Defect Reporting

Resolution: Ongoing throughout the testing phase

- Retesting: [Start Date] - [End Date]
- Test Completion and Reporting: [End Date]
- Risks and Contingencies:

Identify potential risks that may impact the acceptance testing process, such as delays in test data preparation, dependencies on external systems, or resource constraints. Develop contingency plans to mitigate these risks and ensure the timely completion of testing activities.

8. Test Deliverables:

| | | | | | | | | |
|----|-------|------------------|-------------------------|--|--|--|-------------|------|
| 8 | TC008 | Checkout | Proceed to checkout | Verify that users can proceed to the checkout process | 1. View the contents of the cart 2. Click on the "Checkout" button | The user is redirected to the checkout page | As Expected | Pass |
| 9 | TC009 | Shipping Address | Enter shipping address | Verify that users can enter their shipping address during checkout | 1. Navigate to the checkout page 2. Fill in the shipping address details 3. Click on the "Next" button | The shipping address is successfully saved for the order | As Expected | Pass |
| 10 | TC010 | Payment Method | Select a payment method | Verify that users can choose a payment method during checkout | 1. Navigate to the checkout page 2. Select a payment method from the available options 3. Click on the "Next" button | The selected payment method is highlighted and saved for the order | As Expected | Pass |
| 11 | TC011 | Order Placement | Place an order | Verify that users can successfully place an order | 1. Complete the checkout process by entering shipping address, payment method, etc. 2. Click on the "Place | The order is successfully placed and a confirmation message is displayed | As Expected | Pass |

| | | | | | | | | |
|----|-------|------------------|-------------------------|--|--|--|-------------|------|
| 12 | TC012 | Order History | View order history | Verify that users can view their order history | <ol style="list-style-type: none"> 1. Navigate to the user account section 2. Click on the "Order History" link | The order history page displays a list of the user's previous orders | As Expected | Pass |
| 13 | TC013 | Order Details | View order details | Verify that users can view the details of a specific order | <ol style="list-style-type: none"> 1. Navigate to the order history page 2. Click on a specific order to view its details | The order details page displays the relevant information about the order | As Expected | Pass |
| 14 | TC014 | Account Settings | Update account settings | Verify that users can update their account settings | <ol style="list-style-type: none"> 1. Navigate to the account settings page 2. Make the desired changes to the account settings 3. Click on the "Save" button | The account settings are successfully updated | As Expected | Pass |
| 15 | TC015 | Logout | User logout | Verify that users can log out of their account | <ol style="list-style-type: none"> 1. Click on the "Logout" button or link | The user is logged out and redirected to the homepage or login page | As Expected | Pass |

| | | | | | | | | |
|----|-------|-----------------|-------------------------------|--|---|--|-------------|------|
| 16 | TC016 | Product Details | View product details | Verify that users can view the details of a specific product | 1. Navigate to the product listing page 2. Click on a specific product to view its details | The product details page displays the relevant information about the product | As Expected | Pass |
| 17 | TC017 | Product Reviews | Read product reviews | Verify that users can read reviews for a product | 1. Navigate to the product details page 2. Scroll down to the reviews section | The reviews section displays the reviews and ratings given by other users | As Expected | Pass |
| 18 | TC018 | Write Review | Write a product review | Verify that users can write a review for a product | 1. Navigate to the product details page 2. Scroll down to the reviews section 3. Click on the "Write a Review" button 4. Fill in the review form and submit | The review is successfully submitted and displayed in the reviews section | As Expected | Pass |
| 19 | TC019 | Add to Wishlist | Add a product to the wishlist | Verify that users can add products to their wishlist | 1. Navigate to the product details page 2. Click on the | The product is added to the wishlist successfully | As Expected | Pass |

| | | | | | | | | |
|----|-------|----------------------|------------------------------------|---|---|--|-------------|------|
| 19 | TC019 | Add to Wishlist | Add a product to the wishlist | Verify that users can add products to their wishlist | 1. Navigate to the product details page 2. Click on the "Add to Wishlist" button | The product is added to the wishlist successfully | As Expected | Pass |
| 20 | TC020 | View Wishlist | View the wishlist | Verify that users can view their wishlist | 1. Click on the "Wishlist" icon or link | The wishlist page displays the products added by the user | As Expected | Pass |
| 21 | TC021 | Remove from Wishlist | Remove a product from the wishlist | Verify that users can remove products from their wishlist | 1. View the contents of the wishlist 2. Click on the "Remove" button for a product 3. Confirm the removal | The product is successfully removed from the wishlist | As Expected | Pass |
| 22 | TC022 | Compare Products | Compare multiple products | Verify that users can compare the features of multiple products | 1. Select multiple products from the product listing page 2. Click on the "Compare" button | The products are displayed side by side with their respective features for easy comparison | As Expected | Pass |

| | | | | | | | | |
|----|-------|-----------------|---|--|---|---|-------------|------|
| 26 | TC026 | Sort Products | Sort products by price, popularity, etc. | Verify that users can sort the product listing based on different criteria | 1. Select a sort option from the sorting dropdown 2. Observe the changes in the product listing order | The product listing is sorted according to the selected criteria | As Expected | Pass |
| 27 | TC027 | Filter Products | Apply filters to narrow down product search | Verify that users can apply filters to refine their product search | 1. Select filter options from the available filter categories 2. Click on the "Apply Filters" button | The product listing is updated to display products that match the applied filters | As Expected | Pass |
| 28 | TC028 | Add to Compare | Add a product to the compare list | Verify that users can add a product to the compare list | 1. Navigate to the product details page 2. Click on the "Add to Compare" button | The product is added to the compare list successfully | As Expected | Pass |

| | | | | | | | | |
|----|-------|---------------------|---|---|---|--|-------------|------|
| 29 | TC029 | View Compare List | View the compare list | Verify that users can view the compare list | 1. Click on the "Compare" icon or link | The compare list page displays the products added by the user for comparison | As Expected | Pass |
| 30 | TC030 | Remove from Compare | Remove a product from the compare list | Verify that users can remove products from the compare list | 1. View the contents of the compare list 2. Click on the "Remove" button for a product 3. Confirm the removal | The product is successfully removed from the compare list | As Expected | Pass |
| 31 | TC031 | Sort Search Results | Sort search results by price, relevance, etc. | Verify that users can sort the search results based on different criteria | 1. Enter a search query in the search bar 2. Select a sort option from the sorting dropdown 3. Click on the search button 4. Observe the changes in the search results order | The search results are sorted according to the selected criteria | As Expected | Pass |

| | | | | | | | | |
|----|-------|----------------------------------|--|--|---|---|-------------|------|
| 32 | TC032 | Filter Search Results | Apply filters to refine search results | Verify that users can apply filters to narrow down their search results | <p>1. Enter a search query in the search bar </p> <p>2. Select filter options from the available filter categories </p> <p>3. Click on the search button </p> <p>4. Click on the "Apply Filters" button</p> | The search results are updated to display products that match the applied filters | As Expected | Pass |
| 33 | TC033 | Add to Cart from Product Details | Add a product to the cart from the product details page | Verify that users can add a product to the cart directly from the product details page | <p>1. Navigate to the product details page </p> <p>2. Click on the "Add to Cart" button</p> | The product is added to the cart successfully | As Expected | Pass |
| 34 | TC034 | View Cart Subtotal | View the subtotal of the cart | Verify that users can view the subtotal of the items in their cart | <p>1. View the contents of the cart </p> <p>2. Locate the subtotal section</p> | The cart displays the subtotal amount calculated based on the added items | As Expected | Pass |
| 35 | TC035 | View Cart Total | View the total amount of the cart including taxes and shipping charges | Verify that users can view the total amount of the cart including taxes and | <p>1. View the contents of the cart </p> | The cart displays the total amount including taxes and shipping | As Expected | Pass |

| | | | | | | | | |
|----|-------|--------------------|--|---|--|---|-------------|------|
| 35 | TC035 | View Cart Total | View the total amount of the cart including taxes and shipping charges | Verify that users can view the total amount of the cart including taxes and shipping charges | 1. View the contents of the cart 2. Locate the total section | The cart displays the total amount including taxes and shipping charges | As Expected | Pass |
| 36 | TC036 | Apply Coupon Code | Apply a coupon code during checkout | Verify that users can apply a coupon code to avail discounts | 1. Navigate to the checkout page 2. Enter a valid coupon code 3. Click on the "Apply" button | The coupon code is successfully applied and the discounted amount is reflected in the order total | As Expected | Pass |
| 37 | TC037 | View Order Summary | View the summary of the order during checkout | Verify that users can view the order summary including item details, prices, and quantities during checkout | 1. Navigate to the checkout page 2. Locate the order summary section | The order summary displays the relevant details and information for the items in the cart | As Expected | Pass |

38 TC038 Edit Cart Edit the items in the cart Verify that users can edit the items in the cart
 1. View the contents of the cart
 The cart is successfully updated with
As Expected
Pass

| | | | | | | | | |
|----|-------|----------------------------|--|--|---|--|-------------|------|
| 38 | TC038 | Edit Cart | Edit the items in the cart | Verify that users can edit the items in their cart | <ol style="list-style-type: none"> 1. View the contents of the cart 2. Update the quantity or remove items as desired 3. Click on the "Update Cart" button | The cart is successfully updated with the changes made by the user | As Expected | Pass |
| 39 | TC039 | Guest Checkout | Place an order as a guest user | Verify that guest users can place an order without creating an account | <ol style="list-style-type: none"> 1. Add items to the cart 2. Proceed to checkout 3. Fill in the required details including shipping address and payment information 4. Click on the "Place Order" button | The order is successfully placed without requiring the user to create an account | As Expected | Pass |
| 40 | TC040 | Add Multiple Items to Cart | Add multiple items to the cart at once | Verify that users can add multiple items to the cart simultaneously | <ol style="list-style-type: none"> 1. Select multiple products from the product listing page 2. Click on the "Add to Cart" | All the selected items are successfully added to the cart | As Expected | Pass |

| | | | | | | | | |
|----|-------|----------------------------|---|---|---|--|-------------|------|
| 40 | TC040 | Add Multiple Items to Cart | Add multiple items to the cart at once | Verify that users can add multiple items to the cart simultaneously | <ol style="list-style-type: none"> 1. Select multiple products from the product listing page 2. Click on the "Add to Cart" button for all selected products 3. View the cart | All the selected items are successfully added to the cart | As Expected | Pass |
| 41 | TC041 | Cancel Order | Cancel a placed order | Verify that users can cancel a placed order | <ol style="list-style-type: none"> 1. Navigate to the order details page of a placed order 2. Click on the "Cancel Order" button 3. Confirm the cancellation | The order is successfully cancelled and a cancellation confirmation message is displayed | As Expected | Pass |
| 42 | TC042 | Contact Customer Support | Contact customer support for assistance | Verify that users can contact customer support for assistance | <ol style="list-style-type: none"> 1. Locate the customer support contact information 2. Use the provided contact method (phone, email, chat, etc.) to contact customer | The user is able to successfully reach out to customer support for assistance | As Expected | Pass |

| | | | | | | | | |
|----|-------|----------------------|---|---|--|--|-------------|------|
| 43 | TC043 | Track Order | Track the status of an order | Verify that users can track the status of their placed order | <ol style="list-style-type: none"> 1. Navigate to the order details page of a placed order 2. Locate the order tracking section 3. Enter the tracking number or order details to track the status | The order status successfully retrieved and displayed to the user | As Expected | Pass |
| 44 | TC044 | Social Media Sharing | Share a product on social media platforms | Verify that users can share a product on social media platforms | <ol style="list-style-type: none"> 1. Navigate to the product details page 2. Click on the social media sharing icons (Facebook, Twitter, Instagram, etc.) | The product is shared on the respective social media platform successfully | As Expected | Pass |
| 45 | TC045 | Mobile Responsive | Test website responsiveness on mobile devices | Verify that the website is responsive and functions correctly on mobile devices | Access the website using various mobile devices (smartphone, tablet, etc.) | The website displays correctly and all functionalities work properly on mobile devices | As Expected | Pass |

| | | | | | | | | |
|----|-------|-----------------------|--|--|--|--|-------------|------|
| 46 | TC046 | Browser Compatibility | Test website compatibility across different browsers | Verify that the website is compatible with popular web browsers | 1. Access the website using different web browsers (Chrome, Firefox, Safari, Edge, etc.) 2. Perform various actions such as registration, search, add to cart, etc. | The website displays correctly and all functionalities work properly across different browsers | As Expected | Pass |
| 47 | TC047 | Load Time | Test website load time | Verify that the website loads quickly and efficiently | 1. Access the website using different network connections (3G, 4G, Wi-Fi, etc.) 2. Measure the time taken to load the website and its pages | The website loads within an acceptable time frame for different network connections | As Expected | Pass |
| 48 | TC048 | Error Handling | Test error handling scenarios | Verify that error messages are displayed appropriately for different error scenarios | 1. Intentionally trigger various error scenarios (e.g., invalid login credentials, payment failure, page not found, etc.) | The error messages accurately describe the error and guide the user on how to resolve it | As Expected | Pass |

Bibliography :

1) Books :

- React – Accomazzo – The Rode to learn React
- Object Oriented Analysis and Design.
- UML – By Jason T Roff
- Software Project Management – By Bob Hughes, Mike Cotterell
- Node - Craig Buckler - first edition (July, 2022)
- UML 2.0: A programmatic introduction to UML : O'Reilly; 1st edition (16 May 2006)

2) Websites :

- www.tutorialspoint.com
- www.stackoverflow.com
- www.geekforgeeks.com
- www.W3schools.com
- www.W3layouts.com