A Project Report

On

AI Based E-Commerce with Chat-Bot

Submitted by

HARSHIT S – 20BCE2407

NIKHIL AGARWAL – 20BCE2154

VIPIN C – 20BCE2160

UJJWAL AKASH – 20BCE2202

SHASHWAT ASHAR– 19BCE0808

For

Software Engineering

CSE3001

Slot: B1, J Component

B.Tech. in Computer Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

November 2022

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Introduction

The way of life in modern times has changed. People find it time-consuming and uncomfortable to visit crowded markets. E-shopping is thus a blessing because it saves a lot of time. Online shopping is a practise where customers deal directly with sellers to purchase products, services, etc. Customers can browse online stores while lounging in their homes or while seated in front of a computer. Many customers have access to the internet both at work and at home, and online retailers are typically open 24 hours a day. Online shopping is therefore incredibly convenient for them. Online shopping is especially alluring around the holidays since it eliminates the need to stand in line or look around a store for a certain item. Online retailers offer a variety of products.

This application's goal is to educate users on e-commerce and demonstrate how an interactive e-commerce application can be created from scratch. It includes all the implementation necessary for database setup, building session models to link various user interface (UI) pages, figuring out the price of placed orders, etc. By mapping the category or item Id to the corresponding IDs contained in the database, it is in charge of collecting information from the database and making it available to the UI.

The application of artificial intelligence in online purchasing is revolutionizing the e-commerce sector by forecasting consumer behavior based on the goods and times that they purchase. For instance, if a certain brand of rice is commonly purchased by online customers each week, the store could give these customers a customized offer for this product or even make a recommendation for a complementary item that complements rice dishes.

Artificial intelligence (AI) solutions for e-commerce or AI-enabled digital assistants, like the Google Duplex tool, are gaining the ability to make grocery lists using a customer's natural speech and even place online orders on their behalf.

1.2 Problem Definition

Build e-commerce website which has user friendly interface for both the users and the system admin and create a database which will contain the details of the users and the products to be registered. Once a user clicks checkout generation of the bill should be procured. A one click checkout function which will redirect the user to the payment gateway. Create AI based Chat-bot which will interact with the customers for any quires.

1.3 Project Scope

Electronic commerce, also known as e-commerce is a type of industry were buying and selling of a product is conducted over electronic system such as the internet.

E-commerce has boomed over the years and is one of the fastest-growing domains in the online world. Though it took some time for this to be accepted by the end-users, today we are at a point where the majority of the people love to shop online.

In this project we are developing a business to consumer (B2C) application in which the consumer can select the items from the catalogue and add to cart to buy the products. The system generates the billing and consumer can pay through payment gateway. If the consumer needs any assistance he can interact with the chat-bot.

1.4 Motivation

In this project, we'll create a business-to-consumer (B2C) application that allows users to browse a catalogue, choose what they want, and then put it to their shopping cart to purchase it. The system creates the invoices, which the customer can pay for using a payment gateway. One of the most important elements that will contribute to the company's expansion is customer service.

A quick customer support team is not present in the majority of e-commerce systems. Therefore, we have made the decision to create an AI-powered chatbot that will accurately answer consumers' questions in a matter of minutes. This will make human involvement unnecessary, saving many man hours of labour.

1.5 Background Study/Literature Survey

1. The goal of this paper was to create Hebron, a web-based chatbot for the Covenant University Community Mall. The aim of this work is the design and implementation of a chatbot for Covenant

University Shopping Mall. The goal of the chatbot was to converse with the pupils in real time while being knowledgeable and accurate. Students may do this by chatting with the bot to get information about specific things they want to buy and making payments online before they go to the mall. The distress felt by members of the Covenant University Community when they visit to CUSM to source for things only to discover that the desired items are either out of stock or unavailable will be lessened as a result of this research.

2. In this paper, a chatbot for selling services, products, and both digital and physical things is introduced. In this work, the authors presented their chatterbot idea for conversational commerce. The suggested design was made with the intention of enhancing user involvement in social media marketing and increasing the efficiency of social media marketing by applying the rapid order approach. The installed bot is only compatible with WooCommerce, therefore extending compatibility for other shopping carts might increase the use of chat-commerce bots.

3. In this paper, the author proposed to come up with a virtual shopping assistant having image recognition capabilities. For product search, current systems can only recognise text and speech. They do not, however, offer an image-based search feature. In order to provide customers with a strong and seamless search capacity to search for various types of things on e-commerce websites using simply photographs, this project will concentrate on image recognition.

4. In this paper, the authors have tried to evaluate performance of a classic chatbot ALICE, an entertaining chatbot Jabberwacky and a modern chatbot Rose. It makes use of AIML for specifying heuristic conversation rules. They have compared the knowledge bases, conversational capabilities, and capacity to handle unforeseen circumstances of chatbots and people. They’ve evaluated a few chatbots based on their performance and found that their knowledge base and conversational properties are comparable to humans. The conclusion was that chatbots perform equally well as humans but humans still have an edge over chatbots.

5. The key technologies in building social chatbots from core chat to visual awareness to skills was discussed in this paper. In this paper, it’s shown XiaoIce can dynamically recognize emotion and engage the user throughout long conversations with appropriate interpersonal responses. Although social chatbots like XiaoIce have made significant strides, the underlying workings of human-level intelligence, which typically manifest themselves in human-to-human interactions, are still not completely understood.

6. This paper comparatively investigates factors for customers’ satisfaction in voice commerce and ecommerce. This study's primary goal was to discover and comprehend the variables that influence consumer satisfaction with e-commerce and voice commerce technologies. Comparing the outcomes of these two e-commerce platforms was the second goal. They conducted a survey to test the research models

7. The proposed system successfully maps relationships and retrieves data. This study helps us to build the chat bot further to help the users to understand the details of products without logging in on the website, in this case the users who have queries and need information directly to be acquired successfully. This project will alleviate any pitfalls the user comes across during intensive search on the website and ameliorates their efficiency.

8. The suggested system architecture of a distributed chatbot system can be used in e-commerce to automate the human-machine communication using natural language queries. It uses WebSocket communication between user interface and the bot, analyzes the user's query and provides information of the queried orders and supplies. The system uses distributed services that provide both vertical and horizontal scale of the system.

1.6 SDLC approach used to develop project

Among the Waterfall Model, V-shaped Model, Iterative Model, Spiral Model, Big Bang Model and Agile Model we are using the first model - Waterfall Model, which is one of the most flexible, easily understandable in Software Development Life Cycle models.

The waterfall model is a breakdown of project activities into linear sequential phases, where each phase depends on the deliverables of the previous one and corresponds to a specialization of tasks. The approach is typical for certain areas of engineering design. In software development, it tends to be among the less iterative and flexible approaches, as progress flows in largely one direction ("downwards" like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing and deployment.

In our project we are following the Royce's original waterfall model, the following phases are followed in order:

1. System and software requirements: captured in a product requirements document
2. Analysis: resulting in models, schema, and business rules
3. Design: resulting in the software architecture
4. Coding: the development, proving, and integration of software
5. Testing: the systematic discovery and debugging of defects

6. Deployment: Hosting the application.

Thus, the waterfall model maintains that one should move to a phase only when it's preceding phase is reviewed and verified.

The waterfall model provides a structured approach; the model itself progresses linearly through discrete, easily understandable and explainable phases and thus is easy to understand; it also provides easily identifiable milestones in the development process of our project.

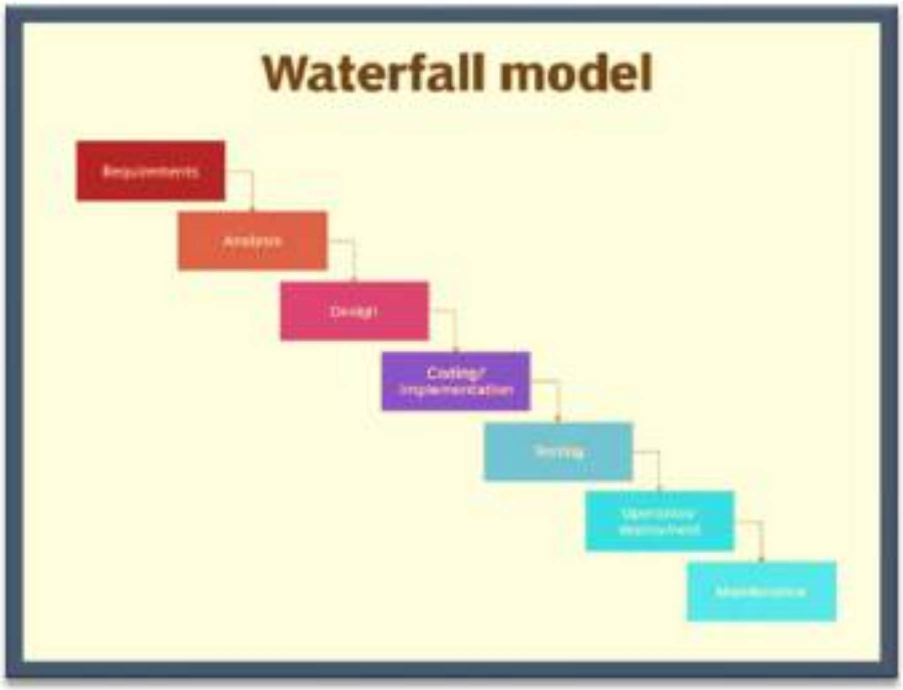


Fig 1.6.1: SDLC model used for implementation of the project

Chapter 2

Project Planning

2.1 Project Schedule

2.1.1 Work breakdown Structure

In project management and systems engineering, a work-breakdown structure is a deliverable-focused division of a project into more manageable parts. A essential project deliverable that divides the team's work into digestible chunks is a work breakdown structure. The work-breakdown structure is described by the Project Management Body of Knowledge as a hierarchical decomposition of the entire scope of work that the project team must carry out in order to achieve the project's goals and produce the necessary deliverables.

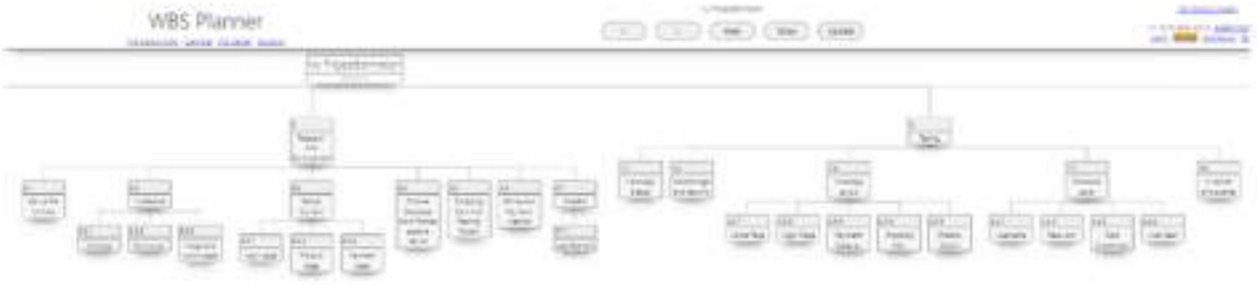


Fig 2.1.1.1: WBS

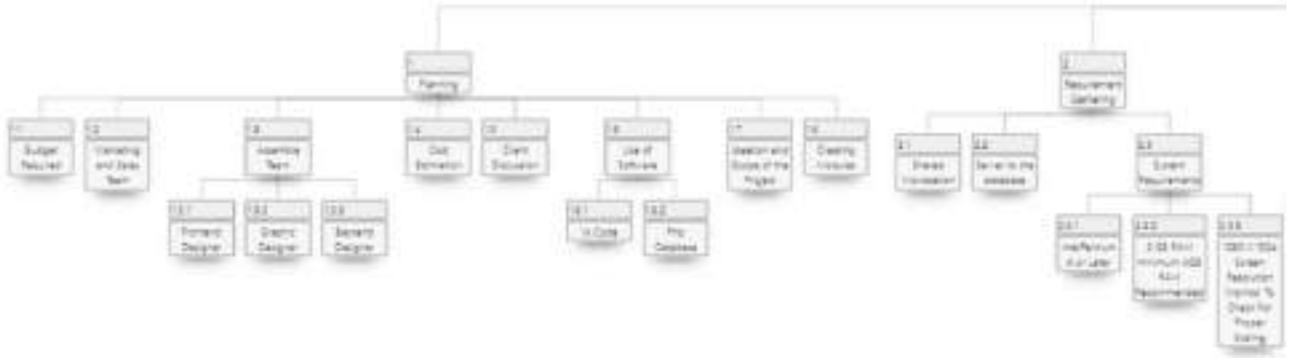


Fig 2.1.1.1: WBS

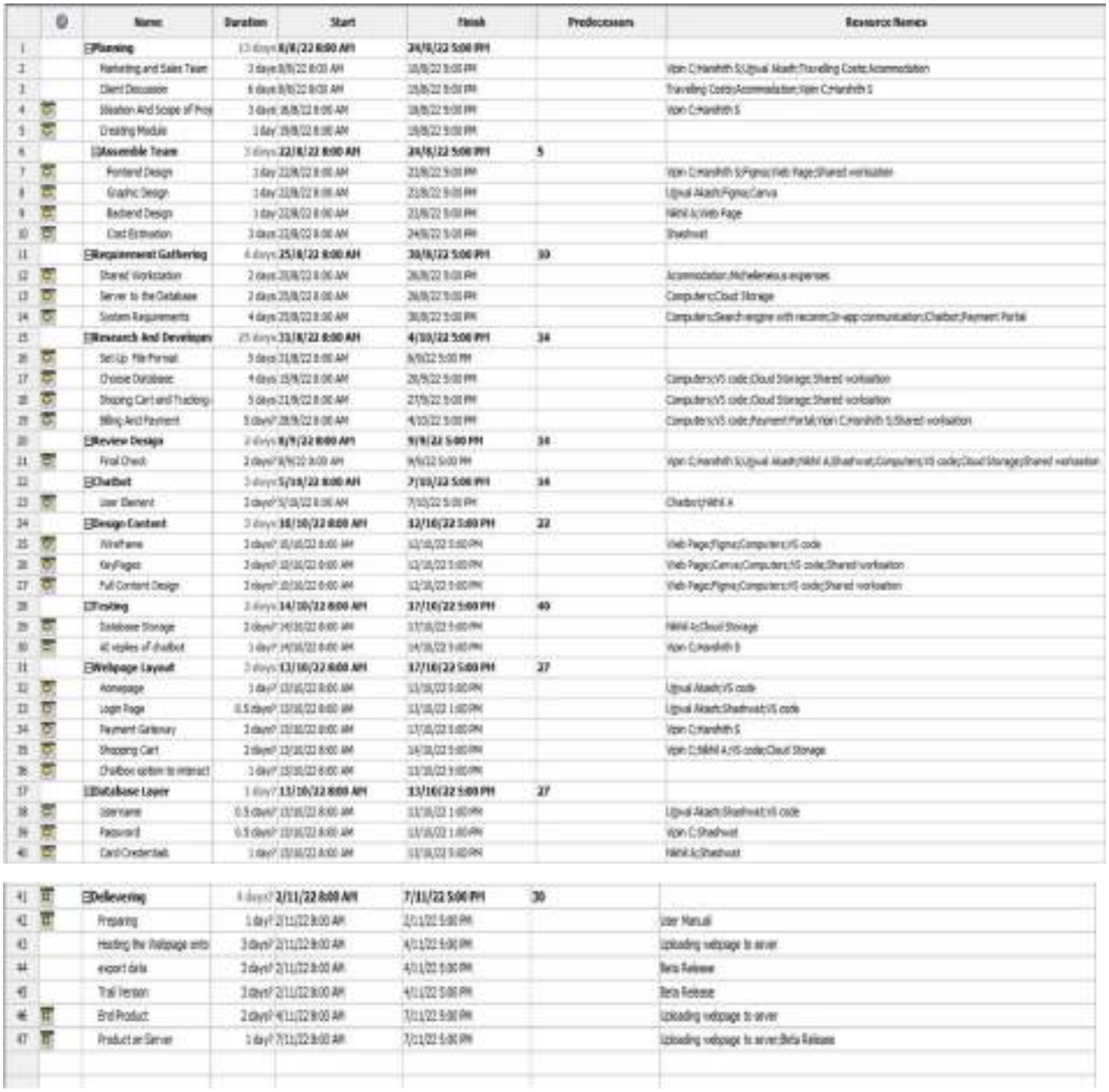
2.1.2 Gantt chart

2.1.2.1 Task scheduling

Scheduling project tasks is an important aspect of project planning. It entails choosing which tasks would be performed when. A software project manager wants to accomplish the following in order to schedule the project plan

List all the tasks that must be performed to finish the project. Divide up major tasks into smaller tasks. Determine how different activities are interdependent. Decide on the most likely size for the amount of time needed to finish the activities. Give resources to projects. Schedule the beginning and ending times of various events. Choose the critical route. The set of activities that determine the project's duration is an important factor.

The first step in arranging a software plan is to list every function necessary to finish the project. The supervisor can more effectively determine the project's vital role if they have a thorough understanding of the complexities of the project and the development process. The main functions are then divided into an appropriate set of tiny operations that would be delegated to different engineers. The logical model of the work After the project manager has carefully broken down the purpose and created the work breakdown structure, formalism aids the manager in identifying the dependencies between the activities. The sequence in which the various occurrences would take place is determined by dependencies among the various activities. If the outcomes of one action A are required by another, then activity A must be scheduled after activity B. The function dependencies, in general, represent a partial ordering of functions, i.e., each service may come before a subset of others, but some functions might not be preceded by any other functions at all (called concurrent function). The structure of an activity network defines how the activities are interdependent.



		Name	Duration	Start	Finish	Predecessors	Resource Names
1		Planning	13 days	8/8/22 8:00 AM	24/8/22 5:00 PM		
2		Marketing and Sales Team	3 days	8/8/22 8:00 AM	10/8/22 5:00 PM		Vipin C;Harshith S;Ujjwal Akash;Traveling Costs;Accommodation
3		Client Discussion	6 days	8/8/22 8:00 AM	15/8/22 5:00 PM		Traveling Costs;Accommodation;Vipin C;Harshith S
4		Ideation And Scope of Proj	3 days	16/8/22 8:00 AM	18/8/22 5:00 PM		Vipin C;Harshith S
5		Creating Module	1 day	19/8/22 8:00 AM	19/8/22 5:00 PM		
6		Assemble Team	3 days	22/8/22 8:00 AM	24/8/22 5:00 PM	5	
7		Frontend Design	1 day	22/8/22 8:00 AM	22/8/22 5:00 PM		Vipin C;Harshith S;Figma;Web Page;Shared workstation
8		Graphic Design	1 day	22/8/22 8:00 AM	22/8/22 5:00 PM		Ujjwal Akash;Figma;Canva
9		Backend Design	1 day	22/8/22 8:00 AM	22/8/22 5:00 PM		Nikhil A;Web Page
10		Cost Estimation	3 days	22/8/22 8:00 AM	24/8/22 5:00 PM		Shashwat
11		Requirement Gathering	4 days	25/8/22 8:00 AM	30/8/22 5:00 PM	10	
12		Shared Workstation	2 days	25/8/22 8:00 AM	26/8/22 5:00 PM		Acommodation;Michelleneous expenses
13		Server to the Database	2 days	25/8/22 8:00 AM	26/8/22 5:00 PM		Computers;Cloud Storage
14		System Requirements	4 days	25/8/22 8:00 AM	30/8/22 5:00 PM		Computers;Search engine with recomm;In-app communication;Chatbot;Payment Portal
15		Research And Developm	25 days	31/8/22 8:00 AM	4/10/22 5:00 PM	14	
16		Set Up File Format	5 days	31/8/22 8:00 AM	6/9/22 5:00 PM		
17		Choose Database	4 days	15/9/22 8:00 AM	20/9/22 5:00 PM		Computers;VS code;Cloud Storage;Shared workstation
18		Shopping Cart and Tracking	5 days	21/9/22 8:00 AM	27/9/22 5:00 PM		Computers;VS code;Cloud Storage;Shared workstation
19		Billing And Payment	5 days?	28/9/22 8:00 AM	4/10/22 5:00 PM		Computers;VS code;Payment Portal;Vipin C;Harshith S;Shared workstation
20		Review Design	2 days	8/9/22 8:00 AM	9/9/22 5:00 PM	14	
21		Final Check	2 days?	8/9/22 8:00 AM	9/9/22 5:00 PM		Vipin C;Harshith S;Ujjwal Akash;Nikhil A;Shashwat;Computers;VS code;Cloud Storage;Shared workstation
22		Chatbot	3 days	5/10/22 8:00 AM	7/10/22 5:00 PM	14	
23		User Element	3 days?	5/10/22 8:00 AM	7/10/22 5:00 PM		Chatbot;Nikhil A
24		Design Content	3 days	10/10/22 8:00 AM	12/10/22 5:00 PM	22	
25		Wireframe	3 days?	10/10/22 8:00 AM	12/10/22 5:00 PM		Web Page;Figma;Computers;VS code
26		KeyPages	3 days?	10/10/22 8:00 AM	12/10/22 5:00 PM		Web Page;Canva;Computers;VS code;Shared workstation
27		Full Content Design	3 days?	10/10/22 8:00 AM	12/10/22 5:00 PM		Web Page;Figma;Computers;VS code;Shared workstation
28		Testing	2 days	14/10/22 8:00 AM	17/10/22 5:00 PM	40	
29		Database Storage	2 days?	14/10/22 8:00 AM	17/10/22 5:00 PM		Nikhil A;Cloud Storage
30		AI replies of chatbot	1 day?	14/10/22 8:00 AM	14/10/22 5:00 PM		Vipin C;Harshith S
31		Webpage Layout	3 days	13/10/22 8:00 AM	17/10/22 5:00 PM	27	
32		Homepage	1 day?	13/10/22 8:00 AM	13/10/22 5:00 PM		Ujjwal Akash;VS code
33		Login Page	0.5 days?	13/10/22 8:00 AM	13/10/22 1:00 PM		Ujjwal Akash;Shashwat;VS code
34		Payment Gateway	3 days?	13/10/22 8:00 AM	17/10/22 5:00 PM		Vipin C;Harshith S
35		Shopping Cart	2 days?	13/10/22 8:00 AM	14/10/22 5:00 PM		Vipin C;Nikhil A;VS code;Cloud Storage
36		Chatbox option to interact	1 day?	13/10/22 8:00 AM	13/10/22 5:00 PM		
37		Database Layer	1 day?	13/10/22 8:00 AM	13/10/22 5:00 PM	27	
38		Username	0.5 days?	13/10/22 8:00 AM	13/10/22 1:00 PM		Ujjwal Akash;Shashwat;VS code
39		Password	0.5 days?	13/10/22 8:00 AM	13/10/22 1:00 PM		Vipin C;Shashwat
40		Card Credentials	1 day?	13/10/22 8:00 AM	13/10/22 5:00 PM		Nikhil A;Shashwat
41		Delievering	4 days?	2/11/22 8:00 AM	7/11/22 5:00 PM	30	
42		Preparing	1 day?	2/11/22 8:00 AM	2/11/22 5:00 PM		User Manual
43		Hosting the Webpage onto	3 days?	2/11/22 8:00 AM	4/11/22 5:00 PM		Uploading webpage to server
44		export data	3 days?	2/11/22 8:00 AM	4/11/22 5:00 PM		Beta Release
45		Trail Version	3 days?	2/11/22 8:00 AM	4/11/22 5:00 PM		Beta Release
46		End Product	2 days?	4/11/22 8:00 AM	7/11/22 5:00 PM		Uploading webpage to server
47		Product on Server	1 day?	7/11/22 8:00 AM	7/11/22 5:00 PM		Uploading webpage to server;Beta Release

Fig 2.1.2.1 Task Scheduling

2.1.2.2 Gantt Chart for the Scheduled task

One of the most common and effective methods of displaying activities displayed against time is a Gantt chart, which is frequently used in project management. A list of the activities is located on the chart's left side, and a suitable time scale is located along the top. A bar is used to symbolize each activity, and the position and length of the bar correspond to the activity's beginning, middle, and finish dates.

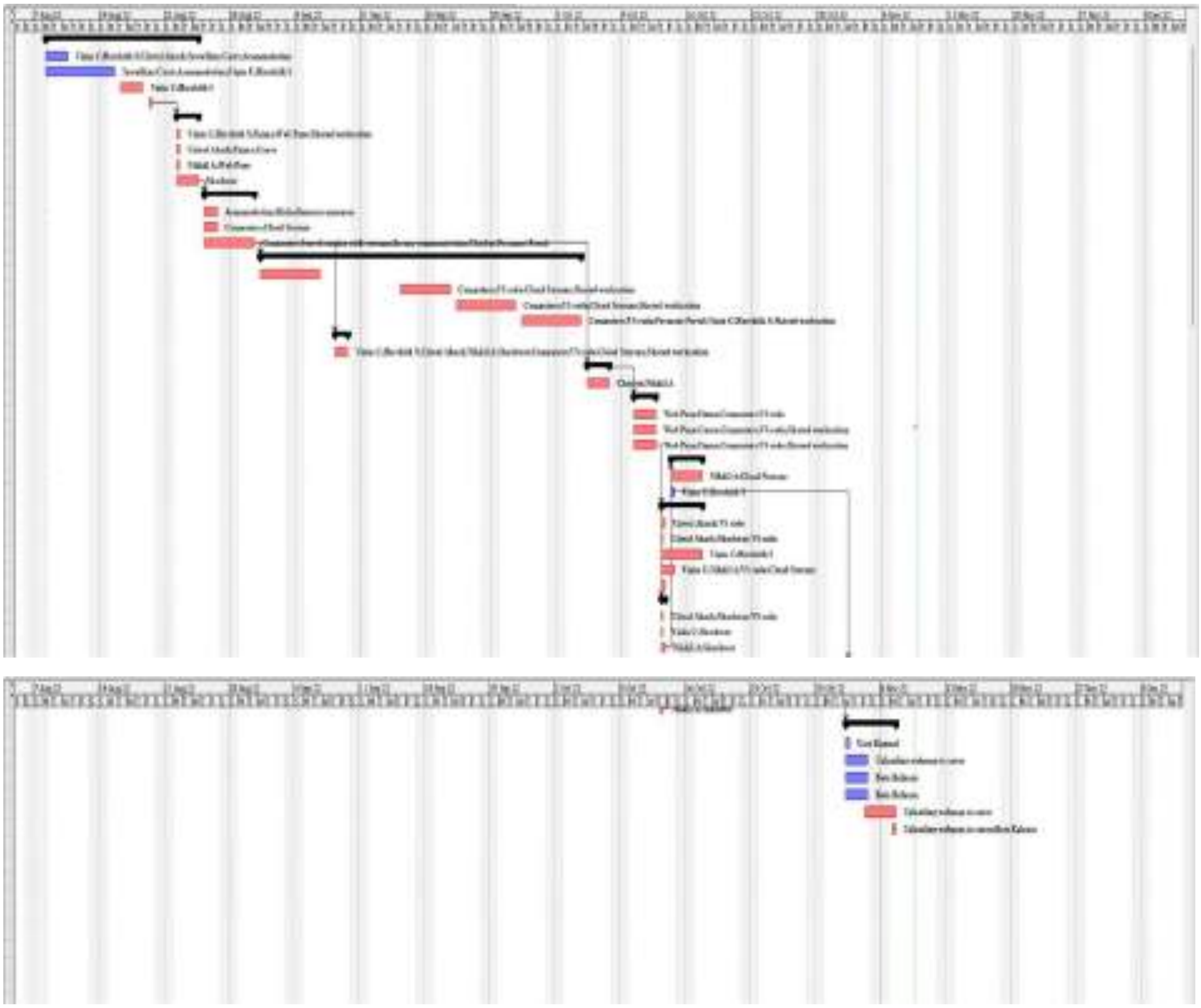


Fig 2.1.2.2 Gantt Chart

2.2 Effort and Resource Estimation

2.2.1 Resource Allocation

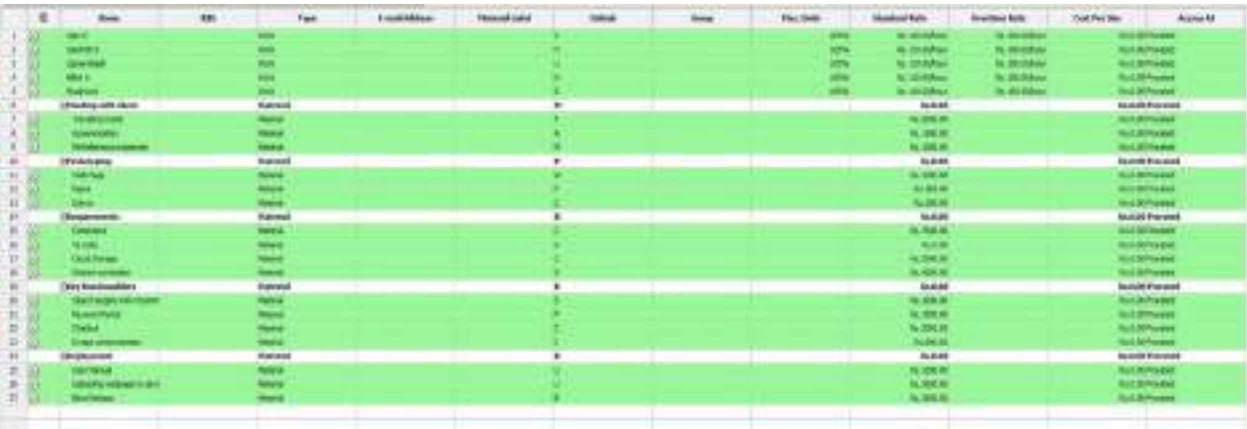
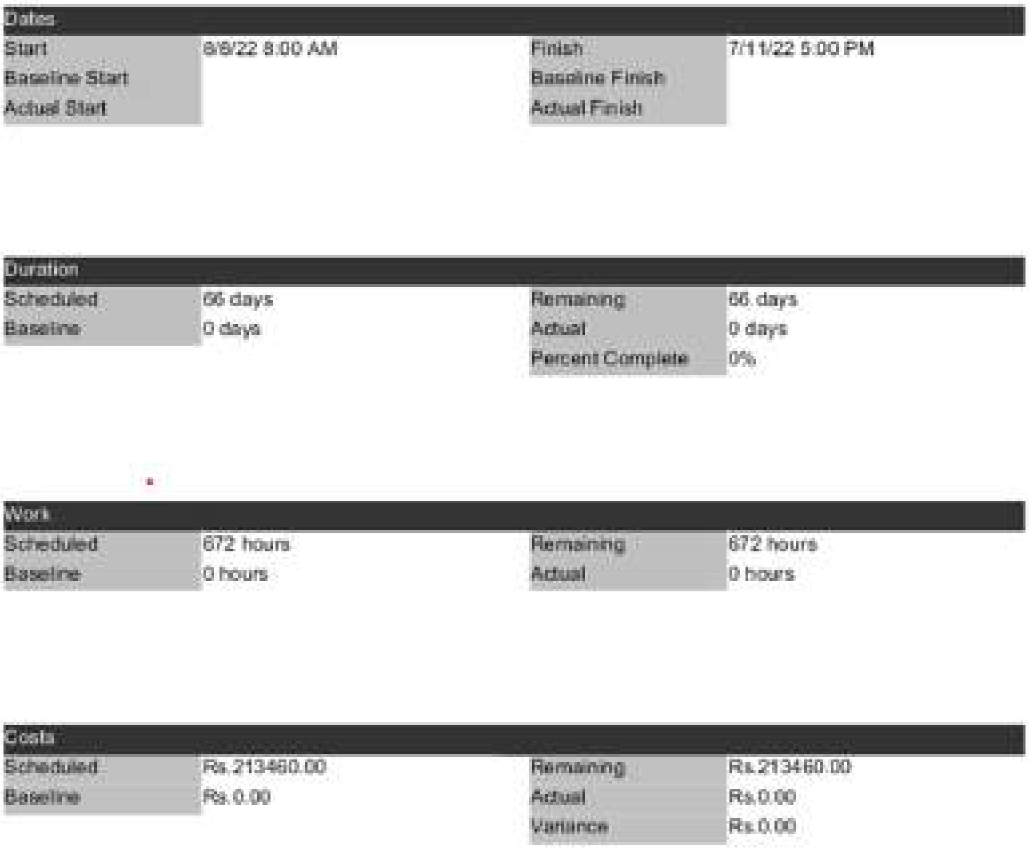


Fig 2.2.1: Resource Allocation

2.2.2 Effort and Cost Estimation

2.2.2.1 System Calculation



Dates			
Start	8/8/22 8:00 AM	Finish	7/11/22 5:00 PM
Baseline Start		Baseline Finish	
Actual Start		Actual Finish	

Duration			
Scheduled	66 days	Remaining	66 days
Baseline	0 days	Actual	0 days
		Percent Complete	0%

Work			
Scheduled	672 hours	Remaining	672 hours
Baseline	0 hours	Actual	0 hours

Costs			
Scheduled	Rs.213460.00	Remaining	Rs.213460.00
Baseline	Rs.0.00	Actual	Rs.0.00
		Variance	Rs.0.00

Fig 2.2.2.1: Cost and work Estimate

2.2.2.2 Manual Calculation

- **Effort= 3 * (0.8)^1.12= 2.336 PM**
- **T dev= 2.5* (2.336)^0.35 =3.364 Months ~ 100 days**

3.1 SRS

3.1.1 Introduction

3.1.1.1 Purpose

The purpose of this document is to present a detailed description of the e-commerce website, which is embedded with AI chatbot, where it holds the reason for the development and explains the features in the system. It also provides us with an insight regarding how it would look like, how it would function and what the inputs and outputs would be generated when tried/tested. This document is for the developers and users of the software system.

3.1.1.2 Document Conventions

The conventions used are:

E-R diagram	Entity Relationship Diagram
E-R model	Entity Relationship
Sign-in/sign-out model	To enter and exit the portal for placing orders
DFD	Data flow diagrams
UML	Unified Modeling Language
AI chatbot	Artificial Intelligence chatbot

Table 3.1.1.1: Document conventions

3.1.1.3 Intended Audience and Reading Suggestions

This document is for the project managers, developers, programmers, testers, documentation writers, user and backend support team. This document consists of the various steps and procedures that could be insisted on the webpage. As we move further, we discuss the project scope, functionality, non-functionality and interfaces of the system.

The website which is being developed is targeted to all the audience who are interested in buying essentials or any general items which are needed online from the comfort of their home.

This is one of the important aspects for people because nowadays after the pandemic has come to a stall people have got habituated to online shopping more in the pandemic than compared to the pre-pandemic which makes them toshop from there phone or any electronic gadget than going to physical store andcheck for the products they need, the chatbot will help them for a more smotheruser experience after the purchase is done.

This SRS document helps convey the applications functionality for the user and the development team. Defining everything from the software interactivity when embedded in hardware, not only will it be specification from where software is developed, it also acts as a guideline as to how to go about developing it. This helps us make vital decisions on our product's lifecycle, and in the unit testing and verification.

This documents also functions as a contract between the software's owner and the developers, allowing the owner to know exactly what the developers aim to do and how they plan to create the software.

3.1.1.4 Product Scope

Electronic commerce, also known as e-commerce, is a type of industry werebuying and selling of a product is conducted over electronic systems such as the internet. E-commerce has boomed over the years and is one of the fastest-growing domains in the online world. Though it took some time for this to be acceptedby the end-users, today we are at a point where the majority of people love toshop online.

In this project we are developing a business to consumer (B2C) application inwhich the consumer can select the items from the catalogue and add to cart to buy the products.
The system generates the billing and consumers can pay through payment gateway. If the consumer needsany assistance he can interact with the chat-bot.

The purpose of the e-commerce website with Ai chatbot embedded into itis to make the user interaction more friendly and hassle-free experience once theorder is placed and when the order is delivered and requires any kind of assistance. With AI chatbot embedded into the system the solution for the queryraised would be obtained instantaneously with no waiting period for technical expert to be connected to the chat for helping with the problems which would bemore time consuming and a tedious task. Thus, we are developing this website with a chatbot for better user experience.

Not only does our project show significant scope in the field of online shopping,but also in everyday, real-life scenarios where people of any age and background can order online with their preference of items for their needs to be full filled with better user experience through website.

3.1.2 Overall Description

3.1.2.1 Product Perspective

This product is to make changes to the already existing systems in the market by enhancing the ordering experience and providing seamless experience of finding the right product and delivering the right product with better customer support after sales. As we are developing a web application users can search for multiple options and an efficient one which fits their needs can be found out and ordered upon. There are already existing models where theuser feedback is poor due to their poor customer support which is provided afterthe order is placed and delivered, which is being improved with the help of AI. It is also a cost-efficient way as you find a one-stop solution as the number of people deployed to attain a customer decreases.

The web application being developed will help the customers to have better
belief in product which is being purchased with no fear in their mind regarding the product being delivered to them is in correct shape or not, as the customer will have a first in segment support with AI and human guidance enabled with afaster solution to the problem which might be encountered during their purchaseperiod.

3.1.2.2 Product Functions

The online shopping-cart application would have the following basic functions:

1. Display all the categories available for shopping on the system’s main page.
2. Display all the items linked to each category listed on the main page.
3. Maintain records for many customers
   - A customer can be either a member or non-member.
   - A customer has a username (unique across all users), password (no restrictions), email address (no restrictions), and postal address (unverified.)
   - Anyone may sign up for a customer account.
   - Allow any customer to become a member.
   - Show a listing of available products
   - Allow customers and administrator to log in and out of the system.
   - Anyone can add one or more products to the shopping cart.
   - The shopping cart needs to allow multiple products of the same type.

- Allow users to remove items

4. Maintain data associated with the inventory (a collection of products)
    - The inventory also keeps track of the stock/quantity of each product
5. Checkout
    - Checkout is only available to logged-in customers. A user that isnot logged in as a customer is given a chance to log in.
    - Collect a 16-digit credit card number from the customer
    - Log/record the transaction

.

6. Customer Support/Chatbot – for any assistance the customer can interact inthe Chatbot.

User Side:

- Login
- Signup
- Search
- Order
- Add to cart
- Checkout
- Payment method
- Order history
- Customer support
    - Chat with human interface
    - Chat with AI

3.1.2.3 User Classes and Characteristic

The users of the online shopping-cart application, based on their roles, are customers (users) and the administrator (owner). These users are identified based on their experience and technical expertise.

1. Users: The users of this online shopping-cart application are all customers who would shop to test the application. These users are anyone with shoppingexperience and the know-how to browse through a shopping-cart application. They must have basic understandings about computers and the internet. The users should be

able to perform the following functions using this system:

- View, browse, and select a category on the home page.
- View, add, and update items in the cart.
- Delete items from the cart.
- Check out the items from the application or continue shopping.
- Sign-on/login using a username and password.
- Place the order by completing the order form
- customer can interact in the Chatbot for any assistance

3.1.2.4 Operating Environment

The operating environment for the system being developed is:

- Operating system: Windows/ MAC OS
- Client/ server system
- Database: MySQL
- Browser: Google Chrome 44+ / Safari 7+ / Mozilla Firefox 40+
- Processor: INTEL CORE PROCESSOR or AMD Ryzen
- Memory: 2GB Ram or more
- Hard disk space: A minimum of 5 Gb for Database Connectivity

3.1.2.5 Design and Implementation Constraints

1. **Hardware Constraints:** The minimum hardware requirement for the systemis 128 MB of Ram and a 32-MB hard-disc drive.

2. **Accessibility Constraints:** Initially, the software should be available as adesktop application for a small set of users to test.

3.**Interface Constraints:** Since this is a web-based application it should workon major browsers like internet Explorer, Mozilla Firefox, Google Chrome, Opera etc.

4. **Safety and Security Constraint:** Since, application is intended for theauthenticated user only, so anonymous person should not be able to access and operate over the user data.

5. **Product Constraints:** The software needs to be designed in a user-friendlymanner to ensure its completeness and effectiveness.

6. **Dataset**: getting the right dataset into the database so that the answersprovided are relevant.

7. **Others:** The application should be built using JavaScript/php inscribed inHTML, and it should, initially, be

accessible through the NetBeans IDE andlater published on a server.

3.1.2.6 User Documentation

A user manual will be provided which will describe the main features and functionality of the software. It will guide the first-time users how to go about in the website like sign in, create account, ordering, Payments etc. There will also be a small 60 second video demo for all the basic tasks which can be done through the website which will guide them even better as it will show them the exact tasks that needs to be done in the same website. We will also have chat assistance which helps the customer to chat with the respected company so that they can a find a solution to the problem which they might face. They will also be provided with a user manual which has a set of instructions.

3.1.2.7 Assumptions and Dependencies

The assumptions and dependencies are as follows:

1. Users and the administrator are accustomed to the paper-based system andwould require training to use the online shopping-cart application.
2. The system is dependent on the availability of an Apache Tomcat Server torun.
3. We assume that system users adhere to the system’s minimum software and hardware requirements.
4. This system will use third-party software, and it is assumed that system usersare familiar with the software

3.1.3 External Interface Requirements

3.1.3.1 User Interface

- Front-end software: HTML 5, CSS and JavaScript
- Back-end software: SQL using PHP

The user interface will be simple and consistent through the webpages, we will use simple terms in the webpage which can be understood by users to whom it’stargeted. The system will have a simple interface and common terminologies sothat no additional training is required to guide.

3.1.3.2 Hardware Interface

There are no special hardware requirements for this project, a normal personal computer can be used to work on this webpage.

- Monitor for displaying the webpage.
- A system with an operating system to run a web browser.
- Keyboard and mouse for navigating and searching.
- A browser which supports CGI, HTML and JavaScript

3.1.3.3 Software Interface

The software used to create the E-commerce website with chatbotintegrated:

Operating system: Windows is being used as it is more user friendly and bettersoftware availability.

Database: MySQL is used to save item records, customer records and other stuff, It is also easy to retrieve data and publish as it just requires a line of codePHP: we use PHP to connect the data with the database and use it to access it and refer it

Programming language: HTML 5 to code the layout of the webpage, CSS to style the website that is being developed and JavaScript to collect data from theuser and to store and access it. Python language is used for implementing the chatbot to the website.

3.1.3.4 Communications Interface

The data is mainly transferred between the website and the respectiveapis. JavaScript functions will be used to make the necessary calls. The exact formats and protocols for incoming and outgoing messages should be abstracted by the APIs. We will be using HTTPs for secure movement through the website during the process of payments. We will also fetch the data from the database for the user interactionwith the chatbot

3.1.4 System Features

3.1.4.1 System feature 1[System Functionality]

- The user needs to search for the items required in the search bar
- The user needs to select from the options provided for the item thatwas searched by them.
- Based on the customer choices for that item, the most sold item would be at the top of the webpage followed by the least sold/ preferred by the customers who visited the site.
- Using the filters option, the user can further filter or shortlist fromthe given results.
- After selecting the desired item, we can add it to the cart.
- Once added to the cart, we click on proceed to check out.
- Where we enter the payment details and the address to bedelivered.

- Where the booking gets confirmed and a pass gets generated into our booking column which acts as a confirmation tool for order placed.

3.1.4.1.1 Description and Priority

The main feature of the website is to search for products which they need by staying at their comfort from a click of a button on their electronic device, it also uses AI search engine to display a better search result and classify better product recommendations to the users based on previous customer transactions which would have happened on the website.

The basic website is built using HTML for the webpage layout and CSS is used to style the webpage and JavaScript is used to collect and do task in a webpage which leads us to the designing of frontend by this tool where we can have a user interface upon which the user can work upon after which we will host theserver onto the web.

We create a SQL query which would be linked to JavaScript for taking inputlike
personal details and payment mode which gets stored in the database for future verification and validation of users as they log in and log out and try to purchasethrough the website.

We will create a secure hypertext transfer protocol for the wallet option where the user can link there bank account or add money to their wallet just like a physical wallet which can be redeemed at the time of ordering through the portal or pay later with the linked wallet by selecting the preferred date and time.
We would create the frontend first which is used for the user interface, it helps us to get a raw idea how the website would look like and next the database which has SQL queries gets linked to it for authentication and validating the user after which it gets hosted to the web.

3.1.4.1.2 Stimulus/Response Sequences

The user creates an account and starts selecting the desired items that they intend to purchase. The recommendations are generated using the results from which the user can be filtered further based on their requirements or select fromany of the recommendations provided in the search. The user can select the product and proceed to checkout

3.1.4.1.3 Functional Requirements

Functional requirements often describe specific actions taken or outputs produced in response to the inputs provided. Sometimes if the user input is wrong an exception is thrown with the error condition printing onto the screen.

We also perform certain specific tasks for the security of the system. These are requirements that the user specifies.

REQ-1: checking for inputs whether they're valid or not

REQ-2: Exact sequence of operation i.e., search, select and order

3.1.4.2 System feature 2[Chatbot for customer assistance with AI enabled]

3.1.4.2.1 Description and Priority

The users can get their queries resolved without any human involvement with the help of an AI powered chatbot.

3.1.4.2.2 Stimulus/Response Sequences

A set of frequently asked questions will be displayed to the user and whenthe user selects the query, the answers to it are fetched from the database and displayed immediately. The AI refines the answers each time a query is being asked. So, the perfection in answers increases as the number of queries on a particular topic increase.

3.1.4.2.3 Functional Requirements

A chatbot is software that simulates human conversations. It enables communication between a human and a machine, which can take the form of messages. A chatbot is designed to work without the assistance of a human operator. AI chatbot responds to questions posed to it in natural language as if were a real person . It responds using a combination of pre-programmed scripts and machine learning algorithms. The chatbot will answer using the knowledge database that is currently available to it if the conversation introduces a concept which isn’t programmed to understand, it will pass it to a human operator. It will learn from that interaction as well as future interactions in either case.

3.1.5 Other Nonfunctional Requirements

3.1.5.1 Performance Requirements

The system must be interactive, and the delay involved must be negligible. So in every action-response of the system, there are no delays in producing the output. Any operation performed to the website should produce an output or a failed output with what error occurred within 2 seconds of the lag. Also, when connecting to the server the delay is based on the distance between the two systems, so the less distance the better the response is and the better configuredthe better response on the website which leads to less latency delay. The systemdata shall load as quickly as possible even with background running.

3.1.5.2 Safety Requirements

The website collects data and other personal information of the users so itis necessary for the admin of the webpage to secure the data, we can encrypt thedata as when it is being fed into the server so that the data breach doesn't take place and as we have wallet option the funds can easily be transferred so we canhave a authenticator in the tool so that whenever we are paying we can use a verification tool like a pin, fingerprint or facial ID to transfer which makes it more secure and safe environment is created.

3.1.5.3 Security Requirements

User Identity authentication requirement:

- The user can create their own account on the website to save theirdata
- This helps them to save personal information and bank details in their own account and cannot be accessed by anyone unless they knowyour username and password

Other than customer name and product order, user reviews. The website being developed doesn't store any data or collect it without any user knowledge.

We have a payment portal which is secured for which the website must obtain security and privacy certifications which must be satisfied so that the transaction can happen on the website without any hassle and adhere to the locallaws which are enforced

3.1.5.4 Software Quality Attributes

Adaptability:

- Any user with a pc or smart device can use our website forordering.
- Our website can run on all operating systems.
- Can adapt easily as the interface is going to similar to that of anyonline shopping portals.

Availability:

- It’s available in all the Operating system from Pc’s to mobilephones
- The order details of items will be saved in the orders column forfuture
- References for easy verification.

Reliability:

- Our website can be used by anyone who wants to purchase itemswhich are needed for user with them going out to multiple shops andverifying the product

3.1.5.5 Business Rules

We will take the APIs of providers who all the products from across platform and will have the dataset present with them, now we will integrate all the dataset into one roof which will provide the users a better experience as they will have a wider range of options to purchase and become a one stop solution for all needs.

3.1.6 Other Requirements

Our project utilities the incremental model because we keep updating thesystem frequently with new features and updates for existing software.

and become a one stop solution for all needs.

3.1.6 Other Requirements

Our project utilities the incremental model because we keep updating thesystem frequently with new features and updates for existing software.

3.2 Data Modelling

3.2.1 Data flow Diagram

A data-flow diagram is a visual representation of how data moves through a system or a process. The DFD additionally gives details about each entity's inputs and outputs as well as the process itself. A data-flow diagram lacks loops, decision rules, and control flows. Using a flowchart, certain operations based on the data can be depicted.

3.2.1.1 Level-0 Data flow Diagram

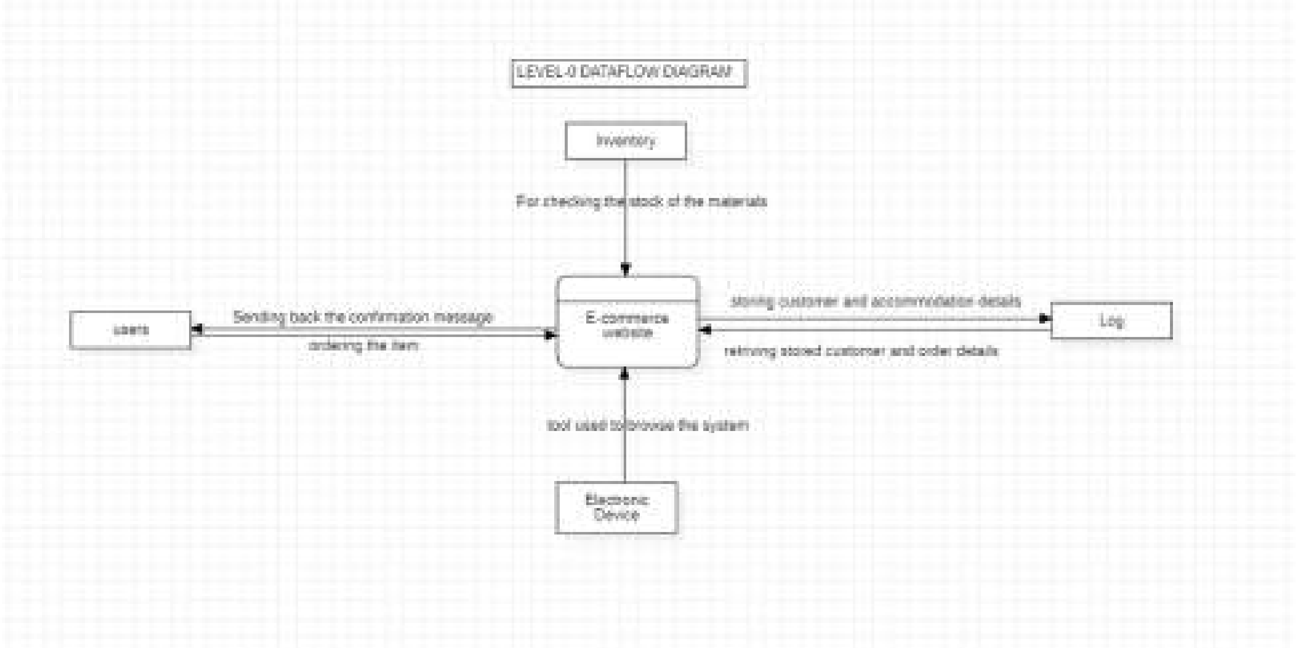

LEVEL-0 DATAFLOW DIAGRAM
Inventory
For checking the stock of the materials
users
Sending back the confirmation message
ordering the item
E-commerce website
storing customer and accommodation details
Log
retriving stored customer and order details
tool used to browse the system
Electronic Device

Fig 3.2.1.1 level-0 DFD

3.2.1.2 Level-1 Data flow Diagram

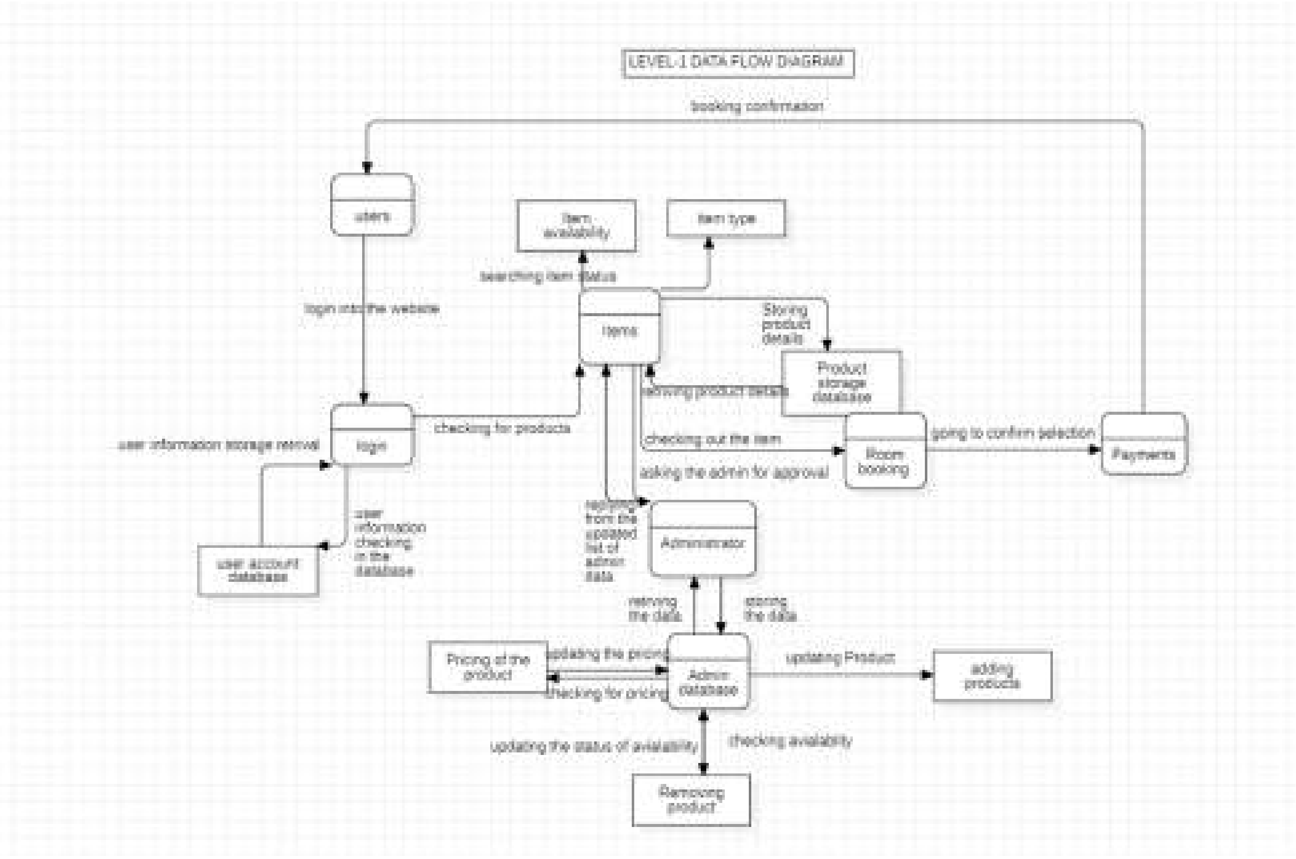


Fig 3.2.1.2 Level-1 DFD

3.2.1.3 Level-2 Data flow Diagram

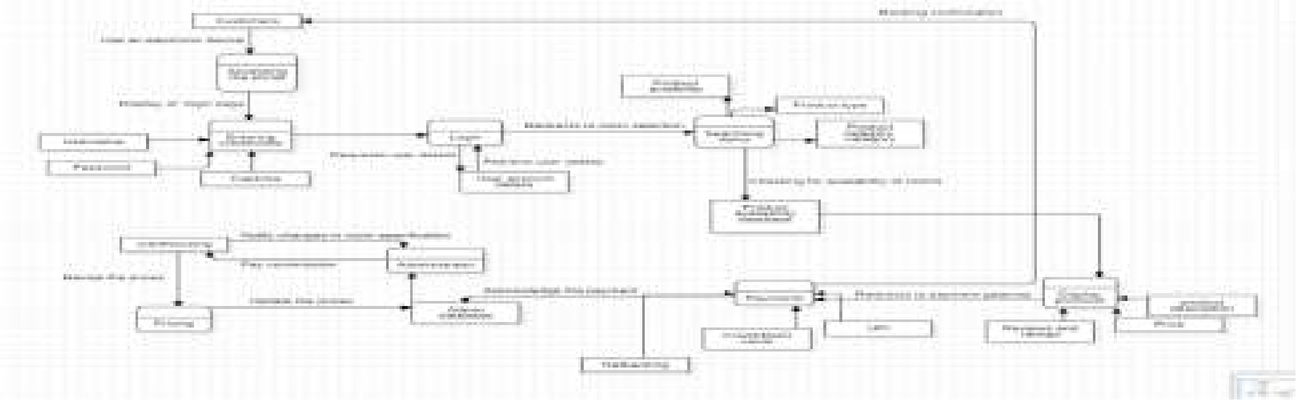


Fig 3.2.1.3 Level-2 DFD

3.2.1.2 Level-1 Data flow Diagram

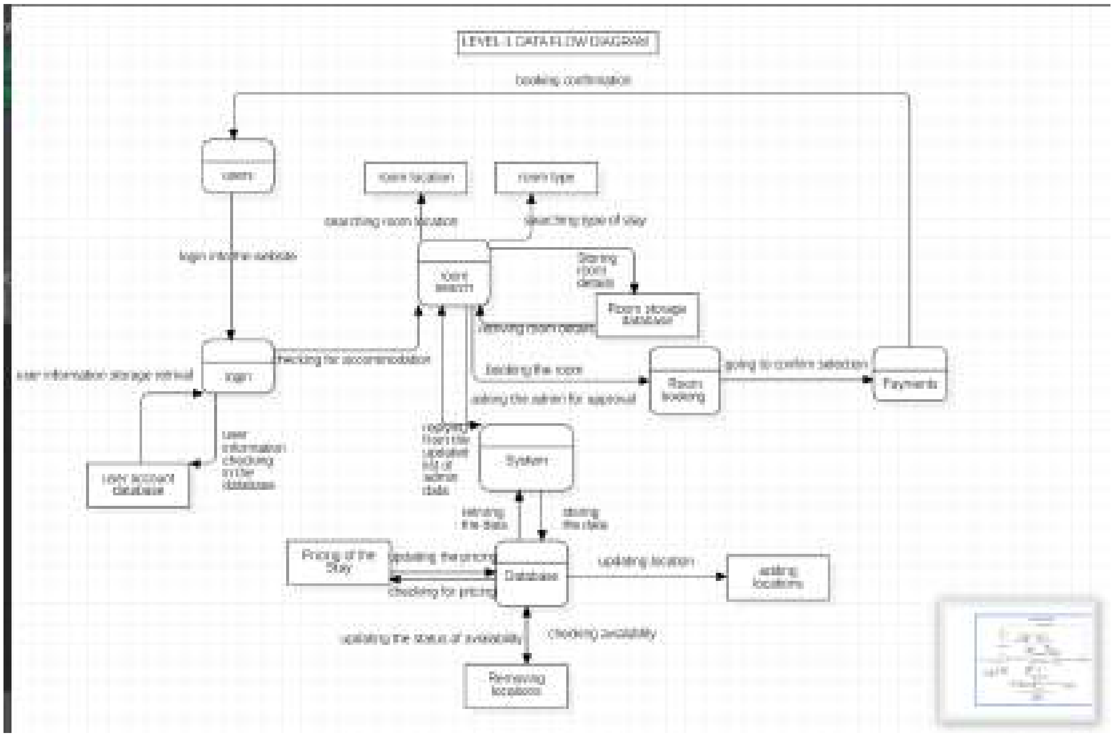


Fig 3.2.1.2 Level-1 DFD

3.2.1.3 Level-2 Data flow Diagram

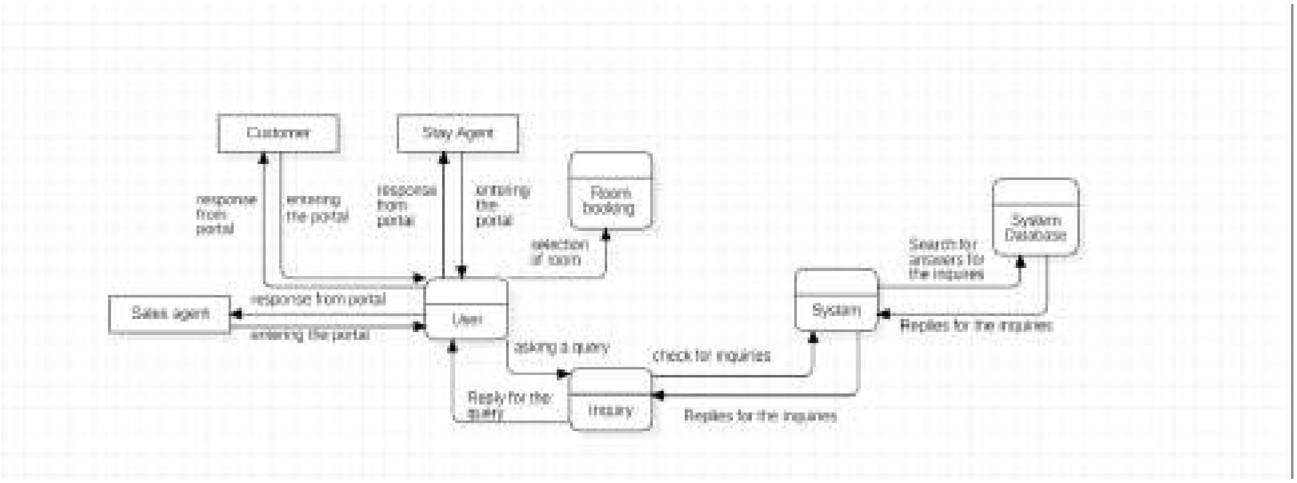


Fig 3.2.1.3 Level-2 DFD

Chapter 4

Designs

4.1 UML Designs

4.1.1 Structural Diagram

4.1.1.1 Class Diagram

The purpose of the diagram is to introduce some common terms, "dictionary" for online shopping - Customer, Web User, Account, Shopping Cart, Product, Order, Payment, etc. and relationships between. Each customer has unique id and is linked to exactly one account. Account owns shopping cart and orders. Orders are sorted and unique. Each order is linked to none to several payments. Customer could register as a web user to be able to buy items online. Web user has login name which also serves as unique id.

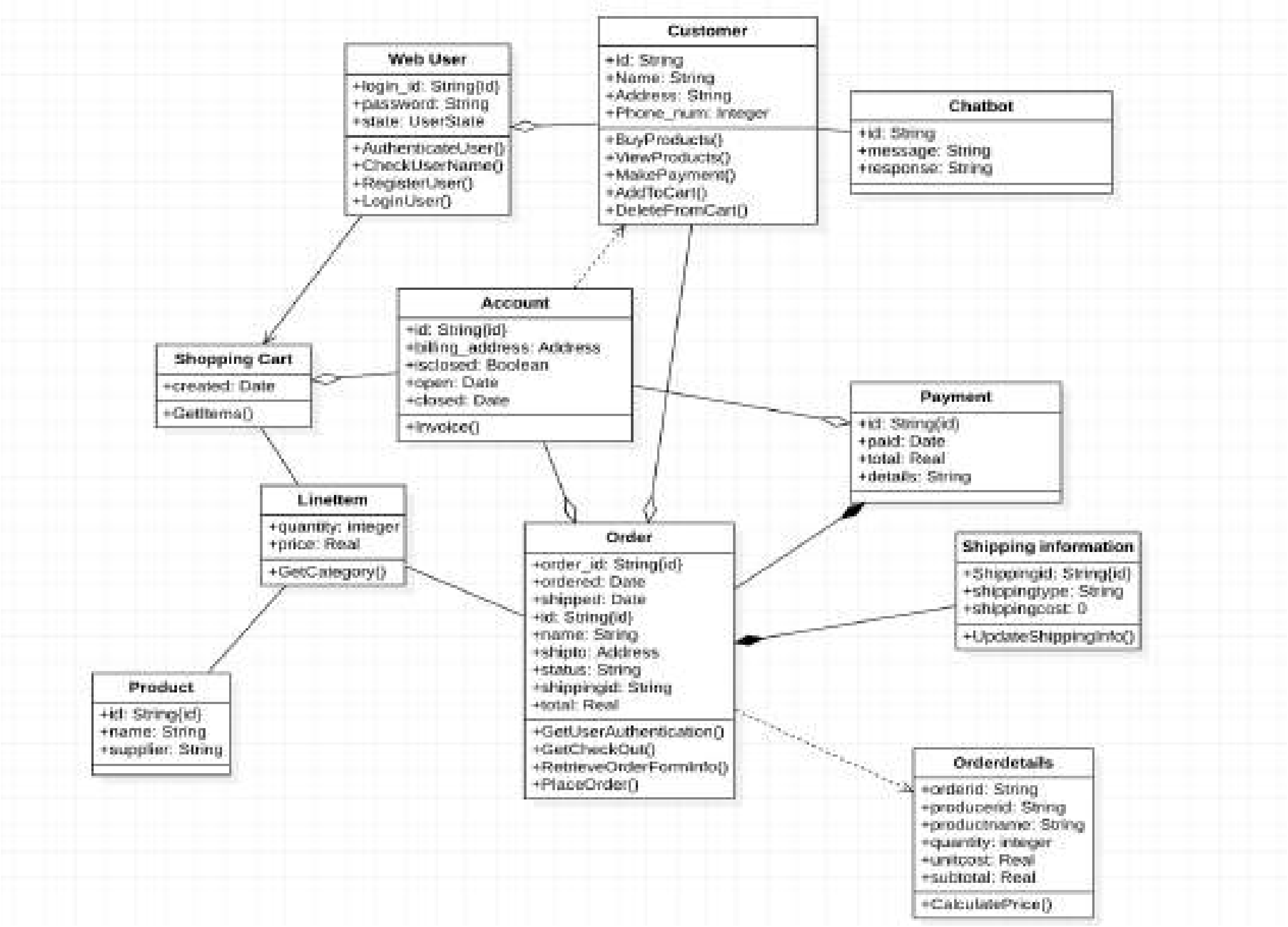


Fig 4.1.1.1 Class Diagram

4.1.1.2 Object Diagram

In the Unified Modeling Language (UML), an object diagram focuses on some particular set of objectsand attributes, and the links between these instances. Object diagrams and class diagrams are closely related and use almost identical notation. Both diagrams are meant to visualize static structure of a system. While class diagrams show classes, object diagrams display instances of classes (objects).

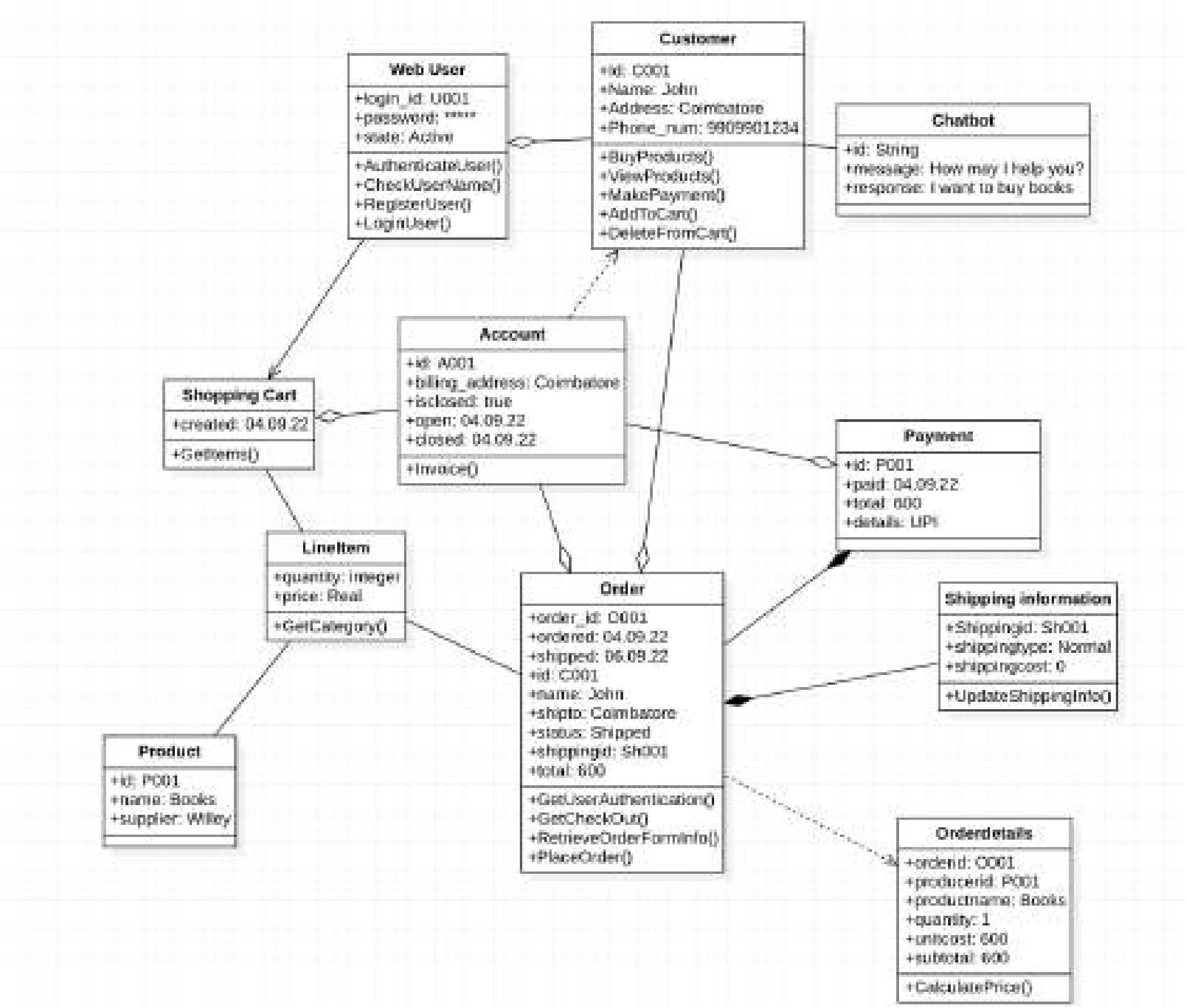


Fig 4.1.1.2: Object Diagram

4.1.1.3 Component Diagram

Search Engine component uses Inventory interface to allow customers to search or browse items. Shopping Cart component uses Orders component during checkout process. Authentication component allows customer to login and binds the customer to Account.

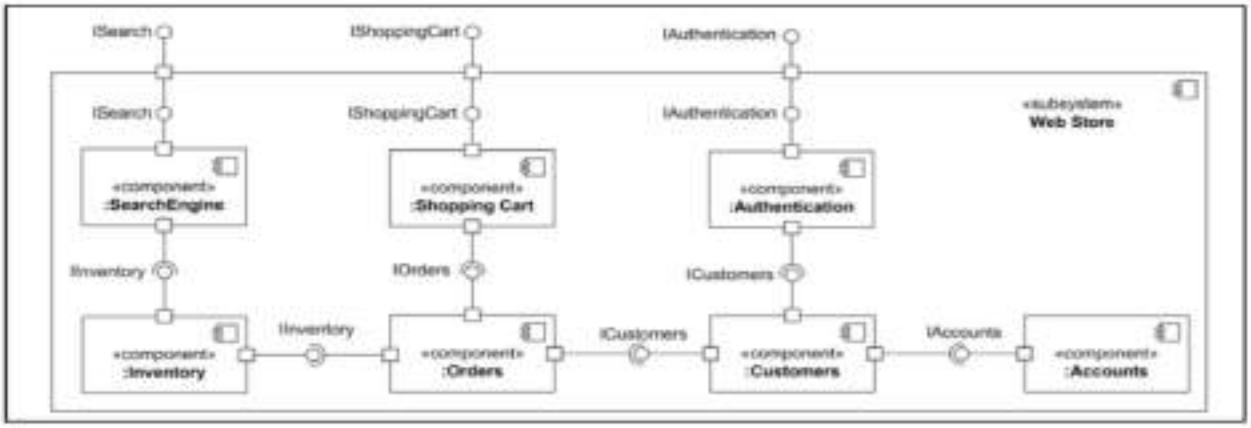


Fig 4.1.1.3 Component Diagram

4.1.1.4 Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So deployment diagrams are used to describe the static deployment of a system. Deployment diagrams consist of nodes and their relationships.

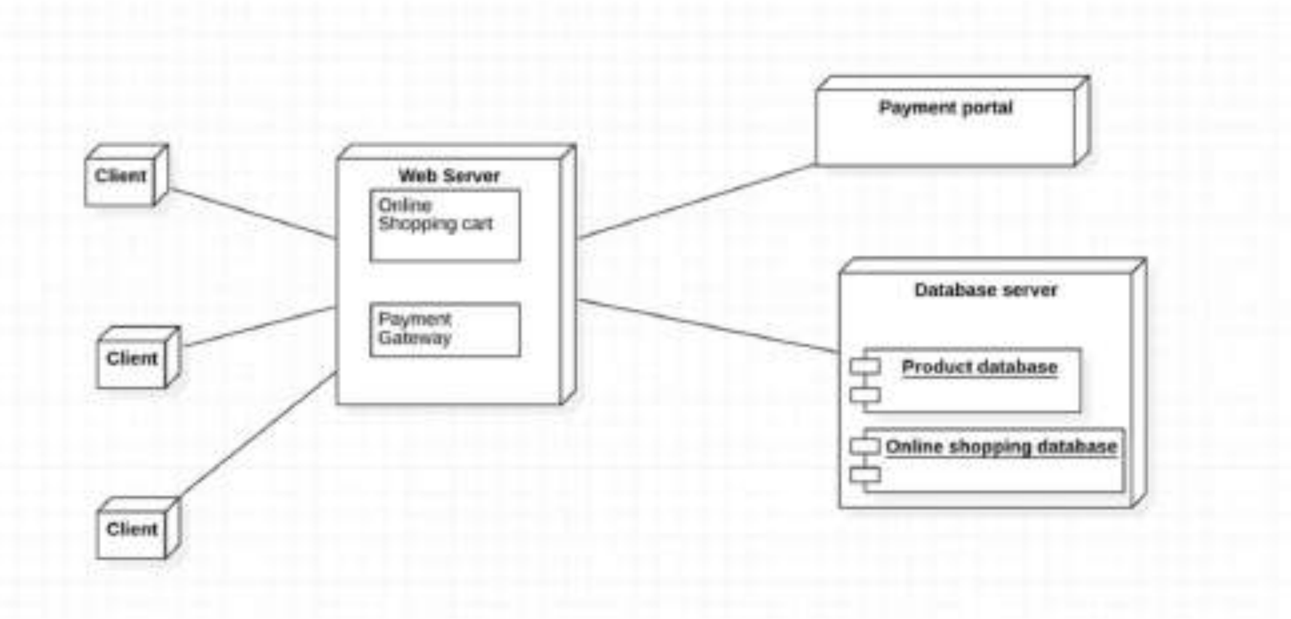


Fig 4.1.1.4 Deployment Diagram

4.1.2 Behavioral Diagram

4.1.2.1 Activity Diagram

This section lists the activity diagram and describes the flow of activities in the system. The figure below demonstrates the activity flow for this online shopping-cart application. The flow of the application is similar for both the user and administrator. The flow begins when the user first runs the application home screen online shopping-cart application that appears in the web browser.

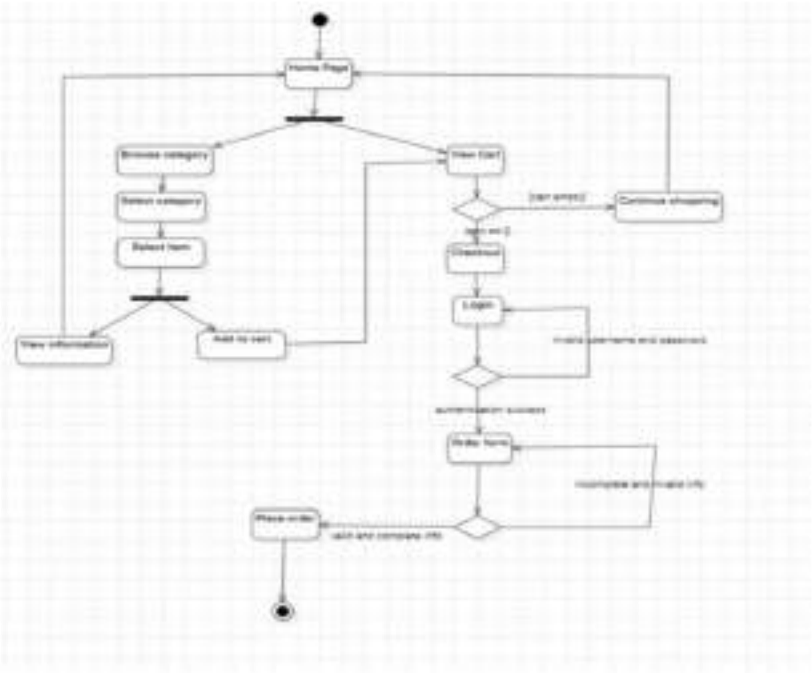


Fig 4.1.2.1 Activity Diagram

4.1.2.2 Collaboration Diagram

This section lists the activity diagram and describes the flow of activities in the system. The figure below demonstrates the activity flow for this online shopping-cart application. The flow of the application is similar for both the user and administrator. The flow begins when the user first runs the application home screen online shopping-cart application that appears in the web browser.

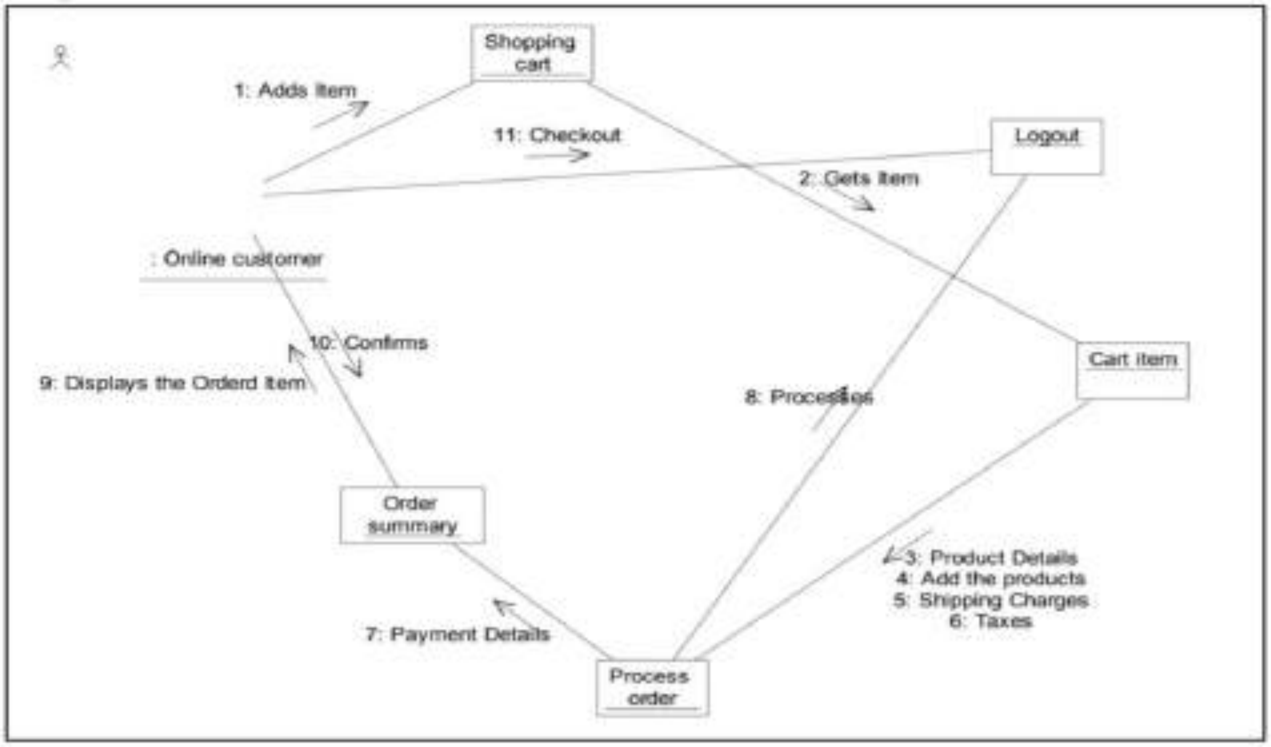


Fig 4.1.2.2 Collaboration Diagram

4.1.2.3 Sequence Diagram

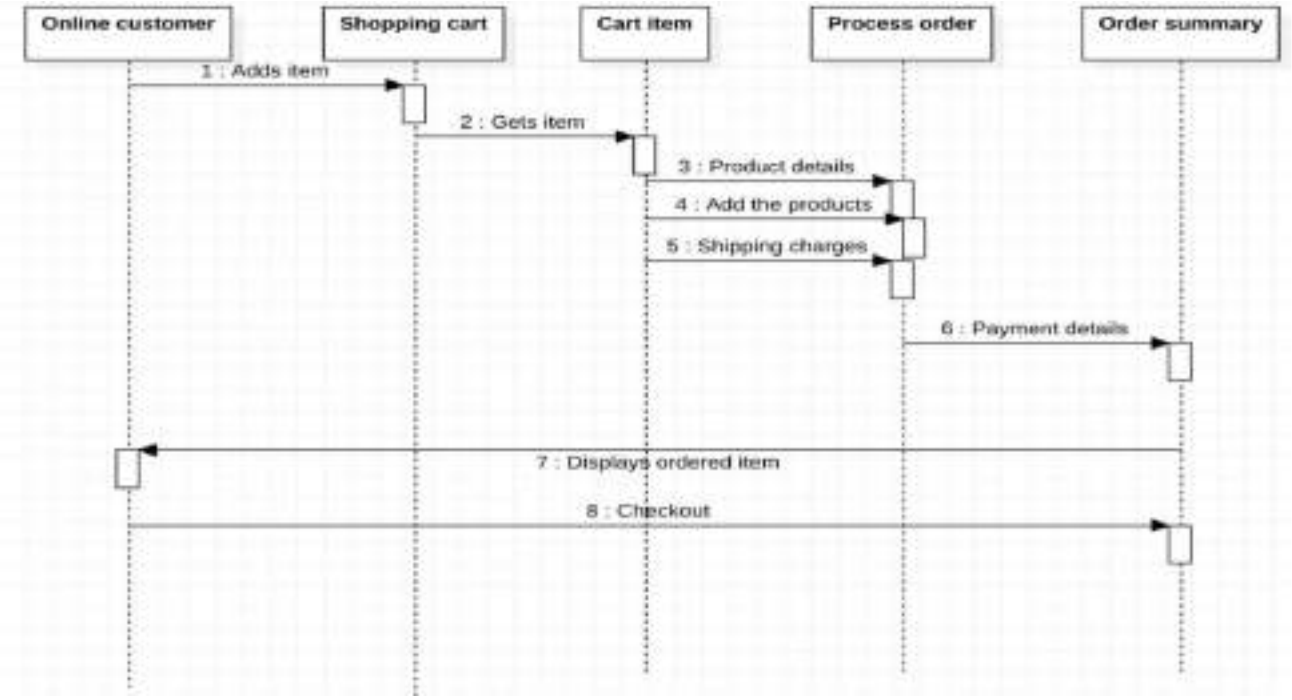


Fig 4.1.2.3 Sequence Diagram

4.1.2.4 Use case Diagram

The system's use case shows the user a detailed view of the system and how the actors would interact with each other and with the system.

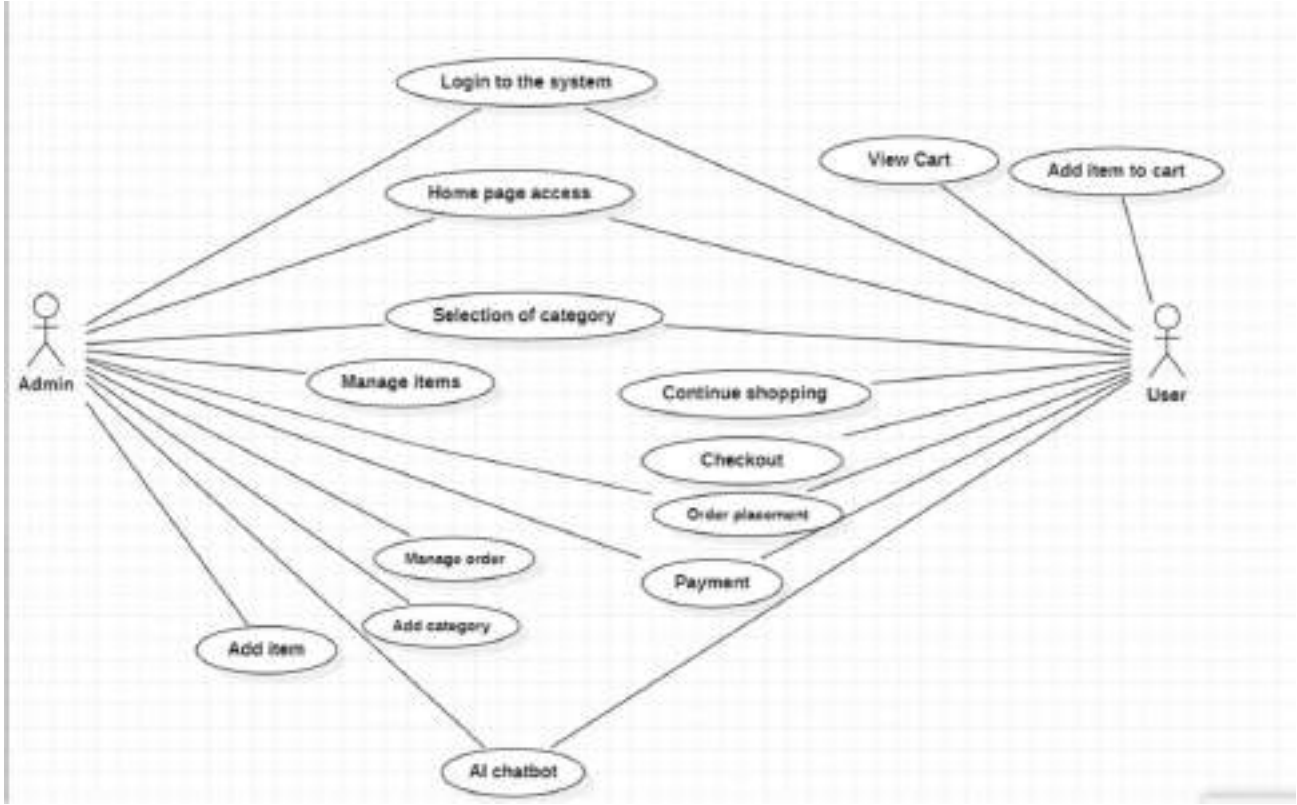


Fig 4.1.2.4 Use case Diagram

4.2 Algorithms and Flow Charts

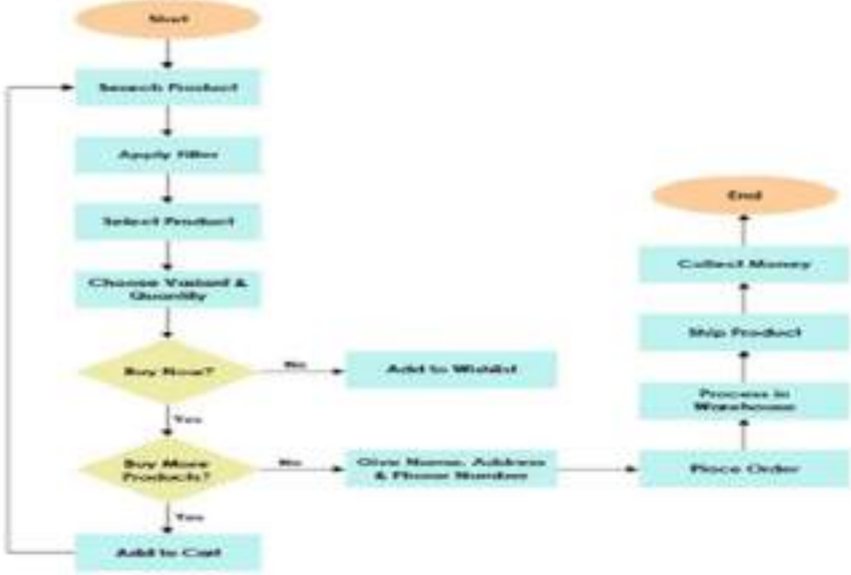


Fig 4.2.1 Flowchart

Chapter 5

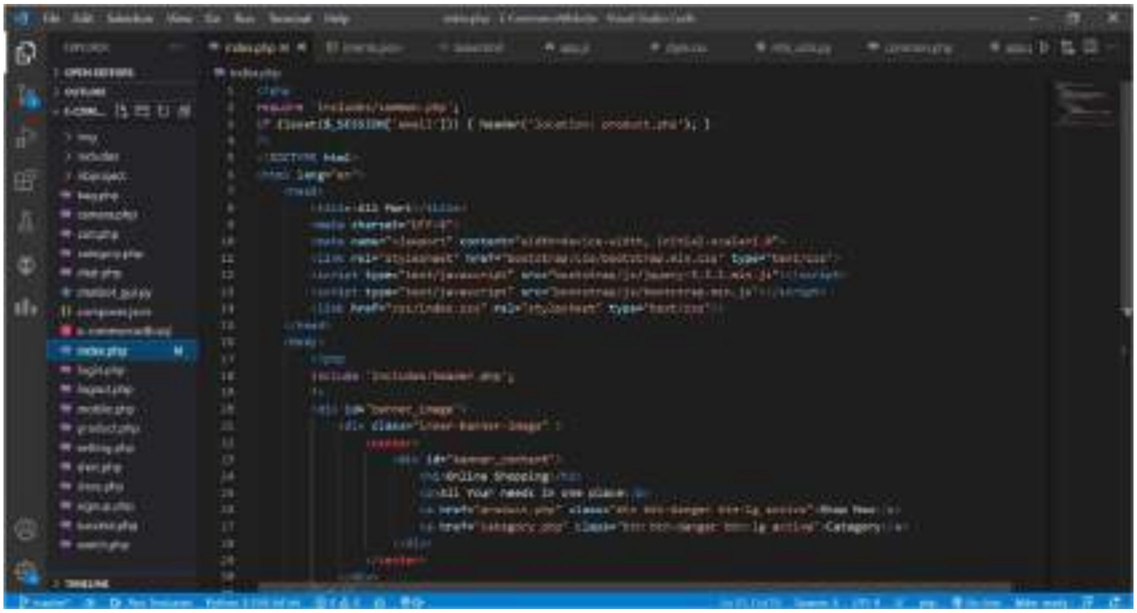
Development

5.1 Tools Description and Development approach used

- **Jupiter notebook**
- **VSCode IDE**
- **Python**
- **Php**
- **Xammp**

5.2 Pseudocodes of Important Modules.

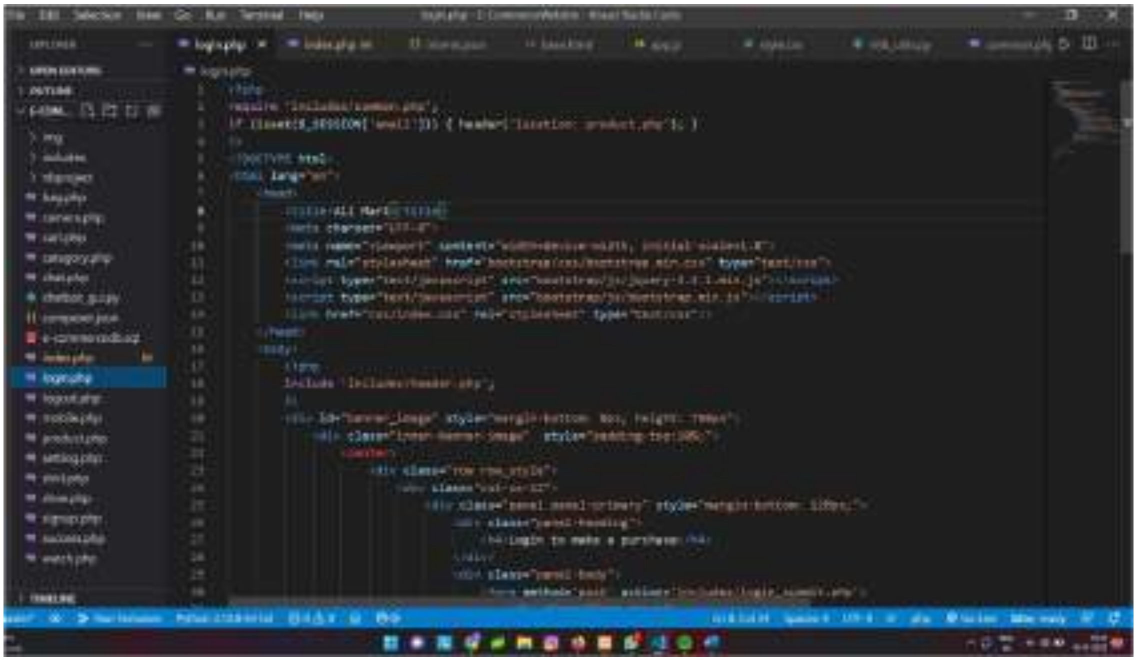
5.2.1 MAIN INDEX CODE THAT IS FIRST OPENED FOR THE SITE



```
<?php
require 'includes/common.php';
if (isset($_SESSION['email'])) { header('location: product.php'); }
?>
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>All Mart</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="stylesheet" href="bootstrap/css/bootstrap.min.css" type="text/css">
        <script type="text/javascript" src="bootstrap/js/jquery-3.3.1.min.js"></script>
        <script type="text/javascript" src="bootstrap/js/bootstrap.min.js"></script>
        <link href="css/index.css" rel="stylesheet" type="text/css">
    </head>
    <body>
        <?php
        include 'includes/header.php';
        ?>
        <div id="banner_image">
            <div class="inner-banner-image">
                <center>
                    <div id="banner_content">
                        <h1>Online Shopping</h1>
                        <p>All Your needs in one place.</p>
                        <a href="product.php" class="btn btn-danger btn-lg active">Shop Now</a>
                        <a href="category.php" class="btn btn-danger btn-lg active">Category</a>
                    </div>
                </center>
            </div>
```

Fig 5.2.1.1: Homepage

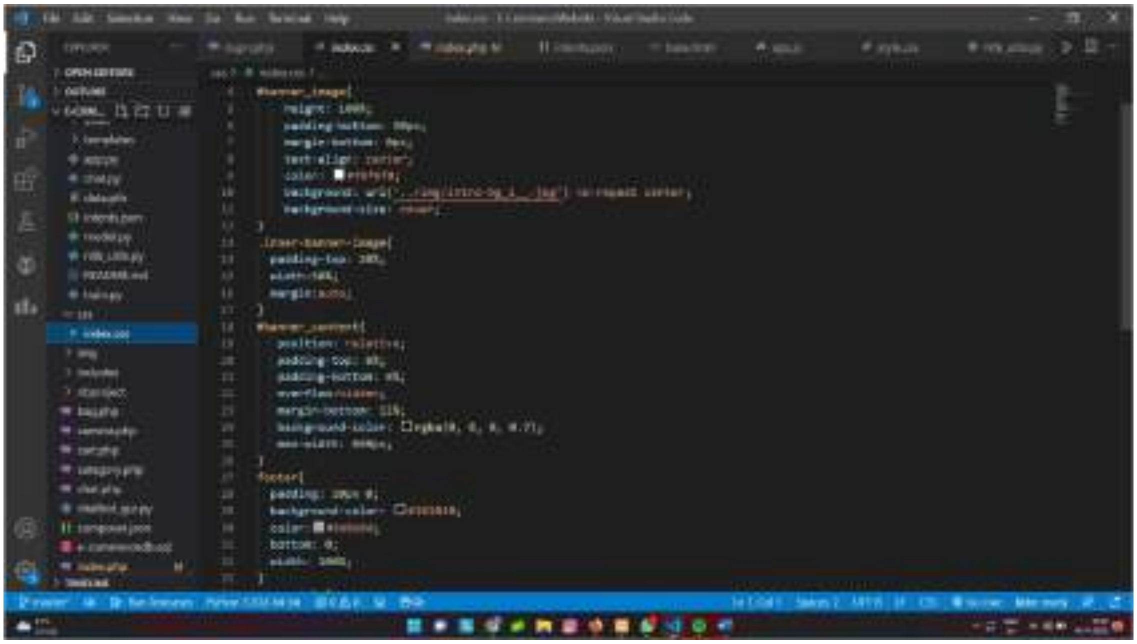
5.2.2 Login Page



```
<?php
require 'includes/common.php';
if (isset($_SESSION['email'])) { header('location: product.php'); }
?>
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Ali Mart</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="stylesheet" href="bootstrap/css/bootstrap.min.css" type="text/css">
        <script type="text/javascript" src="bootstrap/js/jquery-3.3.1.min.js"></script>
        <script type="text/javascript" src="bootstrap/js/bootstrap.min.js"></script>
        <link href="css/index.css" rel="stylesheet" type="text/css">
    </head>
    <body>
        <?php
        include 'includes/header.php';
        ?>
        <div id="banner_image" style="margin-bottom: 0px; height: 700px">
            <div class="inner-banner-image" style="padding-top:10%;">
                <center>
                    <div class="row row_style">
                        <div class="col-xs-12">
                            <div class="panel panel-primary" style="margin-bottom: 120px;">
                                <div class="panel-heading">
                                    <h4>Login to make a purchase</h4>
                                </div>
                                <div class="panel-body">
                                    <form method="post" action="includes/login_submit.php">
```

Fig 5.2.1.2: Login Page

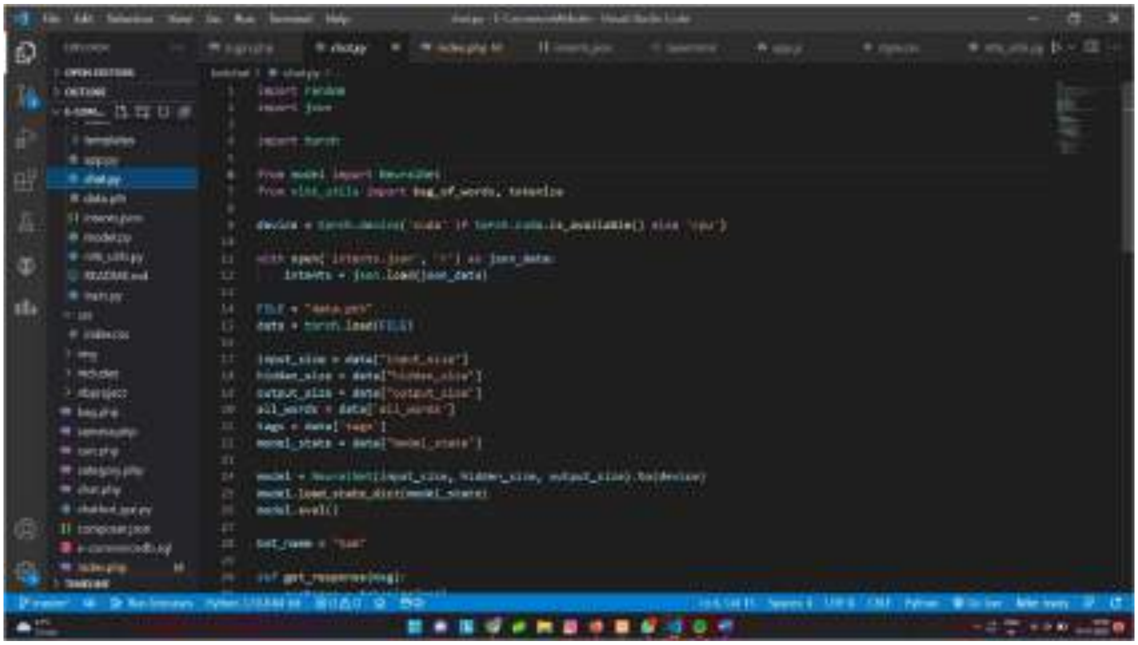
5.2.3 Styling the page



```
#banner_image{
    height: 100%;
    padding-bottom: 50px;
    margin-bottom: 0px;
    text-align: center;
    color: ■#f8f8f8;
    background: url("../img/intro-bg_1_.jpg") no-repeat center;
    background-size: cover;
}
 .inner-banner-image{
  padding-top: 10%;
  width:50%;
  margin:auto;
}
#banner_content{
   position: relative;
   padding-top: 6%;
   padding-bottom: 6%;
   overflow:hidden;
   margin-bottom: 11%;
   background-color: □rgba(0, 0, 0, 0.7);
   max-width: 660px;
}
footer{
  padding: 10px 0;
  background-color: □#101010;
  color: ■#9d9d9d;
  bottom: 0;
  width: 100%;
}
```

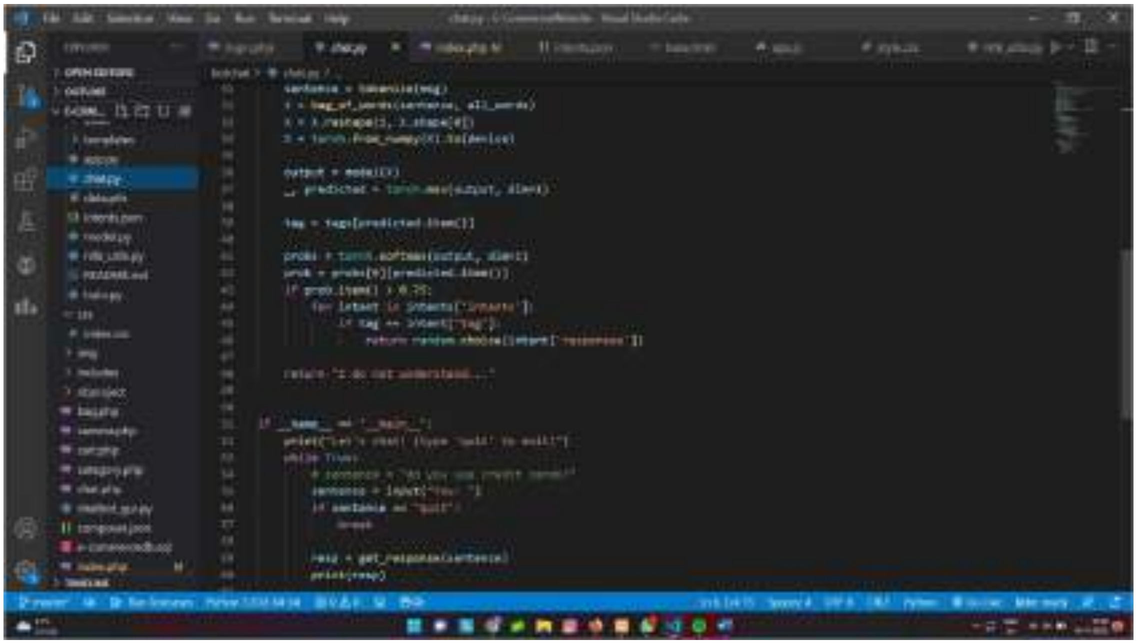
Fig 5.2.1.3: Stylesheet

5.2.4 Chatbot



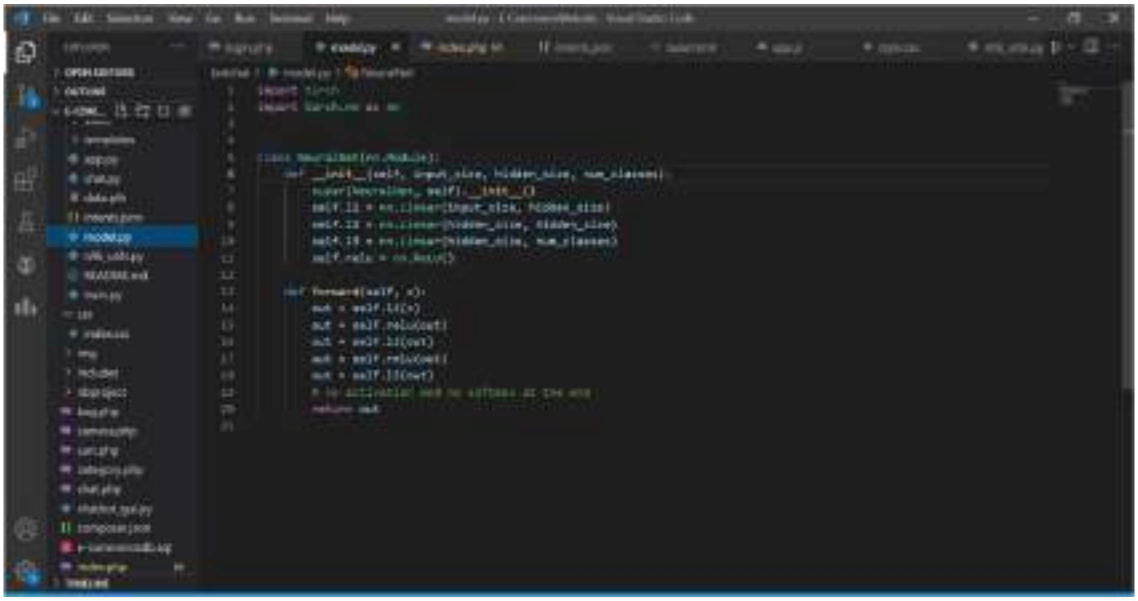
```
import random
import json

import torch

from model import NeuralNet
from nltk_utils import bag_of_words, tokenize

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

with open('intents.json', 'r') as json_data:
    intents = json.load(json_data)

FILE = "data.pth"
data = torch.load(FILE)

input_size = data["input_size"]
hidden_size = data["hidden_size"]
output_size = data["output_size"]
all_words = data['all_words']
tags = data['tags']
model_state = data["model_state"]

model = NeuralNet(input_size, hidden_size, output_size).to(device)
model.load_state_dict(model_state)
model.eval()

bot_name = "Sam"

def get_response(msg):
```

Fig 5.2.1.4: Chatbot



```
    sentence = tokenize(msg)
    X = bag_of_words(sentence, all_words)
    X = X.reshape(1, X.shape[0])
    X = torch.from_numpy(X).to(device)

    output = model(X)
    _, predicted = torch.max(output, dim=1)

    tag = tags[predicted.item()]

    probs = torch.softmax(output, dim=1)
    prob = probs[0][predicted.item()]
    if prob.item() > 0.75:
        for intent in intents['intents']:
            if tag == intent["tag"]:
                return random.choice(intent['responses'])

    return "I do not understand..."


if __name__ == "__main__":
    print("Let's chat! (type 'quit' to exit)")
    while True:
        # sentence = "do you use credit cards?"
        sentence = input("You: ")
        if sentence == "quit":
            break

        resp = get_response(sentence)
        print(resp)
```

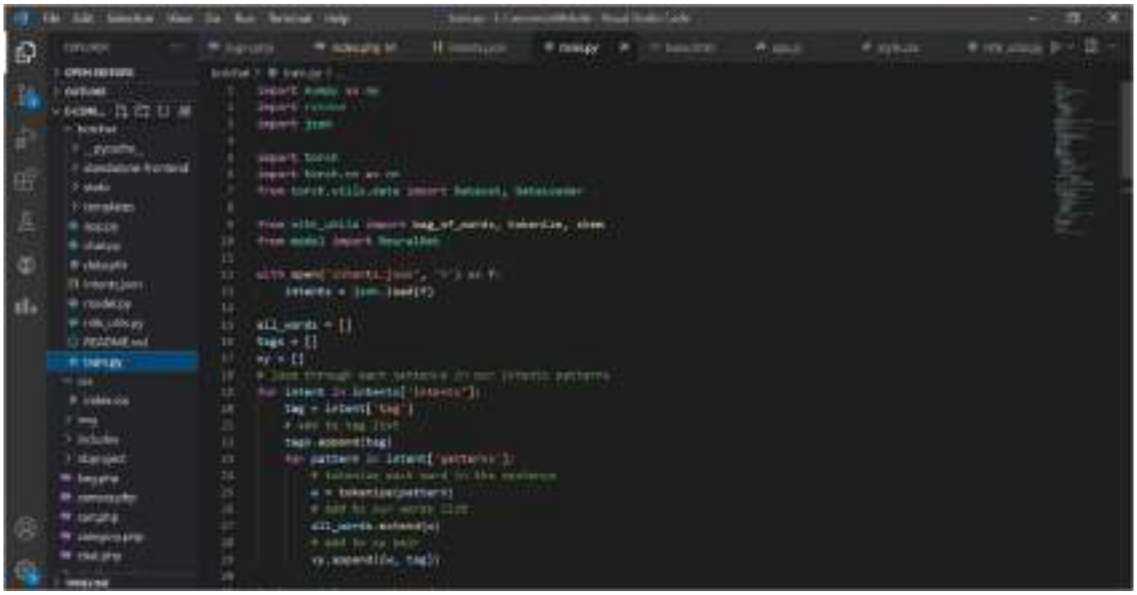
Fig 5.2.1.4a Chatbot cntd



```
import torch
import torch.nn as nn


class NeuralNet(nn.Module):
    def __init__(self, input_size, hidden_size, num_classes):
        super(NeuralNet, self).__init__()
        self.l1 = nn.Linear(input_size, hidden_size)
        self.l2 = nn.Linear(hidden_size, hidden_size)
        self.l3 = nn.Linear(hidden_size, num_classes)
        self.relu = nn.ReLU()

    def forward(self, x):
        out = self.l1(x)
        out = self.relu(out)
        out = self.l2(out)
        out = self.relu(out)
        out = self.l3(out)
        # no activation and no softmax at the end
        return out
```

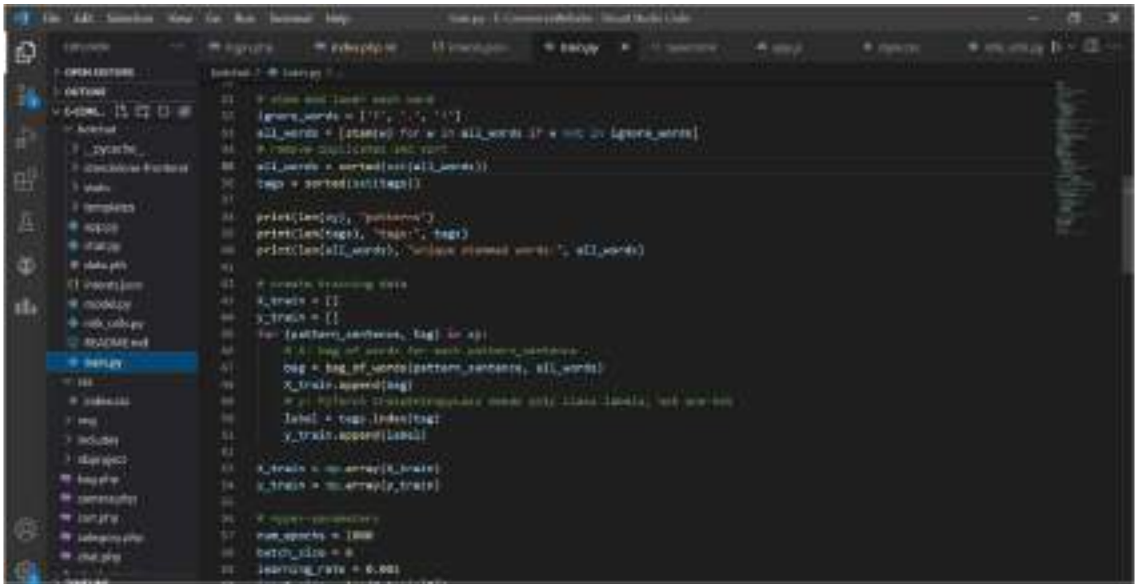
Fig 5.2.1.4b Chatbot cntd

5.2.5 training The Chatbot Model



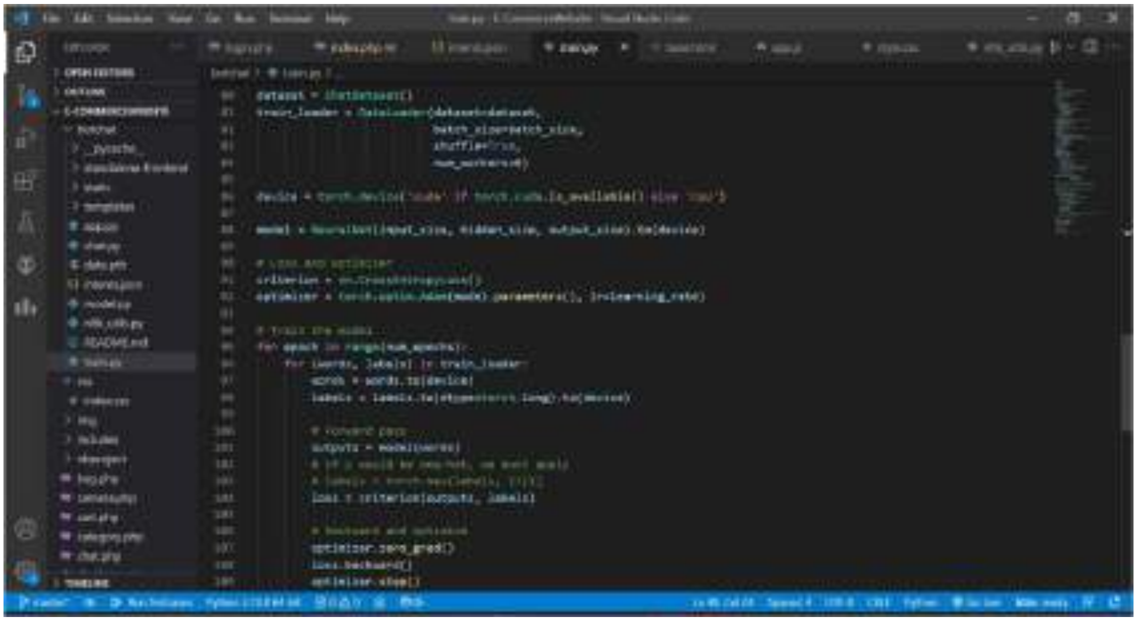
```
import numpy as np
import random
import json

import torch
import torch.nn as nn
from torch.utils.data import Dataset, DataLoader

from nltk_utils import bag_of_words, tokenize, stem
from model import NeuralNet

with open('intents.json', 'r') as f:
    intents = json.load(f)

all_words = []
tags = []
xy = []
# loop through each sentence in our intents patterns
for intent in intents['intents']:
    tag = intent['tag']
    # add to tag list
    tags.append(tag)
    for pattern in intent['patterns']:
        # tokenize each word in the sentence
        w = tokenize(pattern)
        # add to our words list
        all_words.extend(w)
        # add to xy pair
        xy.append((w, tag))
```

Fig 5.2.1.5 Chatbot training



```
# stem and lower each word
ignore_words = ['?', '.', '!']
all_words = [stem(w) for w in all_words if w not in ignore_words]
# remove duplicates and sort
all_words = sorted(set(all_words))
tags = sorted(set(tags))

print(len(xy), "patterns")
print(len(tags), "tags:", tags)
print(len(all_words), "unique stemmed words:", all_words)

# create training data
X_train = []
y_train = []
for (pattern_sentence, tag) in xy:
    # X: bag of words for each pattern_sentence
    bag = bag_of_words(pattern_sentence, all_words)
    X_train.append(bag)
    # y: PyTorch CrossEntropyLoss needs only class labels, not one-hot
    label = tags.index(tag)
    y_train.append(label)

X_train = np.array(X_train)
y_train = np.array(y_train)

# Hyper-parameters
num_epochs = 1000
batch_size = 8
learning_rate = 0.001
```

Fig 5.2.1.5a Chatbot training



```
dataset = ChatDataset()
train_loader = DataLoader(dataset=dataset,
                          batch_size=batch_size,
                          shuffle=True,
                          num_workers=0)

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

model = NeuralNet(input_size, hidden_size, output_size).to(device)

# Loss and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)

# Train the model
for epoch in range(num_epochs):
    for (words, labels) in train_loader:
        words = words.to(device)
        labels = labels.to(dtype=torch.long).to(device)

        # Forward pass
        outputs = model(words)
        # if y would be one-hot, we must apply
        # labels = torch.max(labels, 1)[1]
        loss = criterion(outputs, labels)

        # Backward and optimize
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
```

Fig 5.2.1.5b Chatbot training

Chapter 6

Testing

6.1 Test cases for Sign-in Modules

Table 6.1.1 Sign in module

Test ID	Test Priority	Module Name	Test Designed Data	Test Summary	Pre-Condition	Dependencies	Expected Result	Actual Result	Test Status	Post Conditions
1	High	Signup	Valid: Should not be empty and contain alphabets only. Invalid:If empty or contains any data other than alphabets.	Name	Sign Up for the system.	Mention any dependencies must be mentioned in other test cases or test requirement.	Valid: Allows to enter data in the next field (should not show any error message). Invalid: Display a message "Inavlid Name"	Valid: Allowed to enter data in the next field (should not show any error message). Invalid: A text message stating "Invalid Name" is displayed	Pass	The system must register new user after registration.

2	High	Signup	Valid: Should not be empty and contain email of the form ex:abc@xyz.com Invalid: if empty or not of the form ex:abc@xyz.com	Email	Sign up for the system	the test case must ensure that other test cases too are independent	Valid: Allows to enter data in the next field (should not show any error message). Invalid: Display a message “Invalid Email”	Valid: Allowed to enter data in the next field (should not show any error message).	Pass	The system must register new user after registration.
3	High	Signup	Valid: Should not be empty and Password size should not be less than	Password	Sign up for the system.	the test case must ensure that other test cases too are independent	Valid: Allows to enter the data in the next field(should not show any error	Valid: Allowed to enter data in the next field (shoul	Pass	the system must register new user after registration.

			6 (may contain alphabets, numbers, special characters). Invalid: if empty or password size is less than 6 (may contain alphabets, numbers, special characters).				message).Inavlid :Display a message ”Invalid Password”.	d not show any error message). Invalid: A text message stating “Invalid Password” is displayed.		
4	High	Signup	Valid: Should not be empty and isn’t less than equal to todays date and	Date of birth	Sign up for the system	the test case must ensure that other test cases too are independent.	Valid: Allows to enter data in the next field (should not show any error	Valid: Allowed to enter data in the next field (shoul	Pass	the system must register new user after registration.

			should be of the form dd-mm-yyyy. Invalid: if empty or > todays date and is not in the form of dd-mmyyyy.				message). Invalid: Display a message “Invalid DOB”.	d not show any error message). Invalid: A text message stating “Invalid DOB” is displayed.		
5	High	Signup	Valid: Should not be empty and contain only 10 digits. Test Executed By: This field will be filled after the	Contact no	Sign up for the system.	the test case must ensure that other test cases too are independent.	Valid: Allows to enter data in the next field (should not show any error message). I	Valid: Allowed to enter data in the next field (should not show any error messag	Pass	The system must register new user after registration.

			execution of a test case. Invalid: if empty, contains data other than numbers and also if number of digits is less than or greater than 10.				nvalid: Display a message “Invalid Contact”.	e). Invalid: A text message stating “Invalid Contact” is displayed		
6	High	Signup	Valid: Should not be empty. Invalid: If empty.	Address	Sign up for the system	The test case must ensure that other test cases too are independent	Valid: Allows to enter data in the next field (should not show any error message). I	Valid: Allowed to enter data in the next field (should not show any error message).	Pass	the system must register new user after registration.

							nvalid: Display a message “Invalid Address”.	Invalid: A text message stating “Invalid Address” is displayed.		
7	High	Signup	Valid: Should not be empty. Invalid: If empty.	Gender	Sign up for the system.	the test case must ensure that other test cases too are independent	Valid: Should not show any error message. Invalid: Display a message “Select Gender”.	Valid: Did not show any error message. Invalid: A text message stating “Option not selected” is displayed.	Pass	The system must register new user after registration.

8	High	Signup	Valid: On click, should check all the fields are filled properly or not and the data is stored into the database. Invalid: On click, should check all the fields are not filled properly or the data is not stored into the database.	Sign up button	Sign up for the system.	the test case must ensure that other test cases too are independent	Valid: Should not show any error message. Invalid: Display a message “Select Gender”.	Valid: Did not show any error message. Invalid: An error message is displayed ”data entered is not correct ”	Pass	the system must register new user after registration.

6.2 Test cases for Login Modules

Table 6.2.1 login module

Test ID	Test Priority	Module Name	Test Designed Data	Test Summary	Pre-Condition	Dependencies	Expected Result	Actual Result	Test Status	Post Conditions
9	High	Login	Valid: On click, should check all the fields are filled properly or not and the data is stored into the database. Invalid: On click, should check all the fields are not filled properly or the data is not	Email.	The user should already be a registered customer	the test case must ensure that other test cases too are independent	Valid: Allows to enter data in the next field (should not show any error message). Invalid: Display a message "Invalid Name"	Valid: Allowed to enter data in the next field (should not show any error message). Invalid: A text message stating "Invalid Name" is displayed	Pass	The system must register new user after registration.

			stored into the database							
10	High	Login	Valid: Should not be empty, Password size should not be less than 6 (may contain alphabets, numbers, special characters). And also should match with the data in the database. Invalid: if empty	Password	The user should already be a registered customer	the test case must ensure that other test cases too are independent	Valid: Allows to enter data in the next field (should not show any error message). Invalid: Display a message “Invalid Email”.	Valid: Allowed to enter data in the next field (should not show any error message). Invalid: A text message stating “Invalid Email” is displayed.	Pass	the system must register new user after registration.

			or password size is less than 6 (may contain alphabets, numbers, special characters) or does not match with the database data.							
11	High	Login	Valid: On click, should check all the fields are filled properly or not and the data should match with the data in the database.	Login button	The user should already be a registered customer	the test case must ensure that other test cases too are independent.	Valid: Should not show any error message. Invalid: Display an error	Valid: Did not show any error message. Invalid: An error message is displayed ”data	Pass	the system must register new user after registration.

			Invalid: On click, should check all the fields are not filled properly or the data is not matching with the data in the database.				message.	entered is not Matching”.		

6.3 Test cases for Product Page

Table 6.3.1 Product Page

Test ID	Test Priority	Module Name	Test Designed Data	Test Summary	Pre-Condition	Dependencies	Expected Result	Actual Result	Test Status	Post Conditions
12.	High	Add to cart button.	button	Verify whether the product is added to the cart page	Connected to the Network, supported browser should be there. Account should Exist already.	Item to be added and counted	Check whether more than 1 product can be added to cart page	Check whether more than 1 product can be added to cart page	Pass	To proceed buy or purchasing back
13.	High	Check whether more than 1 product can be added to cart page	Integer, button	Verify whether more than 1 product gets added to the cart page.	Connected to the Network, supported browser should be there. Account should Exist already.	Item to be added and counted	Multiple products should be added to the cart page.	Multiple products should be added to the cart page.	Pass	To proceed buy or purchasing back

6.4 Test cases for Chart Page

Table 6.4.1 Chart Page

Test ID	Test Priority	Module Name	Test Designed Data	Test Summary	Pre-Condition	Dependencies	Expected Result	Actual Result	Test Status	Post Conditions
14	high	Item Removal from Cart	Integer, button	Valid: Clicking on remove item leads to item removal and price deducted from the grand total.	Connected to the Network, supported browser should be there. Account should	Item to be removed and counted	Immediate Removal of Item and cost deduction.	Product got added successfully.	Pass	To proceed buy or purchasing back
		Item Removal from Cart	Integer, button	Invalid: Clicking on remove item leads to item non-removal and price non-deduction from the grand total	Connected to the Network, supported browser should be there. Account should	Item to be removed and counted	Invalid: Item stays in the Cart and gets purchased Without an error	Multiple products got added in the cart page	Fail	To proceed buy or purchasing back

Test ID	**Test Priority**	**Module Name**	**Test Designed Data**	**Test Summary**	**Pre-Conditions**	**Depen -dencies**	**Expected Result**	**Actual Result**	**Test Status**	**Post Conditions**
15.	High	Products added	Character ,Integer	For each item added, a corres-ponding name, price and the total price of all items are shown.	Connected to the Network, supported browser should be there. Account should Exist already.	The items present on the website.	The products should be added with total price.	The products got added and the total price is shown.	Pass	To proceed buy or purchasing back or adding to cart
16	High	Name	Character	Valid: Should not be empty and contain alphabets only.	Connected to Network, supported browser Account should Exist	Object specifica-Tions,valid name	Valid: Allows to enter data in the next field (should not show any error message)..	Valid: Allowed to enter data in the next field (should not show any error message)..	Pass	Adding items and quantity numbers
				Invalid: If empty or contains any data other than alphabets.	Connected to the Network, supported browser should be there. Account should	Object specifica-Tions,valid name	Invalid: Display a message “Invalid Name”	Invalid: A text message stating “Invalid Name” is displayed	Pass	Adding items and quantity numbers

17	High	Email	.charachter	: Valid: Should not be empty and contain email of the form ex:abc@xyz.com.	Connected to the Network, supported browser should be there. Account should	Valid email id	Valid: Allows to enter data in the next field (should not show any error message).	Valid: Allowed to enter data in the next field (should not show any error message).	Pass	
			charachter	Invalid: if empty or not of the form ex:abc@xyz.com	Connected to the Network, supported browser should be there. Account should existed	Valid email Id	Invalid: Display a message “Invalid Email”	Invalid: A text message stating “Invalid Email” is displayed	Pass	
18	High	Contact	integer	Valid: Should not be empty and contain only 10 digits	Connected to the Network, supported browser should be there. Account should existed	Valid contact number	Valid: Allows to enter data in the next field (should not show any error message)	Valid: Allowed to enter data in the next field (should not show any error message).	Pass	
			integer	Invalid: if empty, contains data other than numbers and also if number of digits is less than or grater than 10	Connected to the Network, supported browser should be there.	Valid contact number	Invalid: Display a message “Invalid Contact”	Invalid: A text message stating “Invalid Contact”	Pass	

					Account should			is displayed		
19	High	Address	charachter	Valid: Should not be empty.	Connected to the Network, supported browser should be there. Account should	Valid pincode	Valid: Allows to enter data in the next field (should not show any error message).	Valid: Allowed to enter data in the next field (should not show any error message).	Pass	
			charachter	Invalid: if empty	Connected to the Network, supported browser should be there. Account should	Valid pincode	Invalid: Display a message “Invalid Address”	Invalid: A text message stating “Invalid Address” is displayed	Pass	
20.	High	Payment Mode	Button,	Valid: Should be selected	Connected to the Network, supported browser should be there. Account should,items	Cost calculation,	Valid: Should be any of the 3 given payment methods	Valid: Did not show any error message	Pass	Money pay

					should be selected					
				Invalid: Non selected	Connected to the Network, supported browser should be there. Account should, items should be selected	Cost calculation,	Invalid: Display a message “Select Payment Mode”	Invalid: A text message stating “Select Payment Mode” is displayed	Pass	Money pay
21	High	Confirm Order button	button	Valid: On click ,should check all the fields are filled properly and the data is stored into the database	Connected to the Network, supported browser should be there. Account should		Valid: Should not show any error message and Proceed to Payment Portal [External API]	Valid: Did not show any error message and proceeds to payment Portal [External API]	Pass	Delivery time will come
			button	Invalid: On click ,should check	Connected to the Network, supported		Invalid: Display an error message	Invalid: An error message is	Pass	

				all the fields are not filled properly or the data is not stored into the database	browser should be there. Account should			displayed ”data entered is not correct”		

6.5 Test cases for Chatbot

Table 6.5.1 Chat bot

Test Id	Test Priority	Module Name	Test Designed Data	Test Summary	Pre-Conditions	Dependencies	Expected Result	Actual Result	Test Status	Post Conditions
22	Medium	AI chatbot Interaction/ Validation	"Hi" "what do you sell" "do you use debit card" "Thanks"	To check the whether the inputs result in a successful interaction with the AI chatbot	should have proper knowledge on what to ask and how to frame it	table internet connection and connection to database	Valid: Clicking on the AI chatbot for concerned queries	Valid: Successful interaction with the AI chatbot including End User Satisfaction	Pass	none

								and Query Management		
23	Medium	AI chatbot Interaction/ Validation	"Hi" "what do you sell" "do you use debit card" "Thanks"	To check the whether the inputs result in a successful interaction with the AI chatbot	should have proper knowledge on what to ask and how to frame it	table internet connection and connection to database	Invalid: Non-Interaction with the AI chatbot or not clicking on the AI chatbot	Invalid: None Due to Invalid Input	Pass	none

6.6 Non Functional Requirements test cases

Table 6.6.1: non functional requirements

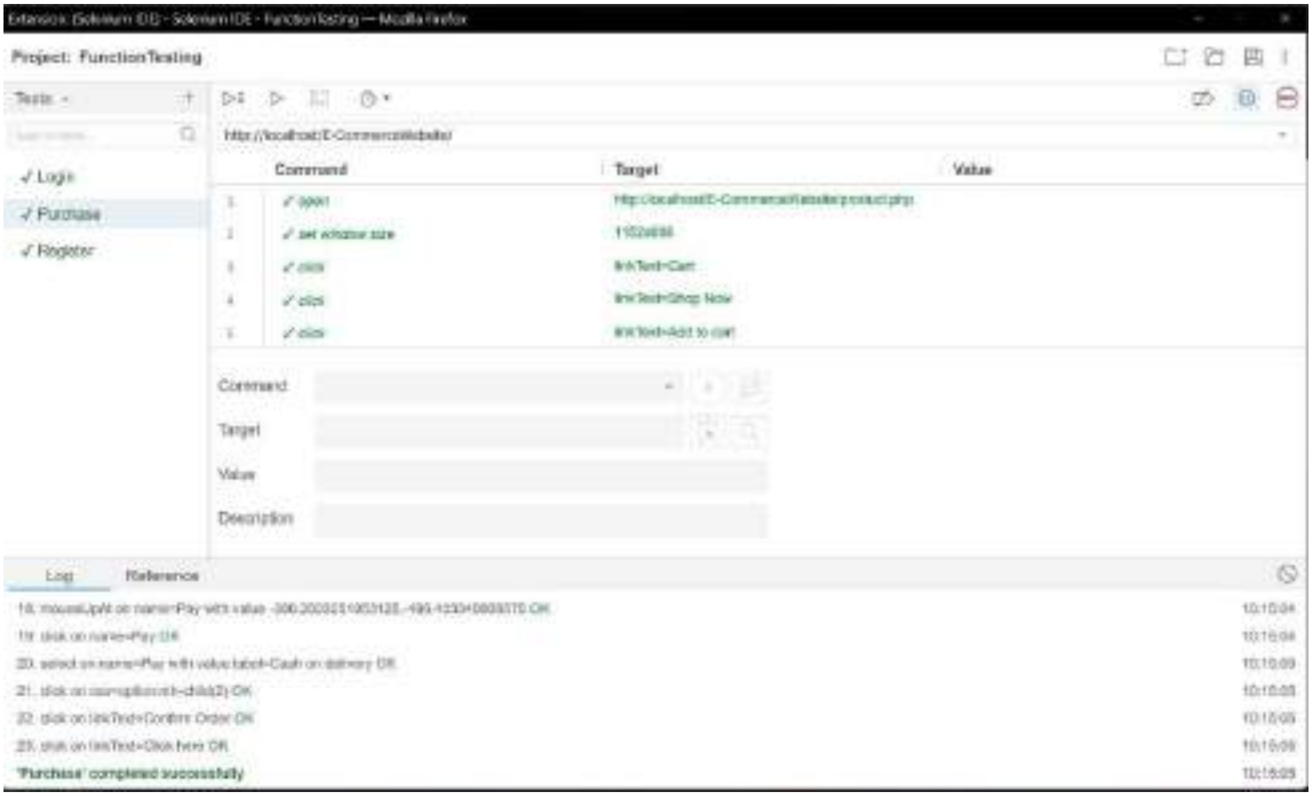
Test Id	Test Priority	Module Name	Test Summary	Pre-condition	Dependency	Expected Result	Actual Result	Test Status	Post-condition
1	high	performance	To find out the response time of the website operations	Website should be active Internet connection	N.A	Load operations <=.5s	Load operations <=.5s	Pass	N.A

2	High	security	It is connected for a software to find out the flaws that may be present in the website which may be lead us to compromise data	Website loaded to homepage.	Dependent on hosting Server	No security vulnerabilities	No security vulnerabilities	Pass	N.A
3	High	Load test	It is conducted for finding out the traffic the website can handle when multiple users enter into website	N.A	Hosting server	Webpage loading with minimal response time	Webpage loading with minimal response time	Pass	Return to initial stage

4	Medium	Stress Testing	Finding out what happens to the software behaviour in the abnormal condition	Abnormal Website	N.A	The software is stable and recoverable	The software is stable and recoverable	Pass	Return to initial stage
5.	High	Recovering Test	Bringing back the website when its down due to crashes or any attacks	Force the system to crash	N.A	Bringing back the website	Bringing back the website	Pass	Bringing back the original state

6.6 Selenium Testing

6.6.1 Testing Output



Extension: (Selenium IDE) - Selenium IDE - FunctionTesting — Mozilla Firefox

Project: FunctionTesting

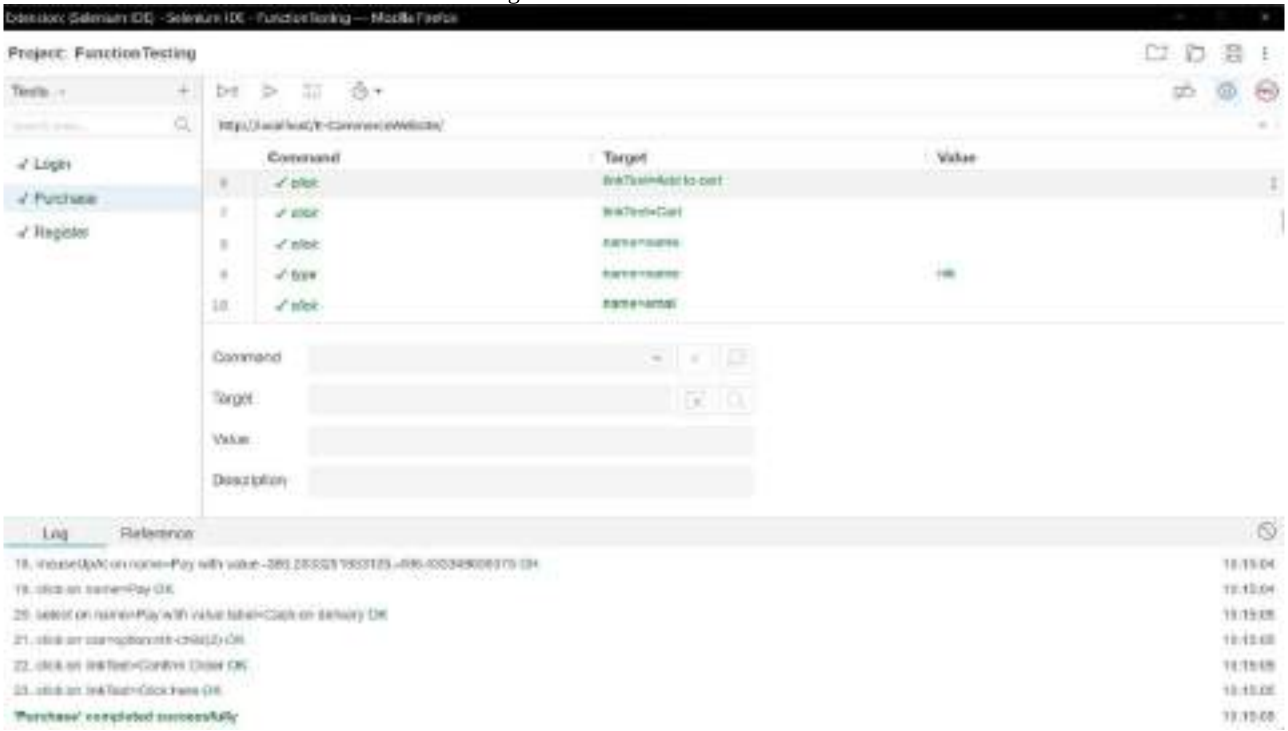
Tests: ✓ Login, ✓ Purchase, ✓ Register

http://localhost/E-CommerceWebsite/

	Command	Target	Value
1	✓ open	http://localhost/E-CommerceWebsite/product.php	
2	✓ set window size	1152x838	
3	✓ click	linkText=Cart	
4	✓ click	linkText=Shop Now	
5	✓ click	linkText=Add to cart	

```
18. mouseUpAt on name=Pay with value -306.2000251953125,-495.4233[illegible] OK    10:15:04
19. click on name=Pay OK                                                          10:15:04
20. select on name=Pay with value label=Cash on delivery OK                       10:15:05
21. click on css=option:nth-child(2) OK                                           10:15:05
22. click on linkText=Confirm Order OK                                            10:15:05
23. click on linkText=Click here OK                                               10:15:05
'Purchase' completed successfully                                                 10:15:05
```

Fig 6.6.1: test 1



Extension: (Selenium IDE) - Selenium IDE - FunctionTesting — Mozilla Firefox

Project: FunctionTesting

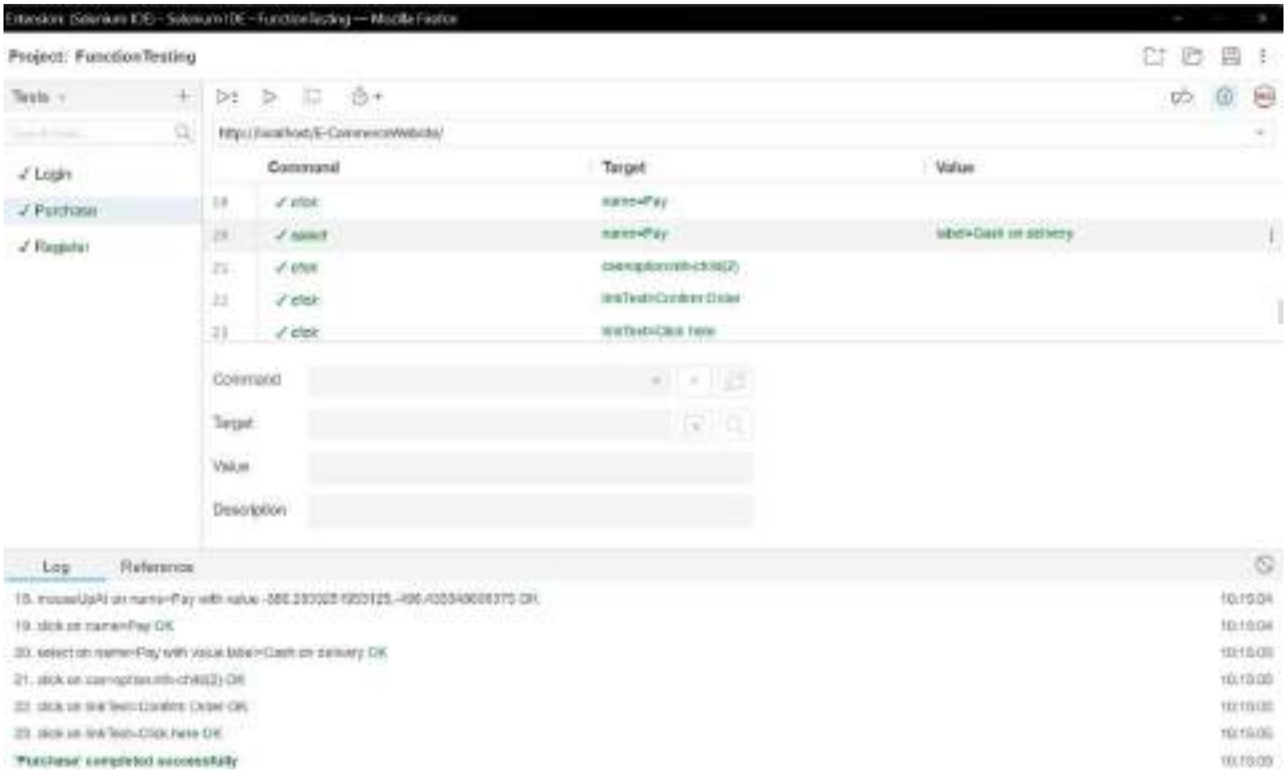
Tests: ✓ Login, ✓ Purchase, ✓ Register

http://localhost/E-CommerceWebsite/

	Command	Target	Value
6	✓ click	linkText=Add to cart	
7	✓ click	linkText=Cart	
8	✓ click	name=name	
9	✓ type	name=name	[illegible]
10	✓ click	name=email	

```
18. mouseUpAt on name=Pay with value -385.283[illegible] OK    11:15:04
19. click on name=Pay OK                                       11:15:04
20. select on name=Pay with value label=Cash on delivery OK    11:15:05
21. click on css=option:nth-child(2) OK                        11:15:05
22. click on linkText=Confirm Order OK                         11:15:05
23. click on linkText=Click here OK                            11:15:05
'Purchase' completed successfully                              11:15:05
```

Fig 6.6.2: test 2

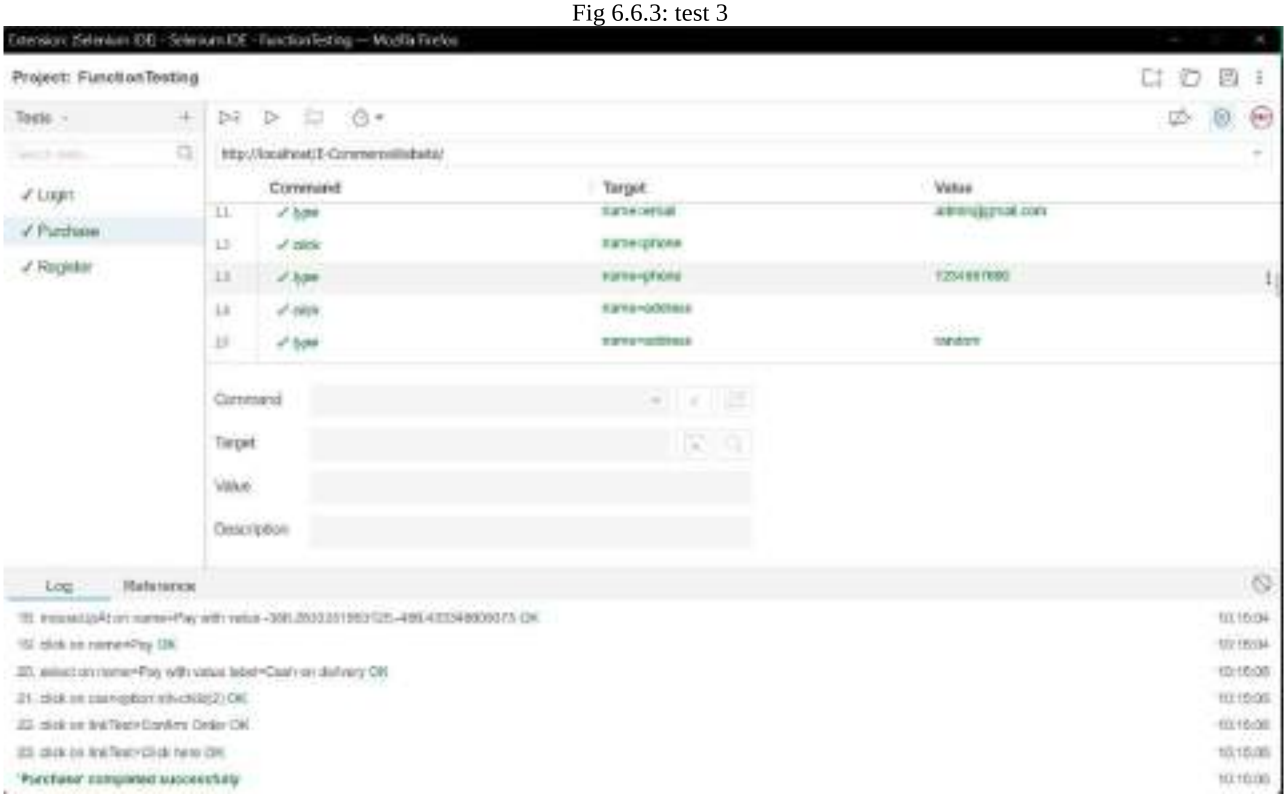


http://localhost/E-CommerceWebsite/

	Command	Target	Value
19	✓ click	name=Pay	
20	✓ select	name=Pay	label=Cash on delivery
21	✓ click	css=option:nth-child(2)	
22	✓ click	linkText=Confirm Order	
23	✓ click	linkText=Click here	

```
18. mouseUpAt on name=Pay with value -388.28302[illegible],-496.[illegible] OK    10:15:04
19. click on name=Pay OK                                                         10:15:04
20. select on name=Pay with value label=Cash on delivery OK                      10:15:05
21. click on css=option:nth-child(2) OK                                          10:15:05
22. click on linkText=Confirm Order OK                                           10:15:05
23. click on linkText=Click here OK                                              10:15:05
'Purchase' completed successfully                                                10:15:05
```

Fig 6.6.3: test 3



http://localhost/E-CommerceWebsite/

	Command	Target	Value
11	✓ type	name=email	admin@gmail.com
12	✓ click	name=phone	
13	✓ type	name=phone	1234567890
14	✓ click	name=address	
15	✓ type	name=address	random

```
18. mouseUpAt on name=Pay with value -388.28302[illegible],-496.[illegible] OK    10:15:04
19. click on name=Pay OK                                                         10:15:04
20. select on name=Pay with value label=Cash on delivery OK                      10:15:05
21. click on css=option:nth-child(2) OK                                          10:15:05
22. click on linkText=Confirm Order OK                                           10:15:05
23. click on linkText=Click here OK                                              10:15:05
'Purchase' completed successfully                                                10:15:05
```

Fig 6.6.4: test 4

Chapter 7

Screenshots of Developed Product

7.1 Homepage

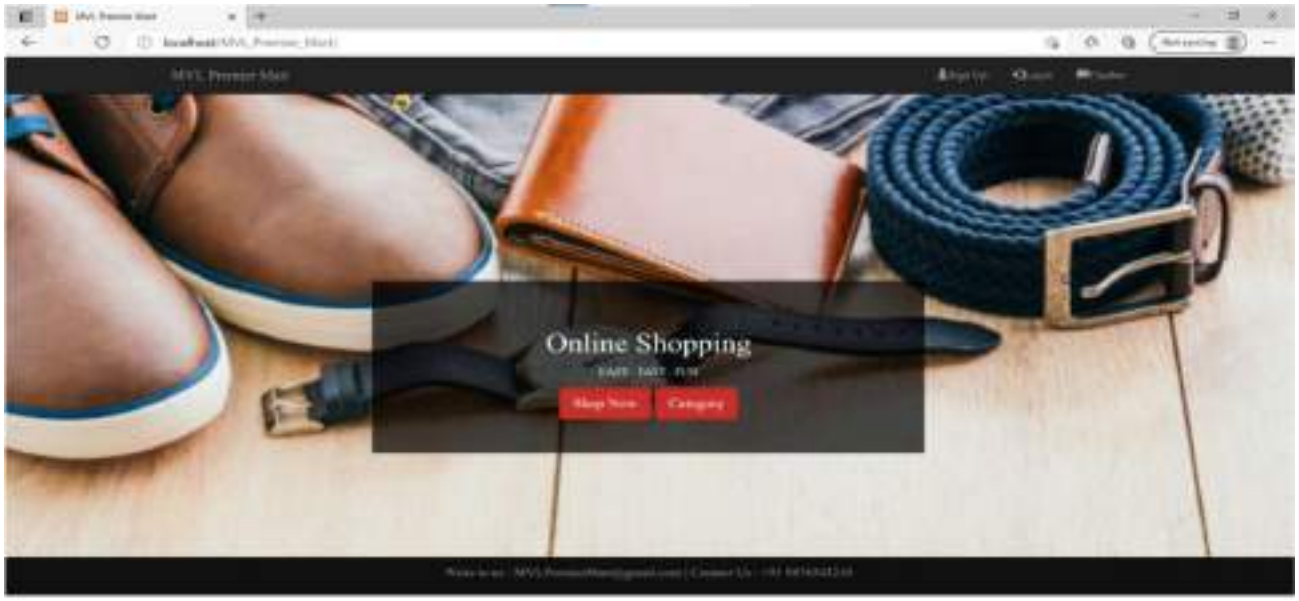


Fig 7.1 Homepage

7.2 category page:

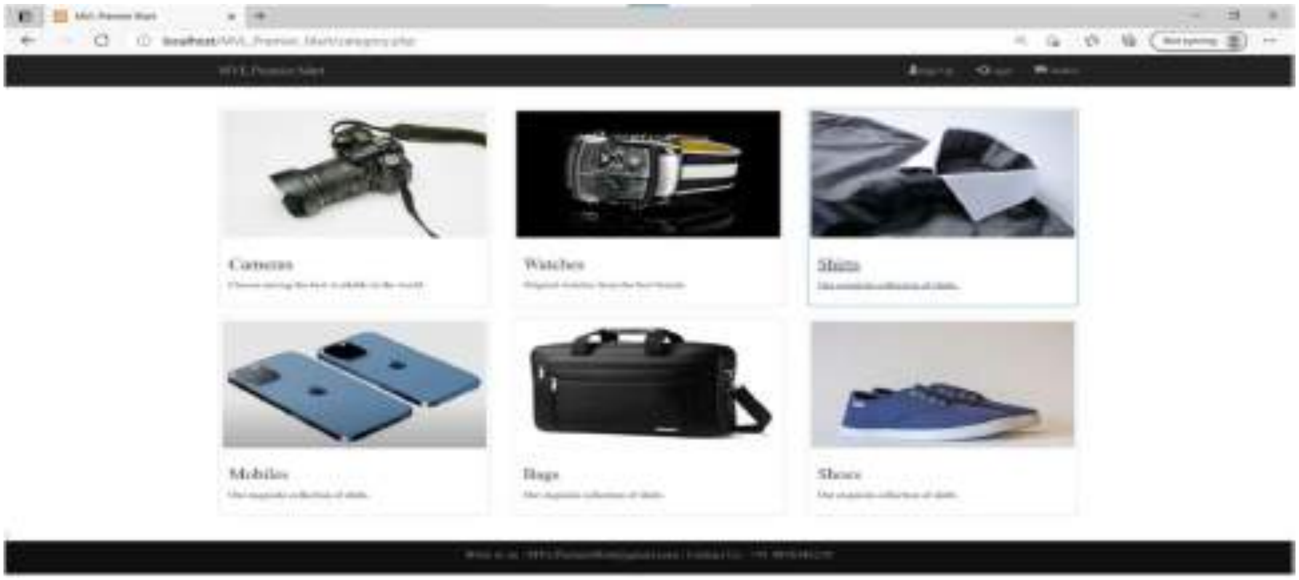
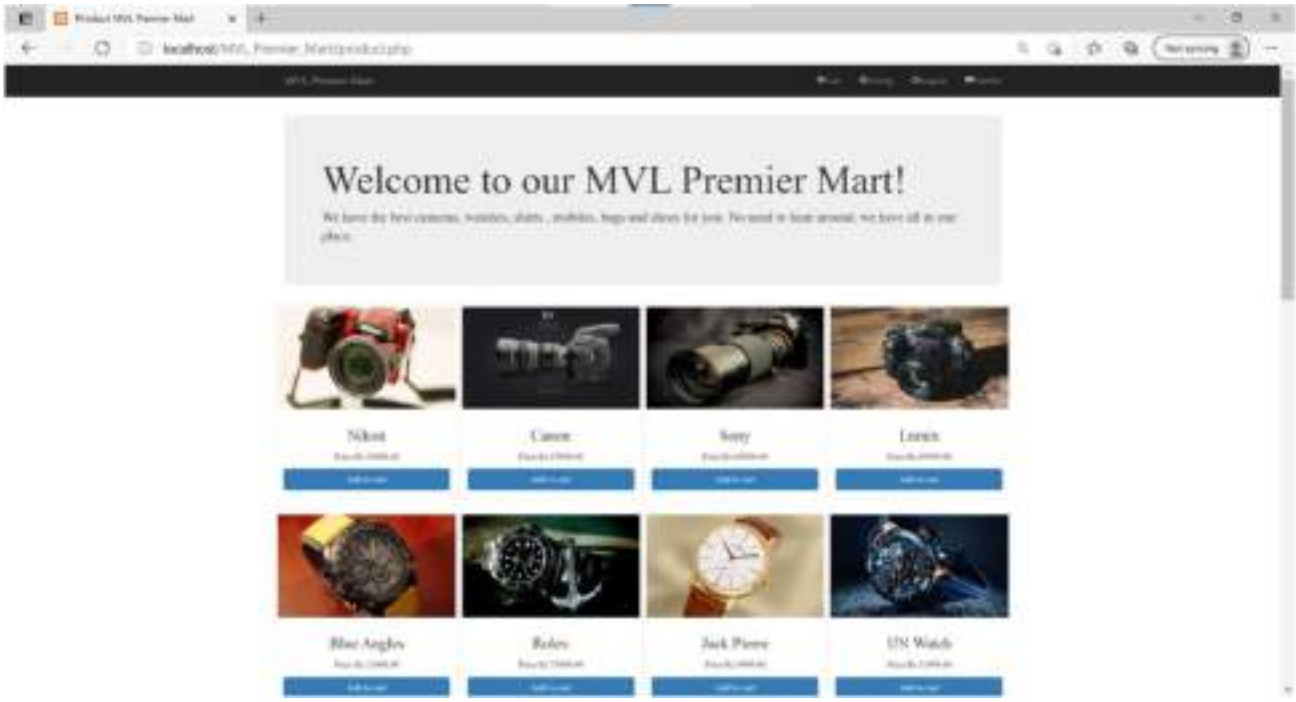


Fig 7.2 Category

7.3 Product page:



localhost/MVL_Premier_Mart/product.php

MVL Premier Mart

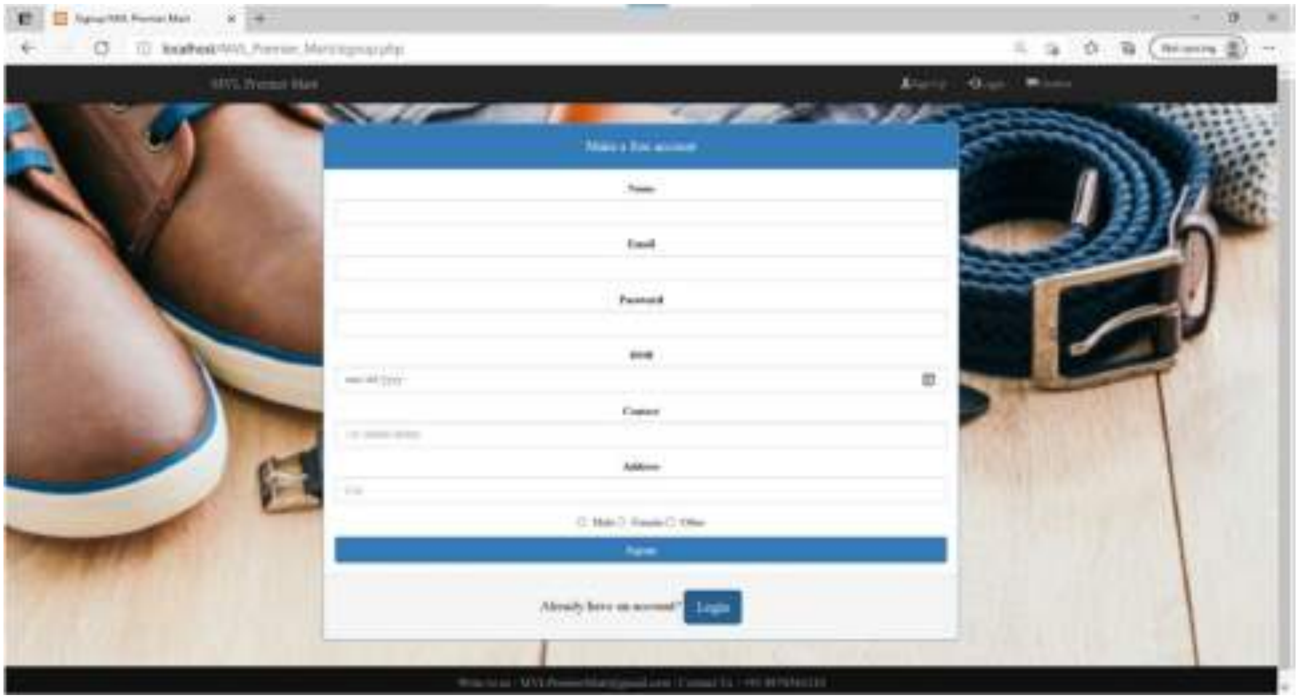
Welcome to our MVL Premier Mart!

We have the best cameras, watches, shirts , mobiles, bags and shoes for you. No need to hunt around, we have all in one place.

Nikon	Canon	Sony	Lorex
Blue Angles	Rolex	Jack Pierre	US Watch

Fig 7.3 Products

7.4 Sign up page:



localhost/MVL_Premier_Mart/signup.php

MVL Premier Mart

Make a free account

Name

Email

Password

DOB

Contact

Address

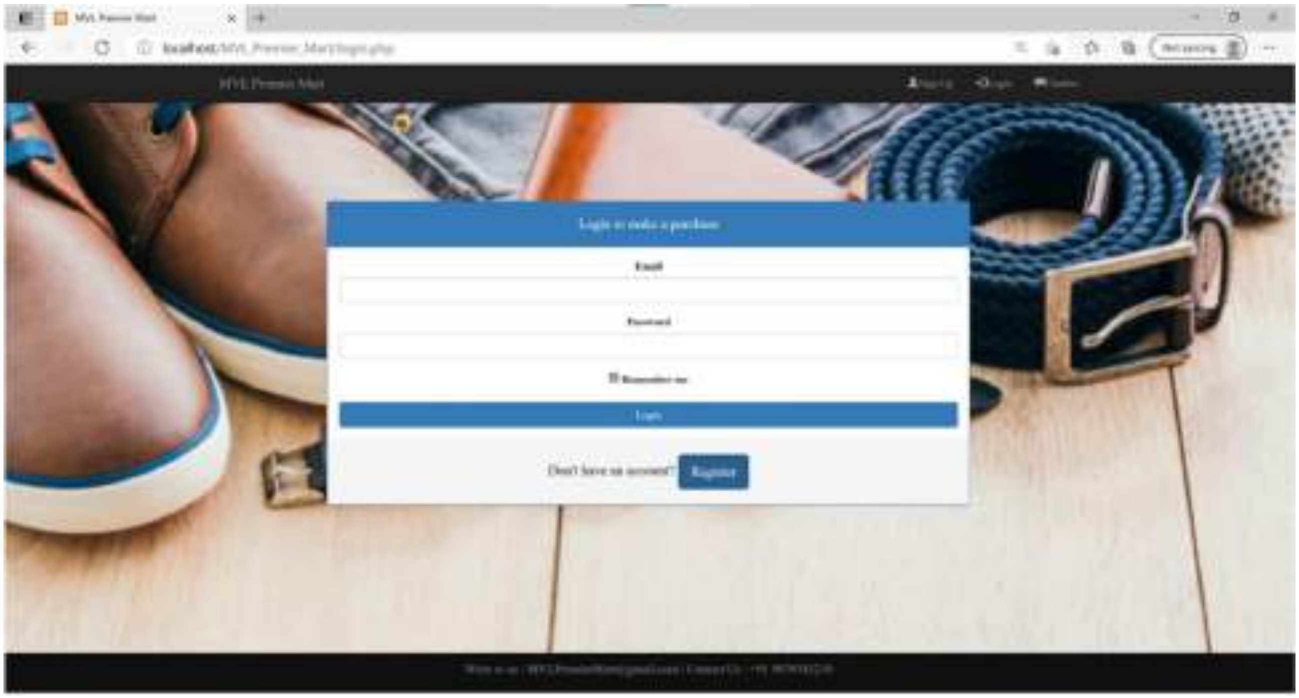
○ Male ○ Female ○ Other

Signup

Already have an account? Login

Fig 7.4 Signup page

7.5 Login page:



Login to make a purchase

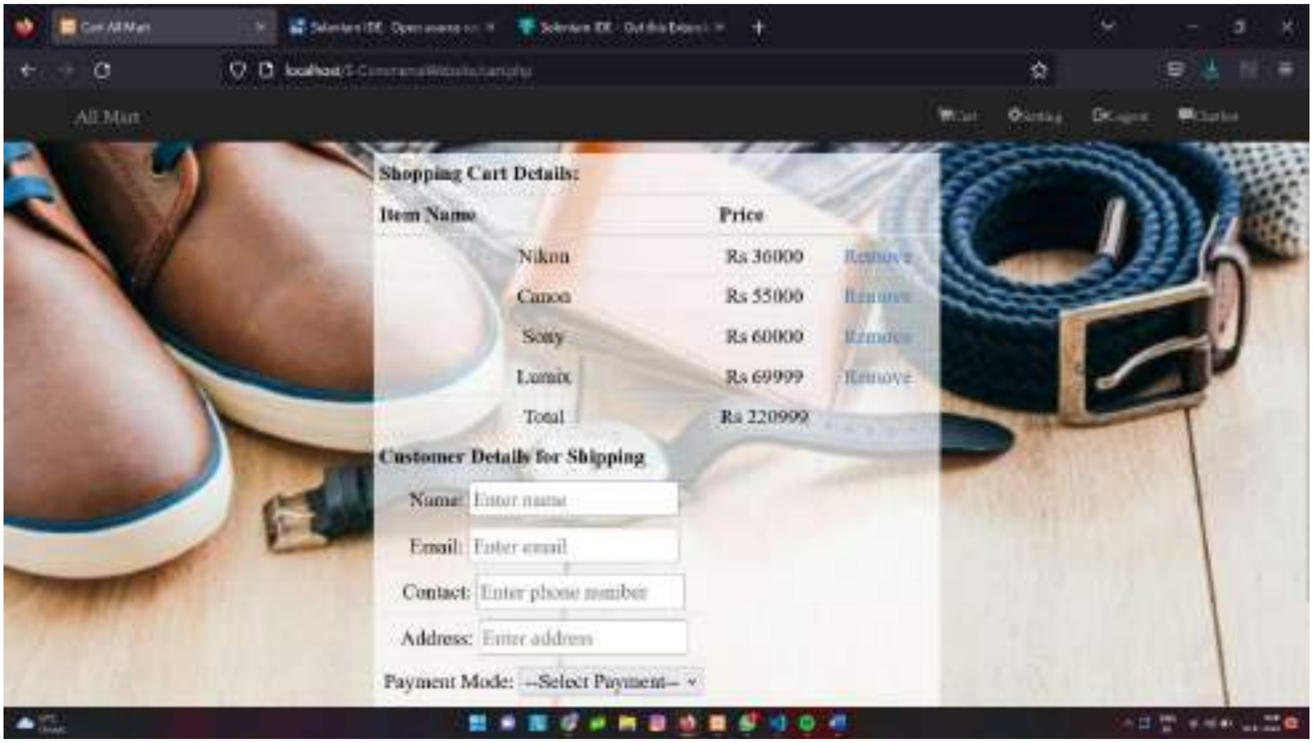
Email

Password

□ Remember me

Fig 7.5 Login page

7.6 Cart page:



Shopping Cart Details:

Item Name	Price	
Nikon	Rs 36000	Remove
Canon	Rs 55000	Remove
Sony	Rs 60000	Remove
Lumix	Rs 69999	Remove
Total	Rs 220999	

Customer Details for Shipping

Name: Enter name

Email: Enter email

Contact: Enter phone number

Address: Enter address

Payment Mode: --Select Payment--

Fig 7.6 Cart page

7.7 Confirmation page:

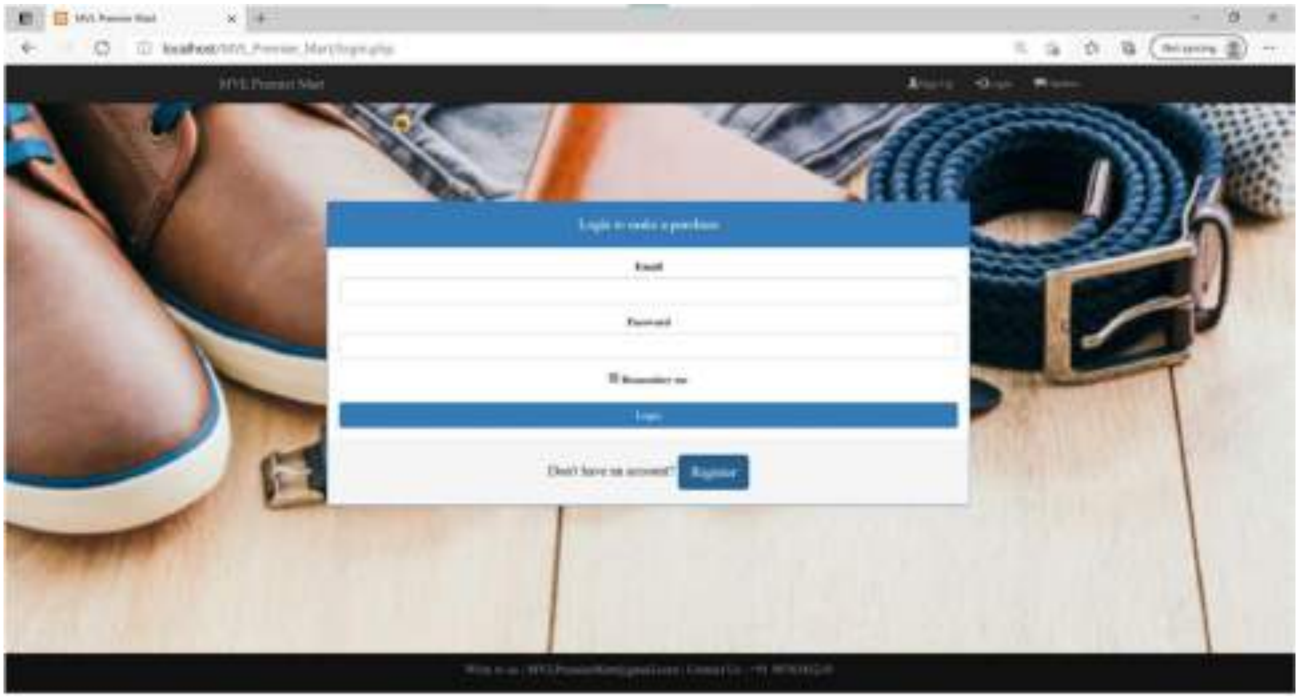


Fig 7.7 Confirmation page

References:

1. https://www.hindawi.com/journals/acisc/2021/6630326/#related-articles

2. https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1516&context=etd_projects

3. https://www.researchgate.net/profile/Amit-Mittal-14/publication/338335856_IJARCCE_A_Comparative_Study_of_Chatbots_and_Humans/links/5e0d912a4585159aa4ab6dcb/IJARCCE-A-Comparative-Study-of-Chatbots-and-Humans.pdf

4. https://sci-hub.se/https://link.springer.com/article/10.1631/FITEE.1700826

5. https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1337&context=wi2019

6. https://www.academia.edu/download/68742433/ijca2018916215.pdf

7. https://sci-hub.se/https://dl.acm.org/doi/abs/10.1145/3351556.3351587

8. https://www.researchgate.net/profile/AmirRAsadi/publication/327931660_Design_and_implementation_of_a_chatbot_for_ecommerce/links/5badda3645851574f7ebe60f/Design-and-implementation-of-a-chatbot-for-e-commerce.pdf

Annexures

1. Review I Presentation

2. Review II Presentation

3. Review III Presentation

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

AI Based E-Commerce Web Application with Integrated Chat-Bot

GROUP MEMBERS

C.VIPIN-20BCE2160
HARSHIT SATISH-20BCE2407
NIKHIL-20BCE2154
UJJWAL AKASH-20BCE2202
SHASHWAT-19BCE0808

Scope statement

- Building the e-commerce website for more user friendly experience.
- building AI search
- Payment gateway
- Chatbot for user quieries.

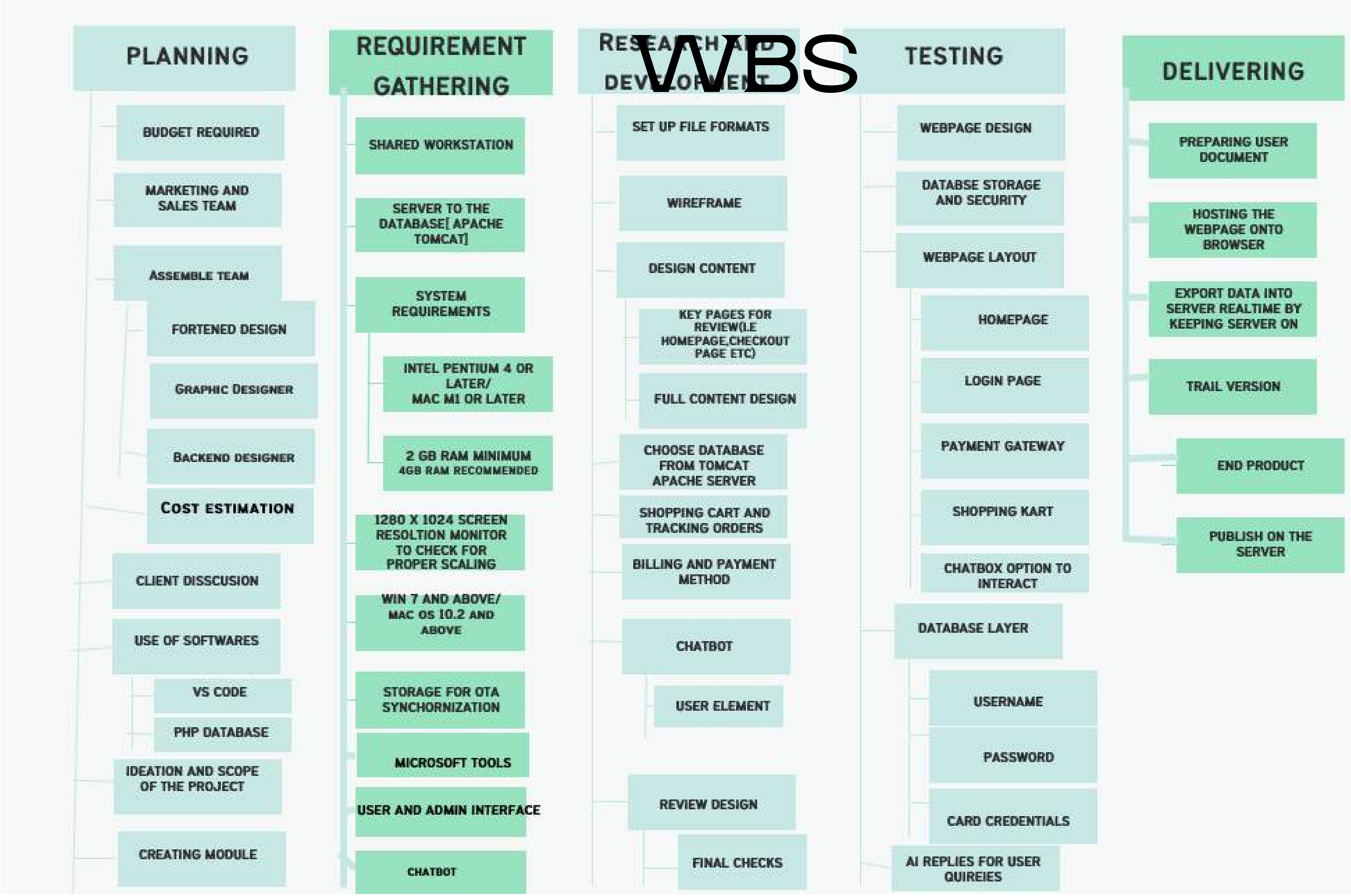
Features of the product

1. Display all the categories available for shopping on the system's main page.
2. Display all the items linked to each category listed on the main page.
3. Maintain records for many customers
4. Allow administrator to update stock quantities
5. Maintain data associated with the inventory (a collection of products)
6. Checkout
7. Allow administrator to specify a stop-order for a product
8. Notify administrator when products need to be reordered
9. Customer Support/Chatbot

MVP

- Chatbot which gives more relatable answers to user which helps us to reduce the operational cost in an organization and create more user-friendly environment which makes the user feel more secure.
- Easy maneuverability for the user through the portal which makes the purchase more convenient and fast tracked

WBS



PLANNING

- BUDGET REQUIRED
- MARKETING AND SALES TEAM
- ASSEMBLE TEAM
  - FORTENED DESIGN
  - GRAPHIC DESIGNER
  - BACKEND DESIGNER
  - COST ESTIMATION
- CLIENT DISSCUSION
- USE OF SOFTWARES
  - VS CODE
  - PHP DATABASE
- IDEATION AND SCOPE OF THE PROJECT
- CREATING MODULE

REQUIREMENT GATHERING

- SHARED WORKSTATION
- SERVER TO THE DATABASE[APACHE TOMCAT]
- SYSTEM REQUIREMENTS
  - INTEL PENTIUM 4 OR LATER/ MAC M1 OR LATER
  - 2 GB RAM MINIMUM 4GB RAM RECOMMENDED
- 1280 X 1024 SCREEN RESOLTION MONITOR TO CHECK FOR PROPER SCALING
- WIN 7 AND ABOVE/ MAC OS 10.2 AND ABOVE
- STORAGE FOR OTA SYNCHRONIZATION
- MICROSOFT TOOLS
- USER AND ADMIN INTERFACE
- CHATBOT

RESEARCH AND DEVELOPMENT

- SET UP FILE FORMATS
- WIREFRAME
- DESIGN CONTENT
  - KEY PAGES FOR REVIEW(I.E HOMEPAGE,CHECKOUT PAGE ETC)
  - FULL CONTENT DESIGN
- CHOOSE DATABASE FROM TOMCAT APACHE SERVER
- SHOPPING CART AND TRACKING ORDERS
- BILLING AND PAYMENT METHOD
- CHATBOT
  - USER ELEMENT
- REVIEW DESIGN
  - FINAL CHECKS

TESTING

- WEBPAGE DESIGN
- DATABSE STORAGE AND SECURITY
- WEBPAGE LAYOUT
  - HOMEPAGE
  - LOGIN PAGE
  - PAYMENT GATEWAY
  - SHOPPING KART
  - CHATBOX OPTION TO INTERACT
- DATABASE LAYER
  - USERNAME
  - PASSWORD
  - CARD CREDENTIALS
- AI REPLIES FOR USER QUIREIES

DELIVERING

- PREPARING USER DOCUMENT
- HOSTING THE WEBPAGE ONTO BROWSER
- EXPORT DATA INTO SERVER REALTIME BY KEEPING SERVER ON
- TRAIL VERSION
  - END PRODUCT
  - PUBLISH ON THE SERVER

Methodology

We are following the Royce's original waterfall model, the following phases are followed in order:

1. System and software requirements: captured in a product requirements document
2. Analysis: resulting in models, schema, and business rules
3. Design: resulting in the software architecture
4. Coding: the development, proving, and integration of software
5. Testing: the systematic discovery and debugging of defects
6. Deployment: Hosting the application.

Thus, the waterfall model maintains that one should move to a phase only when its preceding phase is reviewed and verified.

Milestones and Deliverables

- Source code
- Enhanced website to give better user experience
- Integrating the website with an AI powered chatbot
- We'll be producing the user manual with supporting documents
- The resources should be efficiently used and work should be carried out according to the planned schedule.

Tools Used

- We have used Canva for designing the Work Breakdown Structure (WBS).
- We have used Project Libre for scheduling the tasks and generating the gantt chart. We'll also know the resources allocated to each task and the cost associated with it, we can obtain an estimation cost which needed for the project to be taken up.
- We will be using VS Code to implement the frontend and backend of the web application.
- We will be using PHP for database connectivity because its more user-friendly.

Cost estimation

Effort= 3 * (0.8)^1.12= 2.336 PM
T dev= 2.5* (2.336)^0.35 =3.364 Months ~ 100 days

Thank you!

VIT®

Vellore Institute of Technology

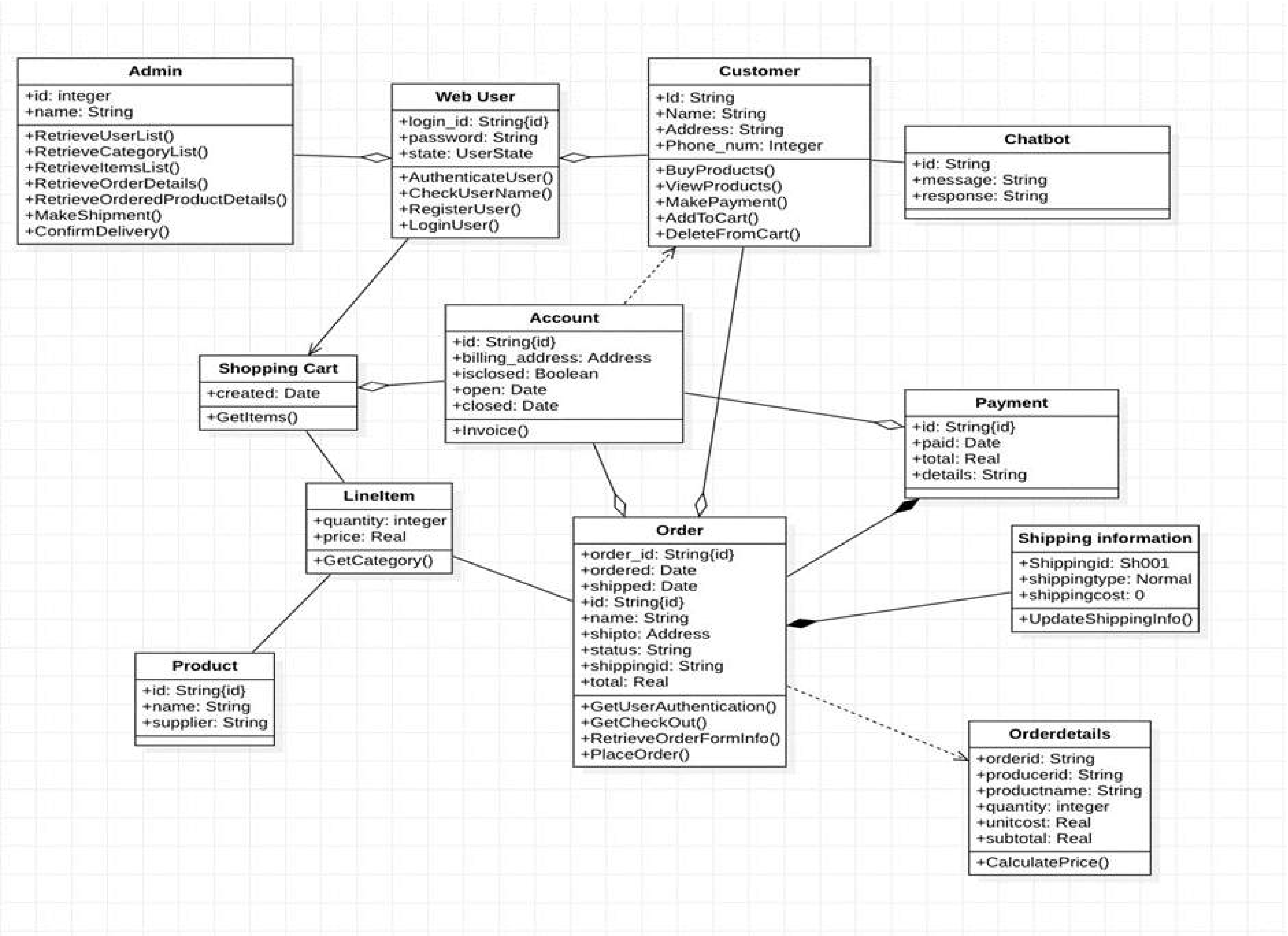
(Deemed to be University under section 3 of UGC Act, 1956)

AI Based E-Commerce Web Application with Integrated Chat-Bot

GROUP MEMBERS

C.VIPIN	20BCE2160
HARSHIT SATISH	20BCE2407
NIKHIL	20BCE2154
UJJWAL AKASH	20BCE2202
SHASHWAT	19BCE0808

CLASS DIAGRAM



Admin
+id: integer
+name: String
+RetrieveUserList()
+RetrieveCategoryList()
+RetrieveItemsList()
+RetrieveOrderDetails()
+RetrieveOrderedProductDetails()
+MakeShipment()
+ConfirmDelivery()

Web User
+login_id: String{id}
+password: String
+state: UserState
+AuthenticateUser()
+CheckUserName()
+RegisterUser()
+LoginUser()

Customer
+Id: String
+Name: String
+Address: String
+Phone_num: Integer
+BuyProducts()
+ViewProducts()
+MakePayment()
+AddToCart()
+DeleteFromCart()

Chatbot
+id: String
+message: String
+response: String

Account
+id: String{id}
+billing_address: Address
+isclosed: Boolean
+open: Date
+closed: Date
+Invoice()

Shopping Cart
+created: Date
+GetItems()

Payment
+id: String{id}
+paid: Date
+total: Real
+details: String

LineItem
+quantity: integer
+price: Real
+GetCategory()

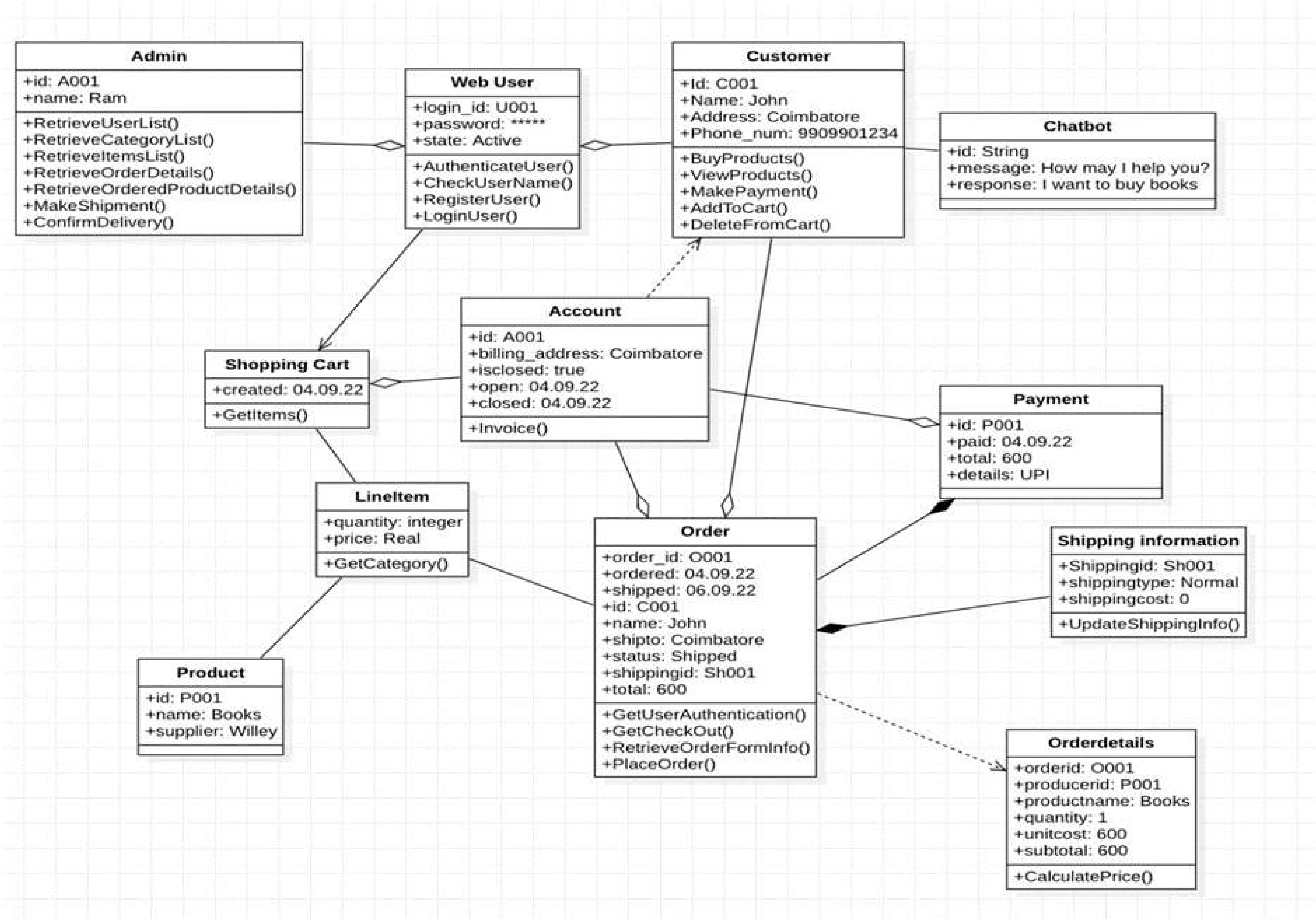
Order
+order_id: String{id}
+ordered: Date
+shipped: Date
+id: String{id}
+name: String
+shipto: Address
+status: String
+shippingid: String
+total: Real
+GetUserAuthentication()
+GetCheckOut()
+RetrieveOrderFormInfo()
+PlaceOrder()

Shipping information
+Shippingid: Sh001
+shippingtype: Normal
+shippingcost: 0
+UpdateShippingInfo()

Product
+id: String{id}
+name: String
+supplier: String

Orderdetails
+orderid: String
+producerid: String
+productname: String
+quantity: integer
+unitcost: Real
+subtotal: Real
+CalculatePrice()

Object Diagram:



Admin
+id: A001
+name: Ram
+RetrieveUserList()
+RetrieveCategoryList()
+RetrieveItemsList()
+RetrieveOrderDetails()
+RetrieveOrderedProductDetails()
+MakeShipment()
+ConfirmDelivery()

Web User
+login_id: U001
+password: *****
+state: Active
+AuthenticateUser()
+CheckUserName()
+RegisterUser()
+LoginUser()

Customer
+Id: C001
+Name: John
+Address: Coimbatore
+Phone_num: 9909901234
+BuyProducts()
+ViewProducts()
+MakePayment()
+AddToCart()
+DeleteFromCart()

Chatbot
+id: String
+message: How may I help you?
+response: I want to buy books

Account
+id: A001
+billing_address: Coimbatore
+isclosed: true
+open: 04.09.22
+closed: 04.09.22
+Invoice()

Shopping Cart
+created: 04.09.22
+GetItems()

Payment
+id: P001
+paid: 04.09.22
+total: 600
+details: UPI

LineItem
+quantity: integer
+price: Real
+GetCategory()

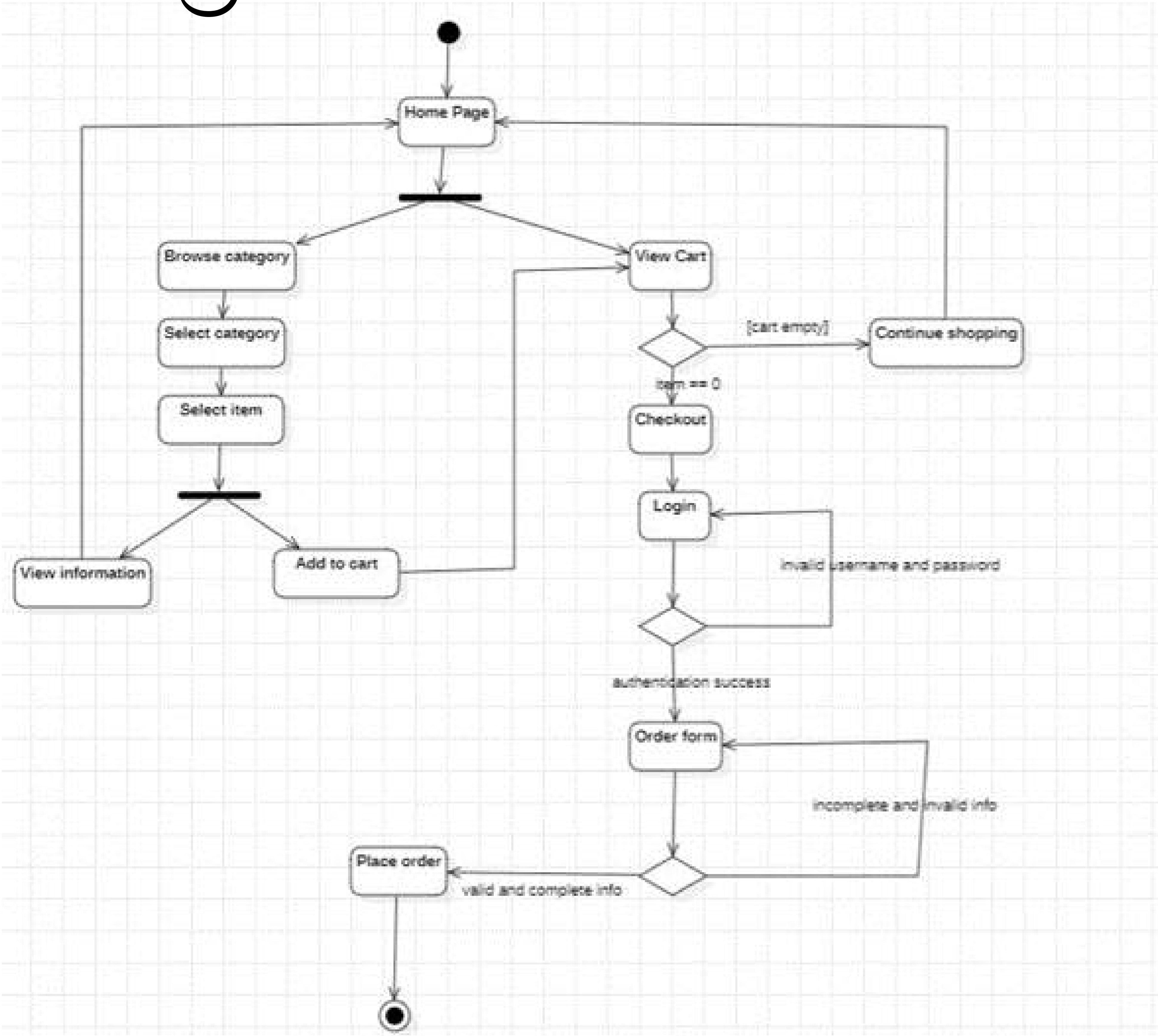
Order
+order_id: O001
+ordered: 04.09.22
+shipped: 06.09.22
+id: C001
+name: John
+shipto: Coimbatore
+status: Shipped
+shippingid: Sh001
+total: 600
+GetUserAuthentication()
+GetCheckOut()
+RetrieveOrderFormInfo()
+PlaceOrder()

Shipping information
+Shippingid: Sh001
+shippingtype: Normal
+shippingcost: 0
+UpdateShippingInfo()

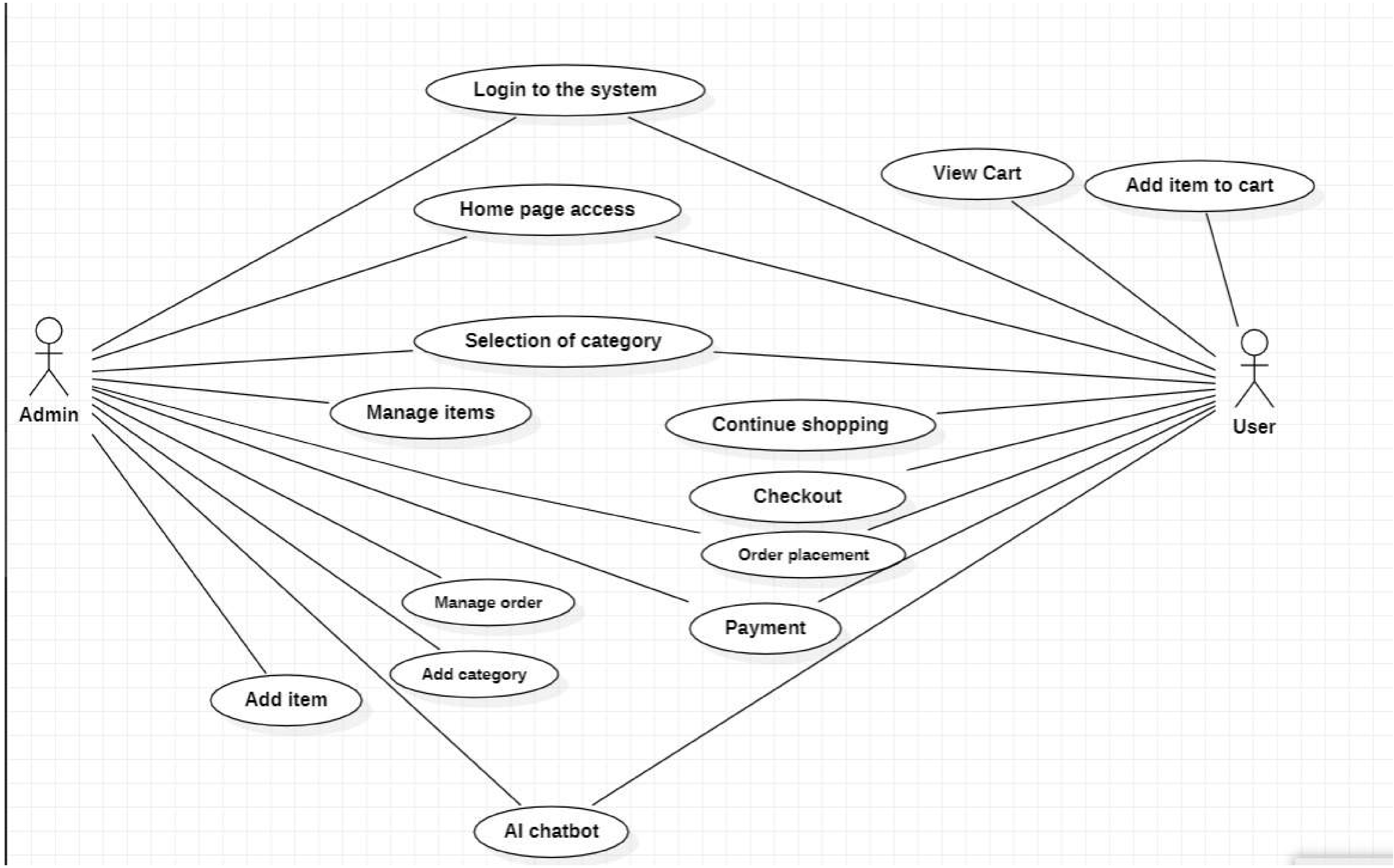
Product
+id: P001
+name: Books
+supplier: Willey

Orderdetails
+orderid: O001
+producerid: P001
+productname: Books
+quantity: 1
+unitcost: 600
+subtotal: 600
+CalculatePrice()

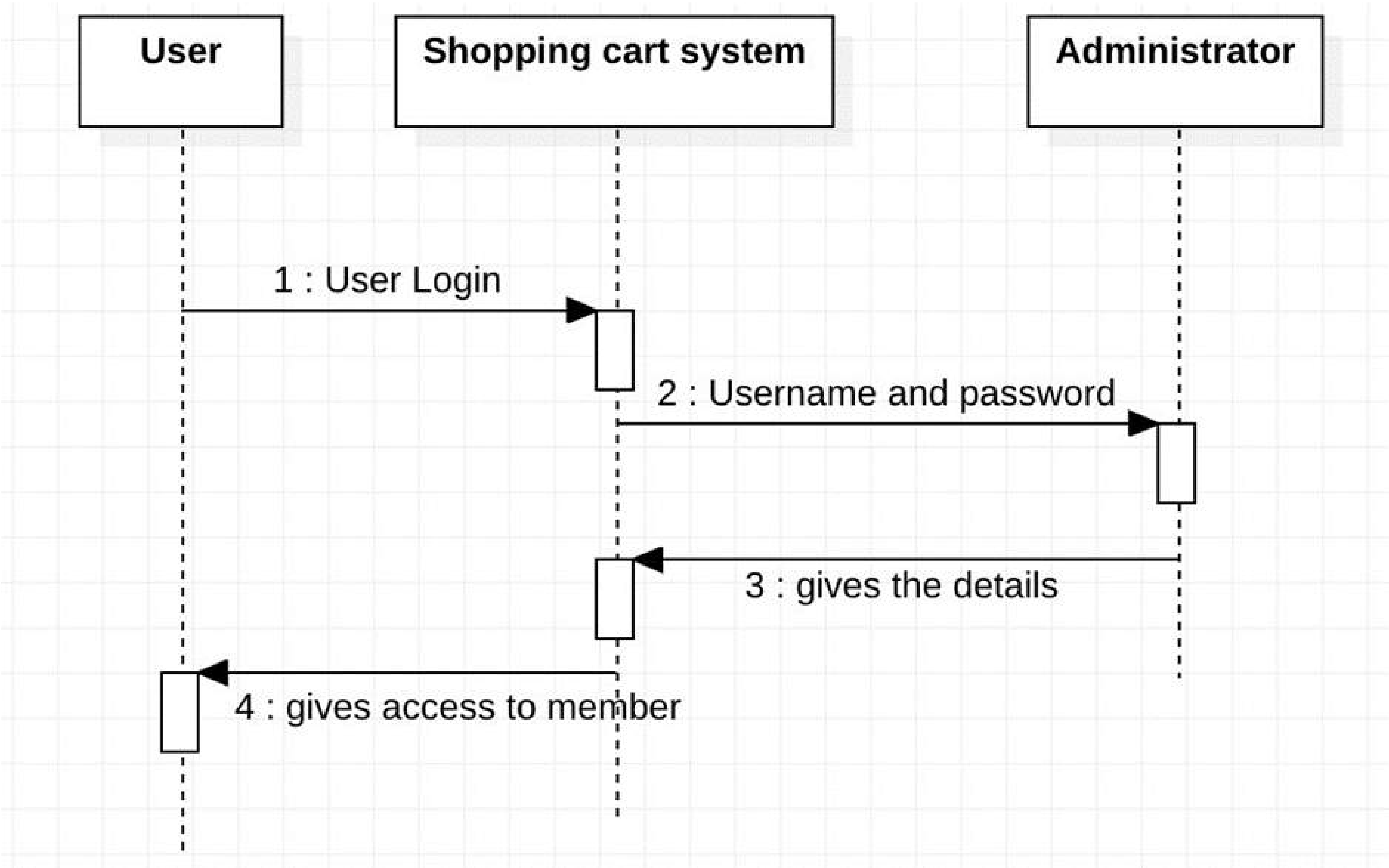
Activity Diagram:



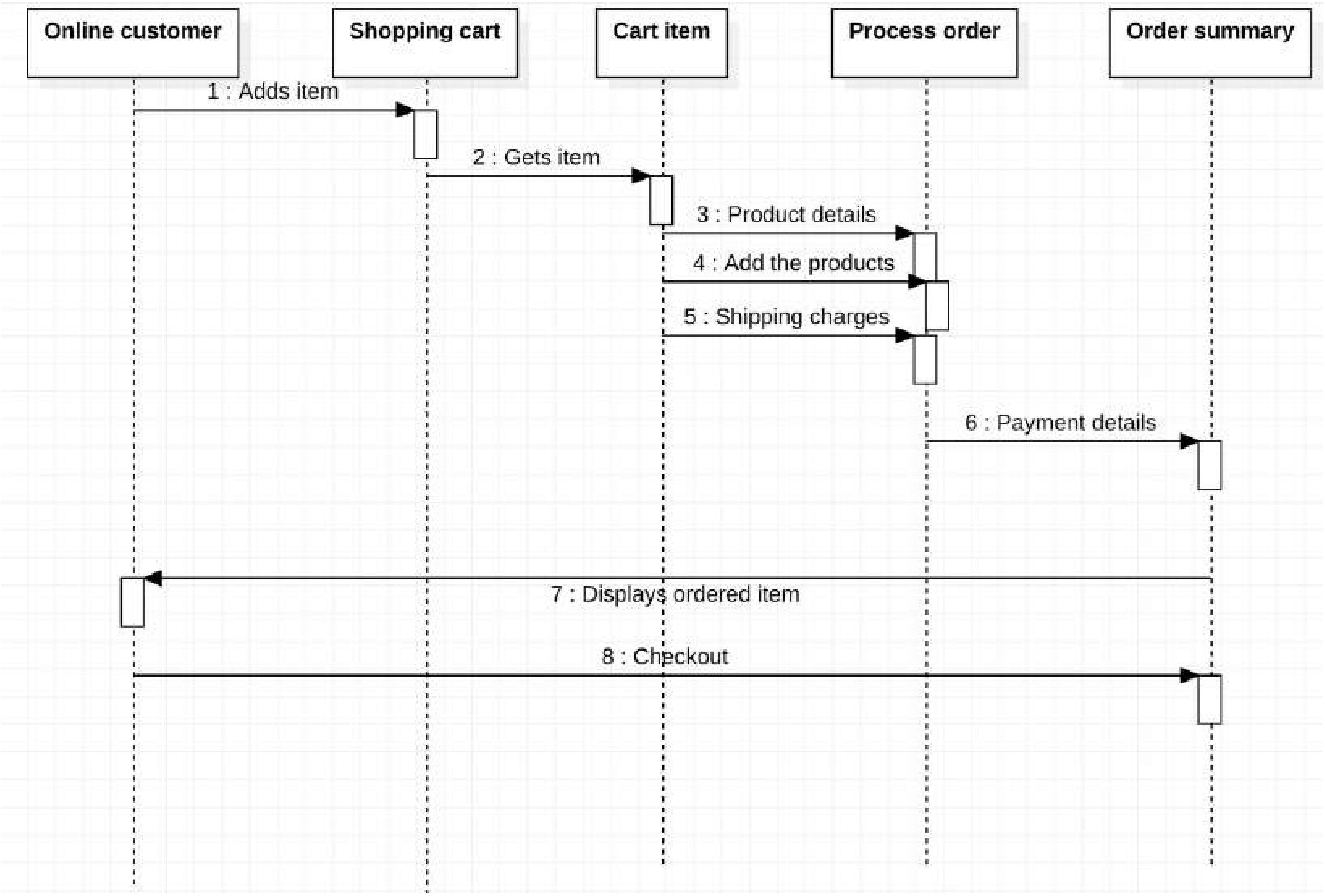
Use case diagram:



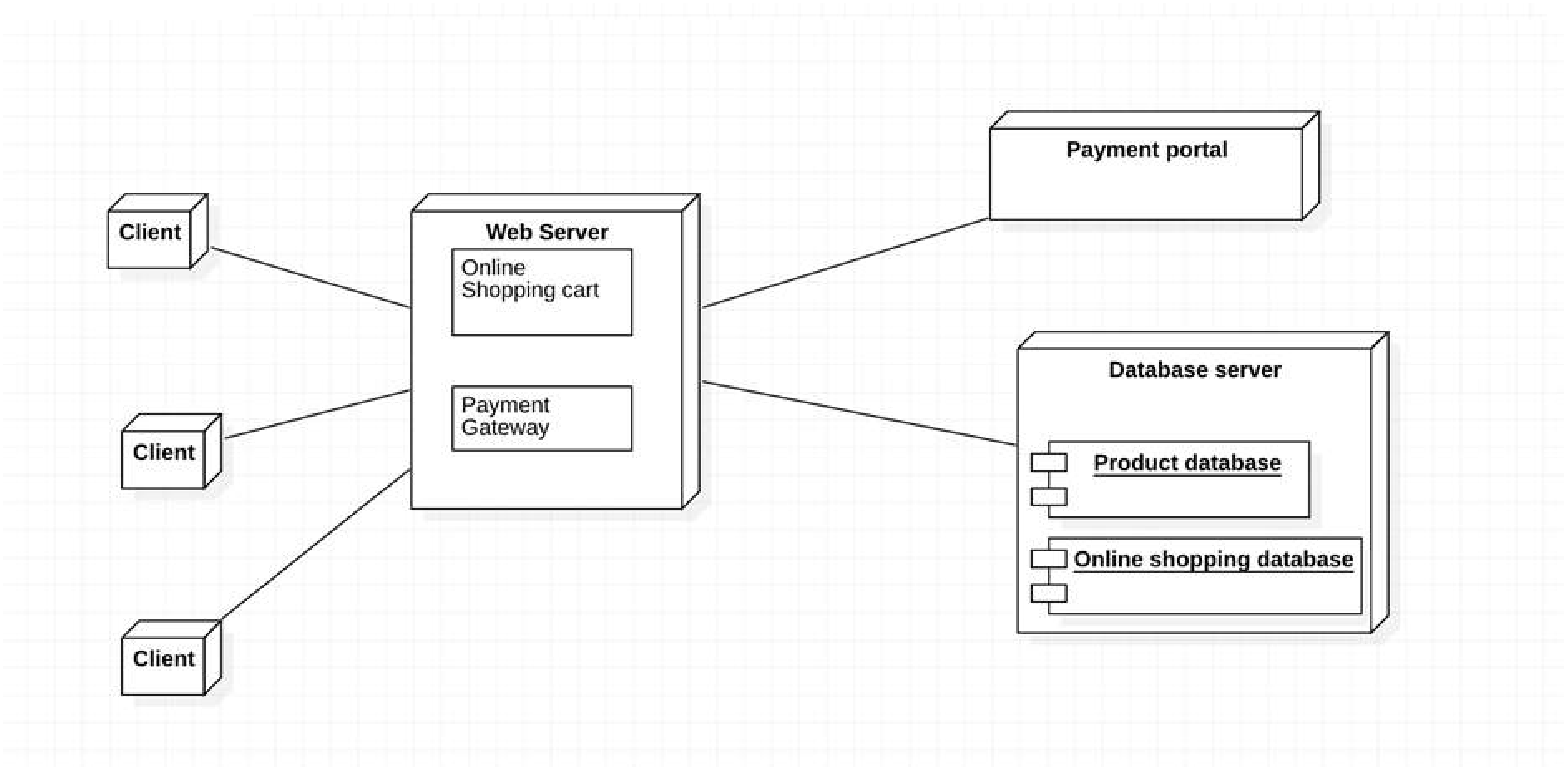
Sequence diagram for user registration



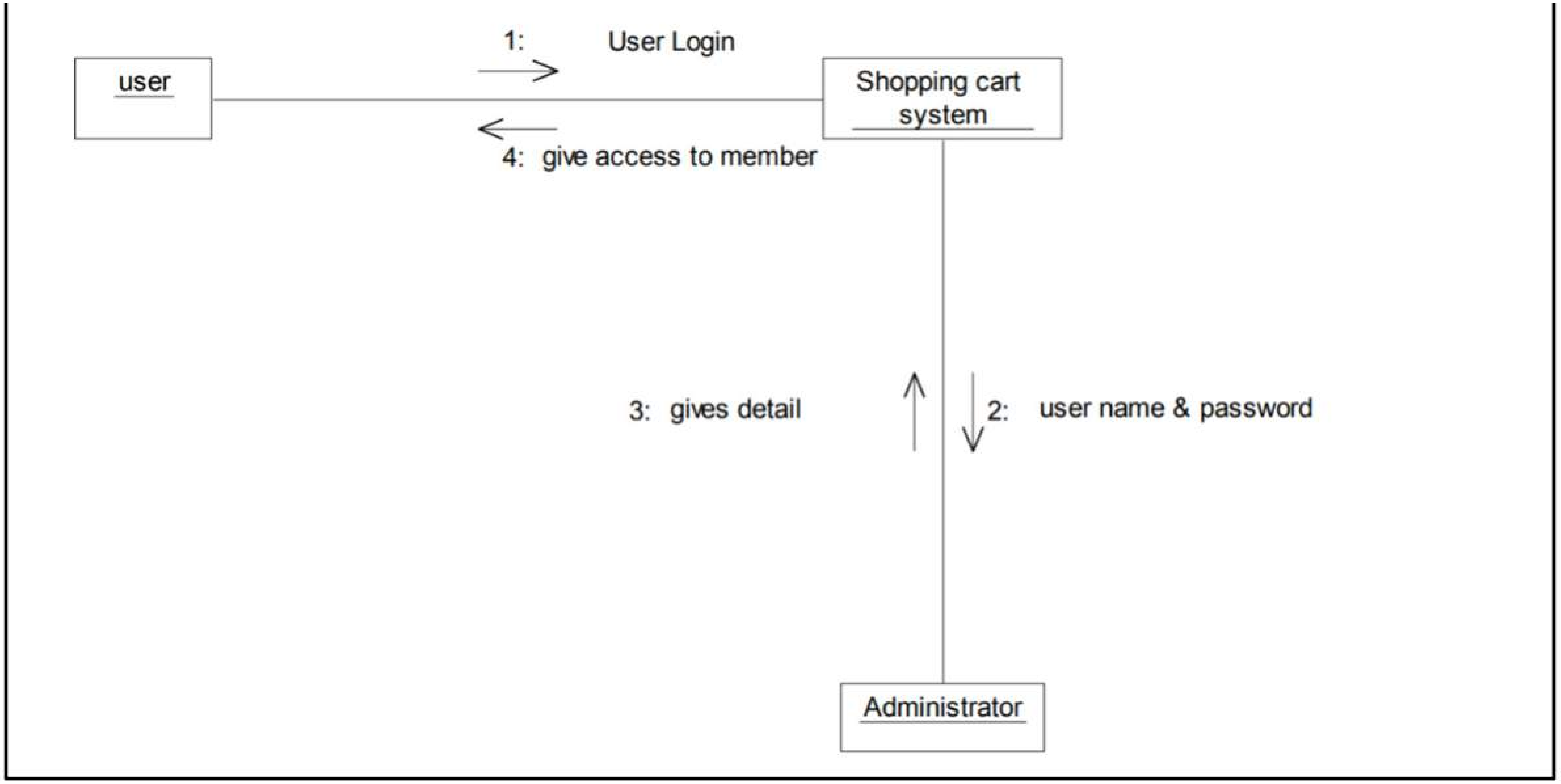
Sequence diagram for customers



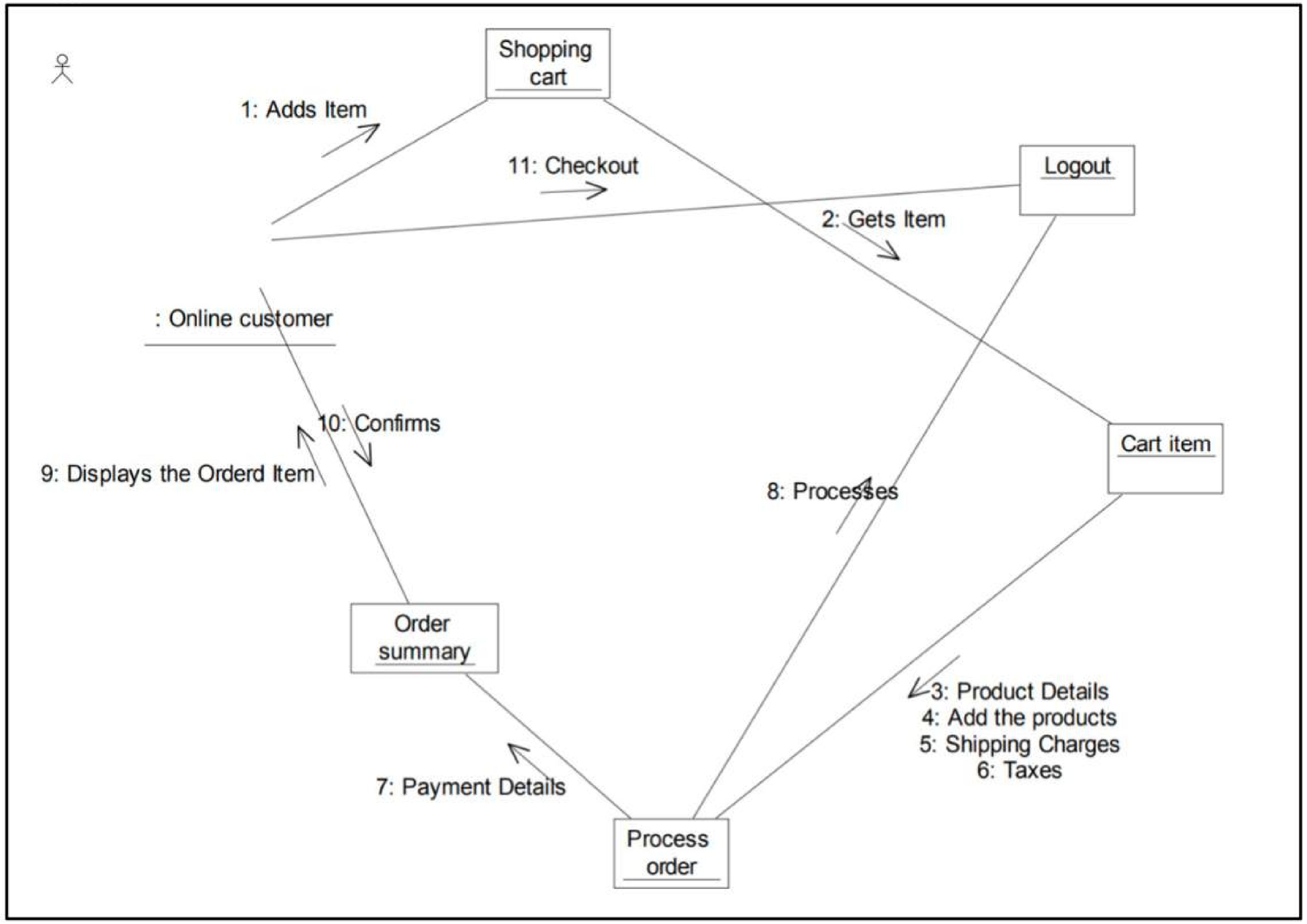
Deployment diagram:



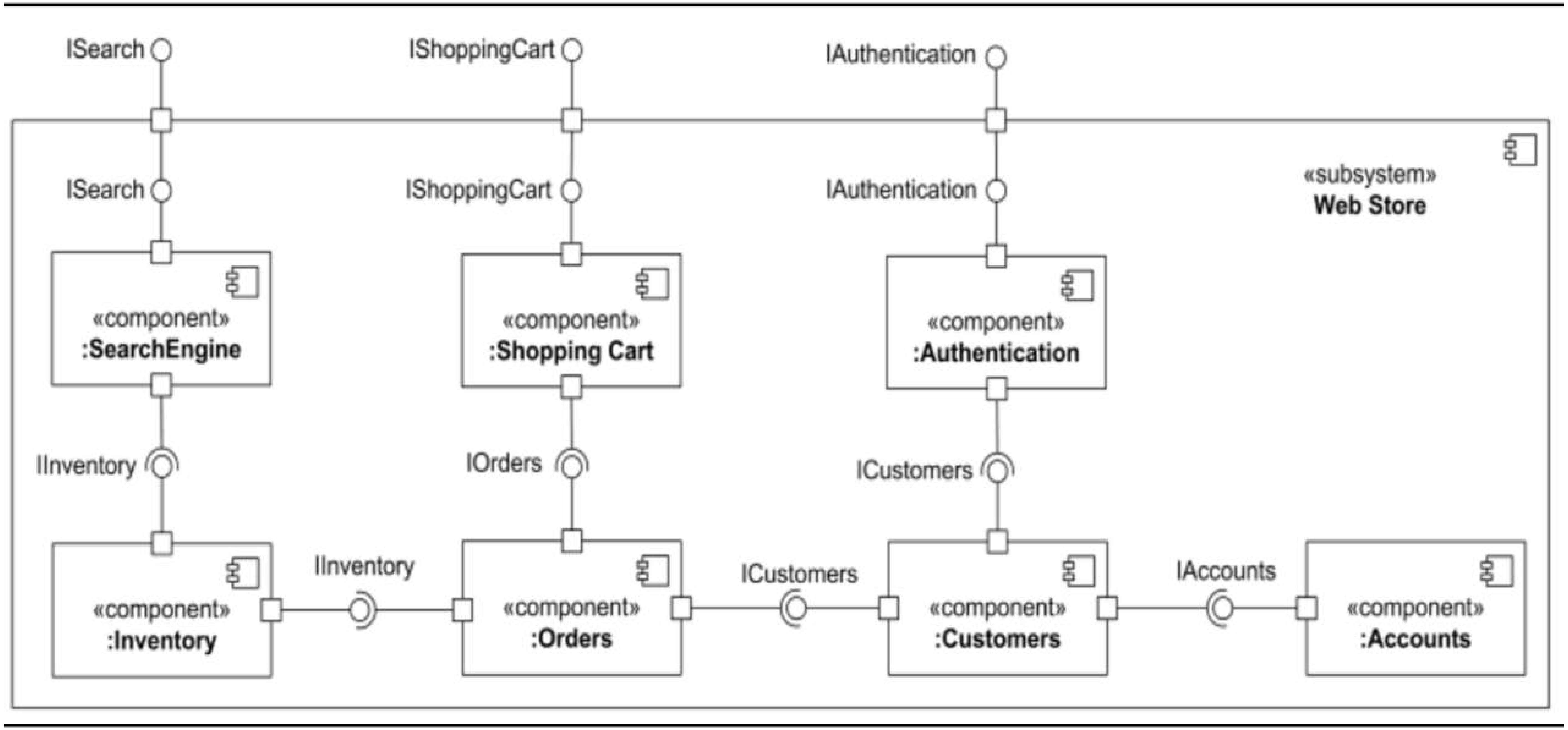
Collaboration diagram for user registration



Collaboration diagram for customers



Component diagram



Thank you!

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

AI Based E-Commerce Web Application with Integrated Chat-Bot

GROUP MEMBERS

C.VIPIN-20BCE2160

HARSHIT SATISH-20BCE2407

NIKHIL-20BCE2154

UJJWAL AKASH-20BCE2202

SHASHWAT-19BCE0808

1. Based on the reviews, what did you feel were useful and what did you wish you had known?

Based on the reviews we got from the trial run of the web application, users have felt that the GUI was very user friendly and attractive. It was easier for them to navigate between different sections of the application. The chatbot was very useful and eased the burden of the users as it eliminated the waiting time to get their queries resolved. We felt that the user interface of the chatbot could be improved as most of the users were actively interacting with it.

2. **Reflect on your experiences, did you feel like you had a better grasp or developing software to solve a given problem? Why or why not?**

Although we've done minor projects earlier, this was something that most of us hadn't ventured into as it had many modules that had to be integrated. Chatbot especially took a lot of time as we had to train the bot by giving various questions and answers. Eventually , when everyone had a grasp of its working, it became easier for us to develop it.

3. How did you end up using the software development tools you used? Did you change the way you used the tools throughout the phase of the project?

All of us had prior knowledge about the front end development tools such as HTML, CSS and JavaScript. Backend development was not known by everyone in the team, so we decided to use PHP and MySQL for the development. We didn't use any of the frameworks and advanced tools such as Node.js, React.js, Angular.js etc.

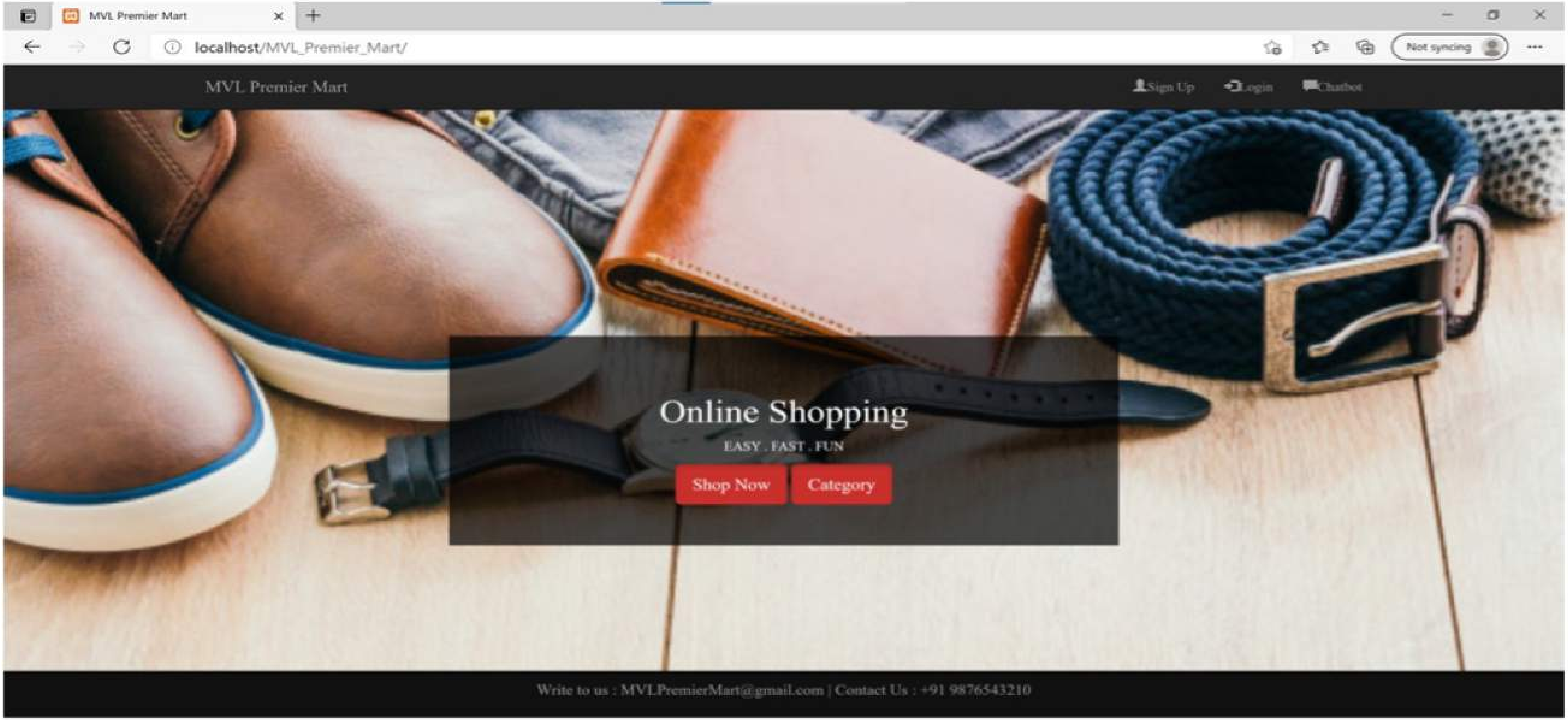
4.Did you change the way you worked, the requirements, etc. while working on the project? Why?

We learnt how to more efficiently work as a team. It is said "Alone we can do so little; together we can do so much." That is indeed true. It was everyone's contribution that made it possible for us to reach towards the completion of the project

5. Did you have conflicts with team members? How did you work to resolve them?

We didn't have any major conflicts but in the initial days, there were a few a squabbles. It did took a while for us to understand each other and the way they operate. So once everyone got to know each other well, it was easier to work on the professional front.

Product demonstration

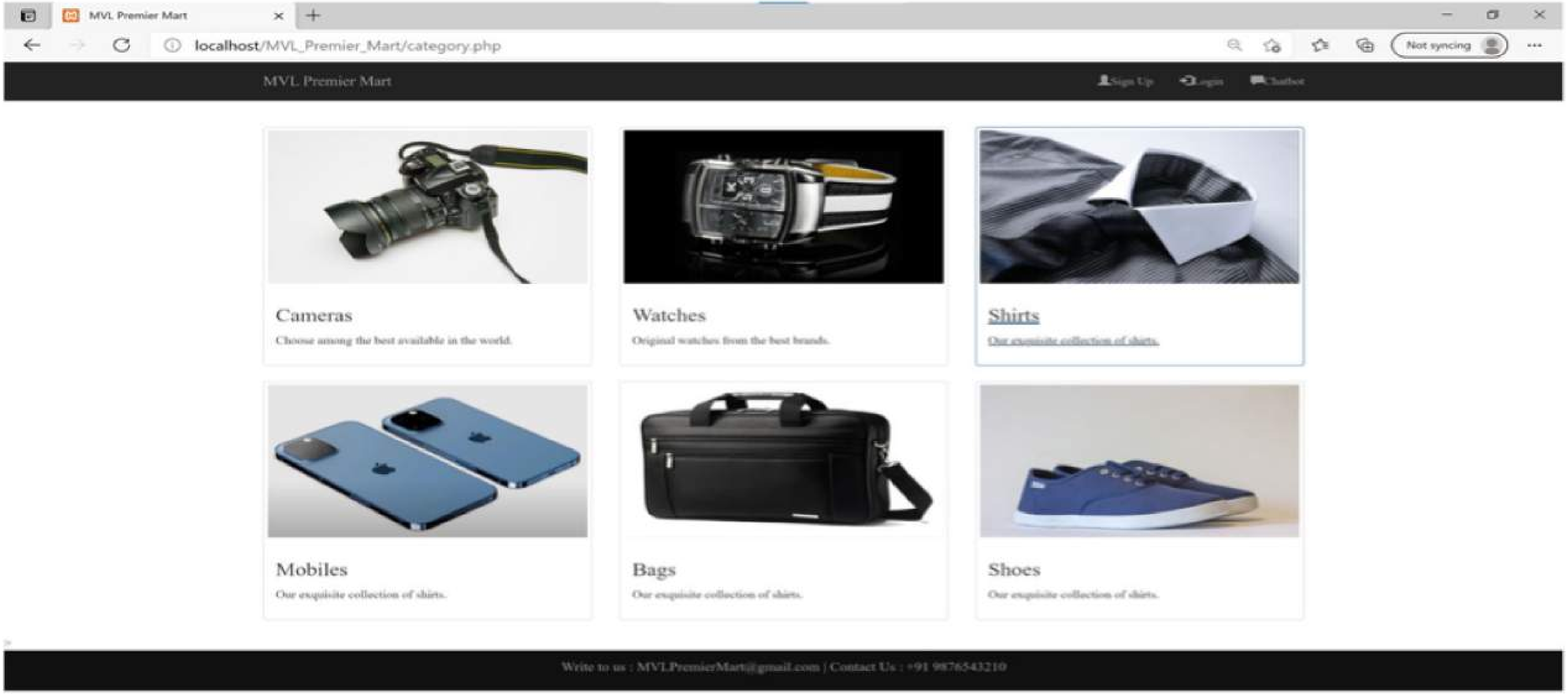


Online Shopping

EASY . FAST . FUN

Write to us : MVLPremierMart@gmail.com | Contact Us : +91 9876543210

Homepage



Cameras

Choose among the best available in the world.

Watches

Original watches from the best brands.

Shirts

Our exquisite collection of shirts.

Mobiles

Our exquisite collection of shirts.

Bags

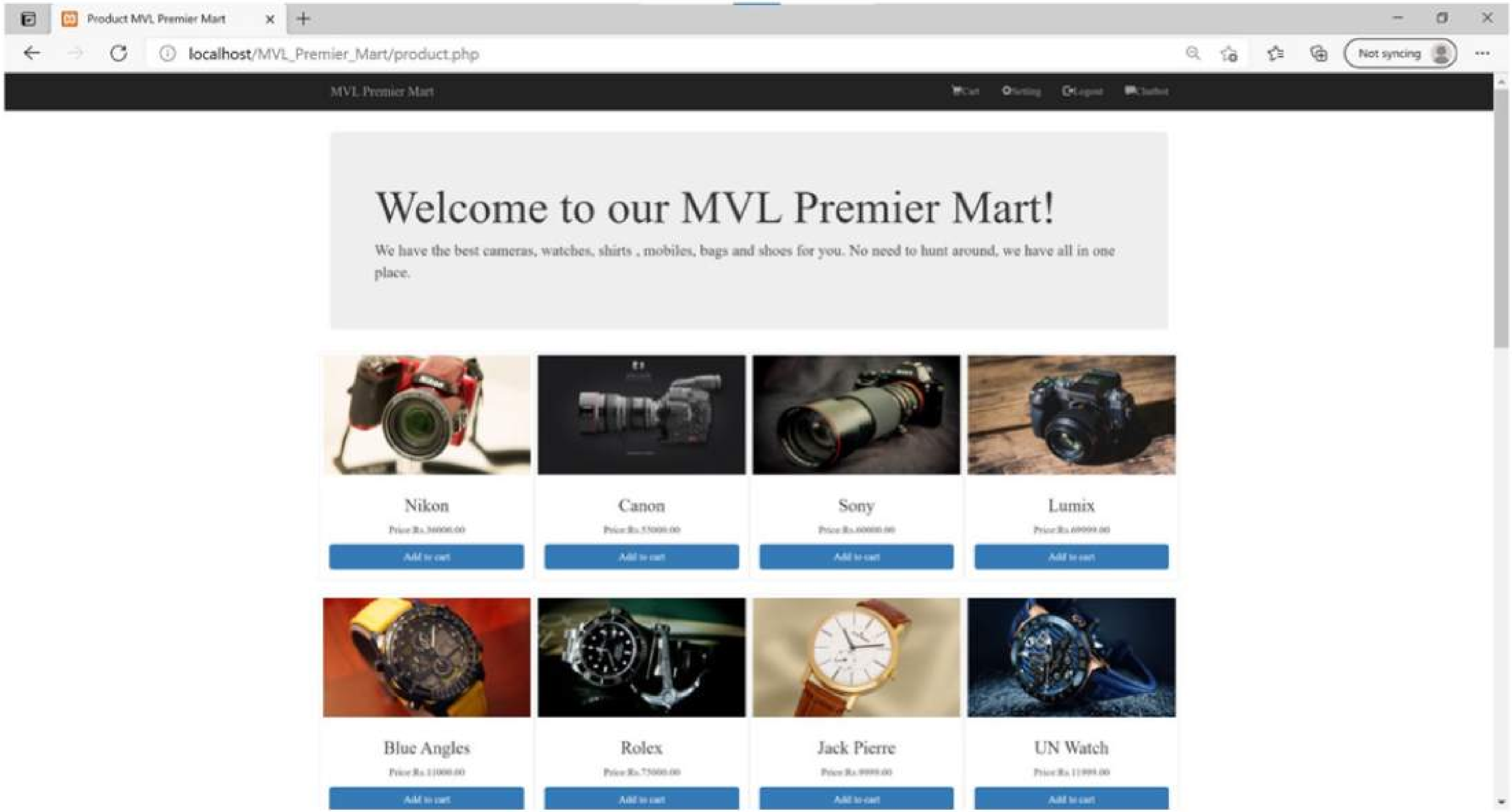
Our exquisite collection of shirts.

Shoes

Our exquisite collection of shirts.

Write to us : MVLPremierMart@gmail.com | Contact Us : +91 9876543210

Category page



Welcome to our MVL Premier Mart!

We have the best cameras, watches, shirts , mobiles, bags and shoes for you. No need to hunt around, we have all in one place.

Nikon

Price:Rs.36000.00

Add to cart

Canon

Price:Rs.55000.00

Add to cart

Sony

Price:Rs.60000.00

Add to cart

Lumix

Price:Rs.69999.00

Add to cart

Blue Angles

Price:Rs.11000.00

Add to cart

Rolex

Price:Rs.75000.00

Add to cart

Jack Pierre

Price:Rs.9999.00

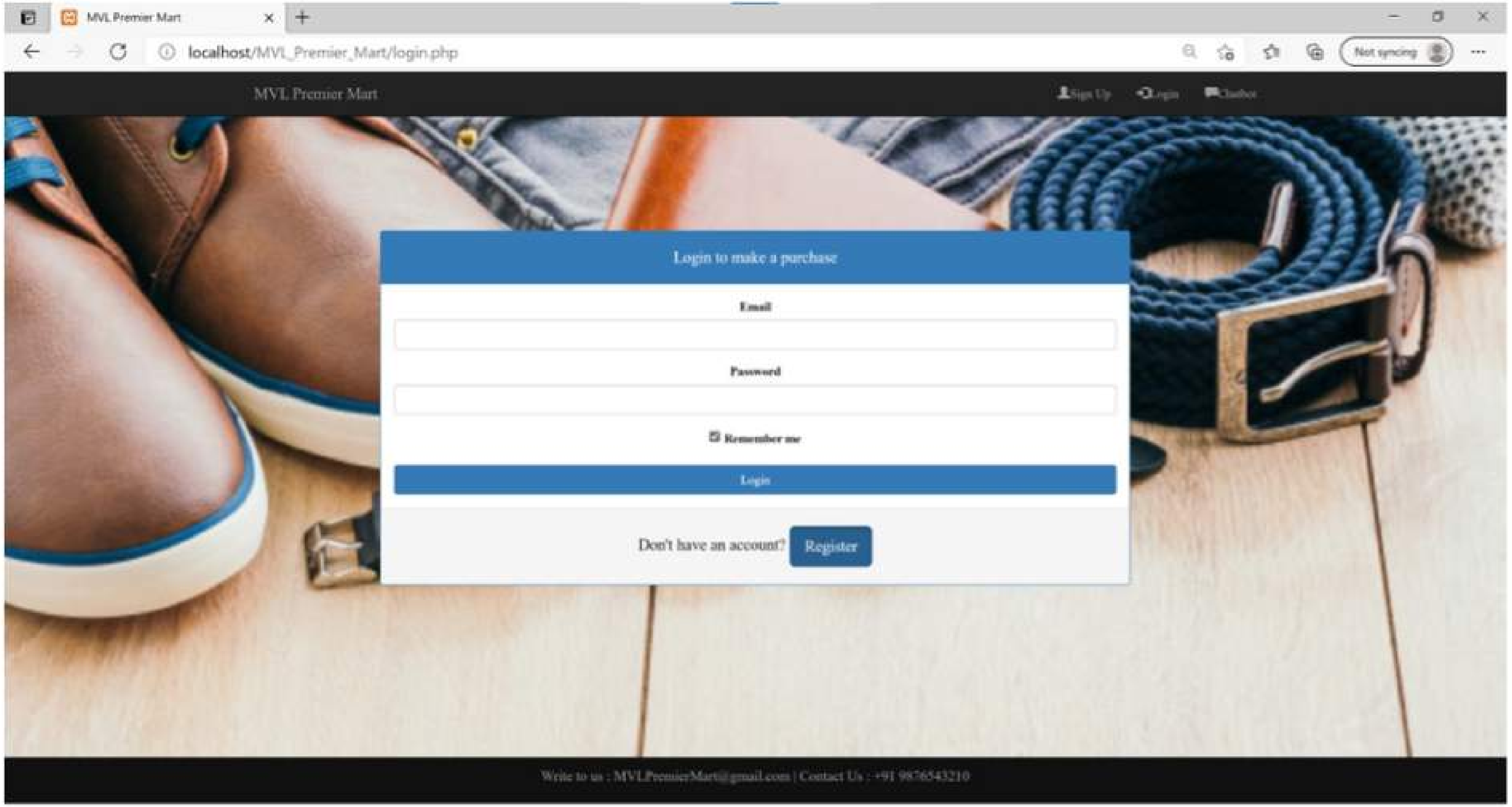
Add to cart

UN Watch

Price:Rs.11999.00

Add to cart

Product page



Login to make a purchase

Email

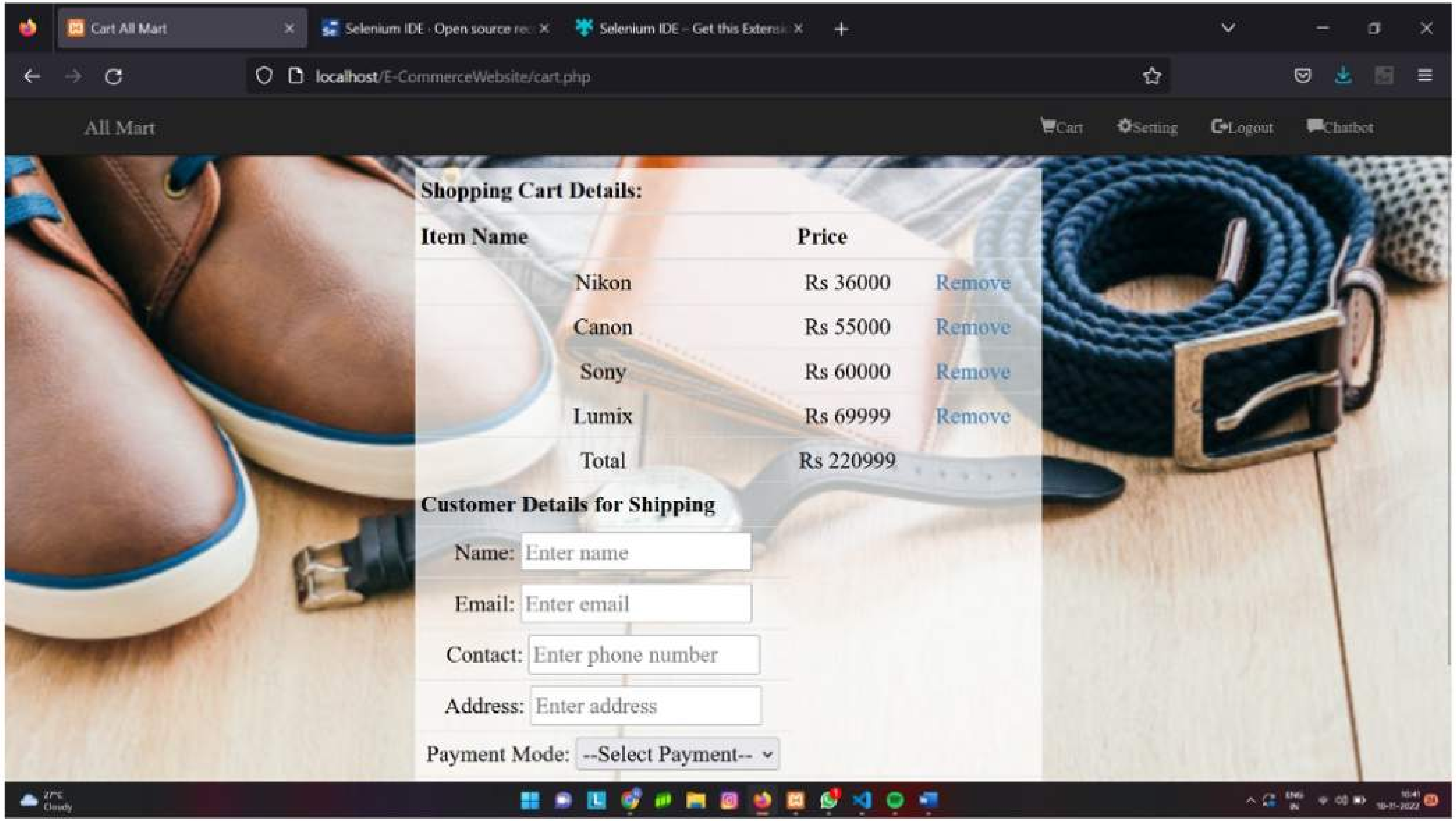
Password

☑ Remember me

Don't have an account?

Write to us : MVLPremierMart@gmail.com | Contact Us : +91 9876543210

Login page



Cart All Mart
Selenium IDE · Open source re
Selenium IDE – Get this Extensi
localhost/E-CommerceWebsite/cart.php
All Mart
Cart
Setting
Logout
Chatbot

Shopping Cart Details:

Item Name	Price	
Nikon	Rs 36000	Remove
Canon	Rs 55000	Remove
Sony	Rs 60000	Remove
Lumix	Rs 69999	Remove
Total	Rs 220999	

Customer Details for Shipping

Name: Enter name

Email: Enter email

Contact: Enter phone number

Address: Enter address

Payment Mode: --Select Payment--

Cart page