SHOPPING MART

A PROJECT REPORT

Submitted By:

ANKIT KUMR SINGH
(University Roll No - 2000290140019)
RUDRA GAHLOT
(University Roll No - 20002901400106)

Submitted in partial fulfillment of the Requirements for the Degree of

MASTER OF COMPUTER APPLICATIONS

Under the Supervision of
Mr.Rabi.N Panda
Associate Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

(JUNE 2022)

DECLARATION

I hereby declare that the work presented in this report entitled “SHOPPING MART", was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources. I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name : Ankit Kumar Singh

Roll. No. : 2000290140019

Branch : MCA

(Candidate Signature)

CERTIFICATE

Certified that Ankit Kumar S**ingh University Roll No - 20002901400198, Rudra Gahlot University Roll No – 20002901400106** have carried out the project work having "**HARDWARE STORE"** for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself / herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date: 13 JAN 2022

Ankit Kumar Singh (2000290140019)

Rudra Gahlot (2000290140088)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Mr.Rabi.N Panda
Associat Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner **Signature of External Examiner**

Dr. Ajay Shrivastava
Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

Hardware Store is process of doing business through computer networks. A person sitting on his chair in front of a computer can access all the facilities of the Internet to buy or sell the products. Unlike traditional commerce that is carried out physically with effort of a person to go & get products, ecommerce has made it easier for human to reduce physical work and to save time. E-Commerce which was started in early 1990’s has taken a great leap in the world of computers, but the fact that has hindered the growth of e-commerce is security. Security is the challenge facing ecommerce today & there is still a lot of advancement made in the field of security. The main advantage of e-commerce over traditional commerce is the user can browse online shops, compare prices and order merchandise sitting at home on their PC. For increasing the use of e-commerce in developing countries the B2B e-commerce is implemented for improving access to global markets for firms in developing countries. For a developing country advancement in the field of e-commerce is essential. The research strategy shows the importance of the e-commerce in developing countries for business applications. The main objective of the E-commerce Website is to manage the details of Products,Customer,Shipping,Payment,Category. It manages all the information about Products, Sales, Category, Products. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Products Customer, Sales, Shipping. It tracks all the details about the Shipping,Payment,Category

ACKNOWLEDGEMENT

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, **Ankit Verma Sir** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Ajay Kumar Shrivastava, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Ankit Kumar Singh

Rudra Gahlot

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Customer get many benefits via online shopping this helps e-commerce companies to build long lasting and profitable relationship with their customers. For making strong relationship with these users it is very important to focus on the customer as a whole and making sense of a flood of real time information that goes well beyond demographics or shopping behaviour. There are two entities who will have the access to the system. One is the admin and another one will be the registered user.

Admin can add product details, view all the order details and can also view the sales of the products. User need to register with basic registration details to generate a valid username and password. After the user logins, it can view all the products that are recommended on the homepage compiled by the system based on user's information. From the recommended products, the user can even further view its details and then if interested to buy, the system gives add to cart option for purchasing the product. The system even has an AI bot with the help of which the user can get answers to queries like features, warranty, price etc. details of the products. This AI Bot even converts text to speech. After selecting the product, user can do payment for the particular product online. Users can view their order history of their purchased product.

1.2 AIM

The main aim of HARDWARE STORE development is to sell products to users. The most successful websites are carefully optimized to achieve a high percentage of purchases. To achieve success ecommerce websites need to integrate all of the latest online closing & upsell techniques available which have been proven to increase chances that a visitor will purchase. There are many important elements that go into building a successful e-commerce website such as removing friction during the purchasing process, making the checkout smooth and easy, making the website fast and attractive, up selling users on related products, incentivizing buyers, reducing cart abandonment, nurturing past buyers to buy again, remarketing to past visitors who haven't yet purchased, using the proper payment options, having a mobile ready design and many more things which are needed to develop and e-commerce website.

1.3 EXISTING SYSTEM

This existing system of buying goods has several disadvantages. It requires lots of time to travel to the particular shop to buy the goods. It is having lots of manual work. Since everyone is leading busy life now a days, time means a lot to everyone. Also there are expenses for travelling from house to shop. It is less user-friendly. In current system user must go to shop and order products. It is difficult to identify the required product. More over the shop from where we would like to buy something may not be open 24*7*365. Hence we have to adjust our time with the shopkeeper's time or vendor's time. In current e-commerce system user have to go shop to view the description of the product. It is unable to generate different kinds of report.

1.4 PROPOSED SYSTEM

The proposed system helps in building a website to buy, sell products or goods online using internet connection. Unlike traditional commerce that is carried out physically with effort of a person to go and get products, eCommerce has made it easier for human to reduce physical work and to save time. The basic concept of the application is to allow the customer to shop virtually using the Internet and allow customers to buy the items and articles of their desire from the store.E-commerce is fast gaining ground as an accepted and used business paradigm.

1.5 HARDWARE & SOFTWARE REQUIREMENTS SPECIFICATION

1.5.1 Hardware Requirements

Number	**Description**
1	PC with 250 GB or more Hard disk.
2	PC with 2 GB RAM.
3	PC with Pentium 1 and above.

1.5.2 Software Requirements

Number	**Description**	**Type**
1	Operating System	Windows
2	Language	JavaScript , HTML
3	Database	MongoDB
4	IDE	VS Code
5	Browser	Chrome, Firefox, Edge

CHAPTER 2

FEASIBILITY STUDY

2.1 FEASIBILITY STUDY

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it’s worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analyzed carefully. There are 3 parts in feasibility study.

1. Technical Feasibility
2. Economical Feasibility
3. Operational Feasibility

2.2 OPERATIONAL FEASIBILITY

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes. To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviours are to be realised. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

2.3 TECHNICAL FEASIBILITY

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on outline design of system requirements in terms of input, processes, output, fields, programs and procedures. This can be qualified in terms of volume of data, trends, frequency of updating inorder to give an introduction to the technical system. The application is the fact that it has been developed on windows XP platform and a high configuration of 1GB RAM on Intel Pentium Dual core processor. This is technically feasible .The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

2.4 ECONOMICAL FEASIBILITY

Establishing the cost-effectiveness of the proposed system i.e. if the benefits do not outweigh the costs then it is not worth going ahead. In the fast paced world today there is a great need of online social networking facilities. Thus the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.

CHAPTER 3

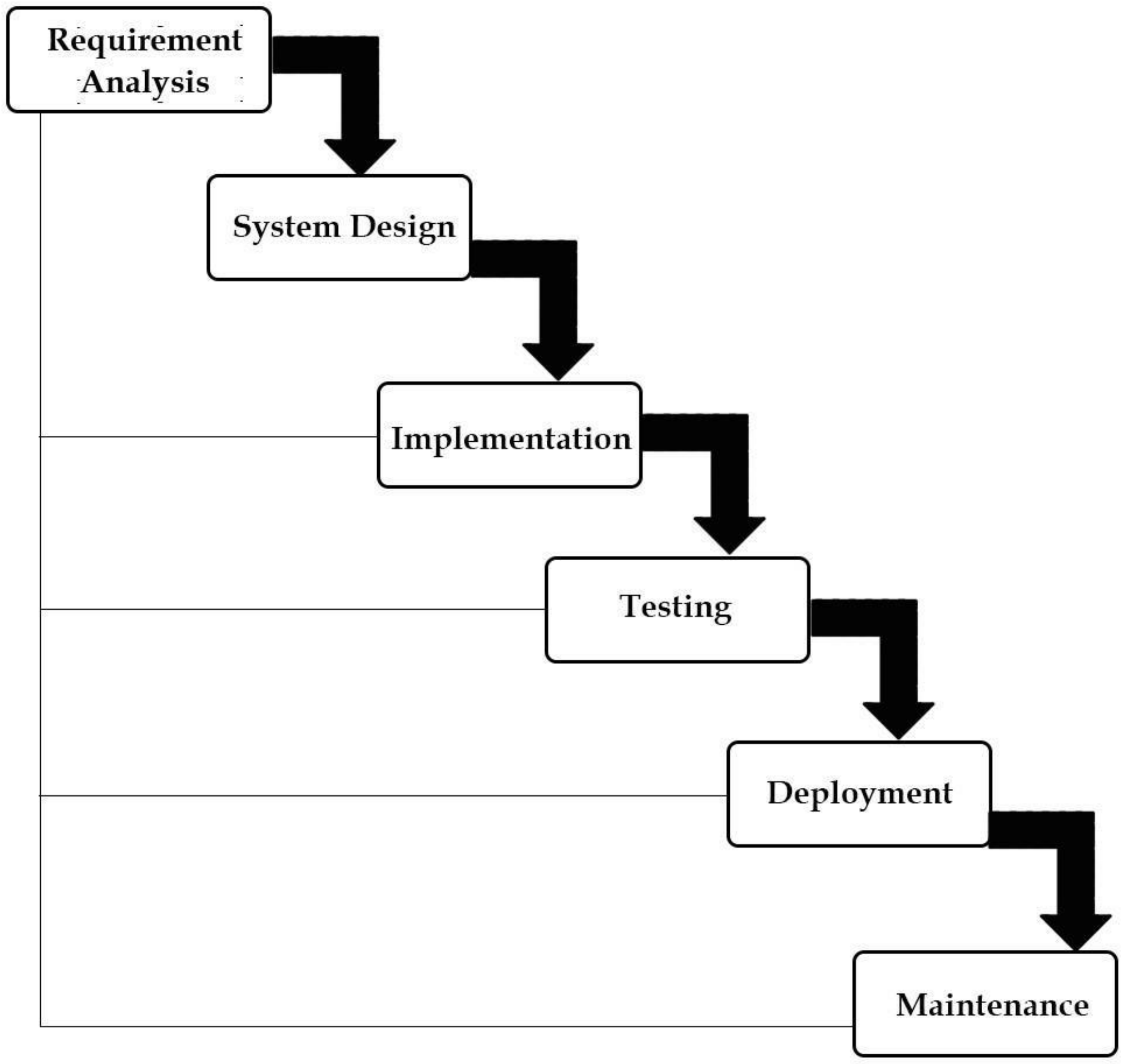
DESIGN & PLANNING

3.1 SOFTWARE DEVELOPMENT LIFE CYCLE MODEL

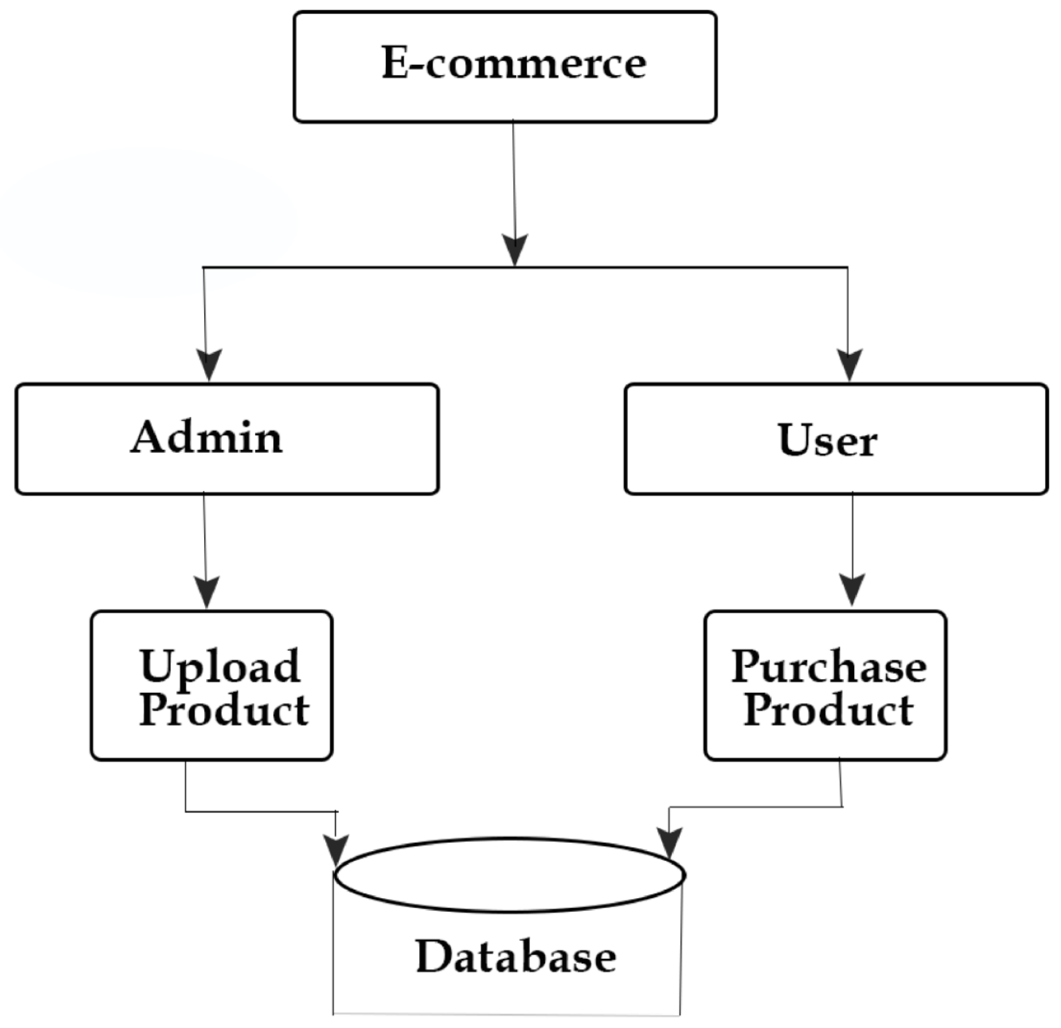
3.1.1 WATERFALL MODEL

The waterfall model was selected as the SDLC model due to the following reasons:

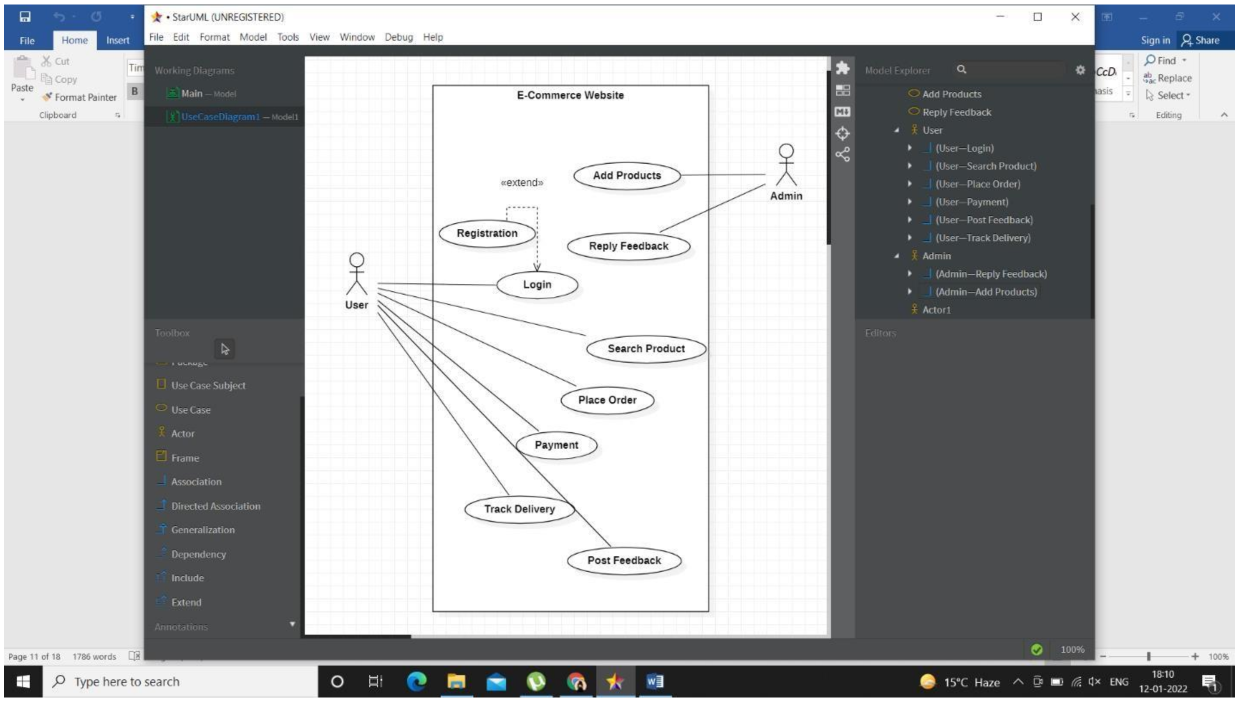
- Technology was adequately understood.
- Simple and easy to understand and use.
- There were no ambiguous requirements.
- Easy to manage due to the rigidity of the model.
- Each phase has specific deliverables and a review process.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.


Requirement Analysis
System Design
Implementation
Testing
Deployment
Maintenance

3.2 GENERAL OVERVIEW

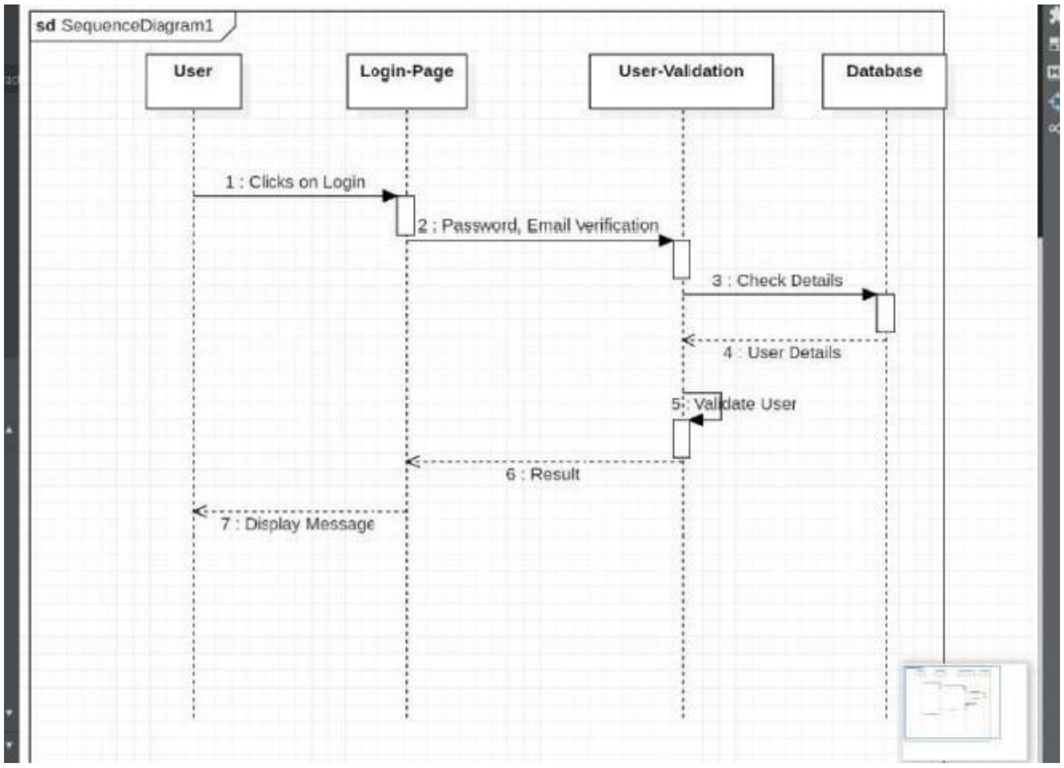


3.3 USE CASE DIAGRAM

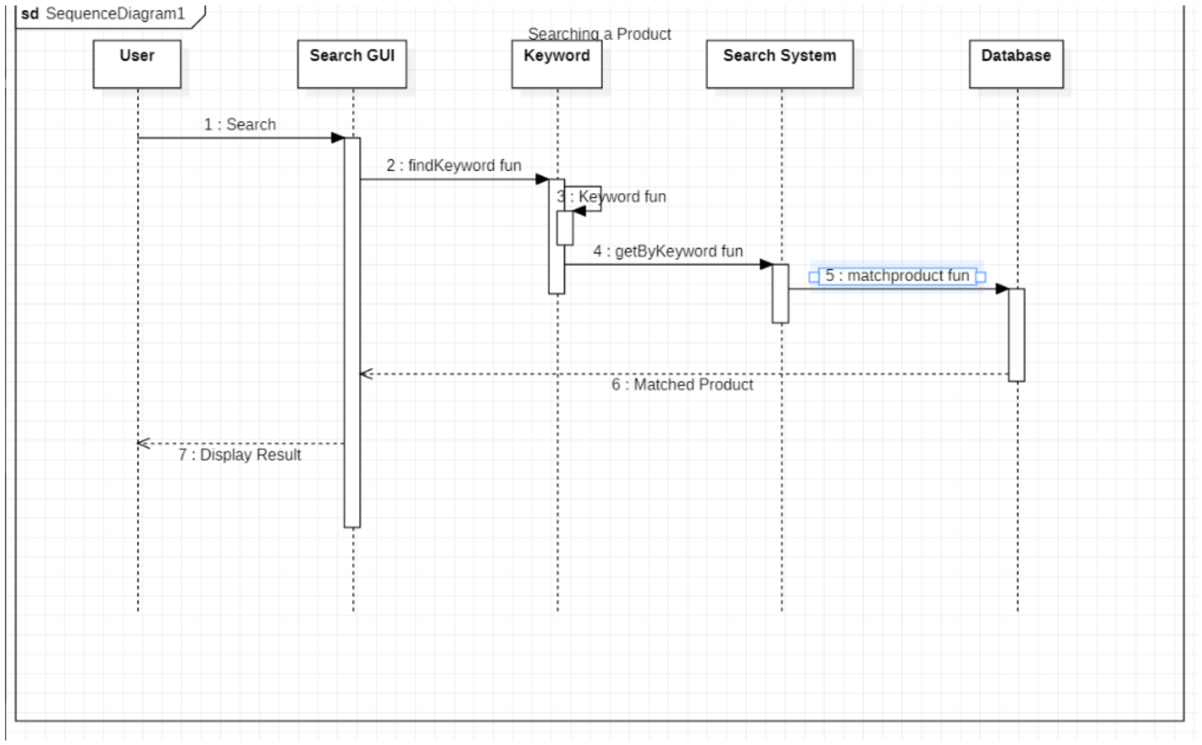


3.4 SEQUENCE DIAGRAM

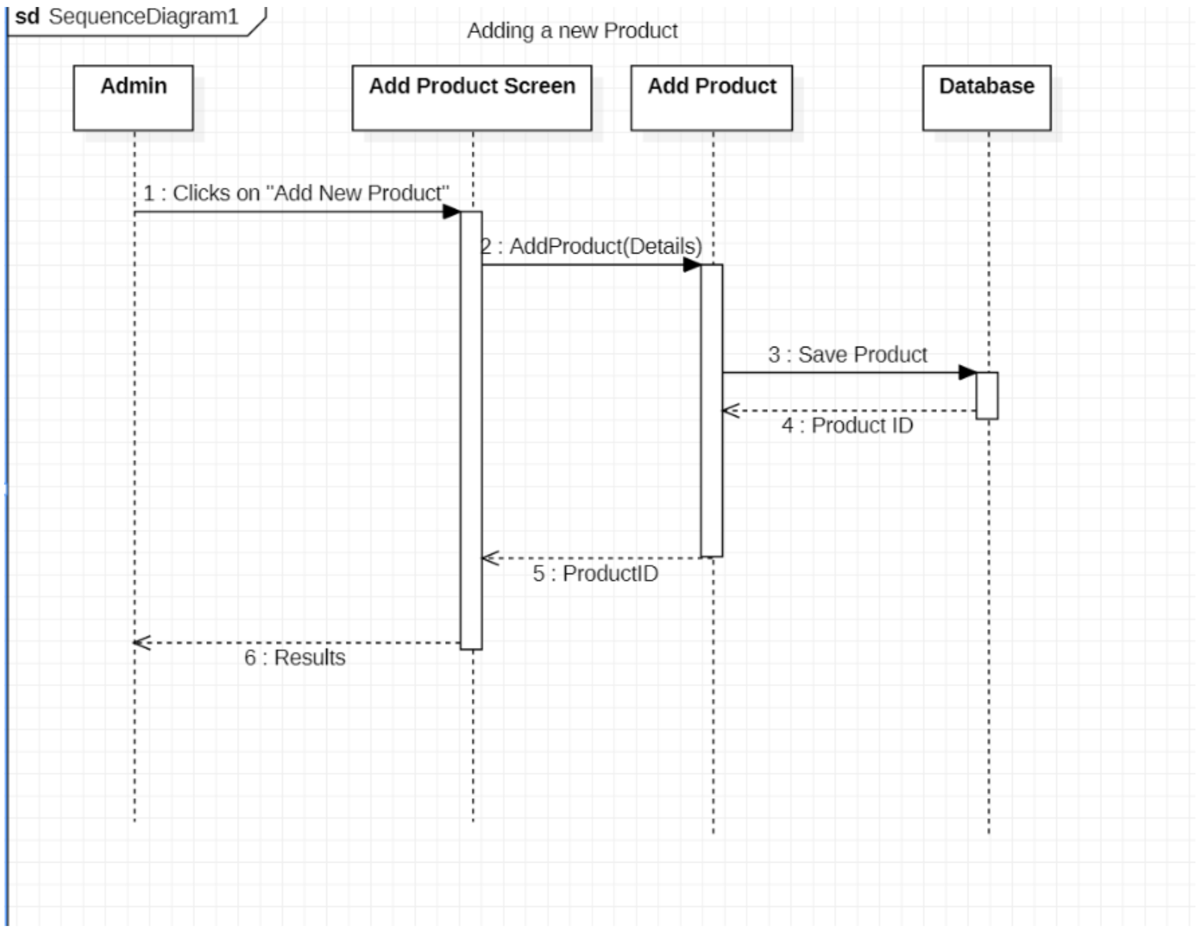
3.4.1 Login



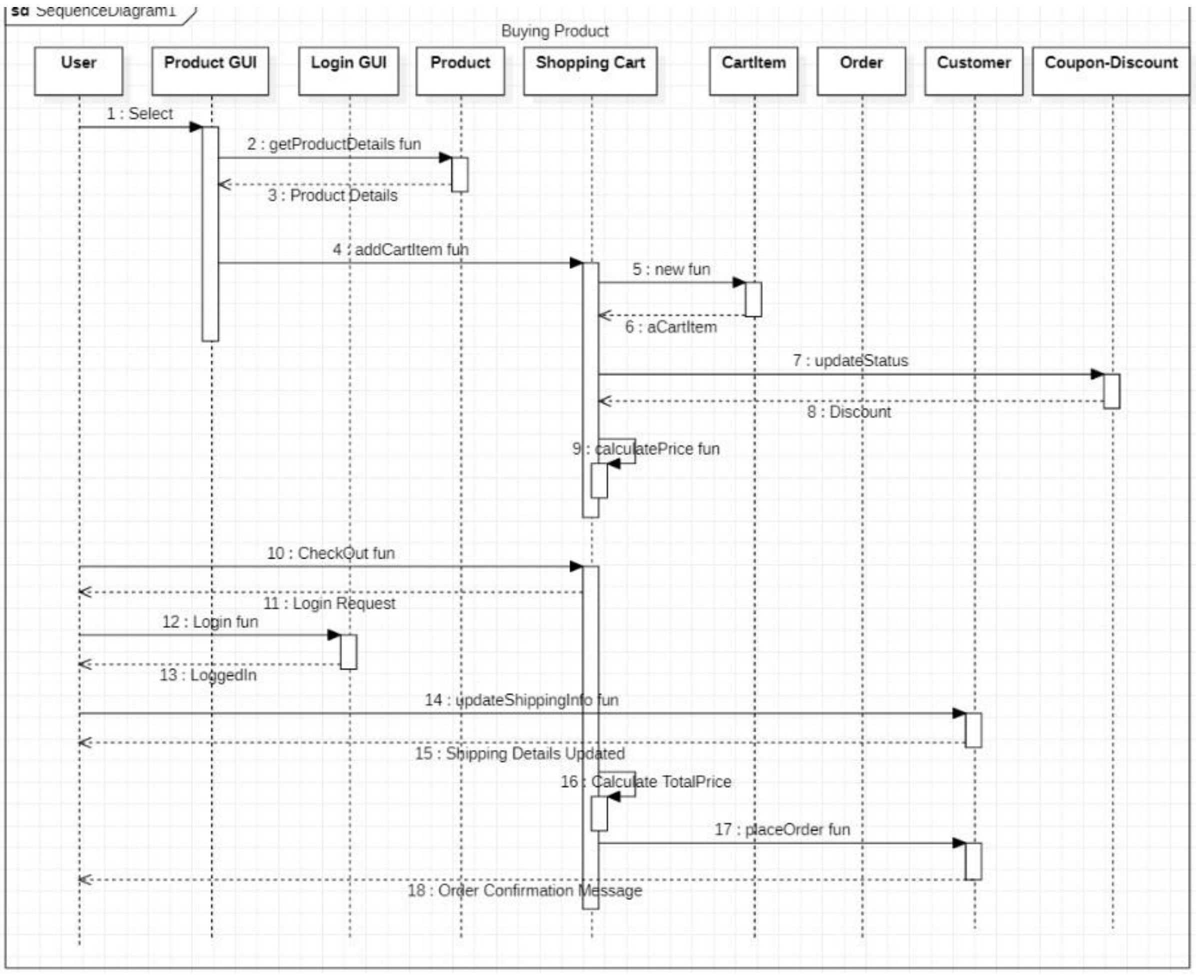
3.4.2 Search


sd SequenceDiagram1
Searching a Product
User
Search GUI
Keyword
Search System
Database
1 : Search
2 : findKeyword fun
3 : Keyword fun
4 : getByKeyword fun
5 : matchproduct fun
6 : Matched Product
7 : Display Result

3.4.3 Add New Product

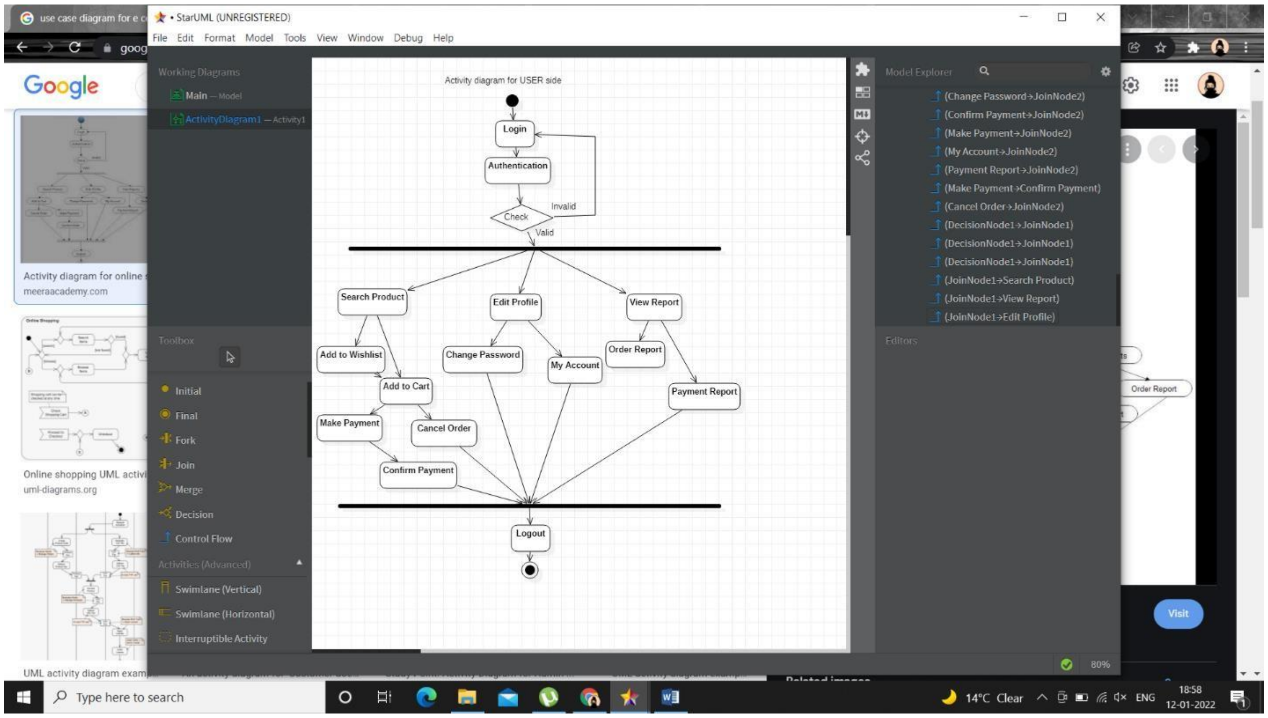


3.4.4 Buy Product

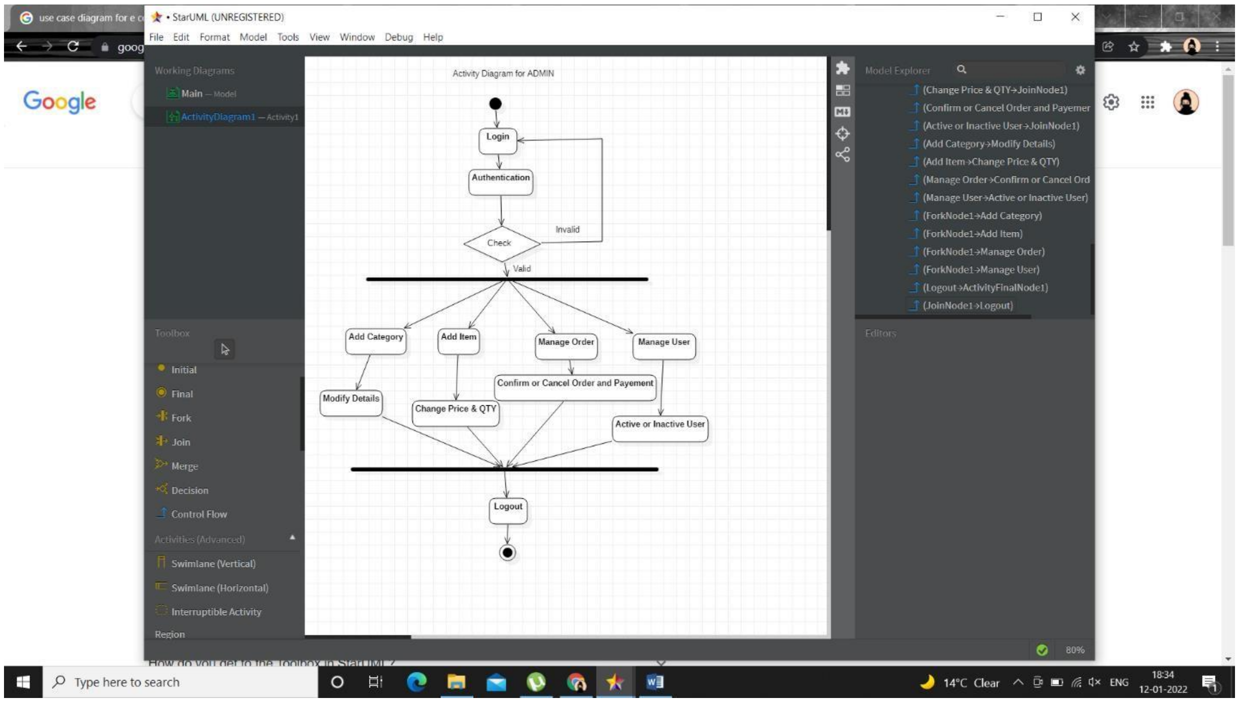


3.5 ACTIVITY DIAGRAM

3.5.1 User Side

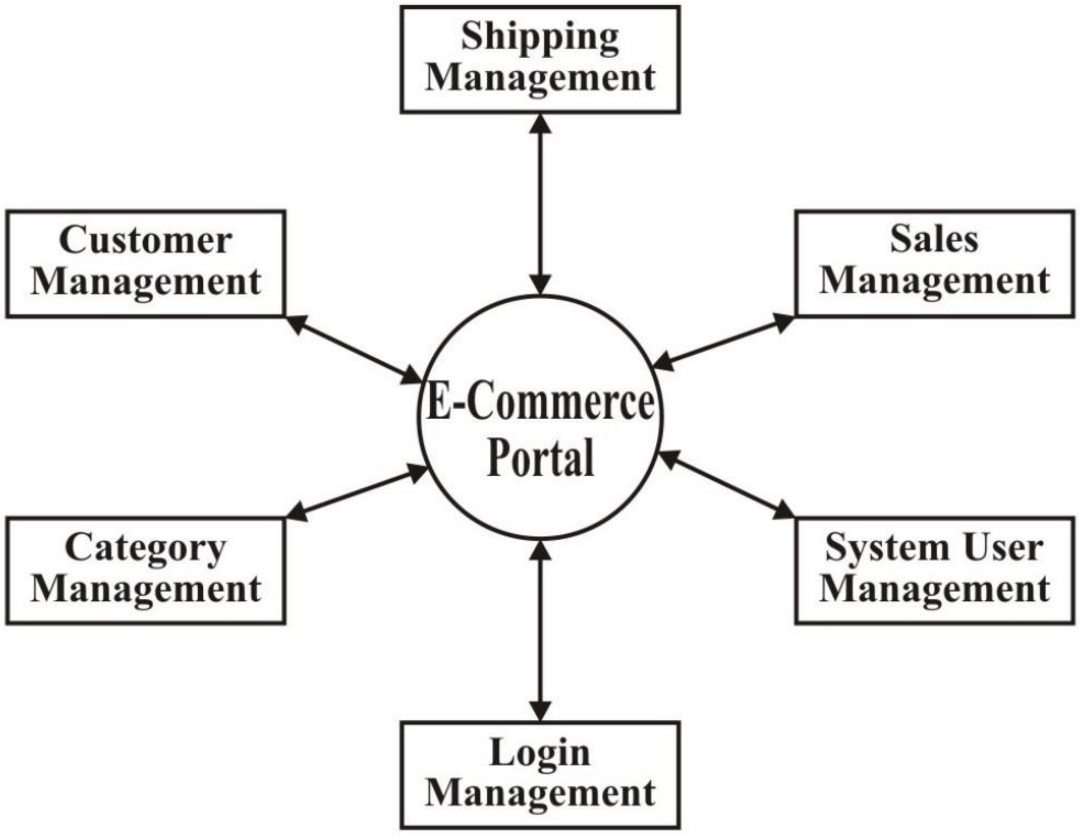


3.5.2 For Admin

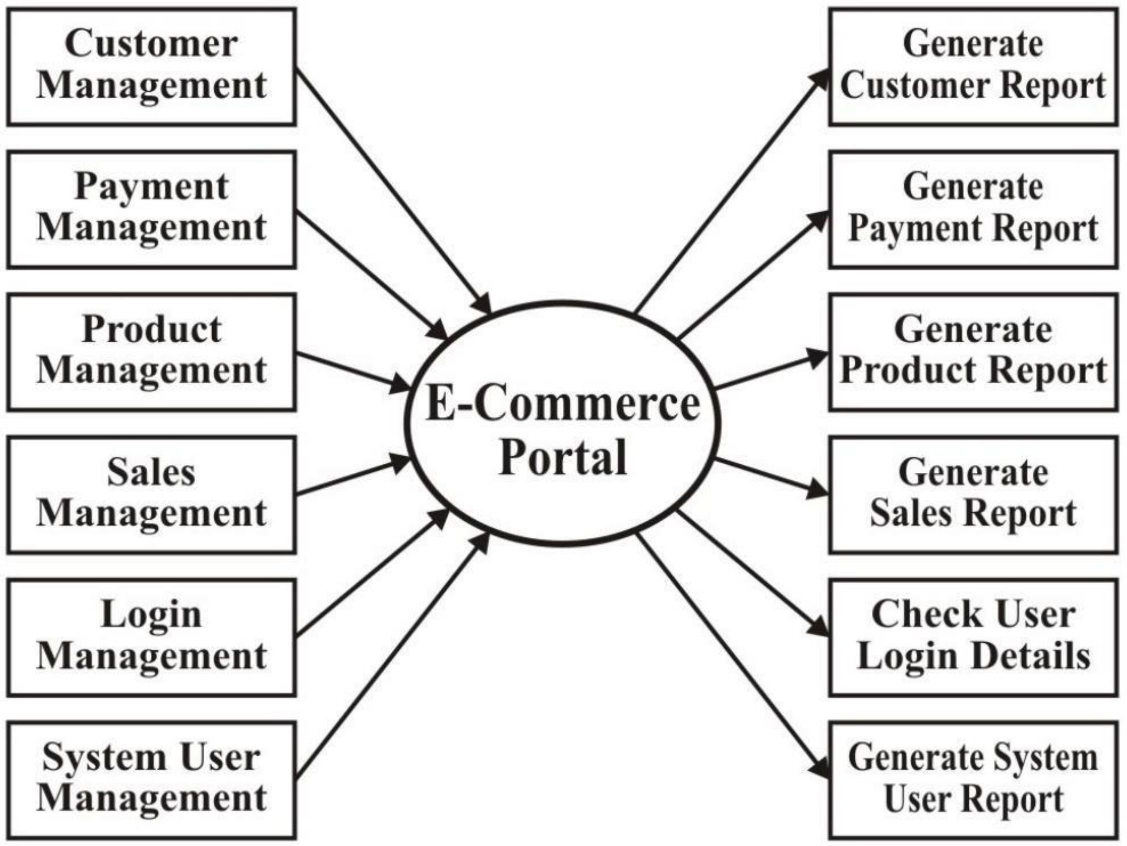


3.6 DFD

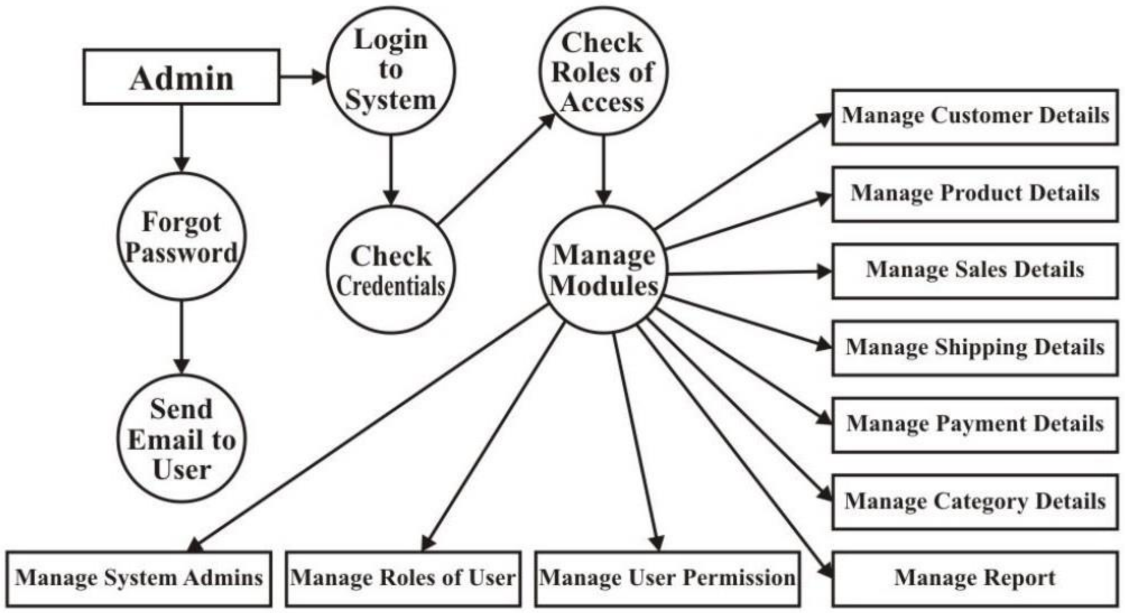
3.6.1 Zero-Level DFD Diagram



3.6.2 First-Level DFD Diagram



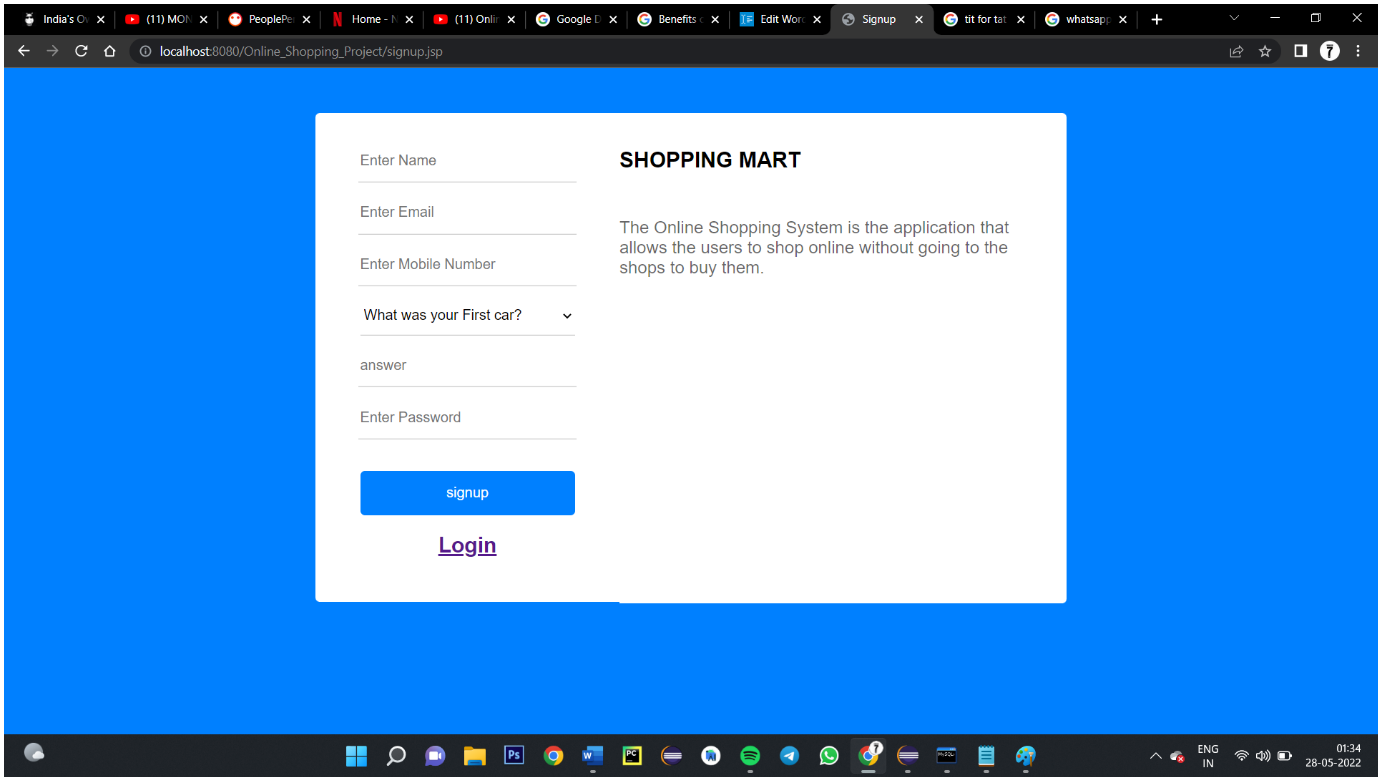
3.6.3 Second-Level DFD Diagram



CHAPTER 4

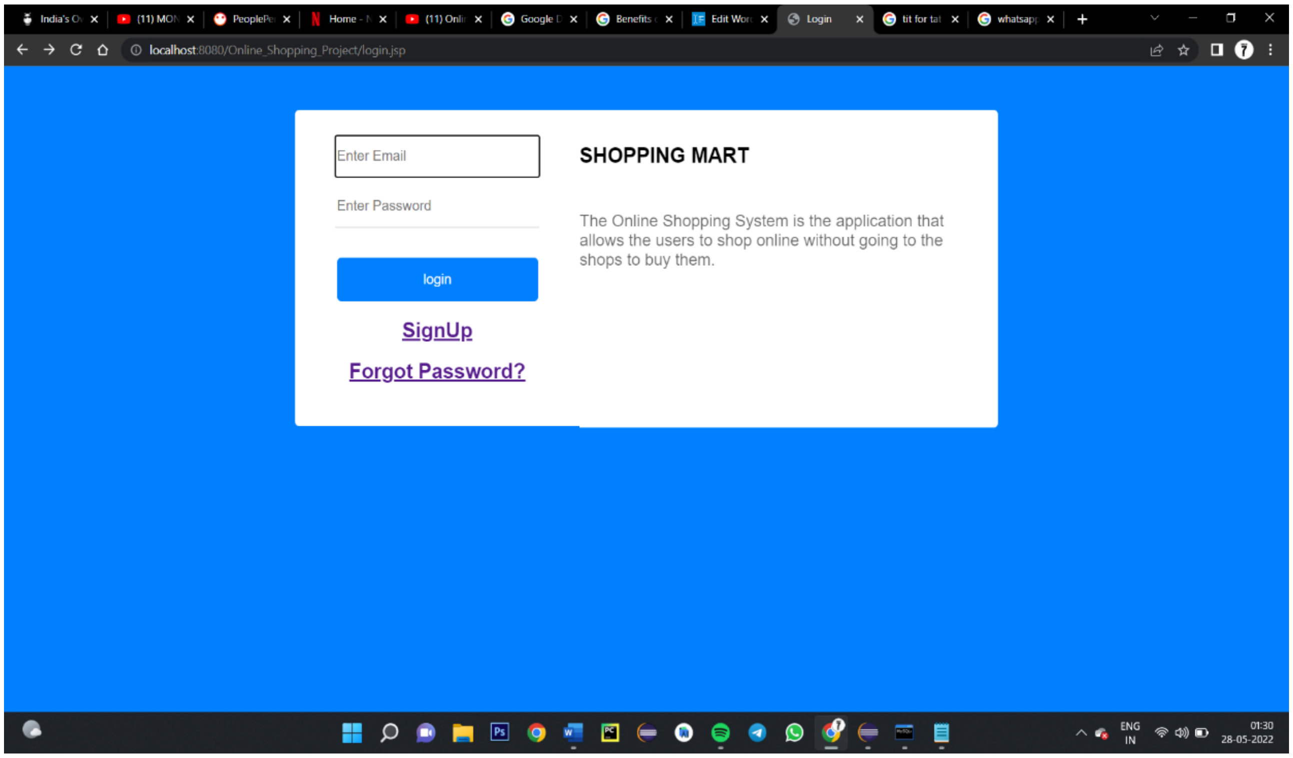
FORM DESIGN

4.1 INPUT / OUTPUT FORM / ADMIN FORM



The Online Shopping System is the application that allows the users to shop online without going to the shops to buy them.

Figure :4.1.1



SHOPPING MART

The Online Shopping System is the application that allows the users to shop online without going to the shops to buy them.

Figure : 4.1.2

CHAPTER 5

CODING

5.1 about.jsp

```
   <%@include file="header.jsp"%>
<%@include file="footer.jsp"%>
<%@page errorPage="error.jsp" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>About</title>
</head>
<body>

<div style="color: white; text-align: center; font-size: 30px;">About <i class="fa fa-address-book"></i></div>
<div style="background-color: white; padding:35px; font-size: 30px;">

<br>
<br>SHOPPING MART
<br>
or
<br>
</div>

</body>



</html>
```

5.2 addChangeAddress.jsp

```
<html>
<head>
<link rel="stylesheet" href="css/changeDetails.css">
<script src='https://kit.fontawesome.com/a076d05399.js'></script>
<title>Message Us</title>
</head>
<body>

<h3 class="alert">Address Successfully Updated !</h3>

<h3 class="alert">Some thing Went Wrong! Try Again!</h3>



<h3>Enter Address</h3>

 <hr>
 <h3>Enter city</h3>

<hr>
<h3>Enter State</h3>

<hr>
<h3>Enter country</h3>

<hr>
 <i class='far fa-arrow-alt-circle-right'></i>

</body>
<br><br><br>
</html>
```

5.3 addressPaymentForOrder.jsp

```
<%@page import="project.ConnectionProvider"%>
<%@page import="java.sql.*"%>
<%@include file="footer.jsp" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">


<html>
<head>
<link rel="stylesheet" href="css/addressPaymentForOrder-style.css">
<script src='https://kit.fontawesome.com/a076d05399.js'></script>
<title>Home</title>


<script>
if(window.history.forward(1) !=null)
        window.history.forward(1);</script>
</head>
<body>
<br>
<table>
<thead>

<%
String email=session.getAttribute("email").toString();
int total=0;
int sno=0;
try
{
        Connection con=ConnectionProvider.getCon();
        Statement st=con.createStatement();
        ResultSet rs1=st.executeQuery("select sum(total) from cart where email='"+email+"' and address is
NULL");
        while(rs1.next())
        {
                total=rs1.getInt(1);
        }
%>


      <tr>
      <th scope="col"><a href="myCart.jsp"><i class='fas fa-arrow-circle-left'> Back</i></a></th>
        <th scope="col" style="background-color: yellow;">Total: <i class="fa fa-inr"></i><%out.println(total); %>
</th>
      </tr>
    </thead>
    <thead>
      <tr>
      <th scope="col">S.No</th>
```

```
          <th scope="col">Product Name</th>
          <th scope="col">Category</th>
          <th scope="col"><i class="fa fa-inr"></i> price</th>
          <th scope="col">Quantity</th>
          <th scope="col">Sub Total</th>
        </tr>
      </thead>
      <tbody>


     <%
    ResultSet rs=st.executeQuery("select * from product inner join cart on product.id=cart.product_id and
cart.email='"+email+"' and cart.address is NULL");
    while(rs.next()){

    %>
        <tr>
        <%sno=sno+1; %>
         <td><%out.println(sno);%></td>
          <td><%=rs.getString(2)%></td>
          <td><%=rs.getString(3)%></td>
          <td><i class="fa fa-inr"></i> ><%=rs.getString(4)%></td>
          <td> <%=rs.getString(8)%></td>
          <td><i class="fa fa-inr"></i><%=rs.getString(10)%> </td>
          </tr>
       <%
       }




    ResultSet rs2=st.executeQuery("select * from users where email='"+email+"'");
    while(rs2.next())
    {

       %>
      </tbody>
    </table>

<hr style="width: 100%">
<form action="addressPaymentForOrderAction.jsp" method="post">
 <div class="left-div">
 <h3>Enter Address</h3>
<input class="input-style" type="text" name="address" value="<%=rs2.getString(7)%>" placeholder="Enter
Address" required>
 </div>

<div class="right-div">
<h3>Enter city</h3>
<input class="input-style" type="text" name="city" value="<%=rs2.getString(8)%>" placeholder="Enter City"
required>
</div>

<div class="left-div">
```

```
<h3>Enter State</h3>
<input class="input-style" type="text" name="state" value="<%=rs2.getString(9)%>" placeholder="Enter State"
required>
</div>



<div class="right-div">
<h3>Enter country</h3>
<input class="input-style" type="text" name="country" value="<%=rs2.getString(10)%>" placeholder="Enter
Country" required>
</div>
<h3 style="color: red">*If there is no address its mean that you did not set you address!</h3>
<h3 style="color: red">*This address will also updated to your profile</h3>
<hr style="width: 100%">
<div class="left-div">
<h3>Select way of Payment</h3>
 <select class="input-style" name="paymentMethod">
 <option value="Cash on delivery(COD)">Cash on delivery(COD)</option>
  <option value="Online Payment">Online Payment</option>
 </select>
</div>

<div class="right-div">
<h3>Pay online on this btechdays@pay.com</h3>
<input class="input-style" type="text" name="transactionId" placeholder="Enter Transaction ID" >
<h3 style="color: red">*If you select online Payment then enter you transaction ID here otherwise leave this
blank</h3>
</div>
<hr style="width: 100%">



<div class="left-div">
<h3>Mobile Number</h3>
<input class="input-style" type="text" name="mobileNumber" value="<%=rs2.getString(3)%>" placeholder="Enter
Mobile Number" required>
<h3 style="color: red">*This mobile number will also updated to your profile</h3>
</div>
<div class="right-div">
<h3 style="color: red">*If you enter wrong transaction id then your order will we can cancel!</h3>
<button class="button" type="submit">Proceed to Generate Bill & Save <i class='far fa-arrow-alt-circle-
right'></i></button>
<h3 style="color: red">*Fill form correctly</h3>
</div>
</form>
```

```
<%
}
}
catch(Exception e)
{
        System.out.println(e);
}
%>
   <br>
   <br>
   <br>

</body>
</html>
```

5.4 addreePaymentForOrderAction

```
<%@page import="project.ConnectionProvider"%>
<%@page import="java.sql.*"%>


<%
String email=session.getAttribute("email").toString();
String address=request.getParameter("address");
String city=request.getParameter("city");
String state=request.getParameter("state");
String country=request.getParameter("country");
String mobileNumber=request.getParameter("mobileNumber");
String paymentMethod=request.getParameter("paymentMethod");
String transactionId="";
transactionId=request.getParameter("tranasctionId");
String status="bill";



try{
        Connection con=ConnectionProvider.getCon();
        PreparedStatement ps=con.prepareStatement("update users set address=?,city=?,state=?,
country=?,mobileNumber=? where email=?");
        ps.setString(1,address);
        ps.setString(2,city);
        ps.setString(3,state);
        ps.setString(4,country);
        ps.setString(5,mobileNumber);
        ps.setString(6,email);
        ps.executeUpdate();

        PreparedStatement ps1=con.prepareStatement("update cart set
address=?,city=?,state=?,country=?,mobileNumber=?,orderDate=now(),deliveryDate=Date_ADD(orderDate,INTER
VAL 7 DAY),paymentMethod=?,transactionId=?,status=? where email=? and address is NULL");


        ps1.setString(1,address);
        ps1.setString(2,city);
        ps1.setString(3,state);
        ps1.setString(4,country);
        ps1.setString(5,mobileNumber);
        ps1.setString(6,paymentMethod);
        ps1.setString(7,transactionId);
        ps1.setString(8,status);
        ps1.setString(9,email);


        ps1.executeUpdate();

        response.sendRedirect("bill.jsp");
}

catch(Exception e)
{
	System.out.println(e);
}
%>
```

5.5 addToCart.jsp

```
<%@page import="project.ConnectionProvider"%>
<%@page import="java.sql.*"%>


<%
String email=session.getAttribute("email").toString();

String address=request.getParameter("address");

String city=request.getParameter("city");

String state=request.getParameter("state");

String country=request.getParameter("country");

String mobileNumber=request.getParameter("mobileNumber");

String paymentMethod=request.getParameter("paymentMethod");


String transactionId="";
transactionId=request.getParameter("tranasctionId");

String status="bill";

try{
        Connection con=ConnectionProvider.getCon();

        PreparedStatement ps=con.prepareStatement("update users set address=?,city=?,state=?,
country=?,mobileNumber=? where email=?");


        ps.setString(1,address);
        ps.setString(2,city);
        ps.setString(3,state);
        ps.setString(4,country);
        ps.setString(5,mobileNumber);
        ps.setString(6,email);
        ps.executeUpdate();

        PreparedStatement ps1=con.prepareStatement("update cart set
address=?,city=?,state=?,country=?,mobileNumber=?,orderDate=now(),deliveryDate=Date_ADD(orderDate,INTER
VAL 7 DAY),paymentMethod=?,transactionId=?,status=? where email=? and address is NULL");


        ps1.setString(1,address);
        ps1.setString(2,city);
        ps1.setString(3,state);
        ps1.setString(4,country);
        ps1.setString(5,mobileNumber);
        ps1.setString(6,paymentMethod);
        ps1.setString(7,transactionId);
        ps1.setString(8,status);
        ps1.setString(9,email);

        ps1.executeUpdate();
        response.sendRedirect("bill.jsp");
}
catch(Exception e)
{
        System.out.println(e);
}
%>
```

5.6 changePassword.jsp

```
<%@page import="project.ConnectionProvider"%>
<%@page import="java.sql.*"%>
<%@include file="changeDetailsHeader.jsp" %>
<%@include file="footer.jsp" %>
<html>
<head>


<link rel="stylesheet" href="css/changeDetails.css">
<script src='https://kit.fontawesome.com/a076d05399.js'></script>
<title>Message Us</title>
</head>
<body>

<%

  String msg=request.getParameter("msg");
if("notMatch".equals(msg)){
%>
<h3 class="alert">New password and Confirm password does not match!</h3>
<%} %>


<%
if("wrong".equals(msg)){
%>
<h3 class="alert">Your old Password is wrong!</h3>
<%} %>


<%
if("done".equals(msg)){
%>
<h3 class="alert">Password change successfully!</h3>
<%} %>


<%
if("invalid".equals(msg)){
%>
<h3 class="alert">Some thing went wrong! Try again!</h3>
<%} %>


<form action="changePasswordAction.jsp" method="post">


<h3>Enter Old Password</h3>
 <input class="input-style" type="password" name="oldPassword" placeholder="Enter Old Password" required>
```

```
  <hr>
 <h3>Enter New Password</h3>
 <input class="input-style" type="password" name="newPassword" placeholder="Enter New Password" required>
 <hr>
<h3>Enter Confirm Password</h3>
<input class="input-style" type="password" name="confirmPassword" placeholder="Enter Confirm Password"
required>
<hr>
 <button class="button" type="submit">Save<i class='far fa-arrow-alt-circle-right'></i></button>
</form>
</body>
<br><br><br>
</html>
```

5.7 forgotpassword.jsp

```
<%@page import="project.ConnectionProvider"%>
<%@page import="java.sql.*"%>

<%
String email=request.getParameter("email");
String mobileNumber=request.getParameter("mobileNumber");
String securityQuestion=request.getParameter("securityQuestion");
String answer=request.getParameter("answer");
String newPassword=request.getParameter("newPassword");

int check=0;
try
{
	Connection con = ConnectionProvider.getCon();
	Statement st = con.createStatement();

	ResultSet rs=st.executeQuery("select * from users where email ='"+email+"' and
mobileNumber='"+mobileNumber+"' and securityQuestion='"+securityQuestion+"' and answer='"+answer+"'");


	while(rs.next())
	{
		check=1;
		st.executeUpdate("update users set password='"+newPassword+"' where email='"+email+"'");
		response.sendRedirect("forgotPassword.jsp?msg=done");
	}
	if(check==0)
{
		response.sendRedirect("forgotPassword.jsp?msg=invalid");
	}
}
catch(Exception e)
{
	System.out.println(e);
}

%>
```

5.8 home.jsp

```
<%@page import="project.ConnectionProvider"%>
<%@page import="java.sql.*"%>
<%@include file="header.jsp" %>
<%@include file="footer.jsp" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Home</title>

<style>
h3
{
	color: yellow;
	text-align: center;
}
</style>

</head>
<body>
<div style="color: white; text-align: center; font-size: 30px;">Home <i class="fa fa-institution"></i></div>

<%
String msg=request.getParameter("msg");
if("added".equals(msg))
{
%>
<h3 class="alert">Product added successfully!</h3>
<%} %>

<%
if("exist".equals(msg))
{
%>
<h3 class="alert">Product already exist in you cart! Quantity  increased!</h3>
<%} %>

<%
if("invalid".equals(msg))
{
%>
<h3 class="alert">Something went wrong! Try again!</h3>
<%} %>


<table>
        <thead>
          <tr>
            <th scope="col">ID</th>
            <th scope="col">Name</th>
            <th scope="col">Category</th>
            <th scope="col"><i class="fa fa-inr"></i> Price</th>
            <th scope="col">Add to cart <i class='fas fa-cart-plus'></i></th>
          </tr>
        </thead>
        <tbody>
<%
try{

             Connection con=ConnectionProvider.getCon();
             Statement st=con.createStatement();

             ResultSet rs=st.executeQuery("select * from product where active='Yes'");
             while(rs.next())
             {
%>
          <tr>
            <td><%=rs.getString(1) %></td>
            <td><%=rs.getString(2) %></td>
            <td><%=rs.getString(3) %></td>
            <td><i class="fa fa-inr"></i><%=rs.getString(4) %> </i></td>
            <td><a href="addToCartAction.jsp?id=<%=rs.getString(1) %>">Add to cart <i class='fas fa-cart-
plus'></i></a></td>
          </tr>
<%}
             }
             catch(Exception e)
{
             System.out.println(e);
             }

             %>
        </tbody>
      </table>
      <br>
      <br>
      <br>

</body>
</html>
```

5.9 login.jsp

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="css/signup-style.css">
<title>Login</title>
</head>
<body>
<div id='container'>
  <div class='signup'>

     <form action="loginAction.jsp" method="post">

             <input type="email" name="email" placeholder="Enter Email" required>
             <input type="password" name="password" placeholder="Enter Password" required>
             <input type="submit" value="login">

     </form>
      <h2><a href="signup.jsp">SignUp</a></h2>
       <h2><a href="forgotPassword.jsp">Forgot Password?</a></h2>
  </div>
  <div class='whysignLogin'>
 <%
String msg=request.getParameter("msg");

 if("notexist".equals(msg))
 {
%>
  <h1>Incorrect Username or Password</h1>
<%
}
%>

<%
if("invalid".equals(msg))
{
%>
<h1>Some thing Went Wrong! Try Again !</h1>
<%
}
%>


    <h2>SHOPPING MART</h2>
    <p>The Online Shopping System is the application that allows the users to shop online without going to the shops
to buy them.</p>

  </div>
</div>
```

```
</body>
</html>
```

5.9 loginAction.jsp

```
<%@page import="project.ConnectionProvider"%>
<%@page import="java.sql.*"%>

<%
String email=request.getParameter("email");
String password=request.getParameter("password");
if("admin@gmail.com".equals(email) && "admin".equals(password))

{
	session.setAttribute("email", email);
	response.sendRedirect("admin/adminHome.jsp");
}
else
{
	int z=0;
	try
	{
		Connection con=ConnectionProvider.getCon();
		Statement st=con.createStatement();
		ResultSet rs=st.executeQuery("select *from users where email='"+email+"' and
password='"+password+"'");
		while(rs.next())
		{
			z=1;
			session.setAttribute("email", email);
			response.sendRedirect("home.jsp");
		}
		if(z==0)
			response.sendRedirect("login.jsp?msg=notexist");
	}
	catch(Exception e)
	{
		System.out.println(e);
		response.sendRedirect("login.jsp?msg=invalid");
	}
}
%>
```

5.10 bill.jsp

```
<%@page import="project.ConnectionProvider"%>
<%@page import="java.sql.*"%>

<%
String email=request.getParameter("email");
String password=request.getParameter("password");
if("admin@gmail.com".equals(email) && "admin".equals(password))
{
	session.setAttribute("email", email);
	response.sendRedirect("admin/adminHome.jsp");
}
else
{
	int z=0;
	try
	{
		Connection con=ConnectionProvider.getCon();
		Statement st=con.createStatement();
		ResultSet rs=st.executeQuery("select *from users where email='"+email+"' and
password='"+password+"'");
		while(rs.next())
		{
			z=1;
			session.setAttribute("email", email);
			response.sendRedirect("home.jsp");
		}
		if(z==0)
			response.sendRedirect("login.jsp?msg=notexist");
	}
	catch(Exception e)
	{
		System.out.println(e);
		response.sendRedirect("login.jsp?msg=invalid");
	}
```

5.11 changeSecurityQuestion.jsp

```
<%@page import="project.ConnectionProvider"%>
<%@page import="java.sql.*"%>

<%
String email=request.getParameter("email");

String password=request.getParameter("password");

if("admin@gmail.com".equals(email) && "admin".equals(password))
{
	session.setAttribute("email", email);
	response.sendRedirect("admin/adminHome.jsp");
}
else
{
	int z=0;
	try
	{
		Connection con=ConnectionProvider.getCon();
		Statement st=con.createStatement();

		ResultSet rs=st.executeQuery("select *from users where email='"+email+"' and 
password='"+password+"'");

		while(rs.next())
		{
			z=1;
			session.setAttribute("email", email);
			response.sendRedirect("home.jsp");
		}
		if(z==0)
			response.sendRedirect("login.jsp?msg=notexist");
	}
	catch(Exception e)
	{
		System.out.println(e);
		response.sendRedirect("login.jsp?msg=invalid");
	}
}
%>
```

5.12 header.jsp

```
<%@page errorPage="error.jsp" %>
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="css/home-style.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
<script src='https://kit.fontawesome.com/a076d05399.js'></script>
</head>
   <!--Header-->
   <br>
   <div class="topnav sticky">
   <%String email=session.getAttribute("email").toString(); %>

           <center><h2>Shopping Mart</h2></center>
           <h2><a href=""><%out.println(email);%><i class='fas fa-user-alt'></i></a></h2>
           <a href="home.jsp">Home<i class="fa fa-institution"></i></a>
           <a href="myCart.jsp">My Cart<i class='fas fa-cart-arrow-down'></i></a>
           <a href="myOrders.jsp">My Orders  <i class='fab fa-elementor'></i></a>
           <a href="changeDetails.jsp">Change Details <i class="fa fa-edit"></i></a>
           <a href="">Message Us <i class='fas fa-comment-alt'></i></a>
           <a href="">About <i class="fa fa-address-book"></i></a>
           <a href="logout.jsp">Logout <i class='fas fa-share-square'></i></a>
           <div class="search-container">


            <form action="searchHome.jsp" method="post">
            <input type="text" placeholder="Search" name="search">
            <button type="submit"><i class="fa fa-search"></i></button>
            </form>



           </div>
          </div>
          <br>
          <!--table-->
```

5.13 mycart.jsp

```
<%@page import="project.ConnectionProvider"%>
<%@page import="java.sql.*"%>
<%@include file="header.jsp" %>
<%@include file="footer.jsp" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>My Cart</title>


<style>
h3
{
	color: yellow;
	text-align: center;
}
</style>


</head>
<body>
<div style="color: white; text-align: center; font-size: 30px;">My Cart <i class='fas fa-cart-arrow-down'></i></div>

<%
String msg=request.getParameter("msg");
if("notPossible".equals(msg)){
%>
<h3 class="alert">There is only one Quantity! So click on remove!</h3>
<%} %>

<%

if("inc".equals(msg)){
%>
<h3 class="alert">Quantity  Increased Successfully!</h3>
<%} %>

<%

if("dec".equals(msg)){
%>
<h3 class="alert">Quantity  Decreased Successfully!</h3>
<%} %>

<%
if("remove".equals(msg)){
%>
<h3 class="alert">Product Successfully Removed!</h3>
```

```
<%} %>

<table>
<thead>
<%
int total=0;
int sno=0;


try
{
        Connection con=ConnectionProvider.getCon();
        Statement st=con.createStatement();
        ResultSet rs1=st.executeQuery("select sum(total) from cart where email='"+email+"' and address is
NULL");
        while(rs1.next())
{
                total=rs1.getInt(1);
        }

%>
      <tr>
        <th scope="col" style="background-color: yellow;">Total: <i class="fa fa-inr"></i><%out.println(total); %>
</th>
       <%if(total>0){ %><th scope="col"><a href="addressPaymentForOrder.jsp">Proceed to
order</a></th><%}%>


      </tr>
    </thead>
    <thead>
      <tr>
      <th scope="col">S.No</th>
        <th scope="col">Product Name</th>
        <th scope="col">Category</th>
        <th scope="col"><i class="fa fa-inr"></i> price</th>
        <th scope="col">Quantity</th>
        <th scope="col">Sub Total</th>
        <th scope="col">Remove <i class='fas fa-trash-alt'></i></th>
      </tr>
    </thead>
    <tbody>


   <%
   ResultSet rs=st.executeQuery("select * from product inner join cart on product.id=cart.product_id and
cart.email='"+email+"' and cart.address is NULL");
   while(rs.next())
{


   %>
      <tr>
      <%sno=sno+1; %>
```

```
           <td><%out.println(sno);%></td>
            <td><%=rs.getString(2)%></td>
            <td><%=rs.getString(3)%></td>
            <td><i class="fa fa-inr"></i> <%=rs.getString(4)%></td>
            <td><a href="incDecQuantityAction.jsp?id=<%=rs.getString(1)%>&quantity=inc"><i class='fas fa-plus-
circle'></i></a><%=rs.getString(8)%>  <a
href="incDecQuantityAction.jsp?id=<%=rs.getString(1)%>&quantity=dec"><i class='fas fa-minus-
circle'></i></a></td>


            <td><i class="fa fa-inr"></i> <%=rs.getString(10)%></td>
            <td><a href="removeFromCart.jsp?id=<%=rs.getString(1)%>">Remove <i class='fas fa-trash-
alt'></i></a></td>
          </tr>
<%
}
}
catch(Exception e)
{}
%>
        </tbody>
      </table>
      <br>
      <br>
      <br>

</body>
</html>
```

5.14 sinupAction.jsp

```
<%@page import="project.ConnectionProvider"%>
<%@page import="java.sql.*"%>

<%
String name =request.getParameter("name");

String email =request.getParameter("email");

String mobileNumber =request.getParameter("mobileNumber");

String securityQuestion =request.getParameter("securityQuestion");

String answer =request.getParameter("answer");

String password =request.getParameter("password");

String address="";
String city="";
String state="";
String country="";
try
{
	Connection con=ConnectionProvider.getCon();
	PreparedStatement ps = con.prepareStatement("insert into users values(?,?,?,?,?,?,?,?,?,?)");
	ps.setString(1, name);
	ps.setString(2, email);
	ps.setString(3, mobileNumber);
	ps.setString(4, securityQuestion);
	ps.setString(5, answer);
	ps.setString(6, password);
	ps.setString(7, address);
	ps.setString(8, city);
	ps.setString(9, state);
	ps.setString(10, country);

	ps.executeUpdate();
	response.sendRedirect("signup.jsp?msg=valid");

}
catch(Exception e)
{
	System.out.println(e);
	response.sendRedirect("signup.jsp?msg=invalid");

}
%>
```

5.15 sinup.jsp

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="css/signup-style.css">
<title>Signup</title>
</head>
<body>
<div id='container'>
  <div class='signup'>


    <form action="signupAction.jsp" method="post">
            <input type="text" name="name" placeholder="Enter Name" required>
            <input type ="email" name ="email" placeholder ="Enter Email" required>
            <input type ="number" name ="mobileNumber" placeholder ="Enter Mobile Number" required>

    <select name ="securityQuestion" required>

            <option value ="What was your First car?">What was your First car?</option>
            <option value ="What is the name of your first pet?">What is the name of your first pet?</option>
            <option value ="What elementary school did you attend?">What elementary school did you
attend?</option>
            <option value ="What is the name of the town where you were born?">What is the name of the town where
you were born?</option>

    </select>


    <input type ="text" name ="answer" placeholder ="answer" required>
    <input type ="password" name ="password" placeholder ="Enter Password" required>
    <input type ="submit" value="signup">
    </form>


      <h2><a href="login.jsp">Login</a></h2>
  </div>
  <div class='whysign'>


<%
String msg=request.getParameter("msg");
if("valid".equals(msg))
{
%>


<h1>Successfully Registered</h1>
<%}%>



<%
if("invalid".equals(msg))
{
%>

            <h1>Some thing Went Wrong! Try Again !</h1>
<%}%>




    <h2>SHOPPING MART</h2>
    <p>The Online Shopping System is the application that allows the users to shop online without going to the shops
to buy them.</p>
  </div>
</div>

</body>
</html>
```

5.16 searchHome.jsp

```
<%@page import="project.ConnectionProvider"%>
<%@page import="java.sql.*"%>
<%@include file="header.jsp" %>
<%@include file="footer.jsp" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Home</title>
</head>
<body>
<div style="color: white; text-align: center; font-size: 30px;">Home <i class="fa fa-institution"></i></div>
<table>

        <thead>
          <tr>
            <th scope="col">ID</th>
            <th scope="col">Name</th>
            <th scope="col">Category</th>
            <th scope="col"><i class="fa fa-inr"></i> Price</th>
            <th scope="col">Add to cart <i class='fas fa-cart-plus'></i></th>
          </tr>
        </thead>

        <tbody>
<%
int z=0;
try
{
        String search=request.getParameter("search");

        Connection con=ConnectionProvider.getCon();

        Statement st=con.createStatement();

        ResultSet rs=st.executeQuery("select * from product where name like '%"+search+"%' or category like
'%"+search+"%' and active='Yes'");

        while(rs.next())
        {
                z=1;


%>
          <tr>
            <td><%=rs.getString(1) %></td>
            <td><%=rs.getString(2) %></td>
            <td><%=rs.getString(3) %></td>
            <td><i class="fa fa-inr"></i><%=rs.getString(4) %> </i></td>
            <td><a href="addToCartAction.jsp?id=<%=rs.getString(1) %>">Add to cart <i class='fas fa-cart-
plus'></i></a></td>


          </tr>
        <%
            }
}
            catch(Exception e)
{
            System.out.println(e);
            }
            %>
        </tbody>
      </table>


            <%if(z==0){ %>
            <h1 style="color:white; text-align: center;">Nothing to show</h1>
            <%} %>
      <br>
      <br>
      <br>
      <div class="footer">
          <p>All right reserved by Shoppimg Mart</p>
      </div>

</body>
</html>
```

CHAPTER 6

TESTING

6.1 : TESTING

6.1.1 INTRODUCTION

Testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use. Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

- meets the requirements that guided its design and development,
- responds correctly to all kinds of inputs,
- performs its functions within an acceptable time,
- it is sufficiently usable,
- can be installed and run in its intended environments, and
- Achieves the general result its stakeholder's desire.

6.1.2 Static vs. dynamic testing:

There are many approaches available in software testing. Reviews, walkthroughs, or inspections are referred to as static testing, whereas executing programmed code with a given set of test cases is referred to as dynamic testing.

Static testing is often implicit, like proofreading, plus when programming tools/text editors check source code structure or compilers (pre-compilers) check syntax and data flow as static program analysis. Dynamic testing takes place when the program itself is run. Dynamic testing may begin before the program is 100% complete in order to test particular sections of code and are applied to discrete functions or modules. Typical techniques for these are either using stubs/drivers or execution from a debugger environment.

6.1.3 White-box testing:

White-box testing (also known as clear box testing, glass box testing, transparent box testing and structural testing) verifies the internal structures or workings of a program, as opposed to the functionality exposed to the end-user. In white-box testing, an internal perspective of the system (the source code), as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g., in-circuit testing (ICT).

While white-box testing can be applied at the unit, integration, and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

Techniques used in white-box testing include:

- API testing – testing of the application using public and private APIs (application programming interfaces)
- Code coverage – creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once)
- Fault injection methods – intentionally introducing faults to gauge the efficacy of testing strategies
- Mutation testing methods
- Static testing methods

Code coverage tools can evaluate the completeness of a test suite that was created with any method, including black-box testing. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested. Code coverage as a software metric can be reported as a percentage for:

- Function coverage, which reports on functions executed
- Statement coverage, which reports on the number of lines executed to complete the test
- Decision coverage, which reports on whether both the True and the False branch of a given tens.

100% statement coverage ensures that all code paths or branches (in terms of control flow) are executed at least once. This is helpful in ensuring correct functionality, but not sufficient since the same code may process different inputs correctly or incorrectly. Pseudo-tested functions and methods are those that are covered but not specified (it is possible to remove their body without breaking any test case).

6.1.4 Black-box testing:

Input
Blackbox
Output

Black-box testing (also known as functional testing) treats the software as a "black box," examining functionality without any knowledge of internal implementation, without seeing the source code. The testers are only aware of what the software is supposed to do, not how it does it. Black-box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, state transition tables, decision table testing, fuzz testing, model-based testing, use case testing, exploratory testing, and specification-based testing.

Specification-based testing aims to test the functionality of software according to the applicable requirements This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case. Test cases are built around specifications and requirements, i.e., what the application is supposed to do. It uses external descriptions of the software, including specifications, requirements, and designs to derive test cases. These tests can be functional or non-functional, though usually functional.

Specification-based testing may be necessary to assure correct functionality, but it is insufficient to guard against complex or high-risk situations.

One advantage of the black box technique is that no programming knowledge is required. Whatever biases the programmers may have had, the tester likely has a different set and may emphasize different areas of functionality. On the other hand, black-box testing has been said to be "like a walk in a dark labyrinth without a flashlight." Because they do not examine the source code, there are situations when a tester writes many test cases to check something that could have been tested by only one test case or leaves some parts of the program untested.

This method of test can be applied to all levels of software testing: unit, integration, system and acceptance. It typically comprises most if not all testing at higher levels, but can also dominate unit testing as well.

6.2 TEST CASES

6.2.1 Testing shopping cart

Some quantity of products was inserted into the shopping cart, and then we proceeded to check out. When we checked out and tested with Paypal, the cart became empty. This indicates that the cart works appropriately as it should. The "index.js" file is responsible for handling the cart of the shop. When an editor runs the file "index.js" the cart() function is called which insert products into the database. After that, the view cart() function can also be invoked to show the product in the cart. The customer can manipulates his cart such as updating the cart or, adding a product to cart.

6.2.2 Customer checking out

The customer can only check out if he has logged in as a customer. Without being a customer, he is unable to checkout successfully.

CHAPTER 7

CONCLUSION

In general, today’s businesses must always strive to create the next best thing that consumers will want because consumers continue to desire their products, services etc. to continuously be better, faster, and cheaper. In this world of new technology, businesses need to accommodate to the new types of consumer needs and trends because it will prove to be vital to their business’ success and survival. E-commerce is continuously progressing and is becoming more and more important to businesses as technology continues to advance and is something that should be taken advantage of and implemented. From the inception of the Internet and e-commerce, the possibilities have become endless for both businesses and consumers. Creating more opportunities for profit and advancements for businesses, while creating more options for consumers. However, just like anything else, e-commerce has its disadvantages including consumer uncertainties, but nothing that can not be resolved or avoided by good decision-making and business practices.

BIBILIOGRAPHY

- https://www.tutorialspoint.com/index.htm
- https://www.javatpoint.com
- https://www.w3schools.com
- https://html.com