MEDICAL E-COMMERCE SYSTEM

CS23333 – Object Oriented Programming using Java Project Report

Submitted by

BHARANIDHARAN R.V -231001026

DEEPAN KUMAR S -231001030

Of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

RAJALAKSHMI ENGINEERING COLLEGE

NOVEMBER-2024

BONAFIDE CERTIFICATE

Certified that this project titled "MEDICAL E-COMMERCE SYSTEM" is the bonafide work of "**BHARANIDHRAN(231001026), DEEPAN KUMAR(231001030)** " who carried out the project work under my supervision.

SIGNATURE
Dr.P.Valarmathie
HEAD OF THE DEPARTMENT

Department of Information Technology
Rajalakshmi Engineering College

SIGNATURE
Mrs.Usha S
COURSE INCHARGE
Assistant Professor(S.G)

Department of Information Technology
Rajalakshmi Engineering College

This project is submitted for CS23333 – Object Oriented Programming using Java held on _____

INTERNAL EXAMINAR **EXTERNAL EXAMINAR**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF Table

1. Abstract

The **Medical E-commerce System** is a comprehensive platform designed to streamline the purchase of medical supplies, including prescription drugs, over-the-counter medications, and healthcare devices. Leveraging technologies like Java for backend logic, MySQL for database management, and HTML/CSS/JavaScript for a user-friendly interface, The platform integrates secure authentication, prescription validation, real-time inventory updates, and seamless online payment gateways the system accommodates healthcare providers, pharmacies, and patients, ensuring a secure and efficient transaction experience.

2. Introduction

In today's digital age, e-commerce has revolutionized the healthcare sector by enabling easy access to medical products. The Medical E-commerce System automates traditional processes such as prescription validation, inventory management, and delivery tracking. With a focus on user-centric design, the system addresses challenges like maintaining data security and ensuring medication authenticity. By providing a convenient solution for patients to order medicines from the comfort of their homes, the system bridges the gap between pharmacies and consumers.

3. Purpose

The primary goal of this project is to create a robust, user-friendly platform that benefits both customers and pharmacy administrators. Objectives include:

- Simplifying the process of purchasing medical products online.
- Provide a user-friendly interface for efficient data entry and retrieval.
- Ensure data consistency and accuracy through secure database connectivity using JDBC
- Automating prescription validation to ensure legal compliance.
- Providing real-time inventory updates to customers.

4. Scope of the Project:

The Medical E-commerce System is intended for diverse users ,including patients, pharmacy owners, and healthcare providers.

The system features:

Secure user authentication for customers and administators Real-time tracking of orders and deliveries Integration with third-party payment gateways for secure transactions Compatibility with various devices, including desktops and mobile phones The project addresses the challenges of manual operations, such as stock discrepancies, data security risks, and slow order fulfillment, offering a modern, automated alternative.

5. Software Requirement Specification

Introduction

This section outlines the essential hardware and software requirements needed to develop and deploy the Medical E-commerce System, ensuring smooth operation, scalability, and user satisfaction. It defines the tools and platforms required for building a reliable and efficient solution.

Product Scope

The Medical E-commerce System is designed to simplify the online purchasing of medical products and services by offering a seamless and secure platform. Developed using Java and MySQL, this system ensures reliable, real-time data management, transaction processing, and customer interaction. It supports a wide range of medical products, including prescription drugs, health equipment, and medical supplies, catering to both individual consumers and healthcare institutions.

References and Acknowledgement

1. https://youtu.be/oiRq01AMeSAsi=K5FEJJbcbPpcf51t
2. https://youtu.be/2NolH6D2xcgsi=3XwbziWVHpvLXr0L

Overall Description

A medical e-commerce website using Java, JSP, and MySQL is an online platform where users can browse, purchase, and manage medical products such as medicines, medical equipment. The website would have features like user authentication, product listings and Java is used for server-side logic, JSP for dynamic web pages, and MySQL for database management to handle products, customer data, and transactions.

Product Perspective

The medical e-commerce website is a user-friendly platform that provides access to a wide range of medical products, The site supports personalized user accounts, order tracking, and product recommendations.It uses Java and JSP for backend processing and dynamic content generation. MySQL handles data storage for products, users, orders, and transactions efficiently.

Product Functionality

a) User Registration & Authentication: Users can create accounts, log in, and manage profiles.
b) Product Search & Filtering: Allows users to search and filter products based on categories (e.g., medications, medical equipment), brand, price, and health conditions.
c) Product Details & Reviews: Provides detailed product information, including descriptions, usage instructions, and customer reviews/ratings.
d) Shopping Cart: Users can add products to their cart, modify quantities, and view a summary of selected items.
e) Order Placement & Payment Integration: Users can proceed to checkout, select payment methods, and make secure online payments (credit card, e-wallets, etc.).

User and Characteristics

Qualification: Users should have at least basic educational qualifications, such as matriculation, and be comfortable with English.

Experience: Familiarity with the university registration process is advantageous.

Technical Experience: Users are expected to have elementary knowledge of computers for optimal system

interaction.

Operating Environment

Hardware Requirements

- Processor: Intel i3 or higher (or equivalent AMD processor)
- Operating System: Windows 8,10, 11
- Processor Speed: 2.0 GHz
- RAM: 4GB
- Hard Disk: 500GB

Software Requirements

- Database: MySQL
- Frontend: JSP
- Technology: Java (JDBC)

Constraints

- System access is limited to authorized personnel for security purposes.
- The system is designed to handle a moderate volume of employee data; scalability for very large datasets may require optimization.

User Interface

The Medical E-Commerce Store provides user-friendly, menu-driven interfaces for:

a) Register: Registering new user.

b) Login: Logging in existing user.

c) Product pages: with detailed information, images, prices.

d) Shopping cart: with visible item summaries and easy quantity adjustments.

e) Add Quantity: increase the quantity of the product.

g) Order: Viewing and Place order.

Hardware Interface

- Screen resolution of at least 640 x 480 or above.
- Compatible with any version of Windows 8, 10, 11.

Software Interface

a) MS-Windows Operating System .

b) JSP for designing the front end .

c) java for the backend & MYSQL as database.

d) Platform: Java Language.

Functional Requirements

User Registration and Authentication:

- Allow users to register with their details (name, email, password, address, etc.).
- Provide secure login/logout functionality using encrypted passwords.

Product Catalog Management:

- Display a categorized list of medical products with filtering options.
- Show detailed product descriptions, images, prices, and stock availability.

Shopping Cart and Checkout:

- Allow users to add, update, or remove products in their shopping cart.

Payment Gateway Integration:

- Support multiple payment option (credit/debit cards).
- Ensure secure transactions through encryption and fraud detection mechanisms.

Admin Functions:

- Allow admins to manage product inventory (add, update, delete products).
- Provide order management features, including viewing and updating order status.

Reports and Analytics:

- Generate sales and user activity reports for business insights.

Non-functional Requirements

1) Performance:

The website should load quickly (within 2-3 seconds) and handle a high volume of traffic without delays or crashes.

2) Scalability:

It should support increasing numbers of users, transactions, and product listings without performance degradation.

3) Security:

Ensure data protection through SSL encryption, secure payment processing, and compliance with privacy regulations (e.g., HIPAA).

4) Availability:

The website should be available 99.9% of the time, with minimal downtime for maintenance or upgrades.

5) Usability:

The site should be user-friendly, with an intuitive interface and smooth navigation for all user groups (patients, healthcare providers, admins).

6) Reliability:

The system must be stable and consistently deliver expected functionality, with minimal errors or disruptions.

7) Compatibility:

The website should be compatible with major browsers (Chrome, Firefox, Safari, etc.) and devices (desktops, tablets, smartphones).

2.SYSTEM FLOW DIAGRAMS

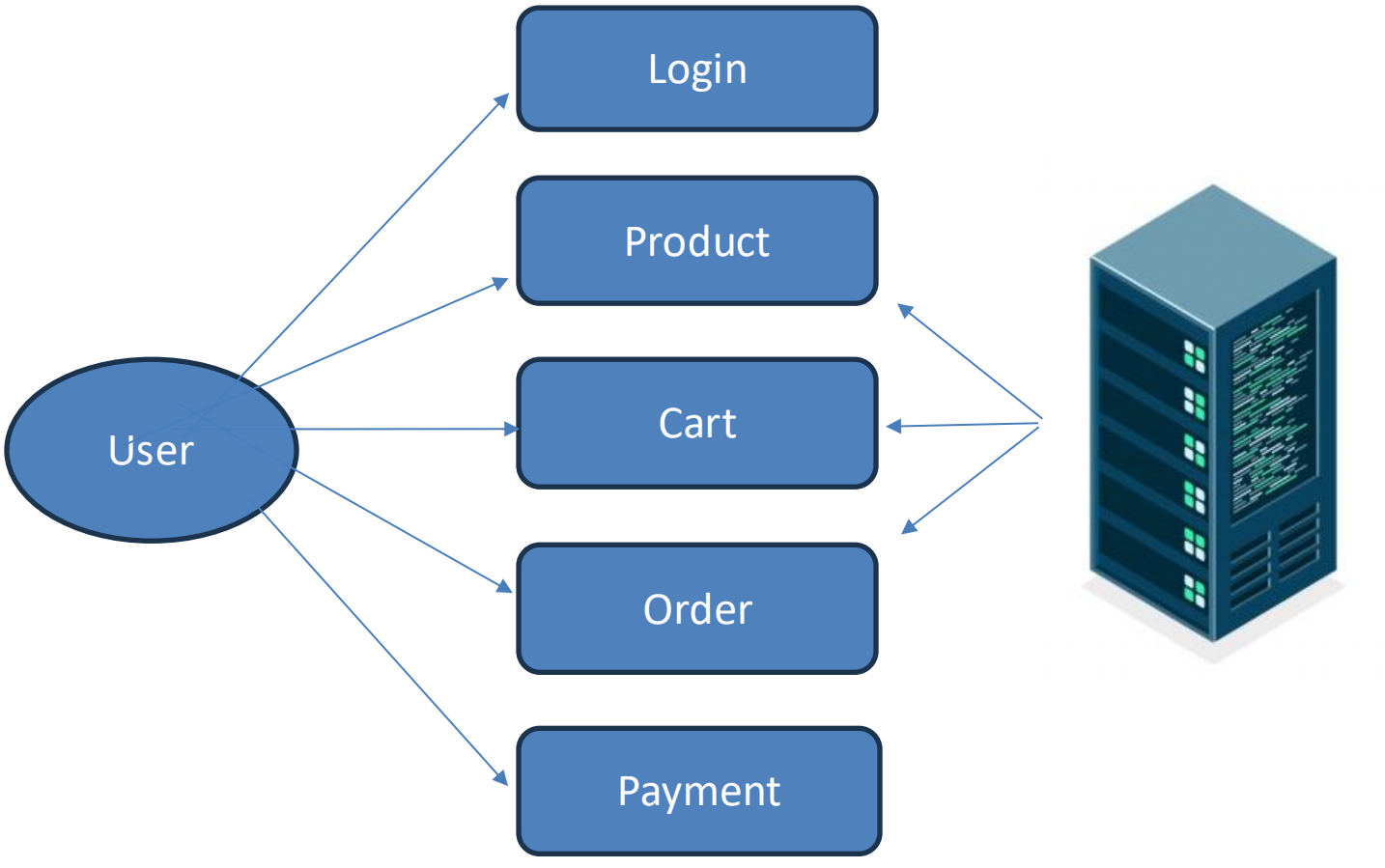

Figure 2.1 Use Case Diagrams

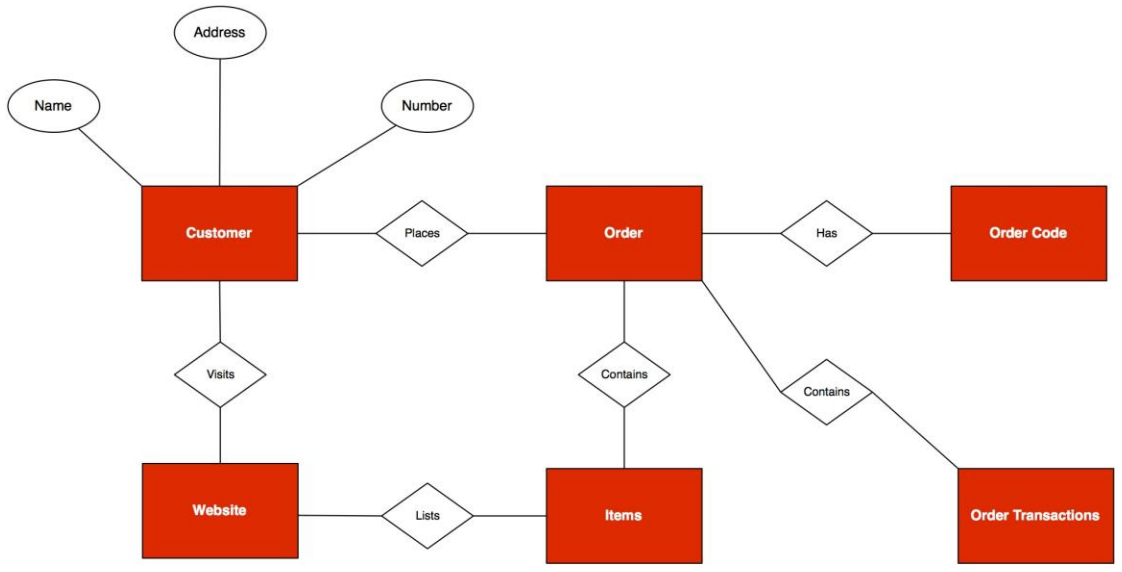

Figure 2.2 Entity Relationship Diagram

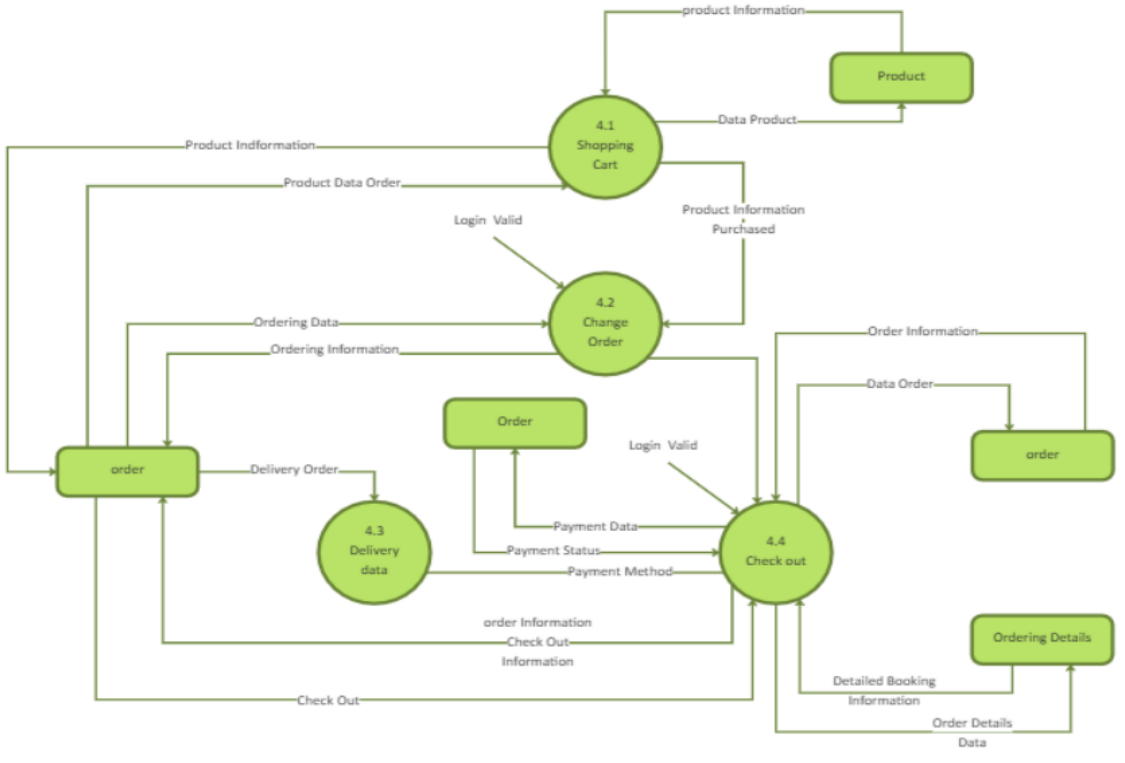

Figure 2.3 Data-flow diagram

3. MODULE DESCRIPTION

1) User Management :Handles user registration, login, profile management, and authentication for customers and admins.

2) Product Management :Allows admins to add, update, delete, and categorize medical products; manage stock levels and product details.

3) Shopping Cart :Manages the user's selected products, including quantity adjustments, view items, and proceed to checkout.

4) Order Management :Handles order placement, order history for users and admins.

5) Payment & Checkout :Integrates secure payment gateways for processing transactions, in including credit cards, digital wallets, and other payment methods.

6) Admin Dashboard Module Provides an interface for admins to monitor sales, manage users, track inventory, and generate reports.

4.1 Design

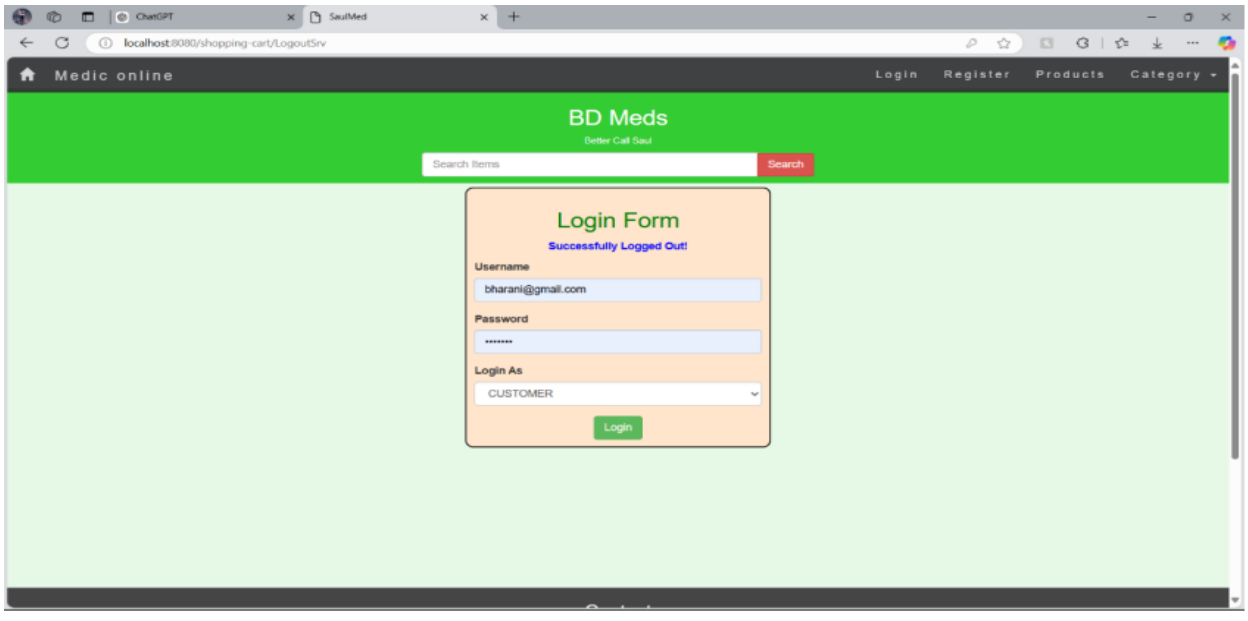

Figure 4.1.1 Login Page

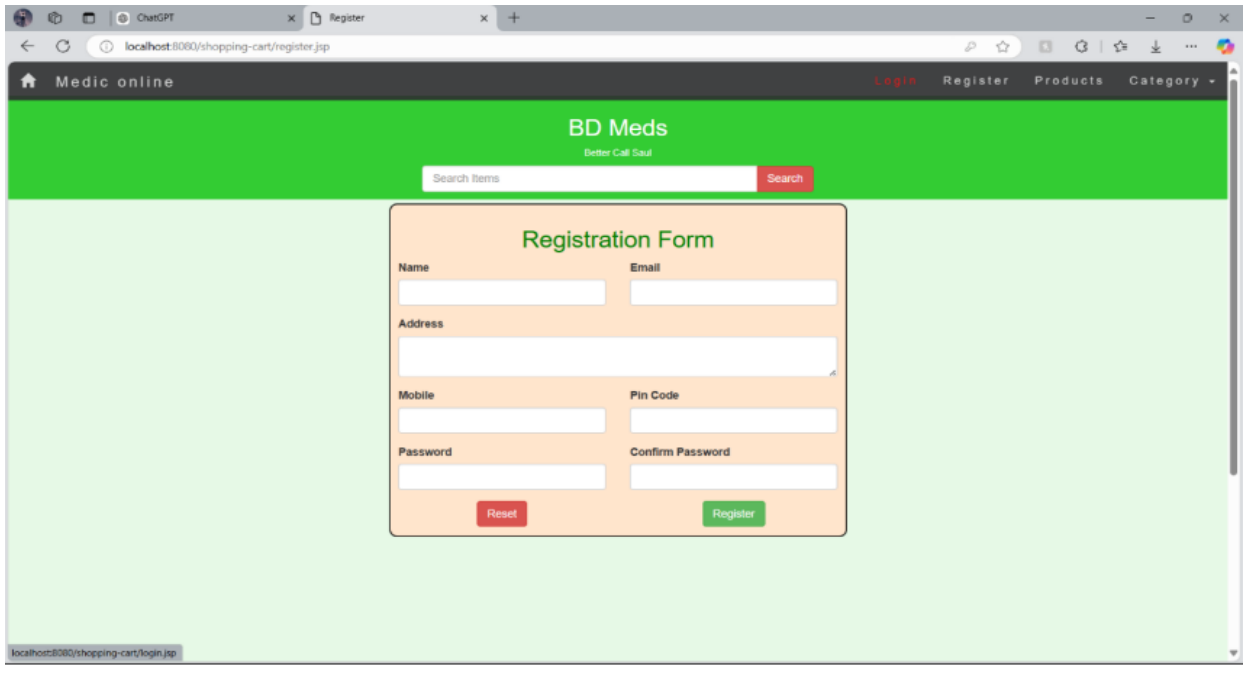

Figure 4.1.2 Registration page

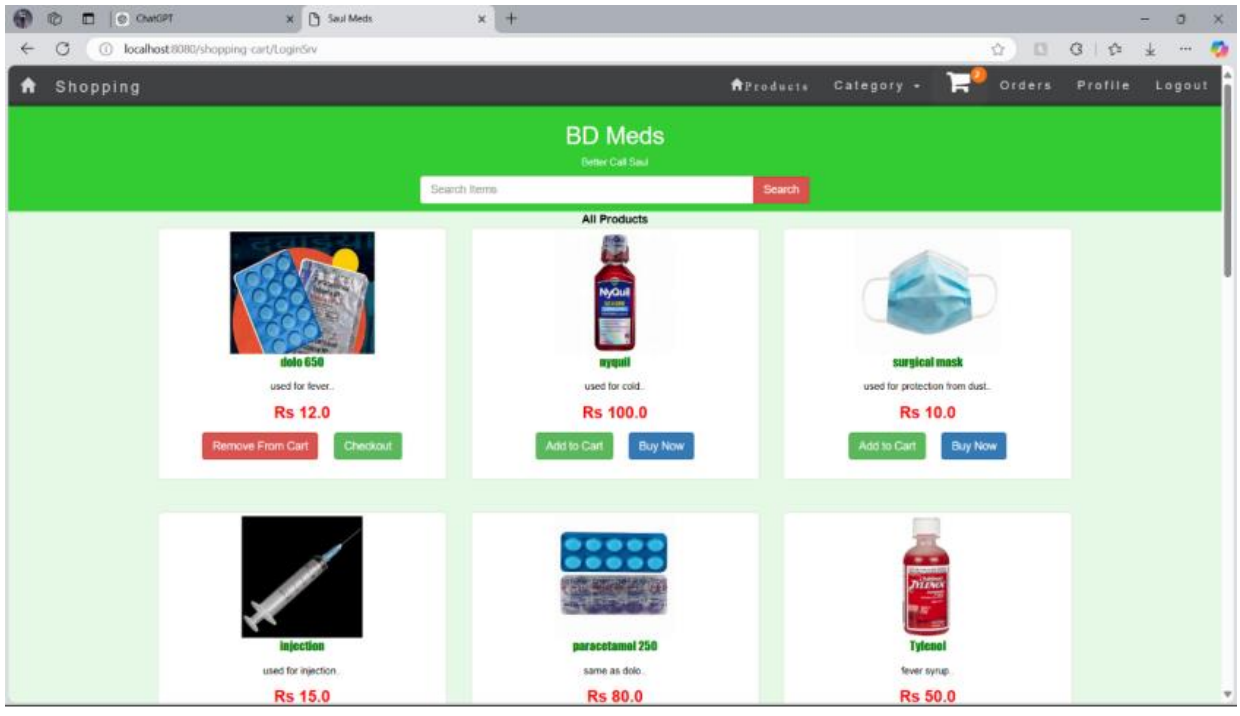

Figure 4.1.3 Home page

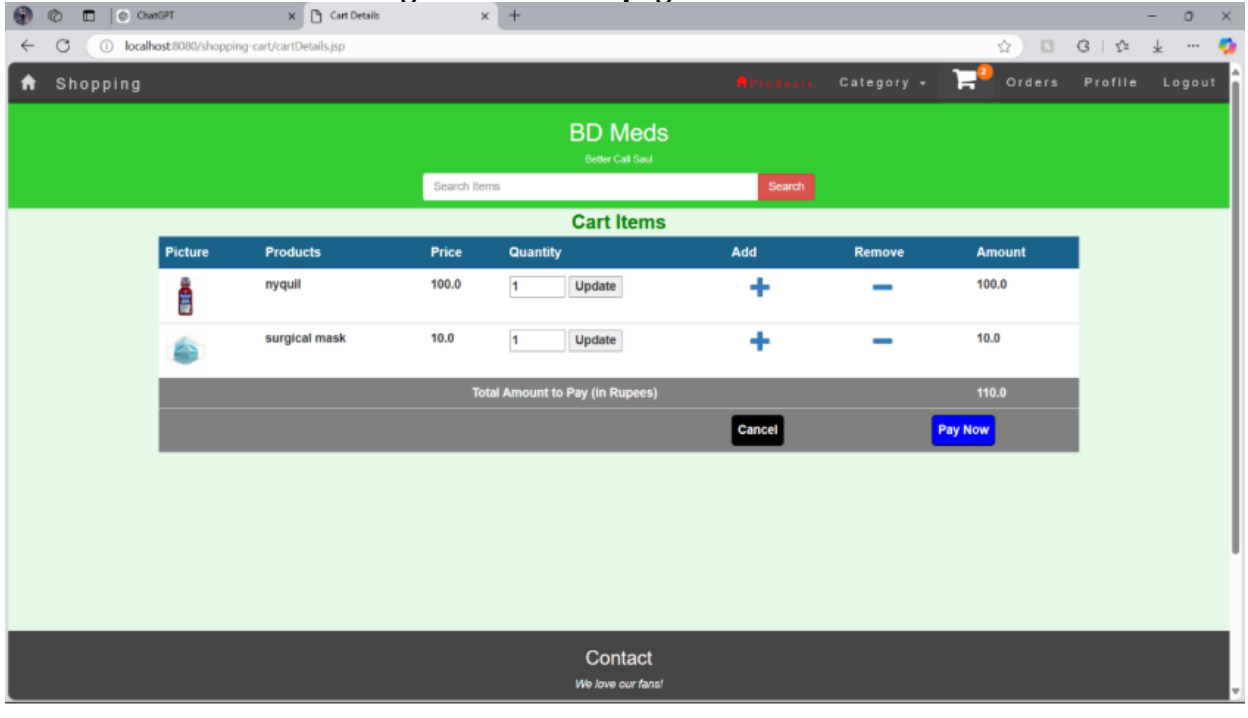

Figure 4.1.4 Cart Selection

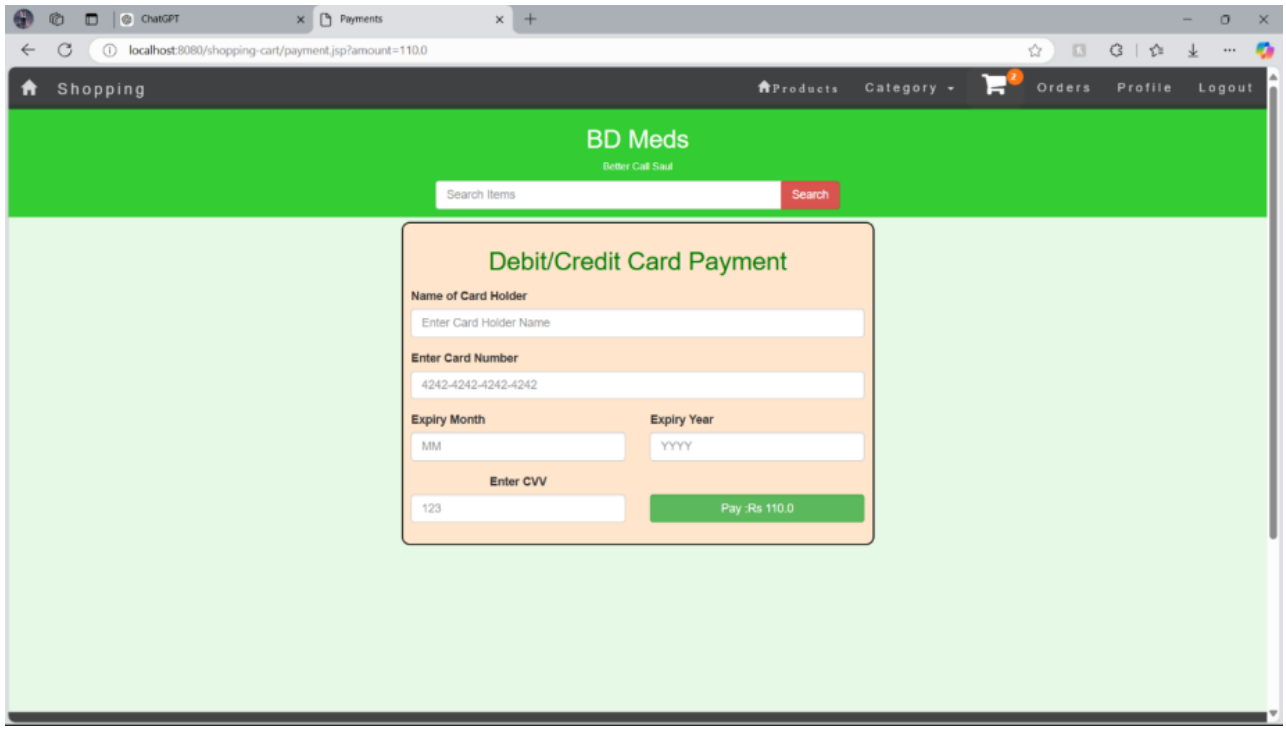

Figure 4.1.5 Payment page

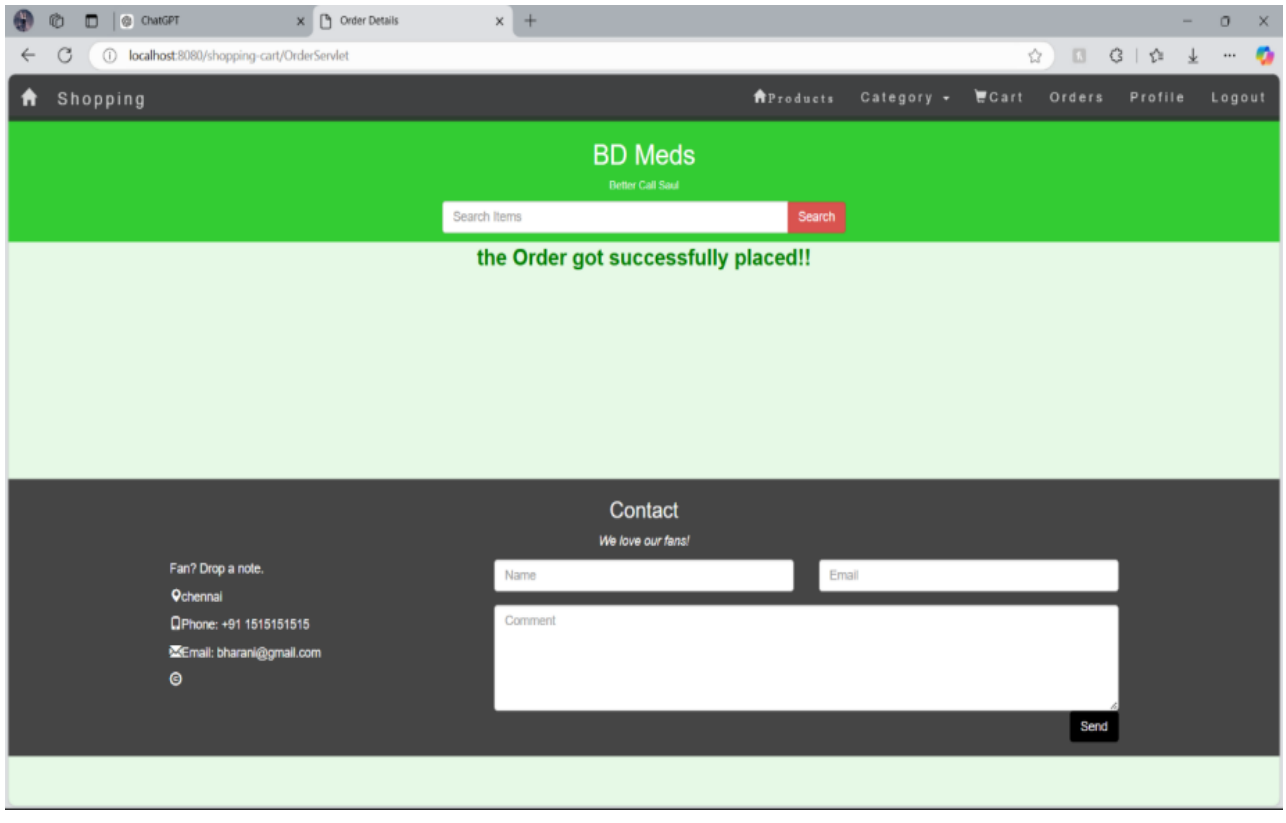

Figure 4.1.6 Order Message

4.2 Database Design

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability.

Medical E-Commerce Web Store which contains 4 MySQL tables :

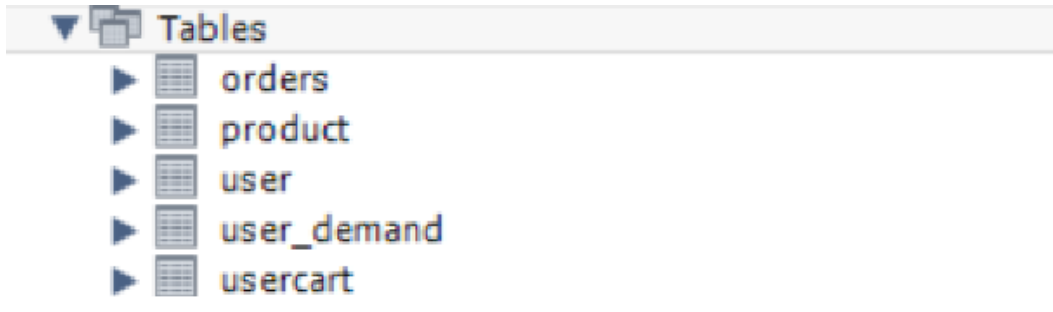

Table 4.2.1 Login Table

4.3 IMPLEMENTATIONS (CODE)

```
package com.shashi.srv;
import java.io.IOException;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import com.shashi.beans.UserBean;
import com.shashi.service.impl.UserServiceImpl;
/**
 * Servlet implementation class LoginSrv
 */
@WebServlet("/LoginSrv")
public class LoginSrv extends HttpServlet {
	private static final long serialVersionUID = 1L;

	public LoginSrv() {
			super();
	}
	protected void doGet(HttpServletRequest request, HttpServletResponse response)
					throws ServletException, IOException {
			String userName = request.getParameter("username");
			String password = request.getParameter("password");
			String userType = request.getParameter("usertype");
			response.setContentType("text/html");
			String status = "Login Denied! Invalid Username or password.";
			if (userType.equals("admin")) {
			if (password.equals("admin") && userName.equals("admin@gmail.com"))
{

	RequestDispatcher rd = request.getRequestDispatcher("adminViewProduct.jsp");
						HttpSession session = request.getSession();
						session.setAttribute("username", userName);
						session.setAttribute("password", password);
						session.setAttribute("usertype", userType);
						rd.forward(request, response);
					} else {
	RequestDispatcher rd = request.getRequestDispatcher("login.jsp?message=" + status);
						rd.include(request, response);
					}
			} else {
					UserServiceImpl udao = new UserServiceImpl();
					status = udao.isValidCredential(userName, password);
					if (status.equalsIgnoreCase("valid")) {
```

```
    resultPanel.setBackground(Color.WHITE);
    // Create the Reservation button panel with center alignment
    JPanel buttonPanel = new JPanel(new GridBagLayout());
    GridBagConstraints gbc = new GridBagConstraints();
    gbc.anchor = GridBagConstraints.CENTER;
    buttonPanel.add(createReservationButton(), gbc);

    frame.add(mainPanel); frame.add(topPanel,
    BorderLayout.NORTH); frame.add(resultPanel,
    BorderLayout.CENTER); frame.add(buttonPanel,
    BorderLayout.SOUTH);


    frame.pack();
    frame.setVisible(tr
    ue);
    setupDatabaseConnection();
}

private JPanel createTopPanel() {


   JPanel topPanel = new JPanel(new FlowLayout());
   topPanel.setBackground(Color.LIGHT_GRAY); // Set background color

    placeComboBox = new JComboBox<>();
    placeComboBox.addItem("None");
    placeComboBox.addItem("Chennai");
    placeComboBox.addItem("Coimbatore");
    placeComboBox.setSelectedItem("None");

    theatreComboBox = new JComboBox<>();
```

```
UserBean user = udao.getUserDetails(userName, password);
        HttpSession session = request.getSession();
        Session.setAttribute("userdata", user);
        session.setAttribute("username", userName);
        session.setAttribute("password", password);
        session.setAttribute("usertype", userType);
RequestDispatcher rd = request.getRequestDispatcher("userHome.jsp");
        rd.forward(request, response);
        } else {
RequestDispatcher rd =request.getRequestDispatcher("login.jsp?message=" + status);
        rd.forward(request, response);
                                        }
                        }
        }
protected void doPost(HttpServletRequest request, HttpServletResponse response)
hrows ServletException, IOException {
                        doGet(request, response);
        }

}
```

```
import com.shashi.service.impl.OrderServiceImpl;
/**
* Servlet implementation class OrderServlet
*/
@WebServlet("/OrderServlet")
public class OrderServlet extends HttpServlet {
private static final long serialVersionUID = 1L;
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
HttpSession session = request.getSession();
String userName = (String) session.getAttribute("username");
String password = (String) session.getAttribute("password");
if (userName == null || password == null) {
response.sendRedirect("login.jsp?message=Session Expired, Login Again!!");
}
double paidAmount = Double.parseDouble(request.getParameter("amount"));
String status = new OrderServiceImpl().paymentSuccess(userName, paidAmount);
PrintWriter pw = response.getWriter();
response.setContentType("text/html");
RequestDispatcher rd = request.getRequestDispatcher("orderDetails.jsp");
rd.include(request, response);
}
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
doGet(request, response);
}
}
```

```
package com.shashi.srv;
import java.io.IOException;
import java.io.InputStream;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import javax.servlet.http.Part;
import com.shashi.service.impl.ProductServiceImpl;
/**
 * Servlet implementation class AddProductSrv
 */
@WebServlet("/AddProductSrv")
@MultipartConfig(maxFileSize = 16177215)
public class AddProductSrv extends HttpServlet {
		private static final long serialVersionUID = 1L;
		protected void doGet(HttpServletRequest request, HttpServletResponse
response)
					throws ServletException, IOException {
				HttpSession session = request.getSession();
				String userType = (String) session.getAttribute("usertype");
				String userName = (String) session.getAttribute("username");
				String password = (String) session.getAttribute("password");
				if (userType == null || !userType.equals("admin")) {
				response.sendRedirect("login.jsp?message=Access Denied!");
				}
				else if (userName == null || password == null) {
		response.sendRedirect("login.jsp?message=Session Expired, Login Again to
Continue!");
				}
				String status = "Product Registration Failed!";
				String prodName = request.getParameter("name");
				String prodType = request.getParameter("type");
				String prodInfo = request.getParameter("info");
				double prodPrice =
Double.parseDouble(request.getParameter("price"));
				int prodQuantity =
Integer.parseInt(request.getParameter("quantity"));
				Part part = request.getPart("image");
				InputStream inputStream = part.getInputStream();
				InputStream prodImage = inputStream;
				ProductServiceImpl product = new ProductServiceImpl();
```

5.CONCLUSION

A Medical e-commerce website plays a crucial role in enhancing accessibility to healthcare products and services. By offering a convenient, secure platform for purchasing medical supplies, pharmaceuticals, and health-related products, it helps meet the growing demand for online healthcare solutions. With user-friendly interfaces, reliable delivery systems, and strong data security measures, such websites can greatly improve the patient experience, foster trust, and contribute to better health outcomes. Ensuring compliance with regulations and maintaining high standards of product quality will be key to long-term success in the medical e-commerce space.

6. REFERENCES

1 Reference link for java-awt at javatpoint : https://www.javatpoint.com/java-awt

2 Reference link for java-swing at javatpoint : https://www.javatpoint.com/java-swing