Medicare .
Course-end Project 4

DESCRIPTION

Create a dynamic and responsive Java e-healthcare web application for ordering medicines of different categories.

Background of the problem statement:
Medicare is a company that supplies medicines and a couple of other healthcare essentials at an affordable price. It was established in 2012 in Delhi, India. It had been serving fine all these years, however, the business analysts noticed a decline in sales since 2017. They found out that online ordering of medicines with companies, such as 100mg and mfine are gaining more profits by eliminating middlemen from the equation. As a result, the team decided to hire a Full Stack developer to develop a healthcare web application with a rich and user-friendly interface.
You are hired as the Full Stack Java developer and are asked to develop the web application. The management team has provided you with the requirements and their business model so that you can easily arrange different components of the application.

Features of the application:

1. Registration
2. Login
3. Payment gateway
4. Searching
5. Filtering
6. Sorting
7. Dynamic data
8. Responsive and compatible with different devices

Recommended technologies:

1. Database management: MySQL and Oracle
2. Backend logic: Java programming, NodeJS
3. Frontend development: JSP, Angular, Bootstrap, HTML/CSS, and Javascript
4. Automation and testing technologies: Selenium, Jasmine (frontend testing), and TestNG
5. DevOps and production technologies: Git, GitHub, Jenkins, Docker, Kubernetes, and AWS

Project development guidelines:

- The project will be delivered within four sprints with every sprint delivering a minimal viable product.
- It is mandatory to perform proper sprint planning with user stories to develop all the components of the project.
- The learner can use any technology from the above-mentioned technologies for different layers of the project.
- The web application should be responsive and should fetch or send data dynamically without hardcoded values.
- The learner must maintain the version of the application over GitHub and every new change should be sent to the repository.
- The learner must implement a CI/CD pipeline using Jenkins.
- The learner should also deploy and host the application on an AWS EC2 instance.
- The learner should also implement automation testing before the application enters the CI/CD pipeline.
- The learner should use Git branching to do basic automation testing of the application in it separately.
- The learner should make a rich frontend of the application, which is user-friendly and easy for the user to navigate through the application.
- There will be two portals in the application, namely admin and user portal.

Admin Portal:
The admin portal deals with all the backend data generation and product information. The admin user should be able to:

- Add or remove medicine details from the application to build a rich product line
- Edit medicine details like name, price, seller, product description, and offers to keep the product information updated with the current prices
- Enable or disable a medicine product

User Portal:
It deals with the user activities. The end-user should be able to:

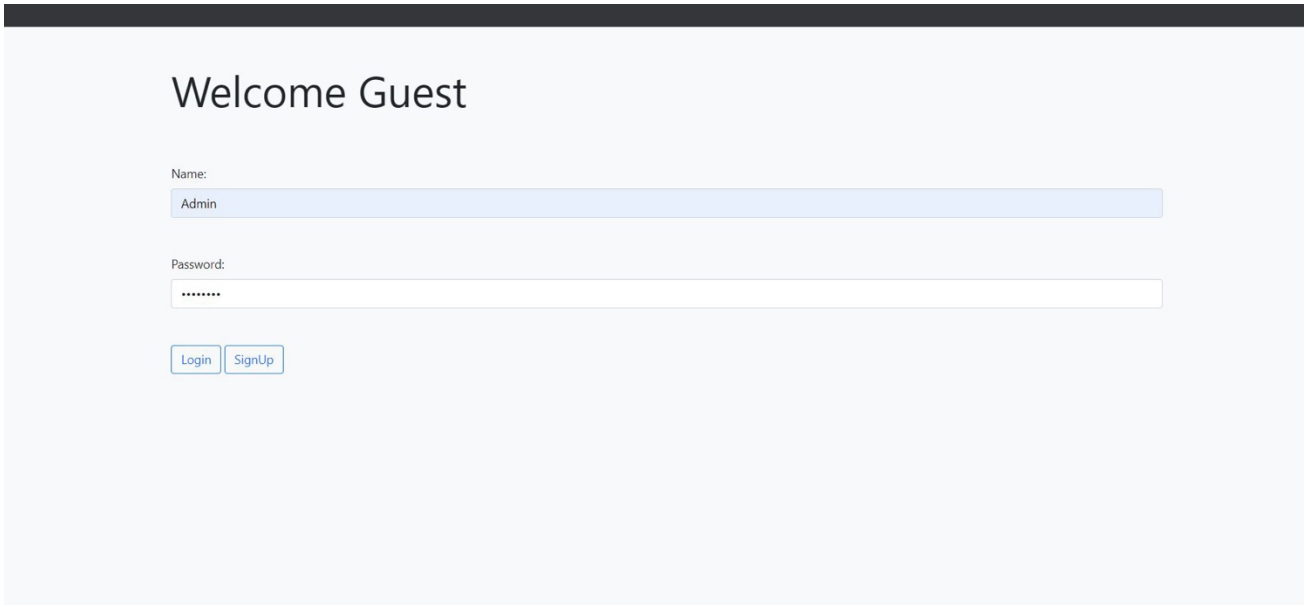
- Sign-in to the application to maintain a record of activities
- Search for products based on the search keyword
- Apply filters and sort results based on different cuisines to get the best deals
- Add all the selected food items to the cart and customize the purchase at the end
- Perform a seamless payment gateway
- Get an order summary details page once the payment is complete

Screenshots

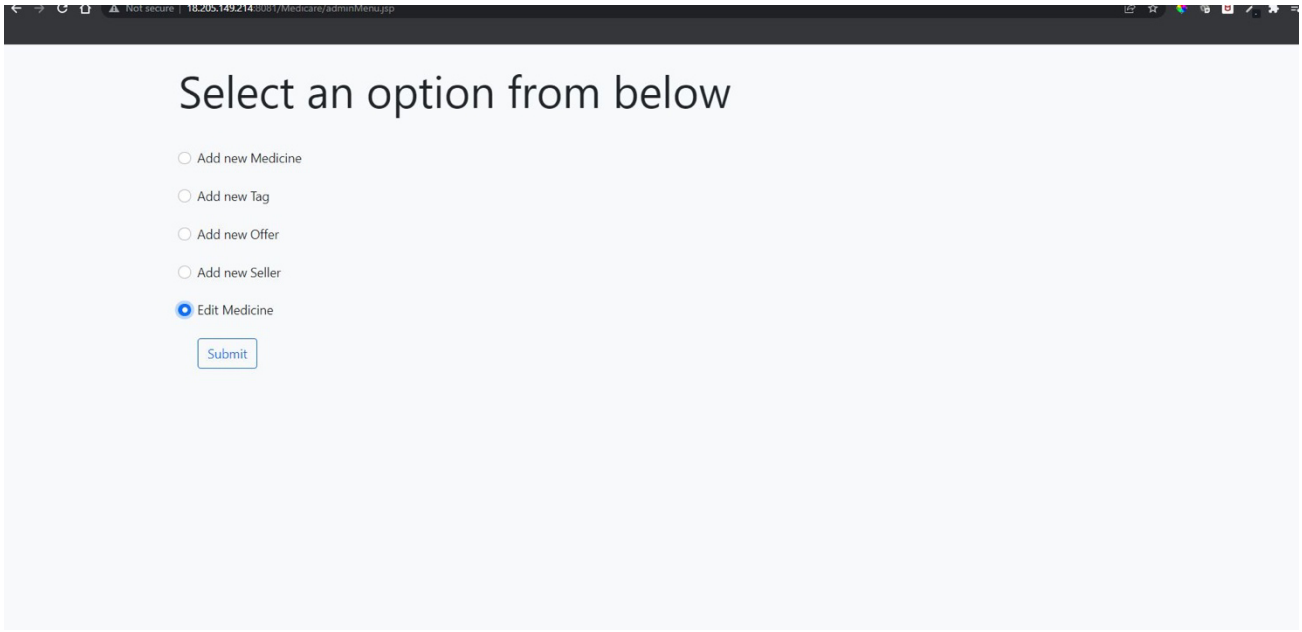
Admin Portal:

The admin portal deals with all the backend data generation and product information. The admin user should be able to:

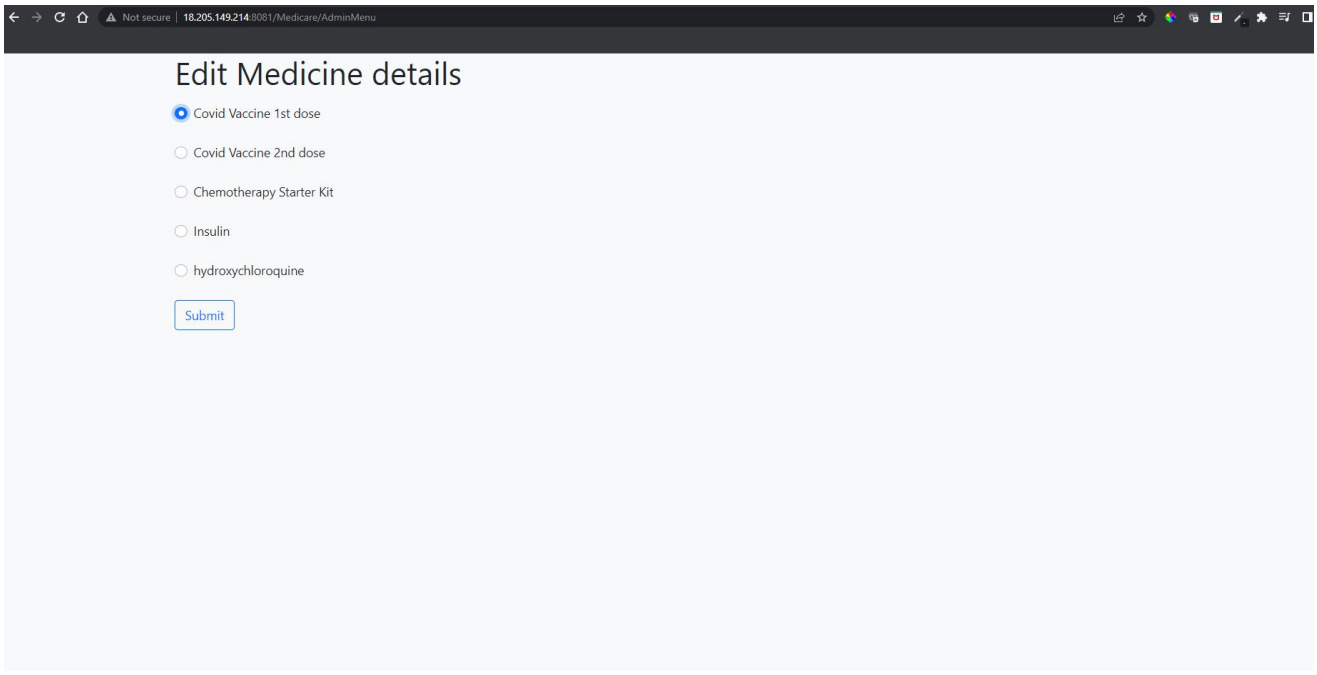
- Add or remove medicine details from the application to build a rich product line
- Edit medicine details like name, price, seller, product description, and offers to keep the product information updated with the current prices
- Enable or disable a medicine product

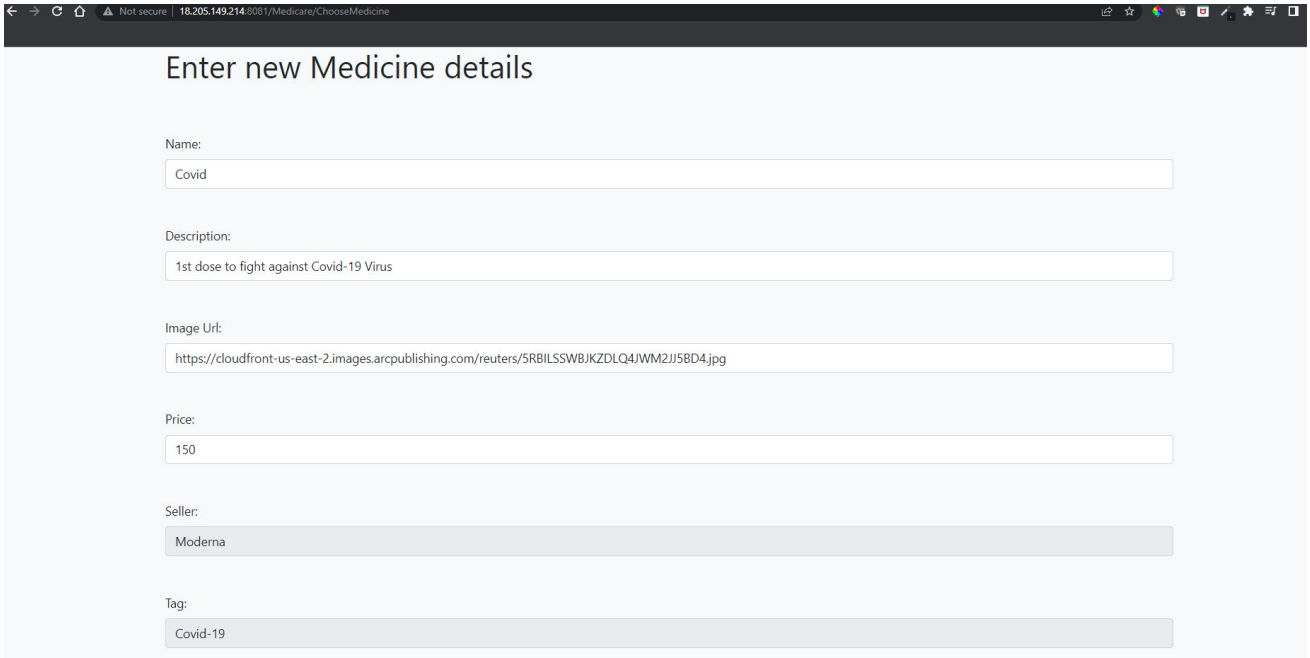


Below includes relevant Admin operations like Add, Edit Medicines, add Sellers, Offers etc.



- Edit medicine details like name, price, seller, product description, and offers to keep the product information updated with the current prices

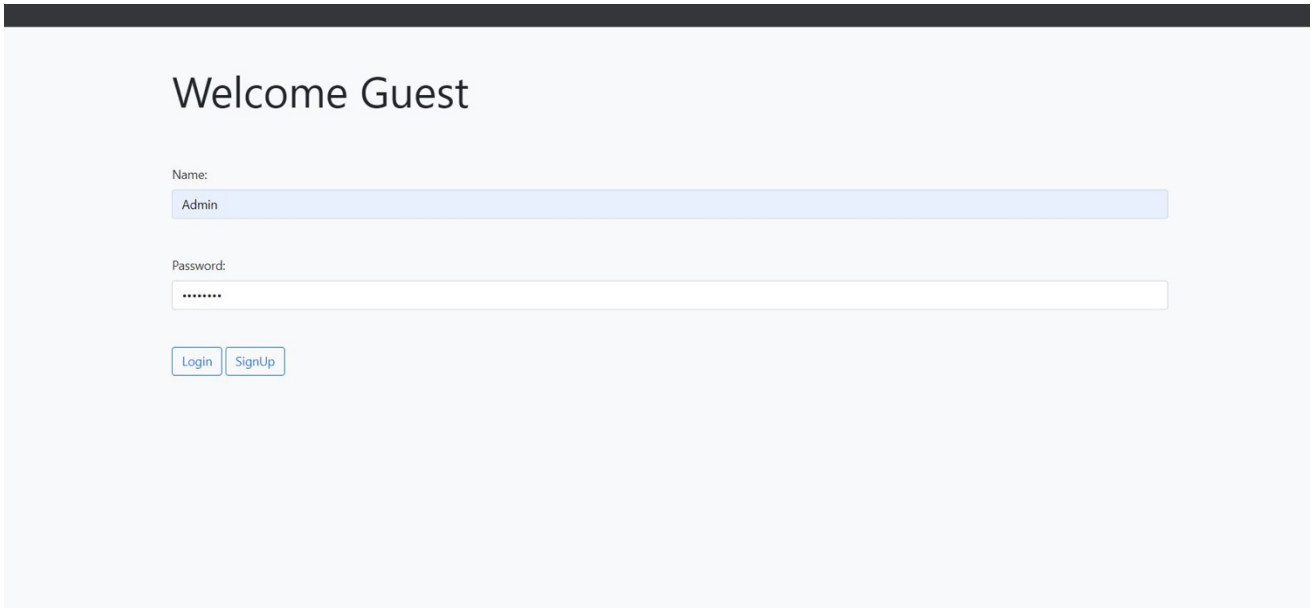


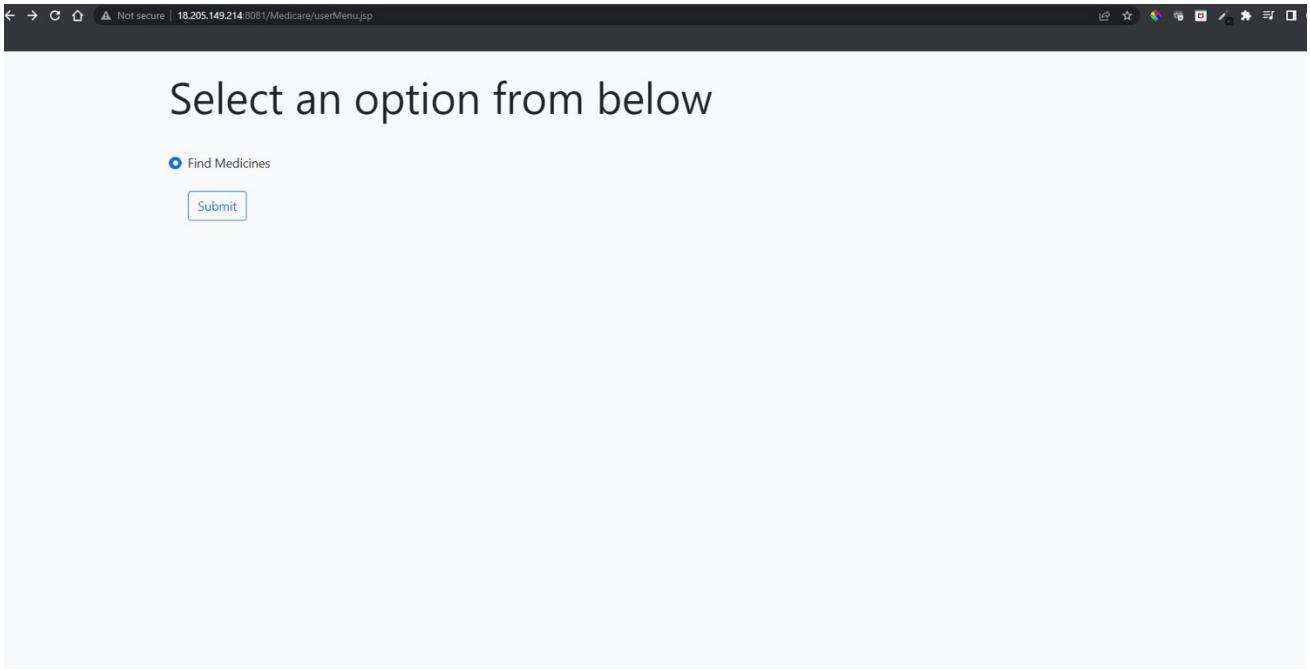


User Portal:
It deals with the user activities. The end-user should be able to:

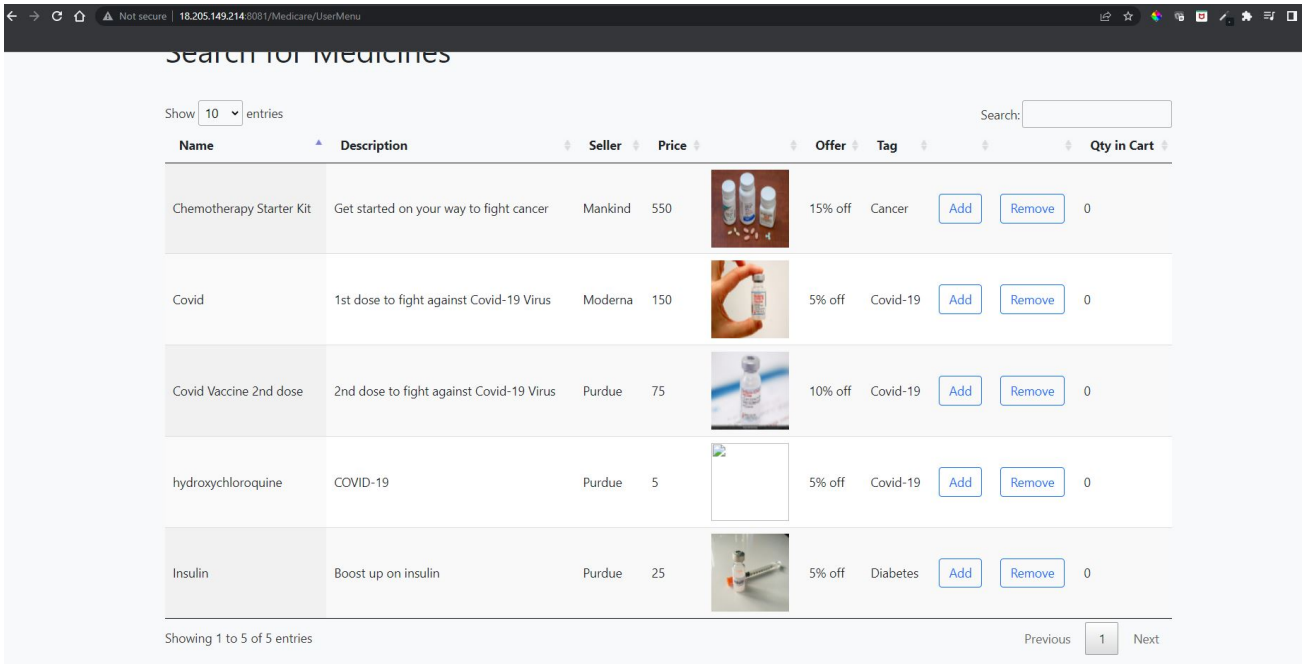
- Sign-in to the application to maintain a record of activities
- Search for products based on the search keyword
- Apply filters and sort results based on different cuisines to get the best deals
- Add all the selected food items to the cart and customize the purchase at the end
- Perform a seamless payment gateway
- Get an order summary details page once the payment is complete

Again I have considered Admin as user to test the further screen.

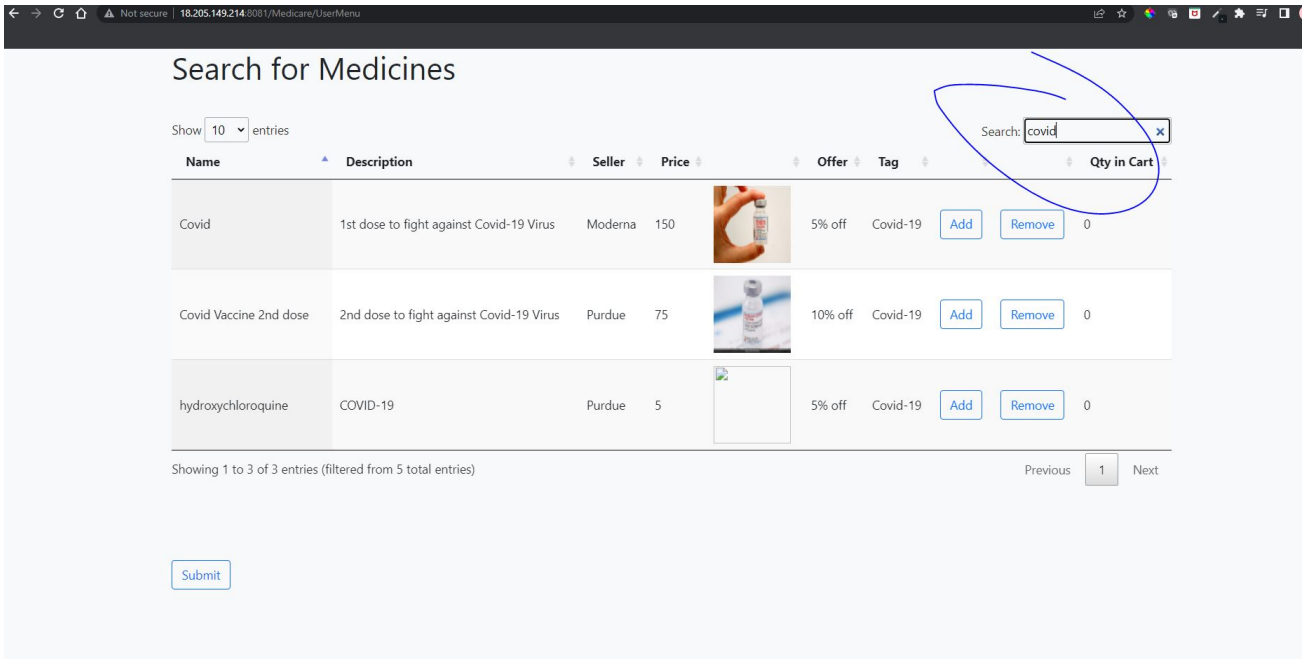




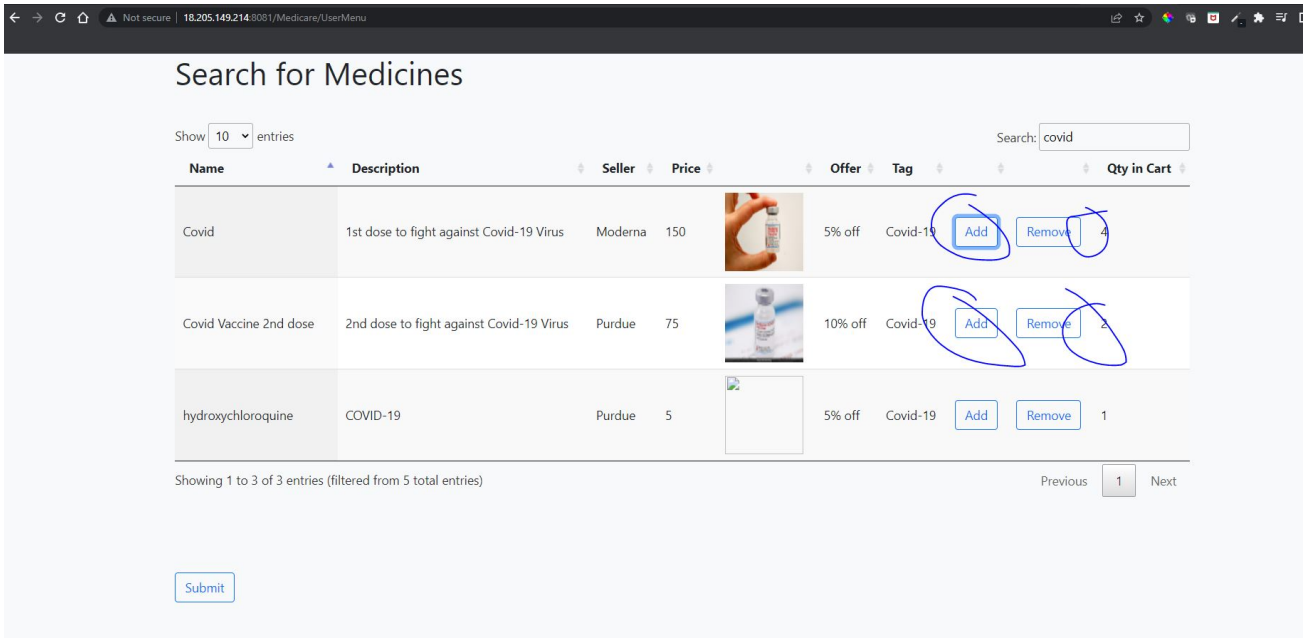
- User Searches for products based on the search keyword as Shown below.



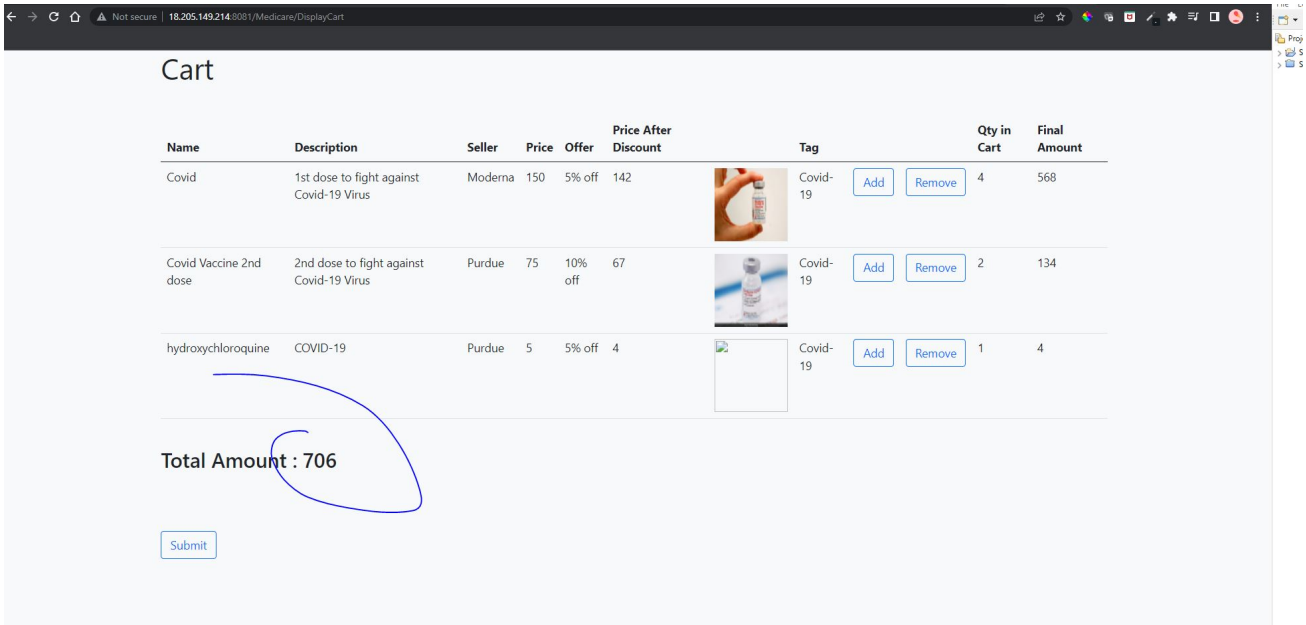
User uses Covid as keyword to search for relevant medicine as shown in the below screenshot.



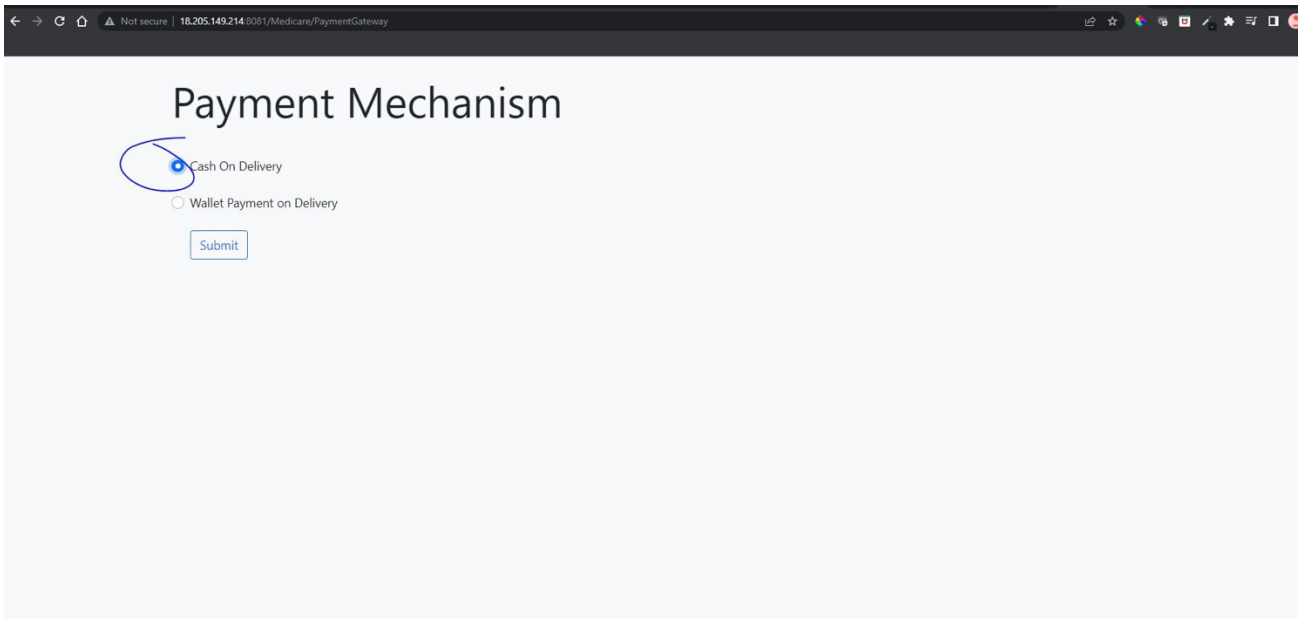
As per the keywork Covid, 3 relevant medicines are displayed for user to add/Remove



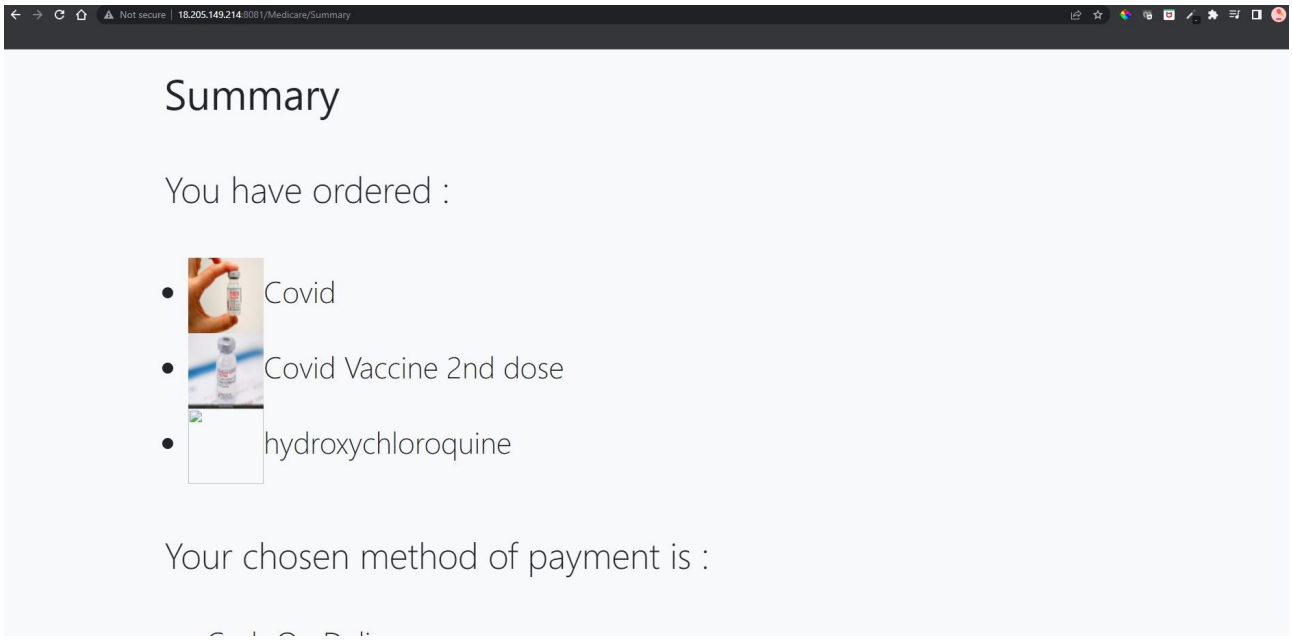
User selected all the 3 Medicines and added to Cart as shown in the screenshot. The system calculates total price asks user to confirm by clicking on Submit button for payment.

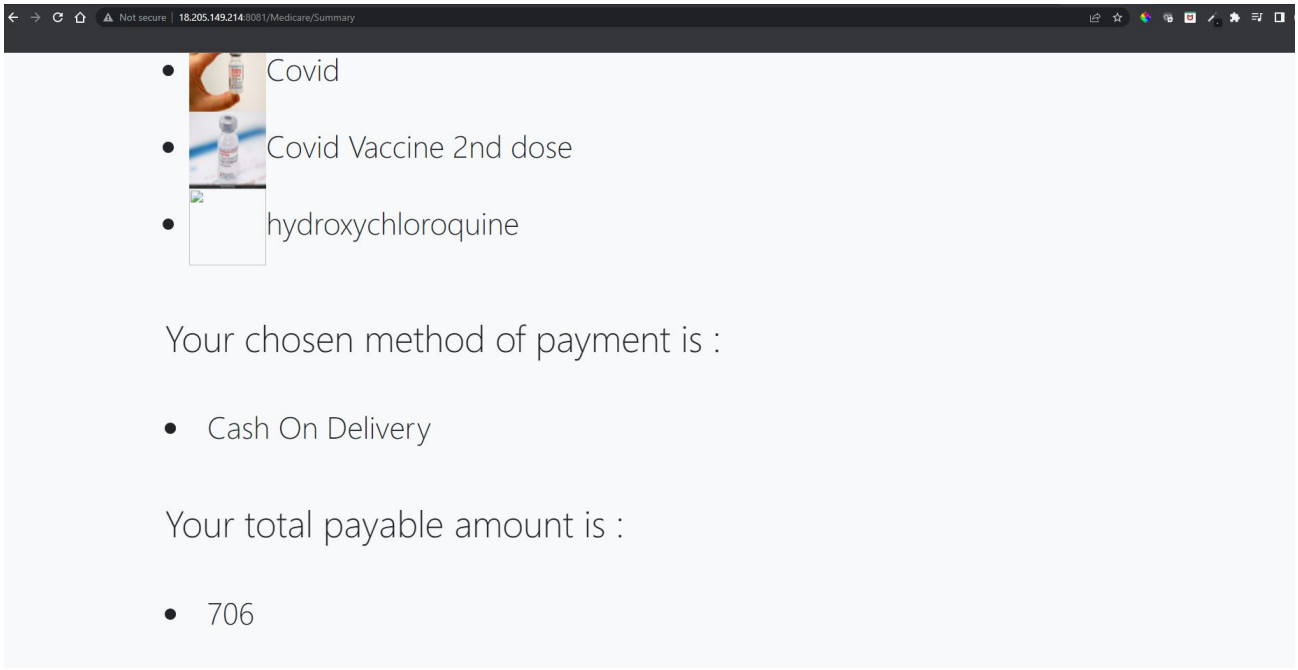


Post clicking on Submit user will be given with 2 Payment options. Cash on Delivery and Wallet Payment on Delivery.



After relevant Payment mechanism selection in the previous step, the Summary page will be shown and Order Confirmation provided.





Source Code

userSignUp.java

```
package service;
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import model.Seller;

public class UserSignup extends HttpServlet {

	private static final long serialVersionUID = 1L;
	private Database database = new Database();

	protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
		response.sendRedirect("index.jsp");
	}
```

```
	public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

		response.setContentType("text/html");
		PrintWriter out = response.getWriter();

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

		if (database.addUser(username, password)) {
			out.print("SignUp Successful");
			response.sendRedirect("userMenu.jsp");
		} else {
			out.print("Error occurred.");
			RequestDispatcher rd = request.getRequestDispatcher("index.jsp");
			rd.include(request, response);
		}

		out.close();
	}

}
```

UserLogin.java

```
package service;
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class UserLogin extends HttpServlet {

	private static final long serialVersionUID = 1L;
	private Database database = new Database();

	protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
		response.sendRedirect("index.jsp");
	}

	public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

		response.setContentType("text/html");
		PrintWriter out = response.getWriter();
```

```
		String username = request.getParameter("username");
		String password = request.getParameter("userpass");

		int loginResponse = database.verifyLogin(username, password);
		if (loginResponse == 2) {
			response.sendRedirect("adminMenu.jsp");
		} else if(loginResponse == 1) {
			response.sendRedirect("userMenu.jsp");
		} else {
			out.print("Entered Credentials are incorrect.");
			RequestDispatcher rd = request.getRequestDispatcher("index.jsp");
			rd.include(request, response);
		}

		out.close();
	}

}
```

UserMenu.java

```
package service;
import java.io.IOException;
import java.util.Map;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import model.*;

public class UserMenu extends HttpServlet {

	private static final long serialVersionUID = 1L;
	private Database database = new Database();

	protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
		response.sendRedirect("index.jsp");
	}

	public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

		Map<Integer, Medicine> medicineMap = database.getAllMedicines();
		Map<Integer, Tag> tagMap = database.getAllTags();
		Map<Integer, Offer> offerMap = database.getAllOffers();
```

```
		Map<Integer, Seller> sellerMap = database.getAllSellers();

		response.setContentType("text/html");

		String choice = request.getParameter("adminOption");

		RequestDispatcher rd = request.getRequestDispatcher("/adminMenu.jsp");
		rd.include(request, response);

		switch (choice) {

		case "search_medicine":
			request.setAttribute("tagMap", tagMap);
			request.setAttribute("offerMap", offerMap);
			request.setAttribute("sellerMap", sellerMap);
			request.setAttribute("medicineMap", medicineMap);
			RequestDispatcher changePasswordDispatcher =
request.getRequestDispatcher("/searchMedicine.jsp");
			changePasswordDispatcher.forward(request, response);
			break;
		}
	}

}
```

chooseMedicine.java

```
package service;
import java.io.IOException;
import java.util.Map;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import model.*;

public class ChooseMedicine extends HttpServlet {

	private static final long serialVersionUID = 1L;
	private Database database = new Database();

	protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
		response.sendRedirect("index.jsp");
	}
```

```
	public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

		Map<Integer, Medicine> medicineMap = database.getAllMedicines();
		Map<Integer, Tag> tagMap = database.getAllTags();
		Map<Integer, Offer> offerMap = database.getAllOffers();
		Map<Integer, Seller> sellerMap = database.getAllSellers();

		response.setContentType("text/html");

		Integer choice = Integer.valueOf(request.getParameter("chosenMedicine"));

		RequestDispatcher rd = request.getRequestDispatcher("/adminMenu.jsp");
		rd.include(request, response);
		request.setAttribute("tagMap", tagMap);
		request.setAttribute("offerMap", offerMap);
		request.setAttribute("sellerMap", sellerMap);
		request.setAttribute("choice", choice);
		request.setAttribute("medicineMap", medicineMap);
		RequestDispatcher changePasswordDispatcher =
request.getRequestDispatcher("/editMedicineDetail.jsp");
		changePasswordDispatcher.forward(request, response);
	}

}
```

paymentGteway.java

```
package service;
import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;



public class PaymentGateway extends HttpServlet {

	private static final long serialVersionUID = 1L;
	protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
		response.sendRedirect("index.jsp");
	}

	public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
```

```
		response.setContentType("text/html");

		String cart=request.getParameter("cart");
		request.setAttribute("cart", cart);
		RequestDispatcher dispatcher =
request.getRequestDispatcher("/paymentGateway.jsp");
		dispatcher.forward(request, response);

	}
}
```

AdminMenu.java

```
package service;
import java.io.IOException;
import java.util.Map;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import model.*;

public class AdminMenu extends HttpServlet {

	private static final long serialVersionUID = 1L;
	private Database database = new Database();

	protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
		response.sendRedirect("index.jsp");
	}

	public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

		Map<Integer, Medicine> medicineMap = database.getAllMedicines();
		Map<Integer, Tag> tagMap = database.getAllTags();
		Map<Integer, Offer> offerMap = database.getAllOffers();
		Map<Integer, Seller> sellerMap = database.getAllSellers();

		response.setContentType("text/html");

		String choice = request.getParameter("adminOption");

		RequestDispatcher rd = request.getRequestDispatcher("/adminMenu.jsp");
```

```
                rd.include(request, response);

                switch (choice) {

                case "add_medicine":
                        request.setAttribute("tagMap", tagMap);
                        request.setAttribute("offerMap", offerMap);
                        request.setAttribute("sellerMap", sellerMap);
                        RequestDispatcher changePasswordDispatcher =
request.getRequestDispatcher("/addMedicine.jsp");
                        changePasswordDispatcher.forward(request, response);
                        break;
                case "add_tag":
                        changePasswordDispatcher =
request.getRequestDispatcher("/addTag.jsp");
                        changePasswordDispatcher.forward(request, response);
                        break;
                case "add_offer":
                        changePasswordDispatcher =
request.getRequestDispatcher("/addOffer.jsp");
                        changePasswordDispatcher.forward(request, response);
                        break;
                case "add_seller":
                        changePasswordDispatcher =
request.getRequestDispatcher("/addSeller.jsp");
                        changePasswordDispatcher.forward(request, response);
                        break;
                case "edit_medicine":
                        request.setAttribute("tagMap", tagMap);
                        request.setAttribute("offerMap", offerMap);
                        request.setAttribute("sellerMap", sellerMap);
                        request.setAttribute("medicineMap", medicineMap);
                        changePasswordDispatcher =
request.getRequestDispatcher("/editMedicine.jsp");
                        changePasswordDispatcher.forward(request, response);
                        break;
                }
        }

}
```

AddMedicine.java

```
package service;
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
```

```
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import model.Medicine;

public class AddMedicine extends HttpServlet {

	private static final long serialVersionUID = 1L;
	private final Database database = new Database();

	protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
		response.sendRedirect("index.jsp");
	}

	public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
		response.setContentType("text/html");
		PrintWriter out = response.getWriter();

		String name = request.getParameter("name");
		String description = request.getParameter("description");
		int price = Integer.valueOf(request.getParameter("price"));
		int sellerId = Integer.valueOf(request.getParameter("seller"));
		int tagId = Integer.valueOf(request.getParameter("tag"));
		int offerId = Integer.valueOf(request.getParameter("offer"));
		String isActiveString = request.getParameter("active");
		String imgUrl = request.getParameter("imageUrl");
		boolean isActive;
		if(isActiveString==null) {
			isActive=false;
		}else {
			isActive=true;
		}
		if (database.addMedicine(new
Medicine(name,description,sellerId,price,offerId,tagId,isActive,imgUrl))) {
			out.print("Medicine added successfully");
		} else {
			out.print("Error occurred.");
		}

		RequestDispatcher rd = request.getRequestDispatcher("adminMenu.jsp");
		rd.include(request, response);

		out.close();
	}
}
```

EditMedicine.java

```
package service;
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import model.Medicine;

public class EditMedicine extends HttpServlet {

	private static final long serialVersionUID = 1L;
	private final Database database = new Database();

	protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
		response.sendRedirect("index.jsp");
	}

	public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
		response.setContentType("text/html");
		PrintWriter out = response.getWriter();

		Integer id = Integer.valueOf(request.getParameter("id"));
		String name = request.getParameter("name");
		String description = request.getParameter("description");
		int price = Integer.valueOf(request.getParameter("price"));
		int sellerId = Integer.valueOf(request.getParameter("seller"));
		int tagId = Integer.valueOf(request.getParameter("tag"));
		int offerId = Integer.valueOf(request.getParameter("offer"));
		String isActiveString = request.getParameter("active");
		String imageUrl = request.getParameter("imageUrl");
		System.out.println("uttu " + imageUrl);
		boolean isActive;
		if(isActiveString==null) {
			isActive=false;
		}else {
			isActive=true;
		}
		if (database.editMedicine(new
Medicine(id,name,description,sellerId,price,offerId,tagId,isActive,imageUrl))) {
			out.print("Medicine edited successfully");
		} else {
			out.print("Error occurred.");
```

```
			}

			RequestDispatcher rd = request.getRequestDispatcher("adminMenu.jsp");
			rd.include(request, response);

			out.close();
		}
}
```

Database.java

```
package service;

import java.util.HashMap;
import java.util.Map;


import constant.Queries;
import model.Medicine;
import model.Offer;
import model.Seller;
import model.Tag;
import java.sql.*;

public class Database {

	public Map<Integer, Medicine> getAllMedicines() {
		Map<Integer, Medicine> medicineMap = new
HashMap<Integer,Medicine>();
		try{
			Class.forName("com.mysql.cj.jdbc.Driver");
			Connection con=DriverManager.getConnection(
			"jdbc:mysql://localhost:3306/medicare","ostl","ostl");
			Statement stmt=con.createStatement();
			ResultSet rs=stmt.executeQuery(Queries.GET_ALL_MEDICINES);
			while(rs.next()) {
				medicineMap.put(rs.getInt(1), new
Medicine(rs.getInt(1),rs.getString(2),rs.getString(3),rs.getInt(4),rs.getInt(5),rs.getInt(6),rs.get
Int(7),rs.getBoolean(8),rs.getString(9)));
			}
			con.close();
			} catch(Exception e) {
				System.out.println(e);
			}
		return medicineMap;
	}

	public Map<Integer, Tag> getAllTags() {
		Map<Integer, Tag> tagMap = new HashMap<Integer,Tag>();
		try{
```

```
                        Class.forName("com.mysql.cj.jdbc.Driver");
                        Connection con=DriverManager.getConnection(
                        "jdbc:mysql://localhost:3306/medicare","ostl","ostl");
                        Statement stmt=con.createStatement();
                        ResultSet rs=stmt.executeQuery(Queries.GET_ALL_TAGS);
                        while(rs.next()) {
                                tagMap.put(rs.getInt(1), new Tag(rs.getInt(1), rs.getString(2)));

                        }
                        con.close();
                        } catch(Exception e){
                                System.out.println(e);
                        }
                return tagMap;
        }

        public Map<Integer, Seller> getAllSellers() {
                Map<Integer, Seller> sellerMap = new HashMap<Integer,Seller>();
                try{
                        Class.forName("com.mysql.cj.jdbc.Driver");
                        Connection con=DriverManager.getConnection(
                        "jdbc:mysql://localhost:3306/medicare","ostl","ostl");
                        Statement stmt=con.createStatement();
                        ResultSet rs=stmt.executeQuery(Queries.GET_ALL_SELLERS);
                        while(rs.next()) {
                                sellerMap.put(rs.getInt(1), new Seller(rs.getInt(1),
rs.getString(2)));
                        }
                        con.close();
                        } catch(Exception e){
                                System.out.println(e);
                        }
                return sellerMap;
        }

        public Map<Integer, Offer> getAllOffers() {
                Map<Integer, Offer> offerMap = new HashMap<Integer,Offer>();
                try{
                        Class.forName("com.mysql.cj.jdbc.Driver");
                        Connection con=DriverManager.getConnection(
                        "jdbc:mysql://localhost:3306/medicare","ostl","ostl");
                        Statement stmt=con.createStatement();
                        ResultSet rs=stmt.executeQuery(Queries.GET_ALL_OFFERS);
                        while(rs.next()) {
                                offerMap.put(rs.getInt(1), new Offer(rs.getInt(1),
rs.getString(2), rs.getInt(3)));
                        }
                        con.close();
                        } catch(Exception e){
                                System.out.println(e);
```

```
			}
		return offerMap;
	}

	//0->not a valid user
	//1->normal user
	//2->admin
	public int verifyLogin(String username, String password) {
		int userStatus = 0;
		try{
			Class.forName("com.mysql.cj.jdbc.Driver");
			Connection con=DriverManager.getConnection(
			"jdbc:mysql://localhost:3306/medicare","ostl","ostl");
			PreparedStatement
stmt=con.prepareStatement(Queries.VERIFY_LOGIN);
			stmt.setString(1,username);
			stmt.setString(2, password);

			ResultSet rs=stmt.executeQuery();
			if(rs.next()) {
				userStatus = 1;
				if(rs.getBoolean(1)==true) {
					userStatus=2;
				}
			}
			con.close();
			} catch(Exception e){
				System.out.println(e);
			}
		return userStatus;
	}

	public boolean addUser(String username, String password) {
		boolean isSuccess = false;
		try{
			Class.forName("com.mysql.cj.jdbc.Driver");
			Connection con=DriverManager.getConnection(
			"jdbc:mysql://localhost:3306/medicare","ostl","ostl");
			PreparedStatement
stmt=con.prepareStatement(Queries.INSERT_USER);
			stmt.setString(1,username);
			stmt.setString(2, password);

			int result = stmt.executeUpdate();
			if(result > 0) {
				isSuccess=true;
			}
			con.close();
			} catch(Exception e){
				System.out.println(e);
```

```
                }
            return isSuccess;
        }

        public boolean addMedicine(Medicine medicine) {
            boolean isSuccess = false;
            try{
                Class.forName("com.mysql.cj.jdbc.Driver");
                Connection con=DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/medicare","ostl","ostl");
                PreparedStatement
stmt=con.prepareStatement(Queries.INSERT_MEDICINE);
                stmt.setString(1,medicine.getName());
                stmt.setString(2, medicine.getDescription());
                stmt.setInt(3, medicine.getSellerId());
                stmt.setInt(4, medicine.getPrice());
                stmt.setInt(5, medicine.getOfferId());
                stmt.setInt(6, medicine.getTagId());
                stmt.setBoolean(7, medicine.isActive());
                stmt.setString(8, medicine.getImageUrl());

                int result = stmt.executeUpdate();
                if(result > 0) {
                    isSuccess=true;
                }
                con.close();
                } catch(Exception e){
                    System.out.println(e);
                }
            return isSuccess;
        }

        public boolean addTag(Tag tag) {
            boolean isSuccess = false;
            try{
                Class.forName("com.mysql.cj.jdbc.Driver");
                Connection con=DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/medicare","ostl","ostl");
                PreparedStatement
stmt=con.prepareStatement(Queries.INSERT_TAG);
                stmt.setString(1,tag.getName());

                int result = stmt.executeUpdate();
                if(result > 0) {
                    isSuccess=true;
                }
                con.close();
                } catch(Exception e){
                    System.out.println(e);
                }
```

```
		return isSuccess;
	}

	public boolean addOffer(Offer offer) {
		boolean isSuccess = false;
		try{
			Class.forName("com.mysql.cj.jdbc.Driver");
			Connection con=DriverManager.getConnection(
			"jdbc:mysql://localhost:3306/medicare","ostl","ostl");
			PreparedStatement
stmt=con.prepareStatement(Queries.INSERT_OFFER);
			stmt.setString(1,offer.getName());
			stmt.setInt(2,offer.getValueInDiscount());

			int result = stmt.executeUpdate();
			if(result > 0) {
				isSuccess=true;
			}
			con.close();
			} catch(Exception e){
				System.out.println(e);
			}
		return isSuccess;
	}

	public boolean addSeller(Seller seller) {
		boolean isSuccess = false;
		try{
			Class.forName("com.mysql.cj.jdbc.Driver");
			Connection con=DriverManager.getConnection(
			"jdbc:mysql://localhost:3306/medicare","ostl","ostl");
			PreparedStatement
stmt=con.prepareStatement(Queries.INSERT_SELLER);
			stmt.setString(1,seller.getName());

			int result = stmt.executeUpdate();
			if(result > 0) {
				isSuccess=true;
			}
			con.close();
			} catch(Exception e){
				System.out.println(e);
			}
		return isSuccess;
	}

	public boolean editMedicine(Medicine medicine) {
		boolean isSuccess = false;
		try{
			Class.forName("com.mysql.cj.jdbc.Driver");
```

```
                        Connection con=DriverManager.getConnection(
                        "jdbc:mysql://localhost:3306/medicare","ostl","ostl");
                        PreparedStatement
stmt=con.prepareStatement(Queries.EDIT_MEDICINE);
                        stmt.setString(1,medicine.getName());
                        stmt.setString(2, medicine.getDescription());
                        stmt.setInt(3, medicine.getSellerId());
                        stmt.setInt(4, medicine.getPrice());
                        stmt.setInt(5, medicine.getOfferId());
                        stmt.setInt(6, medicine.getTagId());
                        stmt.setBoolean(7, medicine.isActive());
                        stmt.setString(8, medicine.getImageUrl());
                        stmt.setInt(9, medicine.getId());

                        int result = stmt.executeUpdate();
                        if(result > 0) {
                                isSuccess=true;
                        }
                        con.close();
                        } catch(Exception e){
                                System.out.println(e);
                        }
                return isSuccess;
        }


}
```