ACADEMIC YEAR: 2024 – 25

Name & Register No of the Candidate:						
Course Code & Title:						
Date of Issue: 16.10.2024			**Date of Submission:**29.10.2024			
Year / Dept. / Sem / Section: III / / V /						
Assignment: II						
Reference(s):						
Marks Details						
Q. No	**1**	**2**	**3**	**4**	**5**	**Total (25)**
COs						
Marks Obtained						

Course In-charge

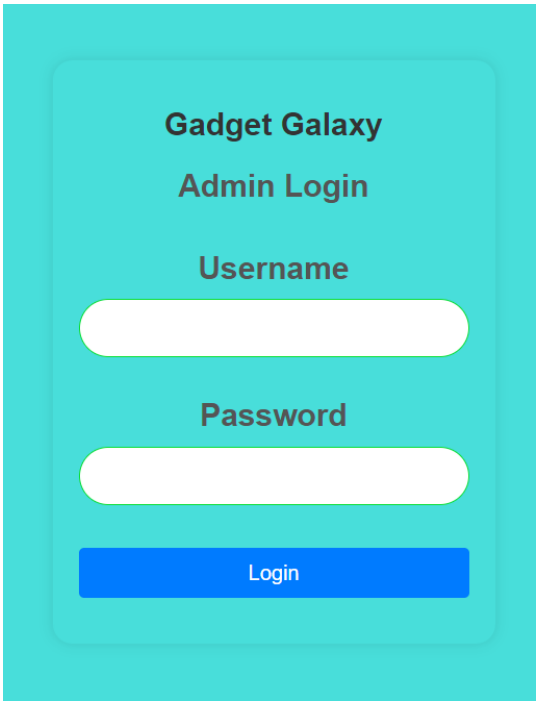
KARPAGAM COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine
(Autonomous)

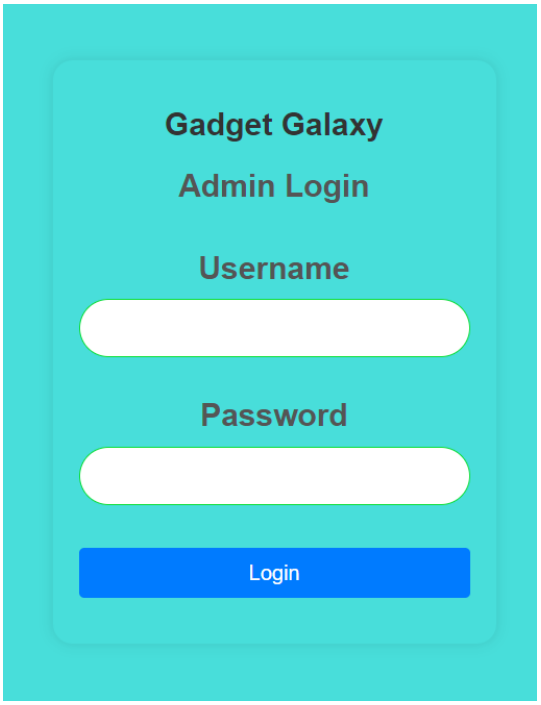
Rubrics for Assignment Evaluation

Criterion	Weightage	Exemplary (5)	Excellent (4)	Adequate (3)	Average (2)	Inadequate (1)	Marks Obtained
Relevance of Content	**5**	Writing shows high degree of attention to logic and reasoning of points.	Writing is logically organized with ideas and paragraphs.	Writing is coherent and logically organized.	Some points remain misplaced and stray from the topic.	Writing lacks logical organization.	
Creativeness	**5**	Content indicates synthesis of ideas, in depth analysis and evidences.	Content indicates original thinking and develops ideas with sufficient and firm evidence.	Content indicates thinking and reasoning applied with original thought.	Content shows average thinking and reasoning.	Shows some thinking and reasoning but most ideas are underdeveloped.	
Clarity of content	**5**	Reveals high degree of critical thinking	Critical thinking is weaved into points	Some critical thinking is present	Critical thinking lacks detailed development.	Ideas are vague with little evidence of critical thinking.	
Grammar & Style	**5**	Creative use of sentence structure	Excellent use of sentence structure	Adequate use of sentence structure	Average use of sentence structure	Inadequate use of sentence structure	
References	**5**	Adequate, complete and standard references	Adequate and complete references but substandard	Adequate but substandard and incomplete references	Adequate and incomplete references	Inadequate and incomplete references	
						Total Marks (out of 25)	

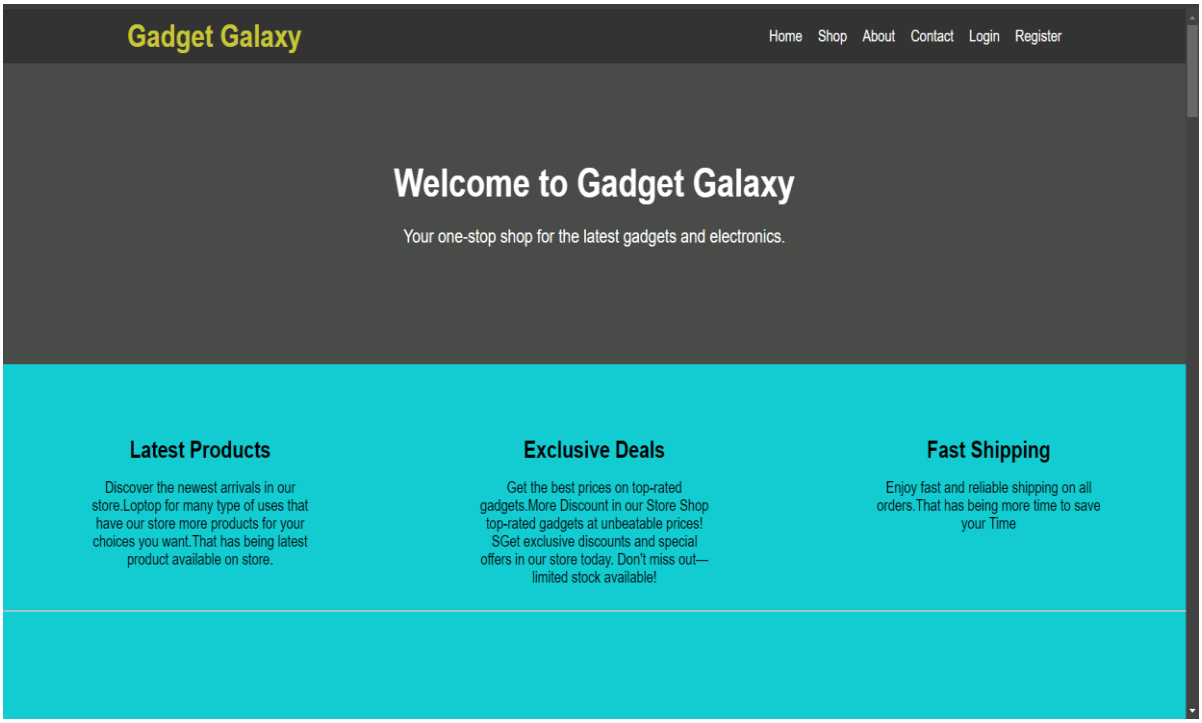
Signature of the Evaluator with date

Login page:

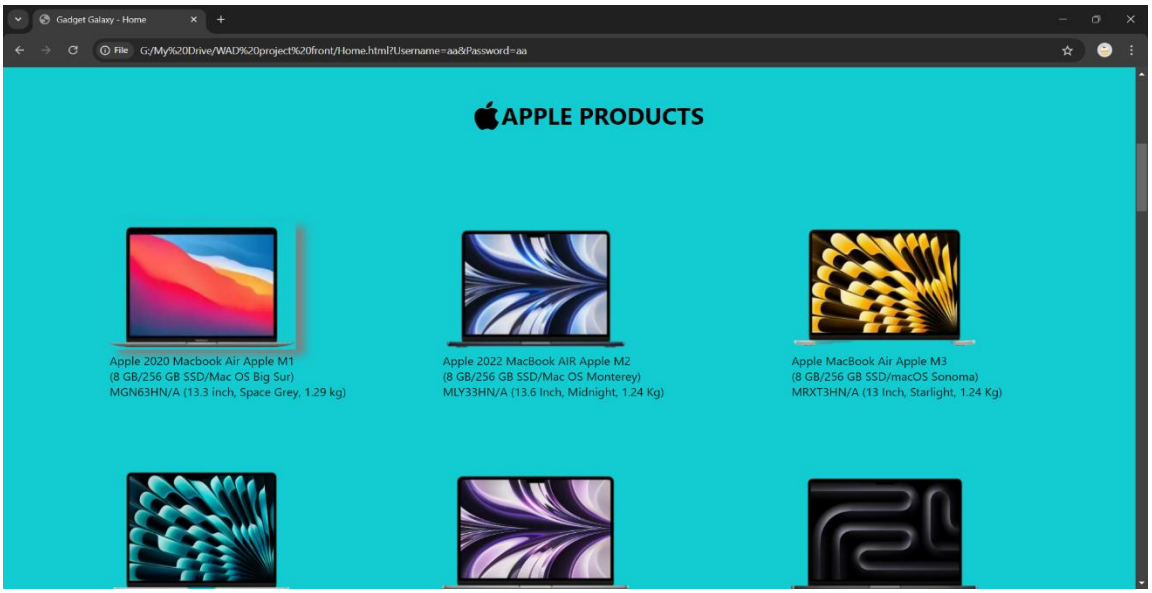




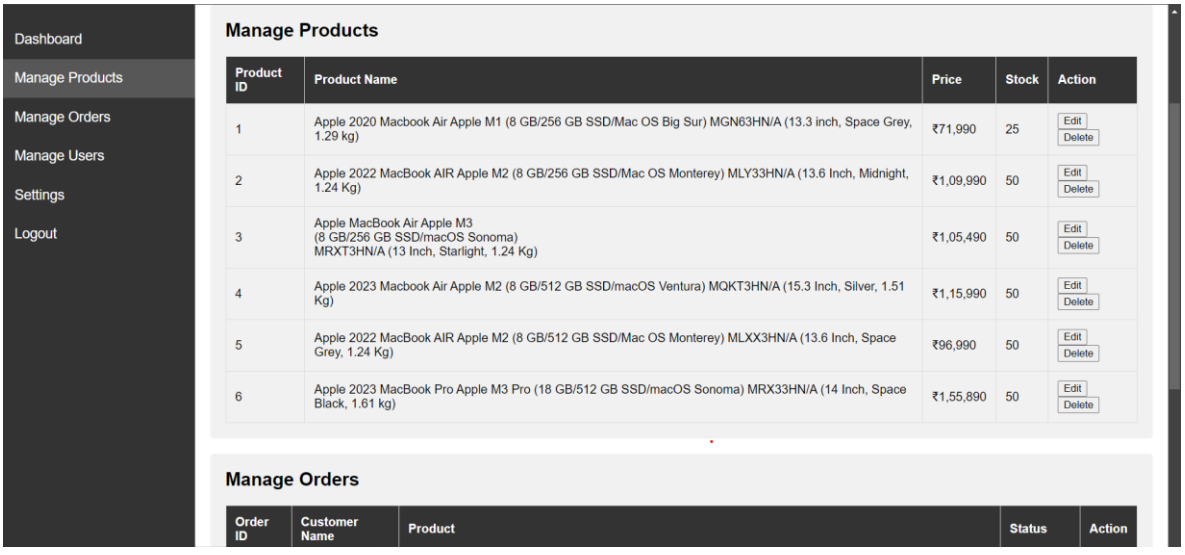
HomePage:



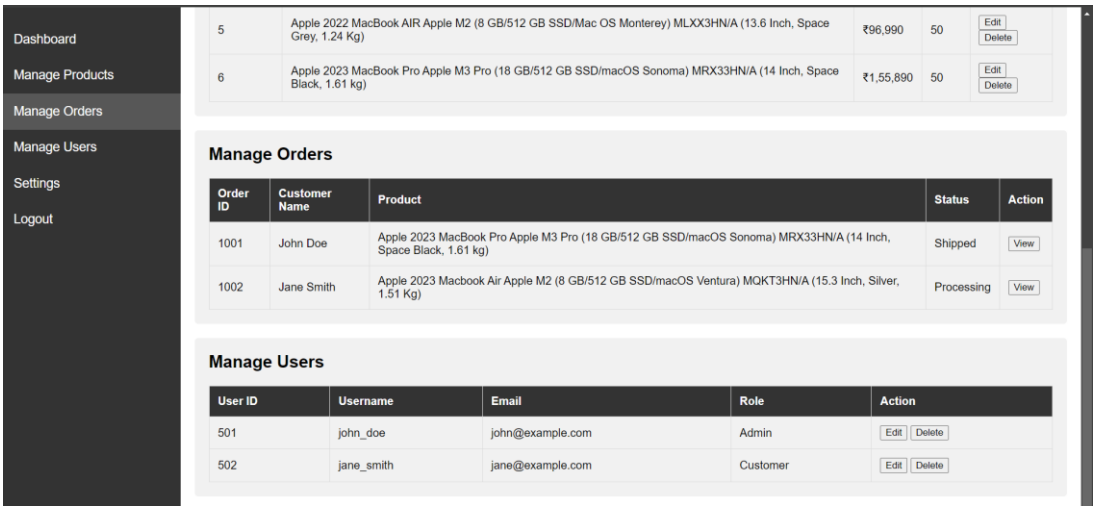
Product Page:



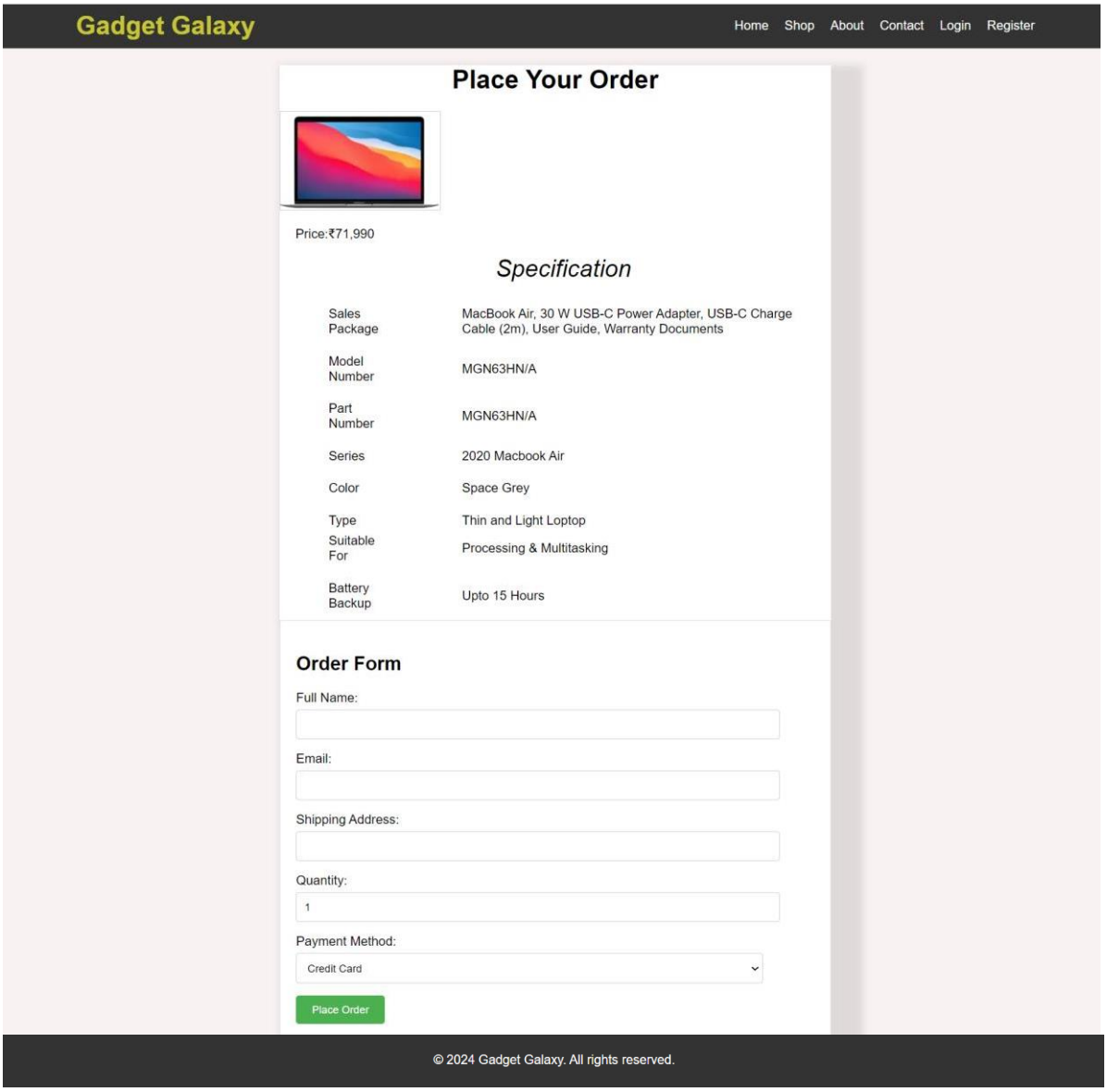
Manage product:



Manage Order:



Order page:



Loginservlet.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
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;

@WebServlet("/LoginServlet")
public class LoginServlet extends HttpServlet {
   protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
       String username = request.getParameter("Username");
       String password = request.getParameter("Password");

       try (Connection conn = DatabaseConnection.getConnection()) {
          String sql = "SELECT * FROM users WHERE username = ? AND password = ?";
          PreparedStatement stmt = conn.prepareStatement(sql);
          stmt.setString(1, username);
          stmt.setString(2, password);

          ResultSet rs = stmt.executeQuery();
          PrintWriter out = response.getWriter();

          if (rs.next()) {
             // Login success: Create a session and redirect to home page
             HttpSession session = request.getSession();
             session.setAttribute("username", username);
             response.sendRedirect("Home.jsp");
          } else {
             // Login failure: Redirect to login page with an error
             response.sendRedirect("login.jsp?error=invalid");
          }
       } catch (SQLException e) {
          e.printStackTrace();
       }
   }
}
```

HomeServlet.java

```
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/products")
public class ProductServlet extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        List<Product> products = new ArrayList<>();

        try (Connection conn = DatabaseConnection.getConnection()) {
            String sql = "SELECT * FROM products";
            PreparedStatement stmt = conn.prepareStatement(sql);
            ResultSet rs = stmt.executeQuery();

            while (rs.next()) {
                Product product = new Product();
                product.setId(rs.getInt("id"));
                product.setName(rs.getString("name"));
                product.setPrice(rs.getDouble("price"));
                product.setDescription(rs.getString("description"));
                product.setImageUrl(rs.getString("imageUrl"));
                products.add(product);
            }
        } catch (SQLException e) {p
            e.printStackTrace();
        }

        // Set products in request scope and forward to JSP
        request.setAttribute("products", products);
        request.getRequestDispatcher("products.jsp").forward(request, response);
    }
}
```

Orderservlet.java

```
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/placeOrder")
public class OrderServlet extends HttpServlet {
   protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
      String productName = "ASUS TUF Gaming F17"; // hardcoded for this example
      String customerName = request.getParameter("name");
      String email = request.getParameter("email");
      String address = request.getParameter("address");
      int quantity = Integer.parseInt(request.getParameter("quantity"));
      String paymentMethod = request.getParameter("payment");

      try (Connection conn = DatabaseConnection.getConnection()) {
         String sql = "INSERT INTO orders (product_name, customer_name, email,
shipping_address, quantity, payment_method) VALUES (?, ?, ?, ?, ?, ?)";
         PreparedStatement stmt = conn.prepareStatement(sql);
         stmt.setString(1, productName);
         stmt.setString(2, customerName);
         stmt.setString(3, email);
         stmt.setString(4, address);
         stmt.setInt(5, quantity);
         stmt.setString(6, paymentMethod);

         int rowsInserted = stmt.executeUpdate();
         if (rowsInserted > 0) {
            // Redirect to order confirmation page with success message
            response.sendRedirect("orderConfirmation.jsp?status=success");
         } else {
            response.sendRedirect("orderConfirmation.jsp?status=error");
         }
      } catch (SQLException e) {
         e.printStackTrace();
         response.sendRedirect("orderConfirmation.jsp?status=error");
      }
   }
}
```

Product Management Servlet

```
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/products")
public class ProductServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        List<Product> products = new ArrayList<>();
        try (Connection conn = DatabaseConnection.getConnection()) {
            String sql = "SELECT * FROM products";
            PreparedStatement stmt = conn.prepareStatement(sql);
            ResultSet rs = stmt.executeQuery();
            while (rs.next()) {
                Product product = new Product();
                product.setId(rs.getInt("id"));
                product.setName(rs.getString("name"));
                product.setPrice(rs.getDouble("price"));
                product.setStock(rs.getInt("stock"));
                products.add(product);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        request.setAttribute("products", products);
        request.getRequestDispatcher("products.jsp").forward(request, response);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        String action = request.getParameter("action");
        try (Connection conn = DatabaseConnection.getConnection()) {
            if ("add".equals(action)) {
                String name = request.getParameter("name");
                double price = Double.parseDouble(request.getParameter("price"));
                int stock = Integer.parseInt(request.getParameter("stock"));
                String sql = "INSERT INTO products (name, price, stock) VALUES (?, ?, ?)";
                PreparedStatement stmt = conn.prepareStatement(sql);
                stmt.setString(1, name);
```

```
  stmt.setDouble(2, price);
                stmt.setInt(3, stock);
                stmt.executeUpdate();
            } else if ("delete".equals(action)) {
                int id = Integer.parseInt(request.getParameter("id"));
                String sql = "DELETE FROM products WHERE id = ?";
                PreparedStatement stmt = conn.prepareStatement(sql);
                stmt.setInt(1, id);
                stmt.executeUpdate();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        response.sendRedirect("products");
    }
}
```

Order Management Servlet.java

```
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/orders")
public class OrderServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        List<Order> orders = new ArrayList<>();
        try (Connection conn = DatabaseConnection.getConnection()) {
            String sql = "SELECT o.id, o.customer_name, p.name AS product, o.status FROM orders o
JOIN products p ON o.product_id = p.id";
            PreparedStatement stmt = conn.prepareStatement(sql);
            ResultSet rs = stmt.executeQuery();
            while (rs.next()) {
                Order order = new Order();
                order.setId(rs.getInt("id"));
                order.setCustomerName(rs.getString("customer_name"));
                order.setProduct(rs.getString("product"));
                order.setStatus(rs.getString("status"));
                orders.add(order);
            }
```

```
    }
  catch (SQLException e) {
            e.printStackTrace();
        }
        request.setAttribute("orders", orders);
        request.getRequestDispatcher("orders.jsp").forward(request, response);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
  ServletException, IOException {
        String action = request.getParameter("action");
        try (Connection conn = DatabaseConnection.getConnection()) {
            if ("updateStatus".equals(action)) {
                int id = Integer.parseInt(request.getParameter("id"));
                String status = request.getParameter("status");
                String sql = "UPDATE orders SET status = ? WHERE id = ?";
                PreparedStatement stmt = conn.prepareStatement(sql);
                stmt.setString(1, status);
                stmt.setInt(2, id);
                stmt.executeUpdate();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        response.sendRedirect("orders");
    }
}
```