VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi – 590 018

Advance Java (21CSE644)

Assignment

"*Product Management System*"

Submitted By

Name: Rakshitha L Hegde **USN: 1GA21CS119**

Name: Rakshith H R **USN: 1GA21CS117**

Under the Guidance of

Mr. Shyam Sundar Bhushan
Assistant Professor
Dept. of CSE

Department of Artificial Intelligence & Data Science
GLOBAL ACADEMY OF TECHNOLOGY
Rajarajeshwarinagar, Bengaluru – 560 098

2023 - 2024

CHAPTER 1

PROBLEM DEFINITION

Managing product data manually is a common challenge across various industries, leading to inefficiencies and potential errors. Many businesses still rely on spreadsheets, paper records, or outdated software systems to handle product information. These methods often result in data redundancy, inconsistency, human errors, and time-consuming processes. Such inefficiencies not only slow down daily operations but also impair the ability to make informed decisions based on real-time data.

The absence of a centralized and secure platform for managing product information increases the risk of unauthorized access, data breaches, and loss of critical business data. As organizations expand, the complexity of managing product inventories, tracking details, and updating information grows, making it difficult to maintain accuracy and efficiency with manual methods.

This project addresses these challenges by developing a web-based Product Management System that centralizes all product-related activities. It allows users to securely log in, add, edit, view, search, and delete product information through a streamlined and user-friendly interface. The system incorporates robust authentication and session management mechanisms to ensure secure access. By leveraging a database-driven approach, it maintains data integrity and provides real-time updates, enabling better decision-making and more efficient management of product inventories.

Ultimately, this project aims to enhance operational efficiency, reduce errors, and provide a reliable, scalable solution for managing product data in a dynamic business environment.

CHAPTER 2

IMPLEMENTATION

2.1 COMPONENTS

1. User Authentication System:

- Manages user login, logout, and session management to ensure secure access to the system.
- Login Functionality: Allows users to enter their credentials (username and password) and verifies them against stored data.
- Session Management: Once authenticated, the system creates a session for the user, tracking their activity until they log out or the session expires.
- Logout Functionality: Ends the user's session and redirects them to the login page.
- Access Control: Ensures that only authenticated users can access the system's features and pages.

2. Product Management:

- Handles the addition, editing, deletion, and viewing of products in the system.
- Add Product: Allows users to input details such as Product

ID, Name, Category, and Price to add a new product to the system.

- Edit Product: Enables users to modify existing product details by fetching the product data, displaying it in editable fields, and saving the updates.
- Delete Product: Provides a way for users to remove a product from the system based on its ID.
- View Products: Displays a list of all products stored in the system with relevant details and action buttons for editing or deleting.
- Search Product: Allows users to search for products by their ID.

3. Data Management:

- Database Integration: A robust backend that stores product information and user credentials securely using SQL*Plus.
- Data Integrity: Ensures consistent and reliable data throughout the system using database constraints and validation.

4. User Interface (UI) Components:

- Provides the visual structure and navigation elements of the system, ensuring a user-friendly experience.
- Header and Navigation Bar: Consistent navigation across all

pages, including links to Home, Add Product, View Products, Search Product, and Logout.

- Styling with CSS: Uses a stylesheet (style.css) to apply consistent visual styling across the application, including button designs, table layouts, and form styles.
- Responsive Layouts: Ensures that the system's interface is accessible and usable on different screen sizes and devices.

5. Error Handling:

- Manages errors and exceptions in the application, providing appropriate feedback to the user.
- Error Pages: Redirects users to an error page when operations like adding, editing, or deleting a product fail.
- Input Validation: Ensures that user inputs are valid and prevents invalid data from being processed.

6. Security:

- Access Control: Restricts access to the system's functionalities based on user authentication.
- Data Protection: Safeguards sensitive information through secure data handling practices.

2.2 PROGRAM CODE

JSP Code.

- **addProduct.jsp**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@page import="dao.ProductManagementDAO"%>
<%@page import="pojo.Product"%>
<%@page import="java.util.*"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Add Product</title>
</head>
<body>
	<%@ include file="header.jsp" %>

	<div align="center">
		<form action="processAddProduct.jsp" method="post">
			<table class="productTable">
				<thead>
					<tr>
						<th colspan="2">
							Product Details
						</th>
					</tr>
				</thead>
				<tr>
					<td>Product ID</td>
					<td><input type="text" name="prodId" size="20"
						class="productTextField" /></td>
				</tr>
				<tr>
					<td>Product Name</td>
					<td><input type="text" name="prodName" size="20"
						class="productTextField" /></td>
				</tr>
				<tr>
					<td>Category</td>
					<td><input type="text" name="prodCategory" size="20"
						class="productTextField" /></td>
				</tr>
				<tr>
					<td>Price</td>
					<td><input type="text" name="prodPrice" size="20"
						class="productTextField" /></td>
				</tr>
			</table>
			<button class="actionBtn" style="margin-top:10px">Add</button>
		</form>
	</div>
</body>
```

```
</html>
```

- **editProduct.jsp**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@page import="dao.ProductManagementDAO"%>
<%@page import="pojo.Product"%>
<%@page import="java.util.*"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Edit Product</title>
</head>
<body>
	<%@ include file="header.jsp" %>
	<%
		String productId = request.getParameter("prodId");
		Product product = ProductManagementDAO.getProductById(productId);

	%>
	<div align="center">
		<form action="processEditProduct.jsp" method="post">
			<table class="productTable">
				<thead>
					<tr>
						<th colspan="2">
							Product Details
						</th>
					</tr>
				</thead>
				<tr>
					<td>Product ID</td>
					<td><input type="text" name="prodId" size="20"
						value="<%=productId%>" class="productTextField"
						readonly/></td>
				</tr>
				<tr>
					<td>Product Name</td>
					<td><input type="text" name="prodName" size="20"
						value="<%=product.getProductName()%>"
						class="productTextField"/></td>
				</tr>
				<tr>
					<td>Category</td>
					<td><input type="text" name="prodCategory" size="20"
						value="<%=product.getProductCategory()%>"
						class="productTextField"/></td>
				</tr>
				<tr>
					<td>Price</td>
					<td><input type="text" name="prodPrice" size="20"
						value="<%=product.getProductPrice()%>"
						class="productTextField"/></td>
```

```
                        </tr>
                    </table>
                    <button class="actionBtn" style="margin-top:10px">Save</button>
                </form>
        </div>
</body>
</html>
```

- **header.jsp**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<link rel="stylesheet" href="style.css">
</head>
<body>
        <%
        if(session.getAttribute("userName") == null)
        {
                response.sendRedirect("login.jsp");
        }
        %>
        <nav class="navbar">
                <ul class="navbar-nav">
                        <li><a href="home.jsp">Home</a></li>
                        <li><a href="addProduct.jsp">Add Product</a></li>
                        <li><a href="viewProducts.jsp">View Products</a></li>
                        <li><a href="searchProduct.jsp">Search Product</a></li>
                        <li style="float:right;margin-right:10px"><a
                        href="logout.jsp">Logout</a></li>
                </ul>
        </nav>
</body>
</html>
```

- **home.jsp**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Home</title>
</head>
<body>
<%@ include file="header.jsp" %>

<%
String userName = (String)session.getAttribute("userName");
```

```
%>

<div align="center">
<h2>Product Management System</h2>
<label>Welcome </label>
</div>

</body>
</html>
```

- **login.jsp**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Login</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
	<div align="center">
		<form action="processLogin.jsp" method="post">
			<table class="loginForm">
				<tr>
					<td><label for="userName">User Name</label></td>
					<td><input type="text" id="userName" name="userName"
					class="searchTextField"/></td>
				</tr>
				<tr>
					<td><label for="password">Password</label></td>
					<td><input type="password" id="password"
					name="password" class="searchTextField"/></td>
				</tr>
				<tr>
					<td colspan="2" align="center">
						<input type="submit" value="Login"
						class="actionBtn" />
					</td>
				</tr>
			</table>
		</form>
	</div>
</body>
</html>
```

- **loginFailed.jsp**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
```

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<body>
<h1>Invalid user credentials</h1>
<a href="login.jsp">Try Again</a>
</body>
</body>
</html>
```

- **loginout.jsp**

```
<%
session.invalidate();
response.sendRedirect("login.jsp");
%>
```

- **processAddProduct.jsp**

```
<%@page import="dao.ProductManagementDAO"%>
<%@page import="pojo.Product"%>

<%

String productId = request.getParameter("prodId");
String productName = request.getParameter("prodName");
String productCategory = request.getParameter("prodCategory");
Integer productPrice = Integer.parseInt(request.getParameter("prodPrice"));

Product product = new Product(productId,productName,productCategory,productPrice);

int status = ProductManagementDAO.addProduct(product);
if(status == 1)
{
	response.sendRedirect("viewProducts.jsp");
}
else
{
	response.sendRedirect("error.jsp");
}

%>
```

- **processDeleteProduct.jsp**

```
<%@page import="dao.ProductManagementDAO"%>
<%@page import="pojo.Product"%>

<%

String productId = request.getParameter("prodId");

int status = ProductManagementDAO.deleteProduct(productId);
```

```
if(status == 1)
{
	response.sendRedirect("viewProducts.jsp");
}
else
{
	response.sendRedirect("error.jsp");
}

%>
```

- **processEditProduct.jsp**

```
<%@page import="dao.ProductManagementDAO"%>
<%@page import="pojo.Product"%>

<%

String productId = request.getParameter("prodId");
String productName = request.getParameter("prodName");
String productCategory = request.getParameter("prodCategory");
Integer productPrice = Integer.parseInt(request.getParameter("prodPrice"));

Product product = new Product(productId,productName,productCategory,productPrice);

int status = ProductManagementDAO.updateProduct(product);
if(status == 1)
{
	response.sendRedirect("viewProducts.jsp");
}
else
{
	response.sendRedirect("error.jsp");
}

%>
```

- **processLogin.jsp**

```
<%@page import="dao.LoginDAO"%>
<%@page import="pojo.LoginInfo"%>
<%

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

if(LoginDAO.isUserValid(new LoginInfo(userName,password)))
{
	session.setAttribute("userName",userName);
	session.setMaxInactiveInterval(200);
	response.sendRedirect("home.jsp");
}
else
{
	response.sendRedirect("loginFailed.jsp");
```

```
}
%>
```

- **searchProduct.jsp**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@page import="dao.ProductManagementDAO"%>
<%@page import="pojo.Product"%>
<%@page import="java.util.*"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Search Product</title>
</head>
<body>
<%@ include file="header.jsp" %>
<div align="center" style="padding-top:25px;">
	<form action="searchProduct.jsp">
		<label>Enter Product ID: </label>
		<input type="text" name="prodId" size="25" class="searchTextField"/>
		<button class="actionBtn">Search</button>
	</form>
</div>
	<table align="center" class="productTable">
		<thead>
			<tr>
				<th>Product ID</th>
				<th>Product Name</th>
				<th>Category</th>
				<th>Price</th>
				<th colspan="2">Actions</th>
			</tr>
		</thead>
		<%
			String productId = request.getParameter("prodId");
			if(productId != null)
			{
				Product p = ProductManagementDAO.getProductById(productId);

				if(p != null)
				{
		%>
			<tr>
				<td><%=p.getProductId()%></td>
				<td><%=p.getProductName()%></td>
				<td><%=p.getProductCategory()%></td>
				<td><%= p.getProductPrice() %></td>
				<td><button class="actionBtn" onclick="location.href =
				'editProduct.jsp?prodId=<%=
				p.getProductId()%>';">Edit</button></td>
				<td><button class="actionBtn" onclick="location.href =
				'processDeleteProduct.jsp?prodId=<%=
				p.getProductId()%>';">Delete</button></td>
```

```
                </tr>
            <%      }
                        else
                        {
              %>
                <tr>
                        <td colspan="5">No record to display</td>
                </tr>
              <%
                        }
                }
                else
                {
              %>
                <tr>
                        <td colspan="5">No record to display</td>
                </tr>
              <%
                }
              %>
        </table>

</body>
</html>
```

- **viewProducts.jsp**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@page import="dao.ProductManagementDAO"%>
<%@page import="pojo.Product"%>
<%@page import="java.util.*"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>View Products</title>
</head>
<body>
<%@ include file="header.jsp" %>

        <table align="center" class="productTable">
                <thead>
                        <tr>
                                <th>Product ID</th>
                                <th>Product Name</th>
                                <th>Category</th>
                                <th>Price</th>
                                <th colspan="2">Actions</th>
                        </tr>
                </thead>
                <%
                        List<Product> productList = ProductManagementDAO.getAllProducts();
                        for (Product p : productList) {
                %>
```

```
			<tr>
				<td><%=p.getProductId()%></td>
				<td><%=p.getProductName()%></td>
				<td><%=p.getProductCategory()%></td>
				<td><%= p.getProductPrice() %></td>
				<td><button class="actionBtn" onclick="location.href =
				'editProduct.jsp?prodId=<%= p.getProductId()%>';">Edit</button></td>
				<td><button class="actionBtn" onclick="location.href =
				'processDeleteProduct.jsp?prodId=<%=
				p.getProductId()%>';">Delete</button></td>
			</tr>
			<%
}
%>
	</table>
</body>
</html>
```

CSS Code.

- **style.css**

```
@CHARSET "ISO-8859-1";

body {
	font-family: Arial;
	background-color: #f4f4f4;
}

.navbar {
	 background-color: #3b5998;
	 overflow: hidden;
	 height: 63px;
}

.navbar a {
	 float: left;
	 display: block;
	 color: #f2f2f2;
	 text-align: center;
	 padding: 14px 16px;
	 text-decoration: none;
	 font-size: 17px;
}

.navbar ul {
	margin: 8px 0 0 0;
	list-style: none;
}

.navbar a:hover {
	background-color: #ddd;
	color: #000;
}
```

```
.productTable {
	padding-top:25px;
	border-spacing: 0px;
}

.productTable thead tr th{
	padding: 15px;
	color: white;
	background-color: #374561;
	font-size:15px;
}

.productTable tbody tr td {
	padding: 13px;
	font-size: 13px
}

.productTable tr:nth-child(even) {
	background-color: #e4e4e4;
}

.productTable tr:nth-child(odd) {
	background-color: white;
}

.actionBtn {
    background-color: #1c54b5;
    padding: 10px;
    color: white;
    border: none;
    width: 75px;
    border-radius: 5px;
    cursor: pointer;
}

.actionBtn:hover  {
	 background-color: #3d74d2;
}

.searchTextField {
	height: 30px;
    border-radius: 5px;
    padding-left: 5px
}

.productTextField {
	height: 25px;
    border-radius: 5px;
    padding-left: 5px
}

.loginForm {
	background-color: #d5dbe4;
	border-spacing: 15px;
	padding: 10px;
	margin-top:100px;
```

```
}
```

Java Code.

- **LoginDAO.java**

```
package dao;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import dbutil.DBUtil;
import pojo.LoginInfo;
public class LoginDAO {
	public static boolean isUserValid(LoginInfo userDetails)
	{
		boolean validStatus = false;
		try{
			Connection conn = DBUtil.getConnection();
			Statement st= conn.createStatement();
			ResultSet rs= st.executeQuery("SELECT * FROM login_info WHERE
			user_name = '"+userDetails.getUserName()+"' AND password =
			'"+userDetails.getPassword()+"'");
			while(rs.next()){
				validStatus = true;
			}
			DBUtil.closeConnection(conn);
		}
		catch(Exception e){
			e.printStackTrace();
		}
		return validStatus;
	}
}
```

- **ProductManagementDAO.java**

```
package dao;
import java.sql.*;
import java.util.*;
import dbutil.DBUtil;
import pojo.Product;

public class ProductManagementDAO {
	public static List<Product> getAllProducts()
	{
		List<Product> productList = new ArrayList<Product>();
		try{
			Connection conn = DBUtil.getConnection();
			Statement st= conn.createStatement();
			ResultSet rs= st.executeQuery("SELECT * FROM product");
			while(rs.next()){
				Product product = new
				Product(rs.getString("prod_id"),rs.getString("prod_name"),rs.
				getString("prod_category"),rs.getInt("prod_price"));
				productList.add(product);
```

```
                }

                DBUtil.closeConnection(conn);
        }
        catch(Exception e)
        {
                e.printStackTrace();
        }

        return productList;
}


public static Product getProductById(String productId)
{
        Product product = null;
        try{
                Connection conn = DBUtil.getConnection();
                PreparedStatement ps= conn.prepareStatement("SELECT * FROM product
                WHERE prod_id = ?");
                ps.setString(1, productId);
                ResultSet rs = ps.executeQuery();
                while(rs.next())
                {
                        product = new
                        Product(rs.getString("prod_id"),rs.getString("prod_name"),rs.
                        getString("prod_category"),rs.getInt("prod_price"));
                }
        }
        catch(Exception e)
        {
                e.printStackTrace();
        }

        return product;
}

public static int addProduct(Product product)
{
        int status = 0;
        try
        {
                Connection conn = DBUtil.getConnection();
                PreparedStatement ps= conn.prepareStatement("INSERT INTO product
                VALUES(?,?,?,?)");
                ps.setString(1, product.getProductId());
                ps.setString(2, product.getProductName());
                ps.setString(3, product.getProductCategory());
                ps.setInt(4, product.getProductPrice());
                status = ps.executeUpdate();
        }
        catch(Exception e)
        {
                e.printStackTrace();
        }
        return status;
```

```
	}

	public static int updateProduct(Product product)
	{
		int status = 0;
		try
		{
			Connection conn = DBUtil.getConnection();
			PreparedStatement ps= conn.prepareStatement("UPDATE product SET
			prod_name=?, prod_category=?, prod_price=? WHERE prod_id=?");
			ps.setString(1, product.getProductName());
			ps.setString(2, product.getProductCategory());
			ps.setInt(3, product.getProductPrice());
			ps.setString(4, product.getProductId());
			status = ps.executeUpdate();
		}
		catch(Exception e)
		{
			e.printStackTrace();
		}
		return status;
	}

	public static int deleteProduct(String productId)
	{
		int status = 0;
		try
		{
			Connection conn = DBUtil.getConnection();
			PreparedStatement ps= conn.prepareStatement("DELETE FROM product
			where prod_id = ?");
			ps.setString(1, productId);
			status = ps.executeUpdate();
		}
		catch(Exception e)
		{
			e.printStackTrace();
		}
		return status;
	}
}
```

- **DBUtil.java**

```
package dbutil;
import java.sql.*;
public class DBUtil {

	public static Connection getConnection()
	{
		Connection conn = null;
		try
		{

			Class.forName("oracle.jdbc.driver.OracleDriver");
```

```
			conn =
			DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system",
			"system");
			}
			catch(Exception e)
			{
				e.printStackTrace();
			}
			return conn;
		}


		public static void closeConnection(Connection conn)
		{
			try {
				conn.close();
			} catch (SQLException e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
			}
		}
}
```

- **LoginInfo.java**

```
package pojo;
public class LoginInfo {

	String userName;
	String password;

	public LoginInfo() {
		// TODO Auto-generated constructor stub
	}

	public LoginInfo(String userName, String password) {
		super();
		this.userName = userName;
		this.password = password;
	}

	public String getUserName() {
		return userName;
	}

	public void setUserName(String userName) {
		this.userName = userName;
	}

	public String getPassword() {
		return password;
	}

	public void setPassword(String password) {
		this.password = password;
	}
```

```
	@Override
	public String toString() {
		return "LoginInfo [userName=" + userName + ", password=" + password + "]";
	}
}
```

- **Product.java**

```
package pojo;
public class Product {

	String productId;
	String productName;
	String productCategory;
	Integer productPrice;

	public Product() {
		// TODO Auto-generated constructor stub
	}

	public Product(String productId, String productName, String productCategory,
	Integer productPrice) {
		super();
		this.productId = productId;
		this.productName = productName;
		this.productCategory = productCategory;
		this.productPrice = productPrice;
	}

	public String getProductId() {
		return productId;
	}

	public void setProductId(String productId) {
		this.productId = productId;
	}

	public String getProductName() {
		return productName;
	}

	public void setProductName(String productName) {
		this.productName = productName;
	}

	public String getProductCategory() {
		return productCategory;
	}

	public void setProductCategory(String productCategory) {
		this.productCategory = productCategory;
	}

	public Integer getProductPrice() {
		return productPrice;
```

```
	}

	public void setProductPrice(Integer productPrice) {
		this.productPrice = productPrice;
	}

	@Override
	public String toString() {
		return "Product [productId=" + productId + ", productName=" + productName +
		", productCategory="+ productCategory + ", productPrice=" + productPrice +
		"]";
	}
}
```

CHAPTER 3

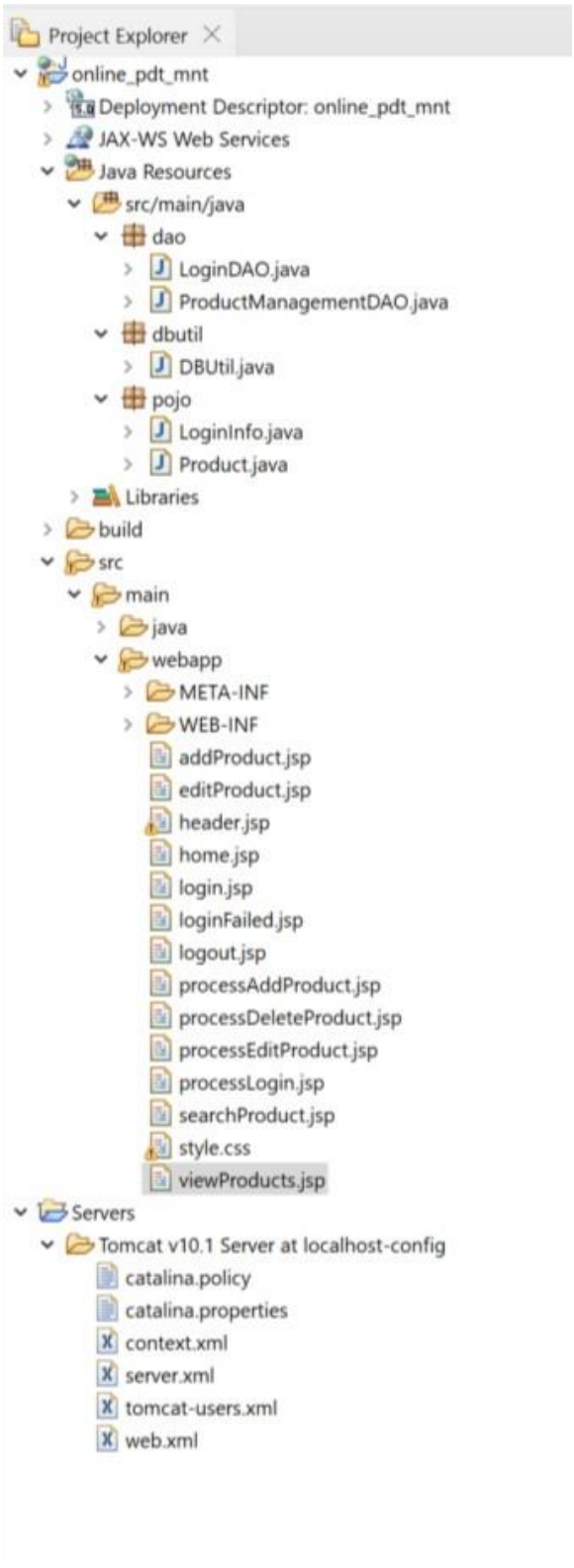
SNAPSHOTS

Backend (SQL*Plus):

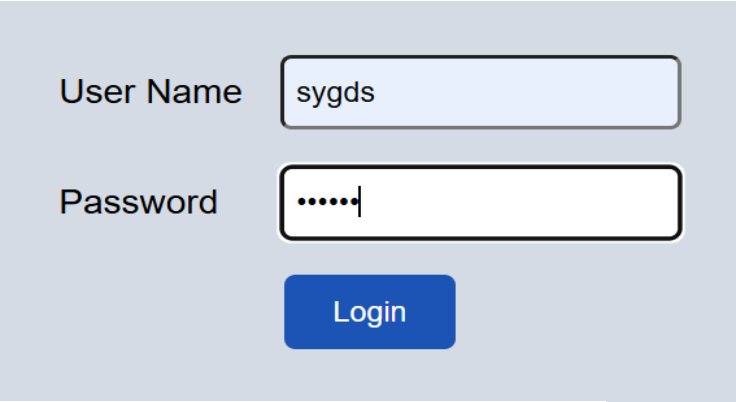
```
SQL*Plus: Release 21.0.0.0.0 - Production on Sat Aug 17 12:04:00 2024
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle.  All rights reserved.

Enter user-name: system
Enter password:
Last Successful login time: Sat Aug 17 2024 11:55:08 +05:30

Connected to:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> set linesize 200;
SQL> CREATE TABLE login_info(    user_name VARCHAR2(30),     password VARCHAR2(30));

Table created.

SQL> INSERT INTO login_info VALUES('raksh', 'raksh123');

1 row created.

SQL> CREATE TABLE product(    prod_id VARCHAR2(30),     prod_name VARCHAR2(30),     prod_category VARCHAR2(30),     prod_price INT);

Table created.

SQL> INSERT INTO product VALUES('P001', 'iPhone', 'Mobile phones', '10000');

1 row created.

SQL> INSERT INTO product VALUES('P002', 'Sony Bravia', 'Television', '7000');

1 row created.

SQL> INSERT INTO product VALUES('P003', 'T-shirt', 'Clothing', '1000');

1 row created.

SQL> INSERT INTO product VALUES('P004', 'Go Pro', 'Camera', '5000');

1 row created.

SQL> select * from login_info;

USER_NAME                      PASSWORD
------------------------------ ------------------------------
raksh                          raksh123

SQL> select*from product;

PROD_ID                        PROD_NAME                      PROD_CATEGORY                  PROD_PRICE
------------------------------ ------------------------------ ------------------------------ ----------
P001                           iPhone                         Mobile phones                       10000
P002                           Sony Bravia                    Television                           7000
P003                           T-shirt                        Clothing                             1000
P004                           Go Pro                         Camera                               5000

SQL>
```

Frontend:

Root directory structure:



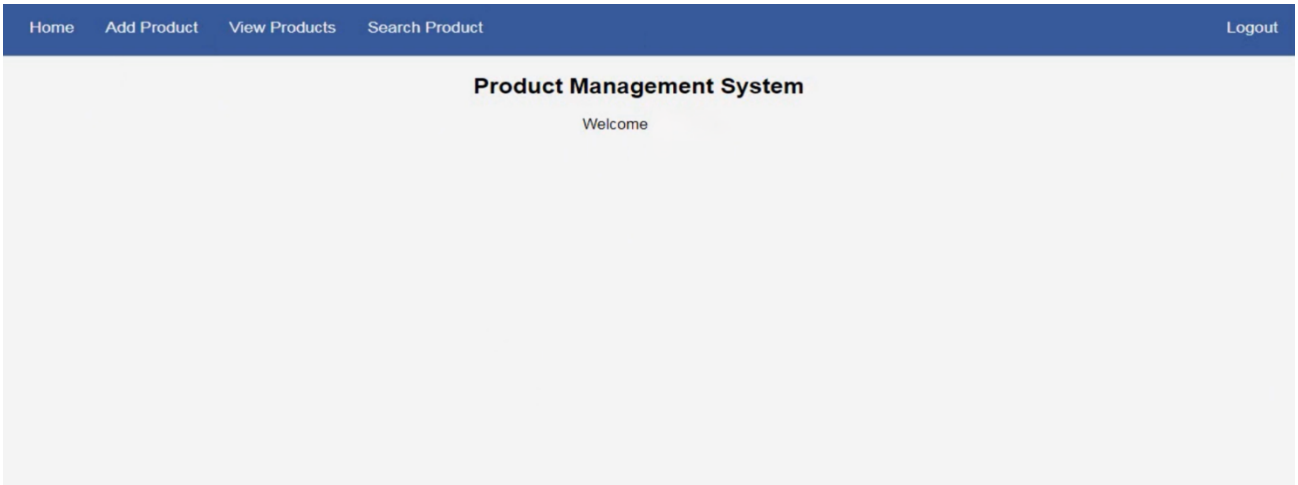
- **Login Functionality**



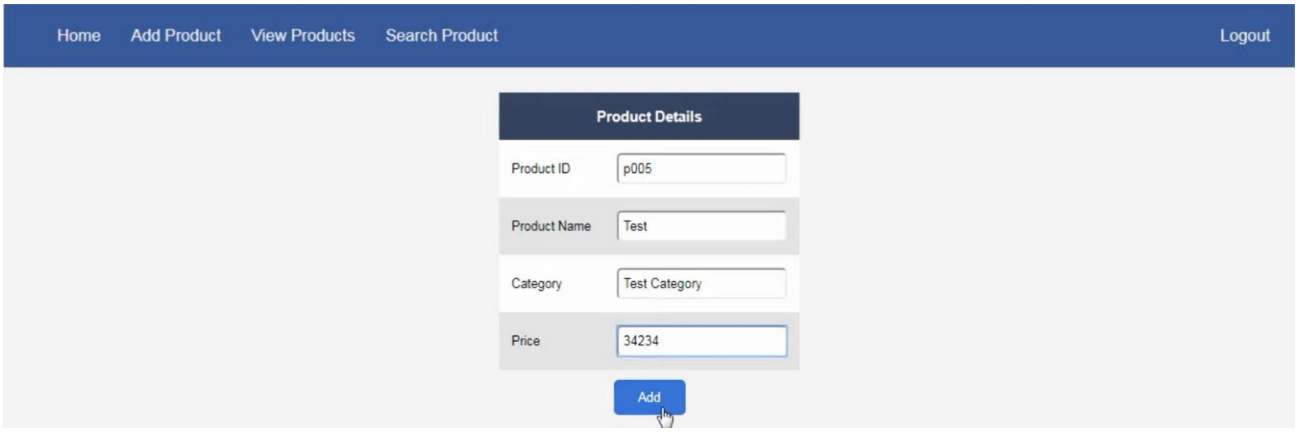
Invalid user credentials

Try Again

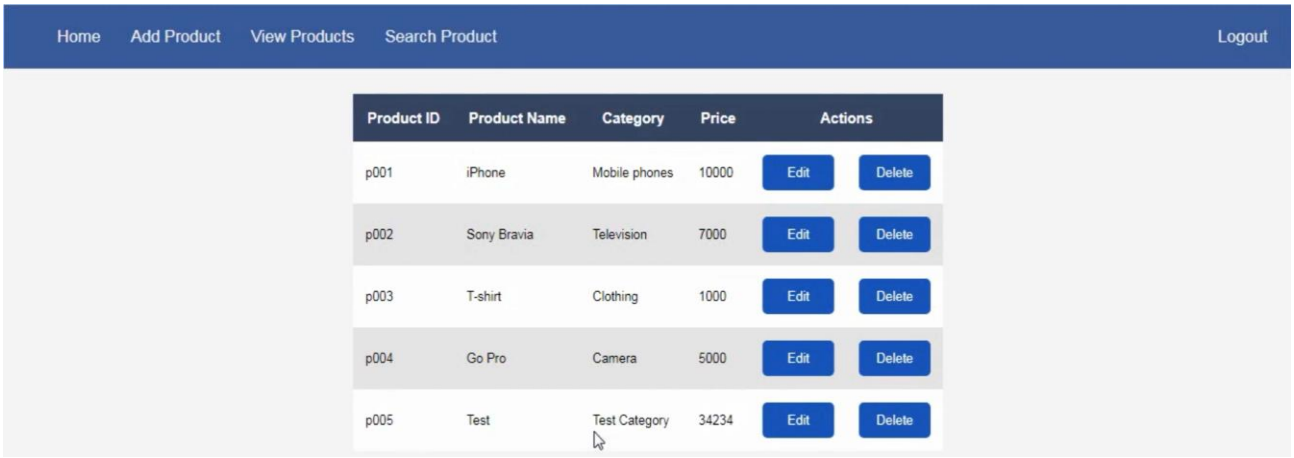
- **Home Page**



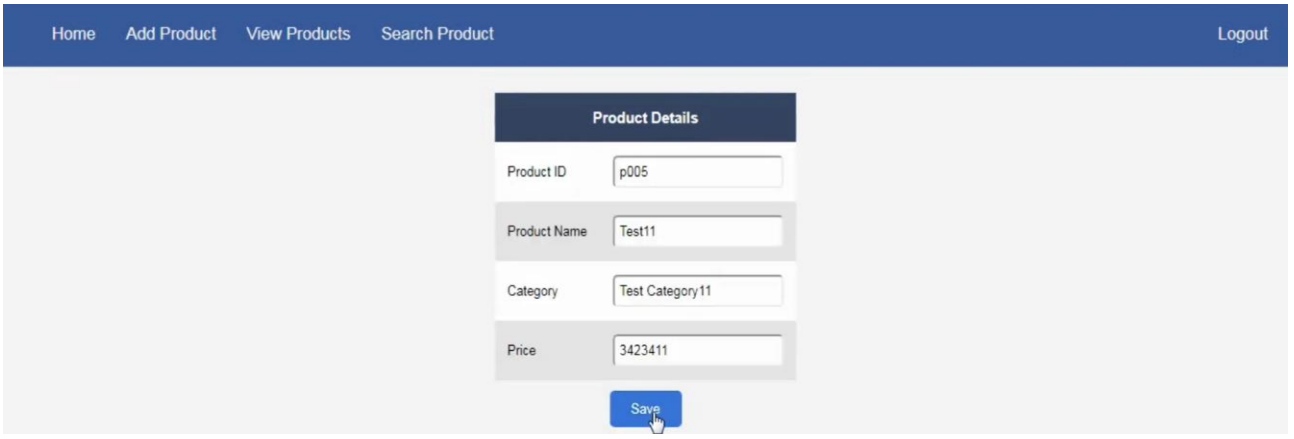
- **Add Product**



- **View Product**



- **Edit Product**



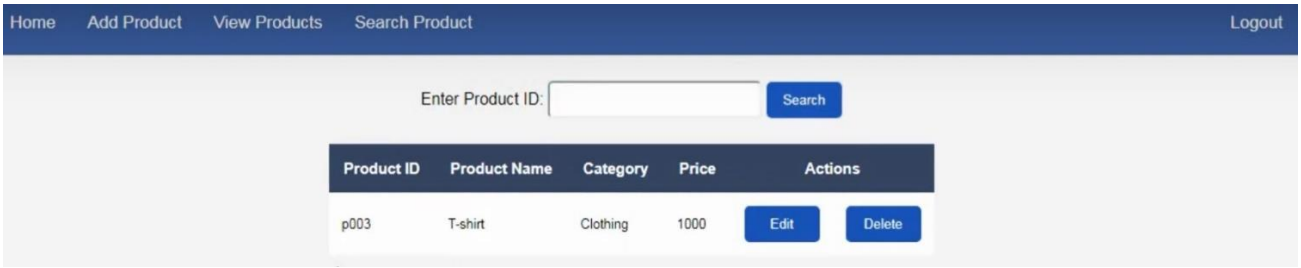
Updates reflected on backend:

```
Copyright (c) 1982, 2014, Oracle.  All rights reserved.

SQL> connect;
Enter user-name: system
Enter password:
Connected.
SQL> set linesize 200;
SQL> select * from product;

PROD_ID                        PROD_NAME                      PROD_CATEGORY                  PROD_PRICE
------------------------------ ------------------------------ ------------------------------ ----------
p001                           iPhone                         Mobile phones                       10000
p002                           Sony Bravia                    Television                           7000
p003                           T-shirt                        Clothing                             1000
p004                           Go Pro                         Camera                               5000
p005                           Test                           Test Category                       34234

SQL> select * from product;

PROD_ID                        PROD_NAME                      PROD_CATEGORY                  PROD_PRICE
------------------------------ ------------------------------ ------------------------------ ----------
p001                           iPhone                         Mobile phones                       10000
p002                           Sony Bravia                    Television                           7000
p003                           T-shirt                        Clothing                             1000
p004                           Go Pro                         Camera                               5000
p005                           Test11                         Test Category11                   3423411
```

- **Delete Product**

```
SQL> select * from product;

PROD_ID                        PROD_NAME                      PROD_CATEGORY                  PROD_PRICE
------------------------------ ------------------------------ ------------------------------ ----------
p001                           iPhone                         Mobile phones                       10000
p002                           Sony Bravia                    Television                           7000
p003                           T-shirt                        Clothing                             1000
p004                           Go Pro                         Camera                               5000

SQL> _
```

- **Search Product**



CONCLUSION

The Product Management System project offers an efficient solution for managing product information and user interactions within an enterprise environment. With its well-structured architecture, including DAO classes for data handling, JSP pages for user interfaces, and utility classes for database connectivity, the application ensures that both functionality and maintainability are prioritized. This project allows users to manage products seamlessly, facilitating operations such as creating, updating, and deleting product entries, while also enabling effective retrieval of data from the database.

The design of the system emphasizes separation of concerns, which not only improves code readability but also ensures scalability and ease of future enhancements. The use of JSP pages for the user interface provides a dynamic and responsive experience, allowing users to interact with the system in a straightforward manner. The integration of a robust database connection utility highlights the project's focus on reliability and efficiency, ensuring that database interactions are handled smoothly.

Overall, this project serves as a solid foundation for building more advanced product management features, such as user authentication, role-based access control, and advanced reporting tools. The modular nature of the application also makes it adaptable to different business requirements, offering flexibility for future growth. By combining essential product management functionalities with a clear, maintainable code structure, this project not only meets the current needs but also positions itself as a valuable tool for any organization aiming to optimize its product management processes.