***Question 1***

public class StringHandlingDemo {

public static void main(String[] args) {

// Original String

String str = " Hello, World! Welcome to Java String Handling. ";

// a. Substring

String substring = str.substring(7, 13);

System.out.println("Substring (7, 13): " + substring);

// b. Replace

String replacedString = str.replace("World", "Universe");

System.out.println("Replaced String: " + replacedString);

// c. Split

String[] splitString = str.split(" ");

System.out.println("Split String:");

for (String s : splitString) {

System.out.println(s);

}

// d. Join

String joinedString = String.join("-", splitString);

System.out.println("Joined String: " + joinedString);

// e. toUpperCase

String upperCaseString = str.toUpperCase();

System.out.println("Upper Case: " + upperCaseString);

// f. toLowerCase

String lowerCaseString = str.toLowerCase();

System.out.println("Lower Case: " + lowerCaseString);

// g. Trim

String trimmedString = str.trim();

System.out.println("Trimmed String: " + trimmedString);

// h. startsWith

boolean startsWithHello = str.startsWith("Hello");

System.out.println("Starts with 'Hello': " + startsWithHello);

// i. endsWith

boolean endsWithHandling = str.endsWith("Handling. ");

System.out.println("Ends with 'Handling. ': " + endsWithHandling);

// j. equalsIgnoreCase

String str2 = " hello, world! welcome to java string handling. ";

boolean equalsIgnoreCase = str.equalsIgnoreCase(str2);

System.out.println("Equals Ignore Case: " + equalsIgnoreCase);

}

}

--end--

***Question 2***

**Step 1: Set Up the Project**

1. **Create a new Dynamic Web Project in Eclipse**:
   * File -> New -> Dynamic Web Project
   * Name your project (e.g., LoginSessionApp).
2. **Configure Deployment Descriptor (web.xml)**:
   * Right-click on WEB-INF -> New -> Other -> XML -> XML File -> Finish.
   * Name it web.xml and configure it.

**Step 2: Create the web.xml File**

xml

Copy code

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee

http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"

version="3.1">

<servlet>

<servlet-name>LoginServlet</servlet-name>

<servlet-class>com.example.LoginServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>LoginServlet</servlet-name>

<url-pattern>/login</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>WelcomeServlet</servlet-name>

<servlet-class>com.example.WelcomeServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>WelcomeServlet</servlet-name>

<url-pattern>/welcome</url-pattern>

</servlet-mapping>

<session-config>

<session-timeout>30</session-timeout>

</session-config>

</web-app>

**Step 3: Create LoginServlet.java**

java

Copy code

package com.example;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import java.io.IOException;

@WebServlet("/login")

public class LoginServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

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

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

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

// For simplicity, assume username is "admin" and password is "password"

if("admin".equals(username) && "password".equals(password)) {

HttpSession session = request.getSession();

session.setAttribute("username", username);

// Create a cookie to remember the login state

Cookie loginCookie = new Cookie("username", username);

loginCookie.setMaxAge(30\*60); // 30 minutes

response.addCookie(loginCookie);

response.sendRedirect("welcome");

} else {

response.sendRedirect("login.html");

}

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

doPost(request, response);

}

}

**Step 4: Create WelcomeServlet.java**

java

Copy code

package com.example;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import java.io.IOException;

import java.io.PrintWriter;

@WebServlet("/welcome")

public class WelcomeServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

HttpSession session = request.getSession(false);

String username = null;

if(session != null) {

username = (String) session.getAttribute("username");

}

if(username == null) {

// Check cookies

Cookie[] cookies = request.getCookies();

if(cookies != null) {

for(Cookie cookie : cookies) {

if(cookie.getName().equals("username")) {

username = cookie.getValue();

break;

}

}

}

}

if(username != null) {

out.println("<h2>Welcome, " + username + "!</h2>");

out.println("<a href='logout'>Logout</a>");

} else {

response.sendRedirect("login.html");

}

}

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

doGet(request, response);

}

}

**Step 5: Create LogoutServlet.java**

java

Copy code

package com.example;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import java.io.IOException;

@WebServlet("/logout")

public class LogoutServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

HttpSession session = request.getSession(false);

if(session != null) {

session.invalidate();

}

Cookie loginCookie = new Cookie("username", "");

loginCookie.setMaxAge(0);

response.addCookie(loginCookie);

response.sendRedirect("login.html");

}

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

doGet(request, response);

}

}

**Step 6: Create HTML Files**

1. **login.html**:

html

Copy code

<!DOCTYPE html>

<html>

<head>

<title>Login</title>

</head>

<body>

<h2>Login</h2>

<form method="post" action="login">

Username: <input type="text" name="username"><br>

Password: <input type="password" name="password"><br>

<input type="submit" value="Login">

</form>

</body>

</html>

1. **welcome.html**:

html

Copy code

<!DOCTYPE html>

<html>

<head>

<title>Welcome</title>

</head>

<body>

<h2>Welcome!</h2>

<a href="logout">Logout</a>

</body>

</html>

**Step 7: Run the Application**

1. **Deploy the project on a server like Apache Tomcat**.
2. **Access the login page** by navigating to http://localhost:8080/LoginSessionApp/login.html.
3. **Log in with the username "admin" and password "password"**.
4. **You should be redirected to the welcome page** with a personalized greeting.
5. **Close the browser and reopen it**. Navigate directly to http://localhost:8080/LoginSessionApp/welcome. You should still see the personalized greeting, demonstrating that the session is maintained using cookies.

***Question 3***

**Step 1: Create a Student Class**

First, create a simple Student class to store student data.

java

Copy code

package com.example;

public class Student {

private String name;

private String grade;

public Student(String name, String grade) {

this.name = name;

this.grade = grade;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getGrade() {

return grade;

}

public void setGrade(String grade) {

this.grade = grade;

}

}

**Step 2: Create a Servlet to Prepare the Data**

Create a servlet that initializes a list of Student objects and forwards the data to the JSP page.

java

Copy code

package com.example;

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 java.io.IOException;

import java.util.ArrayList;

import java.util.List;

@WebServlet("/students")

public class StudentServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

List<Student> students = new ArrayList<>();

students.add(new Student("John Doe", "A"));

students.add(new Student("Jane Smith", "B"));

students.add(new Student("Emily Johnson", "A"));

students.add(new Student("Michael Brown", "C"));

request.setAttribute("students", students);

RequestDispatcher dispatcher = request.getRequestDispatcher("students.jsp");

dispatcher.forward(request, response);

}

}

**Step 3: Configure the Deployment Descriptor (web.xml)**

Add the servlet configuration to the web.xml file.

xml

Copy code

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee

http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"

version="3.1">

<servlet>

<servlet-name>StudentServlet</servlet-name>

<servlet-class>com.example.StudentServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>StudentServlet</servlet-name>

<url-pattern>/students</url-pattern>

</servlet-mapping>

</web-app>

**Step 4: Create the JSP Page**

Create a JSP page named students.jsp to display the list of students and their grades using JSTL.

jsp

Copy code

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>

<html>

<head>

<title>Student Grades</title>

</head>

<body>

<h2>Student Grades</h2>

<table border="1">

<tr>

<th>Name</th>

<th>Grade</th>

</tr>

<c:forEach var="student" items="${students}">

<tr>

<td>${student.name}</td>

<td>${student.grade}</td>

</tr>

</c:forEach>

</table>

</body>

</html>

**Step 5: Deploy and Run the Application**

1. **Deploy the application** on a server like Apache Tomcat.
2. **Access the servlet** by navigating to http://localhost:8080/YourAppName/students.

You should see a list of students and their respective grades displayed in an HTML table.

***Question 4***

**Step 1: Create a MySQL Database**

Run the following SQL commands to create the database and table:

sql

Copy code

CREATE DATABASE school;

USE school;

CREATE TABLE students (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100),

grade VARCHAR(2)

);

**Step 2: Servlet Code**

**Add MySQL Connector Dependency**

Ensure you have the MySQL Connector/J library in your project. If you're using Maven, add the following dependency to your pom.xml:

xml

Copy code

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.26</version>

</dependency>

**Create the Servlet**

java

Copy code

package com.example;

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 java.io.IOException;

import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

@WebServlet("/students")

public class StudentServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

private Connection getConnection() throws SQLException {

String jdbcURL = "jdbc:mysql://localhost:3306/school";

String dbUser = "root";

String dbPassword = "password";

return DriverManager.getConnection(jdbcURL, dbUser, dbPassword);

}

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

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

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

try (Connection connection = getConnection()) {

String sql = "INSERT INTO students (name, grade) VALUES (?, ?)";

PreparedStatement statement = connection.prepareStatement(sql);

statement.setString(1, name);

statement.setString(2, grade);

statement.executeUpdate();

} catch (SQLException e) {

throw new ServletException("Unable to insert student", e);

}

response.sendRedirect("students");

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

List<Student> students = new ArrayList<>();

try (Connection connection = getConnection()) {

String sql = "SELECT name, grade FROM students";

Statement statement = connection.createStatement();

ResultSet resultSet = statement.executeQuery(sql);

while (resultSet.next()) {

String name = resultSet.getString("name");

String grade = resultSet.getString("grade");

students.add(new Student(name, grade));

}

} catch (SQLException e) {

throw new ServletException("Unable to retrieve students", e);

}

request.setAttribute("students", students);

RequestDispatcher dispatcher = request.getRequestDispatcher("students.jsp");

dispatcher.forward(request, response);

}

}

**Create the Student Class**

java

Copy code

package com.example;

public class Student {

private String name;

private String grade;

public Student(String name, String grade) {

this.name = name;

this.grade = grade;

}

public String getName() {

return name;

}

public String getGrade() {

return grade;

}

}

**Step 3: JSP Code**

Create a JSP page named students.jsp to display the list of students and their grades.

**Create students.jsp**

jsp

Copy code

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>

<html>

<head>

<title>Student Grades</title>

</head>

<body>

<h2>Student Grades</h2>

<form action="students" method="post">

<label for="name">Name:</label>

<input type="text" id="name" name="name" required>

<label for="grade">Grade:</label>

<input type="text" id="grade" name="grade" required>

<button type="submit">Add Student</button>

</form>

<hr>

<table border="1">

<tr>

<th>Name</th>

<th>Grade</th>

</tr>

<c:forEach var="student" items="${students}">

<tr>

<td>${student.name}</td>

<td>${student.grade}</td>

</tr>

</c:forEach>

</table>

</body>

</html>

**Step 4: Configure the Deployment Descriptor (web.xml)**

Add the servlet configuration to the web.xml file.

xml

Copy code

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee

http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"

version="3.1">

<servlet>

<servlet-name>StudentServlet</servlet-name>

<servlet-class>com.example.StudentServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>StudentServlet</servlet-name>

<url-pattern>/students</url-pattern>

</servlet-mapping>

</web-app>

**Step 5: Deploy and Run the Application**

1. **Deploy the application** on a server like Apache Tomcat.
2. **Access the servlet** by navigating to http://localhost:8080/YourAppName/students.