Name : Mohil Mokaria

Class : 6CE

Division : E

Subject : Java Experiment 7

Q – Create a web application Student Task Management.

Make necessary assumption whenever required.

 Perform Following steps to implement Session, Cookie and Database Connectivity in JSP.

1. **Task Management and JSP:**
   * Create a Task in which add.jsp.
   * Design a JSP page to perform insertion operation with Database connectivity.
2. **View Task:**
   * Create a Task in which view.jsp.
   * Design a JSP page to display which display number of Task assign to student.
3. **Update Task:**
   * Update a Task in which update.jsp.
   * Design a JSP page to display which update number of Task assign to student.
4. **Delete Task:**
   * Delete a task in which delete.jsp perform usgin Data base connectivity.

### task.java

package myClass;

public class Task {

private int id;

private String name;

private String task;

private String assignedTo;

public Task() {

}

public Task(int id, String name, String task, String assignedTo) {

this.id = id;

this.name = name;

this.task = task;

this.assignedTo = assignedTo;

}

public void setId(int id) {

this.id = id;

}

public void setName(String name) {

this.name = name;

}

public void setTask(String task) {

this.task = task;

}

public void setAssignedTo(String assignedTo) {

this.assignedTo = assignedTo;

}

public int getId() {

return this.id;

}

public String getName() {

return this.name;

}

public String getTask() {

return this.task;

}

public String getAssignedTo() {

return this.assignedTo;

}

}

# view.jsp :

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%@ page import="myClass.Task" %>

<%@ page import="java.util.List" %>

<%@ page import="jakarta.servlet.http.HttpSession" %>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!--GOOGLE ICONS CDN-->

<link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,wght,FILL,GRAD@24,400,0,0" />

<!--BOOTSTRAP CDN-->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

<link rel="stylesheet" href="./view.css" />

<title>View Task</title>

</head>

<body>

<nav class="navbar navbar-expand-lg bg-body-tertiary">

<div class="d-flex justify-content-between my-navbar">

<div>

<h1 class="navbar-brand" href="#">Tasks</h1>

</div>

<div>

<a class="btn btn-primary" href="./add.jsp">Add Task</a>

</div>

</div>

</nav>

<div class="content">

<%

String exceptionMsg = (String) request.getAttribute("exception");

if(exceptionMsg != null && !exceptionMsg.isEmpty()){

%>

<p class="alert alert-danger"><%= exceptionMsg %></p>

<%

}

%>

<%

List<Task> tasks = (List<Task>) session.getAttribute("tasks");

if (tasks != null && !tasks.isEmpty()) {

%>

<table>

<tr>

<th>ID</th>

<th>Name</th>

<th>Task</th>

<th>Student</th>

<th>Edit</th>

<th>Delete</th>

</tr>

<%

for (Task t : tasks) {

%>

<tr>

<td><%= t.getId() %></td>

<td><%= t.getName() %></td>

<td><%= t.getTask() %></td>

<td><%= t.getAssignedTo() %></td>

<td>

<a class="btn btn-outline-primary" href="EditTask?id=<%= t.getId() %>">

<span class="material-symbols-outlined">

edit

</span>

</a>

</td>

<td>

<a class="btn btn-outline-danger" href="DeleteTask?id=<%= t.getId() %>">

<span class="material-symbols-outlined">

delete

</span>

</a>

</td>

</tr>

<%

}

%>

</table>

<%

} else {

%>

<center><h1>No Tasks Added</h1></center>

<%

}

%>

</div>

</body>

</html>

# add.jsp :

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- BOOTSTRAP CDNs -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">

<link rel="stylesheet" href="./add.css" />

<title>Add Task</title>

</head>

<body>

<form id="taskform" action="AddTask" method="post">

<%

String exceptionMsg = (String) request.getAttribute("exception");

if(exceptionMsg != null && !exceptionMsg.isEmpty()){

%>

<p class="alert alert-danger"><%= exceptionMsg %></p>

<%

}

%>

<h1>Add New Task</h1>

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

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

<label for="task">Task : </label>

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

<label for="student">Student : </label>

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

<center><button type="submit">Create</button></center>

</form>

</body>

</html>

### edit.jsp

<%@ page import="myClass.Task" %>

<%@ page import="jakarta.servlet.http.HttpSession" %>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- BOOTSTRAP CDNs -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">

<link rel="stylesheet" href="./add.css" />

<title>Edit Task</title>

</head>

<body>

<form id="taskform" action="EditTask" method="post">

<%

String exceptionMsg = (String) request.getAttribute("exception");

if(exceptionMsg != null && !exceptionMsg.isEmpty()){

%>

<p class="alert alert-danger"><%= exceptionMsg %></p>

<%

}

%>

<%

Task editTask = (Task) session.getAttribute("editTask");

if (editTask != null) {

%>

<h1>Edit Task</h1>

<input type="hidden" id="id" name="id" required value="<%= editTask.getId() %>">

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

<input type="text" id="name" name="name" value="<%= editTask.getName() %>" autofocus required>

<label for="task">Task : </label>

<input type="text" id="task" name="task" value="<%= editTask.getTask() %>" required>

<label for="student">Student : </label>

<input type="text" id="student" name="student" value="<%= editTask.getAssignedTo() %>" required>

<center><button type="submit">Modify</button></center>

<%

} else {

%>

<center><h1>Something Went Wrong!</h1></center>

<%

}

%>

</form>

</body>

</html>

# delete.jsp :

<%@ page import="myClass.Task" %>

<%@ page import="jakarta.servlet.http.HttpSession" %>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- BOOTSTRAP CDNs -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">

<link rel="stylesheet" href="./add.css" />

<title>Add Task</title>

</head>

<body>

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

<%

String exceptionMsg = (String) request.getAttribute("exception");

if(exceptionMsg != null && !exceptionMsg.isEmpty()){

%>

<p class="alert alert-danger"><%= exceptionMsg %></p>

<%

}

%>

<%

Task deleteTask = (Task) session.getAttribute("deleteTask");

if (deleteTask != null) {

%>

<p>Sure Want To Delete (<%= deleteTask.getName() %> - [<%= deleteTask.getTask() %>]), Which Is Assigned to Student (<%= deleteTask.getAssignedTo() %>)</p>

<input type="hidden" id="id" name="id" required value="<%= deleteTask.getId() %>">

<center><button class="btn btn-outline-danger" type="submit">Delete</button></center>

<%

} else {

%>

<center><h1>Something Went Wrong!</h1></center>

<%

}

%>

</form>

</body>

</html>

### ViewTask.java

import jakarta.servlet.RequestDispatcher;

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import jakarta.servlet.http.HttpSession;

import java.io.IOException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import java.util.logging.Level;

import java.util.logging.Logger;

import myClass.Task;

@WebServlet("/ViewTask")

public class ViewTask extends HttpServlet {

private static final long serialVersionUID = 1L;

@Override

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

final String jdbcUri = getServletContext().getInitParameter("jdbcUri");

final String dbUri = getServletContext().getInitParameter("dbUri");

final String dbId = getServletContext().getInitParameter("dbId");

final String dbPass = getServletContext().getInitParameter("dbPass");

HttpSession session = request.getSession();

List<Task> tasks = new ArrayList<>();

try {

Class.forName(jdbcUri);

try (Connection connection = DriverManager.getConnection(dbUri, dbId, dbPass)) {

String sql = "SELECT \* FROM tasks";

try (PreparedStatement statement = connection.prepareStatement(sql)) {

try (ResultSet resultSet = statement.executeQuery()) {

while (resultSet.next()) {

Task t = new Task();

t.setId(resultSet.getInt("id"));

t.setName(resultSet.getString("name"));

t.setTask(resultSet.getString("task"));

t.setAssignedTo(resultSet.getString("assigned\_to"));

tasks.add(t);

}

}

}

session.setAttribute("tasks", tasks);

RequestDispatcher rd = request.getRequestDispatcher("view.jsp");

rd.forward(request, response);

}

} catch (ClassNotFoundException | SQLException ex) {

Logger.getLogger(ViewTask.class.getName()).log(Level.SEVERE, null, ex);

request.setAttribute("exception", ex.getMessage());

RequestDispatcher rd = request.getRequestDispatcher("view.jsp");

rd.forward(request, response);

}

}

}

### AddTask.java

import jakarta.servlet.RequestDispatcher;

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

@WebServlet("/AddTask")

public class AddTask extends HttpServlet {

private static final long serialVersionUID = 1L;

@Override

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

final String jdbcUri = getServletContext().getInitParameter("jdbcUri");

final String dbUri = getServletContext().getInitParameter("dbUri");

final String dbId = getServletContext().getInitParameter("dbId");

final String dbPass = getServletContext().getInitParameter("dbPass");

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

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

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

try {

Class.forName(jdbcUri);

try (Connection con = DriverManager.getConnection(dbUri, dbId, dbPass)) {

PreparedStatement ps = con.prepareStatement("INSERT INTO tasks(name, task, assigned\_to) values(?,?,?)");

ps.setString(1, name);

ps.setString(2, task);

ps.setString(3, student);

int i = ps.executeUpdate();

response.sendRedirect("ViewTask");

}

} catch (ClassNotFoundException | SQLException ex) {

Logger.getLogger(AddTask.class.getName()).log(Level.SEVERE, null, ex);

request.setAttribute("exception", ex.getMessage());

RequestDispatcher rd = request.getRequestDispatcher("add.jsp");

rd.forward(request, response);

}

}

}

### EditTask.java

import jakarta.servlet.RequestDispatcher;

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import jakarta.servlet.http.HttpSession;

import java.io.IOException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

import myClass.Task;

@WebServlet("/EditTask")

public class EditTask extends HttpServlet {

private static final long serialVersionUID = 1L;

@Override

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

final String jdbcUri = getServletContext().getInitParameter("jdbcUri");

final String dbUri = getServletContext().getInitParameter("dbUri");

final String dbId = getServletContext().getInitParameter("dbId");

final String dbPass = getServletContext().getInitParameter("dbPass");

HttpSession session = request.getSession();

String taskIdString = request.getParameter("id");

int taskId = Integer.parseInt(taskIdString);

Task editTask = new Task();

try {

Class.forName(jdbcUri);

try (Connection connection = DriverManager.getConnection(dbUri, dbId, dbPass)) {

String sql = "SELECT \* FROM tasks WHERE id=?";

try (PreparedStatement statement = connection.prepareStatement(sql)) {

statement.setInt(1, taskId);

try (ResultSet resultSet = statement.executeQuery()) {

while (resultSet.next()) {

editTask.setId(resultSet.getInt("id"));

editTask.setName(resultSet.getString("name"));

editTask.setTask(resultSet.getString("task"));

editTask.setAssignedTo(resultSet.getString("assigned\_to"));

}

}

}

session.setAttribute("editTask", editTask);

RequestDispatcher rd = request.getRequestDispatcher("edit.jsp");

rd.forward(request, response);

}

} catch (ClassNotFoundException | SQLException ex) {

Logger.getLogger(ViewTask.class.getName()).log(Level.SEVERE, null, ex);

request.setAttribute("exception", ex.getMessage());

RequestDispatcher rd = request.getRequestDispatcher("edit.jsp");

rd.forward(request, response);

}

}

@Override

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

final String jdbcUri = getServletContext().getInitParameter("jdbcUri");

final String dbUri = getServletContext().getInitParameter("dbUri");

final String dbId = getServletContext().getInitParameter("dbId");

final String dbPass = getServletContext().getInitParameter("dbPass");

String taskIdString = request.getParameter("id");

int taskId = Integer.parseInt(taskIdString);

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

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

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

try {

Class.forName(jdbcUri);

try (Connection con = DriverManager.getConnection(dbUri, dbId, dbPass)) {

PreparedStatement ps = con.prepareStatement("UPDATE tasks SET name = ?, task = ?, assigned\_to = ? WHERE id = ?");

ps.setString(1, name);

ps.setString(2, task);

ps.setString(3, student);

ps.setInt(4, taskId);

int i = ps.executeUpdate();

if (i > 0) {

response.sendRedirect("ViewTask");

} else {

request.setAttribute("exception", "Something Went Wrong!");

RequestDispatcher rd = request.getRequestDispatcher("view.jsp");

rd.forward(request, response);

}

}

} catch (ClassNotFoundException | SQLException ex) {

Logger.getLogger(EditTask.class.getName()).log(Level.SEVERE, null, ex);

request.setAttribute("exception", ex.getMessage());

RequestDispatcher rd = request.getRequestDispatcher("edit.jsp");

rd.forward(request, response);

}

}

}

### DeleteTask.java

import jakarta.servlet.RequestDispatcher;

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import jakarta.servlet.http.HttpSession;

import java.io.IOException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

import myClass.Task;

@WebServlet("/DeleteTask")

public class DeleteTask extends HttpServlet {

private static final long serialVersionUID = 1L;

@Override

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

final String jdbcUri = getServletContext().getInitParameter("jdbcUri");

final String dbUri = getServletContext().getInitParameter("dbUri");

final String dbId = getServletContext().getInitParameter("dbId");

final String dbPass = getServletContext().getInitParameter("dbPass");

HttpSession session = request.getSession();

String taskIdString = request.getParameter("id");

int taskId = Integer.parseInt(taskIdString);

Task deleteTask = new Task();

try {

Class.forName(jdbcUri);

try (Connection connection = DriverManager.getConnection(dbUri, dbId, dbPass)) {

String sql = "SELECT \* FROM tasks WHERE id=?";

try (PreparedStatement statement = connection.prepareStatement(sql)) {

statement.setInt(1, taskId);

try (ResultSet resultSet = statement.executeQuery()) {

while (resultSet.next()) {

deleteTask.setId(resultSet.getInt("id"));

deleteTask.setName(resultSet.getString("name"));

deleteTask.setTask(resultSet.getString("task"));

deleteTask.setAssignedTo(resultSet.getString("assigned\_to"));

}

}

}

session.setAttribute("deleteTask", deleteTask);

RequestDispatcher rd = request.getRequestDispatcher("delete.jsp");

rd.forward(request, response);

}

} catch (ClassNotFoundException | SQLException ex) {

Logger.getLogger(ViewTask.class.getName()).log(Level.SEVERE, null, ex);

request.setAttribute("exception", ex.getMessage());

RequestDispatcher rd = request.getRequestDispatcher("delete.jsp");

rd.forward(request, response);

}

}

@Override

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

final String jdbcUri = getServletContext().getInitParameter("jdbcUri");

final String dbUri = getServletContext().getInitParameter("dbUri");

final String dbId = getServletContext().getInitParameter("dbId");

final String dbPass = getServletContext().getInitParameter("dbPass");

String taskIdString = request.getParameter("id");

int taskId = Integer.parseInt(taskIdString);

try {

Class.forName(jdbcUri);

try (Connection con = DriverManager.getConnection(dbUri, dbId, dbPass)) {

PreparedStatement ps = con.prepareStatement("DELETE FROM tasks WHERE id=?");

ps.setInt(1, taskId);

int rowsAffected = ps.executeUpdate();

if (rowsAffected > 0) {

response.sendRedirect("ViewTask");

} else {

request.setAttribute("exception", "Something Went Wrong!");

RequestDispatcher rd = request.getRequestDispatcher("view.jsp");

rd.forward(request, response);

}

ps.close();

}

} catch (ClassNotFoundException | SQLException ex) {

Logger.getLogger(AddTask.class.getName()).log(Level.SEVERE, null, ex);

request.setAttribute("exception", ex.getMessage());

RequestDispatcher rd = request.getRequestDispatcher("delete.jsp");

rd.forward(request, response);

}

}

}

### web.xml

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

<web-app version="6.0" xmlns="https://jakarta.ee/xml/ns/jakartaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee https://jakarta.ee/xml/ns/jakartaee/web-app\_6\_0.xsd">

<context-param>

<param-name>jdbcUri</param-name>

<param-value>com.mysql.cj.jdbc.Driver</param-value>

</context-param>

<context-param>

<param-name>dbUri</param-name>

<param-value>jdbc:mysql://localhost:3306/mysql</param-value>

</context-param>

<context-param>

<param-name>dbId</param-name>

<param-value>root</param-value>

</context-param>

<context-param>

<param-name>dbPass</param-name>

<param-value/>

</context-param>

<servlet>

<servlet-name>ViewTask</servlet-name>

<servlet-class>ViewTask</servlet-class>

</servlet>

<servlet>

<servlet-name>AddTask</servlet-name>

<servlet-class>AddTask</servlet-class>

</servlet>

<servlet>

<servlet-name>DeleteTask</servlet-name>

<servlet-class>DeleteTask</servlet-class>

</servlet>

<servlet>

<servlet-name>EditTask</servlet-name>

<servlet-class>EditTask</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ViewTask</servlet-name>

<url-pattern>/ViewTask</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>AddTask</servlet-name>

<url-pattern>/AddTask</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>DeleteTask</servlet-name>

<url-pattern>/DeleteTask</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>EditTask</servlet-name>

<url-pattern>/EditTask</url-pattern>

</servlet-mapping>

<welcome-file-list>

<welcome-file>ViewTask</welcome-file>

</welcome-file-list>

</web-app>

## Screenshots:

## 



