Experiment 8

Student Name: Akash Singh **UID: 22BCS12046**
Branch: CSE **Section: 618(A)**
Semester: 6th **DOP: 04/04/2025**
Subject: PBLJ **Subject Code: 22CSH-359**

Aim: Servlet Lifecycle, Generic Servlet, Http Servlet, Linking Servlet to HTML, HTTP Servlet Request and Response, Servlet with JDBC, configuring project using servlet, Servlet Config and Servlet Mapping JSP declaration, JSP directives, JSP Script lets, JSP include tag, JSP page tag

Objective: Develop web applications using Servlets and JSP for user input handling, database integration.

Problem 1.

Write a servlet to accept user credentials through an HTML form and display a personalized welcome message if the login is successful.

Code:

create the HTML login form:

```
<!-- login.html -->
<!DOCTYPE html>
<html>
<head>
  <title>Login Form</title>
  <style> body { font-family: Arial,
    sans-serif; margin: 40px;
    }
    .login-container { width:
      300px; padding: 20px;
      border: 1px solid #ddd;
      border-radius: 5px;
    }
    input[type="text"], input[type="password"]
      { width: 100%; padding: 10px; margin:
      8px 0; box-sizing: border-box;
```

```
        }      input[type="submit"]      {
        background-color:    #4CAF50;
        color:   white;   padding:   10px
        15px;   border:   none;   cursor:
        pointer; width: 100%;
        }
    </style>
</head>
<body>
    <div class="login-container">
        <h2>User Login</h2>
        <form action="LoginServlet" method="post">
            <label for="username">Username:</label>
            <input type="text" id="username" name="username" required>
            <label for="password">Password:</label>
            <input type="password" id="password" name="password" required>
            <input type="submit" value="Login">
        </form>
    </div>
</body> </html>
```

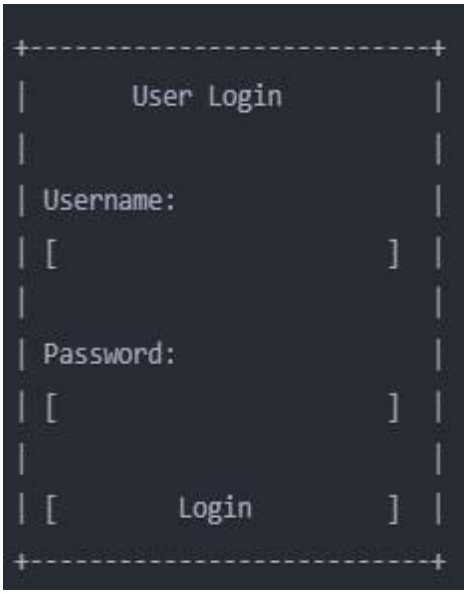
create the servlet to handle the login:

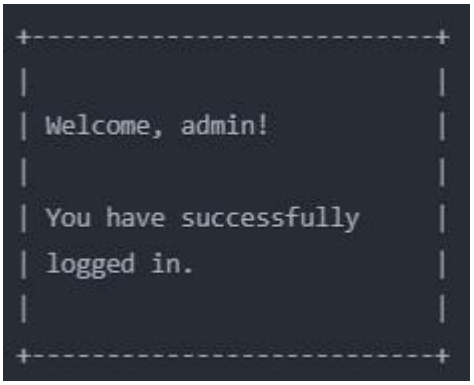
```
// LoginServlet.java import
java.io.IOException; import
java.io.PrintWriter;   import
javax.servlet.ServletExcept
ion;                   import
javax.servlet.annotation.We
bServlet;              import
javax.servlet.http.HttpServl
et;                    import
javax.servlet.http.HttpServl
etRequest;             import
```

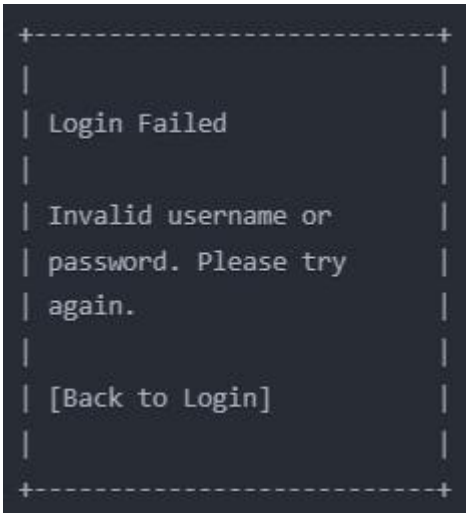
```
javax.servlet.http.HttpServl
etResponse;
@WebServlet("/LoginServlet") public class LoginServlet extends HttpServlet {
private static final long serialVersionUID = 1L; // Hard-coded credentials for
demonstration private static final String VALID_USERNAME = "admin"; private
static final String VALID_PASSWORD = "password"; protected void
doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    // Get the form parameters
    String username = request.getParameter("username"); String password =
    request.getParameter("password"); response.setContentType("text/html");
    PrintWriter out = response.getWriter(); out.println("<!DOCTYPE html>");
    out.println("<html>"); out.println("<head>"); out.println("<title>Login
    Result</title>"); out.println("<style>"); out.println("body { font-family: Arial,
    sans-serif; margin: 40px; }"); out.println(".message { padding: 20px; border-
    radius: 5px; margin-top: 20px; }"); out.println(".success { background-color:
    #dff0d8; color: #3c763d; }"); out.println(".error { background-color: #f2dede;
    color: #a94442; }"); out.println("</style>"); out.println("</head>");
    out.println("<body>");
    // Validate credentials
   if (VALID_USERNAME.equals(username) && VALID_PASSWORD.equals(password)) {
      out.println("<div class='message success'>");
      out.println("<h2>Welcome, " + username + "!</h2>");
      out.println("<p>You have successfully logged in.</p>");
      out.println("</div>");
    } else { out.println("<div class='message error'>");
    out.println("<h2>Login Failed</h2>"); out.println("<p>Invalid username
    or password. Please try again.</p>"); out.println("<a
    href='login.html'>Back to Login</a>"); out.println("</div>");    }
    out.println("</body>"); out.println("</html>");
  }
```

}

Output







Problem 2:

Create a servlet integrated with JDBC to display a list of employees from a database. Include a search form to fetch employee details by ID.

Code:

create the database structure: -

- Create employee table

```
CREATE TABLE employees ( id INT
   PRIMARY KEY, name
   VARCHAR(100) NOT NULL,
   position VARCHAR(100), salary
   DECIMAL(10,2),
   hire_date DATE
);
-- Insert some sample data
INSERT INTO employees VALUES (101, 'John Doe', 'Software Engineer', 75000.00, '2020-01-15');
INSERT INTO employees VALUES (102, 'Jane Smith', 'Project Manager', 85000.00, '2019-05-20');
INSERT INTO employees VALUES (103, 'Bob Johnson', 'UI/UX Designer', 70000.00, '2021-03-10');
INSERT INTO employees VALUES (104, 'Alice Williams', 'Database Administrator', 80000.00, '2018-11-05');
INSERT INTO employees VALUES (105, 'Charlie Brown', 'System Analyst', 72000.00, '2020-09-25');
```

create the HTML form for searching employees:

```
<!-- employeeSearch.html -->
<!DOCTYPE html>

<html>

<head>
   <title>Employee Search</title>
   <style> body
      {
         font-family: Arial, sans-serif; margin:
         40px;
      }
```

```
        .container { width:
          80%; max-width:
          800px; margin: 0
          auto;
        }
        .search-box { padding: 20px;
          background-color: #f5f5f5;
          border-radius: 5px;
          margin-bottom: 20px;
        } input[type="text"] { padding:
        8px; width: 200px; } button {
        padding: 8px 15px;
        background-color: #4CAF50;
        color: white; border: none;
        cursor: pointer;
        }
        a.button {
          padding: 8px 15px;
          background-color: #2196F3;
          color: white; text-
          decoration: none; border-
          radius: 3px; margin-left:
          10px;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Employee Directory</h1>
        <div class="search-box">
```

```
        <h3>Search Employee by ID</h3>
        <form action="EmployeeServlet" method="get">
          <input type="text" name="empId" placeholder="Enter Employee ID">
           <button type="submit">Search</button>
          <a href="EmployeeServlet" class="button">View All Employees</a>
        </form>
      </div>
    </div>
</body> </html>
```

create an Employee model class:

```
// Employee.java import java.util.Date;  public class Employee {
    private int id; private String name; private String position;
    private double salary;
    private Date hireDate;
    // Constructors public Employee() {} public Employee(int id, String name, String position, double salary, Date hireDate)
    { this.id = id;
      this.name = name;
      this.position = position;
      this.salary = salary;
      this.hireDate = hireDate;
```

```
    }
    // Getters and Setters
    public int getId() { return
    id; } public void setId(int
    id) { this.id = id;
    }
    public String getName() { return
       name;
    }
    public void setName(String name) {
       this.name = name;
    }
    public String getPosition() { return
       position;
    } public void setPosition(String position)
    { this.position = position;
    }
    public double getSalary() { return
       salary;

    }
    public void setSalary(double salary) {
       this.salary = salary;
    }
    public Date getHireDate() { return
       hireDate;
    }
    public void setHireDate(Date hireDate) { this.hireDate
       = hireDate;
    }
}
```

create the EmployeeServlet: //

```
EmployeeServlet.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
java.text.SimpleDateFormat;
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("/EmployeeServlet") public class
EmployeeServlet 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();
    String empIdParam = request.getParameter("empId"); Connection
    conn = null;
    try {
      conn = DBUtil.getConnection();
      List<Employee> employees = new ArrayList<>(); if
      (empIdParam != null && !empIdParam.trim().isEmpty()) {
        // Search for specific employee int empId =
        Integer.parseInt(empIdParam);
```

```
        PreparedStatement pstmt = conn.prepareStatement(
                "SELECT * FROM employees WHERE id = ?");
        pstmt.setInt(1, empId);
        ResultSet rs = pstmt.executeQuery(); while
        (rs.next()) {
            Employee emp = new Employee();
            emp.setId(rs.getInt("id"));
            emp.setName(rs.getString("name"));
            emp.setPosition(rs.getString("position"));
            emp.setSalary(rs.getDouble("salary"));
            emp.setHireDate(rs.getDate("hire_date"));
            employees.add(emp);
        } rs.close();
        pstmt.close()
        ;
    } else {
        // Fetch all employees
        PreparedStatement pstmt = conn.prepareStatement("SELECT * FROM employees");
        ResultSet rs = pstmt.executeQuery(); while (rs.next()) {

            Employee emp = new Employee();
            emp.setId(rs.getInt("id"));
            emp.setName(rs.getString("name"));
            emp.setPosition(rs.getString("position"));
            emp.setSalary(rs.getDouble("salary"));
            emp.setHireDate(rs.getDate("hire_date"));
            employees.add(emp);
        }
        rs.close();
        pstmt.close();
    }
```

```
// Generate HTML output out.println("<!DOCTYPE html>"); out.println("<html>");
out.println("<head>"); out.println("<title>Employee Directory</title>");
out.println("<style>"); out.println("body { font-family: Arial, sans-serif; margin: 40px;
}"); out.println(".container { width: 80%; max-width: 800px; margin: 0 auto; }");
out.println("table { width: 100%; border-collapse: collapse; }"); out.println("th, td {
padding: 10px; text-align: left; border-bottom: 1px solid #ddd; }"); out.println("th {
background-color: #f2f2f2; }");
out.println(".search-box { padding: 20px; background-color: #f5f5f5; border-radius: 5px; margin-bottom:
20px; }");
out.println("input[type=\"text\"] { padding: 8px; width: 200px; }");
out.println("button { padding: 8px 15px; background-color: #4CAF50; color: white; border: none; cursor:
pointer; }");
out.println("a.button { padding: 8px 15px; background-color: #2196F3; color: white; text-decoration: none;
border-radius: 3px; margin-left: 10px; display: inline-block; }");
out.println(".no-results { background-color: #f8d7da; color: #721c24; padding: 15px; border-radius: 5px;
}");
out.println("</style>");
out.println("</head>");
out.println("<body>");
out.println("<div class='container'>");
out.println("<h1>Employee Directory</h1>");

out.println("<div class='search-box'>");
out.println("<h3>Search Employee by ID</h3>");
out.println("<form action='EmployeeServlet'
method='get'>");
out.println("<input type='text' name='empId' placeholder='Enter Employee ID' value='" +
(empIdParam != null ? empIdParam : "") + "'>"); out.println("<button
type='submit'>Search</button>");
out.println("<a href='EmployeeServlet' class='button'>View All Employees</a>");
out.println("</form>"); out.println("</div>"); if (employees.isEmpty()) {
```

```
out.println("<div class='no-results'>"); out.println("<h3>No employees
found</h3>"); out.println("</div>");
} else { out.println("<table>");
out.println("<tr>");
out.println("<th>ID</th>");
out.println("<th>Name</th>");
out.println("<th>Position</th>");
out.println("<th>Salary</th>");
out.println("<th>Hire
Date</th>"); out.println("</tr>");
SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd"); for
(Employee emp : employees) {
out.println("<tr>");
out.println("<td>" + emp.getId() + "</td>");
out.println("<td>" + emp.getName() + "</td>"); out.println("<td>"
+ emp.getPosition() + "</td>");
out.println("<td>$" + String.format("%.2f", emp.getSalary()) + "</td>");
out.println("<td>" + dateFormat.format(emp.getHireDate()) + "</td>");
out.println("</tr>");
}
out.println("</table>");
}
out.println("</div>");
out.println("</body>");
out.println("</html>");
} catch (SQLException e) { out.println("<h3>Database Error: " +
e.getMessage() + "</h3>");
e.printStackTrace();
} catch (NumberFormatException e) { out.println("<h3>Invalid
Employee ID format</h3>");
```

```
        } finally {
            DBUtil.closeConnection(conn);
        }
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException { doGet(request, response);
    }
}
```

create a DBUtil class to manage database connections:

```
//      DBUtil.java      import
java.sql.Connection;     import
java.sql.DriverManager;
import java.sql.SQLException;
public class DBUtil {

    private static final String JDBC_URL = "jdbc:mysql://localhost:3306/employeedb";
    private static final String JDBC_USER = "root"; private static final String
    JDBC_PASSWORD = "password";
    static { try
        {
            Class.forName("com.mysql.cj.jdbc.Driver");
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
    public static Connection getConnection() throws SQLException {
        return DriverManager.getConnection(JDBC_URL, JDBC_USER, JDBC_PASSWORD);
    }
    public static void closeConnection(Connection conn) { if
        (conn != null) {
```

```
            try                 {
                conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}
```

Output


Employee Directory
Search Employee by ID
[ ] [Search] [View All Employees]

Discover. Learn. Empower.

DEPARTMENT OF

ENGINEERING

COMPUTER SCIENCE &

```
+-------------------------------------------------+
|            Employee Directory                   |
|                                                 |
| +---------------------------------------------+ |
| |          Search Employee by ID              | |
| |                                             | |
| | [                ] [Search] [View All Employees] |
| +---------------------------------------------+ |
|                                                 |
| +---------------------------------------------+ |
| | ID   | Name          | Position       | Salary   | Hire Date  |
| |------|---------------|----------------|----------|------------|
| | 101  | John Doe      | Software Engineer | $75000.00 | 2020-01-15 |
| | 102  | Jane Smith    | Project Manager   | $85000.00 | 2019-05-20 |
| | 103  | Bob Johnson   | UI/UX Designer    | $70000.00 | 2021-03-10 |
| | 104  | Alice Williams| Database Admin    | $80000.00 | 2018-11-05 |
| | 105  | Charlie Brown | System Analyst    | $72000.00 | 2020-09-25 |
| +---------------------------------------------+ |
|                                                 |
+-------------------------------------------------+
```

```
+-------------------------------------------------+
|            Employee Directory                   |
|                                                 |
| +---------------------------------------------+ |
| |          Search Employee by ID              | |
| |                                             | |
| | [102             ] [Search] [View All Employees] |
| +---------------------------------------------+ |
|                                                 |
| +---------------------------------------------+ |
| | ID   | Name          | Position       | Salary   | Hire Date  |
| |------|---------------|----------------|----------|------------|
| | 102  | Jane Smith    | Project Manager   | $85000.00 | 2019-05-20 |
| +---------------------------------------------+ |
|                                                 |
+-------------------------------------------------+
```

Discover. Learn. Empower.

DEPARTMENT OF ENGINEERING

Problem 3:

Develop a JSP-based student portal. Include a form for entering attendance details and save them to the database Code:

create the database structure: -

- Create students table

```
CREATE TABLE students ( student_id
   INT PRIMARY KEY, name
   VARCHAR(100) NOT NULL, class
   VARCHAR(20), section CHAR(1)
);
-- Create attendance table
CREATE TABLE attendance ( id INT PRIMARY
   KEY AUTO_INCREMENT,
   student_id INT, date DATE NOT NULL, status
   ENUM('Present', 'Absent', 'Late') NOT NULL,
   remarks VARCHAR(255),
```

```
    FOREIGN KEY (student_id) REFERENCES students(student_id)
);
```

-- Insert sample student data

INSERT INTO students VALUES (1001, 'Alex Johnson', '10', 'A');

INSERT INTO students VALUES (1002, 'Sophia Davis', '10', 'A');

INSERT INTO students VALUES (1003, 'Ethan Wilson', '10', 'B');

INSERT INTO students VALUES (1004, 'Olivia Martin', '10', 'B');

INSERT INTO students VALUES (1005, 'Noah Thompson', '10', 'A');

create a DBUtil class:

// com.studentportal.util.DBUtil.java package com.studentportal.util; import java.sql.Connection; import java.sql.DriverManager; import java.sql.SQLException; public class DBUtil { private static final String JDBC_URL = "jdbc:mysql://localhost:3306/studentportal"; private static final String JDBC_USER = "root"; private static final String JDBC_PASSWORD = "password";

```
static {
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
} public static Connection getConnection() throws SQLException { return
DriverManager.getConnection(JDBC_URL, JDBC_USER, JDBC_PASSWORD);
} public static void closeConnection(Connection conn)
{ if (conn != null) {
    try {
        conn.close();
    } catch (SQLException e) {
        e.printStackTrace();
```

```
        }
      }
    }
} create model
```

classes:

```
// com.studentportal.model.Student.java package com.studentportal.model;
public class Student { private int studentId; private String name; private String
className; private char section; // Constructors public Student() {} public
Student(int studentId, String name, String className, char section) {
this.studentId = studentId; this.name = name; this.className = className;
this.section = section;
    }
    // Getters and Setters
    public int getStudentId()
    { return studentId;
    } public void setStudentId(int studentId)
    { this.studentId = studentId;
    }

    public String getName() { return
        name;
    } public void setName(String name)
    { this.name = name;
    }
    public String getClassName() { return
        className;
    } public void setClassName(String className)
    { this.className = className;
    } public char getSection()
    { return section;
```

```
    } public void setSection(char section)
    { this.section = section;
    }
}
// com.studentportal.model.Attendance.java
package com.studentportal.model; import
java.util.Date; public class Attendance {
private int id; private int studentId; private
Date date; private String status; private
String remarks; // Constructors
    public Attendance() {}


    public Attendance(int id, int studentId, Date date, String status, String remarks) {
        this.id = id; this.studentId = studentId; this.date = date; this.status = status;
        this.remarks = remarks;
    }
    // Getters and Setters
    public int getId() { return id; } public
    void setId(int id) { this.id = id; } public
    int getStudentId() { return studentId; }
    public void setStudentId(int studentId) {
    this.studentId = studentId;
    } public Date getDate() { return
    date; } public void setDate(Date
    date) { this.date = date; } public
    String getStatus() { return status; }
    public void setStatus(String status) {
    this.status = status; } public String
    getRemarks() { return remarks;
    } public void setRemarks(String remarks)
    { this.remarks = remarks;
```

```
    }
}
```

create DAO (Data Access Object) classes: //

com.studentportal.dao.StudentDAO.java

```
package com.studentportal.dao; 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 com.studentportal.model.Student;
import com.studentportal.util.DBUtil; public
class StudentDAO { public List<Student>
getAllStudents() throws SQLException {
    List<Student> students = new ArrayList<>();
    Connection conn = null; try { conn =
    DBUtil.getConnection();
      PreparedStatement pstmt = conn.prepareStatement("SELECT * FROM students ORDER BY name");
      ResultSet rs = pstmt.executeQuery();
      while (rs.next()) {
        Student student = new Student();
        student.setStudentId(rs.getInt("student_id"));
        student.setName(rs.getString("name"));
        student.setClassName(rs.getString("class"));
        student.setSection(rs.getString("section").charAt(0));
        students.add(student);
      } rs.close();
      pstmt.close()
      ;
    } finally {
      DBUtil.closeConnection(conn);
```

```
    }           return
    students;
} public Student getStudentById(int studentId) throws
SQLException { Student student = null; Connection conn = null; try
{
        conn = DBUtil.getConnection();
        PreparedStatement pstmt = conn.prepareStatement("SELECT * FROM students WHERE student_id = ?");
        pstmt.setInt(1, studentId);
        ResultSet rs = pstmt.executeQuery(); if (rs.next()) {
        student = new Student();
        student.setStudentId(rs.getInt("student_id"));
        student.setName(rs.getString("name"));
        student.setClassName(rs.getString("class"));
        student.setSection(rs.getString("section").charAt(0));
        } rs.close();
        pstmt.close()
        ;
    } finally {
        DBUtil.closeConnection(conn);
    }


    return student;
}
public List<Student> getStudentsByClassAndSection(String className, char section) throws SQLException {
    List<Student> students = new ArrayList<>(); Connection conn = null; try { conn = DBUtil.getConnection();
        PreparedStatement pstmt = conn.prepareStatement(
                "SELECT * FROM students WHERE class = ? AND section = ? ORDER BY name");
        pstmt.setString(1, className); pstmt.setString(2, String.valueOf(section)); ResultSet rs =
        pstmt.executeQuery(); while (rs.next()) {
```

```
            Student student = new Student();
            student.setStudentId(rs.getInt("student_id"));
            student.setName(rs.getString("name"));
            student.setClassName(rs.getString("class"));
            student.setSection(rs.getString("section").charAt(0));
            students.add(student);
        } rs.close();
        pstmt.close()
        ;
    } finally {
        DBUtil.closeConnection(conn);
    }        return
    students;
  }
}
// com.studentportal.dao.AttendanceDAO.java package
com.studentportal.dao;

import java.sql.Connection; import java.sql.PreparedStatement; import
java.sql.ResultSet; import java.sql.SQLException; import java.util.ArrayList;
import java.util.Date; import java.util.List; import
com.studentportal.model.Attendance; import com.studentportal.util.DBUtil;
public class AttendanceDAO { public boolean saveAttendance(Attendance
attendance) throws SQLException { Connection conn = null; boolean success =
false; try {
        conn = DBUtil.getConnection();
        // Check if an entry already exists for this student on this date
        PreparedStatement checkStmt = conn.prepareStatement(
            "SELECT id FROM attendance WHERE student_id = ? AND date = ?");
        checkStmt.setInt(1, attendance.getStudentId()); checkStmt.setDate(2, new
        java.sql.Date(attendance.getDate().getTime())); ResultSet rs =
        checkStmt.executeQuery();
```

```
        if (rs.next()) {
            // Update existing record int
            id = rs.getInt("id");
            PreparedStatement updateStmt = conn.prepareStatement(
                    "UPDATE attendance SET status = ?, remarks = ? WHERE id = ?");
            updateStmt.setString(1, attendance.getStatus()); updateStmt.setString(2,
            attendance.getRemarks()); updateStmt.setInt(3, id);

            success = updateStmt.executeUpdate() > 0;
            updateStmt.close();
        } else {
            // Insert new record
            PreparedStatement insertStmt = conn.prepareStatement(
                    "INSERT INTO attendance (student_id, date, status, remarks) VALUES (?, ?, ?, ?)");
        insertStmt.setInt(1, attendance.getStudentId()); insertStmt.setDate(2, new
        java.sql.Date(attendance.getDate().getTime())); insertStmt.setString(3,
        attendance.getStatus()); insertStmt.setString(4, attendance.getRemarks()); success =
        insertStmt.executeUpdate() > 0; insertStmt.close();} rs.close(); checkStmt.close();
    } finally {
        DBUtil.closeConnection(conn);
    } return
    success; }

public List<Attendance> getAttendanceByDate(Date date) throws SQLException {
    List<Attendance> attendanceList = new ArrayList<>();
    Connection conn = null; try { conn =
    DBUtil.getConnection();
        PreparedStatement pstmt = conn.prepareStatement(
                "SELECT * FROM attendance WHERE date = ?");
        pstmt.setDate(1, new java.sql.Date(date.getTime()));
        ResultSet rs = pstmt.executeQuery(); while (rs.next()) {
```

```
                Attendance attendance = new Attendance();
                attendance.setId(rs.getInt("id"));
                attendance.setStudentId(rs.getInt("student_id"));
                attendance.setDate(rs.getDate("date"));
                attendance.setStatus(rs.getString("status"));
                attendance.setRemarks(rs.getString("remarks"));
                attendanceList.add(attendance);
            } rs.close();
            pstmt.close()
            ;
        } finally {
            DBUtil.closeConnection(conn);
        }                   return
        attendanceList;
    }
    public List<Attendance> getAttendanceByStudent(int studentId) throws SQLException {
        List<Attendance> attendanceList = new ArrayList<>();
        Connection conn = null; try { conn =
        DBUtil.getConnection();
            PreparedStatement pstmt = conn.prepareStatement(
"SELECT * FROM attendance WHERE student_id = ? ORDER BY date DESC"); pstmt.setInt(1,
                                studentId); ResultSet rs = pstmt.executeQuery(); while (rs.next()) {
                Attendance attendance = new Attendance();
                attendance.setId(rs.getInt("id"));
                attendance.setStudentId(rs.getInt("student_id"));
                attendance.setDate(rs.getDate("date"));
                attendance.setStatus(rs.getString("status"));
                attendance.setRemarks(rs.getString("remarks"));
                attendanceList.add(attendance);
            }

rs.close(); pstmt.close();
```

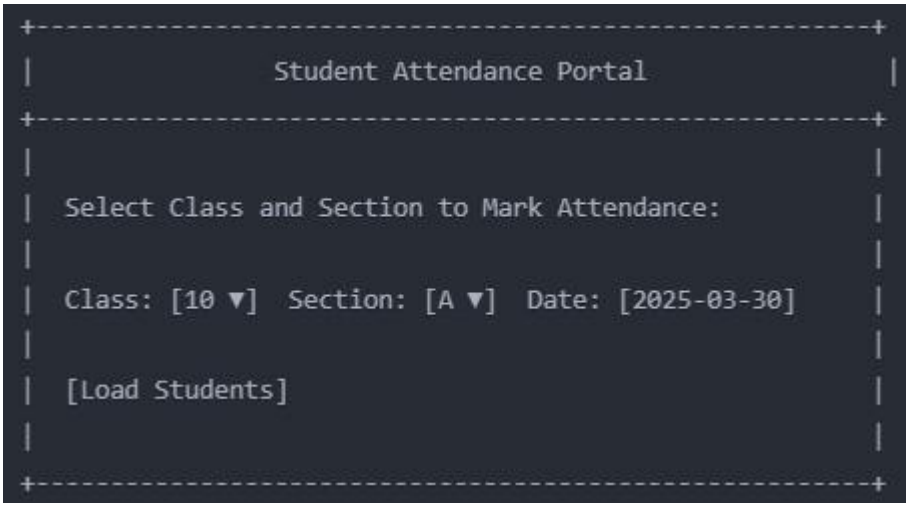
```
        } finally {
            DBUtil.closeConnection(conn);
        }                   return
        attendanceList;
    }
```

} **create the servlet to handle attendance submission:** //

com.studentportal.servlet.AttendanceServlet.java

```
package com.studentportal.servlet; import java.io.IOException; import java.sql.SQLException; import java.text.ParseException; import java.text.SimpleDateFormat; import java.util.Date; import javax.servlet.ServletException; import javax.servlet.annotation.WebServlet; import javax.servlet.http.HttpServlet; import javax.servlet.http.HttpServletRequest; import javax.servlet.http.HttpServletResponse;
```

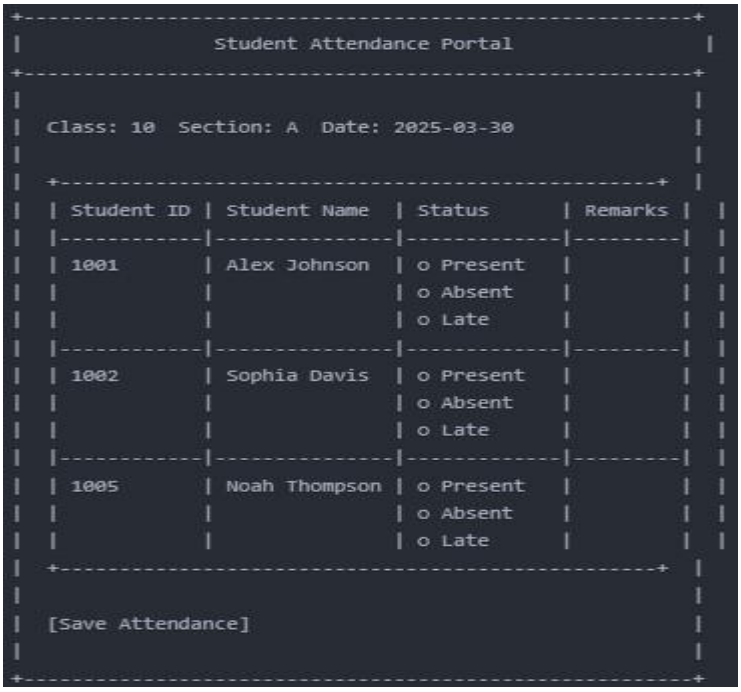
```
import com.studentportal.dao.AttendanceDAO;

import com.studentportal.model.Attendance; @WebServlet("/AttendanceServlet")
public class AttendanceServlet extends HttpServlet { private static final long
serialVersionUID = 1L; protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
    String dateStr = request.getParameter("date");
    String                  className                  =
    request.getParameter("class"); String section =
    request.getParameter("section"); try {
      SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
      Date date = dateFormat.parse(dateStr);
      String[] studentIds = request.getParameterValues("studentId");
```

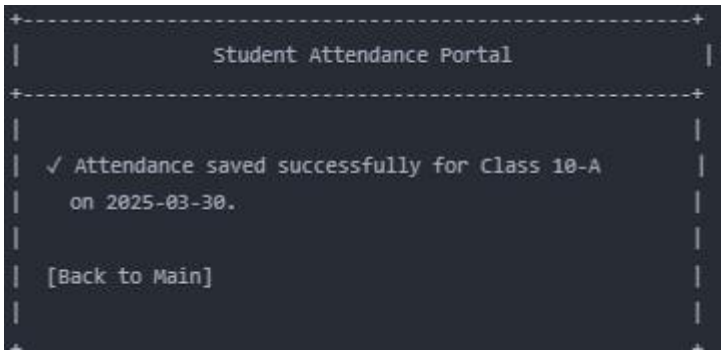


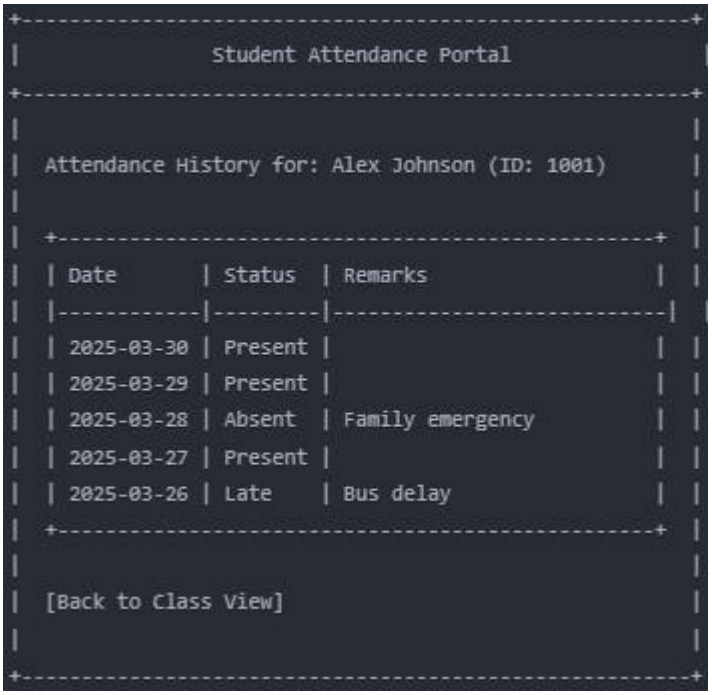

```
      String[] statuses = request.getParameterValues("status");
      String[] remarks = request.getParameterValues("remarks");
```

AttendanceDAO attendanceDAO = new Attendance

Output







DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

Learning Outcomes:

1. Basic servlet lifecycle and HTML form processing
2. JDBC integration with servlets for database operations
3. JSP implementation for dynamic web content generation
4. MVC architecture application in web development
5. Web application configuration and session management