Experiment8

Student Name: Aarav Singh	**UID: 22BCS13329**
Branch:CSE	**Section: 618(A)**
Semester: 6th	**DOP:04/04/2025**
Subject:PBLJ	**SubjectCode: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: DevelopwebapplicationsusingServletsandJSPforuserinputhandling, database integration.

Problem1.

Writeaservlettoacceptusercredentialsthroughan HTMLformanddisplaya personalized welcome message if the login is successful.

Code:

createtheHTMLloginform:

```
<!--login.html-->

<!DOCTYPEhtml>

<html>

<head>

  <title>LoginForm</title>

  <style>

    body{

      font-family:Arial,sans-serif;

      margin:40px;

    }

    .login-container{

      width: 300px;

      padding:20px;

      border:1pxsolid#ddd;

      border-radius:5px;

    }
```
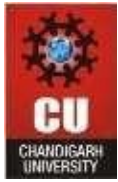
```
input[type="text"],input[type="password"]{

    width: 100%;

    padding:10px;

    margin:8px0;

    box-sizing:border-box;

}

input[type="submit"] {

    background-color:#4CAF50;

    color: white;

    padding:10px15px;

    border: none;cursor:

    pointer;

    width:100%;

}

</style>

</head>

<body>

    <divclass="login-container">

        <h2>UserLogin</h2>

        <formaction="LoginServlet"method="post">

            <labelfor="username">Username:</label>

            <inputtype="text"id="username"name="username"required>

            <labelfor="password">Password:</label>

            <inputtype="password"id="password"name="password"required>

            <inputtype="submit"value="Login">

        </form>

    </div>

</body>

</html>
```

createtheservlettohandlethelogin:

```
// LoginServlet.java

importjava.io.IOException;

import java.io.PrintWriter;
```

```java
import javax.servlet.ServletException;

importjavax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

importjavax.servlet.http.HttpServletResponse;

@WebServlet("/LoginServlet")

public class LoginServlet extends HttpServlet {

    privatestaticfinallongserialVersionUID=1L;

    //Hard-codedcredentialsfordemonstration

    private static final String VALID_USERNAME = "admin";

    privatestaticfinalStringVALID_PASSWORD="password";

    protectedvoiddoPost(HttpServletRequestrequest,HttpServletResponseresponse)

        throws ServletException, IOException {

        //Gettheformparameters

        Stringusername=request.getParameter("username");

        Stringpassword=request.getParameter("password");

        response.setContentType("text/html");

        PrintWriter out = response.getWriter();

        out.println("<!DOCTYPE html>");

        out.println("<html>");

        out.println("<head>");

        out.println("<title>LoginResult</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>");

        //Validatecredentials

        if(VALID_USERNAME.equals(username)&&VALID_PASSWORD.equals(password)){ out.println("<div

            class='message success'>");
```

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

        out.println("<p>Youhavesuccessfullyloggedin.</p>");

        out.println("</div>");

      }else{

        out.println("<divclass='messageerror'>");

        out.println("<h2>Login Failed</h2>");

        out.println("<p>Invalidusernameorpassword.Pleasetryagain.</p>");

        out.println("<a href='login.html'>Back to Login</a>");

        out.println("</div>");        }

      out.println("</body>");

      out.println("</html>");

   }

}
```

Output

```
+---------------------------+
|        User Login         |
|                           |
| Username:                 |
| [                     ]   |
|                           |
| Password:                 |
| [                     ]   |
|                           |
| [          Login      ]   |
+---------------------------+
```

```
+---------------------------+
|                           |
| Welcome, admin!           |
|                           |
| You have successfully     |
| logged in.                |
|                           |
```

```
+--------------------------+
|                          |
| Login Failed             |
|                          |
| Invalid username or      |
| password. Please try     |
| again.                   |
|                          |
| [Back to Login]          |
|                          |
+--------------------------+
```

Problem2:

CreateaservletintegratedwithJDBCtodisplayalistofemployeesfrom adatabase. Include a search form to fetch employee details by ID.

Code:

createthedatabasestructure:

-- Create employee table

CREATETABLEemployees(

  idINTPRIMARYKEY,

  nameVARCHAR(100)NOTNULL,

  positionVARCHAR(100),

  salary DECIMAL(10,2),

  hire_date DATE

);

--Insertsomesampledata

INSERTINTOemployeesVALUES(101, 'JohnDoe', 'SoftwareEngineer', 75000.00, '2020-01-15');

INSERT INTO employees VALUES (102, 'Jane Smith', 'Project Manager', 85000.00, '2019-05-20');

INSERTINTOemployeesVALUES(103,'BobJohnson','UI/UXDesigner',70000.00,'2021-03-10');

INSERTINTOemployeesVALUES(104,'AliceWilliams','DatabaseAdministrator',80000.00,'2018-11-05');

INSERT INTO employees VALUES (105, 'Charlie Brown', 'System Analyst', 72000.00, '2020-09-25');

createtheHTMLformforsearchingemployees:

<!--employeeSearch.html-->

<!DOCTYPEhtml>

```html
<html>

<head>

    <title>EmployeeSearch</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{
```
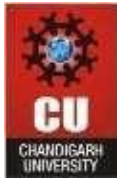
```
            padding: 8px 15px;

            background-color:#2196F3;

            color: white;

            text-decoration:none;

            border-radius: 3px;

            margin-left: 10px;

        }

    </style>

</head>

<body>

    <div class="container">

        <h1>EmployeeDirectory</h1>

        <divclass="search-box">

            <h3>SearchEmployeebyID</h3>

            <formaction="EmployeeServlet"method="get">

                <inputtype="text"name="empId"placeholder="EnterEmployeeID">

                <buttontype="submit">Search</button>

                <ahref="EmployeeServlet"class="button">ViewAllEmployees</a>

            </form>

        </div>

    </div>

</body>

</html>
```
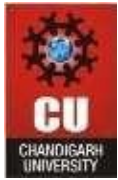
createanEmployeemodelclass:

```
//

Employee.javaimportja

va.util.Date;

publicclassEmployee{

    private int id;

    privateStringname;

    privateStringposition;
```

private double salary;

```java
privateDatehireDate;

//

ConstructorspublicE

mployee(){}

publicEmployee(intid,Stringname,Stringposition,doublesalary,DatehireDate){ this.id

    = id;

    this.name = name;

    this.position = position;

    this.salary = salary;

    this.hireDate=hireDate;

}
//GettersandSetters
public int getId() {

    returnid;

}
publicvoidsetId(intid){

    this.id = id;

}
publicStringgetName(){

    return name;

}
publicvoidsetName(Stringname){

    this.name = name;

}
publicStringgetPosition(){

    return position;

}
publicvoidsetPosition(Stringposition){

    this.position = position;

}
publicdoublegetSalary(){
```
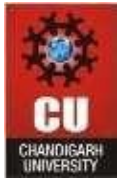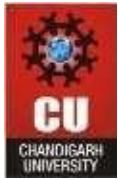
```
    return salary;
```

```java
    }

    publicvoidsetSalary(doublesalary){

        this.salary = salary;

    }

    publicDategetHireDate(){

        return hireDate;

    }

    publicvoidsetHireDate(DatehireDate){

        this.hireDate = hireDate;

    }

}
```

createtheEmployeeServlet:

```java
// EmployeeServlet.java

importjava.io.IOException;

import java.io.PrintWriter;

importjava.sql.Connection;

importjava.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

importjava.text.SimpleDateFormat;

import java.util.ArrayList;

importjava.util.List;

import javax.servlet.ServletException;

importjavax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

importjavax.servlet.http.HttpServletResponse;

@WebServlet("/EmployeeServlet")

publicclassEmployeeServletextendsHttpServlet{

    private static final long serialVersionUID = 1L;

    protectedvoiddoGet(HttpServletRequestrequest,HttpServletResponseresponse)
```

```java
    throwsServletException,IOException{

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String empIdParam = request.getParameter("empId");

Connection conn = null;

try{

    conn = DBUtil.getConnection();

    List<Employee>employees=newArrayList<>();

    if(empIdParam!=null&&!empIdParam.trim().isEmpty()){

        //Searchforspecificemployee

        int empId = Integer.parseInt(empIdParam);

        PreparedStatementpstmt=conn.prepareStatement(

            "SELECT*FROMemployeesWHEREid=?");

        pstmt.setInt(1, empId);

        ResultSetrs=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{

        //Fetchallemployees

        PreparedStatementpstmt=conn.prepareStatement("SELECT*FROMemployees"); ResultSet

        rs = pstmt.executeQuery();

        while(rs.next()){
```

```java
        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("<!DOCTYPEhtml>");

out.println("<html>");

out.println("<head>");

out.println("<title>EmployeeDirectory</title>");

out.println("<style>");

out.println("body { font-family: Arial, sans-serif; margin: 40px; }");

out.println(".container{width:80%;max-width:800px;margin:0auto;}");

out.println("table { width: 100%; border-collapse: collapse; }");

out.println("th,td{padding:10px;text-align:left;border-bottom:1pxsolid#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:8px15px;background-color:#4CAF50;color:white;border:none;cursor: pointer;
}");

out.println("a.button{padding:8px15px;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>");
```
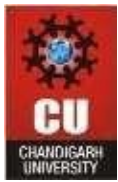
```java
out.println("</head>");
out.println("<body>");
out.println("<divclass='container'>");
out.println("<h1>EmployeeDirectory</h1>");

out.println("<div class='search-box'>");
out.println("<h3>Search Employee by ID</h3>");
out.println("<form action='EmployeeServlet' method='get'>");
out.println("<inputtype='text'name='empId'placeholder='EnterEmployeeID'value='"+
        (empIdParam != null ? empIdParam : "") + "'>");
out.println("<buttontype='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("<divclass='no-results'>");
    out.println("<h3>Noemployeesfound</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>HireDate</th>");
    out.println("</tr>");
    SimpleDateFormatdateFormat=newSimpleDateFormat("yyyy-MM-dd"); for
    (Employee emp : employees) {
        out.println("<tr>");
        out.println("<td>"+emp.getId()+"</td>");
```
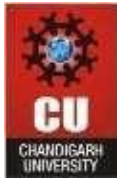
```java
            out.println("<td>" + emp.getName() + "</td>");

            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(SQLExceptione){

        out.println("<h3>DatabaseError:"+e.getMessage()+"</h3>");

        e.printStackTrace();

    } catch (NumberFormatException e) {

        out.println("<h3>InvalidEmployeeIDformat</h3>");

    } finally {

        DBUtil.closeConnection(conn);

    }

}

protectedvoiddoPost(HttpServletRequestrequest,HttpServletResponseresponse)
        throws ServletException, IOException {

    doGet(request,response);

}
}
```
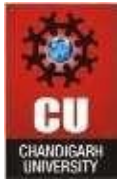
createaDBUtilclasstomanagedatabaseconnections:

```java
//DBUtil.java

import java.sql.Connection;

importjava.sql.DriverManager;

importjava.sql.SQLException;

public class DBUtil {
```

```java
privatestaticfinalStringJDBC_URL="jdbc:mysql://localhost:3306/employeedb";

private static final String JDBC_USER = "root";

privatestaticfinalStringJDBC_PASSWORD="password";

static {

    try{

        Class.forName("com.mysql.cj.jdbc.Driver");

    }catch(ClassNotFoundExceptione){

        e.printStackTrace();

    }

}

publicstaticConnectiongetConnection()throwsSQLException{

    returnDriverManager.getConnection(JDBC_URL,JDBC_USER,JDBC_PASSWORD);

}

publicstaticvoidcloseConnection(Connectionconn){ if

    (conn != null) {

        try{

            conn.close();

        }catch(SQLExceptione){

            e.printStackTrace();

        }

    }

}

}
```

Output

```
+----------------------------------------------+
|                Employee Directory            |
|                                              |
| +------------------------------------------+ |
| |          Search Employee by ID           | |
| |                                          | |
| | [                 ] [Search] [View All Employees] |
| +------------------------------------------+ |
|                                              |
+----------------------------------------------+
```

```
+--------------------------------------------------+
|              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 |
| +----------------------------------------------+ |
|                                                  |
+--------------------------------------------------+
```

Problem3:

DevelopaJSP-basedstudentportal.Includeaformforenteringattendancedetails and save them to the database

Code:

createthedatabasestructure:

```
-- Create students table
CREATETABLEstudents(
    student_id INT PRIMARY KEY,
    nameVARCHAR(100)NOTNULL,
    class VARCHAR(20),
    sectionCHAR(1)
);
-- Create attendance table
CREATETABLEattendance(
    idINTPRIMARYKEYAUTO_INCREMENT,
    student_idINT,
    dateDATENOTNULL,
    statusENUM('Present','Absent','Late')NOTNULL,
    remarks VARCHAR(255),
    FOREIGNKEY(student_id)REFERENCESstudents(student_id)
);
```
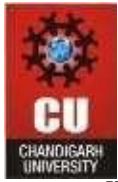
--Insertsamplestudentdata

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');

INSERTINTOstudentsVALUES(1005,'NoahThompson','10','A');

create a DBUtil class:

```java
//com.studentportal.util.DBUtil.java

package com.studentportal.util;

import java.sql.Connection;

importjava.sql.DriverManager;

import  java.sql.SQLException;

public class DBUtil {

  privatestaticfinalStringJDBC_URL="jdbc:mysql://localhost:3306/studentportal";

  private static final String JDBC_USER = "root";

  privatestaticfinalStringJDBC_PASSWORD="password";

  static {

    try{

      Class.forName("com.mysql.cj.jdbc.Driver");

    }catch(ClassNotFoundExceptione){

      e.printStackTrace();

    }

  }

  publicstaticConnectiongetConnection()throwsSQLException{

    returnDriverManager.getConnection(JDBC_URL,JDBC_USER,JDBC_PASSWORD);

  }

  publicstaticvoidcloseConnection(Connectionconn){ if

    (conn != null) {

      try{

        conn.close();

      }catch(SQLExceptione){

        e.printStackTrace();

      }

    }
```
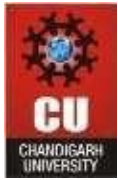
```
        }

    }

}
```

createmodelclasses:

//com.studentportal.model.Student.java

package com.studentportal.model;

```java
public class Student {

    private int studentId;

    private String name;

    privateStringclassName;

    private char section;

    // Constructors

    publicStudent(){}

    publicStudent(intstudentId,Stringname,StringclassName,charsection){

        this.studentId = studentId;

        this.name = name;

        this.className=className;

        this.section=section;

    }

    // Getters and Setters

    publicintgetStudentId(){

        returnstudentId;

    }

    publicvoidsetStudentId(intstudentId){

        this.studentId = studentId;

    }


    publicStringgetName(){

        returnname;

    }

    publicvoidsetName(Stringname) {

        this.name = name;

    }
```
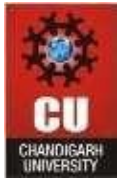
```java
publicStringgetClassName(){

    returnclassName;

}

publicvoidsetClassName(StringclassName){

    this.className=className;

}

publicchargetSection(){

    return section;

}

publicvoidsetSection(charsection){ this.section

    = section;

}

}
//com.studentportal.model.Attendance.java

package com.studentportal.model;

import java.util.Date;

publicclassAttendance{

    private int

    id;privateintstudentI

    d; private Date date;

    privateStringstatus;

    privateStringremarks;

    // Constructors

    publicAttendance(){}


    publicAttendance(intid,intstudentId,Datedate,Stringstatus,Stringremarks){ this.id =

        id;

        this.studentId=studentId;

        this.date = date;this.status

        = status;

        this.remarks=remarks;

    }
//GettersandSetters
```

```java
publicintgetId(){

    returnid;

}

publicvoidsetId(intid){

    this.id=id;

}

publicintgetStudentId(){

    return studentId;

}

publicvoidsetStudentId(intstudentId){

    this.studentId = studentId;

}

publicDategetDate(){

    return date;

}

publicvoidsetDate(Datedate) {

    this.date= date;

}

publicStringgetStatus(){

    returnstatus;

}

publicvoidsetStatus(Stringstatus){

    this.status= status;

}

publicStringgetRemarks(){

    return remarks;

}

publicvoidsetRemarks(Stringremarks){

    this.remarks = remarks;

}

}
```

createDAO(DataAccessObject)classes:

// com.studentportal.dao.StudentDAO.java
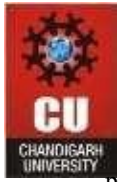
```java
packagecom.studentportal.dao;

import

java.sql.Connection;importjava.sql

.PreparedStatement; import

java.sql.ResultSet;

importjava.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

importcom.studentportal.model.Student;

import com.studentportal.util.DBUtil;

public class StudentDAO {

    publicList<Student>getAllStudents()throwsSQLException{ List<Student>

        students = new ArrayList<>();

        Connectionconn=null;

        try {

            conn=DBUtil.getConnection();

            PreparedStatementpstmt=conn.prepareStatement("SELECT*FROMstudentsORDERBYname");

            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);

        }

        returnstudents;

    }
```

```java
public Student getStudentById(int studentId) throws SQLException {
```
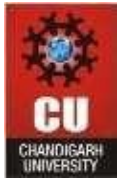
```java
        Student student = null;

        Connectionconn=null;

        try {

            conn=DBUtil.getConnection();

            PreparedStatementpstmt=conn.prepareStatement("SELECT*FROMstudentsWHEREstudent_id=?"); pstmt.setInt(1,

            studentId);

            ResultSetrs=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);

        }


        returnstudent;

    }
    publicList<Student>getStudentsByClassAndSection(StringclassName,charsection)throwsSQLException{ List<Student>

        students = new ArrayList<>();

        Connectionconn=null;

        try {

            conn = DBUtil.getConnection();

            PreparedStatementpstmt=conn.prepareStatement(

                "SELECT*FROMstudentsWHEREclass=?ANDsection=?ORDERBYname"); pstmt.setString(1,

            className);

            pstmt.setString(2,String.valueOf(section));

            ResultSet rs = pstmt.executeQuery();

            while(rs.next()){
```

```java
            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);

    }

    returnstudents;

   }

}
//com.studentportal.dao.AttendanceDAO.java

package com.studentportal.dao;


importjava.sql.Connection;

importjava.sql.PreparedStatement;

import java.sql.ResultSet;

importjava.sql.SQLException;

import java.util.ArrayList;

import java.util.Date;

importjava.util.List;

importcom.studentportal.model.Attendance;

import com.studentportal.util.DBUtil;public

class AttendanceDAO {

   publicbooleansaveAttendance(Attendanceattendance)throwsSQLException{

      Connection conn = null;

      booleansuccess=false;

      try {

         conn=DBUtil.getConnection();
```

```java
//Checkifanentryalreadyexistsforthisstudentonthisdate
PreparedStatement checkStmt = conn.prepareStatement(
        "SELECTidFROMattendanceWHEREstudent_id=?ANDdate=?");
checkStmt.setInt(1, attendance.getStudentId());
checkStmt.setDate(2,newjava.sql.Date(attendance.getDate().getTime()));
ResultSet rs = checkStmt.executeQuery();
if(rs.next()){
    //Updateexistingrecord
    int id = rs.getInt("id");
    PreparedStatementupdateStmt=conn.prepareStatement(
            "UPDATEattendanceSETstatus=?,remarks=?WHEREid=?");
    updateStmt.setString(1, attendance.getStatus());
    updateStmt.setString(2,attendance.getRemarks());
    updateStmt.setInt(3, id);


    success=updateStmt.executeUpdate()>0;
    updateStmt.close();
}else{
    //Insertnewrecord
    PreparedStatementinsertStmt=conn.prepareStatement(
            "INSERTINTOattendance(student_id,date,status,remarks)VALUES(?,?,?,?)");
    insertStmt.setInt(1, attendance.getStudentId());
    insertStmt.setDate(2,newjava.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);
}
returnsuccess;
```

```java
}
publicList<Attendance>getAttendanceByDate(Datedate)throwsSQLException{

    List<Attendance> attendanceList = new ArrayList<>();

    Connectionconn=null;

    try {

        conn = DBUtil.getConnection();

        PreparedStatementpstmt=conn.prepareStatement(

            "SELECT*FROMattendanceWHEREdate=?");

        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);

    }

    returnattendanceList;

}
publicList<Attendance>getAttendanceByStudent(intstudentId)throwsSQLException{

    List<Attendance> attendanceList = new ArrayList<>();

    Connectionconn=null;

    try {

        conn = DBUtil.getConnection();

        PreparedStatementpstmt=conn.prepareStatement(

            "SELECT*FROMattendanceWHEREstudent_id=?ORDERBYdateDESC");
```
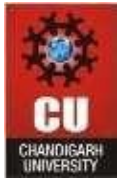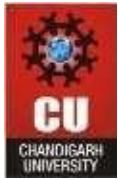
```java
        pstmt.setInt(1,studentId);

        ResultSetrs=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);

    }

    returnattendanceList;

  }

}
```

createtheservlettohandleattendancesubmission:

```java
//com.studentportal.servlet.AttendanceServlet.java

package com.studentportal.servlet;

import java.io.IOException;

import java.sql.SQLException;

importjava.text.ParseException;

importjava.text.SimpleDateFormat;

import java.util.Date;

import javax.servlet.ServletException;

importjavax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

importjavax.servlet.http.HttpServletResponse;
```

```java
importcom.studentportal.dao.AttendanceDAO;

importcom.studentportal.model.Attendance;

@WebServlet("/AttendanceServlet")

publicclassAttendanceServletextendsHttpServlet{

    private static final long serialVersionUID = 1L;

    protectedvoiddoPost(HttpServletRequestrequest,HttpServletResponseresponse)

        throws ServletException, IOException {

    String dateStr = request.getParameter("date");

    String className = request.getParameter("class");

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

    try {

        SimpleDateFormatdateFormat=newSimpleDateFormat("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

```
+--------------------------------------------------------+
|                Student Attendance Portal               |
+--------------------------------------------------------+
|                                                        |
|  Select Class and Section to Mark Attendance:          |
|                                                        |
|  Class: [10 ▼]  Section: [A ▼]  Date: [2025-03-30]     |
|                                                        |
|  [Load Students]                                       |
|                                                        |
+--------------------------------------------------------+
```

```
+------------------------------------------------------+
|                Student Attendance Portal             |
+------------------------------------------------------+
|                                                      |
|  Class: 10  Section: A  Date: 2025-03-30             |
|                                                      |
|  +------------------------------------------------+  |
|  | Student ID | Student Name | Status     | Remarks|  |
|  |------------|--------------|------------|--------|  |
|  | 1001       | Alex Johnson | o Present  |        |  |
|  |            |              | o Absent   |        |  |
|  |            |              | o Late     |        |  |
|  |------------|--------------|------------|--------|  |
|  | 1002       | Sophia Davis | o Present  |        |  |
|  |            |              | o Absent   |        |  |
|  |            |              | o Late     |        |  |
|  |------------|--------------|------------|--------|  |
|  | 1005       | Noah Thompson| o Present  |        |  |
|  |            |              | o Absent   |        |  |
|  |            |              | o Late     |        |  |
|  +------------------------------------------------+  |
|                                                      |
|  [Save Attendance]                                   |
|                                                      |
+------------------------------------------------------+
```

```
+------------------------------------------------------+
|                Student Attendance Portal             |
+------------------------------------------------------+
|                                                      |
|  √ Attendance saved successfully for Class 10-A      |
|     on 2025-03-30.                                    |
|                                                      |
|  [Back to Main]                                      |
|                                                      |
+------------------------------------------------------+
```

```
+------------------------------------------------------+
|                Student Attendance Portal             |
+------------------------------------------------------+
|                                                      |
|  Attendance History for: Alex Johnson (ID: 1001)     |
|                                                      |
|  +------------------------------------------------+  |
|  | Date       | Status  | Remarks                 |  |
|  |------------|---------|-------------------------|  |
|  | 2025-03-30 | Present |                         |  |
|  | 2025-03-29 | Present |                         |  |
|  | 2025-03-28 | Absent  | Family emergency        |  |
|  | 2025-03-27 | Present |                         |  |
|  | 2025-03-26 | Late    | Bus delay               |  |
|  +------------------------------------------------+  |
|                                                      |
|  [Back to Class View]                                |
|                                                      |
+------------------------------------------------------+
```

LearningOutcomes:

1. BasicservletlifecycleandHTMLform processing
2. JDBCintegrationwithservletsfordatabaseoperations
3. JSPimplementation fordynamicwebcontent generation
4. MVCarchitectureapplicationinwebdevelopment
5. Webapplicationconfigurationandsessionmanagement