Views(JSP pages)

index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Learner's Academy</title>
</head>
<body>
      <h1>Welcome to Leaner's Academy Admin portal</h1>
      <form action="login" method="post">

            <table>
                  <tr>
                        <td>Username</td>
                        <td><input type="text" name="username"
placeholder="Username" required/></td>
                  </tr>
                  <tr>
                        <td>Password</td>
                        <td><input type="password" name="password"
placeholder="password" required/></td>
                  </tr>
                  <tr>
                        <td colspan="2"><input type="submit"
value="Login"/></td>
                  </tr>
            </table>


      </form>
</body>
</html>
```

loginerror.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
if (session.getAttribute("isValid") == null)
      response.sendRedirect("index.jsp");
      //System.out.println(session.getAttribute("isValid"));
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Invalid login page</title>
</head>
<body>
      <h1>Invalid login deatils</h1>
      <%
            session.removeAttribute("isValid");
    %>
    <a href="index.jsp">Go to login</a>
</body>
</html>
```

addclass.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
if (session.getAttribute("isValid") == null || !(Boolean)
session.getAttribute("isValid"))
    response.sendRedirect("index.jsp");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Add Class</title>
<script>
    //    alert(new Date());
    function validateForm() {
        let cid = document.getElementById("cid").value;
        let cname = document.getElementById("cname").value;
        let isValid = true;

        if (cid.length > 6) {
            alert("Class ID length should be less than or equal to 6.
Please enter valid id");
            isValid = false;
        }
        if (cname.length > 10) {
            alert("Class name length should be less than or equal to
10. Please enter valid name");
            isValid = false;
        }
        return isValid;
    }
</script>

</head>
<body>
    <h1>Enter class details to add</h1>
    <form action="addclass" method="post" onsubmit="return validateForm()">
        <table>
            <tr>
                <td>New Class ID:</td>
                <td><input type="text" name="classid"
                    placeholder="Class ID" id="cid" required
/></td>
            </tr>
            <tr>
                <td>Subject name:</td>
                <td><input type="text" name="classname"
                    placeholder="Class Name" id="cname" required
/></td>
            </tr>
            <tr>
				<td colspan="2"><input type="submit" value="Add 
Class" /></td>
			</tr>
		</table>
	</form>
</body>
</html>
```

addstudent.jsp

```
<%@page import="java.sql.ResultSet"%>
<%@page import="java.util.ArrayList"%>
<%@page import="java.sql.PreparedStatement"%>
<%@page import="com.scode.util.ConnectionUtil"%>
<%@page import="java.sql.Connection"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
	pageEncoding="UTF-8"%>
<%
if (session.getAttribute("isValid") == null || !(Boolean) 
session.getAttribute("isValid"))
	response.sendRedirect("index.jsp");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Add Student</title>
<script>
	//	alert(new Date());
	function validateForm() {
		let stuid = document.getElementById("stuid").value;
		let stuclass = document.getElementById("classid").value;
		//alert(stuclass);
		let isValid = true;

		if (stuid.length > 10) {
			alert("Student ID length should be less than or equal to 
10. Please enter valid id");
			isValid = false;
		}
		if(stuclass == "-1"){
			alert("Please select a class");
			isValid = false;
		}
		return isValid;
	}
</script>

</head>
<body>
	<h1>Enter student details to add</h1>
	<form action="addstudent" method="post"
		onsubmit="return validateForm()">
		<table>
			<tr>
```

```
                        <td>New Student ID:</td>
                        <td><input type="text" name="studentid"
                              placeholder="Student ID" id="stuid" required
/></td>
                  </tr>
                  <tr>
                        <td>Student name:</td>
                        <td><input type="text" name="studentname"
                              placeholder="Student Name" id="stuname"
required /></td>
                  </tr>
                  <tr>
                        <td>Class ID:</td>
                        <td><select name="classid" id="classid" required>
                                    <option value="-1">Select</option>
                                    <%
                                    Connection con =
ConnectionUtil.getConnection();
                                    PreparedStatement stmt =
con.prepareStatement("select * from classes");
                                    ArrayList<String> classes = new
ArrayList<String>();
                                    ResultSet rs = stmt.executeQuery();
                                    while(rs.next()){
                                          %>
                                          <option value="<%=
rs.getString(1)%>"><%= rs.getString(1) %></option>
                                          <%
                                    }
                                    con.close();
                                    stmt.close();
                                    rs.close();
                                    %>
                        </select></td>
                  </tr>
                  <tr>
                        <td colspan="2"><input type="submit" value="Add
Student" /></td>
                  </tr>
            </table>
      </form>
</body>
</html>
```

addsubject.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>
<%
if (session.getAttribute("isValid") == null || !(Boolean)
session.getAttribute("isValid"))
      response.sendRedirect("index.jsp");
%>
<!DOCTYPE html>
<html>
```

```
<head>
<meta charset="UTF-8">
<title>Add Subject</title>
<script>
      //    alert(new Date());
      function validateForm() {
            let sid = document.getElementById("sid").value;
            let sname = document.getElementById("sname").value;
            let isValid = true;

            if (sid.length > 6) {
                  alert("Subject ID length should be less than or equal to 6. 
Please enter valid id");
                  isValid = false;
            }
            if (sname.length > 25) {
                  alert("Subject name length should be less than or equal to 
25. Please enter valid name");
                  isValid = false;
            }
            return isValid;
      }
</script>

</head>
<body>
      <h1>Enter subject details to add</h1>
      <form action="addsubject" method="post" onsubmit="return 
validateForm()">
            <table>
                  <tr>
                        <td>New Subject ID:</td>
                        <td><input type="text" name="subjectid"
                              placeholder="Subject ID" id="sid" required 
/></td>
                  </tr>
                  <tr>
                        <td>Subject name:</td>
                        <td><input type="text" name="subjectname"
                              placeholder="Subject Name" id="sname" required 
/></td>
                  </tr>
                  <tr>
                        <td colspan="2"><input type="submit" value="Add 
Subject" /></td>
                  </tr>
            </table>
      </form>
</body>
</html>
```

addteacher.jsp

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

```
<%
if (session.getAttribute("isValid") == null || !(Boolean)
session.getAttribute("isValid"))
      response.sendRedirect("index.jsp");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Add Teacher</title>
<script>
      //    alert(new Date());
      function validateForm() {
            let tid = document.getElementById("tid").value;
            let isValid = true;

            if (tid.length > 6) {
                  alert("Teacher ID length should be less than or equal to 6.
Please enter valid id");
                  isValid = false;
            }
            return isValid;
      }
</script>

</head>
<body>
      <h1>Enter teacher details to add</h1>
      <form action="addteacher" method="post" onsubmit="return
validateForm()">
            <table>
                  <tr>
                        <td>New Teacher ID:</td>
                        <td><input type="text" name="teacherid"
                              placeholder="Teacher ID" id="tid" required
/></td>
                  </tr>
                  <tr>
                        <td>Teacher name:</td>
                        <td><input type="text" name="teachername"
                              placeholder="Teacher Name" id="tname" required
/></td>
                  </tr>
                  <tr>
                        <td colspan="2"><input type="submit" value="Add
Teacher" /></td>
                  </tr>
            </table>
      </form>
</body>
</html>
```

assignsubject.jsp

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

```
if (session.getAttribute("isValid") == null || !(Boolean)
session.getAttribute("isValid"))
      response.sendRedirect("index.jsp");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Assign Subject to Class</title>
<script>
      //    alert(new Date());
      function validateForm() {
            let cid = document.getElementById("cid").value;
            let sid = document.getElementById("sid").value;
            let isValid = true;

            if (cid.length > 6) {
                  alert("Class ID length should be less than or equal to 6.
Please enter valid id");
                  isValid = false;
            }
            if (sid.length > 6) {
                  alert("Subject ID length should be less than or equal to 6.
Please enter valid id");
                  isValid = false;
            }
            return isValid;
      }
</script>

</head>
<body>
      <h1>Enter Subject ID and Class ID to assign</h1>
      <form action="assignsubject" method="post"
            onsubmit="return validateForm()">
            <table>
                  <tr>
                        <td>Class ID:</td>
                        <td><input type="text" name="classid"
placeholder="Class ID"
                              id="cid" required /></td>
                  </tr>
                  <tr>
                        <td>Subject ID:</td>
                        <td><input type="text" name="subjectid"
placeholder="Subject ID"
                              id="sid" required /></td>
                  </tr>
                  <tr>
                        <td colspan="2"><input type="submit" value="Assign
subject" /></td>
                  </tr>
            </table>
      </form>
</body>
</html>
```

assignteacher.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>
<%
if (session.getAttribute("isValid") == null || !(Boolean)
session.getAttribute("isValid"))
      response.sendRedirect("index.jsp");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Assign Teacher to Class and Subject</title>
<script>
      //    alert(new Date());
      function validateForm() {
            let tid = document.getElementById("tid").value;
            let cid = document.getElementById("cid").value;
            let sid = document.getElementById("sid").value;
            let isValid = true;

            if (tid.length > 6) {
                  alert("Teacher ID length should be less than or equal to 6.
Please enter valid id");
                  isValid = false;
            }
            if (cid.length > 6) {
                  alert("Class ID length should be less than or equal to 6.
Please enter valid id");
                  isValid = false;
            }
            if (sid.length > 6) {
                  alert("Subject ID length should be less than or equal to 6.
Please enter valid id");
                  isValid = false;
            }
            return isValid;
      }
</script>

</head>
<body>
      <h1>Enter Teacher ID and Class ID and Subject ID assign</h1>
      <form action="assignteacher" method="post"
            onsubmit="return validateForm()">
            <table>
                  <tr>
                        <td>Teacher ID:</td>
                        <td><input type="text" name="teacherid"
                              placeholder="Teacher ID" id="tid" required
/></td>
                  </tr>
                  <tr>
                        <td>Class ID:</td>
                        <td><input type="text" name="classid"
placeholder="Class ID"
```

```
                              id="cid" required /></td>
                  </tr>
                  <tr>
                        <td>Subject ID:</td>
                        <td><input type="text" name="subjectid"
placeholder="Subject ID"
                              id="sid" required /></td>
                  </tr>
                  <tr>
                        <td colspan="2"><input type="submit" value="Assign
Teacher" /></td>
                  </tr>
            </table>
      </form>
</body>
</html>
```

changepassword.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>
<%
if (session.getAttribute("isValid") == null || !(Boolean)
session.getAttribute("isValid"))
      response.sendRedirect("index.jsp");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Change password</title>
<script>

      function validate(){
            let password = document.getElementById("password").value;
            let isValid = true;
            if(password.length >10){
                  isValid = false;
                  alert("Password length should be less than or equeal to
10");
            }
            return isValid;
      }

</script>
</head>
<body>
      <h1>Enter a new password</h1>
      <form action="change" method="post" onsubmit="return validate()">
            <table>
                  <tr>
                        <td>New Password:</td>
                        <td><input type="password" name="password"
                              placeholder="New password" id="password"
required /></td>
                  </tr>
                  <tr>
```

```
                        <td colspan="2"><input type="submit" value="Change
password" /></td>
                  </tr>
            </table>
      </form>
</body>
</html>
```

dashboard.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>
<%
//System.out.println(session.getAttribute("isValid"));
if (session.getAttribute("isValid") == null || !(Boolean)
session.getAttribute("isValid"))
      response.sendRedirect("index.jsp");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Dashboard</title>
</head>
<body>
      <h1>Welcome to dashboard</h1>
      <table>
            <tr>
                  <td><a href="changepassword.jsp">Change password</a></td>
            </tr>
            <tr>
                  <td><a href="addsubject.jsp">Add subject</a></td>
            </tr>
            <tr>
                  <td><a href="addclass.jsp">Add class</a></td>
            </tr>
            <tr>
                  <td><a href="addteacher.jsp">Add teacher</a></td>
            </tr>
            <tr>
                  <td><a href="addstudent.jsp">Add student</a></td>
            </tr>
            <tr>
                  <td><a href="assignsubject.jsp">Subject assignment</a></td>
            </tr>
            <tr>
                  <td><a href="assignteacher.jsp">Teacher assignment</a></td>
            </tr>
            <tr>
                  <td><a href="showsubjects">Show subjects</a></td>
            </tr>
            <tr>
                  <td><a href="showteachers">Show teachers</a></td>
            </tr>
            <tr>
```

```
			<td><a href="showclasses">Show classes</a></td>
		</tr>
		<tr>
			<td><a href="showstudents">Show students</a></td>
		</tr>
		<tr>
			<td><a href="showteacherassignments">Show teacher
					assignments</a></td>
		</tr>
		<tr>
			<td><a href="showsubjectassignments">Show subject
					assignments</a></td>
		</tr>
		<tr>
			<td><a href="showclassinfo.jsp">Show class information
</a></td>
		</tr>
		<tr>
			<td><a href="logout">Logout</a></td>
		</tr>
	</table>
</body>
</html>
```

displayinfo.jsp

```
<%@page import="com.scode.admin.model.Student"%>
<%@page import="com.scode.admin.model.Subject"%>
<%@page import="com.scode.admin.model.Teacher"%>
<%@page import="com.scode.admin.model.Classes"%>
<%@page import="java.util.ArrayList"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
	pageEncoding="UTF-8"%>
<%
if (session.getAttribute("isValid") == null || !(Boolean)
session.getAttribute("isValid"))
	response.sendRedirect("index.jsp");
else if (request.getAttribute("teachers") == null ||
request.getAttribute("isFound") == null || request.getAttribute("students")
== null || request.getAttribute("subjects") == null) {
	response.sendRedirect("dashboard.jsp");
} else {
	boolean isFound = (Boolean) request.getAttribute("isFound");
	String tmessage = (String) request.getAttribute("tmessage");
	String smessage = (String) request.getAttribute("smessage");
	String stumessage = (String) request.getAttribute("stumessage");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Class details</title>
</head>
<body>
	<%
	if (!isFound) {
	%>
	<h1>Class not found</h1>
```

```
      <%
      } else {
      %>
      <h1><%=tmessage%></h1>
      <%
      ArrayList<Teacher> teachers = (ArrayList<Teacher>)
request.getAttribute("teachers");
      if (!teachers.isEmpty()) {
      %>
      <table>
            <tr>
                  <th>Teacher ID</th>
                  <th>Teacher Name</th>
            </tr>
            <%
            for (Teacher teacher : teachers) {
            %>
            <tr>
                  <td><%=teacher.getId()%></td>
                  <td><%=teacher.getName()%></td>
            </tr>
            <%
            }
            %>
      </table>
      <%
      }
      %>
      <h1><%=smessage%></h1>
      <%
      ArrayList<Subject> subjects = (ArrayList<Subject>)
request.getAttribute("subjects");
      if (!subjects.isEmpty()) {
      %>
      <table>
            <tr>
                  <th>Subject ID</th>
                  <th>Subject Name</th>
            </tr>
            <%
            for (Subject subject : subjects) {
            %>
            <tr>
                  <td><%=subject.getId()%></td>
                  <td><%=subject.getName()%></td>
            </tr>
            <%
            }
            %>
      </table>
      <%
      }
      %>
            <h1><%=stumessage%></h1>
      <%
      ArrayList<Student> students = (ArrayList<Student>)
request.getAttribute("students");
```

```
      if (!students.isEmpty()) {
      %>
      <table>
            <tr>
                  <th>Student ID</th>
                  <th>Student Name</th>
            </tr>
            <%
            for (Student stu : students) {
            %>
            <tr>
                  <td><%=stu.getId()%></td>
                  <td><%=stu.getName()%></td>
            </tr>
            <%
            }
            %>
      </table>
      <%
      }
      %>
<%
}
}
%>

      <a href="dashboard.jsp">Go to dashboard</a>
</body>
</html>
```

result.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>
<%
String message = (String) request.getAttribute("message");
if (session.getAttribute("isValid") == null || !(Boolean)
session.getAttribute("isValid"))
      response.sendRedirect("index.jsp");
else if(message == null)
      response.sendRedirect("dashboard.jsp");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title><%=message%></title>
</head>
<body>

      <h1><%=message%></h1>
      <a href="dashboard.jsp">Go to dashboard</a>
</body>
</html>
```

showclasses.jsp

```
<%@page import="com.scode.admin.model.Classes"%>
<%@page import="java.util.ArrayList"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>
<%
if (session.getAttribute("isValid") == null || !(Boolean)
session.getAttribute("isValid"))
      response.sendRedirect("index.jsp");
else if (request.getAttribute("classes") == null) {
      response.sendRedirect("dashboard.jsp");
} else {
      String message = (String) request.getAttribute("message");
      ArrayList<Classes> allclasses = (ArrayList<Classes>)
request.getAttribute("classes");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Classes</title>
</head>
<body>
      <h1><%=message%></h1>
      <%
      if (!allclasses.isEmpty()) {
      %>
      <table>
            <tr>
                  <th>Class ID</th>
                  <th>Class Name</th>
            </tr>
            <%
            for (Classes classes : allclasses) {
            %>
            <tr>
                  <td><%=classes.getId()%></td>
                  <td><%=classes.getName()%></td>
            </tr>
            <%
            }
            %>
      </table>

      <%
      }
      }
      %>
      <a href="dashboard.jsp">Go to dashboard</a>
</body>
</html>
```

showclassinfo.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>
<%
if (session.getAttribute("isValid") == null || !(Boolean)
session.getAttribute("isValid"))
      response.sendRedirect("index.jsp");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Search class</title>
<script>
      //    alert(new Date());
      function validateForm() {
            let cid = document.getElementById("cid").value;
            let isValid = true;

            if (cid.length > 6) {
                  alert("Class ID length should be less than or equal to 6.
Please enter valid id");
                  isValid = false;
            }
            return isValid;
      }
</script>

</head>
<body>
      <h1>Enter class id</h1>
      <form action="showclassinfo" method="post" onsubmit="return
validateForm()">
            <table>
                  <tr>
                        <td>Class ID:</td>
                        <td><input type="text" name="classid"
                              placeholder="Class ID" id="cid" required
/></td>
                  </tr>
                  <tr>
                        <td colspan="2"><input type="submit" value="Search
Class info" /></td>
                  </tr>
            </table>
      </form>
</body>
</html>
```

showstudents.jsp

```
<%@page import="com.scode.admin.model.Student"%>
<%@page import="java.util.ArrayList"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>
```

```
<%
if (session.getAttribute("isValid") == null || !(Boolean) 
session.getAttribute("isValid"))
      response.sendRedirect("index.jsp");
else if (request.getAttribute("students") == null) {
      response.sendRedirect("dashboard.jsp");
} else {
      String message = (String) request.getAttribute("message");
      ArrayList<Student> students = (ArrayList<Student>) 
request.getAttribute("students");
      //System.out.println(message);
      //System.out.println(students);
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Students</title>
</head>
<body>
      <h1><%=message%></h1>
      <%
      if (!students.isEmpty()) {
      %>
      <table>
            <tr>
                  <th>Student ID</th>
                  <th>Student Name</th>
            </tr>
            <%
            for (Student student : students) {
            %>
            <tr>
                  <td><%=student.getId()%></td>
                  <td><%=student.getName()%></td>
            </tr>
            <%
            }
            %>
      </table>

      <%
      }
      }
      %>
      <a href="dashboard.jsp">Go to dashboard</a>
</body>
</html>
```

showsubjectassign.jsp

```
<%@page import="com.scode.admin.model.SubjectAssignment"%>

<%@page import="java.util.ArrayList"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>
<%
```

```
if (session.getAttribute("isValid") == null || !(Boolean)
session.getAttribute("isValid"))
      response.sendRedirect("index.jsp");
else if (request.getAttribute("sas") == null) {
      response.sendRedirect("dashboard.jsp");
} else {
      String message = (String) request.getAttribute("message");
      ArrayList<SubjectAssignment> sas = (ArrayList<SubjectAssignment>)
request.getAttribute("sas");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Class Subject assignments</title>
</head>
<body>
      <h1><%=message%></h1>
      <%
      if (!sas.isEmpty()) {
      %>
      <table>
            <tr>
                  <th>Subject ID</th>
                  <th>Class ID</th>
            </tr>
            <%
            for (SubjectAssignment sa : sas) {
            %>
            <tr>
                  <td><%=sa.getSid()%></td>
                  <td><%=sa.getCid()%></td>
            </tr>
            <%
            }
            %>
      </table>
<%
}
}
%>
      <a href="dashboard.jsp">Go to dashboard</a>
</body>
</html>
```

showsubjects.jsp

```
<%@page import="com.scode.admin.model.Subject"%>
<%@page import="java.util.ArrayList"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>
<%
if (session.getAttribute("isValid") == null || !(Boolean)
session.getAttribute("isValid"))
      response.sendRedirect("index.jsp");
else if (request.getAttribute("subjects") == null) {
      response.sendRedirect("dashboard.jsp");
} else {
      String message = (String) request.getAttribute("message");
      ArrayList<Subject> subjects = (ArrayList<Subject>)
request.getAttribute("subjects");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Subjects</title>
</head>
<body>
      <h1><%=message%></h1>
      <%
      if (!subjects.isEmpty()) {
      %>
      <table>
            <tr>
                  <th>Subject ID</th>
                  <th>Subject Name</th>
            </tr>
            <%
            for (Subject subject : subjects) {
            %>
            <tr>
                  <td><%=subject.getId()%></td>
                  <td><%=subject.getName()%></td>
            </tr>
            <%
            }
            %>
      </table>

<%
}
}
%>
      <a href="dashboard.jsp">Go to dashboard</a>
</body>
</html>
```

showteachersassign.jsp

```
<%@page import="com.scode.admin.model.TeacherAssignment"%>

<%@page import="java.util.ArrayList"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>
<%
if (session.getAttribute("isValid") == null || !(Boolean)
session.getAttribute("isValid"))
      response.sendRedirect("index.jsp");
else if (request.getAttribute("tas") == null) {
      response.sendRedirect("dashboard.jsp");
} else {
      String message = (String) request.getAttribute("message");
      ArrayList<TeacherAssignment> ta = (ArrayList<TeacherAssignment>)
request.getAttribute("tas");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Teacher Class Subject assignments</title>
</head>
<body>
	<h1><%=message%></h1>
	<%
	if (!ta.isEmpty()) {
	%>
	<table>
		<tr>
			<th>Teacher ID</th>
			<th>Subject ID</th>
			<th>Class ID</th>
		</tr>
		<%
		for (TeacherAssignment t : ta) {
		%>
		<tr>
			<td><%=t.getTid()%></td>
			<td><%=t.getSid()%></td>
			<td><%=t.getCid()%></td>
		</tr>
		<%
		}
		%>
	</table>
<%
}
}
%>
	<a href="dashboard.jsp">Go to dashboard</a>
</body>
</html>
```

showteachers.jsp

```
<%@page import="com.scode.admin.model.Teacher"%>
<%@page import="java.util.ArrayList"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
	pageEncoding="UTF-8"%>
<%
if (session.getAttribute("isValid") == null || !(Boolean)
session.getAttribute("isValid"))
	response.sendRedirect("index.jsp");
else if (request.getAttribute("teachers") == null) {
	response.sendRedirect("dashboard.jsp");
} else {
	String message = (String) request.getAttribute("message");
	ArrayList<Teacher> teachers = (ArrayList<Teacher>)
request.getAttribute("teachers");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Teachers</title>
</head>
<body>
    <h1><%=message%></h1>
    <%
    if (!teachers.isEmpty()) {
    %>
    <table>
        <tr>
            <th>Teacher ID</th>
            <th>Teacher Name</th>
        </tr>
        <%
        for (Teacher teacher : teachers) {
        %>
        <tr>
            <td><%=teacher.getId()%></td>
            <td><%=teacher.getName()%></td>
        </tr>
        <%
        }
        %>
    </table>

<%
}
}
%>
    <a href="dashboard.jsp">Go to dashboard</a>
</body>
</html>
```

ConnectionUtil.java

```
package com.scode.util;


import java.sql.*;


import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.sql.DataSource;


public class ConnectionUtil {
    public static Connection getConnection() {
        Connection connection = null;
        try {
```

```
                        InitialContext context = new InitialContext();
                        DataSource dataSource = (DataSource) context.lookup("java:comp/env/myds");

                        connection = dataSource.getConnection();
//                      System.out.println(connection);
                } catch (NamingException e) {
                        e.printStackTrace();
                } catch (SQLException e) {
                        e.printStackTrace();
                }
                return connection;
        }
}
```

Models

Admin.java

```
package com.scode.admin.model;

import java.sql.*;

import com.scode.util.ConnectionUtil;

public class Admin {

        private String username;
        private String password;

        private Connection con;
        private PreparedStatement stmt,stmt1;
```

```
    public Admin() {
        try {
            con = ConnectionUtil.getConnection();
            stmt = con.prepareStatement("select * from admin where username = ?
and password = ?");
            stmt1 = con.prepareStatement("update admin set password = ?");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public void setUsername(String username) {
        this.username = username;
    }

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

    public boolean validate() {
        boolean result = false;
        try {
            stmt.setString(1, username);
            stmt.setString(2, password);
            ResultSet rs = stmt.executeQuery();
            result = rs.next();
```

```java
//			System.out.println(result);
		} catch (SQLException e) {
			e.printStackTrace();
		}
		return result;
	}

	public int updatePassword(String password) {
		int result = 0;
		try {
			stmt1.setString(1, password);
			result = stmt1.executeUpdate();
		} catch (SQLException e) {
			e.printStackTrace();
		}
		return result;
	}

	public void destroy() {
		try {
			con.close();
			stmt.close();
			stmt1.close();
		} catch (SQLException e) {
			e.printStackTrace();
		}
	}
}
```

Classes.java

```
package com.scode.admin.model;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import com.scode.util.ConnectionUtil;

public class Classes {
	private String id;
	private String name;

	private Connection con;
	private PreparedStatement stmt, stmt1, stmt2, stmt3;

	public Classes() {
		try {
			con = ConnectionUtil.getConnection();
			stmt = con.prepareStatement("select * from classes where id = ?");
			stmt1 = con.prepareStatement("select * from classes where name = ?");
			stmt2 = con.prepareStatement("insert into classes values(?,?)");
			stmt3 = con.prepareStatement("select * from classes");
```

```
    } catch (SQLException e) {
        e.printStackTrace();
    }

}

public String getId() {
    return id;
}

public void setId(String id) {
    this.id = id;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

@Override
public String toString() {
    return "Class [id=" + id + ", name=" + name + "]";
}
```

```
    public int addClass() {
        int result = 0;
        try {
            stmt.setString(1, id);
            ResultSet rs = stmt.executeQuery();
            if (rs.next())
                return -1;
            stmt1.setString(1, name);
            rs = stmt1.executeQuery();
            if (rs.next())
                return -2;

            stmt2.setString(1, id);
            stmt2.setString(2, name);
            result = stmt2.executeUpdate();
//          System.out.println(result);
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return result;
    }

    public void destroy() {
        try {
            con.close();
            stmt.close();
            stmt1.close();
            stmt2.close();
            stmt3.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public ArrayList<Classes> getClasses() {
        ArrayList<Classes> allclasses = new ArrayList<Classes>();
        try {
            ResultSet rs = stmt3.executeQuery();
            while (rs.next()) {
                Classes classes = new Classes();
                classes.setId(rs.getString(1));
                classes.setName(rs.getString(2));
                allclasses.add(classes);
                classes.destroy();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return allclasses;
    }



//  public ArrayList<Subject> getSubjects(String id) {
```

```
//
//  }
//
//  public ArrayList<Student> getStudents(String id) {
//
//  }

    public int findClass(String id) {
        int result = 0;
        try {
            stmt.setString(1, id);
            ResultSet rs = stmt.executeQuery();
            if (!rs.next())
                return -1;
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return result;
    }
}
```

Student.java

```
package com.scode.admin.model;


import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;


import com.scode.util.ConnectionUtil;


public class Student {
    private String id;
    private String name;
    private String classid;


    private Connection con;
    private PreparedStatement stmt, stmt1, stmt2,stmt3;


    public Student() {
        try {
            con = ConnectionUtil.getConnection();
            stmt = con.prepareStatement("select * from student where id = ?");
            stmt1 = con.prepareStatement("insert into student values(?,?,?)");
            stmt2 = con.prepareStatement("select * from student");
            stmt3 = con.prepareStatement("select * from student where class_id =
?");
        } catch (SQLException e) {
            e.printStackTrace();
        }


    }


    public String getId() {
```

```
    return id;
}

public void setId(String id) {
    this.id = id;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getClassid() {
    return classid;
}

public void setClassid(String classid) {
    this.classid = classid;
}

@Override
public String toString() {
    return "Student [id=" + id + ", name=" + name + ", classid=" + classid + "]";
}
```

```
public int addStudent() {
    int result = 0;
    try {
        stmt.setString(1, id);
        ResultSet rs = stmt.executeQuery();
        if (rs.next())
            return -1;

        stmt1.setString(1, id);
        stmt1.setString(2, name);
        stmt1.setString(3, classid);
        result = stmt1.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return result;
}

public void destroy() {
    try {
        con.close();
        stmt.close();
        stmt1.close();
        stmt2.close();
        stmt3.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public ArrayList<Student> getStudents() {
    ArrayList<Student> students = new ArrayList<Student>();
    try {
        ResultSet rs = stmt2.executeQuery();
        while (rs.next()) {
            Student student = new Student();
            student.setId(rs.getString(1));
            student.setName(rs.getString(2));
            student.setClassid(rs.getString(3));;
            students.add(student);
            student.destroy();
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return students;
}

public ArrayList<Student> getStudents(String Id) {
    ArrayList<Student> students = new ArrayList<Student>();
    try {
        stmt3.setString(1, Id);
        ResultSet rs = stmt3.executeQuery();
```

```
			while (rs.next()) {
				Student student = new Student();
				student.setId(rs.getString(1));
				student.setName(rs.getString(2));
				students.add(student);
				student.destroy();
			}
		} catch (SQLException e) {
			e.printStackTrace();
		}
		return students;
	}
}
```

Subject.java

```
package com.scode.admin.model;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import com.scode.util.ConnectionUtil;

public class Subject {
	private String id;
	private String name;
```

```
    }

    private Connection con;
    private PreparedStatement stmt, stmt1, stmt2, stmt3,stmt4;

    public Subject() {
        try {
            con = ConnectionUtil.getConnection();
            stmt = con.prepareStatement("select * from subjects where id = ?");
            stmt1 = con.prepareStatement("select * from subjects where name = ?");
            stmt2 = con.prepareStatement("insert into subjects values(?,?)");
            stmt3 = con.prepareStatement("select * from subjects");
            stmt4 = con.prepareStatement(
                    "select * from subjects where id in (select subject_id from
classsubjects where class_id = ?)");
        } catch (SQLException e) {
            e.printStackTrace();
        }

    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }
```

```
public String getName() {
	return name;
}

public void setName(String name) {
	this.name = name;
}

@Override
public String toString() {
	return "Subject [id=" + id + ", name=" + name + "]";
}

public int addSubject() {
	int result = 0;
	try {
		stmt.setString(1, id);
		ResultSet rs = stmt.executeQuery();
		if (rs.next())
			return -1;
		stmt1.setString(1, name);
		rs = stmt1.executeQuery();
		if (rs.next())
			return -2;

		stmt2.setString(1, id);
```

```
        stmt2.setString(2, name);
        result = stmt2.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return result;
}

public void destroy() {
    try {
        con.close();
        stmt.close();
        stmt1.close();
        stmt2.close();
        stmt3.close();
        stmt4.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public ArrayList<Subject> getSubjects() {
    ArrayList<Subject> subjects = new ArrayList<Subject>();
    try {
        ResultSet rs = stmt3.executeQuery();
        while (rs.next()) {
            Subject subject = new Subject();
            subject.setId(rs.getString(1));
            subject.setName(rs.getString(2));
            subjects.add(subject);
            subject.destroy();
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return subjects;
}

public ArrayList<Subject> getSubjects(String id) {
    ArrayList<Subject> subjects = new ArrayList<Subject>();
    try {
        stmt4.setString(1, id);
        ResultSet rs = stmt4.executeQuery();
        while (rs.next()) {
            Subject subject = new Subject();
            subject.setId(rs.getString(1));
            subject.setName(rs.getString(2));
            subjects.add(subject);
            subject.destroy();
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return subjects;
    }
}
```

SubjectAssignment.java

```
package com.scode.admin.model;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Random;

import com.scode.util.ConnectionUtil;

public class SubjectAssignment {
    private String sid;
    private String cid;

    private Connection con;
    private PreparedStatement stmt, stmt1, stmt2, stmt3, stmt4;

    public SubjectAssignment() {
        try {
            con = ConnectionUtil.getConnection();
            stmt = con.prepareStatement("select * from subjects where id = ?");
            stmt1 = con.prepareStatement("select * from classes where id = ?");
            stmt2 = con.prepareStatement("select * from classsubjects where subject_id = ?");
            stmt3 = con.prepareStatement("insert into classsubjects values (?,?,?)");
            stmt4 = con.prepareStatement("select * from classsubjects");
//          stmt3 = con.prepareStatement("select * from subjects");

        } catch (SQLException e) {
            e.printStackTrace();
        }

    }

    public String getSid() {
        return sid;
    }

    public void setSid(String sid) {
        this.sid = sid;
    }

    public String getCid() {
        return cid;
    }

    public void setCid(String cid) {
        this.cid = cid;
    }
    @Override
    public String toString() {
        return "SubjectAssignment [Subjetc ID=" + sid + ", is assigned to Class ID=" + cid + "]";
    }

    public int assignSubject() {
        int result = 0;
        try {
            stmt.setString(1, sid);
            ResultSet rs = stmt.executeQuery();
            if (!rs.next())
                return -1;
            stmt1.setString(1, cid);
            rs = stmt1.executeQuery();
            if (!rs.next())
                return -2;
            stmt2.setString(1, sid);
            rs = stmt2.executeQuery();
            if (rs.next())
                return -3;
            Random random = new Random();
            String characters = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
            int length = 6;
            StringBuilder randomText = new StringBuilder();
            for (int i = 0; i < length; i++) {
```

```
            int randomIndex = random.nextInt(characters.length());
            randomText.append(characters.charAt(randomIndex));
        }

        stmt3.setString(1, randomText.toString());
        stmt3.setString(2, cid);
        stmt3.setString(3, sid);
        result = stmt3.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return result;
}

public void destroy() {
    try {
        con.close();
        stmt.close();
        stmt1.close();
        stmt2.close();
        stmt3.close();
        stmt4.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

```
	public ArrayList<SubjectAssignment> getAssignments() {

		ArrayList<SubjectAssignment> sas = new ArrayList<SubjectAssignment>();

		try {

			ResultSet rs = stmt4.executeQuery();

			while (rs.next()) {

				SubjectAssignment sa = new SubjectAssignment();

				sa.setCid(rs.getString(2));

				sa.setSid(rs.getString(3));

				sas.add(sa);

				sa.destroy();

			}

		} catch (SQLException e) {

			e.printStackTrace();

		}

		return sas;

	}

}
```

Teacher.java

```
package com.scode.admin.model;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import com.scode.util.ConnectionUtil;

public class Teacher {
	private String id;
	private String name;

	private Connection con;
	private PreparedStatement stmt, stmt1, stmt2,stmt3;

	public Teacher() {
		try {
```

```
            con = ConnectionUtil.getConnection();
            stmt = con.prepareStatement("select * from teachers where
id = ?");
            stmt1 = con.prepareStatement("insert into teachers
values(?,?)");
            stmt2 = con.prepareStatement("select * from teachers");
            stmt3 = con.prepareStatement(
                    "select * from teachers where id in (select
teacher_id from teacherassignment inner JOIN classsubjects  on
teacherassignment.classsubject_id = classsubjects.id where
classsubjects.class_id = ?)");
        } catch (SQLException e) {
            e.printStackTrace();
        }

    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    @Override
    public String toString() {
        return "Teacher [id=" + id + ", name=" + name + "]";
    }

    public int addTeacher() {
        int result = 0;
        try {
            stmt.setString(1, id);
            ResultSet rs = stmt.executeQuery();
            if (rs.next())
                return -1;

            stmt1.setString(1, id);
            stmt1.setString(2, name);
            result = stmt1.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return result;
    }

    public void destroy() {
        try {
            con.close();
            stmt.close();
            stmt1.close();
            stmt2.close();
            stmt3.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public ArrayList<Teacher> getTeachers() {
        ArrayList<Teacher> teachers = new ArrayList<Teacher>();
        try {
            ResultSet rs = stmt2.executeQuery();
            while (rs.next()) {
                Teacher teacher = new Teacher();
                teacher.setId(rs.getString(1));
                teacher.setName(rs.getString(2));
                teachers.add(teacher);
                teacher.destroy();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return teachers;
    }

    public ArrayList<Teacher> getTeachers(String id) {
        ArrayList<Teacher> teachers = new ArrayList<Teacher>();
        try {
            stmt3.setString(1, id);
            ResultSet rs = stmt3.executeQuery();
            while(rs.next()) {
                Teacher teacher = new Teacher();
                teacher.setId(rs.getString(1));
                teacher.setName(rs.getString(2));
                teachers.add(teacher);
                teacher.destroy();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return teachers;
    }
}
```

TeacherAssignment.java

```
package com.scode.admin.model;


import java.sql.Connection;

import java.sql.PreparedStatement;
import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.Random;


import com.scode.util.ConnectionUtil;


public class TeacherAssignment {

    private String tid;

    private String sid;

    private String cid;


    private Connection con;

    private PreparedStatement stmt, stmt1, stmt2, stmt3, stmt4, stmt5, stmt6;


    public TeacherAssignment() {

        try {

            con = ConnectionUtil.getConnection();

            stmt = con.prepareStatement("select * from subjects where id = ?");

            stmt1 = con.prepareStatement("select * from classes where id = ?");

            stmt2 = con.prepareStatement("select * from teachers where id = ?");

            stmt3 = con.prepareStatement("select * from classsubjects where
class_id = ? and subject_id=?");

            stmt4 = con.prepareStatement("select * from teacherassignment where
classsubject_id = ?");

            stmt5 = con.prepareStatement("insert into teacherassignment values
(?,?,?)");

            stmt6 = con.prepareStatement(
                    "select teacher_id,class_id,subject_id from classsubjects
inner join teacherassignment on classsubjects.id = teacherassignment.classsubject_id");

        } catch (SQLException e) {
            e.printStackTrace();
        }

    }

    public String getTid() {
        return tid;
    }

    public void setTid(String tid) {
        this.tid = tid;
    }

    public String getSid() {
        return sid;
    }

    public void setSid(String sid) {
        this.sid = sid;
    }

    public String getCid() {
        return cid;
    }

    public void setCid(String cid) {
        this.cid = cid;
    }

    @Override
    public String toString() {
        return "TeacherAssignment [Teacher " + tid + ", is assigned to subject = " + sid + ",
for calss =" + cid + "]";
    }

    public int assignSubject() {
        int result = 0;
        String id = null;
        try {
            stmt.setString(1, sid);
            ResultSet rs = stmt.executeQuery();
            if (!rs.next())
                return -1;
            stmt1.setString(1, cid);
            rs = stmt1.executeQuery();
            if (!rs.next())
                return -2;
            stmt2.setString(1, tid);
            rs = stmt2.executeQuery();
            if (!rs.next())
```

```
                return -3;
            stmt3.setString(1, cid);
            stmt3.setString(2, sid);
            ResultSet rs1 = stmt3.executeQuery();
            if (!rs1.next())
                return -4;
            if (rs1 != null)
                id = rs1.getString(1);

            stmt4.setString(1, id);
            rs = stmt4.executeQuery();
            if (rs.next())
                return -5;

            Random random = new Random();
            String characters = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
            int length = 6;
            StringBuilder randomText = new StringBuilder();
            for (int i = 0; i < length; i++) {
                int randomIndex = random.nextInt(characters.length());
                randomText.append(characters.charAt(randomIndex));
            }

            stmt5.setString(1, randomText.toString());
            stmt5.setString(2, tid);
            stmt5.setString(3, id);
```

```
        result = stmt5.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return result;
}

public void destroy() {
    try {
        con.close();
        stmt.close();
        stmt1.close();
        stmt2.close();
        stmt3.close();
        stmt4.close();
        stmt5.close();
        stmt6.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public ArrayList<TeacherAssignment> getAssignments() {
    ArrayList<TeacherAssignment> tas = new ArrayList<TeacherAssignment>();
    try {
        ResultSet rs = stmt6.executeQuery();
        while (rs.next()) {
                TeacherAssignment ta = new TeacherAssignment();
                ta.setTid(rs.getString(1));
                ta.setCid(rs.getString(2));
                ta.setSid(rs.getString(3));
                tas.add(ta);
                ta.destroy();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return tas;
    }
}
```

Controllers(Sevlets)

AddClassController.java

```
package com.scode.admin.controllers;


import java.io.IOException;


import com.scode.admin.model.Classes;


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;

/**
 * Servlet implementation class AddFlightController
 */
@WebServlet("/addclass")
public class AddClassController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public AddClassController() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        doPost(request, response);
    }

    /**
```

```
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        HttpSession session = request.getSession();
        if (session.getAttribute("isValid") == null || session.getAttribute("isValid") ==
"false") {
            RequestDispatcher rd = request.getRequestDispatcher("index.jsp");
            rd.forward(request, response);
        } else if (request.getParameter("classid") == null ||
request.getParameter("classname") == null) {
            RequestDispatcher rd = request.getRequestDispatcher("dashboard.jsp");
            rd.forward(request, response);
        } else {
            String id = request.getParameter("classid");
            String name = request.getParameter("classname");
//          System.out.println(id);
            Classes classes = new Classes();
            classes.setId(id);
            classes.setName(name);

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

            if (id.length() > 6) {
                request.setAttribute("message",
                        "Class ID length is greter than 6. Please enter a
value less than or equal to 6");
```

```
            }
            if (name.length() > 10) {
                request.setAttribute("message",
                        "Class name length is greter than 10. Please enter a
value less than or equal to 10");
            }
            int result = classes.addClass();
            if (result == -1) {
                request.setAttribute("message", "Class ID " + id
                        + " already exists in the database. Please enter a
new value");
            }
            if (result == -2) {
                request.setAttribute("message", "Class name " + name
                        + " already exists in the database. Please enter a
new value");
            }
            if (result > 0) {
                request.setAttribute("message", "Class " + classes.toString() + "
created successfully");


            }
            classes.destroy();
            rd.forward(request, response);
//      System.out.println(request.getParameter("date"));
//      System.out.println(flight.toString());
        }
    }
```

```
}
```

AddStudentController.java

```
package com.scode.admin.controllers;

import java.io.IOException;

import com.scode.admin.model.Student;

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;

/**
 * Servlet implementation class AddFlightController
 */
@WebServlet("/addstudent")
public class AddStudentController extends HttpServlet {
	private static final long serialVersionUID = 1L;

	/**
	 * @see HttpServlet#HttpServlet()
	 */
```

```
    public AddStudentController() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
     *     response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        doPost(request, response);
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
     *     response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        HttpSession session = request.getSession();
        if (session.getAttribute("isValid") == null || session.getAttribute("isValid") ==
"false") {
            RequestDispatcher rd = request.getRequestDispatcher("index.jsp");
            rd.forward(request, response);
        } else if (request.getParameter("studentid") == null ||
request.getParameter("studentname") == null || request.getParameter("classid") == null) {
            RequestDispatcher rd = request.getRequestDispatcher("dashboard.jsp");
            rd.forward(request, response);

        } else {

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

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

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


            Student student = new Student();

            student.setId(id);

            student.setName(name);

            student.setClassid(classid);


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


            if (id.length() > 10) {

                request.setAttribute("message",

                        "Student ID length is greter than 6. Please enter a
value less than or equal to 6");

            }

            int result = student.addStudent();

            if (result == -1) {

                request.setAttribute("message", "Student ID " + id

                        + " already exists in the database. Please enter a
new value");

            }

            if (result > 0) {

                request.setAttribute("message", "Student " + student.toString() + "
created successfully");
```

```
            }
            student.destroy();
            rd.forward(request, response);
//      System.out.println(request.getParameter("date"));
//      System.out.println(flight.toString());
        }
    }


}
```

AddSubjectController.java

```
package com.scode.admin.controllers;


import java.io.IOException;


import com.scode.admin.model.Subject;


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;


/**
 * Servlet implementation class AddFlightController
 */
```

```
@WebServlet("/addsubject")
public class AddSubjectController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public AddSubjectController() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
     *     response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        doPost(request, response);
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
     *     response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
```

```
        HttpSession session = request.getSession();
        if (session.getAttribute("isValid") == null || session.getAttribute("isValid") ==
"false") {
            RequestDispatcher rd = request.getRequestDispatcher("index.jsp");
            rd.forward(request, response);
        } else if (request.getParameter("subjectid") == null ||
request.getParameter("subjectname") == null) {
            RequestDispatcher rd = request.getRequestDispatcher("dashboard.jsp");
            rd.forward(request, response);
        } else {
            String id = request.getParameter("subjectid");
            String name = request.getParameter("subjectname");

            Subject subject = new Subject();
            subject.setId(id);
            subject.setName(name);

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

            if (id.length() > 6) {
                request.setAttribute("message",
                        "Subject ID length is greter than 6. Please enter a
value less than or equal to 6");
            }
            if (name.length() > 25) {
                request.setAttribute("message",
                        "Subject ID length is greter than 25. Please enter a
value less than or equal to 25");
```

```
            }
            int result = subject.addSubject();
            if (result == -1) {
                request.setAttribute("message", "Subject ID " + id
                        + " already exists in the database. Please enter a
new value");
            }
            if (result == -2) {
                request.setAttribute("message", "Subject name " + name
                        + " already exists in the database. Please enter a
new value");
            }
            if (result > 0) {
                request.setAttribute("message", "Subject " + subject.toString() + "
created successfully");

            }
            subject.destroy();
            rd.forward(request, response);
//          System.out.println(request.getParameter("date"));
//          System.out.println(flight.toString());
        }
    }

}
```

AddTeacherController.java

```
package com.scode.admin.controllers;
```

```java
import java.io.IOException;

import com.scode.admin.model.Classes;
import com.scode.admin.model.Teacher;

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;

/**
 * Servlet implementation class AddFlightController
 */
@WebServlet("/addteacher")
public class AddTeacherController extends HttpServlet {
	private static final long serialVersionUID = 1L;

	/**
	 * @see HttpServlet#HttpServlet()
	 */
	public AddTeacherController() {
		super();
		// TODO Auto-generated constructor stub
	}
```

```
    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
     *     response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        doPost(request, response);
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
     *     response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        HttpSession session = request.getSession();
        if (session.getAttribute("isValid") == null || session.getAttribute("isValid") ==
"false") {
            RequestDispatcher rd = request.getRequestDispatcher("index.jsp");
            rd.forward(request, response);
        } else if (request.getParameter("teacherid") == null ||
request.getParameter("teachername") == null) {
            RequestDispatcher rd = request.getRequestDispatcher("dashboard.jsp");
            rd.forward(request, response);
        } else {
            String id = request.getParameter("teacherid");
            String name = request.getParameter("teachername");
//                      System.out.println(id);
            Teacher teacher = new Teacher();
            teacher.setId(id);
            teacher.setName(name);

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

            if (id.length() > 6) {
                    request.setAttribute("message",
                                    "Teacher ID length is greter than 6. Please enter a
value less than or equal to 6");
            }
            int result = teacher.addTeacher();
            if (result == -1) {
                    request.setAttribute("message", "Teacher ID " + id
                                    + " already exists in the database. Please enter a
new value");
            }
            if (result > 0) {
                    request.setAttribute("message", "Teacher " + teacher.toString() + "
created successfully");

            }
            teacher.destroy();
            rd.forward(request, response);
//              System.out.println(request.getParameter("date"));
//              System.out.println(flight.toString());
        }
```

```
	}


}
```

AssignSubjectController.java

```
package com.scode.admin.controllers;

import java.io.IOException;

import com.scode.admin.model.Student;
import com.scode.admin.model.Subject;
import com.scode.admin.model.SubjectAssignment;

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;

/**
 * Servlet implementation class AddFlightController
 */
@WebServlet("/assignsubject")
public class AssignSubjectController extends HttpServlet {
	private static final long serialVersionUID = 1L;
```

```
    /**
     * @see HttpServlet#HttpServlet()
     */
    public AssignSubjectController() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        doPost(request, response);
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        HttpSession session = request.getSession();
        if (session.getAttribute("isValid") == null || session.getAttribute("isValid") ==
"false") {
            RequestDispatcher rd = request.getRequestDispatcher("index.jsp");
            rd.forward(request, response);
        } else if (request.getParameter("classid") == null ||
request.getParameter("subjectid") == null) {
//          System.out.println("hello");
            RequestDispatcher rd = request.getRequestDispatcher("dashboard.jsp");
            rd.forward(request, response);
        } else {
            String sid = request.getParameter("subjectid");
            String cid = request.getParameter("classid");

            SubjectAssignment sa = new SubjectAssignment();
            sa.setCid(cid);
            sa.setSid(sid);

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

            if (cid.length() > 6) {
                request.setAttribute("message",
                        "Class ID length is greter than 6. Please enter a
value less than or equal to 6");
            }
            if (sid.length() > 6) {
                request.setAttribute("message",
                        "Subject ID length is greter than 6. Please enter a
value less than or equal to 6");
            }
            int result = sa.assignSubject();
            if (result == -1) {
                request.setAttribute("message", "Subject ID " + sid
                        + " does not exists in the database. Please enter a
valid value");
            }
            if (result == -2) {
                request.setAttribute("message", "Class ID " + cid
                        + " does not exists in the database. Please enter a
valid value");
            }
            if (result == -3) {
                request.setAttribute("message", "Subject ID " + sid
                        + " is already assigned to different class. Please
enter a valid value");
            }
            if (result > 0) {
                request.setAttribute("message", "Assignment " + sa.toString() + "
created successfully");

            }
            sa.destroy();
            rd.forward(request, response);
//        System.out.println(request.getParameter("date"));
//        System.out.println(flight.toString());
        }
    }

}
```

AssignTeacherController.java

```
package com.scode.admin.controllers;

import java.io.IOException;

import com.scode.admin.model.Student;
import com.scode.admin.model.TeacherAssignment;

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;

/**
 * Servlet implementation class AddFlightController
 */
@WebServlet("/assignteacher")
public class AssignTeacherController extends HttpServlet {
        private static final long serialVersionUID = 1L;

        /**
         * @see HttpServlet#HttpServlet()
         */
        public AssignTeacherController() {
                super();
                // TODO Auto-generated constructor stub
        }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
         *     response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response)
                throws ServletException, IOException {
                doPost(request, response);
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
         *     response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response)
                throws ServletException, IOException {
                HttpSession session = request.getSession();
                if (session.getAttribute("isValid") == null || session.getAttribute("isValid") ==
"false") {
                RequestDispatcher rd = request.getRequestDispatcher("index.jsp");
                rd.forward(request, response);
                } else if (request.getParameter("teacherid") == null ||
request.getParameter("subjectid") == null || request.getParameter("classid") == null) {
                RequestDispatcher rd = request.getRequestDispatcher("dashboard.jsp");
                rd.forward(request, response);
                } else {
                String tid = request.getParameter("teacherid");
                String sid = request.getParameter("subjectid");
                String cid = request.getParameter("classid");

                TeacherAssignment ta = new TeacherAssignment();
                ta.setCid(cid);
                ta.setSid(sid);
                ta.setTid(tid);

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

                if (tid.length() > 6) {
                    request.setAttribute("message",
                            "Teacher ID length is greter than 6. Please enter a value less than or equal to 6");
                }
                if (sid.length() > 6) {
                    request.setAttribute("message",
                            "Subject ID length is greter than 6. Please enter a value less than or equal to 6");
                }
                if (cid.length() > 6) {
                    request.setAttribute("message",
                            "Class ID length is greter than 6. Please enter a value less than or equal to 6");
                }
                int result = ta.assignSubject();
                if (result == -1) {
                    request.setAttribute("message", "Subject ID " + sid
                            + " does not exists in the database. Please enter a
    new value");
                }
                if (result == -2) {
                    request.setAttribute("message", "Class ID " + sid
                            + " does not exists in the database. Please enter a
    new value");
                }

                if (result == -3) {
                    request.setAttribute("message", "Teacher ID " + sid
                            + " does not exists in the database. Please enter a
    new value");
                }
                if (result == -4) {
                    request.setAttribute("message", "Subject ID " + sid
                            + " is not assigned to Class ID "+cid);
                }
                if (result == -5) {
                    request.setAttribute("message", "Subject ID " + sid
                            + " is for Class ID "+cid+" is already assigned to a
    teacher");
                }
                if (result > 0) {
                    request.setAttribute("message", "Assignmet " + ta.toString() + "
    created successfully");

                }
                ta.destroy();
                rd.forward(request, response);
    //          System.out.println(request.getParameter("date"));
    //          System.out.println(flight.toString());
            }
        }

    }
```

ChangePasswordController.java

```
package com.scode.admin.controllers;

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 com.scode.admin.model.Admin;

/**
 * Servlet implementation class ChangePasswordController
 */
@WebServlet("/change")
```

```
public class ChangePasswordController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ChangePasswordController() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        doPost(request, response);
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        HttpSession session = request.getSession();
        if (session.getAttribute("isValid") == null || session.getAttribute("isValid") ==
"false") {
            RequestDispatcher rd = request.getRequestDispatcher("index.jsp");
            rd.forward(request, response);
        } else if (request.getParameter("password") == null) {
            RequestDispatcher rd = request.getRequestDispatcher("dashboard.jsp");
            rd.forward(request, response);
        } else {
            String password = request.getParameter("password");
            Admin admin = new Admin();
            int result = admin.updatePassword(password);
            RequestDispatcher rd = request.getRequestDispatcher("result.jsp");
//          System.out.println(result);
            if (result > 0) {
                request.setAttribute("message", "Password updated succesfully");
            } else {
                request.setAttribute("message", "Password not updated. Please
try again later.");
            }
            admin.destroy();
//      System.out.println(request.getAttribute("message"));
            rd.forward(request, response);
        }
    }


}
```

LoginController.java

```
package com.scode.admin.controllers;
```

```
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 com.scode.admin.model.Admin;

/**
 * Servlet implementation class LoginController
 */
@WebServlet("/login")
public class LoginController extends HttpServlet {
	private static final long serialVersionUID = 1L;

	/**
	 * @see HttpServlet#HttpServlet()
	 */
	public LoginController() {
		super();
		// TODO Auto-generated constructor stub
	}
```

```
/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
 *      response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response)
                throws ServletException, IOException {
        doPost(request, response);
}

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
 *      response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response)
                throws ServletException, IOException {
        HttpSession session = request.getSession();

        String username = request.getParameter("username");
        String password = request.getParameter("password");
        RequestDispatcher rd = null;
        if (username == null || password == null) {
                rd = request.getRequestDispatcher("index.jsp");
                rd.forward(request, response);
        } else {

                Admin admin = new Admin();
```

```
			admin.setPassword(password);
			admin.setUsername(username);
			boolean isValid = admin.validate();
			if (isValid) {
				rd = request.getRequestDispatcher("dashboard.jsp");

			} else {
				rd = request.getRequestDispatcher("loginerror.jsp");
			}
			if (session.getAttribute("isValid") == null)
				session.setAttribute("isValid", isValid);
			admin.destroy();
			rd.forward(request, response);
		}
	}
}
```

LogoutController.java

```
package com.scode.admin.controllers;

import java.io.IOException;

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;


/**
 * Servlet implementation class LogoutController
 */
@WebServlet("/logout")
public class LogoutController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public LogoutController() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
     *     response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        HttpSession session = request.getSession();
        RequestDispatcher rd = request.getRequestDispatcher("index.jsp");
        if (session.getAttribute("isValid") == null || session.getAttribute("isValid") ==
"false") {
```

```
                rd.forward(request, response);
            } else {
                session.invalidate();
                rd.forward(request, response);
            }
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
                throws ServletException, IOException {
            // TODO Auto-generated method stub
            doGet(request, response);
    }

}
```

ShowClassController.java

```
package com.scode.admin.controllers;

import java.io.IOException;
import java.util.ArrayList;

import com.scode.admin.model.Classes;
import com.scode.admin.model.Teacher;
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;

/**
 * Servlet implementation class ShowPlacesController
 */
@WebServlet("/showclasses")
public class ShowClassesController extends HttpServlet {
          private static final long serialVersionUID = 1L;

   /**
    * @see HttpServlet#HttpServlet()
    */
   public ShowClassesController() {
      super();
      // TODO Auto-generated constructor stub
   }

          /**
           * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
           */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        HttpSession session = request.getSession();
        if(session.getAttribute("isValid") == null || session.getAttribute("isValid") ==
"false") {
            RequestDispatcher rd = request.getRequestDispatcher("index.jsp");
            rd.forward(request, response);
        }else {
            Classes classes = new Classes();
            ArrayList<Classes> allclasses = classes.getClasses();
            RequestDispatcher rd = request.getRequestDispatcher("showclasses.jsp");
            if(allclasses.isEmpty()) {
                request.setAttribute("message", "No classes found in database");
            }else {
                request.setAttribute("message", "Found below classes in
database");
            }
            request.setAttribute("classes", allclasses);
            classes.destroy();
            rd.forward(request, response);
        }
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }

}
```

ShowClassInfoController.java

```
package com.scode.admin.controllers;

import java.io.IOException;
import java.util.ArrayList;

import com.scode.admin.model.Classes;
import com.scode.admin.model.Student;
import com.scode.admin.model.Subject;
import com.scode.admin.model.SubjectAssignment;
import com.scode.admin.model.Teacher;

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;

/**
 * Servlet implementation class ShowPlacesController
 */
@WebServlet("/showclassinfo")
public class ShowClassInfoController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ShowClassInfoController() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        HttpSession session = request.getSession();
        if (session.getAttribute("isValid") == null || session.getAttribute("isValid") ==
"false") {
            RequestDispatcher rd = request.getRequestDispatcher("index.jsp");
            rd.forward(request, response);
        } else if (request.getParameter("classid") == null) {
            RequestDispatcher rd = request.getRequestDispatcher("dashboard.jsp");
            rd.forward(request, response);
        } else {
            String id = request.getParameter("classid");
            Classes ci = new Classes();
            int result = ci.findClass(id);
            RequestDispatcher rd =
request.getRequestDispatcher("displayclassinfo.jsp");
            if (result == -1) {
                request.setAttribute("message", "No class info found in
database");
                request.setAttribute("isFound", false);
                request.setAttribute("teachers", "");
                ci.destroy();
                rd.forward(request, response);
            } else {
                request.setAttribute("isFound", true);
                Teacher teacher = new Teacher();
                ArrayList<Teacher> teachers = teacher.getTeachers(id);
                if(teachers.isEmpty()) {
                    request.setAttribute("tmessage", "No teachers are
assigned to this class");
                }else {
                    request.setAttribute("tmessage", "Found below teachers
assigned to this class");
                }
                request.setAttribute("teachers", teachers);
                Subject subject = new Subject();
                ArrayList<Subject> subjects = subject.getSubjects(id);
                if(subjects.isEmpty()) {
                    request.setAttribute("smessage", "No subjects are
assigned to this class");
                }else {
                    request.setAttribute("smessage", "Found below subjects
assigned to this class");
                }
                request.setAttribute("subjects", subjects);
                Student stu = new Student();
                ArrayList<Student> students = stu.getStudents(id);
                if(students.isEmpty()) {
                    request.setAttribute("stumessage", "No students are
assigned to this class");
                }else {
                    request.setAttribute("stumessage", "Found below students
assigned to this class");
                }
                request.setAttribute("students", students);
                rd.forward(request, response);
            }
        }
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
                    // TODO Auto-generated method stub
                    doGet(request, response);
          }

}
```

ShowStudentsController.java

```
package com.scode.admin.controllers;

import java.io.IOException;
import java.util.ArrayList;

import com.scode.admin.model.Student;

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;

/**
 * Servlet implementation class ShowPlacesController
 */
@WebServlet("/showstudents")
public class ShowStudentsController extends HttpServlet {
          private static final long serialVersionUID = 1L;
```

```
   /**
    * @see HttpServlet#HttpServlet()
    */
   public ShowStudentsController() {
      super();
      // TODO Auto-generated constructor stub
   }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                HttpSession session = request.getSession();
                if(session.getAttribute("isValid") == null || session.getAttribute("isValid") ==
"false") {
                        RequestDispatcher rd = request.getRequestDispatcher("index.jsp");
                        rd.forward(request, response);
                }else {
                        Student student = new Student();
                        ArrayList<Student> students = student.getStudents();
//                      System.out.println(students);
                        RequestDispatcher rd =
request.getRequestDispatcher("showstudents.jsp");
                        if(students.isEmpty()) {
                                request.setAttribute("message", "No students found in
database");
```

```
            }else {
                request.setAttribute("message", "Found below students in
database");
            }
            request.setAttribute("students", students);
            student.destroy();
            rd.forward(request, response);
        }
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }

}
```

ShowSubjectAssignmentController.java

```
package com.scode.admin.controllers;

import java.io.IOException;
import java.util.ArrayList;

import com.scode.admin.model.SubjectAssignment;
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;

/**
 * Servlet implementation class ShowPlacesController
 */
@WebServlet("/showsubjectassignments")
public class ShowSubjectsAssignmentController extends HttpServlet {
	private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ShowSubjectsAssignmentController() {
        super();
        // TODO Auto-generated constructor stub
    }

	/**
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
	 */
```

```
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        HttpSession session = request.getSession();
        if(session.getAttribute("isValid") == null || session.getAttribute("isValid") ==
"false") {
            RequestDispatcher rd = request.getRequestDispatcher("index.jsp");
            rd.forward(request, response);
        }else {
            SubjectAssignment sa = new SubjectAssignment();
            ArrayList<SubjectAssignment> sas = sa.getAssignments();
            RequestDispatcher rd =
request.getRequestDispatcher("showsubjectassign.jsp");
            if(sas.isEmpty()) {
                request.setAttribute("message", "No assignments found in
database");
            }else {
                request.setAttribute("message", "Found below subjects in
database");
            }
            request.setAttribute("sas", sas);
            sa.destroy();
            rd.forward(request, response);
        }
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
     */
```

```
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
		// TODO Auto-generated method stub
		doGet(request, response);
	}

}
```

ShowSubjectsController.java

```
package com.scode.admin.controllers;

import java.io.IOException;
import java.util.ArrayList;

import com.scode.admin.model.Subject;
import com.scode.admin.model.Teacher;

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;

/**
 * Servlet implementation class ShowPlacesController
 */
```

```
@WebServlet("/showsubjects")
public class ShowSubjectsController extends HttpServlet {
          private static final long serialVersionUID = 1L;


   /**
    * @see HttpServlet#HttpServlet()
    */
   public ShowSubjectsController() {
      super();
      // TODO Auto-generated constructor stub
   }


          /**
           * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
           */
          protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                    HttpSession session = request.getSession();
                    if(session.getAttribute("isValid") == null || session.getAttribute("isValid") ==
"false") {
                              RequestDispatcher rd = request.getRequestDispatcher("index.jsp");
                              rd.forward(request, response);
                    }else {
                              Subject subject = new Subject();
                              ArrayList<Subject> subjects = subject.getSubjects();
                              RequestDispatcher rd =
request.getRequestDispatcher("showsubjects.jsp");
                              if(subjects.isEmpty()) {
```

```
                request.setAttribute("message", "No subjects found in database");
            }else {
                request.setAttribute("message", "Found below subjects in database");
            }
            request.setAttribute("subjects", subjects);
            subject.destroy();
            rd.forward(request, response);
        }
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }

}
```

ShowTeachersAssignmentController.java

```
package com.scode.admin.controllers;

import java.io.IOException;
import java.util.ArrayList;
import com.scode.admin.model.Teacher;
import com.scode.admin.model.TeacherAssignment;

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;

/**
 * Servlet implementation class ShowPlacesController
 */
@WebServlet("/showteacherassignments")
public class ShowTeachersAssignmentController extends HttpServlet {
	private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ShowTeachersAssignmentController() {
        super();
        // TODO Auto-generated constructor stub
    }

	/**
```

```
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        HttpSession session = request.getSession();
        if(session.getAttribute("isValid") == null || session.getAttribute("isValid") ==
"false") {
            RequestDispatcher rd = request.getRequestDispatcher("index.jsp");
            rd.forward(request, response);
        }else {
            TeacherAssignment ta = new TeacherAssignment();
            ArrayList<TeacherAssignment> tas = ta.getAssignments();
            RequestDispatcher rd =
request.getRequestDispatcher("showteacherassign.jsp");
            if(tas.isEmpty()) {
                request.setAttribute("message", "No assignments found in
database");
            }else {
                request.setAttribute("message", "Found below teachers in
database");
            }
            request.setAttribute("tas", tas);
            ta.destroy();
            rd.forward(request, response);
        }
    }

    /**
```

```
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
		// TODO Auto-generated method stub
		doGet(request, response);
	}

}
```

ShowTeachersController.java

```
package com.scode.admin.controllers;

import java.io.IOException;
import java.util.ArrayList;

import com.scode.admin.model.Teacher;

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;

/**
 * Servlet implementation class ShowPlacesController
 */
@WebServlet("/showteachers")
public class ShowTeachersController extends HttpServlet {
        private static final long serialVersionUID = 1L;

   /**
    * @see HttpServlet#HttpServlet()
    */
   public ShowTeachersController() {
      super();
      // TODO Auto-generated constructor stub
   }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
                HttpSession session = request.getSession();
                if(session.getAttribute("isValid") == null || session.getAttribute("isValid") == "false") {
                        RequestDispatcher rd = request.getRequestDispatcher("index.jsp");
                        rd.forward(request, response);
                }else {
                        Teacher teacher = new Teacher();
                        ArrayList<Teacher> teachers = teacher.getTeachers();
                        RequestDispatcher rd = request.getRequestDispatcher("showteachers.jsp");
```

```
            if(teachers.isEmpty()) {
                request.setAttribute("message", "No teachers found in
database");
            }else {
                request.setAttribute("message", "Found below teachers in
database");
            }
            request.setAttribute("teachers", teachers);
            teacher.destroy();
            rd.forward(request, response);
        }
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }

}
```