

Source Code of Learner's Academy

Login.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Learners Academy </title>

</head>

<!-- Font Awesome -->

<link

href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css"

rel="stylesheet"

/>

<!-- Google Fonts -->

<link

href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700&display=swap"

rel="stylesheet"

/>

<!-- MDB -->

<link
```

```

href="https://cdnjs.cloudflare.com/ajax/libs/mdb-ui-kit/6.2.0/mdb.min.css"
rel="stylesheet"

/>

<body>

<!-- Navbar -->

<nav class="navbar navbar-expand-lg bg-info navbar-dark fixed-top">

<!-- Container wrapper -->

<div class="container-fluid">

<!-- Navbar brand -->

<a class="navbar-brand" href="#"><strong>Learners Academy <u>Administrator</u>
Access</strong></a>

<!-- Toggle button -->

<button class="navbar-toggler" type="button" data-mdb-toggle="collapse" data-mdb-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">

<i class="fas fa-bars"></i>

</button>

<!-- Collapsible wrapper -->

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav me-auto mb-2 mb-lg-0">

</ul>

</div>

</div>

<!-- Container wrapper -->

</nav>

<!-- Navbar -->

<section class="vh-100" style="position: relative;

```

```

top: 10pc;height: 75vh!important;">

<div class="container-fluid h-custom">

<div class="row d-flex justify-content-center align-items-center h-100">

<div class="col-md-9 col-lg-6 col-xl-5">



</div>

<div class="col-md-8 col-lg-6 col-xl-4 offset-xl-1">

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

<div class="d-flex flex-row align-items-center justify-content-center justify-
content-lg-start">

<p class="lead fw-normal mb-0 me-3">Sign in </p>

</div>

<!-- Email input -->

<div class="form-outline mb-4">

<input type="text" id="form3Example3" class="form-control form-control-lg"
name="username"

placeholder="Enter a valid Username" required/>

<label class="form-label" for="form3Example3">Username</label>

</div>

<!-- Password input -->

<div class="form-outline mb-3">

<input type="password" id="form3Example4" class="form-control form-control-lg"
name="password"

placeholder="Enter password" required/>

<label class="form-label" for="form3Example4">Password</label>

```

```
</div>

<div class="text-center text-lg-start mt-4 pt-2">

<button type="submit" class="btn btn-primary btn-lg"
style="padding-left: 2.5rem; padding-right: 2.5rem;">Login</button>

</div>

</form>

</div>

</div>

</div>

</div>

</section>

<!-- MDB -->

<script
type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/mdb-ui-kit/6.2.0/mdb.min.js"
></script>

</body>

</html>
```

Sidebar.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>

<!doctype html>

<html lang="en" data-bs-theme="auto">

<head><script src="../assets/js/color-modes.js"></script>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta name="description" content="">

<meta name="author" content="Mark Otto, Jacob Thornton, and Bootstrap contributors">

<meta name="generator" content="Hugo 0.111.3">

<title>Sidebars · Bootstrap v5.3</title>

<link rel="canonical" href="https://getbootstrap.com/docs/5.3/examples/sidebars/">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-aFq/bzH65dt+w6FI2ooMVUpc+21e0SRygnTpmBvdBgSdnuTN7QbdlgL+OapgHtvPp"
crossorigin="anonymous">

<style>

.bd-placeholder-img {

font-size: 1.125rem;

text-anchor: middle;

-webkit-user-select: none;

-moz-user-select: none;

user-select: none;

}

@media (min-width: 768px) {

.bd-placeholder-img-lg {
```

```
font-size: 3.5rem;

}

}

.b-example-divider {

width: 100%;

height: 3rem;

background-color: rgba(0, 0, 0, .1);

border: solid rgba(0, 0, 0, .15);

border-width: 1px 0;

box-shadow: inset 0 .5em 1.5em rgba(0, 0, 0, .1), inset 0 .125em .5em rgba(0, 0, 0, .15);

}

.b-example-vr {

flex-shrink: 0;

width: 1.5rem;

height: 100vh;

}

.bi {

vertical-align: -.125em;

fill: currentColor;

}

.nav-scroller {

position: relative;

z-index: 2;

height: 2.75rem;

overflow-y: hidden;
```

```
}

.nav-scroller .nav {

display: flex;

flex-wrap: nowrap;

padding-bottom: 1rem;

margin-top: -1px;

overflow-x: auto;

text-align: center;

white-space: nowrap;

-webkit-overflow-scrolling: touch;

}

.btn-bd-primary {

--bd-violet-bg: #712cf9;

--bd-violet-rgb: 112.520718, 44.062154, 249.437846;

--bs-btn-font-weight: 600;

--bs-btn-color: var(--bs-white);

--bs-btn-bg: var(--bd-violet-bg);

--bs-btn-border-color: var(--bd-violet-bg);

--bs-btn-hover-color: var(--bs-white);

--bs-btn-hover-bg: #6528e0;

--bs-btn-hover-border-color: #6528e0;

--bs-btn-focus-shadow-rgb: var(--bd-violet-rgb);

--bs-btn-active-color: var(--bs-btn-hover-color);

--bs-btn-active-bg: #5a23c8;

--bs-btn-active-border-color: #5a23c8;
```

```

}

.bd-mode-toggle {
  z-index: 1500;
}

li{
  margin-bottom:5px;}

</style>

<!-- Custom styles for this template -->

<link href="sidebars.css" rel="stylesheet">

</head>

<body>

<main class="d-flex flex-nowrap" style=" position: absolute;
left: -23px;
width: 24%;
border: 2px solid;">

<h1 class="visually-hidden"></h1>

<div class="b-example-divider b-example-vr"></div>

<div class="d-flex flex-column flex-shrink-0 p-3 bg-body-tertiary" style="width:
280px;">

<a href="/" class="d-flex align-items-center mb-3 mb-md-0 me-md-auto link-body-
emphasis text-decoration-none">

<svg class="bi pe-none me-2" width="40" height="32"><use
xlink:href="#bootstrap"/></svg>

<span class="fs-4 "><strong>Learner's Academy </strong></span>

</a>

<hr>

<ul class="nav nav-pills flex-column mb-auto">

```



```
<li class="nav-item">
```

```
<a href="Subjects" class="nav-link active" aria-current="page">
```

```
<svg class="bi pe-none me-2" width="16" height="16"><use xlink:href="#home"/></svg>
```

Subjects

```
</a>
```

```
</li>
```

```
<li>
```

```
<a href="Teachers" class="nav-link active link-body-emphasis">
```

```
<svg class="bi pe-none me-2" width="16" height="16"><use  
xlink:href="#speedometer2"/></svg>
```

Teachers

```
</a>
```

```
</li>
```

```
<li>
```

```
<a href="Classes" class="nav-link active link-body-emphasis">
```

```
<svg class="bi pe-none me-2" width="16" height="16"><use xlink:href="#table"/></svg>
```

Classes

```
</a>
```

```
</li>
```

```
<li>
```

```
<a href="Classes-subjects" class="nav-link active link-body-emphasis">
```

```
<svg class="bi pe-none me-2" width="16" height="16"><use xlink:href="#grid"/></svg>
```

Classes-subjects

```
</a>
```

```
</li>
```

```
<li>
```

```
<a href="Teachers-subject" class="nav-link active link-body-emphasis">
```

```
<svg class="bi pe-none me-2" width="16" height="16"><use xlink:href="#people-circle"/></svg>
```

Teachers-subject

```
</a>
```

```
</li>
```

```
<li>
```

```
<a href="Students" class="nav-link active link-body-emphasis">
```

```
<svg class="bi pe-none me-2" width="16" height="16"><use xlink:href="#people-circle"/></svg>
```

Students

```
</a>
```

```
</li>
```

```
</ul>
```

```
<hr>
```

```
<div class="dropdown">
```

```
<strong> <a href="LogoutServlet" >
```

```
log out</a></strong>
```

```
</div>
```

```
</div>
```

```
</main>
```

```
<!-- As a link -->
```

```
<!-- As a heading -->
```

```
<nav class="navbar navbar-dark bg-primary text-center" style="position: relative;
```

```
left: 293px;
```

```
top: 8px;
```

```
width: 77%;
```

```
border: 2px crimson;
```

```
border-radius: 3px">
```

```
<span class="navbar-brand mb-0 h1 " style=" margin: 0 auto;">${Name} Master  
List</span>
```

```
</nav>
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-  
alpha2/dist/js/bootstrap.bundle.min.js" integrity="sha384-  
qKXV1j0HvMUeCBQ+QVp7JcfGL760yU08IQ+GpUo5hLbpg51QRiuqHAJz8+BrxE/N"  
crossorigin="anonymous"></script>
```

```
<script src="sidebars.js"></script>
```

```
</body>
```

```
</html>
```

Students-List.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>

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

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<jsp:include page="Sidebar.jsp" />

<table class="table table-striped" style="position: relative;
left: 19pc;
width: 75%;
top: 29px;">

<thead class="bg-primary">

<tr>

<th scope="col">Id</th>

<th scope="col">First Name</th>

<th scope="col">Last Name</th>

<th scope="col">Age</th>

<th scope="col">Class</th>

</tr>

</thead>
```

```
<tbody>

<c:forEach var="lis" items="${list}">

<tr>

<th scope="row">${lis.id}</th>

<td>${lis.firstname}</td>

<td>${lis.lastname}</td>

<td>${lis.age}</td>

<td>${lis.aclass}</td>

</tr>

</c:forEach>

</tbody>

</table>

</body>

</html>
```

Subjects-List.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>

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

<!DOCTYPE html>

<html>
```

```
<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<jsp:include page="Sidebar.jsp" />

<table class="table table-striped" style="position: relative;
left: 19pc;
width: 75%;
top: 29px;">

<thead class="bg-primary">

<tr>

<th scope="col">Id</th>

<th scope="col">Subject Name</th>

</tr>

</thead>

<tbody>

<c:forEach var="lis" items="${list}">

<tr>

<th scope="row">${lis.id}</th>

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

</tr>

</c:forEach>

</tbody>

</table></body></html>
```

Teachers-list.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>

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

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<jsp:include page="Sidebar.jsp" />

<table class="table table-striped" style="position: relative;
left: 19pc;
width: 75%;
top: 29px;">

<thead class="bg-primary">

<tr>

<th scope="col">Id</th>

<th scope="col">First Name</th>

<th scope="col">Last Name</th>

<th scope="col">Age</th>

</tr>

</thead>

<tbody>
```

```
<c:forEach var="lis" items="${list}">

<tr>

<th scope="row">${lis.id}</th>

<td>${lis.firstname}</td>

<td>${lis.lastname}</td>

<td>${lis.age}</td>

</tr>

</c:forEach>

</tbody>

</table>

</body>

</html>
```


Class-List.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>

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

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<jsp:include page="Sidebar.jsp" />

<table class="table table-striped" style="position: relative;
left: 19pc;
width: 75%;
top: 29px;">

<thead class="bg-primary">

<tr>

<th scope="col">Class Id</th>

<th scope="col">Section</th>

<th scope="col">Subject</th>

<th scope="col">Teacher</th>

<th scope="col">time</th>

<th scope="col">Students</th>

</tr>
```

```
</thead>

<tbody>

<c:forEach var="tempClass" items="${CLASSES_LIST}">

<tr>

<c:url var="tempLink" value="/">

<c:param name="command" value="/ST_LIST" />

<c:param name="classId" value="${tempClass.id}" />

<c:param name="section" value="${tempClass.section}" />

<c:param name="subject" value="${tempClass.subject}" />

</c:url>

<td>${tempClass.id}</td>

<td>${tempClass.section}</td>

<td>${tempClass.subject}</td>

<td>${tempClass.teacher}</td>

<td>${tempClass.time}</td>

<td><a href="${tempLink}">List</a></td>

</tr>

</c:forEach>

</tbody>

</table>

</body>

</html>
```

Class-Students.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>

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

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<jsp:include page="Sidebar.jsp" />

<table class="table table-striped" style="position: relative;
left: 19pc;
width: 75%;
top: 29px;">

<thead class="bg-primary">

<tr>

<th>First Name</th>

<th>Last Name</th>

<th>age</th>

</tr>

</thead>

<tbody>

<c:forEach var="tempStudent" items="${STUDENTS_LIST}">
```

```
<tr>

<td>${tempStudent.firstname}</td>

<td>${tempStudent.lastname}</td>

<td>${tempStudent.age}</td>

</tr>

</c:forEach>

</tbody>

</table>

</body>

</html>
```

Assign_Teacher_Class_For_Subjects.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>

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

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<jsp:include page="Sidebar.jsp" />

<form style="position: relative;
width: 18%;
top: 10pc;
margin: 0 auto;
left: 11pc;" action="Assign_classSubTeacher" method="post">

<strong><h2 style="font-size: .9rem;">Assign Teacher to a Class for a
subject</h2></strong>

<div class="form-group">

<label for="exampleInputEmail1">Select a Class</label>

<select class="form-control w-75" name="class">

<c:forEach var="ca" items="${Classes}">

<option class="option" value="${ca.id}">${ca.id}</option>

</c:forEach>

</select>
```

```
</div>

<div class="form-group">

<label for="exampleInputPassword1">Select a Subject</label>

<select class="form-control w-75" name="subject">

<c:forEach var="sub" items="${Subjects}">

<option class="option" value="${sub.id}">${sub.name}</option>

</c:forEach>

</select>

</div>

<div class="form-group">

<label for="exampleInputPassword1">Select a Teacher</label>

<select class="form-control w-75" name="teacher">

<c:forEach var="teah" items="${Teachers}">

<option class="option" value="${teah.id}">${teah.firstname} ${teah.lastname}</option>

</c:forEach>

</select>

</div>

<br>

${msg}

<br>

<button type="submit" class="btn btn-primary">Assign</button>

</form>

</table>

</body>

</html>
```

Assign_Classes-Subjects.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>

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

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<jsp:include page="Sidebar.jsp" />

<form style="position: relative;
width: 18%;
top: 10pc;
margin: 0 auto;
left: 11pc;" action="Assign_classSub" method="post">

<strong>

<h2 style="font-size: 1.1rem;">Assign Classes For Subjects</h2></strong>

<div class="form-group">

<label for="exampleInputEmail1">Select a Class</label>

<select class="form-control w-75" name="class">

<c:forEach var="ca" items="${Classes}">

<option class="option" value="${ca.id}">${ca.id}</option>
```

```
</c:forEach>

</select>

</div>

<div class="form-group">

<label for="exampleInputPassword1">Select a Subject</label>

<select class="form-control w-75" name="subject">

<c:forEach var="sub" items="${Subjects}">

<option class="option" value="${sub.id}">${sub.name}</option>

</c:forEach>

</select>

</div>

<br> ${msg}<br>

<button type="submit" class="btn btn-primary">Assign</button>

</form>

</table>

</body>

</html>
```


COM.MODELS

Class.java;

```
package com.models;

public class Class {

    private int id;

    private int section;

    private String teacher;

    private String subject;

    private String time;

    public Class(int id, int section, String teacher, String subject, String time) {

        super();

        this.id = id;

        this.section = section;

        this.teacher = teacher;

        this.subject = subject;

        this.time = time;

    }

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

    public int getSection() {

        return section;

    }

}
```

```
}

public void setSection(int section) {
    this.section = section;
}

public String getTeacher() {
    return teacher;
}

public void setTeacher(String teacher) {
    this.teacher = teacher;
}

public String getSubject() {
    return subject;
}

public void setSubject(String subject) {
    this.subject = subject;
}

public String getTime() {
    return time;
}

public void setTime(String time) {
    this.time = time;
}
}
```

Student.java

```
package com.models;

public class Student {

    private int id;

    private String firstname;

    private String lastname;

    private int age;

    private int aclass;

    public Student(int id, String firstname, String lastname, int age, int aclass) {

        super();

        this.id = id;

        this.firstname = firstname;

        this.lastname = lastname;

        this.age = age;

        this.aclass = aclass;

    }

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

    public String getfirstname() {

        return firstname;

    }

}
```

```
public void setfirstname(String firstname) {  
    this.firstname = firstname;  
}  
  
public String getlastname() {  
    return lastname;  
}  
  
public void setlastname(String lastname) {  
    this.lastname = lastname;  
}  
  
public int getAge() {  
    return age;  
}  
  
public void setAge(int age) {  
    this.age = age;  
}  
  
public int getAclass() {  
    return aclass;  
}  
  
public void setAclass(int aclass) {  
    this.aclass = aclass;  
}  
  
@Override  
public String toString() {  
    return "Student [id=" + id + ", firstname=" + firstname + ", lastname=" + lastname +  
        ", age=" + age + ", aclass=" + aclass  
        + "];}}}
```

Subject.java

```
package com.models;

public class Subject {

    private int id;

    private String name;

    public Subject(int id, String name ) {

        super();

        this.id = id;

        this.name = name;

    }

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

    public String getName() {

        return name;

    }

    public void setName(String name) {

        this.name = name;

    }

}
```

Teacher.jsp

```
package com.models;

public class Teacher {

    private int id;

    private String firstname;

    private String lastname;

    private int age;

    public Teacher(int id, String firstname, String lastname, int age) {

        super();

        this.id = id;

        this.firstname = firstname;

        this.lastname = lastname;

        this.age = age;

    }

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

    public String getfirstname() {

        return firstname;

    }

    public void setfirstname(String firstname) {

        this.firstname = firstname;

    }

}
```

```
}

public String getlastname() {

return lastname;

}

public void setlastname(String lastname) {

this.lastname = lastname;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

}
```

COM.CONTROLLER

ServletController.java

package com.controller;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

import javax.annotation.Resource;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import javax.sql.DataSource;

import com.models.Student;

import com.models.Class;

import com.models.Teacher;

import com.utils.DB_Query;

import com.models.Subject;


```

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

    Connection con;
    Statement stmt;

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

    @Resource(name="jdbc/Learners_Academy")
    DataSource datasource;

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        // TODO Auto-generated method stub
        String action=request.getServletPath();
//        PrintWriter out=response.getWriter();
//        out.println(action);
        System.out.println(action);
    }
}

```

```
switch(action) {
case "/LoginServlet":
    login(request,response);
    break;
case "/LogoutServlet":
    logout(request,response);
    break;
case "/Subjects":
    Subjects(request,response);
    break;
case "/Teachers":
    Teachers(request,response);
    break;
case "/Classes":
    Classes(request, response);
    break;
case "/Classes-subjects":
    Assign_Classes_and_subjects(request,response);
    break;
case "/Teachers-subject":
    Assign_Classes_and_Teacher_for_subject(request,response);
    break;
case "/Students":
    Studnets(request,response);;
    break;
case "/Assign_classSubTeacher":
    Assign_classSubTeacher(request,response);
    break;
```

```

        case "/Assign_classSub":
            Assign_classSub(request,response);
            break;
        case "/":
            System.out.println("a");
            classStudentsList(request, response);
            break;
        default:
            break;
    }
}

```

private void login(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

```

    // TODO Auto-generated method stub

    String uname=request.getParameter("username");
    String pass=request.getParameter("password");
    if(uname.equals("admin") && pass.equals("admin")) {

        HttpSession session=request.getSession();
        session.setAttribute("username", uname);
        session.setAttribute("password", pass);
        RequestDispatcher dispatcher =
request.getRequestDispatcher("/Students");
        dispatcher.forward(request, response);
    }
    else {
        System.out.println("<h1>Invalid Login</h1>");
    }
}

```

```

        }

//

    }

    private void logout(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

        // TODO Auto-generated method stub
        request.getSession().invalidate();
        RequestDispatcher dispatcher = request.getRequestDispatcher("/Login.jsp");
        dispatcher.forward(request, response);
    }

    private void Subjects(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

        // TODO Auto-generated method stub
        try {

            DB_Querys Utils=new DB_Querys(datasource);
            request.setAttribute("list", Utils.Subjects());
            request.setAttribute("Name", "Subjects");
            RequestDispatcher dispatcher =
request.getRequestDispatcher("Subjects-List.jsp");
            dispatcher.forward(request, response);

        }
        catch(NumberFormatException ignored) {

        }
    }

```

```

    }

    private void Teachers(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

        // TODO Auto-generated method stub

        try {

            DB_Querys Utils=new DB_Querys(datasource);

            request.setAttribute("list", Utils.Teachers());

            request.setAttribute("Name", "Teachers");

            RequestDispatcher dispatcher = request.getRequestDispatcher("Teachers-
List.jsp");

            dispatcher.forward(request, response);

//

        }catch(NumberFormatException ignored) {

        }

    }

    private void Studnets(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

        // TODO Auto-generated method stub

        try {

            DB_Querys Utils=new DB_Querys(datasource);

            request.setAttribute("list", Utils.Students());

            request.setAttribute("Name", "Students");

            RequestDispatcher dispatcher =
request.getRequestDispatcher("Students-List.jsp");

            dispatcher.forward(request, response);

```

```

//
        }catch(NumberFormatException ignored) {
        }
    }

    private void Classes(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        // TODO Auto-generated method stub
        try {

            DB_Querys Utils=new DB_Querys(datasource);
            request.setAttribute("CLASSES_LIST",
Utils.Classes());

            request.setAttribute("Name", "Classes");

            RequestDispatcher dispatcher =
request.getRequestDispatcher("Classes-List.jsp");
            dispatcher.forward(request, response);

//
        }
        catch(NumberFormatException ignored) {
        }
    }

    private void Assign_classSub(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        int classId = Integer.parseInt(request.getParameter("class"));

```

```

        int subject = Integer.parseInt(request.getParameter("subject"));

        // get subjects from db util
        // System.out.println(classsubject);
        DB_Querys Utils=new DB_Querys(datasource);
        int result=Utils.UpdateClassTable(classId, subject);
        if(result==1) {

            request.setAttribute("msg", "Assigned Succesfully");

            RequestDispatcher dispatcher =
request.getRequestDispatcher("/Assign_Classes-Subjects.jsp");
            dispatcher.forward(request, response);
        }
    }

    private void Assign_classSubTeacher(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        int classId = Integer.parseInt(request.getParameter("class"));
        int teacher = Integer.parseInt(request.getParameter("teacher"));
        int subject = Integer.parseInt(request.getParameter("subject"));

        // get subjects from db util
        System.out.println(classId+teacher+subject);
        DB_Querys Utils=new DB_Querys(datasource);
        int result=Utils.UpdateClassTable(classId, teacher, subject);
        if(result==1) {

            request.setAttribute("msg", "Assigned Succesfully");

```

```

        RequestDispatcher dispatcher =
request.getRequestDispatcher("/Assign_Teacher_Class_For_Subjects.jsp");
        dispatcher.forward(request, response);
    }
}

```

```

private void classStudentsList(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    int classId = Integer.parseInt(request.getParameter("classId"));
    String section = request.getParameter("section");
    String subject = request.getParameter("subject");

    // get subjects from db util
    System.out.println("A");
    DB_Querys Utils=new DB_Querys(datasource);
    // add subjects to the request
    List<Student> stu=new ArrayList<>();
    stu=Utils.loadClassStudents(classId);
    System.out.println(stu.size());
    request.setAttribute("STUDENTS_LIST",stu );
    request.setAttribute("Name", "Students of "+ subject+" class of "+classId+"
section of "+section);

    // send it to the jSP view page
    RequestDispatcher dispatcher = request.getRequestDispatcher("/class-
students.jsp");
    dispatcher.forward(request, response);
}

```



```
}
```

```
private void Assign_Classes_and_subjects(HttpServletRequest request,  
HttpServletResponse response) throws ServletException, IOException {
```

```
    // TODO Auto-generated method stub
```

```
    try {
```

```
        DB_Querys Utils=new DB_Querys(datasource);
```

```
        request.setAttribute("Subjects", Utils.Subjects());
```

```
        request.setAttribute("Classes", Utils.Classes());
```

```
        RequestDispatcher dispatcher =  
request.getRequestDispatcher("/Assign_Classes-Subjects.jsp");
```

```
        dispatcher.forward(request, response);
```

```
//
```

```
    } catch(NumberFormatException ignored) {
```

```
    }
```

```
}
```

```
private void Assign_Classes_and_Teacher_for_subject(HttpServletRequest request,  
HttpServletResponse response) throws ServletException, IOException {
```

```
    // TODO Auto-generated method stub
```

```
    try {
```

```
        DB_Querys Utils=new DB_Querys(datasource);
```

```
        request.setAttribute("Subjects", Utils.Subjects());
```

```

        request.setAttribute("Classes", Utils.Classes());

        request.setAttribute("Teachers", Utils.Teachers());

        RequestDispatcher dispatcher =
request.getRequestDispatcher("/Assign_Teacher_Class_For_Subjects.jsp");

        dispatcher.forward(request, response);

//

        } catch(NumberFormatException ignored) {
        }

    }

/**
 * @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);

    }

}

```

```
DB_QUERY.java

package com.utils;

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

import javax.sql.DataSource;

import com.models.Student;
import com.models.Subject;
import com.models.Teacher;
import com.models.Class;

public class DB_Querys {

    private DataSource datasource;

    Connection con = null;
    Statement stmt = null;

    public DB_Querys(DataSource datasource) {
        this.datasource = datasource;
    }
}
```

```
}
```

```
public List<Subject> Subjects() {  
    List<Subject> temp_list = new ArrayList<>();  
    try {  
        // get connection from connection pool  
        con = this.datasource.getConnection();  
        stmt = con.createStatement();  
        ResultSet resultSet = stmt.executeQuery("select * from subjects");  
  
        while (resultSet.next()) {  
            int id = resultSet.getInt("id");  
            String name = resultSet.getString("name");  
            Subject Subjects= new Subject(id, name);  
            temp_list.add(Subjects);  
            System.out.println(id);  
        }  
  
    } catch (SQLException e) {  
        e.printStackTrace();  
    } finally {  
        close(con,stmt);  
    }  
    return temp_list;  
}  
  
public List<Teacher> Teachers() {
```

```

List<Teacher> temp_list = new ArrayList<>();
try {
    // get connection from connection pool

    con = this.datasource.getConnection();
    stmt = con.createStatement();
    ResultSet resultSet = stmt.executeQuery("select * from teachers");

    while (resultSet.next()) {
        int id = resultSet.getInt("id");
        String fname = resultSet.getString("firstname");
        String lname = resultSet.getString("lastname");
        int age = resultSet.getInt("age");
        Teacher teah = new Teacher(id, fname, lname, age);
        temp_list.add(teah);
        System.out.println(id);

    }
} catch (SQLException e) {
    e.printStackTrace();
} finally {
    close(con, stmt);
}
return temp_list;
}

```

```

public List<Class> Classes() {
    List<Class> classes = new ArrayList<>();
    try {
        // get connection from connection pool
        con = this.datasource.getConnection();
        stmt = con.createStatement();
        ResultSet myRs = stmt.executeQuery("select * from classes");

        while (myRs.next()) {
            int id = myRs.getInt("id");
            int section = myRs.getInt("section");
            int subject = myRs.getInt("subject");
            int teacher = myRs.getInt("teacher");
            String time = myRs.getString("time");

            Teacher tempTeacher = loadTeacher(teacher);
            Subject tempSubject = loadSubject(subject);

            String teacher_name = tempTeacher.getfirstname() + " " +
tempTeacher.getlastname();

            // create new student object
            Class temp = new Class(id, section, teacher_name,
tempSubject.getName(), time);

            // add it to the list of students
            classes.add(temp);
        }
    }
}

```

```

    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        close(con,stmt);
    }
    return classes;
}

public List<Student> Students() {
    List<Student> temp_list = new ArrayList<>();
    try {
        // get connection from connection pool
        con = this.datasource.getConnection();
        stmt = con.createStatement();
        ResultSet resultSet = stmt.executeQuery("select * from students");

        while (resultSet.next()) {
            int id = resultSet.getInt("id");
            String fname = resultSet.getString("firstname");
            String lname = resultSet.getString("lastname");
            int age = resultSet.getInt("age");
            int aclass = resultSet.getInt("class");
            System.out.println(id);
            Student stu = new Student(id, fname, lname, age, aclass);
            temp_list.add(stu);
        }
    }
}

```

```

    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        close(con, stmt);
    }
    return temp_list;
}

```

```

private void close(Connection con, Statement stmt) {

```

```

    try {

        if (stmt != null)
            stmt.close();

        if (con != null)
            con.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

```

public List<Student> loadClassStudents(int classId) {

```

```

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

```



```

Connection myConn = null;
Statement myStmt = null;
System.out.println("IN");
try {

    // get a connection
    myConn = datasource.getConnection();

    // create sql stmt
    String sql = "select * from students WHERE class =" + classId;
    myStmt = myConn.createStatement();

    // execute query
    ResultSet res = myStmt.executeQuery(sql);

    // process result
    int size=0;
    while (res.next()) {
        size++;
        System.out.println(size);

//
        if (res.next()) {

            // retrieve data from result set row
            int id = res.getInt("id");
            String firstName = res.getString("firstname");
            String lastName = res.getString("lastname");
            int age = res.getInt("age");

```

```

        int aclass = res.getInt("class");
        // create new student object
        Student tempStudent = new Student(id, firstName, lastName, age,
aclass);

        students.add(tempStudent);

        System.out.println("B "+tempStudent.getfirstname());

//
//
//          }else {
//
//          size=1000;
//
//          System.out.println("ASAS"+size);
//
//          }

    } catch (Exception e) {
        // TODO: handle exception
    } finally {
        // close JDBC objects
        close(myConn, myStmt);
    }

    System.out.println(students.size());
    return students;

}

```

```

public Teacher loadTeacher(int teacherId) {

```

```

    Teacher theTeacher = null;

```

```

    Connection myConn = null;

```

```

    Statement myStmt = null;

```

```
ResultSet myRs = null;
```

```
try {
```

```
    // get a connection
```

```
    myConn = datasource.getConnection();
```

```
    // create sql stmt
```

```
    String sql = "SELECT * FROM teachers WHERE id = " + teacherId;
```

```
    myStmt = myConn.createStatement();
```

```
    // execute query
```

```
    myRs = myStmt.executeQuery(sql);
```

```
    // process result
```

```
    while (myRs.next()) {
```

```
        // retrieve data from result set row
```

```
        int id = myRs.getInt("id");
```

```
        String fname = myRs.getString("firstname");
```

```
        String lname = myRs.getString("lastname");
```

```
        int age = myRs.getInt("age");
```

```
        theTeacher = new Teacher(id, fname, lname, age);
```

```
    }
```

```
} catch (Exception e) {
```

```
    // TODO: handle exception
```

```

    } finally {
        // close JDBC objects
        close(myConn, myStmt);
    }
    return theTeacher;
}

public Subject loadSubject(int subjectId) {

    Subject theSubject = null;

    Connection myConn = null;
    Statement myStmt = null;
    ResultSet myRs = null;

    try {

        // get a connection
        myConn = datasource.getConnection();

        // create sql stmt
        String sql = "SELECT * FROM subjects WHERE id = " + subjectId;
        myStmt = myConn.createStatement();

        // execute query
        myRs = myStmt.executeQuery(sql);

```

```

        // process result
        while (myRs.next()) {

            // retrieve data from result set row
            int id = myRs.getInt("id");
            String name = myRs.getString("name");

            theSubject = new Subject(id, name);

        }

    } catch (Exception e) {
        // TODO: handle exception
    } finally {
        // close JDBC objects
        close(myConn, myStmt);
    }
    return theSubject;
}

```

```

public Class loadClass(int classId) {

```

```

    Class theClass = null;

```

```

    Connection myConn = null;

```

```

    Statement myStmt = null;

```

```

    ResultSet myRs = null;

```

```

try {

    // get a connection
    myConn = datasource.getConnection();

    // create sql stmt
    String sql = "SELECT * FROM classes WHERE id = " + classId;
    myStmt = myConn.createStatement();

    // execute query
    myRs = myStmt.executeQuery(sql);

    // process result
    while (myRs.next()) {

        // retrieve data from result set row
        int id = myRs.getInt("id");
        int section = myRs.getInt("section");
        int subject = myRs.getInt("subject");
        int teacher = myRs.getInt("teacher");
        String time = myRs.getString("time");

        Teacher tempTeacher = loadTeacher(teacher);
        Subject tempSubject = loadSubject(subject);

        String teacher_name = tempTeacher.getfirstname() + " " +
tempTeacher.getlastname());

```

```

        }

    } catch (Exception e) {
        // TODO: handle exception
    } finally {
        // close JDBC objects
        close(myConn, myStmt);
    }
    return theClass;
}

public int UpdateClassTable(int classId,int teach,int sub ) {
    List<Teacher> temp_list = new ArrayList<>();
    try {
        // get connection from connection pool

        con = this.datasource.getConnection();
        stmt = con.createStatement();
        int result = stmt.executeUpdate("update classes set teacher="+teach+",
subject="+sub+" where id="+classId );

        if (result==1) {
            return 1;
        }
        else {
            return 0;
        }
    }

```

```

        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            close(con,stmt);
        }
        return 0;
    }

    public int UpdateClassTable(int classId,int sub ) {
        try {
            // get connection from connection pool

            con = this.datasource.getConnection();
            stmt = con.createStatement();
            int result = stmt.executeUpdate("update classes set subject="+sub+" where
id="+classId );

            if (result==1) {
                return 1;
            }
            else {
                return 0;
            }
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            close(con,stmt);
        }
    }

```



```
return 0;
```

```
}
```

```
}
```

Context.xml

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

<Context>

<Resource name="jdbc/Learners_Academy"

auth ="Container"

type="javax.sql.DataSource"

maxActive="20" maxIdle="5" maxWait="10000"

username="root" password="root"

driverClassName="com.mysql.cj.jdbc.Driver"

url="jdbc:mysql://localhost:3306/Learners_Academy"/>

</Context>
```