```java
package main.java.admin;

import java.io.IOException;
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.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.sql.DataSource;

import main.java.model.Student;
import main.java.model.Subject;
import main.java.model.Teacher;
import main.java.model.Class;

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

    private DbRetrieve dbRetrieve;

    @Resource(name = "Products")
    private DataSource datasource;

    @Override
    public void init() throws ServletException {

        super.init();

        // create instance of db util, to pass in conn pool object
        try {
            dbRetrieve = new DbRetrieve(datasource);

        } catch (Exception e) {
            throw new ServletException(e);
        }

    }

    /**
     * @see HttpServlet#HttpServlet()
     */
    public AdminControllerServlet() {
        super();
```

```
          // TODO Auto-generated constructor stub
     }

     @Override
     protected void doPost(HttpServletRequest req, HttpServletResponse 
resp) throws ServletException, IOException {

          doGet(req, resp);
     }

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

               // read the "command" parameter
               String command = request.getParameter("command");

               if (command == null) {
                    command = "CLASSES";
               }

               // if no cookeies
               if (!getCookies(request, response) && 
(!command.equals("LOGIN"))) {

                    response.sendRedirect("/Administrative-
Portal/login.jsp");
               }

               else {

                    // if there is no command, how to handle

                    // route the data to the appropriate method
                    switch (command) {

                    case "STUDENTS":
                         studentsList(request, response);
                         break;

                    case "TEACHERS":
                         teachersList(request, response);
                         break;

                    case "SUBJECTS":
```

```
                    subjectList(request, response);
                    break;

                case "CLASSES":
                    classestList(request, response);
                    break;

                case "ST_LIST":
                    classStudentsList(request, response);
                    break;

                case "LOGIN":
                    login(request, response);
                    break;

                default:
                    classestList(request, response);

                }
            }
        } catch (Exception e) {
            throw new ServletException(e);
        }
        // response.getWriter().append("Served at:
").append(request.getContextPath());
    }

    private void studentsList(HttpServletRequest request,
HttpServletResponse response) throws Exception {
        // get students from db util
        List<Student> students = dbRetrieve.getStudents();

        // add students to the request
        request.setAttribute("STUDENT_LIST", students);

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

    }

    private void teachersList(HttpServletRequest request,
HttpServletResponse response) throws Exception {
        // get students from db util
        List<Teacher> teachers = dbRetrieve.getTeachers();

        // add students to the request
        request.setAttribute("TEACHERS_LIST", teachers);

        // send it to the jSP view page
```

```
           RequestDispatcher dispatcher =
request.getRequestDispatcher("/teachers-list.jsp");
           dispatcher.forward(request, response);

      }

      private void subjectList(HttpServletRequest request,
HttpServletResponse response) throws Exception {
           // get subjects from db util
           List<Subject> subjects = dbRetrieve.getSubjects();

           // add subjects to the request
           request.setAttribute("SUBJECTS_LIST", subjects);

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

      }

      private void classestList(HttpServletRequest request,
HttpServletResponse response) throws Exception {
           // get subjects from db util
           List<Class> classes = dbRetrieve.getClasses();

           // add subjects to the request
           request.setAttribute("CLASSES_LIST", classes);

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

      }

      private void login(HttpServletRequest request,
HttpServletResponse response) throws Exception {
           String username = request.getParameter("username");
           String password = request.getParameter("password");

           if (username.toLowerCase().equals("admin") &&
password.toLowerCase().equals("admin")) {

                Cookie cookie = new Cookie(username, password);

                // Setting the maximum age to 1 day
                cookie.setMaxAge(86400); // 86400 seconds in a day

                // Send the cookie to the client
                response.addCookie(cookie);
                classestList(request, response);
```

```
        } else {
            RequestDispatcher dispatcher =
request.getRequestDispatcher("/login.jsp");
            dispatcher.forward(request, response);
        }

    }

    private void classStudentsList(HttpServletRequest request,
HttpServletResponse response) throws Exception {

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

        // get subjects from db util
        List<Student> students =
dbRetrieve.loadClassStudents(classId);

        // add subjects to the request
        request.setAttribute("STUDENTS_LIST", students);
        request.setAttribute("SECTION", section);
        request.setAttribute("SUBJECT", subject);

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

    }

    private boolean getCookies(HttpServletRequest request,
HttpServletResponse response) throws Exception {

        boolean check = false;
        Cookie[] cookies = request.getCookies();
        // Find the cookie of interest in arrays of cookies
        for (Cookie cookie : cookies) {

            if (cookie.getName().equals("admin") &&
cookie.getValue().equals("admin")) {
                check = true;
                break;
            }

        }


        return check;

    }
```

```
}
```

```
package main.java.admin;

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

import javax.sql.DataSource;

import main.java.model.Student;
import main.java.model.Subject;
import main.java.model.Teacher;
import main.java.model.Class;

public class DbRetrieve {

      private DataSource dataSource;

      public DbRetrieve(DataSource dataSource) {
            this.dataSource = dataSource;
      }

      public List<Student> getStudents() {

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

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

            try {

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

                  // create sql stmt
                  String sql = "SELECT * FROM students";
                  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 firstName = myRs.getString("fname");
                        String lastName = myRs.getString("lname");
                        int age = myRs.getInt("age");
                        int aclass = myRs.getInt("class");
```

```
                      // create new student object
                      Student tempStudent = new Student(id, firstName,
lastName, age, aclass);

                      // add it to the list of students
                      students.add(tempStudent);

                }

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

     }

     public List<Teacher> getTeachers() {

           List<Teacher> teachers = new ArrayList<>();

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

           try {

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

                // create sql stmt
                String sql = "SELECT * FROM teachers";
                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 firstName = myRs.getString("fname");
                      String lastName = myRs.getString("lname");
                      int age = myRs.getInt("age");

                      // create new student object
                      Teacher temp = new Teacher(id, firstName,
lastName, age);
```

```
                // add it to the list of students
                teachers.add(temp);

            }

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

    }

    public List<Subject> getSubjects() {

        List<Subject> subjects = new ArrayList<>();

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

        try {

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

            // create sql stmt
            String sql = "SELECT * FROM subjects";
            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");
                String shortcut = myRs.getString("shortcut");

                // create new student object
                Subject temp = new Subject(id, name,shortcut);

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

            }
```

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

	}

	public List<Class> getClasses() {

		List<Class> classes = new ArrayList<>();

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

		try {

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

			// create sql stmt
			String sql = "SELECT * FROM classes";
			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.getFname() + "
" + tempTeacher.getLname();

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

    }

    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("fname");
                String lname = myRs.getString("lname");
                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, myRs);
        }
        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");
                String shortcut = myRs.getString("shortcut");

                theSubject = new Subject(id, name,shortcut);

            }

        } catch (Exception e) {
            // TODO: handle exception
        } finally {
            // close JDBC objects
            close(myConn, myStmt, myRs);
        }
        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 clasess 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.getFname() + "
" + tempTeacher.getLname();

            }

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

    }

    public List<Student> loadClassStudents(int classId) {

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

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

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

            // create sql stmt
            String sql = "SELECT * FROM students WHERE class = " +
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");
                String firstName = myRs.getString("fname");
                String lastName = myRs.getString("lname");
                int age = myRs.getInt("age");
                int aclass = myRs.getInt("class");

                // create new student object
                Student tempStudent = new Student(id, firstName,
lastName, age, aclass);
                students.add(tempStudent);

            }

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

    }

    private void close(Connection myConn, Statement myStmt, ResultSet
myRs) {

        try {
            if (myRs != null) {
                myRs.close();
            }
            if (myStmt != null) {
                myStmt.close();
            }
            if (myConn != null) {
                myConn.close();
            }

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

    }

}
```

```
package main.java.admin;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;

import javax.annotation.Resource;
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.sql.DataSource;

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

     //Define datasource/connection pool for reference

     @Resource(name="Products")
     private DataSource dataSource;




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


           // Set the printwriter
           PrintWriter out = response.getWriter();
           response.setContentType("text/plain");

           // establish connection to the DB
           Connection myConn = null;
           Statement myStmt = null;
           ResultSet myRs = null;

           try {

                myConn = dataSource.getConnection();
           //create a sql statement
```

```
        String sql = "select * from students";
        myStmt = myConn.createStatement();


        //execute the sql statement
        myRs = myStmt.executeQuery(sql);

        //process the resultset
        while(myRs.next()) {
            String fname = myRs.getString("fname");
            out.println(fname);

        }



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






    }

}
```

```
package main.java.model;

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;
        }


}
```

```
package main.java.model;

public class Student {

      private int id;
      private String fname;
      private String lname;
      private int age;
      private int aclass;




      public Student(int id, String fname, String lname, int age, int
aclass) {
            super();
            this.id = id;
            this.fname = fname;
            this.lname = lname;
            this.age = age;
            this.aclass = aclass;
      }


      public int getId() {
            return id;
      }
      public void setId(int id) {
            this.id = id;
      }
      public String getFname() {
            return fname;
      }
      public void setFname(String fname) {
            this.fname = fname;
      }
      public String getLname() {
            return lname;
      }
      public void setLname(String lname) {
            this.lname = lname;
      }
      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 + ", fname=" + fname + ",
lname=" + lname + ", age=" + age + ", aclass=" + aclass
				+ "]";
	}



}
```

```java
package main.java.model;

public class Subject {

	private int id;
	private String name;
	private String shortcut;

	public Subject(int id, String name, String shortcut ) {
		super();
		this.id = id;
		this.name = name;
		this.shortcut = shortcut;
	}

	public int getId() {
		return id;
	}

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

	public String getShortcut() {
		return shortcut;
	}

	public void setShortcut(String shortcut) {
		this.shortcut = shortcut;
	}

	public String getName() {
		return name;
	}

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




}
```

```java
package main.java.model;

public class Teacher {

	private int id;
	private String fname;
	private String lname;
	private int age;

	public Teacher(int id, String fname, String lname, int age) {
		super();
		this.id = id;
		this.fname = fname;
		this.lname = lname;
		this.age = age;
	}

	public int getId() {
		return id;
	}

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

	public String getFname() {
		return fname;
	}

	public void setFname(String fname) {
		this.fname = fname;
	}

	public String getLname() {
		return lname;
	}

	public void setLname(String lname) {
		this.lname = lname;
	}

	public int getAge() {
		return age;
	}

	public void setAge(int age) {
		this.age = age;
	}



}
```

```
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http
://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-
app_3_1.xsd" id="WebApp_ID" version="3.1">
<display-name>Administrative-Portal-For Learner -Academy</display-name>
<welcome-file-list>
<welcome-file>AdminControllerServlet</welcome-file>
<welcome-file>index.jsp</welcome-file>
<welcome-file>index.html</welcome-file>
</welcome-file-list>
</web-app>
```

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>List of Classes</title>
<link type="text/css" rel="stylesheet" href="css/style.css">
</head>
<body style="background-image: url('css/background.jpg');">
    <div id="page">
        <jsp:include page="left-list.jsp" />


        <div id="wrapper">

            <div id="header">
                <h3>Classes</h3>
            </div>
        </div>


        <div id="container">

            <div id="content">

                <table>

                    <tr>

                        <th>Section</th>
                        <th>Subject</th>
                        <th>Teacher</th>
                        <th>Time</th>
                        <th>List of Students</th>

                    </tr>

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

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

                                <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>${tempClass.section}</td>
                            <td>${tempClass.subject}</td>
                            <td>${tempClass.teacher}</td>
```

```
                                               <td>${tempClass.time}</td>
                                               <td><a href="${tempLink
}">List</a></td>



                                          </tr>


                                   </c:forEach>

                            </table>
                     </div>
              </div>
       </div>

</body>
</html>
```

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

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Students of a Class</title>
<link type="text/css" rel="stylesheet" href="css/style.css">
</head>
<body style="background-image: url('css/background.jpg');">
<div id="page" >
    <jsp:include page="left-list.jsp" />


        <div id="wrapper">

            <div id="header">
                <h3>Students of ${SUBJECT} class section ${SECTION} </h3>
            </div>
        </div>


        <div id="container">

            <div id="content">

                <table>

                    <tr>

                        <th>First Name</th>
                        <th>Last Name</th>
                        <th>age</th>

                    </tr>

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

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



                        </tr>


                    </c:forEach>

                </table>
            </div>
        </div>
    </div>
```

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

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

<div class="sidenav">
    <h3 id="Logo">
        Administrative <br /> Academy Portal
    </h3>
    <c:url var="classesLink" value="AdminControllerServlet">
        <c:param name="command" value="CLASSES" />
    </c:url>

    <c:url var="subjectsLink" value="AdminControllerServlet">
        <c:param name="command" value="SUBJECTS" />
    </c:url>

    <c:url var="teachersLink" value="AdminControllerServlet">
        <c:param name="command" value="TEACHERS" />
    </c:url>

    <c:url var="studentsLink" value="AdminControllerServlet">
        <c:param name="command" value="STUDENTS" />
    </c:url>






    <a class="bar-item" href="${classesLink}">Classes</a>
        <a class="bar-item" href="${subjectsLink}">Subjects</a>
        <a class="bar-item" href="${teachersLink}">Teachers</a>
        <a class="bar-item" href="${studentsLink}">Students</a>
        <a class="bar-item" href="login.jsp">Log out</a>

</div>
```

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

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>List of Students</title>
<link type="text/css" rel="stylesheet" href="css/style.css">
</head>
<body style="background-image: url('css/background.jpg');">
<div id="page" >
    <jsp:include page="left-list.jsp" />


        <div id="wrapper">

            <div id="header">
                <h3>Students</h3>
            </div>
        </div>


        <div id="container">

            <div id="content">

                <table>

                    <tr>

                        <th>First Name</th>
                        <th>Last Name</th>
                        <th>age</th>


                    </tr>

                    <c:forEach var="tempStudent" items="${STUDENT_LIST }">
                        <tr>

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



                        </tr>


                    </c:forEach>

                </table>
            </div>
        </div>
```

```
       </div>

</body>
</html>
```

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

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Login</title>
 <link type="text/css" rel="stylesheet" href="css/login.css">
</head>
<body style="background-image: url('css/background.jpg');">

  <center> <h1> Admin Login </h1> </center>
    <form action="AdminControllerServlet" method="POST">
        <div class="container">
             <input type="hidden" name="command" value="LOGIN" />
            <label>Username : </label>
            <br/>
            <input type="text" placeholder="Enter Username" name="username" required>
            <br/>
            <label>Password : </label>
            <br/>
            <input type="password" placeholder="Enter Password" name="password"
required>
            <br/>
            <button type="submit">Login</button>
            <br/>
            <input type="checkbox" checked="checked"> Remember me

        </div>
    </form>


</body>
</html>
```

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>List of Teachers</title>
<link type="text/css" rel="stylesheet" href="css/style.css">
</head>
<body style="background-image: url('css/background.jpg');">
    <div id="page">
        <jsp:include page="left-list.jsp" />


        <div id="wrapper">

            <div id="header">
                <h3>Subjects</h3>
            </div>
        </div>


        <div id="container">

            <div id="content">

                <table>

                    <tr>

                        <th>Name</th>
                        <th>Shortcut</th>


                    </tr>

                    <c:forEach var="tempSubject" items="${SUBJECTS_LIST
}">
                        <tr>

                            <td>${tempSubject.name}</td>
                            <td>${tempSubject.shortcut}</td>


                        </tr>


                    </c:forEach>

                </table>
            </div>
        </div>
    </div>

</body>
</html>
```

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>List of Teachers</title>
<link type="text/css" rel="stylesheet" href="css/style.css">
</head>
<body style="background-image: url('css/background.jpg');">
    <div id="page">
        <jsp:include page="left-list.jsp" />


        <div id="wrapper">

            <div id="header">
                <h3>Teachers</h3>
            </div>
        </div>


        <div id="container">

            <div id="content">

                <table>

                    <tr>

                        <th>First Name</th>
                        <th>Last Name</th>
                        <th>age</th>

                    </tr>

                    <c:forEach var="tempStudent" items="${TEACHERS_LIST
}">
                        <tr>

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



                        </tr>


                    </c:forEach>

                </table>
            </div>
        </div>
    </div>
```

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