Step-1.

Open Eclipse EE > File>New>Other>Maven Project>Next>Create A Simple Maven Project>Next>Fill required Information.

Step-2.

Configure pom.xml file and add following dependencies:-

<%@pagelanguage=*"java"*contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

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

<!DOCTYPEhtml>

<html>

<head>

<metacharset=*"ISO-8859-1"*>

<title>Login page</title>

</head>

<body>

<center><h1> Admin Login </h1></center>

<formaction=*"AdminControllerServlet"*method=*"POST"*>

<inputtype=*"hidden"*name=*"command"*value=*"LOGIN"*/>

Enter Username : <inputtype=*"text"*name=*"username"*/>

<br/><br/>

Enter Password :<inputtype=*"password"*name=*"password"*/>

<br/><br/>

<buttontype = *"submit"*> Login </button>

<br/><br/>

</form>

</body>

</html>

**Output--**

Step-3.

In this step we make and servlet file to take the request and response and implements differnets cases for students , classes, teacher etc

package com.simplilearn.workshop.admin;

import java.io.IOException;

import java.util.List;

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 com.simplilearn.workshop.list.Student;

import com.simplilearn.workshop.list.Subject;

import com.simplilearn.workshop.list.Teacher;

import com.simplilearn.workshop.list.Class;

@WebServlet("/AdminControllerServlet")

public class AdminControllerServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

private DbRetrieve dbRetrieve;

private DataSource datasource;

@Override

public void init() throws ServletException {

super.init();

try {

dbRetrieve = new DbRetrieve(datasource);

} catch (Exception e) {

throw new ServletException(e);

}

}

public AdminControllerServlet() {

super();

// TODO Auto-generated constructor stub

}

@Override

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

doGet(req, resp);

}

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("/learners-acadamy/login.jsp");

}

else {

// if there is no command, how to handle route the data

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);

}

}

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("/students-list.jsp");

dispatcher.forward(request, response);

}

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

// get teachers from db util

List<Teacher> teachers = dbRetrieve.getTeachers();

// add teachers 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 classes from db util

List<Class> classes = dbRetrieve.getClasses();

// add classes 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.equals("admin") && password.equals("admin")) {

Cookie cookie = new Cookie(username, password);

// 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");

List<Student> students = dbRetrieve.loadClassStudents(classId);

// add student class 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("/students-class.jsp");

dispatcher.forward(request, response);

}

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

boolean check = false;

Cookie[] cookies = request.getCookies();

for (Cookie cookie : cookies) {

if (cookie.getName().equals("admin") && cookie.getValue().equals("admin")) {

check = true;

break;

}

}

return check;

}

}

Step-4

Creating a DBConnection file to connect database.

package com.simplilearn.workshop.admin;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DBConnection {

private static Connection connection;

public DBConnection () throws ClassNotFoundException, SQLException {

}

public static Connection getConnection() throws ClassNotFoundException, SQLException {

// load a driver

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

// obtain a connection

connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/administrative-portal?useSSL=false","root","1234");

return connection;

}

public void closeConnection() throws SQLException {

if(this.connection !=null) {

this.connection.close();

}

}

}

**Step-5**

Create a DBretrive file to featch data from database usins Statements , Connection and Resultset.

Make methods to send data from database to different java classes.These classes are using getters and settes to get the data.

package com.simplilearn.workshop.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 com.simplilearn.workshop.list.Student;

import com.simplilearn.workshop.list.Teacher;

import com.simplilearn.workshop.list.Subject;

import com.simplilearn.workshop.list.Class;

public class DbRetrieve {

private DataSource dataSource;

public DbRetrieve(DataSource dataSource) {

this.dataSource = dataSource;

}

public List<Student> getStudents() {

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

Connection myConnection = null;

Statement myStatement = null;

ResultSet myResultset = null;

try {

// get a connection

myConnection = DBConnection.getConnection();

// create sql stmt

String sql = "SELECT \* FROM `administrative-portal`.students";

myStatement = myConnection.createStatement();

// execute query

myResultset = myStatement.executeQuery(sql);

// process result

while (myResultset.next()) {

// retrieve data from result set row

int id = myResultset.getInt("id");

String firstName = myResultset.getString("fname");

String lastName = myResultset.getString("lname");

int age = myResultset.getInt("age");

int aclass = myResultset.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) {

} finally {

close(myConnection, myStatement, myResultset);

}

return students;

}

public List<Teacher> getTeachers() {

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

Connection myConnection = null;

Statement myStatement = null;

ResultSet myResultset = null;

try {

// get a connection

myConnection = DBConnection.getConnection();

// create sql stmt

String sql = "SELECT \* FROM `administrative-portal`.teachers";

myStatement = myConnection.createStatement();

// execute query

myResultset = myStatement.executeQuery(sql);

// process result

while (myResultset.next()) {

// retrieve data from result set row

int id = myResultset.getInt("id");

String firstName = myResultset.getString("fname");

String lastName = myResultset.getString("lname");

int age = myResultset.getInt("age");

// create new teacher object

Teacher temp = new Teacher(id, firstName, lastName, age);

// add it to the list of teachers

teachers.add(temp);

}

} catch (Exception e) {

} finally {

// close JDBC objects

close(myConnection, myStatement, myResultset);

}

return teachers;

}

public List<Subject> getSubjects() {

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

Connection myConnection = null;

Statement myStatement = null;

ResultSet myResultset = null;

try {

// get a connection

myConnection = DBConnection.getConnection();

// create sql stmt

String sql = "SELECT \* FROM `administrative-portal`.subjects";

myStatement = myConnection.createStatement();

// execute query

myResultset = myStatement.executeQuery(sql);

// process result

while (myResultset.next()) {

// retrieve data from result set row

int id = myResultset.getInt("id");

String name = myResultset.getString("name");

String shortcut = myResultset.getString("shortcut");

// create new subject object

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

// add it to the list of subjects

subjects.add(temp);

}

} catch (Exception e) {

} finally {

close(myConnection, myStatement, myResultset);

}

return subjects;

}

public List<Class> getClasses() {

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

Connection myConnection = null;

Statement myStatement = null;

ResultSet myResultset = null;

try {

// get a connection

myConnection = DBConnection.getConnection();

// create sql stmt

String sql = "SELECT \* FROM `administrative-portal`.classes";

myStatement = myConnection.createStatement();

// execute query

myResultset = myStatement.executeQuery(sql);

// process result

while (myResultset.next()) {

// retrieve data from result set row

int id = myResultset.getInt("id");

int section = myResultset.getInt("section");

int subject = myResultset.getInt("subject");

int teacher = myResultset.getInt("teacher");

String time = myResultset.getString("time");

Teacher tempTeacher = loadTeacher(teacher);

Subject tempSubject = loadSubject(subject);

String teacher\_name = tempTeacher.getFname() + "" + tempTeacher.getLname();

// create new class object

Class temp = new Class(id, section, teacher\_name, tempSubject.getName(), time);

// add it to the list of classes

classes.add(temp);

}

} catch (Exception e) {

} finally {

close(myConnection, myStatement, myResultset);

}

return classes;

}

public Teacher loadTeacher(int teacherId) {

Teacher theTeacher = null;

Connection myConnection = null;

Statement myStatement = null;

ResultSet myResultset = null;

try {

// get a connection

myConnection = DBConnection.getConnection();

// create sql stmt

String sql = "SELECT \* FROM `administrative-portal`.teachers WHERE id = " + teacherId;

myStatement = myConnection.createStatement();

// execute query

myResultset = myStatement.executeQuery(sql);

// process result

while (myResultset.next()) {

// retrieve data from result set row

int id = myResultset.getInt("id");

String fname = myResultset.getString("fname");

String lname = myResultset.getString("lname");

int age = myResultset.getInt("age");

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

}

} catch (Exception e) {

} finally {

close(myConnection, myStatement, myResultset);

}

return theTeacher;

}

public Subject loadSubject(int subjectId) {

Subject theSubject = null;

Connection myConnection = null;

Statement myStatement = null;

ResultSet myResultset = null;

try {

// get a connection

myConnection = DBConnection.getConnection();

// create sql stmt

String sql = "SELECT \* FROM `administrative-portal`.subjects WHERE id = " + subjectId;

myStatement = myConnection.createStatement();

// execute query

myResultset = myStatement.executeQuery(sql);

// process result

while (myResultset.next()) {

// retrieve data from result set row

int id = myResultset.getInt("id");

String name = myResultset.getString("name");

String shortcut = myResultset.getString("shortcut");

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

}

} catch (Exception e) {

} finally {

close(myConnection, myStatement, myResultset);

}

return theSubject;

}

public Class loadClass(int classId) {

Class theClass = null;

Connection myConnection = null;

Statement myStatement = null;

ResultSet myResultset = null;

try {

// get a connection

myConnection = DBConnection.getConnection();

// create sql stmt

String sql = "SELECT \* FROM `administrative-portal`.clasess WHERE id = " + classId;

myStatement = myConnection.createStatement();

// execute query

myResultset = myStatement.executeQuery(sql);

// process result

while (myResultset.next()) {

// retrieve data from result set row

int id = myResultset.getInt("id");

int section = myResultset.getInt("section");

int subject = myResultset.getInt("subject");

int teacher = myResultset.getInt("teacher");

String time = myResultset.getString("time");

Teacher tempTeacher = loadTeacher(teacher);

Subject tempSubject = loadSubject(subject);

String teacher\_name = tempTeacher.getFname() + "" + tempTeacher.getLname();

}

} catch (Exception e) {

} finally {

close(myConnection, myStatement, myResultset);

}

return theClass;

}

public List<Student> loadClassStudents(int classId) {

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

Connection myConnection = null;

Statement myStatement = null;

ResultSet myResultset = null;

try {

// get a connection

myConnection = DBConnection.getConnection();

// create sql stmt

String sql = "SELECT \* FROM `administrative-portal`.students WHERE class = " + classId;

myStatement = myConnection.createStatement();

// execute query

myResultset = myStatement.executeQuery(sql);

// process result

while (myResultset.next()) {

// retrieve data from result set row

int id = myResultset.getInt("id");

String firstName = myResultset.getString("fname");

String lastName = myResultset.getString("lname");

int age = myResultset.getInt("age");

int aclass = myResultset.getInt("class");

// create new student object

Student tempStudent = new Student(id, firstName, lastName, age,aclass);

students.add(tempStudent);

}

} catch (Exception e) {

} finally {

close(myConnection, myStatement, myResultset);

}

return students;

}

private void close(Connection myConnection, Statement myStatement, ResultSet myResultset) {

try {

if (myResultset != null) {

myResultset.close();

}

if (myStatement != null) {

myStatement.close();

}

if (myConnection != null) {

myConnection.close();

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

Step-6

All 4 java files Class,Students,Teacher and Subjects.

**Class.Java**

**package** com.simplilearn.workshop.list;

**publicclass** Class {

**privateint**id;

**privateint**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;

}

**publicint** getId() {

**return**id;

}

**publicvoid** setId(**int**id) {

**this**.id = id;

}

**publicint** getSection() {

**return**section;

}

**publicvoid** setSection(**int**section) {

**this**.section = section;

}

**public** String getTeacher() {

**return**teacher;

}

**publicvoid** setTeacher(String teacher) {

**this**.teacher = teacher;

}

**public** String getSubject() {

**return**subject;

}

**publicvoid** setSubject(String subject) {

**this**.subject = subject;

}

**public** String getTime() {

**return**time;

}

**publicvoid** setTime(String time) {

**this**.time = time;

}

}

**Student.java**

**package** com.simplilearn.workshop.list;

**publicclass** Student {

**privateint**id;

**private** String fname;

**private** String lname;

**privateint**age;

**privateint**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;

}

**publicint** getId() {

**return**id;

}

**publicvoid** setId(**int**id) {

**this**.id = id;

}

**public** String getFname() {

**return**fname;

}

**publicvoid** setFname(String fname) {

**this**.fname = fname;

}

**public** String getLname() {

**return**lname;

}

**publicvoid** setLname(String lname) {

**this**.lname = lname;

}

**publicint** getAge() {

**return**age;

}

**publicvoid** setAge(**int**age) {

**this**.age = age;

}

**publicint** getAclass() {

**return**aclass;

}

**publicvoid** setAclass(**int**aclass) {

**this**.aclass = aclass;

}

}

**Subject.java**

**package** com.simplilearn.workshop.list;

**publicclass** Subject {

**privateint**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;

}

**publicint** getId() {

**return**id;

}

**publicvoid** setId(**int**id) {

**this**.id = id;

}

**public** String getName() {

**return**name;

}

**publicvoid** setName(String name) {

**this**.name = name;

}

**public** String getShortcut() {

**return**shortcut;

}

**publicvoid** setShortcut(String shortcut) {

**this**.shortcut = shortcut;

}

}

**Teacher.java**

**package** com.simplilearn.workshop.list;

**publicclass** Teacher {

**privateint**id;

**private** String fname;

**private** String lname;

**privateint**age;

**public** Teacher(**int**id, String fname, String lname, **int**age) {

**super**();

**this**.id = id;

**this**.fname = fname;

**this**.lname = lname;

**this**.age = age;

}

**publicint** getId() {

**return**id;

}

**publicvoid** setId(**int**id) {

**this**.id = id;

}

**public** String getFname() {

**return**fname;

}

**publicvoid** setFname(String fname) {

**this**.fname = fname;

}

**public** String getLname() {

**return**lname;

}

**publicvoid** setLname(String lname) {

**this**.lname = lname;

}

**publicint** getAge() {

**return**age;

}

**publicvoid** setAge(**int**age) {

**this**.age = age;

}

}

**Step-7**

**In this step I created various jsp files for show data in UI.**

**Classes-List.jsp**

<%@pagelanguage=*"java"*contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

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

<!DOCTYPEhtml>

<html>

<head>

<metacharset=*"ISO-8859-1"*>

<title>List of classes</title>

</head>

<body>

<jsp:includepage=*"list.jsp"*/>

<h3>Classes</h3>

<table>

<tr>

<th>Section</th>

<th>Subject</th>

<th>Teacher</th>

<th>Time</th>

<th>List of Students</th>

</tr>

<c:forEachvar=*"tempClass"*items=*"*${CLASSES\_LIST }*"*>

<tr>

<c:urlvar=*"tempLink"*value=*"AdminControllerServlet"*>

<c:paramname=*"command"*value=*"ST\_LIST"*/>

<c:paramname=*"classId"*value=*"*${tempClass.id }*"*/>

<c:paramname=*"section"*value=*"*${tempClass.section }*"*/>

<c:paramname=*"subject"*value=*"*${tempClass.subject }*"*/>

</c:url>

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

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

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

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

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

</tr>

</c:forEach>

</table>

</body>

</html>

Output

list.jsp

<%@pagelanguage=*"java"*contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

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

<!DOCTYPEhtml>

<html>

<head>

<metacharset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<h3> administrative <br/> Academy portal </h3>

<c:urlvar=*"classesLink"*value=*"AdminControllerServlet"*>

<c:paramname=*"command"*value=*"CLASSESS"*/>

</c:url>

<c:urlvar=*"subjectsLink"*value=*"AdminControllerServlet"*>

<c:paramname=*"command"*value=*"SUBJECTS"*/>

</c:url>

<c:urlvar=*"teachersLink"*value=*"AdminControllerServlet"*>

<c:paramname=*"command"*value=*"TEACHERS"*/>

</c:url>

<c:urlvar=*"studentsLink"*value=*"AdminControllerServlet"*>

<c:paramname=*"command"*value=*"STUDENTS"*/>

</c:url>

<aclass=*"bar-item"*href=*"*${classesLink}*"*>Classes</a>

<aclass=*"bar-item"*href=*"*${subjectsLink}*"*>Subjects</a>

<aclass=*"bar-item"*href=*"*${teachersLink}*"*>Teachers</a>

<aclass=*"bar-item"*href=*"*${studentsLink}*"*>Students</a>

<aclass=*"bar-item"*href=*"login.jsp"*>Log out</a>

</body>

</html>

student-class.jsp

<%@pagelanguage=*"java"*contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

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

<!DOCTYPEhtml>

<html>

<head>

<metacharset=*"ISO-8859-1"*>

<title>Students of a class</title>

</head>

<body>

<jsp:includepage=*"list.jsp"*/>

<h3>Students of ${SUBJECT} class section ${SECTION} </h3>

<table>

<tr>

<th>First Name</th>

<th>Last Name</th>

<th>Age</th>

</tr>

<c:forEachvar=*"tempStudent"*items=*"*${STUDENTS\_LIST}*"*>

<tr>

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

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

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

</tr>

</c:forEach>

</table>

</body>

</html>

student-list.jsp

<%@pagelanguage=*"java"*contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

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

<!DOCTYPEhtml>

<html>

<head>

<metacharset=*"ISO-8859-1"*>

<title>List of Students</title>

</head>

<body>

<jsp:includepage=*"list.jsp"*/>

<h3>Students</h3>

<table>

<tr>

<th> ID</th>

<th>First Name</th>

<th>Last Name</th>

<th>Age</th>

</tr>

<c:forEachvar=*"tempStudent"*items=*"*${STUDENT\_LIST}*"*>

<tr>

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

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

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

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

</tr>

</c:forEach>

</table>

</body>

</html>

subject-list.jsp

<%@pagelanguage=*"java"*contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

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

<!DOCTYPEhtml>

<html>

<head>

<metacharset=*"ISO-8859-1"*>

<title>List of subjects</title>

</head>

<body>

<jsp:includepage=*"list.jsp"*/>

<h3>Subjects</h3>

<table>

<tr>

<th> ID </th>

<th>Name</th>

<th>Shortcut</th>

</tr>

<c:forEachvar=*"tempSubject"*items=*"*${SUBJECTS\_LIST }*"*>

<tr>

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

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

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

</tr>

</c:forEach>

</table>

</body>

</html>

teacher-list.jsp

<%@pagelanguage=*"java"*contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

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

<!DOCTYPEhtml>

<html>

<head>

<metacharset=*"ISO-8859-1"*>

<title>List of teachers</title>

</head>

<body>

<jsp:includepage=*"list.jsp"*/>

<h3>Teachers</h3>

<table>

<tr>

<th> ID </th>

<th>First Name</th>

<th>Last Name</th>

<th>Age</th>

</tr>

<c:forEachvar=*"tempStudent"*items=*"*${TEACHERS\_LIST }*"*>

<tr>

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

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

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

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

</tr>

</c:forEach>

</table>

</body>

</html>