|  |
| --- |
| **LearnersAcademyProject by**  **Sivaiah Gunda** |

Date of submission: 13-Sep-2022

Source Code GitHub Location: https://github.com/Sivaiah-Gunda/Java\_Simplilearn.git

Note: Please find the LockedMe.java file in [CompanyLocakersProject](https://github.com/Sivaiah-Gunda/Java_Simplilearn/tree/main/CompanyLocakersProject)/[src](https://github.com/Sivaiah-Gunda/Java_Simplilearn/tree/main/CompanyLocakersProject/src)/[com](https://github.com/Sivaiah-Gunda/Java_Simplilearn/tree/main/CompanyLocakersProject/src/com)/[company](https://github.com/Sivaiah-Gunda/Java_Simplilearn/tree/main/CompanyLocakersProject/src/com/company)/**lockers**/

Models:

|  |
| --- |
| 1. Classes.java   package com.simplilearn.models;  public class Classes  {  private int id;  private int section;  private String teacher;  private String subject;  private String time;  public Classes(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;  }  }  } |
| 2. Student.java  package com.simplilearn.models;  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 + "]";  }  } |
| 3. Subject.java  package com.simplilearn.models;  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;  }  } |
| 4. Teacher.java  package com.simplilearn.models;  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;  }  } |

Admin

|  |
| --- |
| AdimnControlServlet.java  package com.simplilearn.admin;  import java.io.IOException;  import java.util.List;  import javax.servlet.RequestDispatcher;  import javax.servlet.ServletException;  import javax.servlet.http.Cookie;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  import com.simplilearn.dao.DbRetrieve;  import com.simplilearn.models.Classes;  import com.simplilearn.models.Student;  import com.simplilearn.models.Subject;  import com.simplilearn.models.Teacher;  /\*\*  \* Servlet implementation class AdimControlServlet  \*/  public class AdimnControlServlet extends HttpServlet  {  private static final long serialVersionUID = 1L;  private DbRetrieve dbRetrieve;  public void init()  {  String jdbcURL = getServletContext().getInitParameter("jdbcURL");  String jdbcUsername = getServletContext()  .getInitParameter("jdbcUsername");  String jdbcPassword = getServletContext()  .getInitParameter("jdbcPassword");  dbRetrieve = new DbRetrieve(  jdbcURL,  jdbcUsername,  jdbcPassword);  }  @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  {  dbRetrieve.connect();  // 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("/SampleProject/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("/ListStudents.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("/TeacherList.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("/SubjectList.jsp");  dispatcher.forward(request, response);  }  private void classestList(HttpServletRequest request,  HttpServletResponse response) throws Exception  {  // get subjects from db util  List<Classes> classes = dbRetrieve.getClasses();  // add subjects to the request  request.setAttribute("CLASSES\_LIST", classes);  // send it to the jSP view page  RequestDispatcher dispatcher = request  .getRequestDispatcher("/ClassList.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("/ClassStudents.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;  }  } |

DAO

|  |
| --- |
| package com.simplilearn.dao;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import java.util.ArrayList;  import java.util.List;  import com.simplilearn.models.Classes;  import com.simplilearn.models.Student;  import com.simplilearn.models.Subject;  import com.simplilearn.models.Teacher;  public class DbRetrieve  {  private String jdbcURL;  private String jdbcUsername;  private String jdbcPassword;  private Connection jdbcConnection;  public DbRetrieve(String jdbcURL, String jdbcUsername, String jdbcPassword)  {  this.jdbcURL = jdbcURL;  this.jdbcUsername = jdbcUsername;  this.jdbcPassword = jdbcPassword;  }  public void connect() throws SQLException  {  if (jdbcConnection == null || jdbcConnection.isClosed())  {  try  {  Class.forName("com.mysql.jdbc.Driver");  }  catch (Exception e)  {  throw new SQLException(  e);  }  jdbcConnection = DriverManager.getConnection(jdbcURL, jdbcUsername,  jdbcPassword);  }  }  public List<Student> getStudents()  {  List<Student> students = new ArrayList<>();  Statement myStmt = null;  ResultSet myRs = null;  try  {  // create sql stmt  String sql = "SELECT \* FROM students";  myStmt = jdbcConnection.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("aclass");  // 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(myStmt, myRs);  }  return students;  }  public List<Teacher> getTeachers()  {  List<Teacher> teachers = new ArrayList<>();  Statement myStmt = null;  ResultSet myRs = null;  try  {  // create sql stmt  String sql = "SELECT \* FROM teachers";  myStmt = jdbcConnection.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 teacher object  Teacher temp = new Teacher(  id,  firstName,  lastName,  age);  // add it to the list of teachers  teachers.add(temp);  }  }  catch (Exception e)  {  // TODO: handle exception  }  finally  {  // close JDBC objects  close(myStmt, myRs);  }  return teachers;  }  public List<Subject> getSubjects()  {  List<Subject> subjects = new ArrayList<>();  Statement myStmt = null;  ResultSet myRs = null;  try  {  // create sql stmt  String sql = "SELECT \* FROM subjects";  myStmt = jdbcConnection.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(myStmt, myRs);  }  return subjects;  }  public List<Classes> getClasses()  {  List<Classes> classes = new ArrayList<>();  Statement myStmt = null;  ResultSet myRs = null;  try  {  // create sql stmt  String sql = "SELECT \* FROM classes";  myStmt = jdbcConnection.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  Classes temp = new Classes(  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(myStmt, myRs);  }  return classes;  }  public Teacher loadTeacher(int teacherId)  {  Teacher theTeacher = null;  Statement myStmt = null;  ResultSet myRs = null;  try  {  // create sql stmt  String sql = "SELECT \* FROM teachers WHERE id = " + teacherId;  myStmt = jdbcConnection.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  closeDbResources(myStmt, myRs);  }  return theTeacher;  }  public Subject loadSubject(int subjectId)  {  Subject theSubject = null;  Statement myStmt = null;  ResultSet myRs = null;  try  {  // create sql stmt  String sql = "SELECT \* FROM subjects WHERE id = " + subjectId;  myStmt = jdbcConnection.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  closeDbResources(myStmt, myRs);  }  return theSubject;  }  public List<Student> loadClassStudents(int classId)  {  List<Student> students = new ArrayList<>();  Statement myStmt = null;  ResultSet myRs = null;  try  {  // create sql stmt  String sql = "SELECT \* FROM students WHERE aclass = " + classId;  myStmt = jdbcConnection.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("aclass");  // 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(myStmt, myRs);  }  return students;  }  private void close(Statement myStmt, ResultSet myRs)  {  try  {  if (myRs != null)  {  myRs.close();  }  if (myStmt != null)  {  myStmt.close();  }  if (jdbcConnection != null)  {  jdbcConnection.close();  }  }  catch (Exception e)  {  e.printStackTrace();  }  }  private void closeDbResources(Statement myStmt, ResultSet myRs)  {  try  {  if (myRs != null)  {  myRs.close();  }  if (myStmt != null)  {  myStmt.close();  }  }  catch (Exception e)  {  e.printStackTrace();  }  }  } |

JSP

|  |
| --- |
| ClassList.jsp  <%@ 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=*"style.css"*>  </head>  <body style="background-image: *url('paper.gif')*;">  <div id=*"page"*>  <jsp:include page=*"LeftList.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=*"*${CLASSES\_LIST }*"*>  <tr>  <c:url var=*"tempLink"* value=*"AdminControlServlet"*>  <c:param name=*"command"* value=*"ST\_LIST"* />  <c:param name=*"classId"* value=*"*${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> |
| ClassList.jsp  <%@ 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=*”style.css”*>  </head>  <body style=”background-image: *url(‘paper.gif’)*;”>  <div id=*”page”* >  <jsp:include page=*”LeftList.jsp”* />  <div id=*”wrapper”*>  <div id=*”header”*>  <h3>Students of ${SUBJECT} class section ${SECTION} </h3>  </div>  </div>  <div id=*”container”*>  <div id=*”content”*>  <table>  <tr>  <th>ID</th>  <th>First Name</th>  <th>Last Name</th>  <th>Age</th>  <th>Class</th>  </tr>  <c:forEach var=*”tempStudent”* items=*”*${STUDENTS\_LIST}*”*>  <tr>  <td>${tempStudent.id}</td>  <td>${tempStudent.fname}</td>  <td>${tempStudent.lname}</td>  <td>${tempStudent.age}</td>  <td>${tempStudent.aclass}</td>  </tr>  </c:forEach>  </table>  </div>  </div>  </div>  </body>  </html> |
| LeftList.jsp  <%@ taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"*%>  <div class=*"sidenav"*>  <h3 id=*"logo"*>  Learners Academy  </h3>  <c:url var=*"classesLink"* value=*"AdminControlServlet"*>  <c:param name=*"command"* value=*"CLASSES"* />  </c:url>  <c:url var=*"subjectsLink"* value=*"AdminControlServlet"*>  <c:param name=*"command"* value=*"SUBJECTS"* />  </c:url>  <c:url var=*"teachersLink"* value=*"AdminControlServlet"*>  <c:param name=*"command"* value=*"TEACHERS"* />  </c:url>  <c:url var=*"studentsLink"* value=*"AdminControlServlet"*>  <c:param name=*"command"* value=*"STUDENTS"* />  </c:url>    <a class=*"bar-item"* href=*"*${classesLink}*"*>Classes</a>  <a class=*"bar-item"* href=*"*${subjectsLink}*"*>Subjects</a>  <a class=*"bar-item"* href=*"*${teachersLink}*"*>Teachers</a>  <a class=*"bar-item"* href=*"*${studentsLink}*"*>Students</a>  <a class=*"bar-item"* href=*"Login.jsp"*>Log out</a>  </div> |
| ListStudents.jsp  <%@ taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"*%>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"ISO-8859-1"*>  <title>List of Students</title>  <link type=*"text/css"* rel=*"stylesheet"* href=*"style.css"*>  </head>  <body style="background-image: *url('paper.gif')*;">  <div id=*"page"* >  <jsp:include page=*"LeftList.jsp"* />    <div id=*"wrapper"*>  <div id=*"header"*>  <h3>Students</h3>  </div>  </div>  <div id=*"container"*>  <div id=*"content"*>  <table>  <tr>  <th>ID</th>  <th>First Name</th>  <th>Last Name</th>  <th>Age</th>  <th>Class</th>  </tr>  <c:forEach var=*"tempStudent"* items=*"*${STUDENT\_LIST }*"*>  <tr>  <td>${tempStudent.id}</td>  <td>${tempStudent.fname}</td>  <td>${tempStudent.lname}</td>  <td>${tempStudent.age}</td>  <td>${tempStudent.aclass}</td>  </tr>  </c:forEach>  </table>  </div>  </div>  </div>  </body>  </html> |
| Login.jsp  <%@ taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"*%>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"ISO-8859-1"*>  <title>Login</title>  <link type=*"text/css"* rel=*"stylesheet"* href=*"css/login.css"*>  </head>  <body style="background-image: *url('css/loginbg.jpg')*;">  <center> <h1>Administrator Login</h1> </center>  <form action=*"AdimnControlServlet"* method=*"POST"*>  <div class=*"container"*>  <input type=*"hidden"* name=*"command"* value=*"LOGIN"* />  <label>Username : </label>  <br/>  <input type=*"text"* placeholder=*"Enter Username"* name=*"username"* required>  <br/>  <label>Password : </label>  <br/>  <input type=*"password"* placeholder=*"Enter Password"* name=*"password"* required>  <br/>  <button type=*"submit"*>Login</button>  <br/>  <input type=*"checkbox"* checked=*"checked"*> Remember me    </div>  </form>  </body>  </html> |
| SubjectList.jsp  <%@ taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"*%>  <!DOCTYPE html>  <html>  <head>  <meta charset=*"ISO-8859-1"*>  <title>List of Teachers</title>  <link type=*"text/css"* rel=*"stylesheet"* href=*"style.css"*>  </head>  <body style="background-image: *url('paper.gif')*;">  <div id=*"page"*>  <jsp:include page=*"LeftList.jsp"* />  <div id=*"wrapper"*>  <div id=*"header"*>  <h3>Subjects</h3>  </div>  </div>  <div id=*"container"*>  <div id=*"content"*>  <table>  <tr>  <th>Name</th>  <th>Shortcut</th>    </tr>  <c:forEach var=*"tempSubject"* items=*"*${SUBJECTS\_LIST }*"*>  <tr>  <td>${tempSubject.name}</td>  <td>${tempSubject.shortcut}</td>  </tr>  </c:forEach>  </table>  </div>  </div>  </div>  </body>  </html> |
| TeacherList.jsp  <%@ 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=*"style.css"*>  </head>  <body style="background-image: *url('paper.gif')*;">  <div id=*"page"*>  <jsp:include page=*"LeftList.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> |
| Screenshots |
| Login |
| Classes |
| Students in Section wise |

|  |
| --- |
| Subjects |
| Teachers |

|  |
| --- |
| Students |
| End of the Document |