**Project Name: - Learne’s Academy**

This is project created by SHWETA TANWAR. In this project JAVA language has been used. This coding has been done on JDK 11. Mysql has been used for database.

All the coding has available on My github account – [shwetatanwar@github.com](mailto:shwetatanwar@github.com)

**(1)WorkFlow of Project:-**

First create database in my sql. Here in my case database name is**:administrative\_portal.**

* Create table like:
* **for Student List** : Table named as “First year”, **“**SeondYear”, “ThirdYear”, “FourthYear”
* **for Subject List:** Table named as “First yearSubject”,SeondYearSubject, ThirdYearSubject,FourthYearSubject
* **For Teacher List:** Table named as “teacher”
* **For Classes List:** Table named as “Classes”
* **Schema for Student List:** FirstYear,SeondYear, ThirdYear,FourthYear with field (int id primary key, firstname varchar(15),lastname varchar(15),age int,assign\_class varchar(10)
* **Schema for table teacher:** (int id primary key,name varchar(20),gender varchar(10),assign\_class varchar(10),assign\_subject Unique varchar(40))

**Note: More than one subject can assign to teacher with different id for different subject but subject should be unique subject.**

* **Schema for Subject List:** (subjectname Unique varchar(40))
* **Schema for table classes:** (classid int primary key,classname varchar(10))

**(2) Classes Details:**

1. **Package name: com.models** (generic java classes with table fields for different tables)

**Classes list:** **I. Student classes:** named as “First year”, **“**SeondYear”, “ThirdYear”, “FourthYear**”**

**II. Teacher Class**

**III.** **Subject classes:** named as “First yearSubject”,SeondYearSubject, ThirdYearSubject,FourthYearSubject

**IV. Classes class**

1. **Package name: com.dao** (It is a object/interface, which is used to access data from database of data storage)

Classes list: **I.TeacherDao**

**II.StudentDao Classes:** named as FirstDao,SecondDao,ThirdDao,FourthDao

**III. SubjectDao**

**IV. ClassesDao**

Database Connectivity:JDBC

1. **Package name: com.web** (Servlet classes to retrieve data from DAO and display it on web using jsp pages)

Classes list:

**I.TeacherServlet :** It takes object from TeacherDao class and print it over web using teachers.jsp page.

to display list of teacher details with provision to insert teacher details dynamically over web page using “subject\_assign.jsp” page.

**(3)List of methods in TeacherServlet class:**

1. **insertSubject method:** to insert teacher details. It takes object from TeacherDao class and print it over web after inserting again redirect to Teacherservelet to display updated list.
2. **subject\_assign methods:**

**There are seprate subject assign for each year named as subject\_assign, subject\_assign1, subject\_assign2, subject\_assign3, for First Year, Second Year, Third Year and Fourth Year subjects respectively.** It redirect over respective “subject\_assign page” to take dynamic input for teacher table details.(note: assign\_class field is by default selected when you click on respective subject assign year) After taking input from user over web it again redirect to “/**listTeachers” method of** TeacherServlet to see updated teacher list over web.

1. **Service method:** it uses switch cases for various option like ""/assignSubject",/assignSubject1","/assignSubject2","/assignSubject3", "/insertSubject","/listTeachers" with (request, response).
2. **listTeachers method:** to print teacher details using object of TeacherDao class. It takes objects from TeacherDao class and print it over web using teacher.jsp page.

**II.Student Servlets:named as: First Servlet,Second Servlet,ThirdServlet,FourthServlet.** It takes object from **StudentDao classes** and print it over web using **students.jsp pages: named as “first.jsp, second.jsp, third.jsp, fourth.jsp” There is option to add new Student for repective year class.**

**III. SubjectServlet:** It takes object from **SubjectDao class** and print it over web using **subjects.jsp page.**

**IV. ClassesServelet:** It takes object from **ClassesDao** class and print it over web using **classes.jsp page**

**V.LoginServlet:** It has login credentials if matches redirect to **dashboard.jsp** where you can see details of academy like. studentlist, teacherlist,classeslist,subjectlist by clicking on respective tittle. If credential not matches again to **login.jsp page** with error message.

VI.LogoutServelet: to logout from session using **logout.jsp.**

VI. ViewAllRecordServlet: To view all records using jsp page named as allStudent.jsp.

**(4) JSP pages:-**

1. **index.jsp:**  redirect to index page
2. **login.jsp:** goto login page control by LoginServelt to check credential if credential are true redirect to Dashboard where thereis option toview a Class Report which will show all the information about the class, such as the list of student, subjects, and teachers. If credentials are wrong again it redirect to login.jsp
3. **teachers.jsp:** retrieve request from TeacherServlet to display list in tabular form over web page.
4. **Students recods jsp pages:** named as first.jsp,second.jsp,third.jsp,fourth.jsp retrieve request from respective servlet named a FirstServlet, SecondServlet, ThirdServlet, FourthServlet to display list in tabular form over web page.
5. **Addstudent jsp pages form:** named as addstudent1.jsp,addstudent2.jsp, addstudent4.jsp, addstudent4.jsp to take dynamic input from web for every year separately.
6. **classes.jsp:** retrieve request from ClassesServlet to display list in tabular form over web page.
7. **subjects.jsp:** retrieve request from TeacherServlet to display list in tabular form over web page.
8. **subject\_assign.jsp pages:named as subject\_assign,subject\_assign1, subject\_assign2, subject\_assign3.** It provides action form to take dynamic entries for teacher tables and after submit it retrieve request from TeacherServlet to display updated teacher list in tabular form over web page. It redirect to TeacherServlet using request.getContextPath()%>*/insertSubject"*
9. **header.jsp:** list of web reference for web page.
10. **allstudent.jsp:** to print all records using input from ViewAllRecordServlet.
11. **complete\_recod:** web referenced links for every records like View StudentList, View Teacher List, View SubjectList, View ClassesList

**(6) Scrum :-**

During the scrum we do meeting with the product owner, sprint master , developer team members , and understood the requirements , user stories, product log and decided to make 2 sprints to complete this application.

Sprint 1-

1. Time Period for this sprint is 1 week. In this sprint individual person has been assigned different class to create.
2. Different classes has been assigned to different members according to there capabilities.
3. Daily scrum call has been scheduled , so that the members can coordinate with each other.

Sprint 2-

1. Time period for this sprint is 3 days . In this sprint all different classes will be assemble into one project.
2. Database created, and connectivity established.
3. Developer team will assure that all function are working fine together.
4. By giving different type of inputs the functionality will be check in every condition.
5. Hence it’s an application based on coding , member will try to find bug and fix it.
6. Secure access to application , data security ,user access limitation , owner privileges will be assure in this sprints.

**(7) Learner’sAcademy Flow:-**

**Administrator**

**Login Page**

**Check credentials**

False **credentials**

True **credentials**

**Dashboard Page to view classes report with options:** [Login](http://localhost:8082/MyAssignment/login.jsp) | [Logout](http://localhost:8082/MyAssignment/LogoutServlet)

[View Complete Record of Academy](http://localhost:8082/MyAssignment/ViewAllRecordServlet) ||[View StudentList](http://localhost:8082/MyAssignment/studentlist.jsp) [View TeacherList](http://localhost:8082/MyAssignment/TeacherServlet)[View ClassesList](http://localhost:8082/MyAssignment/ClassesServlet) [View SubjectList](http://localhost:8082/MyAssignment/SubjectServlet)

[View Complete Record of Academy](http://localhost:8082/MyAssignment/ViewAllRecordServlet)

[View TeacherList](http://localhost:8082/MyAssignment/TeacherServlet)

[View SubjectList](http://localhost:8082/MyAssignment/SubjectServlet)

[View ClassesList](http://localhost:8082/MyAssignment/ClassesServlet)

[View StudentList](http://localhost:8082/MyAssignment/studentlist.jsp)

**Options like : Assign FirstYearSubject, Assign SecondYearSubject, Assign ThirdYearSubject,Assign FourthYearSubject**

**All Subject List like first year subject, second subjects viz..**

1. **First Year**
2. **Second Year**
3. **Third Year**
4. **Fourth Year**

**Fourth Year**

**Third year**

**First Year**

**Second Year**

**Click on respective year subject assign option, a form will open with year already selected. Fill all Records of Teacher Table in form.**

## Add new Student option for each class separately. Add student details to insert new student

**Submit Submit**

**Records Saved in Respective Class and return back to Updated classes list**

***Back To Teacher List to view Updated list***

**Records Saved in TeacherList**

**(6) Scrum :-**

During the scrum we do meeting with the product owner, sprint master , developer team members , and understood the requirements , user stories, product log and decided to make 2 sprints to complete this application.

Sprint 1-

1. Time Period for this sprint is 1 week. In this sprint individual person has been assigned different class to create.
2. Different classes has been assigned to different members according to there capabilities.
3. Daily scrum call has been scheduled , so that the members can coordinate with each other.

Sprint 2-

1. Time period for this sprint is 3 days . In this sprint all different classes will be assemble into one project.
2. Database created, and connectivity established.
3. Developer team will assure that all function are working fine together.
4. By giving different type of inputs the functionality will be check in every condition.
5. Hence it’s an application based on coding , member will try to find bug and fix it.
6. Secure access to application , data security ,user access limitation , owner privileges will be assure in this sprints.