Learner's Academy Project Specification Document

Project objective:

To design and develop a backend administrative portal for the Learner's Academy.

Background of the problem statement:

Learner's Academy is a school that has an online management system. The system keeps track of its classes, subjects, students, and teachers. It has a back-office application with a single administrator login.

The administrator can:

- Set up a master list of all the subjects for all the classes
- Set up a master list of all the teachers
- Set up a master list of all the classes
- Assign classes for subjects from the master list
- Assign teachers to a class for a subject (A teacher can be assigned to different classes fordifferent subjects)
- Get a master list of students (Each student must be assigned to a single class)

There will be an option to view a Class Report which will show all the information about the class, such as the list of students, subjects, and teachers.

Learner's Academy Algorithm:

1. Create a jsp login file with the post method having username, password and login button.

2. Create login servlet write code in do post and set username and password statically. If the given username and password matches then redirect to welcome.jsp, else throw an error message and redirect to login.html. Provide username on session for security as it checks the credentials forevery login. (Add servlet api jar after creating a servlet.)

3. Welcome.jsp first checks if login was made by checking username, if done, it prints Welcome message and gives hyperlinks to dashboard.html and logout. If not, it redirects to login.html.

4. dashboard.html prints welcome message and gives hyperlinks to view class report and to logout.

5. In dashboard servlet, establish jdbc connection first and create tables for classes, subjects and teachers using sql queries. (Add mysql connector jar here) Insert values to subject table by using prepared statement. Create a table to map and assign teachers to subjects and insert values in it using prepared statement. Create students table using sql query.

6. Create a query which gives a class report including classes, subjects, teachers, students using inner joins in sql. Save the output in result set and print all rows in browser using while loop (.next)

7. Create a logout servlet and if user wants to logout in step 2 or step 4, he will be logged out successfully and was redirected to login.html.

Concepts Used:

1. Java Core concepts

2. Servlets

3. Session login and logout

4. Servlet Request dispatcher.

5. Exceptions

6. HttpServlet Request

7. HttpServlet Response

8. SQL

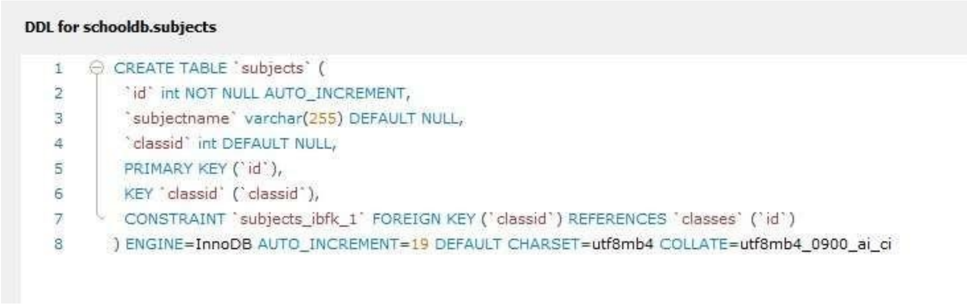
9. Tomcat

10. JSP

11. HTML

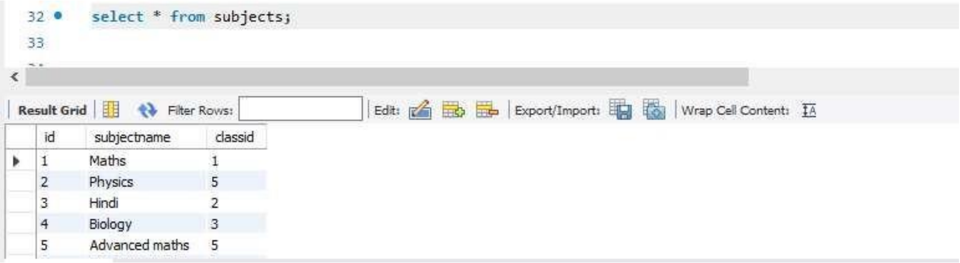
Tables used:

Subjects table:



DDL for schooldb.subjects

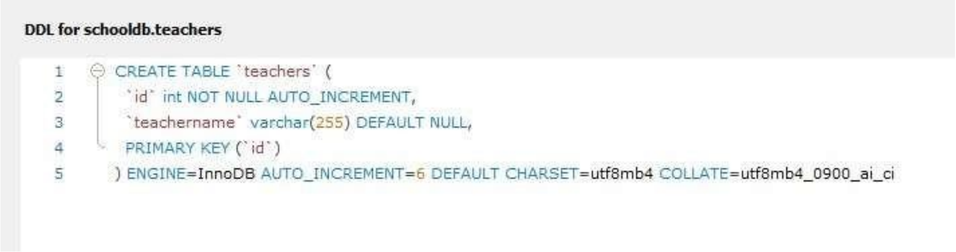
```
CREATE TABLE `subjects` (
  `id` int NOT NULL AUTO_INCREMENT,
  `subjectname` varchar(255) DEFAULT NULL,
  `classid` int DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `classid` (`classid`),
  CONSTRAINT `subjects_ibfk_1` FOREIGN KEY (`classid`) REFERENCES `classes` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=19 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```



```
select * from subjects;
```

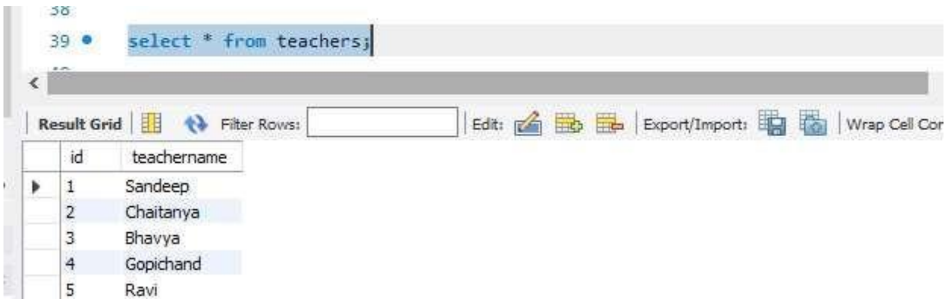
id	subjectname	classid
1	Maths	1
2	Physics	5
3	Hindi	2
4	Biology	3
5	Advanced maths	5

Teachers table:



DDL for schooldb.teachers

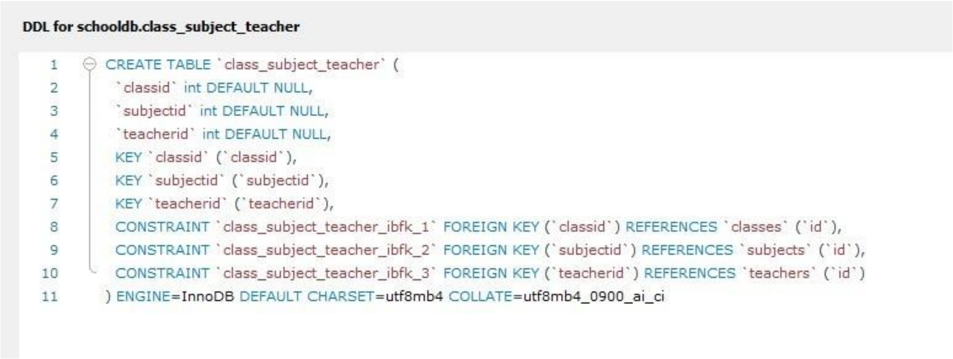
```
CREATE TABLE `teachers` (
  `id` int NOT NULL AUTO_INCREMENT,
  `teachername` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```



```
select * from teachers;
```

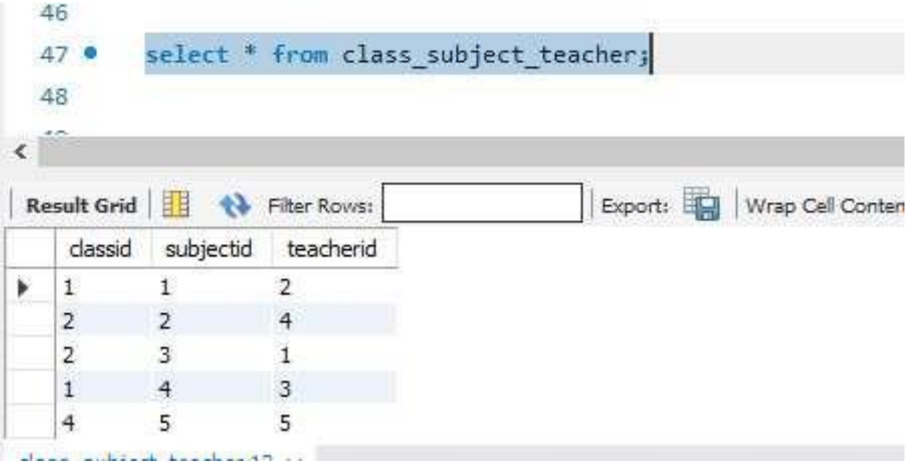
id	teachername
1	Sandeep
2	Chaitanya
3	Bhavya
4	Gopichand
5	Ravi

class_subject_teacher table:



DDL for schooldb.class_subject_teacher

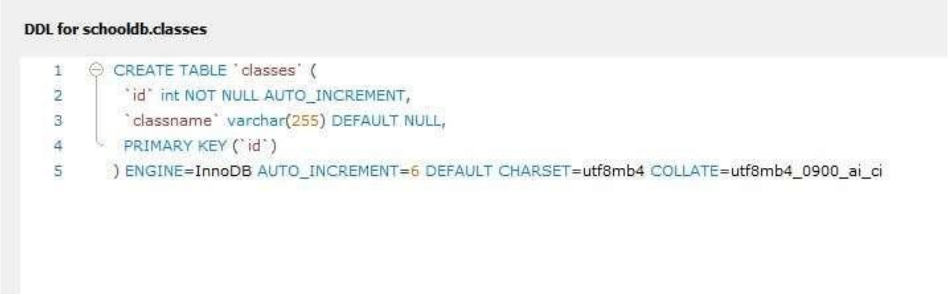
```
CREATE TABLE `class_subject_teacher` (
  `classid` int DEFAULT NULL,
  `subjectid` int DEFAULT NULL,
  `teacherid` int DEFAULT NULL,
  KEY `classid` (`classid`),
  KEY `subjectid` (`subjectid`),
  KEY `teacherid` (`teacherid`),
  CONSTRAINT `class_subject_teacher_ibfk_1` FOREIGN KEY (`classid`) REFERENCES `classes` (`id`),
  CONSTRAINT `class_subject_teacher_ibfk_2` FOREIGN KEY (`subjectid`) REFERENCES `subjects` (`id`),
  CONSTRAINT `class_subject_teacher_ibfk_3` FOREIGN KEY (`teacherid`) REFERENCES `teachers` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```



```
select * from class_subject_teacher;
```

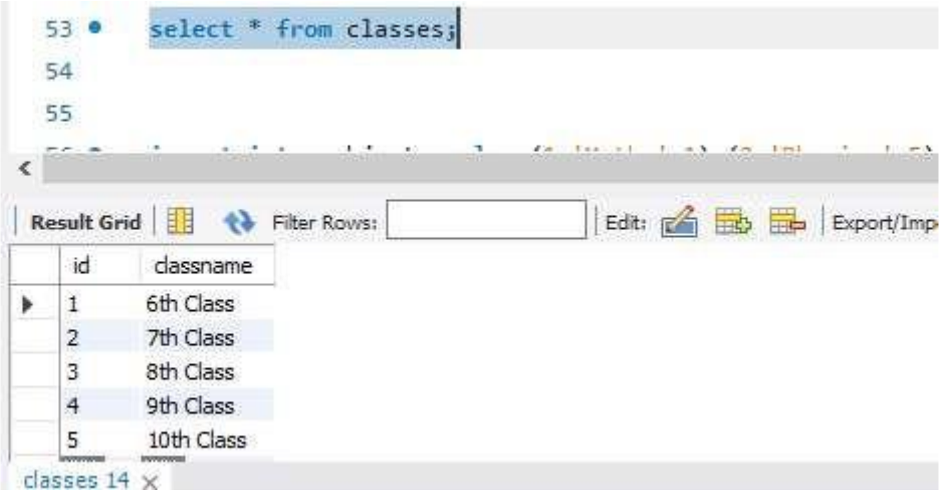
classid	subjectid	teacherid
1	1	2
2	2	4
2	3	1
1	4	3
4	5	5

Classes table:



DDL for schooldb.classes

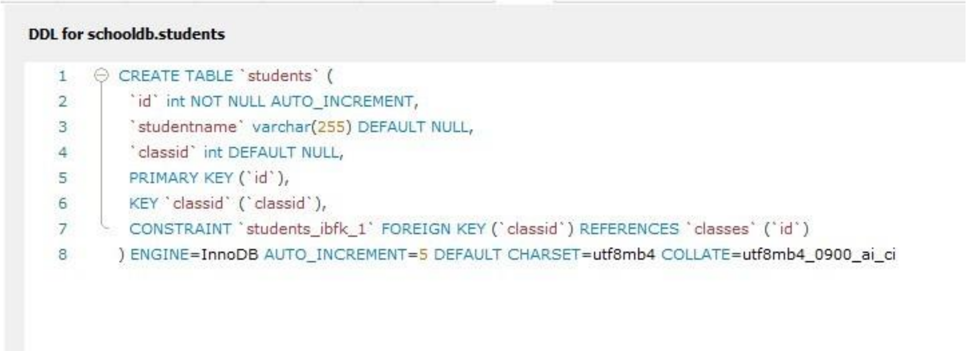
```
CREATE TABLE `classes` (
  `id` int NOT NULL AUTO_INCREMENT,
  `classname` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```



```
select * from classes;
```

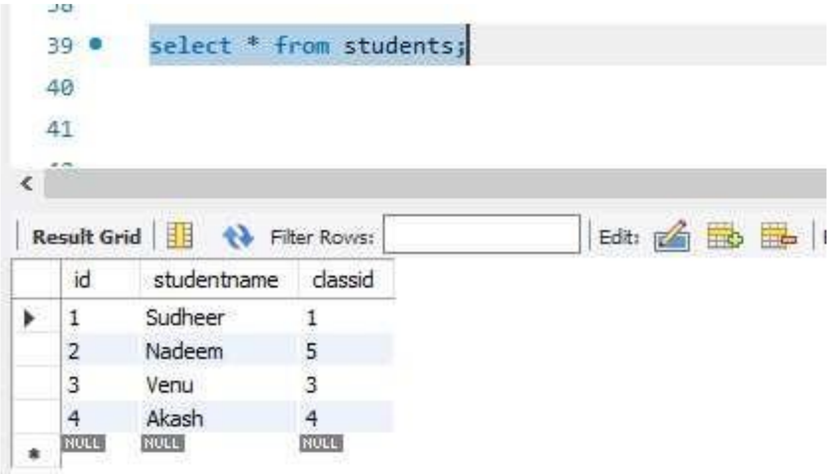
id	classname
1	6th Class
2	7th Class
3	8th Class
4	9th Class
5	10th Class

Students table:



DDL for schooldb.students

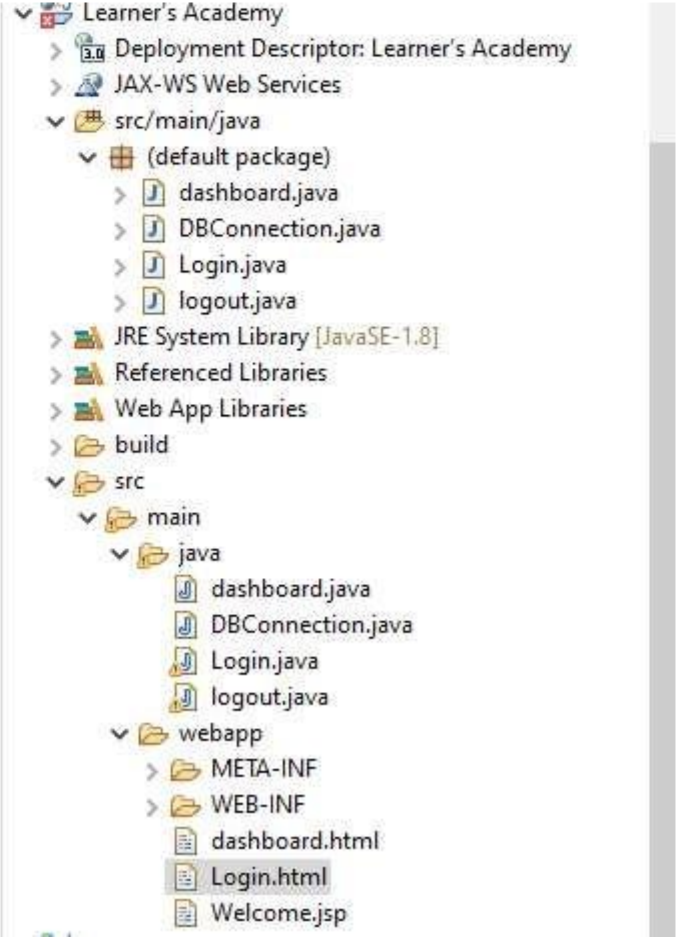
```
CREATE TABLE `students` (
  `id` int NOT NULL AUTO_INCREMENT,
  `studentname` varchar(255) DEFAULT NULL,
  `classid` int DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `classid` (`classid`),
  CONSTRAINT `students_ibfk_1` FOREIGN KEY (`classid`) REFERENCES `classes` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```



```
select * from students;
```

id	studentname	classid
1	Sudheer	1
2	Nadeem	5
3	Venu	3
4	Akash	4
NULL	NULL	NULL

Project Files:



```
Learner's Academy
  Deployment Descriptor: Learner's Academy
  JAX-WS Web Services
  src/main/java
    (default package)
      dashboard.java
      DBConnection.java
      Login.java
      logout.java
  JRE System Library [JavaSE-1.8]
  Referenced Libraries
  Web App Libraries
  build
  src
    main
      java
        dashboard.java
        DBConnection.java
        Login.java
        logout.java
      webapp
        META-INF
        WEB-INF
        dashboard.html
        Login.html
        Welcome.jsp
```

GitHub Link:

https://github.com/Digvijayrajput08/Learners-Academy.git

Conclusion:

Learner's Academy Portal helps Admins of the academy to login and can fetch the details of class, such as the list of students, subjects, and teachers.