**Developing a Backend Admin for Learner’s Academy**

DESCRIPTION

**Project objective:**

As a Full Stack Developer, design and develop a backend administrative portal for the Learner’s Academy. Use the GitHub repository to manage the project artifacts.

**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 for different 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  
       
The goal of the company is to deliver a high-end quality product as early as possible.

**Project:** Develop a Backend Admin for Learner’s Academy

**Developer:** Hridoy Sutar

**Github link:** [**https://github.com/hridoysutar1/Developing-a-Backend-Admin-for-Learner-s-Academy**](https://github.com/hridoysutar1/Developing-a-Backend-Admin-for-Learner-s-Academy)

**Sprint:**

The project is planned to be completed in 1 sprint.

Planning:

1)Creating the flow of the application

2)Writing Servlet, java, jsp code .

3)Testing codes .

4) Pushing code to GitHub.

5)Document writing

**Tool:** Eclipse, MySql workbench

**Server:** Tomcat

**Front End**: JSP

***Backend:*** Servlet

**Databse:** JDBC

**Concept used:**

**Java**: Class, Object,method, access modifier, condition statements, loop, switching statements,list, arraylist, searching, sorting.

**Servlet:** HttpServlet, request,response, session,doget,dopost,init(),service(),destroy(),request dispatcher, lifecycle, parameters, attribute,.

**JSP:** Declarative tag, scriplet tag, expressive tag, page directive, include directive, taglib, custom tag, jsp implicit objects, redirection, jstl.

**JDBC:** Jdbc driver, driver manager, connection, statement, resultset, retrive file.

**Process:**

**Creation of new project in Eclipse:**

File> New> Dynamic web project>enter name, select target runtime and configuration> selct path>finish

My project name Learner Academy Portal For Admin

**Addtion of external .jar :**

Mysql-connector-java-8.0.30.jar and jstl-1.2.jar added to the following path.

Learner Academy Portal For Admin/src/main/webapp/WEB-INF/lib

**Package creation:**

Two package com.admin and com.info created in src/main/java .

**Servlet Creation:**

1)Main.java servlet creation

com.admin>new > servlet> enterd name Main>next> selected servlet method> finish

2)login.java servlet creation

com.admin>new > servlet> enterd name login>next> selected servlet method> finish

3)logout.java servlet creation

com.admin>new > servlet> enterd name logout>next> selected servlet method> finish

**Class creation:**

1) DbConnection.java creation:

com.admin>new> Class> entered name as DbConnection>finish

2)DB.java creation

com.admin>new> Class> entered name as DB >finish

3)Teacher.java creation:

com.info>new>Class> entered name as Teacher> finish

4)Subject.java creation:

com.info>new>Class> entered name as Subject> finish

5)Class.java creation:

com.info>new>Class> entered name as Class> finish

6)Student.java creation:

com.info>new>Class> entered name as Student> finish

**JSP creation:**

1)login.jsp

src>webapp>new>jsp>entered name as login.jsp> finish

2)Report.jsp

src>webapp>new>jsp>entered name as Report.jsp> finish

3)classreport.jsp

src>webapp>new>jsp>entered name as classreport.jsp> finish

4)logout.jsp

src>webapp>new>jsp>entered name as logout.jsp> finish

5)subjectlist.jsp

src>webapp>new>jsp>entered name as subjectlist.jsp> finish

6)studentlist.jsp

src>webapp>new>jsp>entered name as studentlist.jsp> finish

7)teacherlis.jsp

src>webapp>new>jsp>entered name as teacherlis.jsp> finish

**Database and table creation:**

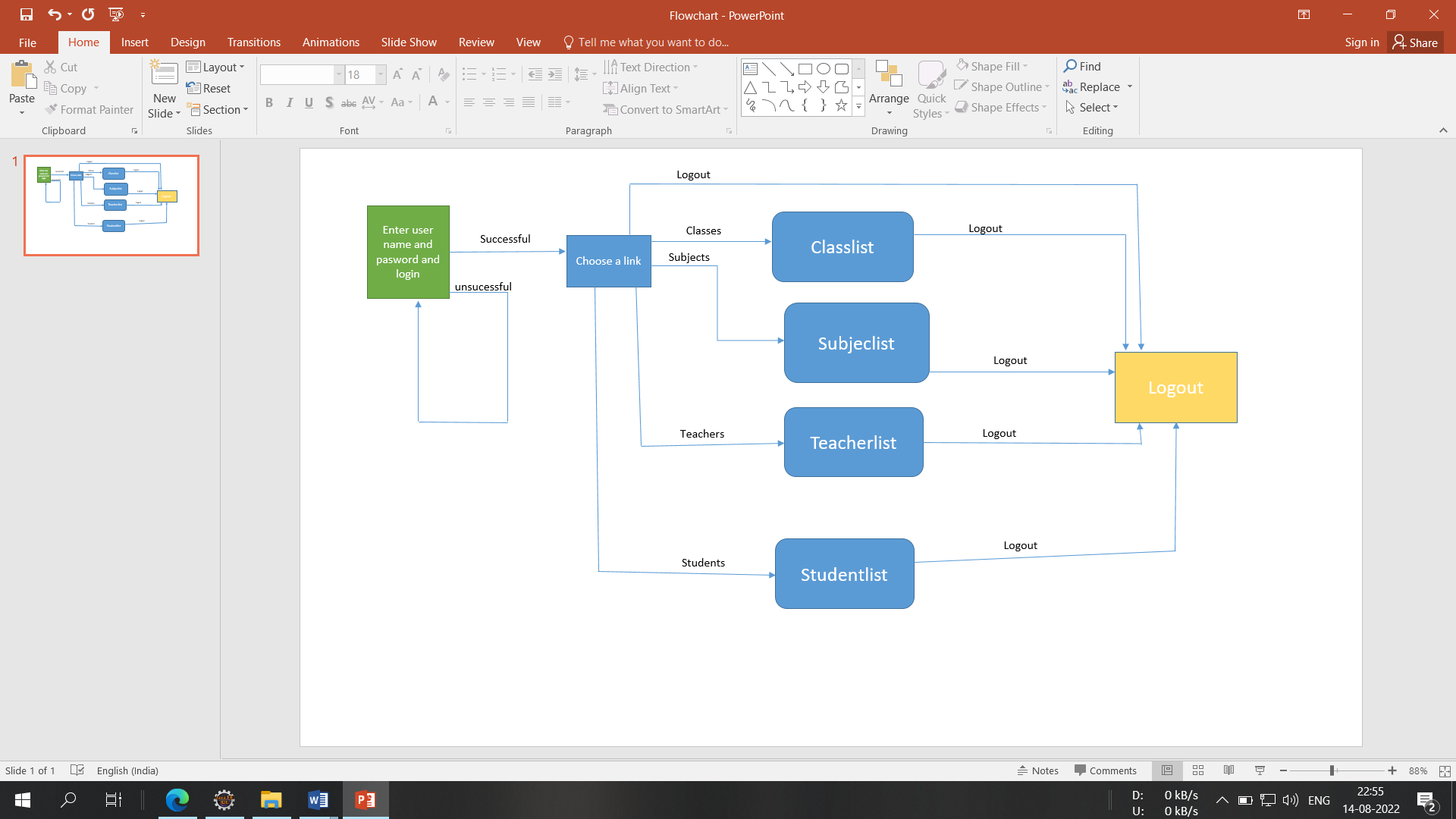
MySql workbench> followed mysql command

Create database academy;

use academy;

select database();

**Flow Chart:**



**Pushing to git repository:**

* Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

* Initialize your repository using the following command:

git init

* Add all the files to your git repository using the following command:

git add .

* Commit the changes using the following command:

git commit . -m “Changes have been committed.”

* Push the files to the folder you initially created using the following command:

git push -u origin master

**Conclusion:**

Admin can login by username and passwoed and can check list of class report, student list, teacher list, subject list.

This application can by further developed by adding features of data insertion and deletion.