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

1. **Developing a Backend Admin for Learner’s Academy:**

**This document contains sections for:**

* Project Description
* [Core concepts used in project](file:///C:\Users\Asus\Downloads\LockedMe%20-%20Virtual%20Key%20for%20Repositories.docx#Core_concepts)
* [Flow of the Application](file:///C:\Users\Asus\Downloads\LockedMe%20-%20Virtual%20Key%20for%20Repositories.docx#Flow).
* Project Users Stories : ( Agile and Scrum )
* Git Repositories
* How to run project
* [Demonstrating the product capabilities, appearance, and user interactions.](file:///C:\Users\Asus\Downloads\LockedMe%20-%20Virtual%20Key%20for%20Repositories.docx#Product_capability)
* [Unique Selling Points of the Application](file:///C:\Users\Asus\Downloads\LockedMe%20-%20Virtual%20Key%20for%20Repositories.docx#USP)
* [Conclusions](file:///C:\Users\Asus\Downloads\LockedMe%20-%20Virtual%20Key%20for%20Repositories.docx#Conclusions)

The code for this project is hosted at :

https://github.com/Karuna-Sri3/Phase2-Learners-academy.git

* 1. **Project 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.

**The flow and features of the application:**

● Plan more than two sprints to complete the application  
● Document the flow of the application and prepare a flow chart   
● List the core concepts and algorithms being used to complete this application  
● Implement the appropriate concepts, such as exceptions, collections, and sorting techniques for source code optimization and increased performance

**You must use the following:**

● Eclipse/IntelliJ: An IDE to code for the application   
● Java: A programming language to develop the web pages, databases, and others  
● SQL: To create tables for admin, classes, students, and other specifics  
● Git: To connect and push files from the local system to GitHub   
● GitHub: To store the application code and track its versions   
● Scrum: An efficient agile framework to deliver the product incrementally   
● Search and Sort techniques: Data structures used for the project   
● Specification document: Any open-source document or Google Docs

**The following requirements should be met:**

● The source code should be pushed to your GitHub repository. You need to document the steps and write the algorithms in it.  
● The submission of your GitHub repository link is mandatory. In order to track your task, you need to share the link of the repository. You can add a section in your document.   
● Document the process step-by-step starting from sprint planning to the product release.   
● The application should not close, exit, or throw an exception if the user specifies an invalid input.  
● You need to submit the final specification document which will include:   
● Project and developer details   
● Sprints planned and the tasks achieved in them   
● Algorithms and flowcharts of the application   
● Core concepts used in the project   
● Links to the GitHub repository to verify the project completion

* 1. **Core concepts used in the project:**

-> Used Java Language in Eclipse IDE.

-> File Handling

->Collections framework

-> Sorting

-> Flow Control

-> Recursion

-> Exception Handling

-> Streams API

1. **Architecture diagram / flow chart**



* 1. **Project Users Stories : ( Agile and Scrum )**

The project is planned to be completed in 3 sprints. Tasks assumed to be completed in the sprint are:

* Creating the flow of the application
* Initializing git repository to track changes as development progresses.
* Writing the Java program to fulfill the requirements of the project.
* Testing the Java program with different kinds of User input
* Pushing code to GitHub.

1) As an admin I can Set up a master list of all the subjects for all the classes

2) As an admin I can Set up a master list of all the teachers

3) As an admin I can Set up a master list of all the classes

4) As an admin I can Assign classes for subjects from the master list

5) As an admin I can Assign teachers to a class for a subject (A teacher can be assigned to different classes for different subjects)

6) As an admin I can Get a master list of students (Each student must be assigned to a single class).

7) As an admin I can create an option to view a Class Report which will show all the information about the class, such as the list of students, subjects, and teacher.

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

Sprint 1

1) As an admin I can Set up a master list of all the subjects for all the classes

2) As an admin I can Set up a master list of all the teachers

3) As an admin I can Set up a master list of all the classes

Sprint 2

4) As an admin I can Assign classes for subjects from the master list

5) As an admin I can Assign teachers to a class for a subject (A teacher can be assigned to different classes for different subjects)

6) As an admin I can Get a master list of students (Each student must be assigned to a single class).

Sprint 3

7) As an admin I can create an option to view a Class Report which will show all the information about the class, such as the list of students, subjects, and teacher.

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

**3. Project git Repositories**

link : https://github.com/Karuna-Sri3/Phase2-Learners-academy.git

**Directory Structure / package:**

## **Demonstrating the product capabilities, appearance, and user interactions**

To demonstrate the product capabilities, below are the sub-sections configured to highlight appearance and user interactions for the project:

## **Step 1:** Creating a new project in Eclipse

* Open Eclipse
* Go to File -> New -> Project ->Maven Project -> Next.
* Type in any project name and click on “Finish.”
* Select your project and go to File -> New -> Class.

**Step 2:**

**Java files**

## **AdminControllerServlet.java**

## **DbRetrieve.java**

## **TestServlet.java**

[Class.java](https://github.com/jyothsnamukhee03/phase2-backend-admin-for-Learners-academy/blob/main/src/main/java/model/Class.java)

[Student.java](https://github.com/jyothsnamukhee03/phase2-backend-admin-for-Learners-academy/blob/main/src/main/java/model/Student.java)

[Subject.java](https://github.com/jyothsnamukhee03/phase2-backend-admin-for-Learners-academy/blob/main/src/main/java/model/Subject.java)

[Teacher.java](https://github.com/jyothsnamukhee03/phase2-backend-admin-for-Learners-academy/blob/main/src/main/java/model/Teacher.java)

**Jsp files**

[class-students.jsp](https://github.com/jyothsnamukhee03/phase2-backend-admin-for-Learners-academy/blob/main/WebContent/class-students.jsp)

[classes-list.jsp](https://github.com/jyothsnamukhee03/phase2-backend-admin-for-Learners-academy/blob/main/WebContent/classes-list.jsp)

[left-list.jsp](https://github.com/jyothsnamukhee03/phase2-backend-admin-for-Learners-academy/blob/main/WebContent/left-list.jsp)

[list-students.jsp](https://github.com/jyothsnamukhee03/phase2-backend-admin-for-Learners-academy/blob/main/WebContent/list-students.jsp)

[login.jsp](https://github.com/jyothsnamukhee03/phase2-backend-admin-for-Learners-academy/blob/main/WebContent/login.jsp)

[subjects-list.jsp](https://github.com/jyothsnamukhee03/phase2-backend-admin-for-Learners-academy/blob/main/WebContent/subjects-list.jsp)

[teachers-list.jsp](https://github.com/jyothsnamukhee03/phase2-backend-admin-for-Learners-academy/blob/main/WebContent/teachers-list.jsp)

**CSS files**

[add-student-style.css](https://github.com/jyothsnamukhee03/phase2-backend-admin-for-Learners-academy/blob/main/WebContent/css/add-student-style.css)

[login.css](https://github.com/jyothsnamukhee03/phase2-backend-admin-for-Learners-academy/blob/main/WebContent/css/login.css)

[style.css](https://github.com/jyothsnamukhee03/phase2-backend-admin-for-Learners-academy/blob/main/WebContent/css/style.css)

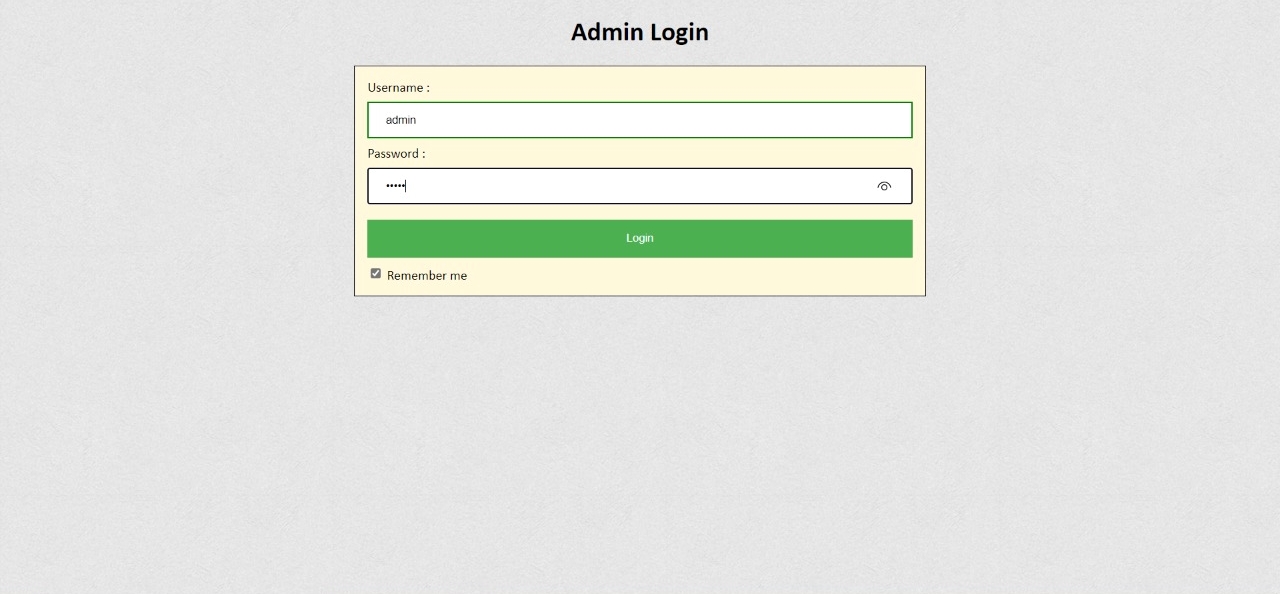
## **administrative-portal.sql**

All above mentioned file are attached via zip file all these are

**Step 3:**

**Outputs:**

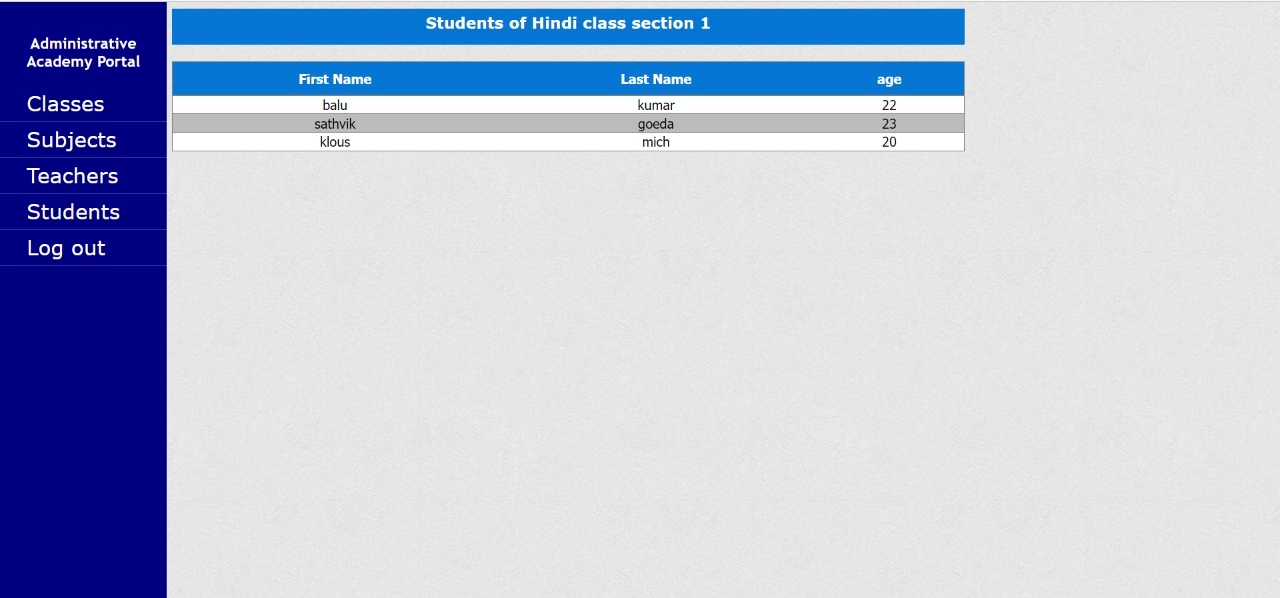
**Admin login**

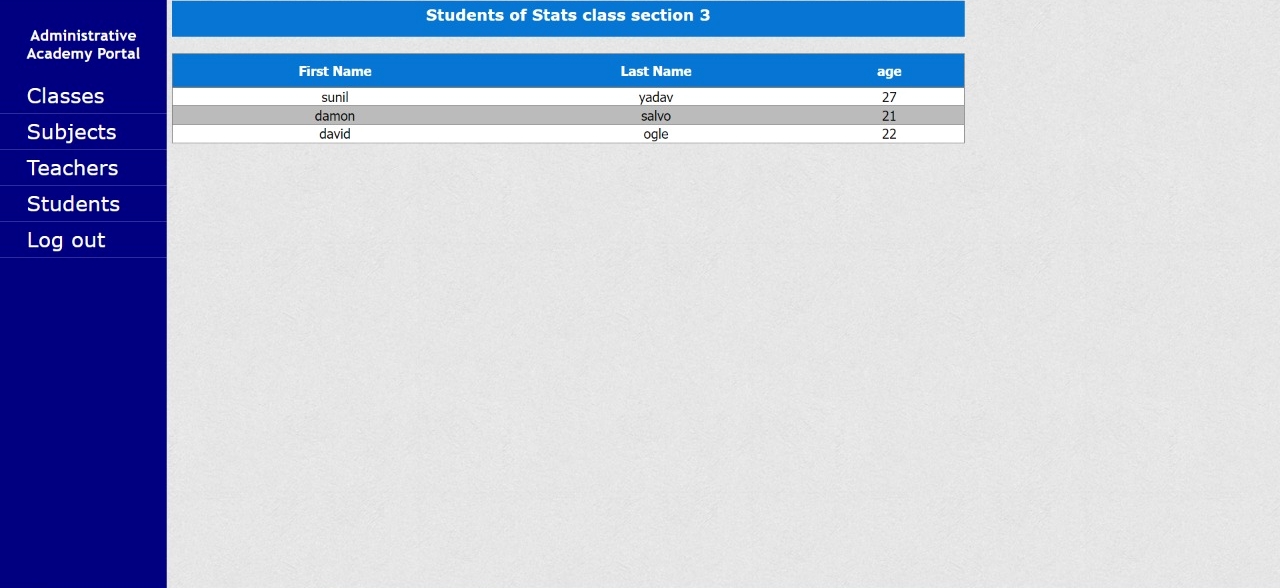


**Classes:**

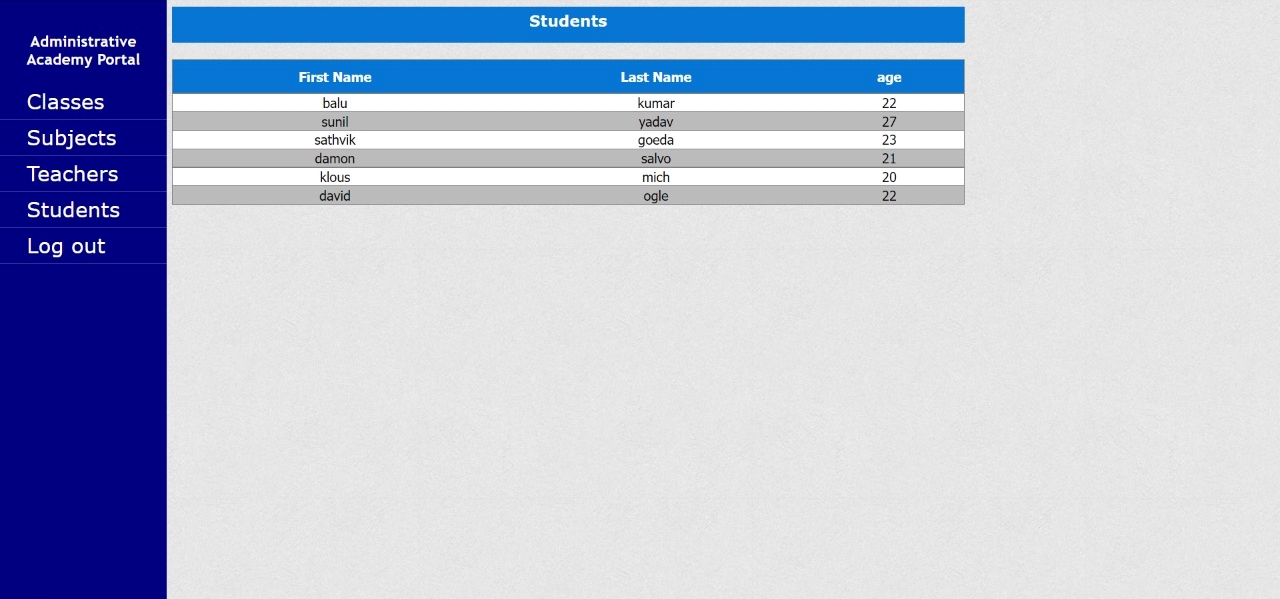


## **On clicking list:**

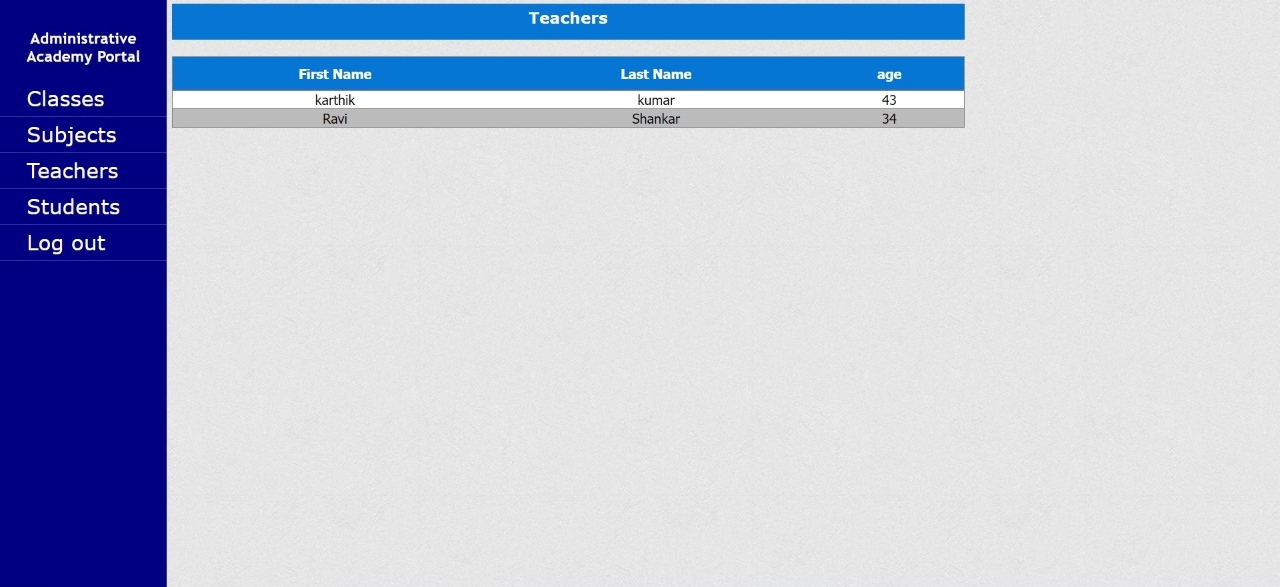




Students



## **Teachers**



## **Step 4:** Pushing the code to GitHub repository

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

*cd <folder path>*

* Initialize 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 <commit message>*

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

*git push -u origin master*

## **Unique Selling Points of the Application**

1. Scheduled timetables for the teachers and students can be maintained easily.
2. The data of the subjects, classes, students and teachers can be edited easily.
3. High security for the data as the admin only can access the data.
4. Searching for any data about classes, subject, students and teachers is made easy

## **Conclusions**

In the program an application has been developed with a duration of two spirits. This application makes handling the data of the learner’s academy. All the data about the classes, subject, teachers, students and their schedule are maintained. The admin can login through a User ID and password and manipulated the data.