Virtual key foryourrepository

# ProjectContains:

1. Projectdetails
2. Developerdetails
3. SprintsPlanning
4. Flowcharts&Algorithms
5. Coreconcepts used fortheproject
6. GithubFiles

# ProjectDetails:

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

# DeveloperDetails:

* Hanuman Prasad (https://github.com/hanumanprasadvishwa/Administrative-portal-for-Learner-Academy-main)
* [hanumanprasad0108@gmail.com](mailto:hanumanprasad0108@gmail.com)

# SprintsPlanning:

Thescopeof projecthasbeendividedinto3sprints.

### Sprint1:

Understandingtherequirementoftheprojectandcreatingaflowchart&algorithm forthesame.

### Sprint2:

### - Writing the SQL Queries to create the tables in database as per the requirement of the application.

- Writing SQL queries to apply constraints and indexes on each table as per requirement.

- Writing SQL query to insert data into the tables.

### Sprint3:

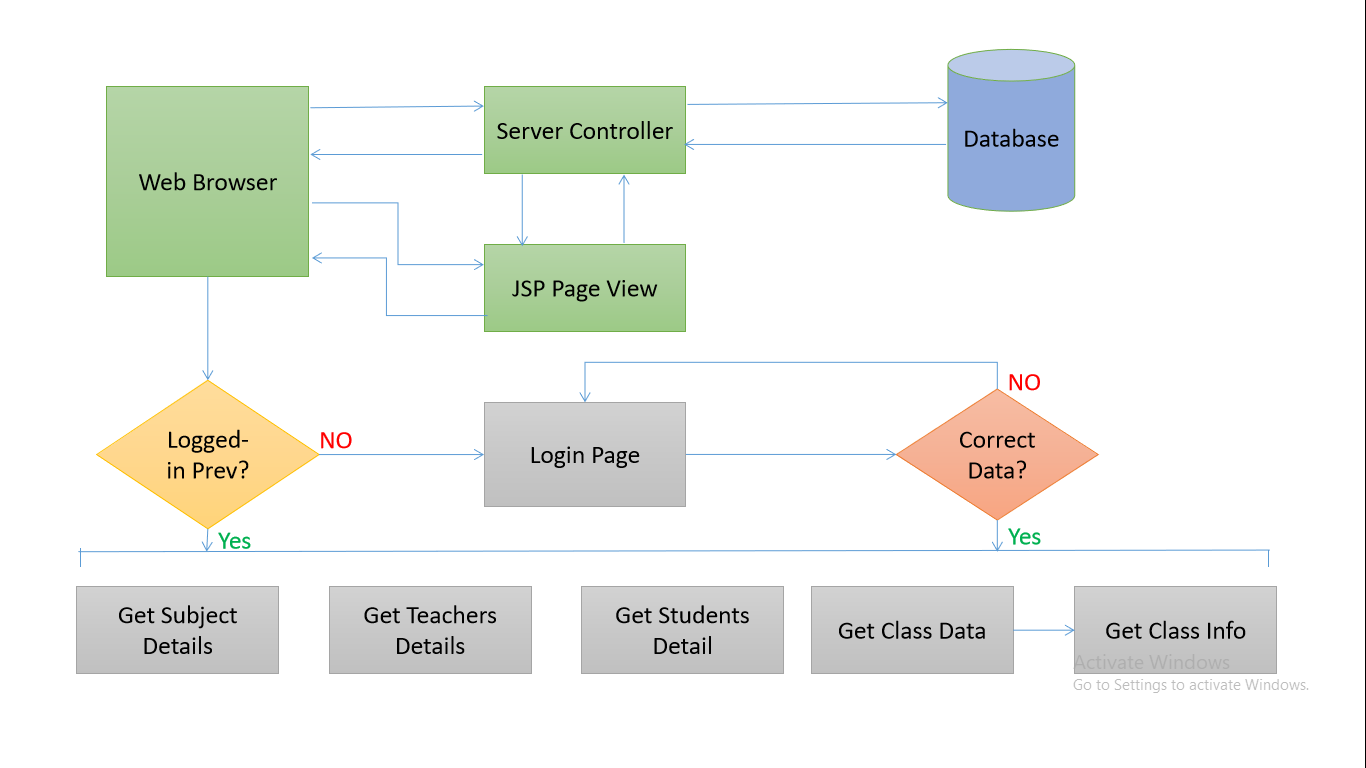
Creating the Login Page & dashboard for the Application. Creating java Programme forthevarioustasksaspertheflowchartcreated.

### Sprint4:

Testing and fixing the issues occurred in the application and validating with various inputs & PushingtheprogramtoGithub.

# FlowCharts&Algorithms:

### **Flowchart:**



**Algorithm:**

Step 1 - Start

Step 2 - Login Using UserName and Password.

If UserName And Password Not correct then Re-Enter UserName and Password.

Step 3 - Login Using UserName and Password.

User and Password is correct then Dashboard page will get open.

Step 4 - Click on CLASSES menu on the left to see the list of classes.

TO see the list of students in a class click on LIST button mentioned on the right side.

Step 5 - Click on SUBJECTS menu on the left to see the list of SUBJECTS.

Step 6 - Click on TEACHERS menu on the left to see the list of TEACHERS.

Step 7 - Click on STUDENTS menu on the left to see the list of STUDENTS.

Step 8 - To LOGOUT from the Dashboard/System click on LOGOUT button appears on the left side.

Step 9 - Stop.

## **Technologies and tools Used :**

• Servlet: To impliment the business logic and works a controller for the project.

• JSP: To handle the presentation view.

• MYSQL: To Create Database and its objects.

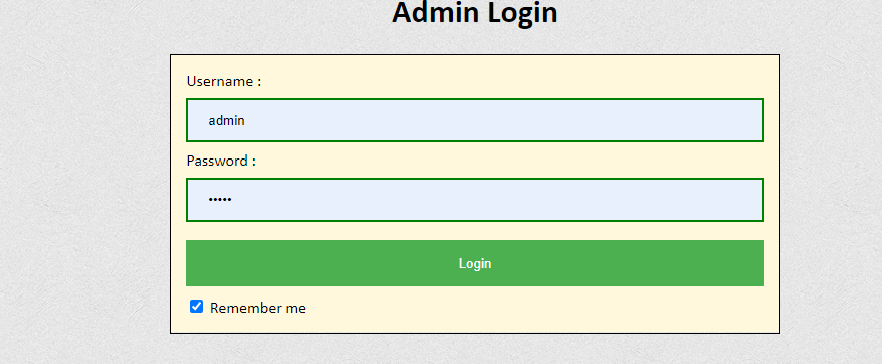
• JDBC: To make operations on the database for the project.

• CSS: To format the contents.

• Eclipse: To write and run the code.

• Tomcat: To run and deploy servlet application.

**//Login page from where ADMIN User can Login into the application using USERNAME and PASSWORD.**

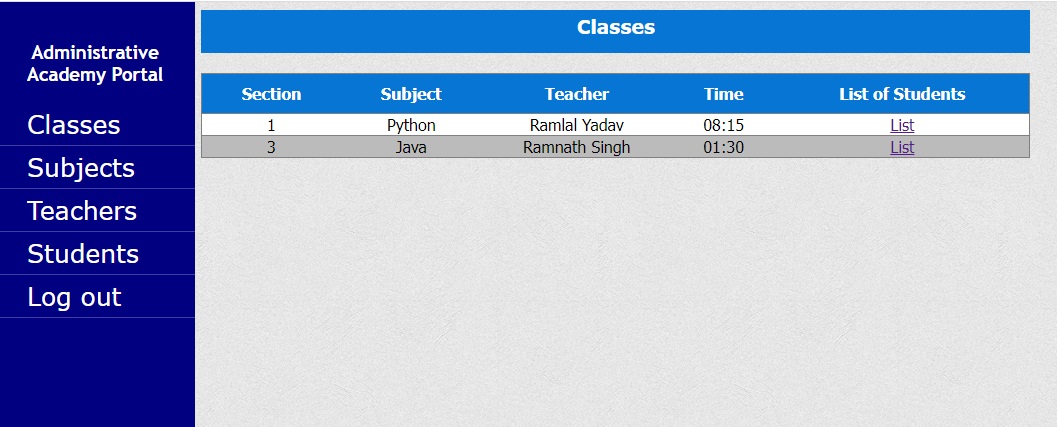
****

**// Dashboard page is as under.**

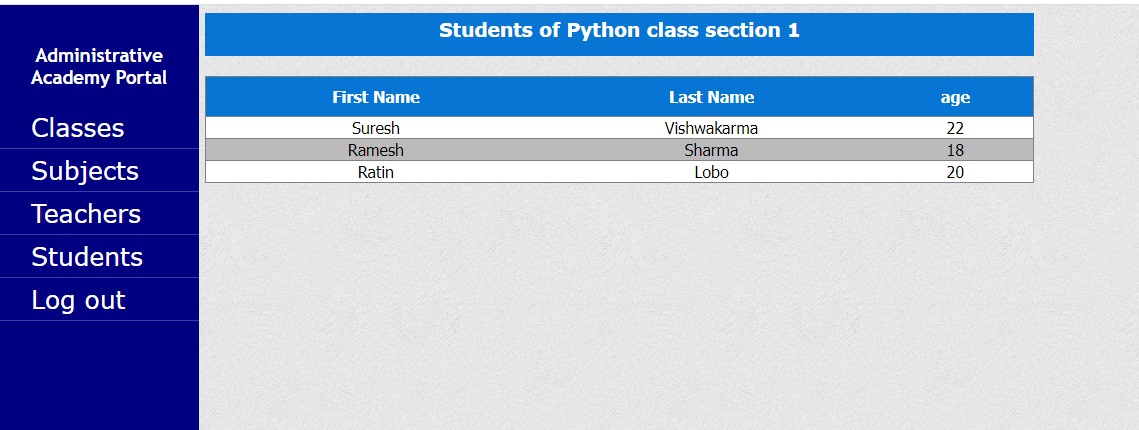
**Click on CLASSES menu on the left to see the list of classes.**

**To see the list of students in a class click on LIST button mentioned on**

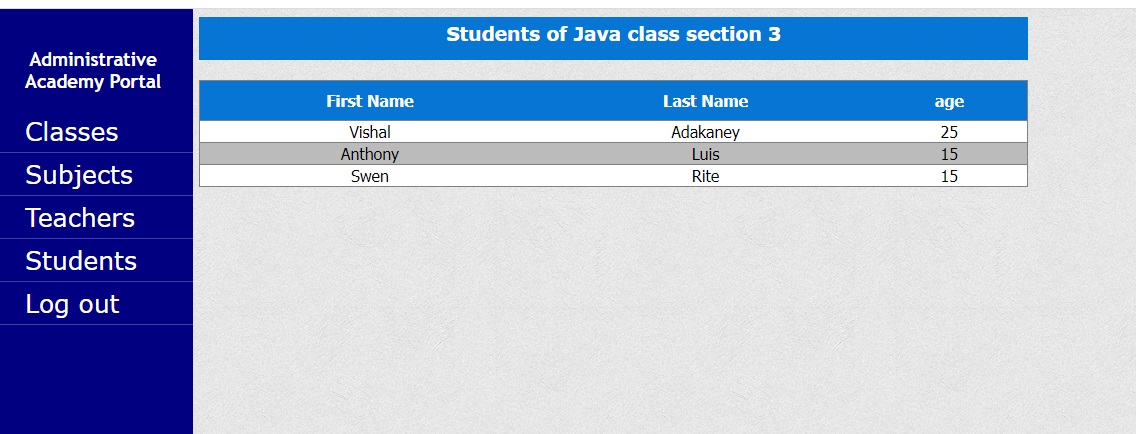
**the right side..**



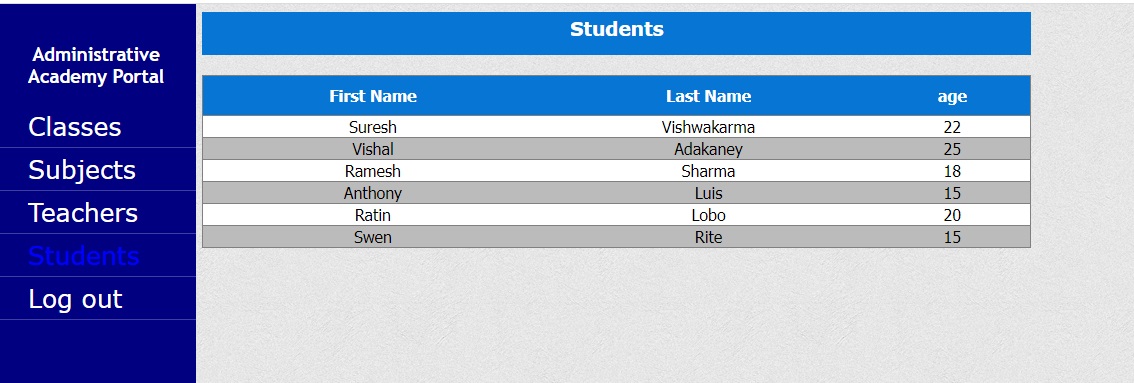
**// List of Students in Section 1.**

****

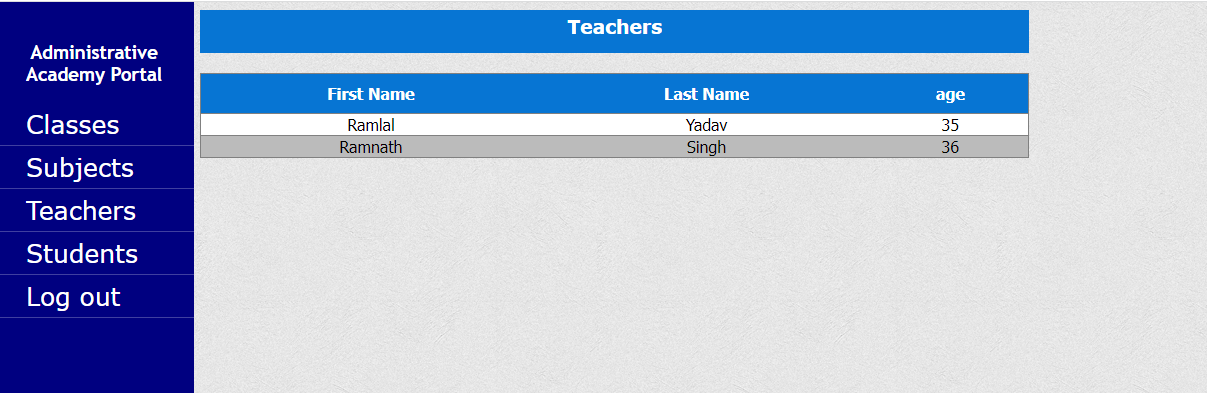
**// List of Students in Section 3.**



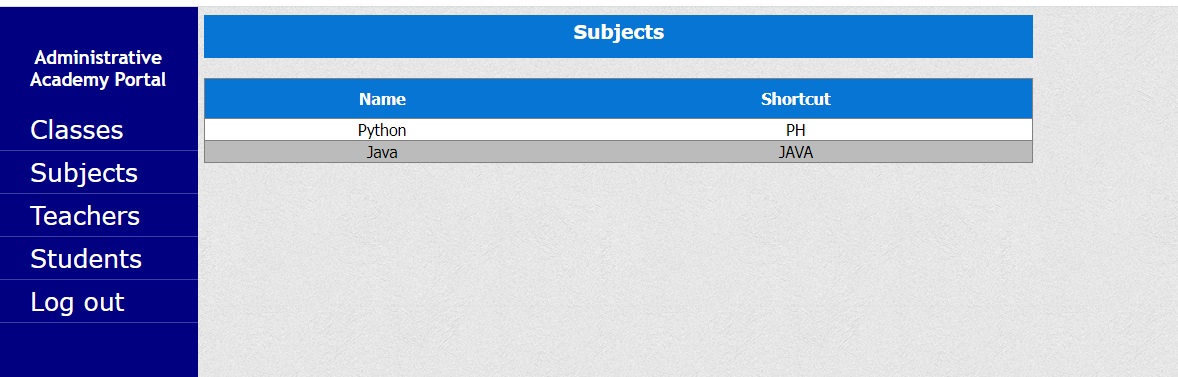
**// List of All Students.**



**// List of All Teachers.**



**// List of All Subjects.**



## Coreconceptsused:

Thefollowingconceptsareusedintheproject,

• Object-Oriented: used to create and model objects for users and their credentials.

• Databases: used to store and retrieve data.

• Data Sources: used to define a set of properties required to identify and access the database.

• Collections: used some collections such array list to store collection of data.

• Exception Handling: used to catch problems that arises in the code especially in I/O blocks.

• Cookies: to store log-in data on the client browser.

## How to run the program :

• clone project

• Import the “database\administrative-portal.sql” file to your database administration tool.

• Go to “\src\main\webapp\META-INF\context.xml” file and open it.

• Edit the database’ properties such as username, password and driverClassName to be suit to your database administration tool.

• Now run program on a server.

• To login you must enter admin for both username and password.

## LinktoGithub:

**https://github.com/hanumanprasadvishwa/Project-2.git**

## 

## Conclusion:

Applicationispreparedaccordingtorequiredfeatures.Also,consideringtediousnessforsomeoftheaskedfeaturestheyaresimplifiedinordertoincreasetheproductivity.