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

This document contains sections for:

* [Sprint planning and Task completion](#Sprint_plan)
* [Core concepts used in project](#Core_concepts)
* [Flow of the Application](#Flow)
* Algorithm and product backlog
* Technologies and tools used
* [Demonstrating the product capabilities.](#Product_capability)
* [Conclusions](#Conclusions)

The code for this project is hosted at ------------------

The project is developed by Mahesh Koutam .

## Sprints planning and Task completion

The project is planned to be completed in 2 sprints.

Tasks assumed to be completed in the 1st 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.
* Writing the jsp programs as per requirements.
* Initializing the database(Mysql)

Tasks assumed to be completed in the 1st sprint are:

* Writing the Java servelts program to fulfill the requirements of the project.
* Testing the Java program with database.
* Pushing code to GitHub.
* Creating this specification document highlighting application capabilities, appearance, and user interactions.

## Core concepts used in project

• 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.

## Flow of the Application

Diagram

Description automatically generated

Algorithm and Product Backlog:

1. Create database and tables.
2. Connect the database to the project.
3. Create models classes.
4. Create a database utility class to retrieve data.
5. Create login page.
6. Create JSP files for all pages of the project.
7. Create a servlet to get requests and send responses to the JSP files.
8. Add cookies.
9. Create a CSS file to format the contents.
10. Debug and Test the project.

Technologies and tools Used

• Servlet: to do the business logic and works a controller for the project.

• JSP: to handle the presentation view.

• SQL: to create and manage the database.

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

• CSS: to format the contents.

• phpMyAdmin: to administrate and manage the database manually.

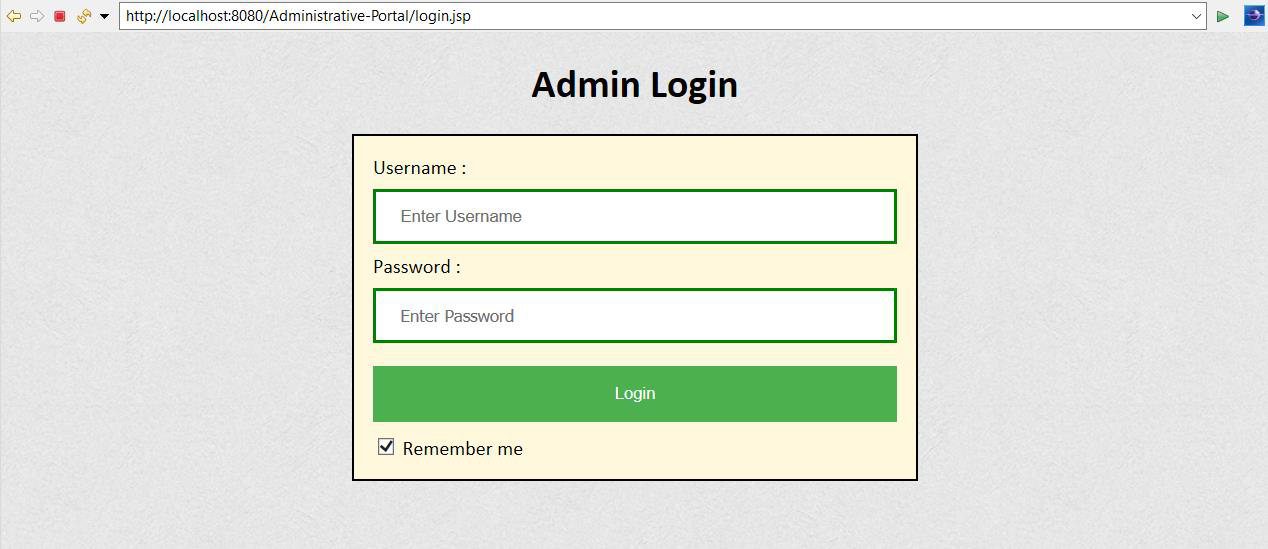
• Eclipse: to write and run the code.

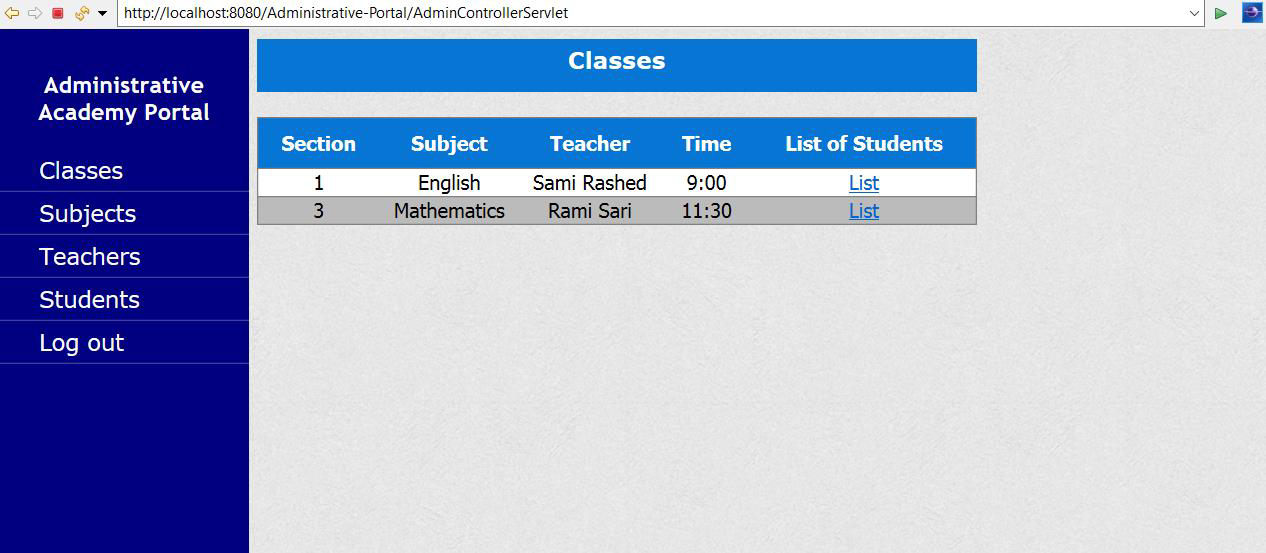
• Tomcat: to run and deploy servlet application.

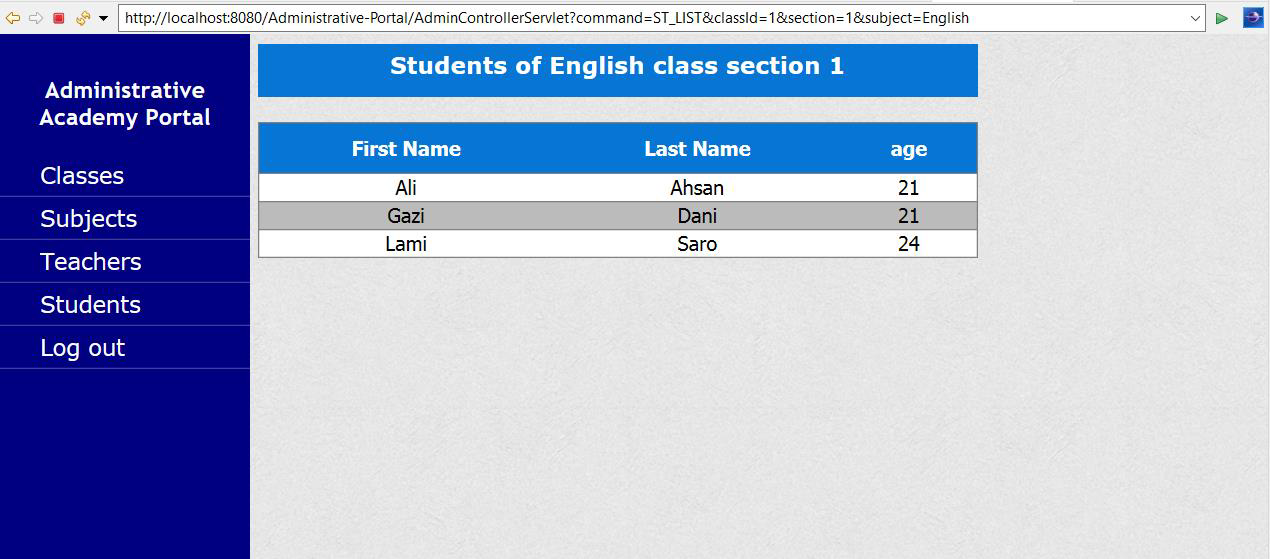
## 

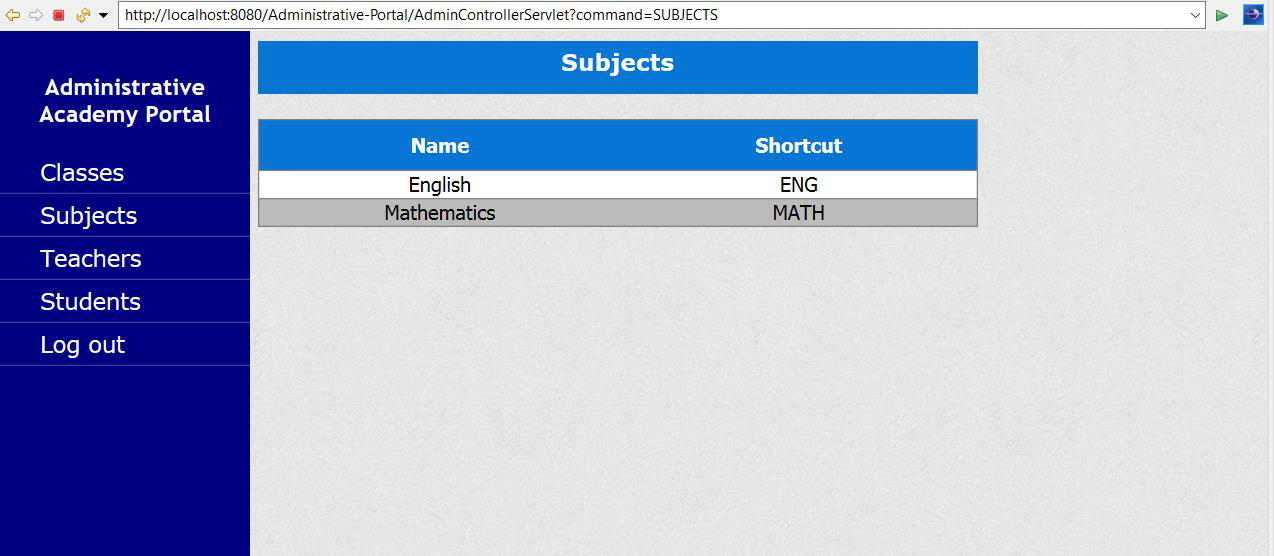
## Demonstrating the product capabilities.

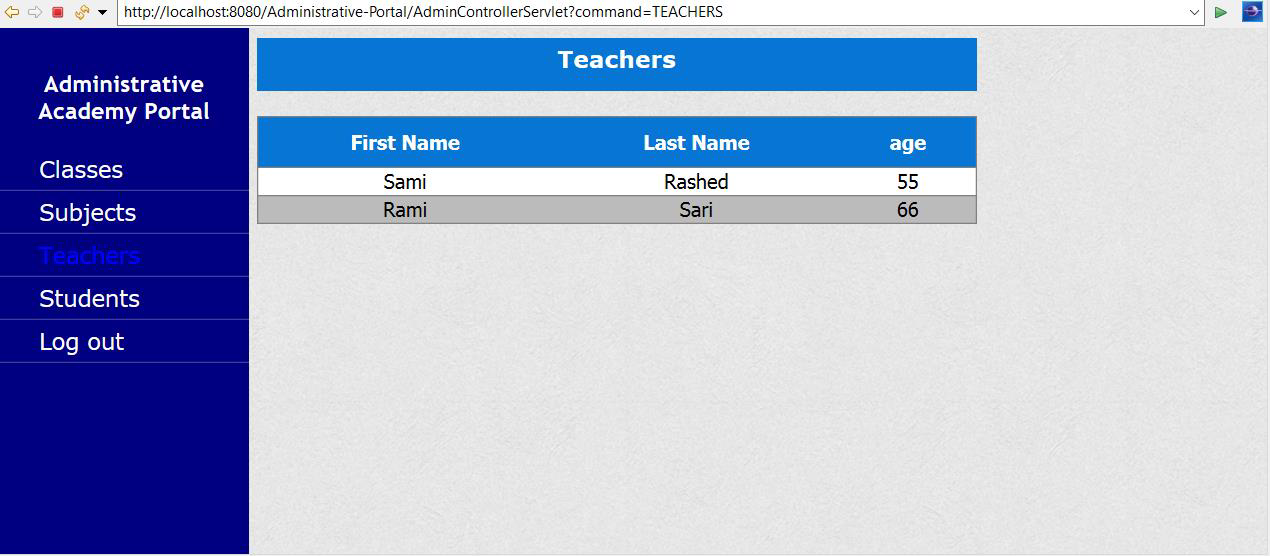
**Admin Login:**

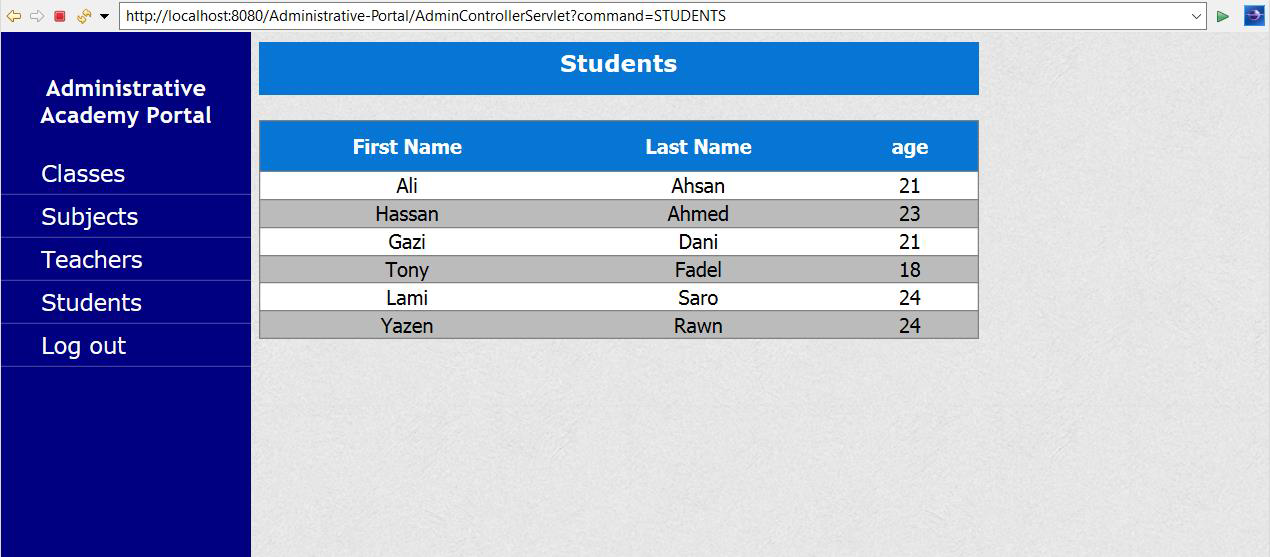
[](https://user-images.githubusercontent.com/64940728/120771774-47dd9600-c528-11eb-86c8-ee8a1b133a23.png)

[](https://user-images.githubusercontent.com/64940728/120771806-4e6c0d80-c528-11eb-97bb-8abe14d8560c.png)

[](https://user-images.githubusercontent.com/64940728/120771819-51ff9480-c528-11eb-98fe-39b7767b8de6.png)

[](https://user-images.githubusercontent.com/64940728/120771833-54fa8500-c528-11eb-9291-2ab6c81528f3.png)

[](https://user-images.githubusercontent.com/64940728/120771849-588e0c00-c528-11eb-89d0-fbc960d1562e.png)

[](https://user-images.githubusercontent.com/64940728/120771867-5c219300-c528-11eb-8aed-e8d137640817.png)

**XML FILE :**

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?> |
|  | <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd" id="WebApp\_ID" version="3.1"> |
|  | <display-name>Administrative-Portal</display-name> |
|  |  |
|  | <welcome-file-list> |
|  | <welcome-file>AdminControllerServlet</welcome-file> |
|  | <welcome-file>index.jsp</welcome-file> |
|  | <welcome-file>index.html</welcome-file> |
|  | </welcome-file-list> |
|  | </web-app> |

**MODELS FOR THE PROJECT**

**CLASS**

public Class(int id, int section, String teacher, String subject, String time) {

super();

this.id = id;

this.section = section;

this.teacher = teacher;

this.subject = subject;

this.time = time;

}

**STUDENT**

public Student(int id, String fname, String lname, int age, int aclass) {

super();

this.id = id;

this.fname = fname;

this.lname = lname;

this.age = age;

this.aclass = aclass;

}

**SUBJECT**

public Subject(int id, String name, String shortcut ) {

super();

this.id = id;

this.name = name;

this.shortcut = shortcut;

}

**TEACHER**

public Teacher(int id, String fname, String lname, int age) {

super();

this.id = id;

this.fname = fname;

this.lname = lname;

this.age = age;

}

**CREATING ADMINCONTROLLER**

@WebServlet("/AdminControllerServlet")

public class AdminControllerServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

private DbRetrieve dbRetrieve;

@Resource(name = "jdbc\_database")

private DataSource datasource;

@Override

public void init() throws ServletException {

super.init();

// create instance of db util, to pass in conn pool object

try {

dbRetrieve = new DbRetrieve(datasource);

} catch (Exception e) {

throw new ServletException(e);

}

}

## 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**