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

This document contains sections for:

* Sprint planning and Task completion
* Core concepts used in the project
* Flow of the Application.
* Demonstrating the product capabilities, appearance, and user interactions.
* Unique Selling Points of the Application
* Conclusions

The code for this project is hosted at

The project is developed by Vishnu vardhan.

## Sprints planning and Task completion

The project is planned to be completed in 1 sprint. 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.
* Creating this specification document highlighting application capabilities, appearance, and user interactions.

## The flow of the Application



## 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:

## Project' Details

This project aims to design and develop a backend administrative portal for the Learner’s Academy using Java EE technologies. I developed it as a project of phase 2 for the Become a back-end expert course. The goal of this project is to apply servlet, jsp and JDBC concepts.

## 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.

## Flowcharts of The Application

[](https://user-images.githubusercontent.com/64940728/120771636-182e8e00-c528-11eb-92bb-f5856138c93f.png)

## Core concepts used in the 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 arraylist 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\database.sql” file to your database administration tool.

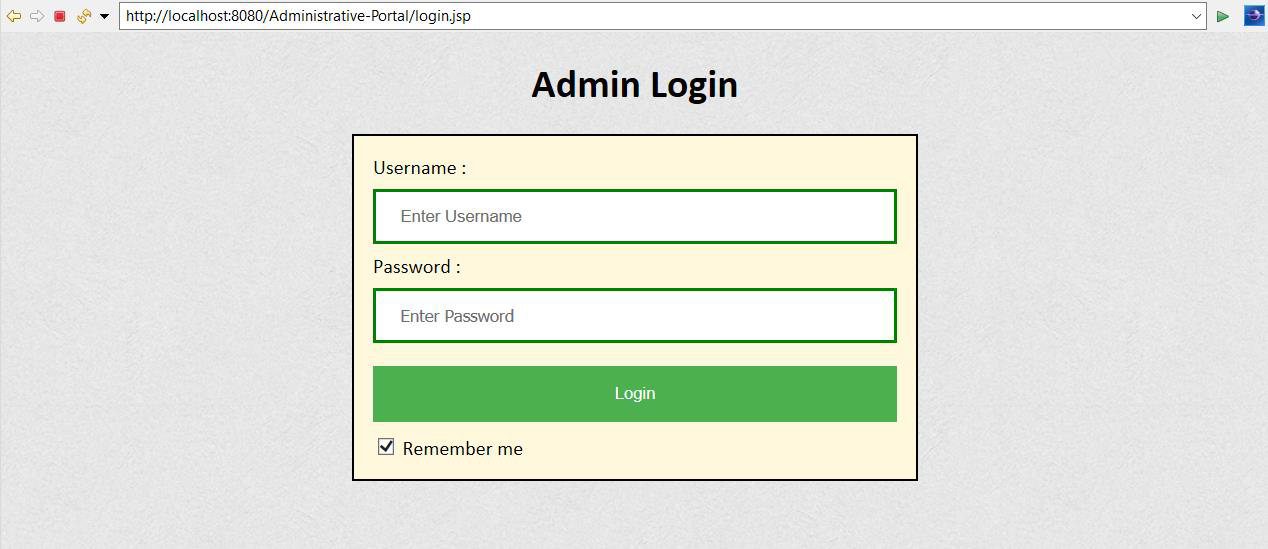
• Go to “\src\main\webapp\META-INF\context.xml” file and open it.

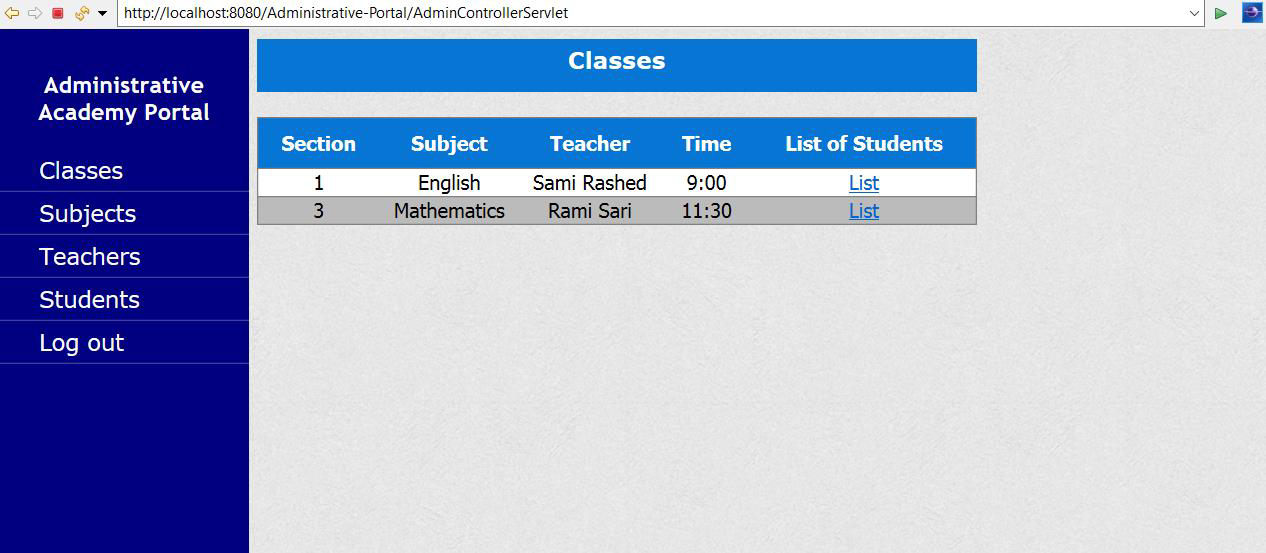
• Edit the database’ properties such as username, password and driverClassName to be suit to your database administration tool.

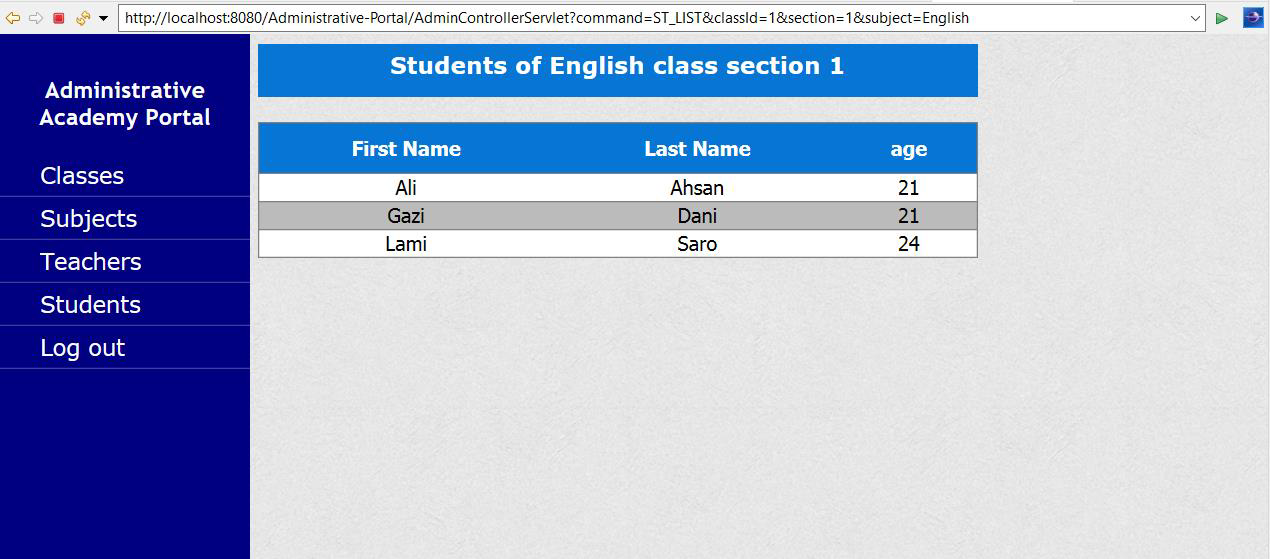
• Now run program on a server.

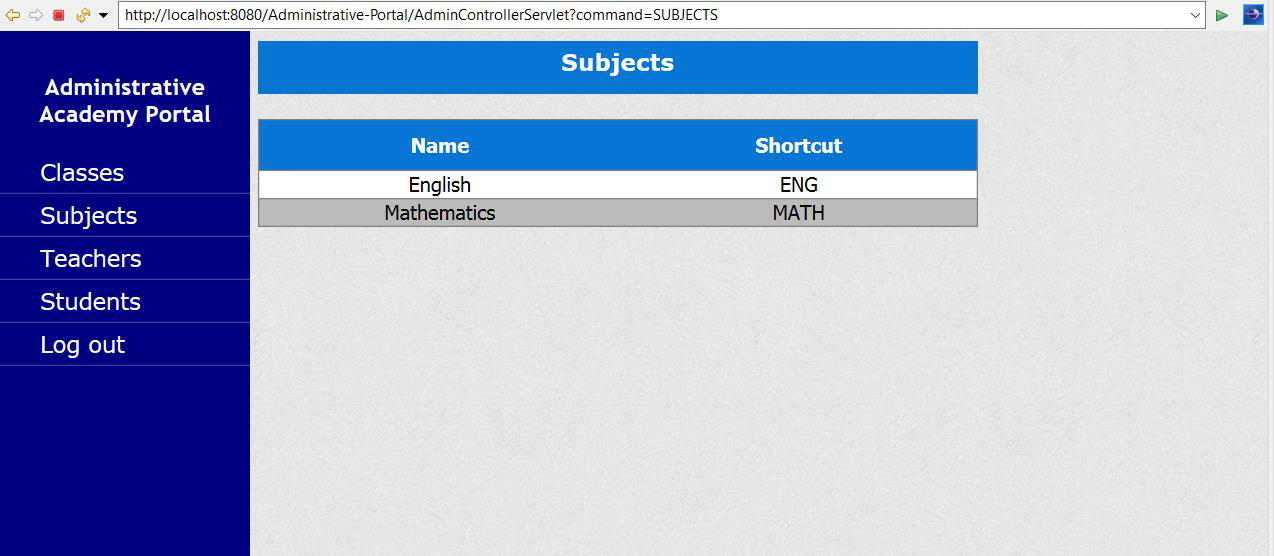
• To login you must enter admin for both username and password.

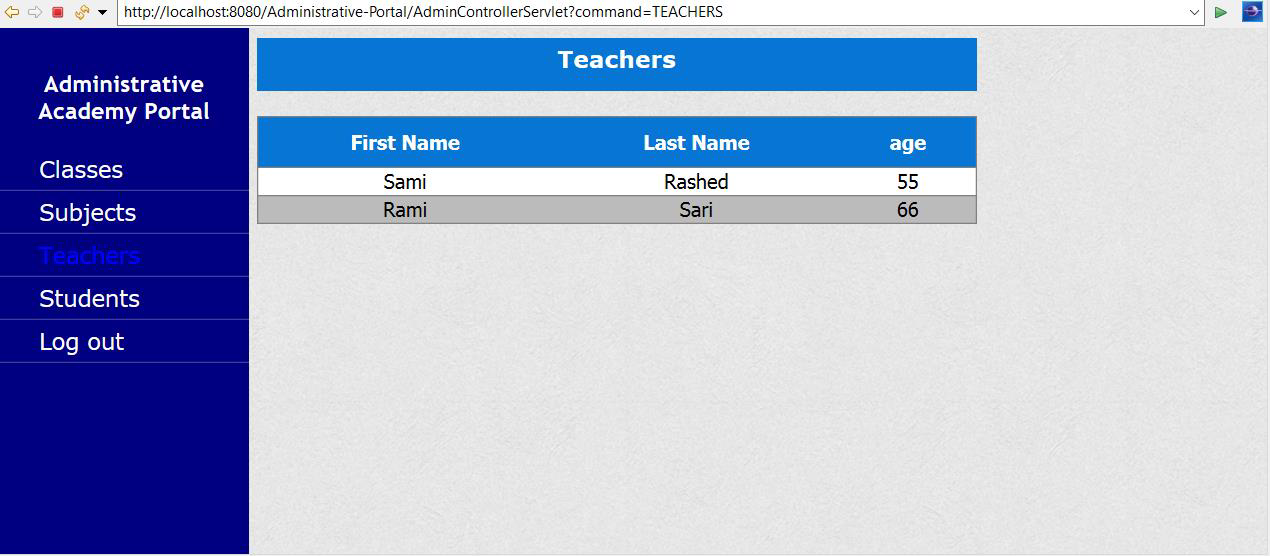
Screenshots:

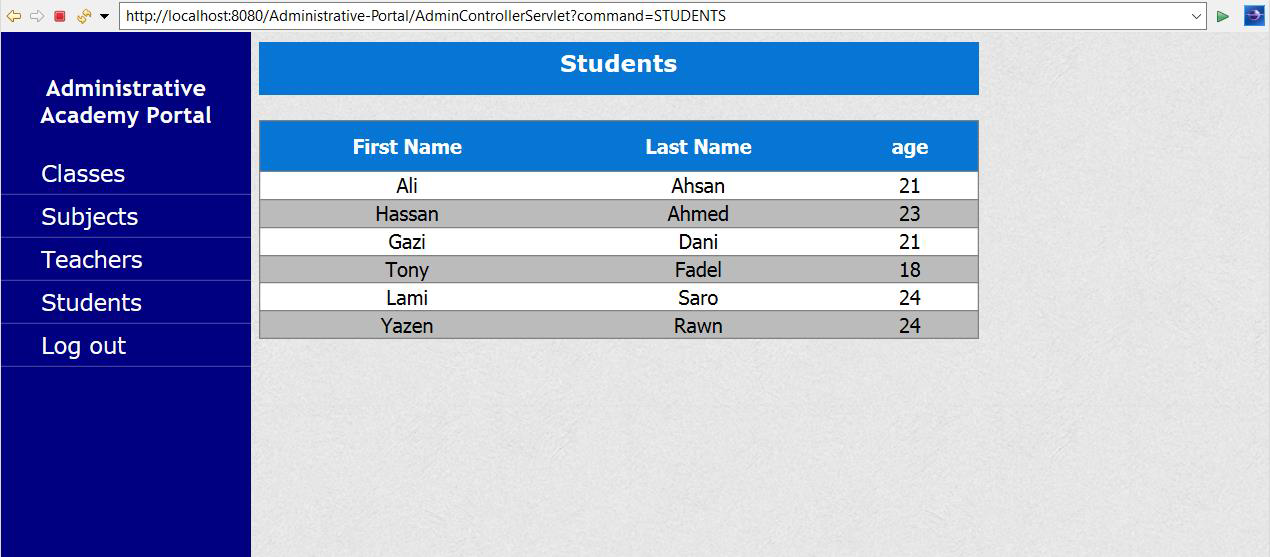
[](https://user-images.githubusercontent.com/64940728/120771774-47dd9600-c528-11eb-86c8-ee8a1b133a23.png)

[](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)

## Pushing the code to the GitHub repository

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

**cd <folder path>**

* Initialize the repository using the following command:

**git init**

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

**gitadd .**

* 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**