Name: Duggasani Naga Surekha

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.

This document contains sections for:

The code for this project is hosted at
https://github.com/surekhaitgithub/Newcodingboard.git

The project is developed by Duggasani Naga Surekha

Sprints planning and Task completion:

The project is planned to be completed in 2 sprint. Tasks assumed to be completed in the sprints 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.

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 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.

Technologies 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.

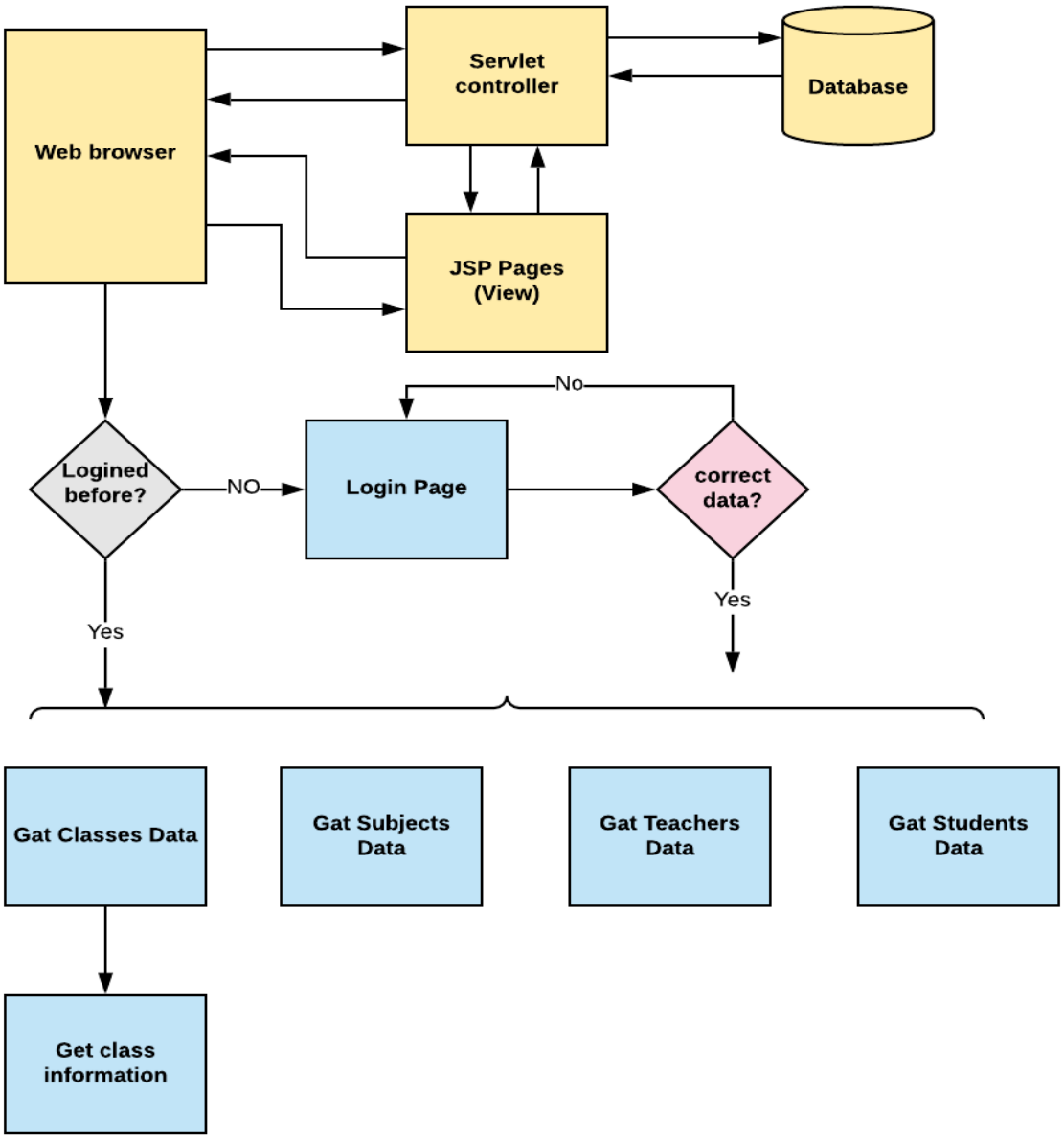
• CSS: to format the contents.

• phpMyAdmin: to administrate and manage the database manually.

• Eclipse: to write and run the code.

• Tomcat: to run and deploy servlet application.

Flow of the Application:



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

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

Student.java

Subject.java

Teacher.java

Jsp files

class-students.jsp

classes-list.jsp

left-list.jsp

list-students.jsp

login.jsp

subjects-list.jsp

- 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

Student.java

Subject.java

Teacher.java

Jsp files

class-students.jsp

classes-list.jsp

left-list.jsp

list-students.jsp

login.jsp

subjects-list.jsp

teachers-list.jsp

CSS files

add-student-style.css

login.css

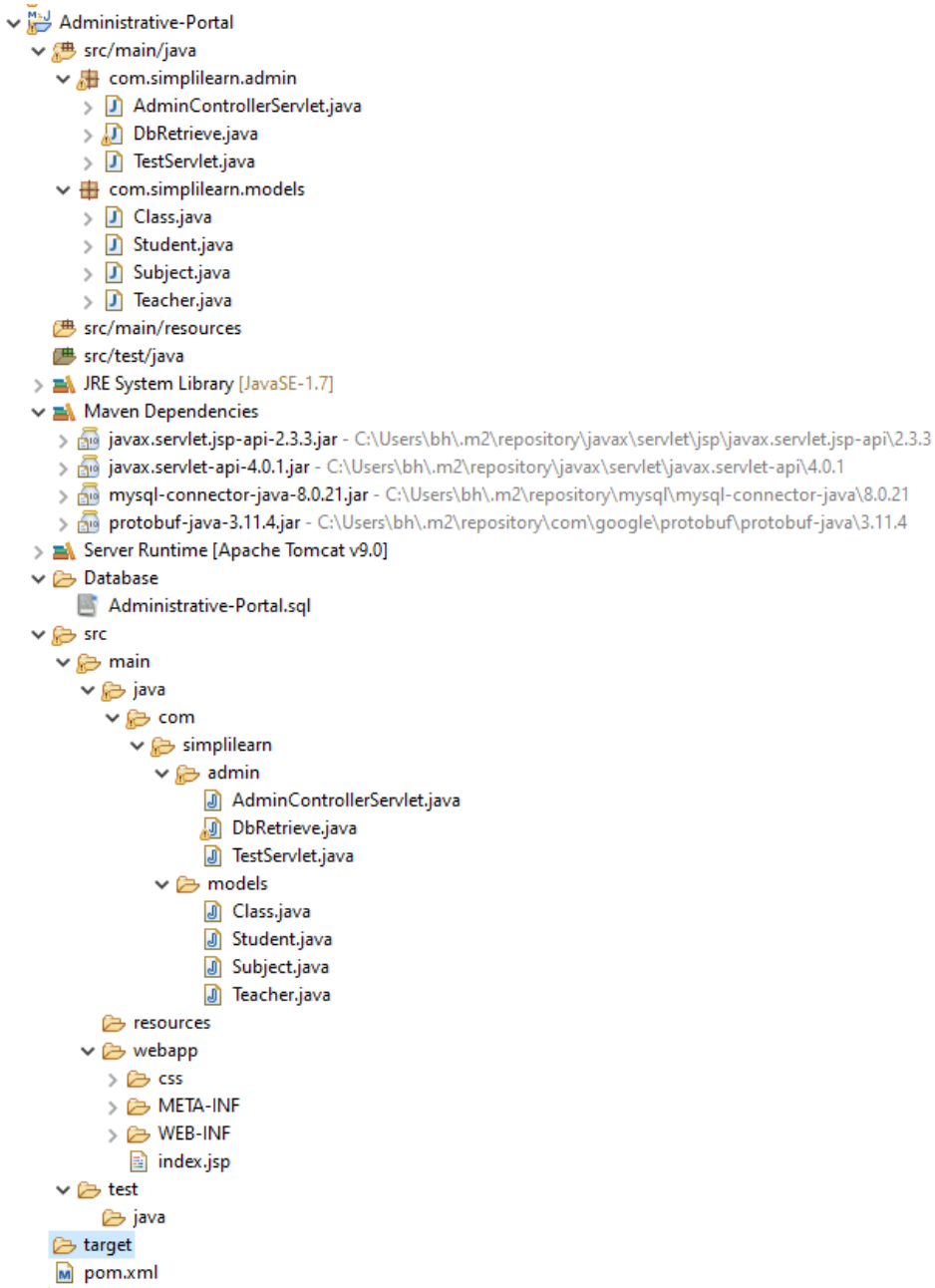
style.css

administrative-portal.sql

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

Step 3:

1.**Creating the project in eclipse.**



1 **Import the “database\database.sql” file to your database administration tool.**

-- Host: 127.0.0.1:3307

-- Generation Time: April 07, 2022 at 04:00PM

-- Server version: 10.4.18-MariaDB

-- PHP Version: 8.0.3

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";

```
START TRANSACTION;

SET time_zone = "+00:00";



/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;

/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;

/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;

/*!40101 SET NAMES utf8mb4 */;



--

-- Database: `administrative-portal`

--



-- --------------------------------------------------------



--

-- Table structure for table `classes`

--



CREATE TABLE `classes` (

  `id` int(11) NOT NULL,

  `section` int(55) NOT NULL,

  `teacher` int(11) NOT NULL,

  `subject` int(11) NOT NULL,
```

```
  `time` varchar(44) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `classes`
--

INSERT INTO `classes` (`id`, `section`, `teacher`, `subject`, `time`) VALUES
(4, 2, 4, 2, '10:25'),
(3, 2, 2, 4, '8:45');
drop table classes;



-- --------------------------------------------------------

--
-- Table structure for table `students`
--

CREATE TABLE `students` (
  `id` int(11) NOT NULL,
  `fname` varchar(55) NOT NULL,
  `lname` varchar(55) NOT NULL,
  `age` int(11) DEFAULT NULL,
```

```
  `class` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;



--
-- Dumping data for table `students`
--

INSERT INTO `students` (`id`, `fname`, `lname`, `age`, `class`) VALUES
(1, 'Akshay', 'kumar', 20, 1),
(2, 'Hasin', 'singh', 19, 2),
(3, 'Gnanavi', 'reddy', 18, 1),
(4, 'Tony', 'Rathod', 19, 2),
(5, 'Lally', 'Simha', 24, 1),
(6, 'sindhu', 'Rawn', 26, 2);



-- --------------------------------------------------------

--
-- Table structure for table `subjects`
--
```

```
CREATE TABLE `subjects` (
  `id` int(11) NOT NULL,
  `name` varchar(55) NOT NULL,
  `shortcut` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `subjects`
--

INSERT INTO `subjects` (`id`, `name`, `shortcut`) VALUES
(3, 'Biology', 'Bio'),
(4, 'English', 'Eng');
select * from subjects;
drop table subjects;

-- --------------------------------------------------------
CREATE TABLE `subjects` (
  `id` int(11) NOT NULL,
  `name` varchar(55) NOT NULL,
  `shortcut` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
```

```
-- Dumping data for table `subjects`
--

INSERT INTO `subjects` (`id`, `name`, `shortcut`) VALUES
(3, 'Biology', 'Bio'),
(4, 'English', 'Eng');
select * from subjects;
drop table subjects;

-- --------------------------------------------------------

--
-- Table structure for table `teachers`
--

CREATE TABLE `teachers` (
  `id` int(11) NOT NULL,
  `fname` varchar(55) NOT NULL,
  `lname` varchar(55) NOT NULL,
  `age` varchar(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `teachers`
--
```

```
INSERT INTO `teachers` (`id`, `fname`, `lname`, `age`) VALUES
(3, 'dwarak', 'pridhvi', '78'),
(4, 'sirigala', 'Sarika', '64');
-- drop table teachers;
select *from teachers;


--
-- Indexes for dumped tables
--


--
-- Indexes for table `classes`
--
ALTER TABLE `classes`
  ADD PRIMARY KEY (`id`),
  ADD KEY `subject_id` (`subject`),
  ADD KEY `teacher_id` (`teacher`);
(3, 'dwarak', 'pridhvi', '78'),
(4, 'sirigala', 'Sarika', '64');
-- drop table teachers;
select *from teachers;


--
-- Indexes for dumped tables
```

```
--


--
-- Indexes for table `classes`
--
ALTER TABLE `classes`
  ADD PRIMARY KEY (`id`),
  ADD KEY `subject_id` (`subject`),
  ADD KEY `teacher_id` (`teacher`);


--
-- Indexes for table `students`
--
ALTER TABLE `students`
  ADD PRIMARY KEY (`id`),
  ADD KEY `class_id` (`class`);


--
-- Indexes for table `subjects`
--
ALTER TABLE `subjects`
  ADD PRIMARY KEY (`id`);


--
-- Indexes for table `teachers`
```

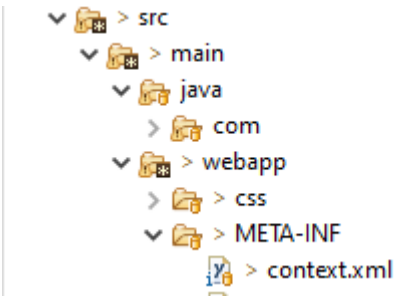
```
--

ALTER TABLE `teachers`

  ADD PRIMARY KEY (`id`);



--

-- AUTO_INCREMENT for dumped tables

--



--

-- AUTO_INCREMENT for table `classes`

--

ALTER TABLE `classes`

  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=3;



--

-- AUTO_INCREMENT for table `students`

--

  ADD PRIMARY KEY (`id`);



--

-- AUTO_INCREMENT for dumped tables

--



--

-- AUTO_INCREMENT for table `classes`
```

```
--

ALTER TABLE `classes`

  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=3;



--

-- AUTO_INCREMENT for table `students`

--

ALTER TABLE `students`

  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=8;



--

-- AUTO_INCREMENT for table `subjects`

--

ALTER TABLE `subjects`

  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=3;



--

-- AUTO_INCREMENT for table `teachers`

--

ALTER TABLE `teachers`

  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=3;



--
```

```
-- Constraints for dumped tables

--



--

-- Constraints for table `classes`

--

ALTER TABLE `classes`

  ADD CONSTRAINT `subject_id` FOREIGN KEY (`subject`) REFERENCES `subjects` (`id`),

  ADD CONSTRAINT `teacher_id` FOREIGN KEY (`teacher`) REFERENCES `teachers` (`id`);



--

-- Constraints for table `students`

--




-- Constraints for dumped tables

--



--

-- Constraints for table `classes`

--

ALTER TABLE `classes`

  ADD CONSTRAINT `subject_id` FOREIGN KEY (`subject`) REFERENCES `subjects` (`id`),
```

```
  ADD CONSTRAINT `teacher_id` FOREIGN KEY (`teacher`) REFERENCES
`teachers` (`id`);



--

-- Constraints for table `students`

--







/*!40101 SET
CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

/*!40101 SET
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;

/*!40101 SET
COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

3. Go to "\src\main\webapp\META-INF\context.xml" file and open it.

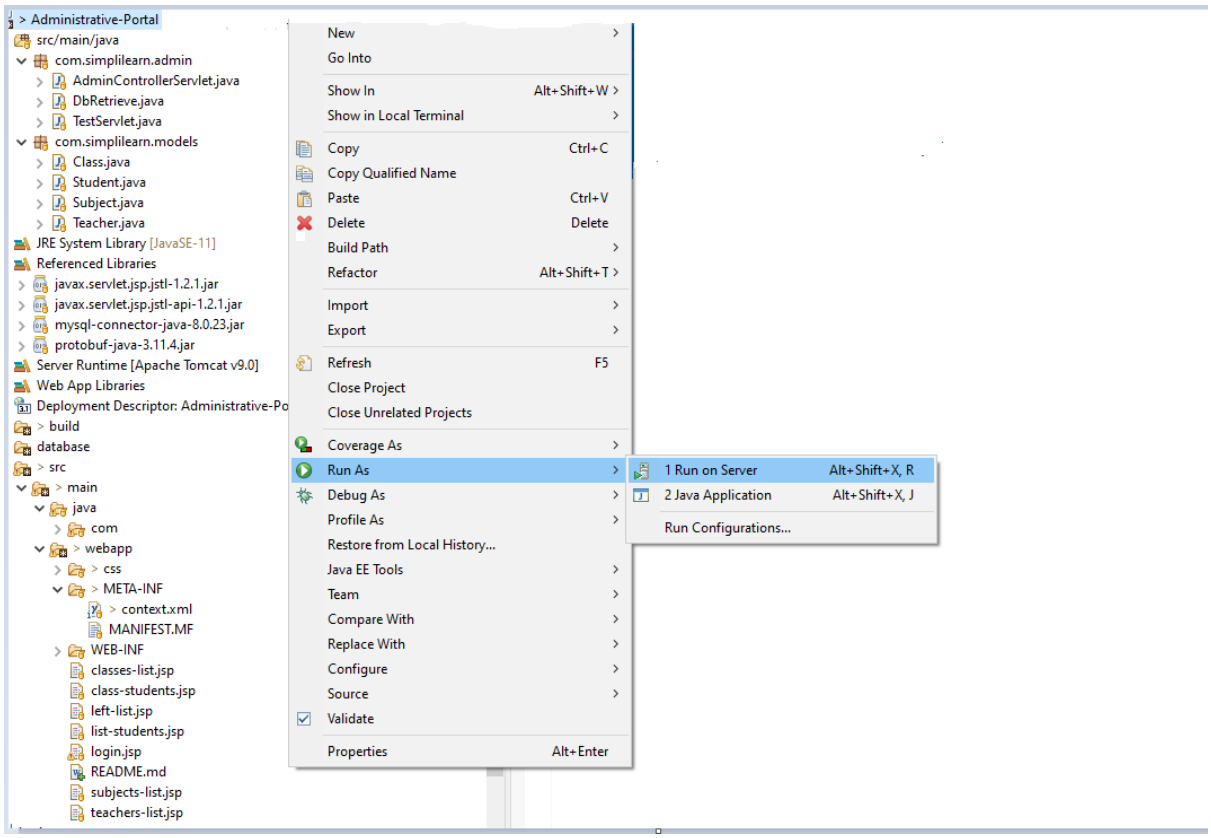


4. Edit the database' properties such as username, password and driverClassName to be suit to your database administration tool.

```
<Context>
```

```
<Resource name="jdbc_database"
                    auth="Container" type="javax.sql.DataSource"
          maxActive="20" maxIdle="5" maxWait="10000"
          username="root" password="Surekha@456"
driverClassName="com.mysql.cj.jdbc.Driver"
          url="jdbc:mysql://localhost:3306/admindb"/>

</Context>
```

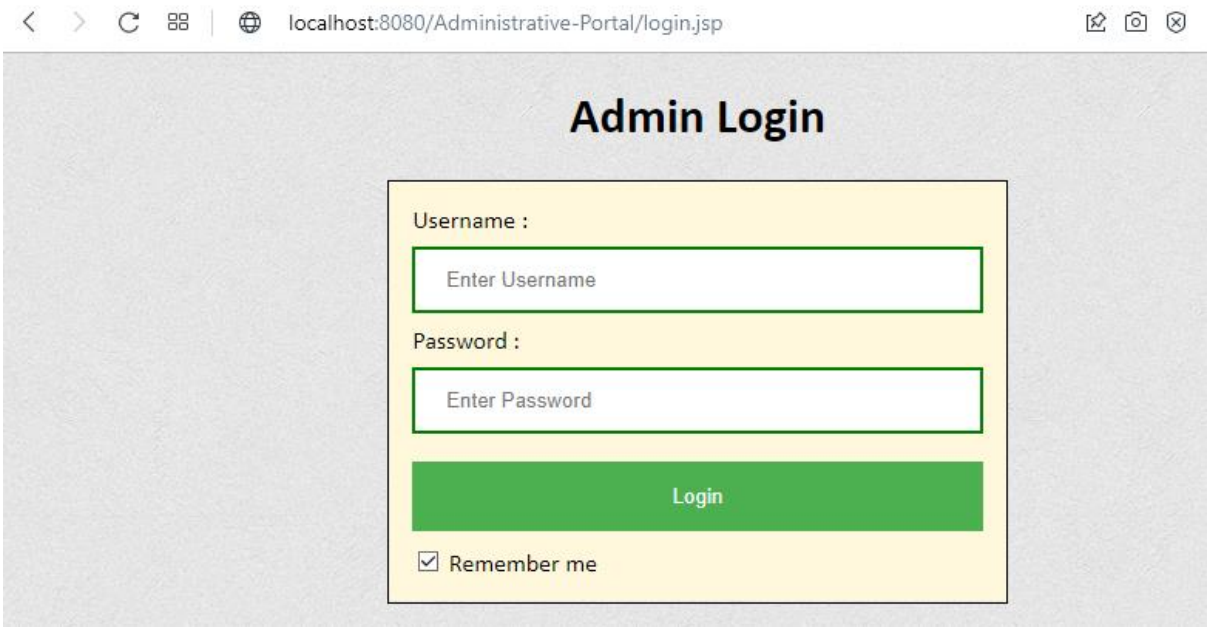
5.Now run program on a server.



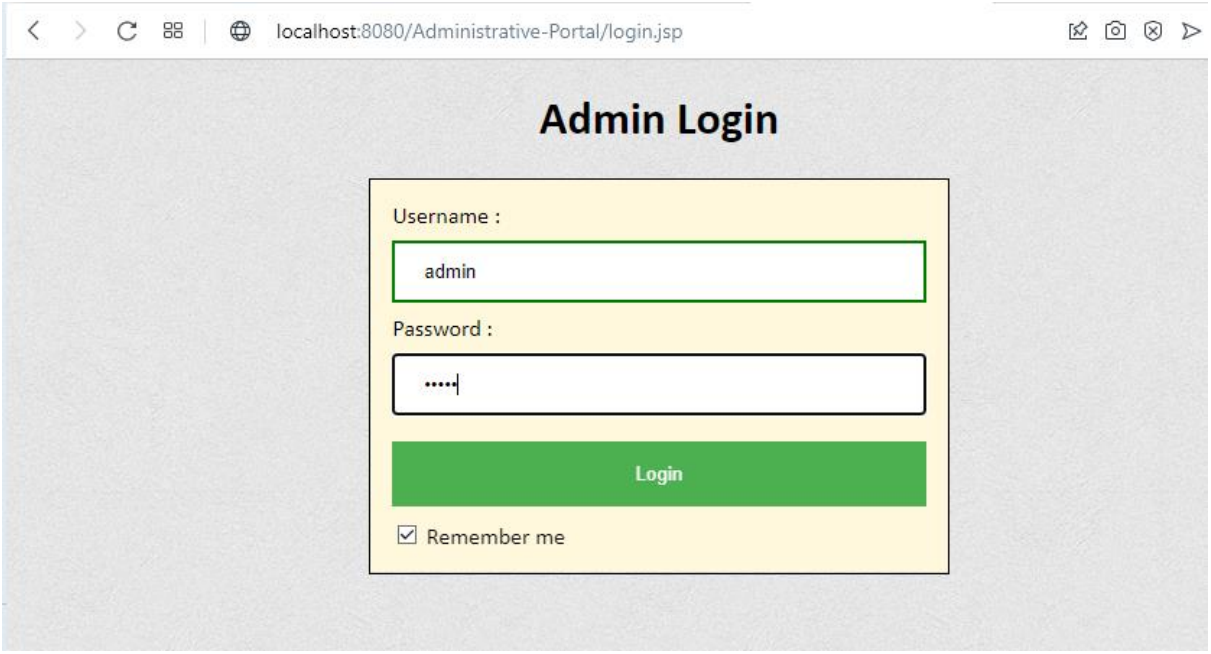
6. To login you must enter admin for both username and password.

Screenshots:

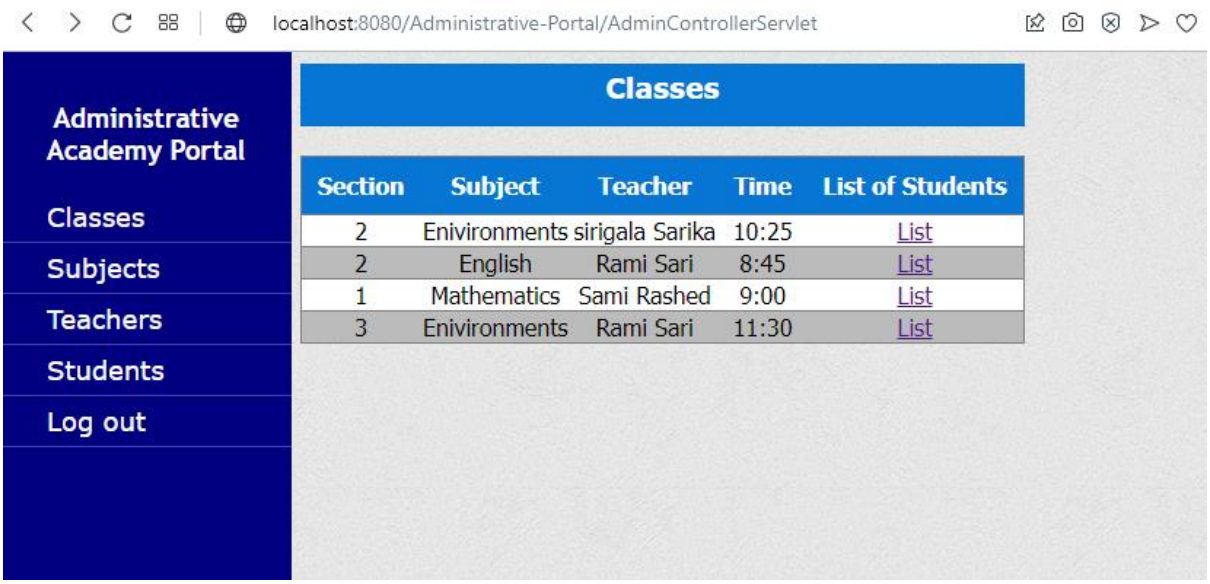
1.Login page:



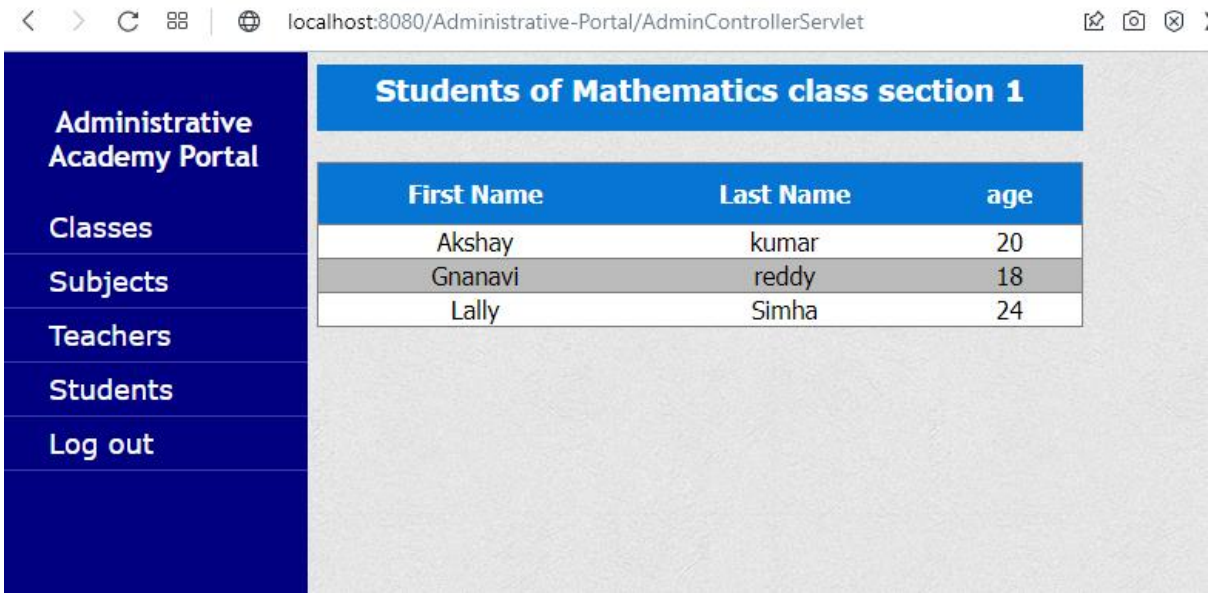
2. Enter the username and password :



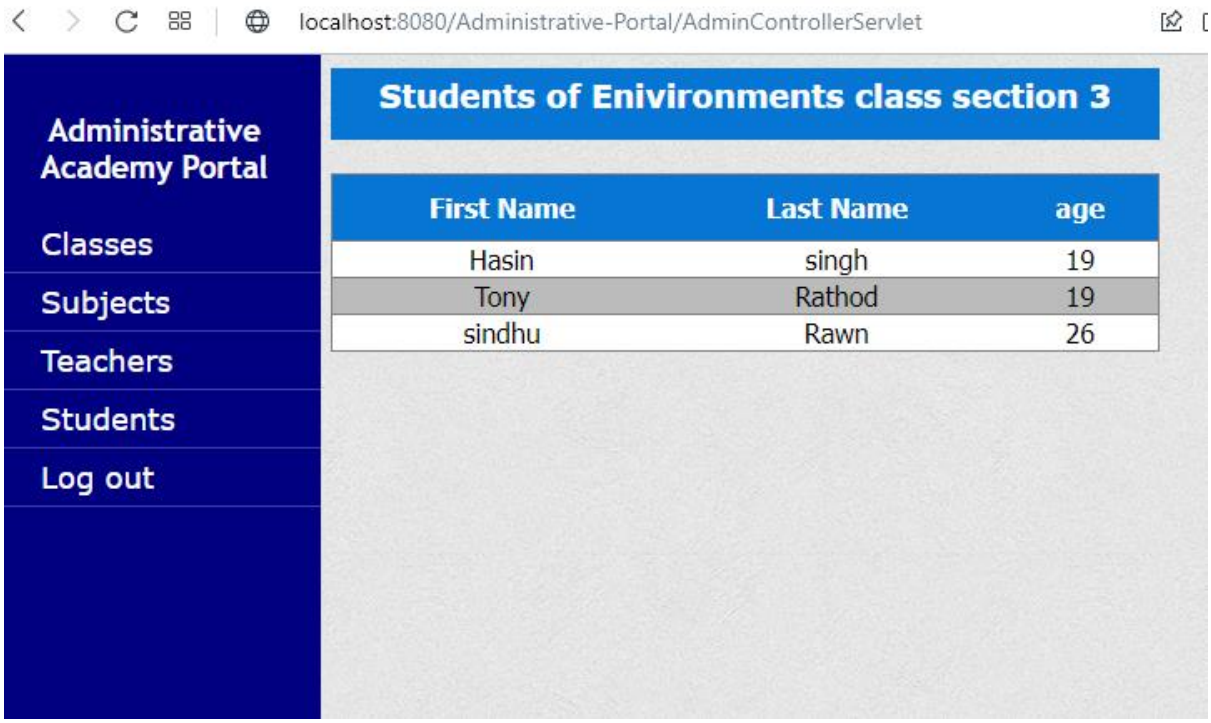
3. Classes:



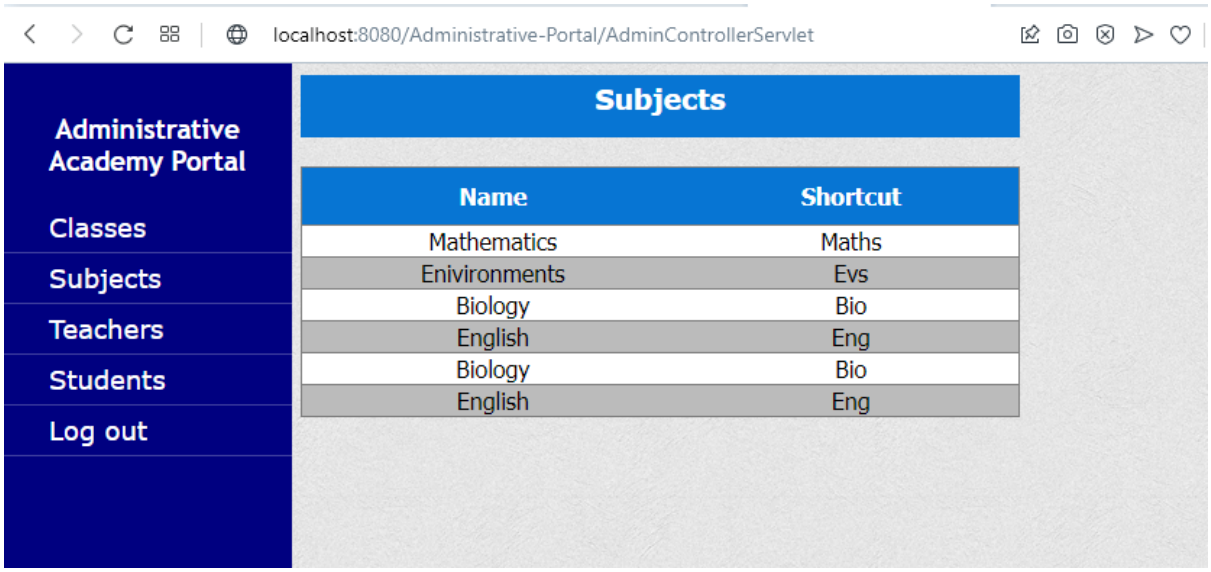
3.1. List of students in mathematics classes section1:



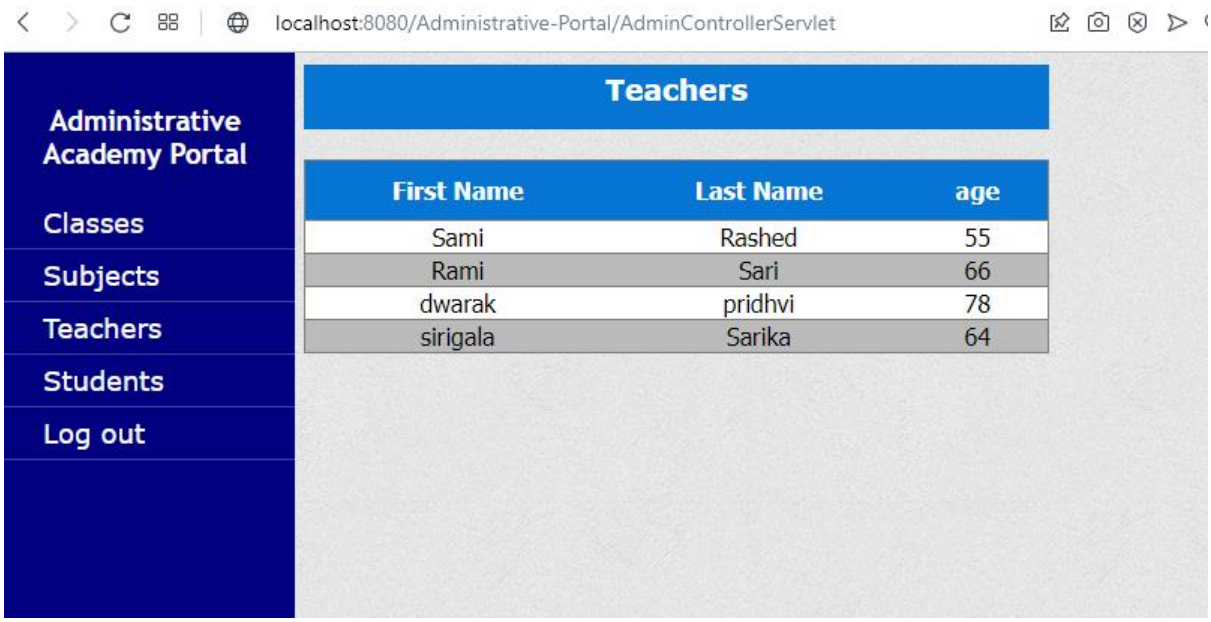
3.2. List of students in environment classes section3:



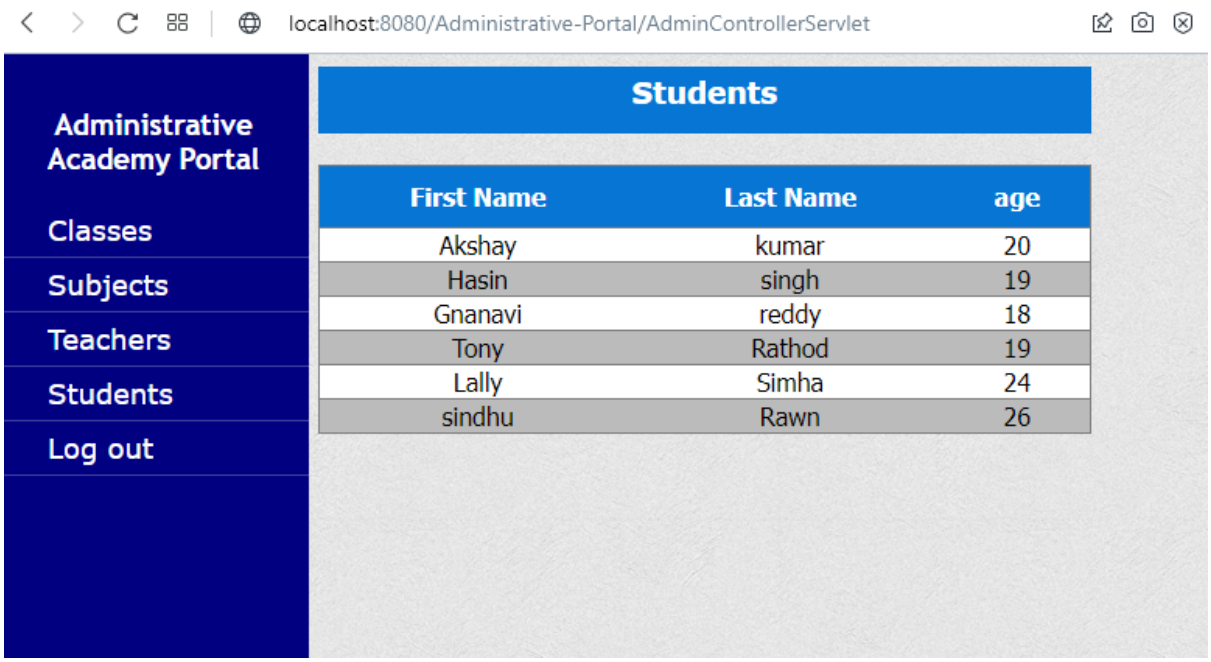
4. Subjects:



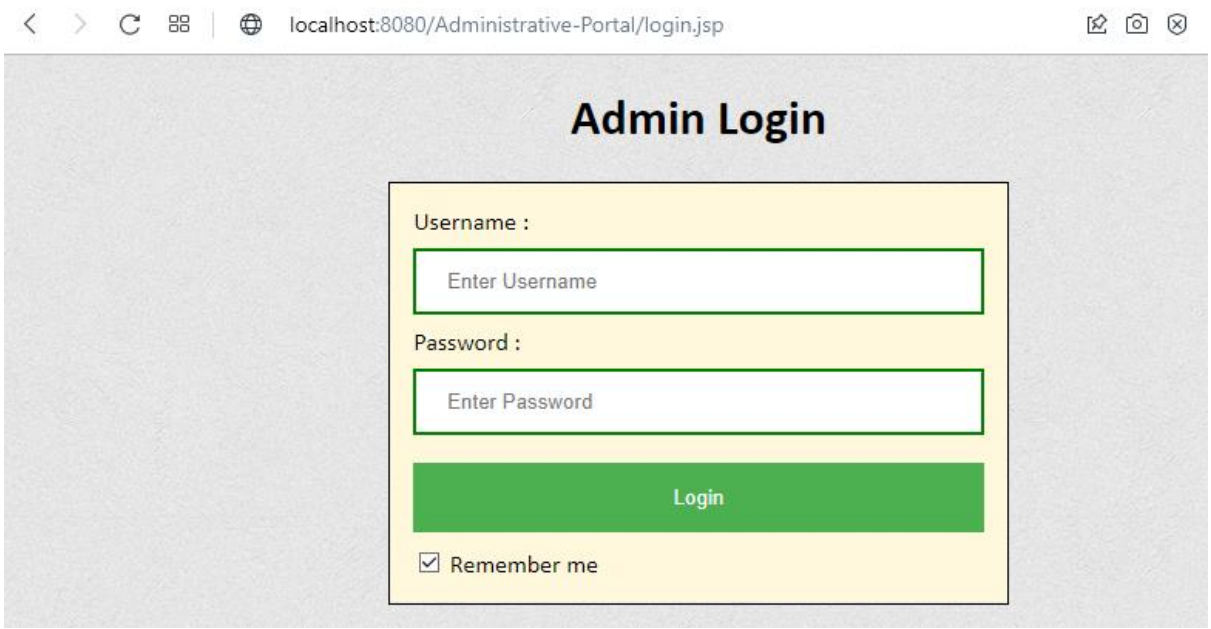
5.Teachers:



6.Students:



7. Logout page:



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.