Learner's Academy

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 admin login details are: username is *admin* and password is *admin*.

This document contains sections for:

- Sprint planning and Task completion
- Core concepts used in project
- Conclusions

The code for this project is available at ServletDemo/LearnerAcadDemo at main · pragathihebbarkm/ServletDemo (github.com) this project is developed by Pragathi Hebbar.

Sprints planning and Task completion

The project is planned to be completed in a single sprint. Tasks that are assumed to be completed in this sprint are :

- Creating the flow of the application
- Initializing git repository to track changes as development progresses.
- Writing the program to fulfil the requirements of the project.
- Pushing code to GitHub.
- Creating this specification document highlighting application capabilities, appearance, and user interactions.

Core concepts used in project

The application is based on JSP, Java, Servlets and MySql database.

Demonstrating the product capabilities, appearance, and user interactions

We have 7 operations available:

1. Admin Login
2. List of Classes
3. List of Students taking a particular subject
4. List of Teachers
5. List of Students
6. List of Subjects
7. Admin Logout

Flow of Application

1. For Login/Logout a simple JSP and Servlet are used (login.jsp), that handles the operation of Login/Logout.
2. The MYSQL connection is achieved through DbRetrieve.java class which establishes the connection between database and the eclipse editor for performing all the operations on database.
3. Servlet for all admin operations like Listing Students, Classes, Subjects and Teachers is created in AdminControllerServlet.
4. The Dashboard Page is created using left-list.jsp which holds the links to display the details of Teachers ,Students, Subjects and Classes.
5. The details of Class, Students, Teachers and Subjects are displayed with the help of jsp files (classes-list.jsp, list-students.jsp, subjects-list.jsp and teachers-list.jsp) created which retrieves the details from database.

6. The class report is displayed by clicking on the List link available in the class list, this is displayed with the help of class-students.jsp file.

SQL Queries

```
--
-- Table structure for table `classes`
--

CREATE TABLE `classes` (
  `id` int NOT NULL,
  `section` int NOT NULL,
  `teacher` int NOT NULL,
  `subject` int NOT NULL,
  `time` varchar(25) NOT NULL
);

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

INSERT INTO `classes` (`id`, `section`, `teacher`, `subject`, `time`)
VALUES
(1, 1, 1, 1, '9:00'),
(2, 3, 2, 2, '11:30');
--------------------------------------------------
--
```

```
-- Table structure for table `students`
--
CREATE TABLE `students` (
  `id` int NOT NULL,
  `fname` varchar(55) NOT NULL,
  `lname` varchar(55) NOT NULL,
  `age` int DEFAULT NULL,
  `class` int NOT NULL
) ;

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

INSERT INTO `students` (`id`, `fname`, `lname`, `age`, `class`)
VALUES
(1, 'Alison', 'Cook', 21, 1),
(2, 'Tim', 'Morgan', 23, 2),
(4, 'Anne', 'Evans', 21, 1),
(5, 'Sam', 'Riddle', 18, 2),
(6, 'Will', 'Smith', 24, 1),
(7, 'Robert', 'Stone', 24, 2);
-- --------------------------------------------------
--
-- Table structure for table `subjects`
```

```
--

CREATE TABLE `subjects` (
  `id` int NOT NULL,
  `name` varchar(55) NOT NULL,
  `shortcut` varchar(50) NOT NULL
);

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

INSERT INTO `subjects` (`id`, `name`, `shortcut`) VALUES
(1, 'Operating System', 'OS'),
(2, 'Basic Electronics', 'BE');
-- --------------------------------------------------------

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

CREATE TABLE `teachers` (
  `id` int NOT NULL,
  `fname` varchar(55) NOT NULL,
  `lname` varchar(55) NOT NULL,
```

```
  `age` varchar(11) DEFAULT NULL
);

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

INSERT INTO `teachers` (`id`, `fname`, `lname`, `age`) VALUES
(1, 'Frank', 'Reed', '55'),
(2, 'Brian', 'Weiss', '66');
--
-- 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`
--
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`
--
ALTER TABLE `students`
  ADD CONSTRAINT `class_id` FOREIGN KEY (`class`)
REFERENCES `classes` (`id`);
COMMIT;
```

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:

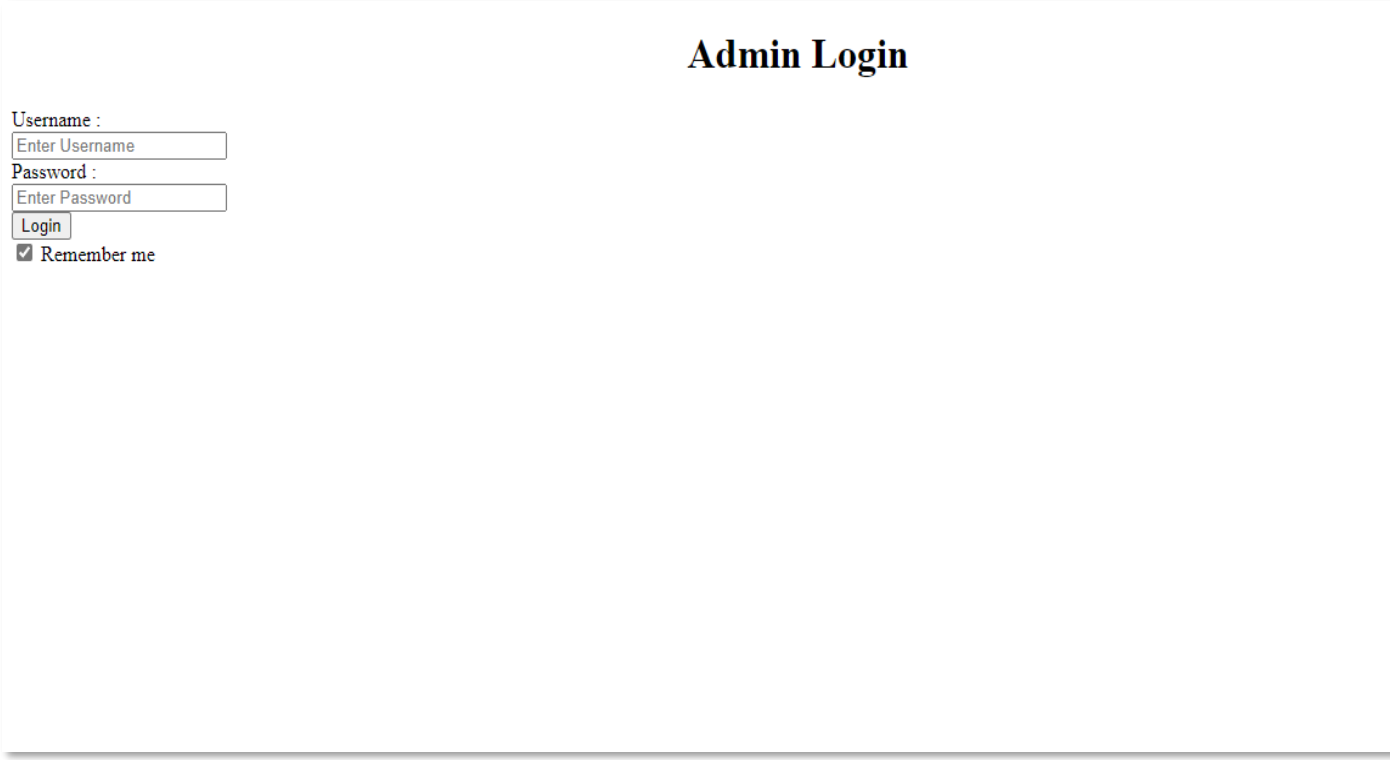
git commit -m <commit message>

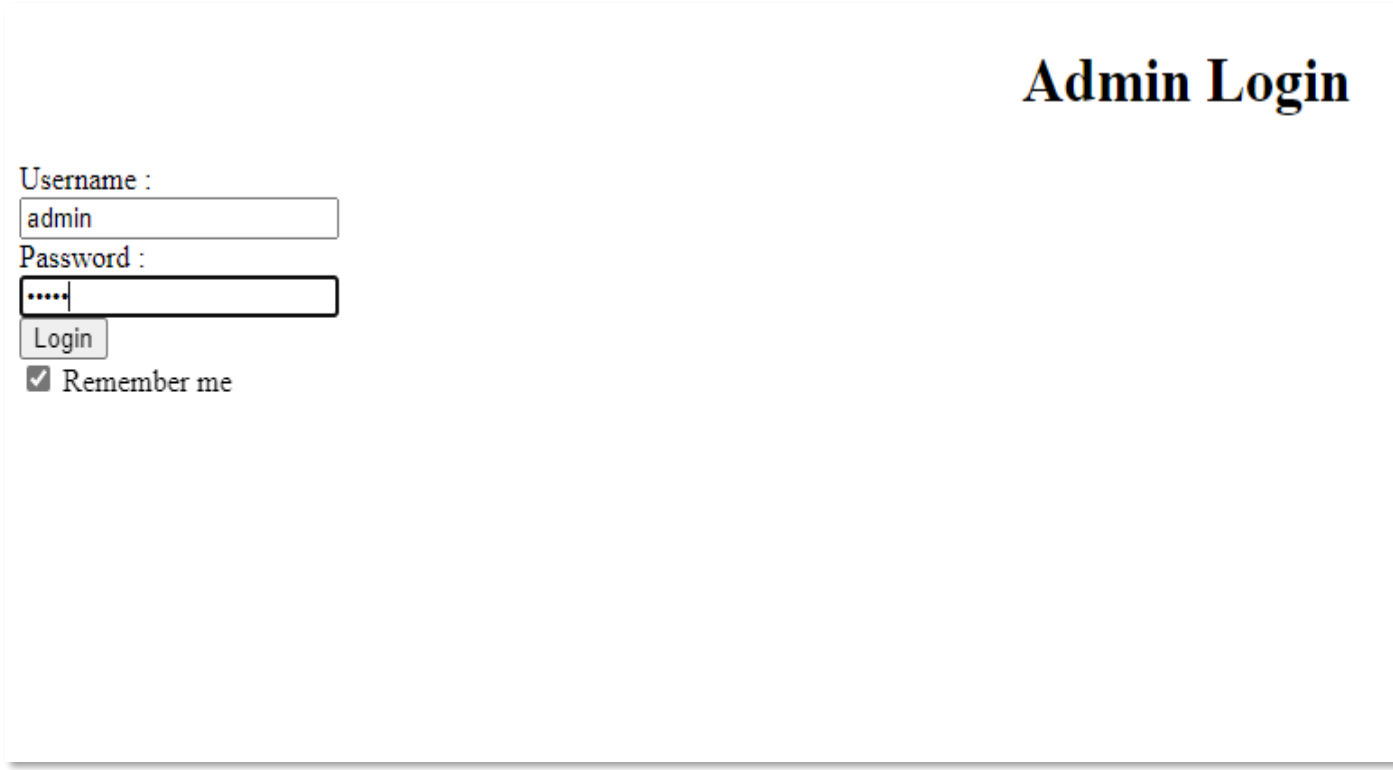
- Push the files to the folder you initially created using the following command:

git push -u origin master

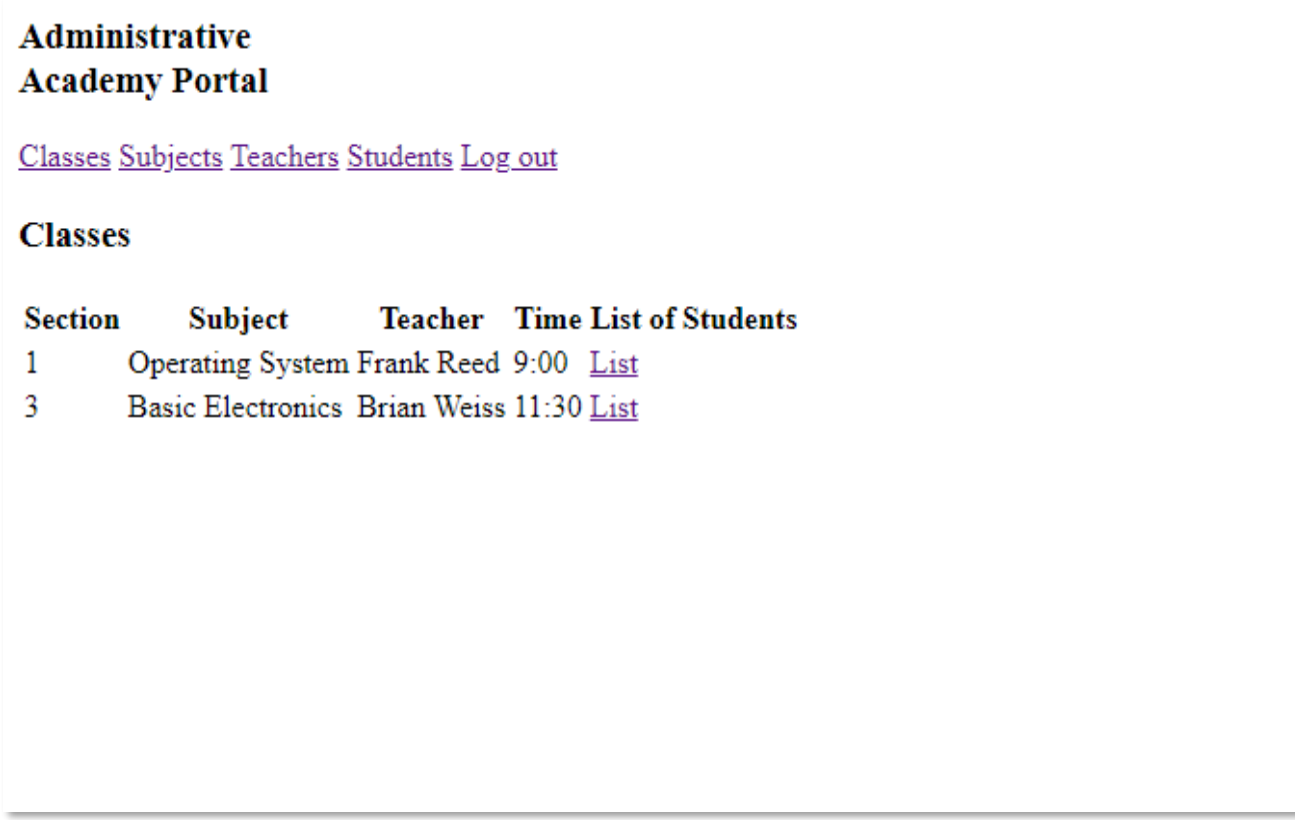
Application View:

1. Login Page

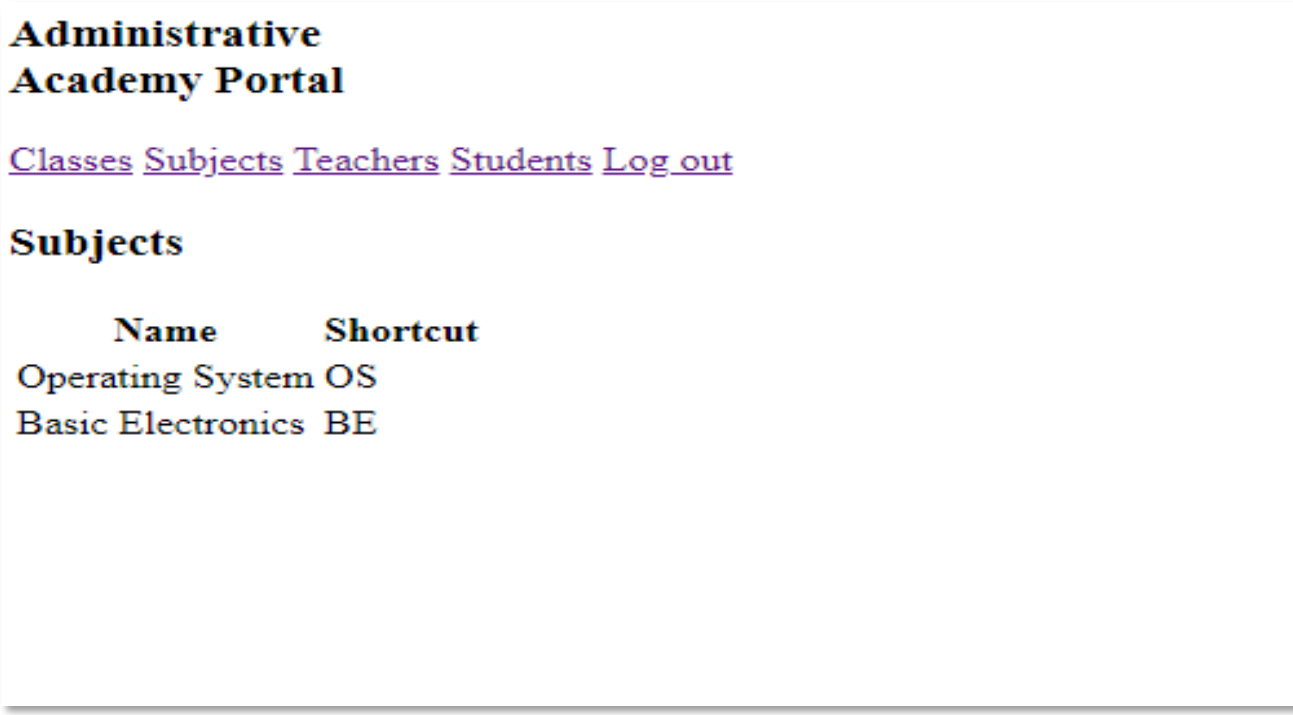



Admin Login
Username :
admin
Password :
Login
Remember me

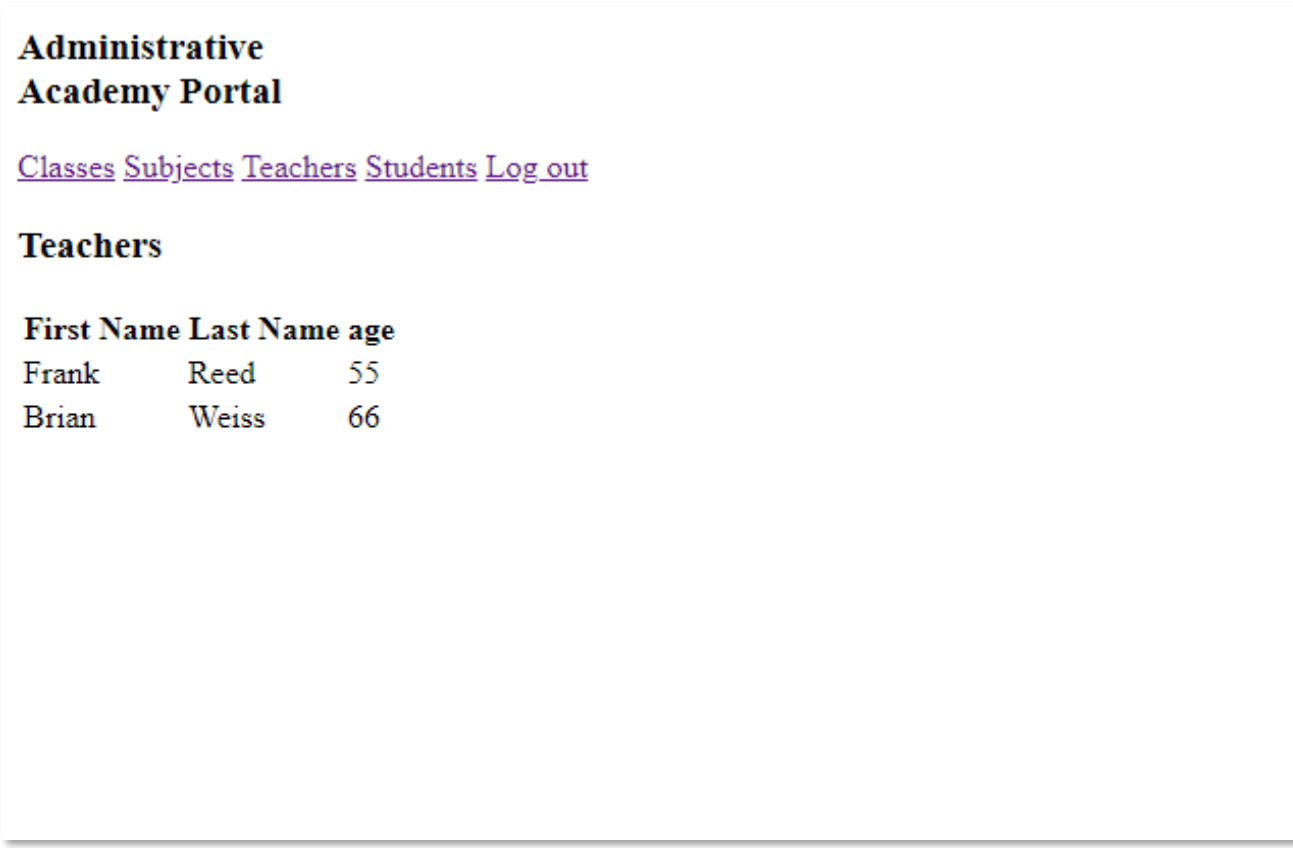
2. Dashboard


Administrative
Academy Portal
Classes Subjects Teachers Students Log out
Classes
Section Subject Teacher Time List of Students
1 Operating System Frank Reed 9:00 List
3 Basic Electronics Brian Weiss 11:30 List

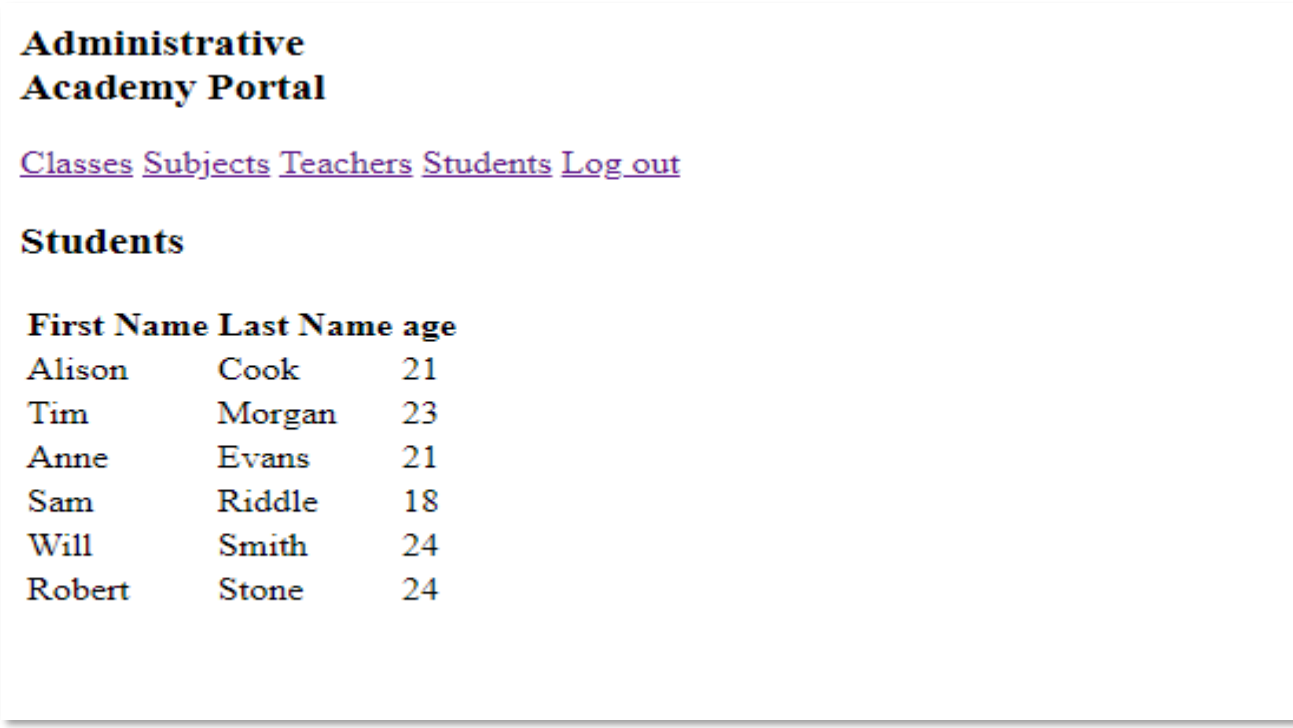
3. List of Subjects



4. List of Teachers



5. List of students



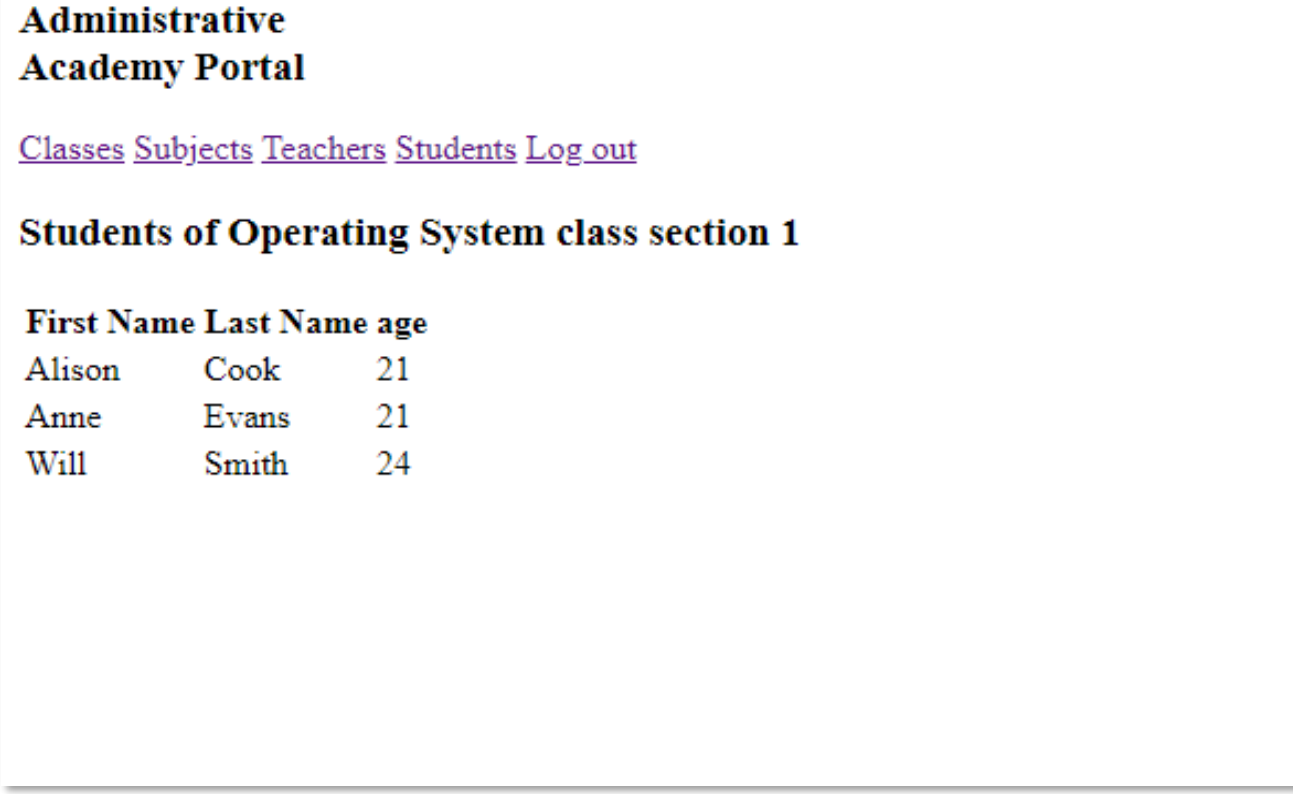
Administrative
Academy Portal

Classes Subjects Teachers Students Log out

Students

First Name	Last Name	age
Alison	Cook	21
Tim	Morgan	23
Anne	Evans	21
Sam	Riddle	18
Will	Smith	24
Robert	Stone	24

6. Class report listing students enrolled in a particular subject



Administrative
Academy Portal

Classes Subjects Teachers Students Log out

Students of Operating System class section 1

First Name	Last Name	age
Alison	Cook	21
Anne	Evans	21
Will	Smith	24