

SMS Testing

Once the application is up and running, there are few assumptions that we made as part of our application. They have been mentioned in our design document as well.

1. Admin account doesn't have a sign-up page. We need to insert an Admin record manually. Use the below query:

```
INSERT INTO user_login (email, first_name, last_name, password, phone_no, role, username)
```

```
VALUES ('admin@gmail.com', 'admin', 'root', '123', '2332312', 'admin', 'admin');
```

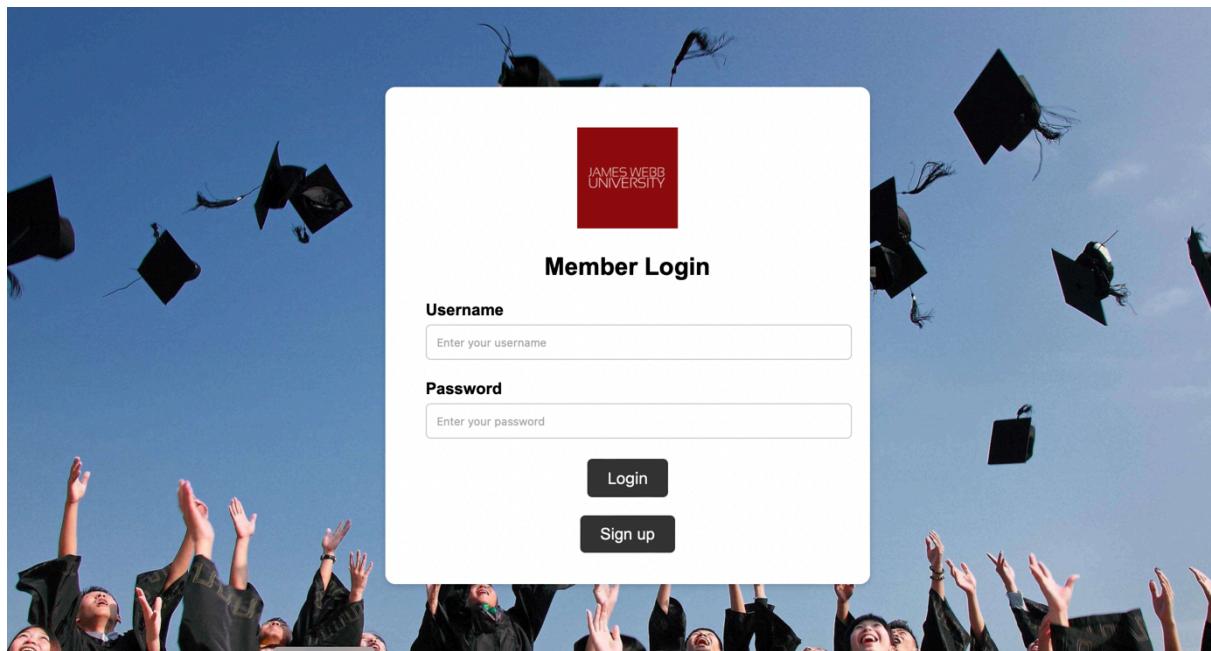
2. Once a teacher signs-up with a new account, they cannot directly login.

Admin needs to map the teacher with a course.

3. One teacher can be mapped to only Course.

Now, once the admin record has been added. We are good to go with the application.

Login screen: <http://127.0.0.1:5500/templates/homelogin.html>





Screenshot

Signup

[Student](#) [Teacher](#)

First Name

Last Name

Phone Number

Email

Username

Password

[Register](#)

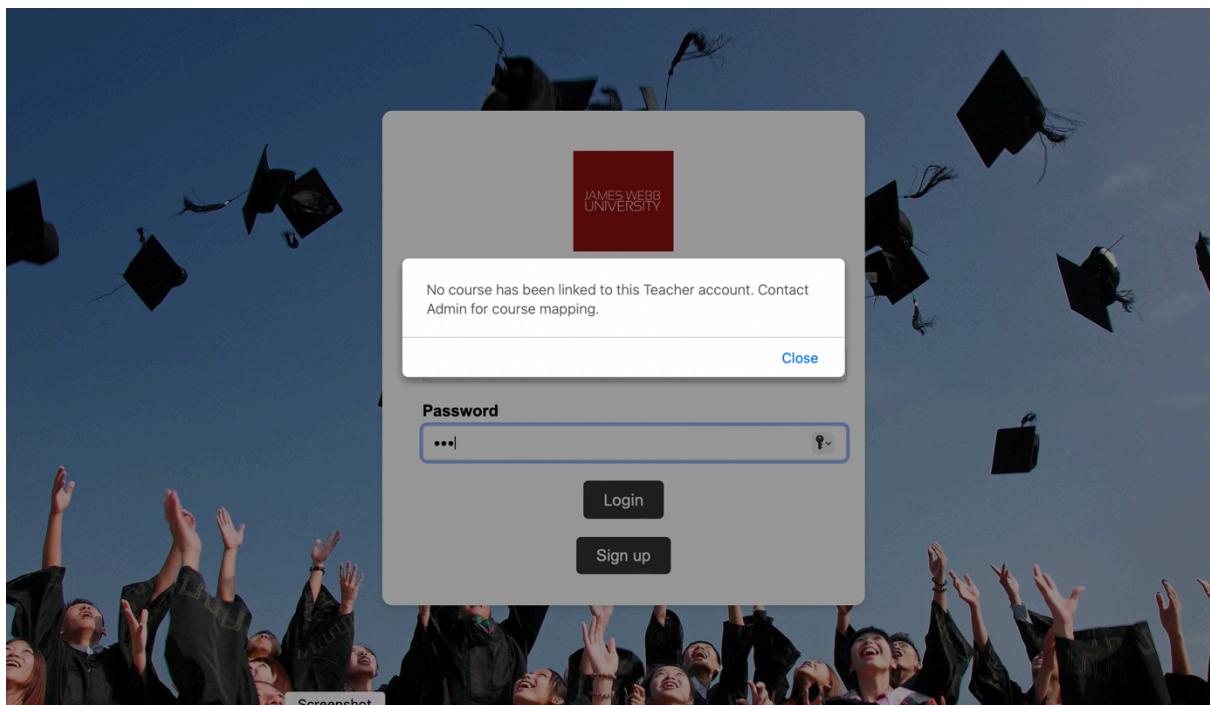
Already have an account? [Click here to login](#)



Now let me sign up 3 teachers.

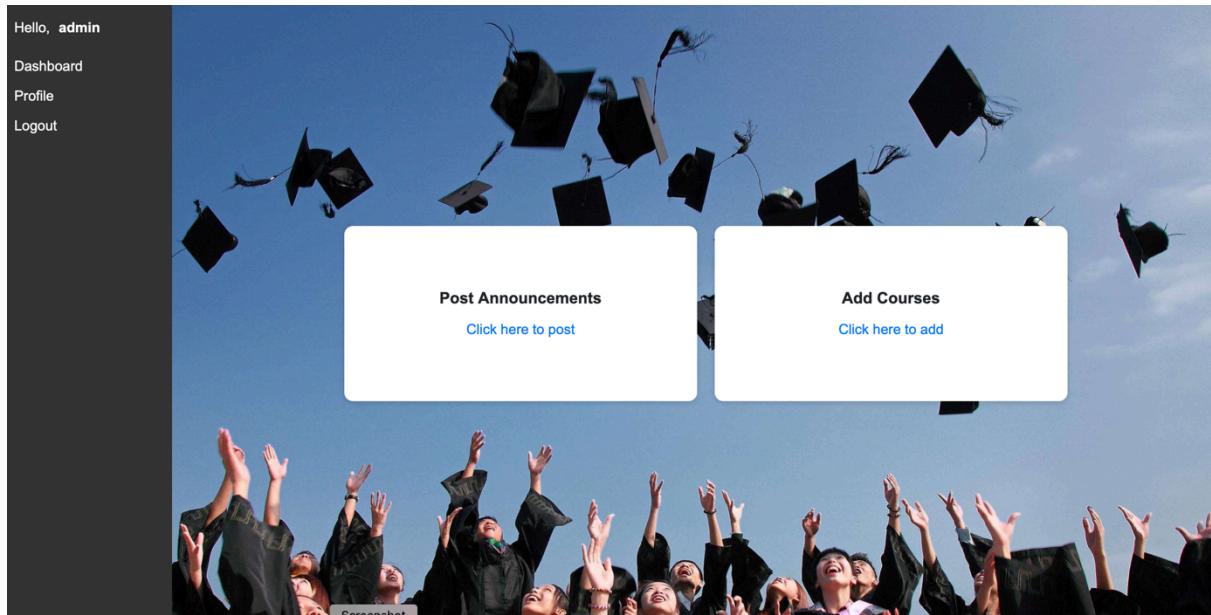
John, Sebastian, Mike – 3 teachers have been signed up recently.

But they won't be able to login until the admin maps them.

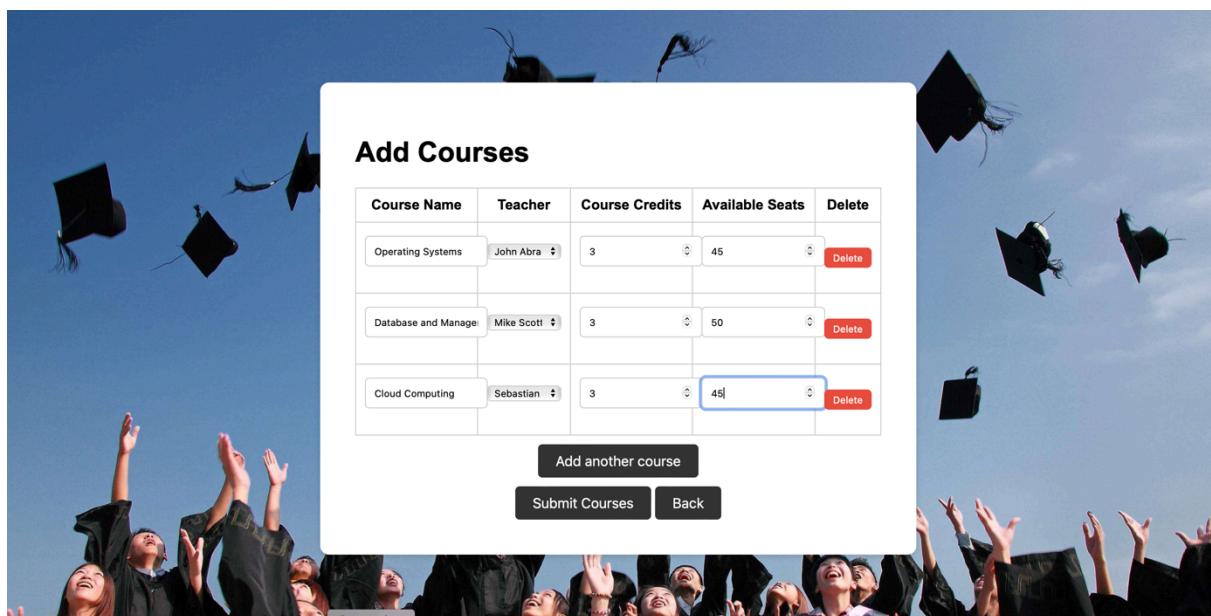


Screenshot

Now, let admin login and map the courses for them.

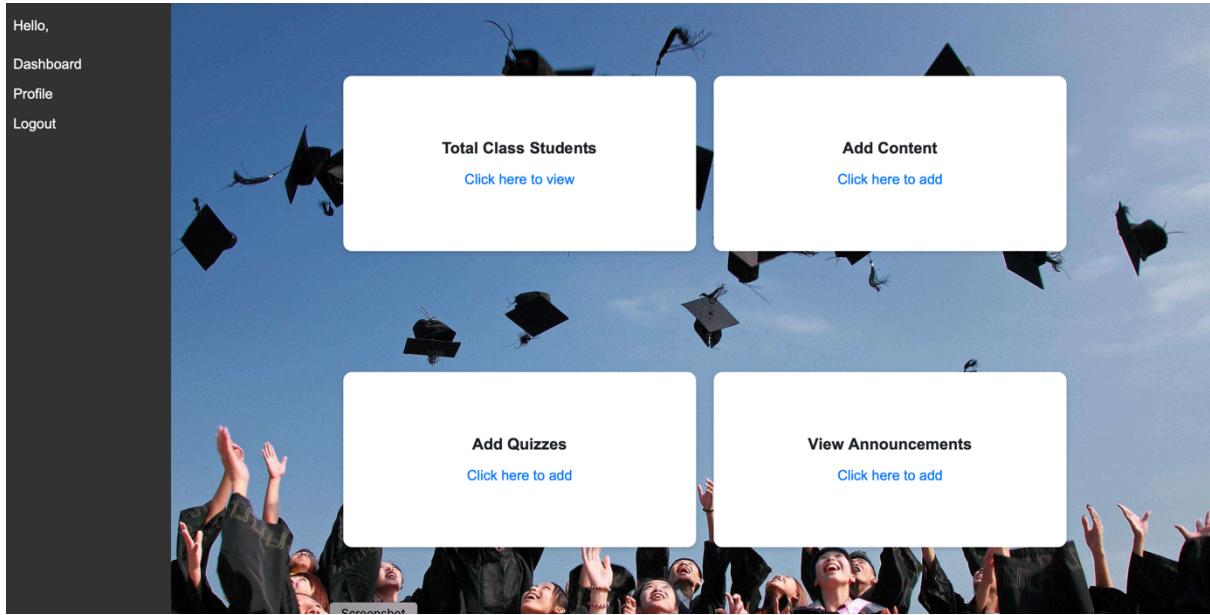


Mapping courses to the teachers:



Now the teachers will be able to login.

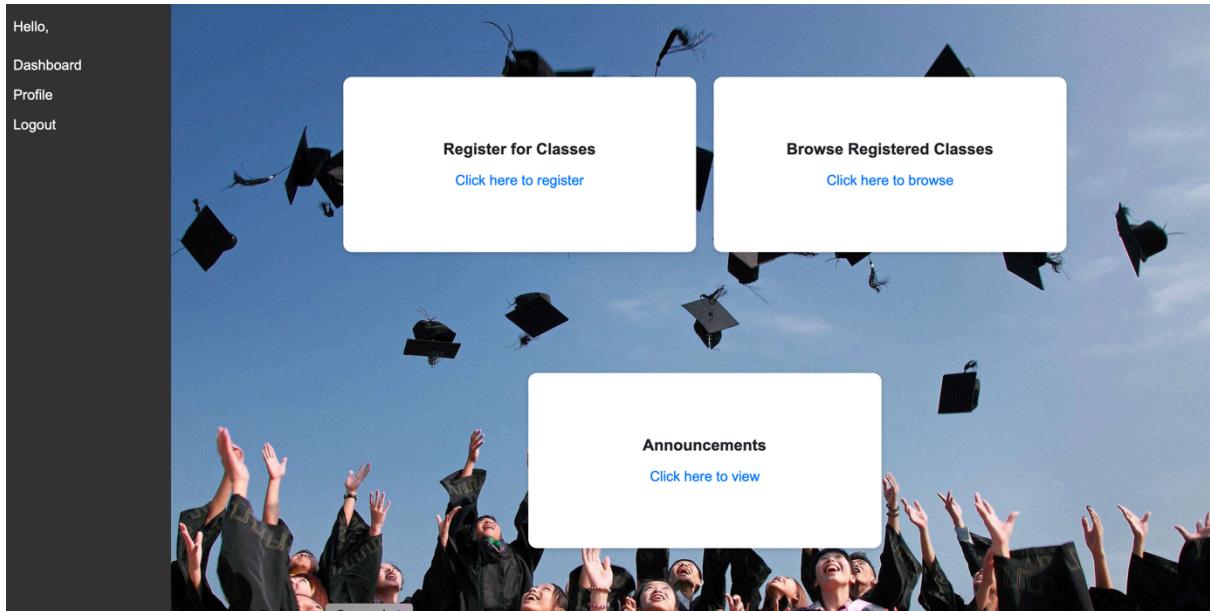
Logged in to a teacher : John



A teacher will be able to do following things :

1. Update his profile
2. View total students enrolled in his course
3. Add content into his course
4. Add quizzes into his course
5. View any announcements posted by the admin

Similarly a new Student will be able to do the following things:



Student:

1. Updates his profile
2. Register for new classes

3. Browse the registered classes
4. View content/quizzes in the registered class
5. View announcements posted by admin

All the Rest APIs that are being called in the front-end are tested via Post man.

The screenshot shows the Postman interface with a collection named 'StudentManagementSystem'. A specific test case titled 'Get All Courses' is selected. The request method is 'GET' and the URL is 'http://localhost:8080/sms/course/getall'. The response status is '200 OK' with a time of '56 ms' and a size of '579 B'. The response body is displayed in JSON format:

```

1 [
2   {
3     "course_id": 1,
4     "course_name": "Operating Systems",
5     "course_credits": "3",
6     "available_seats": 45,
7     "teacher_id": 1
8   },
9   {
10    "course_id": 2,
11    "course_name": "Database and Management",
12    "course_credits": "3",
13    "available_seats": 50,
14  }
]

```

Postman collection for all our REST APIs which acts as a Proof For our Application Testing:

https://github.com/kvpvishnu11/sms/blob/main/StudentManagementSystem.postman_collection.json