

Pave Full Stack Coding Assessment

Background

Tutors need a system where they can manage their students. Tutors and students are identified by their email addresses.

Your Task

Your task is to:

1. Develop a set of API endpoints, listed under Tasks below, for tutors to perform administrative functions for their classes.
 - a. Your code must be hosted on Github, or any other similar service, in a publicly-accessible repository.
 - b. You may assume that login and access control have already been handled.
2. (Optional) Deploy your API to any publicly accessible hosting environment.
3. Develop a web app frontend project
 - a. Your code must be hosted on Github, or any other similar service, in a publicly-accessible repository.
 - b. You may assume that login and access control have already been handled.

You have **one week** to work on this assignment starting from the day you receive it.

When you have completed your assignment, before the given deadline, please submit to us (at mervin@yourpave.com) a link to your code repository.

If you have any queries, feel free to send them to mervin@yourpave.com.

Expectations/Requirements

1. Your code repository should contain a `README.md` that includes the following:
 - a. Link(s) to the hosted API (if applicable)
 - b. Instructions for running local instance of your API server; we need to minimally be able to launch and test your solution locally
2. Please use NodeJS for the backend code.
3. Please use MySQL/SQLite as the database.
4. Please include unit tests.
5. Please use Next.js for the frontend code (with Axios for API requests).
6. If you are selected for next round of interview, you should be prepared to:
 - a. Walk through your code to interviewers
 - b. Explain any design decisions you have made
 - c. Modify the API endpoints, or implement more endpoints (if requested)

Important!

- We will assess your submission holistically (i.e. not just in terms of functionality), including factors such as:
 - Readability and code cleanliness
 - Secure coding practices
 - Code structure/design, e.g. modularity, testability
- Your API will be subjected to automated test tools, so please adhere closely to the given specs.
 - (Optional) You can provide a Postman collection for the APIs that you've implemented, but we can (and likely will) still use our own tools as well to test your API.

Tasks

1. As a tutor, I want to register one or more students to a specified tutor.

A tutor can register multiple students. A student can also be registered to multiple tutors.

```
* Endpoint: `POST /api/register`
* Headers: `Content-Type: application/json`
* Success response status: HTTP 204
* Request body example:
...
{
  "tutor": "tutorken@gmail.com"
  "students":
  [
    "studentholt@gmail.com",
    "studentjohn@gmail.com"
  ]
}
```

2. As a tutor, I want to retrieve a list of students common to a given list of tutors (i.e. retrieve students who are registered to ALL of the given tutors).

```
* Endpoint: `GET /api/getcommonsstudents`
* Success response status: HTTP 200
* Request example 1: `GET /api/getcommonstudents?tutor=tutorken%40gmail.com`
* Success response body 1:
...
{
  "students" :
```

```
[
  "common_student_1@gmail.com",
  "common_student_2@gmail.com",
  "student_only_under_tutor_ken@gmail.com"
]
}
```

* Request example 2: `GET

/api/getcommonsstudents?tutor=tutorken%40gmail.com&tutor=tutorjoe%40gmail.com`

* Success response body 2:

```
{
  "students" :
  [
    "common_student_1@gmail.com",
    "common_student_2@gmail.com"
  ]
}
```

3. As a tutor, I want to retrieve a list of students who can receive a given notification.

A notification consists of:

- * the tutor who is sending the notification, and
- * the text of the notification itself.

To receive notifications from e.g. tutorken@gmail.com', a student:

* AND MUST fulfill **AT LEAST ONE** of the following:

1. is registered with "tutorken@gmail.com"
2. has been @mentioned in the notification

The list of students retrieved should not contain any duplicates/repetitions.

* Endpoint: `POST /api/retrievenotifications`

* Headers: `Content-Type: application/json`

* Success response status: HTTP 200

* Request body example 1:

```
{
  "tutor": "tutorken@gmail.com",
  "notification": "Hello students! @studentanne@gmail.com @studentmary@gmail.com"
}
```

* Success response body 1:

```
{
  "recipients":
```

```
[
  "studentpeter@gmail.com",
  "studentanne@gmail.com",
  "studentmary@gmail.com"
]
}
```

In the example above, studentanne@gmail.com and studentmary@gmail.com can receive the notification from tutorken@gmail.com, regardless whether they are registered to him, because they are @mentioned in the notification text. studentpeter@gmail.com however, has to be registered to tutorken@gmail.com.

* Request body example 2:

```
{
  "tutor": "tutorken@gmail.com",
  "notification": "Hey everyone"
}
```

* Success response body 2:

```
{
  "recipients":
    [
      "studentpeter@gmail.com"
    ]
}
```

4. (Frontend) As a tutor, I want to see a list of students common to a given list of tutors (i.e. retrieve students who are registered to ALL of the given tutors).

Assuming login and authentication has already been done. Build a frontend UI that has the following:

- Develop a form to capture email address of a tutor
- Submission of form will send a POST request to API endpoint at question 2
- Display list of students in a table format returned from API in a table format

API Error Responses

For all the above API endpoints, error responses should:

- have an appropriate HTTP response code
- have a JSON response body containing a meaningful error message:

```
{ "message": "Some meaningful error message" }
```