Fullstack project - read carefully.
Choose between 2 topics provided in the REST API Slides

Installation

You should use either `npm` or `yarn` but not both. **It's recommended** to use `yarn`

This template already comes with all the needed packages. In case you want to install extra ones ie MUI feel free.

you have two different folders:

1. `api:` where your backend lives

2. `client:` Frontend of your application

To install, you need to go to both directories and install their packages as such:

```bash
cd api
yarn install
```

```bash
cd client
yarn install
```

To run the `frontend` make sure you are under the client folder and:

````bash
yarn start
````

To run the `backend` make sure you are under the api folder and:

````bash
yarn start:dev
````

````
<br />

> `Pro tip:`Open 3 different terminals. 1 for client dir (to run the client), 1 for api dir (to run the api), and another in the root directory for committing your changes.

<br />

## `api` folder

1. Create a `.env` file in the root directory and copy the content from `.env.example`
2. Make sure MongoDB is running (if you are using local MongoDB)
3. If you need to customize your env, take a look at `secrets.ts` file
4. Separate routes and functions into routers, controllers, services folders

<br />

## `client` folder

1. Create a `.env` file in the root directory if you need to store secret data
2. You can complete your project using SASS, CSS, or other styling libraries

<br />

## Requirements of the project:

Check the REST API slides for what are the required features

*you are allowed to add more features if you finish the required ones.*

<br />

## How should you approach the assignment?

1. **OPEN a PR** as soon as you get your first task or feature implemented. And keep doing that for every. single. feature.
````

```
   1. **use the** Ready for review label and **leave a comment** to let us know what to review
2. Make sure you don't spend too much time on a task (if you are, it means you need help or you will fall behind if you don't ask for help)
3. **DO NOT** copy-paste a bunch of code and then try to figure out how to make it work. what you should be doing is taking it one step at a time.
4. **MAKE SURE** to **PLAN** then execute. when you plan a feature you should visualise how it would be coded and what you need to complete it. the more time you spend on planning and thinking about the steps you should do, the easier and faster the coding would be.
5. If you have a problem that you are trying to solve, split it into smaller problems and make sure you solve one and then move to the next one.
6. You are advised to follow this approach:
   1. Build one Model at a time, so if you are starting with products then build its model and only products model (if this collection would have a relation with another collection then it's fine to build that too)
   2. Focus on finishing the most basic routes of all CRUD operations in their simplest format. (without fancy controllers or aggregations etc)
   3. The idea is MVP approach you want to build the basic routes and most basic features then start adding up more complex logic as you go further.
   4. once the basic routes are done, move to the client folder and start connecting the routes you built with the frontend. (you have to see the results)
   5. if you start that way you are good to start playing around and have more complex logic whether in backend or frontend
   6. if you follow this approach you would practice, one of the most important things is to add features to an existing codebase.
7. And again, **PLEASE**, **DO NOT** push your code when you have so much code written. it will be so hard to suggest better approaches or solutions and you would end up with a very basic code review. If you don't care and don't want a code review feel free to push your fullstack project at the last minute.

<br />

## Recap on how to start developing
1. Build basic routes
2. Connect some or most of the routes to your frontend
```

3. Start adding Auth after the security lecture (Please, I don't want to see Auth code when you don't have all the routes implemented + some routes being fetched from the frontend)

> PS. *you are only allowed to add Auth before the security lecture **if** you have all routes implemented + some routes fetched from the frontend then*

Do you have a problem and want quick help?

Sending me/Anisul a message saying my app is broken would help **no one**.
you should have the following:

1. what are you trying to do
2. when the problem happens at what point
3. screenshot of the code flow

FAQ

* What if I finished earlier?
  * You can always have extra tasks; reach out to me if you need help on what to add.
* Can I add more features?
  * Sure you can, but first, you are required to finish the requirements
* I'm progressing slowly! I'm stuck!
  * Reach out to me or Anisul. (Remember you have a deadline to meet)
* Deadline?
  * check your calendar (presentation days)