Hi Omer Shafrir,
Hope you are well.

Following our conversation, I've attached the coding task here:

Tom is Josh's mentor for JS, but while Covid is still out there he prefers to do remote sessions.
Tom wants to share with Josh a piece of code, observe him while he is writing and changing the code in r eal time.

Help Tom creating an online coding web application with the following pages and features :

Lobby page (no need for authentication) :
The page should contain the title "Choose code block" and a list of at least 4 items which represents code blocks, each item can be represented by a name (for example - "Async case")
Clicking on an item should take the user to the code block page with the details of the code block he choo ses.

Code block page :
Both users should enter this page. (2 different clients)
Assume that the first user who opens the code block page is the mentor, after that, any other user will be counted as a student.

The mentor will see the code block he choose with a read only mode
The student will see the code block with the ability to change the code
Code changes should be displayed in real-time (Socket)
Use Highlight.js (or any equivalent library) to highlight the syntax
(Support JS code only)

General guidelines:

Code blocks should be created manually, no need for API or UI.
A code block should have the fields 'title' and 'code' (code is a string which represent JS code)
Add clear comments to the code where needed.
This task involves client server and DB, you can use any framework/language you want.

***Submission instructions: ***

1. Deploy the project and supply the url for the app.
You can use any service you would like for hosting your deployment (There are many free services for tha t purpose - railway.app, Netlify, Vercel etc. )

2. Upload your code to GitHub and attach a link to your GitHub repository.

Bonus:
Have a "solution" on a codeblock object (also insert manually), once the student changes the code to be e qual to the solution, show a big smiley face on the screen :)

If you have any questions regarding the assignment do not hesitate to contact me over email or phone.

Good luck!
Moveo's HR team

# Application: Online coding web application

- Next.JS : Socket.io
- MongoDB
- Vercel Hosting

## 1) Lobby Page - (landing page)

- Login mechanism (Authentication) : optional
  - nextAuth / custom session management with Cookies
- "choose code block" button → redirects

## 2) Login Page (Optional)

## 3) Tasks Page

- List of JS Tasks:
  - Each lesson redirects to Editor Page with chosen code block.
  - Lessons are fetched from Mongo database, into state holding Task object: title, content, solution

## 4) Editor Page

- Users session management:
  - 1st impl.: first user to enter is mentor, second is student
  - 2nd impl: Both mentor & student registered: User entity has 'type' field: Mentor or Student.
- Student Editor Page: Read-Write, real-time collaboration with Mentor via socket.io (Mentor Read Only).
  - text highlighting
- Submit Button: Compares JS block with solution (string comparision). Success handling - UI.

Home Page (App comp)

Login (optional)

App Name

Choose code Block

Animation / creative

Login Page

Home

Option 1:

username

password

Option 2:

Authorize with google / github

default:

Sign in as: | student/mentor

Tasks

Home

Choose a code block

1. Lesson #1
2. Lesson #2

mentor

student

Editor

Back

JS code

Read / Write

changes in real time

Socket.io

Submit

Editor

JS code

Read only

- Pages
  - app.js
  - homepage.js
  - Login.js
  - Practices.js
  - Editor.js
- components:
  - CodeFrame.js

Mongo:

task: {'title': string, 'content': string, 'solution': string}

? student: {name, username, password, email, ? completed tastes}

? mentor: {name, username, password, email}

# Progress:

1. Static version of the app, Mongo initialization
   1. Mongo init + communication
   2. util page with: Taste Type
   3. Defining Layout & Header files. Layout will handle styles for all project, Header will have Homepage Link
   4. Defining Homepage, Editor & Tastes pages.

2. Adding interactivity - Editor component, integrating Socket.io
3. Design & css
4. Deploying to Vercel

## Questions:

1) Can I use whatever packages & frameworks as I wish?

2) Do I have to stick to the structure provided in the task summary or can I implement it in a bit different way as I see fit, as long as my app will function as requested?

3) Should my app support more than 1 on 1 sessions at the same time?

   If the answer is no: Implementation details as to how to handle 3rd and 4th users entering the app are left to me?

4) What information should my app maintain? (things like student progress, tasks completed etc.)