Full Stuck Assessment

Thank you for your interest in the full stack position at Modex. As part of our hiring process, we would like to assess your Node.js and react skills. Please complete the following tasks and share a git repository

Task 1: Problem Solving

You are given a programming problem to solve using Node.js. Write a Node.js script that accomplishes the following:

Problem Statement:

You are given an array nums of integers representing a sequence of stock prices for a single stock over a period of n days. Your task is to write a Node.js function maxProfitK(prices, k) that calculates the maximum profit that can be obtained by completing at most k transactions.

Write a Node.js function maxProfitK(prices, k) that takes an array prices of integers representing the stock prices and an integer k representing the maximum number of transactions allowed, and returns the maximum profit that can be obtained.

Input:

An array prices of integers representing the stock prices for each day, where prices[i] is the price of the stock on the i-th day (0-indexed).
An integer k representing the maximum number of transactions allowed.
Output:

An integer representing the maximum profit that can be obtained by completing at most k transactions.

Task 2: API Development

Your task is to implement a solution in Node.js. You may use any external libraries or packages if necessary. Please ensure that your solution is well-documented, efficient, and follows best practices for Node.js development.

Please use docker for this project

Design and implement a simple RESTful API using Node.js and Express.js. The API should include the following endpoints:

GET /api/users: Retrieve a list of users
POST /api/users: Create a new user
GET /api/users/:id: Retrieve a specific user by ID
PUT /api/users/:id: Update an existing user by ID
DELETE /api/users/:id: Delete a user by ID

You may use any data storage solution (e.g., in-memory storage, MongoDB, etc.) to persist user data. Ensure that your API follows RESTful principles, includes error handling, input validation, and proper documentation (e.g., using Swagger or OpenAPI).

Please make sure to include also testing

Task 3 : React Web application

Your task is to create a react web application that implement the use of the full API from Task2

For this task you may use tailwind or any other library

Good Luck!