

Backend Engineer Technical Test

Context:

Imagine you are part of a team developing a software platform aimed at enhancing users' mental well-being through a curated set of activities. These activities are designed to promote relaxation, boost self-esteem, improve productivity, enhance physical health, and foster social connections.

Time

We give the candidate a maximum of 7 days to complete all the tests

Task:

Your task is to develop a secure and efficient Node.js backend that enables users to explore, complete, and track various activities aimed at improving their mental well-being.

Users should be allowed to:

- Register
- Log in
- List all the activities
- Bonus:
 - Mark activities as completed
 - List completed activities

Information Needed:

We recommend generating a set of default activities with random information.

Activities Categories

The activities could be categorized into the following categories:

- Relaxation
- Self-Esteem
- Productivity

- Physical Health
- Social Connection

Activity Attributes

Each activity should have the following attributes (you can use placeholders as “lorem ipsum” for titles, descriptions and content):

- Title
- Description
- Category: The category to which the activity belongs.
- Duration: The estimated time required to complete the activity.
- Difficulty Level: An indication of the complexity or challenge level of the activity.
- Activity content: This is a long text representing the content of the activity.
- Creation Timestamp: The timestamp indicating when the activity was created.

User's Information

User profiles should include basic information such as username, email, and password for authentication purposes.

Requirements:

REST API:

- Users should be allowed to mark activities as completed.
- Users should be able to list their completed activities.
- Bonus:
 - Implement CRUD endpoints for activities, allowing potential integrations with a backoffice system or third-party services.

Database:

- Choose an appropriate database (e.g., MongoDB, PostgreSQL) for storing activity data and user information, ensuring scalability and reliability.

Authentication and Authorization:

- Develop an authentication mechanism (e.g., JWT) to secure API endpoints, ensuring that only authenticated users can access and modify their activity data.

Submission:

- Submit your solution in a GitHub repository. A well-structured commit history will be appreciated.