Back End Developer Task

We appreciate your interest in the position and your enthusiasm for our upcoming projects! As part of our interview process, we assign a task to assess the participant's skills, creativity, and compatibility with our dynamic team. Team building is fundamental to our values, and we are seeking developers who bring exceptional ideas and seamlessly integrate with our team.

We look forward to seeing how your talents can enhance our team and shape the future of our game projects!

Introduction

We appreciate your interest in participating in this interview task. This assignment should be completed within **72 hours** of receiving these instructions, allowing for a fair evaluation of all candidates.

If you require an extension beyond the initial 72 hours, please notify us in advance. Keep in mind that the assessment considers development time, so only request additional time if you are confident it will significantly improve the end result.

Please prioritize the quality and usability of the website you create above all else.

Please note that anything created during this interview will not be used as part of any development process at the company. Feel free to add your watermark to the work as necessary.

Interview Task

We aim to evaluate your **Back End Developer** skills across several key aspects. This includes how well the endpoints are implemented, the quality of the code, proper implementation of authentication and data validation, efficient handling of CRUD operations, database interactions and .

PART ONE:

Develop a basic backend API for a content-sharing platform that performs simple CRUD operations. The focus should be on creating endpoints for managing user-generated content, including games, videos, artwork, and music.

1. **Project Setup:**
    - Use a backend framework (Node.js with Express, Django, Flask, etc.).
    - Set up the project with necessary dependencies.

2. **API Endpoints:**
   - **Content Management:**
     - Create endpoints to handle CRUD operations (Create, Read, Update, Delete) for content.
     - Each endpoint should expect/operate with the attributes defined under section 3 - Database.
     - **(Optional)** Create endpoints that handle operations designed with pagination, filtering and sorting in mind.
   - **User Management:**
     - Create endpoints for user registration and login.
     - Use JWT or any other token-based authentication mechanism for securing endpoints.
   - **Rating System:**
     - Create an endpoint to allow users to rate content (1 to 5 stars).
     - Each rating should be associated with an existing user, along with the content item that it refers to. This relationship is to be further defined by you.
3. **Database:**
   - Use a relational database (PostgreSQL, MySQL) or a NoSQL database (MongoDB).
   - Set up the database schema to support the **media content**, **user** and **rating** data models. Their attributes should be defined as:
     - **Media content:** `media_id`, `title`, `description`, `category` (game, video, artwork, music), `thumbnail_url`, `content_url`, and `created_at`.
     - **User:** `user_id`, `username`, `email`, `rating_count`, `last_login`, and `created_at`.
     - **Rating:** These are to be defined by you.
4. **Sample Data:**
   - Populate the database with some sample data for testing/demonstration purposes. These would include
     - **User data** (at least 2),
     - **media content data** (at least 5),
     - and **ratings data** (examples of:
       1. Two or more ratings from one user on one media content,
       2. Two or more ratings from different users on one media content, and
       3. Two ratings from one user on different media content)
5. **Documentation:**
   - Provide API documentation using Swagger, Postman, or a similar tool.
   - Provide an OpenAPI 3.0 yaml or json.
   - Include details on how to set up and run the project locally.

NG+

PART TWO:

Create a presentation that visually showcases your work.

Create a presentation for your project using pitch.com, and record a comprehensive **5-minute video** - we recommend using OBS, Vimeo, Pitch or Loom. During the presentation, be on camera for a personal touch and use a microphone to ensure clear audio. Prioritize a 720p resolution for the video, and make sure the content is audible and encompasses your entire presentation.

Instructions:

1. Begin with a brief introduction, lasting no more than 20 seconds, where you introduce yourself and outline your career as a Back End Web Developer.
2. Explain the system setup, how to execute the scripts, and an overview of the API and database interactions.
3. Omit the addition of music in the video, focusing solely on the spoken content.

Submission Guidelines

- Host your code on a platform like GitHub and provide the repository link (make sure it's public).
- Include a README file with instructions on how to set up and run the project.
- Include API documentation (Swagger/Postman link or a markdown file).
- Use **pitch.com** for your presentation (please make sure it's public).
- Ensure the presentation is in English.
- Once completed, send the link to the **interviewer email** and mail it to the company's email: **hiring@ngp.careers**

Within the next few days, we will provide feedback along with the results. We wish you the best of luck and encourage you to enjoy the process!