iEKO MEDIA

3D VISUAL DESIGN \ VR \ AR \ COMMERCIAL FILM & PHOTOGRAPHY SERVICES

**Full Stack Developer Technical Assessment**

Objective:
This assessment aims to gauge your technical skills, problem-solving abilities, and proficiency in Full Stack Development, with a focus on React.js, Tailwind CSS, MongoDB, Digital Ocean, and other related technologies. You are expected to develop a small-scale application adhering to the outlined requirements. Typescript is preferred language, preferred database MongoDB
Deliverables:

1. Source code hosted on a GitHub repository.
2. A working version of the application hosted on Digital Ocean.
3. A short video (10 - 15 mins) explaining your code structure, logic, and the decisions made during development.

Time Allocation:
This assessment is designed to be completed within 8 hours. However, quality is paramount over speed. Please ensure your submission is complete and well-structured.

## Part 1: Backend Development (Node.js, MongoDB, REST APIs (GraphQL is more appreciated))

**Task 1.1: Setup REST/GraphQL API**

- Set up a REST/GraphQL API using Apollo Server.
- Implement the following functionalities:
    - Query all users with pagination and sorting options.
    - Query a single user by their ID.
    - Create, update, and delete user mutations with necessary validations and authorization.

**Task 1.2: Implement Job Queue with Redis**

- Integrate Redis into your Node.js application to implement a job queue system with the following features:
    - Job creation: Allow users to create jobs with specific data and enqueue them in the Redis job queue.
    - Job processing: Implement a worker that continuously checks the job queue and processes jobs as they become available, ensuring each job is processed only once.
    - Job status and result: Provide an endpoint or REST/GraphQL query to check the status and result of a specific job by its ID.

**Task 1.3: Caching**

- Add caching functionality to your Node.js application to improve performance. Apply caching to frequently accessed data or expensive operations using a caching mechanism of your choice (e.g., Redis or in-memory caching).

**Task 1.4: Error Handling and Logging**

- Implement a logging mechanism that logs relevant events and errors to a file or a centralized logging system. Ensure that errors are properly handled and appropriate error responses are returned to clients.

## Part 2: Frontend Development (React.js, Tailwind CSS)

**Task 2.1: Setup React Application**

- Set up a React application and style it using Tailwind CSS.
- Implement routing and state management. (React Query is preferred).

**Task 2.2: User Management Interface**

- Create a user management interface where admin can:
    - View a list of users.
    - View details of a single user.
    - Create, update, and delete users.

**Task 2.3: Data Visualization**

- Create a dashboard displaying some mock statistics, charts, and graphs using a library of your choice (e.g., Chart.js, D3.js).

## Part 3: Deployment and Scalability (Digital Ocean)

**Task 3.1: Deployment Preparation**

- Prepare the application for deployment, considering scalability and high availability aspects. Document the steps required, including setting up any necessary infrastructure (such as load balancers or auto-scaling groups) and configuration management.

**Task 3.2: Deployment**

- Deploy the application on Digital Ocean and document the steps involved in the deployment process.

Evaluation Criteria:

- Code Quality and Structuring
- Functionality
- Error Handling
- Scalability and Performance Optimization
- Deployment
- Explanation and Communication of Code and Choices Made