wone

Task Stage Interview for Senior Full-Stack Engineer

Introduction

At Walking on Earth (WONE), measuring stress is at the core of everything we do. We have developed the WONE Method that's a 3 step process to measure and regulate stress:

1. **Measure**: we calculate stress with our WONE index, blending biometric data with a clinical assessment
2. **Recommend**: our AI recommendation systems sends you personalised interventions that meet you where you are
3. **Intervene**: 'micro moments of recovery' delivered by world-leading experts across neuroscience, mind and body based practices

As part of the interview process, we want to focus on the first step 'Measure'. To evaluate your skills and approach, we are asking you to create a full-stack solution that enables users to take the assessment component of the WONE index. We'll divide this into two parts:

1. A take-home task focused on creating the backend
2. A live coding front end task that builds on top of your backend solution

Part 2 will take place in-person, and be followed by a brief system design exercise.

Pre-requisites

Before you begin, it's important to gain a deeper understanding of our assessment.

Our assessment consists of three main sections - Resilience, Energy, and Activity - and a fourth optional section called Company. Each section includes a set of questions. Some of the questions in the main sections contribute to your overall assessment score, which is part of our stress measure known as the WONE Index.

For this task, you may find it useful to walk through our assessment via this link. Additionally, the assessment's structure is provided in Appendix 1: Assessment JSON. If you have any questions or need clarification before you start, please reach out to velislav@walkingonearth.com.

Part 1: Take-Home Task

Objective

Develop a simple backend application to meet the specified requirements.

Requirements

1. **Assessment Management:**
   - Store the assessment structure in a database.
   - Enable admin users to create and modify assessments.
   - Allow admin users to create and deploy new versions of the assessment.
2. **User Interaction:**
   - Allow users to take the assessment.
   - Enable users to resume their progress if they don't complete the assessment in one go.

Your Task

- Identify the necessary database objects and their relationships to meet the requirements.
- Create the data models and populate them using the data from Appendix 1: Assessment JSON.
- Build API endpoints that cover both admin (creating, modifying, deploying assessments) and user (taking and resuming assessments) interactions.

Deliverables

1. **Solution Description:**
   - Provide a brief overview of your approach and solution.
2. **Working Project:**
   - A functional backend application that can be run and tested.
3. **Setup Instructions:**
   - Clear, step-by-step instructions on how to set up and run your project.

Time-frame

Please aim to spend no more than **3 hours** on this exercise. The backend should include the essential features, focusing on clarity and scalability rather than completing every possible scenario.

Constraints

You may use **Python** or **JavaScript/TypeScript** as your programming language, and feel free to choose any frameworks, services, or databases you prefer.

Submission

Submit your solution as an archive of a Git repository or an online link (e.g., GitHub), along with instructions and documentation. Ensure your repository includes all necessary files and setup instructions to run your project.

Send your submission to velislav@walkingonearth.com at least one day before Part 2 of the interview.

Evaluation Criteria

1. **Code Quality:**
   - Clean and readable code, appropriate use of comments, and adherence to best practices.
2. **Solution Design:**
   - Correct identification of necessary objects and relationships, and scalability of the data models.
3. **Functionality:**
   - The backend meets the specified requirements, including error handling and edge cases.
4. **Documentation:**
   - Clear description of the approach and setup instructions.
5. **Testing (Bonus):**
   - Including automated tests for your API endpoints is a plus.

Part 2: Live Coding and Discussion

Objective

Develop a simple web application using **React** and integrate the backend you developed in Part 1.

Time-frame

The live coding exercise will last for **1 hour** followed by a **30-minute system design discussion**.

Detailed Description

1. **Live Coding:**
   - Create a basic React application that allows users to take the assessment.
   - Integrate it with your backend API to demonstrate user interaction with the assessment.
   - Focus on functionality rather than extensive UI styling.
2. **System Design Discussion:**
   - Discuss your backend architecture, design choices, and potential improvements.
   - Explore scalability, security, and other critical aspects of your solution.

Evaluation Criteria

1. **Coding Skills:**
    - Proficiency in **React** and **JavaScript/TypeScript**, with clear and efficient code.
2. **Problem-Solving:**
    - Ability to debug and solve issues on the fly, with a logical approach to integration challenges.
3. **System Design:**
    - Depth and clarity in discussing backend architecture, with considerations for scalability and security.
4. **Communication:**
    - Clear explanation of your thought process and design decisions, and responsiveness to feedback.

Summary

This task is designed to evaluate your ability to design, implement, and integrate a backend system with a front-end application. We are looking for clean code, thoughtful design, and effective communication.

If you have any questions, please reach out to velislav@walkingonearth.com.

Appendix 1: Assessment JSON

Link to download.