Candidate Evaluation Assignment

ASSIGNMENT OVERVIEW

Position: Software Engineer I
Department: Technology
Due Date: 20th May 2025 @ 5:00PM AST

Introduction

As part of the interview process for the Software Engineer I role at Nimble, candidates are required to create a sample application that tests their knowledge of cloud, backend and frontend technologies. There is no right or wrong solution to the assignment, and the objective is to evaluate the candidate's grasp of required technologies, problem solving skills, documentation and the ability to talk through their solution.

Part 1: Cloud-based RESTful API

Description

The first part of the evaluation assignment requires candidates to create a cloud-based RESTful API that allows for the management of data in a domain of the candidate's choice. This could include inventory for a small village shop, patient records for a medical practice, statistics for a local sports club or a music catalog for a popular radio station. The solution should make use of cloud infrastructure and non-local storage services and include instructions on how to deploy or access the API.

Requirements

1. **API Development:**
    - Develop a RESTful API that handles basic CRUD (Create, Read, Update, Delete) operations for a particular domain (See examples in description).
    - Implement authentication and authorization to secure the API endpoints.

2. **Cloud Infrastructure:**
    - Deploy the application using cloud-based infrastructure. The candidate is allowed to use any cloud provider, however preference is given to using AWS infrastructure.

3. **Non-Local Storage:**
    - Integrate a cloud-based database or a storage service to persistently store data. This could include a relational database, a NoSQL database, or a file storage mechanism.
    - The choice of storage should align with the project's requirements and be justifiable during the interview.

4. **Scalability and Reliability:**
    - Implement logging and monitoring strategies to keep track of application performance and errors.

5. **Documentation:**
    - Provide comprehensive documentation of the API, including endpoint descriptions, request/response examples, and authentication details.
    - Include instructions on how to set up the development environment, deploy the application to the chosen cloud provider, and test the functionality.

6. **Code Quality:**
    - Write clean, readable, and maintainable code.
    - Follow best practices in software development, including proper use of version control systems, preferably Git.

Evaluation Criteria

- **Correctness and functionality:** How well does the code meet the requirements?
- **Code quality and organization:** Is the code easy to read and maintain?
- **Use of cloud services:** Effective integration and use of cloud features, with extra points for AWS usage.
- **Documentation:** Clarity and completeness of API and setup documentation.
- **Scalability considerations:** Ability to explain how the application can handle future growth and increased load.

Part 2: Frontend Application

Description

The second part of the assignment requires candidates to create a frontend application. The application should consume the RESTful API delivered in Part 1 to provide users with real-time access and control over the managed data. If the candidate has not completed Part 1 of the assignment, any publicly available API may be used instead.

The application should, at a minimum, allow users to view and manipulate records. Additional functionality is appreciated, but not required. The project is designed to test the candidate's ability to implement a frontend application using MVVM design patterns and best practices in state management, routing, and UI/UX design.

Project Requirements

1. **Platform:**
    - The application can be developed as a web or mobile application. However, preference is given to mobile applications.

2. **Design Patterns:**
    - Implement the MVVM (Model-View-ViewModel) architecture to ensure clean separation of concerns within the application.

3. **Front-end Framework:**
    - Candidates are allowed to use any frontend development framework such as Ionic, React/React Native, Angular, or Flutter, with preference given to Flutter.

4. **RESTful API Integration:**
    - Integrate the API developed in Part 1 of the assignment. If you are unable to complete Part 1, use any publicly available API to fetch and manipulate data.
    - Ensure proper handling of API requests, data parsing, and error handling.

5. **State Management:**
    - Implement efficient state management using appropriate tools or libraries based on the chosen framework.
    - Manage application state to ensure smooth data flow and update between components.

6. **Routing:**
    - Implement routing to allow navigation between different views/screens within the application.

7. **Design and User Experience:**
    - Create a user-friendly and aesthetic interface, allowing for easy navigation and interaction.
    - Candidates can design their own UX/UI, focusing on usability and clarity.

8. **Documentation:**
    - Include documentation that describes the setup process, architecture, and any external libraries or services used.
    - Provide a brief explanation of design decisions, especially around state management and routing.

Evaluation Criteria

- Functionality: How well does the application fulfill the project requirements?
- Code Quality: Is the code well-structured, modular, and easy to understand?
- Use of MVVM: Proper implementation and adherence to the MVVM design pattern.
- Use of Framework: Effective use of the chosen frontend development tool.
- UI/UX Design: Quality of the application's user experience and design.
- Bonus Features: Quality and implementation of animations and theming.

Submission Guidelines

- Candidates should commit all their source code, configuration files, and deployment scripts for Parts 1 and 2 in a clearly organized repository.
- A README file explaining how to run both the frontend and backend, along with considerations for future enhancements or known limitations should be included.
- A link to the repository should be sent to engineering@getnimble.pro by the stated deadline.

Next Steps

Nimble's recruitment team will evaluate your submission and may invite you to a follow up interview to talk through your solution and ask additional questions to evaluate your technical and problem solving skills.