Engineering Intern Interview Preparation Guide

Thank you for applying for the Software Development Intern position at Kalen Technology Solution. We are excited to explore your skills and passion for software development. This document outlines the interview process, including a take-home assignment and expectations for the interview. Please Read And Understand Document for Better Results.

Point To Note

- **Code Scope**: **Focus on quality over quantity. A small, well-designed component (e.g., one API endpoint, one UI component, one mobile function) is sufficient, as long as it demonstrates your approach to the scenario**.

Interview Overview

Position: Software Engineering Intern (Backend, Frontend, or Mobile Development)

Structure:

- Introduction and behavioral questions
- Discussion of your submitted solution to the assignment
- Q&A and wrap-up

Preparation Tips:

- Choose one question from the three provided below based on your strengths and interests.
- Submit your solution, including source code, via GitHub 24 hours before your interview.
- Be prepared to explain your design, code, and trade-offs during the interview.

Assignment Instructions

To Before Interview, we ask you to complete the following:

- **Choose One Question**: Select one of the three scenario-based questions below (Backend, Frontend, or Mobile Development) that best aligns with your skills and interests.
- **Develop a Solution**:
    - Provide a written explanation (e.g., in a README.md) describing your approach, including:
        - System/UI/app design and architecture.
        - Technology choices and trade-offs.
        - How you addressed the specific requirements in the question.
    - Write source code to implement a key component of your solution (e.g., an API endpoint for backend, a UI component for frontend, or a location-tracking function for mobile). The code should be functional but does not need to be a complete application—focus on demonstrating your design and problem-solving.
    - Ensure your code is clear, well-structured, and includes comments where necessary.
- **Submission**:
    - Create a public or private GitHub repository for your solution.
    - Include your source code, README.md, and any necessary setup instructions.
    - Share the GitHub repository link by emailing [Your Email] at least **24 hours before your scheduled interview**
    - Late submissions may not be considered.
- **Interview Preparation**:
    - Bring your source code to the interview (e.g., have the repository accessible or a local copy).
    - Be ready to walk through your code, explain your design decisions, and answer questions about potential improvements or alternative approaches.

Interview Questions (Choose One)

Below are the three scenario-based questions. Choose **one** to solve and submit. Each question requires a written explanation and source code for a key component of the solution.

Backend Development

Scenario: You are tasked with designing a notification system for an e-commerce platform like Amazon. The system must handle sending emails, SMS, and push notifications to millions of users for order updates (e.g., order placed, shipped, delivered). The system should be scalable, reliable, and maintainable.

Question: Design the backend architecture for this notification system. Describe the key components, how they interact, and the technologies or patterns you would use. Address the following:

- How would you ensure the system can handle high throughput during peak times (e.g., Black Friday)?
- How would you handle failures?
- What database structure or message queue system would you use, and why?

Deliverables:

- **Written Explanation**: Describe your architecture, technology choices, and how you address scalability and fault tolerance.
- **Source Code**: Implement a key component, such as an API endpoint to process notifications or a function to handle message queue retries. Use any programming language (e.g., Python, Java, Node.js).

Evaluation Focus: Scalable and modular system design, fault tolerance, and clean, functional code.

Frontend Development

Scenario: You are building a dashboard for a healthcare application used by doctors to monitor patient vitals (e.g., heart rate, blood pressure) in real-time. The dashboard must display data clearly, be responsive across devices, and allow doctors to filter data by patient or time range.

Question: Design the frontend for this dashboard. Describe the UI/UX design principles you would apply, the framework or tools you would use, and how you would structure the code. Address the following:

- How would you ensure the dashboard is intuitive and visually appealing for doctors under time pressure?
- How would you handle real-time data updates efficiently in the UI?
- What steps would you take to make the dashboard accessible and responsive on both desktop and tablet devices?

Deliverables:

- **Written Explanation**: Describe your UI/UX design, framework choice (e.g., React, Vue), and approach to real-time updates and responsiveness.
- **Source Code**: Implement a key component, such as a filter dropdown or a real-time chart, using HTML/CSS/JavaScript or a frontend framework.

Evaluation Focus: Intuitive UI/UX, accessibility, responsiveness, and efficient, well-structured code.

Mobile Development

Scenario: You are developing a ride-sharing mobile app like Uber for both iOS and Android. The app must allow users to book rides, view driver locations in real-time on a map, and receive updates about ride status (e.g., driver assigned, en route, completed).

Question: Propose a solution for implementing the real-time driver location tracking feature in the mobile app. Describe the architecture, including client-side and server-side interactions, and the technologies you would use. Address the following:

- How would you optimize battery and data usage for real-time location updates on the mobile device?
- How would you ensure a smooth and accurate map experience for users?
- What cross-platform development approach (e.g., native, Flutter, React Native) would you choose, and why?

Deliverables:

- **Written Explanation**: Describe your architecture, platform choice (e.g., native Swift/Kotlin, Flutter), and optimization strategies for battery and map performance.
- **Source Code**: Implement a key component, such as a function to throttle location updates or a map integration module, in your chosen language/framework.

Evaluation Focus: Resource-efficient design, smooth map performance, and justified platform choice with clean code.

Additional Notes

- **GitHub Submission**: Ensure your repository is accessible (public or shared with [Your GitHub Username]). Include a README.md with setup instructions (e.g., how to run your code) and your written explanation.
- **Code Scope**: **Focus on quality over quantity. A small, well-designed component (e.g., one API endpoint, one UI component, one mobile function) is sufficient, as long as it demonstrates your approach to the scenario**.
- **Tools for Interview**: Be prepared to share your screen or walk through your code in a text editor or IDE during the interview. We may use a shared whiteboard (e.g., [specify tool, like Excalidraw]) for diagrams.
- **Questions for Us**: Prepare questions about the role, team, or company to discuss during the interview.
- **Contact**: If you have questions about the assignment or interview, contact **Adrian Sengondo Mvungi** at asengondo@kalen.co.tz

- Minimize AI Use

We look forward to reviewing your solution and discussing your approach during the interview. Best of luck!