Fullstack Engineer Hiring Task

Overview:

In this task, you will build a Customer Support App using Django, Angular, TailwindCSS, Python, and TypeScript. This task will assess your ability to design a backend, build APIs, and implement basic frontend functionality.

The task will be evaluated based on both the submitted code and a follow-up pair programming session with me.

Part 1: Take-home Task

Objective:

Develop a Customer Support App that allows users to:

1. List tickets: View all existing support tickets.
2. Create tickets: Add new support tickets.
3. Resolve tickets: Mark tickets as resolved.
4. Delete tickets: Remove tickets.

Backend (Django):

- Design a `Ticket` model with your choice of fields (e.g., title, description, status, created_at).
- Implement CRUD APIs to support the following operations:
  1. List all tickets.
  2. Create a new ticket.
  3. Mark a ticket as resolved.
  4. Delete a ticket.
- Use SQLite for simplicity.
- Write basic tests for the backend to ensure the API works correctly.

Frontend (Angular):

- Basic UI using Angular and TailwindCSS:

  1. A ticket list page that displays all tickets.

     - Show key fields like title, status, and created_at.

     - Include buttons to resolve and delete tickets.

  2. A simple form to create new tickets (only basic fields like title and description).

- Use Angular services to call the backend API.

- Ensure the frontend updates the list dynamically after actions.

Optional (Bonus Points):

- Add basic frontend tests to validate UI logic.

- Use TypeScript effectively for better type safety.

Deliverable:

Push your code to a public GitHub repository with setup instructions in the README.md.

Part 2: Pair Programming Session

After submitting your task, you will participate in a pair programming session with me.

What to Expect:

- Code Review: We will discuss your design choices and thought process.

- Task Extension or Refactoring: You may be asked to extend or refactor the functionality.

  Example tasks:

  - Adding filters or search to the ticket list.

  - Adding pagination for large ticket sets.

  - Discussing potential code optimizations.

- Test Coverage: We may extend or discuss the backend test strategy.

Evaluation Criteria:

- Backend Implementation: Code organization, correctness, and test coverage.
- Frontend Implementation: Minimal but functional UI with proper API integration.
- Usability and Functionality: A working app with the ability to list, create, resolve, and delete tickets.
- Code Quality: Readable, maintainable code following best practices.
- Communication and Problem-solving: Your ability to explain and improve your code during the pair programming session.

Timeline & Submission:

Submit your GitHub repository link within 48 hours of receiving the task.

We will follow up to schedule the pair programming session within a few days of submission.

Stack:

- Backend: Django (Python)
- Frontend: Angular (TypeScript)
- Styling: TailwindCSS
- Database: SQLite (for simplicity)

We look forward to your submission!