HTX xData Technical Test Question – Software Engineer

The purpose of this test is to assess if you have the prerequisite technical ability for the role.

Please follow closely to the Test Instruction

- You can make assumptions to any information not specified within the Test Instruction.
- You can make references to any other technical literature as required.
- Reach out to us if you have any queries via email.
- Submit Deliverables to the sender of this test.
    - **The deliverable for this assignment is a link to your codebase (publicly accessible) for your attempt on the assignment.**
    - All submitted artifacts must be executable without errors. If a Python environment setup is required, please include the setup scripts in the submission. Failure to comply will result in the submission being considered incomplete.
    - You have **a total of 7 calendar days** to complete the list of tasks (1 to 5) below by yourself, upon receiving the email.
    - You will need to document and comment your code properly and state any assumptions made within each task.

Tasks

1. Create a public git repository e.g., https://github.com/fred/myrepo. You may use any of the publicly available repositories like GitHub, Gitlab, etc.

2. Create a directory called backend in your repository for all backend-related code.

    a) Implement a RESTful API using a Python web framework (e.g., Flask, FastAPI) with the following endpoints:

        i. GET /health: Returns the status of the service.
        ii. POST /transcribe: Accepts audio files and returns transcriptions. Files uploaded should be modified to be unique, if the file name already exists in the backend.
        iii. GET /transcriptions: Retrieves all transcriptions from the database.
        iv. GET /search: Query for transcriptions based on the audio file name. (You are given 3 audio files to use for this assignment)

    b) Integrate the Whisper speech recognition model:
        i. Use the openai/whisper-tiny model from Hugging Face.
        ii. Implement necessary audio preprocessing

c) Search Feature:
    i. Use SQLite as the primary database for storing the audio file name, transcribed text and created timestamp.

d) Containerise the backend service.

3. Create a directory called frontend in your repository for all frontend-related code.

    a) Develop a single-page application using a modern JavaScript framework (e.g., React, Vue.js, Angular).

    b) Implement the following features for the UI (User Interface):
        i. File upload for single / batched audio file transcription.
        ii. Display a list of all transcriptions from the database.
        iii. Search functionality to retrieve transcriptions based on the **audio file name** and display results in the UI.

    c) Containerise the frontend service.

4. Testing

    a) Write **three** unit tests for the backend and frontend each. Include instructions on how to run the tests in the README.

5. Based on the full-stack application you developed, create an architecture diagram that outlines the following components:
    a) Frontend,
    b) Backend,
    c) Database,
    d) External Services, if any.

    Your answer can be saved as `architecture.pdf` under the main repository. Please explain your diagram, citing assumptions and considerations.