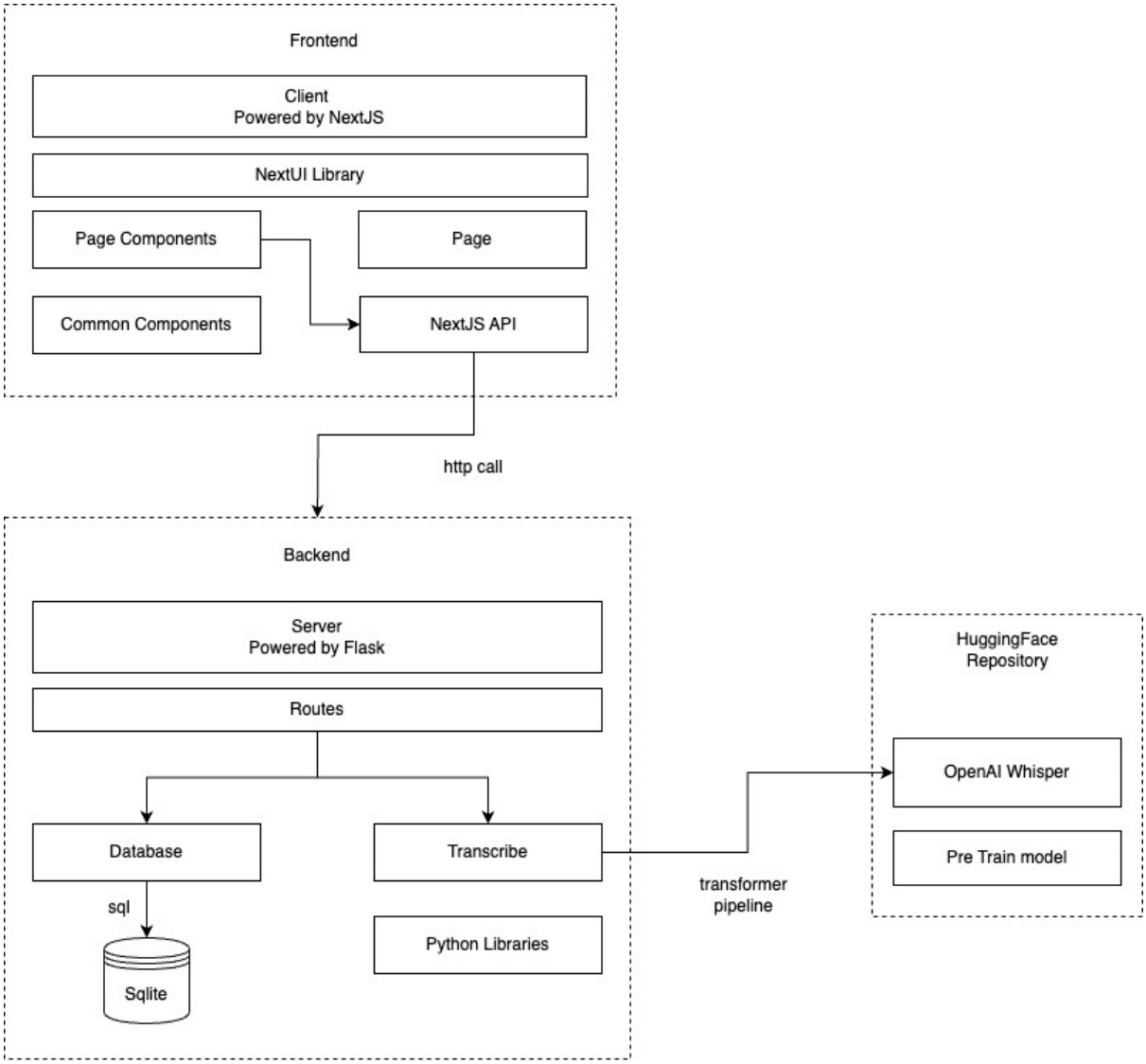
The assignment implements a basic transcription system that is using the Flask framework for the backend and Next.js for the frontend, with SQLite as the database. The architecture is depicted in the below diagram.



Backend

Flask was selected for the backend due to its ease of setup, making it comfortable for prototyping with no or little configuration. It also provides the flexibility to transition to more robust frameworks like FastAPI should the prototype evolve into a full-fledged system requiring high performance and asynchronous capabilities.

Frontend

Next.js was chosen for the frontend for its simplicity and streamlined development process compared to standalone React. Its minimal configuration requirements and capability to function as either a client-side application or a full-stack framework (using Server-Side Rendering) make it a versatile choice for rapid prototyping.

Database

SQLite serves as the database in this prototype for its simplicity and ease of integration with the backend. As the system matures and requires a more scalable and collaborative environment with multiple team members, it can be seamlessly replaced with managed standalone databases like MySQL, PostgreSQL, MS SQL, or MongoDB.

Architectural Design

The system adopts a pragmatic approach, prioritizing the development of the core logic: transcribing audio files into text. Once the transcription logic is validated and capable of handling multiple audio permutation, the architecture can be incrementally refactored for scalability. For instance:

- The database can be migrated to a managed standalone solution for multi-user collaboration on schema design.
- The backend framework can be replaced with another framework.

While the ideal design for security and scalability is to adheres to the strict three-tier architecture—separating the presentation, business, and data layers, this implementation that I adopted focuses on functionality first to get buy-in from stakeholders, before introducing next few layers of complexity.