Project Documentation for Healthcare Translation App

Based on your workspace, I can see you have a multilingual healthcare translation web application with both frontend and backend components. Here's an overview of the project:

Project Structure

The project is organized into two main components:

- Frontend - A web interface built with HTML, CSS, and JavaScript
- Backend - A Python Flask application that handles translation services

Key Components

Frontend

The frontend provides:

- Speech-to-text functionality using the Web Speech API
- Language selection for both input and output languages
- Simple interface for healthcare professionals to use in real-time
- Audio playback of translated text

Backend

The backend handles:

- Processing translation requests through a Flask API
- Integrating with Groq's LLama 3 70B model API for high-quality translations
- Environment variable management for secure API key storage

Deployment

The application is deployed on Vercel:

- Frontend configuration is defined in frontend/vercel.json
- Backend configuration is defined in backend/vercel.json
- The application is accessible at: https://healthcaretranslationapp.vercel.app/

How It Works

1. Users select their input language and speak into the microphone

2. The Web Speech API transcribes the speech to text
3. Users select the desired output language and click "Translate"
4. The frontend sends a request to the backend API
5. The backend uses the Groq API (LLama 3 70B) to translate the text
6. The translated text is returned to the frontend and can be played back as audio

Technical Details

- The backend uses Flask with CORS support to handle cross-origin requests
- API keys are stored in a .env.local file (excluded from version control)
- The application is deployed on Vercel with appropriate routing configurations
- Translation is performed using a large language model with temperature set to 0.3 for consistent results

Running Locally

1. Clone the repository
2. Create a .env.local file with your GROQ_API_KEY
3. Run the frontend by opening index.html in a browser
4. Run the backend with python app.py in the backend directory