Project Documentation

# Conversational AI for Education (Metterian)

A GPT-2 based chatbot for English learners with voice recognition, quizzes, and speaking practice.

## 1. Project Overview

This project aims to build a Conversational AI chatbot to support English language learners by enabling interactive conversation, quizzes, and speaking practice. The chatbot leverages pretrained GPT-2 for natural language generation, integrates speech recognition for spoken inputs, and provides text-to-speech for responses.

## 2. Objectives

* Develop an AI-powered educational assistant for English learners.
* Provide real-time interaction via text and voice.
* Enhance speaking and comprehension skills with practice sessions.
* Integrate quiz-based learning for grammar and vocabulary.
* Make the chatbot accessible across devices.

## 3. Technologies Used

* Python 3.9+ - Core programming language
* PyTorch / TensorFlow - Framework for GPT-2 fine-tuning
* Hugging Face Transformers - Pretrained GPT-2 model & tokenizer
* SpeechRecognition - Voice input (speech-to-text)
* gTTS / pyttsx3 - Text-to-speech output
* Flask / FastAPI - Backend API for chatbot service
* React.js / Streamlit - Frontend user interface
* SQLite / Firebase - User data & quiz score storage
* Docker - Containerization for deployment
* Heroku / AWS / GCP - Cloud hosting for production

## 4. System Architecture

Workflow:  
1. User Input (Text or Speech)  
2. Preprocessing (Tokenization & cleaning)  
3. GPT-2 generates response  
4. Post-processing  
5. Output (Text + TTS)  
6. Database Update (Logs, quiz scores, progress)

## 5. Step-by-Step Guide

1. Step 1: Environment Setup (Python virtual environment, install dependencies)
2. Step 2: Load GPT-2 Model
3. Step 3: Add Speech-to-Text functionality
4. Step 4: Add Text-to-Speech functionality
5. Step 5: Integrate Quiz Module
6. Step 6: Build UI (Streamlit / React.js frontend)
7. Step 7: Deploy (Docker, Heroku/AWS/GCP)

## 6. Features Implemented

* Conversational chatbot using GPT-2
* Text & voice input supported
* Text-to-speech response output
* Vocabulary & grammar quiz module
* Progress tracking & database logging
* User-friendly interface

## 7. Process & Methodology

* Requirement Analysis – Defined use cases for English learners.
* Model Selection – Chose GPT-2 for balance of speed & fluency.
* Integration – Combined GPT-2 with STT (SpeechRecognition) & TTS (gTTS).
* Quiz Development – Designed MCQ quizzes for practice.
* Testing & Evaluation – Conducted user testing for fluency & accuracy.
* Deployment – Packaged chatbot for web/mobile use.

## 8. Limitations

* GPT-2 has limited context (short memory).
* May produce grammatically incorrect or irrelevant answers sometimes.
* Requires internet for speech recognition APIs.
* Quiz database currently limited in scope.

## 9. Next Steps / Future Work

* Fine-tune GPT-2 on English learning datasets (ESL, TOEFL, Duolingo).
* Add BERT-based grammar correction.
* Enhance quiz variety: listening, comprehension, speaking tasks.
* Build mobile app version (React Native / Flutter).
* Integrate personalized learning paths (AI tracks weak areas).
* Upgrade to GPT-3.5 / GPT-4 for better fluency.