🎤Speech-to-Text Transcription Project – Beginner's Guide

This document explains **step-by-step** how our Speech-to-Text project works. It is written for beginners who want to study, understand, and improve the project.

Project Overview

The goal of this project is to: - Capture audio input (either **live from a microphone** or **from uploaded audio files**). - Convert speech into **text transcription** using Google Speech Recognition API. - Allow users to **save transcripts** in multiple formats: `.txt`, `.docx`, `.pdf`. - Provide a **Streamlit-based UI** for easy interaction.

Libraries Used

1. speech_recognition (sr)

- **Purpose:** Handles capturing audio and converting speech to text.
- **Functions used:**
- `sr.Recognizer()` → Creates a recognizer object.
- `sr.Microphone()` → Captures audio from microphone.
- `recognizer.listen(source, timeout, phrase_time_limit)` → Listens and records audio.
- `recognizer.adjust_for_ambient_noise(source, duration)` → Handles background noise.
- `recognizer.recognize_google(audio, language)` → Sends audio to Google API and returns transcription.

- **Arguments:**

- `timeout`: Max waiting time before speech starts.
- `phrase_time_limit`: Max duration of captured speech.
- `language`: Language code like `en-US`, `hi-IN`, `te-IN`.

---

2. streamlit (st)

- **Purpose:** Creates an interactive web interface.
- **Functions used:**
- `st.title()` → Page title.
- `st.info(), st.success(), st.error()` → Display messages with icons.
- `st.button(), st.selectbox(), st.file_uploader()` → User inputs.

- `st.download_button()` → Download files.
- `st.session_state` → Stores values across page reloads.
- `st.tabs()` → Creates tabbed interface.

---

3. datetime

- **Purpose:** Add timestamps to transcripts.
- **Function used:**
- `datetime.now().strftime("%Y-%m-%d %H:%M:%S")` → Formats current time.

---

4. uuid

- **Purpose:** Creates unique IDs for users/files.
- **Function used:**
- `uuid.uuid4().hex[:6]` → Generates a 6-character random ID.

---

5. docx (python-docx)

- **Purpose:** Creates Word documents.
- **Functions used:**
- `Document()` → Create a new Word document.
- `doc.add_paragraph(text)` → Add text.
- `doc.save(file_name)` → Save as `.docx`.

---

6. reportlab

- **Purpose:** Generates PDF files.
- **Modules used:**
- `SimpleDocTemplate(file_name)` → Creates PDF document.
- `Paragraph(text, styles["Normal"])` → Add text.
- `getSampleStyleSheet()` → Provides text styles.

Code Workflow

1. App starts

- Streamlit loads UI (`st.title`, language selector, name input).

2. Capture User's Name

- `get_user_name()` uses microphone.

- Recognized name + unique ID saved in `st.session_state.user_name`.

3. Choose Action (Tabs)

- **Tab 1: Upload Audio** → User uploads `.wav` or `.mp3` file.
- **Tab 2: Live Recording** → Capture audio with microphone.

4. Transcription

- Audio is processed by Google API (`recognizer.recognize_google`).
- Transcript saved in `st.session_state.transcript`.

5. Save & Download

- User selects file type from dropdown.
- Transcript is saved as `.txt`, `.docx`, or `.pdf`.
- `st.download_button()` allows downloading.

🏓 Shell Commands

1. **Install libraries**:

```
pip install streamlit speechrecognition pyaudio python-docx reportlab
```

2. **Run the app**:

```
streamlit run using_Streamlit.py
```

⚠️ **Note:** Running with `python using_Streamlit.py` will cause errors (Streamlit needs `streamlit run`).

1. **Fix pip not recognized issue**:
2. Add to **Environment Variables → Path**:

```
C:\Users\<YourName>\AppData\Roaming\Python\Python311\Scripts
```

Common Errors & Fixes

1. `ModuleNotFoundError: No module named 'speech_recognition'`

2. Fix: Install with `pip install SpeechRecognition`.
3. `pipwin` **errors when installing PyAudio**
4. Fix: Use `pip install pipwin` → `pipwin install pyaudio`.
5. `WaitTimeoutError`
6. Cause: No speech detected.
7. Fix: Increase `timeout` and `phrase_time_limit`.
8. `charmap codec can't encode characters`
9. Cause: Non-UTF-8 characters in transcript.
10. Fix: Save with `encoding="utf-8"`.
11. `unknown encoding: utf-18`
12. Cause: Typo, correct is `utf-8`.
13. **Transcript lost on reload**
14. Fix: Store in `st.session_state.transcript`.

🖌️ Future Improvements

- Add **multi-user login system** instead of only unique IDs.
- Support **continuous recording** instead of short phrases.
- Add **translation support** for multiple languages.
- Improve noise handling with external libraries like `noisereduce`.

Key Takeaways

- **speech_recognition** handles audio input and transcription.
- **Streamlit** provides the UI and session handling.
- **docx & reportlab** help save transcripts in Word/PDF.
- **uuid & datetime** ensure uniqueness and timestamping.
- **Session state** solves the biggest issue of data loss on refresh.

🈶With this knowledge, you can **understand, run, debug, and expand** the project easily as a beginner.