

Ex.No: 7 Date:	TEXT – TO – SPEECH SYSTEM USING GTTS
---------------------------------	---------------------------------------------

TEXT – TO – SPEECH SYSTEM USING GTTS**AIM:**

To create a web-based Text-to-Speech (TTS) application using gTTS and Streamlit that allow users to input text, select a language and voice speed, and convert it into audible speech.

THEORY:

- Text-to-Speech (TTS) is a technology that converts written text into spoken words using speech synthesis.
- gTTS (Google Text-to-Speech): A Python library and API interface to Google Translate's TTS engine.
- Streamlit: A Python framework to quickly build interactive web apps for data and machine learning applications.
- Users can choose different languages and voice speeds, providing flexibility in the generated audio.
- The generated audio is played directly in the web interface using Streamlit's st.audio() component.

IMPLEMENTATION:**Installation of Packages:**

```
pip install streamlit
```

```
pip install Gtts
```

ALGORITHM:

Step 1: Import necessary libraries: streamlit, gTTS, tempfile, and base64.

Step 2: Set up Streamlit page with title, layout, and icon.

Step 3: Initialize session state to store user input.

Step 4: Create UI elements:

Step 5: Text area for user input.

Step 6: Sidebar to select language.

Step 7: Sidebar to select voice speed.

Step 8: Button to trigger speech conversion.

Step 9: Handle button click:

Step 10: Check if input text is empty; display a warning if yes.

Step 11: If not, generate speech using gTTS with the chosen language and speed.

Step 12: Save speech to a temporary MP3 file.

Step 13: Play audio in the app using st.audio().

Step 14: Display success message after generating audio.

Step 15: (Optional) Clear the text input for a new entry.

CODE:

```
import streamlit as st
from gtts import gTTS
import tempfile
import base64
# Page setup
st.set_page_config(page_title="Text-to-Speech Chatbot", page_icon="📢",
layout="centered")
# Initialize session state
if "text_input" not in st.session_state:
    st.session_state.text_input = ""
```



```
# Title
```

```
st.markdown(
```

```
"<h1 style='text-align:center; color:#4CAF50;'>🌐 Text-to-Speech Chatbot 🎤 </h1>"
```

```
"<p style='text-align:center; color:gray;'>Type text, select a language & speed, then hear  
it!</p>,
```

```
unsafe_allow_html=True
```

```
)
```

```
# Sidebar
```

```
languages = {
```

```
    "English": "en", "French": "fr", "Spanish": "es", "German": "de",
```

```
    "Hindi": "hi", "Tamil": "ta", "Italian": "it", "Chinese": "zh-cn",
```

```
    "Japanese": "ja", "Arabic": "ar", "Russian": "ru", "Portuguese": "pt"
```

```
}
```

```
selected_lang = st.sidebar.selectbox("🌐 Choose Language", list(languages.keys()))
```

```
voice_speed = st.sidebar.radio("🎵 Voice Speed", ["Slow", "Normal", "Fast"])
```

```
# Text input
```

```
user_text = st.text_area("💬 Enter text here:", key="text_input") # bind to session_state
```

```
# Button
```

```
if st.button("🔊 Convert to Speech"):
```

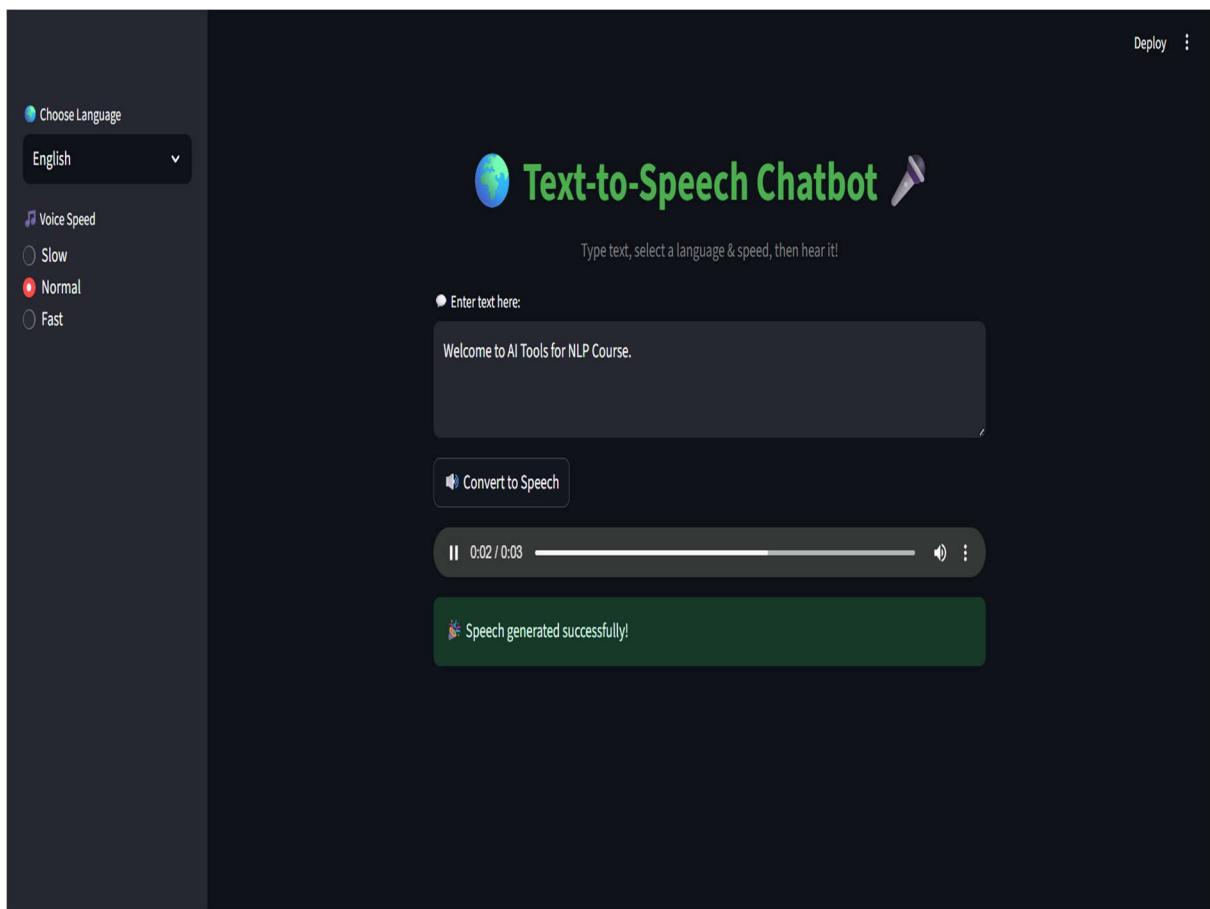
```
    if user_text.strip() == "":
```

```
        st.warning("⚠ Please enter some text!")
```

```
    else:
```

```
        # Generate TTS
```

```
tts = gTTS(
```

OUTPUT:

```

text=user_text,
lang=languages[selected_lang],
slow=True if voice_speed == "Slow" else False
)
# Save temp file
with tempfile.NamedTemporaryFile(delete=False, suffix=".mp3") as tmp_file:
    tts.save(tmp_file.name)
    audio_file = tmp_file.name
    st.audio(audio_file, format="audio/mp3")
    st.success("🔊 Speech generated successfully!")
# Clear text input by resetting session_state variable

```

RUBRICS:

PROBLEM UNDERSTANDING AND OBJECTIVE CLARITY (20)	IMPLEMENTATION AND CODING (40)	OUTPUT EVALUATION AND DISCUSSION (30)	VIVA (10)	TOTAL (100)

RESULT:

Thus, a web-based Text-to-Speech (TTS) application using gTTS and Streamlit that allow users to input text, select a language and voice speed, and convert it into audible speech has been executed successfully.