

```

import streamlit as st
import google.generativeai as genai
from pypdf import PdfReader
import edge_tts
import asyncio
import nest_asyncio

nest_asyncio.apply()

st.set_page_config(page_title="Fazaa", page_icon="🌸", layout="centered")

st.markdown("""
<style>
@import url('https://fonts.googleapis.com/css2?family=Tajawal:wght@400;700&di
html, body, [class*="css"] { font-family: 'Tajawal', sans-serif; direction: r
.stButton>button {
    width: 100%; border-radius: 25px; height: 3.5em;
    background-color: #8A1538; color: white; border: none; font-weight: bold;
}
.stButton>button:hover { background-color: #FCE4EC !important; color: #8A1538
h1, h2, h3 { color: #8A1538; text-align: center; }
.stAudio { margin-top: 20px; }
</style>
""", unsafe_allow_html=True)

try:
    API_KEY = st.secrets["GEMINI_API_KEY"]
except Exception:
    API_KEY = ""

genai.configure(api_key=API_KEY)

async def generate_natural_audio_async(text, output_file):
    voice = "ar-SA-ZariyahNeural"
    communicate = edge_tts.Communicate(text, voice, rate="+10%")
    await communicate.save(output_file)

def generate_audio(text, output_file):
    loop = asyncio.get_event_loop()
    loop.run_until_complete(generate_natural_audio_async(text, output_file))

st.markdown("<h1>Fazaa - Tasolfa</h1>", unsafe_allow_html=True)
st.markdown("<p style='text-align: center;'>Listen to your lecture as a conversat

```

```

uploaded_file = st.file_uploader("Upload PDF lecture file", type="pdf")

if uploaded_file:
    reader = PdfReader(uploaded_file)
    full_text = "".join([p.extract_text() for p in reader.pages if p.extract_text])

if full_text:
    full_text = full_text[:15000]

st.success("File uploaded! Choose what you want:")
col1, col2, col3 = st.columns(3)

base_prompt = (
    "Convert the following academic text into a long detailed conversatio"
    "Nora explains with intelligence and enthusiasm. "
    "Manal asks smart questions and wants to understand the details. "
    "Use a deep engaging Saudi Najdi conversational style. "
    "Explain everything in the text. "
    "Do not write Nora or Manal labels, make it a connected dialogue like"
    "Text: " + full_text
)

final_prompt = ""
if col1.button("Saudi Arabic Style"):
    final_prompt = base_prompt
if col2.button("Translate then Saudi Style"):
    final_prompt = "Translate the following text to Arabic then " + base_
if col3.button("English Session"):
    final_prompt = (
        "Create a deep-dive conversation between two students, Nora and M"
        "discussing this PDF in a friendly English style. Text: " + full_
    )

if final_prompt:
    with st.spinner("Nora and Manal are preparing the session..."):
        try:
            model = genai.GenerativeModel("gemini-1.5-flash")
            response = model.generate_content(final_prompt)
            generated_script = response.text

            audio_file = "faza_audio.mp3"
            generate_audio(generated_script, audio_file)

            st.markdown("---")
            st.markdown("### Ready! Listen to Fazaa:")
            st.audio(audio_file)

```

```
        with open(audio_file, "rb") as f:
            st.download_button("Download MP3", f, file_name="nora_man.mp3")

    except Exception as e:
        st.error("Error: " + str(e))

else:
    st.error("File cannot be read.")
```