FINAL YEAR PROJECT PROTOTYPE

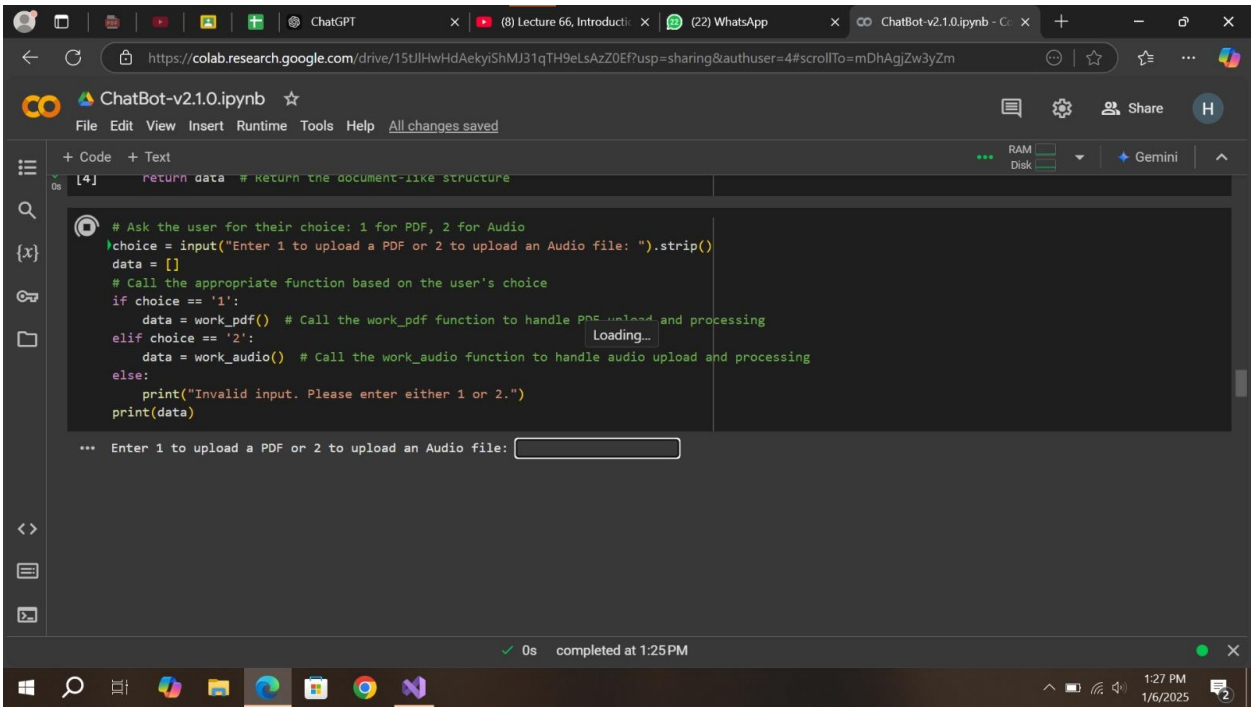
BCSF21M011	Muhammad Haider Hamayoun
BCSF20M015	Tuba Saleem
BCSF21M018	Husna Sarwar
BCSF21M003	Muhammad Umar Qureshi

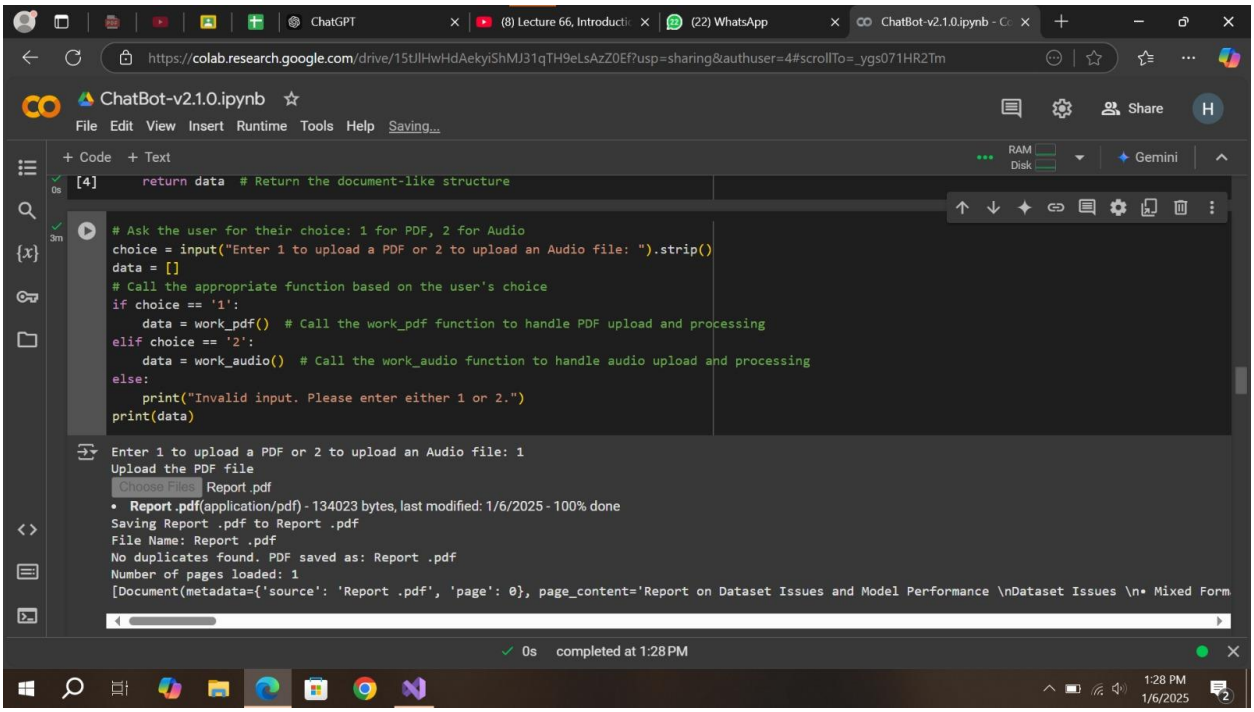
Detailed Prototype Description

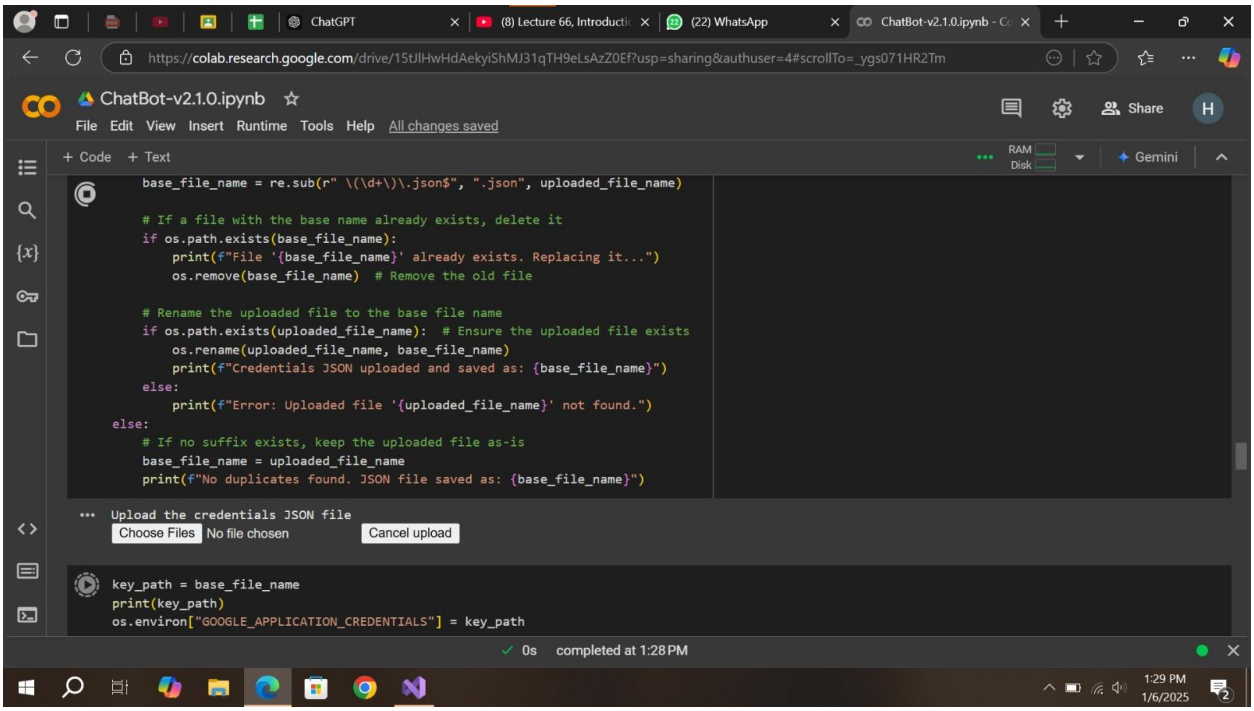
The **Colab Chatbot** prototype demonstrates the functionalities of a voice-driven AI chatbot for early childhood education. This chatbot allows users to interact in both **Urdu** and **English**, processes uploaded **PDFs** or **MP4 files**, and provides responses through **voice-to-text** or **voice-to-voice** interactions.

Input Mechanisms

1. **Voice Input:** Users click on the microphone button and speak their queries. The chatbot processes the spoken input using speech-to-text technology.
2. **File Upload:** Users can upload PDFs or MP4 files. The chatbot extracts content (text or audio) and generates appropriate responses.
3. **Multilingual Input:** Users can ask questions in either **Urdu** or **English**, and the system processes them seamlessly.

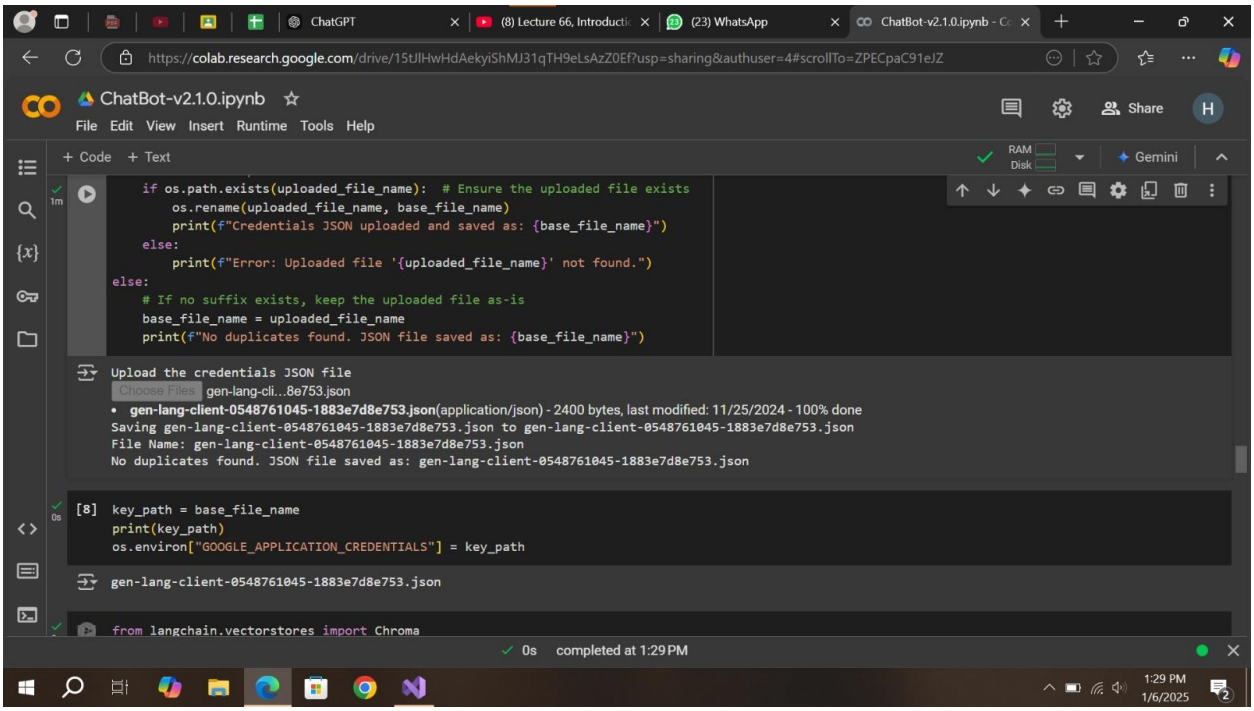

ChatBot-v2.1.0.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Share
+ Code + Text
RAM
Disk
Gemini
[4] return data # Return the document-like structure
# Ask the user for their choice: 1 for PDF, 2 for Audio
choice = input("Enter 1 to upload a PDF or 2 to upload an Audio file: ").strip()
data = []
# Call the appropriate function based on the user's choice
if choice == '1':
data = work_pdf() # Call the work_pdf function to handle PDF upload and processing
elif choice == '2':
data = work_audio() # Call the work_audio function to handle audio upload and processing
else:
print("Invalid input. Please enter either 1 or 2.")
print(data)
Loading...
Enter 1 to upload a PDF or 2 to upload an Audio file:
0s completed at 1:25PM
1:27 PM
1/6/2025

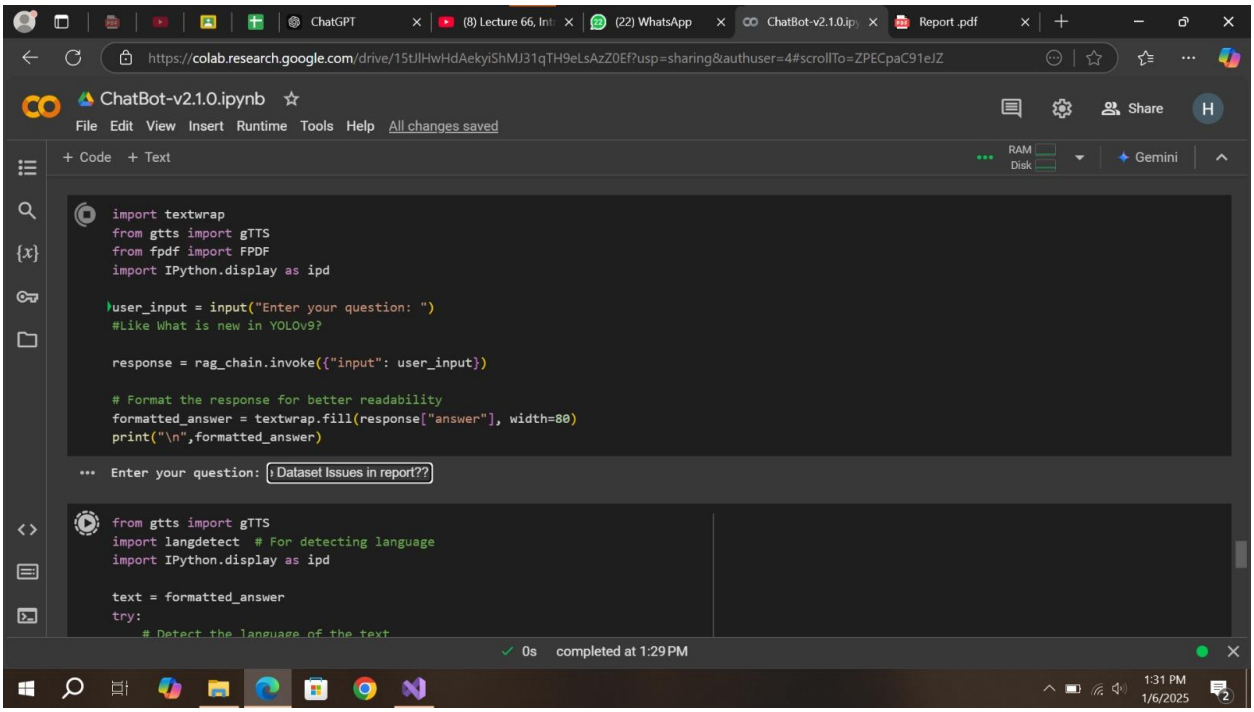

ChatBot-v2.1.0.ipynb
File Edit View Insert Runtime Tools Help Saving...
Share
+ Code + Text
RAM
Disk
Gemini
[4] return data # Return the document-like structure
# Ask the user for their choice: 1 for PDF, 2 for Audio
choice = input("Enter 1 to upload a PDF or 2 to upload an Audio file: ").strip()
data = []
# Call the appropriate function based on the user's choice
if choice == '1':
data = work_pdf() # Call the work_pdf function to handle PDF upload and processing
elif choice == '2':
data = work_audio() # Call the work_audio function to handle audio upload and processing
else:
print("Invalid input. Please enter either 1 or 2.")
print(data)
Enter 1 to upload a PDF or 2 to upload an Audio file: 1
Upload the PDF file
Choose Files Report .pdf
Report .pdf(application/pdf) - 134023 bytes, last modified: 1/6/2025 - 100% done
Saving Report .pdf to Report .pdf
File Name: Report .pdf
No duplicates found. PDF saved as: Report .pdf
Number of pages loaded: 1
[Document(metadata={'source': 'Report .pdf', 'page': 0}, page_content='Report on Dataset Issues and Model Performance \nDataset Issues \n• Mixed Form
0s completed at 1:28PM
1:28 PM
1/6/2025

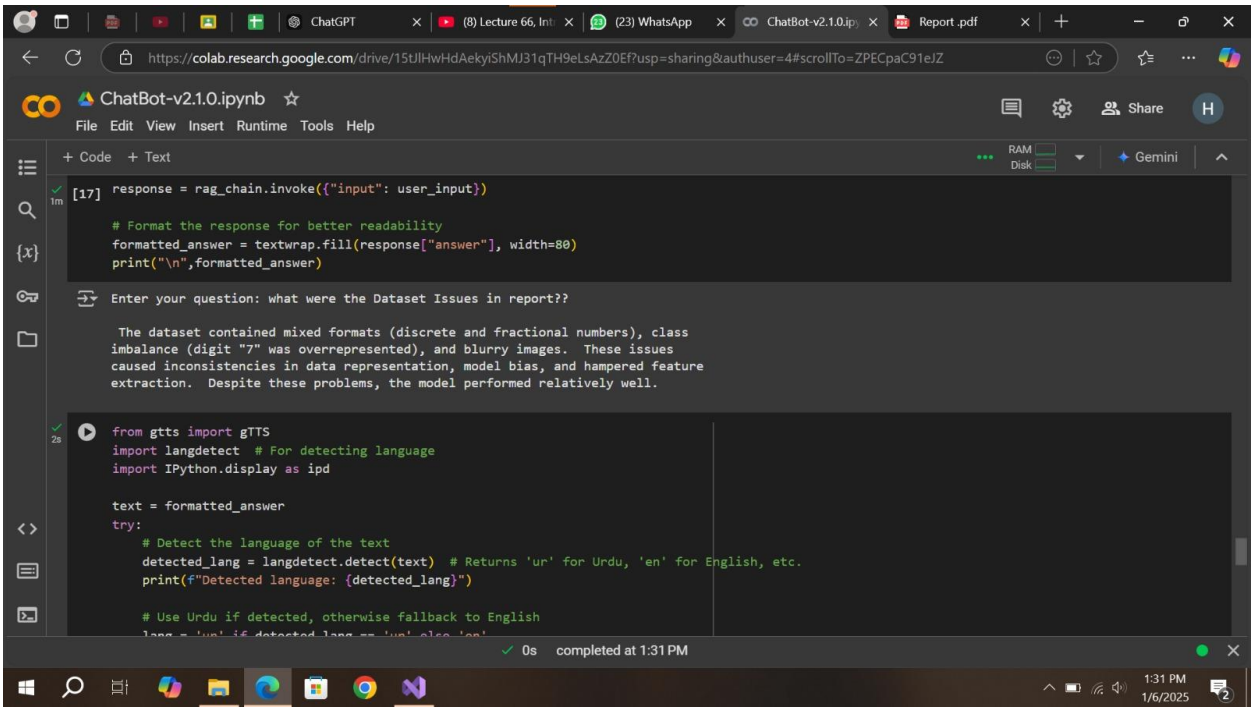


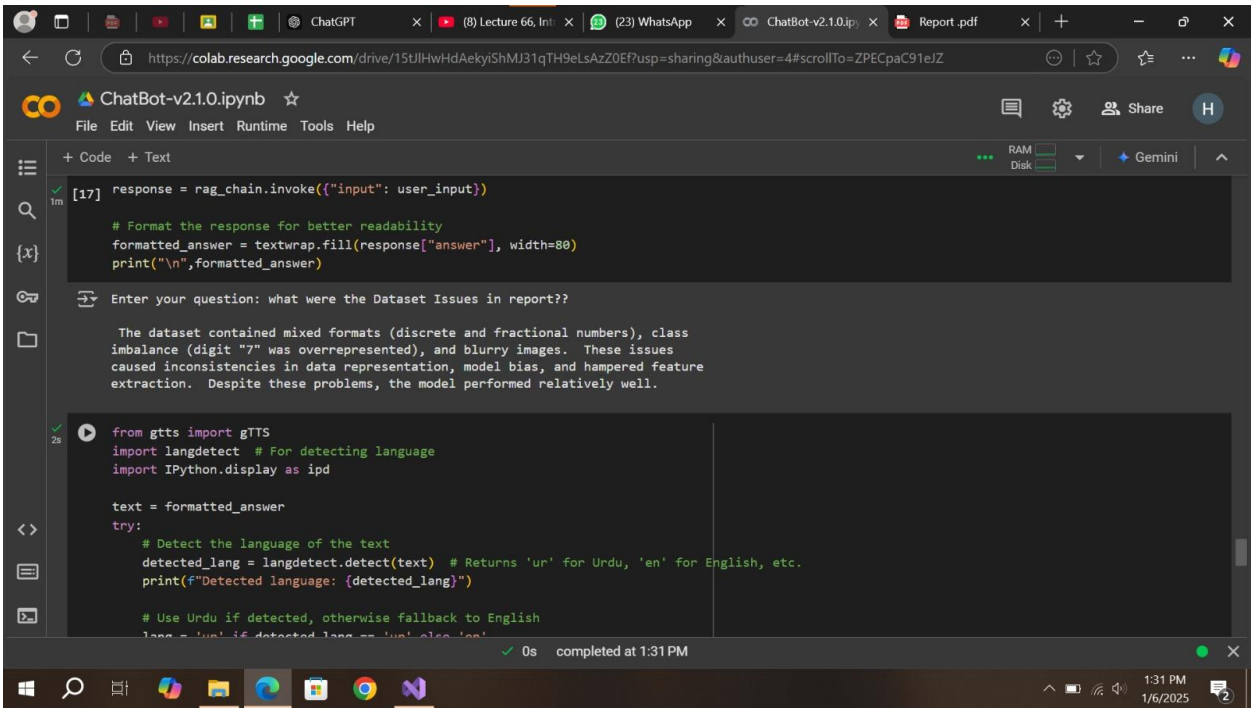
Output Mechanisms

1. **Textual Responses:** The chatbot displays the processed response on the screen in the selected language.
2. **Audio Responses:** Responses are also played back using text-to-speech technology for an engaging voice-based interaction.
3. **Dynamic Feedback:** The system provides visual and auditory feedback during query processing, such as loading animations or progress indicators.


ChatBot-v2.1.0.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
Share
Gemini
if os.path.exists(uploaded_file_name): # Ensure the uploaded file exists
os.rename(uploaded_file_name, base_file_name)
print(f"Credentials JSON uploaded and saved as: {base_file_name}")
else:
print(f"Error: Uploaded file '{uploaded_file_name}' not found.")
else:
# If no suffix exists, keep the uploaded file as-is
base_file_name = uploaded_file_name
print(f"No duplicates found. JSON file saved as: {base_file_name}")
Upload the credentials JSON file
Choose Files gen-lang-cli...8e753.json
gen-lang-client-0548761045-1883e7d8e753.json(application/json) - 2400 bytes, last modified: 11/25/2024 - 100% done
Saving gen-lang-client-0548761045-1883e7d8e753.json to gen-lang-client-0548761045-1883e7d8e753.json
File Name: gen-lang-client-0548761045-1883e7d8e753.json
No duplicates found. JSON file saved as: gen-lang-client-0548761045-1883e7d8e753.json
[8] key_path = base_file_name
print(key_path)
os.environ["GOOGLE_APPLICATION_CREDENTIALS"] = key_path
gen-lang-client-0548761045-1883e7d8e753.json
from langchain.vectorstores import Chroma
0s completed at 1:29 PM
1:29 PM
1/6/2025


ChatBot-v2.1.0.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Share
Gemini
import textwrap
from gtts import gTTS
from fpdf import FPDF
import IPython.display as ipd
user_input = input("Enter your question: ")
#Like What is new in YOLOv9?
response = rag_chain.invoke({"input": user_input})
# Format the response for better readability
formatted_answer = textwrap.fill(response["answer"], width=80)
print("\n",formatted_answer)
Enter your question: Dataset Issues in report??
from gtts import gTTS
import langdetect # For detecting language
import IPython.display as ipd
text = formatted_answer
try:
# Detect the language of the text
0s completed at 1:29 PM
1:31 PM
1/6/2025

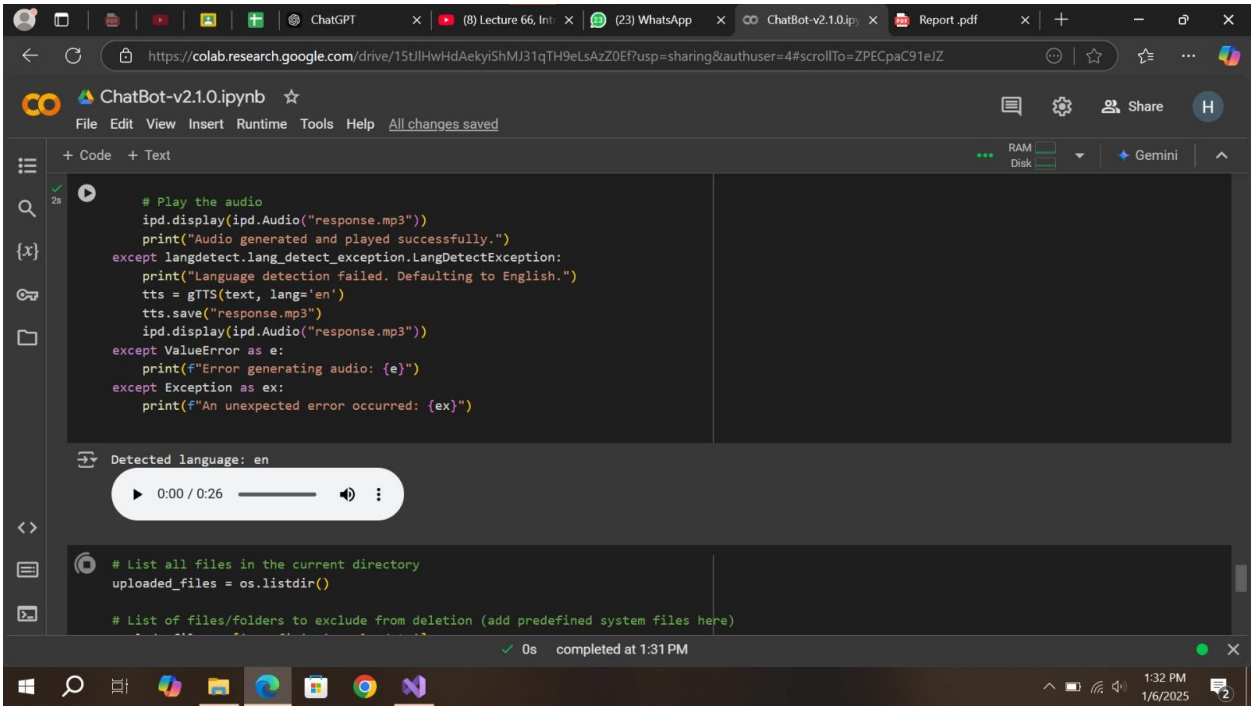

ChatGPT
(8) Lecture 66, Int
(23) WhatsApp
ChatBot-v2.1.0.ipy
Report .pdf
https://colab.research.google.com/drive/15tJlHwHdAekyiShMJ31qTH9eLsAzZ0Ef?usp=sharing&authuser=4#scrollTo=ZPECpaC91eJZ
ChatBot-v2.1.0.ipynb
File Edit View Insert Runtime Tools Help
Share
H
+ Code + Text
RAM
Disk
Gemini
[17] response = rag_chain.invoke({"input": user_input})
# Format the response for better readability
formatted_answer = textwrap.fill(response["answer"], width=80)
print("\n",formatted_answer)
Enter your question: what were the Dataset Issues in report??
The dataset contained mixed formats (discrete and fractional numbers), class
imbalance (digit "7" was overrepresented), and blurry images. These issues
caused inconsistencies in data representation, model bias, and hampered feature
extraction. Despite these problems, the model performed relatively well.
from gtts import gTTS
import langdetect # For detecting language
import IPython.display as ipd
text = formatted_answer
try:
# Detect the language of the text
detected_lang = langdetect.detect(text) # Returns 'ur' for Urdu, 'en' for English, etc.
print(f"Detected language: {detected_lang}")
# Use Urdu if detected, otherwise fallback to English
0s completed at 1:31 PM
1:31 PM
1/6/2025


ChatGPT
(8) Lecture 66, Int
(23) WhatsApp
ChatBot-v2.1.0.ipy
Report .pdf
https://colab.research.google.com/drive/15tJlHwHdAekyiShMJ31qTH9eLsAzZ0Ef?usp=sharing&authuser=4#scrollTo=ZPECpaC91eJZ
ChatBot-v2.1.0.ipynb
File Edit View Insert Runtime Tools Help
Share
H
+ Code + Text
RAM
Disk
Gemini
[17] response = rag_chain.invoke({"input": user_input})
# Format the response for better readability
formatted_answer = textwrap.fill(response["answer"], width=80)
print("\n",formatted_answer)
Enter your question: what were the Dataset Issues in report??
The dataset contained mixed formats (discrete and fractional numbers), class
imbalance (digit "7" was overrepresented), and blurry images. These issues
caused inconsistencies in data representation, model bias, and hampered feature
extraction. Despite these problems, the model performed relatively well.
from gtts import gTTS
import langdetect # For detecting language
import IPython.display as ipd
text = formatted_answer
try:
# Detect the language of the text
detected_lang = langdetect.detect(text) # Returns 'ur' for Urdu, 'en' for English, etc.
print(f"Detected language: {detected_lang}")
# Use Urdu if detected, otherwise fallback to English
0s completed at 1:31 PM
1:31 PM
1/6/2025

Help Document

Steps to Start the Prototype

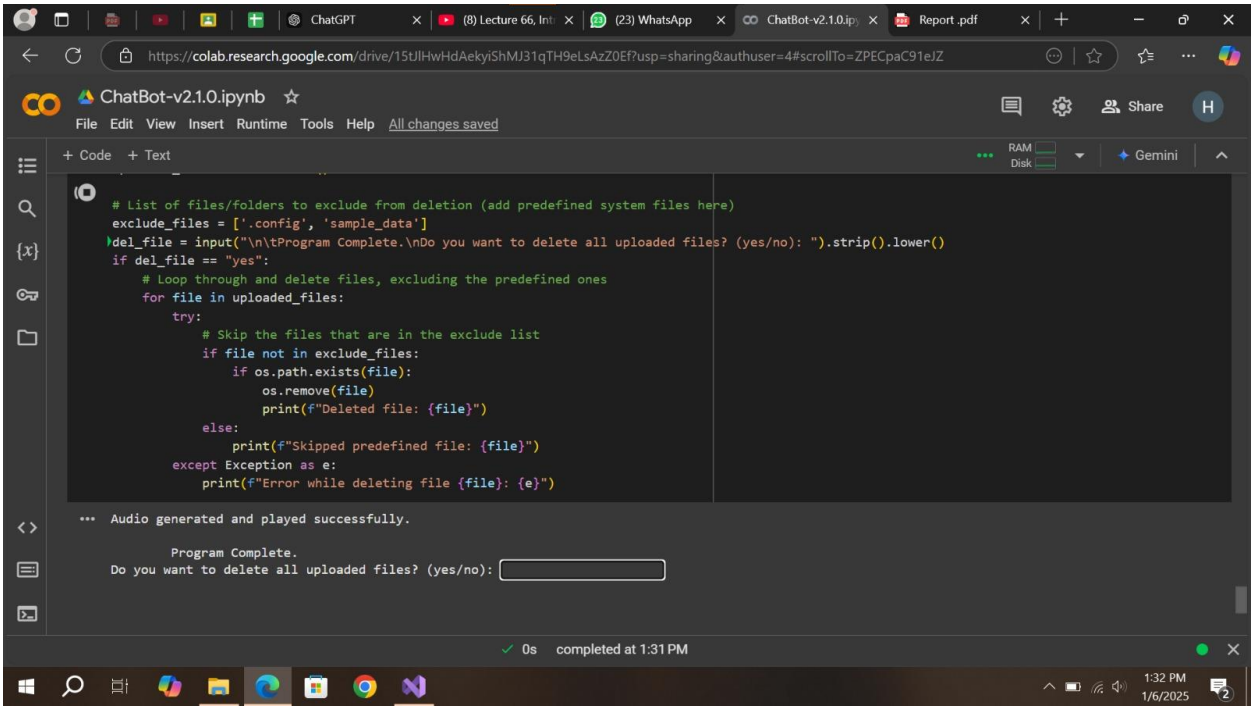
1. Open the Colab Notebook from the provided link.
2. Upload the required files (PDF or MP4) or enable the microphone for real-time voice input.
3. Run the notebook cells step-by-step to initialize the chatbot interface.
4. Begin interacting with the chatbot using voice or file uploads.



Interaction Sequences

1. **Voice Query Interaction:**
    a. Click the microphone icon.
    b. Speak your query (e.g., "What is the capital of Pakistan?").
    c. Wait for the chatbot to transcribe and process your input.
    d. View the response on-screen and hear it through the speakers.
2. **File Upload Interaction:**
    a. Click the upload button to attach a PDF or MP4 file.
    b. The chatbot processes the file's content.
    c. Receive textual and auditory responses based on the uploaded content.
3. **Multilingual Interaction:**
    a. Speak or type a query in Urdu or English.

b. The chatbot detects the language and generates the response accordingly.



Troubleshooting

1. Ensure the microphone permissions are enabled for voice input.
2. Check the internet connection for smooth API interactions.
3. Re-upload files if processing errors occur.

Conclusion

The **Colab Chatbot** prototype integrates voice, file uploads, and multilingual processing to provide an interactive educational experience. The dynamic interface ensures ease of use and accessibility for all users.