Project Report – Amit Meena

Project 1: Custom Chatbot Interface for LLaMA and Mistral Models

Objective:

- Build a **custom chatbot UI** where users interact with **LLaMA 2**, **LLaMA 3.2**, and **Mistral** models.

Implementation:

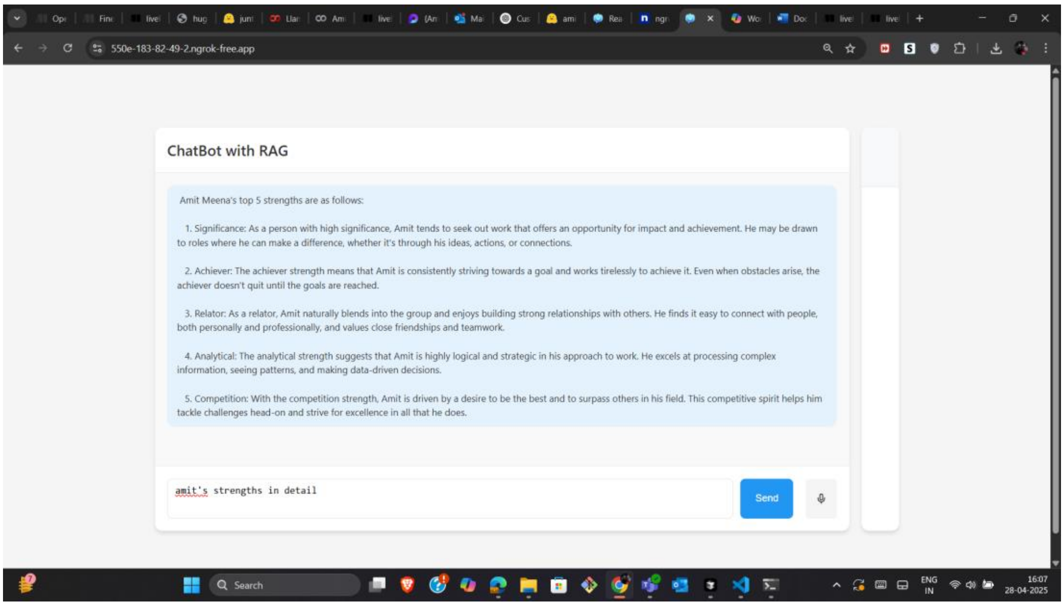
1. **Model Setup**:
    a. Downloaded and hosted **LLaMA 2**, **LLaMA 3.2**, and **Mistral** models locally.
    b. Optimized loading and memory management for efficient inference.
2. **Backend (Python)**:
    a. Developed a Flask-based backend API to handle user prompts and generate model responses.
    b. Integrated an optional retrieval step (RAG) **before** querying the model (detailed in Project 2).
3. **Frontend (React)**:
    a. Built an intuitive chatbot UI in React.
    b. Features include:
        i. Real-time conversation flow.
        ii. Model selection (user can choose between LLaMA 2, LLaMA 3.2, and Mistral).
        iii. Support for additional retrieved context (RAG responses injected dynamically).
        iv. API communication through the **Ngrok URL**.

4. **Testing**:
    a. Validated both direct model responses and RAG-enhanced responses.
    b. Optimized for low latency and smooth user experience.

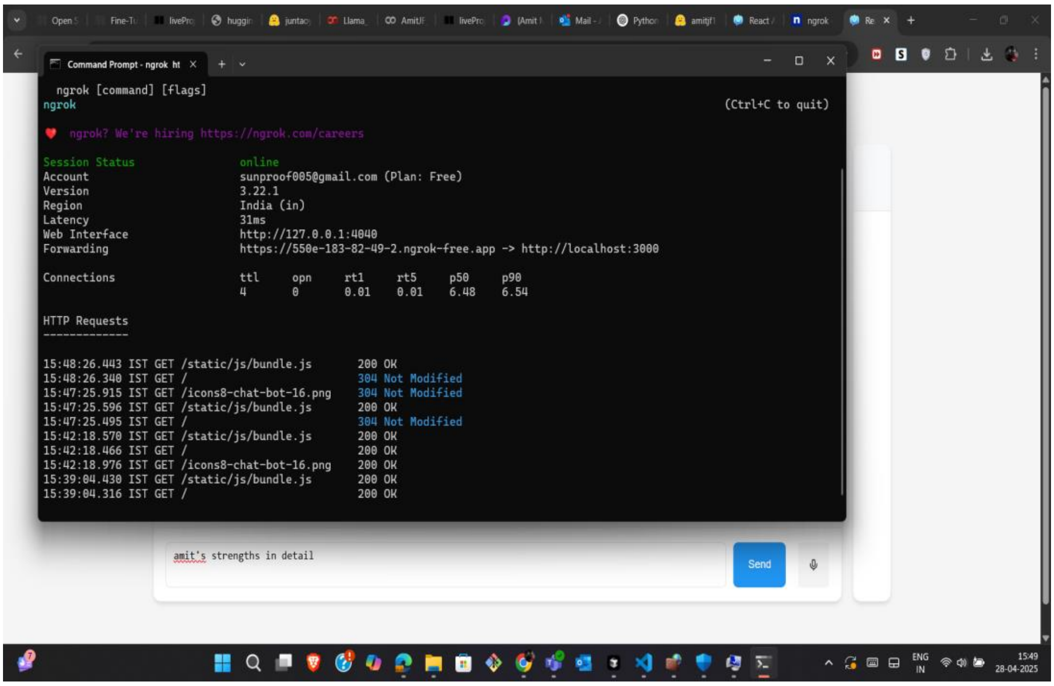
    Note: This chatbot is tightly coupled with a **Retrieval-Augmented Generation (RAG)** system for better, context-aware answers (see Project 2).

Screenshots:

a. *React Chatbot Interface:*



b. *Flask API Exposed via Ngrok:*



Project 2: Retrieval-Augmented Generation (RAG) System Integration

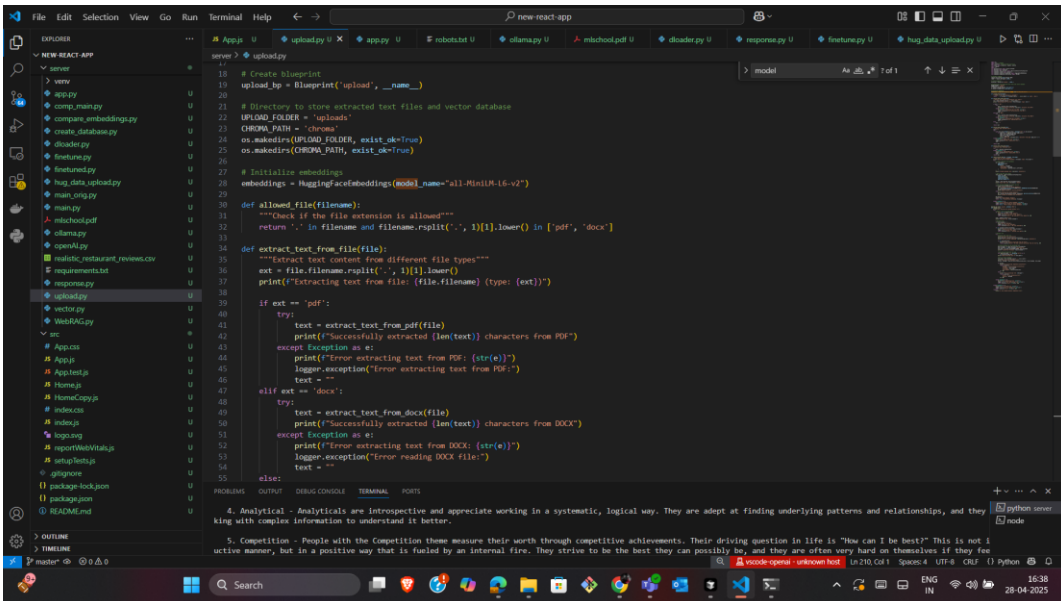
Objective:

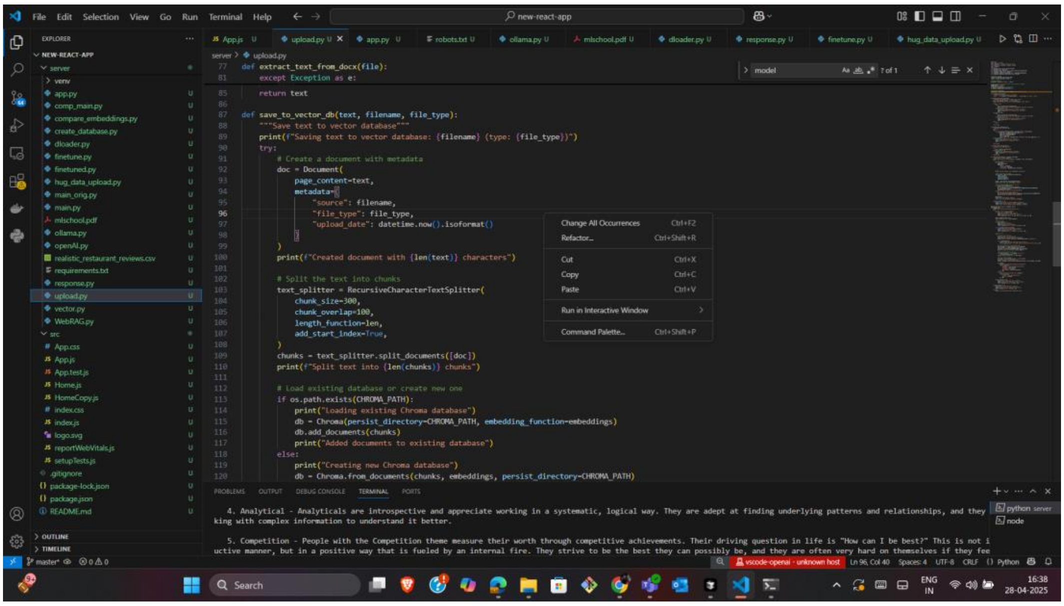
- Implement **Retrieval-Augmented Generation (RAG)** to improve the accuracy and relevance of model outputs.

Implementation:

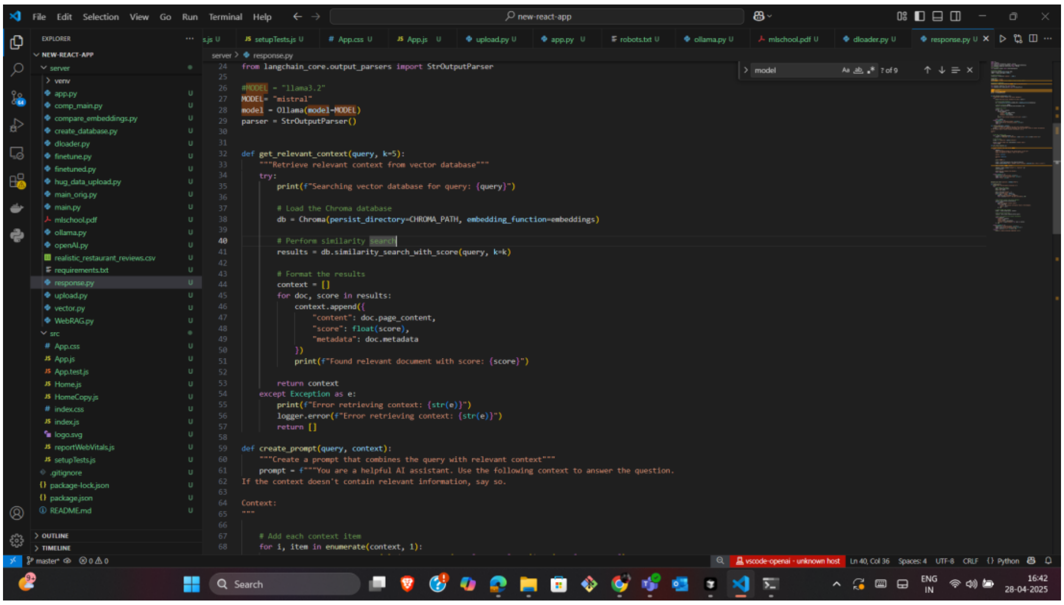
1. **Vector Database Creation**:
    a. Processed large sets of domain-specific documents.
    b. Converted documents into vector embeddings using transformer-based embedding models.
    c. Indexed embeddings into a **FAISS-based** vector database.
2. **Retrieval Pipeline**:
    a. On receiving a user query:
        i. Search the vector database for the most relevant documents.
        ii. Retrieve top-k matches as context snippets.
        iii. Append these snippets to the user prompt.
3. **Response Generation**:
    a. The model (LLaMA or Mistral) generates its answer **based on both** the user's query **and** the retrieved context.
    b. This ensures more **informed and accurate** responses, especially for domain-specific or fact-based questions.
4. **Integration with Chatbot**:
    a. Seamlessly connected the RAG pipeline with the chatbot backend.
    b. User can interact without noticing the retrieval step, but benefits from smarter outputs.
5. **Testing**:
    a. Compared RAG-enhanced outputs vs standard model outputs.
    b. Found significant improvement in answer relevance and factual correctness.
6. **Screenshots**:

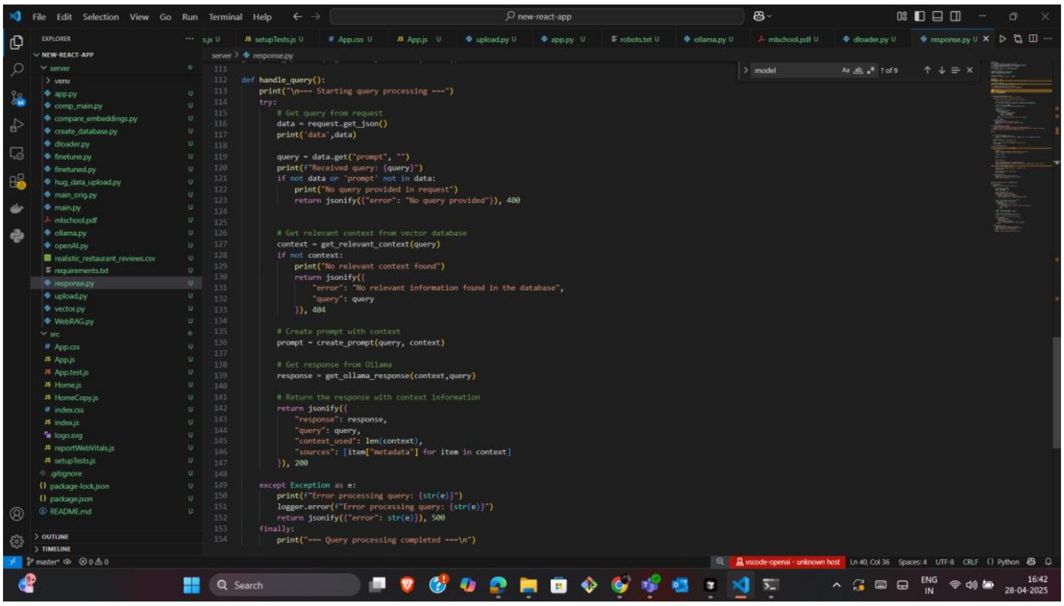
a. *Vector Database Setup (CHROMA DB):*

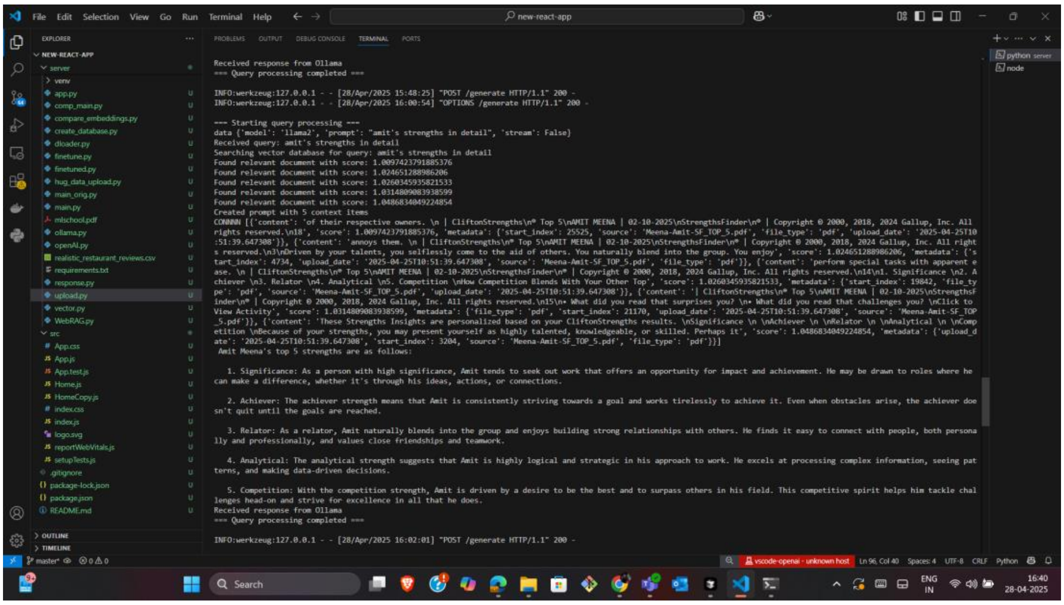




b. *Retrieved Context Added to Prompt:*


new-react-app
server > response.py
from langchain_core.output_parsers import StrOutputParser
#MODEL = "llama3.2"
MODEL= "mistral"
model = Ollama(model=MODEL)
parser = StrOutputParser()
def get_relevant_context(query, k=5):
"""Retrieve relevant context from vector database"""
try:
print(f"Searching vector database for query: {query}")
# Load the Chroma database
db = Chroma(persist_directory=CHROMA_PATH, embedding_function=embeddings)
# Perform similarity search
results = db.similarity_search_with_score(query, k=k)
# Format the results
context = []
for doc, score in results:
context.append({
"content": doc.page_content,
"score": float(score),
"metadata": doc.metadata
})
print(f"Found relevant document with score: {score}")
return context
except Exception as e:
print(f"Error retrieving context: {str(e)}")
logger.error(f"Error retrieving context: {str(e)}")
return []
def create_prompt(query, context):
"""Create a prompt that combines the query with relevant context"""
prompt = f"""You are a helpful AI assistant. Use the following context to answer the question.
If the context doesn't contain relevant information, say so.
Context:
"""
# Add each context item
for i, item in enumerate(context, 1):


new-react-app
server > response.py
def handle_query():
print("\n--- Starting query processing ---")
try:
# Get query from request
data = request.get_json()
print('data',data)
query = data.get("prompt", "")
print(f"Received query: {query}")
if not data or 'prompt' not in data:
print("No query provided in request")
return jsonify({"error": "No query provided"}), 400
# Get relevant context from vector database
context = get_relevant_context(query)
if not context:
print("No relevant context found")
return jsonify({
"error": "No relevant information found in the database",
"query": query
}), 404
# Create prompt with context
prompt = create_prompt(query, context)
# Get response from Ollama
response = get_ollama_response(context,query)
# Return the response with context information
return jsonify({
"response": response,
"query": query,
"context_used": len(context),
"sources": [item["metadata"] for item in context]
}), 200
except Exception as e:
print(f"Error processing query: {str(e)}")
logger.error(f"Error processing query: {str(e)}")
return jsonify({"error": str(e)}), 500
finally:
print("--- Query processing completed ---\n")


File Edit Selection View Go Run Terminal Help
new-react-app
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Received response from Ollama
=== Query processing completed ===
INFO:werkzeug:127.0.0.1 - - [28/Apr/2025 15:48:25] "POST /generate HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [28/Apr/2025 16:00:54] "OPTIONS /generate HTTP/1.1" 200 -
=== Starting query processing ===
data {'model': 'llama2', 'prompt': "amit's strengths in detail", 'stream': False}
Received query: amit's strengths in detail
Searching vector database for query: amit's strengths in detail
Found relevant document with score: 1.0097423791885376
Found relevant document with score: 1.024651288986206
Found relevant document with score: 1.0260345935821533
Found relevant document with score: 1.0314809083938599
Found relevant document with score: 1.0486834049224854
Created prompt with 5 context items
CONNNN [{'content': 'of their respective owners. \n | CliftonStrengths\n® Top 5\nAMIT MEENA | 02-10-2025\nStrengthsFinder\n® | Copyright © 2000, 2018, 2024 Gallup, Inc. All rights reserved.\n18', 'score': 1.0097423791885376, 'metadata': {'start_index': 25525, 'source': 'Meena-Amit-SF_TOP_5.pdf', 'file_type': 'pdf', 'upload_date': '2025-04-25T10:51:39.647308'}}, {'content': 'annoys them. \n | CliftonStrengths\n® Top 5\nAMIT MEENA | 02-10-2025\nStrengthsFinder\n® | Copyright © 2000, 2018, 2024 Gallup, Inc. All rights reserved.\n3\nDriven by your talents, you selflessly come to the aid of others. You naturally blend into the group. You enjoy', 'score': 1.024651288986206, 'metadata': {'start_index': 4734, 'upload_date': '2025-04-25T10:51:39.647308', 'source': 'Meena-Amit-SF_TOP_5.pdf', 'file_type': 'pdf'}}, {'content': 'perform special tasks with apparent ease. \n | CliftonStrengths\n® Top 5\nAMIT MEENA | 02-10-2025\nStrengthsFinder\n® | Copyright © 2000, 2018, 2024 Gallup, Inc. All rights reserved.\n14\n1. Significance \n2. Achiever \n3. Relator \n4. Analytical \n5. Competition \nHow Competition Blends With Your Other Top', 'score': 1.0260345935821533, 'metadata': {'start_index': 19842, 'file_type': 'pdf', 'source': 'Meena-Amit-SF_TOP_5.pdf', 'upload_date': '2025-04-25T10:51:39.647308'}}, {'content': '| CliftonStrengths\n® Top 5\nAMIT MEENA | 02-10-2025\nStrengthsFinder\n® | Copyright © 2000, 2018, 2024 Gallup, Inc. All rights reserved.\n15\n• What did you read that surprises you? \n• What did you read that challenges you? \nClick to View Activity', 'score': 1.0314809083938599, 'metadata': {'file_type': 'pdf', 'start_index': 21170, 'upload_date': '2025-04-25T10:51:39.647308', 'source': 'Meena-Amit-SF_TOP_5.pdf'}}, {'content': 'These Strengths Insights are personalized based on your CliftonStrengths results. \nSignificance \n \nAchiever \n \nRelator \n \nAnalytical \n \nCompetition \nBecause of your strengths, you may present yourself as highly talented, knowledgeable, or skilled. Perhaps it', 'score': 1.0486834049224854, 'metadata': {'upload_date': '2025-04-25T10:51:39.647308', 'start_index': 3204, 'source': 'Meena-Amit-SF_TOP_5.pdf', 'file_type': 'pdf'}}]
Amit Meena's top 5 strengths are as follows:
1. Significance: As a person with high significance, Amit tends to seek out work that offers an opportunity for impact and achievement. He may be drawn to roles where he can make a difference, whether it's through his ideas, actions, or connections.
2. Achiever: The achiever strength means that Amit is consistently striving towards a goal and works tirelessly to achieve it. Even when obstacles arise, the achiever doesn't quit until the goals are reached.
3. Relator: As a relator, Amit naturally blends into the group and enjoys building strong relationships with others. He finds it easy to connect with people, both personally and professionally, and values close friendships and teamwork.
4. Analytical: The analytical strength suggests that Amit is highly logical and strategic in his approach to work. He excels at processing complex information, seeing patterns, and making data-driven decisions.
5. Competition: With the competition strength, Amit is driven by a desire to be the best and to surpass others in his field. This competitive spirit helps him tackle challenges head-on and strive for excellence in all that he does.
Received response from Ollama
=== Query processing completed ===
INFO:werkzeug:127.0.0.1 - - [28/Apr/2025 16:02:01] "POST /generate HTTP/1.1" 200 -
vscode-openai - unknown host
Ln 96, Col 40 Spaces: 4 UTF-8 CRLF Python
Search
16:40
28-04-2025

Project 3: Dataset Creation and Fine-tuning of LLaMA Model

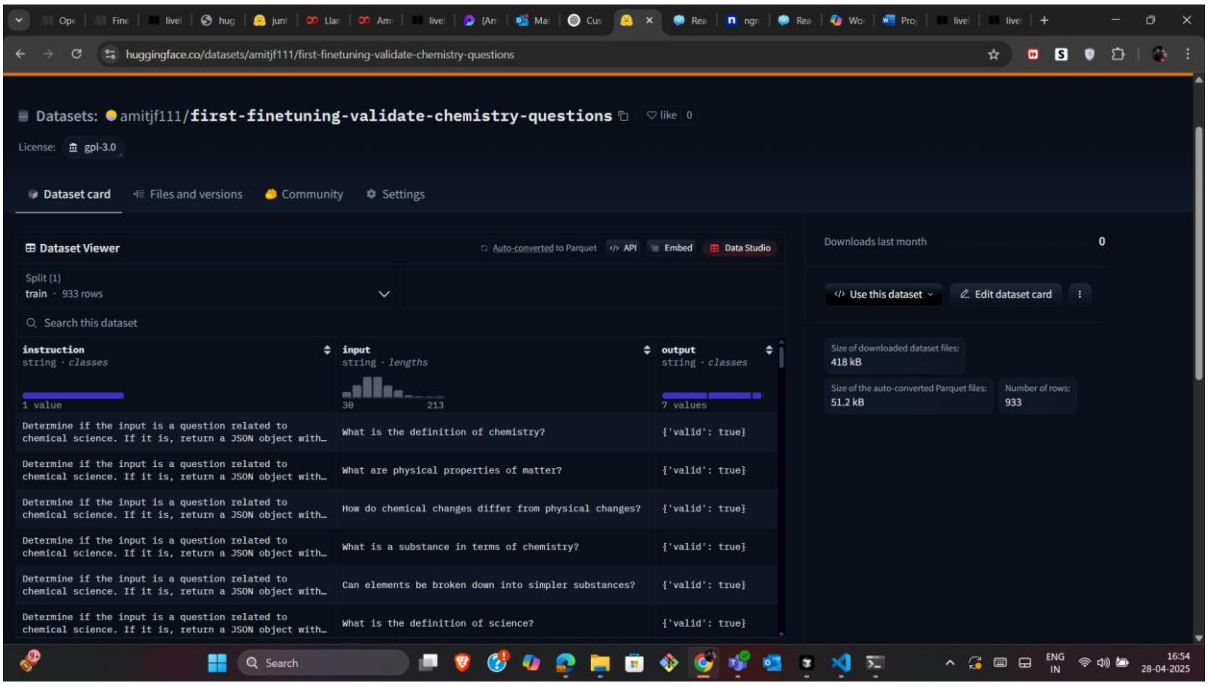
Objective:

- Create a custom dataset.
- Upload it to Hugging Face.
- Fine-tune the LLaMA model to adapt it to specific tasks/domains.

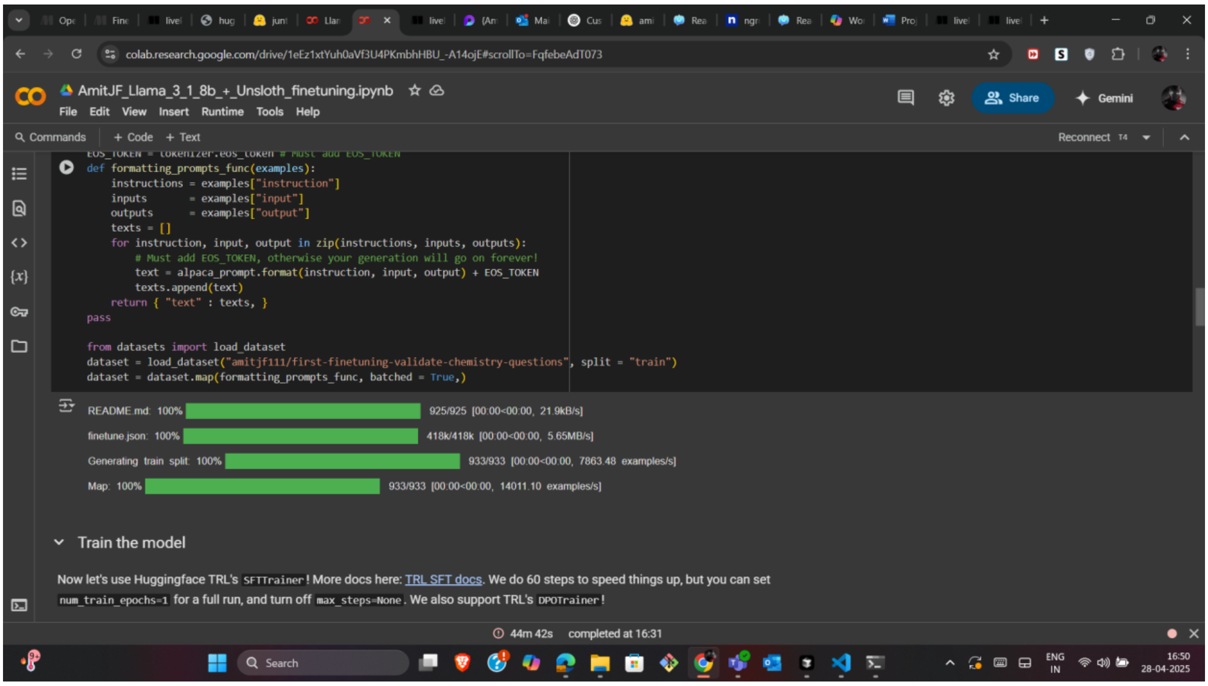
Implementation:

1. **Dataset Creation**:
    a. Designed a structured dataset (e.g., instruction-response pairs, domain-specific Q&A).
    b. Ensured high data quality with clear formatting (JSONL/CSV).
2. **Uploading to Hugging Face**:
    a. Uploaded the dataset to my Hugging Face account for accessibility and versioning.
3. **Fine-tuning**:
    a. Fine-tuned the LLaMA model using the custom dataset.
    b. Techniques used:
        i. LoRA (Low-Rank Adaptation) for efficient fine-tuning.
        ii. Hyperparameter tuning (learning rate, batch size, epochs) for optimal results.
4. **Results**:
    a. Achieved a more specialized LLaMA model that performed better on custom task domains.
    b. Improved coherence, task adherence, and response specificity.

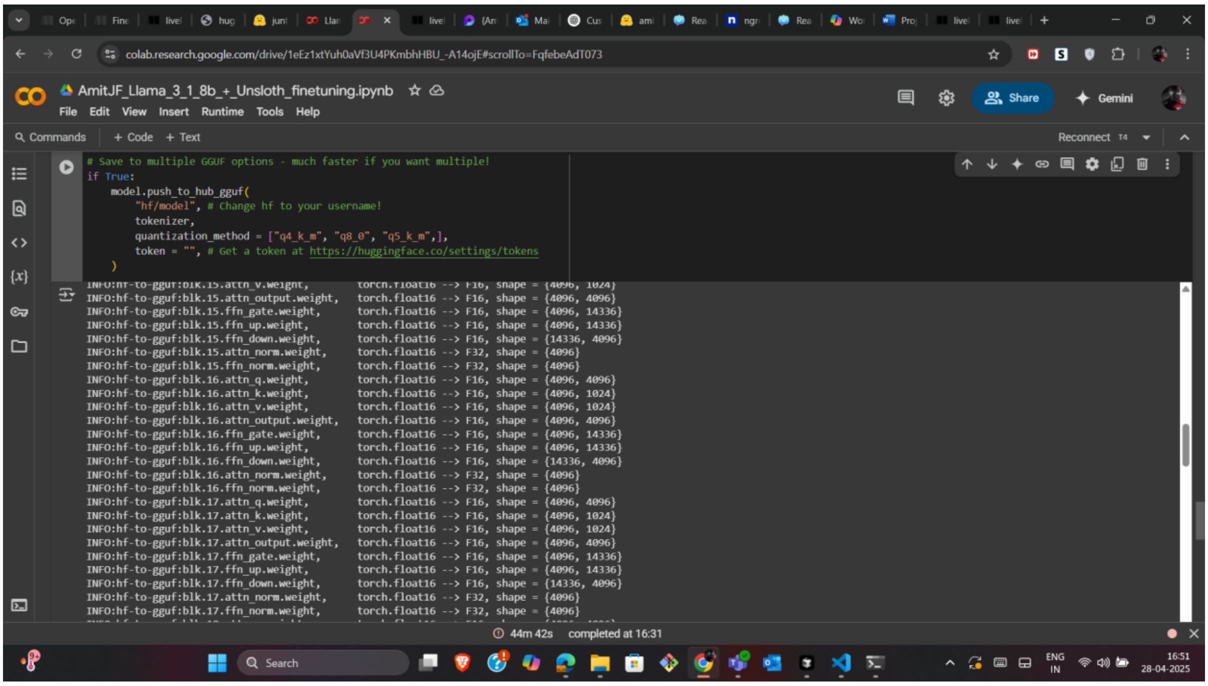
Screenshot - Custom Dataset Example(Hugging Face upload):



Screenshot – use while finetuning:



Screenshot - Fine-tuning Logs:



Conclusion

This project series involved end-to-end development: setting up strong foundation models, creating an interactive chatbot, augmenting it with retrieval for smarter responses, and fine-tuning a model for even more targeted performance.

The experience covered key real-world AI production workflows like serving large models, building retrieval pipelines, integrating frontends, and model personalization.