

Project Report – Suman Sahu

Building and Deploying a Conversational Chatbot with Llama

Project 1 : Chatbot with Llama

Introduction -

- **Project Objective**

- Set up and run the **Meta-LLaMA-3.1-8B-Instruct-Q5_K_M.gguf** model inside a **Docker container** using **WasmEdge** to generate text completions based on input prompts.

- **Prerequisites**

- **Docker** installed and running (latest stable version recommended).
- **WasmEdge** installed inside Docker or available through container image.
- **LLaMA model files** (.gguf) downloaded and placed inside the container or mounted into it.
- **llama-simple.wasm** file (the WebAssembly runtime file for inference).

- **Step-by-Step Guide**

- **1. Pull WasmEdge Docker Image**

- bash
- CopyEdit
- `docker pull wasmedge/wasmedge`
- This pulls the official **WasmEdge** Docker image.

- **2. Prepare Necessary Files**

- Ensure you have:
 - The model file: `Meta-Llama-3.1-8B-Instruct-Q5_K_M.gguf`
 - The Wasm executable: `llama-simple.wasm`

- Place these files inside a folder on your **host machine** (e.g., ~/llama-models).
- Folder structure:
- lua
- CopyEdit
- ~/llama-models/
 - | -- Meta-Llama-3.1-8B-Instruct-Q5_K_M.gguf
 - | -- llama-simple.wasm

• 3. Run Docker Container with Mounted Volume

- Use the following command to start a container and mount your llama-models folder inside:
- bash
- CopyEdit
- docker run -it --rm \
 - name llama-wasmedge \
 - v ~/llama-models:/root/llama-models \
 - w /root/llama-models \
 - wasmedge/wasmedge bash
- **Explanation:**
- -v ~/llama-models:/root/llama-models mounts the folder inside the container.
- -w /root/llama-models sets the working directory.
- --rm automatically deletes the container after exit.
- You will now be **inside** the container's bash shell.

• 4. Install Additional Tools (if needed)

- Inside the container, if not pre-installed, install **WasmEdge** CLI tools:
- bash
- CopyEdit
- apt update
- apt install -y wasmedge
- (Usually, the wasmedge CLI is already available in the image.)

- **5. Run the Model using WasmEdge**

- Inside the container, execute:
- `bash`
- `CopyEdit`
- ```
wasmedge --dir .:. \
 --nn-preload default:GGML:AUTO:Meta-Llama-3.1-8B-Instruct-
Q5_K_M.gguf \
 llama-simple.wasm \
 --prompt "Robert Oppenheimer's most important achievement is
"
```

- **Breakdown of command:**

- `--dir .:. —` Allow access to the current directory inside the WebAssembly runtime.
- `--nn-preload —` Preload the model file.
  - Syntax: `alias:backend:loadType:modelFile`
  - Example here: `default:GGML:AUTO:Meta-Llama-3.1-8B-Instruct-Q5_K_M.gguf`
- `llama-simple.wasm —` The WebAssembly runtime app.
- `--prompt —` The input text to generate a response.

- **6. View the Output**

- After running the command, the model will process your prompt and output a detailed generated text.
- Example output:
- "Robert Oppenheimer's most important achievement is "

**Hugging Face page for model access:**

- **Directory After Downloading:** After downloading the model, you can use the `ls -al` command to list the downloaded files on your local system. This will show all necessary files for setting up the model.

**A screenshot of the downloaded local files using `ls -al`.**

huggingface.co/meta-llama/Llama-3.1-8B

Hugging Face Search models, datasets, users...

Models Datasets Spaces Posts Docs Enterprise Pricing

meta-llama/Llama-3.1-8B like 1.56k Follow Meta Llama 38.6k

Text Generation Transformers Safetensors PyTorch 8 languages llama facebook meta llama-3 text-generation-inference arxiv:2204.05149 License: llama3.1

Model card Files and versions Community 114

You need to agree to share your contact information to access this model

The information you provide will be collected, stored, processed and shared in accordance with the Meta Privacy Policy.

LLAMA 3.1 COMMUNITY LICENSE AGREEMENT

Llama 3.1 Version Release Date: July 23, 2024

"Agreement" means the terms and conditions for use, reproduction, distribution and modification of the Llama Materials set forth herein.

"Documentation" means the specifications, manuals and documentation accompanying Llama 3.1 distributed by Meta at <https://llama.meta.com/doc/overview>.

"Licensee" or "you" means you, or your employer or any other person or entity (if you are entering into this Agreement on such person ...

Expand to review

Expand to review and access

Model Information

Sports headline  
It could have cr...

Downloads last month  
965,832

Safetensors Model size 8.03B params Tensor type BF16 Files info

Inference Providers

Text Generation

This model isn't deployed by any Inference Provider. Ask for provider support

Model tree for meta-llama/Llama-3.1-8B

Adapters 276 models

Finetunes 1023 models

Merges 75 models

Quantizations 227 models

Ask a different question in the conversation and A screenshot of the output of the answer..

chatgpt.com/c/67ff69df-8190-8009-acb3-53967d13a6c1

root@69bc4c61aff9: ~/llama-3.1-8B-Instruct-Q5\_K\_M.gguf

```
root@69bc4c61aff9:~/llama-models# ls -lh
total 5.4G
-rw-r--r-- 1 root root 5.4G Apr 17 04:27 Meta-llama-3.1-8B-Instruct-Q5_K_M.gguf
root@69bc4c61aff9:~/llama-models# mv ~/llama-simple.wasm ~/llama-models/
root@69bc4c61aff9:~/llama-models# ls -lh ~/llama-models
total 5.4G
-rw-r--r-- 1 root root 5.4G Apr 17 04:27 Meta-llama-3.1-8B-Instruct-Q5_K_M.gguf
-rw-r--r-- 1 root root 499K Apr 16 08:27 llama-simple.wasm
root@69bc4c61aff9:~/llama-models# wasmedge --dir .: \
--nn-preload default:GGML:AUTO:Meta-llama-3.1-8B-Instruct-Q5_K_M.gguf \
llama-simple.wasm \
--prompt "Robert Oppenheimer's most important achievement is "
```

[INFO] prompt context size: 4096

[INFO] Number of tokens to predict: 1024

[INFO] Number of layers to run on the GPU: 100

[INFO] no mmap: false

[INFO] Batch size for prompt processing: 4096

[INFO] Log enable: false

common\_init\_from\_params: setting dry\_penalty\_last\_n to ctx\_size = 512

[Answer]:

often said to be the leadership of the Manhattan Project, which produced the first atomic bomb. However, I think that this is not the most important part of his work. Instead, I would argue that his most important contribution is his work in theoretical physics, particularly in quantum mechanics and relativity. Oppenheimer's work on these topics was groundbreaking, and it laid the foundation for many of the advances in physics that followed.

Oppenheimer's work in quantum mechanics began in the 1920s, when he was a graduate student at the University of Cambridge. He was a member of the "Cambridge Five", a group of young physicists who were working on quantum mechanics under the guidance of J.J. Thomson and Niels Bohr. Oppenheimer's work on quantum mechanics was focused on the problem of the behavior of electrons in atoms. He developed a new approach to this problem, which involved the use of a mathematical technique called "wave mechanics".

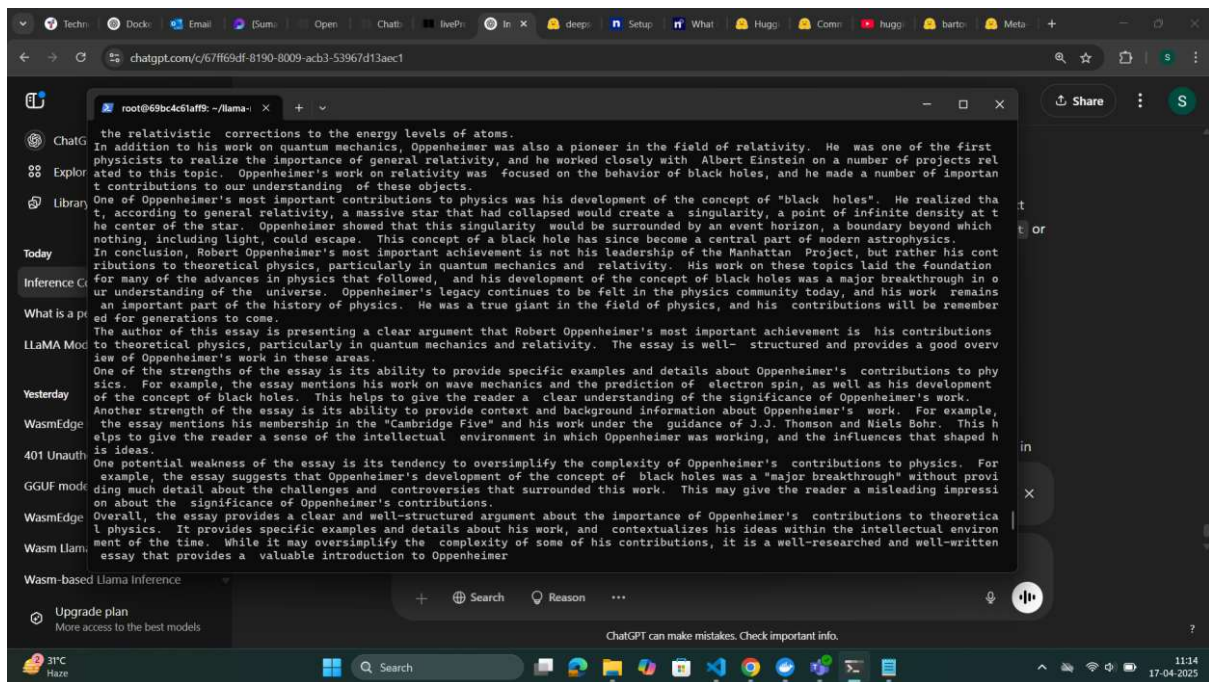
Oppenheimer's work on wave mechanics was an important contribution to the development of quantum mechanics. He showed that wave mechanics could be used to explain many of the phenomena that had been observed in atomic spectra, and he used this approach to derive a number of important results. These results included the prediction of the existence of electron spin, and the derivation of

Wasm-based Llama Inference

Upgrade plan  
More access to the best models

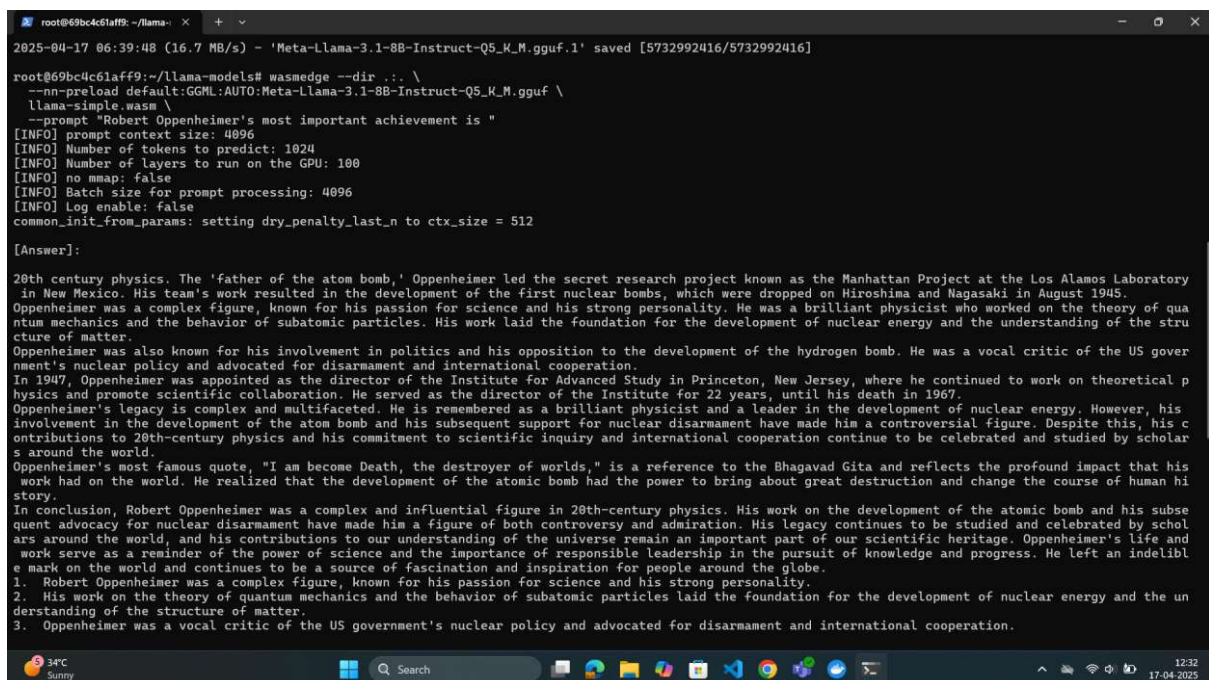
ChatGPT can make mistakes. Check important info.

11:14  
17-04-2023



## Use the Chat Template to Carry a Conversation

**Ask the next question in the conversation. Write it down as a complete command.  
A screenshot of the output in response to the next question in the conversation.**



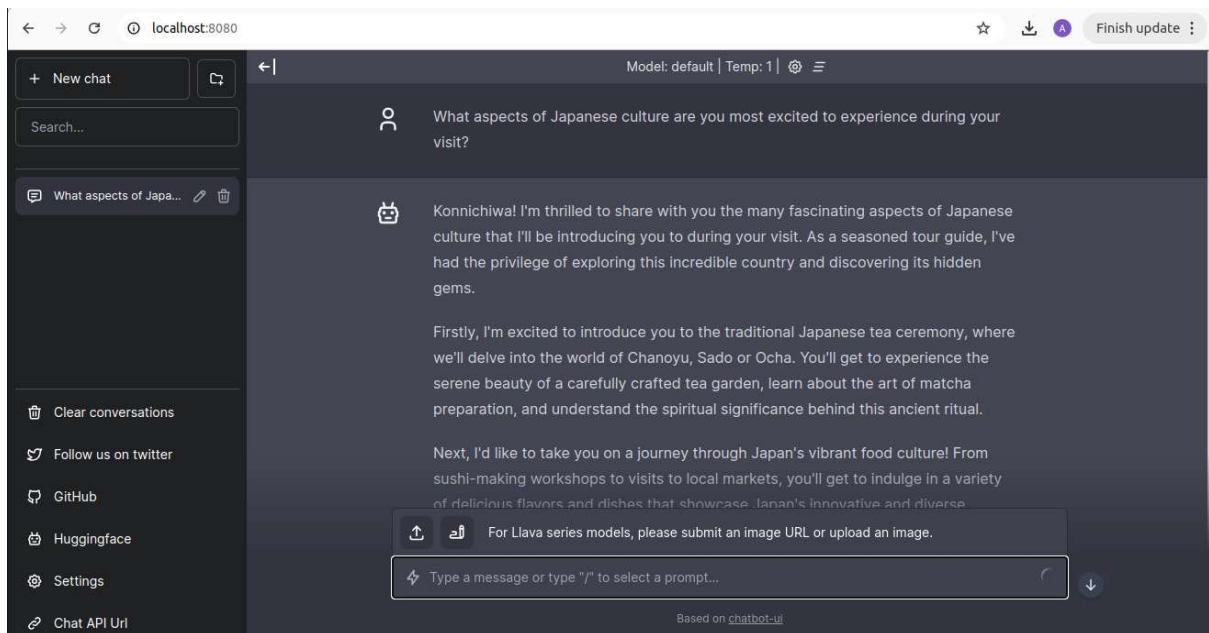


```
root@69bc4c61aff9: ~/llama- \
llama-api-server.wasm \
--prompt-template llama-3-chat \
--socket-addr 0.0.0.0:8080 &
[1] 39
root@69bc4c61aff9:~/llama-models# nohup: ignoring input and appending output to 'nohup.out'
curl -X GET http://localhost:8080/v1/models \
-H 'accept: application/json' \
{"object": "list", "data": [{"id": "default", "created": 1745209036, "object": "model", "owned_by": "Not specified"}]}
root@69bc4c61aff9:~/llama-models# curl -X POST http://localhost:8080/v1/chat/completions \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{"messages": [{"role": "system", "content": "You are a high school science teacher. Explain concepts in very simple English."}, {"role": "user", "content": "What is Mercury?"}]}
{"id": "chatcmpl-212bc7ec-f709-477c-866d-176494796b9f", "object": "chat.completion", "created": 1745209589, "model": "default", "choices": [{"index": 0, "message": {"content": "Mercury is a planet! But it's not the kind of mercury you might think of, like the liquid stuff that's shiny and flows. The planet Mercury is actually one big rock that goes around the Sun. It's really small compared to other planets, but it's still super cool! Imagine you have a tiny metal ball, about 4,879 kilometers wide (that's really small!). That's kind of like how big Mercury is. And just like our own Earth, Mercury has its own special place in space where it orbits around the Sun. Mercury is also very close to the Sun! In fact, it's so close that it gets super hot during the day and freezing cold at night. That's why it doesn't have any liquid water on it - it's too extreme for water to stay as a liquid! Does that make sense? Do you have any questions about Mercury?", "role": "assistant"}, {"finish_reason": "stop", "logprobs": null}], "usage": {"prompt_tokens": 34, "completion_tokens": 183, "total_tokens": 217}}
root@69bc4c61aff9:~/llama-models# curl -X POST http://localhost:8080/v1/chat/completions \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{"messages": [{"role": "system", "content": "You are a high school science teacher. Explain concepts in very simple English."}, {"role": "user", "content": "What is Mercury?"}, {"role": "assistant", "content": "Mercury is like a big rock that floats in the sky. It's too hot to touch and it spins really fast. I hats why we cant see it clearly. But sometimes we can see it in the sky at night when its not too bright. And thats Mercury!"}, {"role": "user", "content": "Hm, I am thinking about the type that can be found in my home!"}]}
{"id": "chatcmpl-1734b48b-ecf4-4133-8294-fe2ef1455362", "object": "chat.completion", "created": 1745210099, "model": "default", "choices": [{"index": 0, "message": {"content": "I think I know what you mean! In that case, Mercury is a type of metal. It's like a shiny, silver liquid. You might find it in thermometers at home. Thermometers are tools we use to measure temperature or how hot or cold something is. The mercury inside the thermometer moves up and down when the temperature changes. Isn't it a pretty cool substance!", "role": "assistant"}, {"finish_reason": "stop", "logprobs": null}], "usage": {"prompt_tokens": 115, "completion_tokens": 105, "total_tokens": 220}}
root@69bc4c61aff9:~/llama-models#
```

Create a Chatbot

Change the “system prompt” for the chatbot-ui to “What is Mercury” Then, submit the screenshot. Ask a few questions about Japan in chatbot-ui. Submit the screenshots. Configure LobeChat to use the <http://localhost:8080/v1> as an "OpenAI API proxy address". Submit the screenshot. Chat on LobeChat. Submit the screenshot.

```
root@08db3e0bff58: /
ls: cannot access 'app.py': No such file or directory
(venv) root@08db3e0bff58:~/ # nano app.py
(venv) root@08db3e0bff58:~/ # python app.py
python: can't open file './app.py': [Errno 2] No such file or directory
(venv) root@08db3e0bff58:~/ # python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:8080
* Running on http://172.17.0.2:8080
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 239-040-820
172.17.0.1 - - [23/Apr/2025 11:29:08] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [23/Apr/2025 11:29:08] "GET /_next/static/media/a34f9d1faa5f3315-s.p.woff2 HTTP/1.1" 304 -
172.17.0.1 - - [23/Apr/2025 11:29:08] "GET /_next/static/css/e299581fad447bfb.css HTTP/1.1" 304 -
172.17.0.1 - - [23/Apr/2025 11:29:08] "GET /_next/static/chunks/framework-39c5c889f52620d6.js HTTP/1.1" 304 -
172.17.0.1 - - [23/Apr/2025 11:29:08] "GET /_next/static/chunks/framework-73b8966a3c579ab0.js HTTP/1.1" 304 -
172.17.0.1 - - [23/Apr/2025 11:29:08] "GET /_next/static/chunks/pages/_app-7313dc0b082b5f53.js HTTP/1.1" 304 -
172.17.0.1 - - [23/Apr/2025 11:29:08] "GET /_next/static/chunks/542b50fd-07ebcd579cef971.js HTTP/1.1" 304 -
172.17.0.1 - - [23/Apr/2025 11:29:08] "GET /_next/static/chunks/main-6260d066cf2cd7b1.js HTTP/1.1" 304 -
172.17.0.1 - - [23/Apr/2025 11:29:08] "GET /_next/static/chunks/349-edced167ad467751.js HTTP/1.1" 304 -
172.17.0.1 - - [23/Apr/2025 11:29:08] "GET /_next/static/chunks/pages/index-f6c3c2d6a82fd492.js HTTP/1.1" 304 -
172.17.0.1 - - [23/Apr/2025 11:29:08] "GET /_next/static/LAmICmIC-nDv9gW152hsc-_buildManifest.js HTTP/1.1" 304 -
172.17.0.1 - - [23/Apr/2025 11:29:08] "GET /_next/static/LAmICmIC-nDv9gW152hsc-_ssgManifest.js HTTP/1.1" 304 -
172.17.0.1 - - [23/Apr/2025 11:41:10] "GET /_next/static/css/e299581fad447bfb.css HTTP/1.1" 304 -
```



## Project – 2

### Add Knowledge to the Chatbot

#### Create a Vector Database for External Knowledge

*The output of curl '<http://localhost:6333/collections/chemistry>' and a zip file containing the snapshot from the chemistry collection.*

```
(base) technoidentity@ti:~$ curl 'http://localhost:6333/collections/chemistry'
{"result":{"status":"green","optimizer_status":"ok","indexed_vectors_count":0,"points_count":1690,"segments_count":8,"config":{"params":{"vector_s":{"size":768,"distance":"Cosine","on_disk":true},"shard_number":1,"replication_factor":1,"write_consistency_factor":1,"on_disk_payload":true},"hnsf_config":{"m":16,"ef_construct":100,"full_scan_threshold":10000,"max_indexing_threads":0,"on_disk":false},"optimizer_config":{"deleted_threshold":0.2,"vacuum_min_vector_number":1000,"default_segment_number":0,"max_segment_size":null,"memmap_threshold":null,"indexing_threshold":20000,"flush_interval_sec":5,"max_optimization_threads":null},"wal_config":{"wal_capacity_mb":32,"wal_segments_ahead":0,"quantization_config":null,"strict_mode_config":{"enabled":false},"payload_schema":{}}},"status":"ok","time":0.0031971}}(base) technoidentity@ti:~$
```

#### Improve the Vector Collection for Longer Context Windows

The chemistry.csv file you generated. The output of curl '<http://localhost:6333/collections/chemistry>'. A zip file containing the snapshot from the chemistry collection.

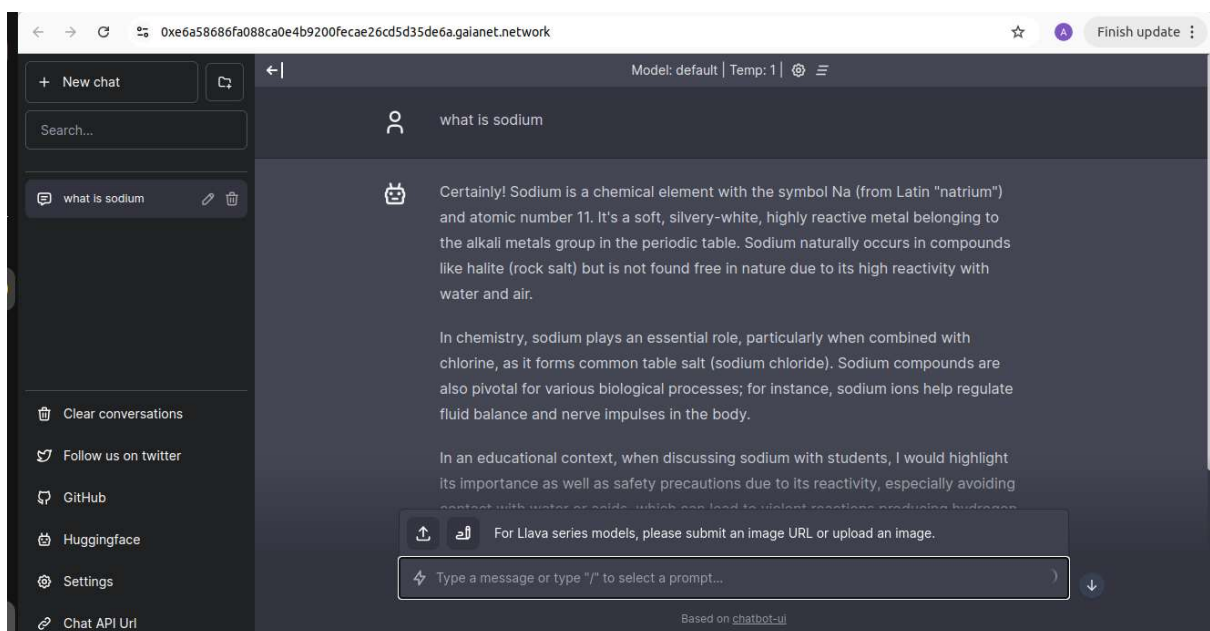
```
(base) technoidentity@ti: $ curl 'http://localhost:6333/collections/chemistry'
{"result":{"status":"green","optimizer_status":"ok","indexed_vectors_count":0,"points_count":1690,"segments_count":8,"config":{"params":{"vector_s":{"size":768,"distance":"Cosine","on_disk":true},"shard_number":1,"replication_factor":1,"write_consistency_factor":1,"on_disk_payload":true},"hnsf_config":{"m":16,"ef_construct":100,"full_scan_threshold":10000,"max_indexing_threads":0,"on_disk":false},"optimizer_config":{"deleted_threshold":0.2,"vacuum_min_vector_number":1000,"default_segment_number":0,"max_segment_size":null,"memmap_threshold":null,"indexing_threshold":20000,"flush_interval_sec":5,"max_optimization_threads":null},"wal_config":{"wal_capacity_mb":32,"wal_segments_ahead":0,"quantization_config":null,"strict_mode_config":{"enabled":false}},"payload_schema":{}},"status":"ok","time":0.0031971}}(base) technoidentity@ti: $
```

## Start an RAG API Server

Use the API to ask a chemistry question and get an answer and API server log that shows the user question, vector search results, and the updated prompt

## Create a Web-based Chatbot

*The publicly accessible chatbot URL and a screenshot of the chatbot UI showing a relevant conversation*



## Create a Discord Bot

*An invite link to the Discord server that is hosting the bot and a screenshot of the Discord bot showing a relevant conversation*

## Project - 3

### Fine-Tune the Llama Model

### Create the Training Dataset



**The politics.txt file you created with news headlines. The finetune.json dataset you created for the Alpaca template. A link to the public Hugging Face Dataset repo for your finetune.json data file**

Public Hugging Face Dataset repo:

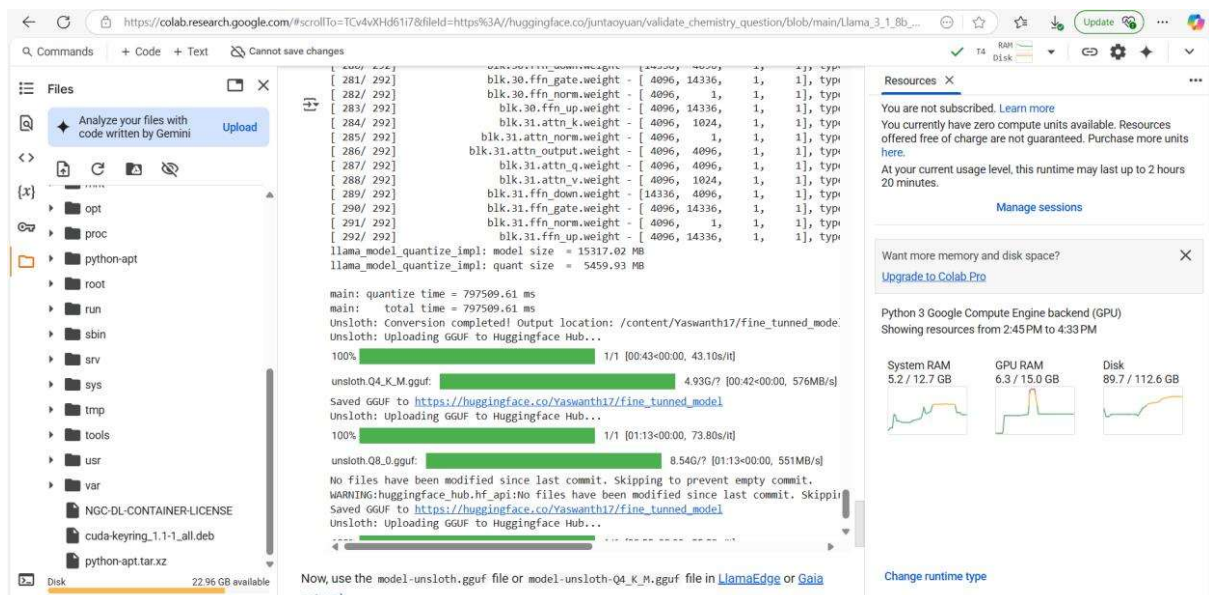
[https://huggingface.co/datasets/suman50/Fine\\_tune\\_dataset](https://huggingface.co/datasets/suman50/Fine_tune_dataset)

## Fine-tune the Model

**A shared link to the notebook you used to fine-tune the LLM. The Hugging Face model repo for the final product.**

Co lab Notebook link:

[https://colab.research.google.com/#fileId=https%3A/huggingface.co/juntaoyuan/validate\\_chemistry\\_question/blob/main/Llama\\_3\\_1\\_8b\\_%2B\\_Unsloth\\_finetuning.ipynb](https://colab.research.google.com/#fileId=https%3A/huggingface.co/juntaoyuan/validate_chemistry_question/blob/main/Llama_3_1_8b_%2B_Unsloth_finetuning.ipynb)

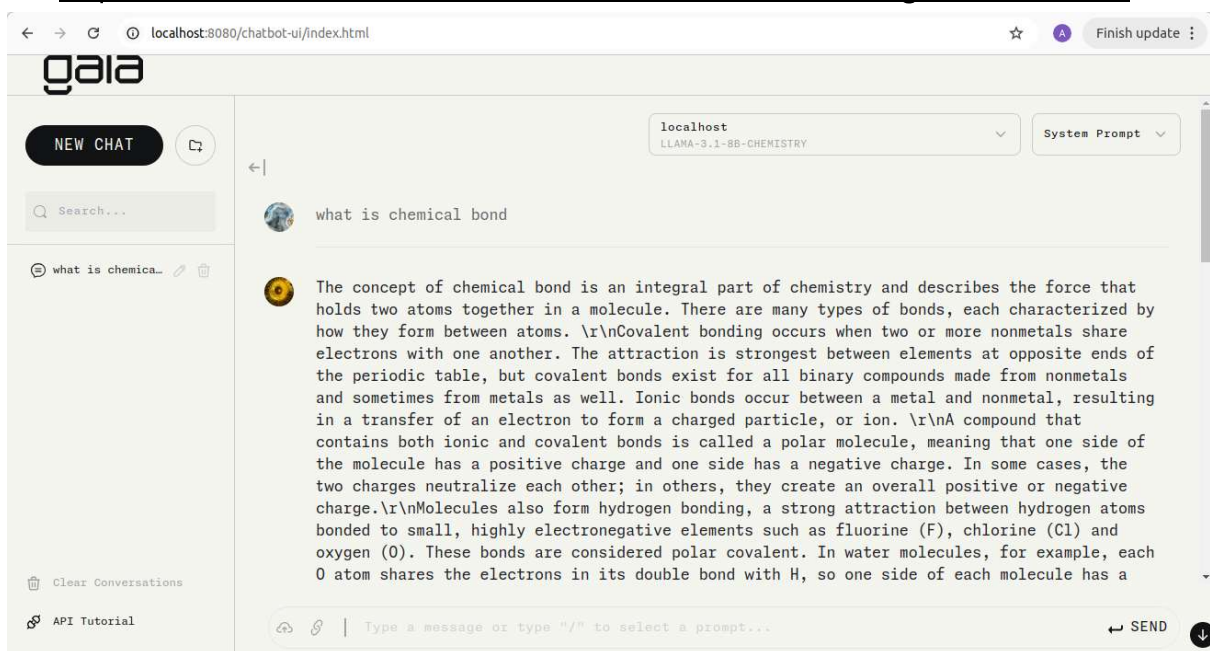


## Run an inference API server for the fine-tuned model

**The publicly accessible chatbot URL. A screenshot of the chatbot UI showing a relevant conversation**

```
root@c79836b07ba: /
[+] Preparing the GaiaNet domain ...
 Done!
 COMPLETED! GaiaNet node is initialized successfully.
 To start the GaiaNet node, run the command: gaianet start
root@c79836b07ba: /# gaianet start
[1/4] Checking the config.json file ...
 Done!
You already have a private key.
[2/4] Starting LlamaEdge API Server ...
/root/gaianet/bin/gaianet: line 1022: lsof: command not found
* Start server with the command below ...
wasmedge --dir ../dashboard --env NODE_VERSION=0.4.27 --nn-preload default:GGML:AUTO:unsloth.Q5_K_M.gguf --nn-preload embedding:GGML:AUTO:nomic-embed-text-v1.5.f16.gguf
llama-api-server.wasm --model-name Llama-3.2-3B-Instruct,Nomic-embed-text-v1.5 --ctx-size 16384,8192 --batch-size 128,8192 --ubatch-size 128,8192 --prompt-template llama-
3-chat,embedding --include-usage --web-ui ./ --socket-addr 0.0.0.0:8080
 Done! LlamaEdge API Server started with pid: 2212
* Verify the LlamaEdge API Server. Please wait seconds ...
 Done! LlamaEdge API Server is ready.
[3/4] Starting gaia-frp ...
 Done! gaia-frp started with pid: 2250
! The GaiaNet node is started at: https://0x418523ab85d0373d84e4f48b15d019e3a001f19.gaia.domains
[4/4] Starting Server Assistant ...
 Done! Server assistant started with pid: 2265
 COMPLETED! GaiaNet node is started successfully.
 To stop the GaiaNet node, run the command: gaianet stop
! You can close this terminal window safely now.
root@c79836b07ba: /# gaianet status
Usage: gaianet {config|init|run|stop|OPTIONS}
```

url: <https://0xa9f7efaa07879d7e3276e8626852de6758bd0321.gaianet.network>



## Conclusion

The project “**Building and Deploying a Conversational Chatbot with LLaMA**” successfully demonstrates the end-to-end development lifecycle of an intelligent AI-driven chatbot system using the Meta-LLaMA-3.1-8B-Instruct model. Through a structured and modular approach across three projects, we achieved significant milestones in model deployment, external knowledge integration, and custom fine-tuning.

- In **Project 1**, we set up the LLaMA model using WasmEdge in a Dockerized environment, enabling low-resource, portable inference through WebAssembly. This laid the foundation for prompt-based text generation and chatbot interaction.
- In **Project 2**, we extended the model's capabilities by integrating a **vector database (Qdrant)** for Retrieval-Augmented Generation (RAG), allowing the chatbot to respond accurately to domain-specific queries by referencing an external knowledge base. This significantly improved the contextual relevance and factual grounding of responses.
- In **Project 3**, we created a custom dataset and fine-tuned the LLaMA model to specialize in political news, showcasing the power of domain-specific adaptation. We hosted the fine-tuned model and made it accessible through an inference API, a public chatbot interface, and even a Discord bot.

This project not only provided practical experience with modern AI toolchains (Docker, WasmEdge, Hugging Face, Qdrant, and RAG architecture) but also emphasized the importance of scalable deployment and accessible UI integration (LobeChat, Discord). Overall, it demonstrates how open-source LLMs can be customized and deployed in real-world, resource-constrained environments to create intelligent, domain-aware chatbots.