**Name: Neha Joshi**
**UIN: 734003138**
**HW3 Report**

---

HW3 includes creation of a LLM Personal Assistant that has the capabilities of:

1. Sending emails
    a. Finding the contacts from directory
    b. Drafting a subject, body and sending the email
2. Scheduling meeting
    a. handling conflicts with existing meetings on calendar
    b. Emailing the invite to attendee
3. Searching the internet
    a. summarizing the search
    b. giving top 10 search result links and snippets
4. Answering questions based on multiple pdfs
5. Asking questions whenever required
6. Everything runs locally- Your information is safe!

To run the code and get the tasks done, please implement the following steps:

1. Get credentials.json file from your google calendar API and paste it in the confidential folder.
2. Get your google application password and paste it in confidential/email_pass.txt
3. Get your google custom search api and paste it in confidential/search_api.txt
4. Get your google custom search cx and paste it in confidential/search_cx.txt
5. Create a database of your contacts and paste it in confidential/email_pass.txt. The format should be like:
   NAME EMAIL
   abc: vedantjoshi370@gmail.com
   Neha Joshi: nehayj100@gmail.com
6. Add all the pdf files you need to answer your questions in paper folder.
7. Install all dependencies from requirements.txt
8. The pipeline.py file has the main code- Run python pipeline.py
9. Now add the tasks as the following tasks and let the assistant work!
    a. Send an email to abc saying Hi
    b. Schedule a meeting
        i. This now asks you questions regarding the meeting
        ii. Further you are asked if the tools should recommend slots- if you say yes- the tool will find a free slot from your calendar and recommend it.
        iii. If you have a time in mind- you will be allowed to add it and then the recommendation feature is not used.
    c. Answer using pdfs: “What is SimCLR?”
    d. Search the internet: “Who is Kamala Harris?”

10. All the above tasks will be performed with an average of the following mentioned latency. As mentioned, the tool will keep asking questions to the users wherever necessary.

Tool Details:

1. Llama 3.2 from ollama
2. Python and Goog;e APIs for email, calendar and custom google search
3. Chroma DB for RAG

Latency:

| Send email | Schedule meeting | Answer using pdf | Search internet |
|---|---|---|---|
| 62.578 | 58.85 | 13.318 | 18.869 |

Challenges faced:

1. The major challenge was writing a right prompt. Although I can explain the task to the LLM in a long detailed prompt- I realized that a short and precise prompt worked the best.
2. Latency is not constant – the same tasks sometimes take 60 seconds while sometimes its 15 seconds. The good thing is it takes no more than 60 – 75 seconds in my case.
3. Increasing the number of actions led to the LLM diverge from the task. TO handle this- short and precise prompts helped me.
4. In email application, the LLM diverged from the contact name ,may times. Sometimes, the output was the correct email id but was given in a lost format which led to errors. To handle this a feedback loops helped- The LLM knows what error it did the last time.
5. Another challenge was speed. When I was using LLAMA 3.1- I was getting relatively lower speed but upgrading to LLAMA 3.2 helped.

Improvements possible:

1. Latency reduction.
2. Reduction in divergence of LLM's thoughts. The chain of thoughts can be modified a bit to reduce this.
3. More depth in actions- for instance if the schedule meeting, you can create a google meet or zoom link and add it.
4. Adding voice API so you don't even type the task – just speak it.
5. Using a smaller model (like Llama 3.2 1B) to decrease the latency.