Name : Zekaj Vorname : Butrind
Name : Richani Vorname : Oliver

Assignment 4

1.

a) IT infrastructure requirements:

Server with REST API architecture hosted locally (within the company's intranet) with **Node.js**. It will provide the webpages of the app and coordinate the different requests handling within the back end. Domain name can be locally set. The data should be hosted on a **DB** (**MySQL or MongoDB**) which will need to not be connected to the internet for security purposes. The DB shall be separated in at least two separate instances, one for company documents and one for user credentials. The REST API and the DB can either be hosted on the same machine or on different machines, depending on what the IT infrastructure of the company is like. Separated would be ideal for compartmentalization and possible scaling. We will need at least 8GB of RAM or 4GB GPU and for storage space it will depend on the number of documents and company employees, but we will probably need at least 500GB. The OS deployed on the machine would ideally be Debian for compatibility and ease of use.

2a)

i)

- **Feature 1: Retrieve document**

  The user writes a prompt in the chat (here, asking the assistant to retrieve a document for him by its name). The prompt is sent to the company server where the REST API will interpret it as a user prompt it redirects it to the LLM API. The LLM API feeds the user prompt to the LLM, which will output a text. The LLM API interprets the text and standardizes the appropriate request that must be done and returns it to the REST API.

  The REST API will continue the request by asking for the document to the DB API, which will provide it after having checked the access rights of the user. The document is found and finally sent back to the user, with the text output of the LLM by the REST API.

- **Feature 2: Summarize document**

  The user writes a prompt in the chat (here, asking to summarize a document or a text). The prompt is sent to the company server where the REST API will interpret it as a user prompt it redirects it to the LLM API. The LLM API feeds the user prompt to the LLM, which will output a text. The LLM API interprets the text and standardizes the appropriate request that must be done and returns it to the REST API.

  Here the REST API can directly send back the text output (which contains the summary) to the user.

- **Feature 3: Send notifications**

  The user writes a prompt in the chat (here, asking to send a notification to other users). The prompt is sent to the company server where the REST API will interpret it as a user prompt it redirects it to the LLM API. The LLM API feeds the user prompt to the LLM, which will output a text. The LLM API interprets the text and standardizes the appropriate request (by filling-in the appropriate template with the information the user gave) that must be done and returns it to the REST API.

  The request is forwarded to the DB API by the REST API, where the recipients list will be parsed to determine which users have to be notified. Once the recipients IDs have been found in the DB, they are sent back to the REST API which can formulate the notification and send it to each of the user IDs in the list.

- **Feature 4: Schedule**

  The user writes a prompt in the chat (here, asking schedule a meeting). The prompt is sent to the company server where the REST API will interpret it as a user prompt it redirects it to the LLM API. The LLM API feeds the user prompt to the LLM, which will output a text. The LLM API interprets the text and standardizes the appropriate request (by filling-in the appropriate template with the information the user gave) that must be done and returns it to the REST API.

  The REST API will have to ask the DB API to save the meeting information in the DB, and then to retrieve the participants of the meeting to notify them. Once the DB API has saved the meeting information and determined the user IDs of the participants, the REST API can send a confirmation to the sender and a notification to all the participants of the upcoming meeting.