Course Project Contribution Agreement

This agreement is made between the student(s) listed below to document each team member's contributions to the course project. It outlines the individual roles, contributions and involvement of each student in the project.

Project Title: Pharmaceutical Risk Interaction System Management

Team Member: Celian, Sebastian, Isac, Clement, Mahmut

Description of Contributions:

By signing this contract, each team member agrees to the following contributions to the project:

Mahmut

- Organized meetings & GitHub issues.
- Fixed Workflow Diagram. Defined Project Milestones. Explored to-be-used project tools. Created SPARQL queries that accomplish the following tasks:
    - Get the English drug label given the internationally set drug id.
    - Get All Interacting Drugs with respect to another Drug (e.g., Zopiclone)
    - Get Side Effects of a Drug (e.g., Zopiclone)
    - Get all instances of "medicines" in the WikiData Knowledge Base.
    - Get the count of all medicines within WikiData.
    - Find Chemical Substance used as Medicine through Query by name.
    - Get Best Matched Meds based on Wrongly Type Input Medication.
        - Tools: Python Module "*fuzzywuzzy*"
    - Saves the interactions between proposed medicine and the ones it has a significant reaction with creating side effects.
- Created a Pinecone index.
    - Embedded all medications and upserted them into the Pinecone database, within a predefined namespace.
    - Queried the namespace through the use of similarity search, works.
- Built Chainlit basic interface and seamlessly integrated it with newly release Gemini 2.0 through API calls.
- Built Chainlit basic interface and seamlessly integrated it with local models phi3 and llama3.2:1b through API calls.
- Created a new workflow diagram.
- Created a Custom Logo for our Application, both for the dark and light color setting in our UI. Integrated it with the chainlit application, works.
- Created a more intuitive UI, played with welcome text, user prompt, including font size, colors, etc.
- Added initial prompting questions in UI to user, extracts current medications and the potentially new medication, uses fuzzywuzzy to match against existing medical substances in knowledge base, works.
- LLM extractions before fuzzywuzzy for extra precision in extractions

- Add the queries for side effects, display results
- Added radio buttons "Ask for further details" and "Query again". The first saves the current context and prompts Gemini 2.0 via an API call with the new user question and the save context. Query again clears the current context and prompts the user to re-enter medications of interest.
- Fixed four starter profiles, integrated current "side-effects-profile" into base-architecture. Removed unnecessary files. Added icons for each profile. Reviewed pull-requests.
- Try-catch: ed getting the list of all medications from the knowledge-graph, if it fails, defaults to most recent list of medications extracted from the Knowledge-graph Wikidata.
- In large parts finished the general chat. Added general chat integration with rest of application, added a general chat prompt so as to make the LLM respond as expected. Lastly, conducted test, does it respond as we expect? Does it recall context in past messages? All worked fine.
- Implemented Medication Recommendation chat. Symptoms are inputted, converted into Latin, best match against SPARQL queried symptoms CSV using fuzzywuzzy, returns corresponding medicine for the input. Necessary prompts are also created.
- Created workflow diagrams for the overall application, and three out of four chats.
- Read research papers on erroneous medication prescriptions, rewrite the introduction and included these sources.
- Created the final presentation and split up the work.
- Wrote on the report.

Isac

- Create a Logo for our Project.
- Fixed Workflow Diagram. Defined Project Milestones. Explored to-be-used project tools.
- LLM extractions before fuzzywuzzy for extra precision in extractions
- Implemented proof of concept using local llama3.2 LLM and SPARQL data
- Add the queries for side effects, display results
- Implemented Medication Recommendation chat. Symptoms are inputted, converted into Latin, best match against SPARQL queried symptoms CSV using fuzzywuzzy, returns corresponding medicine for the input. Necessary prompts are also created.
- Explored gemini system instructions, tested profile selection with ai prompt, not sufficiently precise.
- Bugfix for chat reset when changing profile.
- Fixed overall workflow workflow diagram, general questions workflow diagram, side-effects-identifier workflow diagram and medication recommender workflow diagram.
- Created the final presentation and split up the work.
- Wrote on the report.

Sebastian

- Fixed Workflow Diagram. Defined Project Milestones. Explored to-be-used project tools.
- Created a query for getting drug interaction & its side effects, results are stored in a separate Json file.
- LLM extractions before fuzzywuzzy for extra precision in extractions
  Add the queries for side effects, display results
- Fixed four starter profiles, integrated current "side-effects-profile" into base-architecture. Removed unnecessary files. Added icons for each profile. Reviewed pull-requests.
- Try-catch: ed getting the list of all medications from the knowledge-graph, if it fails, defaults to most recent list of medications extracted from the Knowledge-graph Wikidata.
- Alternative Medicine Chat Profile => Created appropriate SPARQL queries for the task, Prompt Engineering with regards to user input as well as gemini output.
- Created the final presentation and split up the work.
- Wrote on report.
-

Célian

- Help to find subject for the project.
- Created the final presentation and split up the work.
- Wrote on the report.
- More testing and understand code, for technologies and concepts of the project. And try to improve some methods of coding and use some tools.
- Help to fix the bug we talked about during the meetings.
- Reviewed SPARQL queries.

Clément

- Wrote on the report.
- Created the final presentation and split up the work.
- Contributed to Medication Recommender, added the symptoms and signs query and parsing.
- Tried to add example of messages to test and give user examples but they disappear after the first instructions are send so we removed them.

Project Owner: Mahmut

Scrum Master: Sebastian

Development Team: Celian, Sebastian, Isac, Clement, Mahmut

Points of Agreement

- Each student must ask other team members for assistance if they get stuck.
- Each student shall attend all meetings unless a valid excuse is given.
- Team will work in an agile manner.

Project Meeting Log

As part of this contract, the project team is required to document the project meetings organized, the members who attended, and the topic(s) discussed. Will be submitted through an attached excel file.

Acknowledgment of Accountability:

By signing this contract, each undersigned student acknowledges and agrees to the following:

- **Balanced Contributions**: All students contribute fairly and equitably to the project. We have completed our assigned tasks on time and have actively contributed to the project.
- **Project Meetings**: Each student attended and participated in project meetings. We have documented all meetings attended and tracked the action items assigned to us.
- **Timeliness and Integrity**: We confirm that we have met all deadlines for our tasks, contributed fairly to the project, and ensured all work is our own. We understand that failure to meet these expectations may affect our individual grades or the overall project grade.

Signatures:

By signing below, we confirm that the information provided in this contract is accurate and that each of us has contributed as described above. This agreement is made in good faith to ensure accountability for each team member's role in the project.

Student Name _____

Signature _____

Student Name _____

Signature _____

Student Name _____

Signature _____

Student Name _____

Signature _____

Student Name _____

Signature _____

Date: _____