School of Computing
CA326 Year 3 Project Proposal Form

SECTION A
Project Title ____ChatSQL : Smart data chatbot____.
Student 1 Name____Georgijs Pitkevics____ ID Number 19355266
Student 2 Name____Chee Hin Choa________ ID Number _21100497_
(A third team member is exceptional and requires detailed justification.)
Staff Member Consulted____Brian Davis_____.

Project Description (1-2 pages):

Description

Minimum 250 word description of the proposed project.

ChatSQL is a smart data assistant that redefines how users interact with databases. ChatSQL is a user-friendly assistant that the user can talk to and engage in conversation about data and databases, the user will not require extensive knowledge of databases or SQL.

The core principles of ChatSQL:

1. User-Centric Design: ChatSQL is catered to people with minimal IT literacy required. The only two skills required are the ability to upload a file to a web page and send a message, this ensures that users with minimal technical proficiency can interact with databases.
2. Understanding of Natural Language: ChatSQL is capable of understanding natural language and taking in the user's queries in English without the need for SQL input.
3. Data Privacy and Security: Data security is important in today's world, ChatSQL is designed to only query the database, without the ability to perform modifications or deletions of tables or data within the database to protect it from unintentional or malicious modification.

How it works:

1. User selects the database they wish to query. ie; covid-vaccinations.sqlite3
2. The user can ask the chatbot a question about the database, ie; "What was the average vaccination rate in Dublin?"
3. The chatbot queries our chosen LLM which converts natural language into an SQL query and replies with the result in natural language form. ie; "The average vaccination rate in Dublin is 95.4%"
4. Conversation can continue further if the user requires more information.

Example scenario:

A startup's Data Analyst is on annual leave. The Project Manager needs information on some data for an upcoming meeting with stakeholders. The data analyst has preloaded the databases onto ChatSQL. The project manager goes onto the companies configuration of ChatSQL and queries the chatbot about the data he needs for his meeting. ChatSQL allows the PM to understand the data so that he can get ready for his meeting without bothering the Data Analyst.

Division of Work

Outlines how the work is envisaged to be split equally among the team members.

We will work simultaneously together to approach this task so that we stay on the right track. The complex parts of the project can be grouped into:

1. “Natural language to SQL” LLM development, this may include Natural Language Processing (NLP) as well 1-2 LLMs to generate an SQL statement from a given sentence, with option to utilize the table names taken from the database for better accuracy and efficiency.
2. HTML/HTMX/TailWindCSS Frontend: UI/UX design, implementing the chat UI, making it easy to understand and use for first-time users. Tying it in with the backend with HTMX and WebSockets (Flask SocketIO).
3. Flask backend: Overall structure of the code to handle the POST/GET requests from the frontend, hosting the web application, handling networking to and from the client, and Implementing the “Natural language to SQL” within the Web Application as well as handling database access.
4. Security Layer: This will be an important undertaking in implementing security measures within the web application to ensure the user cannot modify or delete data from the database, ie; Filtering out and refusing to perform requests such as ‘DROP TABLE your_table_name’.

Chee Hin will undertake task 2, and Georgijs will undertake tasks 3 & 4.

While we are dividing the main workload, we will first develop the LLM functionality (Task 1) together in a pair programming approach to ensure we both understand how the essential functionality of ChatSQL works.

Programming language(s)

List the proposed language(s) to be used

- Python
- HTML
- HTMX
- CSS
- JavaScript (Optional)
- TailWindCSS
- SQL/SQLite/PostgreSQL Databases

Programming tool(s)

List tools (compiler, database, web server, etc.) to be used

- Github - for version control, tracking changes during development
- Visual Studio Code - IDE for writing, debugging, and managing your code effectively.
- Powershell/Bash Terminal - for testing the flask application locally during development stages.
- Potential research into utilising prisma db services to better visualize the models and their associated databases. ie; user login, chat history, SQL generation history with accompanying prompts.

Learning Challenges

List the main new things (technologies, languages, tools, etc) that you will have to learn

Developing ChatSQL will be an exciting and equally challenging learning opportunity for the two of us. Here are the main challenges that we expect to encounter:

1. HTMX: HTMX is a fairly new up-and-coming tool based on hypertext that handles "AJAX, CSS Animations, WebSockets and Server Sent Events directly in HTML". This will allow us to speed up the development of the application as it eliminates the need for JavaScript altogether, giving us more flexibility in our choice of technologies and time to develop/refine features.
2. TailWindCSS: TailWind is a modern CSS framework which offers a unique way of styling web applications directly within the html. We'll need to become familiar with this framework to design and appealing and easy-to-use interface.
3. Flask: Flask is a Python web framework that we'll be utilizing to create the backend of our web app. It offers a multitude of additional libraries such as 'Flask Socket-IO', 'Flask Login' and 'SQLAlchemy'. It will be quite challenging to learn how to utilize it for this project however we believe this is the best framework for this, as python is one of the best languages for utilizing LLMs and NLP.
4. User Experience: To stay true to our core principle of User-Centric Design, we need to research and understand how to create an intuitive user-friendly interface that is easy to use and requires no prior training by utilizing UI design guidelines as well as a simple prompting system that the average user can understand from the moment they enter the application.
5. Large Language Model (LLM): We will need to research LLM's and how they work. We need to decide whether we will use an off-the-shelf LLM like OpenAI's GPT4 API which would prove more efficient for hardware, meaning we would not have to worry about using powerful GPUs for computation, or if we would like to have the model run locally on the webserver, utilizing open source models like the many that can be found on huggingface.co and further trained.

6. Database Administration: To work with various databases, we need to understand how they work and what level of complexity in SQL statements we need to aim for when querying databases.

Hardware / software platform

State the hardware and software platform for development, eg. PC, Linux, etc.

Personal machines/laptops:

- Georgijs will utilize his Windows 11 Laptop and Dualboot (Windows 11 & Arch Linux) Desktop
- Chee Hin will utilize his Linux (Ubuntu 22.04.2 LTS) Laptop

We plan to look into jupyter notebook (as found in Google Collab) for research and development of the LLM functionality.

Special hardware / software requirements

Describe any special requirements.

- ○ Note 1 - In general the School of Computing is not in a position to supply and support special hardware / software for 3rd years projects. Accordingly, any special needs should be provided by the students and discussed with your supervisor.
- ○ Note 2 - It is assumed that all projects will be developed / demonstrated using standard lab machines. Students may use their own hardware, but all projects must be demonstrated in a School of Computing lab, either on a lab machine or the students own machine.

NOT APPLICABLE