**Student Name:** Felix Nguyen

**Student Number:** 222521972

**3.1C**

* Github URL for the quiz app:
* Github URL for the calculator app:

**Report on ideas to harness LLAMA2:**

The most important functionality of a quiz app is to provide entertainment or knowledge checking through a series of questions and answers. The Q&A aspect of this type of application resembles the kind of interactions between a chatbot and the client, which is one of the many forte that LLAMA2 models can take advantage of. In this report, I will be referring to “traditional quiz app” as those not powered by an AI element, and “AI-powered quiz app” as those leveraging on AI models such as LLAMA2. From the developers’ point of view, designing and developing a traditional quiz app involves manually maintaining the list of questions and corresponding answers. While this approach has worked over a few decades, the complexity of the app multiplicatively increases as the number of questions grows and introduces a number of nuances that LLMs can help resolve or improve.

First off, the traditional quiz app is expected to retrieve the set of questions and answers from a database. This allows developers to modify the questions and answers without requiring a rebuild and re-deploying to clients. However, maintaining a database for questions and answers retrieval requires a database with sufficient power to cater to heavy read operations, which will be costly to setup either on-premise or on the cloud. With LLAMA2 and other LLMs, questions and corresponding correct and misleading answers can be automatically generated and a database for storing the Q&A set is no longer necessary.

Secondly, maintaining the Q&A set introduces a lot of operation cost as the number of questions grows. Factual quiz rarely gets updated, but questions on fleeting matters in the world may have their correct answers changed quickly. For example, the answer to the question “Who is currently the richest man in the world?” can vary depending on the day or time when the quiz is accessed. For this reason, LLMs may make use of RAG technique to pull information from unstructured text, our written documents or online websites, to automatically generate the correct up-to-date answer without having to manually update the app’s database. As an example as to how LLAMA2 works with RAG, Andrew has published an article to cover the benefits of RAG and the technical setup requirements to facilitate this approach.[[1]](#ref1)

On the other hand, from the user experience, having to repetitively answer the same set of questions can be boring and unproductive. Even though developers may store a set of thousands of questions, there is still a limit to how many questions can be implemented for the traditional quiz application. Utilizing the high capability to generate natural language texts of LLM models, questions may be randomly generated to create more variations to the pool. Having varied question sets means that the users will actually have to understand the concept and the knowledge that the app is trying to deliver, instead of forcefully remember the answers.

**References:**

1. Andrew, “*How to use Retrieval-Augmented Generation (RAG) with Llama 2*”, agi-sphere.com, <https://agi-sphere.com/retrieval-augmented-generation-llama2/>, accessed on 7 Apr 2025.
2. Quiz App: https://github.com/hellosagar/Quiz-App/blob/main/app/src/main/res/drawable/arrow.xml