The App Features, Modern Android Practices and LLMS

The Features:

User Creation:

A user account can be created by entering details such as a unique login, username, email, password and phone number. The database stores this information, and it's necessary to use these details to login to the app and access the content. This ensures that each user has access to a learning path design for them, and no one else can access their details.

User Interests:

The user can select the topics that interests them the most. There were only 4 options displayed, but if the app ever goes live, more diverse options can be added.

Loging In:

Loging in requires the valid user details. A username and password. If the wrong details are provided, the user can't enter the app and use it's features.

Quiz Generation:

The Quiz Generation in the App was intended to be generated by LAMA, but as the deadline approached 6.1P, I had to settle for getting the data through Volley Requests to the Open Trivia DB API. This was due to the app being run on my actual device, which was causing issues with using LAMA (The RAM was too low to use a Virtual Device for testing purposes)

Doing the Quiz:

The user is able to choose their quiz, and answer the questions generated. They get 3 options and can only choose 1.

Results Recording:

The user is able to see how they went, what questions they got right, and what they got wrong. As well as what the actual correct answer was so that they could learn from their mistakes.

Results Recording:

The user is able to see how they went, what questions they got right, and what they got wrong. As well as what the actual correct answer was so that they could learn from their mistakes.

<u>Upgrading Payment:</u>

The user is able to choose from 3 types of upgrades, and make the relevant payment using Stripe.

<u>Sharing User Details:</u>

The user is able to share their account details through different mediums, such as messenger and WhatsApp.

Adhering to modern android development practices:

Modern android development practices were adhered to, by using well-structured and documented code. Since the code is so compartmentalised, and the variables are named relevant names, the code can easily be further developer by another developer.

Relevant techniques, such as requests through Volley, using JSON Data, and Stripe Payment were used. The UI was also easy to use and flowed well.

User data was also handled well since SQLite was used to store data, and variables that needed to be private, were made private.

CI/CD practices were followed through the use of GitHub. The project was regularly backed up and appropriate commenting was provided do that context and understanding was given to the different 'pushes' to the final repository. This is good in case there is a need to go back in time and understand a feature more.

How LLMs can be used for further improvement

LLMS can be used for further improvement by generating more questions, and questions that adjust to the difficulty of the user as they answer more questions. Maybe more engaging activities could be created, not just quizzes, but something like 'complete the sentence', or 'hangman'. LLMs could be used so that the user can 'chat' with an AI agent that has a lot of knowledge about a particular area, so that the user can learn more.

The quiz could be translated to different languages so that users are able to use the app in the language they're familiar with, thus increasing usability.

They could also be used for security, e.g. asking the user questions that only they would know, in case they forget their password.