Participant: Kizito Richard

**Theme 1: Project Overview**

1. Question: Could you provide an overview of your project, "Rugby Mobile"?

- Answer: "Rugby Mobile" is a sports application focused on providing rugby information, including statistics, live scores, team details, and player information, primarily for Uganda Rugby Union. It also incorporates a machine learning model for predicting match outcomes, enhancing sports betting and fan engagement.

2. Question: What was the process of developing the project, from concept to implementation?

- Answer: We began by creating a concept paper, then submitted it for approval. Once approved, we worked on a proposal paper, detailing the steps to develop a mobile application and integrate a machine learning model. Research was conducted on obtaining data for both aspects. However, it's important to note that we faced challenges in the initial concept approval stage, which added complexity to the process.

**Theme 2: Data Challenges**

3. Question: What challenges did you face when obtaining data for your project?

- Answer: Data acquisition posed significant challenges. People were hesitant to share data, requiring proof that the project was legitimate and intended for educational purposes. This verification process sometimes caused delays and confusion. Additionally, some data sources had limitations, affecting the completeness of our data.

4. Question: How did you handle the issues related to data acquisition and prove that your project was for educational purposes?

- Answer: We had to address these concerns by providing evidence and assuring the data providers of our legitimate intentions, which could be time-consuming. We also had to explore alternative data sources to compensate for the limitations of some datasets, adding complexity to the data collection process.

**Theme 3: Technical Debt in Implementation**

5. Question: Could you share examples of technical challenges or errors (technical debt) that you encountered during the implementation of your project?

- Answer: While developing our prototype from scratch using Flutter, we didn't face complex bugs since we were writing most of the code ourselves. However, adding new features posed time challenges, and one instance where a feature addition significantly extended our project timeline was mentioned. Additionally, there were instances where our initial code structure had to be refactored, leading to technical debt.

6. Question: How did you manage time constraints and deal with the challenge of adding new features while maintaining code quality?

- Answer: Adding new features within time constraints required a lot of effort. For example, we requested additional time when it became clear that certain new features were necessary for the project's success. This extension helped us avoid compromising code quality due to rushing. However, it's worth noting that accommodating these changes also added to the overall complexity of the project.

**Theme 4: Importance of Documentation**

7. Question: Did you maintain documentation throughout the development process, and how important was it in addressing technical gaps?

- Answer: Documentation was crucial. While developing the mobile application and machine learning model, we documented our process and any changes made. It helped us understand and address technical gaps, especially when we revisited certain aspects during the project. However, there were instances when keeping documentation up-to-date became challenging due to the rapid pace of development.

8. Question: How did you ensure that documentation was consistently updated as the project evolved?

- Answer: We included documentation as an integral part of our development process. Whenever we made changes or additions, we updated the documentation accordingly to ensure it remained accurate and useful. However, maintaining documentation sometimes required additional effort, which could have otherwise been allocated to development tasks.

**Theme 5: Addressing Technical Debt**

9. Question: What strategies or approaches did you use to prioritize and address technical gaps in your project?

- Answer: Technical gap prioritization mainly occurred during the implementation phase. We prioritized tasks based on the impact on the project's quality and reviewed code to identify critical gaps that required immediate attention. However, as the project progressed, we encountered situations where addressing all technical debt became challenging due to resource constraints.

10. Question: In your experience, how does addressing technical debt early in a project affect its overall quality and success?

- Answer: Addressing technical debt early is essential for maintaining high project quality and meeting deadlines. Failing to address issues promptly can lead to a decline in project quality and increased challenges down the line. However, it's important to note that even with early awareness, resource limitations can impact the ability to address all technical debt effectively.

**Theme 6: Framework and Future Development**

11. Question: Can you suggest insights or additions to a framework that would help young developers effectively manage technical debt in their projects?

- Answer: Understanding the importance of code and acknowledging it as a valuable asset is essential. A framework could emphasize the significance of comprehending code to reduce technical debt effectively. Additionally, young developers should be encouraged to proactively identify and address technical debt during the development process.

12. Question: Do you think that awareness of emerging trends and tools in the IT field should be integrated into such a framework for young developers?

- Answer: Yes, it's crucial to keep such a framework updated with emerging trends and tools in the IT field. Being aware of these trends can help developers work more efficiently and maintain high code quality. However, it's also important to strike a balance between adopting new technologies and ensuring they align with the project's goals and constraints.