BrainStorming Summary:

- As an IT team, we conducted 2–3 formal brainstorming sessions along with several informal discussions.

  We considered many ideas, rejecting a few due to impracticality or feasibility concerns.

- In the very first session, we mainly focused on the UI features of the application. We discussed that our app should look modern with a minimalistic design and professional color palette while still providing a professional and smooth experience. All features should be easily accessible and presented in a clean, organized manner. We analyzed the UI/UX of various applications that offer messaging features, with particular attention to the design of passive message handling.

- Another important aspect we brainstormed was Privacy, as it directly influences the number of users who would trust and adopt our app. Initially, we considered encrypting all messages before storing them in the database. However, due to the high computational load of frequent encryption and decryption, we decided instead to store messages locally on the user's device. As application managers, we will only store the User ID along with the password, email addresses, and mobile numbers. This approach reduces the system load while also maintaining user privacy, provided the API calls are made securely. Additionally, the user should be able to log in to their device securely.

- We also discussed that scam messages are not limited to English. A large number of scams are conducted in different regional languages. Therefore, our model must be capable of identifying suspicious messages in multiple languages. Similarly, many users are not comfortable with English, so the app's UI should support multilingual options. This would not only improve accessibility but also increase user diversity and adoption.

- Users should be able to connect multiple email accounts, since most people use more than one email ID. For phone numbers, however, the limit will depend on what the mobile device itself supports. To fetch SMS, the app requires the SMS service to run locally on the device. With the right permissions, the app should automatically scan both emails and SMS messages as they are received. Based on the severity of the detected scam or phishing attempt, notifications should be sent promptly to alert the user before any potential damage occurs.

- Apart from this, we discussed some additional features. If the user chooses not to grant permission to scan their emails or messages, they should still be able to manually input messages and access the same features, such as Summary/Explanation and Text Highlight. For messages with a very high potential for fraud, users will be notified if permission is granted.

- The Text Highlight and Explanation functionalities will also be available for previously scanned documents. On the main dashboard, users will be able to see a division of their messages into Scam, Safe, and other categories through graphical representations, along with additional insights.

- In the final brainstorming sessions, we concentrated on the ML model, which is the core of our application since it performs the scam detection. The model should deliver results as quickly as possible while maintaining high accuracy. It also needs to adapt to new and evolving scam techniques. To support this, we agreed on the importance of a feedback loop, where users can provide input on the confidence scores generated by the app. This feedback will later be used to retrain and improve the model, ensuring it remains effective against future scams.