Email-assistant

- *What we have:*
  - Assume there are some financial news emails coming from certain email addresses, each containing text and images for multiple events. Financial Email Example
- *What we need:*
  - Access all such news email messages inside the user inbox from the past 24hrs, break down all messages into events, and display the events in a table.
  - The table needs at least 2 columns: events, and any source link address if exist.
  - In the table of events, merge and summarize repeated events.
- *Important UI/UX features:*
  - When the user hovers over an event in the table, display a small side panel that shows source email titles.
  - When the user clicks on a specific email title from the side panel, jump to the email content and highlight the corresponding original text in the original email.
  - We want to generate the summaries asynchronously at a certain time at night, and have the results ready for access in real-time in the morning.
  - We want the app built as a Chrome extension and connects to their personal Gmail or Outlook account.

Report Generator

- *What we have:*
  - a chatbot that allows single correspondence (i.e. one question from user and one answer from the chatbot).
- *What we need:*
  - when the chatbot answers further questions, integrate memory from previous Q&As for context.
- *Important UI/UX features:*
  - The stack/architecture needs to be prepared for long memory context and maintaining details from previous answer.
  - Display stages of workflow in real-time (e.g. querying database, found data, etc.) or stream answer.
  - Leave room for tapping into external APIs for data, which will be used in our RAG workflow.

Questions for Engineering Team

- What are the considerations for recommending certain architectures? For example, are vector databases the most *economical and efficient* choice, or is there a special problem structure we may rely on for the mapping-to-original-source function?
- What are the considerations for recommending a specific stack over others? For example, which vector search database (e.g. PineCone vs Milvus vs PostgreSQL's pgvector), which embedding model, etc.

- What are the considerations for recommending cloud computing platforms and services on those platforms? (We are currently using AWS Lambda, PostgreSQL, and SQS for the backend)