Silicon Slippers

“Discover Your Perfect Financial Fit with Discovery Bank: The Silicon Slipper for Your Wealth Journey.”

Ryan Murphy - Tumi Mokoka - Michael Gamsu

Problem Statement:

What we aim to address :	**How we aim to address it:**
Cost of Financial Advisors	Providing financial advice to low/middle income households who wish to improve financial situation but cannot afford personal financial advisory - free to use and easy to understand
Availability	24/7 personal support, instant responses
Insights into customer behaviour	Discovery is a behavioural bank and this platform will provide feedback on customer needs and behaviour
Cost effectiveness	Provides cost-effective and high-quality personal financial advice to a larger number of clients
Navigating discovery services	Personalised recommendations for specific discovery financial plans and services

Challenges we Faced

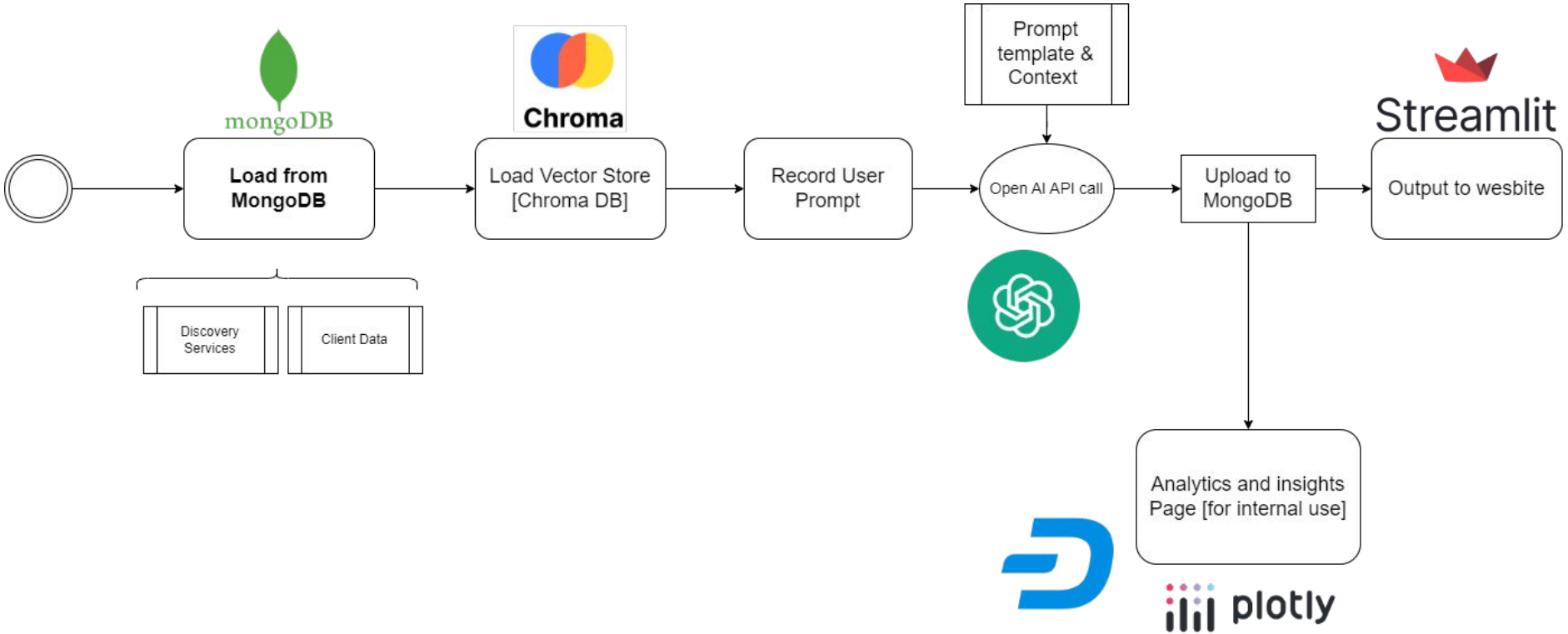
- We faced numerous challenges over the weekend.
- One of the main challenges was to get the chatbot to give personal advice and not generic advice.
- All being aligned with the same vision for the project
- Conflicting ideas & opinions from mentors
- Integrating the pdfs into MongoDB and aligning that with Chroma DB.
- Difficult to test because outputs are non-deterministic
- Not a lot of sleep.

How we Overcame Them

- Luckily, we managed to get over the hurdles.
- We did some precise fine tuning of the model by using clever prompt engineering.
- We all decided on a clear solution by Saturday morning.
- Taking everything in consideration, we had to believe in our vision and focus on the goal.
- Watching and following many tutorials on how to configure each DB.
- Had to ask it loads of questions and fine--tuning it to when it gave us a good answer
- Red Bull.

Technical Implementation

- The main driver in this project is a Python library called Langchain. This allowed us to create a chat bot without training a GPT.
- We used MongoDB to store all of our unstructured data about discovery and our client.
- OpenAI embeddings. This is what put our words and tokens into vectors to be stored in a vector db.
- We used Retrieval Augmented Generation(RAG) with a vector database called Chroma to pull the write content based on the question.
- We used a GPT- 4 API to output the answer to the user.
- For the front end we used a library called Streamlit for the chatbot interface and we used the libraries Plotly and Dash for the analytics front end.


mongoDB
Load from MongoDB
Discovery Services
Client Data
Chroma
Load Vector Store [Chroma DB]
Record User Prompt
Prompt template & Context
Open AI API call
Upload to MongoDB
Streamlit
Output to wesbite
Analytics and insights Page [for internal use]
plotly

How This Product is Beneficial

How are we going to make Discovery Money?

- This product will help people to save money which will result in Discovery having more money.
- This is a unique product that no other bank has so it could attract more customers and increase customer loyalty.
- Good way to onboard new clients
- The analytics section can help Discovery gain more insight to their customers
- Discovery could choose to charge for this application which would increase their revenue.

Next Steps

How we'd integrate this with Discovery

- Integration with discovery customer profiles
- Fine tuning an LLM with a deeper understanding of the Discovery ecosystem
  - Integrate with other parts of discovery like medical aid and Vitality
- Take the POPI act into consideration
  - Exchanging open-source and API tech with internal resources to ensure security and privacy
  - Session-by-session private keys (no data storage beyond session)
- Improve insights dashboard

Q & A