**Frax Telegram Bot Project**

1. The data is currently stored in my Mongodb Atlas. The access link is hard-coded in both script\_mongodb.py and db\_monitor.py. You can change it as required.
2. Database structure

**frax *(database)***

**wallets*****(collection 1):*** *Frax data as provided*

*document keys:*

**\_id:** *Autogenerated Mongodb id*

**wallet\_address**

**Feature\_1**

**Feature\_2**

**Feature\_3**

**Feature\_4**

**subscription *(collection 2):*** *Tracks which user is subscribed to which wallet*

*document keys:*

**\_id :** *Autogenerated Mongodb id*

**wallet\_id:** *Maps to wallet address in “wallets” collection*

**user\_id:** *Maps to unique id of Telegram user who has interacted with Bot*

**doc\_id:** *Maps to “\_id” in “wallets” collection and corresponding to wallet\_id*

**metadata *(collection 3):*** *Stores data fetched from Telegram API depending on interactions with db\_monitor bot*

*document keys:*

**\_id:** *Autogenerated Mongodb id*

**message\_id**

**user\_id:** *Unique for each Telegram user*

**update\_id**

**is\_bot**

**first\_name**

**last\_name**

**date**

**user\_account\_type**

**message**

**language**

**queried\_wallet:** *Helps track which wallet the user is interested in*

**frequency:** *User is automatically subscribed to a wallet (i.e. queried\_wallet) if frequency >= 3*

1. Steps for Execution:
   1. Create a virtual env
   2. Install required packages from requirements.txt
   3. Open 2 cmd windows and activate the virtual environment in both
   4. In one window, run **!python script\_mongodb.py** to establish backend functions of Telegram bot
   5. In another, run **!python db\_monitor.py** to track changes made to the “wallets” collection in “frax” database
2. Experimentation:
   1. Viewing and subscribing to wallet through Telegram:
      1. Open Telegram app and search for **new\_db\_monitor**
      2. To View the last entry for a wallet id, use format **/select {wallet\_id}**
      3. To subscribe to a specific wallet, use format **/subscribe {wallet\_id}**
      4. To unsubscribe to a specific wallet, use format **/unsubscribe {wallet\_id}**
   2. Receiving notifications for changes to subscribed wallets
      1. Make sure that you are subscribed to the wallet you want to receive notifications
      2. Make changes (update, replace, delete) to the wallet through Mongodb database