Current architecture :

Data mgmt :

1. Screenr.com webscraped using making\_bse.py
2. Columns sorted using sorting\_bse.py
3. Modelling performed using modelling.py ( 1000 x 134 granularity)

The resulted dataframe from modelling.py is stored in a csv file called bse\_modellingX.csv.

1. API calls made to forumscout using free tier access token (.py) and added to database.
2. Type discerned using sentiment\_{}search.py
3. Finbert sentiment analysis peformed using sentiment\_joiner.py ( 3000 x 12 )

The resulted dataframe from sentiment\_joiner.py is stored in a csv file called sm\_sentimentX.csv.

Database : (port 3306)

1. MySQL installer : <https://dev.mysql.com/downloads/installer/>
2. This downloads MySQLserver and workbench.
3. Option to create server given once when installation is happening.
4. This server image is used to create docker container using docker-compose.yaml.
5. Status of The yaml file also mentions the local mounted folder for data persistency.
6. server can be checked using services in settings ( MYSQL80 )
7. MYSQL workbench is used to connect to MySQLsever ( username : root , password : stockpulse )
8. Set database to default
9. Fillmeup.sql script used to populate database(FinancialData + SentimentData)
10. csv files uploaded to local mounted folder and then uploaded using script.

Docker-compose.yaml file starts 3 containers the API py backend, database and R server.

FLASK py backend : ( port : 5000 )

1. Connects to the database to obtain financial data table + (sentiment data)
2. /stocks GET API : returns entire Name column from database table
3. /stock/{stock name} GET API : returns all columns for that specific stock row
4. /stock/subindustry/{stock name} GET API : returns all stocks in the same sub industry as the sent stock name . all stocks contain all columns.
5. /predict POST API : sends all stocks in same sub industry, then runs ML model and returns predicted stock price.

PLUMBR R backend : ( port 8000)

1. RMySQL , DBI packages use to connect to db
2. 6 visulizations plots API generated and hosted locally on Rstudio plumber APi

Frontend :

1. Home.vue : main opening page of the app
2. Homecom.vue : modelling page ( box insertion for changing growth rates n more )
3. Streetcom.vue : relative valuation page ( ML model perfection )
4. Aboutcom.vue : sentiment analysis page ( upvotes , retweets not in picture )
5. Login.vue : login page ( 2FA email authentication mobile sms )