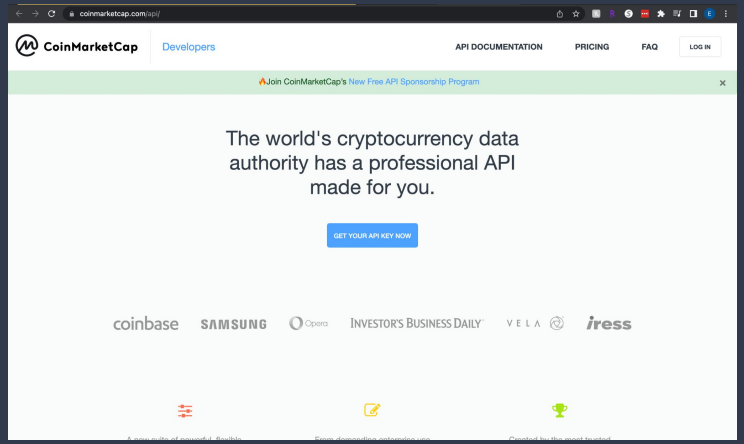
Project 3 - Cryptocurrency Dashboard

Elza Hayyat, Gustavo Gyotoku, Xander Polny, Ismael Rodriguez

Project Summary

1. API call
2. ETL
3. Store on MongoDB
4. Render Flask
5. HTML script
6. Visualizations through JavaScript
7. Add user-driven interactions through HTML
8. Populate dashboard

Dataset Used



- API
- Link: https://coinmarketcap.com/api/
  - Contains data for 1000 different cryptocurrencies
- Main Columns Used:
  - Name
  - Price
  - Volume Change: 24h
  - Market Cap Dominance
  - Percent Change: 1h, 24h, 7d, 30d, 60d, 90d

Analysis Focus

- Given the size of the dataset, we chose to focus most graphs on the top 10 cryptocurrencies
- Top 10 ranked off of market cap dominance

1. Bitcoin
2. Ethereum
3. Tether
4. BNB
5. USD Coin
6. XRP
7. Solana
8. Terra
9. Cardano
10. Avalanche

Extraction

```
%load_ext nb_black
from requests import Request, Session
from requests.exceptions import ConnectionError, Timeout, TooManyRedirects
import json
import pandas as pd
import pprint
from config import app_key
import pymongo
from time import time, sleep

url = "https://pro-api.coinmarketcap.com/v1/cryptocurrency/listings/latest"
parameters = {"start": 1, "limit": 1000, "convert": "USD"}
headers = {
    "Accept": "application/json",
    "X-CMC_PRO_API_KEY": app_key,
}
```

```
session = Session()
session.headers.update(headers)
data_retrieved = []
pp = pprint.PrettyPrinter(width=10, compact=True)

try:
    response = session.get(url, params=parameters)
    data = json.loads(response.text)
    pp.pprint(data)
    data_retrieved.append(data)

except (ConnectionError, Timeout, TooManyRedirects) as error:
    data_retrieved(error)
```

Transformation

```
docs = pd.DataFrame(columns=[])
items = db.items.find()
for num, doc in enumerate(items):
    doc["_id"] = str(doc["_id"])
    doc_id = doc["_id"]
    series_obj = pd.Series(doc, name=doc_id)
    docs = docs.append(series_obj)
```

```
final_df = pd.concat([data_df, df], axis=1)
final_df = final_df.drop(columns="quote")
final_df
```

```
data_df = pd.DataFrame(docs["data"][0])
data_df["date_added"] = pd.to_datetime(data_df["date_added"])
data_df["year"] = pd.DatetimeIndex(data_df["date_added"]).year
data_df.head(2)
```

Load

- Stored on MongoDB

```
conn = "mongodb://localhost:27017"
client = pymongo.MongoClient(conn)
db = client.cryptoCurrency_db
collection = db.items
```

Flask

```python
from flask import Flask, render_template, redirect
from flask_pymongo import PyMongo
import json
from bson import ObjectId
import flask
from flask_cors import CORS, cross_origin

class JSONEncoder(json.JSONEncoder):
    def default(self, o):
        if isinstance(o, ObjectId):
            return str(o)
        return json.JSONEncoder.default(self, o)

app = Flask(__name__)
CORS(app, support_credentials=True)
```

```python
app.config["MONGO_URI"] = "mongodb://localhost:27017/cryptoCurrency_db"
mongo = PyMongo(app)

# create my own API from the data stored in Mongo database
@app.route('/get_data')
def index():
    listings = mongo.db.items.find_one()
    return JSONEncoder().encode(listings)

@app.route('/')
def home():
    return render_template('index.html')
```

HTML - Head

- Used bootstrap, plotly and d3
- Worked on style to make the dashboard prettier

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Cryptocurrency Dashboard</title>
    <link rel="stylesheet" href="/static/style.css">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
    <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
    <script src="https://d3js.org/d3.v5.min.js"></script>

    <style>
    body {
    background-color: #ff8811;
    };

    h1 {
        color: #FFF8F0;
    }

    h5 {
        color: #FFF8F0;
    }
    select option {

        background-color: #FFF8F0;

    }
    .shadow {

box-shadow: 8px 7px 5px 0px rgba(244,208,111,0.54);
    }

    </style>
```

HTML - Body

```
<div class="container">
    <div class="row">
        <div class="col-md-12 text-center my-4">
            <h1></h1>
        </div>
```

- Created a drop down menu, added options manually, bar graph changes according to cryptocurrency chosen

```
<select class="shadow" id="selDataset">
    <option value="Bitcoin"> Bitcoin </option>
    <option value="Ethereum"> Ethereum </option>
    <option value="Tether"> Tether </option>
    <option value="BNB"> BNB </option>
    <option value="USD Coin"> USD Coin </option>
    <option value="Solana"> Solana </option>
    <option value="XRP"> XRP </option>
    <option value="Terra"> Terra </option>
    <option value="Cardano"> Cardano </option>
    <option value="Dogecoin"> Dogecoin </option>
</select>
```

HTML - Body (cont.)

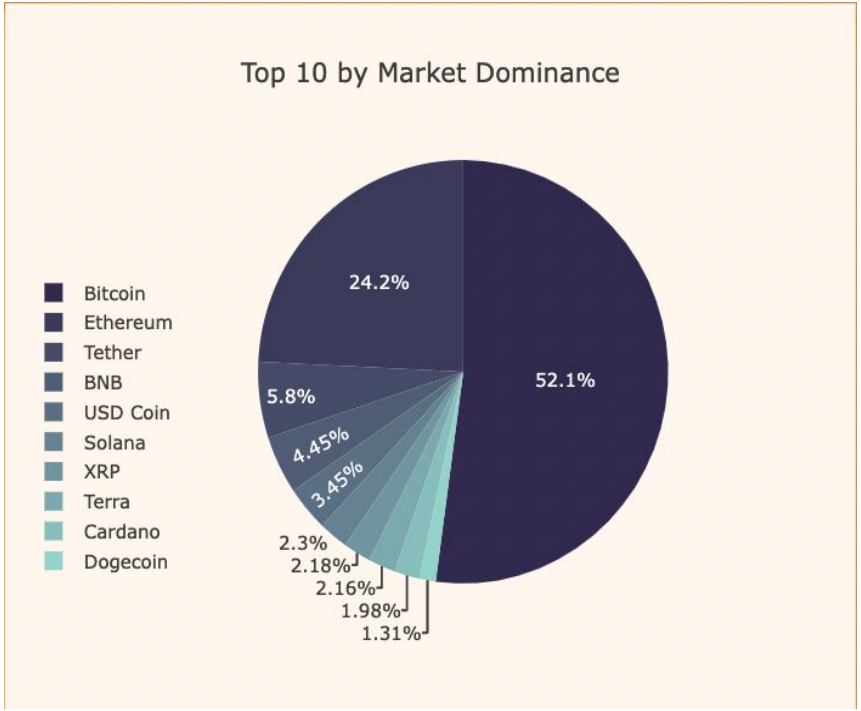
```
<div class="col-md-6">
    <div class = "shadow" id="BarChart2"></div>
</div>
<div class="col-md-6">
    <br>
    <br>
    <h1 style="color: #FFF8F0; font-size:75px; text-align: center;">Cryptocurrency Dashboard</h1>
</div>
```

- Created divs for each graph

```
<div class="row">
    <div class="col-md-6">
        <div class = "shadow", id="PieChart"></div>
    </div>
    <div class="col-md-6">
        <div class = "shadow", id="BarChart"></div>
    </div>
```

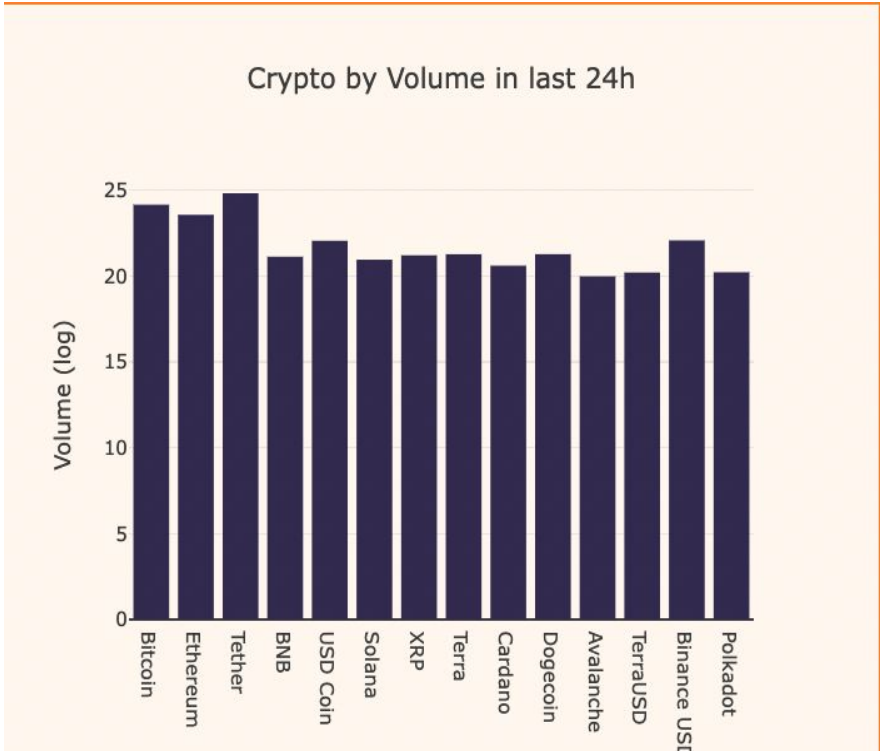
Pie Chart

- Top 10 Cryptos
- Market Cap Dominance (% of total MC) at time of API call



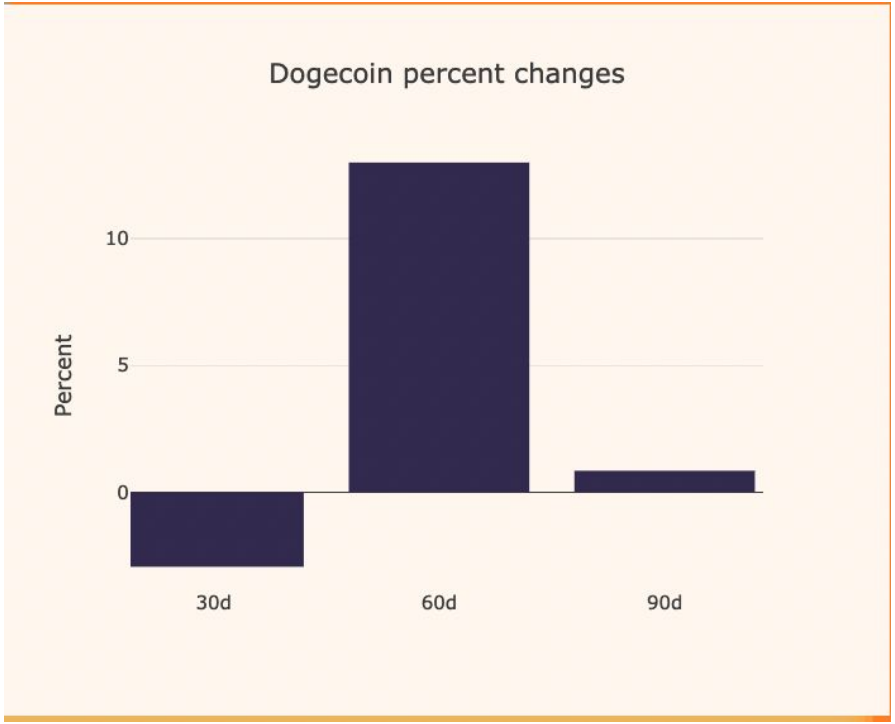
Bar Chart

- Top 10 Cryptos
- 24-Hour Volume Change at time of API call



2nd Bar Chart

- Percent price changes over a certain time period for a specified crypto
- 1hr, 24hr, 7d, 30d, 60d, 90d intervals



Limitations

- Only able to to get the current data at the time of the API call
  - Limited our time trend visualization capabilities
- Bitcoin data heavily skewed many columns, so it had to be separated from the rest of the cryptocurrencies
- Mac security troubleshooting
- Different operating system within group

Finalized Dashboard Showcase

Thank You