

# Web based app for DeFi tokens

## GitHub Project Description:

A Flask-based web app for visualizing historical market data of DeFi tokens using CoinGecko API, with interactive Plotly.js charts.

## Project Summary:

I developed a Flask web application that allows users to view and analyze historical market data (price, market cap, volume) for DeFi tokens. The data is fetched from the CoinGecko API and presented in an interactive chart using Plotly.js.

## Development Process:

- 1. Data Fetching:** Implemented a Python function to fetch historical data from the CoinGecko API.
- 2. Data Processing:** Used Pandas to process and prepare data for visualization.
- 3. Backend Setup:** Created a Flask server to handle web requests and serve data.
- 4. Frontend Creation:** Developed an HTML form to input user selections and display results.
- 5. Charting:** Integrated Plotly.js for dynamic and interactive data visualization.
- 6. Debugging:** Ensured correct data fetching and representation through debugging and error handling.
- 7. Enhancements:** Improved the script for better data normalization and visualization clarity.
- 8. Deployment:** Set up the application to run on a local server, ready to be deployed on any web hosting platform supporting Python.

## User Manual:

### 1. Starting the Application:

- Ensure Python and Flask are installed.
- Run the `app.py` script in your terminal or command prompt to start the server.

### 2. Accessing the Application:

- Open a web browser and navigate to `http://127.0.0.1:5001/`.

### 3. Using the Application:

- Enter the token ID (e.g., 'bitcoin', 'ethereum', 'dai') into the 'Token ID' field.
- Enter the desired number of days for historical data into the 'Days' field.
- Click the 'Get Data' button to submit your request.
- The historical market data chart for the selected token will be displayed on the page.

### 4. Interacting with the Chart:

- Hover over the chart to see detailed data points.
- Use the toolbar to zoom in/out, pan, or reset the chart view.

### 5. Adding More Tokens:

- To add more tokens to the application, update the `token\_ids` list in the `app.py` script with new CoinGecko token IDs.
- Restart the Flask server for the changes to take effect.

## Conclusion:

This project showcases a full-stack development process, from data fetching to front-end presentation. The application serves as a powerful tool for crypto enthusiasts and investors to make informed decisions based on historical data trends.