

Final Python Project Presentation

Currency Exchanger Dashboard

Dallas Baptist University – 6341-AIP01 – Python

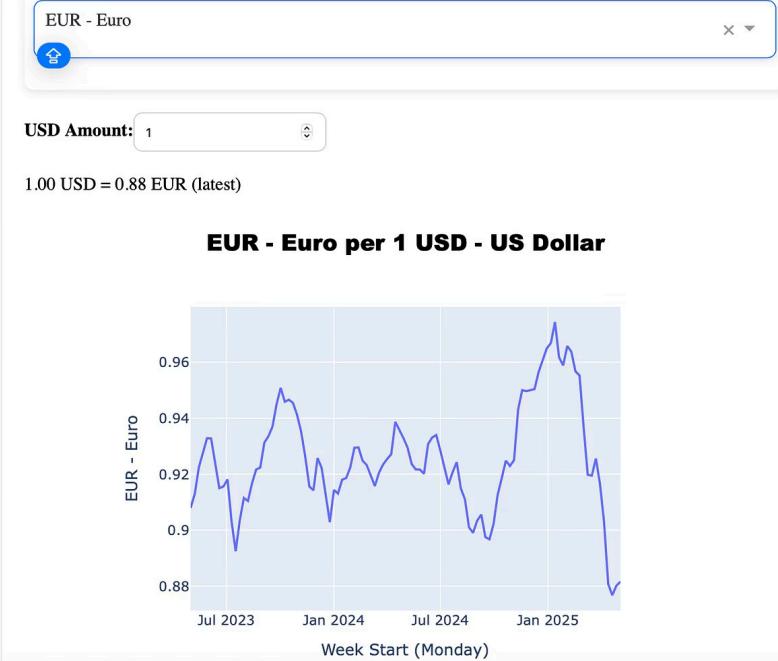
Instructor: Dennis Wang

Phat Vo, Noel Erkie, Laura Castano, Miryan Danila Urrea Gil, Enrique
Castaneira

 May 5, 2025

Currency Exchange Rates vs USD

Current Date: 2025-05-05





Introduction



Web-based Dashboard

A web-based dashboard to visualize and analyze USD exchange rates.



Target Users

Built for travelers, international students, expats, and global users.



Key Features

View historical exchange rate trends.

Convert USD to other currencies.

See annual % change.

Visualize volatility.

Data Source and API

API Used

Frankfurter API (European Central Bank).

Reasons for Choosing:

- Free & no authentication.
- Reliable historical data.
- Simple JSON structure.

Limitations

Limitations:

- Only major currencies supported.

Currencies Used:

- KRW, AUD, CAD, PLN, MXN, EUR, INR, CNY, HKD, THB, SGD.

```
    nolbet Cafe)          net settaller.01);
    22                      hat inspecter for Lager.gag(-425);
    74                      regal = querhaen(19.0.49)
    44                      want contactelf{
    15                      patlonty feresgntins the sl:
    18                      from rtafl,f gunneent:01pagersior,68
    18                      nextil):
    18                      );
    18                      );
    18
```

Tech Stack & Packages

Language

Python

Data Handling

requests – API access.

pandas, datetime, os – data handling & caching.

Visualization & Web

Dash, Plotly Express – graphs & interactivity.

Dash DataTable – interactive data tables.

 Focused on simplicity + functionality.

Currency Selection

Selection Criteria

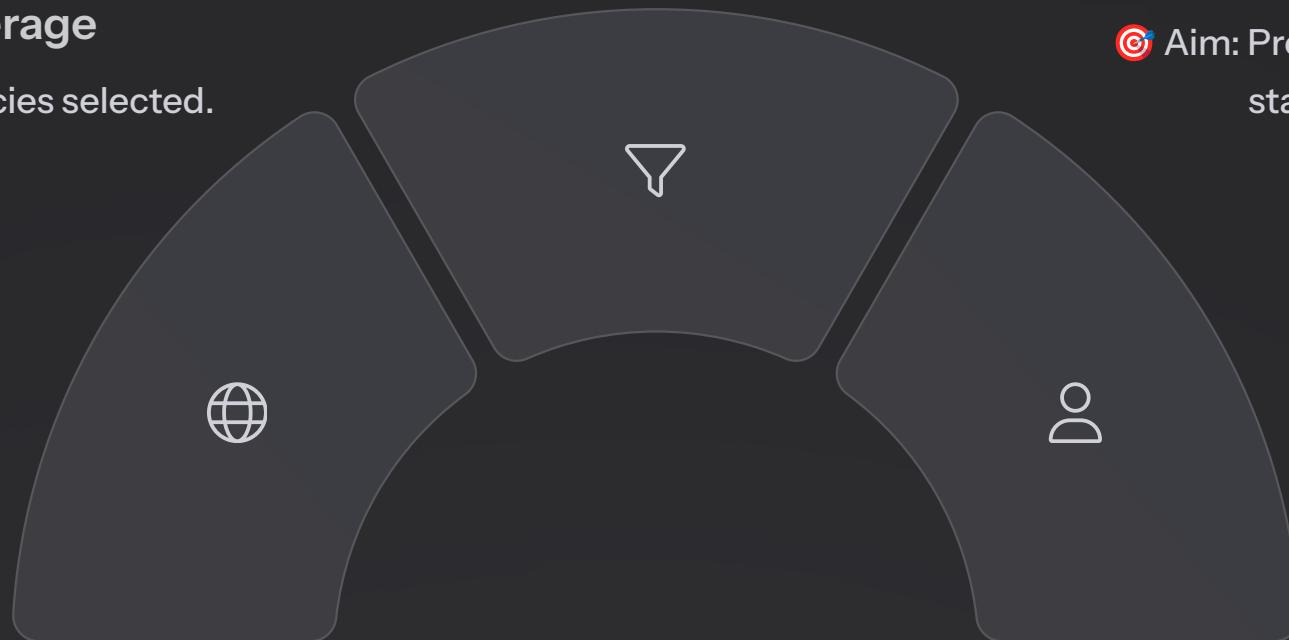


Criteria:

- Geographical diversity.
- Real-world relevance (travel, business, education).

Global Coverage

11 major global currencies selected.



User Experience

🧠 User-friendly: Full currency names added to dropdown.

🎯 Aim: Prevent UI overload while staying practical.

Euro rates:

10%

20%

25%

30%

1021

1022

2015

2021

2019

2004

2020

1010

Recent transactions

1/23

\\$17

Dashboard Features & Design

Dashboard Features

- ▼ Currency Dropdown
- \$ USD Input & Real-time Conversion
- 📈 Line Chart – Historical trends (2 years)
- 📊 Bar Chart – % change over 1 year
- 📋 Data Table – Live exchange rates
- 📉 Volatility Line Chart – Weekly std deviation

Styling & Design

- ✍️ Fonts: Bold, readable (Arial Black)
- 🎨 Colors: Light theme, soft shadows
- 📱 Responsive layout: Fits wide/narrow screens
- 💡 UX touches: Hover effects, rounded corners, clean spacing
- 💻 Goal: Make it functional *and* visually appealing

Deployment

Platform Selection

🚀 Hosted using **Render**

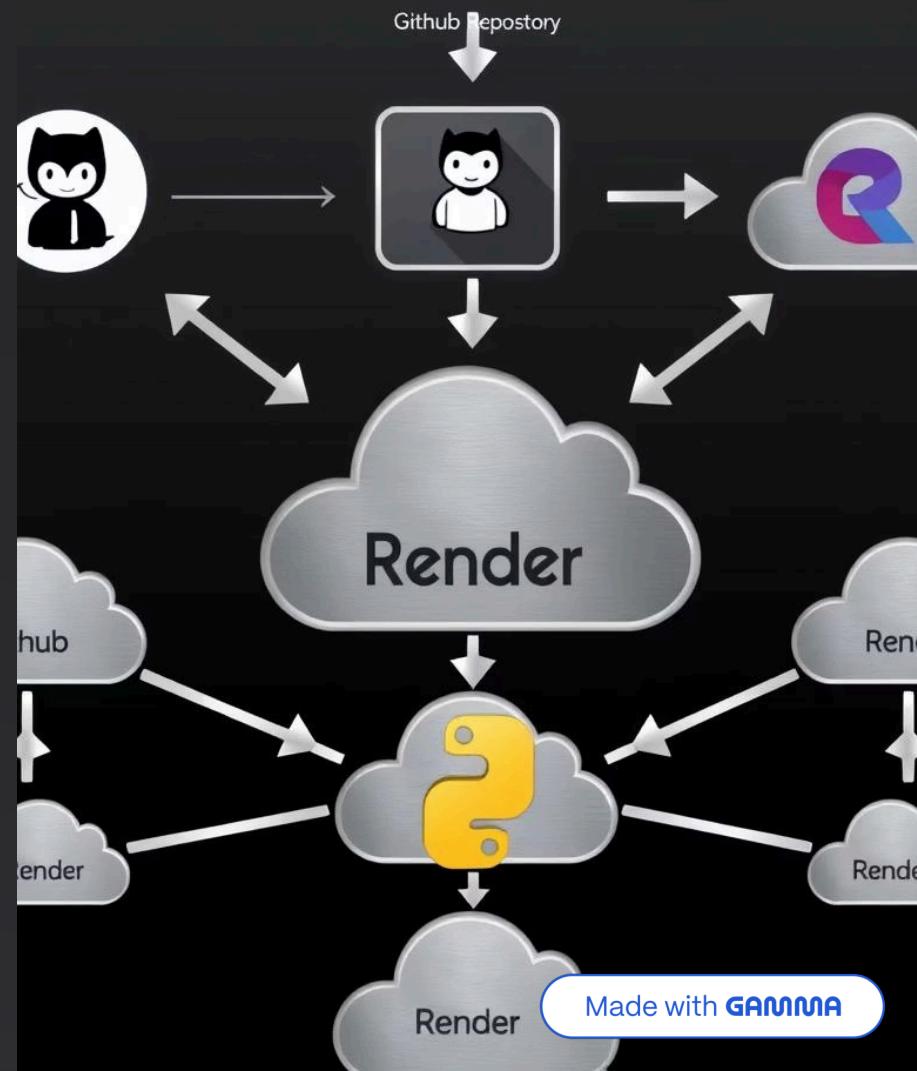
Deployment Process

Steps:

- GitHub → Render
- Requirements.txt specified
- Python 3.9+ environment

Maintenance

🔄 Easy updates and redeployment



Challenges & Alternatives



Challenges

Limited currency list, missing data, balancing simplicity vs. interactivity



Lessons Learned

API integration, data visualization, web deployment, UX design importance



Alternatives Explored

API and framework options we considered



API Alternatives:

exchangerate.host

open.er-api.com

currencyapi.net



Framework Alternatives:

Streamlit – easier UI, but limited customization

Flask – flexible, but more code-heavy

✓ Why Dash?

Best mix of interactivity + simplicity

Easy integration with Plotly for visuals

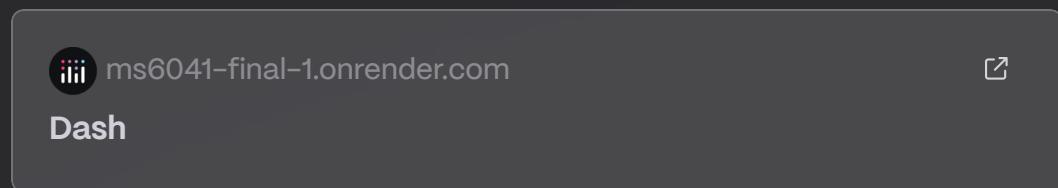
Minimal frontend skills required

Link & Code Snapshot



Dashboard Web

The final application as deployed on Render:



```
 5  from dash import Dash, dcc, html, Input, Output, dash_table
 6  import plotly.express as px
 7
 8
 9  # Define time range
10 end_date = date.today()
11 start_date = end_date - timedelta(days=365 * 2)
12 today = date.today().strftime('%Y-%m-%d')
13
14 # Base currency and symbols
15 base = 'USD'
16 symbols = 'KRW,AUD,CAD,PLN,MXN,EUR,INR,CNY,HKD,THB,SGD'
17
18 # Currency full names
19 currency_names = {
20     'KRW': 'KRW - South Korean Won',
21     'AUD': 'AUD - Australian Dollar',
22     'CAD': 'CAD - Canadian Dollar',
23     'PLN': 'PLN - Polish Zloty',
24     'MXN': 'MXN - Mexican Peso',
25     'EUR': 'EUR - Euro',
26     'INR': 'INR - Indian Rupee',
27     'CNY': 'CNY - Chinese Yuan',
28     'HKD': 'HKD - Hong Kong Dollar',
29     'THB': 'THB - Thai Baht',
30     'SGD': 'SGD - Singapore Dollar',
31     'USD': 'USD - US Dollar'
32 }
```

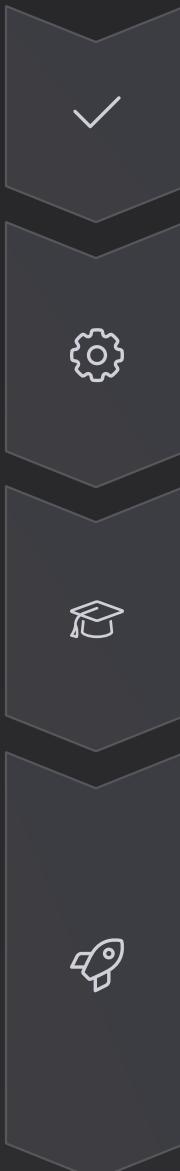
Code

currency_exchange_final_project.py

[Github Click here](#)



Conclusion



Accomplishments

- ✓ Created a live, interactive dashboard from scratch.

Integration

- 🔧 Integrated real-time data + backend logic + frontend visuals.

Learning

- 🎓 Learned full-stack Python (Dash + API + Data + Deployment).

Future Improvements

- 🚀 Future Improvements:
- Add more currencies.
- Use mobile-friendly design.
- Offer multi-language support.