import streamlit as st

import yfinance as yf

import pandas as pd

import pandas\_ta as ta

import plotly.graph\_objects as go

from datetime import datetime, timedelta

# Set page configuration

st.set\_page\_config(page\_title="Forex Chart Pattern Scanner", layout="wide")

# Fetch forex data from Yahoo Finance

def get\_forex\_data(symbol, period='1y', interval='1d'):

try:

ticker = yf.Ticker(f"{symbol}=X")

data = ticker.history(period=period, interval=interval)

if data.empty:

st.error(f"No data available for {symbol}")

return None

return data

except Exception as e:

st.error(f"Error fetching data: {str(e)}")

return None

# Detect chart patterns

def detect\_patterns(df):

if df is None or df.empty:

return {}

patterns = {}

# Double Top Pattern

if len(df) > 20:

highs = df['High'].rolling(window=5).max()

if any(highs.diff().abs() < 0.0010):

patterns['Double Top'] = 'Bearish'

# Double Bottom Pattern

if len(df) > 20:

lows = df['Low'].rolling(window=5).min()

if any(lows.diff().abs() < 0.0010):

patterns['Double Bottom'] = 'Bullish'

# Head and Shoulders Pattern (Simple Detection)

if len(df) > 40:

if df['High'].rolling(window=10).max().diff().abs().mean() < 0.0015:

patterns['Head and Shoulders'] = 'Bearish'

return patterns

# Main function to display everything in Streamlit

def main():

st.title("Forex Chart Pattern Scanner")

# Sidebar for user input

currency\_pairs = ['EUR/USD', 'GBP/USD', 'USD/JPY', 'AUD/USD']

selected\_pair = st.sidebar.selectbox("Select Currency Pair", currency\_pairs)

timeframe = st.sidebar.selectbox("Select Timeframe",

['1d', '5d', '1mo', '3mo', '6mo', '1y'])

# Get data from Yahoo Finance

symbol = selected\_pair.replace('/', '')

df = get\_forex\_data(symbol, period=timeframe)

if df is not None and not df.empty:

# Detect patterns

patterns = detect\_patterns(df)

# Display detected patterns

st.subheader("Detected Patterns")

if patterns:

for pattern, direction in patterns.items():

st.write(f"📊 {pattern}: {direction}")

else:

st.write("No significant patterns detected in the current timeframe")

# Create candlestick chart

fig = go.Figure(data=[go.Candlestick(x=df.index,

open=df['Open'],

high=df['High'],

low=df['Low'],

close=df['Close'])])

fig.update\_layout(title=f"{selected\_pair} Chart",

xaxis\_title="Date",

yaxis\_title="Price",

height=600)

st.plotly\_chart(fig, use\_container\_width=True)

# Run the Streamlit app

if \_\_name\_\_ == "\_\_main\_\_":

main()