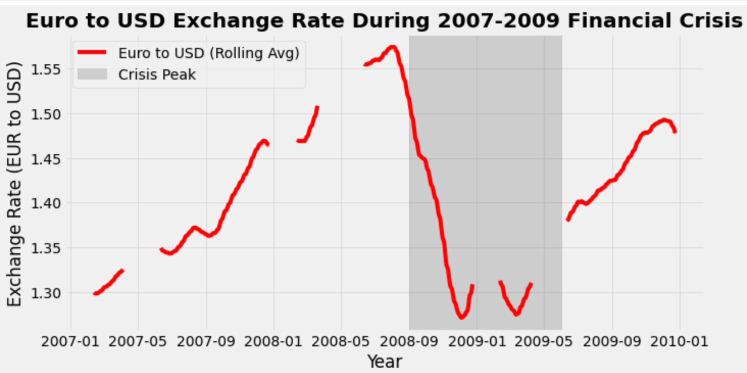
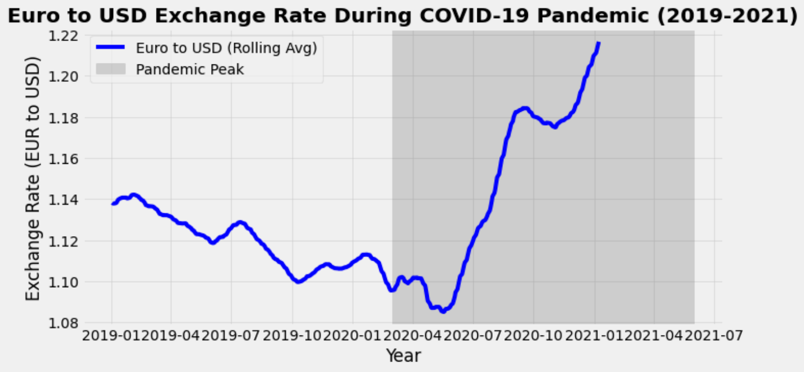
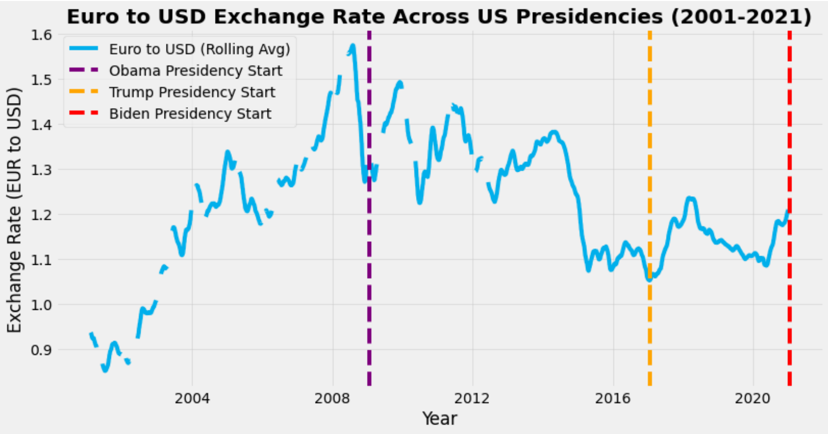
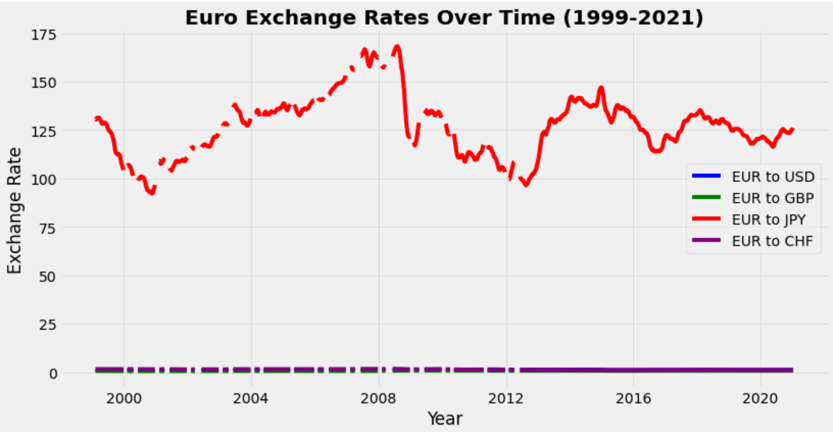
In [12]:

```python
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.style as style
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.arima.model import ARIMA
from pmdarima import auto_arima

import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
warnings.filterwarnings("ignore", message="Covariance matrix calculated using the outer product of gradients")



# Load dataset
file_path = "euro-daily-hist_1999_2020.csv"
exchange_rates = pd.read_csv(file_path)

# Data Cleaning and Preprocessing
exchange_rates.rename(columns={
    '[US dollar ]': 'US_dollar',
    'Period\\Unit:': 'Time',
    '[UK pound sterling ]': 'GBP',
    '[Japanese yen ]': 'JPY',
    '[Swiss franc ]': 'CHF'
}, inplace=True)

exchange_rates['Time'] = pd.to_datetime(exchange_rates['Time'])
exchange_rates.sort_values('Time', inplace=True)
exchange_rates.reset_index(drop=True, inplace=True)

# Select relevant columns
currencies = ['Time', 'US_dollar', 'GBP', 'JPY', 'CHF']
euro_to_other = exchange_rates[currencies].copy()

# Handle missing data (replace '-' with NaN and convert to float)
for col in ['US_dollar', 'GBP', 'JPY', 'CHF']:
    euro_to_other[col] = euro_to_other[col].replace('-', pd.NA)  # Replace '-' with NaN
    euro_to_other[col] = euro_to_other[col].str.strip()  # Remove leading/trailing spaces (if any)
    euro_to_other[col] = pd.to_numeric(euro_to_other[col], errors='coerce')  # Convert to float, forcing errors to NaN


# Calculate rolling mean (30-day moving average)
for col in ['US_dollar', 'GBP', 'JPY', 'CHF']:
    euro_to_other[f"{col}_rolling"] = euro_to_other[col].rolling(30).mean()

# Define periods of interest
financial_crisis = euro_to_other[(euro_to_other['Time'].dt.year >= 2007) & (euro_to_other['Time'].dt.year <= 2009)]
covid_pandemic = euro_to_other[(euro_to_other['Time'].dt.year >= 2019) & (euro_to_other['Time'].dt.year <= 2021)]
us_presidents = euro_to_other[(euro_to_other['Time'].dt.year >= 2001) & (euro_to_other['Time'].dt.year < 2021)]

# Apply FiveThirtyEight style
style.use('fivethirtyeight')

# Plot Financial Crisis (2007-2009)
plt.figure(figsize=(10, 5))
plt.plot(financial_crisis['Time'], financial_crisis['US_dollar_rolling'], color='red', label='Euro to USD (Rolling Avg)')
plt.axvspan(pd.Timestamp('2008-09-01'), pd.Timestamp('2009-06-01'), color='gray', alpha=0.3, label='Crisis Peak')
plt.title("Euro to USD Exchange Rate During 2007-2009 Financial Crisis", weight='bold')
plt.xlabel("Year")
plt.ylabel("Exchange Rate (EUR to USD)")
plt.legend()
plt.grid(alpha=0.5)
plt.show()

# Plot COVID-19 Pandemic (2019-2021)
plt.figure(figsize=(10, 5))
plt.plot(covid_pandemic['Time'], covid_pandemic['US_dollar_rolling'], color='blue', label='Euro to USD (Rolling Avg)')
plt.axvspan(pd.Timestamp('2020-03-01'), pd.Timestamp('2021-06-01'), color='gray', alpha=0.3, label='Pandemic Peak')
plt.title("Euro to USD Exchange Rate During COVID-19 Pandemic (2019-2021)", weight='bold')
plt.xlabel("Year")
plt.ylabel("Exchange Rate (EUR to USD)")
plt.legend()
plt.grid(alpha=0.5)
plt.show()

# Plot Exchange Rate Under US Presidents
plt.figure(figsize=(12, 6))
plt.plot(us_presidents['Time'], us_presidents['US_dollar_rolling'], color='#00B2EE', label="Euro to USD (Rolling Avg)")
plt.axvline(pd.Timestamp('2009-01-20'), color='purple', linestyle='--', label="Obama Presidency Start")
plt.axvline(pd.Timestamp('2017-01-20'), color='orange', linestyle='--', label="Trump Presidency Start")
plt.axvline(pd.Timestamp('2021-01-20'), color='red', linestyle='--', label="Biden Presidency Start")
plt.title("Euro to USD Exchange Rate Across US Presidencies (2001-2021)", weight='bold')
plt.xlabel("Year")
plt.ylabel("Exchange Rate (EUR to USD)")
plt.legend()
plt.grid(alpha=0.5)
plt.show()

# Multi-Currency Comparison (EUR to USD, GBP, JPY, CHF)
plt.figure(figsize=(12, 6))
plt.plot(euro_to_other['Time'], euro_to_other['US_dollar_rolling'], color='blue', label="EUR to USD")
plt.plot(euro_to_other['Time'], euro_to_other['GBP_rolling'], color='green', label="EUR to GBP")
plt.plot(euro_to_other['Time'], euro_to_other['JPY_rolling'], color='red', label="EUR to JPY")
plt.plot(euro_to_other['Time'], euro_to_other['CHF_rolling'], color='purple', label="EUR to CHF")
plt.title("Euro Exchange Rates Over Time (1999-2021)", weight='bold')
plt.xlabel("Year")
plt.ylabel("Exchange Rate")
plt.legend()
plt.grid(alpha=0.5)
plt.show()
```



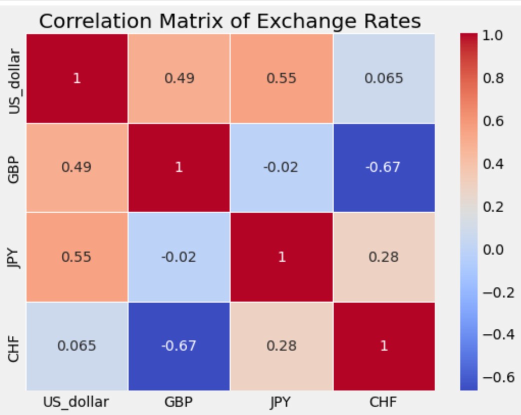




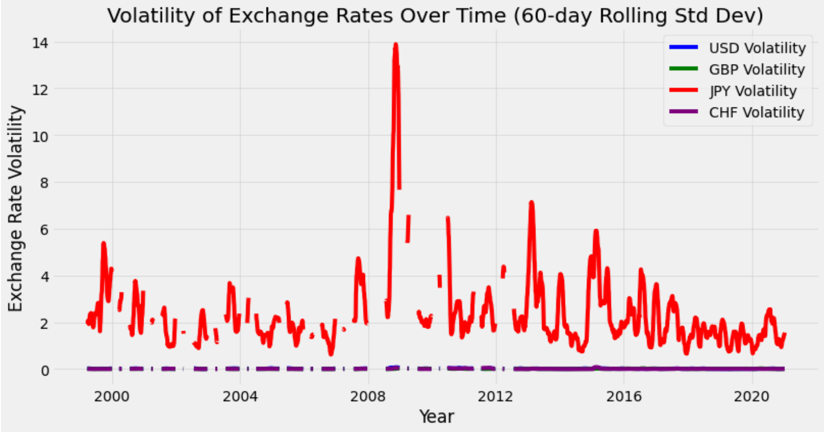


In [16]:

```
# Correlation Analysis
correlation_matrix = euro_to_other[['US_dollar', 'GBP', 'JPY', 'CHF']].corr()
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title("Correlation Matrix of Exchange Rates")
plt.show()

print(correlation_matrix)


# Volatility Analysis (Rolling Standard Deviation)
volatility_window = 60  # 60-day rolling window
for col in ['US_dollar', 'GBP', 'JPY', 'CHF']:
    euro_to_other[f"{col}_volatility"] = euro_to_other[col].rolling(volatility_window).std()

# Plot Volatility Trends
plt.figure(figsize=(12, 6))
plt.plot(euro_to_other['Time'], euro_to_other['US_dollar_volatility'], label="USD Volatility", color='blue')
plt.plot(euro_to_other['Time'], euro_to_other['GBP_volatility'], label="GBP Volatility", color='green')
plt.plot(euro_to_other['Time'], euro_to_other['JPY_volatility'], label="JPY Volatility", color='red')
plt.plot(euro_to_other['Time'], euro_to_other['CHF_volatility'], label="CHF Volatility", color='purple')
plt.title("Volatility of Exchange Rates Over Time (60-day Rolling Std Dev)")
plt.xlabel("Year")
plt.ylabel("Exchange Rate Volatility")
plt.legend()
plt.grid(alpha=0.5)
plt.show()
```




```
           US_dollar       GBP       JPY       CHF
US_dollar   1.000000  0.493000  0.547829  0.065139
GBP         0.493000  1.000000 -0.019830 -0.667594
JPY         0.547829 -0.019830  1.000000  0.281510
CHF         0.065139 -0.667594  0.281510  1.000000
```



Key Takeaways:

1. **EUR to USD (US_dollar) vs. Other Currencies:**
   - **Strong correlation with JPY (0.55)** → Suggests that movements in the EUR/USD exchange rate are somewhat aligned with EUR/JPY.
   - **Moderate correlation with GBP (0.49)** → Indicates that EUR/USD and EUR/GBP tend to move together but are not perfectly aligned.
   - **Weak correlation with CHF (0.06)** → Suggests EUR/USD and EUR/CHF move mostly independently.
2. **EUR to GBP (GBP) vs. Other Currencies:**
   - **Negative correlation with CHF (-0.67)** → Indicates that when the EUR/GBP exchange rate increases, EUR/CHF tends to decrease (strong inverse relationship).
   - **Minimal correlation with JPY (-0.02)** → Suggests little relationship between EUR/GBP and EUR/JPY movements.
3. **EUR to JPY (JPY) vs. Other Currencies:**
   - **Positive correlation with USD (0.55)** → Suggests that when EUR/USD rises, EUR/JPY also tends to rise.
   - **Moderate correlation with CHF (0.28)** → Indicates some alignment, but not a strong relationship.
4. **EUR to CHF (CHF) vs. Other Currencies:**
   - **Strong negative correlation with GBP (-0.67)** → Suggests that GBP and CHF exchange rates move in opposite directions against the Euro.

Implications:

- **Portfolio Diversification**: Since EUR/USD and EUR/JPY are **positively correlated**, they might not offer much diversification. However, **EUR/GBP and EUR/CHF move inversely**, meaning they could act as hedging instruments.
- **Market Sensitivity**: The **strong negative correlation between EUR/GBP and EUR/CHF** could reflect different monetary policies of the **Bank of England (BoE) and Swiss National Bank (SNB)**.
- **Currency Pair Trading Strategies**: Traders could use these correlations to anticipate **hedging opportunities** or construct **mean-reverting strategies**.