Import Required Libraries

Import the necessary libraries, including NumPy, pandas, and matplotlib.

In [39]:

```python
# Import the necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Load EEG Data

Load the EEG data from the CSV file using pandas.

In [40]:

```python
# Load the EEG data from the CSV file using pandas
file_path = '../data/mindMonitor_2025-02-15--19-18-29.csv'
eeg_data = pd.read_csv(file_path)

# Display the first few rows of the data
eeg_data.head()
```

Out[40]:

	TimeStamp	Delta_TP9	Delta_AF7	Delta_AF8	Delta_TP10	Theta_TP9	Theta_AF7	Theta
0	2025-02-15 19:18:29.097	NaN	NaN	NaN	NaN	NaN	NaN	
1	2025-02-15 19:18:29.257	NaN	NaN	NaN	NaN	NaN	NaN	
2	2025-02-15 19:18:29.261	NaN	NaN	NaN	NaN	NaN	NaN	
3	2025-02-15 19:18:29.597	0.0	0.285171	0.135338	0.0	0.0	0.011946	0.1
4	2025-02-15 19:18:30.098	0.0	0.247423	0.417576	0.0	0.0	-0.103462	0.1

5 rows × 39 columns

Preprocess Data

Clean and preprocess the EEG data, including handling missing values and selecting relevant columns.

In [41]:

```
# Convert 'TimeStamp' to datetime
eeg_data['TimeStamp'] = pd.to_datetime(eeg_data['TimeStamp'])

# Fill missing values
eeg_data.fillna(method='ffill', inplace=True)
print("\nEEG Data After Preprocessing:")
print(eeg_data.head())

# Select relevant columns for Alpha waves
alpha_columns = ['Alpha_AF7', 'Alpha_AF8']

# Extract Alpha wave data
alpha_data = eeg_data[alpha_columns]
print("\nAlpha Wave Data:")
print(alpha_data.head())

# Convert the data to numeric type
alpha_data_numeric = alpha_data.apply(pd.to_numeric, errors='coerce')
print("\nAlpha Wave Data (Numeric):")
print(alpha_data_numeric.head())

# Check for any remaining NaN values and handle them
print("\nChecking for NaN values:")
print(alpha_data_numeric.isna().sum())

# Fill any remaining NaN values with the mean of the column
alpha_data_numeric.fillna(alpha_data_numeric.mean(), inplace=True)
print("\nAlpha Wave Data (Numeric) After Filling NaNs:")
print(alpha_data_numeric.head())
```

```
EEG Data After Preprocessing:
                TimeStamp  Delta_TP9  Delta_AF7  Delta_AF8  Delta_TP10  \
0 2025-02-15 19:18:29.097        NaN        NaN        NaN         NaN   
1 2025-02-15 19:18:29.257        NaN        NaN        NaN         NaN   
2 2025-02-15 19:18:29.261        NaN        NaN        NaN         NaN   
3 2025-02-15 19:18:29.597        0.0   0.285171   0.135338         0.0   
4 2025-02-15 19:18:30.098        0.0   0.247423   0.417576         0.0   

   Theta_TP9  Theta_AF7  Theta_AF8  Theta_TP10  Alpha_TP9  ...    Gyro_X  \
0        NaN        NaN        NaN         NaN        NaN  ...       NaN   
1        NaN        NaN        NaN         NaN        NaN  ...       NaN   
2        NaN        NaN        NaN         NaN        NaN  ...       NaN   
3        0.0   0.011946   0.148115         0.0        0.0  ...  0.575714   
4        0.0  -0.103462   0.144801         0.0        0.0  ...  0.328979   

     Gyro_Y    Gyro_Z  HeadBandOn  HSI_TP9  HSI_AF7  HSI_AF8  HSI_TP10  \
0       NaN       NaN         NaN      NaN      NaN      NaN       NaN   
1       NaN       NaN         NaN      NaN      NaN      NaN       NaN   
2       NaN       NaN         NaN      NaN      NaN      NaN       NaN   
3 -2.183228 -0.194397         1.0      2.0      1.0      1.0       4.0   
4 -2.272949 -0.216827         1.0      4.0      1.0      1.0       4.0   

   Battery                         Elements  
0      NaN  /muse/event/connected Muse-0167  
1      NaN             /muse/elements/blink  
2      NaN             /muse/elements/blink  
3     50.0             /muse/elements/blink  
4     50.0             /muse/elements/blink  

[5 rows x 39 columns]

Alpha Wave Data:
   Alpha_AF7  Alpha_AF8
0        NaN        NaN
1        NaN        NaN
2        NaN        NaN
3   0.167010    0.60120
4   0.137827    0.59584

Alpha Wave Data (Numeric):
   Alpha_AF7  Alpha_AF8
0        NaN        NaN
1        NaN        NaN
2        NaN        NaN
3   0.167010    0.60120
4   0.137827    0.59584

Checking for NaN values:
Alpha_AF7    3
Alpha_AF8    3
dtype: int64

Alpha Wave Data (Numeric) After Filling NaNs:
   Alpha_AF7  Alpha_AF8
0   0.403290   0.404602
1   0.403290   0.404602
```

```
2   0.403290   0.404602
3   0.167010   0.601200
4   0.137827   0.595840
```

```
/tmp/ipykernel_3263454/1920047010.py:5: FutureWarning: DataFrame.fillna with 'metho
d' is deprecated and will raise in a future version. Use obj.ffill() or obj.bfill()
instead.
  eeg_data.fillna(method='ffill', inplace=True)
```

Perform FAA

To calculate Frontal Alpha Asymmetry (FAA), we use the formula FAA = log(Alpha_AF8) - log(Alpha_AF7). This involves taking the natural logarithm of the Alpha_AF8 and Alpha_AF7 columns and then subtracting the log values of Alpha_AF7 from Alpha_AF8. The resulting FAA values are then plotted over time to visualize the changes.

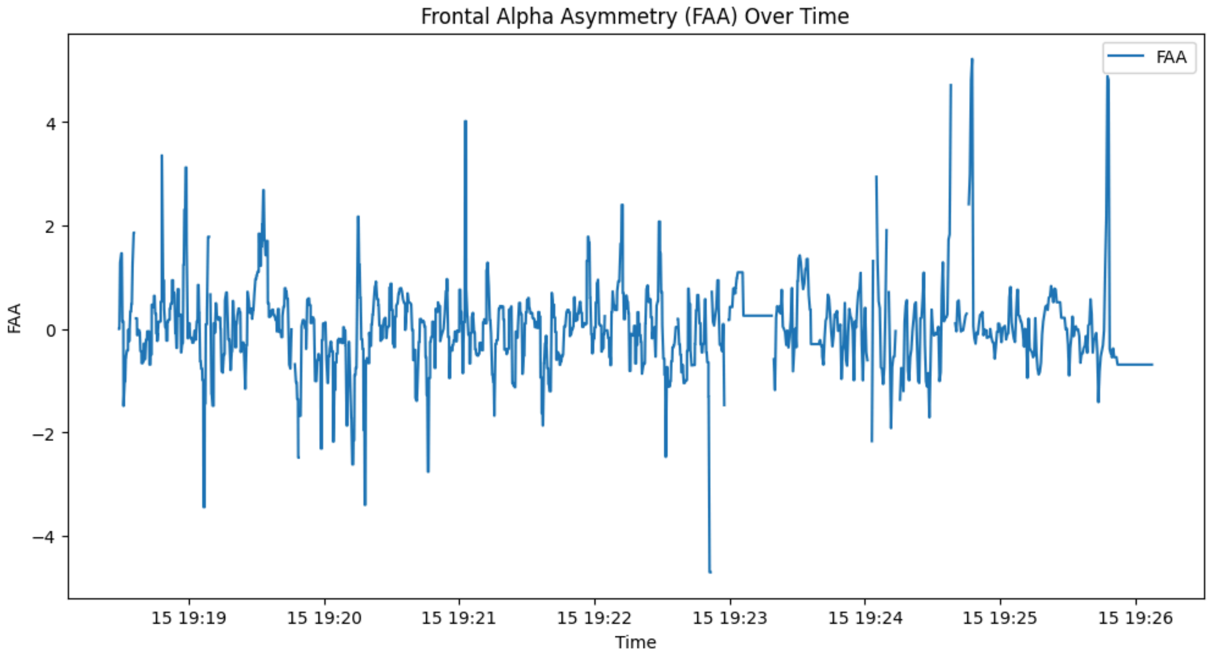
In [42]:

```
# Calculate Frontal Alpha Asymmetry (FAA)
# FAA = log(Alpha_AF8) - log(Alpha_AF7)
alpha_data_numeric['FAA'] = np.log(alpha_data_numeric['Alpha_AF8']) - np.log(alpha_
print("\nFAA Data:")
print(alpha_data_numeric['FAA'].head())

# Plot FAA over time
plt.figure(figsize=(12, 6))
plt.plot(eeg_data['TimeStamp'], alpha_data_numeric['FAA'], label='FAA')
plt.title('Frontal Alpha Asymmetry (FAA) Over Time')
plt.xlabel('Time')
plt.ylabel('FAA')
plt.legend()
plt.show()
```

```
FAA Data:
0    0.003249
1    0.003249
2    0.003249
3    1.280872
4    1.463972
Name: FAA, dtype: float64
```

```
/mnt/nvme/workspace/playground/eeg/venv/lib/python3.12/site-packages/pandas/core/arr
aylike.py:399: RuntimeWarning: invalid value encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```


Frontal Alpha Asymmetry (FAA) Over Time
FAA
FAA
4
2
0
−2
−4
15 19:19
15 19:20
15 19:21
15 19:22
15 19:23
15 19:24
15 19:25
15 19:26
Time