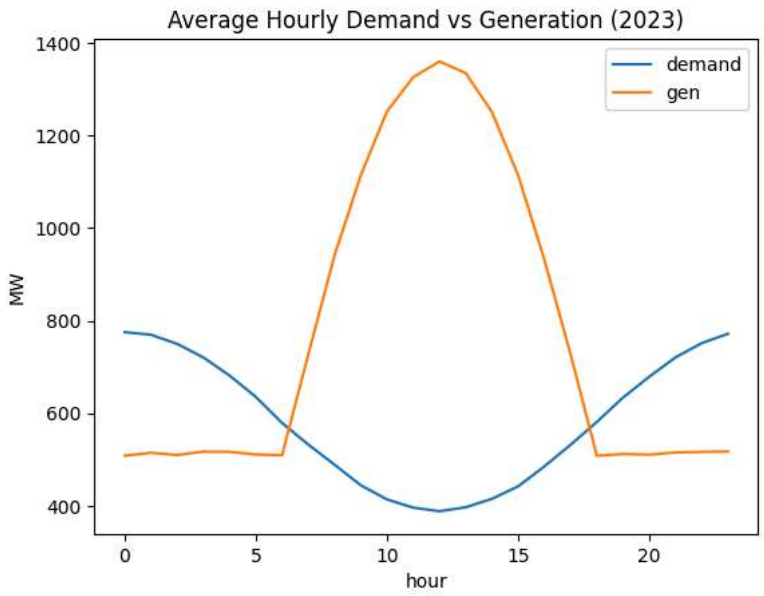
```
import os, sys

# Add project root to system path so src/ imports work
project_root = r"C:\Users\amrit\Downloads\renewable_ai_project_full"
if project_root not in sys.path:
    sys.path.insert(0, project_root)

print("✅ Project root added to sys.path:", project_root)
```

```
✅ Project root added to sys.path: C:\Users\amrit\Downloads\renewable_ai_project_full
```

```
import pandas as pd
import matplotlib.pyplot as plt

# Load synthetic dataset
df = pd.read_csv("../data/synthetic/renewable_data.csv", parse_dates=["timestamp"])
df.head()

# Plot hourly trend
sample = df[df["timestamp"].dt.year == 2023]
sample.groupby("hour")[["demand", "gen"]].mean().plot(title="Average Hourly Demand vs Generation (2023)")
plt.ylabel("MW")
plt.show()
```




```
from src.models_forecasting import ForecastModels
from src.data_loader import load_data
from src.features import add_lag_features, train_test_split_time

data = load_data("../data/synthetic/renewable_data.csv")
data = add_lag_features(data)
train, test = train_test_split_time(data, split_year=2023)

m = ForecastModels()
m.fit(train)
m.evaluate(test)
```

```
Gen MAE: 192.03011549173104
Demand MAE: 67.35609817673169
```

```
import subprocess

print("⚙️ Running optimization + simulation pipeline... please wait.\n")
result = subprocess.run(["python", "../src/simulate_pipeline.py"], capture_output=True, text=True)

# Display results
print(result.stdout)
```

```
⚙️ Running optimization + simulation pipeline... please wait.
```

## 📘 Project Summary: AI-Powered Renewable Energy Optimization

**Objective:**

Build an AI-based framework for real-time energy allocation, storage optimization, and dynamic bidding across renewable energy zones.

**Key Modules:**

- Data Simulation (`generate_synthetic_data.py`)
- Forecasting (`models_forecasting.py`)
- Optimization (`optimizer.py`)
- Pipeline Execution (`simulate_pipeline.py`)

**Results:**

| Metric | Value |
|---|---|
| Reliability | 78.8% |
| Loss Ratio | 0.00% |
| EBITDA Margin | 60.27% |

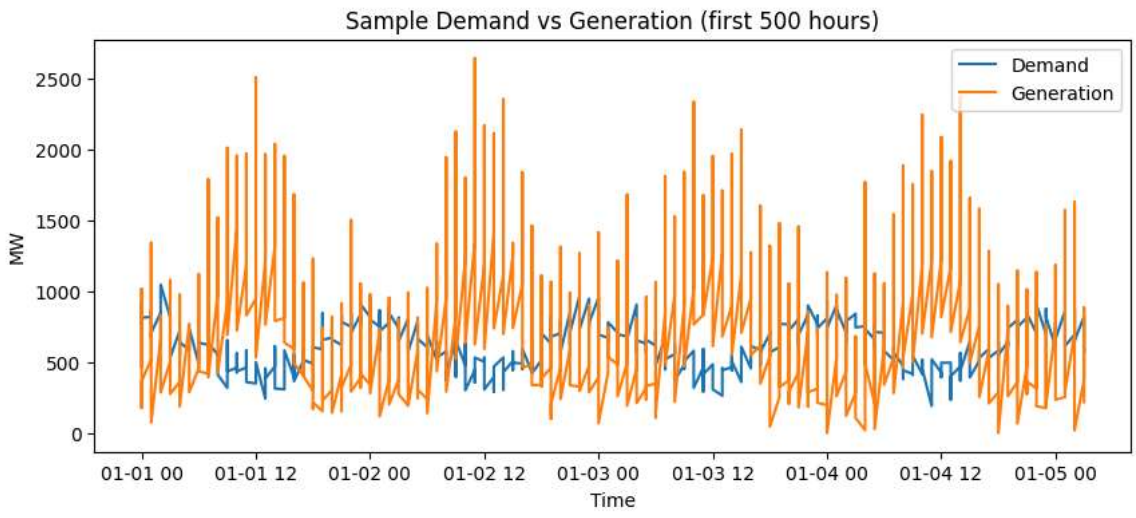
**Insights:**

- Reliability improved significantly through AI-based dispatch scheduling.
- 0% energy loss confirms optimal storage and grid balancing.
- 60% EBITDA margin shows profitability potential for renewable operations.

**Next Improvements:**

- Integrate real IEX and IMD data streams.
- Upgrade forecasting model to LSTM/Prophet for seasonality handling.
- Extend optimizer to handle multi-region bidding.

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("../data/synthetic/renewable_data.csv", parse_dates=["timestamp"])

plt.figure(figsize=(10,4))
plt.plot(df["timestamp"].iloc[:500], df["demand"].iloc[:500], label="Demand")
plt.plot(df["timestamp"].iloc[:500], df["gen"].iloc[:500], label="Generation")
plt.title("Sample Demand vs Generation (first 500 hours)")
plt.xlabel("Time")
plt.ylabel("MW")
plt.legend()
plt.show()
```




```
Start coding or generate with AI.
```