SOFTWARE DEVOLEPMENT PROCESS

Programming Project: Utilization of Open Data

BY:AMALRAJ MULLIKKOL

Task Description:

The goal of this project is to develop a Python program that uses open data to analyze the economic development of the United Arab Emirates (UAE). The project

focuses on **GDP per capita** as the selected economic indicator and examines how it changed over the period **2010–2023**.

The program loads the dataset, processes it, calculates basic statistics, computes economic growth, and generates a visual chart showing how GDP per capita changed over time.

Objectives:

* Use real **open data** from a public, verified source.
* Load and process the data using Python.
* Filter the data by country (UAE) and year range (2010–2023).
* Compute summary statistics (mean, min, max).
* Calculate growth from the first to the last year.
* Create a visualization (line chart) of GDP per capita.
* Apply programming best practices and project structure.
* Publish the code in a **GitHub repository**.

Chosen Open Data Source:

* **Dataset Name:** GDP per capita (current US$)
* **Provider:** World Bank Open Data
* **Link:** https://data.worldbank.org/indicator/NY.GDP.PCAP.CD?locations=AE

Dataset Description:

This dataset includes the annual GDP per capita for the United Arab Emirates from 2010 to 2023.
Values represent the average income per person in current US dollars. The dataset was manually extracted from the World Bank website and formatted into a CSV file containing three columns:

country, year, gdp_per_capita

2. Project Architecture and Structure **Programming**

Language Justification:

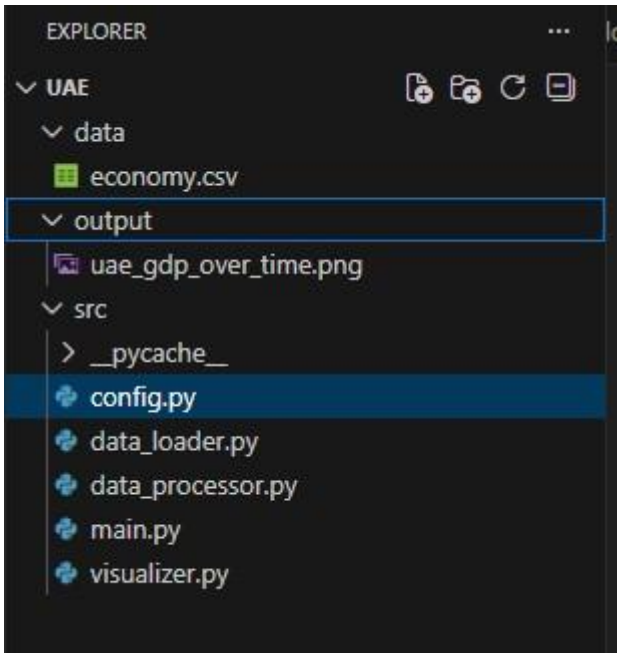
Python was chosen due to its strong ecosystem for data analysis. Libraries such as **pandas** and **matplotlib** make it easy to load, clean, process, and visualize open data.

Python is beginner-friendly, follows the **PEP8** style standard, and is widely used in data science and academic research.
Its readability and large community support make it the ideal choice for this project.

Program Structure:

The project follows a modular structure to keep the code clean and organized.

Project Folder Structure:



Module Descriptions:

***config.py**
Stores global configuration settings, including file paths, selected country, year range, and output folder location.

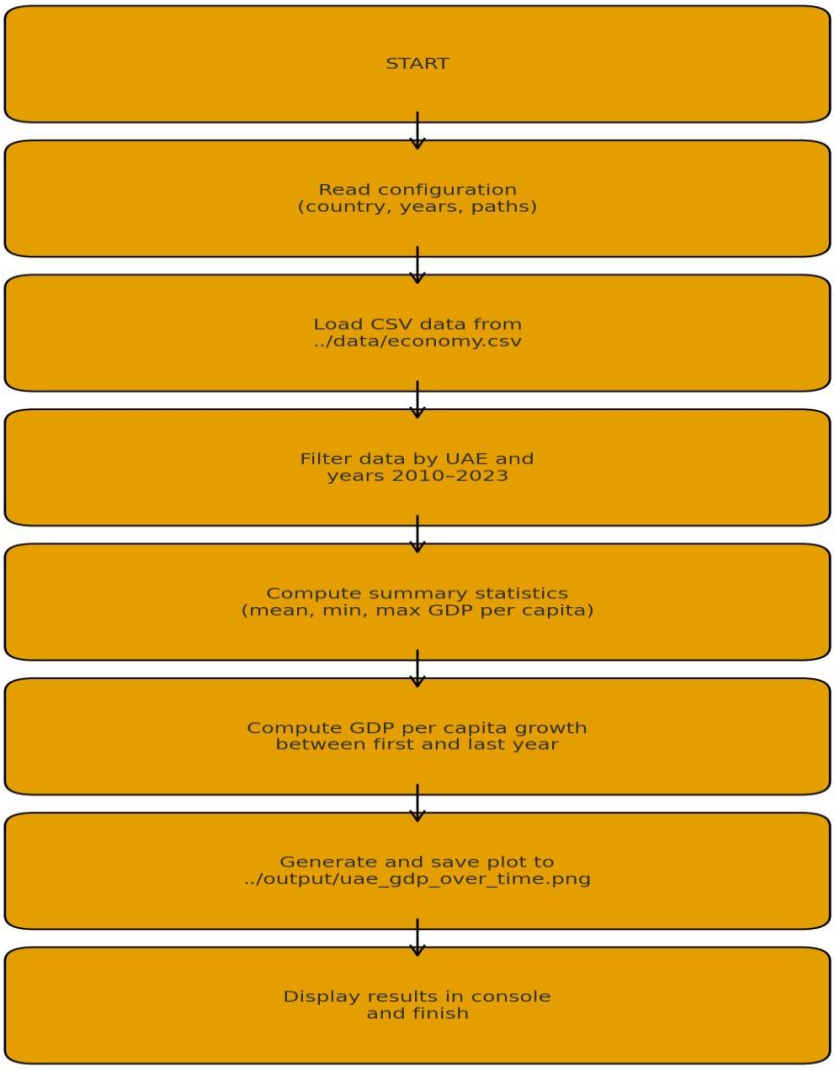
* **data_loader.py**
Loads the CSV dataset using pandas and ensures proper data types.

* **data_processor.py**
Filters the dataset by country and year, calculates summary statistics, and computes GDP per capita growth.

* **visualizer.py**
Generates a line chart of GDP per capita over time and saves it as a PNG image.

* **main.py**
The entry point of the application. Coordinates loading, processing, analyzing, and visualizing data.

Algorithm / Flowchart Description:



3. Result Examples:

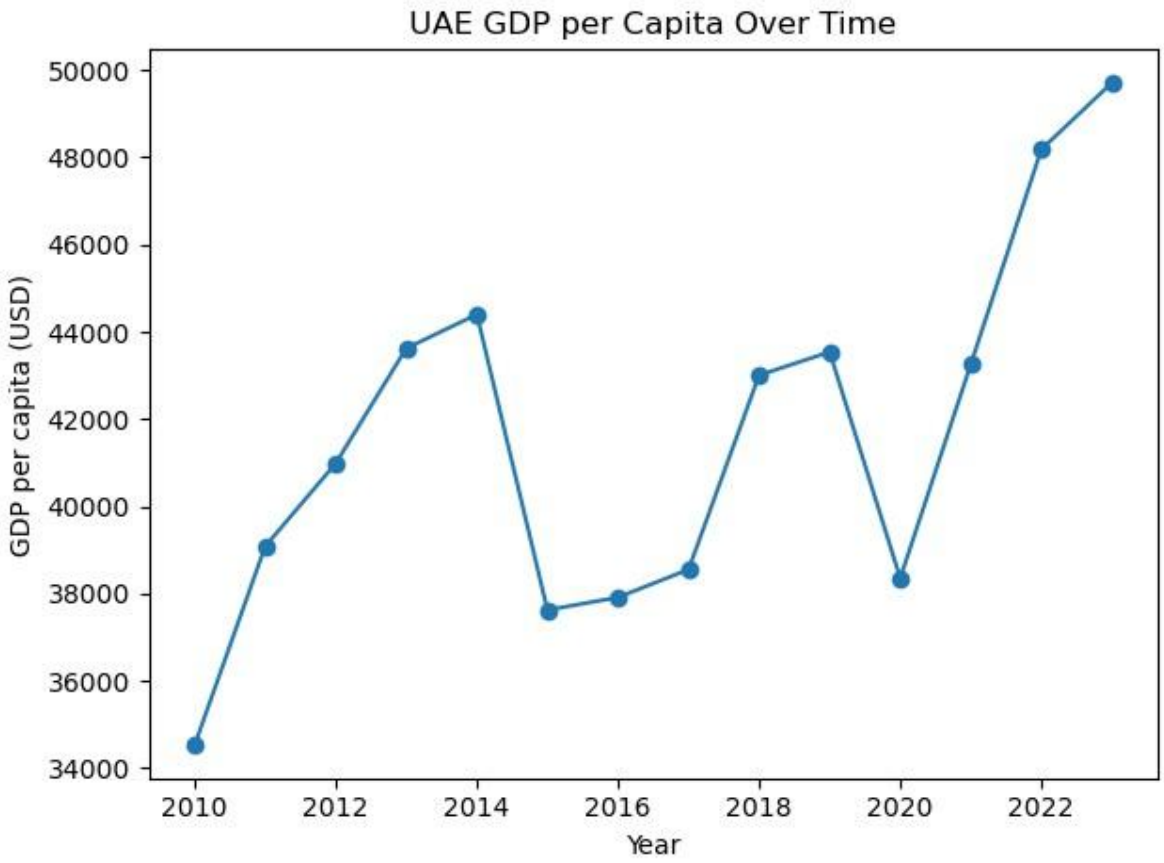
filtered Data (UAE GDP per capita 2010–2023)

```
Filtered data:
   country  year  gdp_per_capita
0      UAE  2010        34527.37
1      UAE  2011        39080.88
2      UAE  2012        40984.78
3      UAE  2013        43622.57
4      UAE  2014        44398.83
5      UAE  2015        37622.96
6      UAE  2016        37911.81
7      UAE  2017        38551.31
8      UAE  2018        43005.73
9      UAE  2019        43538.35
10     UAE  2020        38353.05
11     UAE  2021        43255.24
12     UAE  2022        48182.47
13     UAE  2023        49711.33
```

Summary Statistics:

```
Summary statistics (GDP per capita):
country  mean_gdp   min_gdp   max_gdp
    UAE 41,624.76 34,527.37 49,711.33
```

Generated Visualization:



Interpretation:

The chart shows a generally upward trend in UAE GDP per capita between 2010 and 2023, with some fluctuations around 2015–2020.
The GDP per capita increased significantly after 2021, reaching its highest value in 2023.
Overall, the UAE experienced **approximately 44% growth** in GDP per capita over the analyzed period, indicating strong economic development.

Program Code :

Appendix A — config.py:

```
from pathlib import Path


DATA_FILE = Path("..") / "data" / "economy.csv"


COUNTRIES = ["UAE"]


START_YEAR = 2010
```

```
END_YEAR = 2023

OUTPUT_DIR = Path("..") / "output"
```

Appendix B — data_loader.py:

```
from pathlib import Path
from typing import Union
import pandas as pd
from config import DATA_FILE

def load_data(path: Union[str, Path] = None) -> pd.DataFrame:
    csv_path = Path(path) if path is not None else DATA_FILE

    if not csv_path.exists():
        raise FileNotFoundError(f"Data file not found: {csv_path}")

    df = pd.read_csv(csv_path)
    df["year"] = df["year"].astype(int)
    df["gdp_per_capita"] = df["gdp_per_capita"].astype(float)
    df["country"] = df["country"].astype(str)
    return df
```

Appendix C — data_processor.py:

```
import pandas as pd
from typing import List

def filter_data(df, countries: List[str], start_year, end_year):
```

```
    mask = (
        df["country"].isin(countries)
        & (df["year"] >= start_year)
        & (df["year"] <= end_year)
    )
    filtered = df.loc[mask].copy()
    filtered.sort_values(["year"], inplace=True)
    return filtered

def compute_summary_statistics(df: pd.DataFrame) -> pd.DataFrame:
    grouped = df.groupby("country")["gdp_per_capita"]
    summary = grouped.agg(mean_gdp="mean", min_gdp="min", max_gdp="max")
    return summary.reset_index()

def compute_growth(df: pd.DataFrame) -> pd.DataFrame:
    results = []

    for country, group in df.groupby("country"):
        group = group.sort_values("year")
        first_val = group.iloc[0]["gdp_per_capita"]
        last_val = group.iloc[-1]["gdp_per_capita"]
        growth_percent = (last_val - first_val) / first_val * 100

        results.append({
            "country": country,
            "first_year": group.iloc[0]["year"],
            "last_year": group.iloc[-1]["year"],
```

```
            "first_value": first_val,
            "last_value": last_val,
            "growth_percent": growth_percent,
        })

    return pd.DataFrame(results)
```

Appendix D — visualizer.py

```
from pathlib import Path
import matplotlib.pyplot as plt
import pandas as pd

def ensure_output_dir(path: Path):
    path.mkdir(parents=True, exist_ok=True)

def plot_gdp_over_time(df: pd.DataFrame, output_dir: Path):
    ensure_output_dir(output_dir)

    plt.figure()
    plt.plot(df["year"], df["gdp_per_capita"], marker="o")
    plt.xlabel("Year")
    plt.ylabel("GDP per capita (USD)")
    plt.title("UAE GDP per Capita Over Time")
    plt.tight_layout()

    file = output_dir / "uae_gdp_over_time.png"
    plt.savefig(file)
```

```
    plt.close()
    return file
```

Appendix E — main.py

```
from config import COUNTRIES, END_YEAR, OUTPUT_DIR, START_YEAR
from data_loader import load_data
from data_processor import compute_growth, compute_summary_statistics, filter_data
from visualizer import plot_gdp_over_time

def main():
    print("Loading data...")
    df = load_data()

    print("Filtering data...")
    filtered_df = filter_data(df, COUNTRIES, START_YEAR, END_YEAR)
    print(f"\nFiltered data:\n{filtered_df}")

    print("\nComputing summary statistics...")
    summary_df = compute_summary_statistics(filtered_df)
    print("\nSummary statistics (GDP per capita):")
    print(summary_df.to_string(index=False))

    print("\nComputing growth between first and last year...")
    growth_df = compute_growth(filtered_df)
    print("\nGDP per capita growth (%):")
    print(growth_df.to_string(index=False))
```

```
    print("\nCreating plot...")
    gdp_over_time_path = plot_gdp_over_time(filtered_df, OUTPUT_DIR)
    print(f"\nSaved line chart to: {gdp_over_time_path}")


    print("\nDone.")


if __name__ == "__main__":
    main()
```

Conclusion

This project successfully analyzed UAE GDP per capita from 2010 to 2023 using open data from the World Bank. The Python program processed the dataset, calculated key statistics, and generated a visualization showing how the UAE's economy changed over time. The results indicate steady long-term growth, with GDP per capita increasing by about **44%** over the period. Overall, the project demonstrates how open data and Python can be used effectively to extract meaningful economic insights.