K.RAMAKRISHNAN COLLEGE OF ENGINEERING

(AUTONOMOUS)

FUEL PRIZE MONITORING AND ANALYSIS TOOL

Presented by,

Name : B.BHARATH RAJ

Register No. : 8115U23EC009

Roll. No. : ECA2309

Year & Section : I ECA A

Date of Review : 20/06/2024

PRESENTATION OVERVIEW

- Objective
- System Architecture(Flow Diagram)
- Modules
- CodeTantra Implementation Screenshot
- Future Scope
- Conclusion
- Refernce

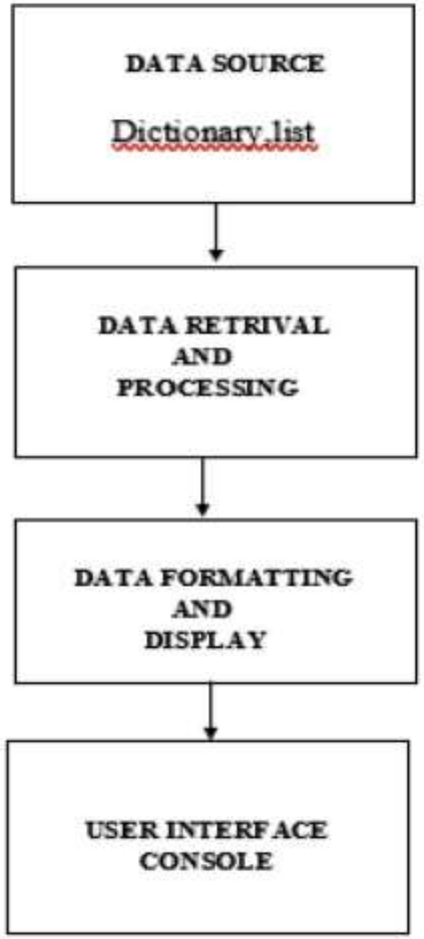
OBJECTIVE OF THE PROJECT

OBJECTIVE :

The objective of the Petrol Price Monitoring and Analysis Tool is to provide real-time monitoring and analysis of petrol prices, enabling users to track trends, make informed decisions, and understand factors influencing price fluctuations.

PROPOSED SYSTEM ARCHITECTURE/FLOW DIAGRAM

.



MODULES

1.USER INTERFACE: The User Interface (UI) module handles user interactions, taking input from the user and displaying the output.

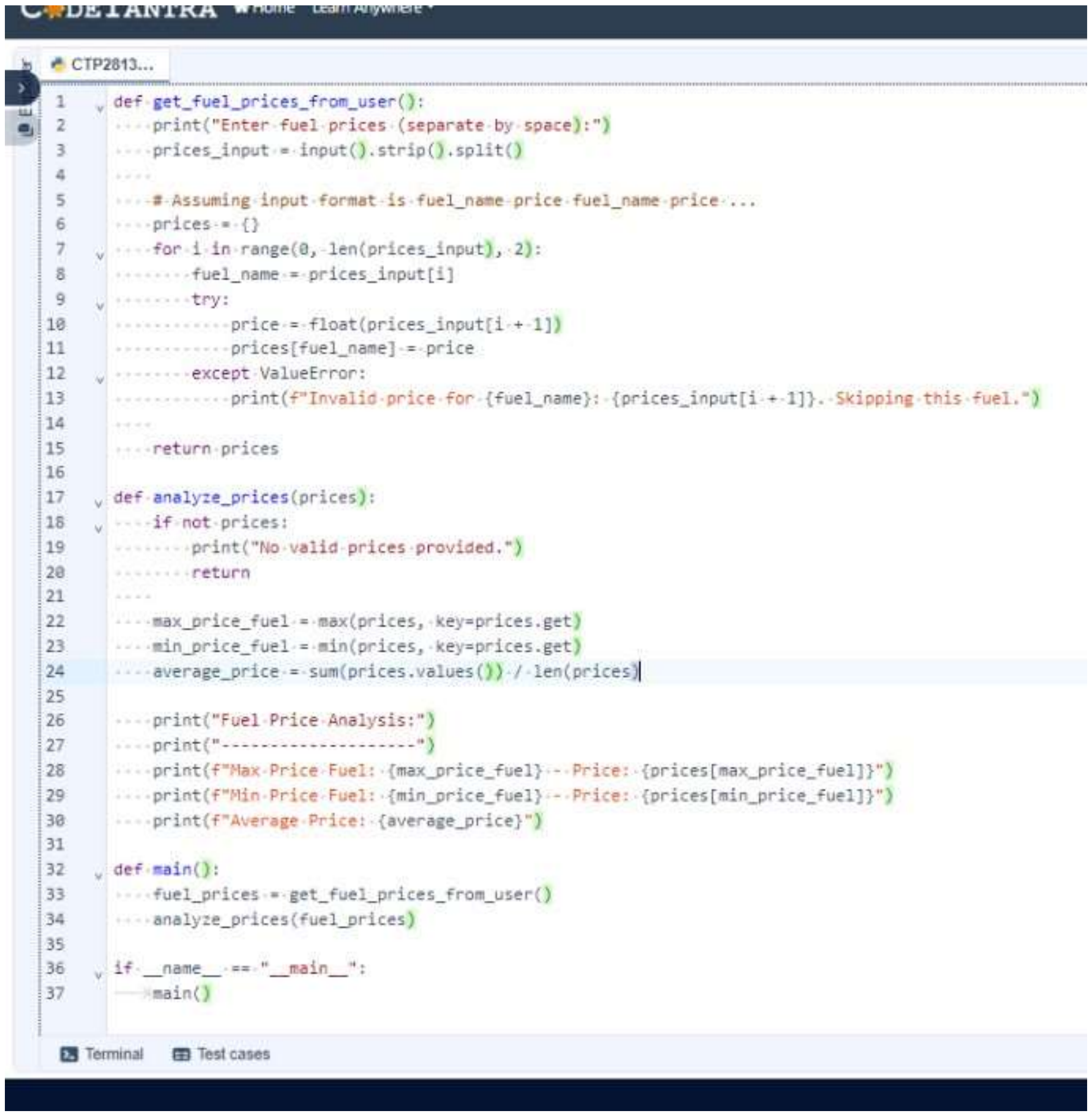
2.DATA COLLETION: This module will be responsible for collecting and updating the fuel prices.

3. DATA PROCESSING: This module will process the fetched data if needed. It can include functions to format or validate the data before displaying it.

4. DISPLAY THE RESULT: The Display the result module is responsible for formatting and displaying the today's and yesterday's fuel prices

CodeTantra Implementation Screenshot

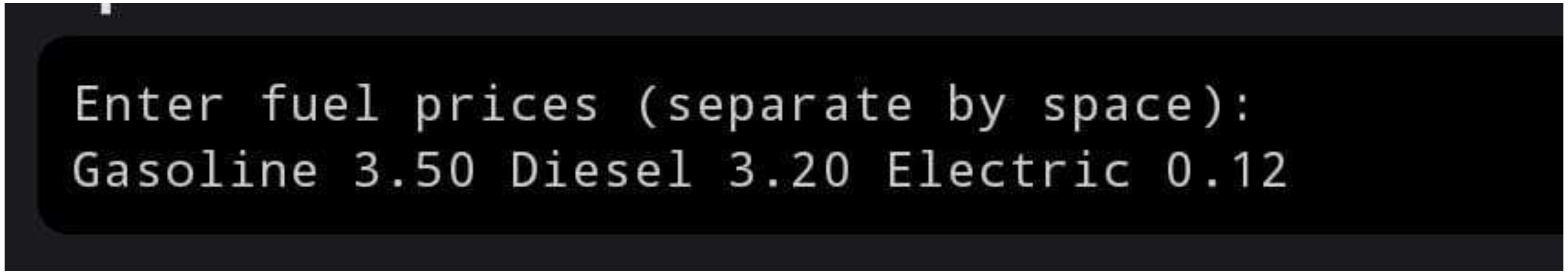
PROGRAM



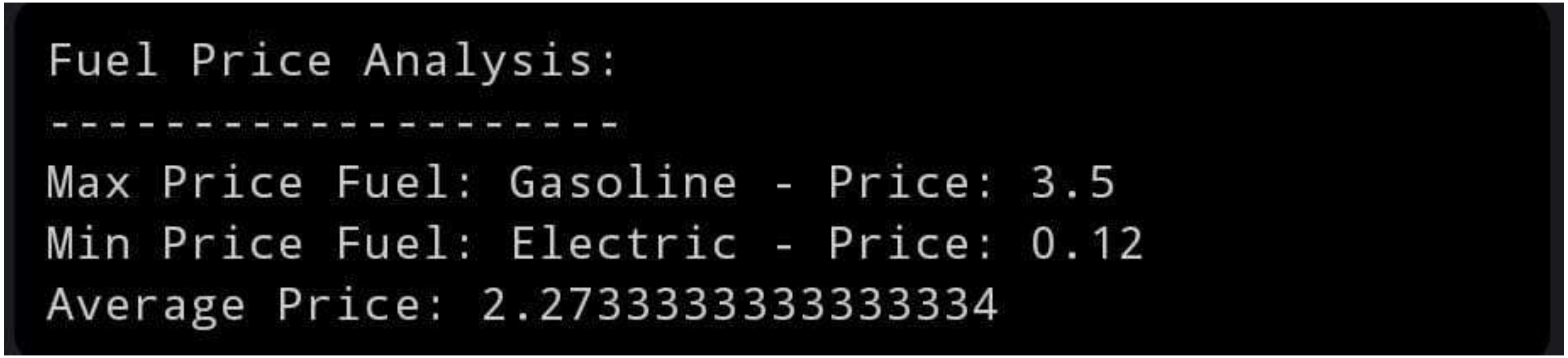
```python
def get_fuel_prices_from_user():
    print("Enter fuel prices (separate by space):")
    prices_input = input().strip().split()
    
    # Assuming input format is fuel_name price fuel_name price ...
    prices = {}
    for i in range(0, len(prices_input), 2):
        fuel_name = prices_input[i]
        try:
            price = float(prices_input[i + 1])
            prices[fuel_name] = price
        except ValueError:
            print(f"Invalid price for {fuel_name}: {prices_input[i + 1]}. Skipping this fuel.")
    
    return prices

def analyze_prices(prices):
    if not prices:
        print("No valid prices provided.")
        return
    
    max_price_fuel = max(prices, key=prices.get)
    min_price_fuel = min(prices, key=prices.get)
    average_price = sum(prices.values()) / len(prices)

    print("Fuel Price Analysis:")
    print("--------------------")
    print(f"Max Price Fuel: {max_price_fuel} - Price: {prices[max_price_fuel]}")
    print(f"Min Price Fuel: {min_price_fuel} - Price: {prices[min_price_fuel]}")
    print(f"Average Price: {average_price}")

def main():
    fuel_prices = get_fuel_prices_from_user()
    analyze_prices(fuel_prices)

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

CodeTantra Implementation Screenshot

OUTPUT



```
Enter fuel prices (separate by space):
Gasoline 3.50 Diesel 3.20 Electric 0.12
```



```
Fuel Price Analysis:
----------------------
Max Price Fuel: Gasoline - Price: 3.5
Min Price Fuel: Electric - Price: 0.12
Average Price: 2.2733333333333334
```

Future Scope:

- A fuel management system is designed to monitor, control, and optimize the use of fuel in various applications.
- It tracks fuel use, detects leaks or theft, and provides information to improve efficiency.

Conclusion:

- As the landscape evolves, Fuel Price Monitoring and Analysis Tools will play a crucial role in navigating the complexities of fuel economics, driving efficiency, and promoting sustainability in the energy sector.

Reference:

- Fluent Python: Clear, Concise, and Effective Programming by Luciano Ramalho
- www.geeksforgreeks.org

Thank
You