WEEK 3 PROJECT REPORT:

ENHANCEMENTS AND OPTIMIZATION

NAME: MINURANISAHU, ANANYA, RITHIKA

DATE: 03/02/2026

Objective:

The objective of week 3was to enhance and optimize the existing project by improving data visualization, structuring output effectively, and enhancing user interactions. This week focused on organizing nutritional information clearly, creating graphical representations, and improving the overall usability of the project.

Implementation Details:

1.Data structuring using pandas:

- Nutritional data of selected recipes was structured using a pandas data frame.
- This allowed tabular representation of calories, proteins, carbohydrates, fats, and fibre in a clear and readable format.

2.Visualization using matplotlib:

- Bar charts were generated to represent nutritional breakdown visually.
- This helped users quickly understand the composition of recipes and compare nutrient amounts effectively.

3.Output optimization:

- Printed the data in a formatted table using df.to_string (index=False) for better readability.
- Enhanced console messages to make the outputs more user-friendly.

4.Future enhancements considered:

- Implementing multithreading to handle multiple API requests efficiency.
- Further improving UI/UX for a smother user experience, including integrating interactive charts in the future.

Tools and technologies:

- Python
- Pandas (for data structuring)
- Matplotlib (for visualization)
- Visual studio code

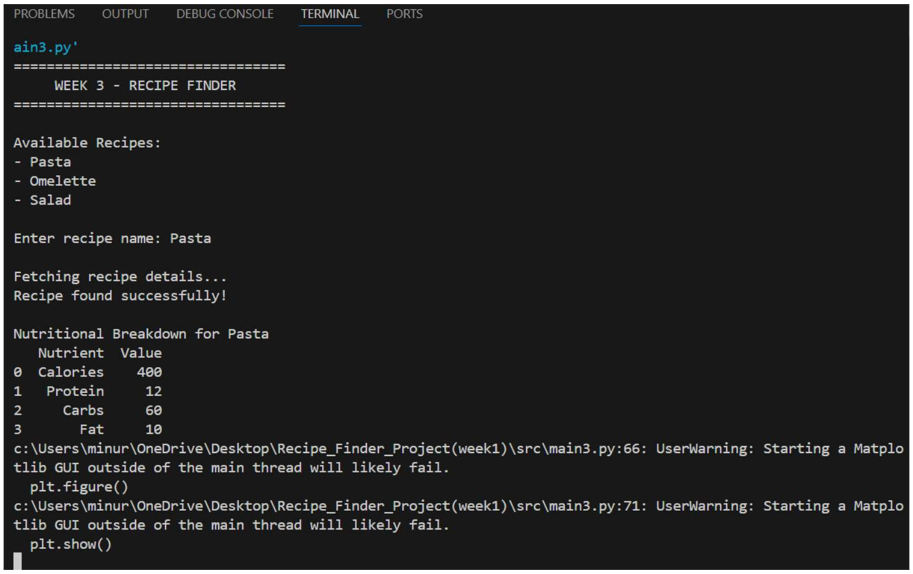
OUTCOME:

- The application now presents nutritional data in a structured and visually appealing format.
- Users can easily see the nutritional breakdown of selected recipe.
- The project is prepared for future enhancements like multithreading and API integration.

MULTITHREADING EXPLANATION:

Multithreading is a technique that allows a program to rum multiple threads (smaller processes) at the same.

In the context of this project, multithreading can be used to handle multiple API requests simultaneously in head of one by one.
This reduces the total execution time and makes the program faster and more efficient. “Multithreading was studied and conceptually integrated into the project design to support efficient handling of multiple API requests. This prepares the application for improved performance in future extensions,”



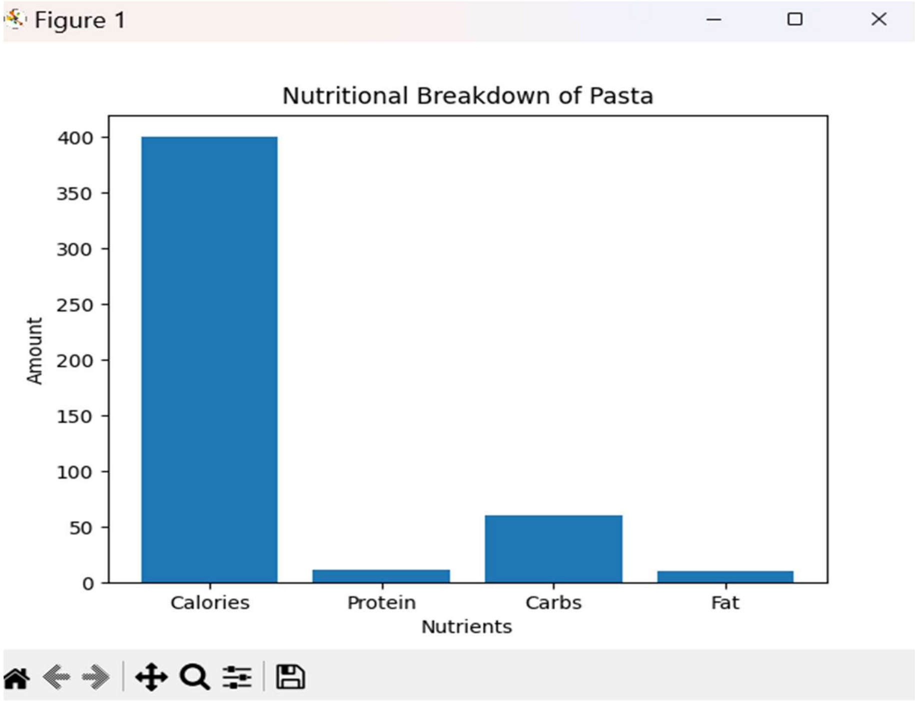
```
ain3.py'
================================
     WEEK 3 - RECIPE FINDER
================================

Available Recipes:
- Pasta
- Omelette
- Salad

Enter recipe name: Pasta

Fetching recipe details...
Recipe found successfully!

Nutritional Breakdown for Pasta
   Nutrient  Value
0  Calories    400
1   Protein     12
2     Carbs     60
3       Fat     10
c:\Users\minur\OneDrive\Desktop\Recipe_Finder_Project(week1)\src\main3.py:66: UserWarning: Starting a Matplo
tlib GUI outside of the main thread will likely fail.
  plt.figure()
c:\Users\minur\OneDrive\Desktop\Recipe_Finder_Project(week1)\src\main3.py:71: UserWarning: Starting a Matplo
tlib GUI outside of the main thread will likely fail.
  plt.show()
```



CHALLENGES AND SOLUTIONS

Challenge 1: Understanding data visualization

Problem: Initially, visualizing nutrition data in a clear and meaningful way was difficult.

Solution: Learned basic plotting techniques in Matplotlib and created bar charts to show the nutritional breakdown of recipes. This made the output more understandable and visually appealing.

Challenge 2: Structuring raw data effectively

Problem: Raw API or sample data was not easy to read when printed directly in the console.

Solution: Used **Pandas Data Frame** to structure the data in a table format, making it easier to read and compare different nutritional components.

Challenge 3: Improving console output readability

Problem: Printed data looked cluttered without formatting.

Solution: Applied formatting methods like df.to_string(index=False) and added descriptive messages to improve readability.

Challenge 4: Handling multiple API requests (planned)

Problem: Making multiple requests at once could slow the program.

Solution: Researched multithreading concepts. Though not implemented fully this week, multithreading was considered for future optimization to improve performance.