

WEEK 1: PROJECT REPORT

NAME: MINURANISAHU, ANANYA, RITHIKA

DATE: 27/01/2026

1. Final Week 1 Look

The project is organized into a professional directory structure to ensure maintainability and scalability

docs: Contains project documentation and visual proof of progress.

src: Contains the core application logic, including main.py.

tests: Reserved for unit testing and code verification scripts.

requirements.txt: A complete list of all external Python libraries required for the project.

2. Summary of Achievements

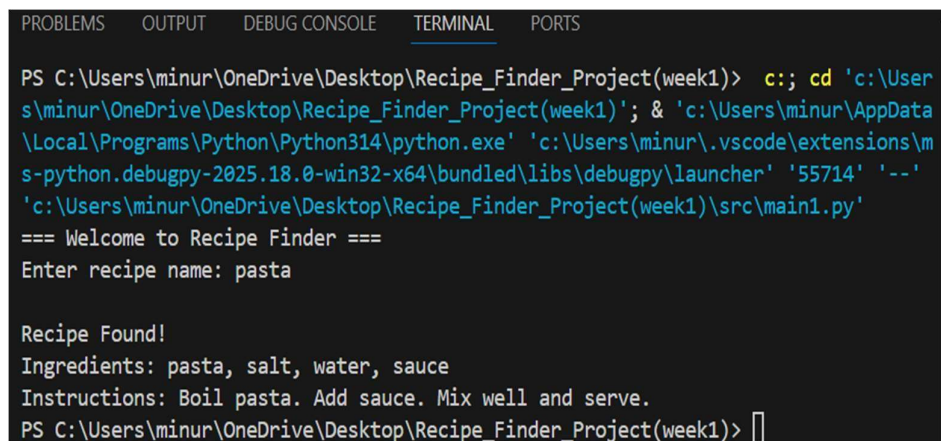
Environment Setup: Successfully configured the Python environment and installed the core tech stack including Pandas, Matplotlib, and Requests.

File Configuration: Resolved file extension issues and established a working script execution path in the terminal.

Initial GUI Development: Designed and launched the "Recipe Finder with Nutrition Analysis" interface using Tkinter, featuring user input fields and search functionality.

API Integration Readiness: Integrated the Spoonacular API key and verified the ability to fetch data from external sources.

WEEK 1 OUTPUT:

A screenshot of a terminal window with a dark background and light-colored text. The terminal shows the execution of a Python script. The prompt is 'PS C:\Users\minur\OneDrive\Desktop\Recipe_Finder_Project(week1)>'. The command entered is 'c.:: cd 'c:\Users\minur\OneDrive\Desktop\Recipe_Finder_Project(week1)'; & 'c:\Users\minur\AppData\Local\Programs\Python\Python314\python.exe' 'c:\Users\minur\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundle\libs\debugpy\launcher' '55714' '--' 'c:\Users\minur\OneDrive\Desktop\Recipe_Finder_Project(week1)\src\main1.py''. The output of the script is: '=== Welcome to Recipe Finder ===', 'Enter recipe name: pasta', 'Recipe Found!', 'Ingredients: pasta, salt, water, sauce', and 'Instructions: Boil pasta. Add sauce. Mix well and serve.'. The terminal prompt is now 'PS C:\Users\minur\OneDrive\Desktop\Recipe_Finder_Project(week1)>'.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\minur\OneDrive\Desktop\Recipe_Finder_Project(week1)> c.:: cd 'c:\Users\minur\OneDrive\Desktop\Recipe_Finder_Project(week1)'; & 'c:\Users\minur\AppData\Local\Programs\Python\Python314\python.exe' 'c:\Users\minur\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundle\libs\debugpy\launcher' '55714' '--' 'c:\Users\minur\OneDrive\Desktop\Recipe_Finder_Project(week1)\src\main1.py'
=== Welcome to Recipe Finder ===
Enter recipe name: pasta

Recipe Found!
Ingredients: pasta, salt, water, sauce
Instructions: Boil pasta. Add sauce. Mix well and serve.
PS C:\Users\minur\OneDrive\Desktop\Recipe_Finder_Project(week1)> 
```

PROJECT CHALLENGES AND SOLUTIONS

Phase: Research and Setup

Challenge 1: Understand the Spoonacular API

Description: Initially, it was challenging to understand the Spoonacular API documentation, including the available endpoints, required parameters, and response format for recipe search and nutrition analysis.

Solution: The API documentation was studied thoroughly. Sample API were tested using a browser and python scripts to understand the structure

of requests and responses. Relevant endpoints for recipe search and nutritional data were identified and noted for implementation.

Challenge 2: API Key Generation and Authentication

Description: The API requests failed in the initial stage due to missing or incorrect API key configuration.

Solutions: An account was created on the Spoonacular platform and a valid API key was generated. The API key was securely integrated into the Python program and tested using sample requests to ensure successful authentication.

Challenge 3: Development Environment Setup

Description: Required Python libraries such as requests, tkinter, pandas, and matplotlib were not available by default, leading to execution errors.

Solution: All required libraries were installed using the pip package manager. Python and Tkinter installations were verified. A proper development environment was configured to ensure smooth execution of the application.

Challenge 4: Designing the Initial GUI Layout

Description: Designing a user-friendly and structured GUI using Tkinter was challenging due to lack of prior layout planning.

Solution: A basic GUI layout was first sketched, including input fields, buttons, and result display areas. Tkinter widgets such as Label, Entry,

Button, and Text were implemented. The layout was refined after testing to improve usability.

Challenge 5: Fetching Data from the API

Description: Errors occurred while fetching recipe data due to incorrect API URLs and parameter formatting.

Solution: The requests library was used to send API requests with proper query parameters. Debugging was performed by printing API responses and correcting URL formats and parameter values accordingly.

Challenge 6: Parsing JSON Response Data

Description: The API returned complex JSON responses, making it difficult to extract relevant information such as recipe name and nutritional values.

Solution: The JSON response structure was analyzed carefully. Required data fields were extracted using Python dictionaries and lists. Only essential information was selected for display in the application.

Challenge 7: Integrating GUI with API Functionality

Description: Linking user input from the GUI with API calls did not work correctly during initial attempts.

Solution: The button click event was properly connected to the API calling function. User input from the Entry widget was passed as a parameter to the API request, and the fetched recipe results were displayed in the GUI output section.

OUTCOME OF WEEK 1

By the end of Week 1, a basic GUI layout was successfully developed, API integration was tested, and recipe data was fetched and displayed. The foundation for further development of filtering, nutrition analysis, and visualization features was established.