Personal Information Manager – Documentation

1. Project Overview and Objectives

The Personal Information Manager is a beginner-level Python project developed as part of Task 1 – Week 1: Python Basics. The main objective of this project is to strengthen the programming foundation by applying core Python concepts such as variables, user input and output, string formatting, loops, and basic validation. This project demonstrates the ability to build a complete, interactive Python program that runs successfully in a development environment.

2. Setup and Installation Instructions

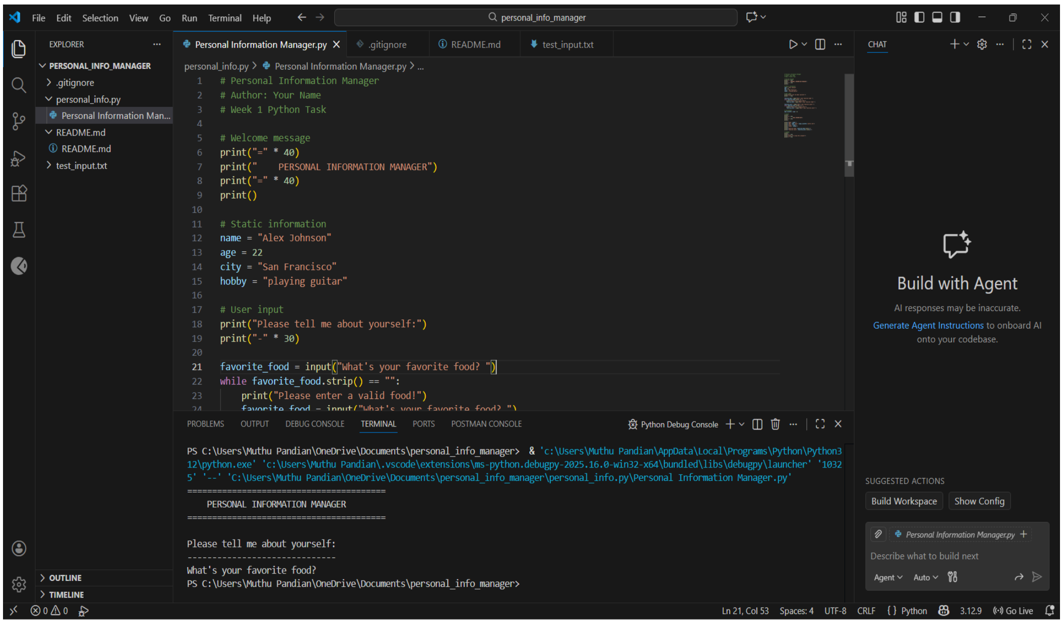
- Install Python 3 on the system and ensure it is added to the system PATH.
- Install Visual Studio Code (VS Code) as the code editor.
- Create a project folder named 'week1-personal-info'.
- Open the folder in VS Code using File → Open Folder.
- Open the terminal in VS Code and run the program using the command: python personal_info.py.
- Follow the prompts displayed in the terminal to enter user information.

3. Code Structure Explanation

The project consists of a primary Python file named personal_info.py. The program begins with a welcome message followed by static personal information stored using variables such as name, age, city, and hobby. It then collects dynamic user input for favorite food and favorite color using the input() function. A while loop is implemented to validate user input and prevent empty values. The program also performs a basic calculation to convert age into months. Finally, all information is displayed using formatted output with f-strings, ensuring clarity and readability.

4. Screenshots of Working Application

The following screenshot shows the Personal Information Manager program running successfully in the Visual Studio Code terminal. It demonstrates proper execution, user input prompts, and formatted output.


File Edit Selection View Go Run Terminal Help
personal_info_manager
EXPLORER
PERSONAL_INFO_MANAGER
.gitignore
personal_info.py
Personal Information Man...
README.md
README.md
test_input.txt
OUTLINE
TIMELINE
Personal Information Manager.py
.gitignore
README.md
test_input.txt
personal_info.py > Personal Information Manager.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
Python Debug Console
CHAT
SUGGESTED ACTIONS
Build Workspace
Show Config
Personal Information Manager.py
Describe what to build next
Agent
Auto
Ln 21, Col 53 Spaces: 4 UTF-8 CRLF {} Python 3.12.9 Go Live

```python
# Personal Information Manager
# Author: Your Name
# Week 1 Python Task

# Welcome message
print("=" * 40)
print("    PERSONAL INFORMATION MANAGER")
print("=" * 40)
print()

# Static information
name = "Alex Johnson"
age = 22
city = "San Francisco"
hobby = "playing guitar"

# User input
print("Please tell me about yourself:")
print("-" * 30)

favorite_food = input("What's your favorite food? ")
while favorite_food.strip() == "":
    print("Please enter a valid food!")
    favorite_food = input("What's your favorite food? ")
```

```
PS C:\Users\Muthu Pandian\OneDrive\Documents\personal_info_manager>  & 'c:\Users\Muthu Pandian\AppData\Local\Programs\Python\Python3
12\python.exe' 'c:\Users\Muthu Pandian\.vscode\extensions\ms-python.debugpy-2025.16.0-win32-x64\bundled\libs\debugpy\launcher' '1032
5' '--' 'C:\Users\Muthu Pandian\OneDrive\Documents\personal_info_manager\personal_info.py\Personal Information Manager.py'
========================================
    PERSONAL INFORMATION MANAGER
========================================

Please tell me about yourself:
------------------------------
What's your favorite food?
PS C:\Users\Muthu Pandian\OneDrive\Documents\personal_info_manager>
```

Build with Agent

AI responses may be inaccurate.

Generate Agent Instructions to onboard AI onto your codebase.

5. Explanation of How Technical Requirements Were Met

- Variables were used to store more than four pieces of personal information.
- User input was collected using the input() function for dynamic data.
- Basic input validation was implemented using a while loop and string methods.
- String formatting was achieved using Python f-strings for structured output.
- The program was tested and executed successfully in VS Code, as shown in the screenshot.