TASK MANAGER AND REMINDER –PYTHON

NAME : MUGLE SRUTHI

REG NO : 12109334

SECTION : K21CK

LOVELY PROFESSIONAL UNIVERSITY ,PUNJAB.

LOVELY PROFESSIONAL UNIVERSITY

Introduction

In today's busy world, effective task management is crucial. Our project offers a simple yet powerful solution for organizing your tasks and staying on top of deadlines. With features like adding tasks, displaying them, marking completion, and setting reminders, we aim to enhance your productivity. Our user-friendly interface ensures ease of use for everyone. Join us on a journey to streamline your daily life and boost your efficiency with the "Task Manager and Reminder" project.

Why Python?

Use Python for this project because of its simplicity and extensive libraries, making development faster and more e£icient.

PROJECT OVERVIEW

The "Task Manager and Reminder" project is a Python-based application designed to streamline task management and enhance productivity. This project addresses the need for efficient task organization in our fast-paced lives and offers a user-friendly solution with key features:

1.Add Task: Users can input and store tasks with titles, descriptions, and due dates, making it easy to keep track of various commitments.

2.Display Tasks: The application allows users to view their task list, providing essential details such as titles, due dates, and completion statuses, ensuring efficient organization.

3.Mark Task as Completed: Users can mark tasks as completed, providing a clear overview of their progress and helping them stay on top of pending tasks.

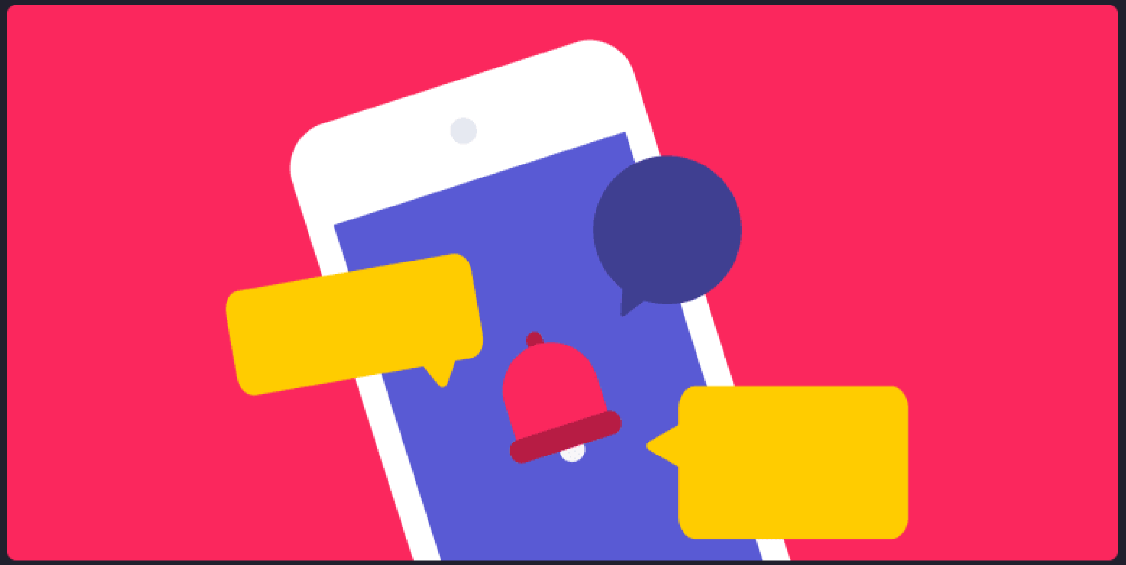
4. **Set Reminders** : Project includes a reminder system that checks for tasks due on the current day and sends timely reminders , reducing the risk of missed deadlines.

5. **User-Friendly Interface**: With intuitive menus and a straightforward interface,

"Task Manager and Reminder" project ensures accessibility and users of all backgrounds.This project is an invaluable tool for individuals professionals, and students ,seeking to optimize their task management,
Prioritize efectively, and reduce the stress associated with missed deadlines.

With Python as the programming language, the "Task Manager and Reminder" project combines simplicity and functionality to o£er an e£icient and user - friendly solution for daily

task management .



How to Use the "Task Manager and Reminder" Project:

Launch the Application: Run the Python script to start the application.

Main Menu: Choose an option from the main menu:

1: Add a new task with title, description, and due date.

2: Display your task list with details.

3: Add a Task: Select option 1, provide task details, and the task will be added to your list.

4: View Tasks: Choose option 2 to see your task list, including titles, due dates, and statuses.

5:Mark Completion: Use option 3 to mark a task as completed when finished.

6: Set Reminders: Select option 4 to receive reminders for tasks due today, helping you stay organized and on top of deadlines.

Microsoft To Do

Code explanation

import datetime import time

The first two lines import the datetime and time modules. These modules provide functions for working with dates and times.

Python

tasks = []

The next line defines a list called tasks. This list will store the tasks that are added by the user.

```
def add_task():
 title = input("Enter task title: ")
 description = input("Enter task description: ")
 due_date = input("Enter due date (YYYY-MM-DD): ")
 tasks.append({"title": title, "description": description, "due_date": due_date, "completed": False})
 print("Task added successfully!")
```

The add_task() function takes four arguments: the task title, the task description, and the due date. The function then creates a dictionary with these values and appends it to the tasks list. The function also prints a message to the user indicating that the task was added successfully.

```
def display_tasks(): print("\nTask List:")
 for idx, task in enumerate(tasks, start=1):
 status = "Completed" if task["completed"] else "Pending"
print(f"{idx}. Title: {task['title']}\tDue Date: {task['due_date']}\tStatus: {status}")
```

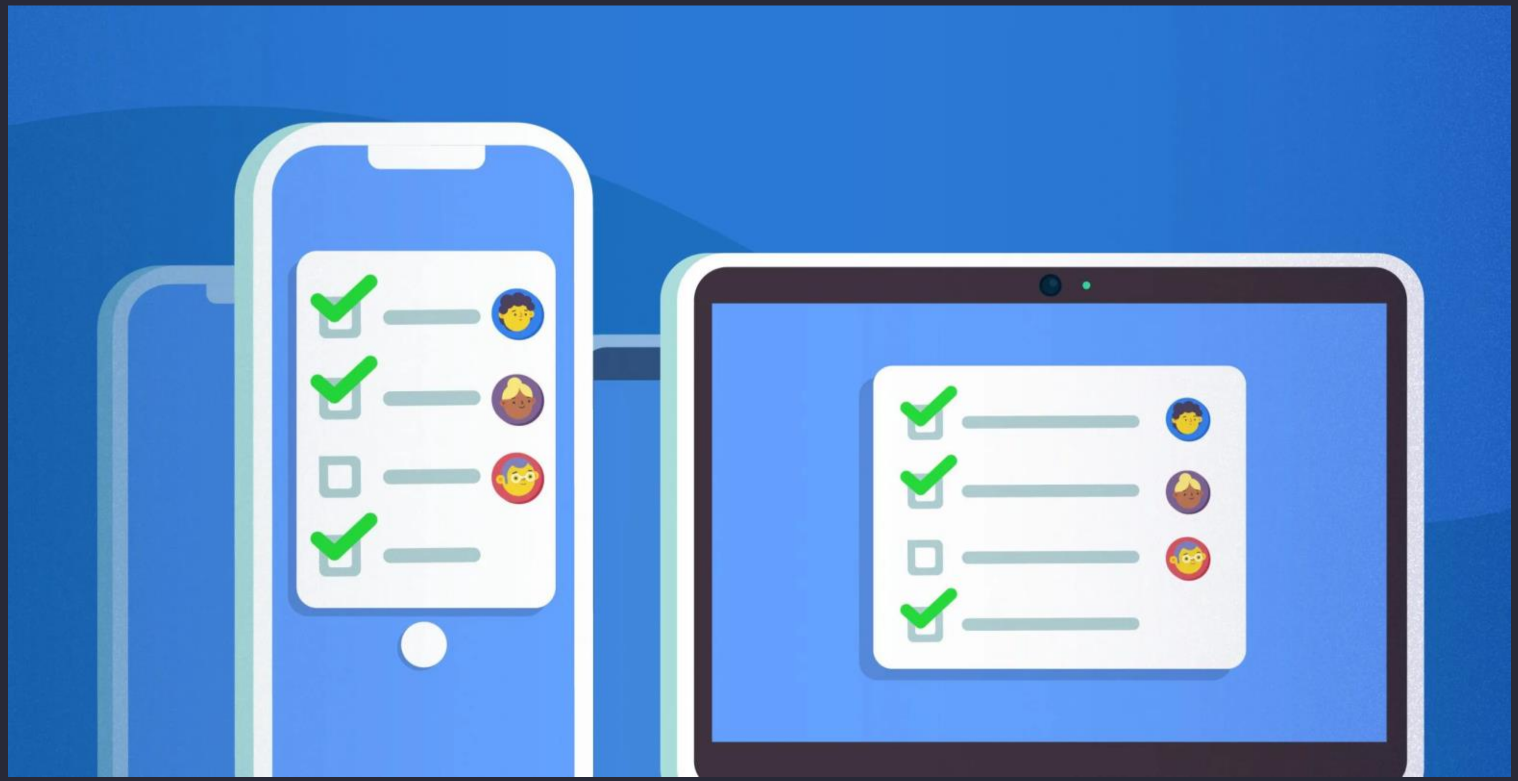
The display_tasks() function prints a list of all the tasks in the tasks list.
The function also indicates the status of each task (completed or pending).

```
def mark_completed():
 display_tasks() choice = int(input("Enter the task number to mark as completed: ")) - 1
if 0 <= choice < len(tasks): tasks[choice]["completed"] = True print("Task marked as completed!")
else: print("Invalid choice.")
```

The mark_completed() function prompts the user to enter the number of
the task to mark as completed. The function then checks if the number is valid.
If it is, the function sets the completed flag for that task to True. Otherwise, the function prints an error message.

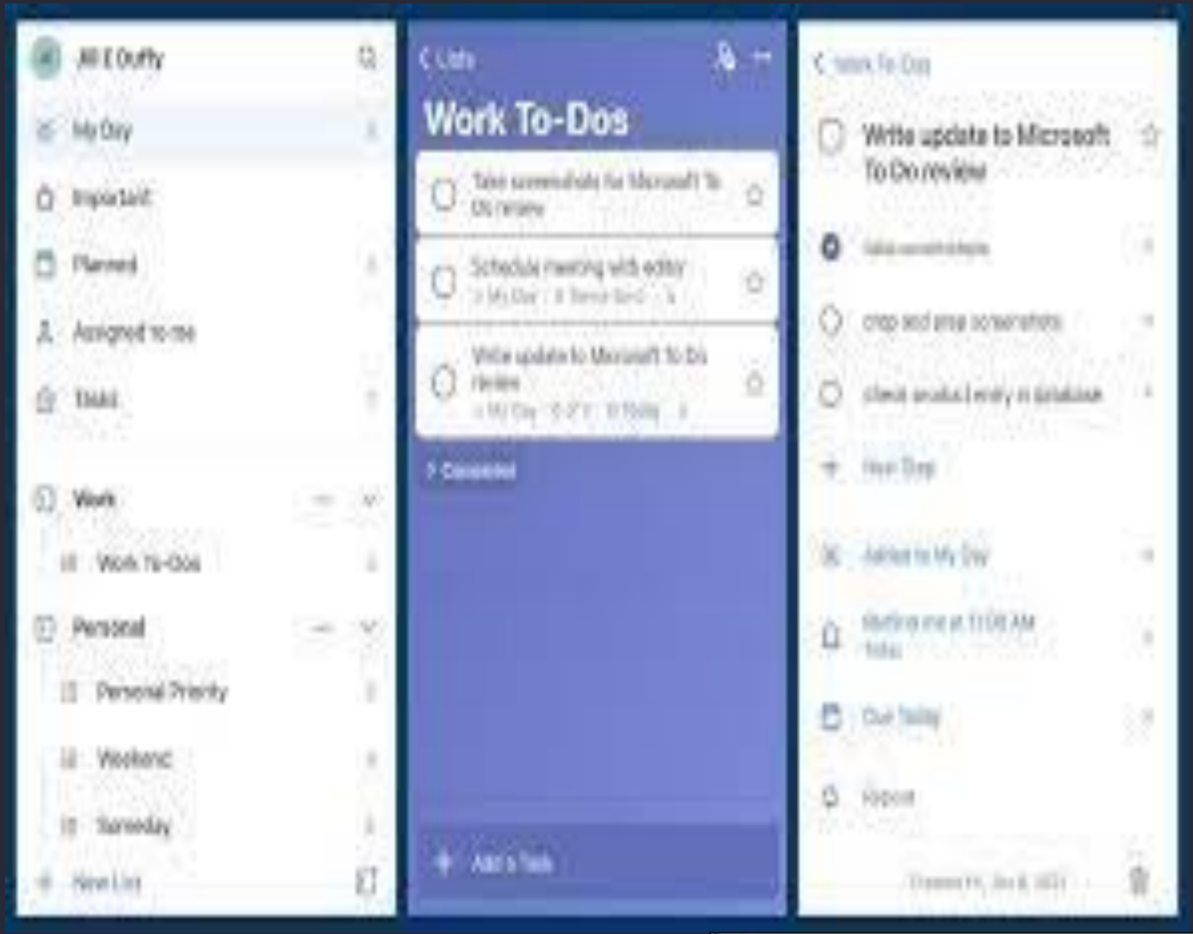
```
def set_reminders():
while True: current_time = datetime.datetime.now()
for task in tasks: due_date = datetime.datetime.strptime(task["due_date"], "%Y-%m-%d")
 if not task["completed"] and current_time.date() == due_date.date():
 print(f"Reminder: Task '{task['title']}' is due today!") time.sleep(60)
```

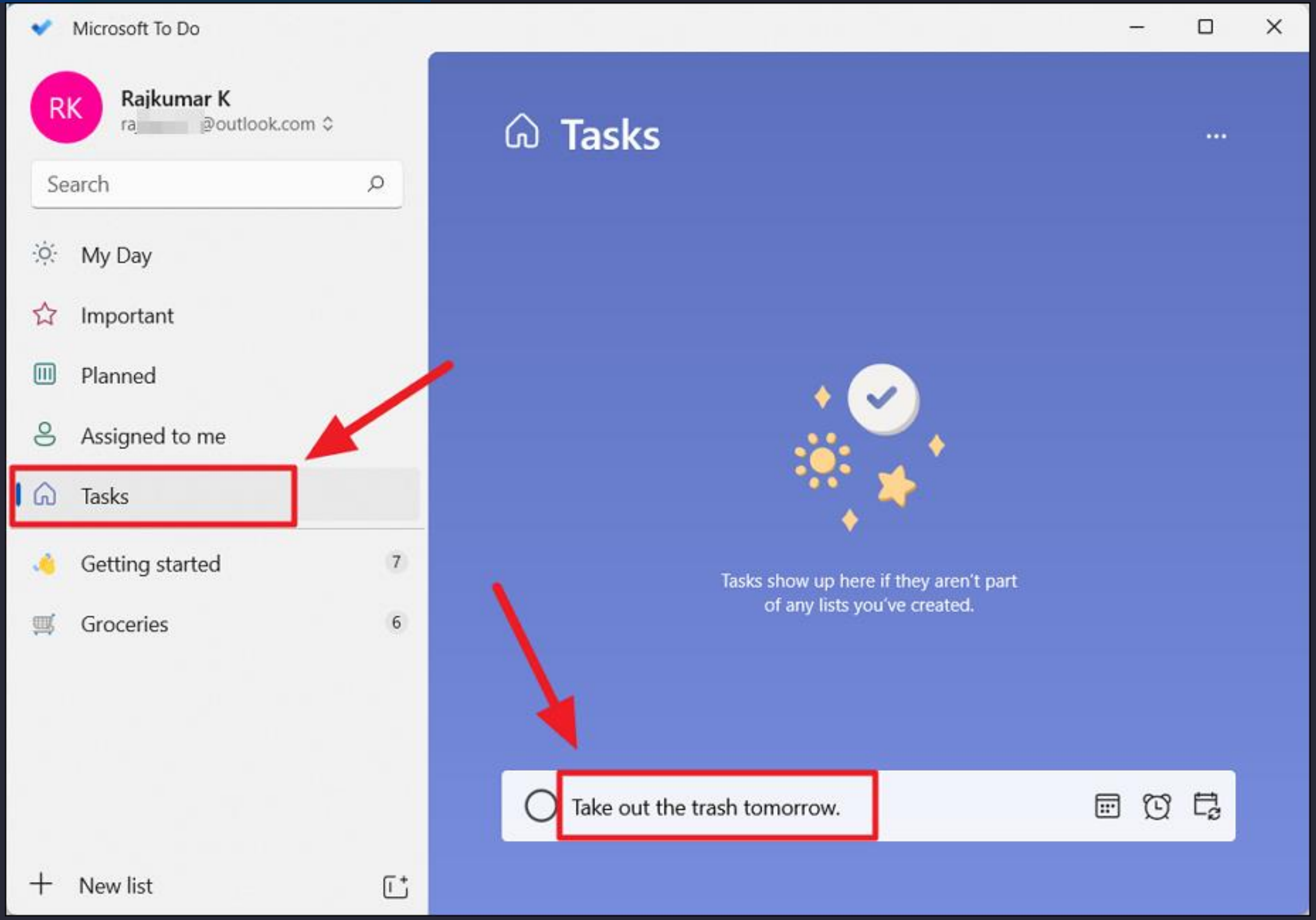
The set_reminders() function checks for tasks that are due today.
If it finds any, the function prints a reminder message. The function then sleeps
for 60 seconds (1 minute) and checks again for due tasks.



```
def main():
 print("Task Manager and Reminder")
 while True: print("\nMenu:")
 print("1. Add Task")
 print("2. Display Tasks")
print("3. Mark Task as Completed")
print("4. Set Reminders")
print("5. Exit") choice = input("Enter your choice: ")
 if choice == "1":
 add_task()
 elif choice == "2":
 display_tasks()
elif choice == "3":
mark_completed()
elif choice == "4":
 set_reminders()
elif choice == "5":
 print("Exiting.")
Break
 else:
 print("Invalid choice. Please select a valid option.")
 if __name__ == "__main__": main()
```

The main() function is the entry point for the program.
It prints a welcome message and then enters a loop. In the loop,
the function prints a menu of options and prompts the user to select an option.
The function then calls the appropriate function based on the user's choice.
The loop continues until the user chooses to exit the program.

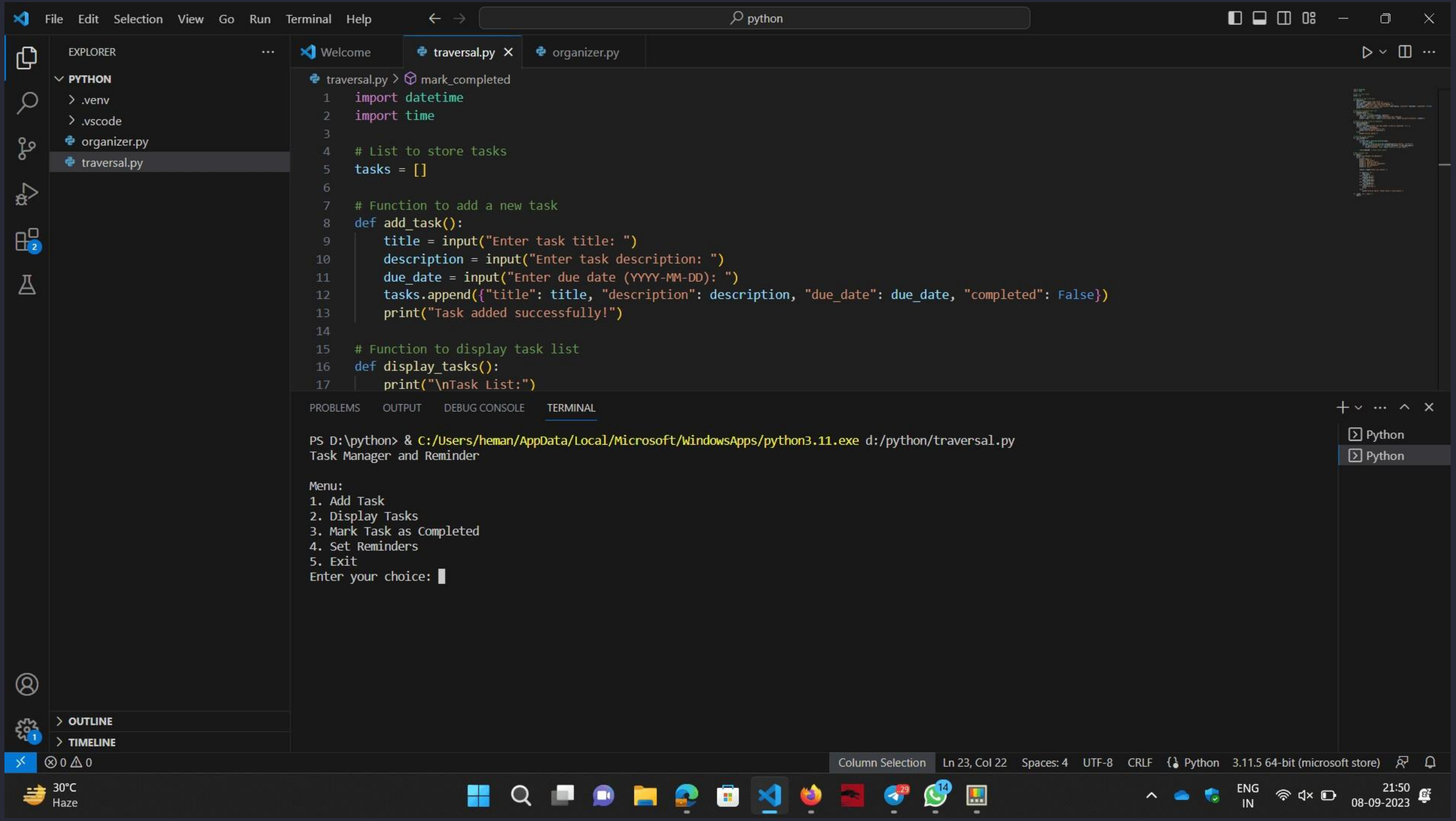




Tasks show up here if they aren't part of any lists you've created.

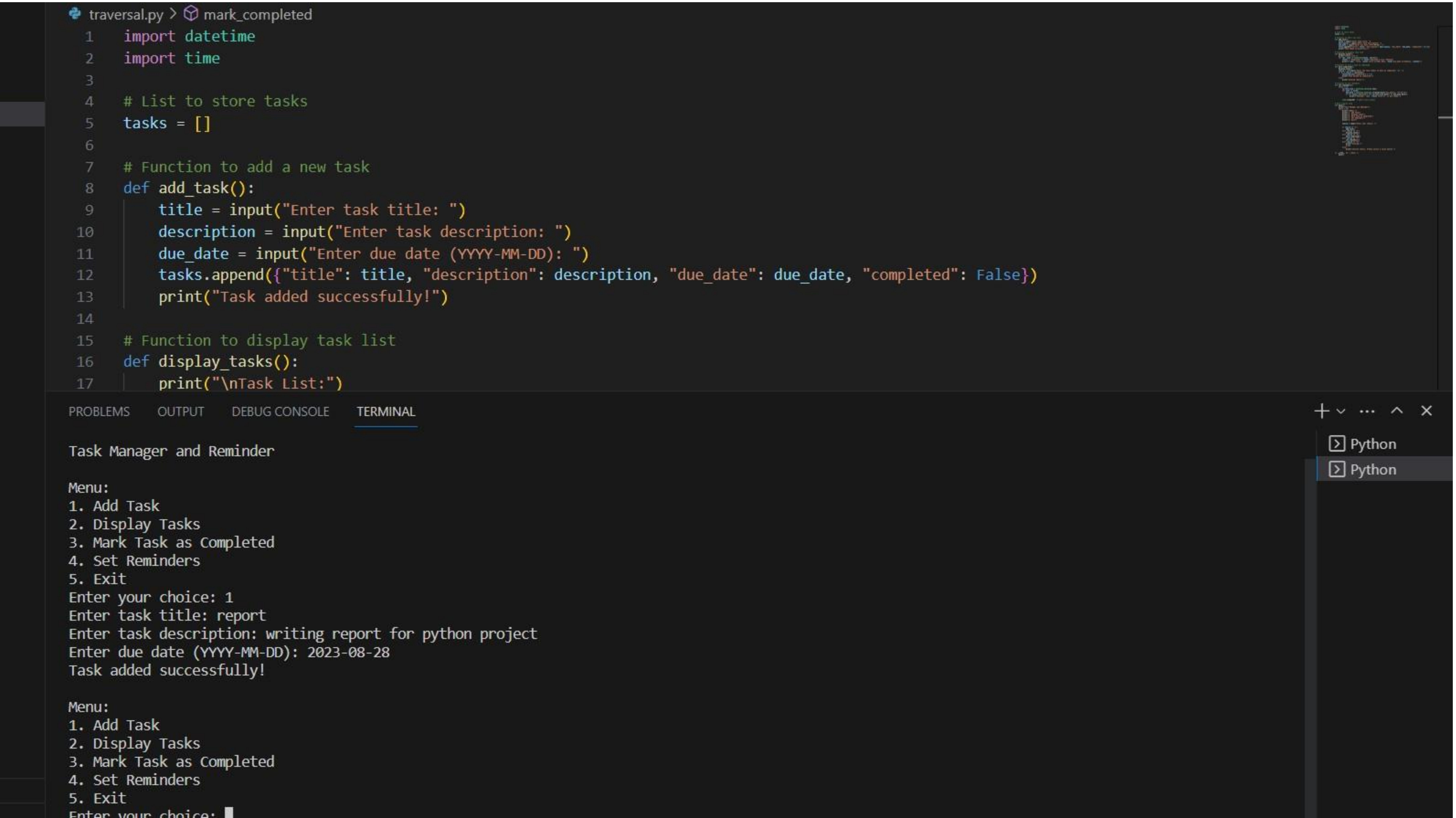
Take out the trash tomorrow.

Output images


File Edit Selection View Go Run Terminal Help
python
EXPLORER
PYTHON
.venv
.vscode
organizer.py
traversal.py
Welcome
traversal.py
organizer.py
traversal.py > mark_completed
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python
Python
OUTLINE
TIMELINE
Column Selection Ln 23, Col 22 Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit (microsoft store)
30°C Haze
ENG IN
21:50 08-09-2023

```python
import datetime
import time

# List to store tasks
tasks = []

# Function to add a new task
def add_task():
    title = input("Enter task title: ")
    description = input("Enter task description: ")
    due_date = input("Enter due date (YYYY-MM-DD): ")
    tasks.append({"title": title, "description": description, "due_date": due_date, "completed": False})
    print("Task added successfully!")

# Function to display task list
def display_tasks():
    print("\nTask List:")
```

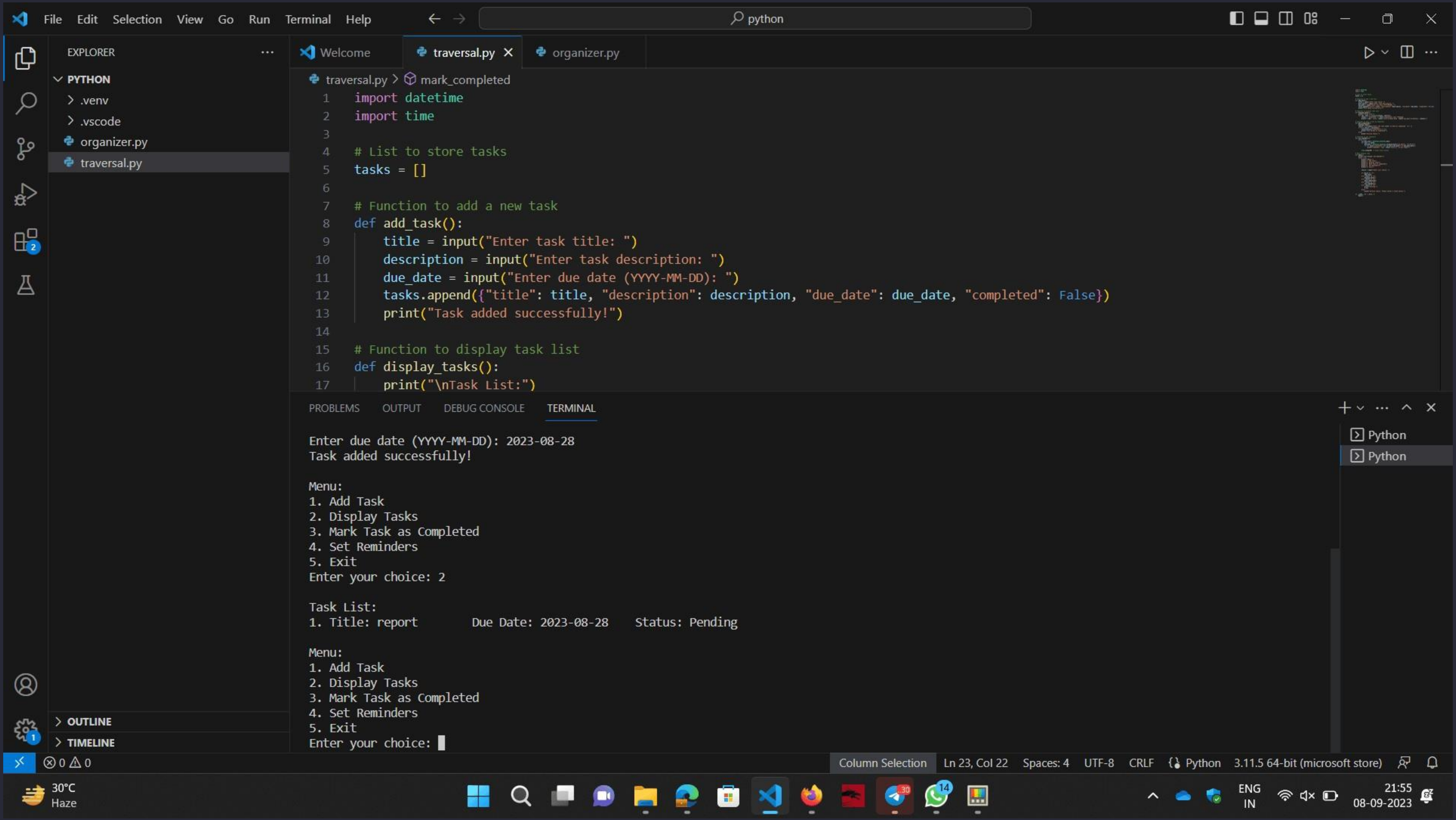
```
PS D:\python> & C:/Users/heman/AppData/Local/Microsoft/WindowsApps/python3.11.exe d:/python/traversal.py
Task Manager and Reminder

Menu:
1. Add Task
2. Display Tasks
3. Mark Task as Completed
4. Set Reminders
5. Exit
Enter your choice: 
```



traversal.py > mark_completed

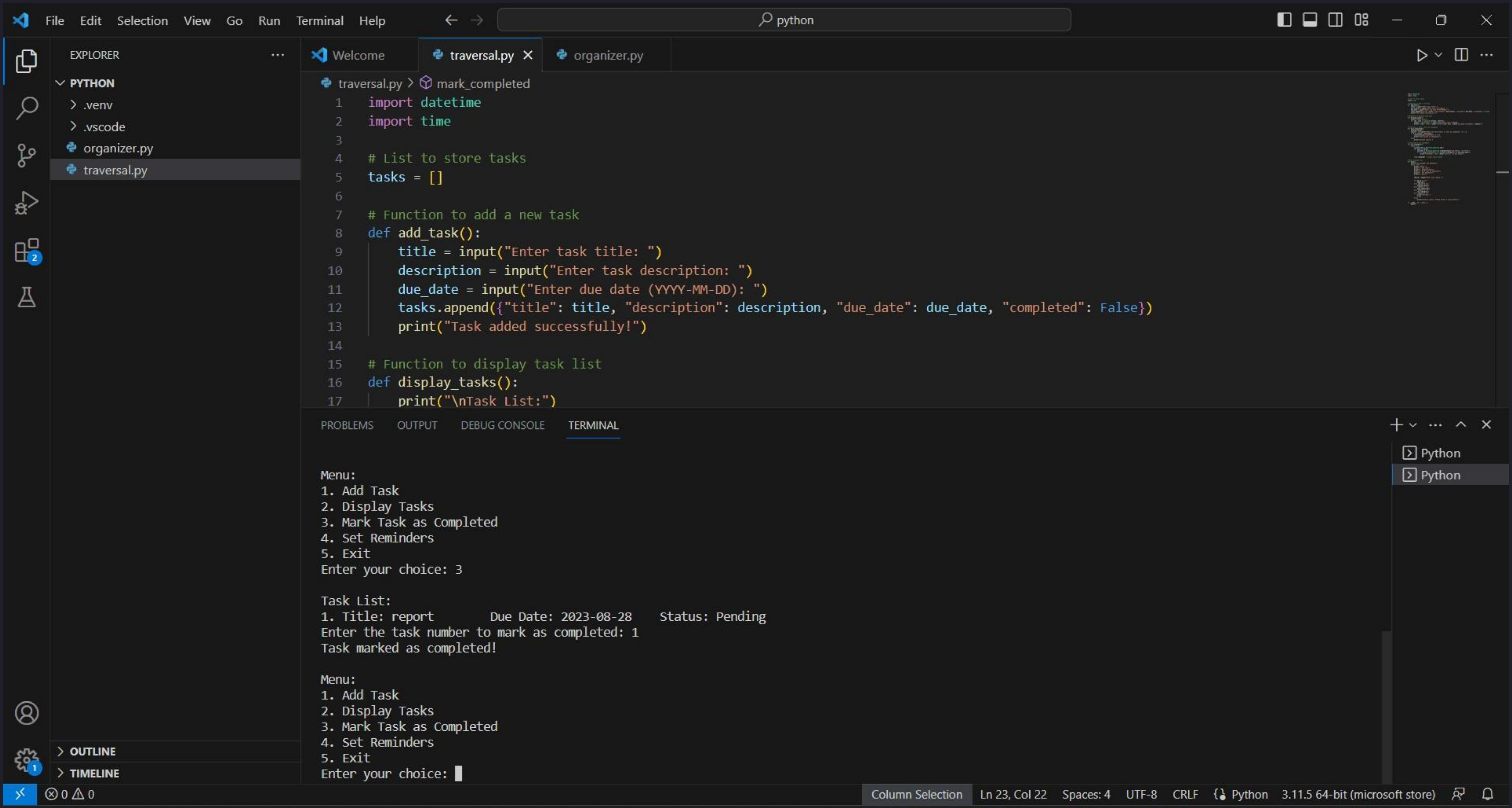
```python
import datetime
import time

# List to store tasks
tasks = []

# Function to add a new task
def add_task():
    title = input("Enter task title: ")
    description = input("Enter task description: ")
    due_date = input("Enter due date (YYYY-MM-DD): ")
    tasks.append({"title": title, "description": description, "due_date": due_date, "completed": False})
    print("Task added successfully!")

# Function to display task list
def display_tasks():
    print("\nTask List:")
```

```
Task Manager and Reminder

Menu:
1. Add Task
2. Display Tasks
3. Mark Task as Completed
4. Set Reminders
5. Exit
Enter your choice: 1
Enter task title: report
Enter task description: writing report for python project
Enter due date (YYYY-MM-DD): 2023-08-28
Task added successfully!

Menu:
1. Add Task
2. Display Tasks
3. Mark Task as Completed
4. Set Reminders
5. Exit
Enter your choice:
```



```python
import datetime
import time

# List to store tasks
tasks = []

# Function to add a new task
def add_task():
    title = input("Enter task title: ")
    description = input("Enter task description: ")
    due_date = input("Enter due date (YYYY-MM-DD): ")
    tasks.append({"title": title, "description": description, "due_date": due_date, "completed": False})
    print("Task added successfully!")

# Function to display task list
def display_tasks():
    print("\nTask List:")
```

```
Enter due date (YYYY-MM-DD): 2023-08-28
Task added successfully!

Menu:
1. Add Task
2. Display Tasks
3. Mark Task as Completed
4. Set Reminders
5. Exit
Enter your choice: 2

Task List:
1. Title: report        Due Date: 2023-08-28    Status: Pending

Menu:
1. Add Task
2. Display Tasks
3. Mark Task as Completed
4. Set Reminders
5. Exit
Enter your choice: 
```



```python
import datetime
import time

# List to store tasks
tasks = []

# Function to add a new task
def add_task():
    title = input("Enter task title: ")
    description = input("Enter task description: ")
    due_date = input("Enter due date (YYYY-MM-DD): ")
    tasks.append({"title": title, "description": description, "due_date": due_date, "completed": False})
    print("Task added successfully!")

# Function to display task list
def display_tasks():
    print("\nTask List:")
```

```
Menu:
1. Add Task
2. Display Tasks
3. Mark Task as Completed
4. Set Reminders
5. Exit
Enter your choice: 3

Task List:
1. Title: report        Due Date: 2023-08-28    Status: Pending
Enter the task number to mark as completed: 1
Task marked as completed!

Menu:
1. Add Task
2. Display Tasks
3. Mark Task as Completed
4. Set Reminders
5. Exit
Enter your choice: 
```

Conclusion

In summary, the "Task Manager and Reminder" project, driven by Python's simplicity and functionality, ofers a practical solution for e£ective task management and timely reminders. By using this application, you can efficiently organize your tasks, mark them as completed, and receive reminders for imminent deadlines. Its user-friendly interface ensures accessibility for all users. Embrace this tool to streamline your daily life, prioritize tasks, and reduce the risk of missing important deadlines, ultimately leading to enhanced productivity and peace of mind.

THANKYOU!

Do you have any questions?