Mini Project

Command-Line To-Do List Manager

Source code:

```
import json
from datetime import datetime, timedelta

# File where tasks will be saved
TASKS_FILE = 'tasks.json'

# Priority levels
PRIORITY_LEVELS = ['low', 'medium', 'high']


# Load tasks from file
def load_tasks():
    try:
        with open(TASKS_FILE, 'r') as file:
            tasks = json.load(file)
    except (FileNotFoundError, json.JSONDecodeError):
        tasks = []
    return tasks


# Save tasks to file
def save_tasks(tasks):
    with open(TASKS_FILE, 'w') as file:
        json.dump(tasks, file, indent=4)
```

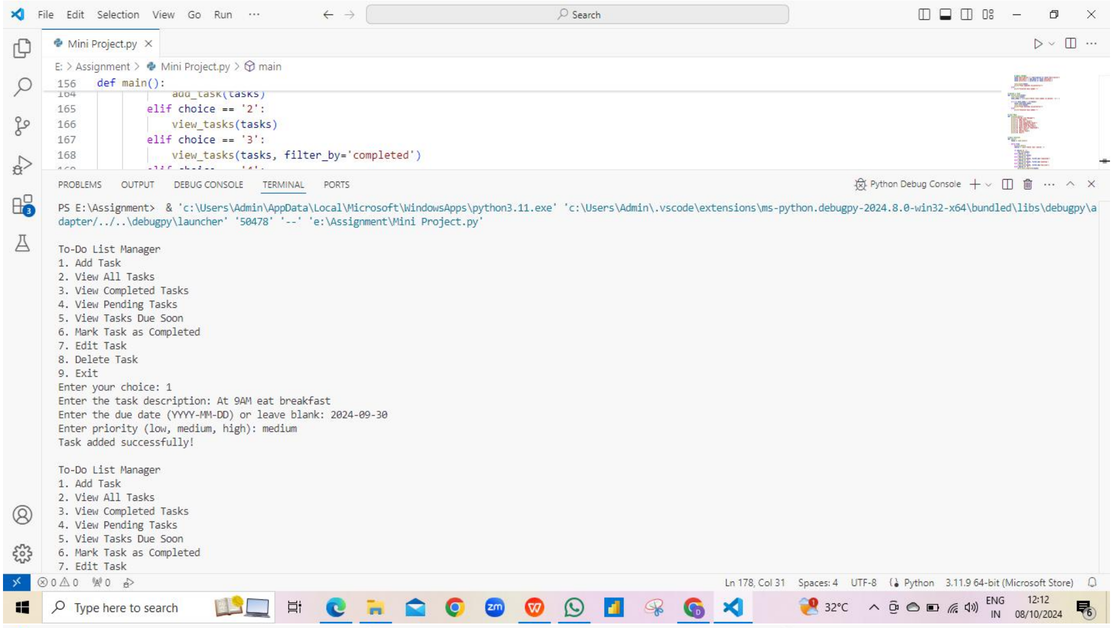
```python
# Add a new task
def add_task(tasks):
    description = input("Enter the task description: ")
    due_date = input("Enter the due date (YYYY-MM-DD) or leave blank: ")

    if due_date:
        try:
            due_date = datetime.strptime(due_date, '%Y-%m-%d').strftime('%Y-%m-%d')
        except ValueError:
            print("Invalid date format. Task not added.")
            return

    priority = input(f"Enter priority (low, medium, high): ").lower()
    if priority not in PRIORITY_LEVELS:
        print("Invalid priority. Task not added.")
        return

    task = {
        'description': description,
        'due_date': due_date or None,
        'completed': False,
        'priority': priority
    }
    tasks.append(task)
    save_tasks(tasks)
    print("Task added successfully!")
```

```
# View tasks based on filter
def view_tasks(tasks, filter_by=None):
    if not tasks:
        print("No tasks available.")
        return

    filtered_tasks = tasks

    if filter_by == 'completed':
        filtered_tasks = [task for task in tasks if task['completed']]
    elif filter_by == 'pending':
        filtered_tasks = [task for task in tasks if not
task['completed']]
    elif filter_by == 'due_soon':
        today = datetime.now().date()
        soon = today + timedelta(days=3)
        filtered_tasks = [task for task in tasks if task['due_date']
and datetime.strptime(task['due_date'], '%Y-%m-%d').date() <= soon and
not task['completed']]

    if not filtered_tasks:
        print("No tasks found for the selected filter.")
        return

    for idx, task in enumerate(filtered_tasks, 1):
        status = "Completed" if task['completed'] else "Pending"
        print(f"{idx}. {task['description']} | Due: {task['due_date']}
| Status: {status} | Priority: {task['priority']}")


# Mark task as complete
def mark_task_completed(tasks):
    view_tasks(tasks, filter_by='pending')
    task_index = int(input("Enter task number to mark as completed: "))
- 1

    if 0 <= task_index < len(tasks):
        tasks[task_index]['completed'] = True
        save_tasks(tasks)
        print("Task marked as completed!")
    else:
        print("Invalid task number.")
```

```python
# Edit a task
def edit_task(tasks):
    view_tasks(tasks)
    task_index = int(input("Enter task number to edit: ")) - 1

    if 0 <= task_index < len(tasks):
        task = tasks[task_index]
        print(f"Editing task: {task['description']}")
        description = input(f"Enter new description (leave blank to keep '{task['description']}'): ")
        due_date = input(f"Enter new due date (YYYY-MM-DD) or leave blank to keep '{task['due_date']}': ")

        if due_date:
            try:
                due_date = datetime.strptime(due_date, '%Y-%m-%d').strftime('%Y-%m-%d')
            except ValueError:
                print("Invalid date format. Task not updated.")
                return

        priority = input(f"Enter new priority (low, medium, high) or leave blank to keep '{task['priority']}': ").lower()
        if priority and priority not in PRIORITY_LEVELS:
            print("Invalid priority. Task not updated.")
            return

        # Apply changes
        task['description'] = description or task['description']
        task['due_date'] = due_date or task['due_date']
        task['priority'] = priority or task['priority']

        save_tasks(tasks)
        print("Task updated successfully!")
    else:
        print("Invalid task number.")


# Delete a task
def delete_task(tasks):
    view_tasks(tasks)
    task_index = int(input("Enter task number to delete: ")) - 1

    if 0 <= task_index < len(tasks):
        tasks.pop(task_index)
        save_tasks(tasks)
        print("Task deleted successfully!")
    else:
        print("Invalid task number.")
```

```python
# User Menu
def display_menu():
    print("\nTo-Do List Manager")
    print("1. Add Task")
    print("2. View All Tasks")
    print("3. View Completed Tasks")
    print("4. View Pending Tasks")
    print("5. View Tasks Due Soon")
    print("6. Mark Task as Completed")
    print("7. Edit Task")
    print("8. Delete Task")
    print("9. Exit")


# Main function
def main():
    tasks = load_tasks()

    while True:
        display_menu()
        choice = input("Enter your choice: ")

        if choice == '1':
            add_task(tasks)
        elif choice == '2':
            view_tasks(tasks)
        elif choice == '3':
            view_tasks(tasks, filter_by='completed')
        elif choice == '4':
            view_tasks(tasks, filter_by='pending')
        elif choice == '5':
            view_tasks(tasks, filter_by='due_soon')
        elif choice == '6':
            mark_task_completed(tasks)
        elif choice == '7':
            edit_task(tasks)
        elif choice == '8':
            delete_task(tasks)
        elif choice == '9':
            print("Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.")


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

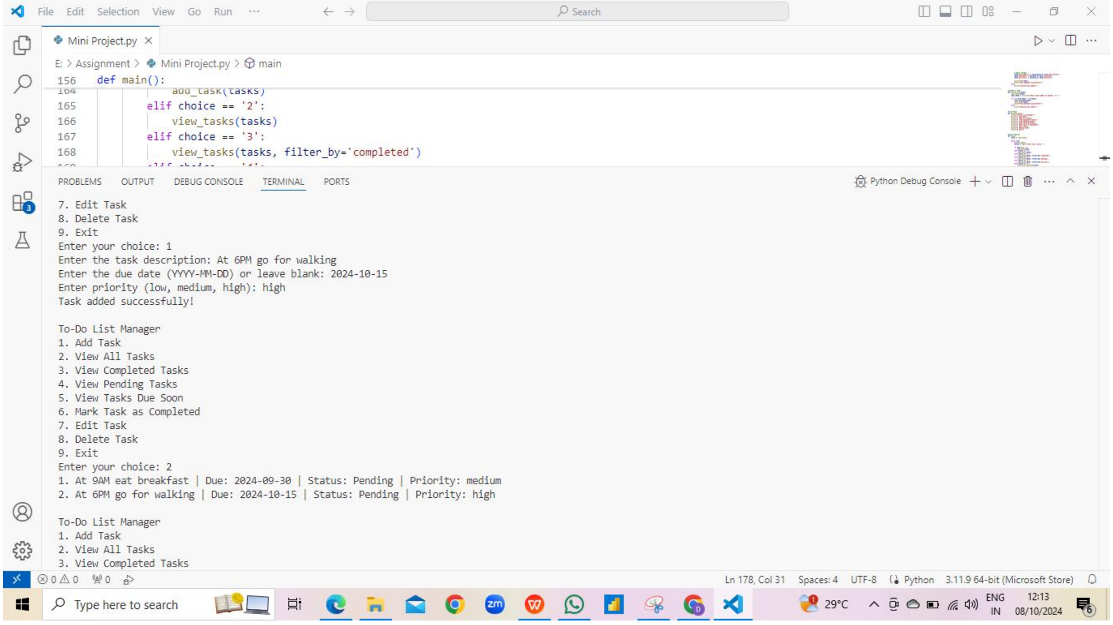
Output Screens:



File Edit Selection View Go Run ···
Search
Mini Project.py
E: > Assignment > Mini Project.py > main
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python Debug Console
Ln 178, Col 31 Spaces: 4 UTF-8 Python 3.11.9 64-bit (Microsoft Store)
Type here to search
32°C
ENG IN
12:12
08/10/2024

```
def main():
            add_task(tasks)
        elif choice == '2':
            view_tasks(tasks)
        elif choice == '3':
            view_tasks(tasks, filter_by='completed')
```

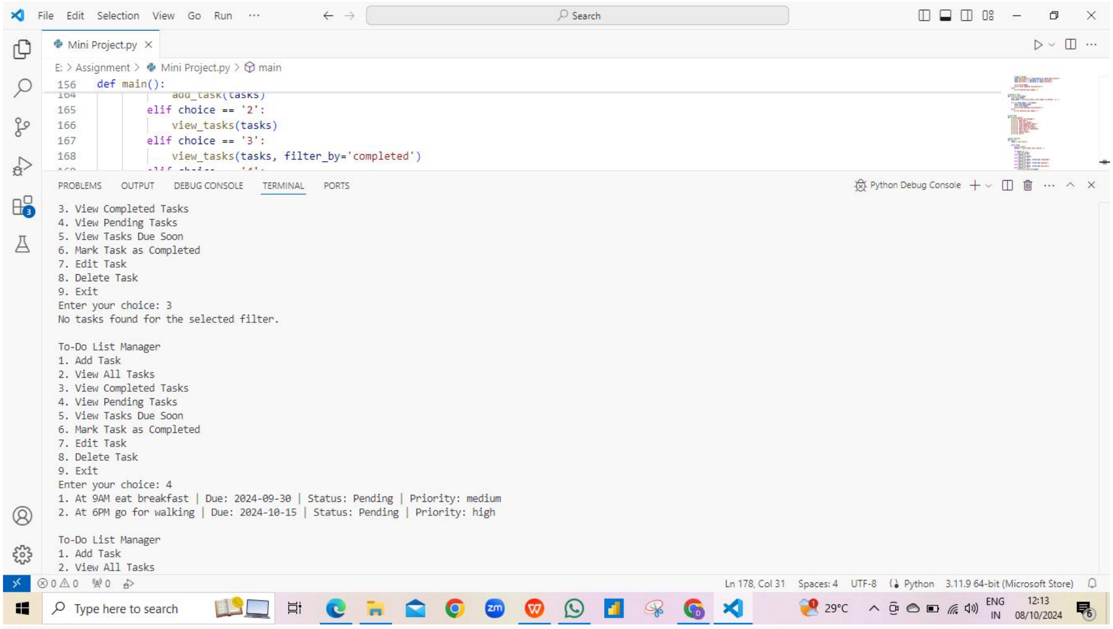
```
PS E:\Assignment>  & 'c:\Users\Admin\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\Admin\.vscode\extensions\ms-python.debugpy-2024.8.0-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '50478' '--' 'e:\Assignment\Mini Project.py'

To-Do List Manager
1. Add Task
2. View All Tasks
3. View Completed Tasks
4. View Pending Tasks
5. View Tasks Due Soon
6. Mark Task as Completed
7. Edit Task
8. Delete Task
9. Exit
Enter your choice: 1
Enter the task description: At 9AM eat breakfast
Enter the due date (YYYY-MM-DD) or leave blank: 2024-09-30
Enter priority (low, medium, high): medium
Task added successfully!

To-Do List Manager
1. Add Task
2. View All Tasks
3. View Completed Tasks
4. View Pending Tasks
5. View Tasks Due Soon
6. Mark Task as Completed
7. Edit Task
```



File Edit Selection View Go Run ···
Search
Mini Project.py
E: > Assignment > Mini Project.py > main
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python Debug Console
Ln 178, Col 31 Spaces: 4 UTF-8 Python 3.11.9 64-bit (Microsoft Store)
Type here to search
29°C
ENG IN
12:13
08/10/2024

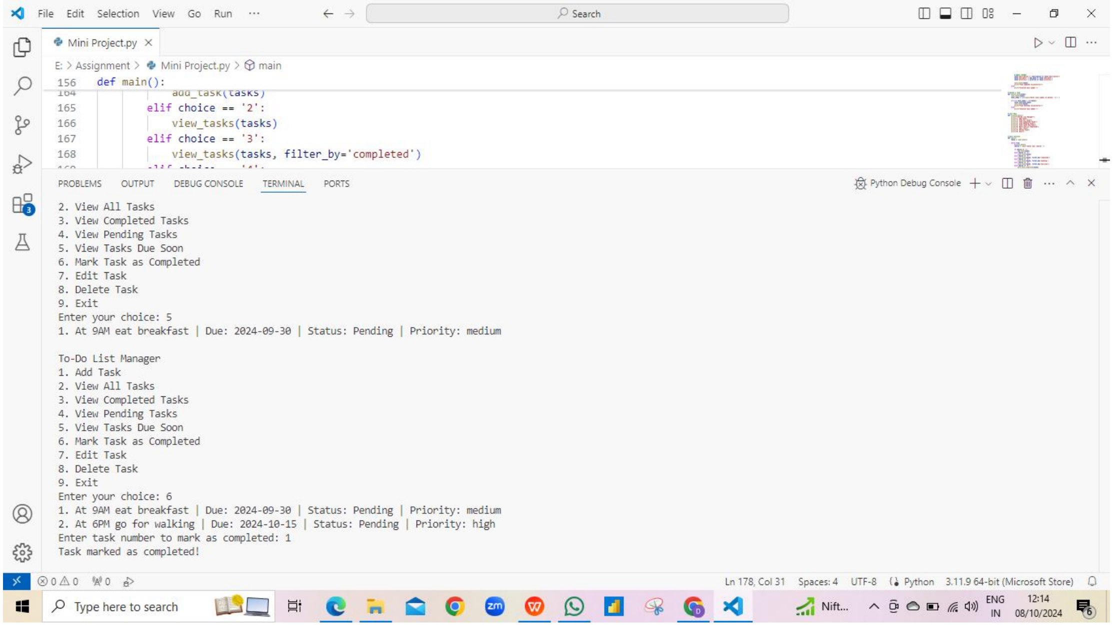
```
def main():
            add_task(tasks)
        elif choice == '2':
            view_tasks(tasks)
        elif choice == '3':
            view_tasks(tasks, filter_by='completed')
```

```
7. Edit Task
8. Delete Task
9. Exit
Enter your choice: 1
Enter the task description: At 6PM go for walking
Enter the due date (YYYY-MM-DD) or leave blank: 2024-10-15
Enter priority (low, medium, high): high
Task added successfully!

To-Do List Manager
1. Add Task
2. View All Tasks
3. View Completed Tasks
4. View Pending Tasks
5. View Tasks Due Soon
6. Mark Task as Completed
7. Edit Task
8. Delete Task
9. Exit
Enter your choice: 2
1. At 9AM eat breakfast | Due: 2024-09-30 | Status: Pending | Priority: medium
2. At 6PM go for walking | Due: 2024-10-15 | Status: Pending | Priority: high

To-Do List Manager
1. Add Task
2. View All Tasks
3. View Completed Tasks
```


File Edit Selection View Go Run
Search
Mini Project.py
E: > Assignment > Mini Project.py > main
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python Debug Console
Ln 178, Col 31 Spaces: 4 UTF-8 Python 3.11.9 64-bit (Microsoft Store)
Type here to search
29°C
ENG IN 12:13 08/10/2024

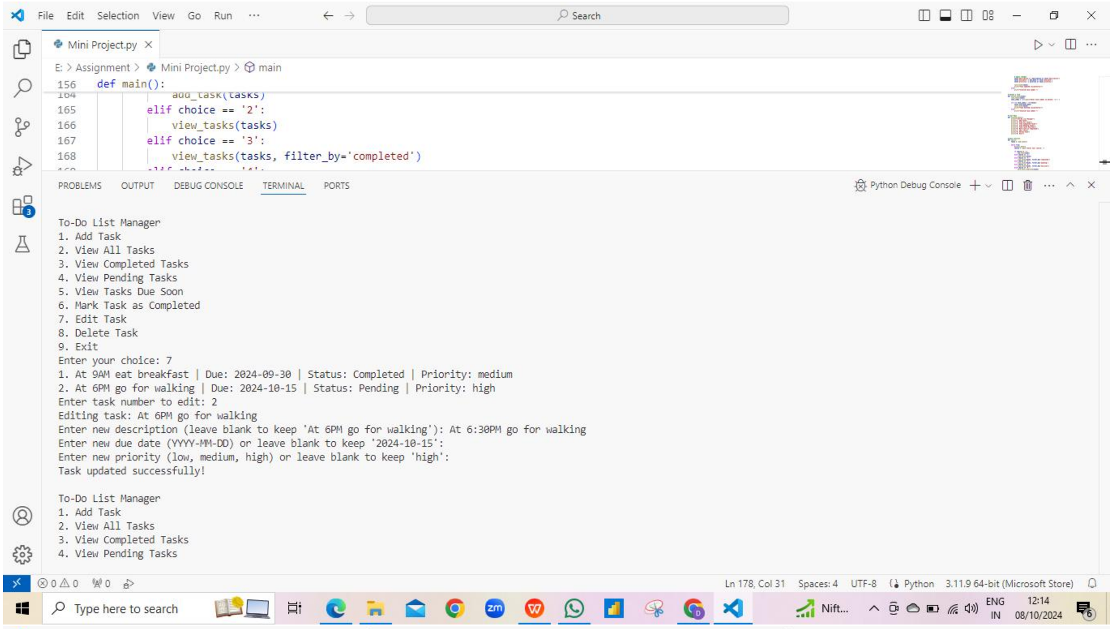
```python
def main():
            add_task(tasks)
        elif choice == '2':
            view_tasks(tasks)
        elif choice == '3':
            view_tasks(tasks, filter_by='completed')
```

```
3. View Completed Tasks
4. View Pending Tasks
5. View Tasks Due Soon
6. Mark Task as Completed
7. Edit Task
8. Delete Task
9. Exit
Enter your choice: 3
No tasks found for the selected filter.

To-Do List Manager
1. Add Task
2. View All Tasks
3. View Completed Tasks
4. View Pending Tasks
5. View Tasks Due Soon
6. Mark Task as Completed
7. Edit Task
8. Delete Task
9. Exit
Enter your choice: 4
1. At 9AM eat breakfast | Due: 2024-09-30 | Status: Pending | Priority: medium
2. At 6PM go for walking | Due: 2024-10-15 | Status: Pending | Priority: high

To-Do List Manager
1. Add Task
2. View All Tasks
```


File Edit Selection View Go Run
Search
Mini Project.py
E: > Assignment > Mini Project.py > main
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python Debug Console
Ln 178, Col 31 Spaces: 4 UTF-8 Python 3.11.9 64-bit (Microsoft Store)
Type here to search
Nift...
ENG IN 12:14 08/10/2024

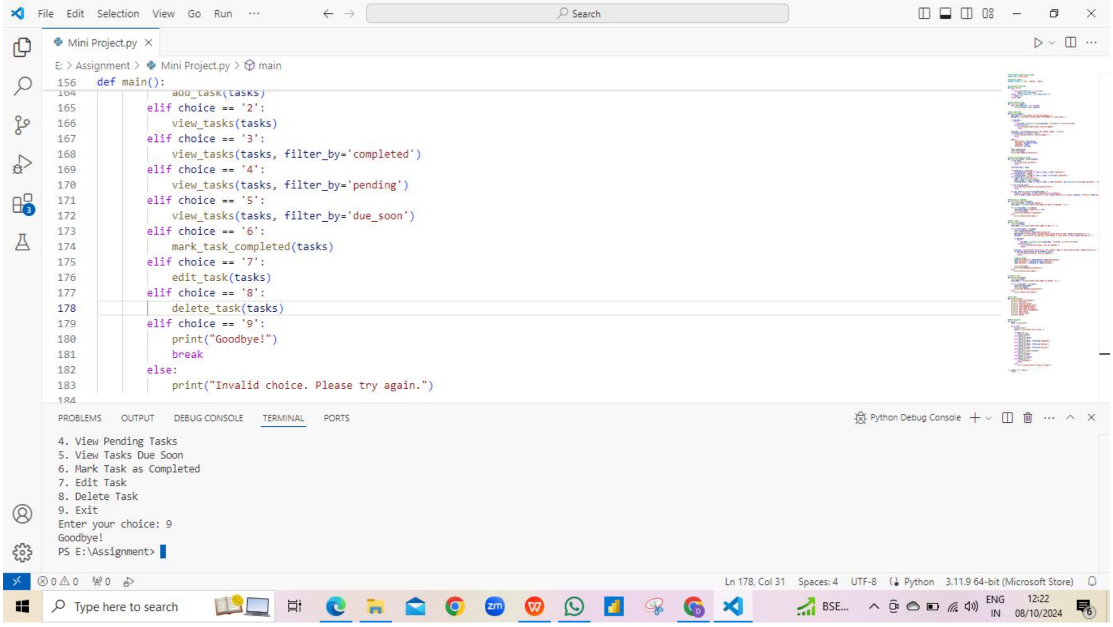
```python
def main():
            add_task(tasks)
        elif choice == '2':
            view_tasks(tasks)
        elif choice == '3':
            view_tasks(tasks, filter_by='completed')
```

```
2. View All Tasks
3. View Completed Tasks
4. View Pending Tasks
5. View Tasks Due Soon
6. Mark Task as Completed
7. Edit Task
8. Delete Task
9. Exit
Enter your choice: 5
1. At 9AM eat breakfast | Due: 2024-09-30 | Status: Pending | Priority: medium

To-Do List Manager
1. Add Task
2. View All Tasks
3. View Completed Tasks
4. View Pending Tasks
5. View Tasks Due Soon
6. Mark Task as Completed
7. Edit Task
8. Delete Task
9. Exit
Enter your choice: 6
1. At 9AM eat breakfast | Due: 2024-09-30 | Status: Pending | Priority: medium
2. At 6PM go for walking | Due: 2024-10-15 | Status: Pending | Priority: high
Enter task number to mark as completed: 1
Task marked as completed!
```


File Edit Selection View Go Run ···
Search
Mini Project.py
E: > Assignment > Mini Project.py > main
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python Debug Console
Ln 178, Col 31 Spaces: 4 UTF-8 Python 3.11.9 64-bit (Microsoft Store)
Type here to search
Nift...
ENG IN 12:14 08/10/2024

```python
def main():
            add_task(tasks)
        elif choice == '2':
            view_tasks(tasks)
        elif choice == '3':
            view_tasks(tasks, filter_by='completed')
```

```
To-Do List Manager
1. Add Task
2. View All Tasks
3. View Completed Tasks
4. View Pending Tasks
5. View Tasks Due Soon
6. Mark Task as Completed
7. Edit Task
8. Delete Task
9. Exit
Enter your choice: 7
1. At 9AM eat breakfast | Due: 2024-09-30 | Status: Completed | Priority: medium
2. At 6PM go for walking | Due: 2024-10-15 | Status: Pending | Priority: high
Enter task number to edit: 2
Editing task: At 6PM go for walking
Enter new description (leave blank to keep 'At 6PM go for walking'): At 6:30PM go for walking
Enter new due date (YYYY-MM-DD) or leave blank to keep '2024-10-15':
Enter new priority (low, medium, high) or leave blank to keep 'high':
Task updated successfully!

To-Do List Manager
1. Add Task
2. View All Tasks
3. View Completed Tasks
4. View Pending Tasks
```


File Edit Selection View Go Run ···
Search
Mini Project.py
E: > Assignment > Mini Project.py > main
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python Debug Console
Ln 178, Col 31 Spaces: 4 UTF-8 Python 3.11.9 64-bit (Microsoft Store)
Type here to search
BSE...
ENG IN 12:22 08/10/2024

```python
def main():
            add_task(tasks)
        elif choice == '2':
            view_tasks(tasks)
        elif choice == '3':
            view_tasks(tasks, filter_by='completed')
        elif choice == '4':
            view_tasks(tasks, filter_by='pending')
        elif choice == '5':
            view_tasks(tasks, filter_by='due_soon')
        elif choice == '6':
            mark_task_completed(tasks)
        elif choice == '7':
            edit_task(tasks)
        elif choice == '8':
            delete_task(tasks)
        elif choice == '9':
            print("Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.")
```

```
4. View Pending Tasks
5. View Tasks Due Soon
6. Mark Task as Completed
7. Edit Task
8. Delete Task
9. Exit
Enter your choice: 9
Goodbye!
PS E:\Assignment> 
```