

## Mini Project :- Command Line To-Do List Manager

### Code:-

```
#this is the mini project i have choosen "command line To-do list manager"
import json
import datetime
import os

# File where tasks will be stored
TASKS_FILE = "tasks.json"

# Load tasks from file
def load_tasks():
    if os.path.exists(TASKS_FILE):
        with open(TASKS_FILE, "r") as f:
            return json.load(f)
    return []

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

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

    task = {
        "description": description,
        "due_date": due_date if due_date else None,
        "status": "Pending",
    }
    tasks.append(task)
    save_tasks(tasks)
    print("Task added successfully!")

# View tasks (all / filtered)
def view_tasks(tasks, filter_by=None):
    if not tasks:
        print("No tasks available.")
        return

    today = datetime.date.today()
```

```

for i, task in enumerate(tasks, start=1):
    if filter_by == "completed" and task["status"] != "Completed":
        continue
    elif filter_by == "pending" and task["status"] != "Pending":
        continue
    elif filter_by == "due_soon":
        if not task["due_date"]:
            continue
        due = datetime.datetime.strptime(task["due_date"], "%Y-%m-%d").date()
        if (due - today).days > 3:
            continue

    print(
        f"{i}. {task['description']} | Due: {task['due_date']} | Status: {task['status']}"
    )

# Mark task as completed
def mark_completed(tasks):
    view_tasks(tasks, filter_by="pending")
    task_no = int(input("Enter task number to mark as completed: "))
    if 0 < task_no <= len(tasks):
        tasks[task_no - 1]["status"] = "Completed"
        save_tasks(tasks)
        print("Task marked as completed!")
    else:
        print("Invalid task number.")

# Edit a task
def edit_task(tasks):
    view_tasks(tasks)
    task_no = int(input("Enter task number to edit: "))
    if 0 < task_no <= len(tasks):
        new_desc = input("Enter new description (leave blank to keep same): ")
        new_due = input("Enter new due date (YYYY-MM-DD) or leave blank: ")

        if new_desc:
            tasks[task_no - 1]["description"] = new_desc
        if new_due:
            tasks[task_no - 1]["due_date"] = new_due

        save_tasks(tasks)
        print("Task updated successfully!")
    else:

```

```

        print("Invalid task number.")

# Delete a task
def delete_task(tasks):
    view_tasks(tasks)
    task_no = int(input("Enter task number to delete: "))
    if 0 < task_no <= len(tasks):
        tasks.pop(task_no - 1)
        save_tasks(tasks)
        print("Task deleted successfully!")
    else:
        print("Invalid task number.")

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

    while True:
        print("\n===== To-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 (within 3 days)")
        print("6. Mark Task as Completed")
        print("7. Edit Task")
        print("8. Delete Task")
        print("9. Exit")

        choice = input("Enter your choice (1-9): ")

        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_completed(tasks)
        elif choice == "7":
            edit_task(tasks)

```

```

elif choice == "8":
    delete_task(tasks)
elif choice == "9":
    print("Exiting program. Goodbye!")
    break
else:
    print("Invalid choice, please try again.")

# Run the program
if __name__ == "__main__":
    main()

```

## Result :-

The screenshot shows a Visual Studio Code editor with a Python file named `todo-manager.py`. The code defines a `delete_task` function and a `main` function. The terminal output shows the program running and displaying a menu of options. The user has entered '1' to add a task, and the program has successfully added the task 'task one add' with a due date of '30-08-2025'.

```

87
88 # Delete a task
89 def delete_task(tasks):
90     view_tasks(tasks)
91     task_no = int(input("Enter task number to delete: "))
92     if 0 < task_no <= len(tasks):
93         tasks.pop(task_no - 1)
94         save_tasks(tasks)
95         print("Task deleted successfully!")

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

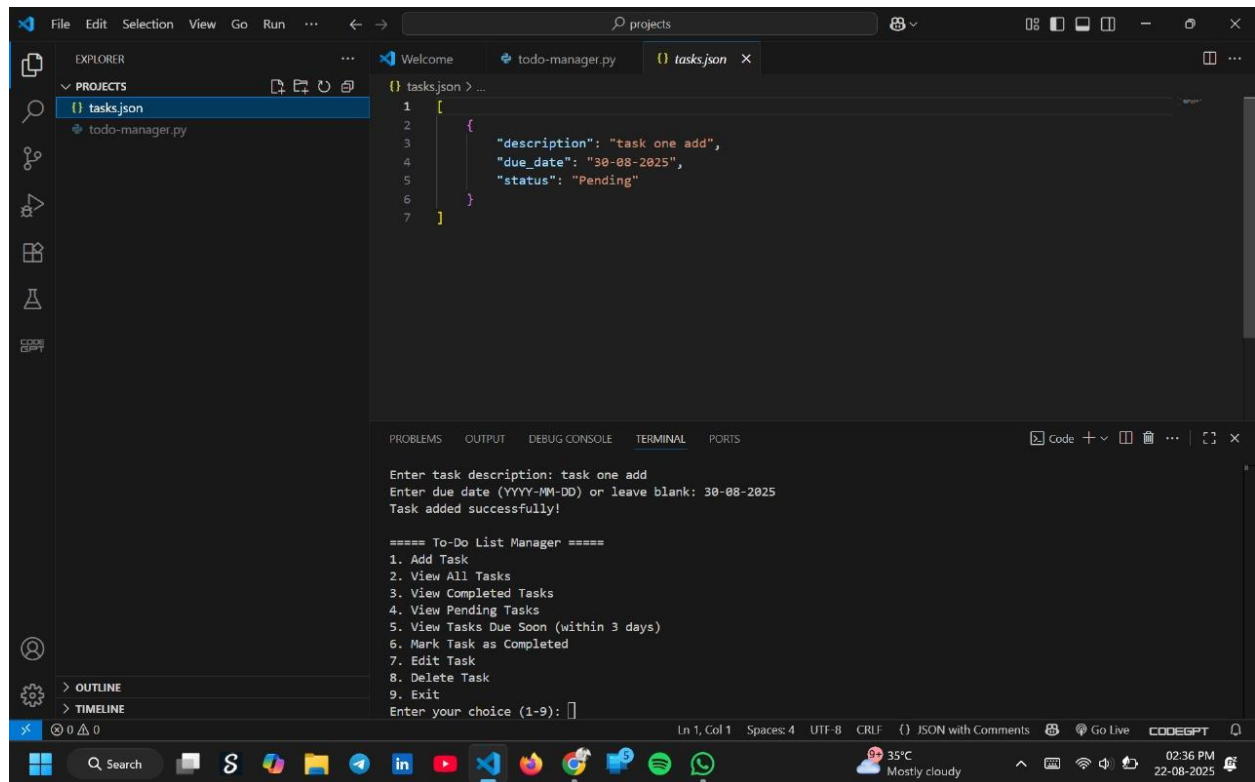
2. View All Tasks
3. View Completed Tasks
4. View Pending Tasks
5. View Tasks Due Soon (within 3 days)
6. Mark Task as Completed
7. Edit Task
8. Delete Task
9. Exit
Enter your choice (1-9): 1
Enter task description: task one add
Enter due date (YYYY-MM-DD) or leave blank: 30-08-2025
Task added successfully!

===== To-Do List Manager =====
1. Add Task
2. View All Tasks
3. View Completed Tasks
4. View Pending Tasks
5. View Tasks Due Soon (within 3 days)
6. Mark Task as Completed
7. Edit Task
8. Delete Task
9. Exit
Enter your choice (1-9):

```

Ln 83, Col 26 Spaces: 4 UTF-8 CRLF {} Python 3.13.2 Go Live CODEGPT

35°C Mostly cloudy 02:36 PM 22-08-2025



**Github repository link:**

<https://github.com/Vishalmohil63/vaultofcodes-internship>