Project Title: To-Do List Manager in Python

Description:

The To-Do List project is a simple “command-line application” that helps users organize their daily tasks. It allows users to “add new tasks”, “view pending/completed tasks”, “mark tasks as completed”, and “delete tasks” once they are done. The program ensures smooth interaction with clear instructions and real-time feedback.

Key Features:

- Add new tasks with a description
- View all tasks with status (✅ Pending / ✔ Completed)
- Mark tasks as completed
- Delete tasks by their number
- Interactive menu-driven interface.

Skills Demonstrated:

- **Object-Oriented Programming (OOP):** Classes (Task, ToDoList) with methods for modular design
- **Data Structures:** Lists for storing and managing tasks
- **Control Flow:** Loops and conditionals for menu handling
- **User Input Handling:** Taking and validating inputs from the user
- **Code Reusability:** Encapsulation of functionality into methods.

Applications:

This project demonstrates the basics of *task management software* and can be further extended into:

- GUI-based applications (Tkinter, PyQt)
- Web-based to-do apps (Flask/Django)
- Persistent task storage using files or databases.

Python code:

```
class Task:
    def __init__(self, description):
        self.description = description
        self.completed = False
    def mark_completed(self):
        self.completed = True
    def __str__(self):
        status = "✔ Completed" if self.completed else "❌ Pending"
        return f"{self.description} - {status}"
class ToDoList:
    def __init__(self):
        self.tasks = []
    def add_task(self, description):
        task = Task(description)
        self.tasks.append(task)
        print(f"Task '{description}' added successfully!")
    def view_tasks(self):
        if not self.tasks:
            print("No tasks available.")
        else:
            print("\nYour Tasks:")
            for index, task in enumerate(self.tasks, start=1):
```

```
            print(f"{index}. {task}")
    def mark_task_completed(self, index):
        if 0 < index <= len(self.tasks):
            self.tasks[index - 1].mark_completed()
            print("Task marked as completed!")
        else:
            print("Invalid task number!")
    def delete_task(self, index):
        if 0 < index <= len(self.tasks):
            removed = self.tasks.pop(index - 1)
            print(f"Task '{removed.description}' deleted!")
        else:
            print("Invalid task number!")
def main():
    todo = ToDoList()
    while True:
        print("\n===== TO-DO LIST MENU =====")
        print("1. Add Task")
        print("2. View Tasks")
        print("3. Mark Task as Completed")
        print("4. Delete Task")
        print("5. Exit")
        choice = input("Enter your choice (1-5): ")
```

```
        if choice == "1":
            desc = input("Enter task description: ")
            todo.add_task(desc)
        elif choice == "2":
            todo.view_tasks()
        elif choice == "3":
            todo.view_tasks()
            num = int(input("Enter task number to mark as completed: "))
            todo.mark_task_completed(num)
        elif choice == "4":
            todo.view_tasks()
            num = int(input("Enter task number to delete: "))
            todo.delete_task(num)
        elif choice == "5":
            print("Exiting... Have a productive day!")
            break
        else:
            print("Invalid choice! Please try again.")
if __name__ == "__main__":
    main()
```

Output:

===== TO-DO LIST MENU =====

1. Add Task

2. View Tasks

3. Mark Task as Completed

4. Delete Task

5. Exit