PROJECT ON PYTHON

Project: To-Do List Application

Overview

A simple To-Do List Application helps users manage daily tasks by allowing them to add, view, and remove tasks. This project is an excellent way to practice Python fundamentals, such as working with lists, functions, loops, and conditionals.

Tasks

- Create a Python script to manage a to-do list.
- Implement a function to add tasks to the list.
- Implement a function to display all tasks.
- Implement a function to remove tasks from the list by task name or index.
- Use loops to keep the program running until the user chooses to exit.
- Optional: Implement a function to mark tasks as "completed" or "pending."

Features Overview

- Add a Task
- Display Tasks
- Remove a Task
- Mark a Task as Completed
- Mark a Task as Pending
- Exit the Program

Add a Task

- When the user selects option 1, they are prompted to enter the name of the task. The task is added to the todo_list with an initial status of "pending".

Display Tasks

When the user selects option `2`, the program displays all tasks in the `todo_list`. Each task is shown with its index and its status (either "✓" for completed or "✗" for pending).

Remove a Task

When the user selects option 3, they are prompted to either:

- Enter the index (number) of the task to be removed.
- Enter the task name to be removed.

Mark a Task as Completed

When the user selects option 4, they are prompted to mark a task as completed by either:

- Task number (index).
- Task name.

Mark a Task as Pending

When the user selects option 5, they can mark a task as pending using either the task's number or name. This updates the task's status to "pending".

Exit the Program

Selecting option 6 exits the program. The loop is terminated, and a goodbye message is displayed.

Input Handling and Task Identification

The program handles user input by checking if the input is a digit (interpreted as an index) or a string (interpreted as the task name). For tasks, input is case-insensitive, allowing flexibility when removing or marking tasks.

Status Representation

- Completed tasks are displayed with a "✓" symbol.

Pending tasks are displayed with a "✗" symbol.

CODE:-

```
todo_list = []
continue_program = True

while continue_program:

    print("\n--- To-Do List Manager ---")
    print("1. Add a task")
    print("2. Display tasks")
    print("3. Remove a task")
    print("4. Mark a task as completed")
    print("5. Mark a task as pending")
    print("6. Exit")

    choice = input("Choose an option (1-6): ")

    if choice == '1':  # Add a task
```

```
        task_name = input("Enter the task name: ")
        task = {"task": task_name, "status": "pending"}
        todo_list.append(task)
        print(f'Added task: "{task_name}"')

    elif choice == '2':  # Display tasks
        if not todo_list:
            print("Your to-do list is empty.")
        else:
            print("\nYour To-Do List:")
            for index, task in enumerate(todo_list, 1):
                status = "✓" if task["status"] == "completed" else "✗"
                print(f"{index}. {task['task']} [{status}]")

    elif choice == '3':  # Remove a task
        identifier = input("Enter the task number or name to remove:
")
        if identifier.isdigit():  # Treat as index if input is a number
            index = int(identifier) - 1
            if 0 <= index < len(todo_list):
                removed_task = todo_list.pop(index)
```

```
            print(f'Removed task: "{removed_task["task"]}"')
        else:
            print("Invalid task number.")
    else:
        for task in todo_list:
            if task["task"].lower() == identifier.lower():
                todo_list.remove(task)
                print(f'Removed task: "{task["task"]}"')
                break
        else:
            print("Task not found.")

elif choice == '4':
    identifier = input("Enter the task number or name to mark as completed: ")
    if identifier.isdigit():
        index = int(identifier) - 1
        if 0 <= index < len(todo_list):
            todo_list[index]["status"] = "completed"
            print(f'Marked task "{todo_list[index]["task"]}" as completed.')
        else:
```

```
            print("Invalid task number.")
        else:  # Treat as task name
            for task in todo_list:
                if task["task"].lower() == identifier.lower():
                    task["status"] = "completed"
                    print(f'Marked task "{task["task"]}" as completed.')
                    break
            else:
                print("Task not found.")

    elif choice == '5':
        identifier = input("Enter the task number or name to mark as pending: ")
        if identifier.isdigit():
            index = int(identifier) - 1
            if 0 <= index < len(todo_list):
                todo_list[index]["status"] = "pending"
                print(f'Marked task "{todo_list[index]["task"]}" as pending.')
            else:
                print("Invalid task number.")
        else:
```

```
            for task in todo_list:
                if task["task"].lower() == identifier.lower():
                    task["status"] = "pending"
                    print(f'Marked task "{task["task"]}" as pending.')
                    break
            else:
                print("Task not found.")

    elif choice == '6':
        print("Exiting the To-Do List Manager. Goodbye!")
        break

    else:
        print("Invalid option, please choose again.")
```

OUTPUT:-

jupyter Untitled7 Last Checkpoint: 22 minutes ago

File Edit View Run Kernel Settings Help

Code

```
--- To-Do List Manager ---
1. Add a task
2. Display tasks
3. Remove a task
4. Mark a task as completed
5. Mark a task as pending
6. Exit
Choose an option (1-6):  1
Enter the task name:  weakup ealry morning
Added task: "weakup ealry morning"

--- To-Do List Manager ---
1. Add a task
2. Display tasks
3. Remove a task
4. Mark a task as completed
5. Mark a task as pending
6. Exit
Choose an option (1-6):  2

Your To-Do List:
1. weakup ealry morning [X]
```

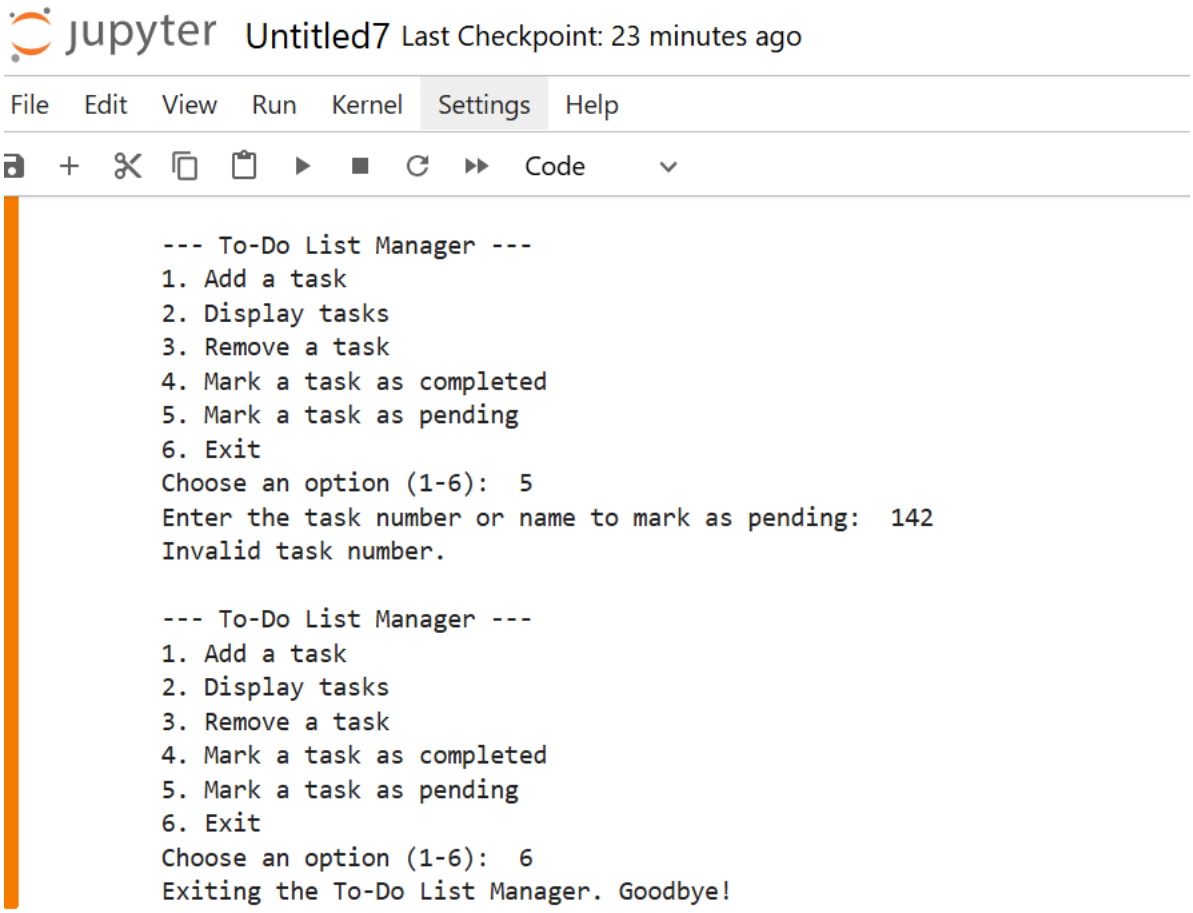
jupyter Untitled7 Last Checkpoint: 22 minutes ago

File Edit View Run Kernel Settings Help

Code

```
--- To-Do List Manager ---
1. Add a task
2. Display tasks
3. Remove a task
4. Mark a task as completed
5. Mark a task as pending
6. Exit
Choose an option (1-6):  3
Enter the task number or name to remove:  142
Invalid task number.

--- To-Do List Manager ---
1. Add a task
2. Display tasks
3. Remove a task
4. Mark a task as completed
5. Mark a task as pending
6. Exit
Choose an option (1-6):  4
Enter the task number or name to mark as completed:  142
Invalid task number.
```



Conclusion

The To-Do List Manager is a simple and interactive command-line tool that allows users to efficiently manage tasks. It supports adding tasks, displaying tasks with their status (pending or completed), removing tasks, and updating task statuses. With a user-friendly menu and flexible input options, this project is an excellent basic implementation of task management using Python. The application is straightforward, making it easy to track and organize tasks effectively.