|  |  |
| --- | --- |
| Project Title : | CLI Based ToDo List Application |
| Name : | **Aman Ali** |
| Roll No : | **195** |
| Class : | BSAI |
| Section : | **3C** |
| Submission Date : | August 28, 2025 |

TODO List Application

**Intoduction:**

This project is a Command Line TODO application built in python. It allow the user to manage the tasks by adding , viewing , marking as completed , unmarking , delete a specific task and clearing all tasks.

The program use a class based structure means OOP and tasks are store in a dictionary with the priority numbers. The output is display in the console/terminal.

The main thing about this app is it is **Dynamic** because firstly this show the tasks in the sorted form and this app is fully flexible user can perform action either by task name or the priority number.

**Features Implemented:**

* **Add Task** -> Add a new task with optional priority number
* **View Task** ->Display all tasks sorted format
* **Mark as Done**  -> Mark a task as completed by providing either task name or priority number
* **Unmark Task** -> Unmark a marked task by providing either task name or priority number
* **Delete Task**  -> Delete a task either by providing task name or priority number
* **Clear all Tasks**  -> Removes all tasks from the list
* **Exit App**  -> Quit the application safely with a simple dot animation

**Code Explanation ( How & Why ) :**

1. **Class TODO**

The application is structured in a TODO clas.

* Self.task -> store all tasks with priorities in unsorted way
* Self.marked\_tasl -> stores all the marked task as completed

1. **add\_tasks(self, task , priority=None )**

add a new task . it takes ask and priority as parameters . It prevent the task duplication by checking the task name is the name is already avalible in dictionary or not. And If priority is not provided in parameters then it get the maximum priority number in the **self.task** dictionary and then +1 in this and set to the given task.

For example is priority 1 & 2 is exists and new priority is not given then the new task get the priority number 3 automatically

1. **view\_tasks(self)**

this does not required any parameter. It merges the incompleted and completed tasks into a new dictionary and then return the merged dictionary and also sort the dictionary in an order.

1. **mark\_tasks(self, task= None , priority = None )**

Allow user to mark a task done by adding a tick mark in the value of the dictionary

Internally replace [ ] with [✔] for a visual indicator. Flexible because user can mark any task by either providing the task name or priority number.

1. **umark\_tasks(self, task= None , priority = None )**

Allow user to unmark a marked task .

Internally replace [✔] with [ ] for a visual indicator. Flexible because user can mark any task by either providing the task name or priority number.

1. **delete\_tasks(self, task= None , priority = None )**

Delete a task permanently. This is also flexible because user can delete a task either by providing task name and priority number.

It remove the task from the **self.task** and also make sure to delete the task from **self.marked\_tasks** dictionary if it exist in it.

1. **clear\_all\_tasks( self)**

It does not require any parameters it clear the both self.task and self.marked\_tasks dictionary by using python’s built-in function **.clear().**

1. **run() Function ( Menu System )**

Provides a menu driven interface in the terminal to the user.

It use a while loop to repeatedly show the options and gets the choice as input until the user exits with a basic animation. Handle user input safely with the checks and also solve the casing like lower and upper cases .

For example in this app the input abC , AbC , ABC , abc & ABC are same everything will be saved in lowercase and this will b convert automatically.

**Why this project and this approach was chosen :**

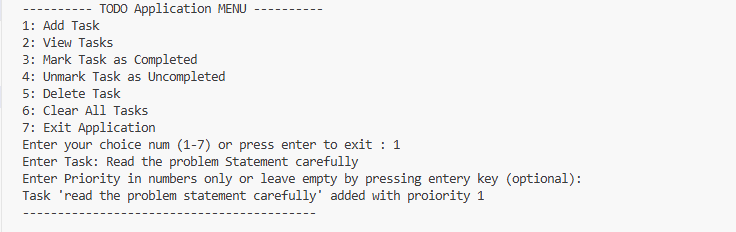
I chose the TODO list application over other projects because I really want to know how a todo app works .Also because a todo app is very important for any professional and student in their daily use.

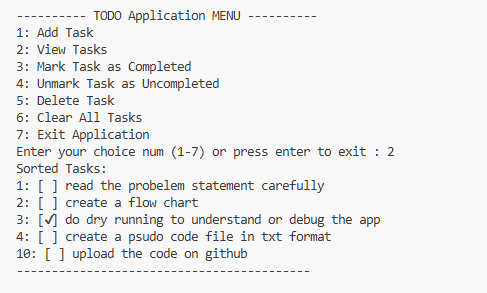
Secondly why I chose this approach . I chose this approach specially object oriented structure to make the code reusable and each operation separate into separate into methods.

I chose **Dictionary** over other data structures because this store data in key value pairs that are very helpful in display the task in sorted format.

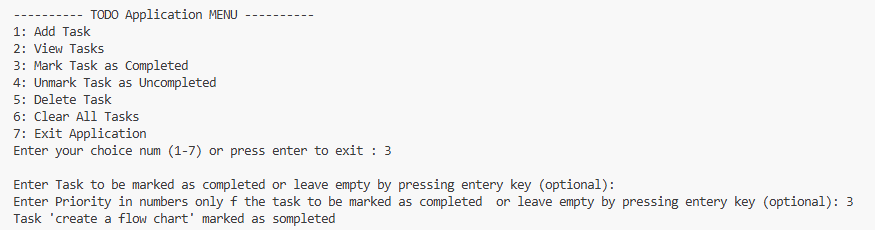
**Screenshots of output:**

Add Task:

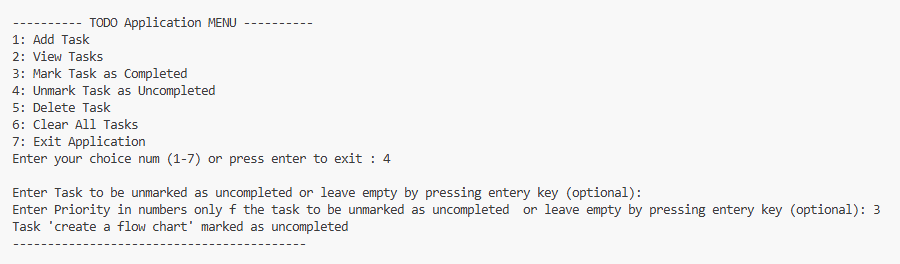


View Task:  


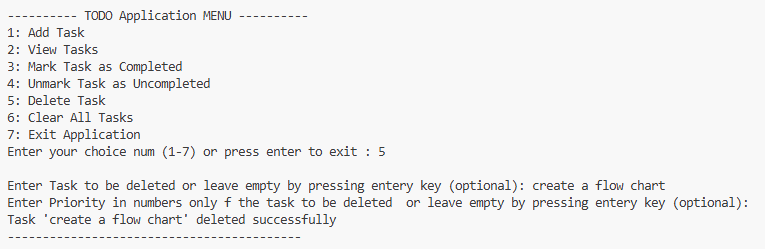
Mark as completed:



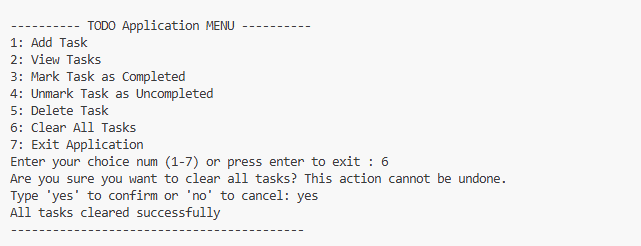
Unmark a marked task:



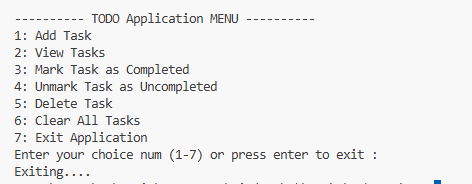
Delete a task:



Clear all tasks:



Exit :

****

**Conclusion:**

A simple CLI based TODO app that perform various of tasks like add , view in sorted form , add , remove , mark as done and unmark any task with the flexibity or priority number or task name. It use Object oriented programming and dictionary to store data.