A PROJECT ON

TO-DO LIST APP

DevTown||Prompt Engineering&Generative AI

Name : K.AKHILA

Department : CSE

Institution : Mahatma Gandhi University,Nalgonda

INTRODUCTION

- Overwhelming workload and constant distractions
- Tighter deadlines than ever
- Time is limited, but goals are not
- "You can do anything, but not everything." – David Allen

What is a To-Do List?

- A structured set of tasks to manage time, focus, and responsibilities.
- Includes:
  - Tasks & Subtasks
  - Deadlines
  - Priorities
  - Reminders

Benefits of a To-Do List

- Clears your mind and reduces anxiety
- Enhances time management
- Helps prioritize what matters
- Increases motivation and satisfaction
- Tracks progress effectively

Common Types of To-Do Lists

- Daily Tasks – Today's priorities
- Weekly Plans – Structure your week
- Project-Based – Break down large goals
- Kanban Boards – Visual task management

Essential Features of a Smart To-Do List

- Task categorization (e.g., Work, Personal)
- Priority tagging (High, Medium, Low)
- Due dates and deadlines
- Notifications & reminders
- Progress indicators

Top To-Do List Tools

- Digital Tools:
- Todoist, Microsoft To Do, Notion, Trello, Google Keep
- Physical Tools:
- Journals, Sticky Notes, Paper Planners

How to Create an Effective To-Do List

- 1. Brain dump all tasks
- 2. Categorize them
- 3. Assign priorities
- 4. Add deadlines
- 5. Review daily
- 6. Check off completed items

Sample Layout

- Task | Priority | Due Time | Status
- Study for TCSS exam | High | 4 PM | Pending
- Complete assignment | Medium | 6 PM | In Progress
- Exercise | Low | 7 PM | Done

IMPLEMENTATION
python code(part 1)

```
class ToDoList:
    def __init__(self):
        self.tasks = []

    def add_task(self, task):
        self.tasks.append({"task": task, "completed": False})
        print(f"Task '{task}' added to the list.")
```

Python code(part 2)

```
print(f"Task '{self.tasks[task_number - 1]['task']}' marked as completed.")
        else:
            print("Invalid task number.")
    def remove_task(self, task_number):
        if 0 < task_number <= len(self.tasks):
            removed_task = self.tasks.pop(task_number - 1)
            print(f"Removed: {removed_task['task']}")
        else:
            print("Invalid task number.")
```

Python code(part 3)

- print(f"Task '{self.tasks[task_number - 1]['task']}' marked as completed.")
- else:
- print("Invalid task number.")

- def remove_task(self, task_number):
- if 0 < task_number <= len(self.tasks):
- removed_task = self.tasks.pop(task_number - 1)
- print(f"Removed: {removed_task['task']}")
- else:
- print("Invalid task number.")

Main Function & Main

```
def main():
    to_do_list = ToDoList()
    while True:
        print("1. Add Task\n2. View\n3. Complete\n4. Remove\n5. Exit")
        choice = input("Choice: ")
        if choice == "1":
            to_do_list.add_task(input("Enter task: "))
        elif choice == "2":
            to_do_list.view_tasks()
        elif choice == "3":
            to_do_list.mark_complete(int(input("Task #: ")))
        elif choice == "4":
```

- to_do_list.remove_task(int(input("Task #: ")))
- elif choice == "5":
- break;

OUTPUT:

- To-Do List Options:
    1. Add Task
    2. View Tasks
    3. Mark Task as Complete
    4. Remove Task
    5. Exit

Enter task : Study python

Tips for Success

- Avoid overloading your list
- Break large tasks into smaller steps
- Be realistic with time estimates
- Review and revise regularly
- Celebrate small wins

Final Thoughts

- "Productivity is not about doing more. It’ s about doing what matters."
- Start your to-do list today and take control of your time and goals!