MINI PROJECT

Develop a basic to-do list application using functions and data structures

Objective: Develop a simple to-do list application using Python with an emphasis on functions and data structures.

Key Components:

1. Functions: implementing various functions to handle different aspects of the to-do list application. Functions are modular blocks of code that perform specific tasks, making our code more organized and easier to understand.
   Function to add a task
   Function to delete a task
   Function to display the list of tasks
   Function to mark a task as complete
2. Data Structures: Utilizing appropriate data structures to store and manage the to-do list. A common choice would be a list or a dictionary, but we can explore other options based on our creativity , requirements and understanding.

Code:

```
# todo_list.py
from datetime import datetime
class Task:
    def __init__(self, description, priority, due_date):
        self.description = description
        self.priority = priority
        self.due_date = due_date
        self.completed = False
        self.start_date = datetime.now().strftime("%d-%m-%Y")
        self.completed_date = None
    def mark_as_completed(self):
        self.completed = True
        self.completed_date = datetime.now().strftime("%d-%m-%Y")
    def __str__(self):
        status = "Completed" if self.completed else "Not Completed"
        if self.completed:
            return f"Description: {self.description}\n  Priority:
{self.priority}\n  Due Date: {self.due_date}\n  Start Date:
{self.start_date}\n  Completed Date: {self.completed_date}\n  Status:
{status}\n"
        else:
            return f"Description: {self.description}\n  Priority:
{self.priority}\n  Due Date: {self.due_date}\n  Start Date:
{self.start_date}\n  Status: {status}\n"
class ToDoList:
    def __init__(self):
        self.tasks = []
    def add_task(self, task):
        self.tasks.append(task)
    def display_tasks(self):
        for i, task in enumerate(self.tasks, start=1):
            print(f"{i}. {task}")
    def delete_task(self, index):
        try:
            del self.tasks[index - 1]
            print(f"Task {index} deleted")
        except IndexError:
            print("Invalid index")
    def mark_task_as_completed(self, index):
```

```
        try:
            self.tasks[index - 1].mark_as_completed()
            print(f"Task {index} marked as completed")
            self.display_tasks()
        except IndexError:
            print("Invalid index")
def main():
    todo_list = ToDoList()
    while True:
        print("\nTo-Do List App")
        print("1. Add task")
        print("2. Display tasks")
        print("3. Mark task as completed")
        print("4. Delete task")
        print("5. Exit")
        choice = input("Choose an option: ")
        if choice == "1":
            description = input("Enter task description: ")
            priority = input("Enter task priority (High/Medium/Low): ")
            due_date = input("Enter task due date (DD-MM-YYYY): ")
            task = Task(description, priority, due_date)
            todo_list.add_task(task)
        elif choice == "2":
            todo_list.display_tasks()
        elif choice == "3":
            index = int(input("Enter the index of the task to mark as
completed: "))
            todo_list.mark_task_as_completed(index)
        elif choice == "4":
            index = int(input("Enter the index of the task to delete: "))
            todo_list.delete_task(index)
        elif choice == "5":
            print("Thankyou!")
            break
        else:
            print("Invalid option")
if __name__== "__main__":
    main()
```

Functions:

1. **__init__:** Initializes an object of the Task or ToDoList class.
2. **mark_as_completed**: Marks a task as completed and updates the completion date.
3. **__str__:** Returns a string representation of a task.
4. **add_task:** Adds a task to the to-do list.
5. **display_tasks:** Displays all tasks in the to-do list.
6. **delete_task:** Deletes a task from the to-do list.
7. **mark_task_as_completed:** Marks a task as completed and updates the to-do list.
8. **main:** The main function that runs the to-do list app.

Data Structures:

Class: Task and ToDoList are classes that define the structure and behavior of tasks and to-do lists, respectively.

List: The tasks attribute in the ToDoList class is a list that stores all tasks in the to-do list.

Object: Each task is an object of the Task class, which has attributes like description, priority, due_date, completed, start_date, and completed_date.

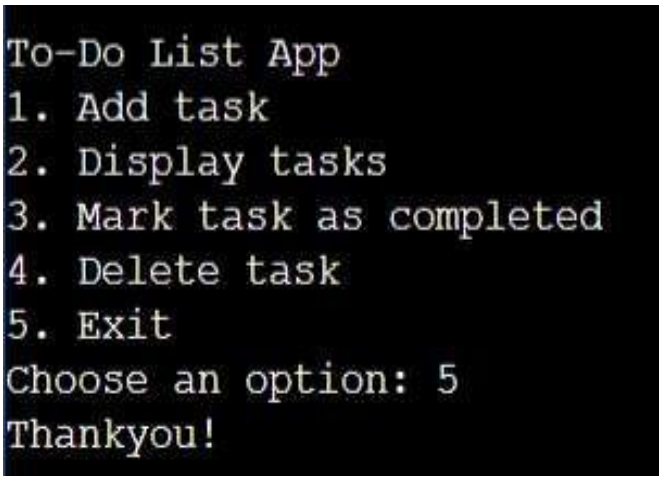
Output:

```
To-Do List App
1. Add task
2. Display tasks
3. Mark task as completed
4. Delete task
5. Exit
Choose an option: 1
Enter task description: Placement
Enter task priority (High/Medium/Low): h
Enter task due date (DD-MM-YYYY): 01-07-2024

To-Do List App
1. Add task
2. Display tasks
3. Mark task as completed
4. Delete task
5. Exit
Choose an option: 2
1. Description: Placement
   Priority: h
   Due Date: 01-07-2024
   Start Date: 30-06-2024
   Status: Not Completed
```

```
To-Do List App
1. Add task
2. Display tasks
3. Mark task as completed
4. Delete task
5. Exit
Choose an option: 3
Enter the index of the task to mark as completed: 1
Task 1 marked as completed
1. Description: Placement
   Priority: h
   Due Date: 01-07-2024
   Start Date: 30-06-2024
   Completed Date: 30-06-2024
   Status: Completed


To-Do List App
1. Add task
2. Display tasks
3. Mark task as completed
4. Delete task
5. Exit
Choose an option: 4
Enter the index of the task to delete: 1
Task 1 deleted
```



Done by,

Manmohan Kumar