MINI PROJECT
VAULTOFCODES

WEEK-3:

⇨ Develop a basic to-do-list application using functions and data structure…

Source code:

```
def add_task(todo_list, task):
    todo_list.append(task)
    print(f'Task "{task}" added successfully.')
def delete_task(todo_list, task):
    if task in todo_list:
        todo_list.remove(task)
        print(f'Task "{task}" deleted successfully.')
    else:
        print(f'Task "{task}" not found in the to-do list.')
def display_tasks(todo_list):
    if not todo_list:
        print('To-do list is empty.')
    else:
        print('To-do list:')
        for index, task in enumerate(todo_list, start=1):
            print(f'{index}. {task}')
def mark_complete(todo_list, task):
    if task in todo_list:
        print(f'Task "{task}" marked as complete.')
    else:
        print(f'Task "{task}" not found in the to-do list.')
def main():
    todo_list = []
```

```
    while True:
        print('\nTo-Do List Application:')
        print('1. Add Task')
        print('2. Delete Task')
        print('3. Display Tasks')
        print('4. Mark Task as Complete')
        print('5. Exit')

        choice = input('Enter your choice (1-5): ')

        if choice == '1':
            task = input('Enter the task to add: ')
            add_task(todo_list, task)
        elif choice == '2':
            task = input('Enter the task to delete: ')
            delete_task(todo_list, task)
        elif choice == '3':
            display_tasks(todo_list)
        elif choice == '4':
            task = input('Enter the task to mark as complete: ')
            mark_complete(todo_list, task)
        elif choice == '5':
            print('Exiting the application. Goodbye!')
            break
        else:
            print('Invalid choice. Please enter a number between 1 and 5.')

if __name__ == "__main__":
    main()
```

OUTPUT:

- **Adding a task into the To-DO-List :-**

```
To-Do List Application:
1. Add Task
2. Delete Task
3. Display Tasks
4. Mark Task as Complete
5. Exit
Enter your choice (1-5): 1
Enter the task to add: eating
Task "eating" added successfully.
```

- **Displaying the LIST :-**

```
To-Do List Application:
1. Add Task
2. Delete Task
3. Display Tasks
4. Mark Task as Complete
5. Exit
Enter your choice (1-5): 3
To-do list:
1. eating

To-Do List Application:
1. Add Task
2. Delete Task
3. Display Tasks
4. Mark Task as Complete
5. Exit
Enter your choice (1-5):
```

- **Mark the TASK as completed :-**

```
To-Do List Application:
1. Add Task
2. Delete Task
3. Display Tasks
4. Mark Task as Complete
5. Exit
Enter your choice (1-5): 4
Enter the task to mark as complete: eating
Task "eating" marked as complete.

To-Do List Application:
1. Add Task
2. Delete Task
3. Display Tasks
4. Mark Task as Complete
5. Exit
Enter your choice (1-5):
```

- **Adding new TASK to the list :-**

```
To-Do List Application:
1. Add Task
2. Delete Task
3. Display Tasks
4. Mark Task as Complete
5. Exit
Enter your choice (1-5): 1
Enter the task to add: running
Task "running" added successfully.

To-Do List Application:
1. Add Task
2. Delete Task
3. Display Tasks
4. Mark Task as Complete
5. Exit
Enter your choice (1-5):
```

- **Displaying all the TASK wich are present in the LIST :-**

```
To-Do List Application:
1. Add Task
2. Delete Task
3. Display Tasks
4. Mark Task as Complete
5. Exit
Enter your choice (1-5): 3
To-do list:
1. eating
2. running

To-Do List Application:
1. Add Task
2. Delete Task
3. Display Tasks
4. Mark Task as Complete
5. Exit
Enter your choice (1-5):
```

- **Deleting a TASK from the LIST :-**

```
To-Do List Application:
1. Add Task
2. Delete Task
3. Display Tasks
4. Mark Task as Complete
5. Exit
Enter your choice (1-5): 3
To-do list:
1. running

To-Do List Application:
1. Add Task
2. Delete Task
3. Display Tasks
4. Mark Task as Complete
5. Exit
Enter your choice (1-5):
```

- **After completing all TASKs we choose the 5^{th} option to EXIT from the LIST.**