### **Project 3: To-Do List Application**

#### **Objective:**

Create a command-line to-do list application where users can add, view, edit, and delete tasks. Tasks should be saved to a file so that they persist even after the program is closed.

### **Instructions**

#### **Step 1: Set Up the Environment**

1. Create a new Python file called todo\_list.py.
2. Open the file in your preferred IDE or text editor.

#### **Step 2: Create a Function to Load and Save Data**

Create a function called load\_tasks() that reads the tasks from a text file called tasks.txt and returns them as a list.  
python  
Copy code  
def load\_tasks():

tasks = []

try:

with open("tasks.txt", "r") as file:

tasks = file.readlines()

except FileNotFoundError:

pass

return [task.strip() for task in tasks]

Create a function called save\_tasks(tasks) that writes the tasks to tasks.txt.  
python  
Copy code  
def save\_tasks(tasks):

with open("tasks.txt", "w") as file:

for task in tasks:

file.write(task + "\n")

#### **Step 3: Create Functions for Adding, Editing, and Deleting Tasks**

1. **Adding a Task**
   * Create a function called add\_task(tasks) that prompts the user to enter a new task and appends it to the tasks list.

python  
Copy code  
def add\_task(tasks):

new\_task = input("Enter a new task: ")

tasks.append(new\_task)

print(f"Task '{new\_task}' added.")

1. **Editing a Task**
   * Create a function called edit\_task(tasks) that displays all tasks with their index, allows the user to select a task to edit, and updates it.

python  
Copy code  
def edit\_task(tasks):

display\_tasks(tasks)

index = int(input("Enter the task number to edit: ")) - 1

if 0 <= index < len(tasks):

new\_content = input("Enter new content for the task: ")

tasks[index] = new\_content

print("Task updated.")

else:

print("Invalid task number.")

1. **Deleting a Task**
   * Create a function called delete\_task(tasks) that displays all tasks, allows the user to select a task to delete, and removes it.

python  
Copy code  
def delete\_task(tasks):

display\_tasks(tasks)

index = int(input("Enter the task number to delete: ")) - 1

if 0 <= index < len(tasks):

print(f"Task '{tasks.pop(index)}' deleted.")

else:

print("Invalid task number.")

#### **Step 4: Create a Function to Display Tasks**

Create a function called display\_tasks(tasks) that prints the list of tasks.  
python  
Copy code  
def display\_tasks(tasks):

print("\n--- To-Do List ---")

if tasks:

for i, task in enumerate(tasks, 1):

print(f"{i}. {task}")

else:

print("No tasks available.")

#### **Step 5: Create the Main Function**

1. Create a main() function to serve as the program's entry point.
2. Load the tasks using load\_tasks() and display a menu with options:
   * View Tasks
   * Add Task
   * Edit Task
   * Delete Task
   * Save & Exit

Based on the user's input, call the appropriate function and loop until the user chooses to exit.  
python  
Copy code  
def main():

tasks = load\_tasks()

while True:

print("\n--- To-Do List Menu ---")

print("1. View Tasks")

print("2. Add Task")

print("3. Edit Task")

print("4. Delete Task")

print("5. Save & Exit")

choice = input("Enter your choice: ")

if choice == "1":

display\_tasks(tasks)

elif choice == "2":

add\_task(tasks)

elif choice == "3":

edit\_task(tasks)

elif choice == "4":

delete\_task(tasks)

elif choice == "5":

save\_tasks(tasks)

print("Tasks saved. Exiting the program.")

break

else:

print("Invalid choice. Please select an option from the menu.")

#### **Step 6: Run the Program**

Add a line at the end to ensure main() runs when the script is executed:  
python  
Copy code  
if \_\_name\_\_ == "\_\_main\_\_":

main()