**Project Description**

The To-Do List project is a console-based application designed to help users manage their tasks efficiently. Users can add tasks, view pending and completed tasks, edit task details, delete tasks, and mark tasks as completed. The code ensures that tasks are saved and can be retrieved even after the program is closed.

**Features Implemented**

1. **Add Task:**

* Users can add a new task with a unique name, description, and deadline.
* The task is marked as 'Pending' by default.

1. **View Tasks:**

* Users can view all pending and completed tasks.
* Tasks are displayed with their name, description, and deadline.

1. **Edit Task:**

* Users can edit the description and deadline of an existing task by providing the task name.

1. **Delete Task:**

* Users can delete a task by providing its name.

1. **Mark Task as Completed:**

* Users can mark a task as completed by providing its name.
* The task's status is updated to 'Completed'.

1. **Storage:**

* Task data is stored in a file (tasks.txt) and is retrieved upon starting the application.
* The file is updated with changes made to the tasks.

**How to Run the Project**

1. **Ensure Python is Installed:**

* Make sure you have Python installed on your computer.

1. **Run the Script:**

* Open console.
* Navigate to the directory where Source code.py is saved.
* Run the script
* Ensure the tasks.txt file is in the same directory as your Source code.py script.

Running the script will create the tasks.txt file if it doesn't already exist.

**Source Code**

import os

TASKS\_FILE = 'tasks.txt'

def read\_tasks():

tasks = []

if os.path.exists(TASKS\_FILE):

with open(TASKS\_FILE, 'r') as file:

for line in file:

name, description, deadline, status = line.strip().split(',')

tasks.append({

'name': name,

'description': description,

'deadline': deadline,

'status': status

})

return tasks

def write\_tasks(tasks):

with open(TASKS\_FILE, 'w') as file:

for task in tasks:

file.write(f"{task['name']},{task['description']},{task['deadline']},{task['status']}\n")

def add\_task(tasks):

name = input("Enter task name: ")

description = input("Enter task description: ")

deadline = input("Enter deadline (YYYY-MM-DD): ")

tasks.append({

'name': name,

'description': description,

'deadline': deadline,

'status': 'Pending'

})

write\_tasks(tasks)

print("Task added successfully!")

def view\_tasks(tasks):

print("To-Do List:")

print("[Pending]")

for task in tasks:

if task['status'] == 'Pending':

print(f"Name: {task['name']}, Description: {task['description']}, Deadline: {task['deadline']}")

print("\n[Completed]")

for task in tasks:

if task['status'] == 'Completed':

print(f"Name: {task['name']}, Description: {task['description']}, Deadline: {task['deadline']}")

def edit\_task(tasks):

name = input("Enter task name to edit: ")

for task in tasks:

if task['name'] == name:

task['description'] = input("Enter new task description: ")

task['deadline'] = input("Enter new deadline (YYYY-MM-DD): ")

write\_tasks(tasks)

print("Task edited successfully!")

return

print("Task not found!")

def delete\_task(tasks):

name = input("Enter task name to delete: ")

tasks = [task for task in tasks if task['name'] != name]

write\_tasks(tasks)

print("Task deleted successfully!")

def mark\_task\_completed(tasks):

name = input("Enter task name to mark as completed: ")

for task in tasks:

if task['name'] == name:

task['status'] = 'Completed'

write\_tasks(tasks)

print("Task marked as completed!")

return

print("Task not found!")

def main():

tasks = read\_tasks()

while True:

print("Welcome to To-Do List Manager!")

print("1. Add Task")

print("2. View Tasks")

print("3. Edit Task")

print("4. Delete Task")

print("5. Mark Task as Completed")

print("6. Exit")

choice = input("Enter your choice: ")

if choice == '1':

add\_task(tasks)

elif choice == '2':

view\_tasks(tasks)

elif choice == '3':

edit\_task(tasks)

elif choice == '4':

delete\_task(tasks)

elif choice == '5':

mark\_task\_completed(tasks)

elif choice == '6':

break

else:

print("Invalid choice! Please try again.")

if \_\_name\_\_ == "\_\_main\_\_":

main()