\*) to-do list program

import tkinter as tk

from tkinter import messagebox

class Task:

def \_\_init\_\_(self, description, priority, due\_date):

self.description = description

self.priority = priority

self.due\_date = due\_date

self.completed = False

class ToDoListApp:

def \_\_init\_\_(self, root):

self.root = root

self.root.title("To-Do List App")

self.tasks = []

# Create labels

tk.Label(root, text="Task Description").grid(row=0, column=0)

tk.Label(root, text="Priority (High/Medium/Low)").grid(row=1, column=0)

tk.Label(root, text="Due Date").grid(row=2, column=0)

# Entry fields

self.description\_entry = tk.Entry(root)

self.priority\_entry = tk.Entry(root)

self.due\_date\_entry = tk.Entry(root)

self.description\_entry.grid(row=0, column=1)

self.priority\_entry.grid(row=1, column=1)

self.due\_date\_entry.grid(row=2, column=1)

# Buttons

tk.Button(root, text="Add Task", command=self.add\_task).grid(row=3, column=0)

tk.Button(root, text="Remove Task", command=self.remove\_task).grid(row=3, column=1)

tk.Button(root, text="Mark as Completed", command=self.mark\_completed).grid(row=3, column=2)

# Task Listbox

self.task\_listbox = tk.Listbox(root, width=50, height=15)

self.task\_listbox.grid(row=4, columnspan=3)

# Load tasks from file

self.load\_tasks()

self.update\_listbox()

def add\_task(self):

description = self.description\_entry.get()

priority = self.priority\_entry.get()

due\_date = self.due\_date\_entry.get()

if description and priority and due\_date:

task = Task(description, priority, due\_date)

self.tasks.append(task)

self.update\_listbox()

self.save\_tasks()

self.clear\_entries()

else:

messagebox.showwarning("Warning", "Please fill in all fields.")

def remove\_task(self):

selected\_task\_index = self.task\_listbox.curselection()

if selected\_task\_index:

task\_index = selected\_task\_index[0]

del self.tasks[task\_index]

self.update\_listbox()

self.save\_tasks()

def mark\_completed(self):

selected\_task\_index = self.task\_listbox.curselection()

if selected\_task\_index:

task\_index = selected\_task\_index[0]

self.tasks[task\_index].completed = True

self.update\_listbox()

self.save\_tasks()

def update\_listbox(self):

self.task\_listbox.delete(0, tk.END)

for task in self.tasks:

status = "[x]" if task.completed else "[ ]"

self.task\_listbox.insert(tk.END, f"{status} {task.description} (Priority: {task.priority}, Due: {task.due\_date})")

def clear\_entries(self):

self.description\_entry.delete(0, tk.END)

self.priority\_entry.delete(0, tk.END)

self.due\_date\_entry.delete(0, tk.END)

def save\_tasks(self):

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

for task in self.tasks:

file.write(f"{task.description},{task.priority},{task.due\_date},{task.completed}\n")

def load\_tasks(self):

try:

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

lines = file.readlines()

for line in lines:

data = line.strip().split(",")

description, priority, due\_date, completed = data

task = Task(description, priority, due\_date)

if completed == "True":

task.completed = True

self.tasks.append(task)

except FileNotFoundError:

pass

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

root = tk.Tk()

app = ToDoListApp(root)

root.mainloop()

0utput:

