**TASK 1(COMMAND LINE INTERFACE)**

print ("~~~~COMMAND LINE INTERFACE~~~~")

import json

from datetime import datetime

# File to store tasks

FILE\_NAME = "tasks.json"

def load\_tasks():

try:

with open(FILE\_NAME, 'r') as file:

return json.load(file)

except FileNotFoundError:

return []

def save\_tasks(tasks):

with open(FILE\_NAME, 'w') as file:

json.dump(tasks, file, indent=4)

def add\_task(tasks):

title = input("Enter task title: ")

description = input("Enter task description: ")

due\_date = input("Enter due date (YYYY-MM-DD) or press enter to skip: ")

task = {

"title": title,

"description": description,

"due\_date": due\_date if due\_date else None,

"completed": False

}

tasks.append(task)

save\_tasks(tasks)

print("Task added!")

def view\_tasks(tasks):

if not tasks:

print("No tasks to show.")

return

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

status = "Completed" if task['completed'] else "Incomplete"

due = f"Due: {task['due\_date']}" if task['due\_date'] else "No due date"

print(f"{i}. {task['title']} - {status} ({due})")

def mark\_complete(tasks):

view\_tasks(tasks)

task\_num = int(input("Enter task number to mark as complete: "))

tasks[task\_num - 1]['completed'] = True

save\_tasks(tasks)

print("Task marked as complete!")

def delete\_task(tasks):

view\_tasks(tasks)

task\_num = int(input("Enter task number to delete: "))

tasks.pop(task\_num - 1)

save\_tasks(tasks)

print("Task deleted!")

def main():

tasks = load\_tasks()

while True:

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

print("1. View Tasks")

print("2. Add Task")

print("3. Mark Task as Complete")

print("4. Delete Task")

print("5. Exit")

choice = input("Enter your choice: ")

if choice == "1":

view\_tasks(tasks)

elif choice == "2":

add\_task(tasks)

elif choice == "3":

mark\_complete(tasks)

elif choice == "4":

delete\_task(tasks)

elif choice == "5":

print("Goodbye!")

break

else:

print("Invalid option. Please try again.")

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

main()

**TASK 1(GRAPHICAL USER INTERFACE)**

GRAPHICAL USER INTERFACE

import tkinter as tk

from tkinter import messagebox

import json

FILE\_NAME = "tasks.json"

def load\_tasks():

try:

with open(FILE\_NAME, 'r') as file:

return json.load(file)

except FileNotFoundError:

return []

def save\_tasks(tasks):

with open(FILE\_NAME, 'w') as file:

json.dump(tasks, file, indent=4)

def add\_task():

task\_title = entry\_task.get()

if task\_title:

tasks.append({"title": task\_title, "completed": False})

save\_tasks(tasks)

update\_task\_listbox()

entry\_task.delete(0, tk.END)

else:

messagebox.showwarning("Input Error", "Please enter a task title.")

def update\_task\_listbox():

listbox\_tasks.delete(0, tk.END)

for task in tasks:

status = "✔" if task['completed'] else "✘"

listbox\_tasks.insert(tk.END, f"{task['title']} [{status}]")

def mark\_complete():

try:

index = listbox\_tasks.curselection()[0]

tasks[index]['completed'] = True

save\_tasks(tasks)

update\_task\_listbox()

except IndexError:

messagebox.showwarning("Selection Error", "Please select a task.")

def delete\_task():

try:

index = listbox\_tasks.curselection()[0]

tasks.pop(index)

save\_tasks(tasks)

update\_task\_listbox()

except IndexError:

messagebox.showwarning("Selection Error", "Please select a task.")

# Initialize main window

root = tk.Tk()

root.title("To-Do List")

tasks = load\_tasks()

# Task entry

entry\_task = tk.Entry(root, width=40)

entry\_task.pack(pady=10)

# Add task button

button\_add = tk.Button(root, text="Add Task", width=42, command=add\_task)

button\_add.pack(pady=5)

# Task listbox

listbox\_tasks = tk.Listbox(root, height=10, width=42)

listbox\_tasks.pack(pady=10)

update\_task\_listbox()

# Buttons for task actions

button\_complete = tk.Button(root, text="Mark Complete", width=42, command=mark\_complete)

button\_complete.pack(pady=5)

button\_delete = tk.Button(root, text="Delete Task", width=42, command=delete\_task)

button\_delete.pack(pady=5)

# Start the application

root.mainloop()

**TASK 2**

print ("~~~~Simple Calculator~~~~")

num1 = float(input("enter first number here: "))

num2 = float(input("enter second number here: "))

print ("press 1 for addition \npress 2 for subtraction \npress 3 for multiplication \npress 4 for division")

choice = int(input("enter your choice from 1-4: "))

if choice == 1:

print ("The addition of given two numbers is", num1 + num2)

elif choice == 2:

print ("The subtraction of given two numbers is:", num1 - num2)

elif choice == 3:

print ("The multiplication of given two numbers is:", num1 \* num2)

elif choice == 4:

print ("The division of given two numbers is:", num1 / num2)

else:

print ("Invalid Input")

**TASK 3**

print ("~~~~Password Generator~~~~")

import string

import random

if \_\_name\_\_ == '\_\_main\_\_':

s1 = string.ascii\_uppercase

s2 = string.ascii\_lowercase

s3 = string.digits

s4 = string.punctuation

passwordlength = int(input("Enter password length\n"))

s = []

s.extend(list(s1))

s.extend(list(s2))

s.extend(list(s3))

s.extend(list(s4))

print(s)

random.shuffle(s)

print("Your password is: ")

print("".join(s[0:passwordlength]))

**TASK 4**

import random

while True:

print ("~~~~~~~WELCOME TO ROCK PAPER SCISSORS GAME~~~~~~~")

user\_score = 0

comp\_score = 0

ties = 0

name = input("Enter your name here: ")

print ("""

Winning Rules:

1. Paper vs Rock ---> Paper

2. Rock vs Scissors ---> Rock

3. Scissors vs Paper ---> Scissors""")

print()

print("""Choices are:

1. Rock

2. Paper

3. Scissors""")

choice = int(input("enter your choice from 1-3: "))

print()

while choice > 3 or choice < 1:

choice = int(input("enter valid input"))

if choice == 1:

user\_choice = "Rock"

elif choice == 2:

user\_choice = "Paper"

else:

user\_choice = "Scissors"

print ("The user's choice is",user\_choice)

print ("Now it is Computer's turn")

computer = random.randint(1,3)

if computer == 1:

comp\_choice = "Rock"

elif computer == 2:

comp\_choice = "Paper"

else:

comp\_choice = "Scissors"

print ("The computer's choice is",comp\_choice)

if ((user\_choice == "Paper" and comp\_choice == "Rock") or(user\_choice == "Rock" and comp\_choice == "Paper")):

print ("Paper wins")

result = "Paper"

elif ((user\_choice == "Scissors" and comp\_choice == "Rock") or(user\_choice == "Rock" and comp\_choice == "Scissors")):

print ("Rock wins")

result = "Rock"

elif (user\_choice == comp\_choice):

print ("It is a tie")

result = "Tie"

else:

print ("Scissors wins")

result = "Scissors"

if result == "Tie":

ties += 1

elif result == user\_choice:

print("user wins")

user\_score += 1

else:

print("computer wins")

comp\_score += 1

print ("Scores are")

print (name,"'s score is",user\_score)

print ("computer's score is",comp\_score)

print ("ties are",ties)

repeat = input("Do you want to play again? ")

if repeat == "No" or repeat == "No":

break

print ("Game over!")

print ("Thanks for playing")

**TASK 5**

class ContactBook:

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

self.contacts = {}

def add\_contact(self, name, phone\_number, email):

self.contacts[name] = {

"phone\_number": phone\_number,

"email": email

}

print(f"Contact {name} added successfully!")

def view\_contacts(self):

for name, details in self.contacts.items():

print(f"Name: {name}")

print(f"Phone Number: {details['phone\_number']}")

print(f"Email: {details['email']}")

print("------------------------")

def search\_contact(self, name):

if name in self.contacts:

print(f"Name: {name}")

print(f"Phone Number: {self.contacts[name]['phone\_number']}")

print(f"Email: {self.contacts[name]['email']}")

else:

print(f"Contact {name} not found!")

def update\_contact(self, name, phone\_number=None, email=None):

if name in self.contacts:

if phone\_number:

self.contacts[name]["phone\_number"] = phone\_number

if email:

self.contacts[name]["email"] = email

print(f"Contact {name} updated successfully!")

else:

print(f"Contact {name} not found!")

def delete\_contact(self, name):

if name in self.contacts:

del self.contacts[name]

print(f"Contact {name} deleted successfully!")

else:

print(f"Contact {name} not found!")

def main():

contact\_book = ContactBook()

while True:

print("1. Add Contact")

print("2. View Contacts")

print("3. Search Contact")

print("4. Update Contact")

print("5. Delete Contact")

print("6. Exit")

choice = input("Enter your choice: ")

if choice == "1":

name = input("Enter name: ")

phone\_number = input("Enter phone number: ")

email = input("Enter email: ")

contact\_book.add\_contact(name, phone\_number, email)

elif choice == "2":

contact\_book.view\_contacts()

elif choice == "3":

name = input("Enter name to search: ")

contact\_book.search\_contact(name)

elif choice == "4":

name = input("Enter name to update: ")

phone\_number = input("Enter new phone number (press enter to skip): ")

email = input("Enter new email (press enter to skip): ")

contact\_book.update\_contact(name, phone\_number or None, email or None)

elif choice == "5":

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

contact\_book.delete\_contact(name)

elif choice == "6":

break

else:

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

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

main()