Weekly Progress Report

Name: Ashutosh Kumar Muni
Domain: Python
Date of submission:23/2/2024

Week Ending: 01

I. Overview:

This week, the primary focus was on understanding USC_TIA and contributing to Python projects. Additionally, efforts were made to leverage learning resources for skill enhancement. As the python projects assigned are providing different application of the python.

II. Achievements:

1. Problem solving skill:
   - When worked under several projects the problem solving skills are improved.
   - Successfully executed and got ideas for solution.

2. Python Project Contributions:

URL shortener:-

```
import pyshorteners
import time
def shorten_url(url):
    s = pyshorteners.Shortener()
    return s.tinyurl.short(url)

original_url = input("Enter a url link :")
print("___wait a while___")
shortened_url = shorten_url(original_url)
time.sleep(5)
```

```
print("Successful Shortened URL:", shortened_url)
```

- This code worked on the basic of library i.e python shortener to short the URL provided and output I s generate to get the short link .

File Organizer :-

```
import os
from tkinter import filedialog

folderpath = filedialog.askdirectory()
#filepath = filedialog.askopenfilename()

if folderpath  != "":
    os.chdir(folderpath)
else:
    exit()

for i in os.listdir():
    text_touple = os.path.splitext(i)
    extension = text_touple[-1]

    if extension != "":
        extension = extension.replace(".",",")

        try:
            os.mkdir(f"{extension}_files")
        except FileExistsError:
            pass
        except Exception as e:
            print(e)

    else:
         print(f"FOLDER_{i}")
        pass

print("\n\n\t\t:")
```

Password Manager :-

```
import tkinter as tk
from tkinter import messagebox

def save_password():
    website = website_entry.get()
    username = username_entry.get()
    password = password_entry.get()



    if not website or not username or not password:
        messagebox.showerror("Error", "Please fill in all fields.")
        return



    with open("passwords.txt", "a") as file:
        file.write(f"Website: {website}\n")
        file.write(f"Username: {username}\n")
        file.write(f"Password: {password}\n\n")

    website_entry.delete(0, tk.END)
    username_entry.delete(0, tk.END)
    password_entry.delete(0, tk.END)



def retrieve_passwords():
    passwords = ""
    try:
        with open("passwords.txt", "r") as file:
            passwords = file.read()
    except FileNotFoundError:
        passwords = "Password file not found."

    messagebox.showinfo("Stored Passwords", passwords)

root = tk.Tk()
root.title("Password Manager")
root.geometry("400x300")



website_label = tk.Label(root, text="Website:")
website_label.pack()
website_entry = tk.Entry(root)
website_entry.pack()

username_label = tk.Label(root, text="Username:")
username_label.pack()
username_entry = tk.Entry(root)
username_entry.pack()

password_label = tk.Label(root, text="Password:")
password_label.pack()
password_entry = tk.Entry(root, show="*")
password_entry.pack()



save_button = tk.Button(root, text="Save Password", command=save_password)
save_button.pack()

retrieve_button= tk.Button(root, text="Retrieve Passwords",
command=retrieve_passwords)
```

```
retrieve_button.pack()
root.mainloop()
```

Quiz Game :-

```
from tkinter import *
from tkinter import ttk  # Fix the import statement typo

y = 0
a = ttk.Notebook()

frame1 = ttk.Frame(a)
frame2 = ttk.Frame(a)
frame3 = ttk.Frame(a)
frame4 = ttk.Frame(a)
frame5 = ttk.Frame(a)
frame6 = ttk.Frame(a)
frame7 = ttk.Frame(a)
frame8 = ttk.Frame(a)
frame9 = ttk.Frame(a)
frame10 = ttk.Frame(a)

root = ttk.Frame(a)

def quiz(y):
    a.add(frame1, text="Q1")
    a.add(frame2, text="Q2")
    a.add(frame3, text="Q3")
    a.add(frame4, text="Q4")
    a.add(frame5, text="Q5")
    a.add(frame6, text="Q6")
    a.add(frame7, text="Q7")
    a.add(frame8, text="Q8")
    a.add(frame9, text="Q9")
    a.add(frame10, text="Q10")


    Label(frame1, text="Total keywords in python ?", font=("Arial", 40, "bold")).grid(row=2, column=2)
    Button(frame1, text="33", font=("calibri", 30, "bold"), bg="light
blue",command=a1_right).grid(row=3, column=1)
    Button(frame1, text="31", font=("calibri", 30, "bold"), bg="light
green",command=a1_wrong).grid(row=3, column=2)
    Button(frame1, text="30", font=("calibri", 30, "bold"), bg="light
pink",command=a1_wrong).grid(row=3, column=3)

    Label(frame2, text="Output of 2**3 ?", font=("Arial", 40, "bold")).grid(row=2, column=2)
    Button(frame2, text="6", font=("calibri", 30, "bold"), bg="light
blue",command=a2_wrong).grid(row=3, column=1)
    Button(frame2, text="8", font=("calibri", 30, "bold"), bg="light
green",command=a2_right).grid(row=3, column=2)
    Button(frame2, text="9", font=("calibri", 30, "bold"), bg="light
pink",command=a2_wrong).grid(row=3, column=3)

    Label(frame3, text="Output of np.arrange(1,5) ?", font=("Arial", 40, "bold")).grid(row=2, column=2)
    Button(frame3, text="[1,2,3,4]", font=("calibri", 30, "bold"), bg="light
blue",command=a3_right).grid(row=3, column=1)
    Button(frame3, text="[0,1,2,3,4]", font=("calibri", 30, "bold"), bg="light
green",command=a3_wrong).grid(row=3, column=2)
    Button(frame3, text="[1,2,3,4,5]", font=("calibri", 30, "bold"), bg="light
pink",command=a3_wrong).grid(row=3, column=3)

    Label(frame4, text="Output of all([2,4,0,6]) ?", font=("Arial", 40, "bold")).grid(row=2, column=2)
    Button(frame4, text="true", font=("calibri", 30, "bold"), bg="light
blue",command=a4_wrong).grid(row=3, column=1)
    Button(frame4, text="false", font=("calibri", 30, "bold"), bg="light
green",command=a4_right).grid(row=3, column=2)
    Button(frame4, text="0", font=("calibri", 30, "bold"), bg="light
pink",command=a4_wrong).grid(row=3, column=3)

    Label(frame5, text="Output of 2*12 ?", font=("Arial", 40, "bold")).grid(row=2, column=2)
    Button(frame5, text="24", font=("calibri", 30, "bold"), bg="light
blue",command=a5_right).grid(row=3, column=1)
    Button(frame5, text="28", font=("calibri", 30, "bold"), bg="light
green",command=a5_wrong).grid(row=3, column=2)
    Button(frame5, text="32", font=("calibri", 30, "bold"), bg="light
pink",command=a5_wrong).grid(row=3, column=3)
```

```
    Label(frame6, text="Python is ____ programming lang ?", font=("Arial", 30, "bold")).grid(row=2, column=2)
    Button(frame6, text="high level", font=("calibri", 30, "bold"), bg="light blue",command=a6_right).grid(row=3, column=1)
    Button(frame6, text="low level", font=("calibri", 30, "bold"), bg="light green",command=a6_wrong).grid(row=3, column=2)
    Button(frame6, text="none", font=("calibri", 30, "bold"), bg="light pink",command=a6_wrong).grid(row=3, column=3)

    Label(frame7, text="Python operator always yields the result of ____ ?", font=("Arial", 30, "bold")).grid(row=2, column=2)
    Button(frame7, text="integer", font=("calibri", 25, "bold"), bg="light blue",command=a7_wrong).grid(row=3, column=1)
    Button(frame7, text="complex", font=("calibri", 25, "bold"), bg="light green",command=a7_wrong).grid(row=3, column=2)
    Button(frame7, text="floating point", font=("calibri", 25, "bold"), bg="light pink",command=a7_right).grid(row=3, column=3)

    Label(frame8, text="String in python are ___ ?", font=("Arial", 40, "bold")).grid(row=2, column=2)
    Button(frame8, text="mutable", font=("calibri", 30, "bold"), bg="light blue",command=a8_wrong).grid(row=3, column=1)
    Button(frame8, text="immutable", font=("calibri", 30, "bold"), bg="light green",command=a8_right).grid(row=3, column=2)
    Button(frame8, text="fixed", font=("calibri", 30, "bold"), bg="light pink",command=a8_wrong).grid(row=3, column=3)

    Label(frame9, text="Which statement is used for error checking ?", font=("Arial", 30, "bold")).grid(row=2, column=2)
    Button(frame9, text="list", font=("calibri", 25, "bold"), bg="light blue",command=a9_wrong).grid(row=3, column=1)
    Button(frame9, text="Assert", font=("calibri", 25, "bold"), bg="light green",command=a9_right).grid(row=3, column=2)
    Button(frame9, text="tuple", font=("calibri", 25, "bold"), bg="light pink",command=a9_wrong).grid(row=3, column=3)

    Label(frame10, text=" which is used for multiline comment?", font=("Arial", 40, "bold")).grid(row=2, column=2)
    Button(frame10, text="//", font=("calibri", 30, "bold"), bg="light blue",command=a10_wrong).grid(row=3, column=1)
```

```
   Button(frame10, text="//", font=("calibri", 30, "bold"), bg="light
green",command=a10_right).grid(row=3, column=2)
   Button(frame10, text=" #", font=("calibri", 30, "bold"), bg="light
pink",command=a10_wrong).grid(row=3, column=3)


def a1_right():

Label(frame1,text="Correct",font=("Arial",40,"bold"),background="green",fg="yellow").grid(row=1,col
umn=2)
   Label(frame1,text="Marks obtained :
1",font=("Arial",30,"bold"),background="black",fg="white").grid(row=1,column=3)

def a1_wrong():

Label(frame1,text="Incorrect",font=("Arial",40,"bold"),background="green",fg="yellow").grid(row=1,c
olumn=2)
   Label(frame1,text="Marks obtained :
0",font=("Arial",30,"bold"),background="black",fg="white").grid(row=1,column=3)


def a2_right():

Label(frame2,text="Correct",font=("Arial",40,"bold"),background="green",fg="yellow").grid(row=1,col
umn=2)
   Label(frame2,text="Marks obtained :
1",font=("Arial",30,"bold"),background="black",fg="white").grid(row=1,column=3)

def a2_wrong():

Label(frame2,text="Incorrect",font=("Arial",40,"bold"),background="green",fg="yellow").grid(row=1,c
olumn=2)
   Label(frame2,text="Marks obtained :
0",font=("Arial",30,"bold"),background="black",fg="white").grid(row=1,column=3)


def a3_right():
```

```
Label(frame3,text="Correct",font=("Arial",40,"bold"),background="green",fg="yellow").grid(row=1,col
umn=2)
    Label(frame3,text="Marks obtained :
1",font=("Arial",30,"bold"),background="black",fg="white").grid(row=1,column=3)

def a3_wrong():

Label(frame3,text="Incorrect",font=("Arial",40,"bold"),background="green",fg="yellow").grid(row=1,c
olumn=2)
    Label(frame3,text="Marks obtained :
0",font=("Arial",30,"bold"),background="black",fg="white").grid(row=1,column=3)



def a4_right():

Label(frame4,text="Correct",font=("Arial",40,"bold"),background="green",fg="yellow").grid(row=1,col
umn=2)
    Label(frame4,text="Marks obtained :
1",font=("Arial",40,"bold"),background="black",fg="white").grid(row=1,column=3)

def a4_wrong():

Label(frame4,text="Incorrect",font=("Arial",40,"bold"),background="green",fg="yellow").grid(row=1,c
olumn=2)
    Label(frame4,text="Marks obtained :
0",font=("Arial",30,"bold"),background="black",fg="white").grid(row=1,column=3)



def a5_right():

Label(frame5,text="Correct",font=("Arial",40,"bold"),background="green",fg="yellow").grid(row=1,col
umn=2)
    Label(frame5,text="Marks obtained :
1",font=("Arial",30,"bold"),background="black",fg="white").grid(row=1,column=3)

def a5_wrong():
```

```
Label(frame5,text="Incorrect",font=("Arial",40,"bold"),background="green",fg="yellow").grid(row=1,c
olumn=2)
   Label(frame5,text="Marks obtained :
0",font=("Arial",30,"bold"),background="black",fg="white").grid(row=1,column=3)

def a6_right():

Label(frame6,text="Correct",font=("Arial",40,"bold"),background="green",fg="yellow").grid(row=1,col
umn=2)
   Label(frame6,text="Marks obtained :
1",font=("Arial",20,"bold"),background="black",fg="white").grid(row=1,column=3)

def a6_wrong():

Label(frame6,text="Incorrect",font=("Arial",40,"bold"),background="green",fg="yellow").grid(row=1,c
olumn=2)
   Label(frame6,text="Marks obtained :
0",font=("Arial",20,"bold"),background="black",fg="white").grid(row=1,column=3)



def a7_right():

Label(frame7,text="Correct",font=("Arial",40,"bold"),background="green",fg="yellow").grid(row=1,col
umn=2)
   Label(frame7,text="Marks obtained :
1",font=("Arial",20,"bold"),background="black",fg="white").grid(row=1,column=3)

def a7_wrong():

Label(frame7,text="Incorrect",font=("Arial",30,"bold"),background="green",fg="yellow").grid(row=1,c
olumn=2)
   Label(frame7,text="Marks obtained :
0",font=("Arial",20,"bold"),background="black",fg="white").grid(row=1,column=3)



def a8_right():
```

```
Label(frame8,text="Correct",font=("Arial",40,"bold"),background="green",fg="yellow").grid(row=1,col
umn=2)
   Label(frame8,text="Marks obtained :
1",font=("Arial",30,"bold"),background="black",fg="white").grid(row=1,column=3)

def a8_wrong():

Label(frame8,text="Incorrect",font=("Arial",40,"bold"),background="green",fg="yellow").grid(row=1,c
olumn=2)
   Label(frame8,text="Marks obtained :
0",font=("Arial",30,"bold"),background="black",fg="white").grid(row=1,column=3)



def a9_right():

Label(frame9,text="Correct",font=("Arial",40,"bold"),background="green",fg="yellow").grid(row=1,col
umn=2)
   Label(frame9,text="Marks obtained :
1",font=("Arial",25,"bold"),background="black",fg="white").grid(row=1,column=3)

def a9_wrong():

Label(frame9,text="Incorrect",font=("Arial",40,"bold"),background="green",fg="yellow").grid(row=1,c
olumn=2)
   Label(frame9,text="Marks obtained :
0",font=("Arial",25,"bold"),background="black",fg="white").grid(row=1,column=3)



def a10_right():

Label(frame10,text="Correct",font=("Arial",40,"bold"),background="green",fg="yellow").grid(row=1,c
olumn=2)
   Label(frame10,text="Marks obtained :
1",font=("Arial",30,"bold"),background="black",fg="white").grid(row=1,column=3)

def a10_wrong():
   Label(frame10,text="better luck next
time",font=("Arial",40,"bold"),background="green",fg="yellow").grid(row=1,column=2)
```

```
        Label(frame10,text="Marks obtained :
      0",font=("Arial",30,"bold"),background="black",fg="white").grid(row=1,column=3)


quiz(y)
a.pack()
a.mainloop()
```

3.Learning Python:

- Acquired proficiency in essential Python libraries, such as tkinter ,date and time ,py shorterner, default libarary.
- Applied Python skills to real-world problems within USC_TIA context.

III. Challenges:

1. USC_TIA Integration:

- Encountered challenges during USC_TIA integration with VS code editor ,sublime text 3 editor and Python IDE.
- Ongoing efforts to troubleshoot and ensure successful integration.

2. Python Project Complexity:

- Faced complexity in understanding fetching the database, creating accurate output, working on performed skill for debugging of the Python project.
- Seeking guidance to overcome challenges and enhance understanding.
- Issue in making the quiz game with proper output.

IV. Learning Resources:

1. USC_TIA Documentation:

- Utilized USC_TIA official documentation for reference and troubleshooting.
- Attended relevant webinars and online tutorials to deepen understanding.

2. Python Learning Resources:

- https://youtu.be/lFSkmIEjckc?si=uPYzULKc78vzwqAy
- https://youtu.be/KBjBPQExJLw?si=1CX63Kj5Vki6YI1u
- Python Tutorial | Learn Python Programming (geeksforgeeks.org)
- Python GUI - tkinter - GeeksforGeeks

V. Next Week's Goals:

1. USC_TIA Enhancement:

- Address integration challenges and explore advanced USC_TIA features.
- Collaborate with peers to contribute to USC_TIA improvement discussions.

2.Python Project Development:

- Tackle more complex tasks within the Python project to increase contribution.
- Seek feedback from mentors and peers for continuous improvement.

VI. Additional Comments:

I have made commendable progress this week, excelling in collaboration and meeting project milestones. My efforts in working on the quiz game, file organizer, password manager, and URL shortener have been seamless, showcasing a versatile skill set. Engaging discussions on quiz game design and insightful considerations for file organization have enriched my experiences. Challenges, particularly in seamless integration, were overcome with effective problem-solving skills. Learning experiences involve advanced Python libraries for encryption in the password manager and efficient algorithms for the URL shortener. Looking ahead, I aim to implement multi-user support and analytics for enhanced functionality. Regular updates and check-ins have maintained transparent communication. The project has fostered my awareness about security considerations in handling sensitive information. I am poised for the next phase, anticipating further innovation and success. Overall, this week's efforts reflect strong dedication, collaboration, and a commitment to delivering a comprehensive Python application.