CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

Name : JOSANTH SMILAN A
Stream : I MCA-B
Reg No : 2347228

```
import tkinter as tk
from tkinter import ttk
import re

# Function to validate the Name
def validate_name(name):
    # Name should only contain letters, and spaces
    return re.match(r"^[A-Za-z\s]+$", name)

# Function to validate the Register No
def validate_register_no(register_no):
    # Register No validation (customize this as per your requirements)
    return re.match(r"^[A-Z0-9]+$", register_no)

# Function to validate the Email
def validate_email(email):
    # Email validation using a simple regular expression
    return re.match(r"^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$", email)

# Function to validate the Phone Number
def validate_phone(phone):
    # Phone number validation using a simple regular expression
```

```
    return re.match(r"^[0-9]{10}$", phone)

# Create the main window
root = tk.Tk()
root.title("User Profile Form")

# Create and pack a frame for the form elements
frame = ttk.Frame(root)
frame.pack(padx=20, pady=20)

# Create and pack form labels and entry widgets
ttk.Label(frame, text="Name :").grid(row=0, column=0, sticky="w")
name_entry = ttk.Entry(frame)
name_entry.grid(row=0, column=1, padx=10, pady=5)

# Create a label and an entry widget for the "Register No" field
ttk.Label(frame, text="Register No :").grid(row=1, column=0, sticky="w")
register_no_entry = ttk.Entry(frame)
register_no_entry.grid(row=1, column=1, padx=10, pady=5)

ttk.Label(frame, text="Email Id :").grid(row=2, column=0, sticky="w")
email_entry = ttk.Entry(frame)
email_entry.grid(row=2, column=1, padx=10, pady=5)

ttk.Label(frame, text="Phone Number :").grid(row=3, column=0, sticky="w")
phone_entry = ttk.Entry(frame)
phone_entry.grid(row=3, column=1, padx=10, pady=5)

# Create radio buttons for the "Gender" field
ttk.Label(frame, text="Gender :").grid(row=4, column=0, sticky="w")
gender_var = tk.StringVar()
```

```
male_radio = ttk.Radiobutton(frame, text="Male", variable=gender_var, value="Male")
female_radio = ttk.Radiobutton(frame, text="Female", variable=gender_var, value="Female")
other_radio = ttk.Radiobutton(frame, text="Other", variable=gender_var, value="Other")
male_radio.grid(row=4, column=1, padx=10, pady=5)
female_radio.grid(row=4, column=2, padx=10, pady=5)
other_radio.grid(row=4, column=3, padx=10, pady=5)

ttk.Label(frame, text="Year of Birth :").grid(row=5, column=0, sticky="w")
dob_spinbox = ttk.Spinbox(frame, from_=1900, to=2023)
dob_spinbox.grid(row=5, column=1, padx=10, pady=5)

# List of course options for the drop-down list
course_options = ["Math", "Science", "History", "English", "Computer Science"]

# Create a label and a Combobox for the "Course" field
ttk.Label(frame, text="Select Stream :").grid(row=6, column=0, sticky="w")
course_var = tk.StringVar()
course_combobox = ttk.Combobox(frame, textvariable=course_var, values=course_options)
course_combobox.grid(row=6, column=1, padx=10, pady=5)



# Function to handle form submission
def submit_form():
    name = name_entry.get()
    register_no = register_no_entry.get()  # Retrieve the entered register number
    email = email_entry.get()
    phone = phone_entry.get()
    gender = gender_var.get()  # Retrieve the selected gender
    course = course_var.get()  # Retrieve the selected course
```

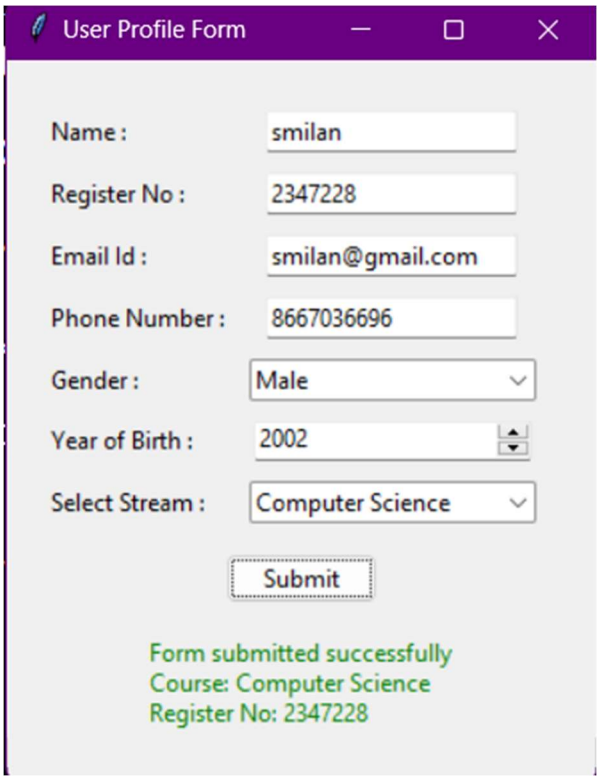
```
    # Validate Name, Email, Phone Number, and Register No
    if not validate_name(name):
        result_label.config(text="Invalid Name", foreground="red")
    elif not validate_register_no(register_no):
        result_label.config(text="Invalid Register No", foreground="red")
    elif not validate_email(email):
        result_label.config(text="Invalid Email", foreground="red")
    elif not validate_phone(phone):
        result_label.config(text="Invalid Phone Number", foreground="red")
    else:
        result_label.config(text=f"Form submitted successfully\nGender: {gender}\nCourse: {course}\nRegister No: {register_no}", foreground="green")

# Create and pack a Submit button
submit_button = ttk.Button(frame, text="Submit", command=submit_form)
submit_button.grid(row=7, column=0, columnspan=2, pady=10)

# Create a label for displaying validation results
result_label = ttk.Label(frame, text="", foreground="green")
result_label.grid(row=8, column=0, columnspan=2, pady=5)

# Start the Tkinter main loop
root.mainloop()
```

Output :



LAB : 11

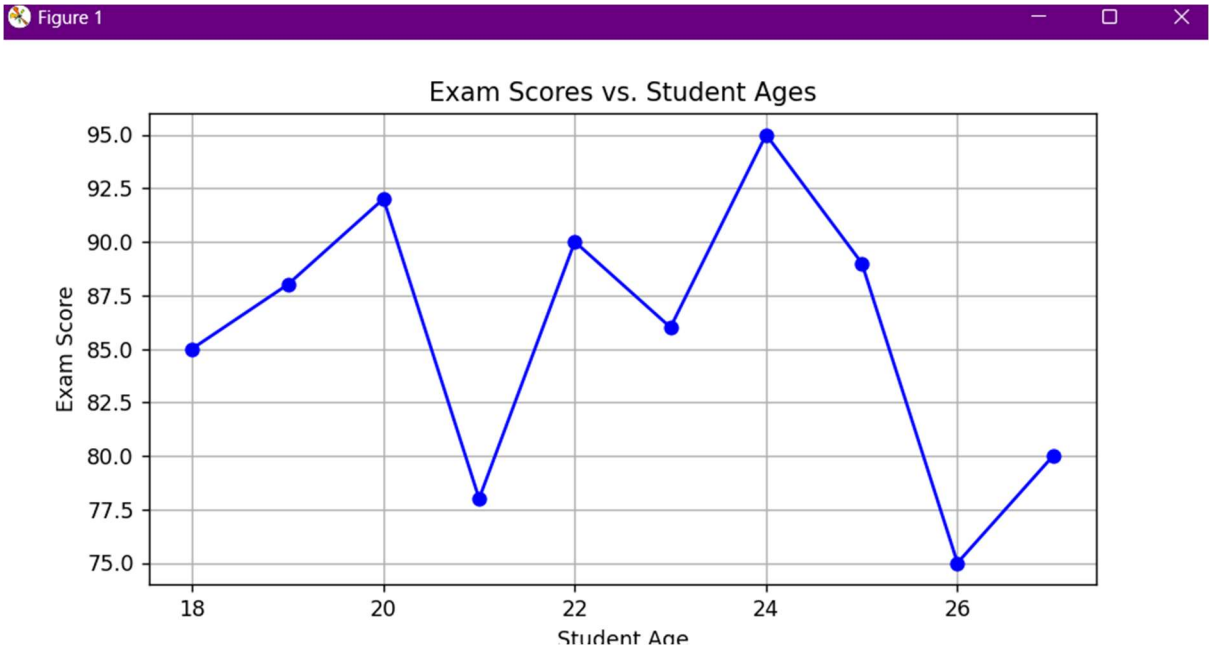
```
import matplotlib.pyplot as plt
import numpy as np

# Hypothetical dataset (student ages and exam scores)
student_ages = [18, 19, 20, 21, 22, 23, 24, 25, 26, 27]
exam_scores = [85, 88, 92, 78, 90, 86, 95, 89, 75, 80]

# Line Plot: Visualizing the relationship between age and exam score
plt.figure(figsize=(8, 4))
plt.plot(student_ages, exam_scores, marker='o', linestyle='-', color='b')
plt.title('Exam Scores vs. Student Ages')
plt.xlabel('Student Age')
plt.ylabel('Exam Score')
plt.grid(True)
plt.show()
```

```
# Scatter Plot: Visualizing the distribution of exam scores
plt.figure(figsize=(8, 4))
plt.scatter(student_ages, exam_scores, color='r', alpha=0.7)
plt.title('Scatter Plot of Exam Scores')
plt.xlabel('Student Age')
plt.ylabel('Exam Score')
plt.grid(True)
plt.show()
```

Output :



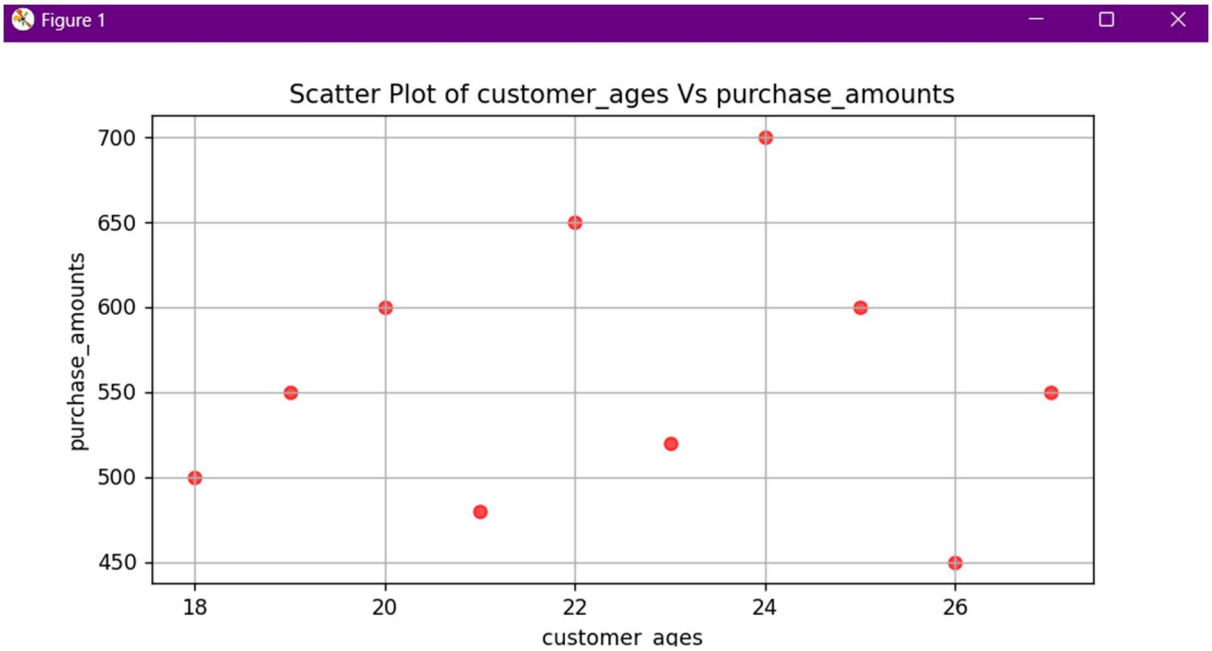


Figure 1
Scatter Plot of customer_ages Vs purchase_amounts
purchase_amounts
700
650
600
550
500
450
18
20
22
24
26
customer_ages

p