

Q1. Apply regular expression for form validation. Create your domain-form using Tkinter Module. Form should contain Text box [For Name, Email Id, Phone number], Dropdown [for Gender], Spinbox [for Year/DoB] and other necessary widgets required for your domain. Validate Your Name, Email Id, Phone number in the form.

```
In [ ]: import tkinter as tk
from tkinter import ttk
from tkinter import messagebox
import re

def validate_age(age):
    try:
        age = int(age)
        if age < 18:
            return False
        return True
    except ValueError:
        return False

def validate_name(name):
    return bool(re.match(r'^[A-Za-z\s]+$', name))

def validate_phone_number(phone):
    return bool(re.match(r'^\d{10}$', phone))

def submit_form():
    Voter_name = name_entry.get()
    Parties = Parties_var.get()
    Gender = gender_var.get()
    Age = age_entry.get()
    Location = location_entry.get()
    Address = address_entry.get()
    Party_Symbol = party_symbol_var.get()
    State = state_var.get()
    Phone = phone_entry.get()
    DateOfBirth = f"{day_spinbox.get()} {month_spinbox.get()} {year_spinbox.get()}"

    # Validate Age
    if not validate_age(Age):
        messagebox.showerror("Error", "Age must be a valid number and at least 18")
        return

    # Validate Voter Name
    if not validate_name(Voter_name):
        messagebox.showerror("Error", "Voter Name must contain only letters and spaces")
        return

    # Validate Location
    if not validate_name(Location):
        messagebox.showerror("Error", "Location must contain only letters")
        return

    # Validate Address
    if not validate_name(Address):
        messagebox.showerror("Error", "Address must contain only letters and numbers")
        return
```

```

        messagebox.showerror("Error", "Address must contain only letters")
        return

    # Validate Phone Number
    if not validate_phone_number(Phone):
        messagebox.showerror("Error", "Phone number must contain exactly")
        return

    # Display a confirmation message
    confirmation_label.config(text=f"Thank you, {Voter_name}! Your vote f

# Create the main window
root = tk.Tk()
root.title("Online Election System")

# Create a label for the title
title_label = tk.Label(root, text="Online Election Voting Form", font=("H
title_label.pack(pady=10)

# Create a frame to group related widgets
form_frame = ttk.Frame(root)
form_frame.pack(pady=20)

# Voter_Name
name_label = ttk.Label(form_frame, text="Voter Name:", foreground="yellow")
name_label.grid(row=0, column=0, padx=10, pady=5, sticky="w")
name_entry = ttk.Entry(form_frame)
name_entry.grid(row=0, column=1, padx=10, pady=5)

# Parties
Parties_label = ttk.Label(form_frame, text="Parties:", foreground="yellow")
Parties_label.grid(row=1, column=0, padx=10, pady=5, sticky="w")
Parties_var = tk.StringVar()
Parties_combobox = ttk.Combobox(form_frame, textvariable=Parties_var, val
Parties_combobox.grid(row=1, column=1, padx=10, pady=5)

# Gender
gender_label = ttk.Label(form_frame, text="Gender:", foreground="yellow")
gender_label.grid(row=2, column=0, padx=10, pady=5, sticky="w")
gender_var = tk.StringVar()
male_radio = ttk.Radiobutton(form_frame, text="Male", variable=gender_var
female_radio = ttk.Radiobutton(form_frame, text="Female", variable=gender
male_radio.grid(row=2, column=1, padx=10, pady=5, sticky="w")
female_radio.grid(row=2, column=2, padx=10, pady=5, sticky="w")

# Age
age_label = ttk.Label(form_frame, text="Age:", foreground="yellow")
age_label.grid(row=3, column=0, padx=10, pady=5, sticky="w")
age_entry = ttk.Entry(form_frame)
age_entry.grid(row=3, column=1, padx=10, pady=5)

# Location
location_label = ttk.Label(form_frame, text="Location:", foreground="yell
location_label.grid(row=4, column=0, padx=10, pady=5, sticky="w")
location_entry = ttk.Entry(form_frame)
location_entry.grid(row=4, column=1, padx=10, pady=5)

# Address
address_label = ttk.Label(form_frame, text="Address:", foreground="yellow")
address_label.grid(row=5, column=0, padx=10, pady=5, sticky="w")

```

```
address_entry = ttk.Entry(form_frame)
address_entry.grid(row=5, column=1, padx=10, pady=5)

# Party Symbol
party_symbol_label = ttk.Label(form_frame, text="Party Symbol:", foreground="red")
party_symbol_label.grid(row=6, column=0, padx=10, pady=5, sticky="w")
party_symbol_var = tk.StringVar()
party_combobox = ttk.Combobox(form_frame, textvariable=party_symbol_var, values=party_symbols)
party_combobox.grid(row=6, column=1, padx=10, pady=5)

# State
state_label = ttk.Label(form_frame, text="State:", foreground="yellow")
state_label.grid(row=7, column=0, padx=10, pady=5, sticky="w")
state_var = tk.StringVar()
state_combobox = ttk.Combobox(form_frame, textvariable=state_var, values=states)
state_combobox.grid(row=7, column=1, padx=10, pady=5)

# Phone Number
phone_label = ttk.Label(form_frame, text="Phone Number:", foreground="yellow")
phone_label.grid(row=8, column=0, padx=10, pady=5, sticky="w")
phone_entry = ttk.Entry(form_frame)
phone_entry.grid(row=8, column=1, padx=10, pady=5)

# Date of Birth
dob_label = ttk.Label(form_frame, text="Date of Birth:", foreground="yellow")
dob_label.grid(row=9, column=0, padx=10, pady=5, sticky="w")

day_spinbox = ttk.Spinbox(form_frame, from_=1, to=31, width=3)
day_spinbox.grid(row=9, column=1, padx=5, pady=5, sticky="w")

month_spinbox = ttk.Spinbox(form_frame, from_=1, to=12, width=3)
month_spinbox.grid(row=9, column=2, padx=5, pady=5, sticky="w")

year_spinbox = ttk.Spinbox(form_frame, from_=1900, to=2023, width=5)
year_spinbox.grid(row=9, column=3, padx=5, pady=5, sticky="w")

# Submit Button
submit_button = ttk.Button(root, text="Submit", command=submit_form, style=submit_style)
submit_button.pack(pady=10)

# Confirmation Label
confirmation_label = ttk.Label(root, text="", foreground="red")
confirmation_label.pack()

# Style for Submit Button
style = ttk.Style()
style.configure("Submit.TButton", foreground="white", font=("Helvetica", 12))

root.mainloop()
```