P9_2347138

September 27, 2023

Apply regular expressions for form validation(TkInter).

1. Create a form using the following widgets for your domain. Label, Entry, Button, RadioButton, OptionMenu, Checkbutton, message box
2. Apply regular expression to validate the input of all widgets. Reuse your code from P6: Implement 're' module
3. Make a simple calculation related to your domain. E.g. Age from DOB, Amount to be paid, Year of experience from date of joining, etc.
4. Display all widget inputs that are received from the user.

[ ]:
```
# P9
import tkinter as tk
from tkinter import messagebox
import re
from datetime import datetime


# Function to calculate age from DOB
def calculate_age():
    dob = entry_dob.get()
    try:
        dob_date = datetime.strptime(dob, "%d-%m-%Y")
        today = datetime.today()
        age = (
            today.year
            - dob_date.year
            - ((today.month, today.day) < (dob_date.month, dob_date.day))
        )
        result_label.config(text=f"Age: {age} years")
    except ValueError:
        messagebox.showerror("Error", "Invalid Date of Birth format. Use␣
↪DD-MM-YYYY")


# Function to calculate fee amount
def calculate_fee():
    subscription = subscription_var.get()
    fee = 0
```

```
    if subscription == "Basic":
        fee = 100
    elif subscription == "Pro":
        fee = 200
    elif subscription == "Premium":
        fee = 300

    result_label.config(text=f"Fee Amount: ${fee}")


# Function to validate and display all inputs, including age
def validate_and_display():
    name = entry_name.get()
    email = entry_email.get()
    phone = entry_phone.get()
    dob = entry_dob.get()
    gender = gender_var.get()
    subscription = subscription_var.get()
    is_student = student_var.get()  # Checkbutton value

    # Regular expressions for validation
    email_pattern = r"^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$"
    phone_pattern = r"^\d{10}$"

    # Check if the "Student" checkbox is ticked
    if not is_student:
        messagebox.showerror("Error", "You must accept the terms and conditions.
 ↪")
        return

    # Check if inputs match the regular expressions
    if not re.match(email_pattern, email):
        messagebox.showerror("Error", "Invalid Email Address")
    elif not re.match(phone_pattern, phone):
        messagebox.showerror("Error", "Invalid Phone Number")
    else:
        try:
            dob_date = datetime.strptime(dob, "%d-%m-%Y")
            today = datetime.today()
            age = (
                today.year
                - dob_date.year
                - ((today.month, today.day) < (dob_date.month, dob_date.day))
            )

            # Create a message with all the information
```

```
            result_text = (
                f"Name: {name}\nEmail: {email}\nPhone: {phone}\n"
                f"DOB: {dob}\nGender: {gender}\nSubscription: {subscription}\n"
                f"Is Student: {is_student}\nAge: {age} years"
            )

            # Display the information in a message box
            messagebox.showinfo("Registration Details", result_text)

            # Set the result_label to display the age and fee
            result_label.config(text=f"Age: {age} years")
            calculate_fee()  # Calculate and display fee
        except ValueError:
            messagebox.showerror(
                "Error", "Invalid Date of Birth format. Use DD-MM-YYYY"
            )


# Create the main window
root = tk.Tk()
root.title("Student Registration Form")

# Set the window size
window_width = 500
window_height = 500
screen_width = root.winfo_screenwidth()
screen_height = root.winfo_screenheight()

x = (screen_width / 2) - (window_width / 2)
y = (screen_height / 2) - (window_height / 2)

root.geometry(f"{window_width}x{window_height}+{int(x)}+{int(y)}")

# Create a frame for the registration form
frame = tk.Frame(root)
frame.place(relx=0.5, rely=0.5, anchor=tk.CENTER)

# Create and place widgets on the form
label_name = tk.Label(frame, text="Name:")
entry_name = tk.Entry(frame)

label_email = tk.Label(frame, text="Email:")
entry_email = tk.Entry(frame)

label_phone = tk.Label(frame, text="Phone:")
entry_phone = tk.Entry(frame)
```

```
label_dob = tk.Label(frame, text="Date of Birth (DD-MM-YYYY):")
entry_dob = tk.Entry(frame)

gender_var = tk.StringVar()
label_gender = tk.Label(frame, text="Gender:")
male_radio = tk.Radiobutton(frame, text="Male", variable=gender_var,␣
 ↪value="Male")
female_radio = tk.Radiobutton(frame, text="Female", variable=gender_var,␣
 ↪value="Female")

subscription_var = tk.StringVar()
label_subscription = tk.Label(frame, text="Subscription:")
subscription_menu = tk.OptionMenu(frame, subscription_var, "Basic", "Pro",␣
 ↪"Premium")

student_var = tk.BooleanVar()
checkbutton_student = tk.Checkbutton(
    frame, text="Terms and Conditions", variable=student_var
)

validate_button = tk.Button(frame, text="Register",␣
 ↪command=validate_and_display)

result_label = tk.Label(frame, text="", font=("Helvetica", 12))

# Place widgets using the grid layout
label_name.grid(row=0, column=0, sticky="w", padx=10, pady=5)
entry_name.grid(row=0, column=1, padx=10, pady=5)

label_email.grid(row=1, column=0, sticky="w", padx=10, pady=5)
entry_email.grid(row=1, column=1, padx=10, pady=5)

label_phone.grid(row=2, column=0, sticky="w", padx=10, pady=5)
entry_phone.grid(row=2, column=1, padx=10, pady=5)

label_dob.grid(row=3, column=0, sticky="w", padx=10, pady=5)
entry_dob.grid(row=3, column=1, padx=10, pady=5)

label_gender.grid(row=4, column=0, sticky="w", padx=10, pady=5)
male_radio.grid(row=4, column=1, padx=10, pady=5)
female_radio.grid(row=4, column=2, padx=10, pady=5)

label_subscription.grid(row=5, column=0, sticky="w", padx=10, pady=5)
subscription_menu.grid(row=5, column=1, padx=10, pady=5)

checkbutton_student.grid(row=6, column=0, columnspan=2, padx=10, pady=5)
```

```
validate_button.grid(row=7, column=0, columnspan=2, padx=10, pady=10)

result_label.grid(row=8, column=0, columnspan=2, padx=10, pady=10)

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

[ ]: