In Partial Fulfillment of the Requirements for the

ITE 121 - Intermediate Programming

**Student Profiling System**

Presented to:

Dr. Unife O. Cagas

**Professor**

Prepared by:

Justin Mae V. Repolidon   
**Student**

BSCS-1B2: Computer Science

May. 2024

**Description:**

This application is a student profiling system designed to allow students to register for a temporary ID required for entering school. It utilizes the Tkinter library for the graphical user interface (GUI) and the Pillow library for handling profile pictures. The system consists of registration and login functionalities, where students can create accounts, log in, and upload their profile pictures.

**Objectives:**

* Facilitate student registration: Provide a simple and intuitive interface for students to register for a temporary ID.
* Secure login system: Ensure that only registered students can log in using their credentials.
* Profile management: Allow students to upload and view their profile pictures within the application.
* Data management: Store student credentials and profile pictures securely within the application.

**Significance:**

This application is significant for educational institutions aiming to streamline the process of student registration and entry management. By digitizing the registration and login process, it enhances operational efficiency and reduces manual workload. Additionally, it helps in maintaining a secure and organized database of student profiles.

**Benefits:**

* Efficiency: Speeds up the student registration process by eliminating paperwork.
* Security: Ensures that only authorized students can access the school premises.
* Convenience: Offers a user-friendly interface for students to register and log in.
* Data Integrity: Maintains accurate and consistent records of student information.
* Profile Customization: Allows students to personalize their profiles with pictures.

**Features:**

* User Registration: Allows students to create accounts by providing a username and password. Ensures that passwords are confirmed correctly before registration. Includes a feature to upload a profile picture during registration.
* User Login: Validates student credentials to grant access. Displays an error message if the credentials are incorrect.
* Profile Picture Upload: Provides an option for students to upload and display a profile picture.
* Dashboard: Displays a welcome message and the student's profile picture after successful login. Includes a logout button to end the session.

**Code:**

import tkinter as tk

from tkinter import messagebox, filedialog

from PIL import Image, ImageTk

# Initialize the main application window

root = tk.Tk()

root.title("Application Form")

root.minsize(300, 300)

root.configure(borderwidth=10, relief="groove", bd=10)

# Initialize the database (dictionary)

database = {}

# Function to clear entry fields

def clear\_entry\_fields():

    # Clear all entry fields for username and password in both login and registration forms.

    username\_entry.delete(0, tk.END)

    password\_entry.delete(0, tk.END)

    if 'register\_username\_entry' in globals():

        register\_username\_entry.delete(0, tk.END)

    if 'register\_password\_entry' in globals():

        register\_password\_entry.delete(0, tk.END)

    if 'register\_confirm\_password\_entry' in globals():

        register\_confirm\_password\_entry.delete(0, tk.END)

# Function to validate login credentials

def validate\_login():

    # Validate the login credentials entered by the user.

    username = username\_entry.get()

    password = password\_entry.get()

    if username in database and database[username]["password"] == password:

        messagebox.showinfo("Login Success", "You have successfully logged in.")

        dashboard(username)

        clear\_entry\_fields()

    else:

        messagebox.showerror("Login Failed", "Invalid username or password.")

# Function to display the register form

def show\_register\_form():

    #Display the registration form for new users to register.

    login\_frame.pack\_forget()

    global register\_frame

    register\_frame = tk.Frame(root, bg="#3C3B6E")

    register\_frame.pack(fill="both", expand=True)

    # Create register form

    global register\_username\_entry, register\_password\_entry, register\_confirm\_password\_entry

    register\_username\_label = tk.Label(register\_frame, text="Username:", bg="#C3E7FD", fg="#4E6D7A", relief="solid")

    register\_username\_label.pack(fill='x', expand=True, padx=20)

    register\_username\_entry = tk.Entry(register\_frame, bg="#C3E7FD", fg="#4E6D7A", bd=2)

    register\_username\_entry.pack(fill='both', expand=True, padx=20)

    register\_password\_label = tk.Label(register\_frame, text="Password:", bg="#C3E7FD", fg="#4E6D7A", relief="solid")

    register\_password\_label.pack(fill='x', expand=True, padx=20)

    register\_password\_entry = tk.Entry(register\_frame, show="\*", bg="#C3E7FD", fg="#4E6D7A", bd=2)

    register\_password\_entry.pack(fill='both', expand=True, padx=20)

    register\_confirm\_password\_label = tk.Label(register\_frame, text="Confirm Password:", bg="#C3E7FD", fg="#4E6D7A", relief="solid")

    register\_confirm\_password\_label.pack(fill='x', expand=True, padx=20)

    register\_confirm\_password\_entry = tk.Entry(register\_frame, show="\*", bg="#C3E7FD", fg="#4E6D7A", bd=2)

    register\_confirm\_password\_entry.pack(fill='both', expand=True, padx=20)

    register\_profile\_button = tk.Button(register\_frame, text="Upload Profile Picture", command=upload\_profile\_picture, bg="#FFD700", fg="#3C3B6E")

    register\_profile\_button.pack(fill='x', expand=True, padx=50, pady=(0, 10))

    register\_submit\_button = tk.Button(register\_frame, text="Submit", command=register\_user, bg="#FFD700", fg="#3C3B6E")

    register\_submit\_button.pack(fill='x', expand=True, padx=50, pady=(0, 10))

# Function to upload profile picture

def upload\_profile\_picture():

    #Allow the user to upload a profile picture and display it in the registration form.

    filename = filedialog.askopenfilename(title="Choose your profile picture", filetypes=[("Image files", "\*.jpg \*.jpeg")])

    if filename:

        load = Image.open(filename)

        load = load.resize((100, 100))

        global profile\_image

        profile\_image = ImageTk.PhotoImage(load)

        profile\_frame = tk.Frame(register\_frame, bg="#C3E7FD", width=120, height=120)

        profile\_frame.pack(pady=10)

        profile\_frame.grid\_propagate(False)

        profile\_frame.columnconfigure(0, weight=1)

        profile\_frame.rowconfigure(0, weight=1)

        profile\_picture = tk.Label(profile\_frame, image=profile\_image, bg="#C3E7FD")

        profile\_picture.image = profile\_image

        profile\_picture.grid(row=0, column=0, sticky="nsew")

        profile\_picture.pack()

# Function to register a new user

def register\_user():

    #Register a new user with a username and password, and a profile picture.

    username = register\_username\_entry.get()

    password = register\_password\_entry.get()

    confirm\_password = register\_confirm\_password\_entry.get()

    if username.strip() != "" and password.strip() != "" and confirm\_password.strip() != "":

        if password == confirm\_password:

            if username not in database:

                database[username] = {"password": password}

                if 'profile\_image' in globals():

                    database[username]["profile\_image"] = profile\_image

                print(f"User registered successfully: {username}", database[username])

                register\_frame.pack\_forget()

                login\_frame.pack(fill='both', expand=True)

                clear\_entry\_fields()

                messagebox.showinfo("Registration Success", "You have successfully registered.")

            else:

                messagebox.showerror("Registration Failed", "Username already exists")

        else:

            messagebox.showerror("Registration Failed", "Passwords do not match")

    else:

        messagebox.showerror("Registration Failed", "Please fill out all fields")

# Function to display the dashboard

def dashboard(username):

    #Display the dashboard for the logged-in user, showing their profile picture and a welcome message.

    login\_frame.pack\_forget()

    global dashboard\_frame

    dashboard\_frame = tk.Frame(root, bg="#3C3B6E")

    dashboard\_frame.pack(fill='both', expand=True)

    if username in database and "profile\_image" in database[username]:

        profile\_frame = tk.Frame(dashboard\_frame, bg="#C3E7FD", width=120, height=120)

        profile\_frame.pack(pady=10)

        profile\_frame.grid\_propagate(False)

        profile\_frame.columnconfigure(0, weight=1)

        profile\_frame.rowconfigure(0, weight=1)

        profile\_picture = tk.Label(profile\_frame, image=database[username]["profile\_image"], bg="#C3E7FD")

        profile\_picture.image = database[username]["profile\_image"]

        profile\_picture.grid(row=0, column=0, sticky="nsew")

        profile\_picture.pack()

    welcome\_label = tk.Label(dashboard\_frame, text="Welcome, " + username, bg="#C3E7FD", fg="#4E6D7A")

    welcome\_label.pack(padx=20, pady=10)

    logout\_button = tk.Button(dashboard\_frame, text="Logout", command=logout, bg="#FFD700", fg="#3C3B6E")

    logout\_button.pack(pady=10)

# Function to log out

def logout():

    #Log out the current user and return to the login screen.

    dashboard\_frame.pack\_forget()

    login\_frame.pack(fill='both', expand=True)

    clear\_entry\_fields()

# Creating the login frame

login\_frame = tk.Frame(root, bg="#3C3B6E")

login\_frame.pack(fill='both', expand=True)

# Username label and entry field

username\_label = tk.Label(login\_frame, text="Username", bg="#C3E7FD", fg="#4E6D7A", relief="solid")

username\_label.pack(fill='x', expand=True, padx=20)

username\_entry = tk.Entry(login\_frame, bg="#C3E7FD", fg="#4E6D7A", bd=2)

username\_entry.pack(fill='both', expand=True, padx=20)

# Password label and entry field

password\_label = tk.Label(login\_frame, text="Password", bg="#C3E7FD", fg="#4E6D7A", relief="solid")

password\_label.pack(fill='x', expand=True, padx=20)

password\_entry = tk.Entry(login\_frame, show="\*", bg="#C3E7FD", fg="#4E6D7A", bd=2)

password\_entry.pack(fill='both', expand=True, padx=20)

# Login and Register buttons

login\_button = tk.Button(login\_frame, text="Login", command=validate\_login, bg="#FFD700", fg="#3C3B6E")

login\_button.pack(fill='x', expand=True, padx=50, pady=(10, 5))

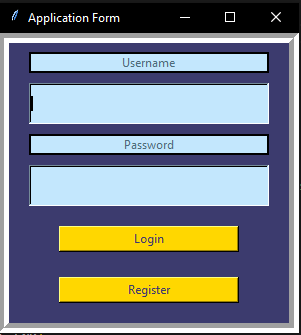
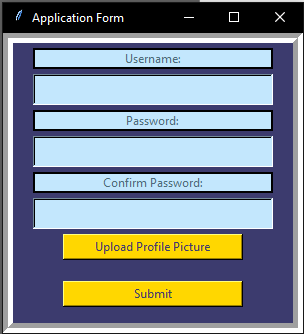
register\_button = tk.Button(login\_frame, text="Register", command=show\_register\_form, bg="#FFD700", fg="#3C3B6E")

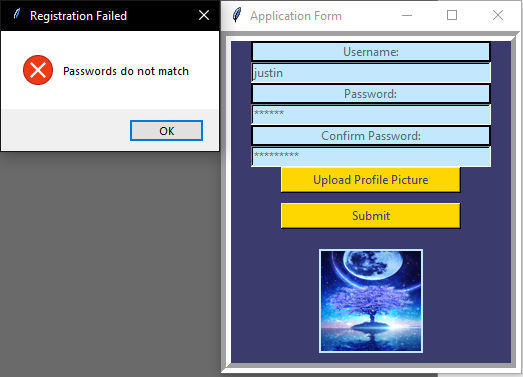
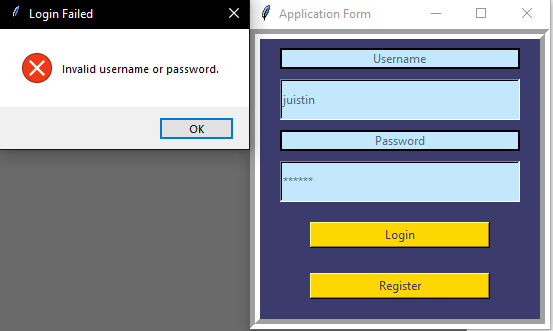
register\_button.pack(fill='x', expand=True, padx=50, pady=(0, 10))

# Start the main loop

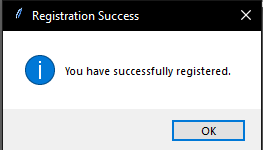
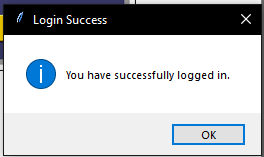
root.mainloop()

**Output:**

**Login Form Register Form  
** ****

**Error Message  
** 

**Success Registration and Login:**

**** 

**Dashboard**

