

Student Feedback System

Report

Guided By –

Dr Santosh Kumar

Submitted By –

Aditya Raj (221010203)

Abstract

Feedback is very important in any field. The project is all about Academic and mess feedback. In which we have to give our comments for the system. Also, it can show the review about the subject and respective faculty members. It brings people together and creates a healthy communication flow. The project has major work of oops terminology like class, constructor, method, inheritance. The company uses customer feedback to improve its products; For school and college, it gives all information & rating about academic system of college.

Introduction

| **TERM** | **DEFINITION** |
| --- | --- |
| Feedback | Feedback refers to information, opinions, or evaluations provided to someone regarding their performance, behaviour, or work. It is a response or reaction to a particular action, task, or project, with the intention of providing guidance, improvement, or reinforcement. |
| Object Oriented | Object-oriented programming (OOP) is a programming paradigm that organizes software design around objects, which are instances of classes. In OOP, objects are the fundamental building blocks of a program and encapsulate data (attributes) and behaviour (methods) into a single entity. |
| GUI | GUI stands for Graphical User Interface. It refers to the visual components and interactive elements of a software application that allow users to interact with the program by creating the project into widgets. |

Methodology

Display the main GUI window of login system to enter username and password, if both are not correct then message showing incorrect username or password; if correct then proceeding it to feedback form where students have to enter their name, email choose the subject and faculty and give their respective comments and rating after clicking next button new window will appear where choose the best option according to them.

The uses of Feedback class where have to add several attributes, the uses of polymorphism in labelling the widget in constructive way and uses of inheritance to create objects.

Creating a Json file where have to add some question and call it in new python file.

Then at last, add all the file in login file by import subprocess and python path file.

To store data, create a csv file where have to add all the feedback ratings of respective form.

In Monitoring all the data and ratings programmer operate csv and find average ratings of a particular subjects.

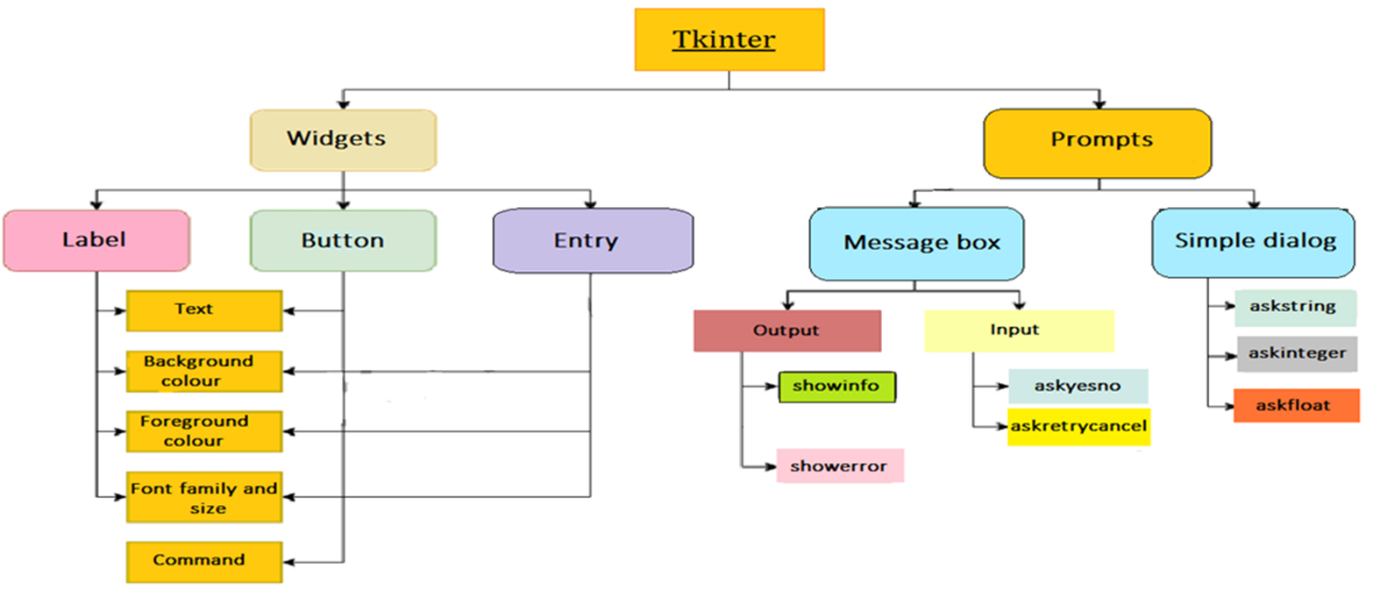
Some Library used in this project:

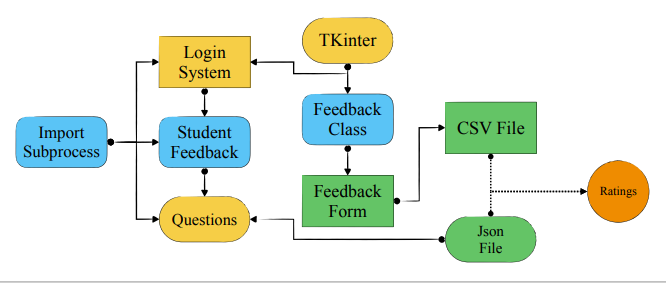
* from tkinter import\*: In order to work with a tkinter application, we have to install and import the tkinter library in our environment. Generally, we import the tkinter library in the environment by using from tkinter import \* command. The significance of "import \*" **represents all the functions and built-in modules in the tkinter library**.
* Tkinter Library: Tkinter is **the standard GUI library for Python**. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Creating a GUI application using Tkinter is an easy task.
* from tkinter import messagebox: **The tk messagebox module is used to display message boxes in your applications**. This module provides a number of functions that you can use to display an appropriate message. Some of these functions are showinfo, showwarning.
* class Login: In Python, object-oriented Programming (OOPs) is **a programming paradigm that uses objects and classes in programming**.
* Use of the code for adding background image self.bg=PhotoImage (file="C:\\Users\\adity\OneDrive \\Pictures\\Screenshots\\Screenshot\_20230214\_201241.png") for background image in project.
* Import json: In this file some questions are added and adding it to in python file.
* Import csv: The csv file operation is get operated through this library.
* Import subprocess: Through this all the separated file get joined in single file.
* In this project all uses of object-oriented concepts like inheritance, polymorphism, methods, constructor, destructor, encapsulation & abstraction.

After all of this the feedback form data will be stored in feedback.csv file and the json question file is saved on feedback\_text.csv and to taking together data will merge into single merged.csv file with header name like name, email, subject, faculty, comments, rating, content, instructor, assignment, overall column.

At last, the overall rating and average ratings of each subject can also be calculated with average operation of csv rating column then user can check all ratings and average rating of each subject.

Flowcharts





Code :

First python file is login system, the code is as follows:

from tkinter import \*

from PIL import ImageTk

from tkinter import messagebox

import subprocess

class Login:

    def \_\_init\_\_(self, root):

        self.root = root

        self.root.title("Login System")

        self.root.geometry("1199x600+90+45")

        self.bg=PhotoImage(file=r"C:\Users\adity\OneDrive\Pictures\Screenshots\Screenshot 2023-06-20 155153.png")

        self.bg\_image=Label(self.root,image=self.bg).place(x=0,y=0,relwidth=1,relheight=1)

        Frame\_login = Frame(self.root,bg="White")

        Frame\_login.place(x=330,y=150,width=500,height=400)

        title = Label(Frame\_login,text="IIIT Naya Raipur",font=("Goudy old style",15,"bold"),fg="black",bg="white").place(x=200,y=10)

        title = Label(Frame\_login,text="Login Here",font=("Impact",25,"bold"),fg="green",bg="white").place(x=200,y=40)

        subtitle = Label(Frame\_login,text="User Login Area",font=("Goudy old style",15,"bold"),fg="black",bg="white").place(x=90,y=100)

        lbl\_user = Label(Frame\_login,text="Username",font=("Goudy old style",15,"bold"),fg="brown",bg="white").place(x=90,y=140)

        self.username = Entry(Frame\_login,font=("Goudy old style",15,"bold"),bg="#E7E6E6")

        self.username.place(x=90,y=170,width=320,height=35)

        lbl\_password = Label(Frame\_login,text="Password",font=("Goudy old style",15,"bold"),fg="brown",bg="white").place(x=90,y=210)

        self.password = Entry(Frame\_login,font=("Goudy old style",15,"bold"),bg="#E7E6E6")

        self.password.place(x=90,y=240,width=320,height=35)

        forget = Button(Frame\_login,text="Forget Password?",bd=0,cursor="hand2",font=("Goudy old style",12),fg="blue",bg="white").place(x=90,y=280)

        submit = Button(Frame\_login,command=self.check\_function,text="Login?",bd=0,cursor="hand2",font=("Goudy old style",15),bg="blue",fg="white").place(x=90,y=320,height=40,width=180)

    def check\_function(self):

     if self.username.get() == "" or self.password.get() == "":

        messagebox.showerror("Error", "All fields are required", parent=self.root)

     elif self.username.get() != "IIIT\_NR" or self.password.get() != "iiitnr":

        messagebox.showerror("Error", "Invalid username or password", parent=self.root)

     else:

        messagebox.showinfo("Welcome", f"Welcome {self.password.get()}")

        root.destroy()  # Close the login window

      #   root\_feedback = Tk()  # Create a new window for the feedback form

        # feedback\_form = FeedbackForm(root\_feedback)

        # feedback\_form.run()

        subprocess.Popen(['python', r'C:\Users\adity\C Language\Python\Project in Python\sub2.py'])

root = Tk()

obj = Login(root)

root.mainloop()

In this code the import subprocess file has been seen which is sub2.py (Feedback Form), the code is as follows

import tkinter as tk

from tkinter import \*

from tkinter import ttk

from tkinter import messagebox

import subprocess

import csv

class FeedbackForm:

    def \_\_init\_\_(self, root):

        self.root = root

        self.root.title("Feedback")

        self.root.geometry("1199x600+90+45")

        self.root.configure(bg="light yellow")

        # self.bg=PhotoImage(file=r""C:\Users\adity\OneDrive\Pictures\Screenshots\Screenshot 2023-06-26 184845.png")

        # self.bg\_image=Label(self.root,image=self.bg).place(x=0,y=0,relwidth=1,relheight=1)

        # Frame\_login = Frame(self.root,bg="White")

        # Frame\_login.place(x=330,y=150,width=500,height=400)

        # Frame\_login = Frame(self.root,bg="White")

        # Frame\_login.place(x=330,y=150,width=500,height=400)

        self.create\_widgets()

    def create\_widgets(self):

        self.frame\_header = ttk.Frame(self.root)

        self.frame\_header.pack(pady=10)

        self.header\_label = ttk.Label(self.frame\_header, text='Student Feedback', foreground='blue',background='pink', font=('Arial', 40, 'bold'))

        self.header\_label.pack()

        self.frame\_content = ttk.Frame(self.root)

        self.frame\_content.pack(pady=10)

        self.frame\_content.configure(style="Content.TFrame")

        style = ttk.Style()

        style.configure("Header.TFrame", background="light pink")  # Set background color for frame\_header

        style.configure("Content.TFrame", background="light yellow")  # Set background color for frame\_content

        self.name\_label = ttk.Label(self.frame\_content, text='Name:', font=('Arial', 18))

        self.name\_label.grid(row=0, column=0, sticky='w')

        self.entry\_name = ttk.Entry(self.frame\_content, width=30, font=('Arial', 14))

        self.entry\_name.grid(row=0, column=1, padx=10, pady=5)

        self.email\_label = ttk.Label(self.frame\_content, text='Email:', font=('Arial', 18))

        self.email\_label.grid(row=1, column=0, sticky='w')

        self.entry\_email = ttk.Entry(self.frame\_content, width=30, font=('Arial', 14))

        self.entry\_email.grid(row=1, column=1, padx=10, pady=5)

        self.subject\_label = ttk.Label(self.frame\_content, text='Subject:', font=('Arial', 18))

        self.subject\_label.grid(row=2, column=0, sticky='w')

        subjects = ["OOP", "Quantum Mechanics", "Calculus", "IOT", "3D Printing","AIML","CDS","OT"]

        self.subject\_combobox = ttk.Combobox(self.frame\_content, values=subjects, font=('Arial', 14))

        self.subject\_combobox.grid(row=2, column=1, padx=10, pady=5)

        self.faculty\_label = ttk.Label(self.frame\_content, text='Faculty:', font=('Arial', 18))

        self.faculty\_label.grid(row=3, column=0, sticky='w')

        self.entry\_faculty = ttk.Entry(self.frame\_content, width=30, font=('Arial', 14))

        self.entry\_faculty.grid(row=3, column=1, padx=10, pady=5)

        self.comment\_label = ttk.Label(self.frame\_content, text='Comment:', font=('Arial', 18))

        self.comment\_label.grid(row=4, column=0, sticky='w')

        self.text\_comment = tk.Text(self.frame\_content, width=40, height=10, font=('Arial', 12))

        self.text\_comment.grid(row=4, column=1, padx=10, pady=5)

        self.rating\_label = ttk.Label(self.frame\_content, text='Rating:', font=('Arial', 18))

        self.rating\_label.grid(row=5, column=0, sticky='w')

        self.rating\_scale = Scale(self.frame\_content, from\_=1, to=5, orient=tk.HORIZONTAL, length=200, tickinterval=1, font=('Arial', 14))

        self.rating\_scale.grid(row=5, column=1, padx=10, pady=5)

        self.submit\_button = ttk.Button(self.frame\_content, text='Next', cursor="hand2", command=self.submit, width=10)

        self.submit\_button.grid(row=7, column=0, sticky='e', padx=10, pady=10)

        self.clear\_button = ttk.Button(self.frame\_content, text='Clear', cursor="hand2", command=self.clear, width=10)

        self.clear\_button.grid(row=7, column=1, sticky='w', padx=10, pady=10)

    def clear(self):

        self.entry\_name.delete(0, tk.END)

        self.entry\_email.delete(0, tk.END)

        self.text\_comment.delete("1.0", tk.END)

        self.rating\_scale.set(1)

    def save\_feedback\_to\_csv(self, name, email, subject, faculty, comment, rating):

        fieldnames = ["Name", "Email", "Subject", "Faculty", "Comment", "Rating"]

        with open('feedback.csv', 'a', newline='') as file:

            writer = csv.DictWriter(file, fieldnames=fieldnames)

            # Write column headers in the first row

            if file.tell() == 0:

                writer.writeheader()

            writer.writerow({

                "Name": name,

                "Email": email,

                "Subject": subject,

                "Faculty": faculty,

                "Comment": comment,

                "Rating": rating

            })

    def submit(self):

        name = self.entry\_name.get()

        email = self.entry\_email.get()

        subject = self.subject\_combobox.get()

        faculty = self.entry\_faculty.get()

        comment = self.text\_comment.get("1.0", tk.END)

        rating = self.rating\_scale.get()

        print('Name:', name)

        print('Email:', email)

        print('Subject:', subject)

        print('Faculty:', faculty)

        print('Comment:', comment)

        print('Rating:', rating)

        self.save\_feedback\_to\_csv(name, email, subject, faculty, comment, rating)

        # messagebox.showinfo(title='Submit', message='Thank you for your feedback. Your comment has been submitted.')

        subprocess.Popen(['python', r'C:\Users\adity\C Language\Python\Project in Python\am.py'])

        self.clear()

    def run(self):

        self.root.mainloop()

if \_\_name\_\_ == "\_\_main\_\_":

    root = tk.Tk()

    feedback\_form = FeedbackForm(root)

    feedback\_form.run()

Now the code of am.py (json questions file), the code is as follows:

import tkinter as tk

from tkinter import ttk

from tkinter import messagebox

import json

import csv

class Question:

    def \_\_init\_\_(self, question, options):

        self.question = question

        self.options = options

class FeedbackForm:

    def \_\_init\_\_(self, root, questions):

        self.root = root

        self.questions = questions

        self.answers = {}

        self.root.title("Feedback")

        self.create\_widgets()

    def create\_widgets(self):

        self.frame\_content = ttk.Frame(self.root)

        self.frame\_content.pack(pady=10)

        for i, question in enumerate(self.questions):

            label = ttk.Label(self.frame\_content, text=question.question, font=('Arial', 12))

            label.grid(row=i, column=0, sticky='w', padx=5, pady=5)

            options = ttk.Combobox(self.frame\_content, values=question.options, font=('Arial', 12))

            options.grid(row=i, column=1, padx=5, pady=5)

            options.bind("<<ComboboxSelected>>", lambda event, index=i, combo=options: self.handle\_selection(event, index, combo))

        self.submit\_button = ttk.Button(self.frame\_content, text='Submit', cursor="hand2", command=self.submit)

        self.submit\_button.grid(row=len(self.questions), column=0, sticky='e', padx=5, pady=10)

        self.clear\_button = ttk.Button(self.frame\_content, text='Clear', cursor="hand2", command=self.clear)

        self.clear\_button.grid(row=len(self.questions), column=1, sticky='w', padx=5, pady=10)

    def handle\_selection(self, event, index, combo):

        self.answers[index] = combo.get()

    def submit(self):

        fieldnames = ["Content", "Instructor", "Assignment", "Overall"]

        with open('feedback\_text.csv', 'a', newline='') as file:

            writer = csv.DictWriter(file, fieldnames=fieldnames)

            # Write column headers in the first row

            if file.tell() == 0:

                writer.writeheader()

            writer.writerow({

                "Content": self.answers.get(0, "Not answered"),

                "Instructor": self.answers.get(1, "Not answered"),

                "Assignment": self.answers.get(2, "Not answered"),

                "Overall": self.answers.get(3, "Not answered")

            })

        messagebox.showinfo(title='Submit', message='Thank you for your feedback. Your responses have been submitted.')

        self.clear()

    def clear(self):

        self.answers = {}

        for child in self.frame\_content.winfo\_children():

            if isinstance(child, ttk.Combobox):

                child.set("")

def load\_questions\_from\_json(file\_path):

    with open(file\_path) as file:

        data = json.load(file)

        feedback\_questions = data["feedback\_questions"]

        questions = []

        for question\_data in feedback\_questions:

            question = Question(question\_data["question"], question\_data["options"])

            questions.append(question)

        return questions

if \_\_name\_\_ == "\_\_main\_\_":

    root = tk.Tk()

    # Load questions from the JSON file

    questions = load\_questions\_from\_json(r"C:\Users\adity\C Language\Python\Project in Python\feed.json")

    # Create an instance of the FeedbackForm class

    feedback\_form = FeedbackForm(root, questions)

    feedback\_form.root.mainloop()

After all the code of json file (feed. json) is as follows:

{

    "feedback\_questions": [

      {

        "question": "How satisfied are you with the course content?",

        "options": ["Very Satisfied", "Satisfied", "Neutral", "Dissatisfied", "Very Dissatisfied"]

      },

      {

        "question": "Did the instructor effectively explain the concepts?",

        "options": ["Yes", "No", "Partially"]

      },

      {

        "question": "Were the assignments helpful in reinforcing the learning?",

        "options": ["Yes", "No", "Partially"]

      },

      {

        "question": "How would you rate the overall learning experience?",

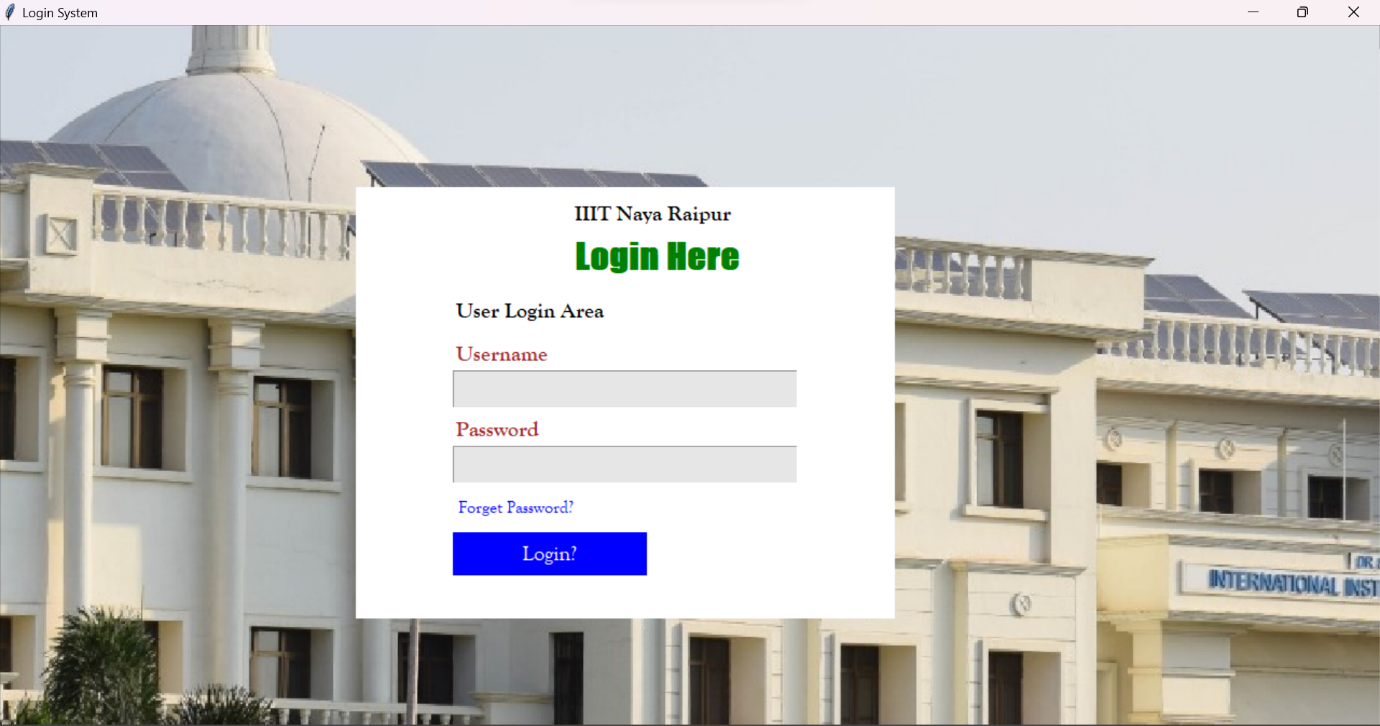
        "options": ["Excellent", "Good", "Fair", "Poor", "Very Poor"]

      }

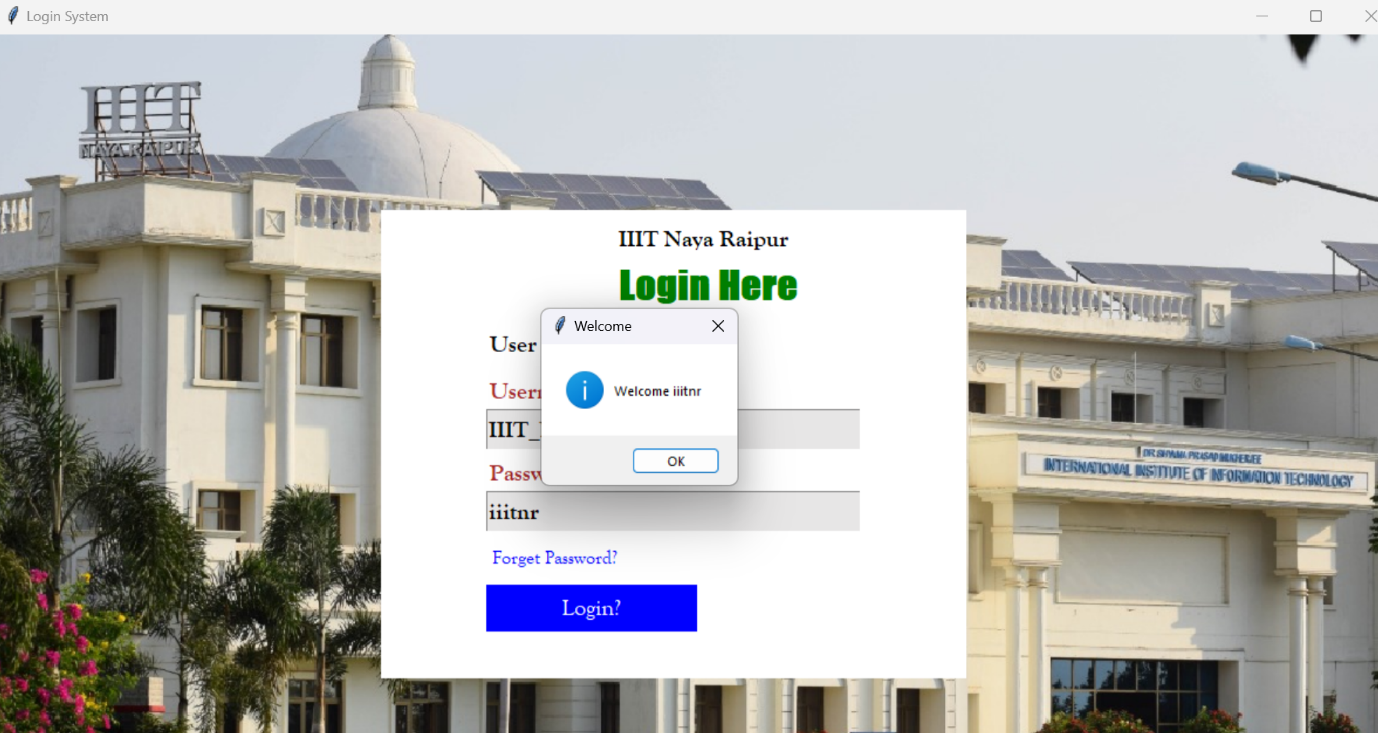
    ]

  }

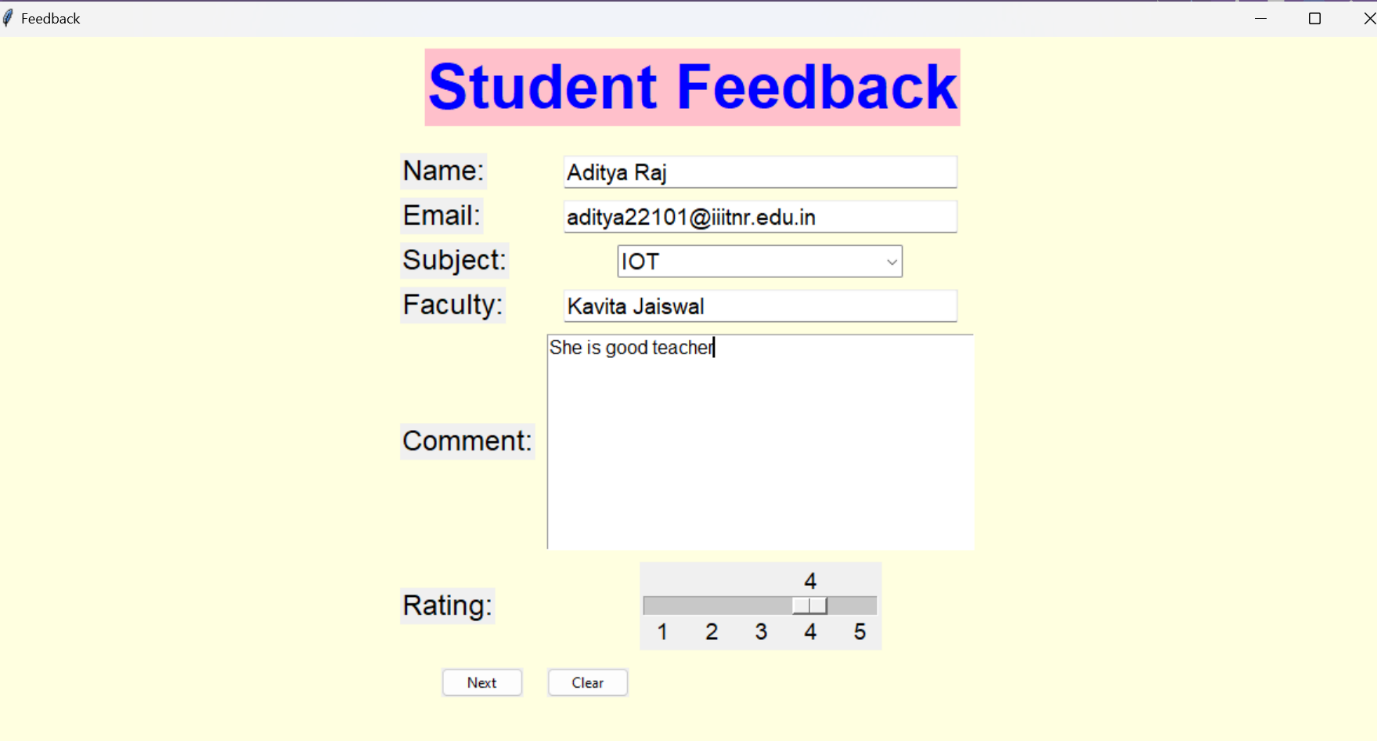
Result



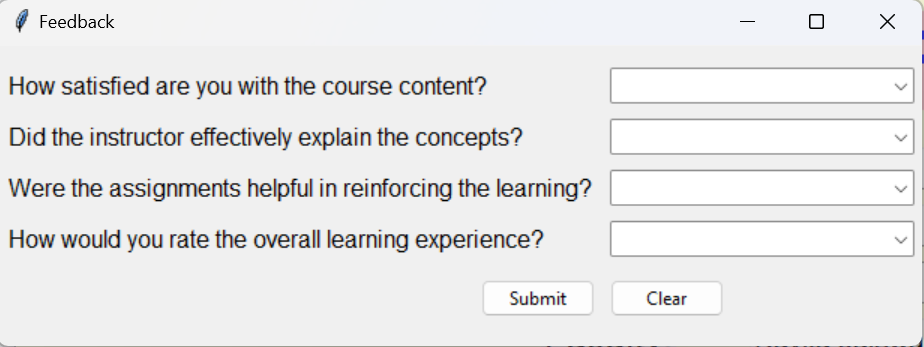
Then enter Username & Password



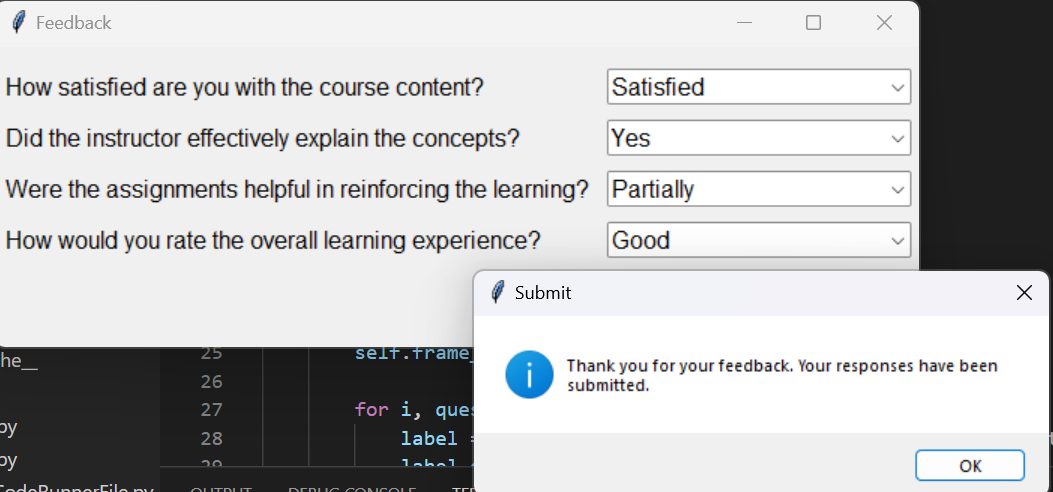
After clicking ok we have got the main part of our project that is feedback:



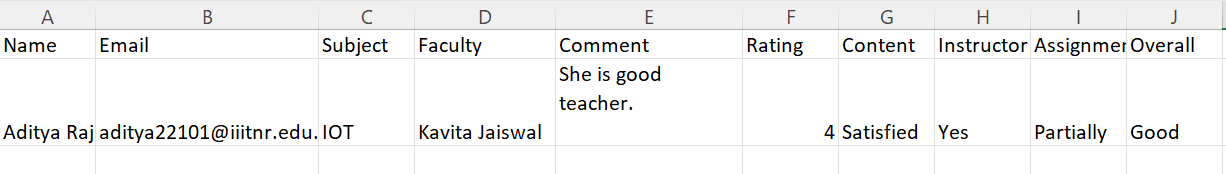
After clicking Next Button: A New window will open



Fill the form and click Submit:



After submitting, the data will store in respective csv file:



In this project the creating of python file which gives overall and average ratings of each subject that is as follows:

The code for that is:

import csv

def calculate\_average\_ratings():

    subject\_ratings = {}

    with open('feedback.csv', 'r') as file:

        reader = csv.DictReader(file)

        for row in reader:

            subject = row['Subject']

            rating = row['Rating']

            # Skip rows with missing or non-numeric ratings

            if rating is None or not rating.isdigit():

                continue

            rating = float(rating)

            if subject in subject\_ratings:

                subject\_ratings[subject].append(rating)

            else:

                subject\_ratings[subject] = [rating]

    for subject, ratings in subject\_ratings.items():

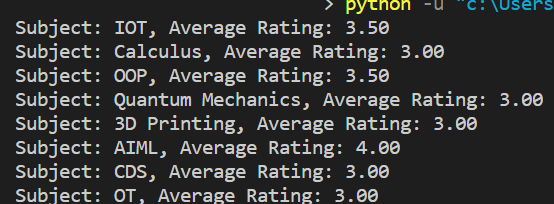
        average\_rating = sum(ratings) / len(ratings)

        print(f"Subject: {subject}, Average Rating: {average\_rating:.2f}")

if \_\_name\_\_ == "\_\_main\_\_":

    calculate\_average\_ratings()

The Output of this file is as:



Conclusion

Feedback is about listening actively, taking the time to analyse, and then thinking of the best possible solution to perform better. It provides positive criticism and allows to see what everyone can change to improve their focus and results. It brings people together and creates a healthy communication flow.

Now the era is technical we prefer smart work rather than hard work, also online system is so easy for accessing and giving the feedback.

PYTHON is easy computer language project is very important for learning and visualizing. In this project there is uses of simple tkinter library function, oops in python, function, conditional operator and make the success of project.

Feedback refers to information, opinions, or evaluations provided to someone regarding their performance, behaviour, or work. It is a response or reaction to a particular action, task, or project, with the intention of providing guidance, improvement, or reinforcement.

References:

<https://www.geeksforgeeks.org/python-gui-tkinter/>

<https://itsourcecode.com/free-projects/python-projects/student-feedback-system>

<https://chat.openai.com/c/c4ef8f20-b3f8-4c78-a07a-1d85f411216e>

THANK YOU