UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

NAAC GRADE A+
Accredited University

PROJECT

Name: Aditya kumar

Subject: PL/SQL LAB

Class: MCA(AIML)

Subject Code: 24CAP-602

UID: 24MCI10006

Date of Performance: 1-11-24

Section: 1-A

1.Aim

The aim of this project is to create a simple application with a user-friendly interface where users can register their college details and save this information in a MySQL database.

2.Objective / Problem Definition

The main goal of this application is to make the registration process easier for colleges and courses. It solves the problem of handling registration data manually by offering a digital way to enter, check, and store data efficiently.

3.Programming Languages Used

- Python: The main language used to develop the application.
- MySQL: Used to manage the database for storing registration data.

4.Block Diagram / Design Flow / Flow Chart

The application has a simple flow:

1. Start Application: Open the GUI.
2. Input Data: User enters college details.
3. Validation: Check if all fields are filled.
4. Database Operation: If everything is correct, save data in the database.
5. Feedback: Show a message about success or failure.
6. Reset Fields: Clear input fields for the next entry.

7. End Application: Close the application.

5.Algorithm or Pseudo Code

Step 1- **Import Required Libraries**:

- The program begins by importing necessary libraries: `tkinter` for GUI creation, `messagebox` for displaying messages, and `mysql.connector` for database interaction.

Step2- **Database Connection**:

- A connection to the MySQL database is established using `mysql.connector.connect()`, specifying the host, user, password, and database name. A cursor is created to execute SQL commands.

Step3- **Define the `submit()` Function:**

- This function handles the submission of the registration form.
- It retrieves values from the input fields using the `get()` method on the associated `StringVar` variables.
- It checks if all fields are filled. If not, an error message is shown.

Step4- **Data Insertion**:

- If all fields are valid:
  - The function constructs an SQL `INSERT` query to add the collected data into the `detail` table in the database. `o` The cursor executes this query, and changes are committed to the database.
  - A success message is displayed, and input fields are cleared for the next entry.
- If an error occurs during database operations, an error message is displayed to the user.

Step4- **Set Up the Main Window**:

- A main window (`k`) is created with a specified size and title.
- A title label is added to the top of the window.

Step5- **Create Labels for Input Fields**:

- Labels for each field (College Name, Course Name, etc.) are created and placed in a grid layout.

Step6- **Create Input Fields**:

- Entry widgets are created for each input field, linked to their corresponding `StringVar` variables to manage their values.
- These entry fields are positioned in the grid layout next to the labels.

Step7- **Create a Checkbox**:

- A checkbox is added to allow users to confirm that they want to submit the form.

Step8- **Create the Submit Button**:

- A button is created that, when clicked, will call the `submit()` function to process the data.

Step9- **Start the Main Loop**:

- Finally, the `mainloop()` method is called to run the application. This method waits for user interaction and keeps the window open.

6.Implementation

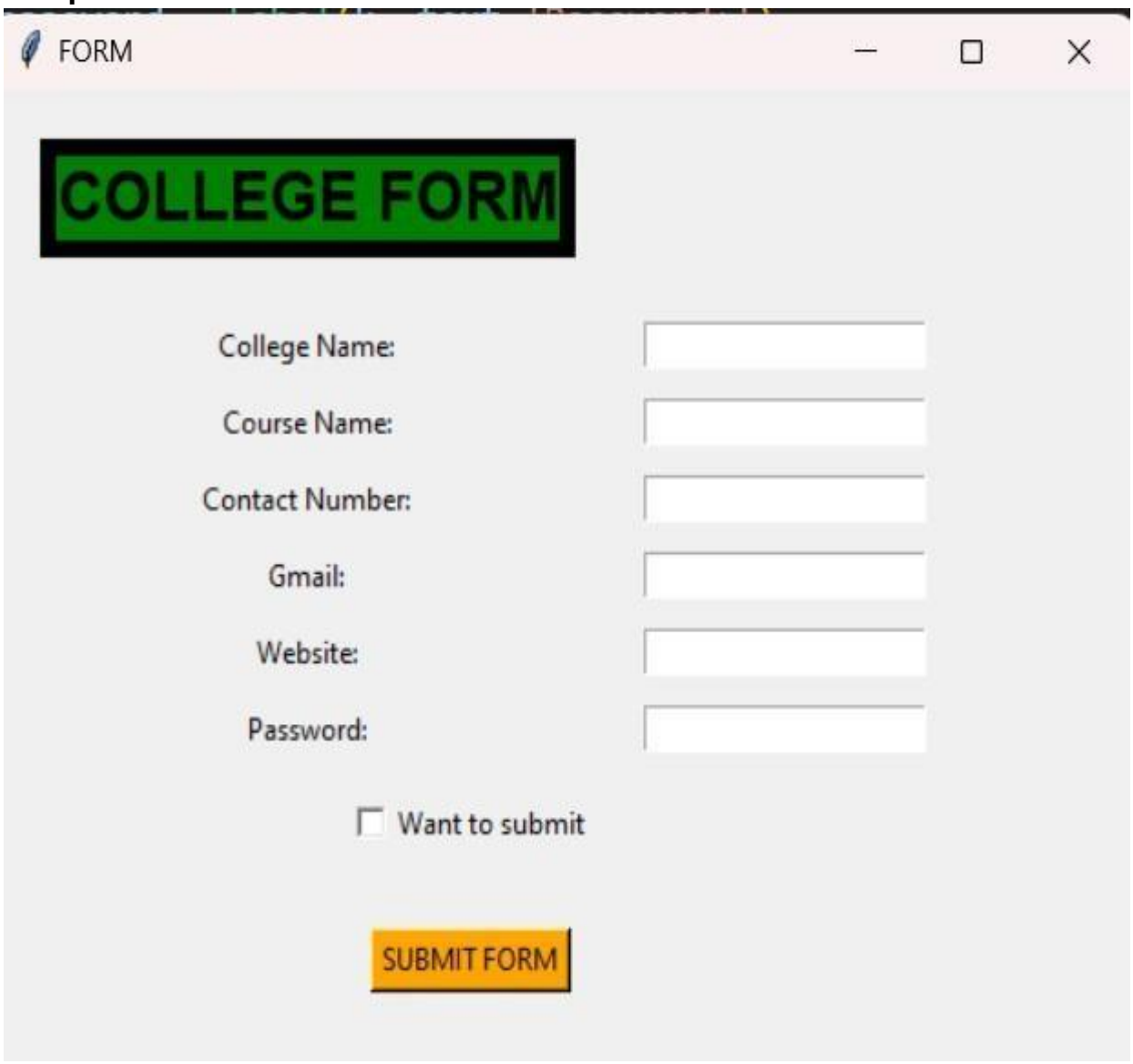
```
from tkinter import *
from tkinter import messagebox
import mysql.connector
con = mysql.connector.connect(
    host = "localhost",
    user = "root",
    passwd = "oracle",
    database = "project"
)
cursor = con.cursor()
def submit():
    collagename = variablecollagename.get()
    corsename = variablecoursename.get()
    contactnumber = variablecontactnumber.get()
    Gmail = variablegmailcollage.get()
    website = variablewebsite.get()
    password = variablepassword.get()
    if collagename and corsename and contactnumber and Gmail and website and password:
        try:
            query = "insert into detail values(%s, %s, %s, %s, %s, %s)"
            values = (collagename, corsename, contactnumber, Gmail, website, password)
            cursor.execute(query,values)
            con.commit()
            messagebox.showinfo("Success", "Registration successful!")
            variablecollagename.set('')
            variablecoursename.set('')
            variablecontactnumber.set('')
            variablegmailcollage.set('')
            variablewebsite.set('')
            variablepassword.set('')
```

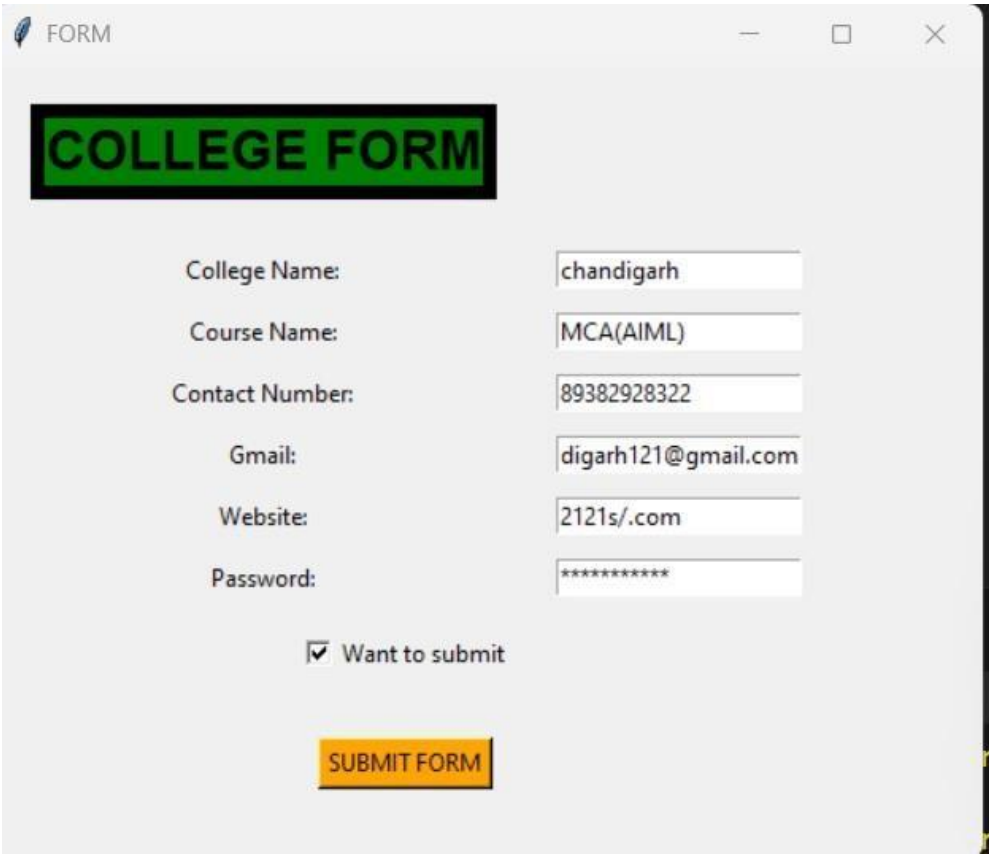
```
        except mysql.connector.Error as err:
            messagebox.showinfo("Database Error", f"Error : {err}")
    else:
        messagebox.showinfo("Input Error", "All fields are requied")
k = Tk()
k.geometry('500x400')
k.title('FORM')

labeloftitle = Label(k, text='COLLEGE FORM', font='comicsansms 20 bold', background='green', borderwidth=7, relief=SOLID)
labeloftitle.grid(row=0, column=0, padx=20, pady=20)

collagename = Label(k, text='College Name:')
corsename = Label(k, text='Course Name:')
contactnumber = Label(k, text='Contact Number:')
Gmail = Label(k, text='Gmail:')
website = Label(k, text='Website:')
password = Label(k, text='Password:')

collagename.grid(row=1, column=0, padx=10, pady=5)
corsename.grid(row=2, column=0, padx=10, pady=5)
contactnumber.grid(row=3, column=0, padx=10, pady=5)
Gmail.grid(row=4, column=0, padx=10, pady=5)
website.grid(row=5, column=0, padx=10, pady=5)
password.grid(row=6, column=0, padx=10, pady=5)

variablecollagename = StringVar()
variablecoursename = StringVar()
variablecontactnumber = StringVar()
```

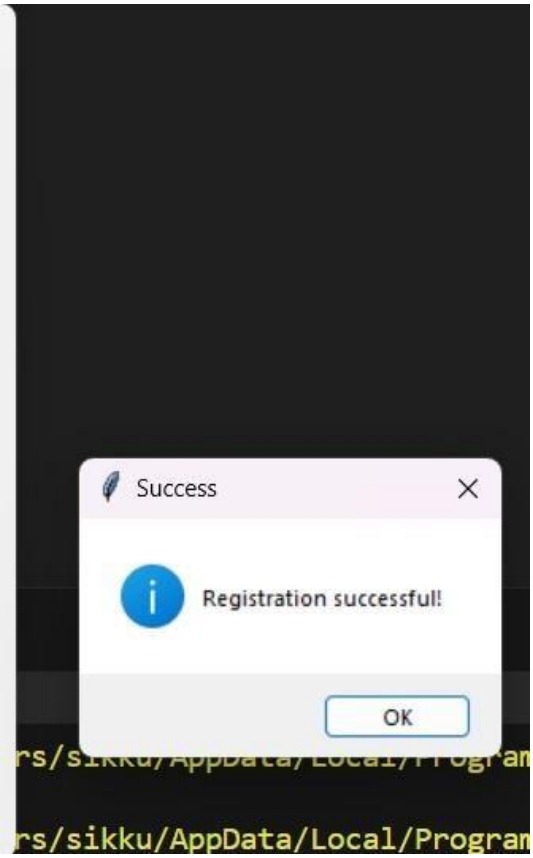
```
variablegmailcollage = StringVar()
variablewebsite = StringVar()
variablepassword = StringVar()
cheakboxvalue = IntVar()

enter_1 = Entry(k, textvariable=variablecollagename)
enter_2 = Entry(k, textvariable=variablecoursename)
enter_3 = Entry(k, textvariable=variablecontactnumber)
enter_4 = Entry(k, textvariable=variablegmailcollage)
enter_5 = Entry(k, textvariable=variablewebsite)
enter_6 = Entry(k, textvariable=variablepassword,show='*')

enter_1.grid(row=1, column=1, padx=10, pady=5)
enter_2.grid(row=2, column=1, padx=10, pady=5)
enter_3.grid(row=3, column=1, padx=10, pady=5)
enter_4.grid(row=4, column=1, padx=10, pady=5)
enter_5.grid(row=5, column=1, padx=10, pady=5)
enter_6.grid(row=6, column=1, padx=10, pady=5)

# Checkbox
checkbutton = Checkbutton(k, text='Want to submit', variable=cheakboxvalue)
checkbutton.grid(row=7, column=0, columnspan=2, pady=10)

# Submit button
button_form = Button(k, text='SUBMIT FORM', bg='orange', fg='black', command=submit)
button_form.grid(row=8, column=0, columnspan=2, pady=20)

k.mainloop()
```

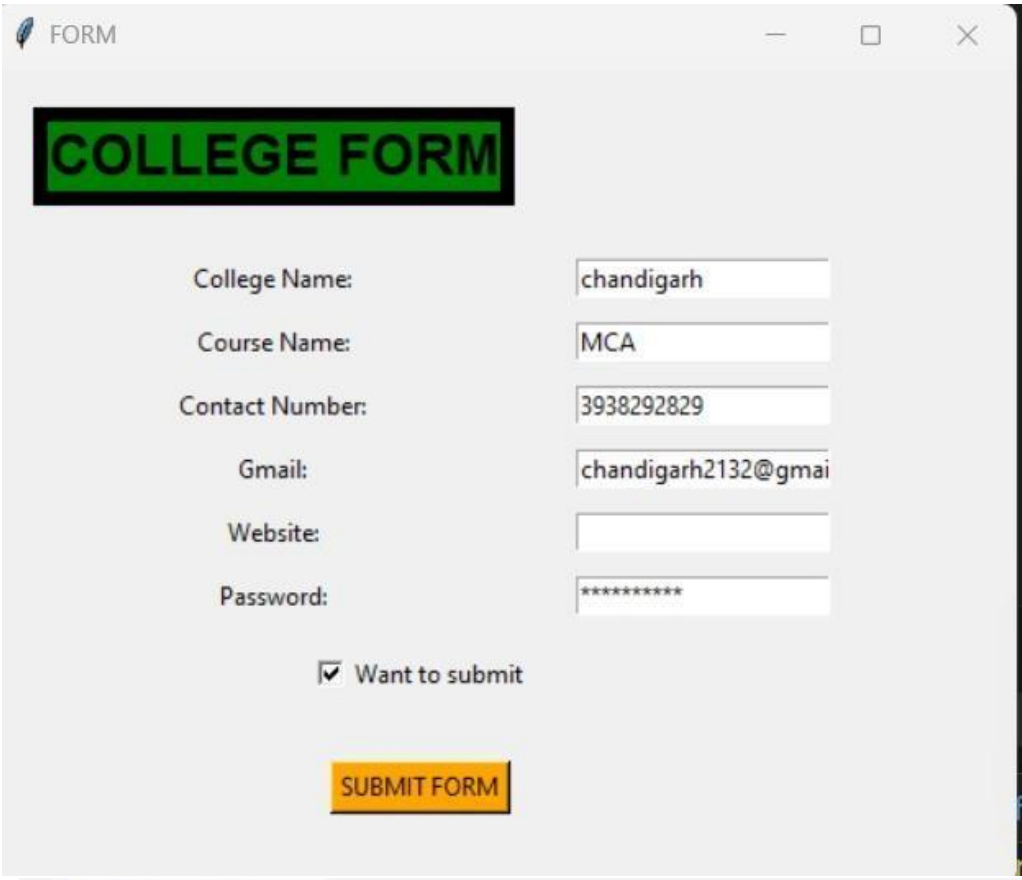
```
1 • use project;
2 • select * from detail;
```

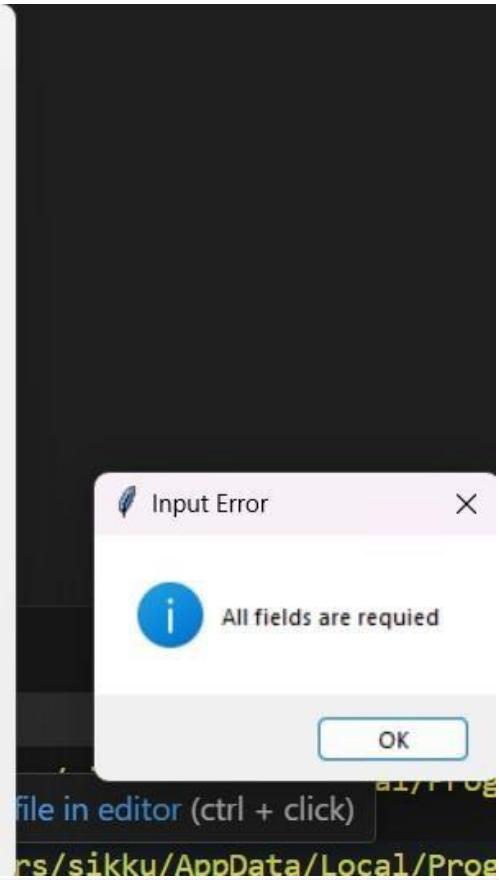
7-Output


FORM
COLLEGE FORM
College Name:
Course Name:
Contact Number:
Gmail:
Website:
Password:
Want to submit
SUBMIT FORM

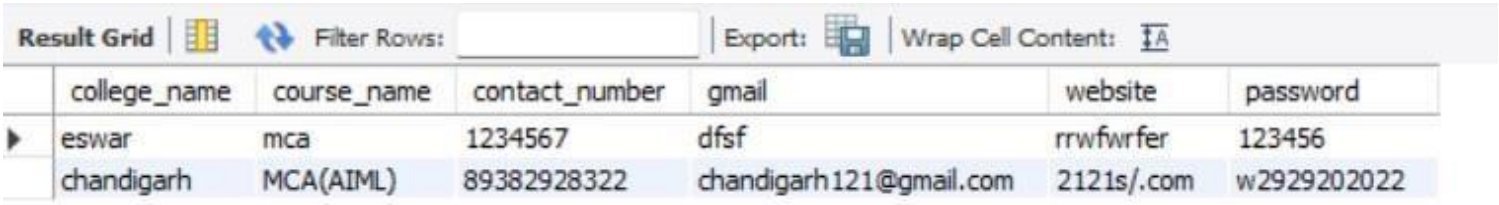








#	Time	Action	Message
1	13:56:37	use project	0 row(s) affected
2	13:56:37	select * from detail LIMIT 0, 1000	2 row(s) returned



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

college_name	course_name	contact_number	gmail	website	password
eswar	mca	1234567	dfsf	rrwfwrfer	123456
chandigarh	MCA(AIML)	89382928322	chandigarh121@gmail.com	2121s/.com	w2929202022

8.Conclusion

The College Registration Form application makes it easy to manage college registration data with a simple and clear interface. By using Tkinter for the GUI and MySQL for data storage, it ensures quick and secure data entry.

Future Framework

9.Future improvements could include:

- Adding security features, like encrypting passwords.
- Ensuring data is in the right format (e.g., proper email and phone number formats).
- Allowing users to view or edit existing entries.
- Enabling bulk registrations or file uploads for multiple entries.

10.Learning Outcomes

this project, I learned:

- How to develop GUIs using Tkinter in Python.
- How to manage a database using MySQL.
- The importance of data validation and error handling in applications.
- The overall process of software development from planning to implementation.