```
from tkinter import *
from tkinter import ttk
from types import CodeType
from PIL import Image,ImageTk
import os
import pickle
import mysql.connector  as sql
from tkinter import messagebox
import login
import dashboard
import create_account
import csv


def store_data(a,b,c,d):
    f=open("Credentials.csv","a",newline="")
    s=csv.writer(f,delimiter="-")
    s.writerow([a,b,c,d])
    f.close()



def user_interfun():

    def change_sign():
        create_account.signup()
        root.withdraw()

    def check():
        try:

            host = host_entry.get()
            port = port_entry.get()
            username = username_entry.get()
            password = password_entry.get()
            #database="cars"
            store_data(host,port,username,password)
            spec=sql.connect(host=host,user=username,password=password,port=port)
            mycur = spec.cursor()
            if spec.is_connected():
                messagebox.showinfo("Connected"," Database connected Sucessfully")
                #dashboard.dashboard()
        except BaseException:
            messagebox.showerror("User","User Doesnt exist")
        try:
            #spec2 = sql.connect(host="localhost", user=username,password=password,port=port)
            #mycur=spec2.cursor()
            #mycur.execute("Flush privileges")
            mycur.execute('create database sms;')
            messagebox.showinfo("Success", "Database \n cms\n created Successfully")
        except:
            mycur.execute('use sms;')
            messagebox.showerror("Error", "Database Creation Failed, \nDatabase May already exists!")

        try:
            mycur.execute('use sms;')

            mycur.execute('create table batch( batch_id int NOT NULL AUTO_INCREMENT, batch_name varchar(50) NOT NULL,batch_year varchar(10) NOT NULL , batch_intake varchar(20) NOT NULL,PRIMARY KEY ( batch_id), UNIQUE KEY (batch_name), reg_date date);')


            mycur.execute('create table course( course_id int NOT NULL AUTO_INCREMENT, course_name varchar(50) NOT NULL,course_duration varchar(10) NOT NULL, course_credit varchar(20) NOT NULL, reg_date date,PRIMARY KEY (course_id), UNIQUE KEY (course_name ));')


            mycur.execute('create table section( section_id int NOT NULL AUTO_INCREMENT, section_code varchar(50) NOT NULL,section_name varchar(50) NOT NULL, section_capacity int NOT NULL,PRIMARY KEY (
section_id), UNIQUE KEY (section_name), reg_date date);')


            mycur.execute ('create table department(department_id int NOT NULL AUTO_INCREMENT, department_code varchar(50) NOT NULL,department_name varchar(50) NOT NULL,PRIMARY KEY (department_id), UNIQUE KEY (department_name), reg_date date);')


            mycur.execute('create table students(student_id int NOT NULL AUTO_INCREMENT,'
                    'username varchar(254) NOT NULL, email varchar(50) NOT NULL,'
                    'password varchar(254) NOT NULL, f_name varchar(50) NOT NULL,'
                    'l_name varchar(50), dob varchar(20),gender varchar(10),'
                    'address varchar(30), contact_no int(13) NOT NULL,shift varchar(20) NOT NULL,'
                    'course_enrolled varchar(50) NOT NULL,batch varchar(50) NOT NULL,'
                    'section_enrolled varchar(20) NOT NULL, reg_date date, PRIMARY KEY (student_id),'
                    'FOREIGN KEY (course_enrolled) REFERENCES course (course_name),'
                    'FOREIGN KEY (batch) REFERENCES batch (batch_name),'
                    'CONSTRAINT UC_username UNIQUE (username,email));')


            mycur.execute('create table employees(employee_id int NOT NULL AUTO_INCREMENT,'
                    'username varchar(254) NOT NULL, email varchar(50) NOT NULL,'
                    'password varchar(254) NOT NULL, f_name varchar(50) NOT NULL,'
                    'l_name varchar(50), dob varchar(20),gender varchar(10),'
                    'address varchar(30), contact_no
 int(13) NOT NULL,job_type varchar(20) NOT NULL,'
                    'registered_as varchar(50) NOT
NULL,qualification varchar(50) NOT NULL,'
                    'department varchar(20) NOT NULL
, reg_date date, PRIMARY KEY (employee_id),'
                    'FOREIGN KEY (department)
REFERENCES department (department_name),'
                    'CONSTRAINT UC_username UNIQUE (
username,email));')

            spec.commit()
            messagebox.showinfo("Success", "All
Table are created successfully")
            spec.close()

            #dashboard.dashboard()
        except BaseException as msg:
            #f=open("log.txt","w")
            #f.write(msg)
            #f.close()
            messagebox.showerror("Error", f"Database
 Table Creation Failed {msg}")

    def login1():

        #try:
            host = host_entry.get()
            port = port_entry.get()
            username = username_entry.get()
            password = password_entry.get()
            #database="cars"

            spec=sql.connect(host=host,user=username
,password=password,port=port)
            if spec.is_connected():
                messagebox.showinfo("Connected","
Database connected Sucessfully")
                dashboard.dashboard()
                root.withdraw()
            spec.close()

            #dashboard.dashboard()
        #except BaseException as msg:
            #print(msg)
            #messagebox.showerror("User","User Doesnt exist")




    root = Tk()
    root.geometry("1067x600")
    root.configure(background="black")
    root.resizable(False, False)
    root.title("School Diaries")


    #background image
    bg = ImageTk.PhotoImage(file="files\Sublime Light1.jpg")
    lbl_bg_1 = Label(root,image=bg)
    lbl_bg_1.place(x=0,y=0,relwidth=1,relheight=1)

    #Labels
    host_label = Label(root, text="Host Name ", bg="white", fg="#4f4e4d",font=("yu gothic ui", 12, "bold"))
    host_label.place(x=675, y=115)
    host_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg="#6b6a69",font=("yu gothic ui semibold", 12))
    host_entry.insert(0, "localhost")
    host_entry.place(x=687, y=139, width=145)

    port_label = Label(root, text="Port ", bg="white", fg="#4f4e4d",font=("yu gothic ui", 13, "bold"))
    port_label.place(x=675, y=190)
    port_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg="#6b6a69",font=("yu gothic ui semibold", 12))
    port_entry.insert(0, "3307")
```

```python
    port_entry.place(x=690, y=213, width=145)

    username_label = Label(root, text="Username ", bg="white", fg="#4f4e4d",font=("yu gothic ui", 13, "bold"))
    username_label.place(x=675, y=265)
    username_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg="#6b6a69",font=("yu gothic ui semibold", 12))
    #username_entry.insert(0, "root")
    username_entry.place(x=687, y=287, width=145)

    password_label = Label(root, text="Password ", bg="white", fg="#4f4e4d",font=("yu gothic ui", 13, "bold"))
    password_label.place(x=675, y=338)
    password_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg="#6b6a69",font=("yu gothic ui semibold", 12))
    #password_entry.insert(0, "root")
    password_entry.place(x=687, y=361, width=145)

    #buttons
    submit = ImageTk.PhotoImage(file='Pics\connect_database.png')
    submit_button = Button(root, image=submit,font=("yu gothic ui", 13, "bold"), relief=FLAT, activebackground="white",borderwidth=0, background="white", cursor="hand2",command=check)
    submit_button.place(x=655, y=443)

    login_pic = ImageTk.PhotoImage(file='Pics\login.png')
    login_button_1 = Button(root, image=login_pic, font=("yu gothic ui", 13, "bold"), relief=FLAT, activebackground="white",borderwidth=0, background="white", cursor="hand2",command=login1)
    login_button_1.place(x=785, y=442)

    #sign_up = ImageTk.PhotoImage(file='Pics\\register.png')
    #sign_up_button = Button(root, image=sign_up, font=("yu gothic ui", 13, "bold"), relief=FLAT, activebackground="white",borderwidth=0, background="white", cursor="hand2",command=change_sign)
    #sign_up_button.place(x=785,y=490)

    root.mainloop()


if __name__=='__main__':
    user_interfun()
```