```python
from distutils import command
from tkinter import *
from tkinter import ttk
from PIL import Image,ImageTk
import os
import pickle
import mysql.connector  as sql
from tkinter import messagebox
from datetime import date
from datetime import time
from datetime import *
import requests
from bs4 import BeautifulSoup
import time
import user_inter
import csv
import course_screen

def regist_depart():
    def load_data():
        f=open("Credentials.csv","r")
        s=csv.reader(f,delimiter="-")
        d=[]
        for i in s:
            d.append(i)
        a=d[::-1]
        return (a[0])

    def back():
        root.destroy()

    def click_clear_button():
        department_code_entry.delete(0, END)
        department_name_entry.delete(0, END)

    def validation():
        """this will validate if the department code and name of entry fields are already in database table named
        department or not if return True, error message is thrown displaying department code/name already exists """
        try:
            #obj_section_database = Model_class.department_registration.GetDatabase('use cms;')
            #db_connection.create(obj_section_database.get_database())
            a=load_data()
            host=a[0]
            username = a[2]
            password = a[3]
            port=a[1]

            spec=sql.connect(host=host,user=username,password=password,port=port,database="sms")
            mycur=spec.cursor()
            query = "select * from department;"
            mycur.execute(query)
            data = mycur.fetchall()
            # print(data)

            code_list = []
            name_list = []

            for values in data:
                code_data_list = values[1]
                code_list.append(code_data_list)
                name_data_list = values[2]
                name_list.append(name_data_list)
                # print(code_list)
                print(name_list)

        except BaseException as msg:
            print(msg)

        if department_code_entry.get() == "" or department_name_entry.get() == "":
            messagebox.showwarning("Warning", "All Fields are Required\n Please fill all required fields")

        elif department_code_entry.get() in code_list:
            messagebox.showerror("Already Exists", f"{department_code_entry.get()} Department code Already Exists")
            # print(department_code_entry.get())
        elif department_name_entry.get() in name_list:
            messagebox.showerror("Already Exists", f"{department_name_entry.get()} Department name Already Exists")



        else:
            click_submit()

    def click_submit():
        """initialize when click submit button, which will take data from entry box
        and insert those data into student table after successful validation of those data"""
        try:
            #obj_department_database = Model_class.department_registration.GetDatabase('use cms;')
            #db_connection.create(obj_department_database.get_database())
            a=load_data()
            host=a[0]
            username = a[2]
            password = a[3]
            port=a[1]

            spec=sql.connect(host=host,user=username,password=password,port=port,database="sms")
            mycur=spec.cursor()

            #obj_department_database = Model_class.department_registration.DepartmentRegistration(
                #department_code_entry.get(), department_name_entry.get(), reg_date)

            query = f"insert into department (department_code,department_name,reg_date) values ('{department_code_entry.get()}','{department_name_entry.get()}','{reg_date}');"
            mycur.execute(query)
            spec.commit()
            #values = (obj_department_database.get_code(), obj_department_database.get_name(),
                        #obj_department_database.get_reg_date())
            # print(values)
            #db_connection.insert(query, values)
            # print(values)
            messagebox.showinfo("Success", f"Department added Successfully\n Section code={department_code_entry.get()},\n "
                                            f"Section name={department_name_entry.get()}")

        except BaseException as msg:
            print(msg)
            messagebox.showerror("Error", "There is some error Submitting Credentials ")




    root = Toplevel()
    root.title('DEPARTMENT REGISTRATION FORM - COLLEGE MANAGEMENT SYSTEM')
    root.geometry('1067x600')
    root.config(bg="#f29844")
    root.resizable(False,False)

    # ======================Backend connection============
    #db_connection = Backend.connection.DatabaseConnection()

    # creating frame for Register
    # img = img
    # dummylabel = Label(root, image=img)
    # dummylabel.place(x=30, y=30)

    reg_frame = Frame(root, bg="#ffffff", width=1000, height=560)
    reg_frame.place(x=30, y=30)




    heading = Label(reg_frame, text="Department Registration Form", font=('yu gothic ui', 20,
"bold"), bg="white",
                        fg='black',
                        bd=5,
                        relief=FLAT)
    heading.place(x=200, y=0, width=600)
    #slider()
    #heading_color()

    dept_frame = LabelFrame(reg_frame, text="Department Details", bg="white", fg="#4f4e4d",
height=380,
                                width=800, borderwidth=2.4,
                                font=("yu gothic ui", 13, "bold"))
    dept_frame.config(highlightbackground="red")
    dept_frame.place(x=100, y=90)

    # ====================================================================
    # ============================Key Bindings====================================
    # ====================================================================

    #root.bind("<Return>", click_enter_submit)

    # ====================================================================
    # ============================Department Code label=====================================
    # ====================================================================

    department_code_label = Label(dept_frame, text="Department Code ", bg="white", fg="#4f4e4d
",
                                    font=("yu gothic ui", 13, "bold"))
    department_code_label.place(x=160, y=80)

    department_code_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg="#
6b6a69",
                                    font=("yu gothic ui semibold", 12))
    department_code_entry.place(x=450, y=227, width=295)  # trebuchet ms

    department_code_line = Canvas(root, width=295, height=1.5, bg="#bdb9b1",
highlightthickness=0)
    department_code_line.place(x=450, y=249)

    # ========================================================================
    # ============================Department Name=====================================
    # ========================================================================

    department_name_label = Label(dept_frame, text="Department Name ", bg="white", fg="#4f4e4d
",
                                    font=("yu gothic ui", 13, "bold"))
    department_name_label.place(x=160, y=140)

    department_name_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg="#
6b6a69",
                                    font=("yu gothic ui semibold", 12))
    department_name_entry.place(x=450, y=287, width=295)  # trebuchet ms

    department_name_line = Canvas(root, width=295, height=1.5, bg="#bdb9b1",
highlightthickness=0)
    department_name_line.place(x=450, y=309)

    reg_date = time.strftime("%Y/%m/%d")



    # ========================================================================
    # ============================Register options====================================
    # ========================================================================
    submit_img = ImageTk.PhotoImage(file='Pics\\submit.png')
    submit = Button(dept_frame, image=submit_img,
                            font=("yu gothic ui", 13, "bold"), relief=FLAT, activebackground="
white"
                            , borderwidth=0, background="white", cursor="hand2",command=
validation)
    submit.image = submit_img
    submit.place(x=90, y=267)

    clear_img = ImageTk.PhotoImage(file='Pics\\clear.png')
    clear_button = Button(dept_frame, image=clear_img,
                                font=("yu gothic ui", 13, "bold"), relief=FLAT,
activebackground="white"
                                , borderwidth=0, background="white", cursor="hand2",
                                command=click_clear_button)
    clear_button.image = clear_img
    clear_button.place(x=250, y=270)

    back_img = ImageTk.PhotoImage(file='Pics\\back.png')
    back_button = Button(dept_frame, image=back_img,
                                font=("yu gothic ui", 13, "bold"), relief=FLAT,
activebackground="white"
                                , borderwidth=0, background="white", cursor="hand2",command=
back)
    back_button.image = back_img
    back_button.place(x=410, y=270)


    exit_img = ImageTk.PhotoImage(file='Pics\\exit.png')
    exit_button = Button(dept_frame, image=exit_img,
                                font=("yu gothic ui", 13, "bold"), relief=FLAT,
activebackground="white"
                                , borderwidth=0, background="white", cursor="hand2", command=
exit)
    exit_button.image = exit_img
    exit_button.place(x=570, y=270)


    #root.mainloop()


if __name__ =="__main__":
    regist_depart()
```