```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 sec_reg():

    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 click_clear_button():
            section_name_entry.delete(0, END)
            section_code_entry.delete(0, END)
            section_capacity_entry.delete(0,END)



    def click_enter_submit():
        validation()


    def validation():
        """this will validate if the section code and name of entry fields are already in database table named
        section or not if return True, error message is thrown displaying section code/name already exists"""
        try:
            #obj_section_database = Model_class.section_registration.GetDatabase('use cms;')
            #self.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 section;"
            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)

        except BaseException as msg:
            print(msg)

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

        elif section_code_entry.get() in code_list:
            messagebox.showerror("Already Exists", f"{section_code_entry.get()} Section code Already Exists")

        elif section_name_entry.get() in name_list:
            messagebox.showerror("Already Exists", f"{section_name_entry.get()} Section Already Exists")

        else:
            click_submit()

    def back():
        root.destroy()

    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_section_database = Model_class.section_registration.GetDatabase('use cms;')
            #self.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()

            #obj_section_database = Model_class.section_registration.SectionRegistration(self.section_code_entry.get(),
                                                                                          #self.section_name_entry.get(),
                                                                                          #self.section_capacity_entry.get(),
                                                                                          #self.reg_date)

            query = f"insert into section (section_code,section_name,section_capacity,reg_date) values ('{section_code_entry.get()}','{section_name_entry.get()}','{section_capacity_entry.get()}','{reg_date}');"
            mycur.execute(query)
            spec.commit()
            #values = (obj_section_database.get_code(),obj_section_database.get_name(),
                      #obj_section_database.get_capacity(),obj_section_database.get_reg_date
```

```
())
            # print(values)
            #self.db_connection.insert(query, values)
            # print(values)
            messagebox.showinfo("Success", f"Admin Data inserted Successfully\n Section code={
section_code_entry.get()},\n "
                                            f"Section name={section_name_entry.get()}")

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



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

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

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



    heading = Label(reg_frame, text="Section Registration Form", font=('yu gothic ui', 30, "
bold"), bg="white",
                           fg='black',
                           bd=5,
                           relief=FLAT)
    heading.place(x=200, y=0, width=600)


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

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

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

    # # ======================================================================
    # # ============================Section ID label==================================
    # # ======================================================================

    # ======================================================================
    # ==========================Subject Code =================================
    # ======================================================================

    section_code_label = Label(section_frame, text="Section Code ", bg="white", fg="#4f4e4d",
                                   font=("yu gothic ui", 13, "bold"))
    section_code_label.place(x=160, y=50)

    section_code_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg="#
6b6a69",
                                   font=("yu gothic ui semibold", 12))
```

```python
    section_code_entry.place(x=405, y=197, width=340)  # trebuchet ms

    section_code_line = Canvas(root, width=340, height=1.5, bg="#bdb9b1", highlightthickness=0
)
    section_code_line.place(x=405, y=219)

    # ========================================================================
    # ===========================Section Name=================================
    # ========================================================================

    section_name_label = Label(section_frame, text="Section Name ", bg="white", fg="#4f4e4d",
                                font=("yu gothic ui", 13, "bold"))
    section_name_label.place(x=160, y=100)

    section_name_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg="#
6b6a69",
                                font=("yu gothic ui semibold", 12))
    section_name_entry.place(x=410, y=247, width=335)  # trebuchet ms

    section_name_line = Canvas(root, width=335, height=1.5, bg="#bdb9b1", highlightthickness=0
)
    section_name_line.place(x=410, y=269)

    # ========================================================================
    # ===========================Section capacity=================================
    # ========================================================================
    root.option_add("*TCombobox*Listbox*Foreground", '#f29844')

    section_capacity_label = Label(section_frame, text="Section Capacity ", bg="white", fg="#
4f4e4d",
                                    font=("yu gothic ui", 13, "bold"))
    section_capacity_label.place(x=160, y=150)

    section_capacity_entry = Entry(root, highlightthickness=0, relief=FLAT, bg="white", fg="#
6b6a69",
                                    font=("yu gothic ui semibold", 12))
    section_capacity_entry.place(x=430, y=297, width=315)  # trebuchet ms

    section_capacity_line = Canvas(root, width=315, height=1.5, bg="#bdb9b1",
highlightthickness=0)
    section_capacity_line.place(x=430, y=319)

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



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


    clear_img = ImageTk.PhotoImage(file='Pics\\clear.png')
    clear_button = Button(section_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(section_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(section_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__":
    sec_reg()

```