Libraries:

import tkinter

from tkinter import ttk

from tkinter import messagebox

import openpyxl

import os

**The tkinter library is the standard library for Tcl/TK GUI toolkit. With TTK being a set of themed widgets which offers a more modern look. Tkinter messagbox allows you to use a message box to ask a yes/no, cancel, or ok input via a button input and display a message. Openpyxl is a library that prints, reads, and writes to excel files. Importing os provides functions for interacting with the operating system. The if not os.path.exists(filepath): #checks for a file in the computers directory, creates a workbook which uses openpyxl to write to the workbook.**

Get var and Enter Data:

**The enter\_data() function gathers user input from the form, validates required fields and creates an Excel File. If the file does not exist, the code creates one as well as displaying appropriate messages with messagebox library. The accepted = accept\_var.get() creates an accepted variable which stores the values in tkinter.StringVar. First\_name\_entry is a tkinter.Entry widget which is where the user types in their name in the GUI. The .get() method retrieves the text and stores it into the var. The enter\_data() function is called when the user clicks on the Enter Data button in the form.**

def enter\_data():

    accepted = accept\_var.get()

    if accepted == "Accepted":

        firstname = first\_name\_entry.get()

        lastname = last\_name\_entry.get()

        if firstname and lastname:

            street\_address = Street\_Address\_entry.get()

            city = city\_entry.get()

            state = state\_combobox.get()

            zip\_code = zip\_code\_entry.get()

            registration\_status = reg\_status\_var.get()

            numcourses = numcourses\_spinbox.get()

            numsemesters = numsemesters\_spinbox.get()

            selected\_education = [level for level, var in education\_vars.items() if var.get() == 1]

            education\_str = ", ".join(selected\_education) if selected\_education else "None"

Excel Workbook:

**This line of code uses the os library and the openpyxl library to operate in the operating system and to use/create an excel file.**

if not os.path.exists(filepath):

                workbook = openpyxl.Workbook()

                sheet = workbook.active

                sheet.append(["First Name", "Last Name", "Street Address", "City", "State", "Zip Code", "Registration Status", "# Courses", "# Semesters", "Education levels"])

                workbook.save(filepath)

workbook = openpyxl.load\_workbook(filepath)

            sheet = workbook.active

            sheet.append([firstname, lastname, street\_address, city, state, zip\_code, registration\_status, numcourses, numsemesters, education\_str])

            workbook.save(filepath)

            messagebox.showinfo("Success", "Data entered successfully!")

**After creating the workbook in the first section of code and writing labels into the excel document using sheet.append. The second section then uses sheet.append to enter the text got from .get() into the excel sheet. Saving the workbook with workbook.save. and displaying a messagebox.showinfo showing text that reads “Success”, “Data entered successfully!”.**

Filepath**:**

filepath = "data.xlsx"

**This is a filepath where the excel file is created and stored in their directory.**

Reset Fields:

**The reset\_fields() function clears all inputs after the form is submitted. This function also resets the dropdowns, checkboxes, and spinboxes to their original values. Places the cursor onto the first name field.**

def reset\_fields():

    """Clears all input fields and resets the cursor to 'First Name'."""

    first\_name\_entry.delete(0, tkinter.END)

    last\_name\_entry.delete(0, tkinter.END)

    Street\_Address\_entry.delete(0, tkinter.END)

    city\_entry.delete(0, tkinter.END)

    zip\_code\_entry.delete(0, tkinter.END)

    state\_combobox.set("Select One")

    reg\_status\_var.set("Not Registered")

    numcourses\_spinbox.delete(0, tkinter.END)

    numcourses\_spinbox.insert(0, "0")

    numsemesters\_spinbox.delete(0, tkinter.END)

    numsemesters\_spinbox.insert(0, "0")

    for var in education\_vars.values():

        var.set(0)

    accept\_var.set("Not Accepted")

    first\_name\_entry.focus()

Window and Frame:

**Sets up a basic form interface window and frame.**

window = tkinter.Tk()

window.title("Data Entry Form")

frame = tkinter.Frame(window)

frame.pack()

#Saving User Info:

**User\_info\_frame user the tkinter library fram and assignes text to the frame. The .grid places the frame on the grid via the row, column, padx, pady, and sticky=”ew”. In this section Labels are created, placed on the grid, and given entry widgets, which are placed on the grid. The user information collected includes first name, last name, address, city, state, and ZIP code. Includes the state Combobox which uses a themed tkinter library.**

user\_info\_frame = tkinter.LabelFrame(frame, text="User Information")

user\_info\_frame.grid(row=0, column=0, padx=10, pady=5, sticky="ew")

first\_name\_label = tkinter.Label(user\_info\_frame, text="First Name")

first\_name\_label.grid(row=0, column=0, sticky="w")

last\_name\_label = tkinter.Label(user\_info\_frame, text="Last Name")

last\_name\_label.grid(row=0, column=1, sticky="w")

first\_name\_entry = tkinter.Entry(user\_info\_frame, width=18)

last\_name\_entry = tkinter.Entry(user\_info\_frame, width=35)

first\_name\_entry.grid(row=1, column=0, padx=5, pady=5, sticky="w")

last\_name\_entry.grid(row=1, column=1, padx=5, pady=5, sticky="w")

Street\_Address\_label = tkinter.Label(user\_info\_frame, text="Street Address")

Street\_Address\_label.grid(row=2, column=0, columnspan=2, sticky="w")

Street\_Address\_entry = ttk.Entry(user\_info\_frame, width=60)

Street\_Address\_entry.grid(row=3, column=0, columnspan=2, padx=5, pady=5, sticky="w")

city\_label = tkinter.Label(user\_info\_frame, text="City")

city\_label.grid(row=4, column=0, sticky="w")

state\_label = tkinter.Label(user\_info\_frame, text="State")

state\_label.grid(row=4, column=1, sticky="w")

zip\_code\_label = tkinter.Label(user\_info\_frame, text="Zip Code")

zip\_code\_label.grid(row=4, column=2, sticky="w")

city\_entry = tkinter.Entry(user\_info\_frame, width=25)

city\_entry.grid(row=5, column=0, padx=5, pady=5, sticky="w")

state\_combobox = ttk.Combobox(user\_info\_frame, width=25, values=[

    "Alaska", "Alabama", "Arakansas", "Arizona", "California", "Colorado", "Connecticut", "Delaware", "Florida", "Georgia", "Hawaii", "Iowa",

    "Idaho", "Ilinois", "Indiana", "Kansas", "Kentucky", "Louisiana", "Massachussets", "Maryland", "Maine", "Michigan", "Minnesota", "Missouri",

    "Mississippi", "Montana", "North Carolina", "North Dakota", "Nebraska", "New Hampshire", "New Jersey", "New Mexico", "Nevada", "New York", "Ohio", "Oklahoma",

    "Oregon", "Pennsylvania", "Rhode Island", "South Carolina", "South Dakota", "Tennessee", "Texas", "Utah", "Virginia", "Vermont", "Washington", "Wisconsin",

    "West Virgina", "Wyoming"])

state\_combobox.grid(row=5, column=1, padx=5, pady=5, sticky="w")

state\_combobox.set("Select One")

zip\_code\_entry = tkinter.Entry(user\_info\_frame, width=10)

zip\_code\_entry.grid(row=5, column=2, padx=5, pady=5)

#Saving Course Info

**Creates a new courses\_frame to separate the widgets contained from the user\_info\_frame.**

courses\_frame = tkinter.LabelFrame(frame, text="Courses Information")

courses\_frame.grid(row=1, column=0, padx=10, pady=5, sticky="ew")

#Education Level Checkboxes

**Uses the courses\_frame to create the education checkboxes which includes the education\_levels of HS, AAS, BS, MS, PhD. The education\_vars = {} includes an empty dictionary that stores IntVar() variables. The for level in education\_levels: code includes a loop to create checkboxes, each checkbox is stored with an IntVar(), which is either 0 for unchecked or 1 for a checked box. A chk.pack is then used to arrange the checkboxes horizontally.**

education\_label = tkinter.Label(courses\_frame, text="Education Level")

education\_label.grid(row=0, column=0, sticky="w")

education\_levels = ["HS", "AAS", "BS", "MS", "PhD"]

education\_vars = {}

education\_frame = tkinter.Frame(courses\_frame)

education\_frame.grid(row=0, column=1, columnspan=5, sticky="w")

for level in education\_levels:

    education\_vars[level] = tkinter.IntVar()  # Create an IntVar for each checkbox

    chk = tkinter.Checkbutton(education\_frame, text=level, variable=education\_vars[level])

    chk.pack(side="left", padx=5)  # Use pack to arrange the checkboxes

#Registration Status

**Creates a checkbutton in a new registration\_frame which separates the code within it from the other frames on the form.**

registration\_frame = ttk.LabelFrame(frame, text="Registration Status")

registration\_frame.grid(row=2, column=0, padx=10, pady=5, sticky="ew")

reg\_status\_var = tkinter.StringVar(value="Not Registered")

registered\_check = tkinter.Checkbutton(registration\_frame, text="Currently Registered", variable=reg\_status\_var, onvalue="Registered", offvalue="Not Registered")

registered\_check.grid(row=0, column=0, sticky="w")

#Completed Courses and Semesters

**Creates a numcourses and a numsemesters spinbox in the registration\_frame. The numcourses goes from 0 to 100 and the numsemesters goes from 0 to 20.**

numcourses\_label = tkinter.Label(registration\_frame, text="# Completed Courses")

numcourses\_label.grid(row=0, column=1, padx=5)

numcourses\_spinbox = tkinter.Spinbox(registration\_frame, from\_=0, to=100, width=5)

numcourses\_spinbox.grid(row=0, column=2, padx=5)

numsemesters\_label = tkinter.Label(registration\_frame, text="# Semesters")

numsemesters\_label.grid(row=0, column=3, padx=5)

numsemesters\_spinbox = tkinter.Spinbox(registration\_frame, from\_=0, to=20, width=5)

numsemesters\_spinbox.grid(row=0, column=4, padx=5)

#Terms and Conditions

**Another frame is created called terms\_frame to separate terms and conditions in the GUI and grid. Creates a checkbutton which has the variable=accept\_var which triggers the Accepted or Not accepted in the enter\_data() code.**

terms\_frame = tkinter.LabelFrame(frame, text="Terms & Conditions", padx=5, pady=2)

terms\_frame.grid(row=3, column=0, padx=10, pady=5, sticky="w")

accept\_var = tkinter.StringVar(value="Not Accepted")

terms\_check = tkinter.Checkbutton(terms\_frame, text="I accept the terms and conditions", variable=accept\_var, onvalue="Accepted", offvalue="Not Accepted")

terms\_check.pack(anchor="w")

#Submit Button

**Creates a tkinter.button in a new button\_frame which separates the button. The command=enter\_data activates the defined function. Window.mainloop() keeps the python program operational until the loop is closed.**

button\_frame = tkinter.Frame(frame)

button\_frame.grid(row=4, column=0, padx=10, pady=5, sticky="w")

button = tkinter.Button(button\_frame, text="Enter Data", command=enter\_data, width=8, height=1)

button.pack()

window.mainloop()