SBOA SCHOOL & JUNIOR COLLEGE

Anna Nagar West Extension, Chennai -600101

COMPUTER SCIENCE PROJECT REPORT

2024 - 2025

Topic :

MANAGEMENT SOFTWARE FOR ANIMAL RESCUE NGO

Project Done By :

1. S. YUVAN
2. TARUN SURYA
3. R. NISHANTHRAJAN

Table of Contents

S.NO	TOPIC	PAGE NUMBER
1	Acknowledgement	
2	Objective	
3	Source Code	
4	Output Screen	
5	Bibliography	
6		

Acknowledgment

I would like to state that this project is my original work and would like to thank all those people who have wholeheartedly extended their cooperation and guidance for making it possible to complete this project on time.

My sincere gratitude to Our School Management for providing us the best infrastructure and all the required resources. My special thanks to school Principal **Mrs. Sharahda Ramamurthy** and Vice Principal **Mrs. Mahalakshmi** for their unconditional support. Many many thanks to my Computer Science teacher **Mrs. K. Bhavani** for her valuable guidance and support. I would also like to thank my family members and friends for their cooperation in completing this project within stipulated time.

Objective

The objective of the project is to create a useful program for an imaginary Animal Rescue NGO.

The NGO which, obviously, rescues animals will have the need to store some stuff. This application will help the NGO to store the required data.

Using this application, the NGO can store information about:

- The Animals Rescued
- Details of the Volunteers
- Their Expenses
- The Donation Amount Received

Source Code

```
#Importing the required modules
import tkinter as tk
from tkinter import ttk, messagebox
import csv

# File paths for CSV files
animal_file = "animals_rescued.csv"
expense_file = "expenses.csv"
donation_file = "donations.csv"
volunteer_file = "volunteers.csv"

# Utility functions to handle CSV operations (will be used for any
reading / write operation)
def read_csv(file_path, headers):
    data = []
    try:
        with open(file_path, mode='r', newline='') as file:
            reader = csv.reader(file)
            for row in reader:
                data.append(row)
    except FileNotFoundError:
        with open(file_path, mode='w', newline='') as file:
            writer = csv.writer(file)
            writer.writerow(headers)
    return data

def write_csv(file_path, data):
    with open(file_path, mode='w', newline='') as file:
        writer = csv.writer(file)
        for row in data:
            writer.writerow(row)
```

```
#Adding headers inside the CSV file
animals_rescued = read_csv(animal_file, ["ID", "Name", "Age",
"Rescue Date", "Status"])
expenses = read_csv(expense_file, ["ID", "Category", "Amount"])
donations = read_csv(donation_file, ["ID", "Donor", "Amount"])
volunteers = read_csv(volunteer_file, ["ID", "Name", "Age",
"Role", "Contact", "Availability"])

# Function to refresh table data
def refresh_table(tree, data):
    for row in tree.get_children():
        tree.delete(row)
    for row in data[1:]:
        tree.insert("", "end", values=row)

# Function to add new data
def add_data_form(data, file_path, columns, tree):
    form_window = tk.Toplevel(root)
    form_window.title("Add New Data")
    form_window.geometry("400x400")
    form_window.configure(bg='#333333')

    tk.Label(form_window, text="Add New Data",
font=("Elephant", 20), bg='#333333', fg='white').pack(pady=10)

    inputs = []
    for column in columns:
        tk.Label(form_window, text=column, bg='#333333',
fg='white').pack(pady=5)
        entry = tk.Entry(form_window, width=30)
```

```
        entry.pack()
        inputs.append(entry)

    def save_data():
        new_row = []
        for entry in inputs :
            new_row.append(entry.get())
        if "" in new_row: # i.e., if the field is left empty
            messagebox.showerror("Error", "All fields are required!")
            return
        new_row[0] = str(len(data))
        data.append(new_row)
        write_csv(file_path, data)
        refresh_table(tree, data)
        messagebox.showinfo("Success", "Data added successfully!")

    tk.Button(form_window, text="Save", width=25, height=2,
command=save_data, bg='#FFFFFF', font=("Consolas", 12),
fg='#333333').pack(pady=20)

# Function to delete data
def delete_data_form(data, file_path, tree):
    delete_window = tk.Toplevel(root)
    delete_window.title("Delete Data")
    delete_window.geometry("400x200")
    delete_window.configure(bg='#333333')

    tk.Label(delete_window, text="Delete Data", font=("Elephant",
20), bg='#333333', fg='white').pack(pady=10)
    tk.Label(delete_window, text="Enter the ID to delete:",
bg='#333333', fg='white').pack(pady=5)
```

```
    id_entry = tk.Entry(delete_window, width=30)
    id_entry.pack(pady=5)

    def delete_data():
        id_to_delete = id_entry.get()
        if not id_to_delete:
            messagebox.showerror("Error", "ID is required!")
            return

        updated_data = [data[0]]  # Start with the header row
        deleted = False
        for row in data[1:]:
            if row[0] != id_to_delete:
                updated_data.append(row)
            else:
                deleted = True

        if not deleted:
            messagebox.showerror("Error", f"No record found with ID:
{id_to_delete}")
        else:
            write_csv(file_path, updated_data)
            data[:] = updated_data
            refresh_table(tree, updated_data)
            messagebox.showinfo("Success", f"Record with ID
{id_to_delete} deleted!")


    tk.Button(delete_window, text="Delete", width=25, height=2,
command=delete_data, bg='#FFFFFF', font=("Consolas", 12),
```

```
fg='#333333').pack(pady=20)


# Function to create the main menu
def main_menu():
    for widget in root.winfo_children():
        widget.destroy()

    tk.Label(root, text="NGO Management System",
font=("Elephant", 30), bg='#333333', fg='white').pack(pady=20)

    tk.Button(root, text="Info about Animals Rescued", width=30,
height=3, font=("Consolas", 14), command=open_animal_info,
bg='#fed700', fg='#333333').pack(pady=10)

    tk.Button(root, text="Expenses", width=30, height=3,
font=("Consolas", 14), command=open_expenses, bg='#fed700',
fg='#333333').pack(pady=10)

    tk.Button(root, text="Donation Amount Received", width=30,
height=3, font=("Consolas", 14), command=open_donations,
bg='#fed700', fg='#333333').pack(pady=10)

    tk.Button(root, text="Volunteer Details", width=30, height=3,
font=("Consolas", 14), command=open_volunteer_details,
bg='#fed700', fg='#333333').pack(pady=10)

# Function to open a new window for data display
def create_new_window(title, data, columns, file_path):
    new_window = tk.Toplevel(root)
    new_window.title(title)
```

```
    new_window.geometry("600x400")
    new_window.configure(bg='#333333')

    tk.Label(new_window, text=title, font=("Elephant", 20),
bg='#333333', fg='white').pack(pady=10)

    tree = ttk.Treeview(new_window, columns=columns,
show="headings")
    for col in columns:
        tree.heading(col, text=col)

    refresh_table(tree, data)
    tree.pack(pady=20)

    tk.Button(new_window, text="Add", width=25, height=2,
command=lambda: add_data_form(data, file_path, columns,
tree), bg='#FFFFFF', font=("Consolas", 12),
fg='#333333').pack(pady=5)

    tk.Button(new_window, text="Delete", width=25, height=2,
command=lambda: delete_data_form(data, file_path, tree),
bg='#FFFFFF', font=("Consolas", 12),
fg='#333333').pack(pady=5)
    tk.Button(new_window, text="Close", width=25, height=2,
command=new_window.destroy, bg='#FFFFFF',
font=("Consolas", 12), fg='#333333').pack(pady=10)

def open_animal_info():
    create_new_window("Animals Rescued", animals_rescued,
["ID", "Name", "Age", "Rescue Date", "Status"], animal_file)
```

```
def open_expenses():
    create_new_window("Expenses", expenses, ["ID", "Category",
"Amount"], expense_file)

def open_donations():
    create_new_window("Donations", donations, ["ID", "Donor",
"Amount"], donation_file)

def open_volunteer_details():
    create_new_window("Volunteer Details", volunteers, ["ID",
"Name", "Age", "Role", "Contact", "Availability"], volunteer_file)

# Main application setup
root = tk.Tk()
root.title("NGO Management System")
root.geometry("1080x720")
root.configure(bg='#333333')

main_menu()
root.mainloop()
```

Output Screen

Main Menu:

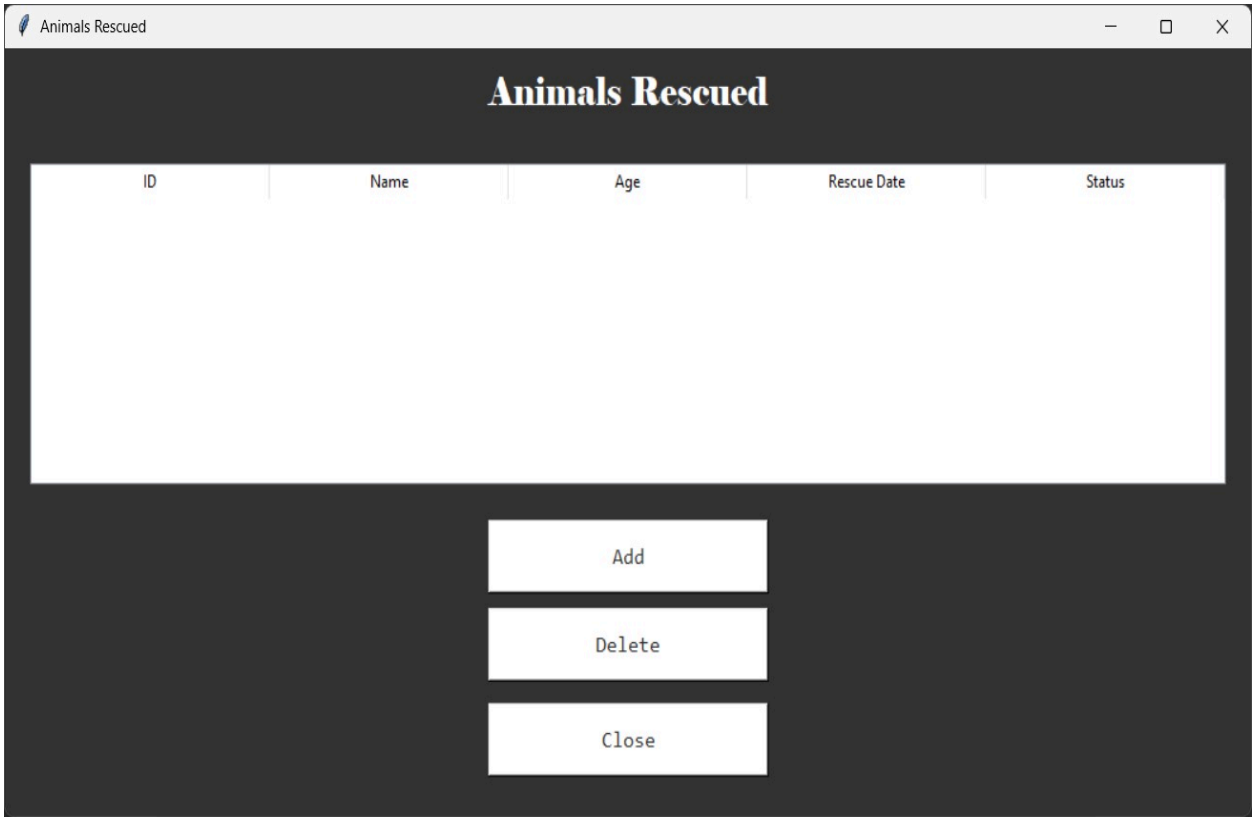
- This is the Main Menu window that the user sees:

- Now the user is met with 4 options, which they can choose to add/view/delete data for 4 different scenarios, namely:
    1. Info about Animals Rescued
    2. Expenses
    3. Donation Amount Received
    4. Volunteer Details
- The user can do 2 operations in all four options. These are adding and deleting data.

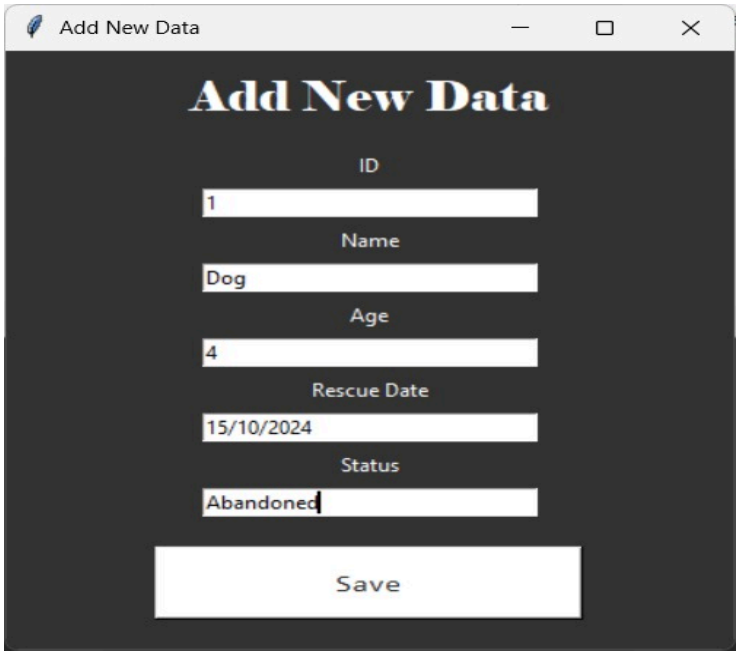
Adding Data:

- At Info about Animals Rescued GUI :

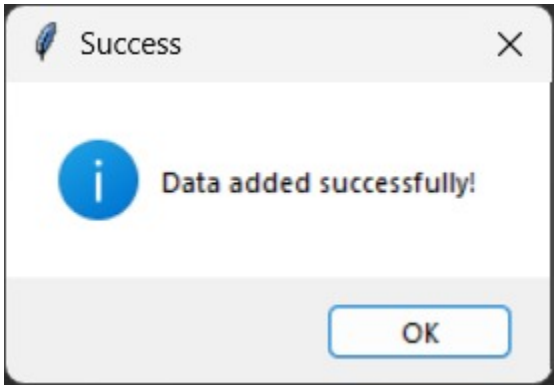
When we press the 1st Option, we see the window shown below



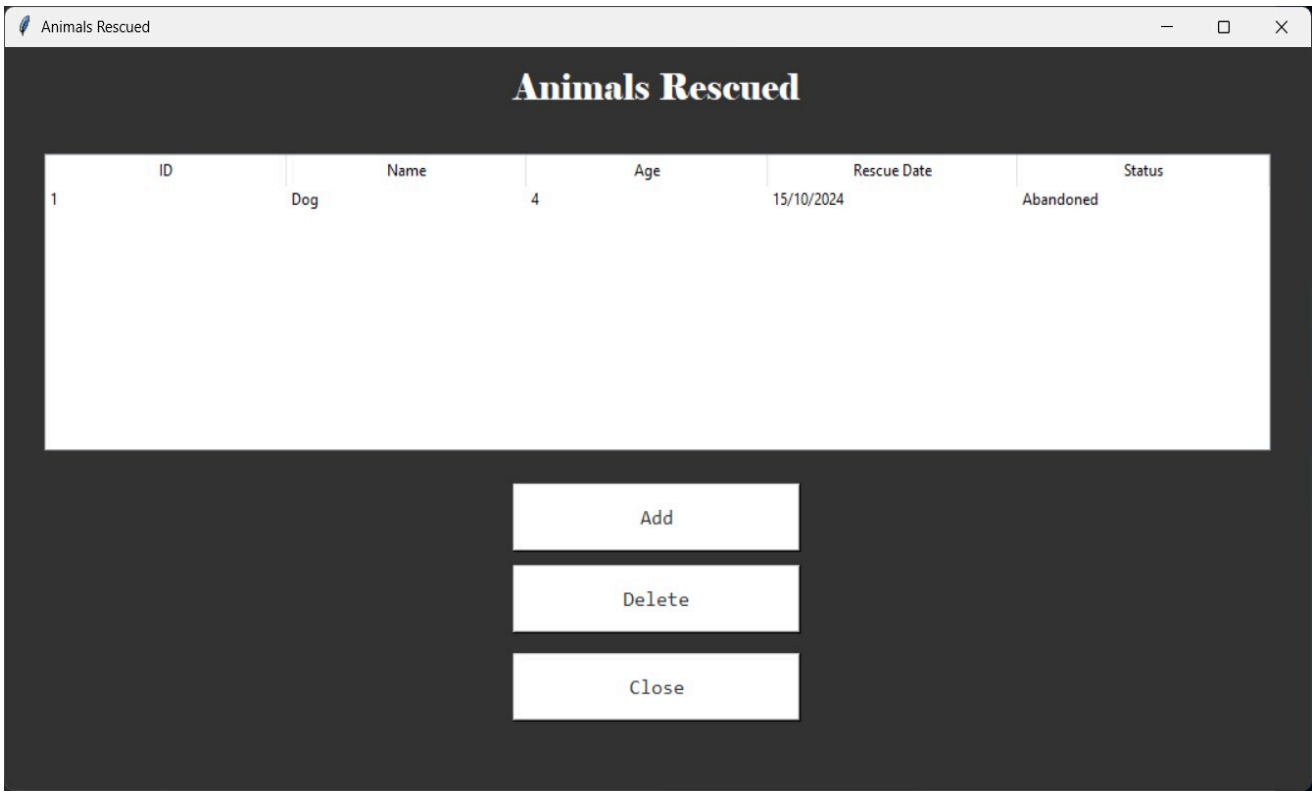
❖ When we press the Add button in order to add data, we see this window below, where the user can enter data.



❖ As we press Save button we get this MessageBox:

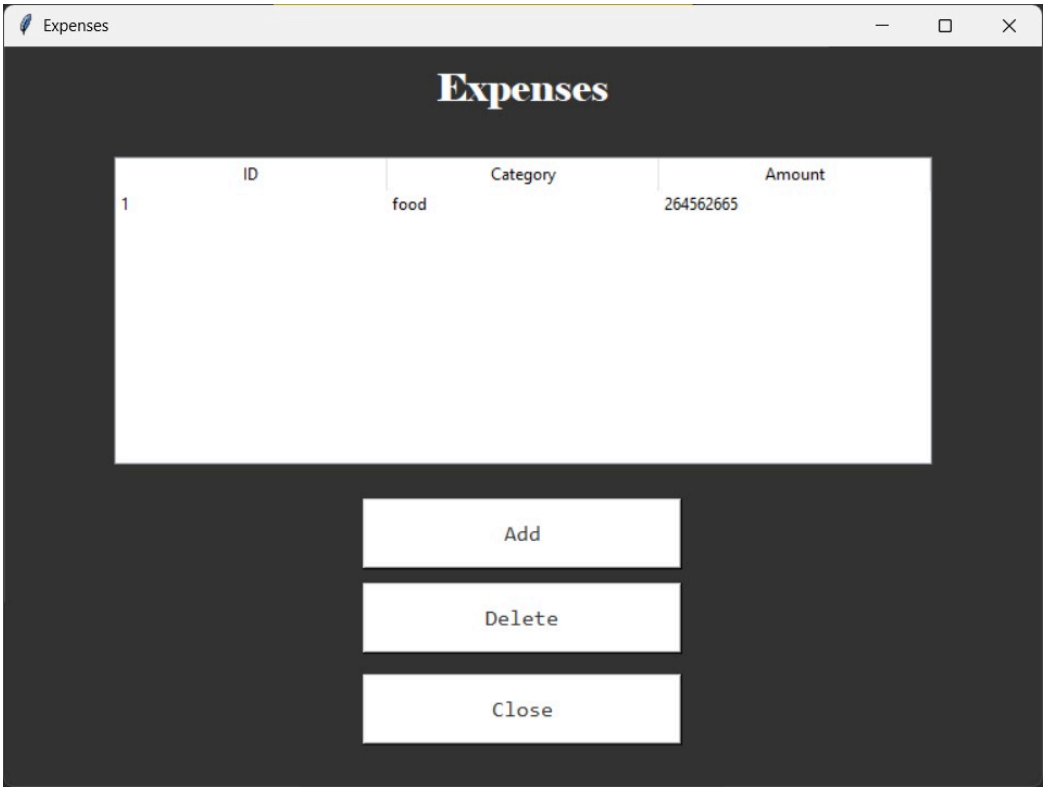


❖ The Screen after adding Data:

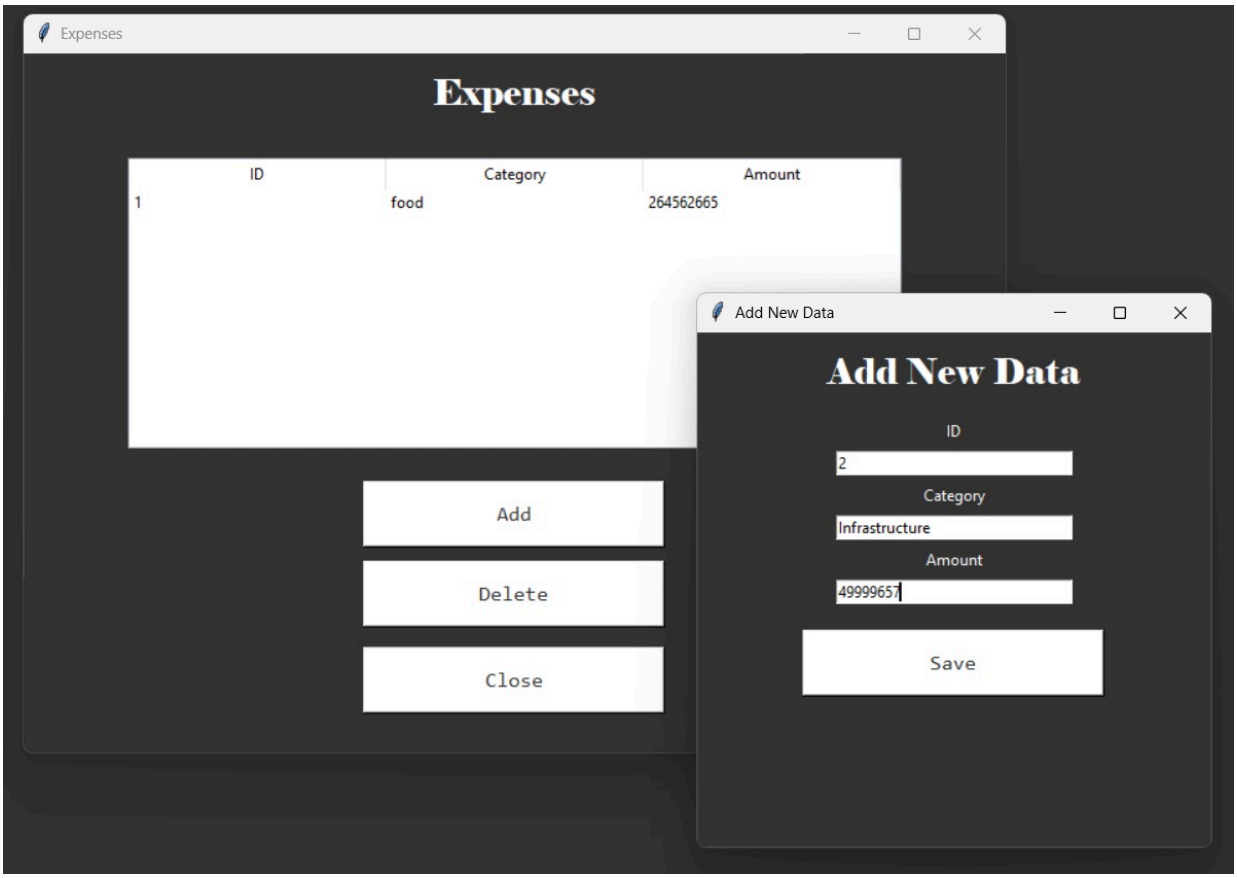


At Expenses GUI :

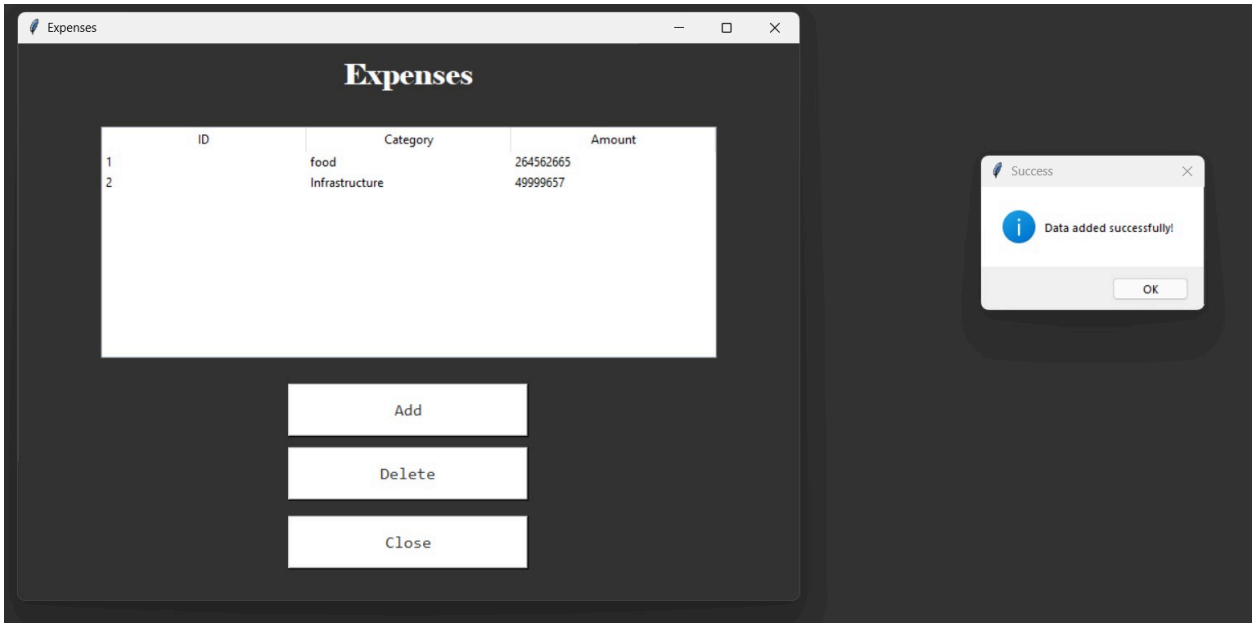
- When you press the second option you enter the Expenses table :



❖ When you press ‘add’ button, it takes you to this window which is shown below



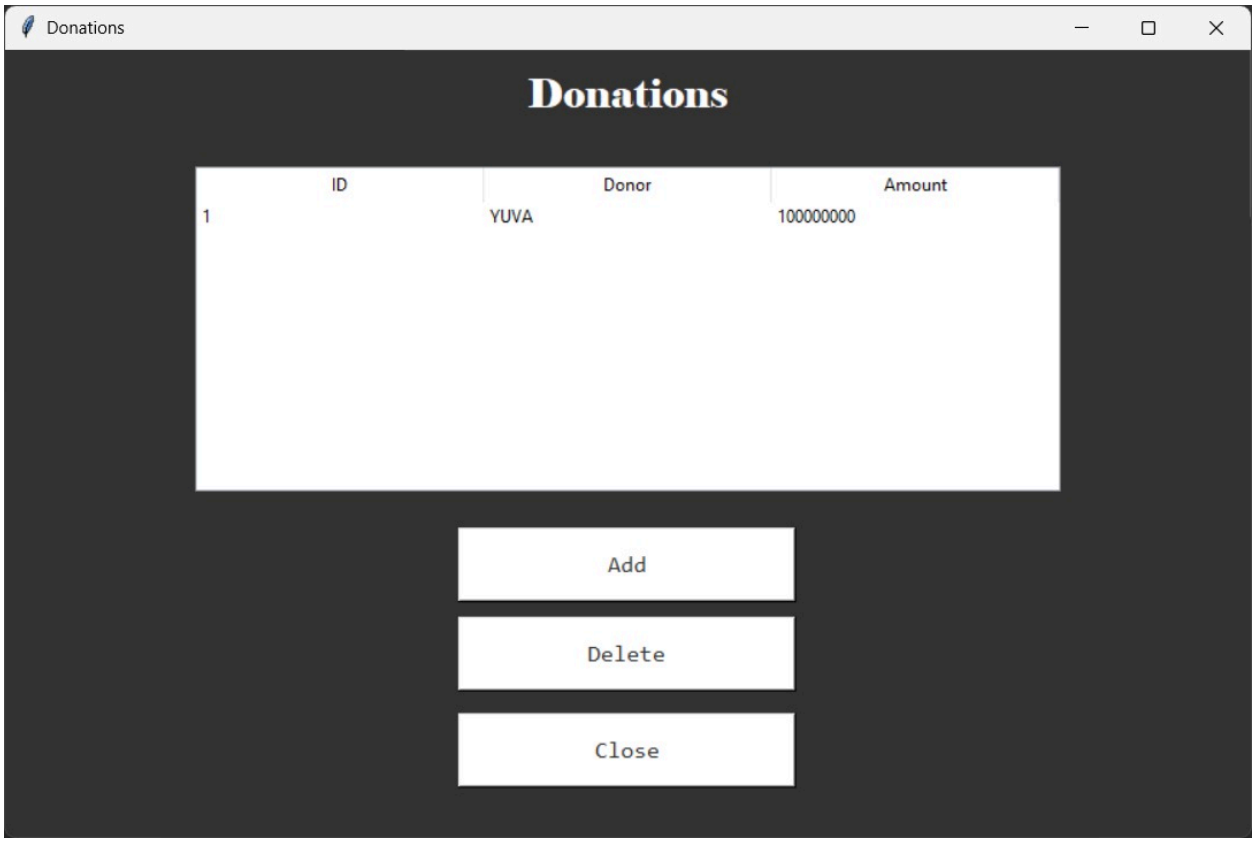
❖ Here you add the record :



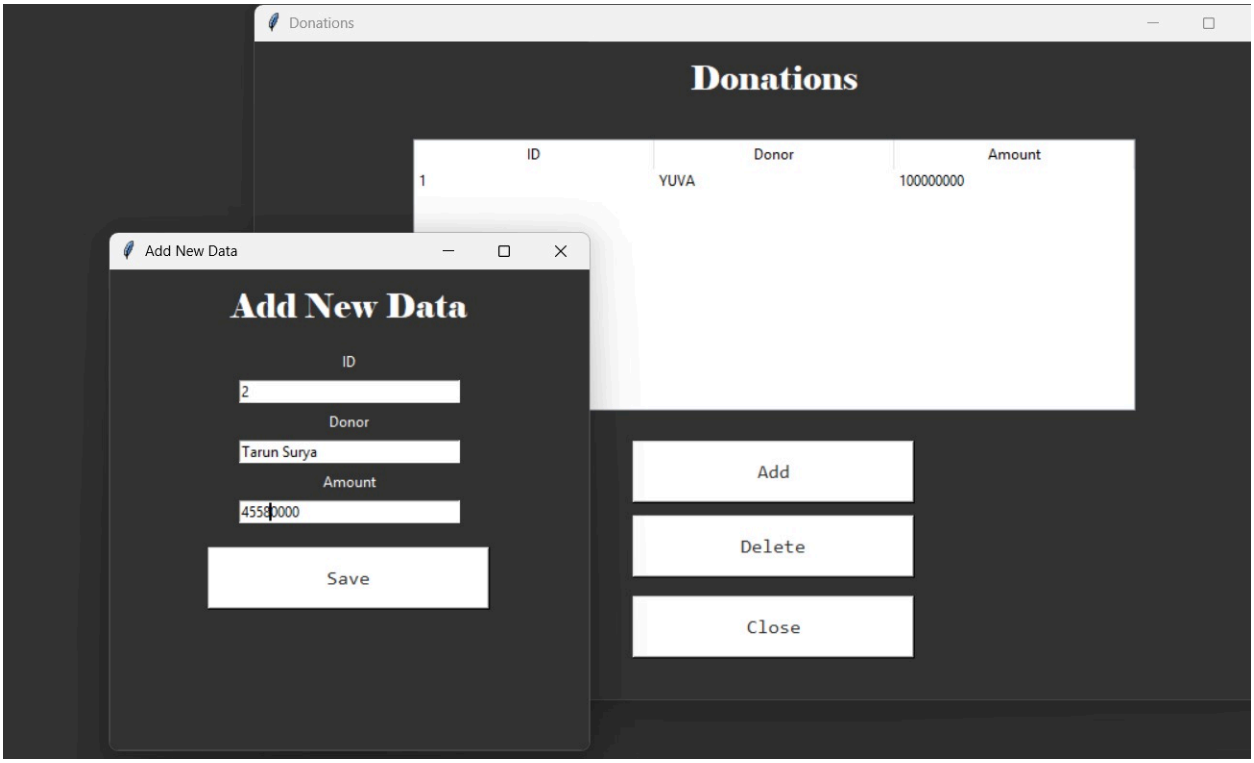
❖ Then you press save button and the credentials would get added to the table and you get a success message.

At Donations GUI:

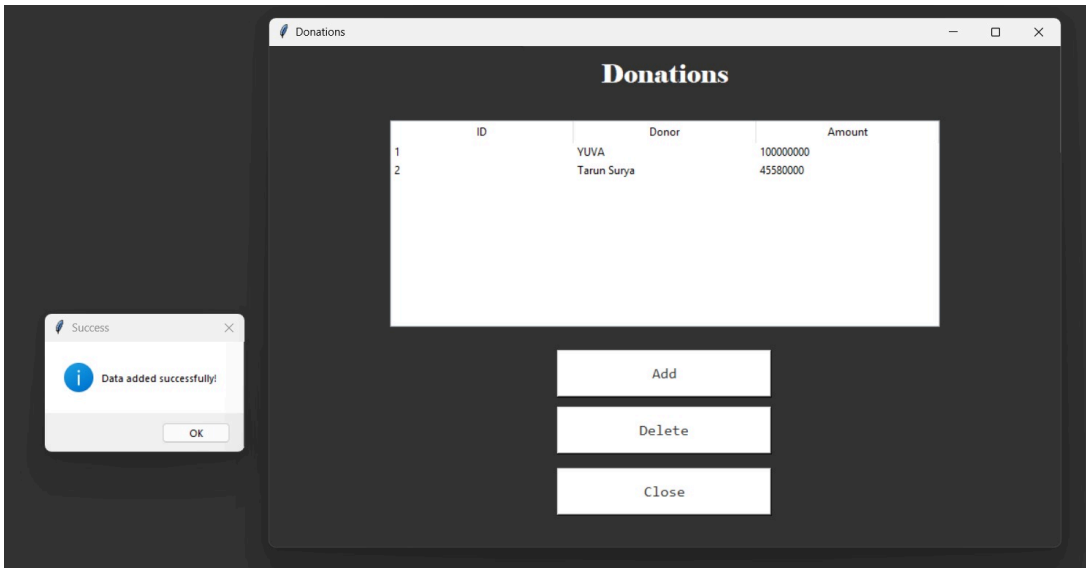
- When you press the third option you enter the Donations table :



❖ Here just like the previous add button, this does the same.

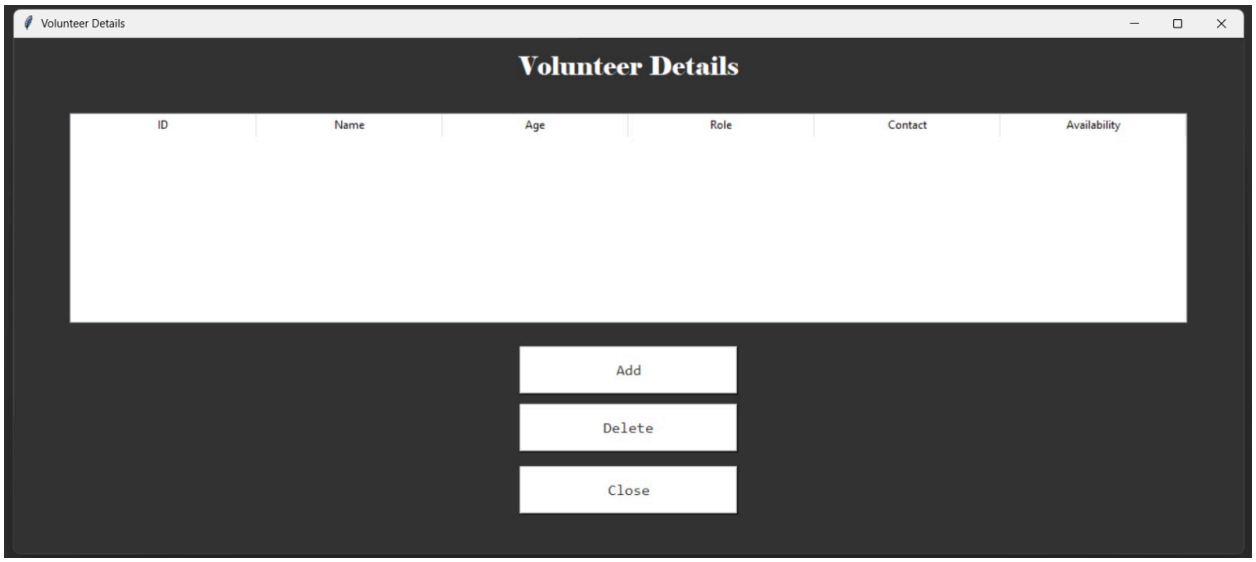


❖ The screen after adding the data :

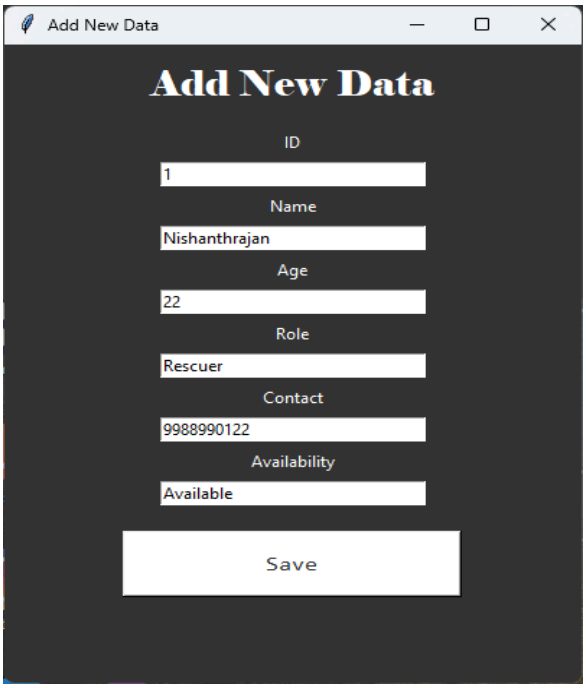


At Volunteer Details GUI:

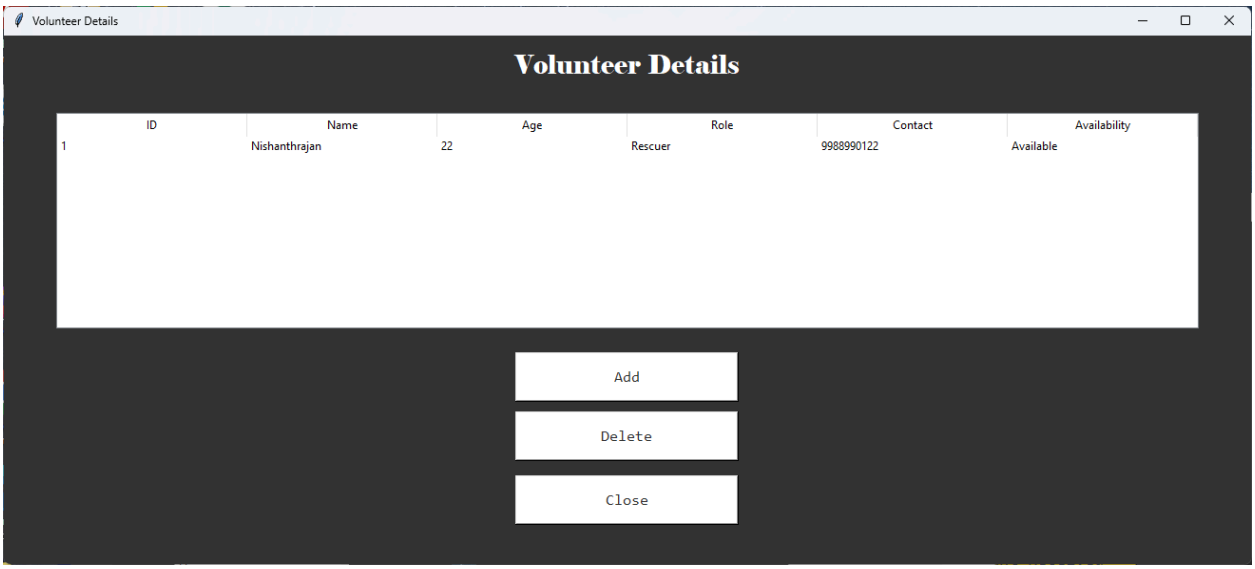
- When you press the fourth option you enter the Volunteer Details GUI:



- When you press the add button you get the below window:



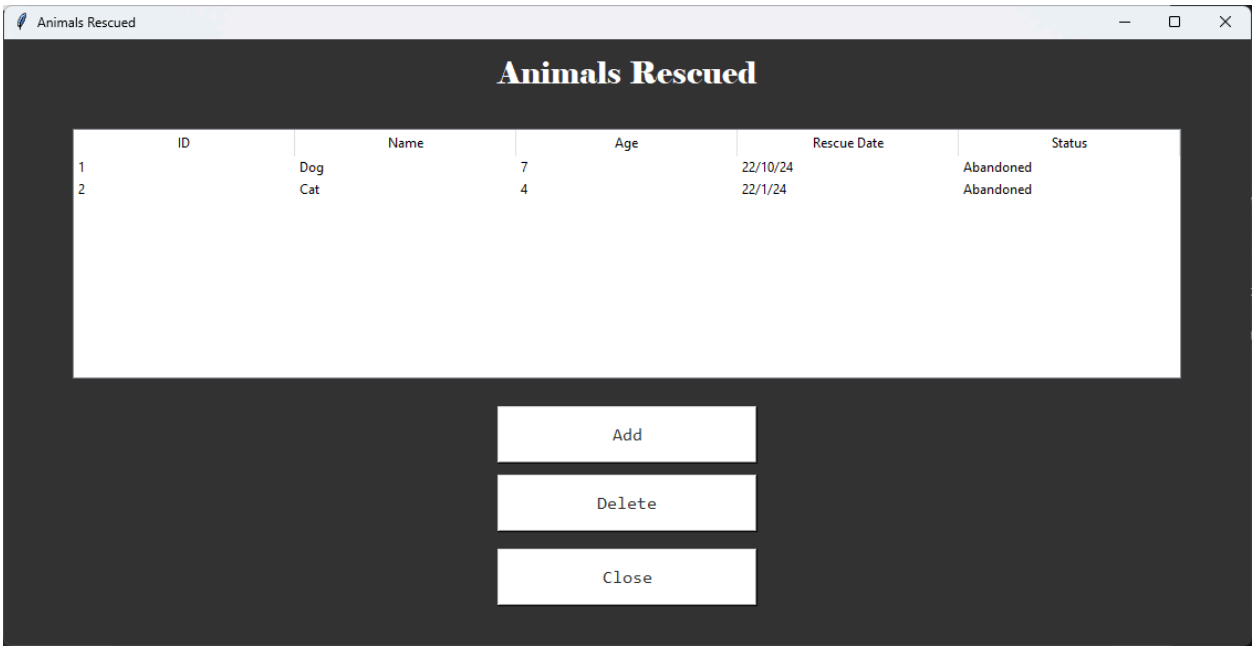
❖ Once the data has been added it looks like this:



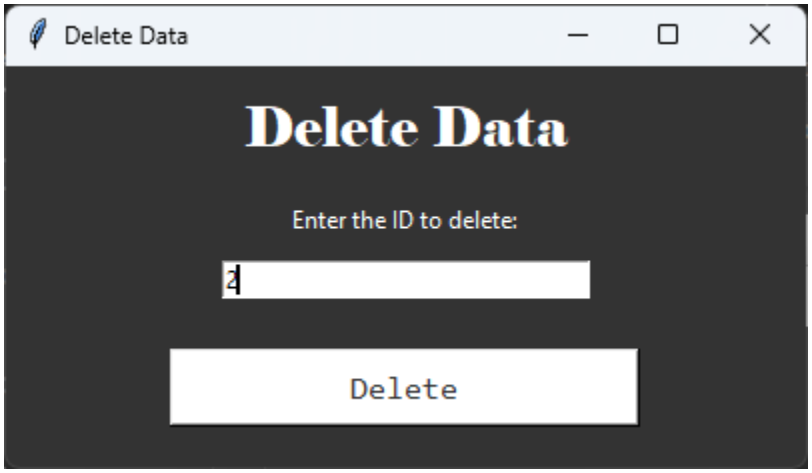
Deleting Data :

At Info about Animals Rescued GUI :

After pressing the first option, we see the window shown below. Consider the following data :

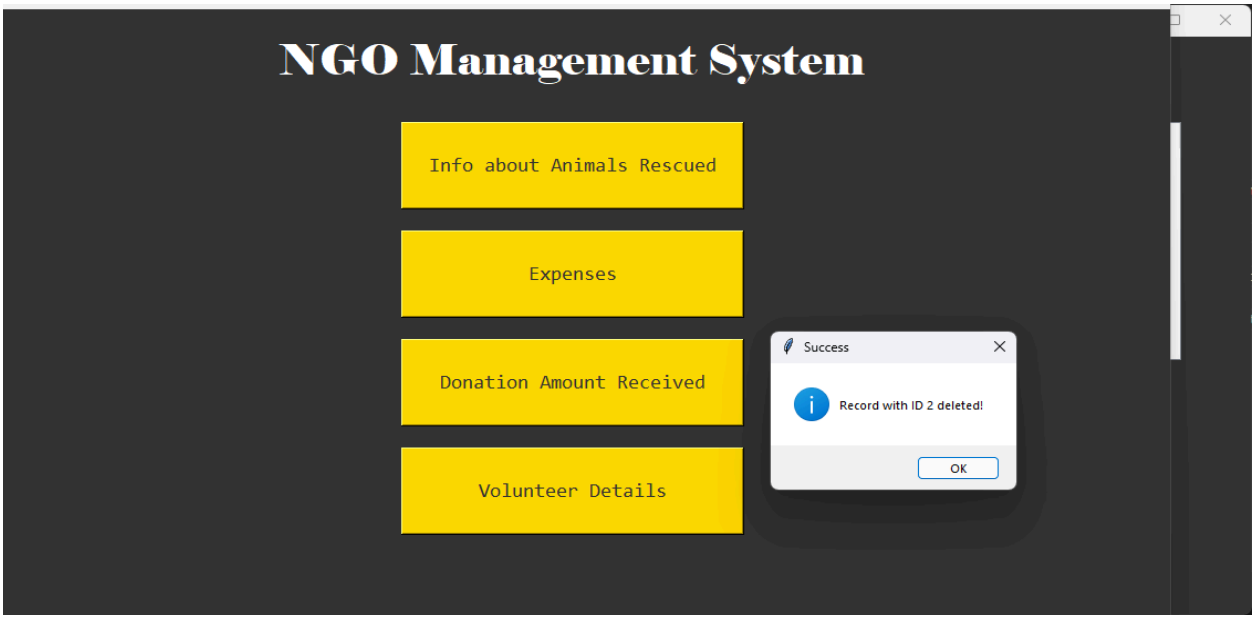


❖ To delete data, press the Delete button.We see the following window :

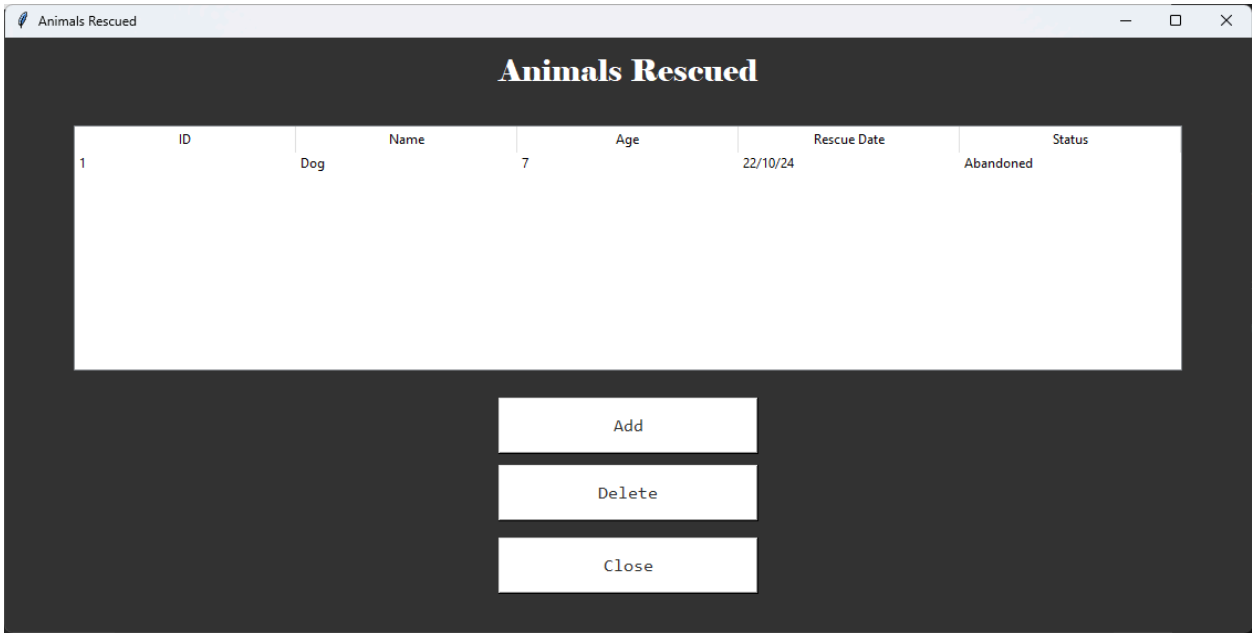


❖ After typing the ID to be deleted, press the delete button to delete the record with the given ID.

❖ We get the following message after successful deletion:



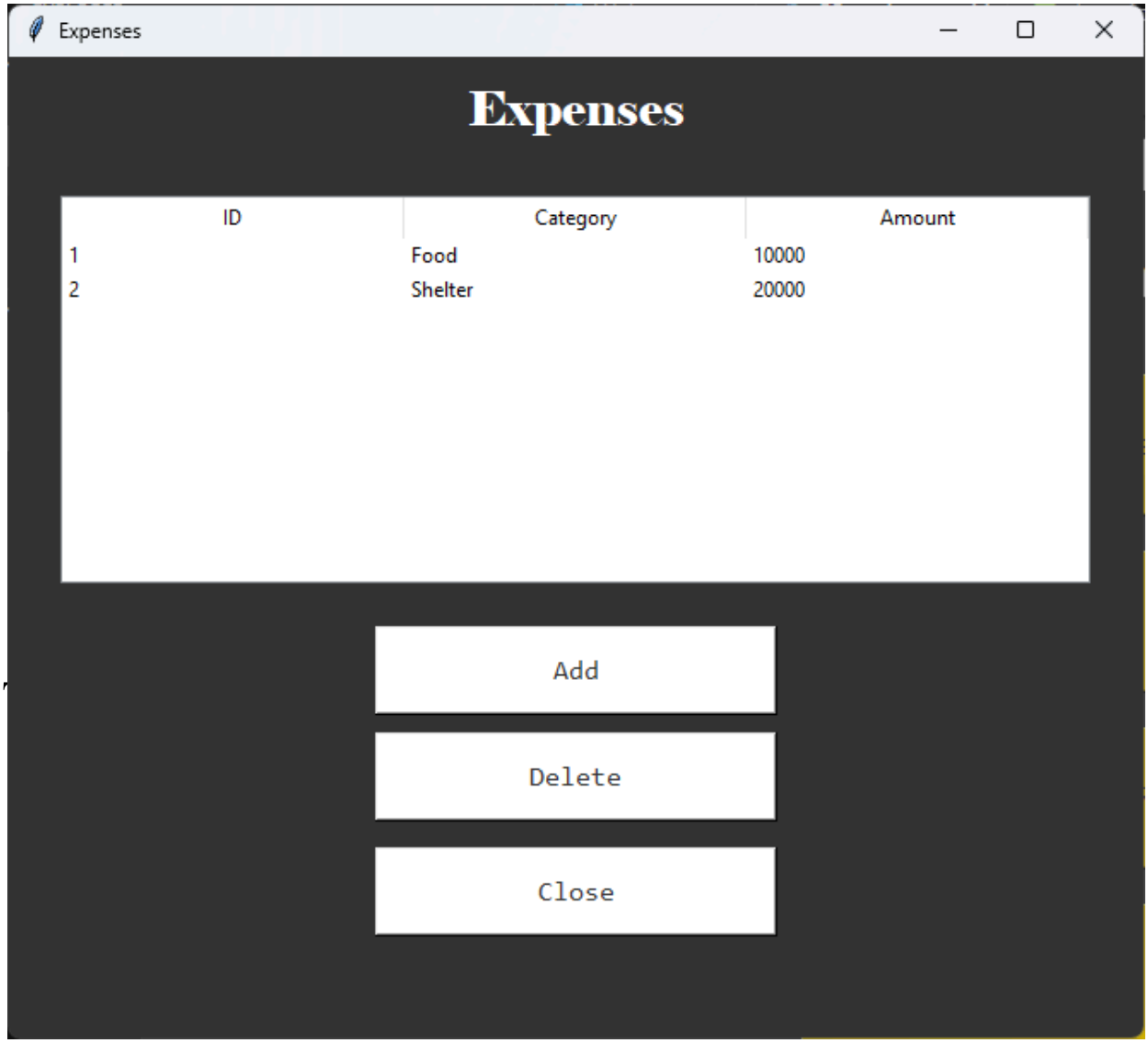
❖ The Animals rescued GUI after the deletion of the record with ID = 2 :

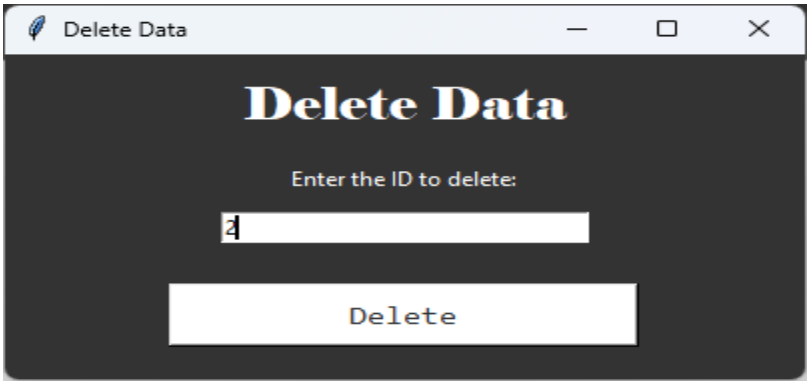


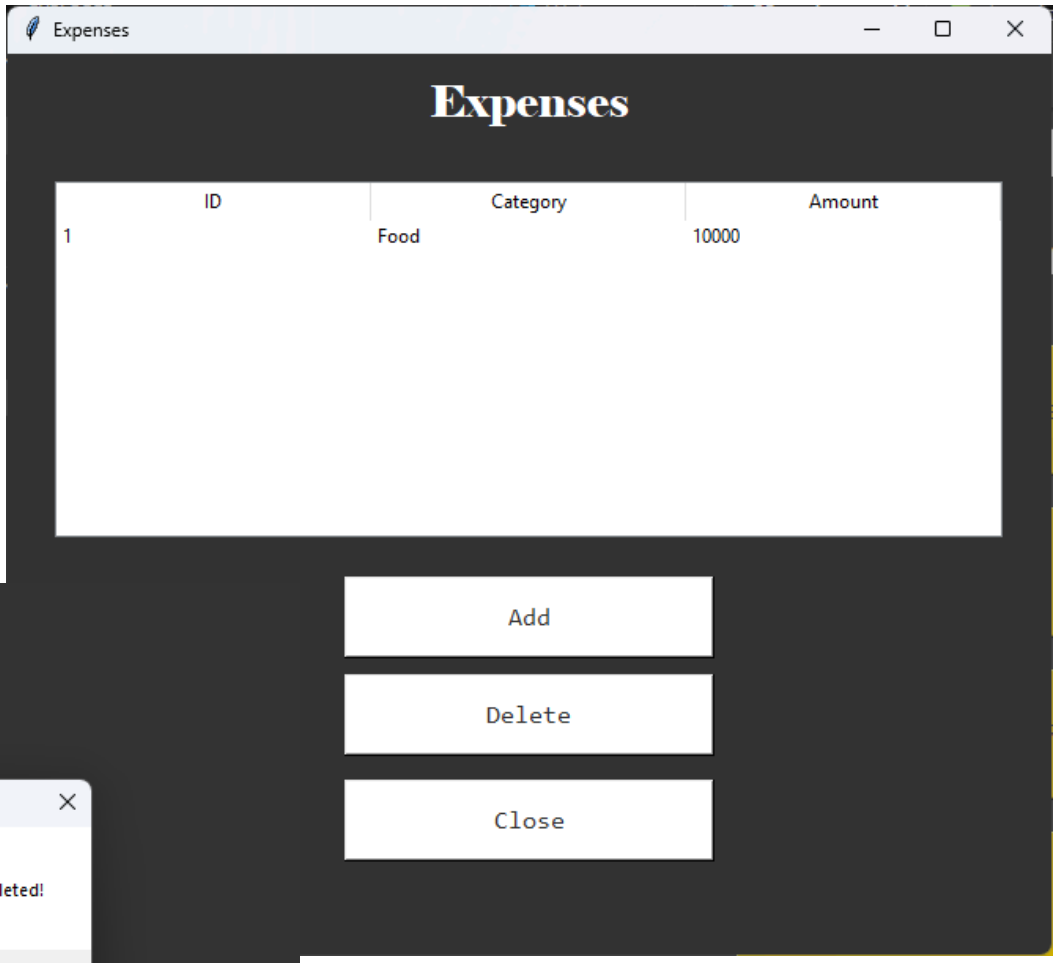
At Expenses GUI :

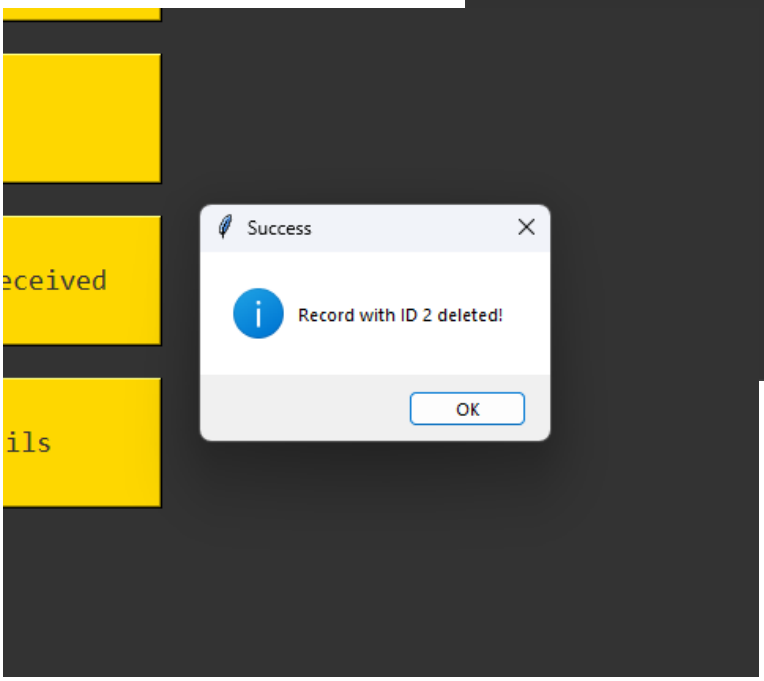
- After pressing the second option, we see this window. Consider the following data:

- Deletion of data will be similar to the previous option.




Delete Data
Delete Data
Enter the ID to delete:
2
Delete

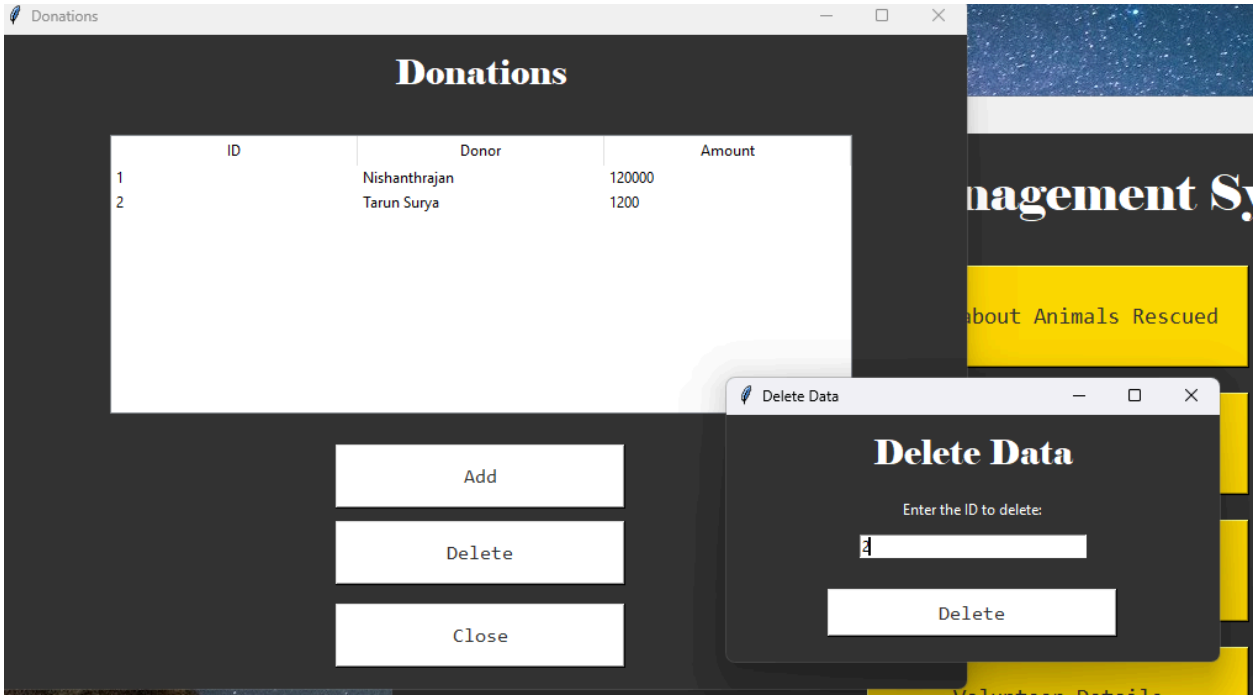

Expenses
Expenses
ID
Category
Amount
1
Food
10000
Add
Delete
Close

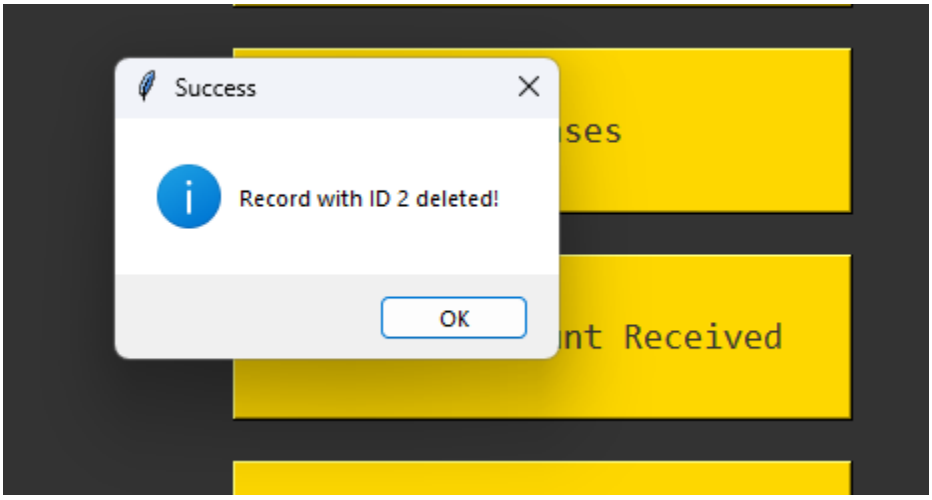

eceived
ils
Success
Record with ID 2 deleted!
OK

At Donations GUI :

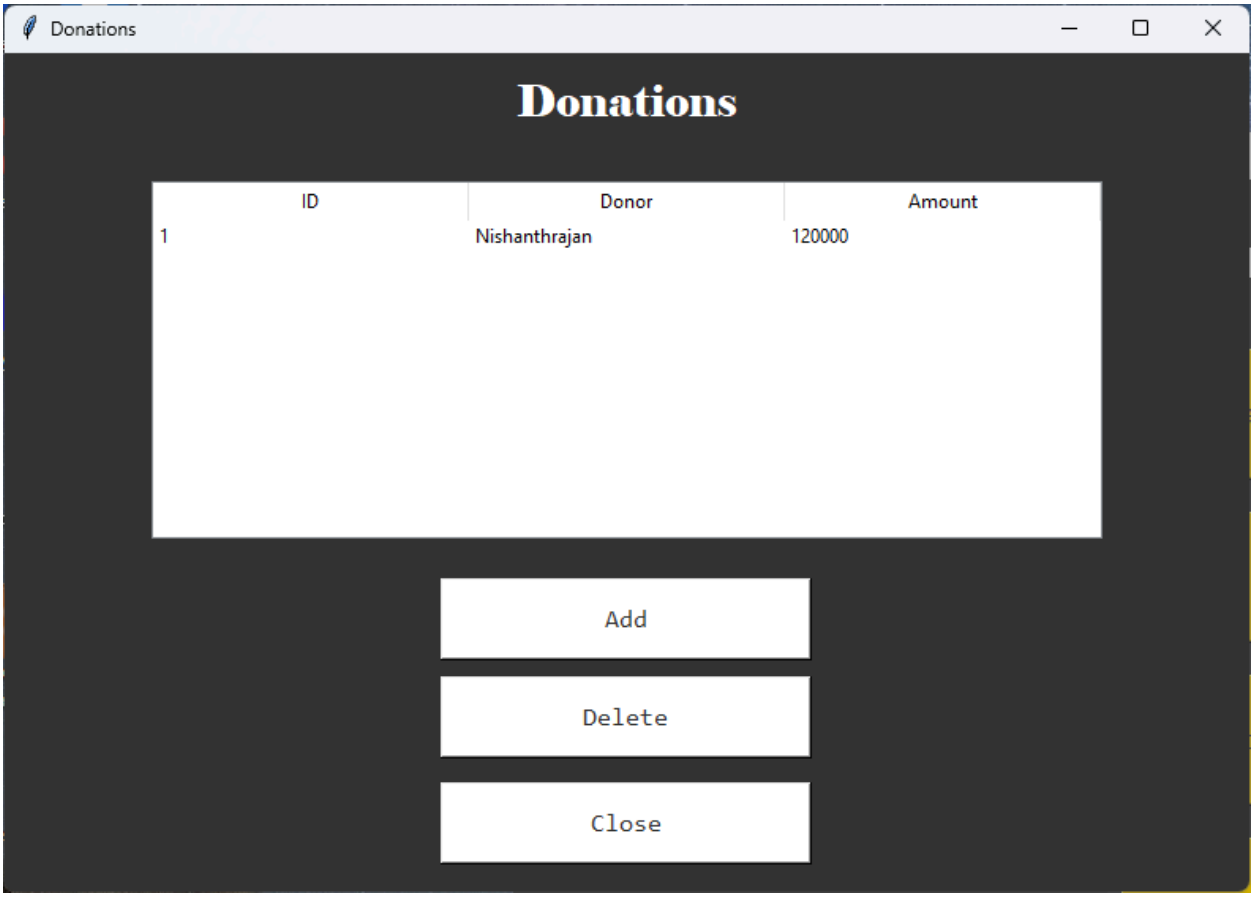
- After pressing the third option, we see this window. Consider the following data:

  Deletion of data will be similar to the previous options.





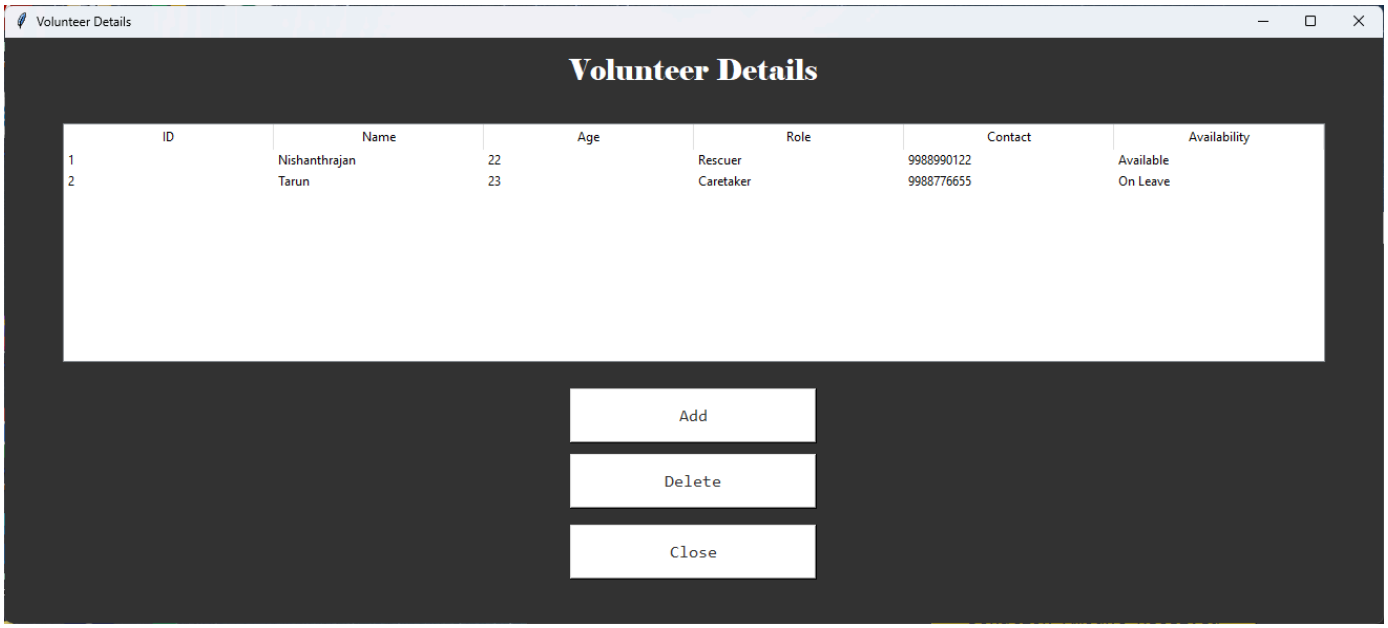
❖ The screen after the removal of record:

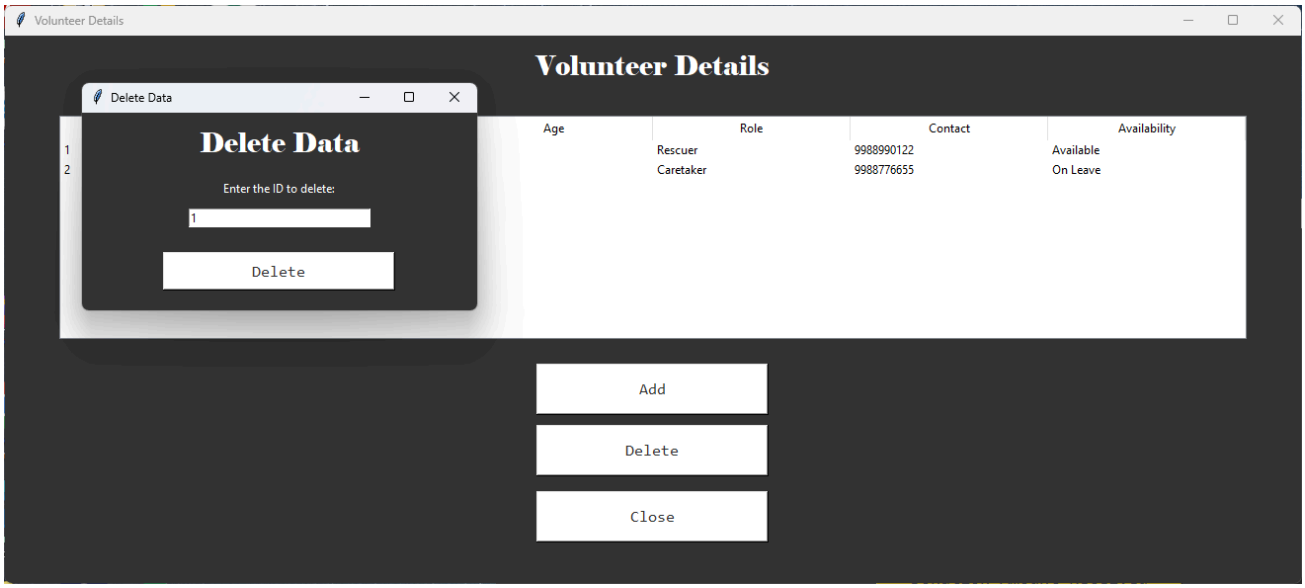


At Volunteer Details GUI:

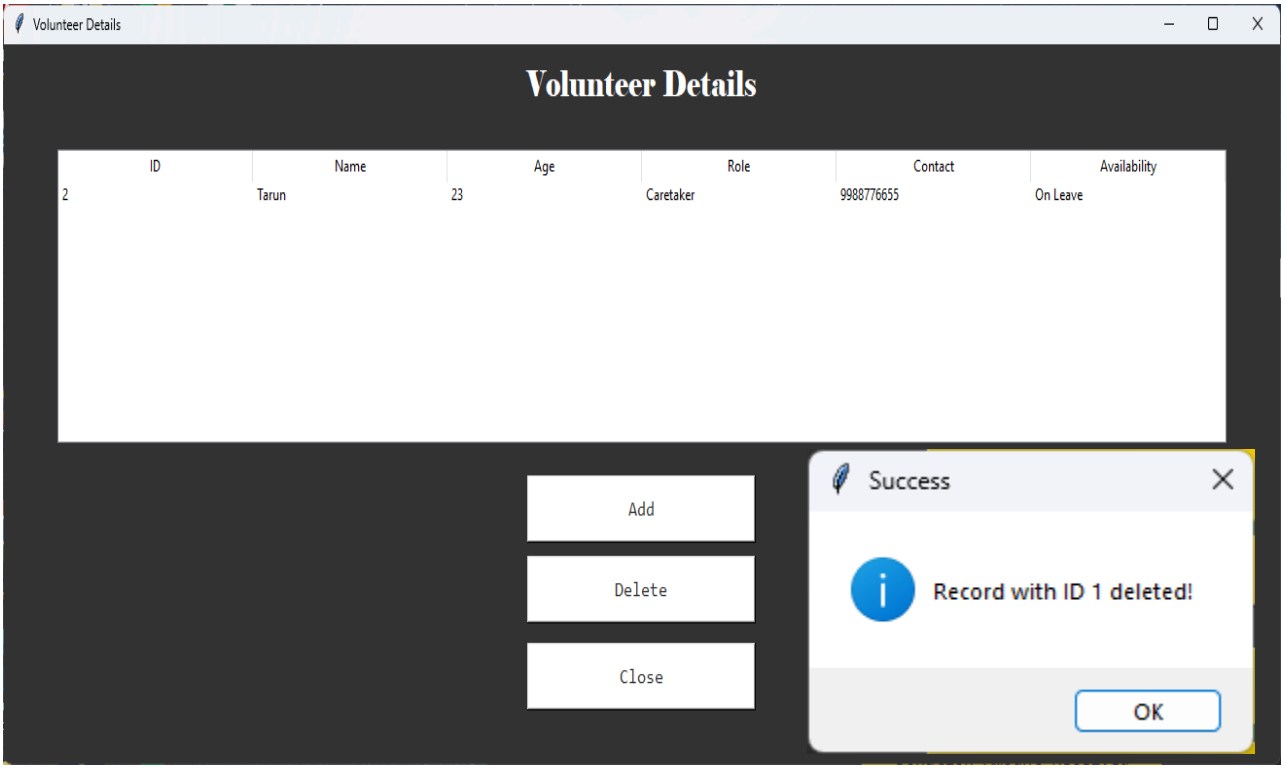
- After pressing the fourth option, we see this window. Consider the following data:

  Deletion of data will be similar to the previous options.





❖ The screen after the removal of record:



Bibliography

1). Python Tkinter Beginner Course :

https://youtu.be/YXPyB4XeYLA?si=osOdV9IS7RF5BpkU

2). https://youtu.be/rtR5wHXPKZ4?si=ItNNJF_hYttFkpJO

3). https://docs.python.org/3/library/tk.html

4). https://www.geeksforgeeks.org/python-tkinter-treeview-scrollbar/