Database Management Project Week -3 Submission

CS262 | ZOO MANAGEMENT SYSTEM

Group Partners	
201951007	Abhishek Kumar Meena
201951131	Sagar ved Bairwa
201951164	Urmila Rathore
201951169	Vishal Ashok Daragade
201951172	Vivek kumar Goyal

INDEX

ZOO MANAGEMENT SYSTEM

1. Introduction

In the early days, the manual usage causes many mistakes by the user and administrative. Using manual properties in the fields was not comfortable for the consumers because it was slower than technical usages, caused wastages of the consumers' time and contained many formalities in usage. The goal of this project is to make the activities of the park easier and modernized.

2. System Analysis

2.1 Problem Definition

The need for Zoo information management System (ZIMS) is concerned With Zoo information handling and Keeping all the data in a Proper way that can be maintained without any error Data. This project provides a new way to maintain the visitor entries by providing the entry tickets which will be saved in the data base by that any time the admin can view the entries details of the visitor as well as the total amount collected by the entries.

It also keeps the track of animals data's with its unique id in such a way that the data's of the animals are loaded into the database. The unique id also is used to display the loaded data.

2.2 Proposed System

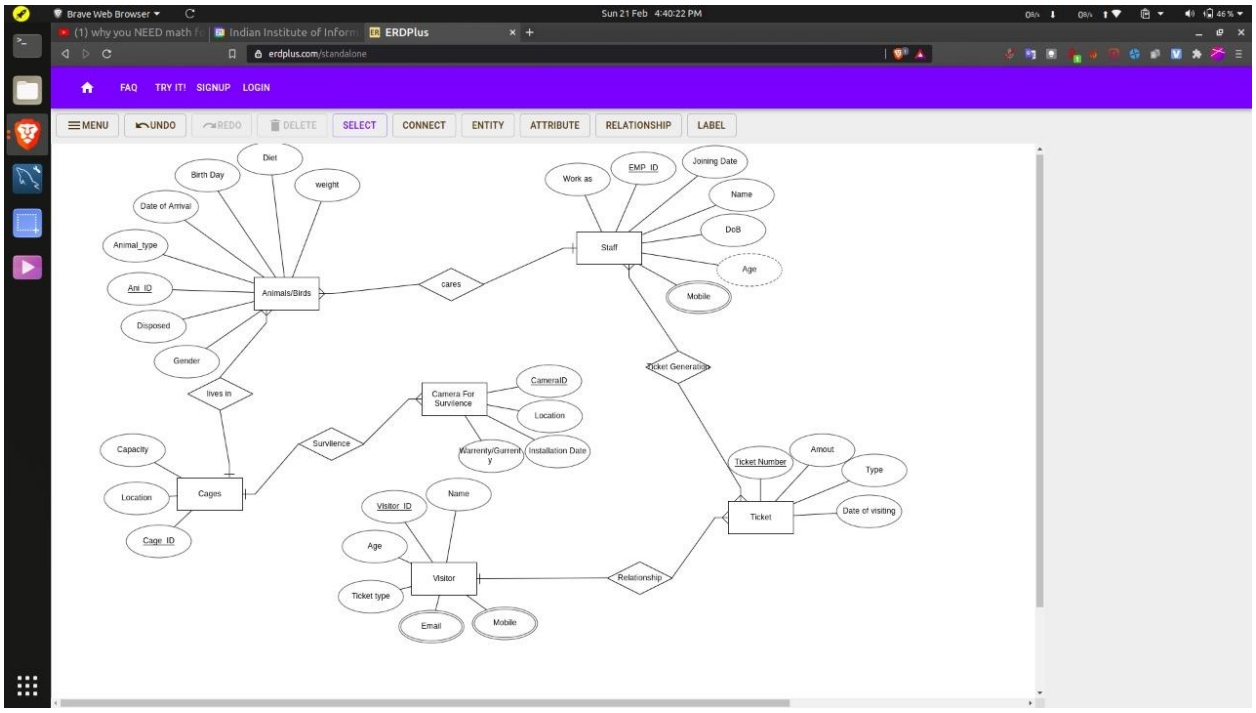
The zoo can tackle the problem by upgrading to Computerized Systems. The system should be able to generate tickets based on the request of the user and must save the data back to the database on successfully generating of tickets. And Easy to keeping all the Animals data in a Proper way that can be maintained without any error Data.

The data will be provided in simple to understand graphical and chart form.

3. System Design

3.1 Activity Form

3.2 ER diagram



3.3 Table Design

3.4 GUI Diagram

4.Project Description

The Following are the main functional modules of this project

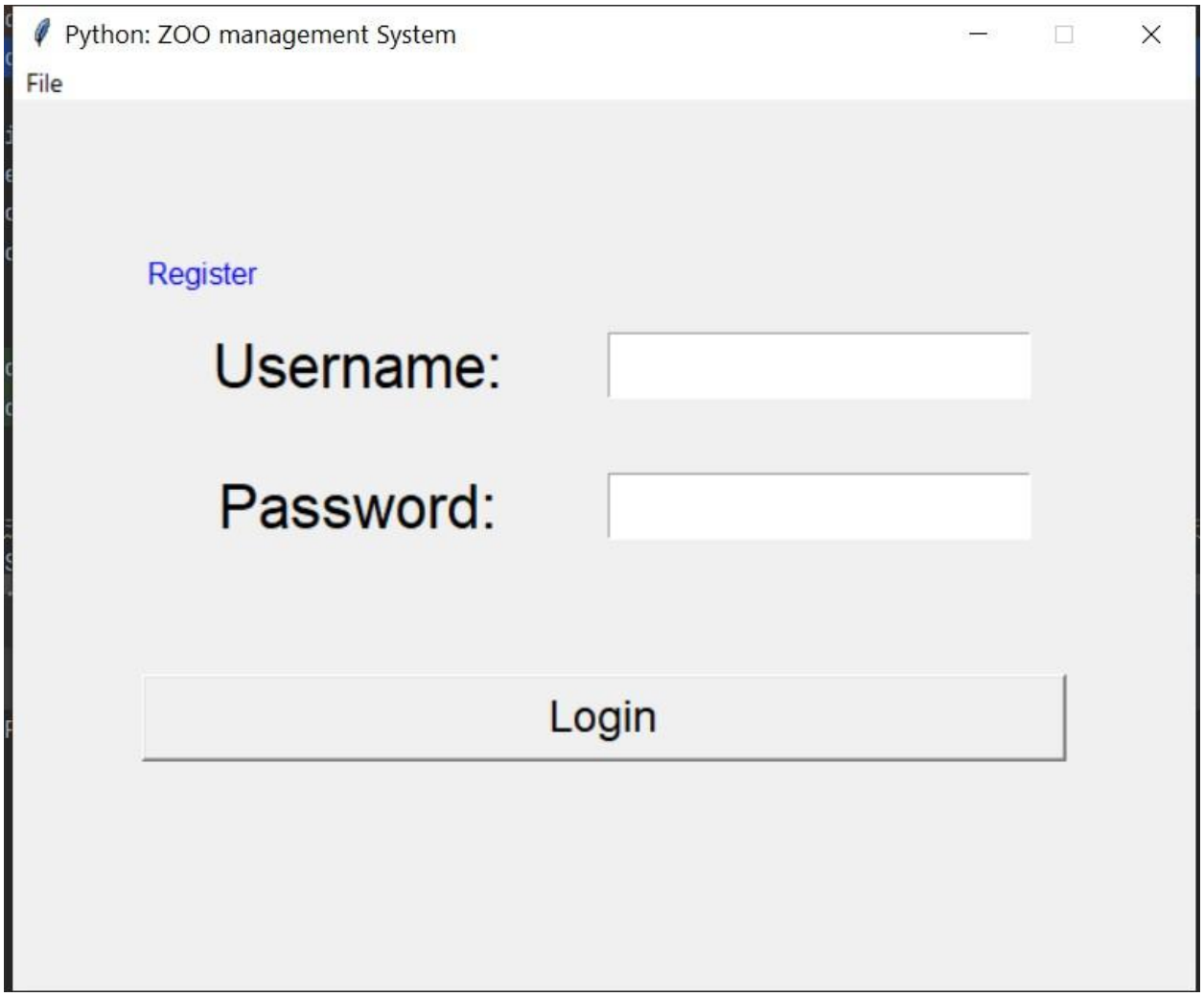
- ➔ Animals data entry
- ➔ Animals data update
- ➔ Animals data chart
- ➔ Entrance ticket
- ➔ Ticket data
- ➔ Day to day ticketing information

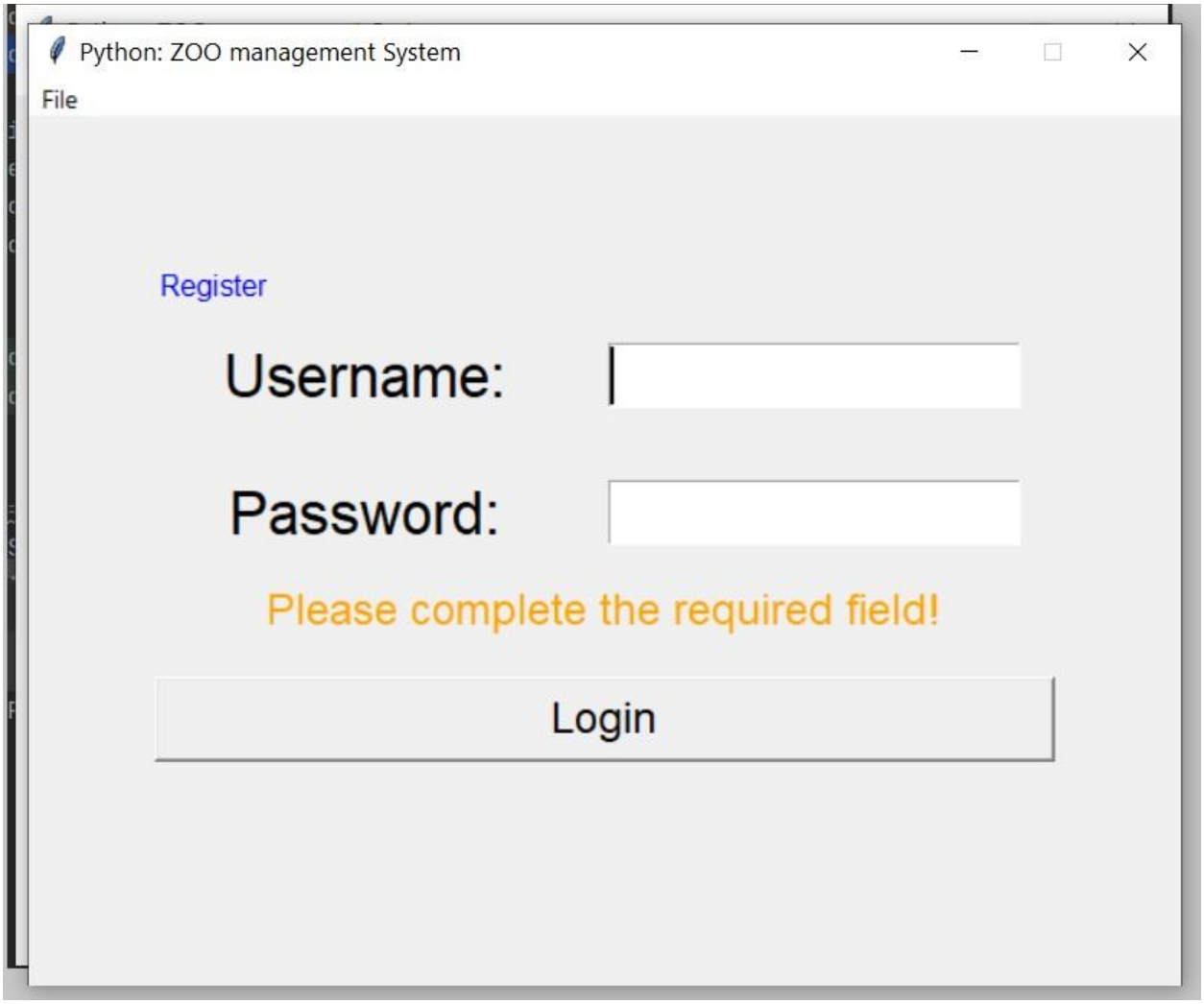
To start off with the user entry will be authenticated by getting the user name and password the admin home user menus open when the admin officer open with admin id and password and the admin menu don't open when the users open with user id and password.

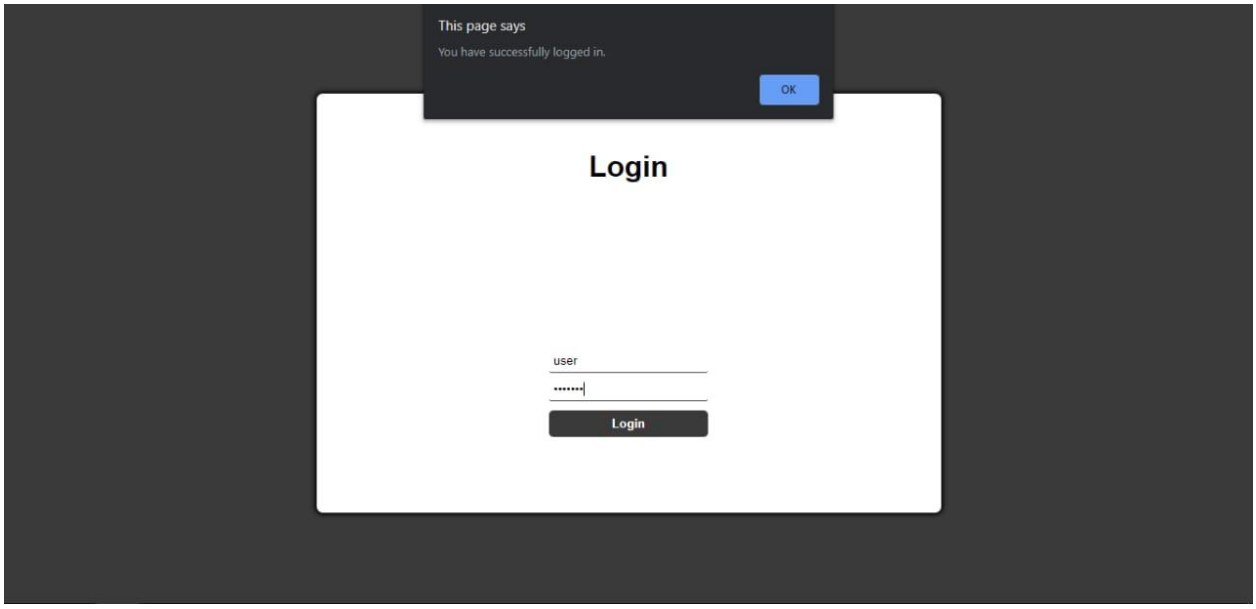
Facility to create the user account is not visible on the front page of starting because the user account doesn't have to create the new user. Admin only has to create the new user account.

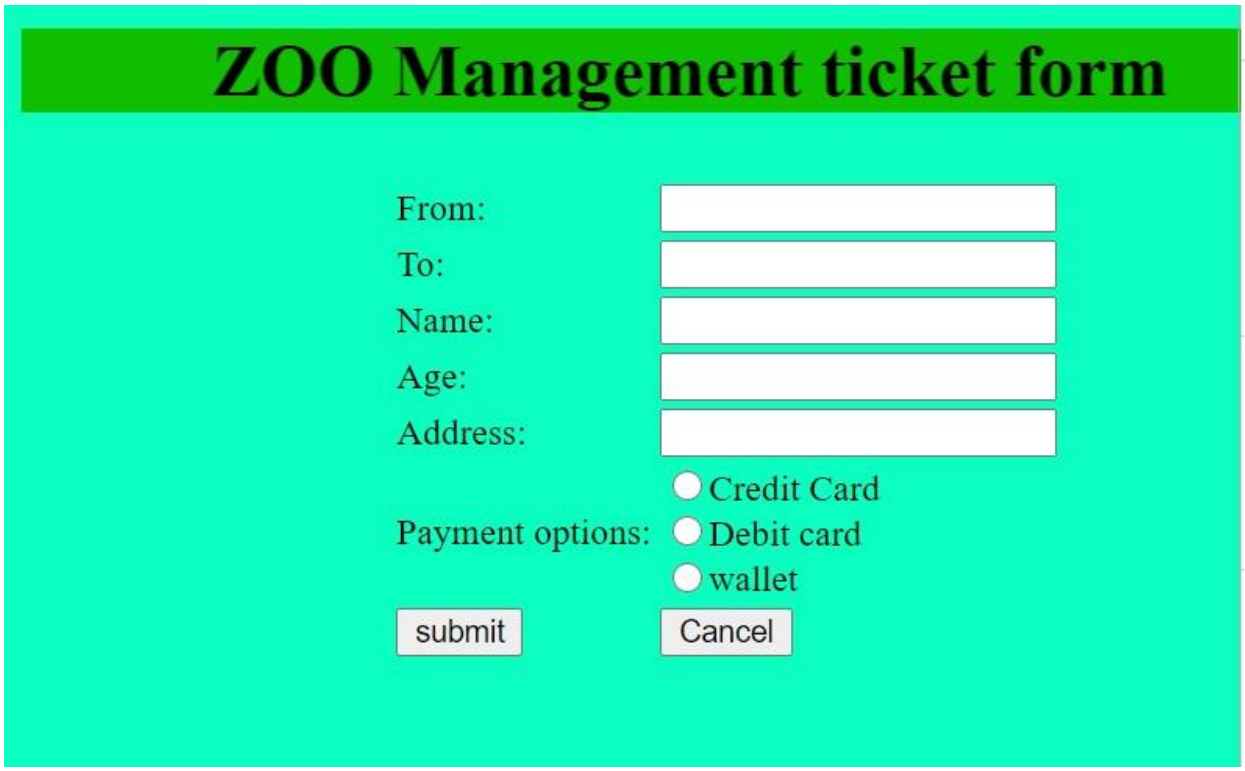
1. Animal Data Entry: This module is to maintain basic details of all types of animals that is being in the whole zoo such as
    a. Name of the species
    b. Unique id of the animal
    c. Date of the arrival
    d. Number of Male, Female, and Unsex

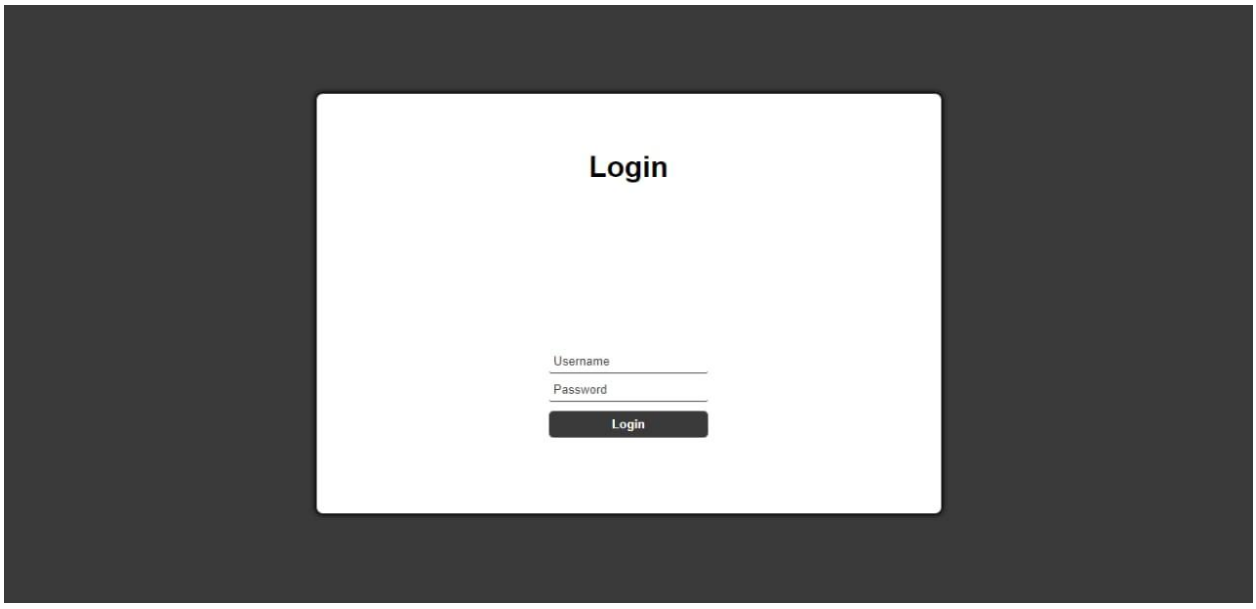
5. Screen Shots

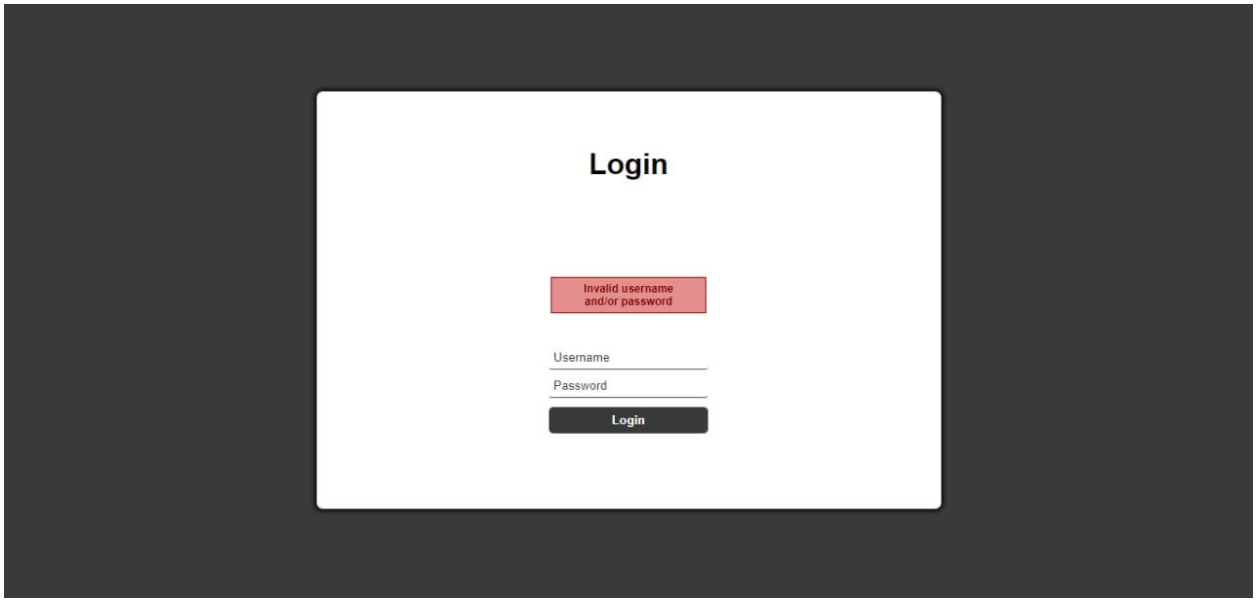



Python: ZOO management System
File
Register
Username:
Password:
Please complete the required field!
Login


This page says
You have successfully logged in.
OK
Login
user
Login

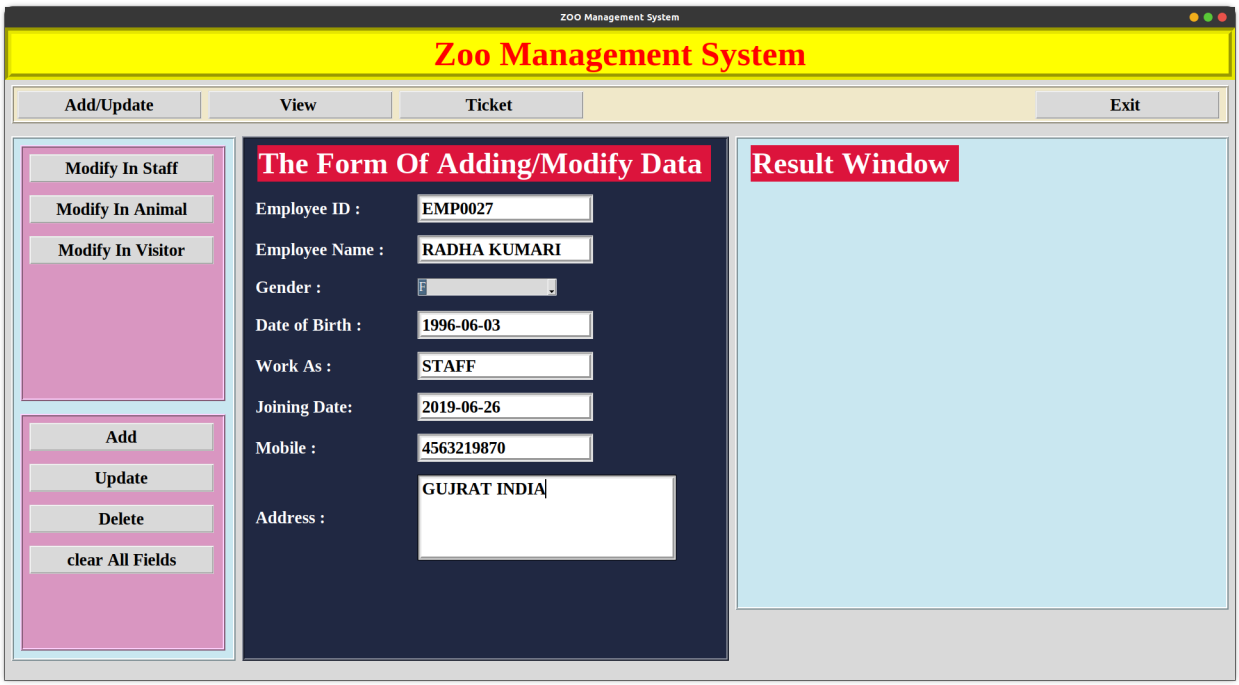

ZOO Management ticket form
From:
To:
Name:
Age:
Address:
Payment options:
Credit Card
Debit card
wallet
submit
Cancel


Login
Username
Password
Login



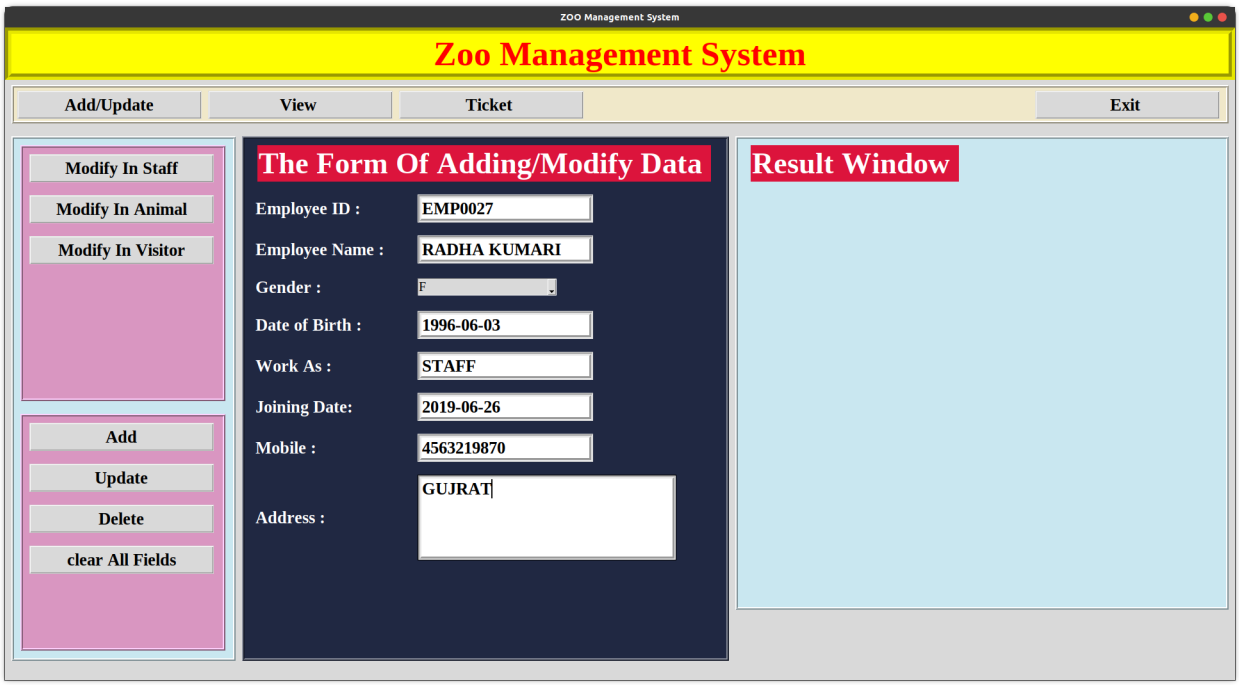
ScreenShot OF FORM (Staff)

1. ADDING A NEW STAFF



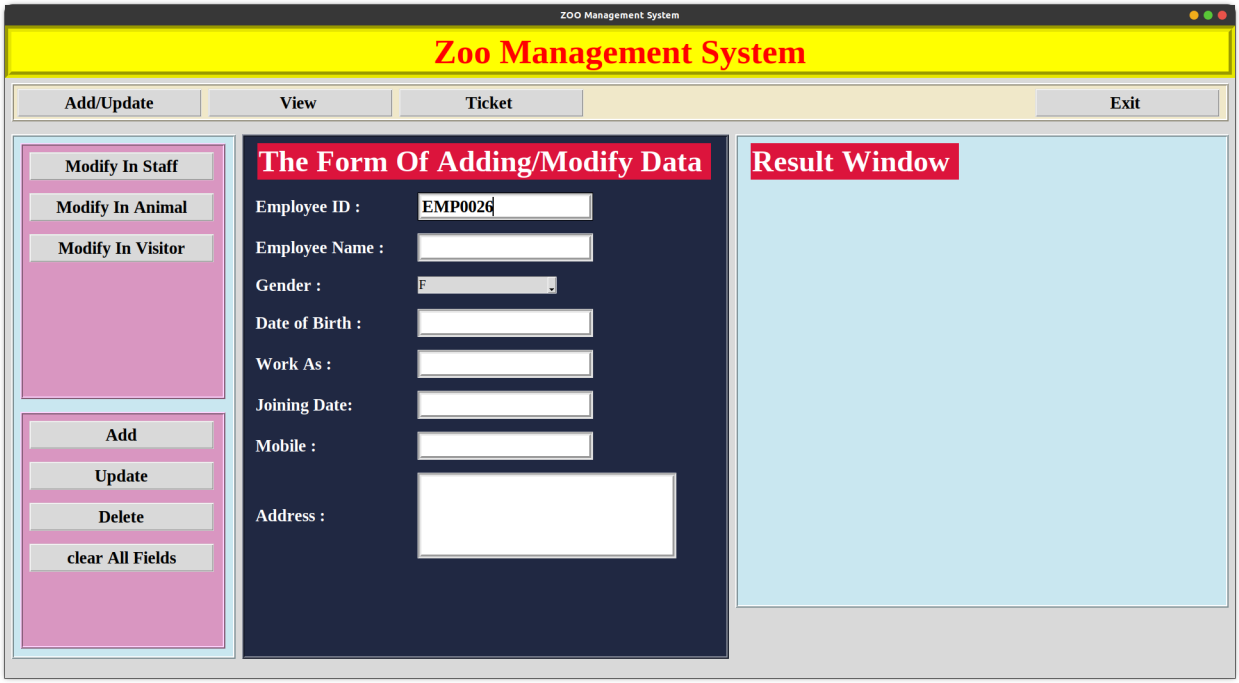
```
Run: main
/home/ubuntu/PycharmProjects/Gui_walaChidiyaghar/venv/bin/python /home/ubuntu/PycharmProjects/Gui_walaChidiyaghar/main.py
Query Executed
```

Update a STAFF

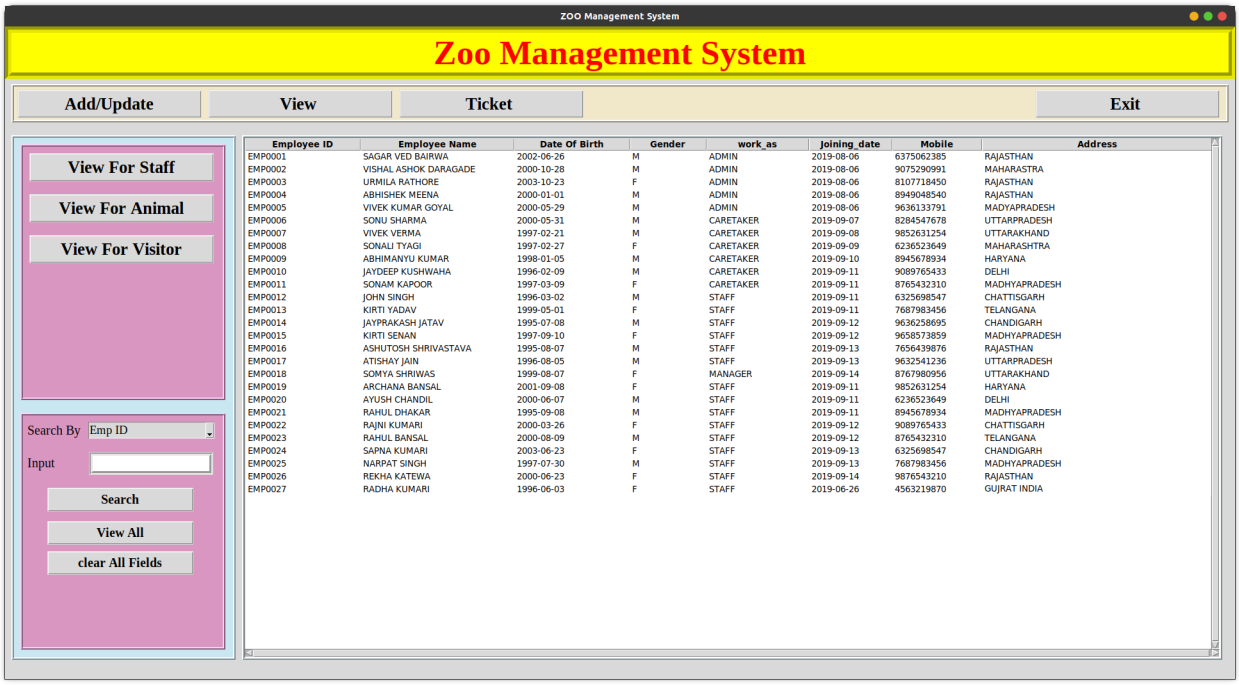



```
Run: main
/home/ubuntu/PycharmProjects/Gui_walaChidiyaghar/venv/bin/python /home/ubuntu/PycharmProjects/Gui_walaChidiyaghar/main.py
Query Executed
Query Executed
```

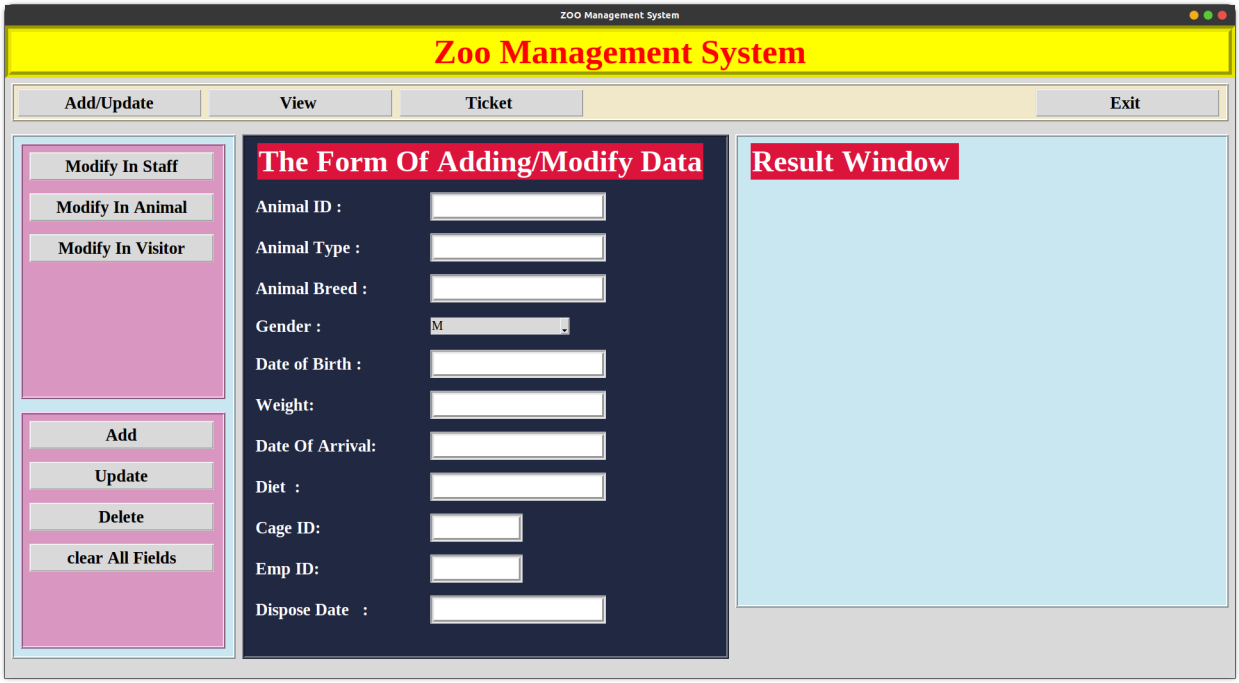
DELETE A STAFF

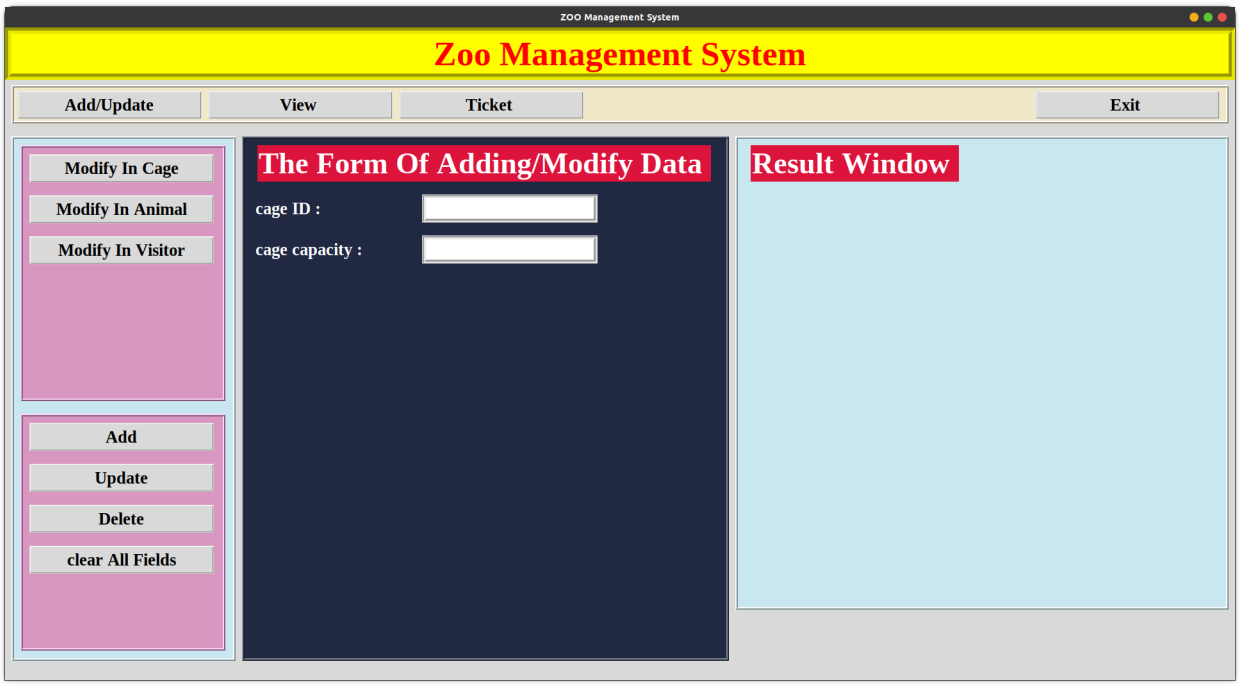


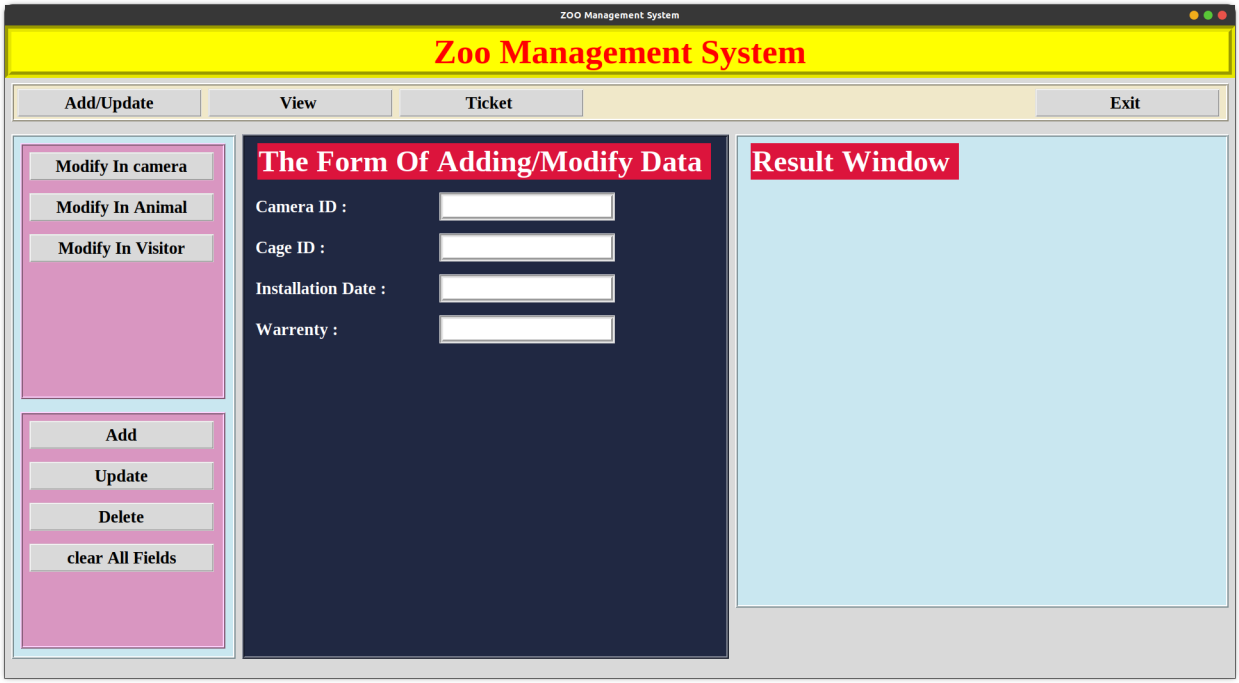
VIEW STAFF DATA



Employee ID	Employee Name	Date Of Birth	Gender	work_as	Joining_date	Mobile	Address
EMP0001	SAGAR VED BAIRWA	2002-06-26	M	ADMIN	2019-08-06	6375062385	RAJASTHAN
EMP0002	VISHAL ASHOK DARAGADE	2000-10-28	M	ADMIN	2019-08-06	9075290991	MAHARASTRA
EMP0003	URMILA RATHORE	2003-10-23	F	ADMIN	2019-08-06	8107718450	RAJASTHAN
EMP0004	ABHISHEK MEENA	2000-01-01	M	ADMIN	2019-08-06	8949048540	RAJASTHAN
EMP0005	VIVEK KUMAR GOYAL	2000-05-29	M	ADMIN	2019-08-06	9636133791	MADYAPRADESH
EMP0006	SONU SHARMA	2000-05-31	M	CARETAKER	2019-09-07	8284547678	UTTARPRADESH
EMP0007	VIVEK VERMA	1997-02-21	M	CARETAKER	2019-09-08	9852631254	UTTARAKHAND
EMP0008	SONALI TYAGI	1997-02-27	F	CARETAKER	2019-09-09	6236523649	MAHARASHTRA
EMP0009	ABHIMANYU KUMAR	1998-01-05	M	CARETAKER	2019-09-10	8945678934	HARYANA
EMP0010	JAYDEEP KUSHWAHA	1996-02-09	M	CARETAKER	2019-09-11	9089765433	DELHI
EMP0011	SONAM KAPOOR	1997-03-09	F	CARETAKER	2019-09-11	8765432310	MADHYAPRADESH
EMP0012	JOHN SINGH	1996-03-02	M	STAFF	2019-09-11	6325698547	CHATTISGARH
EMP0013	KIRTI YADAV	1999-05-01	F	STAFF	2019-09-11	7687983456	TELANGANA
EMP0014	JAYPRAKASH JATAV	1995-07-08	M	STAFF	2019-09-12	9636258695	CHANDIGARH
EMP0015	KIRTI SENAN	1997-09-10	F	STAFF	2019-09-12	9658573859	MADHYAPRADESH
EMP0016	ASHUTOSH SHRIVASTAVA	1995-08-07	M	STAFF	2019-09-13	7656439876	RAJASTHAN
EMP0017	ATISHAY JAIN	1996-08-05	M	STAFF	2019-09-13	9632541236	UTTARPRADESH
EMP0018	SOMYA SHRIWAS	1999-08-07	F	MANAGER	2019-09-14	8767980956	UTTARAKHAND
EMP0019	ARCHANA BANSAL	2001-09-08	F	STAFF	2019-09-11	9852631254	HARYANA
EMP0020	AYUSH CHANDIL	2000-06-07	M	STAFF	2019-09-11	6236523649	DELHI
EMP0021	RAHUL DHAKAR	1995-09-08	M	STAFF	2019-09-11	8945678934	MADHYAPRADESH
EMP0022	RAJNI KUMARI	2000-03-26	F	STAFF	2019-09-12	9089765433	CHATTISGARH
EMP0023	RAHUL BANSAL	2000-08-09	M	STAFF	2019-09-12	8765432310	TELANGANA
EMP0024	SAPNA KUMARI	2003-06-23	F	STAFF	2019-09-13	6325698547	CHANDIGARH
EMP0025	NARPAT SINGH	1997-07-30	M	STAFF	2019-09-13	7687983456	MADHYAPRADESH
EMP0026	REKHA KATEWA	2000-06-23	F	STAFF	2019-09-14	9876543210	RAJASTHAN
EMP0027	RADHA KUMARI	1996-06-03	F	STAFF	2019-06-26	4563219870	GUJRAT INDIA


ZOO Management System
Zoo Management System
Add/Update
View
Ticket
Exit
Modify In Staff
Modify In Animal
Modify In Visitor
Add
Update
Delete
clear All Fields
The Form Of Adding/Modify Data
Animal ID :
Animal Type :
Animal Breed :
Gender :
M
Date of Birth :
Weight:
Date Of Arrival:
Diet :
Cage ID:
Emp ID:
Dispose Date :
Result Window


ZOO Management System
Zoo Management System
Add/Update
View
Ticket
Exit
Modify In Cage
Modify In Animal
Modify In Visitor
Add
Update
Delete
clear All Fields
The Form Of Adding/Modify Data
cage ID :
cage capacity :
Result Window



6.MySql Queries

```
CREATE DATABASE ZOOMANAGEMENT;

USE ZOOMANAGEMENT;

-- CREATE staff

CREATE TABLE staff(
        Emp_ID CHAR(7) PRIMARY KEY,
   emp_name VARCHAR(30),
   dob DATE,
   gender CHAR(1),
   work_as VARCHAR(10),
   joining_date DATE,
   mobile VARCHAR(10),

   CHECK (gender = "m" or gender = "f")
);

-- CREATE visitor

CREATE TABLE visitor(
```

```
        visitor_ID CHAR(7) PRIMARY KEY,
    visitor_name VARCHAR(30),
    age TINYINT,
    country VARCHAR(15),
    ticket_type VARCHAR(10),
    mobile VARCHAR(10),
    email VARCHAR(20)
);

-- CREATE ticket AND add foreign key

CREATE TABLE ticket(
        ticket_ID INTEGER PRIMARY KEY AUTO_INCREMENT,
    Emp_ID CHAR(7),
    visitor_ID CHAR(7),
    date_of_visiting DATE,
    amount NUMERIC(5, 2)
);

ALTER TABLE ticket ADD FOREIGN KEY (Emp_ID) REFERENCES staff(Emp_ID) ON
DELETE SET NULL;
ALTER TABLE ticket ADD FOREIGN KEY (visitor_ID) REFERENCES visitor(visitor_ID) ON
DELETE SET NULL;



-- CREATE cage

CREATE TABLE cage(
        cage_ID CHAR(7) PRIMARY KEY,
        capacity TINYINT
);

-- CREATE animal AND add foreign key

CREATE TABLE animal(
        animal_ID CHAR(7) PRIMARY KEY,
    animal_type VARCHAR(15),
    breed VARCHAR(20),
    date_of_arrival DATE,
    gender CHAR(1),
    diet VARCHAR(15),
    weight DECIMAL(3, 2),
    birth_date DATE,
    disposed DATE ,
```

```
   Emp_ID CHAR(7),
   cage_ID CHAR(7)
);


ALTER TABLE animal ADD FOREIGN KEY (Emp_ID) REFERENCES staff(Emp_ID) ON
DELETE SET NULL;
ALTER TABLE animal ADD FOREIGN KEY (cage_ID) REFERENCES cage(cage_ID) ON
DELETE SET NULL;



-- CREATE camera_for_survilence AND add foreign key

CREATE TABLE camera_for_survilence(
        camera_ID CHAR(5) PRIMARY KEY,
   cage_ID CHAR(7),
   installation_date DATE,
   warranty DATE
);


ALTER TABLE camera_for_survilence ADD FOREIGN KEY (cage_ID) REFERENCES
cage(cage_ID) ON DELETE SET NULL;
```

7. Codes

Main.py

```
from ZooStaff import Staff
from Animalclass import Animal
from cage import CageClass
from camera import cameraClass
from viewdetails import view
```

```
from Ticket_Generation import Ticket
from tkinter import *

root = Tk()
#obj = view(root)
#obj = Staff(root)
#obj = Ticket(root)
#obj2 = Animal(root)
#obj3 = CageClass(root)
obj4 = cameraClass(root)
root.mainloop()
```

Staff.py

```
from tkinter import *
from tkinter import ttk

import pymysql


class Staff:
    def __init__(self, root):
        self.root = root #main window for GUI
        self.root.title("ZOO Management System")
        self.root.geometry("1600x1024+0+0")

        title = Label(self.root,text = "Zoo
Management System",bd=10,relief = GROOVE,font =
("times new roman",40,"bold"),bg="yellow",fg="red"
)
        title.pack(side=TOP,fill = X)
        #-------------------------------VARIABLES
```

```
--------------------------
        self.Emp_ID_var = StringVar()
        self.emp_name_var = StringVar()
        self.dob_var = StringVar()
        self.gender_var = StringVar()
        self.work_as_var =StringVar()
        self.joining_date_var =StringVar()
        self.mobile_var =StringVar()


#=========================================MANAGE
FRAMES
=================================================
=======
        Manage_Frame =
Frame(self.root,bd=4,relief=RIDGE,bg="#f0e8c9")

Manage_Frame.place(x=10,y=90,width=1900,height=60)

        AddUpdateBtn =
Button(Manage_Frame,text="Add/Update",font =
("times new roman",20,"bold"),
width=20).grid(row=0,column=0,padx=5,pady=5)
        VieweBtn = Button(Manage_Frame, text="View
", font=("times new roman", 20,
"bold"),width=20).grid(row=0, column=1, padx=5,
pady=5)
        TicketBtn = Button(Manage_Frame,
text="Ticket ", font=("times new roman", 20,
"bold"), width=20).grid(row=0, column=2, padx=5,
pady=5)
        ExitBtn = Button(Manage_Frame, text="Exit
",command=exit, font=("times new roman", 20,
"bold"), width=20).grid(row=0, column=3, padx=700,
pady=5)
```

```
        #
=====================================MENU FRAMES
=================================================
=======
        Menu_Frame =
Frame(self.root,bd=4,relief=RIDGE,bg="#c9e7f0")
        Menu_Frame.place(x=10, y=170, width=380-30,
height=820)

        #+++++++++++++++++++++++++++++++++++Frame 1
+++++++++++++++++++++++++++
        Btn_Frame = Frame(Menu_Frame, bd=4,
relief=RIDGE, bg="#d996c1")
        Btn_Frame.place(x=10, y=10, width=320,
height=400)
        ModifyStaffBtn = Button(Btn_Frame,
text="Modify In Staff", font=("times new roman",
20, "bold"), width=20).grid(row=0, column=0,
padx=10, pady=10)
        ModifyAnimalBtn = Button(Btn_Frame,
text="Modify In Animal", font=("times new roman",
20, "bold"), width=20).grid(row=1,
column=0,padx=10, pady=10)
        ModifyVisitorBtn = Button(Btn_Frame,
text="Modify In Visitor", font=("times new roman",
20, "bold"), width=20).grid(row=2, column=0,
padx=10, pady=10)
        #+++++++++++++++++++++++++++++++Frame 2
+++++++++++++++++++++++++++
        Btn_Frame2 = Frame(Menu_Frame, bd=4,
relief=RIDGE, bg="#d996c1")
        Btn_Frame2.place(x=10, y=430, width=320,
height=370)
        AddBtn = Button(Btn_Frame2,
text="Add",command =self.addStaff, font=("times
```

```
new roman", 20, "bold"), width=20).grid(
            row=0, column=0, padx=10, pady=10)
        UpdateBtn = Button(Btn_Frame2,
text="Update",command =self.updatestaff,
font=("times new roman", 20, "bold"),

width=20).grid(row=1, column=0, padx=10, pady=10)
        DeleteBtn = Button(Btn_Frame2,
text="Delete",command = self.delete_staff,
font=("times new roman", 20, "bold"),

width=20).grid(row=2, column=0, padx=10, pady=10)
        ClearBtn = Button(Btn_Frame2, text="clear
All Fields",command=self.clear, font=("times new
roman", 20, "bold"),
                                width=20).grid(row=3,
column=0, padx=10, pady=10)


        #
=========================================FORM FRAMES
=================================================
=======
        Form_Frame = Frame(self.root, bd=4,
relief=RIDGE, bg="#202842")
        Form_Frame.place(x=400-30, y=170,
width=380*2, height=820)

        mtitle = Label(Form_Frame,text="The Form Of
Adding/Modify Data
",bg="crimson",fg="white",font=("times new roman",
35, "bold") )
        mtitle.grid(row=0, columnspan=4, pady=10,
padx=20,sticky="w")
```

```
#>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>Employee ID
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        lbl_EmpID = Label(Form_Frame,
text="Employee ID : ", bg="#202842",
fg="white",font=("times new roman", 20, "bold"))
        lbl_EmpID.grid(row=1, column=0, pady=10,
padx=10+5,sticky="w")

        txt_EmpID =
Entry(Form_Frame,textvariable=self.Emp_ID_var,
font=("times new roman", 20,
"bold"),bd=5,relief=GROOVE)
        txt_EmpID.grid(row=1, column=1, pady=10,
padx=5, sticky="w")
        #>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>Emp Name
>>>>>>>>>>>>>>>>>>>>>>>>>>>
        lbl_EmpName = Label(Form_Frame,
text="Employee Name : ", bg="#202842", fg="white",
                         font=("times new roman",
20, "bold"))
        lbl_EmpName.grid(row=2, column=0, pady=10,
padx=10 + 5, sticky="w")

        txt_EmpName = Entry(Form_Frame,textvariable
= self.emp_name_var, font=("times new roman", 20,
"bold"), bd=5, relief=GROOVE)
        txt_EmpName.grid(row=2, column=1, pady=10,
padx=5, sticky="w")
        # >>>>>>>>>>>>>>>>>>>>>>>COMBOBOX FOR
GENDER >>>>>>>>>>>>>>>>>>>>>>>>>
        lbl_Empgender = Label(Form_Frame,
text="Gender : ", bg="#202842", fg="white",
                             font=("times new
roman", 20, "bold"))
        lbl_Empgender.grid(row=3, column=0,
pady=10, padx=10 + 5, sticky="w")
```

```
        Empgender =
ttk.Combobox(Form_Frame,textvariable=self.gender_v
ar, font=("times new roman", 15), state
='readonly')
        Empgender['values'] = ('M', 'F', 'O')
        Empgender.current(0)
        Empgender.grid(row=3, column=1, pady=10,
padx= 5, sticky="w")

        #>>>>>>>>>>>>>>>>>>>>>>>>>>>DATE OF BIRTH
>>>>>>>>>>>>>>>>>>>>>>>>>
        lbl_Empdob = Label(Form_Frame, text="Date
of Birth : ", bg="#202842", fg="white",
                                font=("times new roman",
20, "bold"))
        lbl_Empdob.grid(row=4, column=0, pady=10,
padx=15, sticky="w")

        txt_Empdob = Entry(Form_Frame,textvariable
= self.dob_var, font=("times new roman", 20,
"bold"), bd=5, relief=GROOVE)
        txt_Empdob.grid(row=4, column=1, pady=10,
padx=5, sticky="w")


        #>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>WORK AS
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        lbl_WorkAs = Label(Form_Frame,text="Work As
: ", bg="#202842", fg="white",font=("times new
roman", 20, "bold"))
        lbl_WorkAs.grid(row=7, column=0, pady=10,
padx=10 + 5, sticky="w")

        txt_WorkAs =
Entry(Form_Frame,textvariable=self.work_as_var,
```

```
font=("times new roman", 20, "bold"), bd=5,
relief=GROOVE)
        txt_WorkAs.grid(row=7, column=1, pady=10,
padx=5, sticky="w")

        #>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>JOINING
DATE >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        lbl_Join = Label(Form_Frame, text="Joining
Date: ", bg="#202842", fg="white",
                                font=("times new roman",
20, "bold"))
        lbl_Join.grid(row=8, column=0, pady=10,
padx=10 + 5, sticky="w")

        txt_Join =
Entry(Form_Frame,textvariable=self.joining_date_va
r, font=("times new roman", 20, "bold"), bd=5,
relief=GROOVE)
        txt_Join.grid(row=8, column=1, pady=10,
padx=5, sticky="w")



        #>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>MOBILE
FIELD >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        lbl_Mobile = Label(Form_Frame, text="Mobile
: ", bg="#202842", fg="white",
                                font=("times new roman",
20, "bold"))
        lbl_Mobile.grid(row=9, column=0, pady=10,
padx=10 + 5, sticky="w")

        txt_Mobile =
Entry(Form_Frame,textvariable=self.mobile_var,
font=("times new roman", 20, "bold"), bd=5,
relief=GROOVE)
```

```
        txt_Mobile.grid(row=9, column=1, pady=10,
padx=5, sticky="w")


#>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>ADDRESS
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        lbl_Address = Label(Form_Frame,
text="Address : ", bg="#202842", fg="white",
                                font=("times new roman",
20, "bold"))
        lbl_Address.grid(row=10, column=0, pady=10,
padx=10 + 5, sticky="w")

        self.txt_Address=Text(Form_Frame, width
=30,height = 4, font=("times new roman", 20,
"bold"), bd=5, relief=GROOVE)
        self.txt_Address.grid(row=10, column=1,
pady=10, padx=5, sticky="w")

        #
=========================================RESULT
FRAMES
=====================================================
=======
        Result_Frame = Frame(self.root, bd=4,
relief=RIDGE, bg="#c9e7f0")
        Result_Frame.place(x=400+380*2+10-30,
y=170, width=360+380+30, height=820-80)

        Rtitle = Label(Result_Frame, text="Result
Window ", bg="crimson", fg="white", font=("times
new roman", 35, "bold"))
        Rtitle.grid(row=0, columnspan=4, pady=10,
padx=20, sticky="w")
```

```
#Functions
    #-----------------------------Function to Add
in Staff ---------------------------------
    def addStaff(self):
        con = pymysql.connect(user='root',
password='Password@21',
                              host='localhost',
database='ZOOMANAGEMENT')
        cur = con.cursor()
        query= "insert into staff(Emp_ID,
emp_name,dob, gender, work_as,
joining_date,mobile,Address) values('{}', '{}',
'{}','{}','{}', '{}',
'{}','{}')".format(self.Emp_ID_var.get(),

self.emp_name_var.get(),

self.dob_var.get(),

self.gender_var.get(),

self.work_as_var.get(),

self.joining_date_var.get(),

self.mobile_var.get(),

self.txt_Address.get('1.0',END))
        print("Query Executed")

        cur.execute(query)
        con.commit()
        con.close()
    #--------------------Function to Update in
staff ----------------------------------------
```

```
    def updatestaff(self):
        con = pymysql.connect(user='root',
password='Password@21',
                              host='localhost',
database='ZOOMANAGEMENT')
        cur = con.cursor()
        query = "update staff set emp_name = '{}',
dob = '{}', gender = '{}', work_as =
'{}',joining_date = '{}', mobile = '{}' where
Emp_ID = '{}'".format(
            self.emp_name_var.get(),
self.dob_var.get(), self.gender_var.get(),
            self.work_as_var.get(),
self.joining_date_var.get(),
self.mobile_var.get(),self.Emp_ID_var.get())
        print("Query Executed")

        cur.execute(query)
        con.commit()
        con.close()
   #Delete Staff
    def delete_staff(self):
        con = pymysql.connect(user='root',
password='Password@21', host='localhost',
database='ZOOMANAGEMENT')
        cur = con.cursor()
        query = "delete from staff where Emp_ID =
'{}'".format(self.Emp_ID_var.get())
        con.commit()
        con.close()
   #-----------------------Function To Clear All
Field ----------------------------------
    def clear(self):
       self.Emp_ID_var.set(" ")
       self.emp_name_var.set(" ")
```

```
        self.dob_var.set(" ")
        self.gender_var.set(" ")
        self.work_as_var.set(" ")
        self.joining_date_var.set(" ")
        self.mobile_var.set(" ")

    # ---------------------------Function To Exit
The Program -----------------------------
    def exit(self):
        exit()
```

Animalclass.py

```
from tkinter import *
from tkinter import ttk

import pymysql


class Animal:
    def __init__(self, root):
        self.root = root #main window for GUI
        self.root.title("ZOO Management System")
        self.root.geometry("1600x1024+0+0")

        title = Label(self.root,text = "Zoo
Management System",bd=10,relief = GROOVE,font =
("times new roman",40,"bold"),bg="yellow",fg="red"
)
        title.pack(side=TOP,fill = X)
        #--------------------------------VARIABLES
-------------------------
        self.animal_ID_var = StringVar()
```

```
        self.animal_type_var = StringVar()
        self.animal_breed_var = StringVar()
        self.ani_dob_var = StringVar()
        self.animal_dateofarrival_var = StringVar()
        self.gender_var = StringVar()
        self.diet_var =StringVar()
        self.weight_var =StringVar()
        self.disposedate_var =StringVar()
        self.cageID = StringVar()
        self.Emp_ID = StringVar()


#=====================================MANAGE
FRAMES
=================================================
=======
        Manage_Frame =
Frame(self.root,bd=4,relief=RIDGE,bg="#f0e8c9")

Manage_Frame.place(x=10,y=90,width=1900,height=60)

        AddUpdateBtn =
Button(Manage_Frame,text="Add/Update",command=self
.addanimal,font = ("times new roman",20,"bold"),
width=20).grid(row=0,column=0,padx=5,pady=5)
        VieweBtn = Button(Manage_Frame, text="View
", font=("times new roman", 20,
"bold"),width=20).grid(row=0, column=1, padx=5,
pady=5)
        TicketBtn = Button(Manage_Frame,
text="Ticket ", font=("times new roman", 20,
"bold"), width=20).grid(row=0, column=2, padx=5,
pady=5)
        ExitBtn = Button(Manage_Frame, text="Exit
",command=exit, font=("times new roman", 20,
"bold"), width=20).grid(row=0, column=3, padx=700,
```

```
pady=5)


        #
=====================================MENU FRAMES
=================================================
=======
        Menu_Frame =
Frame(self.root,bd=4,relief=RIDGE,bg="#c9e7f0")
        Menu_Frame.place(x=10, y=170, width=380-30,
height=820)

        #+++++++++++++++++++++++++++++++++++Frame 1
+++++++++++++++++++++++++++
        Btn_Frame = Frame(Menu_Frame, bd=4,
relief=RIDGE, bg="#d996c1")
        Btn_Frame.place(x=10, y=10, width=320,
height=400)
        ModifyStaffBtn = Button(Btn_Frame,
text="Modify In Staff", font=("times new roman",
20, "bold"), width=20).grid(row=0, column=0,
padx=10, pady=10)
        ModifyAnimalBtn = Button(Btn_Frame,
text="Modify In Animal", font=("times new roman",
20, "bold"), width=20).grid(row=1,
column=0,padx=10, pady=10)
        ModifyVisitorBtn = Button(Btn_Frame,
text="Modify In Visitor", font=("times new roman",
20, "bold"), width=20).grid(row=2, column=0,
padx=10, pady=10)
        #++++++++++++++++++++++++++++++++Frame 2
+++++++++++++++++++++++++++
        Btn_Frame2 = Frame(Menu_Frame, bd=4,
relief=RIDGE, bg="#d996c1")
        Btn_Frame2.place(x=10, y=430, width=320,
height=370)
```

```
        AddBtn = Button(Btn_Frame2,
text="Add",command =self.addanimal, font=("times
new roman", 20, "bold"), width=20).grid(
            row=0, column=0, padx=10, pady=10)
        UpdateBtn = Button(Btn_Frame2,
text="Update",command =self.updateanimal,
font=("times new roman", 20, "bold"),

width=20).grid(row=1, column=0, padx=10, pady=10)
        DeleteBtn = Button(Btn_Frame2,
text="Delete",command = self.delete_animal,
font=("times new roman", 20, "bold"),

width=20).grid(row=2, column=0, padx=10, pady=10)
        ClearBtn = Button(Btn_Frame2, text="clear
All Fields",command=self.clear, font=("times new
roman", 20, "bold"),
                            width=20).grid(row=3,
column=0, padx=10, pady=10)


        #
=========================================FORM FRAMES
=================================================
=======
        Form_Frame = Frame(self.root, bd=4,
relief=RIDGE, bg="#202842")
        Form_Frame.place(x=400-30, y=170,
width=380*2, height=820)

        mtitle = Label(Form_Frame,text="The Form Of
Adding/Modify
Data",bg="crimson",fg="white",font=("times new
roman", 35, "bold") )
        mtitle.grid(row=0, columnspan=4, pady=10,
padx=20,sticky="w")
```

```
#>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>Animal ID
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
       lbl_AniID = Label(Form_Frame, text="Animal
ID : ", bg="#202842", fg="white",font=("times new
roman", 20, "bold"))
       lbl_AniID.grid(row=1, column=0, pady=10,
padx=10+5,sticky="w")

       txt_AniID =
Entry(Form_Frame,textvariable=self.animal_ID_var,
font=("times new roman", 20,
"bold"),bd=5,relief=GROOVE)
       txt_AniID.grid(row=1, column=1, pady=10,
padx=5, sticky="w")

       #>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>Animal
Type >>>>>>>>>>>>>>>>>>>>>>>>>>>
       lbl_AniType = Label(Form_Frame,
text="Animal Type : ", bg="#202842", fg="white",
                         font=("times new roman",
20, "bold"))
       lbl_AniType.grid(row=2, column=0, pady=10,
padx=10 + 5, sticky="w")

       txt_AniType = Entry(Form_Frame,textvariable
= self.animal_type_var, font=("times new roman",
20, "bold"), bd=5, relief=GROOVE)
       txt_AniType.grid(row=2, column=1, pady=10,
padx=5, sticky="w")


#>>>>>>>>>>>>>>>>>>>>>>>>>Breed>>>>>>>>>>>>>>>>>>>
>>>>>>>>>.
       lbl_Animal_breed = Label(Form_Frame,
```

```
text="Animal Breed : ", bg="#202842", fg="white",
                                font=("times new
roman", 20, "bold"))
        lbl_Animal_breed.grid(row=3, column=0,
pady=10, padx=10 + 5, sticky="w")

        txt_Animal_breed = Entry(Form_Frame,
textvariable=self.animal_breed_var, font=("times
new roman", 20, "bold"), bd=5,
                                relief=GROOVE)
        txt_Animal_breed.grid(row=3, column=1,
pady=10, padx=5, sticky="w")

        # >>>>>>>>>>>>>>>>>>>>>>>>>COMBOBOX FOR
GENDER >>>>>>>>>>>>>>>>>>>>>>>>>>
        lbl_Anigender = Label(Form_Frame,
text="Gender : ", bg="#202842", fg="white",
                                  font=("times new
roman", 20, "bold"))
        lbl_Anigender.grid(row=4, column=0,
pady=10, padx=10 + 5, sticky="w")

        Anigender =
ttk.Combobox(Form_Frame,textvariable=self.gender_v
ar, font=("times new roman", 15), state
='readonly')
        Anigender['values'] = ('M', 'F')
        Anigender.current(0)
        Anigender.grid(row=4, column=1, pady=10,
padx= 5, sticky="w")

        #>>>>>>>>>>>>>>>>>>>>>>>>>>>>DATE OF BIRTH
>>>>>>>>>>>>>>>>>>>>>>>>>>
        lbl_Anidob = Label(Form_Frame, text="Date
of Birth : ", bg="#202842", fg="white",
                              font=("times new roman",
```

```
20, "bold"))
        lbl_Anidob.grid(row=5, column=0, pady=10,
padx=15, sticky="w")

        txt_Anidob = Entry(Form_Frame,textvariable
= self.ani_dob_var, font=("times new roman", 20,
"bold"), bd=5, relief=GROOVE)
        txt_Anidob.grid(row=5, column=1, pady=10,
padx=5, sticky="w")


        #>>>>>>>>>>>>>>>>>>r>>>>>>>>>>>>>>>>Weight
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        lbl_Weight = Label(Form_Frame,text="Weight:
", bg="#202842", fg="white",font=("times new
roman", 20, "bold"))
        lbl_Weight.grid(row=7, column=0, pady=10,
padx=10 + 5, sticky="w")

        txt_Weight =
Entry(Form_Frame,textvariable=self.weight_var,
font=("times new roman", 20, "bold"), bd=5,
relief=GROOVE)
        txt_Weight.grid(row=7, column=1, pady=10,
padx=5, sticky="w")

        #>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>JOINING
DATE >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        lbl_Join = Label(Form_Frame, text="Date Of
Arrival: ", bg="#202842", fg="white",
                                font=("times new roman",
20, "bold"))
        lbl_Join.grid(row=8, column=0, pady=10,
padx=10 + 5, sticky="w")

        txt_Join =
```

```
Entry(Form_Frame,textvariable=self.animal_dateofar
rival_var, font=("times new roman", 20, "bold"),
bd=5, relief=GROOVE)
        txt_Join.grid(row=8, column=1, pady=10,
padx=5, sticky="w")




#>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>Diet>>>>>>>>>>>>
>>>>>>>>>>>>>>>>>>>>
        lbl_Diet = Label(Form_Frame, text="Diet  :
", bg="#202842", fg="white",
                                font=("times new roman",
20, "bold"))
        lbl_Diet.grid(row=9, column=0, pady=10,
padx=10 + 5, sticky="w")

        txt_Diet =
Entry(Form_Frame,textvariable=self.diet_var,
font=("times new roman", 20, "bold"), bd=5,
relief=GROOVE)
        txt_Diet.grid(row=9, column=1, pady=10,
padx=5, sticky="w")
        # >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>Dispose
Date >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        lbl_Dispose_Date = Label(Form_Frame,
text="Dispose Date   : ", bg="#202842",
fg="white",
                                font=("times new roman",
20, "bold"))
        lbl_Dispose_Date.grid(row=12, column=0,
pady=10, padx=10 + 5, sticky="w")

        txt_Dispose_Date = Entry(Form_Frame,
textvariable=self.disposedate_var, font=("times
```

```
new roman", 20, "bold"), bd=5,
                                relief=GROOVE)
        txt_Dispose_Date.grid(row=12, column=1,
pady=10, padx=5, sticky="w")


        # >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>Cage
ID >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        lbl_CageID = Label(Form_Frame, text="Cage
ID: ", bg="#202842", fg="white",
                                font=("times new
roman", 20, "bold"))
        lbl_CageID.grid(row=10, column=0, pady=10,
padx=10 + 5, sticky="w")

        txt_CageID = Entry(Form_Frame,textvariable
= self.cageID, width=10, font=("times new roman",
20, "bold"), bd=5,
                                    relief=GROOVE)
        txt_CageID.grid(row=10, column=1, pady=10,
padx=5, sticky="w")
        #>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>Emp ID
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        lbl_EmpID = Label(Form_Frame, text="Emp ID:
", bg="#202842", fg="white",
                                font=("times new roman",
20, "bold"))
        lbl_EmpID.grid(row=11, column=0, pady=10,
padx=10 + 5, sticky="w")

        txt_EmpID =
Entry(Form_Frame,textvariable=self.Emp_ID,
width=10, font=("times new roman", 20, "bold"),
bd=5,
                                    relief=GROOVE)
        txt_EmpID.grid(row=11, column=1, pady=10,
```

```
padx=5, sticky="w")


        #
=====================================RESULT
FRAMES
=================================================
=======
        Result_Frame = Frame(self.root, bd=4,
relief=RIDGE, bg="#c9e7f0")
        Result_Frame.place(x=400+380*2+10-30,
y=170, width=360+380+30, height=820-80)

        Rtitle = Label(Result_Frame, text="Result
Window ", bg="crimson", fg="white", font=("times
new roman", 35, "bold"))
        Rtitle.grid(row=0, columnspan=4, pady=10,
padx=20, sticky="w")



#Functions
    #-----------------------------Function to Add
in Animal --------------------------------
    def addanimal(self):
        con = pymysql.connect(user='root',
password='Password@21',
                              host='localhost',
database='ZOOMANAGEMENT')
        cur = con.cursor()
        query= "insert into
animal(animal_ID,animal_type,breed,date_of_arrival
,gender,diet,weight,birth_date,disposed,Emp_ID,cag
e_ID)
values('{}','{}','{}','{}','{}','{}','{}','{}','{}
','{}','{}')".format(self.animal_ID_var.get(),self
```

```
.animal_type_var.get(),self.animal_breed_var.get()
,self.animal_dateofarrival_var.get(),self.gender_v
ar.get(),self.diet_var.get(),self.weight_var.get()
,self.ani_dob_var.get(),self.disposedate_var.get()
,self.Emp_ID.get(),self.cageID.get())


        print("Query Executed")

        cur.execute(query)
        con.commit()
        con.close()
    #--------------------Function to Update in
animal ----------------------------------------

    def updateanimal(self):
        con = pymysql.connect(user='root',
password='Password@21',
                              host='localhost',
database='ZOOMANAGEMENT')
        cur = con.cursor()
        query = "update animal set animal_ID =
'{}', animal_type = '{}', breed = '{}',
date_of_arrival = '{}',gender = '{}',diet = '{}',
weight = '{}', birth_date = '{}', disposed = '{}',
Emp_ID = '{}', cage_ID = '{}' where animal_ID =
'{}'".format(

self.animal_ID.get(),self.animal_type_var.get(),
self.animal_breed_var.get(),
self.ani_dob_var.get(),    self.gender_var.get(),
self.animal_dateofarrival_var.get(),
self.diet_var.get(),
self.weight_var.get(),self.disposedate_var.get(),s
elf.cageID.get(),self.Emp_ID.get())
```

```
        print("Query Executed")

        cur.execute(query)
        con.commit()
        con.close()


    #Delete animal

    def delete_animal(self):
        con = pymysql.connect(user='root',
password='Password@21', host='localhost',
database='ZOOMANAGEMENT')
        cur = con.cursor()
        query = "delete from animal where
animal_ID =
'{}'".format(self.animal_breed_var.get())
        con.commit()
        con.close()

    #------------------------Function To Clear All
Field ----------------------------------
    def clear(self):
        self.animal_ID_var.set("")
        self.animal_type_var.set("")
        self.animal_breed_var.set("")
        self.ani_dob_var.set("")
        self.animal_dateofarrival_var.set("")
        self.gender_var.set("")
        self.diet_var.set("")
        self.weight_var.set("")
        self.disposedate_var.set("")
        self.cageID.set("")
        self.Emp_ID.set("")

    # ---------------------------Function To Exit
```

```
The Program ------------------------------
    def exit(self):
        exit()
```

Camera.py

```
from tkinter import *
from tkinter import ttk

import pymysql


class cameraClass:
    def __init__(self, root):
        self.root = root #main window for GUI
        self.root.title("ZOO Management System")
        self.root.geometry("1600x1024+0+0")

        title = Label(self.root,text = "Zoo
Management System",bd=10,relief = GROOVE,font =
("times new roman",40,"bold"),bg="yellow",fg="red"
)
        title.pack(side=TOP,fill = X)
        #--------------------------------VARIABLES
-------------------------
        self.camera_ID_var = StringVar()
        self.cage_ID_var = StringVar()
        self.Installation_date_var = StringVar()
        self.warranty_var = StringVar()
```

```
#=========================================MANAGE
FRAMES
=================================================
=======
        Manage_Frame =
Frame(self.root,bd=4,relief=RIDGE,bg="#f0e8c9")

Manage_Frame.place(x=10,y=90,width=1900,height=60)

        AddUpdateBtn =
Button(Manage_Frame,text="Add/Update",font =
("times new roman",20,"bold"),
width=20).grid(row=0,column=0,padx=5,pady=5)
        VieweBtn = Button(Manage_Frame, text="View
", font=("times new roman", 20,
"bold"),width=20).grid(row=0, column=1, padx=5,
pady=5)
        TicketBtn = Button(Manage_Frame,
text="Ticket ", font=("times new roman", 20,
"bold"), width=20).grid(row=0, column=2, padx=5,
pady=5)
        ExitBtn = Button(Manage_Frame, text="Exit
",command=exit, font=("times new roman", 20,
"bold"), width=20).grid(row=0, column=3, padx=700,
pady=5)


        #
=========================================MENU FRAMES
=================================================
=======
        Menu_Frame =
Frame(self.root,bd=4,relief=RIDGE,bg="#c9e7f0")
        Menu_Frame.place(x=10, y=170, width=380-30,
height=820)
```

```
        #+++++++++++++++++++++++++++++++++++Frame 1
++++++++++++++++++++++++++
        Btn_Frame = Frame(Menu_Frame, bd=4,
relief=RIDGE, bg="#d996c1")
        Btn_Frame.place(x=10, y=10, width=320,
height=400)
        ModifycameraBtn = Button(Btn_Frame,
text="Modify In camera", font=("times new roman",
20, "bold"), width=20).grid(row=0, column=0,
padx=10, pady=10)
        ModifyAnimalBtn = Button(Btn_Frame,
text="Modify In Animal", font=("times new roman",
20, "bold"), width=20).grid(row=1,
column=0,padx=10, pady=10)
        ModifyVisitorBtn = Button(Btn_Frame,
text="Modify In Visitor", font=("times new roman",
20, "bold"), width=20).grid(row=2, column=0,
padx=10, pady=10)
        #+++++++++++++++++++++++++++++++Frame 2
+++++++++++++++++++++++++++
        Btn_Frame2 = Frame(Menu_Frame, bd=4,
relief=RIDGE, bg="#d996c1")
        Btn_Frame2.place(x=10, y=430, width=320,
height=370)
        AddBtn = Button(Btn_Frame2,
text="Add",command =self.addcamera, font=("times
new roman", 20, "bold"), width=20).grid(
            row=0, column=0, padx=10, pady=10)
        UpdateBtn = Button(Btn_Frame2,
text="Update",command =self.updatecamera,
font=("times new roman", 20, "bold"),

width=20).grid(row=1, column=0, padx=10, pady=10)
        DeleteBtn = Button(Btn_Frame2,
text="Delete",command = self.delete_camera,
```

```
font=("times new roman", 20, "bold"),

width=20).grid(row=2, column=0, padx=10, pady=10)
        ClearBtn = Button(Btn_Frame2, text="clear
All Fields",command=self.clear, font=("times new
roman", 20, "bold"),
                                width=20).grid(row=3,
column=0, padx=10, pady=10)


        #
=====================================FORM FRAMES
=================================================
=======
        Form_Frame = Frame(self.root, bd=4,
relief=RIDGE, bg="#202842")
        Form_Frame.place(x=400-30, y=170,
width=380*2, height=820)

        mtitle = Label(Form_Frame,text="The Form Of
Adding/Modify Data
",bg="crimson",fg="white",font=("times new roman",
35, "bold") )
        mtitle.grid(row=0, columnspan=4, pady=10,
padx=20,sticky="w")


#>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>camera ID
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        lbl_camera = Label(Form_Frame, text="Camera
ID : ", bg="#202842", fg="white",font=("times new
roman", 20, "bold"))
        lbl_camera.grid(row=1, column=0, pady=10,
padx=10+5,sticky="w")

        txt_camera =
```

```
Entry(Form_Frame,textvariable=self.camera_ID_var,
font=("times new roman", 20,
"bold"),bd=5,relief=GROOVE)
        txt_camera.grid(row=1, column=1, pady=10,
padx=5, sticky="w")
        #>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>cage ID
>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        lbl_cageID = Label(Form_Frame, text="Cage
ID : ", bg="#202842", fg="white",
                                font=("times new roman",
20, "bold"))
        lbl_cageID.grid(row=2, column=0, pady=10,
padx=10 + 5, sticky="w")

        txt_cageID = Entry(Form_Frame,textvariable
= self.cage_ID_var, font=("times new roman", 20,
"bold"), bd=5, relief=GROOVE)
        txt_cageID.grid(row=2, column=1, pady=10,
padx=5, sticky="w")
        # >>>>>>>>>>>>>>>>>>>>>>>>>>>>>COMBOBOX FOR
warranty >>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        lbl_Installation_date = Label(Form_Frame,
text="Installation Date : ", bg="#202842",
fg="white",

font=("times new roman", 20, "bold"))
        lbl_Installation_date.grid(row=3, column=0,
pady=10, padx=15, sticky="w")

        txt_Installation_date = Entry(Form_Frame,
textvariable=self.Installation_date_var,

font=("times new roman", 20, "bold"), bd=5,
relief=GROOVE)
        txt_Installation_date.grid(row=3, column=1,
pady=10, padx=5, sticky="w")
```

```
        #>>>>>>>>>>>>>>>>>>>>>>>>>>>>DATE OF BIRTH
>>>>>>>>>>>>>>>>>>>>>>>>>
        lbl_Warrenty_date = Label(Form_Frame,
text="Warrenty : ", bg="#202842", fg="white",
                                font=("times new roman",
20, "bold"))
        lbl_Warrenty_date.grid(row=4, column=0,
pady=10, padx=15, sticky="w")

        txt_Warrenty_date =
Entry(Form_Frame,textvariable =
self.Installation_date_var, font=("times new
roman", 20, "bold"), bd=5, relief=GROOVE)
        txt_Warrenty_date.grid(row=4, column=1,
pady=10, padx=5, sticky="w")




        #
=========================================RESULT
FRAMES
=================================================
=======
        Result_Frame = Frame(self.root, bd=4,
relief=RIDGE, bg="#c9e7f0")
        Result_Frame.place(x=400+380*2+10-30,
y=170, width=360+380+30, height=820-80)

        Rtitle = Label(Result_Frame, text="Result
Window ", bg="crimson", fg="white", font=("times
new roman", 35, "bold"))
        Rtitle.grid(row=0, columnspan=4, pady=10,
padx=20, sticky="w")
```

```
#Functions
    #-----------------------------Function to Add
in camera ---------------------------------
    def addcamera(self):
        con = pymysql.connect(user='root',
password='Password@21',
                              host='localhost',
database='ZOOMANAGEMENT')
        cur = con.cursor()
        query= "insert into camera(camera_ID,
cage_ID,Installation_date, warranty) values('{}',
'{}', '{}')".format(self.camera_ID_var.get(),

self.cage_ID_var.get(),

self.Installation_date_var.get(),

self.warranty_var.get(),

)
        print("Query Executed")

        cur.execute(query)
        con.commit()
        con.close()
    #--------------------Function to Update in
camera ----------------------------------------

    def updatecamera(self):
         con = pymysql.connect(user='root',
password='Password@21',
                              host='localhost',
database='ZOOMANAGEMENT')
         cur = con.cursor()
```

```
        query = "update camera set cage_ID = '{}',
Installation_date = '{}', warranty = '{}' where
camera_ID = '{}'".format(
            self.cage_ID_var.get(),
self.Installation_date_var.get(),
self.warranty_var.get(),
            self.camera_ID_var.get())
        print("Query Executed")

        cur.execute(query)
        con.commit()
        con.close()
    #Delete camera
    def delete_camera(self):
        con = pymysql.connect(user='root',
password='Password@21', host='localhost',
database='ZOOMANAGEMENT')
        cur = con.cursor()
        query = "delete from camera where
camera_ID = '{}'".format(self.camera_ID_var.get())
        con.commit()
        con.close()
    #-----------------------Function To Clear All
Field ---------------------------------
    def clear(self):
        self.camera_ID_var.set(" ")
        self.cage_ID_var.set(" ")
        self.Installation_date_var.set(" ")
        self.warranty_var.set(" ")


    # ---------------------------Function To Exit
The Program -----------------------------
    def exit(self):
        exit()
```

Cage.py

```
from tkinter import *
from tkinter import ttk

import pymysql


class CageClass:
    def __init__(self, root):
        self.root = root #main window for GUI
        self.root.title("ZOO Management System")
        self.root.geometry("1600x1024+0+0")

        title = Label(self.root,text = "Zoo
Management System",bd=10,relief = GROOVE,font =
("times new roman",40,"bold"),bg="yellow",fg="red"
)
        title.pack(side=TOP,fill = X)
        #-------------------------------VARIABLES
-------------------------
        self.Cage_ID_Var = StringVar()
        self.capacity_var = StringVar()



#=====================================MANAGE
FRAMES
=================================================
=======
        Manage_Frame =
Frame(self.root,bd=4,relief=RIDGE,bg="#f0e8c9")
```

```
Manage_Frame.place(x=10,y=90,width=1900,height=60)

        AddUpdateBtn =
Button(Manage_Frame,text="Add/Update",font =
("times new roman",20,"bold"),
width=20).grid(row=0,column=0,padx=5,pady=5)
        VieweBtn = Button(Manage_Frame, text="View
", font=("times new roman", 20,
"bold"),width=20).grid(row=0, column=1, padx=5,
pady=5)
        TicketBtn = Button(Manage_Frame,
text="Ticket ", font=("times new roman", 20,
"bold"), width=20).grid(row=0, column=2, padx=5,
pady=5)
        ExitBtn = Button(Manage_Frame, text="Exit
",command=exit, font=("times new roman", 20,
"bold"), width=20).grid(row=0, column=3, padx=700,
pady=5)


        #
=====================================MENU FRAMES
=================================================
=======
        Menu_Frame =
Frame(self.root,bd=4,relief=RIDGE,bg="#c9e7f0")
        Menu_Frame.place(x=10, y=170, width=380-30,
height=820)

        #+++++++++++++++++++++++++++++++++++Frame 1
+++++++++++++++++++++++++++
        Btn_Frame = Frame(Menu_Frame, bd=4,
relief=RIDGE, bg="#d996c1")
        Btn_Frame.place(x=10, y=10, width=320,
height=400)
        ModifyCageBtn = Button(Btn_Frame,
```

```
text="Modify In Cage", font=("times new roman",
20, "bold"), width=20).grid(row=0, column=0,
padx=10, pady=10)
        ModifyAnimalBtn = Button(Btn_Frame,
text="Modify In Animal", font=("times new roman",
20, "bold"), width=20).grid(row=1,
column=0,padx=10, pady=10)
        ModifyVisitorBtn = Button(Btn_Frame,
text="Modify In Visitor", font=("times new roman",
20, "bold"), width=20).grid(row=2, column=0,
padx=10, pady=10)
        #+++++++++++++++++++++++++++++++Frame 2
++++++++++++++++++++++++++++++
        Btn_Frame2 = Frame(Menu_Frame, bd=4,
relief=RIDGE, bg="#d996c1")
        Btn_Frame2.place(x=10, y=430, width=320,
height=370)
        AddBtn = Button(Btn_Frame2,
text="Add",command =self.addCage, font=("times new
roman", 20, "bold"), width=20).grid(
            row=0, column=0, padx=10, pady=10)
        UpdateBtn = Button(Btn_Frame2,
text="Update",command =self.updateCage,
font=("times new roman", 20, "bold"),

width=20).grid(row=1, column=0, padx=10, pady=10)
        DeleteBtn = Button(Btn_Frame2,
text="Delete",command = self.delete_cage,
font=("times new roman", 20, "bold"),

width=20).grid(row=2, column=0, padx=10, pady=10)
        ClearBtn = Button(Btn_Frame2, text="clear
All Fields",command=self.clear, font=("times new
roman", 20, "bold"),
                            width=20).grid(row=3,
column=0, padx=10, pady=10)
```

```
        #
=====================================FORM FRAMES
=================================================
=======
        Form_Frame = Frame(self.root, bd=4,
relief=RIDGE, bg="#202842")
        Form_Frame.place(x=400-30, y=170,
width=380*2, height=820)

        mtitle = Label(Form_Frame,text="The Form Of
Adding/Modify Data
",bg="crimson",fg="white",font=("times new roman",
35, "bold") )
        mtitle.grid(row=0, columnspan=4, pady=10,
padx=20,sticky="w")




#>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>CageID
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        lbl_cageID = Label(Form_Frame, text="cage
ID : ", bg="#202842", fg="white",font=("times new
roman", 20, "bold"))
        lbl_cageID.grid(row=1, column=0, pady=10,
padx=10+5,sticky="w")

        txt_cageID =
Entry(Form_Frame,textvariable=self.Cage_ID_Var,
font=("times new roman", 20,
"bold"),bd=5,relief=GROOVE)
        txt_cageID.grid(row=1, column=1, pady=10,
padx=5, sticky="w")
        #>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Capacity
```

```
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        lbl_capacity = Label(Form_Frame, text="cage
capacity : ", bg="#202842", fg="white",
                                font=("times new roman",
20, "bold"))
        lbl_capacity.grid(row=2, column=0, pady=10,
padx=10 + 5, sticky="w")

        txt_capacity =
Entry(Form_Frame,textvariable = self.capacity_var,
font=("times new roman", 20, "bold"), bd=5,
relief=GROOVE)
        txt_capacity.grid(row=2, column=1, pady=10,
padx=5, sticky="w")



        #
=====================================RESULT
FRAMES
=================================================
=======
        Result_Frame = Frame(self.root, bd=4,
relief=RIDGE, bg="#c9e7f0")
        Result_Frame.place(x=400+380*2+10-30,
y=170, width=360+380+30, height=820-80)

        Rtitle = Label(Result_Frame, text="Result
Window ", bg="crimson", fg="white", font=("times
new roman", 35, "bold"))
        Rtitle.grid(row=0, columnspan=4, pady=10,
padx=20, sticky="w")



#Functions
```

```
    #-----------------------------Function to Add
in Cage --------------------------------
    def addCage(self):
        con = pymysql.connect(user='root',
password='Password@21',
                              host='localhost',
database='ZOOMANAGEMENT')
        cur = con.cursor()
        query= "insert into Cage(cage_ID,
capacity,) values('{}',
'{}')".format(self.Cage_ID_Var.get(),

self.capacity_var.get())

        print("Query Executed")

        cur.execute(query)
        con.commit()
        con.close()
    #--------------------Function to Update in
Cage ----------------------------------------

    def updateCage(self):
         con = pymysql.connect(user='root',
password='Password@21',
                              host='localhost',
database='ZOOMANAGEMENT')
         cur = con.cursor()
         query = "update Cage set capacity = '{}'
where cage_ID =
'{}'".format(self.capacity_var.get(),
self.Cage_ID_Var.get())
         print("Query Executed")

         cur.execute(query)
         con.commit()
```

```
        con.close()
    #Delete Cage
    def delete_cage(self):
        con = pymysql.connect(user='root',
password='Password@21', host='localhost',
database='ZOOMANAGEMENT')
        cur = con.cursor()
        query = "delete from Cage where cage_ID =
'{}'".format(self.Cage_ID_Var.get())
        con.commit()
        con.close()
    #-----------------------Function To Clear All
Field ----------------------------------
    def clear(self):
       self.Cage_ID_Var.set(" ")
       self.capacity_var.set(" ")


    # --------------------------Function To Exit
The Program -----------------------------
    def exit(self):
        exit()
```

Viewdetails.py

```
from tkinter import *
from tkinter import ttk
from ZooStaff import Staff
import pymysql

'''Author : SAGAR VED BAIRWA
   Machine: Ubuntu 20.04'''

class view:
```

```
    def __init__(self, root):
        self.root = root  # main window for GUI
        self.root.title("ZOO Management System")
        self.root.geometry("1600x1024+0+0")

        title = Label(self.root, text="Zoo
Management System", bd=10, relief=GROOVE,
                      font=("times new roman", 40,
"bold"), bg="yellow", fg="red")
        title.pack(side=TOP, fill=X)
        # ------------------------------VARIABLES
-------------------------
        self.Emp_ID_var = StringVar()
        self.emp_name_var = StringVar()
        self.dob_var = StringVar()
        self.gender_var = StringVar()
        self.work_as_var = StringVar()
        self.joining_date_var = StringVar()
        self.mobile_var = StringVar()

        #
=====================================MANAGE
FRAMES
=================================================
=======
        Manage_Frame = Frame(self.root, bd=4,
relief=RIDGE, bg="#f0e8c9")
        Manage_Frame.place(x=10, y=90, width=1900,
height=60)

        AddUpdateBtn = Button(Manage_Frame,
text="Add/Update", font=("times new roman", 20,
"bold"), width=20).grid(
            row=0, column=0, padx=5, pady=5)
        VieweBtn = Button(Manage_Frame, text="View
", font=("times new roman", 20, "bold"),
```

```
width=20).grid(row=0,

column=1,

padx=5,

pady=5)
        TicketBtn = Button(Manage_Frame,
text="Ticket ", font=("times new roman", 20,
"bold"), width=20).grid(row=0,

column=2,

padx=5,

pady=5)
        ExitBtn = Button(Manage_Frame, text="Exit
",command=exit, font=("times new roman", 20,
"bold"), width=20).grid(row=0,

column=3,

padx=700,

pady=5)

        #
=========================================MENU FRAMES
=================================================
=======
        Menu_Frame = Frame(self.root, bd=4,
relief=RIDGE, bg="#c9e7f0")
        Menu_Frame.place(x=10, y=170, width=380 -
30, height=820)
        Btn_Frame = Frame(Menu_Frame, bd=4,
relief=RIDGE, bg="#d996c1")
```

```
        Btn_Frame.place(x=10, y=10, width=320,
height=400)
        View_StaffBtn = Button(Btn_Frame,
text="View For Staff", font=("times new roman",
20, "bold"), width=20).grid(
            row=0, column=0, padx=10, pady=10)
        View_AnimalBtn = Button(Btn_Frame,
text="View For Animal", font=("times new roman",
20, "bold"),

width=20).grid(row=1, column=0, padx=10, pady=10)
        View_VisitorBtn = Button(Btn_Frame,
text="View For Visitor", font=("times new roman",
20, "bold"),

width=20).grid(row=2, column=0, padx=10, pady=10)

        Btn_Frame2 = Frame(Menu_Frame, bd=4,
relief=RIDGE, bg="#d996c1")
        Btn_Frame2.place(x=10, y=430, width=320,
height=370)


        SearchTypelbl =
Label(Btn_Frame2,text="Search
By",bg="#d996c1",font=("times new roman", 15) )
        SearchTypelbl.grid(row=0,column=0, pady=10,
padx=5, sticky="w")

        SearchType = ttk.Combobox(Btn_Frame2,
font=("times new roman", 14),state='readonly')
        SearchType['values'] = ('Emp ID', 'Name ',
'Mobile No.')
        SearchType.current(0)
        SearchType.grid(row=0,column=1, pady=10,
padx=5, sticky="w")
```

```
        SearchTypetxt = Label(Btn_Frame2,
text="Input ",bg="#d996c1", font=("times new
roman", 15))
        SearchTypetxt.grid(row=1, column=0,
pady=10, padx=5, sticky="w")

        txt_search =
Entry(Btn_Frame2,textvariable=self.Emp_ID_var,
font=("times new roman", 14,
"bold"),bd=5,relief=GROOVE)
        txt_search.grid(row=1, column=1, pady=10,
padx=5)
        Search_Btn = Button(Btn_Frame2,
text="Search",font=("times new roman", 15,
"bold"),
                            width=20).grid(
            row=2, columnspan=2, padx=10, pady=10)
        ViewAll_Btn = Button(Btn_Frame2, text="View
All", font=("times new roman", 15, "bold"),
                              width=20).grid(row=3,
columnspan=2, padx=5, pady=5)
        #DeleteBtn = Button(Btn_Frame2,
text="Delete", font=("times new roman", 20,
"bold"),
                              #width=20).grid(row=2,
column=0, padx=10, pady=10)
        ClearBtn = Button(Btn_Frame2, text="clear
All Fields",font=("times new roman", 15, "bold"),
                             width=20).grid(row=4,
columnspan=2, padx=5, pady=5)

        #
=====================================VIEW FRAMES
=====================================================
=======
```

```
        View_Frame = Frame(self.root, bd=4,
relief=RIDGE, bg="#c9e7f0")
        View_Frame.place(x=370, y=170, width=1530,
height=820)

        scroll_x = Scrollbar(View_Frame,
orient=HORIZONTAL)
        scroll_y = Scrollbar(View_Frame,
orient=VERTICAL)

        self.Staff_table = ttk.Treeview(View_Frame,
columns=(
        "Emp_ID", "emp_name", "dob", "gender",
"work_as", "Joining_date", "Mobile", "Address"),

xscrollcommand=scroll_x.set,
yscrollcommand=scroll_y.set)
        scroll_x.pack(side=BOTTOM, fill=X)
        scroll_y.pack(side=RIGHT, fill=Y)

scroll_x.config(command=self.Staff_table.xview)

scroll_y.config(command=self.Staff_table.yview)

        self.Staff_table.heading("Emp_ID",
text="Employee ID")
        self.Staff_table.heading("emp_name",
text="Employee Name")
        self.Staff_table.heading("dob", text="Date
Of Birth")
        self.Staff_table.heading("gender",
text="Gender")
        self.Staff_table.heading("work_as",
text="work_as")
        self.Staff_table.heading("Joining_date",
text="Joining_date")
```

```
        self.Staff_table.heading("Mobile",
text="Mobile")
        self.Staff_table.heading("Address",
text="Address")
        self.Staff_table['show'] = 'headings'

        self.Staff_table.column("Emp_ID",
width=180)
        self.Staff_table.column("emp_name",
width=240)
        self.Staff_table.column("dob", width=180)
        self.Staff_table.column("gender",
width=120)
        self.Staff_table.column("work_as",
width=160)
        self.Staff_table.column("Joining_date",
width=130)
        self.Staff_table.column("Mobile",
width=140)
        self.Staff_table.column("Address",
width=360)
        self.Staff_table.bind("<ButtonRelease-1>",
self.get_cursor)
        self.Staff_table.pack(fill=BOTH, expand=1)
        self.fetchdata()



    # Functions
    def fetchdata(self):
        con = pymysql.connect(user='root',
password='Password@21',
                              host='localhost',
database='ZOOMANAGEMENT')
        cur = con.cursor()
        query = "select * from staff"
```

```
        cur.execute(query)
        rows = cur.fetchall()
        if len(rows)!=0:

self.Staff_table.delete(*self.Staff_table.get_chil
dren())
            for row in rows:
                self.Staff_table.insert('', END,
values=row)
            con.commit()
        con.close()

    def get_cursor(self, ev):
        cursor_row = self.Staff_table.focus()
        contents =
self.Staff_table.item(cursor_row)
        row = contents['values']
        print(row)

    def exit(self):
        exit()
```

Ticket.py

Thank you