DBMS - Mini Project

Zoo Management System

Submitted By:

Name: Arun Kumar Rath
SRN: PES1UG20CS076
V Semester Section : B

<u>Short Description and Scope of the Project</u>

My project name is "Zoo Management System".
My goal is to show a very initial overview of a zoo and how it is managed centrally by using database management system.

My idea can be divided into four parts which represents my whole database shortly. These parts hold the data of every possible field in the zoo management system.

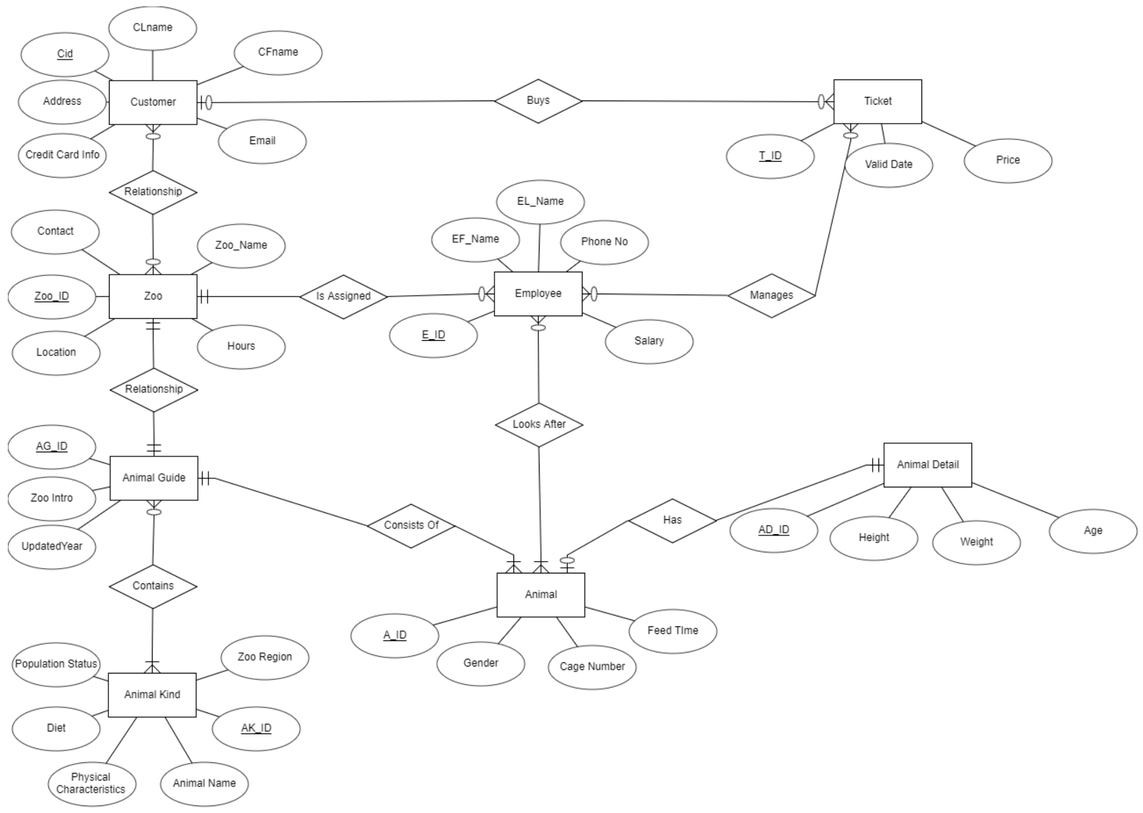
Firstly
<u>Animal Details</u>, we can have the proper datails about animal_id, animal_name, cage_num, gender, height, weight, age, diet, status.

<u>Staff field</u> is used to store all the valid datas of the staff i.e emp_id, emp_name, emp_des, phone_num, salary.

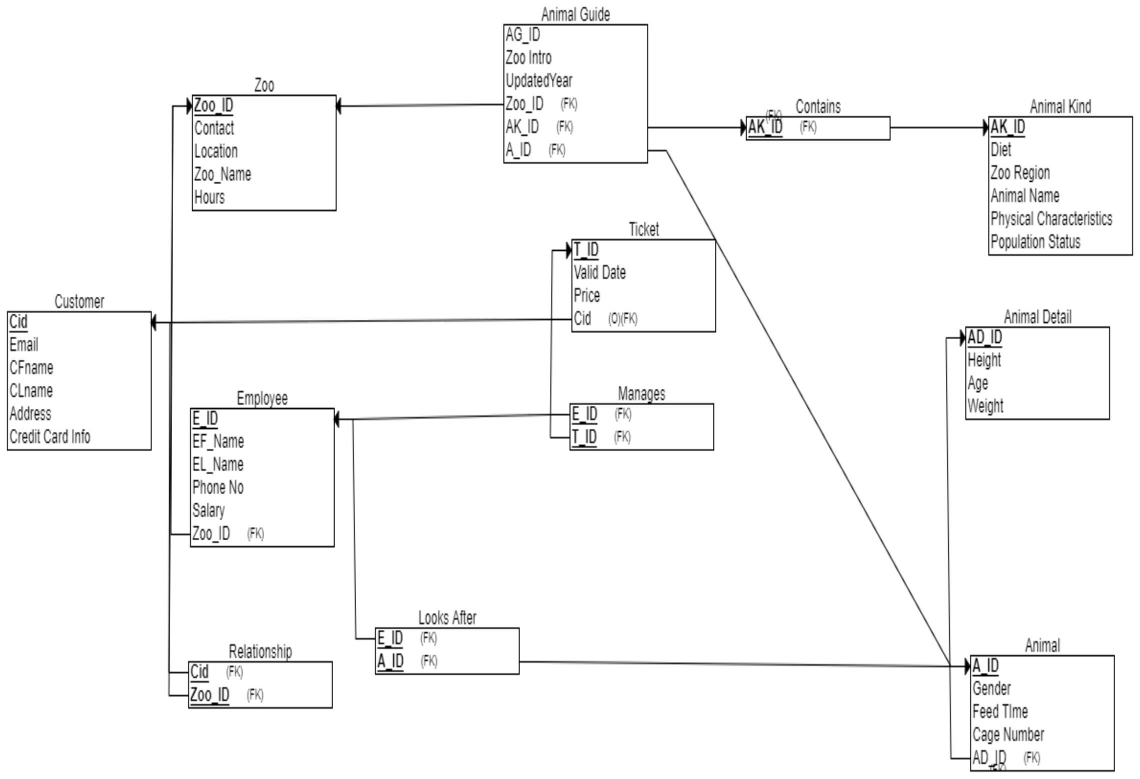
<u>Ticket Field</u> is used to store the ticket details such as ticket_id, order_date, price, age.

A zoo is incomplete without **<u>Customers</u>** . From **<u>Customers</u>** zoo gets it's partial profits.
Visitors access all the facilities of the zoo through the column coupon and ticket.
We obtain he customer details
.

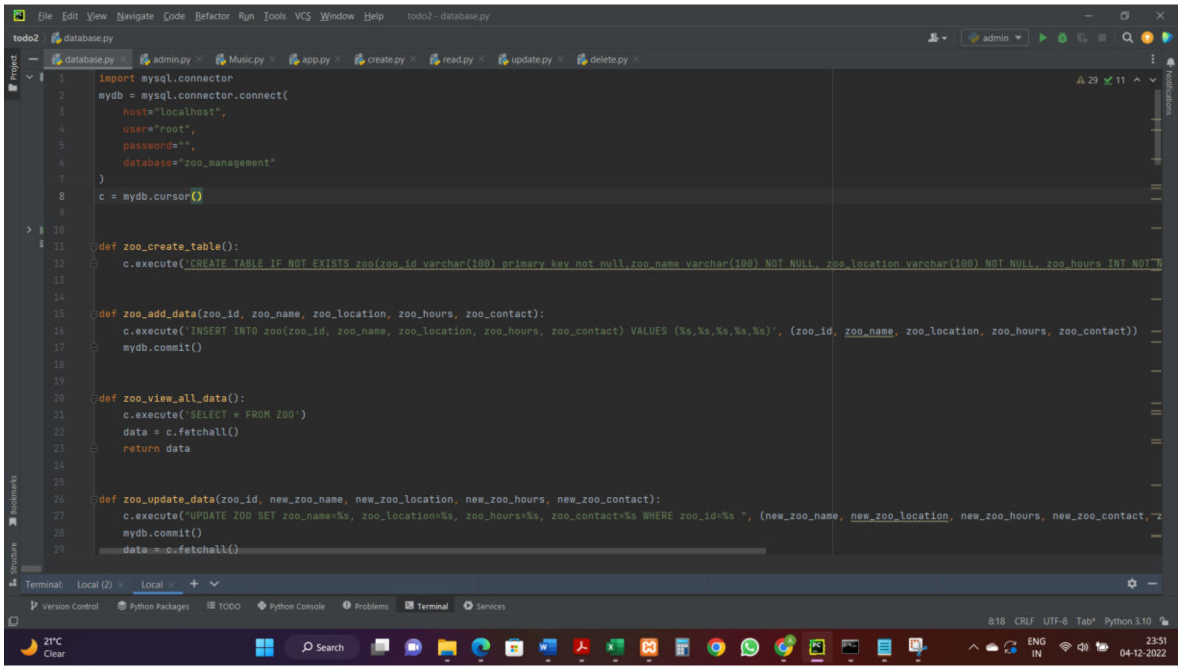
ER Diagram


CLname
Cid
CFname
Address
Customer
Email
Credit Card Info
Buys
Ticket
T_ID
Valid Date
Price
Relationship
EL_Name
EF_Name
Phone No
Contact
Zoo_Name
Zoo_ID
Zoo
Is Assigned
Employee
Manages
Location
Hours
E_ID
Salary
Relationship
Looks After
AG_ID
Animal Guide
Animal Detail
Zoo Intro
Consists Of
Has
AD_ID
Height
Weight
Age
UpdatedYear
Contains
Animal
A_ID
Feed TIme
Gender
Cage Number
Population Status
Zoo Region
Animal Kind
Diet
AK_ID
Physical Characteristics
Animal Name

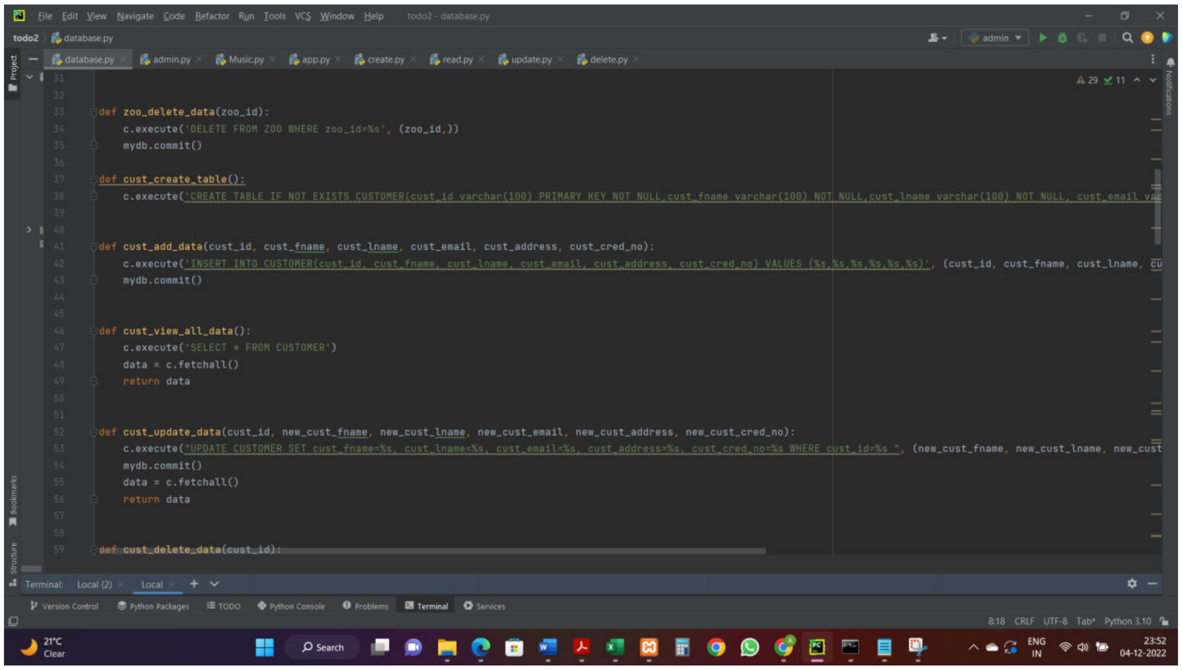
Relational Schema



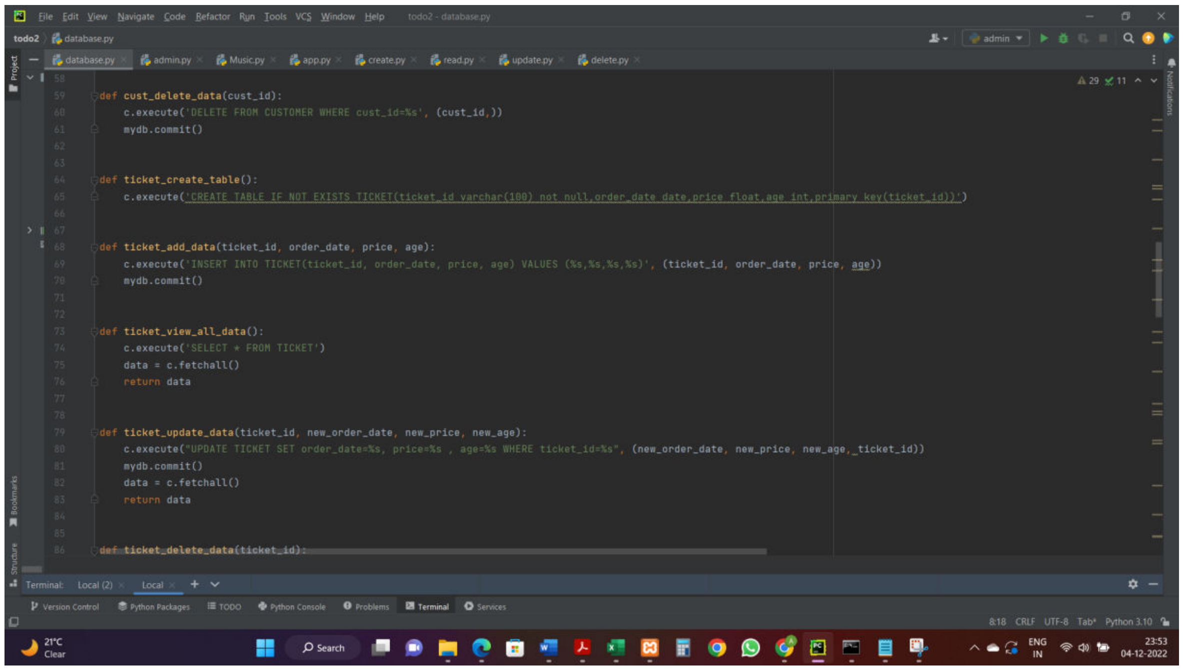
DDL statements - Building the database and populating the database



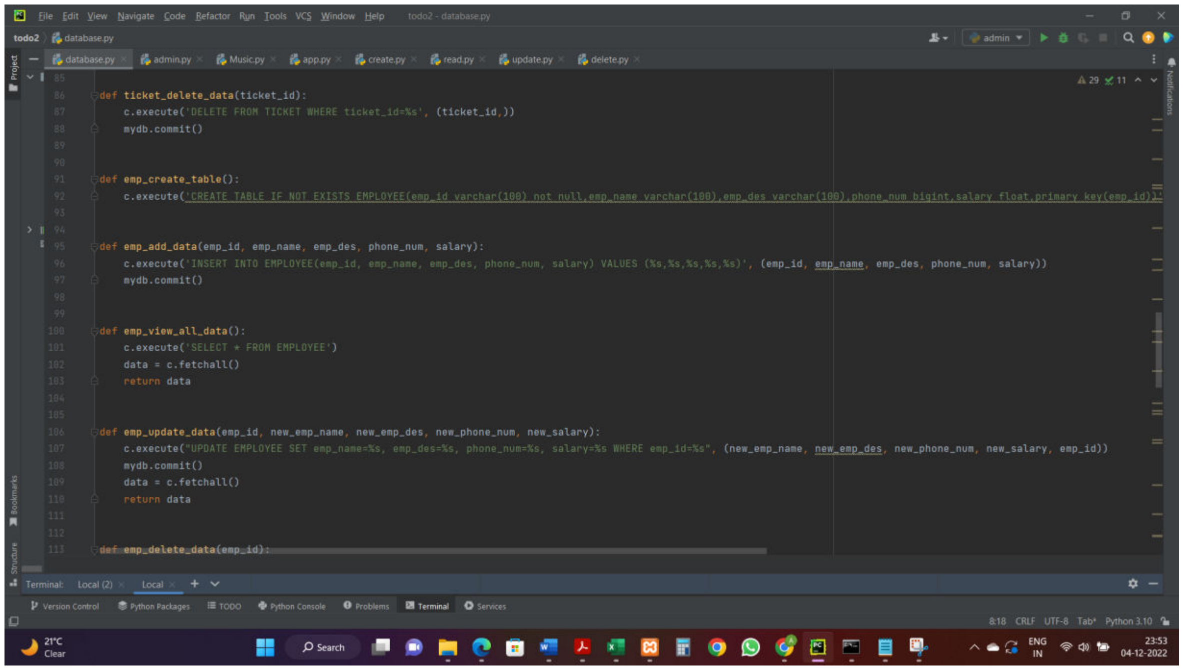
```python
import mysql.connector
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="zoo_management"
)
c = mydb.cursor()


def zoo_create_table():
    c.execute('CREATE TABLE IF NOT EXISTS zoo(zoo_id varchar(100) primary key not null,zoo_name varchar(100) NOT NULL, zoo_location varchar(100) NOT NULL, zoo_hours INT NOT


def zoo_add_data(zoo_id, zoo_name, zoo_location, zoo_hours, zoo_contact):
    c.execute('INSERT INTO zoo(zoo_id, zoo_name, zoo_location, zoo_hours, zoo_contact) VALUES (%s,%s,%s,%s,%s)', (zoo_id, zoo_name, zoo_location, zoo_hours, zoo_contact))
    mydb.commit()


def zoo_view_all_data():
    c.execute('SELECT * FROM ZOO')
    data = c.fetchall()
    return data


def zoo_update_data(zoo_id, new_zoo_name, new_zoo_location, new_zoo_hours, new_zoo_contact):
    c.execute("UPDATE ZOO SET zoo_name=%s, zoo_location=%s, zoo_hours=%s, zoo_contact=%s WHERE zoo_id=%s ", (new_zoo_name, new_zoo_location, new_zoo_hours, new_zoo_contact,
    mydb.commit()
    data = c.fetchall()
```



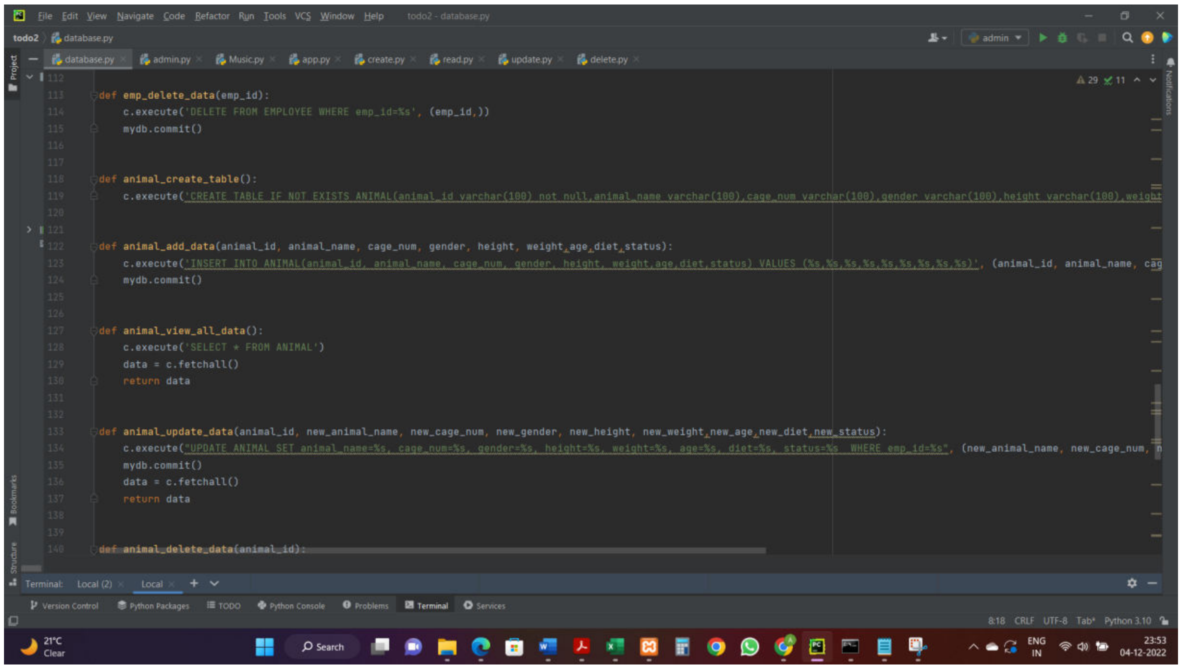
```python


def zoo_delete_data(zoo_id):
    c.execute('DELETE FROM ZOO WHERE zoo_id=%s', (zoo_id,))
    mydb.commit()

def cust_create_table():
    c.execute('CREATE TABLE IF NOT EXISTS CUSTOMER(cust_id varchar(100) PRIMARY KEY NOT NULL,cust_fname varchar(100) NOT NULL,cust_lname varchar(100) NOT NULL, cust_email


def cust_add_data(cust_id, cust_fname, cust_lname, cust_email, cust_address, cust_cred_no):
    c.execute('INSERT INTO CUSTOMER(cust_id, cust_fname, cust_lname, cust_email, cust_address, cust_cred_no) VALUES (%s,%s,%s,%s,%s,%s)', (cust_id, cust_fname, cust_lname,
    mydb.commit()


def cust_view_all_data():
    c.execute('SELECT * FROM CUSTOMER')
    data = c.fetchall()
    return data


def cust_update_data(cust_id, new_cust_fname, new_cust_lname, new_cust_email, new_cust_address, new_cust_cred_no):
    c.execute("UPDATE CUSTOMER SET cust_fname=%s, cust_lname=%s, cust_email=%s, cust_address=%s, cust_cred_no=%s WHERE cust_id=%s ", (new_cust_fname, new_cust_lname, new_cust
    mydb.commit()
    data = c.fetchall()
    return data


def cust_delete_data(cust_id):
```



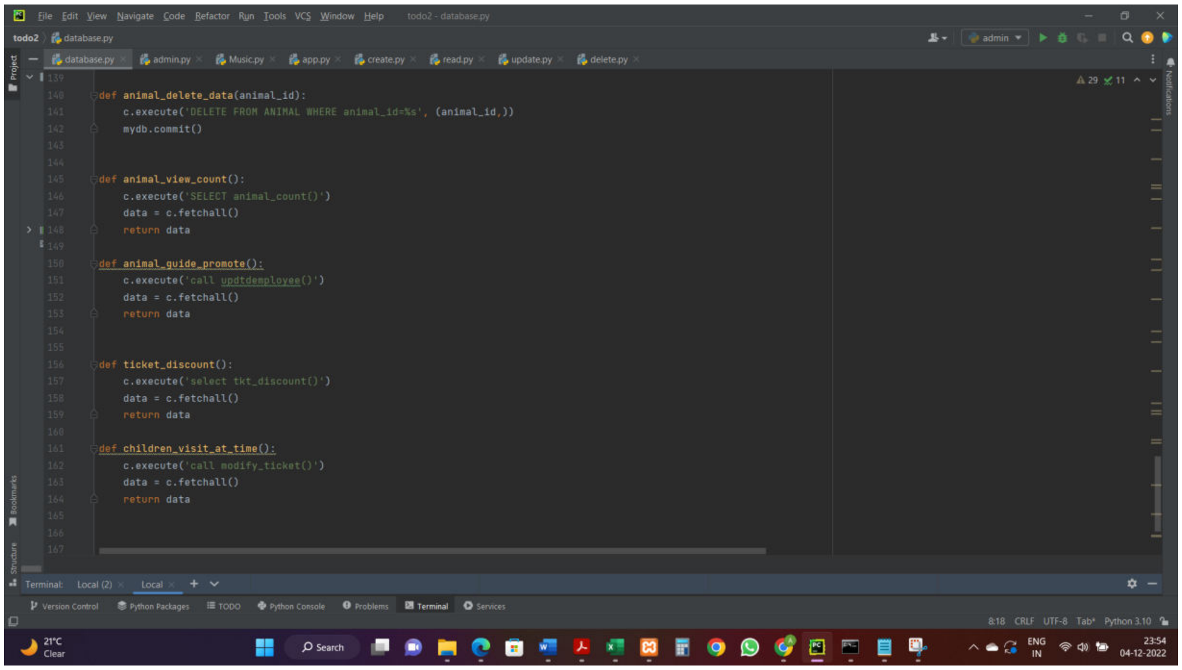
```
def cust_delete_data(cust_id):
    c.execute('DELETE FROM CUSTOMER WHERE cust_id=%s', (cust_id,))
    mydb.commit()


def ticket_create_table():
    c.execute('CREATE TABLE IF NOT EXISTS TICKET(ticket_id varchar(100) not null,order_date date,price float,age int,primary key(ticket_id))')


def ticket_add_data(ticket_id, order_date, price, age):
    c.execute('INSERT INTO TICKET(ticket_id, order_date, price, age) VALUES (%s,%s,%s,%s)', (ticket_id, order_date, price, age))
    mydb.commit()


def ticket_view_all_data():
    c.execute('SELECT * FROM TICKET')
    data = c.fetchall()
    return data


def ticket_update_data(ticket_id, new_order_date, new_price, new_age):
    c.execute("UPDATE TICKET SET order_date=%s, price=%s , age=%s WHERE ticket_id=%s", (new_order_date, new_price, new_age, ticket_id))
    mydb.commit()
    data = c.fetchall()
    return data


def ticket_delete_data(ticket_id):
```



```
def ticket_delete_data(ticket_id):
    c.execute('DELETE FROM TICKET WHERE ticket_id=%s', (ticket_id,))
    mydb.commit()


def emp_create_table():
    c.execute('CREATE TABLE IF NOT EXISTS EMPLOYEE(emp_id varchar(100) not null,emp_name varchar(100),emp_des varchar(100),phone_num bigint,salary float,primary key(emp_id))'


def emp_add_data(emp_id, emp_name, emp_des, phone_num, salary):
    c.execute('INSERT INTO EMPLOYEE(emp_id, emp_name, emp_des, phone_num, salary) VALUES (%s,%s,%s,%s,%s)', (emp_id, emp_name, emp_des, phone_num, salary))
    mydb.commit()


def emp_view_all_data():
    c.execute('SELECT * FROM EMPLOYEE')
    data = c.fetchall()
    return data


def emp_update_data(emp_id, new_emp_name, new_emp_des, new_phone_num, new_salary):
    c.execute("UPDATE EMPLOYEE SET emp_name=%s, emp_des=%s, phone_num=%s, salary=%s WHERE emp_id=%s", (new_emp_name, new_emp_des, new_phone_num, new_salary, emp_id))
    mydb.commit()
    data = c.fetchall()
    return data


def emp_delete_data(emp_id):
```



```
def emp_delete_data(emp_id):
    c.execute('DELETE FROM EMPLOYEE WHERE emp_id=%s', (emp_id,))
    mydb.commit()


def animal_create_table():
    c.execute('CREATE TABLE IF NOT EXISTS ANIMAL(animal_id varchar(100) not null,animal_name varchar(100),cage_num varchar(100),gender varchar(100),height varchar(100),weight


def animal_add_data(animal_id, animal_name, cage_num, gender, height, weight,age,diet,status):
    c.execute('INSERT INTO ANIMAL(animal_id, animal_name, cage_num, gender, height, weight,age,diet,status) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s)', (animal_id, animal_name, cag
    mydb.commit()


def animal_view_all_data():
    c.execute('SELECT * FROM ANIMAL')
    data = c.fetchall()
    return data


def animal_update_data(animal_id, new_animal_name, new_cage_num, new_gender, new_height, new_weight,new_age,new_diet,new_status):
    c.execute("UPDATE ANIMAL SET animal_name=%s, cage_num=%s, gender=%s, height=%s, weight=%s, age=%s, diet=%s, status=%s  WHERE emp_id=%s", (new_animal_name, new_cage_num, n
    mydb.commit()
    data = c.fetchall()
    return data


def animal_delete_data(animal_id):
```



```
def animal_delete_data(animal_id):
    c.execute('DELETE FROM ANIMAL WHERE animal_id=%s', (animal_id,))
    mydb.commit()


def animal_view_count():
    c.execute('SELECT animal_count()')
    data = c.fetchall()
    return data

def animal_guide_promote():
    c.execute('call updtdemployee()')
    data = c.fetchall()
    return data


def ticket_discount():
    c.execute('select tkt_discount()')
    data = c.fetchall()
    return data

def children_visit_at_time():
    c.execute('call modify_ticket()')
    data = c.fetchall()
    return data
```

Join Queries

Showcase at least 4 join queries
Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

Question 1: Which animals are terristerial and herbivore

SELECT l.animal_id,life_duration,class,Age FROM animal_details d JOIN animal_life l ON (d.a_name = l.a_name) and d.food_type ="Herbivore" and l.Animal_type = "Terrestrial";

Question 2: Which vet gives a treatment to sick animal and show his details

SELECT r.cage_no, r.record_details , v.vet_name, v.speciality,v.veternity_experience,s.job_type, s.shifting_time . s.date_of_joining,s.job_duration FROM record r JOIN staff_details s ON r.staff_id = s.staff_id JOIN vet_details v ON s.staff_id = v.staff_id;

Question 3: Show the total cost of cage where reptile animals lived.

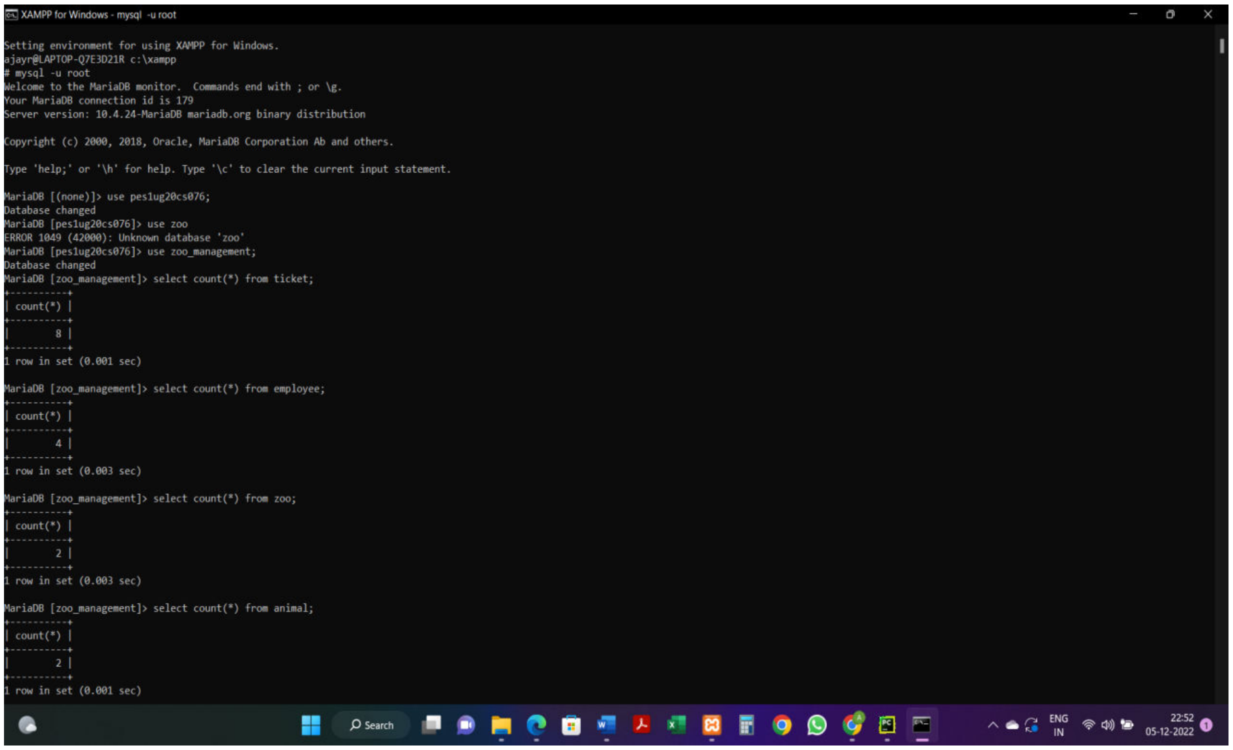
SELECT a.cage_no, sum(SERVICE_CHARGE)+(MEDICINE_COST)+(CLEANING_COST)+FOOD_COST "Total cost" FROM animal_details a JOIN expense e ON (a.cage_no = e.cage_no) and a.class = "reptile" ;

Question 4: HAVING ALL INFO OF THE CAGES BY JOINING TWO TABLES SELECT

cage_no,animal_details.a_name,animal_name,gender,food_type,life_duration, age,Birth_year FROM animal_details RIGHT outer JOIN animal_life ON animal_details.a_name = animal_life.a_name;

Aggregate Functions

Showcase at least 4 Aggregate function queries
Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

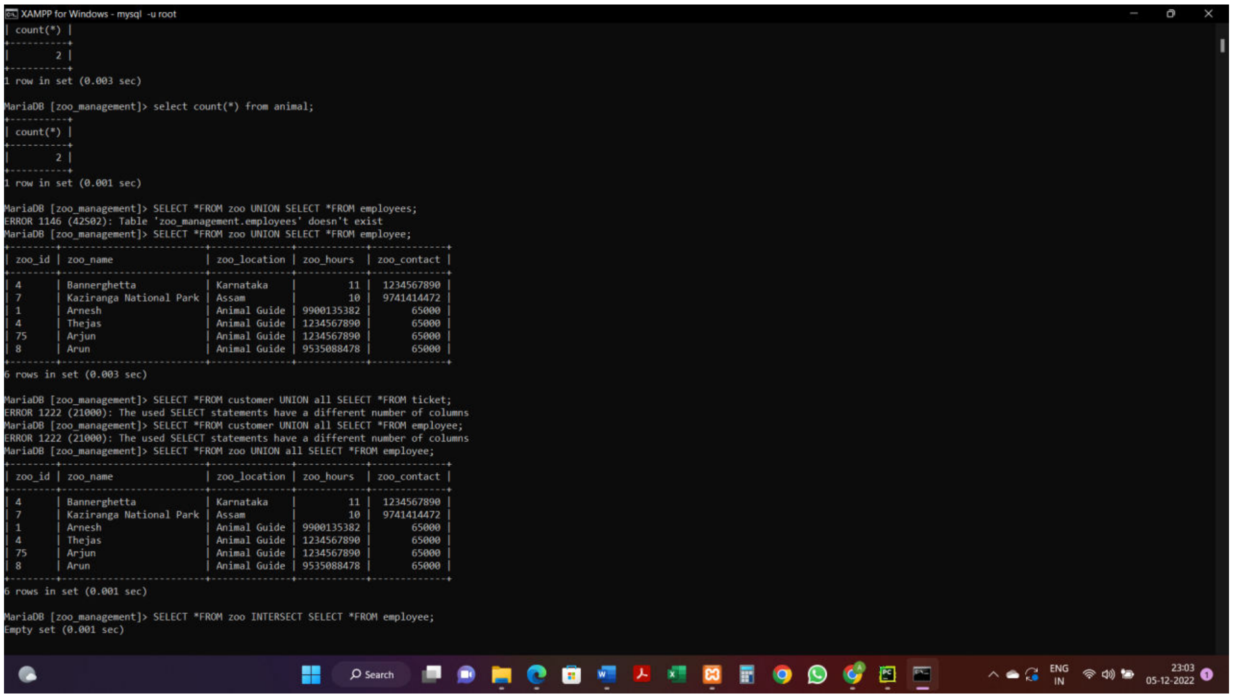


```
Setting environment for using XAMPP for Windows.
ajayr@LAPTOP-Q7E3D21R c:\xampp
# mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 179
Server version: 10.4.24-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use pes1ug20cs076;
Database changed
MariaDB [pes1ug20cs076]> use zoo
ERROR 1049 (42000): Unknown database 'zoo'
MariaDB [pes1ug20cs076]> use zoo_management;
Database changed
MariaDB [zoo_management]> select count(*) from ticket;
+----------+
| count(*) |
+----------+
|        8 |
+----------+
1 row in set (0.001 sec)

MariaDB [zoo_management]> select count(*) from employee;
+----------+
| count(*) |
+----------+
|        4 |
+----------+
1 row in set (0.003 sec)

MariaDB [zoo_management]> select count(*) from zoo;
+----------+
| count(*) |
+----------+
|        2 |
+----------+
1 row in set (0.003 sec)

MariaDB [zoo_management]> select count(*) from animal;
+----------+
| count(*) |
+----------+
|        2 |
+----------+
1 row in set (0.001 sec)
```

Set Operations

Showcase at least 4 Set Operations queries

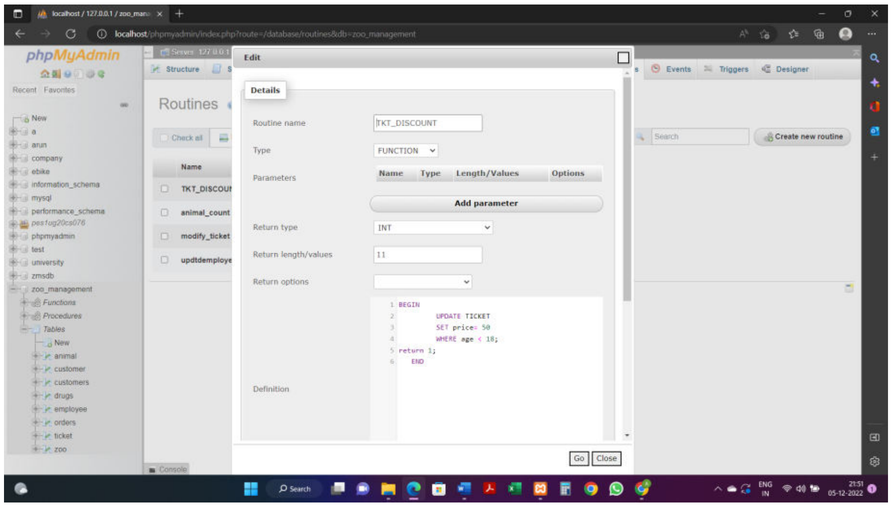
Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results



```
| count(*) |
+----------+
|        2 |
+----------+
1 row in set (0.003 sec)

MariaDB [zoo_management]> select count(*) from animal;
+----------+
| count(*) |
+----------+
|        2 |
+----------+
1 row in set (0.001 sec)

MariaDB [zoo_management]> SELECT *FROM zoo UNION SELECT *FROM employees;
ERROR 1146 (42S02): Table 'zoo_management.employees' doesn't exist
MariaDB [zoo_management]> SELECT *FROM zoo UNION SELECT *FROM employee;
+--------+-------------------------+--------------+------------+-------------+
| zoo_id | zoo_name                | zoo_location | zoo_hours  | zoo_contact |
+--------+-------------------------+--------------+------------+-------------+
| 4      | Bannerghetta            | Karnataka    |         11 |  1234567890 |
| 7      | Kaziranga National Park | Assam        |         10 |  9741414472 |
| 1      | Arnesh                  | Animal Guide | 9900135382 |       65000 |
| 4      | Thejas                  | Animal Guide | 1234567890 |       65000 |
| 75     | Arjun                   | Animal Guide | 1234567890 |       65000 |
| 8      | Arun                    | Animal Guide | 9535088478 |       65000 |
+--------+-------------------------+--------------+------------+-------------+
6 rows in set (0.003 sec)

MariaDB [zoo_management]> SELECT *FROM customer UNION all SELECT *FROM ticket;
ERROR 1222 (21000): The used SELECT statements have a different number of columns
MariaDB [zoo_management]> SELECT *FROM customer UNION all SELECT *FROM employee;
ERROR 1222 (21000): The used SELECT statements have a different number of columns
MariaDB [zoo_management]> SELECT *FROM zoo UNION all SELECT *FROM employee;
+--------+-------------------------+--------------+------------+-------------+
| zoo_id | zoo_name                | zoo_location | zoo_hours  | zoo_contact |
+--------+-------------------------+--------------+------------+-------------+
| 4      | Bannerghetta            | Karnataka    |         11 |  1234567890 |
| 7      | Kaziranga National Park | Assam        |         10 |  9741414472 |
| 1      | Arnesh                  | Animal Guide | 9900135382 |       65000 |
| 4      | Thejas                  | Animal Guide | 1234567890 |       65000 |
| 75     | Arjun                   | Animal Guide | 1234567890 |       65000 |
| 8      | Arun                    | Animal Guide | 9535088478 |       65000 |
+--------+-------------------------+--------------+------------+-------------+
6 rows in set (0.001 sec)

MariaDB [zoo_management]> SELECT *FROM zoo INTERSECT SELECT *FROM employee;
Empty set (0.001 sec)
```

Functions

Create a Function and Procedure. State the objective of the function / Procedure. Run and display the results.



Edit

Details

Routine name: TKT_DISCOUNT

Type: FUNCTION

Parameters: Name | Type | Length/Values | Options

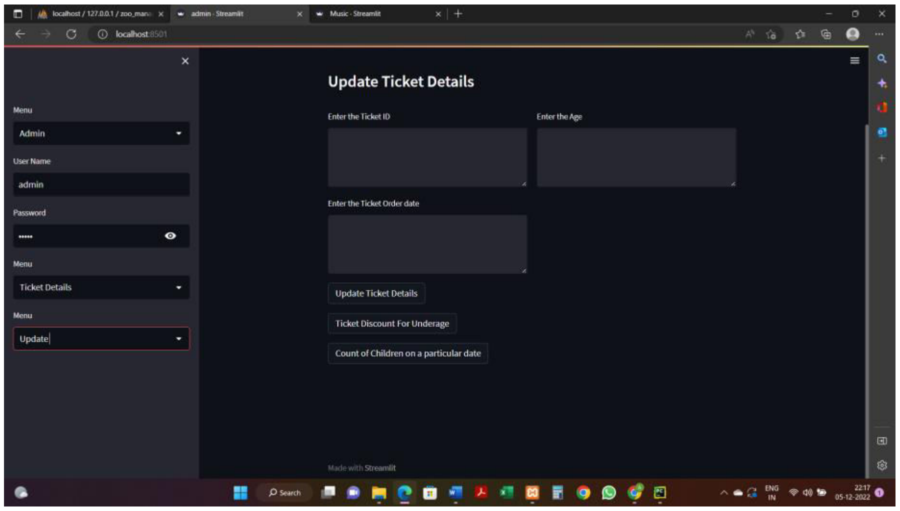
Return type: INT

Return length/values: 11

Return options:

Definition:

```
BEGIN
        UPDATE TICKET
        SET price= 50
        WHERE age < 18;
return 1;
   END
```



Update Ticket Details

Enter the Ticket ID:

Enter the Age:

Enter the Ticket Order date:

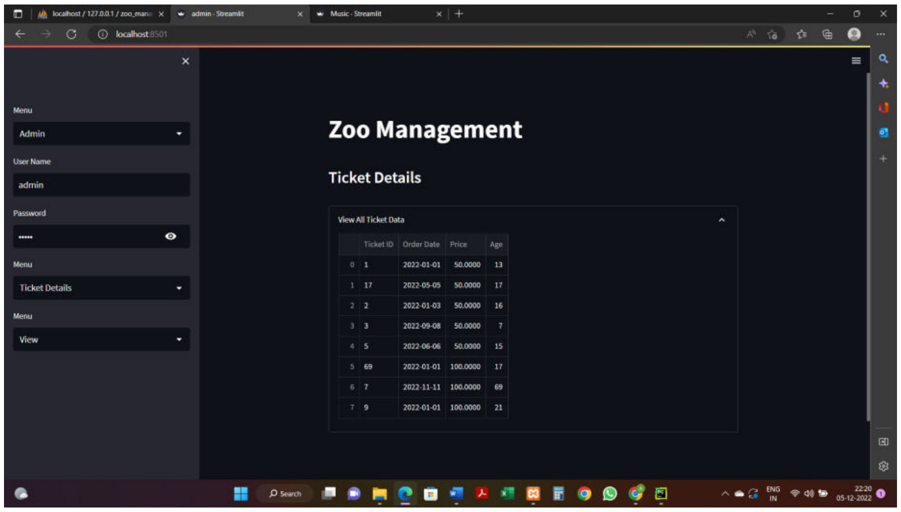
Menu: Admin

User Name: admin

Menu: Ticket Details

Menu: Update

Update Ticket Details
Ticket Discount For Underage
Count of Children on a particular date
Made with Streamlit



Zoo Management

Ticket Details

View All Ticket Data

	Ticket ID	Order Date	Price	Age
0	1	2022-01-01	50.0000	13
1	17	2022-05-05	50.0000	17
2	2	2022-01-03	50.0000	16
3	3	2022-09-08	50.0000	7
4	5	2022-06-06	50.0000	15
5	69	2022-01-01	100.0000	17
6	7	2022-11-11	100.0000	69
7	9	2022-01-01	100.0000	21

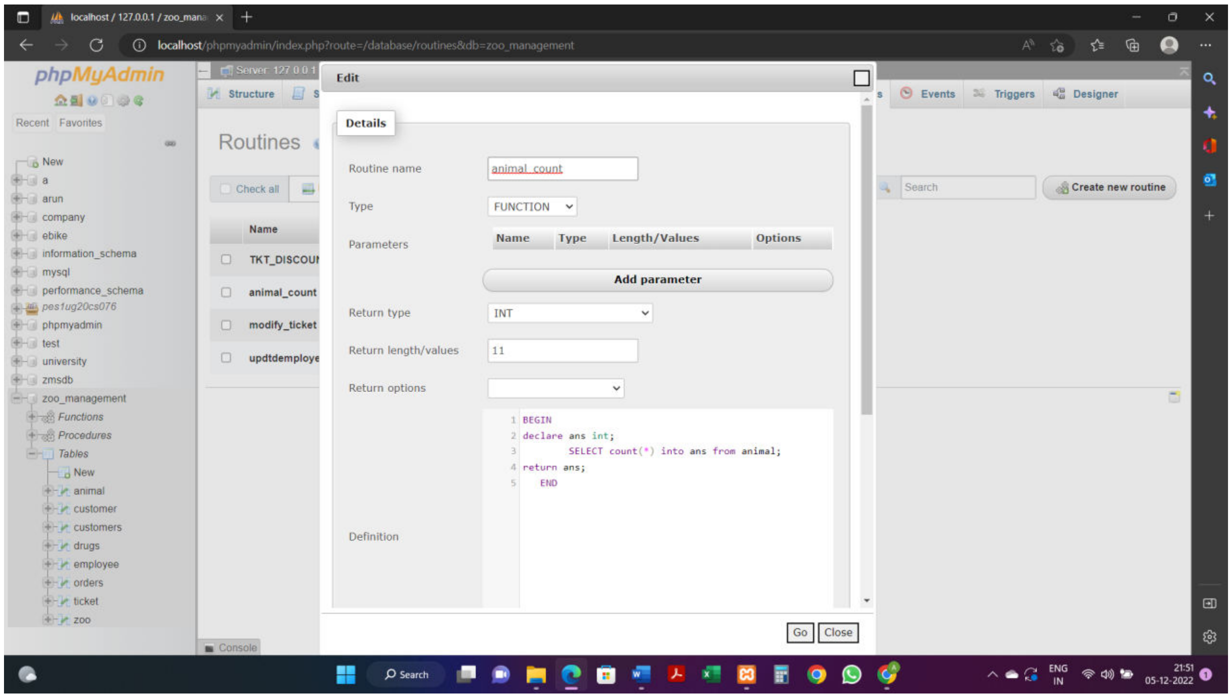
Menu: Admin

User Name: admin

Menu: Ticket Details

Menu: View

TKT_DISCOUNT offers discount of 50 rupees for children visitors.


Routine name: animal_count

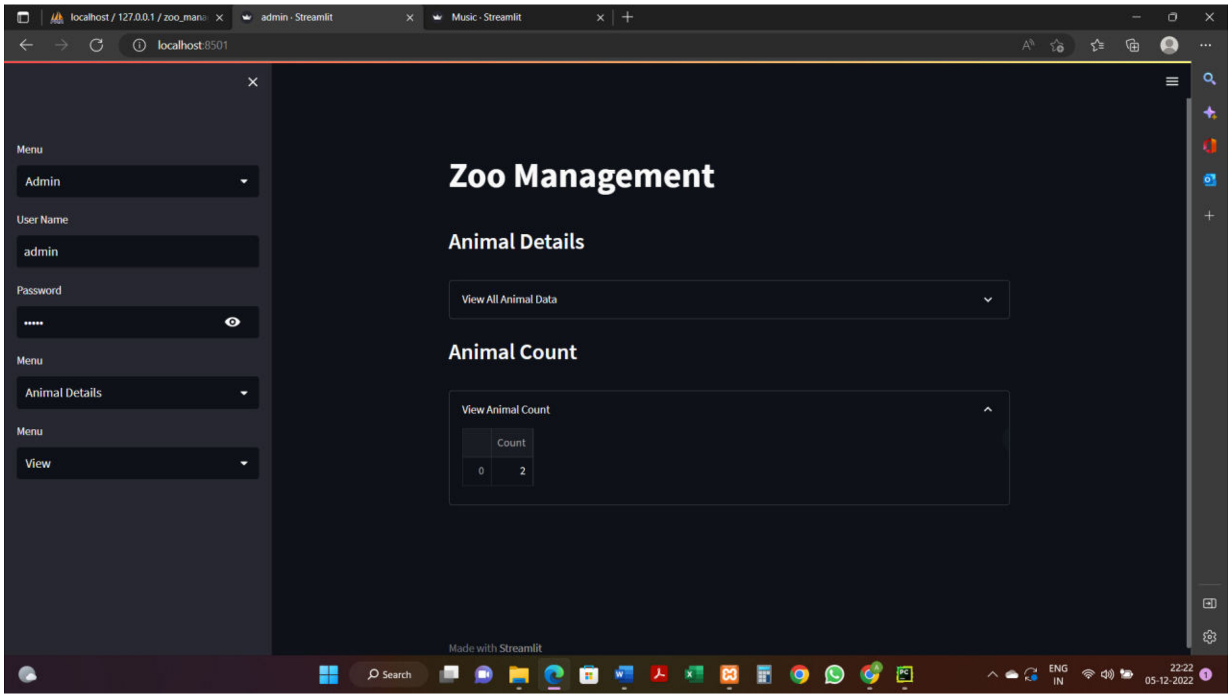
Type: FUNCTION

Return type: INT

Return length/values: 11

Definition:

```
BEGIN
declare ans int;
      SELECT count(*) into ans from animal;
return ans;
   END
```


Menu: Admin

User Name: admin

Password: •••••

Menu: Animal Details

Menu: View

Zoo Management

Animal Details

View All Animal Data

Animal Count

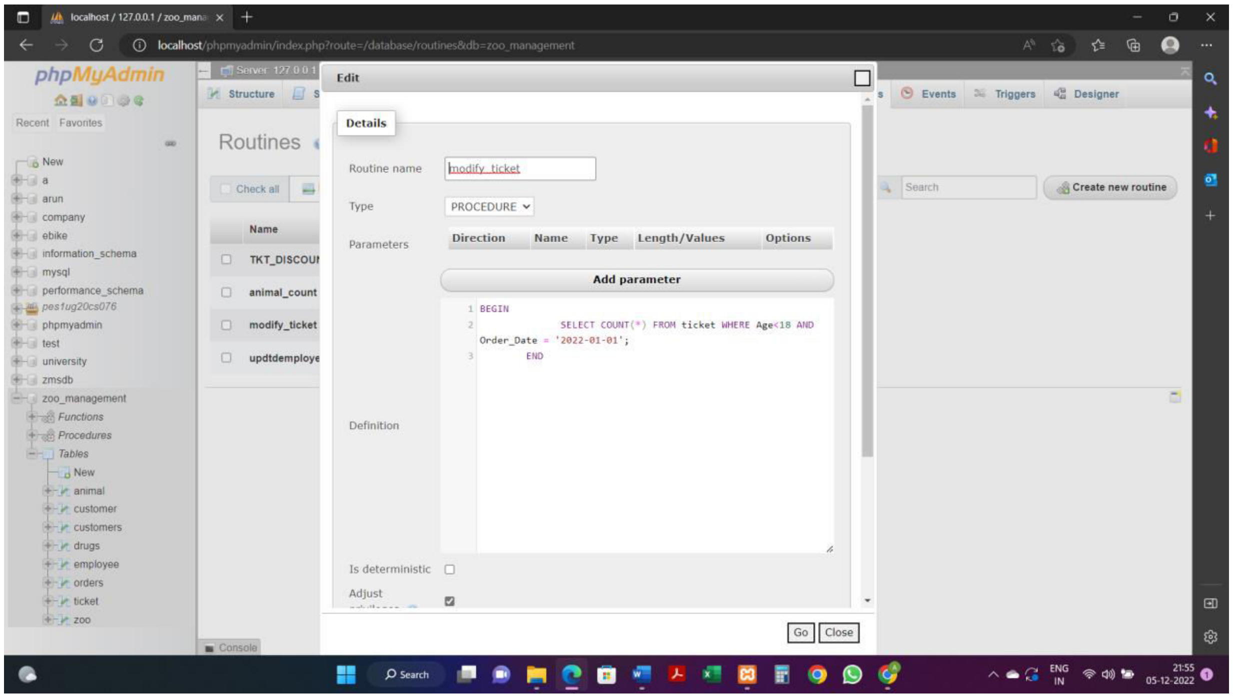
View Animal Count

	Count
0	2

Made with Streamlit

Animal_count shows the count of animals in the animal table

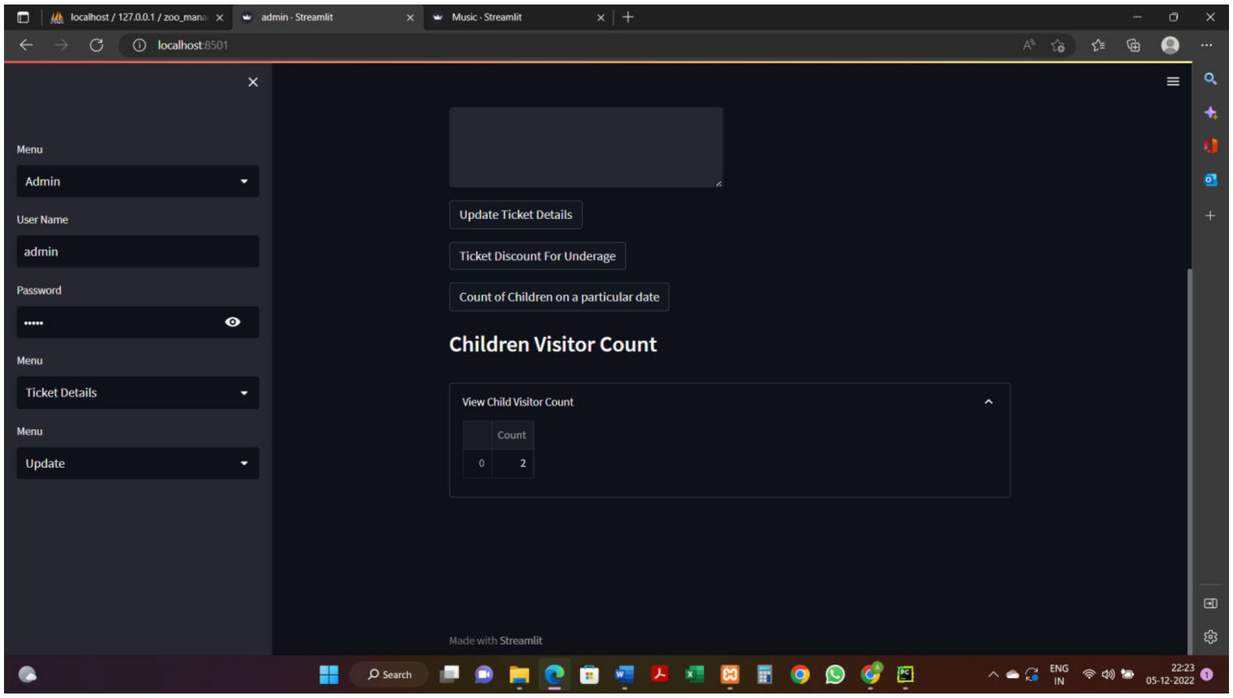
Procedures



Routine name: modify_ticket

Type: PROCEDURE

```
BEGIN
        SELECT COUNT(*) FROM ticket WHERE Age<18 AND
Order_Date = '2022-01-01';
        END
```



Menu: Admin

User Name: admin

Menu: Ticket Details

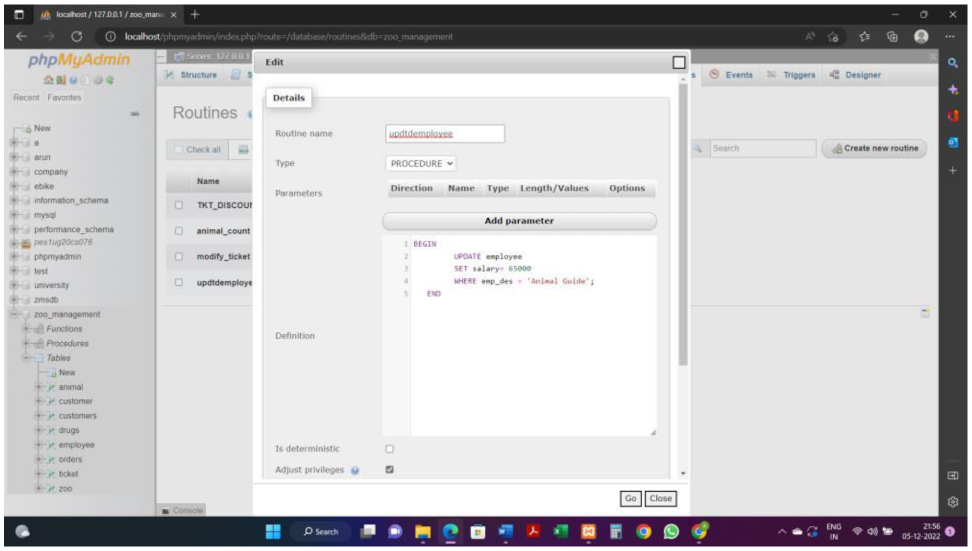
Menu: Update

Children Visitor Count

View Child Visitor Count

	Count
0	2

Shows count of child visitors at a particular date.

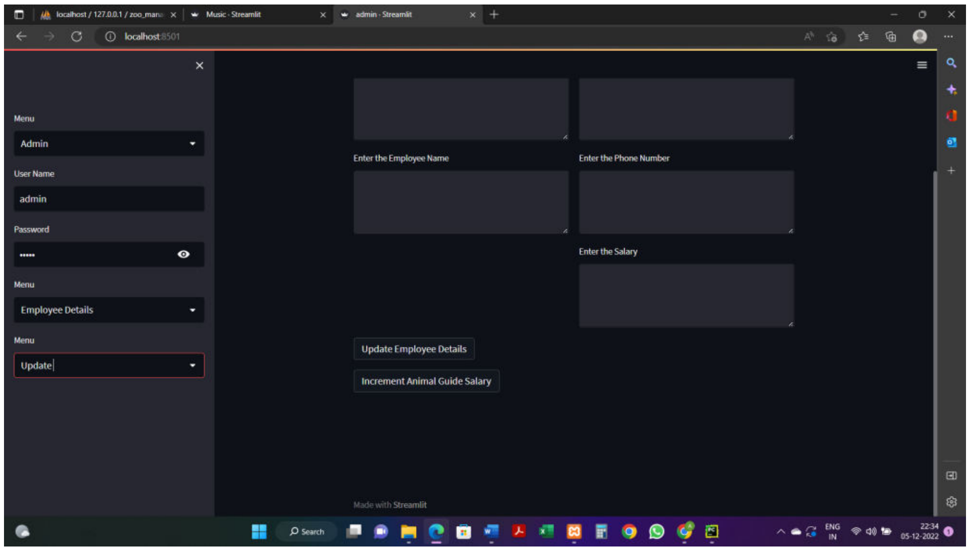


Routine name: updtdemployee

Type: PROCEDURE

Definition:

```
BEGIN
        UPDATE employee
        SET salary= 65000
        WHERE emp_des = 'Animal Guide';
   END
```



Menu: Admin

User Name: admin

Password: •••••

Menu: Employee Details

Menu: Update

Enter the Employee Name

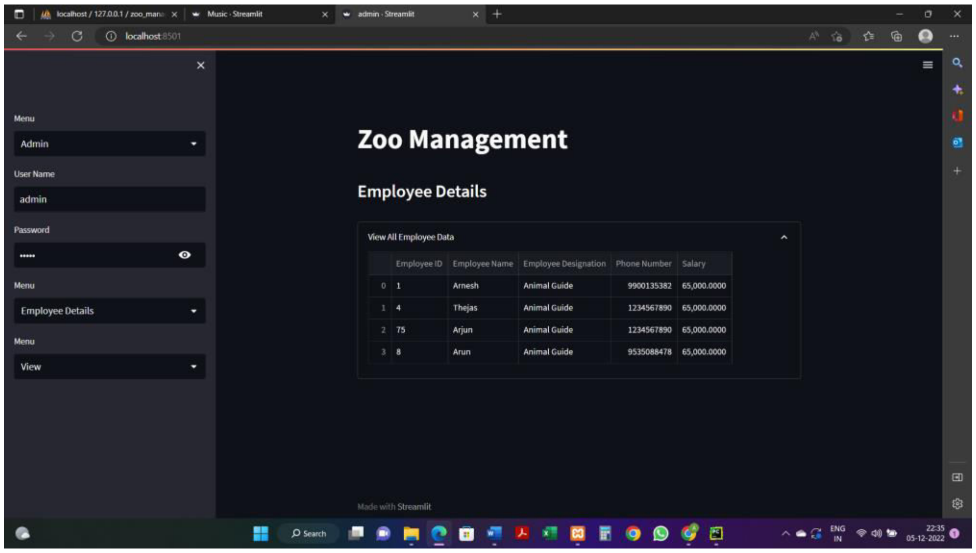
Enter the Phone Number

Enter the Salary

Update Employee Details

Increment Animal Guide Salary

Made with Streamlit



Menu: Admin

User Name: admin

Password: •••••

Menu: Employee Details

Menu: View

Zoo Management

Employee Details

View All Employee Data

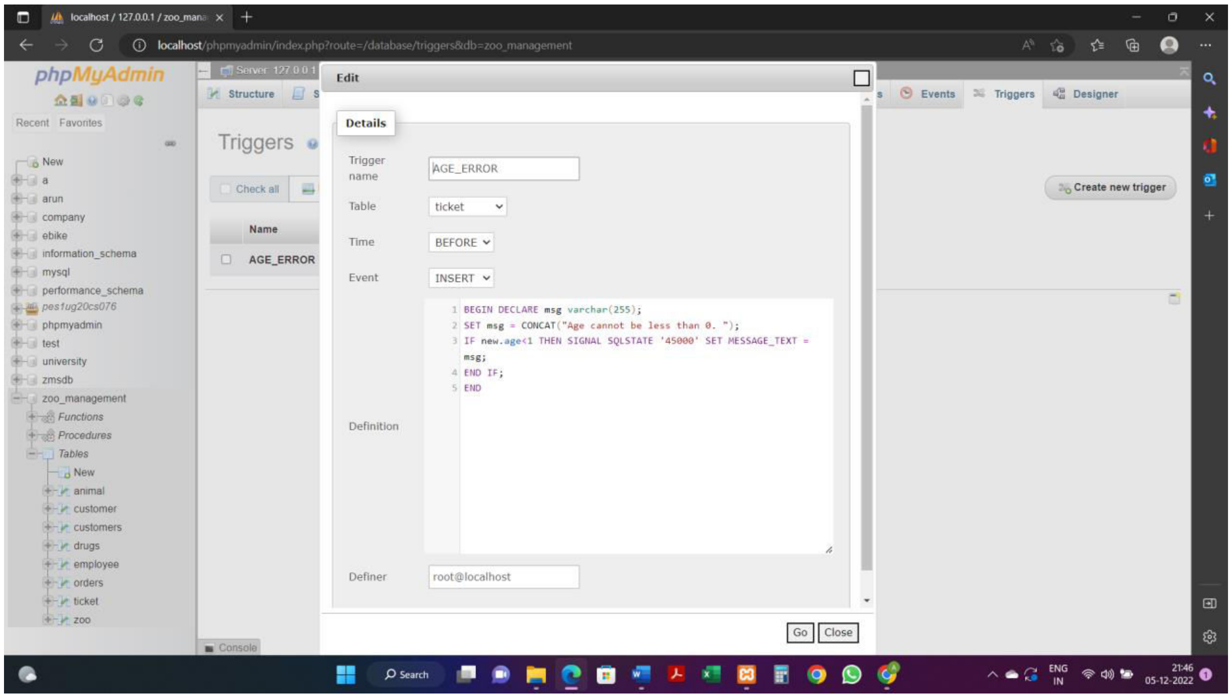
	Employee ID	Employee Name	Employee Designation	Phone Number	Salary
0	1	Arnesh	Animal Guide	9900135382	65,000.0000
1	4	Thejas	Animal Guide	1234567890	65,000.0000
2	75	Arjun	Animal Guide	1234567890	65,000.0000
3	8	Arun	Animal Guide	9535088478	65,000.0000

Made with Streamlit

Updates Salary of animal guides on the click of a button

Triggers

Create a Trigger and a Cursor. State the objective. Run and display the results.

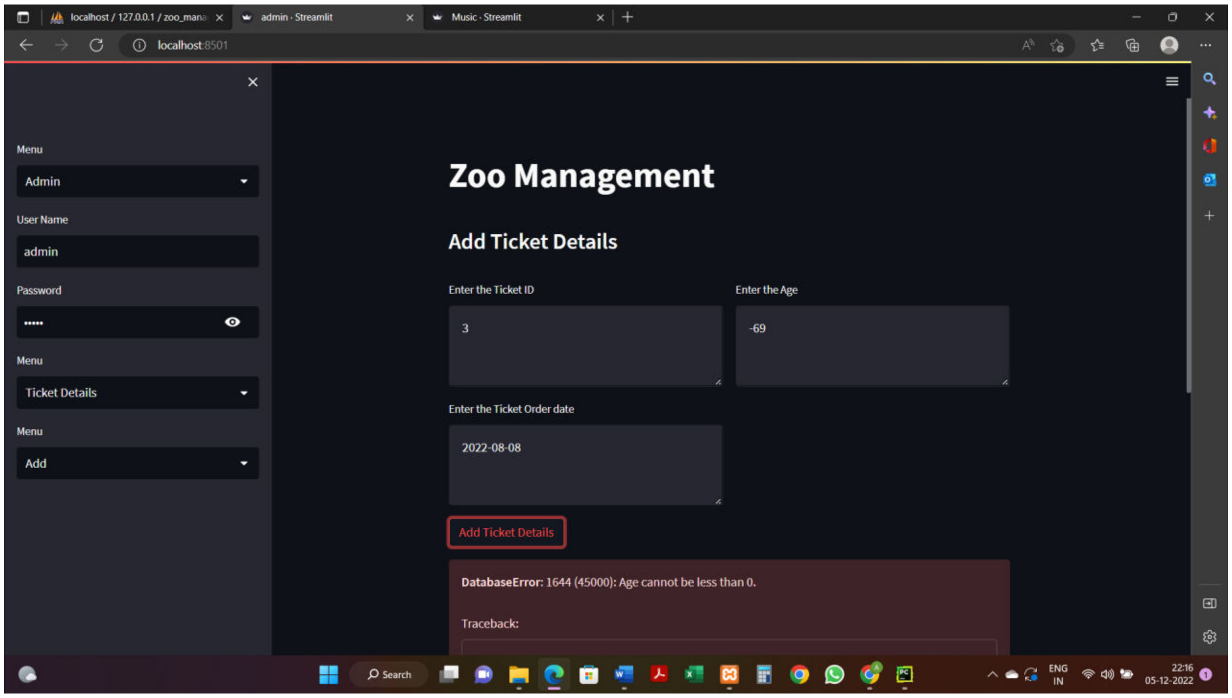


Edit — Details

Trigger name	AGE_ERROR
Table	ticket
Time	BEFORE
Event	INSERT
Definer	root@localhost

Definition

```
BEGIN DECLARE msg varchar(255);
SET msg = CONCAT("Age cannot be less than 0. ");
IF new.age<1 THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT =
msg;
END IF;
END
```



Zoo Management

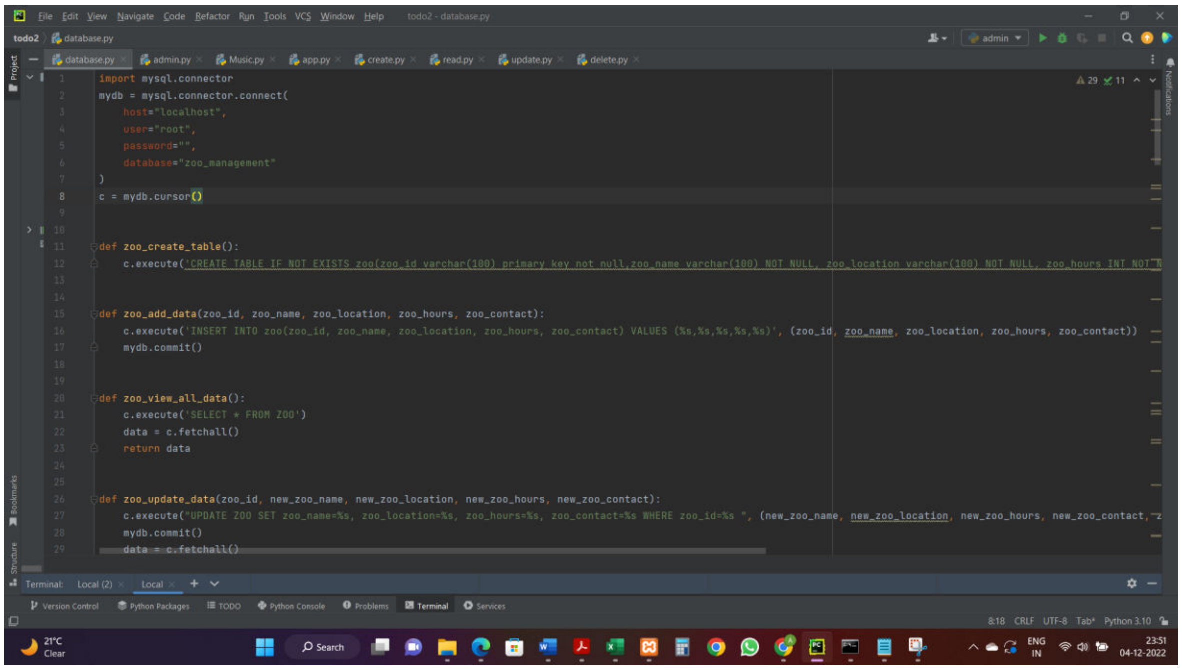
Add Ticket Details

Menu	Admin
User Name	admin
Menu	Ticket Details
Menu	Add
Enter the Ticket ID	3
Enter the Age	-69
Enter the Ticket Order date	2022-08-08

```
DatabaseError: 1644 (45000): Age cannot be less than 0.

Traceback:
```

AGE_ERROR Trigger prevents users to put unrealistic age of 0 and below while filling the ticket details.

CURSORS



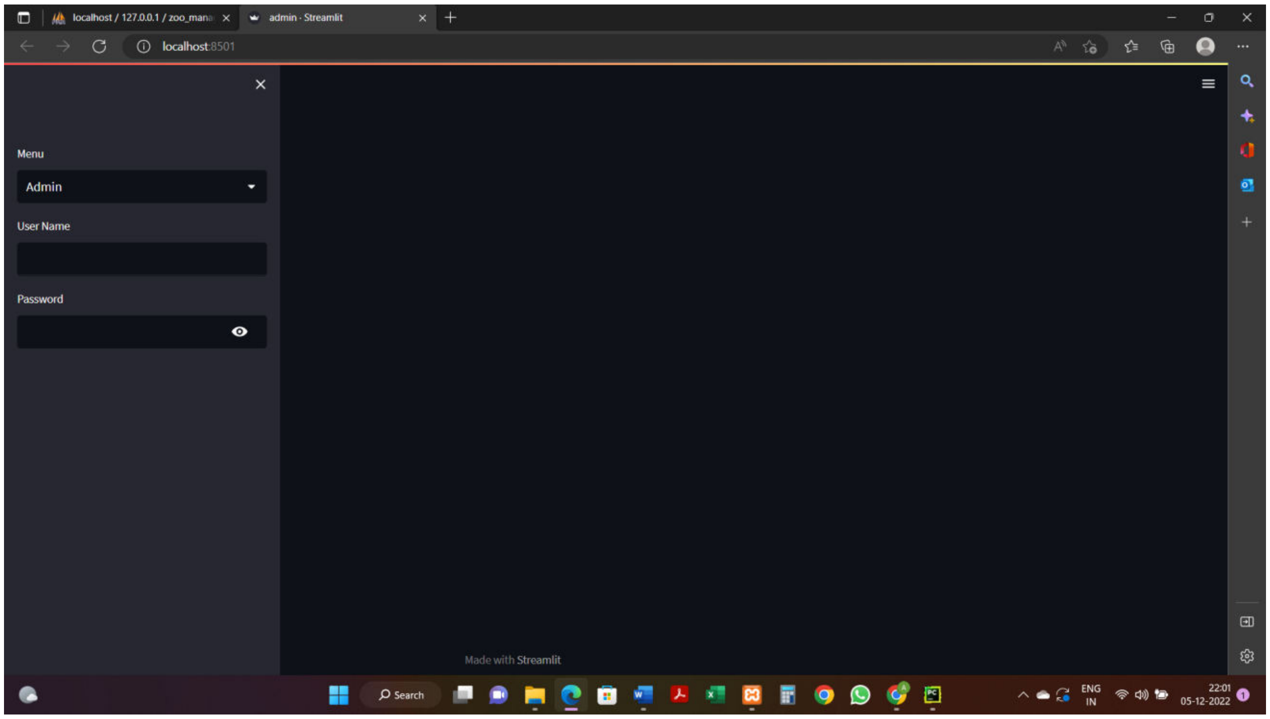
```python
import mysql.connector
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="zoo_management"
)
c = mydb.cursor()


def zoo_create_table():
    c.execute('CREATE TABLE IF NOT EXISTS zoo(zoo_id varchar(100) primary key not null,zoo_name varchar(100) NOT NULL, zoo_location varchar(100) NOT NULL, zoo_hours INT NOT N


def zoo_add_data(zoo_id, zoo_name, zoo_location, zoo_hours, zoo_contact):
    c.execute('INSERT INTO zoo(zoo_id, zoo_name, zoo_location, zoo_hours, zoo_contact) VALUES (%s,%s,%s,%s,%s)', (zoo_id, zoo_name, zoo_location, zoo_hours, zoo_contact))
    mydb.commit()


def zoo_view_all_data():
    c.execute('SELECT * FROM ZOO')
    data = c.fetchall()
    return data


def zoo_update_data(zoo_id, new_zoo_name, new_zoo_location, new_zoo_hours, new_zoo_contact):
    c.execute("UPDATE ZOO SET zoo_name=%s, zoo_location=%s, zoo_hours=%s, zoo_contact=%s WHERE zoo_id=%s ", (new_zoo_name, new_zoo_location, new_zoo_hours, new_zoo_contact, z
    mydb.commit()
    data = c.fetchall()
```

Developing a Frontend

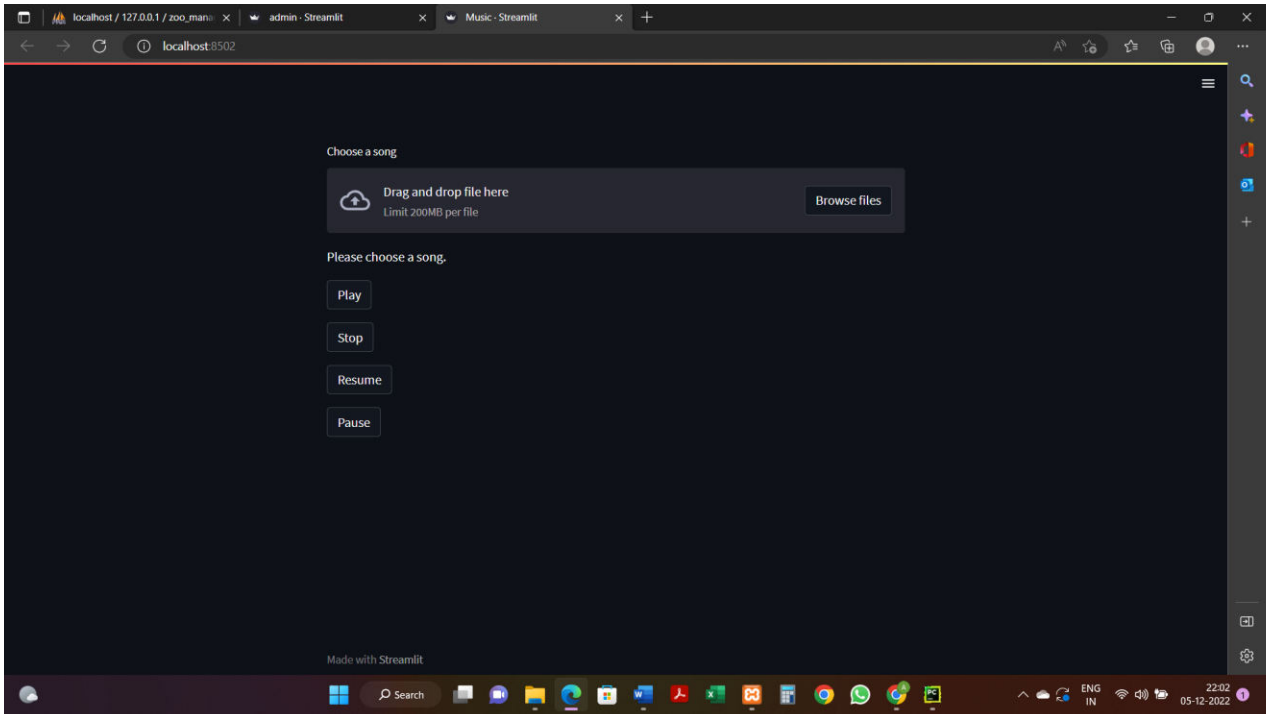
The frontend should support

1. Addition, Modification and Deletion of records from any chosen table
2. There should be an window to accept and run any SQL statement and display the result

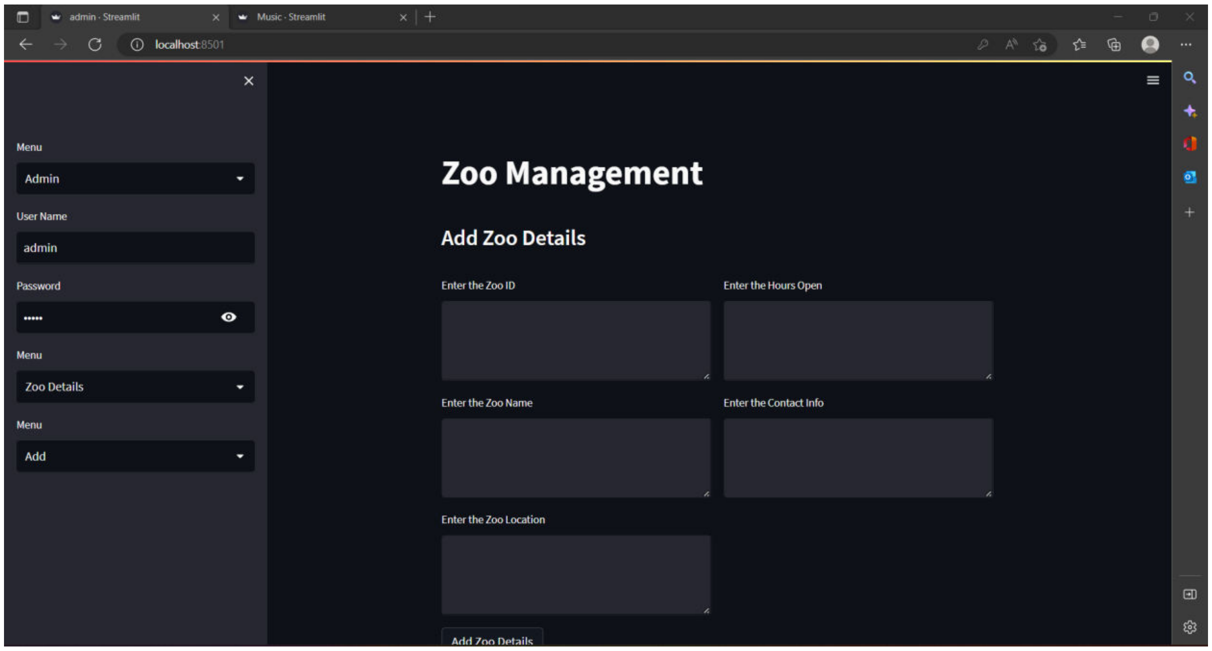
ADMIN



MUSIC



CREATE



admin · Streamlit | Music · Streamlit

localhost:8501

Menu: Admin

User Name: admin

Password: •••••

Menu: Zoo Details

Menu: Add

Zoo Management

Add Zoo Details

Enter the Zoo ID

Enter the Hours Open

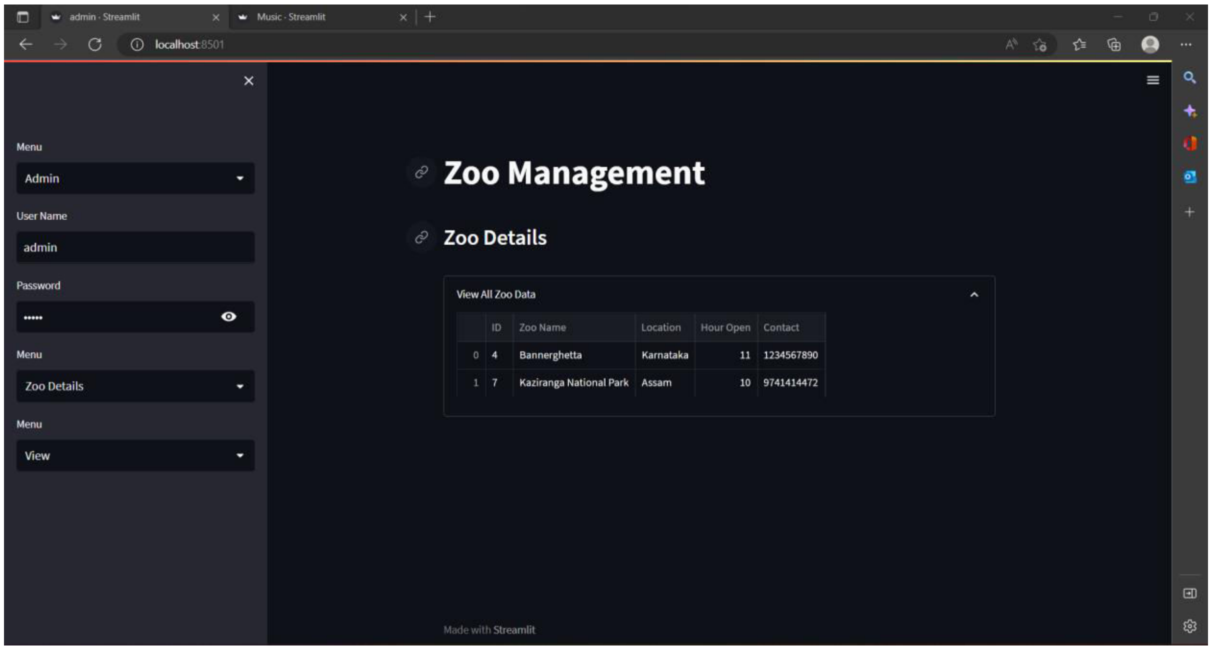
Enter the Zoo Name

Enter the Contact Info

Enter the Zoo Location

Add Zoo Details

READ



admin · Streamlit | Music · Streamlit

localhost:8501

Menu: Admin

User Name: admin

Password: •••••

Menu: Zoo Details

Menu: View

Zoo Management

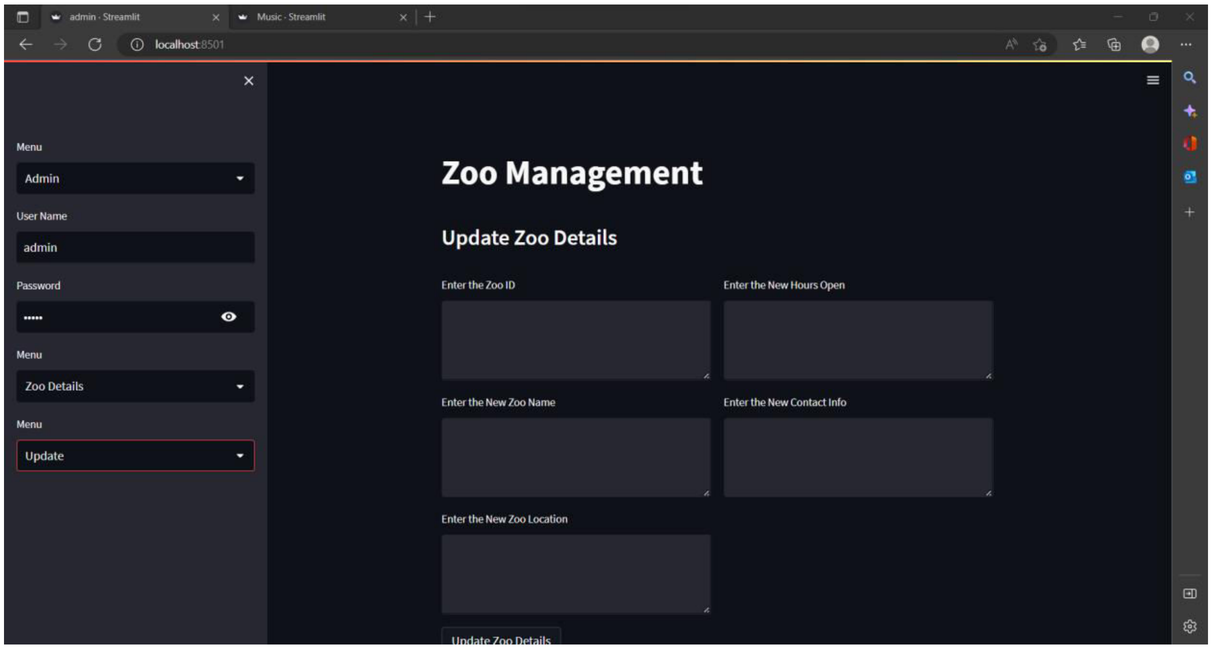
Zoo Details

View All Zoo Data

	ID	Zoo Name	Location	Hour Open	Contact
0	4	Bannerghetta	Karnataka	11	1234567890
1	7	Kaziranga National Park	Assam	10	9741414472

Made with Streamlit

UPDATE



Zoo Management

Update Zoo Details

Enter the Zoo ID

Enter the New Hours Open

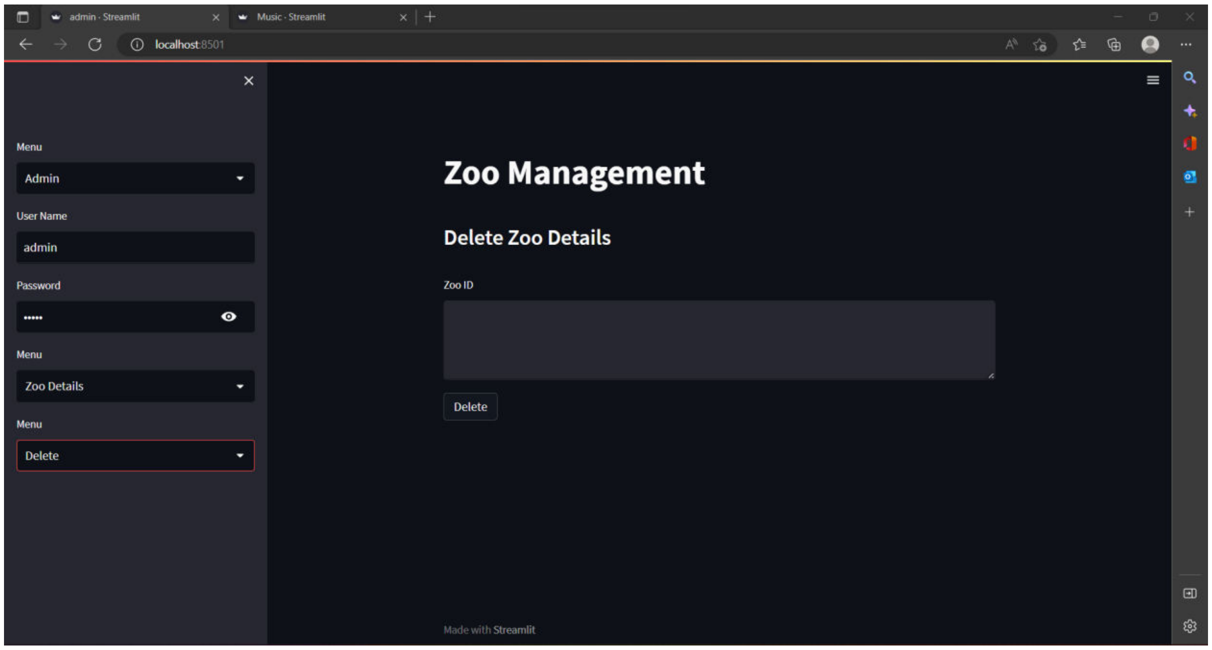
Enter the New Zoo Name

Enter the New Contact Info

Enter the New Zoo Location

Update Zoo Details

DELETE



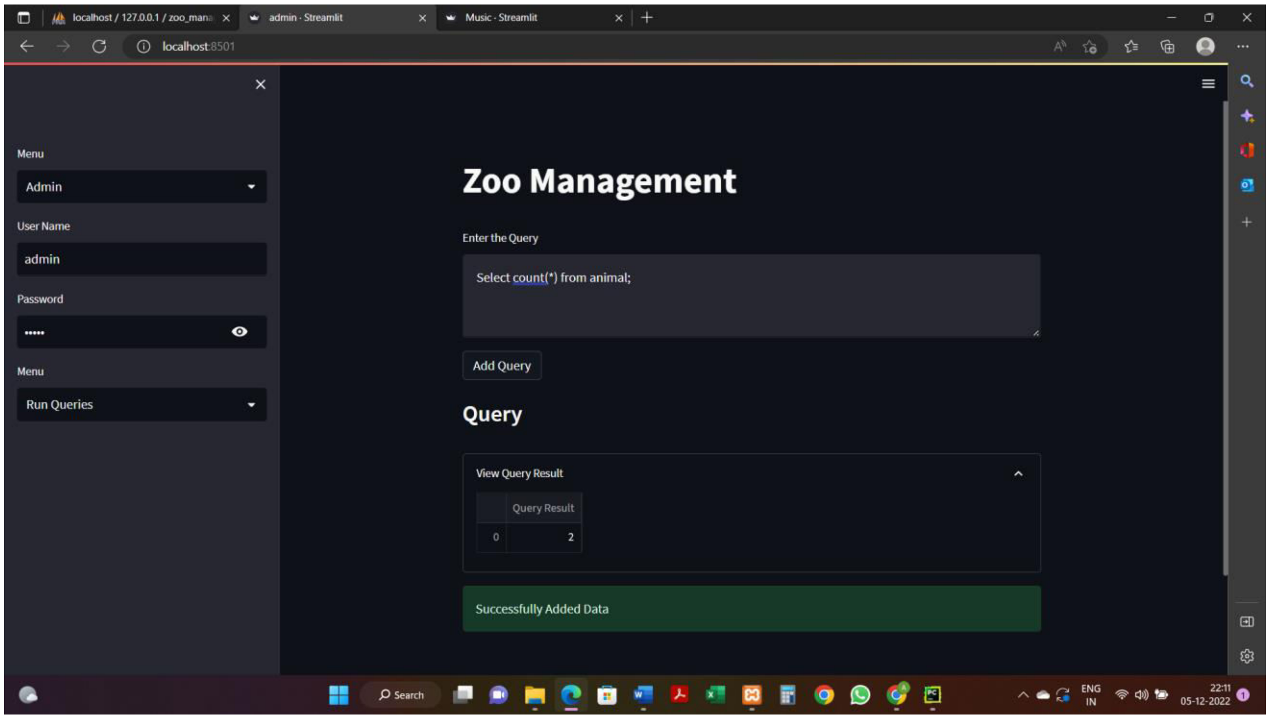
Zoo Management

Delete Zoo Details

Zoo ID

Delete

QUERY



Menu

Admin

User Name

admin

Password

•••••

Menu

Run Queries

Zoo Management

Enter the Query

```
Select count(*) from animal;
```

Add Query

Query

View Query Result

	Query Result
0	2

Successfully Added Data