VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI– 590018, KARNATAKA, INDIA

A MINI-PROJECT REPORT

on

"Animal Rescue Database Management"

Submitted in partial fulfillment of the requirements for the award of

BACHELOR OF ENGINEERING

in

ARTIFICIAL INTELLIGENCE & MACHINE LEARNING DEPARTMENT

Submitted By

MEGHASHREE K   **4VP21AI030**

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING
VIVEKANANDA COLLEGE OF ENGINEERING & TECHNOLOGY

[A Unit of Vivekananda Vidyavardhaka Sangha Puttur(R)]

Affiliated to Visvesvaraya Technological University and Approved by AICTE New Delhi & Govt., of Karnataka

Nehru Nagar, Puttur - 574 203, DK, Karnataka, India.

March, 2024

VIVEKANANDA COLLEGE OF ENGINEERING & TECHNOLOGY

[A Unit of Vivekananda Vidyavardhaka Sangha Puttur (R)]

Affiliated to Visvesvaraya Technological University and Approved by AICTE New Delhi & Govt. of Karnataka

Nehru Nagar, Puttur - 574203, DK, Karnataka, India

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

CERTIFICATE

Certified that **Database Management System Laboratory with mini project** work entitled "**Animal Rescue Database Management**" is carried out by **MEGHASHREE K bearing** USNs **4VP21AI030** respectively bonafide student of **Vivekananda College of Engineering & Technology, Puttur** in partial fulfillment for the award of **Bachelor of Engineering** in **Artificial Intelligence & Machine Learning** of the **Visvesvaraya Technological University,Belagavi** during year 2023-24. It is certified that all corrections/suggestions indicated during Internal Assessment have been incorporated in the report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

_____   _____   _____

Signature of the Guide   **Signature of the Guide**   **Signature of the HOD**
Prof. Abhishek Kumar K   **Prof. Harshitha K**   **Dr. Govindaraj P**

EXTERNAL VIVA

Name of the Examiners   Signature with date

1…………………………............   ....

2…………………………............   ....

ACKNOWLEDGEMENT

I take this opportunity to express my deep heartfelt gratitude to all those people who have helped us in the successful completion of the project.

First and foremost, I would like to express my sincere gratitude to our guides, **Prof. Abhishek Kumar K, Prof. Harshitha K** for providing excellent guidance, encouragement and inspiration throughout the project work. Without their invaluable guidance, this work would never have been a successful one.

I would like to express my sincere gratitude to my Head of the Department of Artificial Intelligence & Machine Learning, **Dr. Govindaraj P** for his guidance and inspiration.

I would like to thank my Principal, **Dr. Mahesh Prasanna K** for providing all the facilities and a proper environment to work in the college campus.

I thankful to all the teaching and non-teaching staff members of Artificial Intelligence & Machine Learning Department for their help and needed support rendered throughout the project.

DECLARATION

I,MEGHASHREE K, 4VP21AI030 student of fifth semester B.E. in Artificial Intelligence & Machine Learning, **Vivekananda College of Engineering & Technology**, **Puttur,** hereby declare that the project work entitled **"Animal Rescue Database Management"** has been carried out and duly executed by me at VCET, Puttur, under the guidance of **Prof. Abhishek Kumar K, Prof. Harshitha K**, Assistant Professors, Department of Artificial Intelligence & Machine Learning, Vivekananda College of Engineering & Technology, Puttur, and submitted in partial fulfillment of the requirements for the award of degree in **Bachelor of Engineering in Artificial Intelligence & Machine Learning** by **Visvesvaraya Technological University**, Belagavi during the academic year 2023-2024.

MEGHASHREE K   4VP21AI030

Date:18/03/2024

Place: VCET.

ABSTRACT

The Database Management System (DBMS) Mini-Project is a practical aimed at designing and implementing an efficient and well-structured relational database system for a specific organizational or domain scenario. The main aim of the project is the management of the database of the Animal Rescue management. The development and deployment of an Animal Rescue Database Management System (ARDBMS) is presented in this abstract, highlighting the efficacy of utilizing relational database technology, specifically MySQL, to support and streamline animal rescue operations. This system is designed to efficiently collect, organize, and utilize a variety of data relevant to animal rescue efforts, ultimately aiding organizations in making well-informed decisions and promoting responsible animal welfare practices.

The DBMS serves as a comprehensive repository for a wide range of animal-related information, encompassing data such as animal profiles, pet ,animal and bird details and sold products. The interface of the system provides real-time access to data and insights, empowering rescue teams with timely information crucial for decision-making. The MySQL database backbone ensures data integrity, reliability, and security, safeguarding sensitive animal-related information and adhering to privacy regulations. In conclusion, the Animal Rescue Database Management System demonstrates the potential of leveraging relational database technology to enhance animal rescue operations. By centralizing diverse data within MySQL, it empowers rescue organizations to contribute effectively to animal welfare efforts and sustainable practices.

Overall, the DBMS Mini Project serves as a valuable learning opportunity, bridging theoretical knowledge with practical application in the realm of database management.

Table of Contents

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Introduction to Database Management System

A database management system (DBMS) refers to the technology for creating and managing databases. DBMS is a software tool to organize (create, retrieve, update, and manage) data in a database. A software for storing and retrieving user by considering appropriate security measures. It allows users to create their own databases as per their requirement.

It consists of group of programs which manipulate the database and provide an interface between the database. It includes the user of the database and other application programs. The DBMS accepts the request for data from an application and instructs the operating system to provide the specific data. In large systems, a DBMS helps users and other third-party software to store and retrieve data.

If it is any field that has contributed to the greatest advancements in the world today, it is the field of information technology, commonly known by the acronym IT. When most people hear of the phrase 'information technology', they tend to picture several Tecno-geeks seated by the computer and fixing some computer software.

1.2 Types of DBMS:

There are mainly 4 types of DBMS, which are Hierarchical, Relational, Network, and Object Oriented DBMS

- **Hierarchical DBMS**: As the name suggests, this type of DBMS has a style of predecessor-successor type of relationship. So, it has a structure similar to that of a tree, wherein the nodes represent records and the branches of the tree represent fields.

- **Relational DBMS (RDBMS)**: This type of DBMS, uses a structure that allows the users to identify and access data in relation to another piece of data in the database.

- **Network DBMS**: This type of DBMS supports many to many relations wherein multiple member records can be linked.

- **Object-oriented DBMS**: This type of DBMS uses small individual software called objects. Each object contains a piece of data, and the instructions for the actions to be done with the data.

1.3 Applications of DBMS

1. **Library Management System**: There are lots of books in the library so; it is tough to store the record of all the books in a register or copy. So, the database management system (DBMS) is used to maintain all the information related to the name of the book, issue date, availability of the book, and its author.

2. **Railway Reservation System**: In the railway reservation system, the database is required to store the record or data of ticket bookings, status about train's arrival, and departure. Also, if trains get late, people get to know it through database update.

3. **Banking**: Database management system is used to store the transaction information of the customer in the database.

4. **Credit card transactions**: Database Management system is used for purchasing on credit cards and generation of monthly statements.

5. **Education Sector**: Presently, examinations are conducted online by many colleges and universities. They manage all examination data through the database management system (DBMS). In spite that student's registrations details, grades, courses, fee, attendance, results, etc. all the information is stored in the database.

6. **Social Media Sites**: We all use of social media websites to connect with friends and to share our views with the world. Daily, millions of peoples sign up for these social

media accounts like Pinterest, Facebook, Twitter, and Google plus. By the use of the database management system, all the information of users are stored in the database and, we become able to connect with other people.

7. **Telecommunication**: Without DBMS any telecommunication company can't think. The Database management system is necessary for these companies to store the call details and monthly postpaid bills in the database.

8. **Finance**: The database management system is used for storing information about sales, holding and purchases of financial instruments such as stocks and bonds in a database.

9. **Online Shopping**: Everyone wants to shop through online shopping websites (such as Amazon, Flipkart, snap deal) from home. So, all the products are sold and added only with the help of the database management system (DBMS). Invoice bills, payments, purchase information all of these are done with the help of DBMS.

10. **Human Resource Management**: Big firms or companies have many workers or employees working under them. They store information about employee's salary, tax, and work with the help of database management system (DBMS).

11. **Manufacturing**: Manufacturing companies make different types of products and sale them on a daily basis. In order to keep the information about their products like bills, purchase of the product, quantity, supply chain management, database management system (DBMS) is used.

12. **Airline Reservation System**: This system is the same as the railway reservation system. This system also uses a database management system to store the records of flights departure, arrival, and delay status.

1.4 Introduction to MySQL

MySQL is a Relational Database Management System ("RDBMS"). It is used by most modern websites and web-based services as a convenient and fast-access storage and retrieval for large volumes of data. MySQL is open-source and free software under the GNU license. It is supported by Oracle Company.

It is developed, marketed, and supported by MySQL AB, a Swedish company, and written in C programming language and C++ programming language. MySQL supports many Operating Systems like Windows, Linux, MacOS, etc. with C, C++, and Java languages.

MySQL can also be accessed using many tools. It can be easily communicated with via PHP (PHP Hypertext Preprocessor), a scripting language whose primary focus is to manipulate HTML for a webpage on the server before it is delivered to a client's machine. A user can submit queries to a database via PHP, allowing insertion, retrieval and manipulation of information into/from the database.

MySQL server design is multi-layered with independent modules and is fully multithreaded by using kernel threads. It can handle multiple CPUs if they are available. MySQL Server works in client/server or embedded systems. and it works on many different platforms.

1.5 MYSQL command syntax

The four main categories of SQL statements are as follows

DML (Data Manipulation Language)

DML statements affect records in a table. These are basic operations we perform on data such as selecting a few records from a table, inserting new records, deleting unnecessary records, and updating/modifying existing records.

- **SELECT** – select records from a table

    SELECT column1, column2, ... FROM table_name;

- **INSERT** – insert new records

    INSERT INTO table_name (column1, column2, column3, ...)

    VALUES (value1, value2, value3, ...);

- **UPDATE** – update/Modify existing records

    UPDATE table_name

    SET column1 = value1, column2 = value2, ...

    WHERE condition;

- **DELETE** – delete existing records

    DELETE FROM table_name WHERE condition;

DDL (Data definition Language)

DDL statements are used to alter/modify a database or table structure and schema. These statements handle the design and storage of database objects.

- **CREATE** – create a new Table, database, schema

    CREATE TABLE table_name(

    column 1 datatype,

    column2 datatype,

    column3 datatype, ...

    );

- **ALTER** – alter existing table, column description

   ALTER TABLE table_name ADD column_name datatype;

- **DROP** – delete existing objects from database

   DROP TABLE table_name;

DCL (Data Control Language)

DCL statements control the level of access that users have on database objects.

- **GRANT** – allows users to read/write on certain database objects

   GRANT ALL PRIVILEGES ON database_name.*

   TO 'username'@'localhost';

- **REVOKE** – keeps users from read/write permission on database objects

   REVOKE privileges ON object FROM user;

TCL (Transaction Control Language)

TCL statements allow you to control and manage transactions to maintain the integrity of data within SQL statements.

- **BEGIN** Transaction – opens a transaction

   [begin_label:] BEGIN

   [statement_list]

   END [end_label]

- **COMMIT** Transaction – commits a transaction

    COMMIT [{TRAN |TRANSACTION} [transaction_name | @tran_name_ variable]] [WITH (DELAYED_DURABILITY = {OFF | ON})]

    [;]

- **ROLLBACK** Transaction – ROLLBACK a transaction in case of any error

    ROLLBACK {TRAN|TRANSACTION}

    [transaction_name | @tran_name_variable

    | savepoint_name | @savepoint_variabl] ;

    ]

CHAPTER 2

ANALYSIS AND REQUIREMENT SPECIFICATION

2.1 Purpose of this project

Its aim at bringing a platform which is user friendly and helpful regarding the information of animals and birds. This project is to create a centralized platform that optimizes the operations of an animal rescue management organization. It achieves this by efficiently managing data related to rescued animals, streamlining process such as intake, sold products, customer information, sales details, and ensuring compliance with regulations, and ultimately enhancing the welfare and outcomes for rescued anima, Scope of this project:

- Developing user-friendly interfaces for various for various stakeholders such as customers, salesman s to input, access, and manage data effectively.

- To improve efficiency and reduce manual efforts.

- Providing features to manage and promote animal adoption, including profiles of available animals.

- system complies with relevant regulations and standards related to animal welfare, data privacy, and organizational practices.

2.2 Scope of this project

- Managing information about rescued animals.

- Facilitating communication and coordination among rescue organizations.

- Ensuring data security and privacy compliance.

2.3 Functional Requirements

Modules

- **Login Module**: Used for managing the login details. We can access the page by logging in by giving the username and password.

- **Animal Intake Module**: Used for managing the details of animals. It provides a set of services to the users.

- **Sold Pets/Sold Products Module**: Used for managing the information and details of the sales product and animals(animal/birds). It is used generally by the system users.

2.4 Non-Functional Requirements

Hardware requirements

- **Processor**   **:** Intel® Celeron® CPU N3060 @1.60GHz
- **RAM**   **:** 4.00 GB
- **Hard Disk**   **:** 1 TB
- **Compact Disk** **:** CD-ROM, CD-R, CD-RW
- **Input device**   **:** Keyboard, mouse
- **Output device** **:** Monitor screen

Software requirements

- **Operating System** **:** 64bit operating system, x64-based processor
- **Database**   **:** MYSQL
- **Tools**   **:** PHP, Xampp Server 3.2.2

CHAPTER 3

DESIGN

3.1 Entity Relationship Diagram

Data schema in graphical form is called ER Diagram. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities. An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of animal rescue.
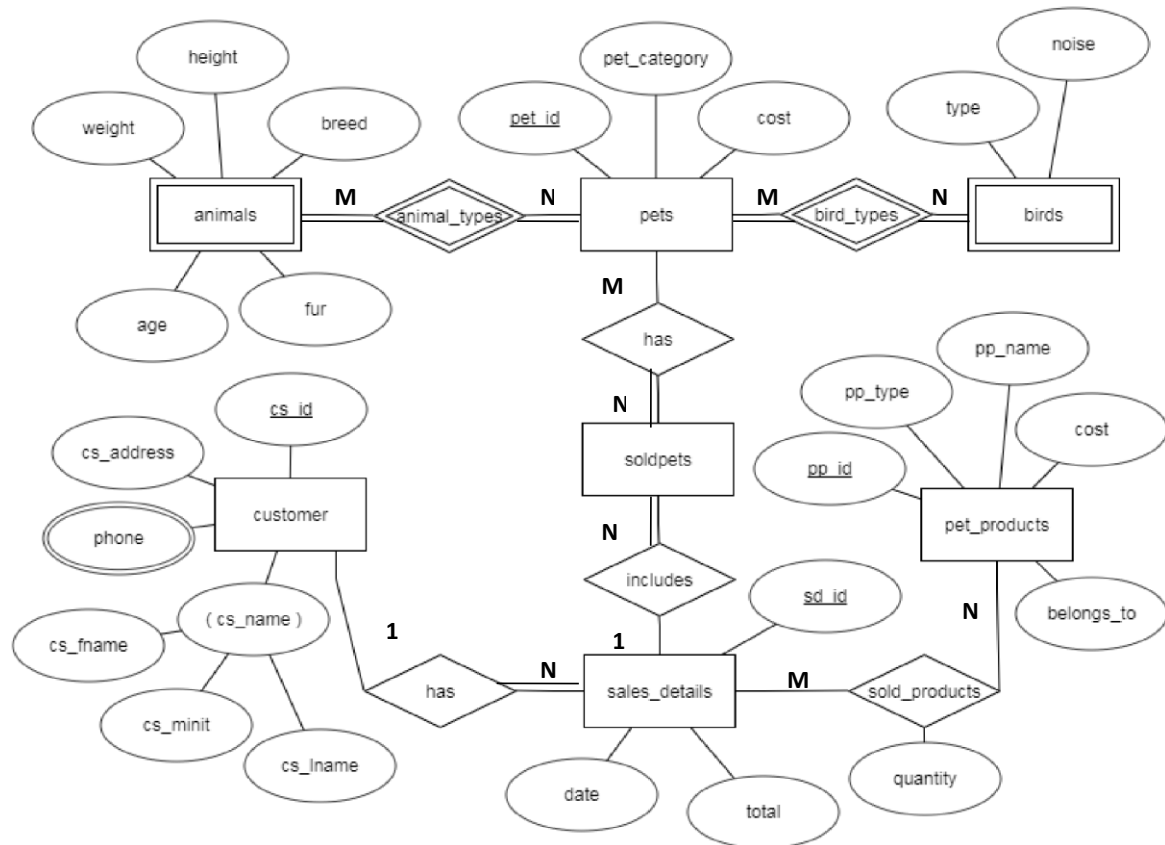

Fig 3.1 ER diagram for Animal rescue database

3.2 Schema Diagram

The term "schema" refers to the organisation of data as a blueprint of how the database is constructed. The formal definition of a database schema is a set of formulas called integrity constraints imposed on a database. A relational schema shows references among fields in the database. When a primary key is referenced in another table in the database, it is called a foreign key. This is denoted by an arrow with the head pointing at the referenced key attribute.
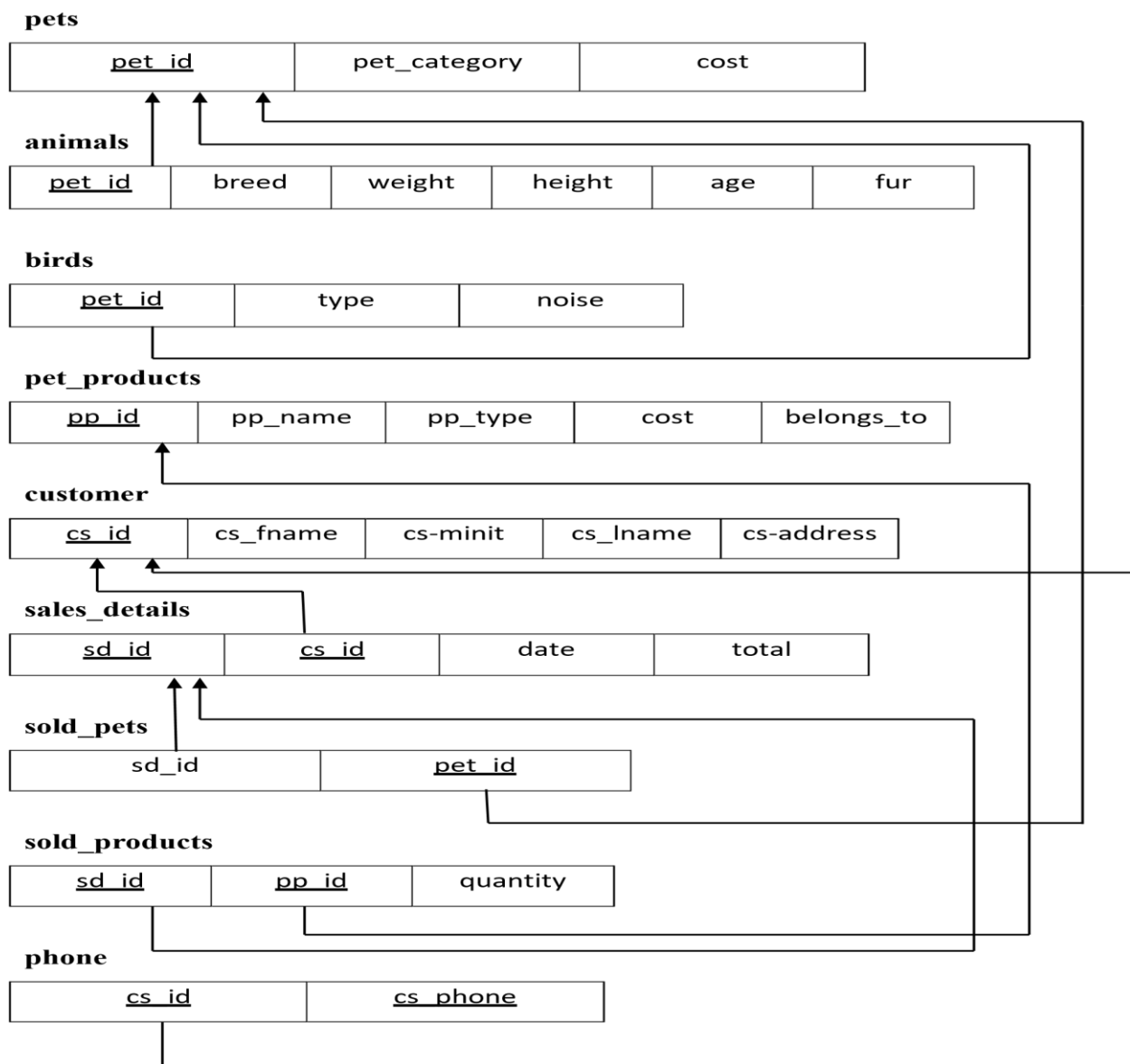
pets

pet_id	pet_category	cost

animals

pet_id	breed	weight	height	age	fur

birds

pet_id	type	noise

pet_products

pp_id	pp_name	pp_type	cost	belongs_to

customer

cs_id	cs_fname	cs-minit	cs_lname	cs-address

sales_details

sd_id	cs_id	date	total

sold_pets

sd_id	pet_id

sold_products

sd_id	pp_id	quantity

phone

cs_id	cs_phone

Fig 3.2: Schema diagram for Animal rescue database

CHAPTER 4
IMPLEMENTATION

4.1 Table structure, Data Dumping and Constraints for tables

Creation of pets table

```
create table pets (pet_id varchar (9) not null,
pet_category varchar(15) not null,
cost int (11) not null,
primary key(pet_id));
```

Insertion of pets table

```
INSERT INTO 'pets` (`pet_id`, `pet_category`, `cost`) VALUES
('pa01', 'dog','8000'),
('pa02', 'cat', '3000'),
('pa03', 'dog', '8500'),
('pa04', 'dog','15000'),
('pa05', 'cat', '3500');
```

Creation of animals table

```
create table animals (pet_id varchar (9) not null,
breed varchar (30) not null,
weight float not null,
height float not null,
age    int (11) not null,
fur varchar (15) not null,
primary key(pet_id),
forgien key(pet_id) references pets(pet_id) on delete cascade);
```

Insertion of animals table

INSERT INTO animals (`pet_id`, `breed`, `weight`, `height`, `age`, `fur`)VALUES

('pa01', 'labrador','11.3', '30', '2', 'white'),

('pa02', 'parsian', '3.6', '20','2', 'white'),

('pa03', 'goldenretriever', '12.5', '40', '2', 'gloden'),

('pa04','boxer', '11.5', '45', '3', 'black'),

('pa05', 'rag doll', '2.6', '20', '5','white');

Creation of birds table

create table birds (pet_id varchar (9) not null,

type varchar (25) not null,

noise varchar (10) not null,

primary key (pet_id),

forgien key(pet_id) references pets(pet_id) on delete cascade);

Insertion of birds table

INSERT INTO `birds` (`pet_id`, `type`, `noise`) VALUES

('pb01', 'grey parrot', 'moderate'),

('pb02', 'black cheeked', 'low'),

('pb03', 'grey headed', 'moderate'),

('pb04', 'lilian', 'moderate'),

('pb05', 'white cockatoo', 'moderate') ;

Creation of pet products table

create table pet_products (pp_id varchar (9) not null,

pp_name varchar(30) not null,

pp_type varchar (20) not null,

cost int (11) not null,

belongs_to varchar (20) not null,

primary key(pp_id));

Insertion of pet products table

INSERT INTO `pet_products` (`pp_id`, `pp_name`, `pp_type`, `cost`, `belongs_to`) VALUES

('pp01', 'dog collar', 'accesories', '500', 'dog'),

('pp02', 'chain', 'accesories', '100', 'cat'),

('pp03', 'pedigree', 'food', '1500', 'dog'),

('pp04', 'mouth mask', 'accesories', '250', 'dog'),

('pp05', 'food bowl', 'accesories', '250', 'dog ') ;

Creation of customer table

create table customer (cs_id varchar (9) not null,

cs_fname varchar(10) not null, cs_minit varchar

(10) not null,   cs_lname varchar (10) not null,

cs_address varchar (30) not null,

primary key(cs_id));

Insertion of customer table

INSERT INTO `customer`(`cs_id`,`cs_fname`,`cs_minit`,`cs_lname`,`cs_address`) VALUES

('cs01', 'Naveen', 'kumar', 'k', 'Mandya'),

('cs02', 'manjunath', 'kumar', 'h v', 'BENGALURU'),

('cs03', 'pavan', 'chikkanna', 'gowda', 'BENGALURU'),

('cs04', 'kushal', 'kumar', 'k', 'BENGALURU'),

('cs05', 'ravi', 'shankar', 'c', 'BENGALURU') ;

Creation of phone table

create table phone (cs_id varchar (9) not null,

cs_phone bigint(10) not null,   primary key(cs_id,cs_phone),

foreign key(cs_id) references customer(cs_id)on delete cascade);

Insertion of phone table

INSERT INTO `phone`(`cs_id`, `cs_phone`) VALUES

('cs01', '8867762336'),

('cs01', '9902587276'),

('cs03', '9845034784'),

('cs04', '6361261639'),

('cs05', '86660873855') ;

Creation of sales details table

create table sales_details(sd_id varchar(9) not null,

cs_id varchar (9) not null,

date date not null,

total int (11) not null,

primary key(sd_id,cs_id),

foreign key(cs_id) references customer(cs_id)on delete cascade);

Insertion of sales details table

INSERT INTO `sales_details` (`sd_id`, `cs_id`, `date`, `total`) VALUES

('sd01', 'cs03', '2018-10-26', '9500'),

('sd02', 'cs01', '2018-11-01', '3000'),

('sd03', 'cs03', '2018-11-08', '500'),

('sd04', 'cs04', '2018-11-15', '250'),

('sd05', 'cs02', '2018-11-17', '9350') ;

Creation of sold pets table

create table sold_pets(sd_id varchar(9) not null,

pet_id varchar (9) not null,

primary key(pet_id),

foreign key(sd_id) references sales_details(sd_id) on delete cascade,

foreign key(pet_id)references pets(pet_id)on delete cascade);

Insertion of sold pets table

INSERT INTO `sold_pets` (`sd_id`, `pet_id`) VALUES

('sd01', 'pa01'),

('sd02', 'pa02'),

('sd05', 'pa03'),

('sd06', 'pb02'),

('sd06', 'pb04');

Creation of sold products table

create table sold_products(sd_id varchar(9) not null,

pp_id varchar (9) not null,

quantity int (11) not null,

primary key(pet_id,pp_id),

foreign key(sd_id) references sales_details(sd_id) on delete cascade,

foreign key(pp_id)references pet_products(pp_id)on delete cascade);

Insertion of sold products table

INSERT INTO `sold_products` (`sd_id`, `pp_id`, `quantity`) VALUES

('sd01', 'pp03', '1'),

('sd03', 'pp01', '1'),

('sd04', 'pp04', '1'),

('sd05', 'pp05', '1'),

('sd05', 'pp06', '2');

Constraints for table animal

ALTER TABLE animals

ADD **CONSTRAINT** animals_ibfk_1 FOREIGN KEY (pet_id)

REFERENCES pets (pet_id) ON DELETE CASCADE;

Constraints for table birds

ALTER TABLE birds

ADD CONSTRAINT birds_ibfk_1 FOREIGN KEY (pet_id)

REFERENCES pets (pet_id) ON DELETE CASCADE**;**

Constraints for table phone

ALTER TABLE phone

ADD CONSTRAINT phone_ibfk_1 FOREIGN KEY (cs_id)

REFERENCES customer (cs_id) ON DELETE CASCADE;

Constraints for table sales_details

ALTER TABLE sales_details

ADD CONSTRAINT sales_details_ibfk_1 FOREIGN KEY

(cs_id) REFERENCES customer (cs_id) ON DELETE

CASCADE;

Constraints for table sold_pets

ALTER TABLE sold_pets

ADD CONSTRAINT sold_pets_ibfk_1 FOREIGN KEY (pet_id)

REFERENCES pets (pet_id) ON DELETE CASCADE,

ADD CONSTRAINT sold_pets_ibfk_2 FOREIGN KEY (sd_id)

REFERENCES sales_details (sd_id) ON DELETE CASCADE;

Constraints for table sold_products

ALTER TABLE sold_products

ADD CONSTRAINT sold_products_ibfk_1 FOREIGN KEY

(sd_id) REFERENCES sales_details (sd_id) ON DELETE

CASCADE,

ADD CONSTRAINT sold_products_ibfk_2 FOREIGN KEY

(pp_id) REFERENCES pet_products (pp_id) ON DELETE

CASCADE

CHAPTER 5

SNAPSHOTS
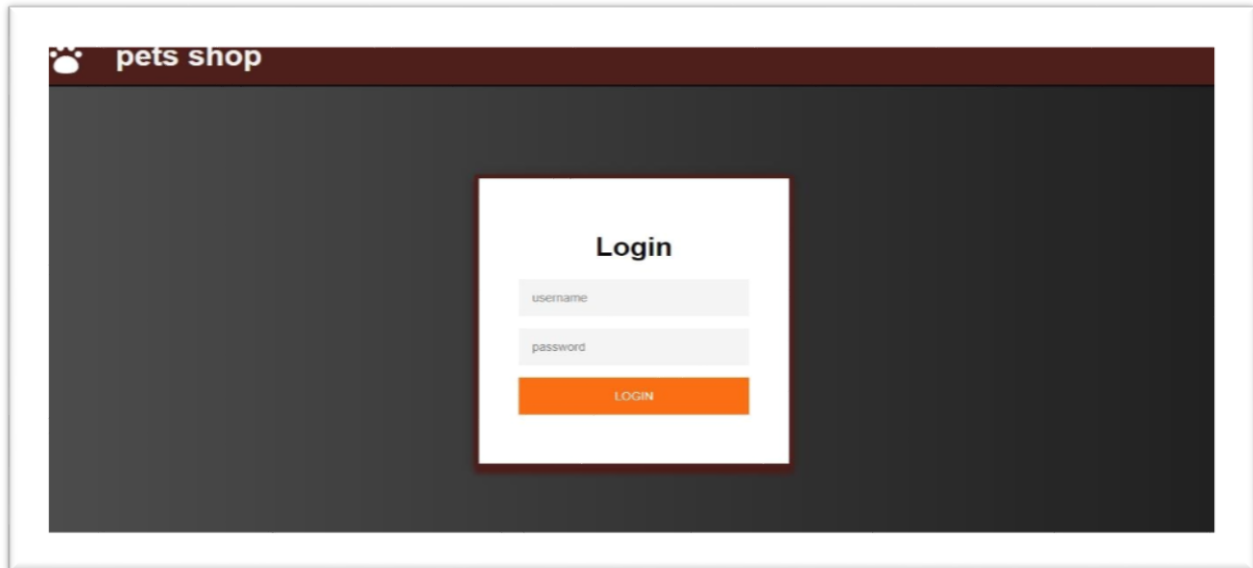
Fig 5.1: Login Page

This page asks admin username and password for authentication, If the authentication is successful then it loads home page.
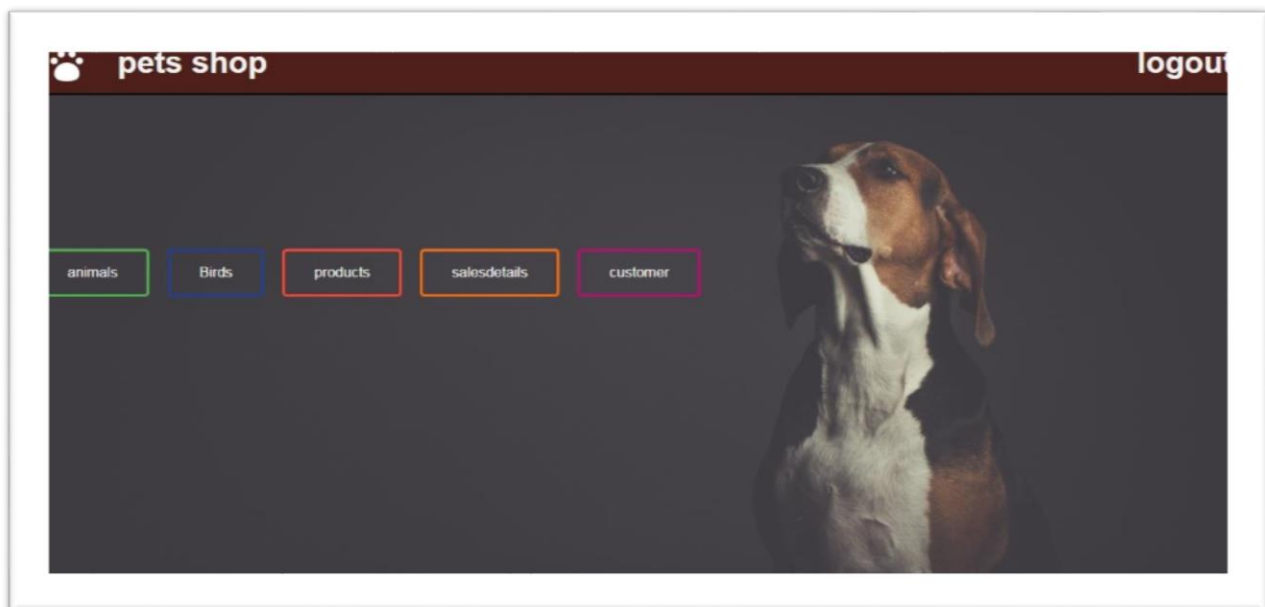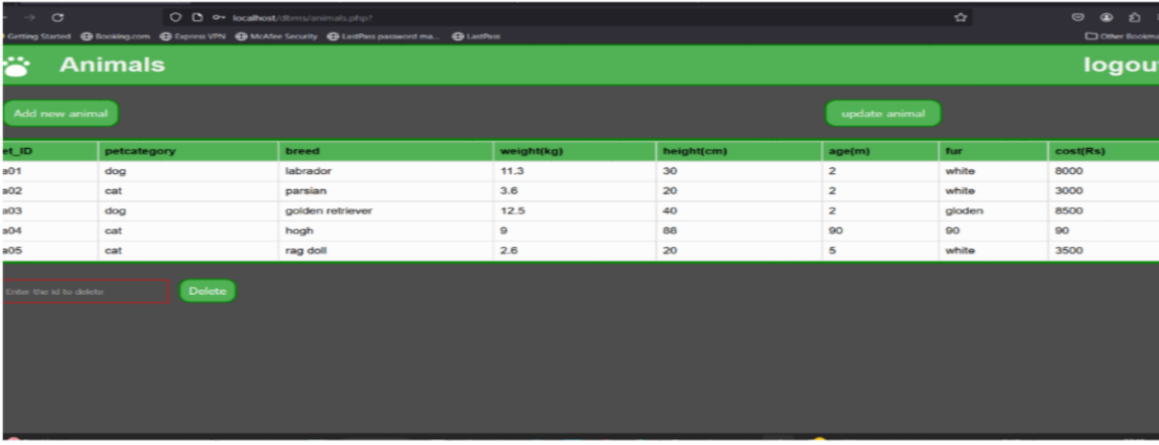
Fig 5.2: Home Page
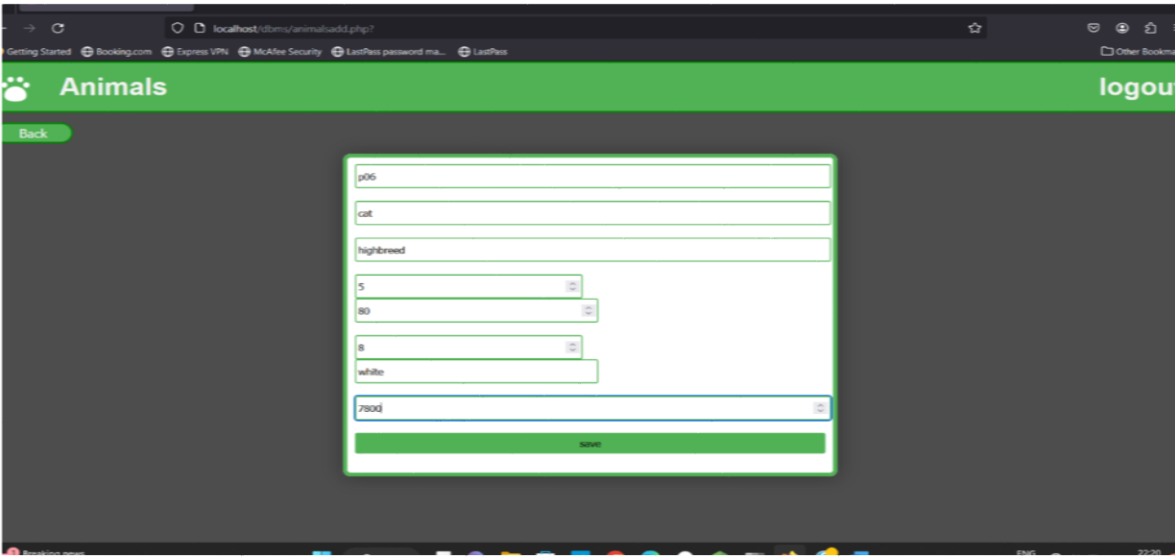
This page provides links to animals page, birds page, products page and customer page.

Fig 5.3: Animals table

This page displays the animals data and also provides link to access insertion and updation page of animals and also at left bottom of the page it gives an option for deletion.

Fig 5.4: Values to insert in Animals table

This page displays the data to insert the values of animals details.
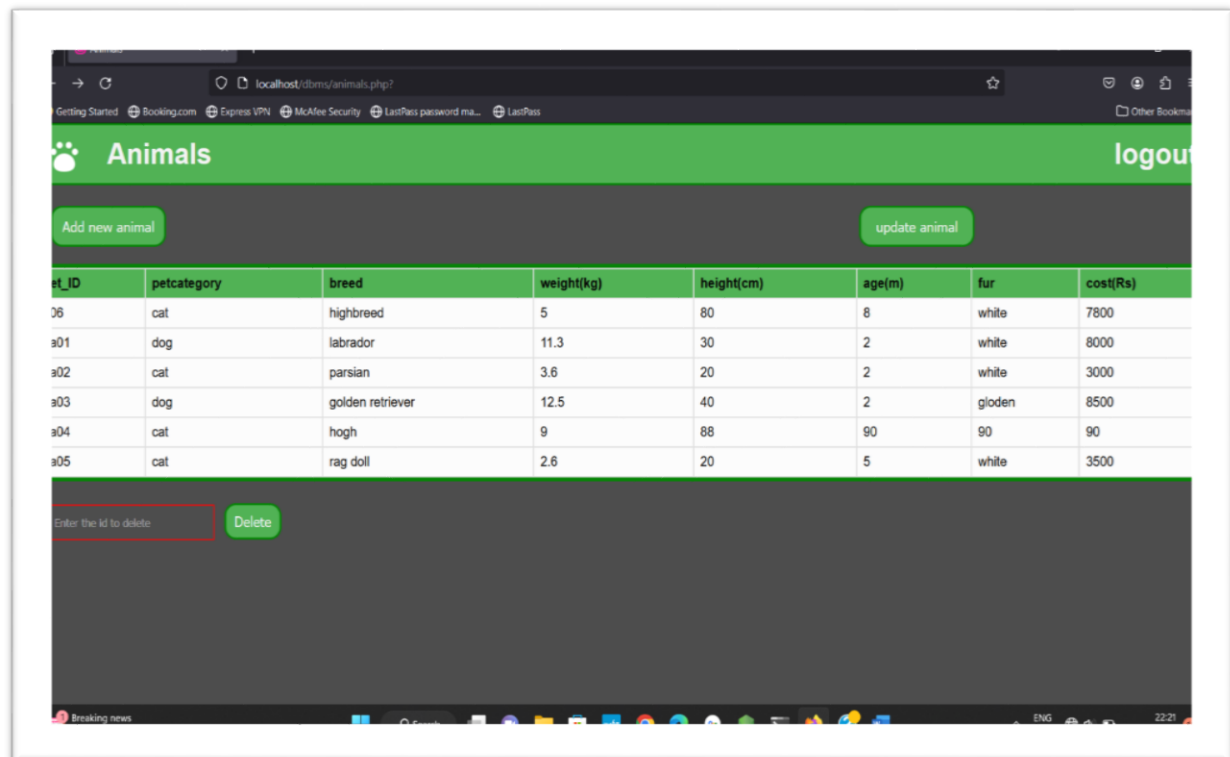
Fig 5.5: Insertion of Animal table

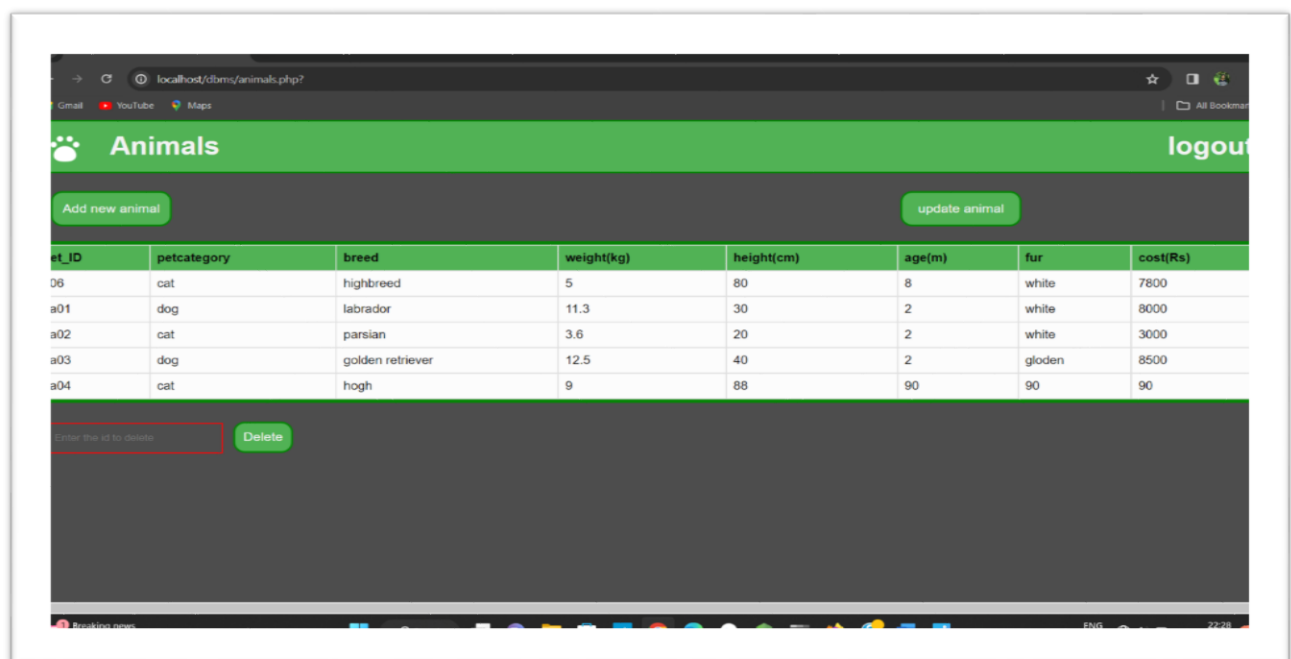This page accept the data to save in animals entity and pet entity.

Fig 5.6: Deletion of animal table

This page accept the data to delete in animals entity and pet entity.
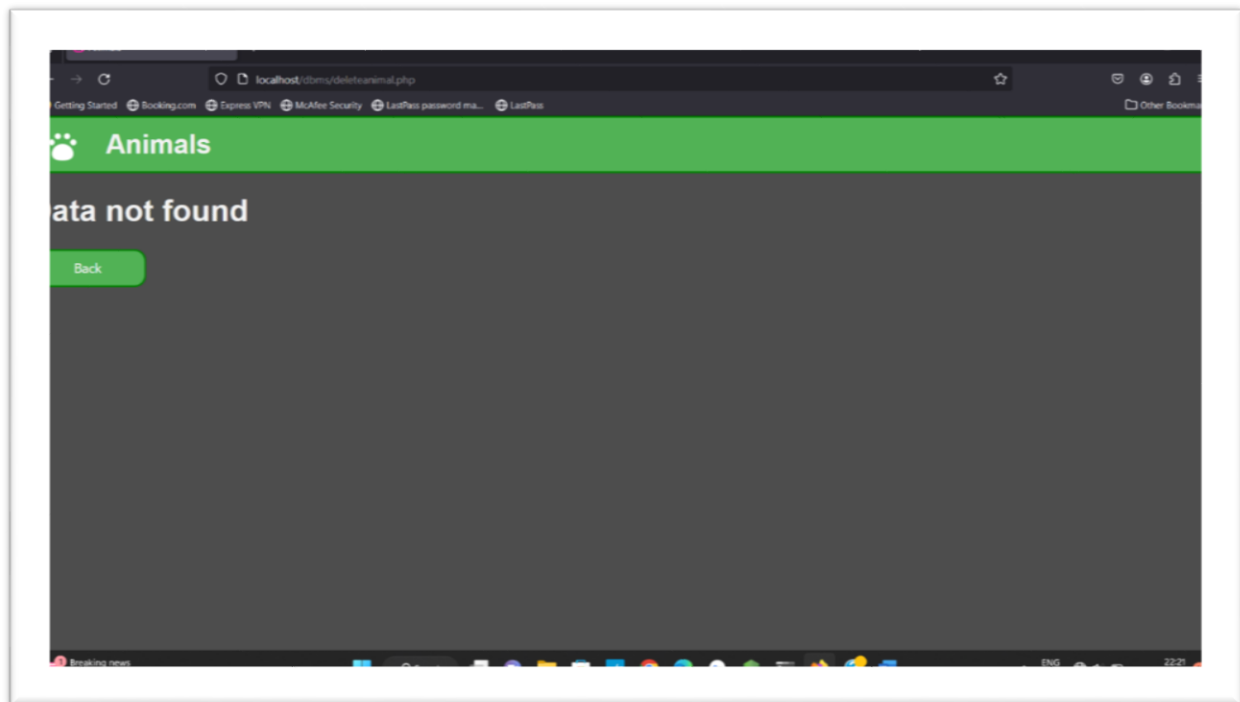
Fig 5.7: Unknown values table

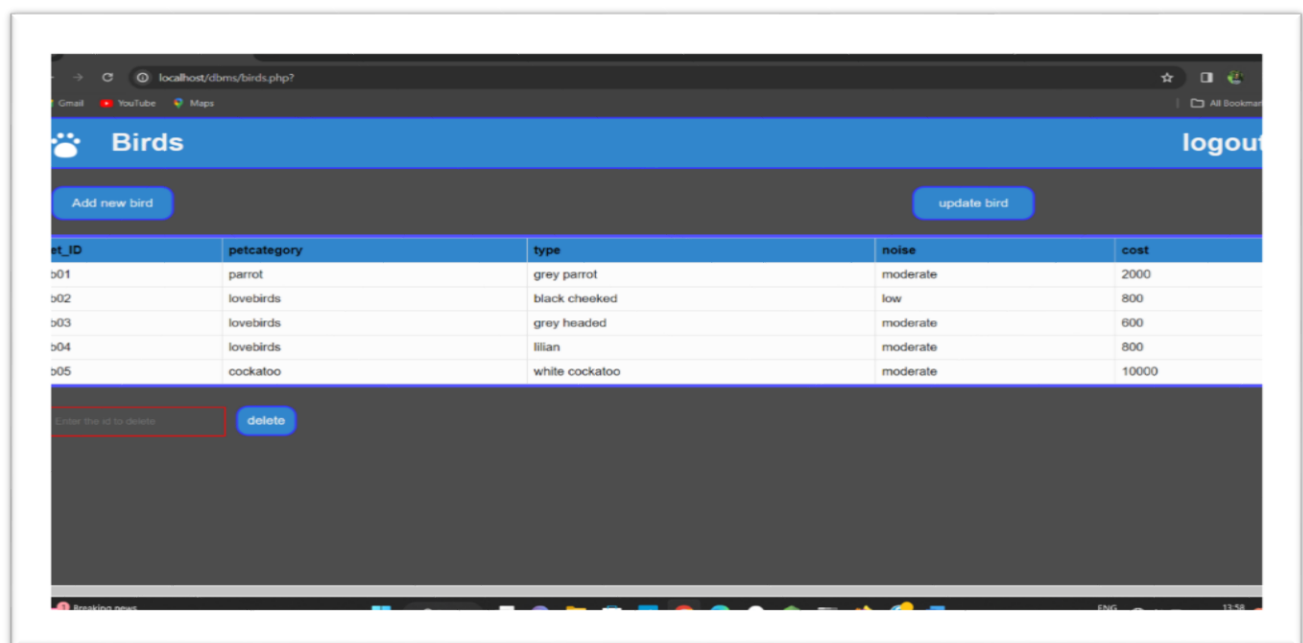Delete the unknown values from the animal table that shows result as data not found.

Fig 5.8: Birds table

This page displays the birds data and also provides link to access insertion and updation page of birds and also at left bottom of the page it gives an option for deletion.

Fig 5.9: Pets product table

This page displays the pets product data and also provides link to access insertion and updation page of pets product and also at left bottom of the page it gives an option for deletion.

Fig 5.10: Sales details table

This page displays the sales details data and also provides link to access insertion and updation page of sales detail and also at left bottom of the page it gives an option for delet

CHAPTER 6
CONCLUSION

In conclusion, database management systems (DBMS) play a crucial role in various sectors such as education, finance, telecommunications, and more, by efficiently organizing and managing data. MySQL, as a relational database management system, offers a robust solution for storing and retrieving data, supporting modern websites and web-based services. With its comprehensive command syntax and functionality, MySQL facilitates the implementation of projects like the one outlined here, which aims to streamline processes and enhance user experiences. So, the development of this Animal Rescue Management System is great improvement over the manual system which uses lots of manual work and paper.

The computerization n of the system speeds up the process and it is fast, efficient and reliable, avoids data redundancy and inconsistency. It contains all the functional features described in objective of the project.

REFERENCES

[1] Fundamentals of Database Systems, Ramez Elmasri and Shamkant B.Navathe,7th Edition, 2017, Pearson.

[2] Database management systems, Ramakrishnan Gehrke, 3rd Edition, 2014, McGraw Hill.