

## Chapter 1

### Preamble

#### 1.1 Introduction

A database is an organised collection of data. A relational database, more restrictively, is a collection of schemas, tables, queries, reports, views and other elements. A database management system (DBMS) is a computer-software application that interacts with end users, other applications, and the database itself to capture and analyze data. A general purpose DBMS allows the definition, creation, querying, update and administration of databases. There is a need for an application to make it easy for user and the organisation to effectively automate the existing manual system by the help of computerized equipment and full-fledged computer software. A PET SHOP MANAGEMENT SYSTEM is a smart approach towards this.

##### 1.1.1 Database Management System (DBMS)

Following the technology progress in the area of processors, computer memory, computer storage, and computer networks, the sizes, capabilities and performance of databases and their respective DBMSs have grown in order of magnitude. The development of database technology can be divided into three eras based on data model or structure: navigational, SQL/relational, and post-relational. The two main early navigational data models were the hierarchical model, epitomized by IBM's IMS system and CODASYL model (network model), implemented in a number of products such as IDMS.

The relational model employs set of ledger-style tables, each used for a different type of entity. Only in the mid-1980s did computing hardware become powerful enough to allow the wide deployment of relational systems (DBMSs plus applications). By the early 1990s, however, relational systems dominated in all large-scale data processing applications, and as of 2015 they remain dominant: IBM DB2, Oracle, MySQL, and

Microsoft SQL Server the top DBMS. The dominant database language, standardized SQL for the relational model, has influenced database languages for other data models.

### 1.1.2 MySQL

MySQL is an open-end relational database management system (RDBMS). MySQL is written in C and C++. Its SQL parser is written in YACC, but it uses a home-brewed lexical analyser. MySQL works on many system platforms, including Linux, macOS, Microsoft Windows, NetBSD. MySQL Community Server and the proprietary Enterprise Server. MySQL Enterprise Server is differentiated by a series of proprietary extensions which install as server plugins, but otherwise shares the version numbering system and is built from the same code base.\

Major features that are available in MySQL are a broad subset of ANSI SQL 99, as well as extensions, cross-platform support, stored procedures, using a procedural language that closely adheres to SQL/PSM, triggers, cursors, update views, online DDL when using the InnoDB Storage Engine. Many programming languages with language-specific APIs include libraries for accessing MySQL databases. These including MySQL Connector/NET for integration with Microsoft's Visual Studio and the JDBC driver for Java. In addition, an ODBC interface called MySQL Connector/ODBC allows additional programming languages that support the ODBC interface to communicate with a MySQL database, such as ASP or ColdFusion.

### 1.1.3 PHP

PHP: Hypertext Preprocessor is a server-side scripting language designed for Web development, and also used as a general-purpose programming language.

PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platforms, free of charge.

### 1.1.4 Normalization

Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.

There are three main types of normal forms:

- First normal form(1NF)
- Second normal form(2NF)
- Third normal form(3NF)
- Boyce and Codd normal form(BCNF)

#### 1. First normal form (1NF)

- a) As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values.
- b) It should hold only atomic values.

#### 2. Second normal form (2NF)

A table is said to be in 2NF if both the following conditions hold:

- a) Table is in 1NF (First normal form)
- b) No non-prime attribute is dependent on the proper subset of any candidate key of table.
- c) An attribute that is not part of any candidate key is known as non-prime attribute

#### 3. Third Normal form (3NF)

A table design is said to be in 3NF if both the following conditions hold:

- a) Table must be in 2NF
- b) Transitive functional dependency of non-prime attribute on any super key should be removed.
- c) An attribute that is not part of any candidate key is known as non-prime attribute.

In other words 3NF can be explained like this: A table is in 3NF if it is in 2NF and for each functional dependency  $X \rightarrow Y$  at least one of the following conditions hold:

X is a super key of table

Y is a prime attribute of table

An attribute that is a part of one of the candidate keys is known as prime attribute.

The relations are already in the normalized form in the schema diagram without any redundancy.

#### **4. Boyce and Codd normal form**

Boyce and Codd normal form (BCNF) is the advance version of 3NF that's why it is also referred as 3.5NF. A table complies with BCNF if it is in 3NF and for every functional dependency  $X \rightarrow Y$ , X should be the super key of the table.

### 1.1.5 Proposed System

The title of the project is “Pet Shop Management System”. This is a website project developed using language: PHP script, HTML and back-end: MySQL. Xampp server 3.2.4 is also used in this project.

Nowadays selling pets and their products has been an emerging business, so is the need for a Pet shop management system. The proposed system keeps record of all the pets that include pet animals and birds. It also keeps record of all their related products required for their care and welfare. The project has an interactive interface to enter and update records. Customer records and sales details are also maintained efficiently in this project making this project faster and centralised and reducing paperwork.

All the procedure in the proposed system is done online. So the project is computerized and easier as compared to existing manual systems.

### Existing system

The existing system is a manual one, requiring much time with paper and ink. It is a very tedious task for any shop owner to maintain records of each pet and products by writing them down in a sheet. This method proves to be uneconomical and errors in data entry may occur as well.

The case is not so with the proposed online system. This system makes it very easier for Owners to maintain records in a centralised way. Customers can view respective pets and their related products by visiting the website.

### 1.2 Objectives

The objective of the project is to provide web based interface to a petshop owner to manages his petshop activities.

To provide an option for storing and managing the basic information about pets and pet products in the shop.

To provide an option for storing and managing the sales details of the shop.

To provide an option for storing and managing the basic information about the customer

To track the information about sold pets and products to a customer.

### 1.3 Organisation of the report

Chapter 1 is the brief introduction about the backend software SQL, front end software HTML and the project report outline details are specified in Chapter 2 is the basic software requirements and hardware requirements needed to do this project are mentioned. Chapter 3 gives idea for the implementation parts for developing the project giving the SQL code for triggers and stored procedures. Chapter 4 is the results with the snapshots for the various operations are displayed with the snapshots. Chapter 5 concludes by giving the direction for future enhancement.

### 1.4 Summary

The chapter discussed before is an overview about the DBMS history, MySQL and PHP. The scope of study and objectives of the project are also mentioned clearly. The organisation of the report is been pictured to increase the readability. Further coming up chapters depicts the use of various queries to implement various changes like insert, delete and also triggers and stored procedures to perform various functions.

## **Chapter 2**

### **Requirement Specifications**

#### **2.1 SOFTWARE REQUIREMENTS**

Operating System : 64bit operating system, x64-based processor

Database : MYSQL

Tools : PHP, Xampp Server 3.2.4

#### **2.2 HARDWARE REQUIREMENTS**

Processor : Intel® Celeron® CPU N3060 @1.60GHz

RAM : 4.00 GB

Hard Disk : 1 TB

Input device : Keyboard(104 keys standard),Mouse(Optical)

Output device : Monitor screen

#### **2.3 FUNCTIONAL REQUIREMENTS**

- Pet shop management system should have collection of pets and their products.
- Each pet should belong to a particular category.
- Customer based on his preferences can view the pets and buy products.
- Billing is generated for each products.



## **2.4 USER CHARACTERSTICS**

Every user:

- Should be comfortable with basic knowledge of computer.
- Must have basic knowledge of English
- Must carry a valid user name and password for authentication.

## Chapter 3

### System Design and Implementation

#### 3.1 Introduction

System design is the art of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development.

In software engineering an ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model, that defines a data or information structure which can be implemented in a database. Some ER models show super and subtype entities connected by generalization-specialisation relationships, and an ER model can be used also in the specification of domain-specific ontologies.

A schema is a pictorial representation of relationship between the database tables in the database that is created. The database schema of a database system is its structure described in a formal language supported by the database management systems (DBMS). A database can be considered a structure in realization of the database language. The states of a created conceptual schema are transformed into an explicit mapping, the database schema. This describes how real-world entities are modelled in the database.

This project is implemented using PHP which is a very efficient language in the field of web development. It is done in windows platform. PHP and HTML is used to implement the entire code. Interface to the program is provided with the help of MySQL database.

### 3.2 ER Diagram

An entity-relationship model (ER model) describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.

ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure that can be implemented in a database, typically a relational database.

Fig no: 3.1 is the ER diagram of “Petshop Management System” with entities pets, animals, birds, pet products, sales details, customer, sold pets.

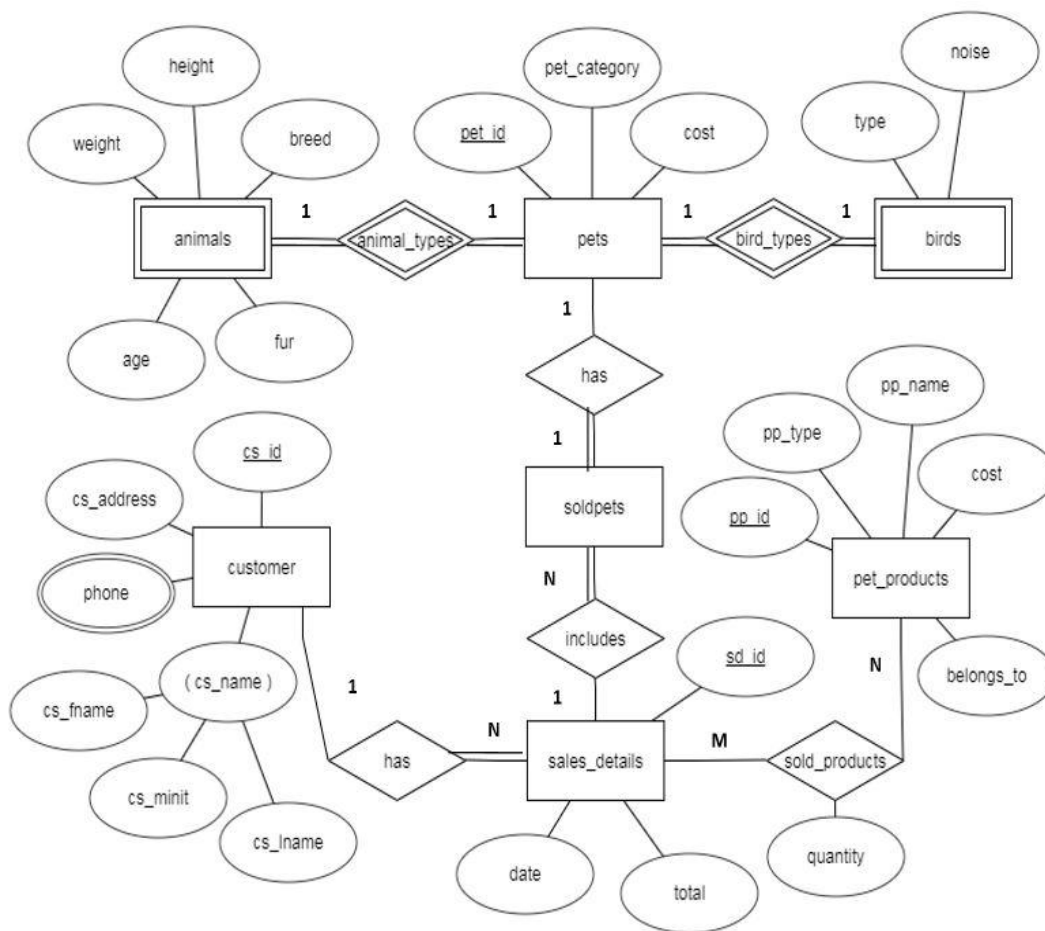


Fig no: 3.1 ER diagram

### 3.3 Schema Diagram

The Schema Diagram of a database described in a formal language supported by the database management systems (DBMS). The formal definition of a database schema is set of formulas called integrity constraints imposed on a database.

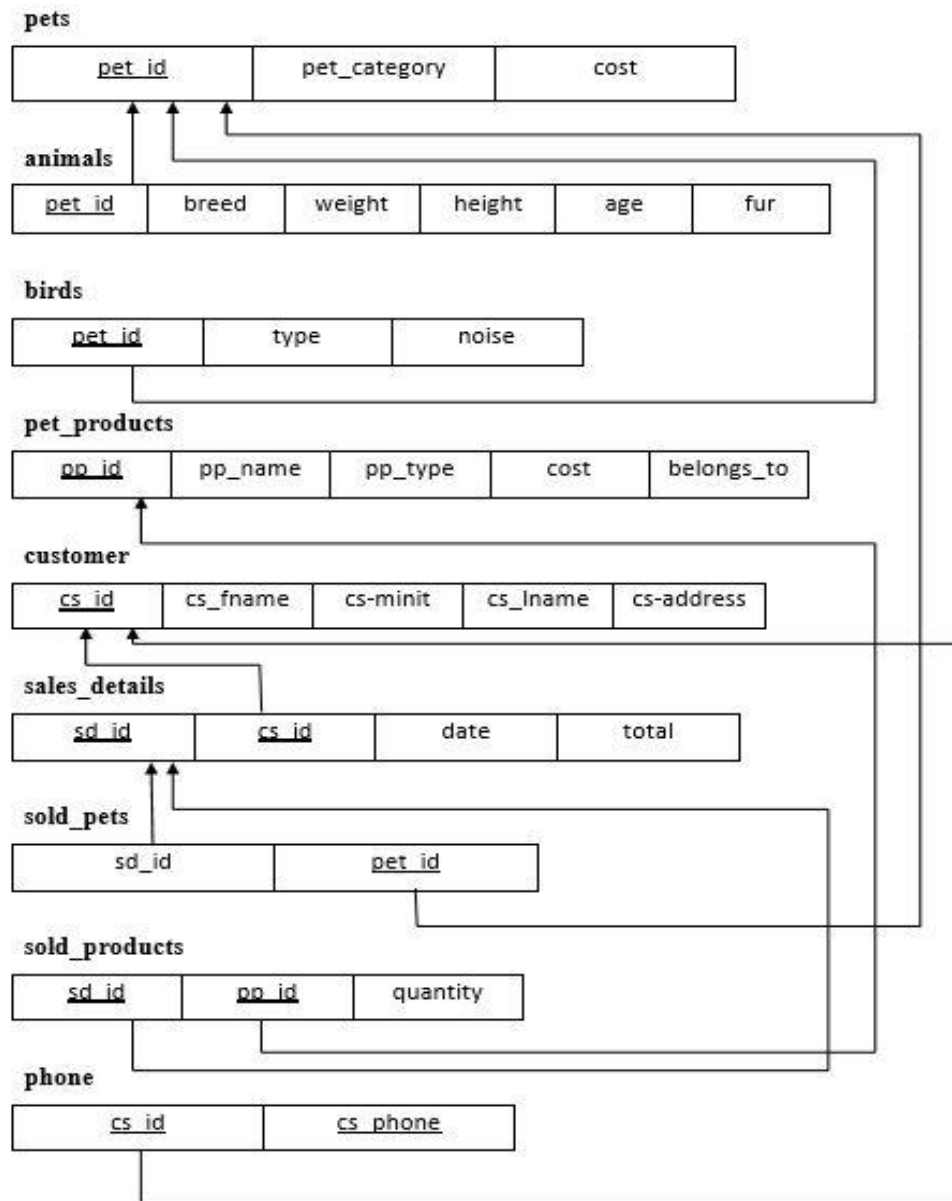


Fig 3.2 Schema Diagram

### 3.4 Queries

The below mentioned are all the queries used to perform various tasks in MySQL such as insert, delete, update. A short description is also provided.

#### 1. CREATION OF PETS TABLE

**Query:** create table pets( pet\_id varchar(9) not null,  
pet\_category varchar(15) not null,  
cost int(11) not null,  
primary key(pet\_id));

**Description:** This query is used to create a table called 'pets' which will store all pets information, the table is identified with the primary key 'pet\_id'.

#### 2 .CREATION OF ANIMALS TABLE

**Query:** 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),  
foreign key(pet\_id) references pets(pet\_id) on delete cascade);

**Description:** This query is used to create a table called 'animals' which will store all animal information about a particular breed(to which breed it belongs), the table is identified with the primary key 'pet\_id'.

### 3. CREATION OF BIRDS TABLE

**Query:** create table birds(pet\_id varchar(9) not null,  
type varchar(25) not null,  
noise varchar(10) not null,  
primary key(pet\_id),  
foreign key(pet\_id) references pets(pet\_id) on delete cascade);

**Description:** This query is used to create a table called 'birds' which will store all birds information about a particular type (to which type it belongs), the table is identified with the primary key 'pet\_id'.

### 4. CREATION OF PET\_PRODUCTS TABLE

**Query:** 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));

**Description:** This query is used to create a table called 'pet\_products' which will store all pet products details having product's name, type and its type and which product belong to which pet, the table is identified with the primary key 'pp\_id'.

### 5. CREATION OF CUSTOMER TABLE

**Query:** 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));
```

**Description:** This query is used to create a table called 'customer' which will store all customer details having customer's name, customer id and address, the table is identified with the primary key 'cs\_id'.

### 6. CREATION OF PHONE TABLE

**Query:** 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);
```

**Description:** This query is used to create a table called 'phone' which will store all customer phone details having customer id and phone number, the table is identified with the primary key 'cs\_id' and 'cs\_phone'.

### 7. CREATION OF SALES\_DETAILS TABLE

**Query:** 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);
```

**Description:** This query is used to create a table called 'sales\_details' which will store all the sales details of the pet shop having a unique sales id, customer id and total the table is identified with the primary key 'cs\_id' and 'sd\_id'.

### 8. CREATION OF SOLD\_PETS TABLE

**Query:** 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);

**Description:** This query is used to create a table called 'sold\_pets' which will store all the details of the sold pets having details such as sd\_id, pet\_id, the table is identified with the primary key 'pet\_id'.

### 9. CREATION OF SOLD\_PRODUCTS TABLE

**Query:** create table sold\_products(sd\_id varchar(9) not null,  
pp\_id varchar(9) not null,  
quantity int(11) not null,  
primary key(sd\_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 );

**Description:** This query is used to create a table called 'sold\_products' which will store all details of all sold products having sd\_id, pp\_id, quantity, the table is identified with the primary keys 'sd\_id' and 'pp\_id'.



### 1. INSERTION OF PETS TABLE

**Query:** INSERT INTO 'pets' (`pet\_id`, `pet\_category`, `cost`) VALUES ('pa01', 'dog', '8000'), ('pa02', 'cat', '3000'), ('pa03', 'dog', '8500'), ('pa04', 'dog', '15000'), ('pa05', 'cat', '3500');

**Description:** This query is used to insert into table 'pets'. All the values to be inserted in a chronological order and are inserted into the table according to the order.

### 2. INSERTION OF ANIMALS TABLE

**Query:** 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', 'golden retriever', '12.5', '40', '2', 'golden'), ('pa04', 'boxer', '11.5', '45', '3', 'black'), ('pa05', 'rag doll', '2.6', '20', '5', 'white');

**Description:** This query is used to insert into table 'animals'. All the values to be inserted in a chronological order and are inserted into the table according to the order.

### 3. INSERTION OF BIRDS TABLE

**Query:** 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');

**Description:** This query is used to insert into table 'birds'. All the values to be inserted in a chronological order and are inserted into the table according to the order.

### 4. INSERTION OF PET\_PRODUCTS TABLE

**Query:** INSERT INTO `pet\_products`(`pp\_id`, `pp\_name`, `pp\_type`, `cost`, `belongs\_to`) VALUES ('pp01', 'dog collar', 'accessories', '500', 'dog'), ('pp02', 'chain', 'accessories', '100', 'cat'), ('pp03', 'pedigree', 'food', '1500', 'dog'), ('pp04', 'mouth mask', 'accessories', '250', 'dog'), ('pp05', 'food bowl', 'accessories', '250', 'dog');

**Description:** This query is used to insert into table 'pet\_products'. All the values to be inserted in a chronological order and are inserted into the table according to the order.

### 5. INSERTION OF CUSTOMER TABLE

**Query:** 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');

**Description:** This query is used to insert into table 'customer'. All the values to be inserted in a chronological order and are inserted into the table according to the order.

### 6. INSERTION OF PHONE TABLE

**Query:** INSERT INTO `phone`(`cs\_id`, `cs\_phone`) VALUES('cs01', '8867762336'),('cs01', '9902587276'),('cs03', '9845034784'),('cs04', '6361261639'),('cs05', '86660873855');

**Description:** This query is used to insert into table 'phone'. All the values to be inserted in a chronological order and are inserted into the table according to the order.

### 7. INSERTION OF SALES\_DETAILS TABLE

**Query:** 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');

**Description:** This query is used to insert into table 'sales\_details'. All the values to be inserted in a chronological order and are inserted into the table according to the order.

### 8. INSERTION OF SOLD\_PETS TABLE

**Query:** INSERT INTO `sold\_pets` (`sd\_id`, `pet\_id`) VALUES('sd01', 'pa01'),('sd02', 'pa02'),('sd05', 'pa03'),('sd06', 'pb02'),('sd06', 'pb04');

**Description:** This query is used to insert into table 'sold\_pets'. All the values to be inserted in a chronological order and are inserted into the table according to the order.

### 9. INSERTION OF SOLD\_PRODUCTS TABLE

**Query:** 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')

**Description:** This query is used to insert into table 'sold\_products'. All the values to be inserted in a chronological order and are inserted into the table according to the order.

### 3.4.1 Stored Procedures

A stored procedure is a set of structured query language(SQL) statements with an assigned name, which are stored in a relational database management system as a group, so it can be reused and shared by multiple programs. Stored procedures can access or modify data in a database.

- **Procedure:**

```
create procedure calculations_for_pets(in pid varchar(9),in sid varchar(9))

BEGIN

DECLARE

cpid ,csid int DEFAULT 0;

set cpid=(select cost from pets where pet_id=pid);

set csid=(select total from sales_details where sd_id=sid); set csid=csid+cpid;

update sales_details set total=csid where sd_id=sid;

END
```

**Description:** calculations\_for\_pets : it calculates the cost of pet sold to a particular sale and updates that in sales\_details entity by adding the cost with the old total value of that sale.

- **Procedure:**

```
create procedure calculations_for_product(in ppid varchar(9),in sid varchar(9),in qnty int(11))

BEGIN

DECLARE

cppid ,csid int DEFAULT 0;

set cppid=(select cost from pet_products where pp_id=ppid); set csid=(select total from sales_details where sd_id=sid); set csid=csid+qnty*cppid;

update sales_details set total=csid where sd_id=sid; END
```

**Description:** calculations\_for\_product: it calculates the cost of product sold to a particular sale and updates that in sales\_details entity by adding the cost with the old total value of that sale.

### 3.4.1 Triggers

Triggers are stored procedures which are automatically executed or fired when some events occur. Triggers are stored into database and invoked repeatedly, when specific condition match.

- **Trigger:**

create or replace trigger check\_sold

before update on pets

for each row

BEGIN DECLARE checking int;

set checking=(select count(\*) from sold\_pets where pet\_id=old.pet\_id); if (checking > 0) then

    signal sqlstate '45000' set message\_text = 'cannot update sold pet'; end if;

END

**Description:** Here in this database, trigger avoids the updation of sold pet values in pet entity.

### 3.5 Pseudocode

Pseudocode is an informal high-level description of the operating system principle of a computer program or other algorithm. It uses the structural conventions of a normal programming language, but is intended for human reading rather than machine reading.

#### 3.5.1 Algorithm for login

**Step 1: BEGIN**

**Step 2:** Enter username and password.

**Step 3:** verify the credentials entered with that in the login table.

**Step 4:** if credentials match then proceed to the webpage else show login  
failed.

**Step 5: End if**

**Step 6: END**

#### 3.5.2 Algorithm for table display

**Step 1: BEGIN**

**Step 2:** Establish connection with the database using the username and password.

**Step 3:** Define ArrayList to return all the values from method passed.

**Step 4:** Define the select query to retrieve all the values from the DBMS.

**Step 5:** pass all the values into the ArrayList method.

**Step 6:** define DefaultTableModel for the table and use the ArrayList created to  
display all the values stored the table.

**Step 7: End if**

**Step 8: END**

### 3.5.3 Algorithm for Insert

**Step 1: BEGIN**

**Step 2:** Get all necessary values required for insertion.

**Step 3:** Define the query for insertion.

**Step 4:** Execute the query.

**Step 5: END**

### 3.5.4 Algorithm for update

**Step 1: BEGIN**

**Step 2:** Get all necessary values required for the values to be updated.

**Step 3:** Define the query for updating.

**Step 4:** Execute the query.

**Step 5: END**

### 3.5.5 Algorithm for delete

**Step 1: BEGIN**

**Step 2:** Get the pet details which is to be deleted.

**Step 3:** Define the query for deleting.

**Step 4:** Execute the query.

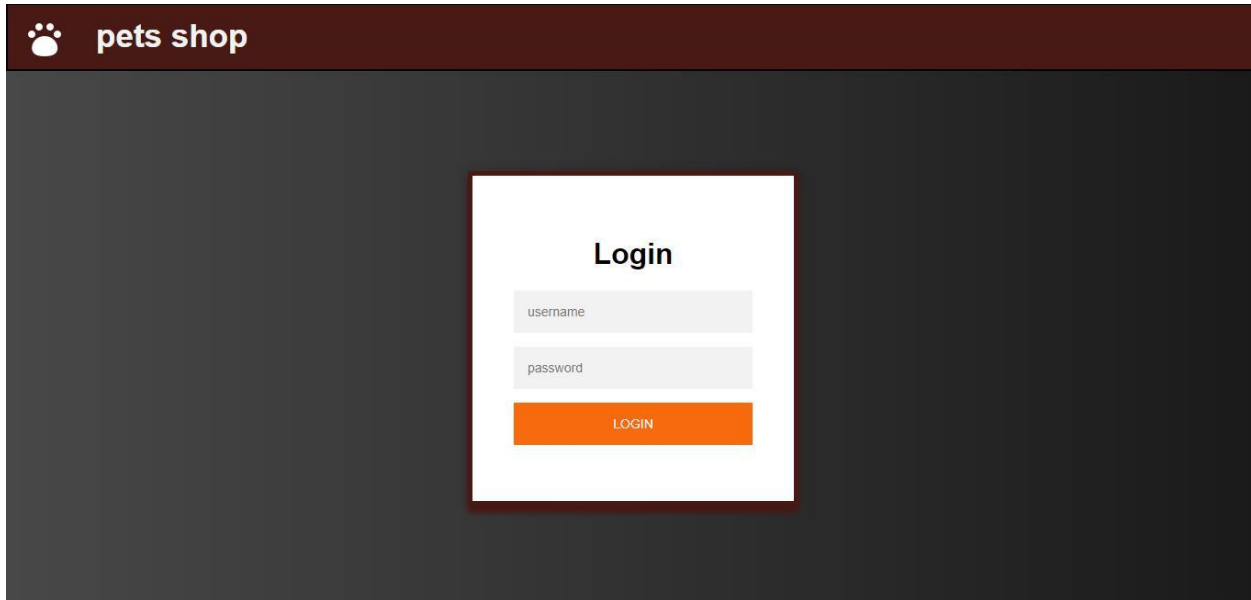
**Step 5: END**

## CHAPTER 4

### RESULTS AND DISCUSSIONS

This section describes the screens of the “PETHOP MANAGEMENT SYSTEM”. The snapshots are shown below for each module.

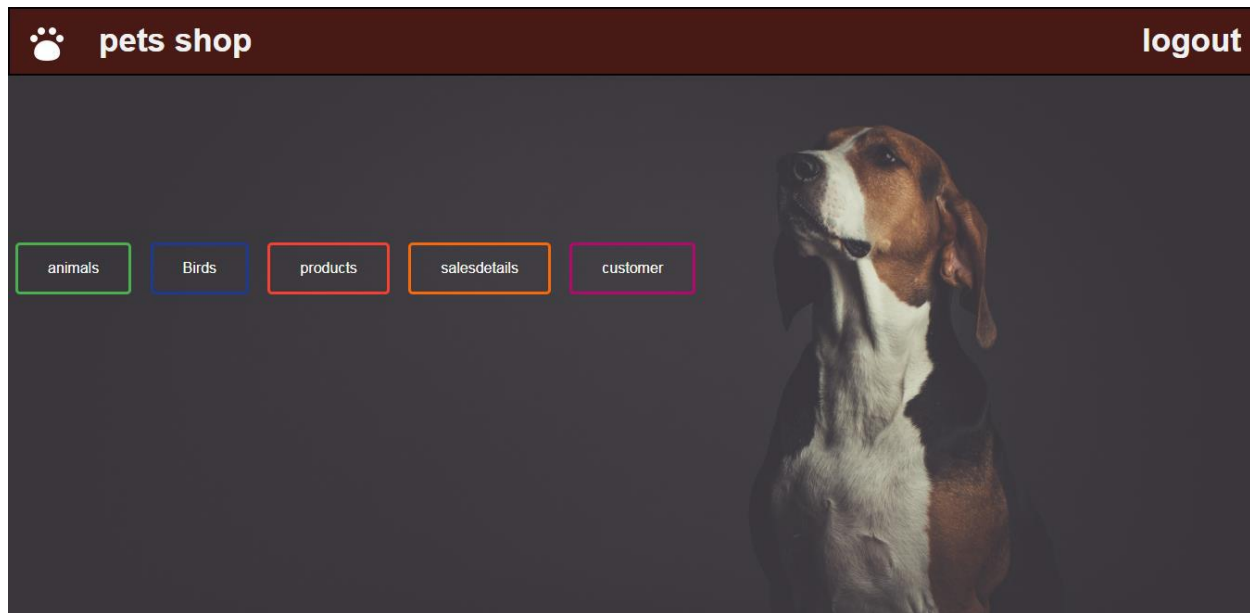
#### 4.1 SNAPSHOTS



**Snapshots 4.1.1: Login page**

This page asks admin username and password for authentication, if the authentication is successful then it loads home page.






**Snapshots 4.1.2: Home page**

This page provides links to animals page, birds page, products page, salesdetails page and customer page.

| pet_ID | petcategory | breed            | weight(kg) | height(cm) | age(m) | fur             | cost(Rs) |
|--------|-------------|------------------|------------|------------|--------|-----------------|----------|
| pa01   | dog         | labrador         | 11.3       | 30         | 2      | white           | 8000     |
| pa02   | cat         | parsian          | 3.6        | 20         | 2      | white           | 3000     |
| pa03   | dog         | golden retriever | 12.5       | 40         | 2      | gloden          | 8500     |
| pa04   | dog         | boxer            | 11.5       | 45         | 3      | black           | 15000    |
| pa05   | cat         | rag doll         | 2.6        | 20         | 5      | white           | 3500     |
| pa06   | dog         | st bernard       | 10.8       | 35         | 3      | brownish yellow | 10500    |
| pa07   | dog         | bulldog          | 8          | 25         | 3      | white           | 12000    |
| pa08   | dog         | german shepard   | 12         | 42         | 4      | black           | 10000    |

**Snapshots 4.1.3: Animals page**


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.

 **Animals** logout

[Back](#)

**Snapshots 4.1.4: Animal insertion page**

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

 **Birds** logout

[Add new bird](#) [update bird](#)

| pet_ID | petcategory | type           | noise    | cost  |
|--------|-------------|----------------|----------|-------|
| pb01   | parrot      | grey parrot    | moderate | 2000  |
| pb02   | lovebirds   | black cheeked  | low      | 800   |
| pb03   | lovebirds   | grey headed    | moderate | 600   |
| pb04   | lovebirds   | lilian         | moderate | 800   |
| pb05   | cockatoo    | white cockatoo | moderate | 10000 |

**Snapshots 4.1.5: Birds page**

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.

## PET SHOP MANAGEMENT SYSTEM

 **pets products** logout

[Add new product](#) [update product](#)

| pp_ID | pp_name    | pp_type     | cost | belongs_to |
|-------|------------|-------------|------|------------|
| pp01  | dog collar | accessories | 500  | dog        |
| pp02  | chain      | accessories | 100  | cat        |
| pp03  | pedigree   | food        | 1500 | dog        |
| pp04  | mouth mask | accessories | 250  | dog        |
| pp05  | food bowl  | accessories | 250  | dog        |
| pp06  | bird feeds | food        | 300  | birds      |

[delete](#)

### Snapshots 4.1.6: pet products page

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

 **Sales details** logout

[Add new details](#) [update details](#) [sold products](#) [sold pets](#)


| sd_ID | cs_id | date       | total |
|-------|-------|------------|-------|
| 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  |
| sd06  | cs05  | 2018-11-20 | 1600  |

[Delete](#)

### Snapshots 4.1.7: sales details page

This page displays sales details data and also provides link to access insertion and updation page of sales details and also at left bottom of the page it gives an option for deletion. It also provides link to access sold pet and sold products page.

## PET SHOP MANAGEMENT SYSTEM

 **Customers** logout

Add new customer

update customer

phone nos.


| cs_ID | cs_fname  | cs_minit  | cs_lname | cs_address |
|-------|-----------|-----------|----------|------------|
| 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  |

Enter the id to delete

Delete

### Snapshots 4.1.8: customers page

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

 **Animals** logout

Back

Enter pet\_id

Enter pet\_category

Enter breed

Enter weight

Enter height

Enter age

Enter fur

Enter cost

update

### Snapshots 4.1.9: Animal insertion page

This page accepts the data to update in animals entity and pet entity.

## **CHAPTER 5**

### **CONCLUSION AND FUTURE ENHANCEMENTS**

#### **5.1 CONCLUSION**

The development of this Pet shop Management System is great improvement over the manual system which uses lots of manual work and paper. The computerization of the system speeds up the process.

The Petshop Management System is fast, efficient and reliable, Avoids data redundancy and inconsistency. It contains all the functional features described in objective of the project.

#### **5.2 FUTURE ENHANCEMENTS**

The future scope of our project is vast and can be used in extensive ways:

- Inclusion of human carer of the pet and its details
- Care and welfare aim to provide individuals with the skills and knowledge needed to best provide for the animals in their care.
- Implement the backup mechanism for taking backup of codebase and database on regular basis.
- Chart and graph analysis of project.

## **REFERENCES**

[1] Elmasri, Ramez, Fundamentals of database systems. Pearson education in india 2008.

[2] <https://programmerblog.net/createmysal-trigger-php>

[3]<http://www.tutorialspoint.com>

[4] <http://www.w3schools.com>

[5]<https://stackoverflow.com>

.