CHAPTER-1

SYNOPSIS

1.1 TITLE OF THE PROJECT:

"PAWSPOT"

1.2 INTRODUCTION:

The main idea of this project is to provide a user friendly interface to automate the process of serving towards the welfare of the pets by giving the abandoned pets a place of shelter and care them with affection. The application also gives guidelines for caring towards the pets, adoption procedure of a pet. The user can do the adoption process through the application as this process is time consuming if done manually. The user can register themselves to the application and adopt a pet or purchase the products related to pets. For adoption, the user can choose the pets that are available, view their details and if they wish they can continue with the process of adoption. In case of purchase of products related to the pets, the user can choose the products they require from the application that are organized based on categories (i.e. food,combos,supplements etc). Payment of the purchased products is done by using by credit card, debit card or cash on delivery. The administrator of the application will be able to maintain the records of the adopters of the pets, purchasers of the products making the manual work easier. The administrator can also maintain the details of pets and products.

1.3 OBJECTIVE:

The main objective of this project is to automate the process of serving towards the welfare of the pets by giving the abandoned pets a place of shelter with care and affection.

1.4 PROJECT CATEGORY:

Web Application

1.5 TOOLS/PLATFORM, HARDWARE AND SOFTWARE REQUIREMENT SPECIFICATION:

1.5.1 HARDWARE REQUIREMENTS:

RAM : 4GB or above

Hard Disk : 20GB Hard disk or above

Operating System : Windows

1.5.2 SOFTWARE REQUIREMENTS:

Language :PHP

Front- end/GUI Tool : HTML,CSS,JQUERY

Back-end : MySQL

Server : XAMPP

1.6 MODULES LIST:

USER SIDE:

1. Register : This module helps the user to register through this website.
2. Login : This module allows the user to login after the registration is done.
3. Product details : This module helps the user to view products, add product to the cart, buy products and cancel the ordered products.
4. View order details : This module allows the user to view orders.
5. Payment : This module allows the user to make payment through credit/debit.
6. Search pet donors : This module allows the user to view the pet details and the contact information of the donor.
7. Choose to adopt : Here the user can send a message to the organization showing interest in adopting a pet.
8. Feedback : This module allows the user to provide a feedback on our service.
9. Logout : This module allows the user to logout.

PET DONOR SIDE:

1. Register : This module helps the pet donor to register to this website.
2. Login : This module allows the pet donor to login after the registration is done.
3. Add pet details : This is where the pet donor can add the details of the pet
4. View requests : This module shows the requests of the people interested in adopting.
5. Delete details : This is where the pet donor deletes the details once the adoption has been done.
6. Feedback : This module allows the user to provide a feedback on our service.
7. Logout : This module allows the pet donor to logout.

PET SHOP SIDE:

1. Register : This module helps the pet shop owner to register through this website.
2. Login : This module allows the pet shop owner to login after the request is been accepted by the admin.
3. Add product details : Here the pet shop owner can add their products.
4. Update products : This module is used by the pet shop owner to update product details and delete the products.
5. View orders : This module allows the pet shop owner to view orders placed by the user.
6. Report : Here the pet shop owner can view all the report of his sales..
7. Change password : This module helps the pet shop owner to change the password.
8. Feedback : This module allows the user to provide a feedback on our service.
9. Logout : This module allows the pet shop owner to logout.

ADMIN SIDE:

1. Login : This module allows the admin to login after the registration is done.
2. View user details : This module will display all the users.
3. View shop user details : This module will display all the shop users.
4. View pet donor details : This module will display all the pet donors.
5. View feedback : This module allows the admin to view all the feedbacks of the application users.
6. Logout : This module allows the admin to logout.

1.7 TYPE OF THE PROJECT:

User Defined Project.

1.8 LIMITATIONS OF THE PROJECT:

Limitation of the project is that only the static details is been discussed.

1.9 FUTURE SCOPES AND FURTHER ENHANCEMENT OF THE PROJECT:

The present system is being developed as a web application. Although the application can be accessed by portable devices like cell phones through browsers. In future, we would like to develop as an android application.

CHAPTER-2

SOFTWARE REQUIREMENT SPECIFICATION

2.1 Introduction

Software Requirement Specification (SRS) is a document, which describes completely the external behavior of the software. First and foremost the work of a software developer is to study the system to be developed and specify the user requirements before going for the designing phase. This document will let us know how this system behaves and responds.

2.1.1 Purpose

The main idea of this project is to provide a user friendly interface to automate the process of serving towards the welfare of the pets by giving the abandoned pets a place of shelter and care them with affection. The application also gives guidelines for caring towards the pets, adoption procedure of a pet. The user can do the adoption process through the application as this process is time consuming if done manually.The user can register themselves to the the application and adopt a pet or purchase the products related to pets. For adoption, the user can choose the pets that are available, view their details and if they wish they can continue with the process of adoption. In case of purchase of products related to the pets, the user can choose the products they require from the application the products are organized based on categories (i.e.food,combos,supplements etc). Payment of the purchased products is done by using by credit card, debit card or cash on delivery. The administrator of the application will be able to maintain the records of the adopters of the pets, purchasers of the products making the manual work easier. The administrator can also maintain the details of pets and products.

2.1.2 Scope:

This SRS describes the requirement of the system. It is meant for use of the developer that will be the basis for validating the final delivered system; any changes made to the requirement in future will have to go through the formal change approval process. This document contains a complete description of the final functioning of PawSpot.

- The system will provide the user to register as customer and look at the details of the pet adoption and pet products.

- The Admin posts the details of the pets available for adoption on the site.
- The users who are registered as customer can easily buy the pet products or adopt a pet at their convenience.

2.1.3 Definition, Acronyms and Abbreviations:

DFD : Data Flow Diagram.

E-R : Entity-Relationship Diagram.

SRS : Software Requirement Analysis and Specification.

SQL : Structured Query Language.

RAM : Random Access Memory.

XAMPP : Cross-Platform Apache MariaDB/MySQL PHP and Perl.

HTML : HyperTextMarkup Language.

CSS : Cascading Style Sheets

2.1.4 Overview:

This document completely describes what the proposed system should do without describing how the system will do it. This provides the purpose as well as the complete behaviour of the system. The document us started with an overall view about the proposed system and then analyzed all the features and functions of the same.

2.2 Overall Description

2.2.1 Product perspective

Product perspective is essentially the relationship of the product to another products defining if the product is independent or a part of large product. This web application is self contained and works relatively as efficient as other packages related to the subject. It provides simple database rather than complex ones for high requirements and it provides good and easy graphical user interface to both new, naive as well as experienced users of computers.

2.2.2 Product Function

All kinds of users should be able to use the system only after entering valid username and password.

- Four types of users are maintained in the system (i.e.User,Pet Donor,Pet Care Shop Owner and Admin).

The User can

- View product details.
- Add products to the cart.
- Buy the products.
- Cancel the products.

The Pet Donor can

- Add pet details.
- Delete details.
- View interested users.

The Pet Shop can

- Add product details.
- Delete product details.
- View orders.

The Admin can

- View user details.
- View shop user details.
- View pet donor details.

2.2.3 User Characteristics:

The system is designed with the intension to provide easy to use simple system so that no cumbersome and elaborate training for operation is needed.

- The user must have a basic knowledge about the computer.

- The user need not be a technical person i.e. no prior knowledge about programing is required.
- User should have the knowledge about the internet.
- User should be able to understand the graphical user interface.

2.2.4 General Constraints:

- Requires all the mandatory fields to be filled with proper information.
- Requires the detailed description of the cleaning events.
- The user should be familiar with primary usage of web application andinternet.
- The developed system can run on any platform (Linux, Windows or MAC) but only with a latest web browser installed.

2.2.5 Assumption and Dependencies:

- Internet connection required.
- The user should have basic mobile knowledge.
- User should be familiar with basic Computer knowledge.
- User should be familiar with usage of internet and web sites.
- The developed web application should run on any platform (UNIX, Linux, Mac, Windows etc) that contains a latest JavaScript enabled web browser.

2.3 Specific Requirements

This section describes all the details that the system developer needs to know for designing and developing this system. These requirements can be organized by the modes of operation, class objects, features and functional hierarchies.

2.3.1 External Interface Requirement

2.3.1.1 User Interface

The web application provides good graphical user interface for the front end of the system so that users can make use of the system with ease. A login page is provided for the users for

authentication. On successful authentication, the permission to use the web application is provided.

2.3.1.2 Hardware Interface

- Processor : Intel Pentium 4 or above / AMD A8 or above
- Ram : 4GB or more
- Hard Disk : 20GB free space or more

2.3.1.3 Software Interface

The following software tools are used in the development of the system:

- Apache Server
- MySQL for database
- PHP for backend design
- HTML, CSS and JS for front end

2.3.1.4 Communication Interface

The system works with help of internet connection. The application is linked to the database maintained in the server.

2.3.2 Functional Requirements

User Side:

Register : This module helps the user to register through this website.

Login : This module allows the user to login after the registration is done.

Product details : This module helps the user to view products, add products to the cart,buy the products and cancel the ordered products.

View order details : This module allows the user to view orders.

Payment : This module allows the user to make payment through credit/debit.

Search pet donors : This module allows the user to view the pet details and the contact information of the pet donors.

Choose to adopt : Here the user can send a message to the donor showing interest in adopting a pet.

Feedback	: This module allows the user to provide a feedback on our service.
Logout	: This module allows the user to logout.

Pet Donor Side:

Register	: This module helps the pet donor to register to this website.
Login	: This module allows the pet donor to login once the registration is done.
Add pet details	: This is where the pet donors can add the details of the pet.
View interested users	: This module shows the details and messages of the people interested in adopting.
Delete details	: This is where the pet donor deletes the details once the adoption is done.
Feedback	: This module allows the user to provide a feedback on our service.
Logout	: This module allows the pet donor to logout.

Pet Shop Side:

Register	: This module helps the pet shop owner to register through this website.
Login	: This module allows the pet shop owner to login after the request is been accepted by the owner.
Add product details	: Here the pet shop owners can add their products.
Update products	: This module is used by the pet shop owner to update product detailsand delete the products.
View orders	: This module allows the pet shop owner to view orders placed by the user.
Report	: Here the pet shop owner user can view all the report of his sales..
Change password	: This module helps the pet shop owner to change the password.
Feedback	: This module allows the user to provide a feedback on our service.
Logout	: This module allows the pet shop owner to logout.

Admin Side:

Login	: This module allows the admin to login after the registration is done.
View User details	: This module will display all the users.
View Shop User details	: This module will display all the shop users.
View Pet Donor details	: This module will display all the Organization users.
View Feedback	: This module allows the admin to view all the feedbacks of theapplication users.
Change password	: This module helps the admin to change the password.
Logout	: This module allows the admin to logout.

2.3.3 Nonfunctional Requirements

2.3.3.1 Performance Requirements

The system will run efficiently on any modern JavaScript enabled web browser. The server will require 2GB or more RAM, Intel Pentium 4 or above processor, 20GB free space in Hard Disk, a latest version of Debian Linux distro.

2.3.3.2 Safety Requirements

The system is safe to use. The data is stored in the central database and it is made sure that there is minimum data loss in case of any mishaps. In case the user forgets password, forgot password functionality helps to set a new password. Authorization helps to check the permission level of the user accessing the site.

2.3.3.3 Security Requirements

- Users shall be required to log in to the application for all operations for the first time
- This application will permit only the admin to modify the records.

2.3.3.4 Software Quality Attributes

If the connection between the user and the application is broken the application will alert the user about the connectivity issues.

- **RELIABILITY**

  It is tested for all the constraints at development stage

- **AVAILABILITY**

  This system will only available till the system on which it is installed is running.

- **SECURITY**

  This system is provided with authentication without which no user can pass. So only the legitimate users are allowed to use the application. If the legitimate users share the authentication information, then the system is open to outsiders.

- **MAINTAINABILITY**

  There will be no maintenance required for this application. The database is provided by the end-user and therefore is maintained by this user.

- **PORTABILITY**

  The system works anywhere with the internet connection.

- **USABILITY**

  The interface is user friendly; it is easy to access and understand.

CHAPTER-3

SYSTEM ANALYSIS AND DESIGN

3.1 Introduction:

System analysis is the application of the system approach to problem solving using computers. Analyst must consider its elements like outputs and input processors, control, feedback and environment when constructing the system.

The purpose of the design phase is to plan a solution of the problem specified by the requirement document. This phase is the step-1 moving from the problem domain to solution domain. It is also a bridge between requirements specifications and final solution for satisfying requirements.

The design activity is often divided into two separate process – System design and Detailed Design. System Design is also called as Top-Level design. It aims to identify the modules that should be in the system, specifications of these modules and how they interact with each other to produce the desired result. At the end of System Design all the major data structures, file formats, major modules in the system and their specifications are decided.

3.2 Logical Design:

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modeling, using an over-abstract (and sometimes graphical) representation of the actual system. In the context of system design, modeling can undertake the following parts, including:

- Data Flow Diagram
- Entity Relationship Diagram

3.3 Physical Design:

The physical design relates to the actual input and output processes of the system. This is laid down in terms of how data is input into a system, how it is verified.

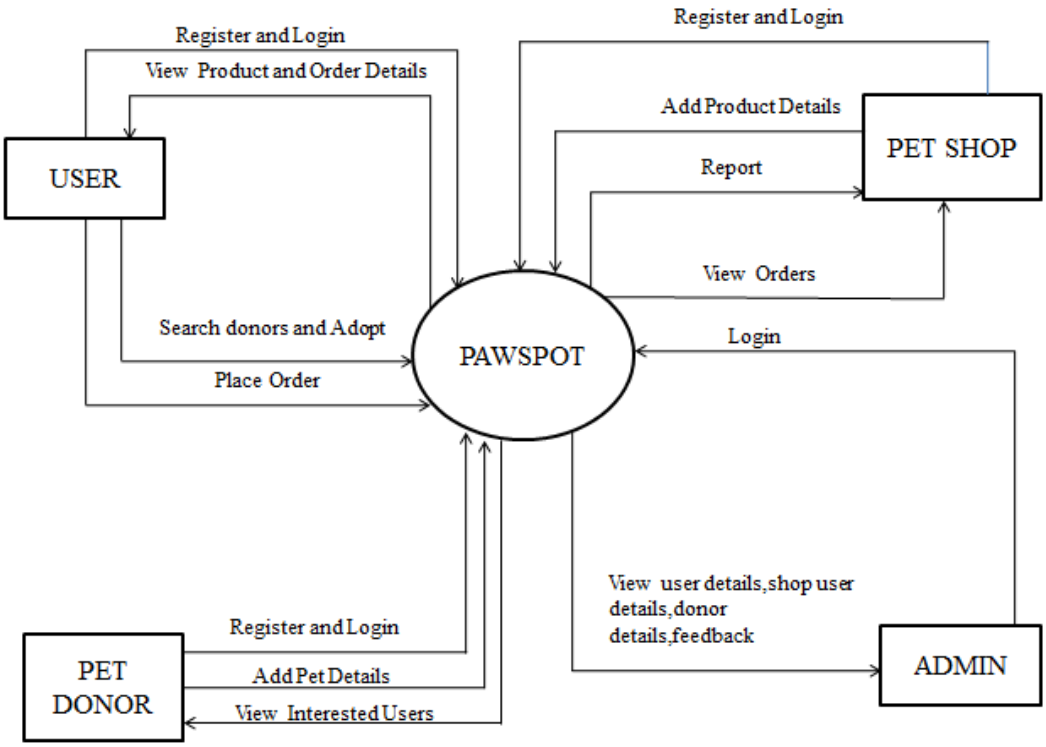
3.4 Applicable Document:

The applicable document to be referred here is Software Requirement Specification.

3.5 Description of Programs

3.5.1 Context Flow Diagram:

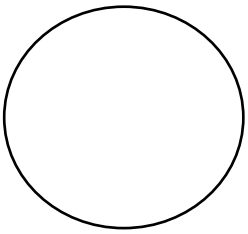
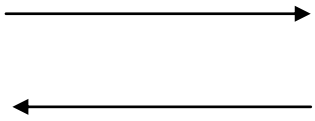
Context flow diagram is a top level data flow diagram. It only contains one process node that generalize the function of the entire system in relationship to external entities. In context diagram the entire system is treated as a single process and all its inputs, outputs, sinks and sources are identified and shown.



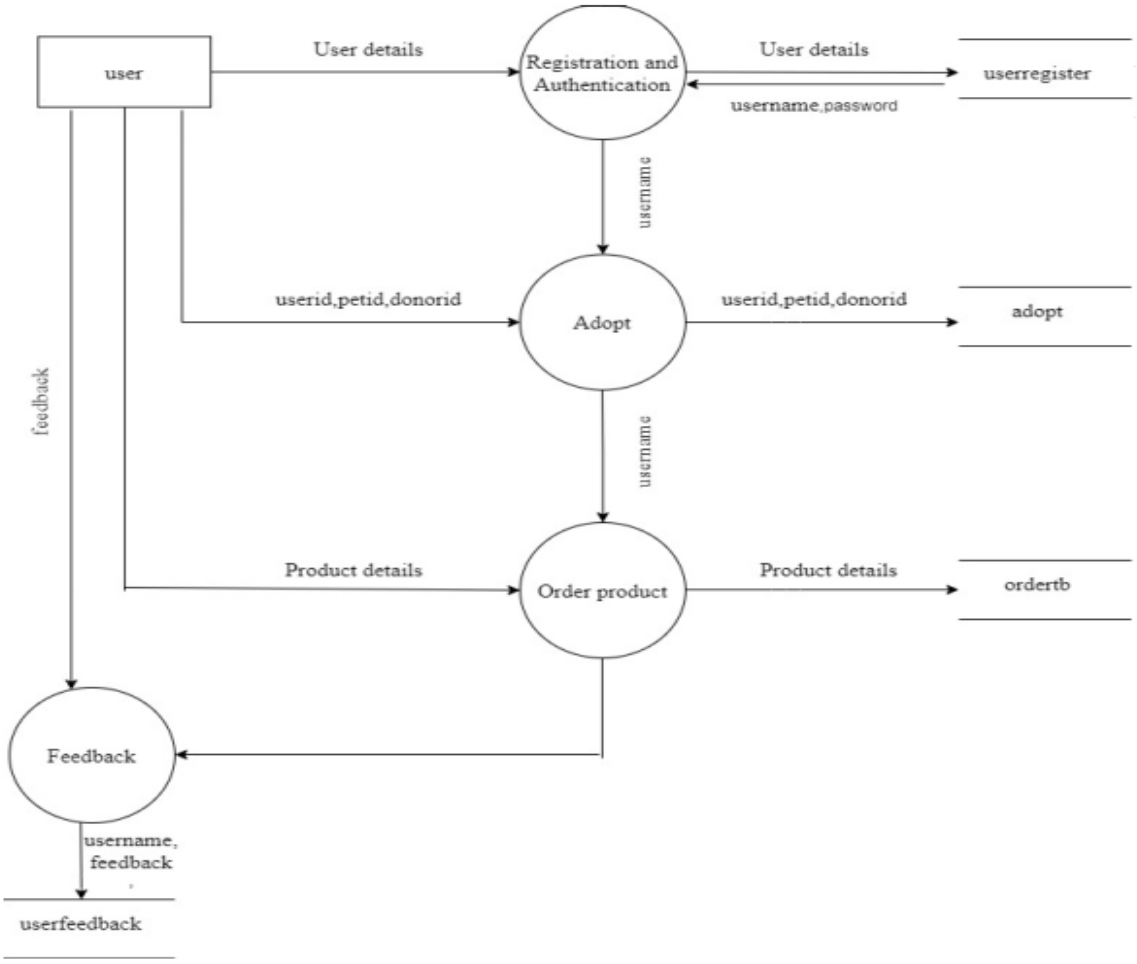
3.5.2 Data Flow Diagrams:

A DFD shows the flow of data through system. It views a system as a function that transforms the input into desired output. Data flow diagram is drawn to pictorially represent the data stored, processed, entities involved in the system and information flow. DFDs are useful in understanding a system and can be efficiently used during problem analysis.

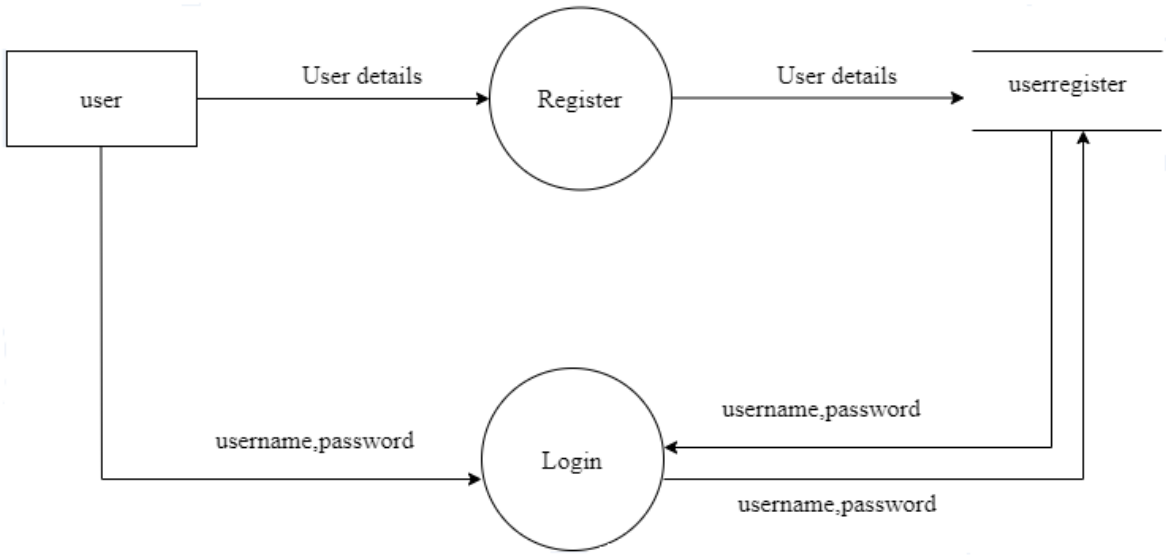
DFD Symbols:

Name	Notation	Description
Process		A process transforms incoming data flow into outgoing data flow. The processes are shown by named circles.
Datastore		Data stores are repositories of data in the system. They are sometimes also referred to as files.
Dataflow		Data flows are pipelines through which packets of information flow. Label the arrows with the name of the data that moves through it.
External Entity		External entities are objects outside the system with which the system communicates. External Entities are sources and destinations of the system's inputs and output.

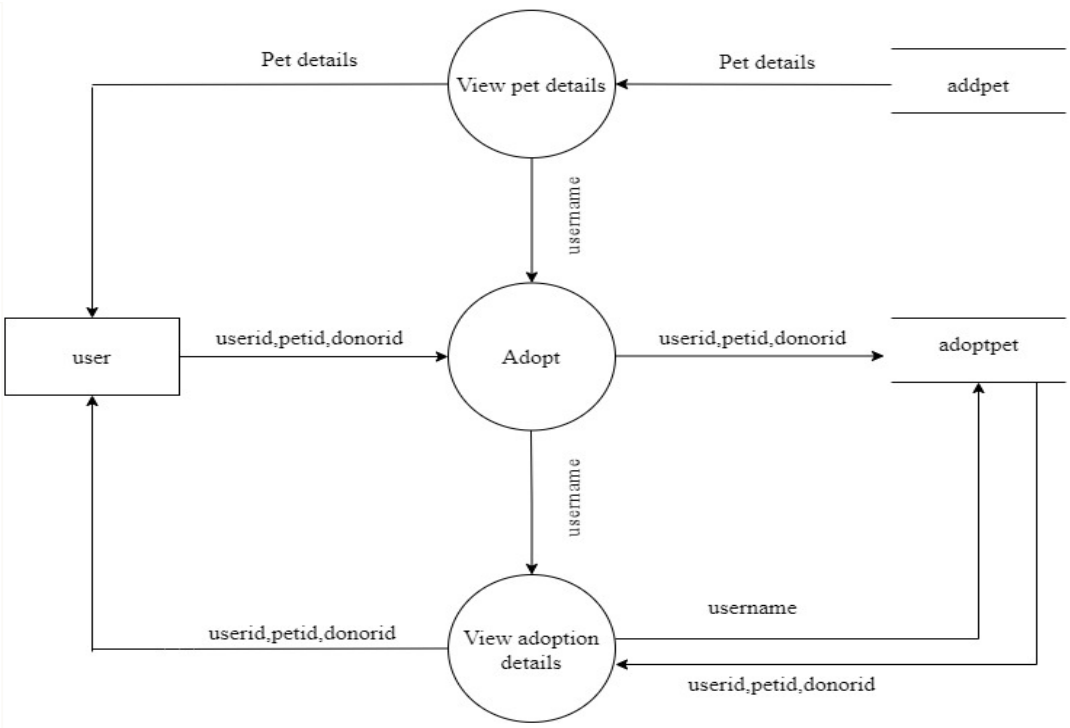
LEVEL 1 DFD USER



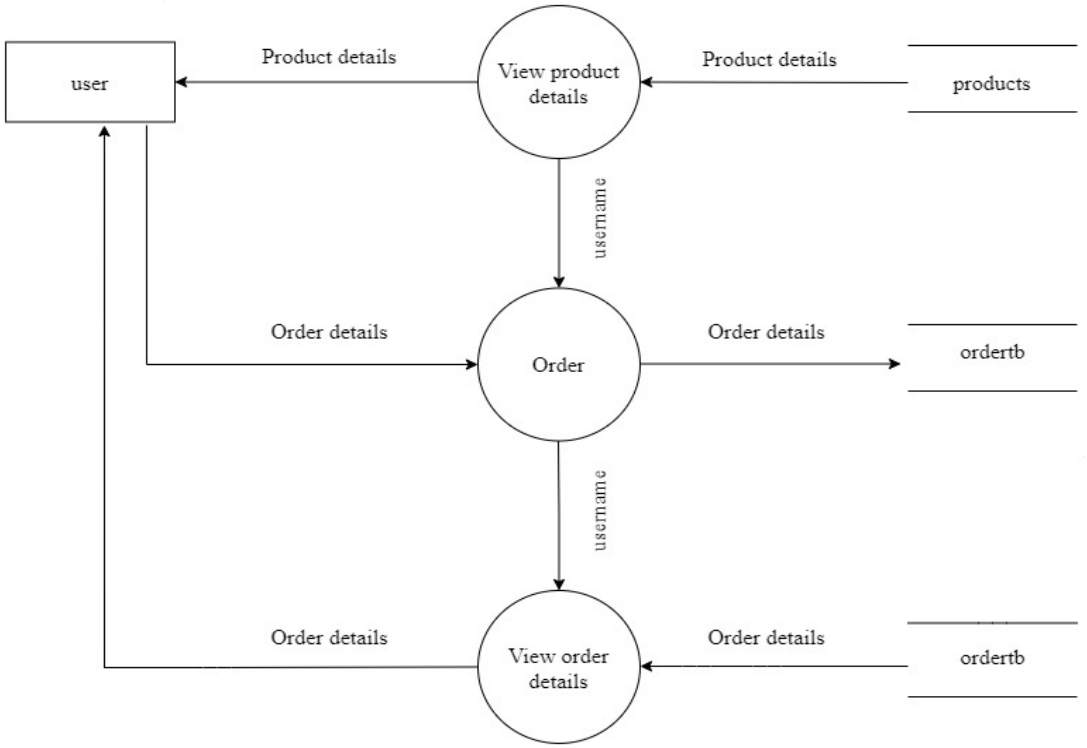
LEVEL 2 DFD REGISTRATION AND AUTHENTICATION



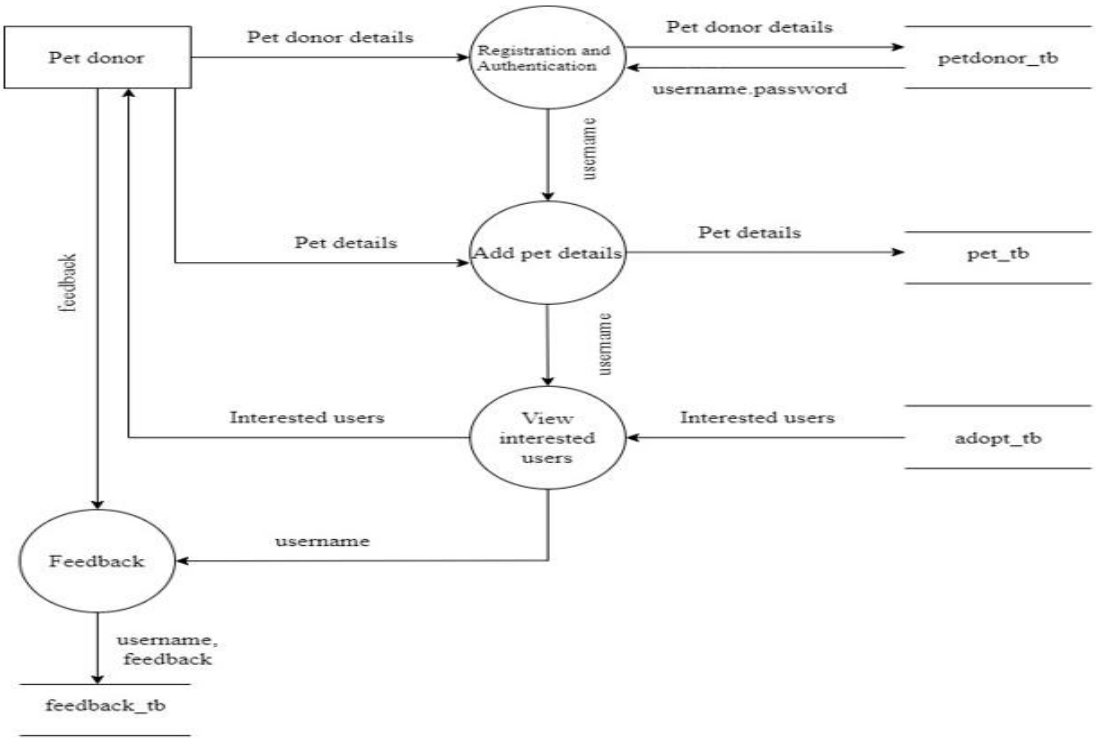
LEVEL 2 DFD ADOPT



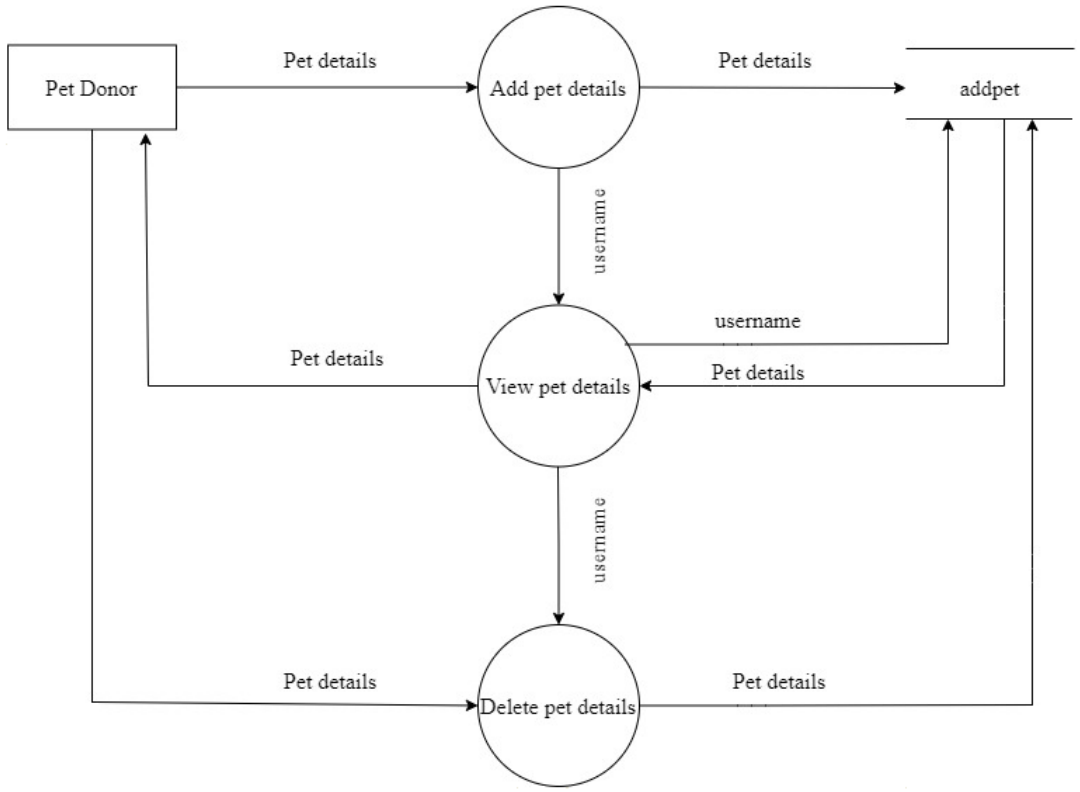
LEVEL 2 DFD ORDER PRODUCT



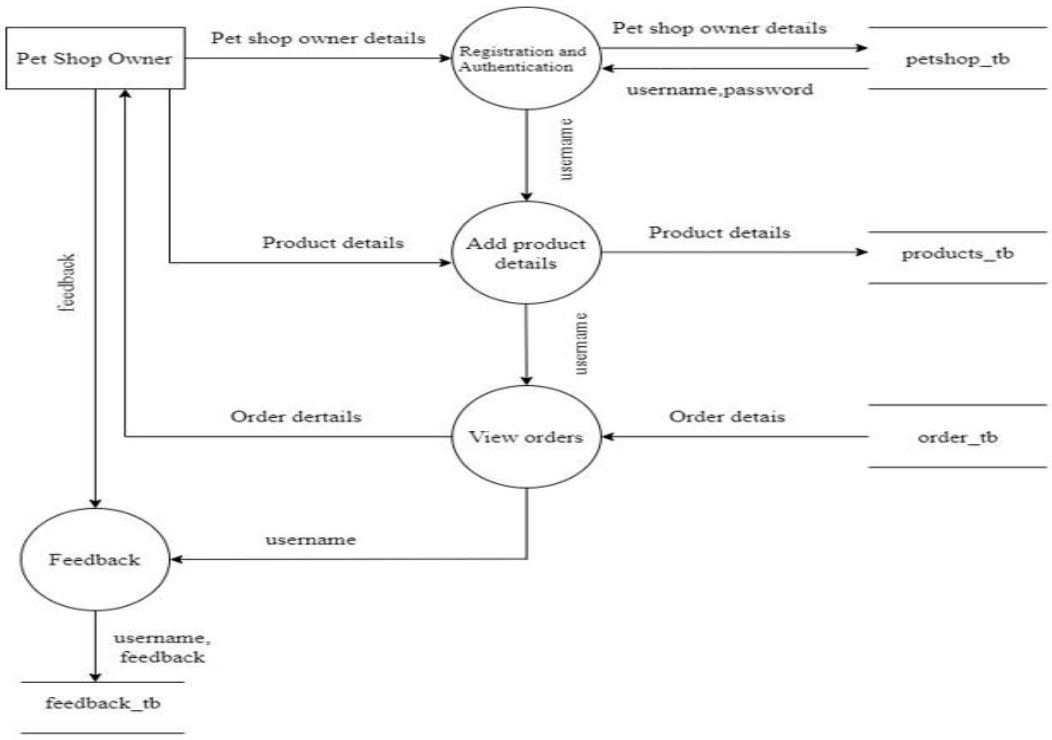
LEVEL 1 DFD PET DONOR



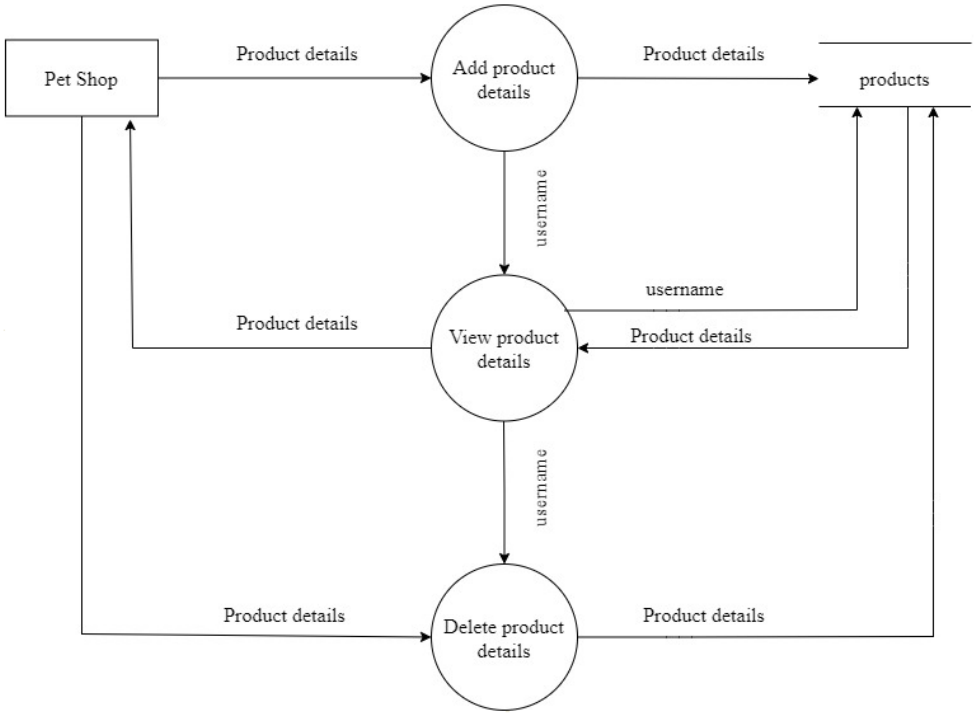
LEVEL 2 DFD ADD PET DETAILS



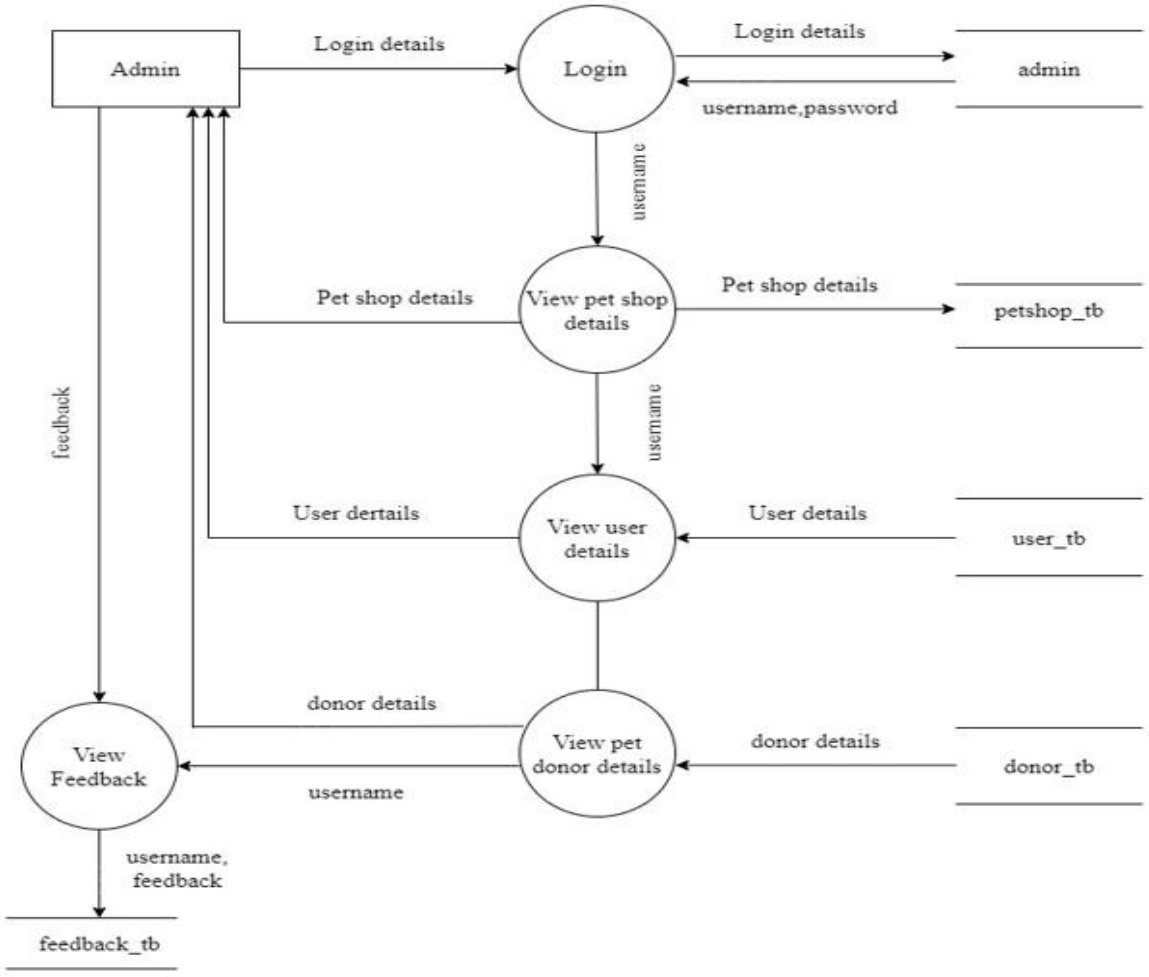
LEVEL 1 DFD PET SHOP



LEVEL 2 DFD PRODUCT DETAILS



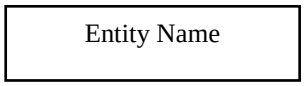
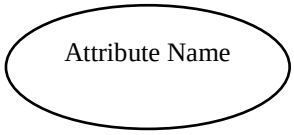
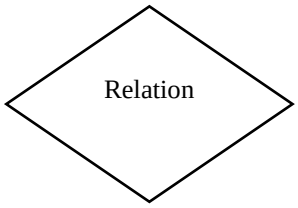
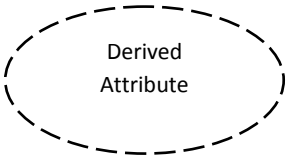
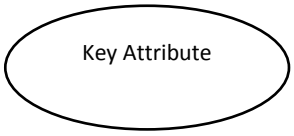
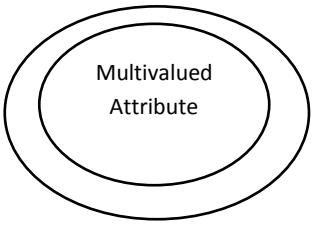
LEVEL 1 DFD ADMIN



3.4 Entity Relationship Diagram:

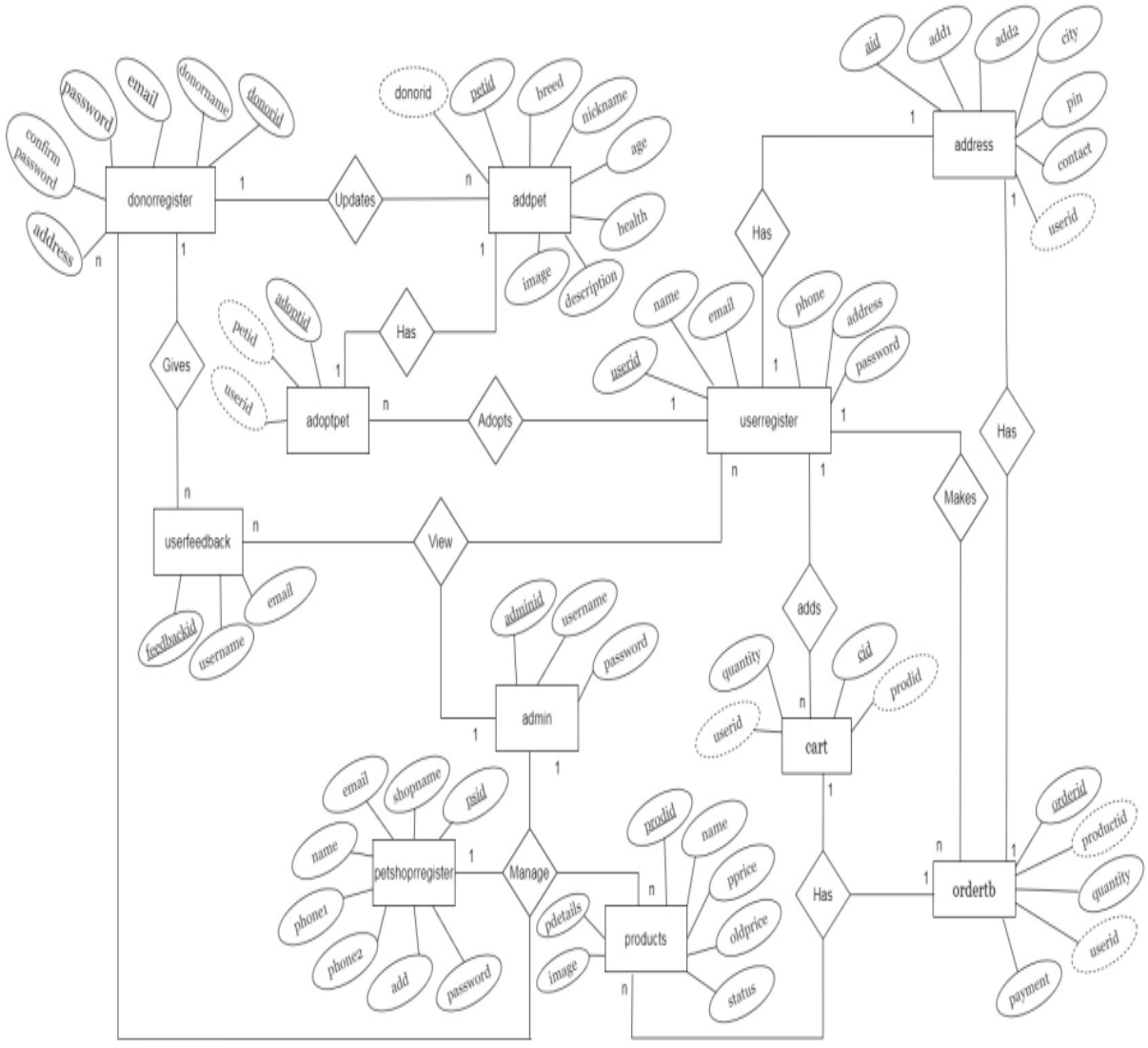
The basic objective of the ER model representation is an entity which is a "thing" in a real world with an independent existence. Entities are physical items or aggregations of data items that are important to the business we analyse or to the system; we intend to build. An entity represents an object defined within the information system about which you want to store information. Entities are named as singular nouns and are shown in rectangles in an ER Diagram.

ER-Diagram Symbols and Description:

Name	Notation	Description
Entity		It may be an object with the physical existence or conceptual existence. It is represented by a Rectangle.
Attribute		The properties of the entity can be an attribute. It is represented by an ellipse.
Relationship		Whenever an attribute of one entity refers to another entity, some relationship exists. It is represented by a diamond
Link		Lines links attributes to entity sets and entity sets to relation
Derived Attribute		Dashed ellipse denotes derived attribute
Key Attribute		An entity type usually has an attribute whose values are distinct for each individual entry in the entity set.it is denoted by an underlined word in an ellipse.
Multivalued Attribute		Attributes that have different numbers of values for a particular attribute. It is represented by a double ellipse.

Cardinality Ratio	1. 1:1 2. 1:M 3. M:1 4. M:M	It specifies the maximum number of relationships that an entity can participate in. There are four cardinality ratios.

ER Diagram:



CHAPTER-4

DETAILED DESIGN

4.1 Introduction:

Detailed design starts after the system design phase and the system has been certified through the review. The goal of this phase is to develop the internal logic of each of the modules identified during the system design.

In the system design, the focus is on identifying the modules, whereas during detailed design the focus is on designing the logic for the modules, In other words in system design attention is on what components are needed, while in the detailed design emphasis is on is on the issue of how the various components are implemented in the software.

The design process for software system has two levels. At the first level focus is on deciding which modules are needed for the system, the specification of these modules and how the modules should be interconnected. This is called system design or top level design, in the specification of the module can be satisfied is decided. This design level is often called detailed design or logic design.

Because the detailed design is extension of system design, system design controls the major structural characteristics of the system. The system design has a major impact on the testability and modifiability of a system and impacts its efficiency. Much of the design efforts for the designing of software are spent creating the system design.

4.2 Applicable Documents:

- Synopsis document for **"PawSpot".**
- Software Requirement Specification (SRS) for **"PawSpot".**
- Database Design document for **"PawSpot".**

4.3 Component Description

Login:

In this module, User/ Pet Donor/ Pet Care shop owner logs into PawSpot webapp

- Input : username, password

- Output : Redirection to User, Pet Donor, or Pet care Shop Owner Dashboard
- Processing : Check if the user exists in the system and authenticate the user

Admin has a separate Login page different from the user Login page

- Input : username, password
- Output : Redirection to Admin Dashboard
- Processing : Check if the Admin account exists on the system and authenticate.

.

Admin

- **View User Details**

  In this module,admin can view the user details.

- **View Shop User Details**

  In this module,admin can view all the shop users.

- **View Pet Donor Details**

  In this module,admin can view all the pet donors.

- **View Feedback**

  Lets the admin to view the feedback.

User

- **Register**

  a. **Input** : Username,password,address,phone
  b. **Output** : The user registers in to the application and is redirected to the login page

- **Choose to adopt**

  This module allows the users to send the message to the admin showing interest to adopt a pet.

- **Search the pet donor**

  This module allow the user to view the pet details and contact information of the donor.

- **Products details**

  a. **Input** : Clicking on product details
  b. **Output** : The user can view products,add products to the cart,buy products and cancel the ordered products.

- **View Order details**

a. Input : Clicking on order details
b. **Output :**The user can view the order details

- **Feedback**

a. **Input :** username,feedback
b. **Output :** Feedback details are stored and displayed.

- **Payment**

This module allows user to make payments through credit/debit cards.

Pet Donor

- **Register**

a. Input : Username,password,address,phone
b. Output : The user registers in to the application and is redirected to the login page.

- **Add Pet Details**

a. **Input :** petanimal,breed,color,age,img1,img2, donorid

b. **Output :** Donor can add the details of the pet

- **ViewInterested Users**

This module shows the details and message of people who is interested in adopting.

- **Delete details**

a. **Input :** petid

b. **Output :** Deletes the pet details once the adoption is done

- **Feedback**

a. **Input :** userid,feedback

b. **Output :** Feedback details are stored and displayed

Pet Shop

- **Register**

a. Input **:** Username,password,address,phone
b. Output :The user registers in to the application and is redirected to the login page

- **Add Product Details**

a. Input **:** productid,prodname,description,price

b. Output **:**Owner can add the product details

- **Update Products**

a. Input **:**productid
b. Output **:**Owner can update the pet products.

- **View Orders**

a. **Input** **:**username,productid
b. **Output** **:** Orders for the product are shown.

- **Report**

a. **Input** **:**username,productid
b. **Output** **:**Pet Care Shop owner can view the reports of his sales

- **Feedback**

a. **Input** **:** userid,feedback
b. **Output** **:** Feedback details are stored and displayed

CHAPTER-4

DATABASE DESIGN

5.1 Introduction:

A Database is a collection of related data, which can be of any size and complexity, By using the concept of Database, we can easily store and retrieve data. The major purpose of a database is to provide the information, which utilizes it with the information that the system needs according to its own requirements.

Database design is done before building it to meet needs of end users within a given Information system that a database intended to support to stop the database defines the needed data and data structures that such a database comprises Manila major physical implemented using SQL server.

Along with documenting your database design from the most important design wall you should have his to create a table structure theories table has a primary key and foreign key primary key should meet the following goals.

● Each record is unique within a table and no other record within the table has all of its columns equal to any other.

● For record to be unique all the columns are necessary that is data in one column should not be repeated anywhere else in the table regarding the second goal the column that has completely unique data throughout the table is known as a primary key field a foreign key field is a field that links one table to another.

5.2 Normalized Tables:

5.2.1 **admin** Table :

ATTRIBUTE	TYPE	LENGTH	CONSTRAINTS	DESCRIPTION
adminid	int	20	Primary key	Admin ID
username	varchar	50	Not null	User name
password	varchar	50	Not null	Password

5.2.2 **userregister** Table:

ATTRIBUTE	TYPE	LENGTH	CONSTRAINTS	DESCRIPTION
userid	int	20	Primary key	User ID
name	varchar	50	Not null	Name
email	varchar	50	Not null	Email-ID
password	varchar	50	Not null	Password
address	varchar	50	Not null	Address
contact	int	20	Not null	Contact Number
dob	date	20	Not null	Date Of Birth

5.2.3 **addpet** Table:

ATTRIBUTE	TYPE	LENGTH	CONSTRAINTS	DESCRIPTION
petid	int	20	Primary key	Pet ID
nickname	varchar	50	Not null	Pet name
breed	varchar	50	Not null	Breed name
age	int	15	Not null	Age
health	varchar	100	Not null	Health condition
description	varchar	100	Not null	Discription
image	varchar	100	Not null	Image
donorid	int	20	Foreign key	Donor ID

5.2.4 **adoptpet** Table:

ATTRIBUTE	TYPE	LENGTH	CONSTRAINTS	DESCRIPTION
adoptid	int	20	Primary key	Adopter ID
userid	int	20	Foreign key	User ID
petid	int	20	Foreign key	Pet ID

5.2.5 **donorregister** Table:

ATTRIBUTE	TYPE	LENGTH	CONSTRAINTS	DESCRIPTION
donorid	int	20	Primary key	Pet Donor ID
name	varchar	50	Not null	Name
address	varchar	50	Not null	Address
email	varchar	50	Not null	Email-ID
password	varchar	50	Not null	Password
contact	int	20	Not null	Contact number

5.2.6 **ordertb** Table:

ATTRIBUTE	TYPE	LENGTH	CONSTRAINTS	DESCRIPTION
orderid	int	20	Primary key	Order ID
userid	int	20	Foreign key	User ID
productid	int	20	Foreign key	Product ID
quantity	int	20	Not null	Quantity
payment	int	20	Not null	Payment

5.2.7 **petshopregister** Table:

ATTRIBUTE	TYPE	LENGTH	CONSTRAINTS	DESCRIPTION
psid	int	20	Primary key	Pet Care ID
shopname	varchar	50	Not null	Shop name
ownername	varchar	50	Not null	Owner name
address	varchar	50	Not null	Address
email	varchar	50	Not null	Email-ID
password	varchar	20	Not null	Password
phone	varchar	12	Not null	Phone

5.2.8 **products** Table:

ATTRIBUTE	TYPE	LENGTH	CONSTRAINTS	DESCRIPTION
prodid	int	20	Primary key	Product ID
prodname	varchar	50	Not null	Product name
pdetails	varchar	100	Not null	Description
pprice	int	20	Not null	Product Price
oldprice	int	20	Not null	Old Price
image	varchar	50	Not null	Image

5.2.9 **address** Table:

ATTRIBUTE	TYPE	LENGTH	CONSTRAINTS	DESCRIPTION
aid	int	20	Primary key	Address ID
add1	varchar	100	Not null	Current Address
add2	varchar	100	Not null	Alternate Address
city	varchar	50	Not null	City
pin	int	20	Not null	Pincode
phone	varchar	12	Not null	Phone
userid	int	20	Not null	User ID

5.2.10 **cart** Table:

ATTRIBUTE	TYPE	LENGTH	CONSTRAINTS	DESCRIPTION
cid	int	20	Primary key	Cart ID
prodid	int	20	Foreign key	Product ID
userid	int	20	Foreign key	User ID
quantity	int	20	Not null	Quantity

5.2.11 **Feedback** Table:

ATTRIBUTE	TYPE	LENGTH	CONSTRAINTS	DESCRIPTION
feedbackid	int	20	Primary key	Feedback ID
username	varchar	20	Not null	Username
email	varchar	20	Not null	Email
feedback	varchar	50	Not null	Feedback

CHAPTER-6

CODING

6.1 Introduction

All the software engineering steps that have been discussed so far are directed towards a final objective of translating the modules into the software form that could be understood by the computer. Coding is an important aspect in the system development and proper care has been taken to ensure that the application is efficient and suits requirements. Coding is viewed as a natural consequence of design.

6.2 Programming Style

It is impossible to provide an exhaustive list of what to do and what no to produce simple readable code. Being able to do this will amount to providing an algorithm for writing good code.

- **Names:**

  Selecting module and variable names often not considered important by programmers. Only when one starts reading program written by others where the variables name are cryptic and not representative does on realize the importance of selecting proper names.

- **Control Construct:**

  It is desirable that as much as possible single entry, single exit construct be used, it is desirable to use few standards control constructs rather than using a wide variety of constructs, just because they are available in the language.

- **Information Hiding:**

  Information hiding is supported where possible. Only the access function for the data structure should be made visible while hiding the data structure behind the function.

- **User-defined types:**

  Modern language allow user to define type like the enumerated type. When such facilities are available, they should be exploited where applicable.

- **Nesting:**

  The different control constructs, the if-then-else, can be nested. If the nesting becomes too deep, the programs become harder to understand.

- **Module size:**

  A programmer should carefully examine any routine with very few statements or too many statements. Large module often will not functionally cohesive and too small modules unnecessary overhead.

- **Module interface:**

  A module with a complex interface should be carefully examined .Such modules might not be functionally cohesive and might be implementing multiple functions.

- **Program Layout:**

  How the program is organized and presented can have affect on readability of it. Proper indentation, bank spaces and parentheses should be used to enhance the readability of programs.

- **Side Effect:**

  When a module is invoked, it sometimes has side effect of modifying program state beyond the modification of parameters listed in the module interface definition, for example, modifying global variables.

6.3 Coding:

Register:

```
<?php
if(isset($_POST['register']))
{
	error_reporting(1);
	include("config.php");
	$Email=$_POST['email'];
	$sql = "select * from userregister where email='$Email'";
	$result = mysqli_query($con,$sql);
	$count=mysqlI_num_rows($result);
	if($count>0)
	{
			echo "<script>
				alert('There is an existing account associated with this email.');
			</script>";
			echo "<script> location.href='userreg.php'; </script>";
	}
	else
	{
		$Name=$_POST['name'];
		$Email=$_POST['email'];
		$question=$_POST['question'];
		$answer=$_POST['answer'];
		$Phone=$_POST['phone'];
		$Address=$_POST['address'];
		$password=$_POST['password'];
		if ($_POST['password'] != $_POST['cpassword']) {
```

```
  // fail!
   echo "<script>
                        alert('Password invalid.');
                        </script>";
}
else {
  // success :(
        $query = "insert into
userregister(name,email,question,answer,phone,address,password)
values('".$Name."','".$Email."','".$question."','".$answer."','".$Phone."','".$Address."','".$pass
word."')";
        mysqli_query($con,$query) or die(mysqli_error($con));
                echo "<script>
                                alert('Registeration Completed, Please Login.');
                        </script>";
                        echo "<script> location.href='userlogin.php'; </script>";
                }
        }
}
?>
```

Login:

```
<?php
session_start();
?>
<?php
if(isset($_POST['login']))
{
        error_reporting(1);
```

```
include("config.php");
$Email=$_POST['email'];
$Password=$_POST['password'];
$sql = "select * from userregister where email='$Email' and password='$Password'";
$result = mysqli_query($con,$sql);
$count=mysqlI_num_rows($result);
if($count>0)
{
        session_start();
        $_SESSION['petuser']=$Email;
        $id=$_SESSION['petuser'];
        echo "<script>
                        alert('Login Successful $id');
                </script>";
        echo "<script> location.href='user/index.php'; </script>";
}
else
{
   echo "<script>
                        alert('Invalid Email or Password');
                        </script>";
}
}
?>
```

Admin side:

View Feedback:

```
<?php
	include("config.php");
	$sql = "select * from userfeedback";
	$result = mysqli_query($con,$sql);
	$count=mysqlI_num_rows($result);
if($count>0)
	{
		while($row = mysqli_fetch_array($result))
		{
			$id=$row[0];
			$uname=$row['uname'];
			$uemail=$row['uemail'];
			$umessage=$row['umessage'];
			?>
				<tbody >
					<tr>
						<td><?php echo $id; ?></td>
						<td>
							<h5 class="font-16"><?php echo
$uname; ?></h5>
						</td>
						<td><?php echo $uemail; ?></td>
						<td><?php echo $umessage; ?></td>
						<td>

							<form method="post" action="">
<input type="text" name="id[]" value="<?php echo $id; ?>" hidden="true">
```

```
<button name="delete"><i class="dw dw-delete-3"></i>Delete</button>
</form>
                                                    </td>
                                                </tr>
<?php
                }


        }
?>
                                        </tbody>
                                    </table>
<?php
if(isset($_POST['delete']))
{
        foreach ($_POST['id'] as $key => $value)
        {
                $selectid=$value;
        }
        $query1 = "delete from userfeedback where id='".$selectid."'";
        mysqli_query($con,$query1) or die(mysqli_error($con));
        echo "<script> location.href='index.php'; </script>";
}
?>
```

View User Table:

```
<?php
	include("config.php");
	$sql = "select * from userregister";
	$result = mysqli_query($con,$sql);
	$count=mysqlI_num_rows($result);
if($count>0)
	{
		while($row = mysqli_fetch_array($result))
		{
			$id=$row[0];
			$name=$row['name'];
			$email=$row['email'];
			$question=$row['question'];
			$answer=$row['answer'];
			$phone=$row['phone'];
			$password=$row['password'];
			?>
<tbody>
		<tr>
		<td><?php echo $id; ?></td>
		<td>
		<h5 class="font-16"><?php echo $name; ?></h5>
		</td>
		<td><?php echo $email; ?></td>
		<td><?php echo $question; ?></td>
		<td><?php echo $answer; ?></td>
		<td><?php echo $phone; ?></td>
		<td><?php echo $password; ?></td>
```

```
                <td>
                <form method="post" action="">
                <input type="text" name="id[]" value="<?php echo $id; ?>" hidden="true">
                <button name="delete"><i class="dw dw-delete-3"></i>Delete</button>
                </form>
                </td>
                </td>
                </tr>
        <?php
                }
        }
?>
        </tbody>
        </table>
<?php
if(isset($_POST['delete']))
{
        foreach ($_POST['id'] as $key => $value)
        {
                $selectid=$value;
        }
        $query1 = "delete from userregister where id='".$selectid."'";
        mysqli_query($con,$query1) or die(mysqli_error($con));
        echo "<script> location.href='usertable.php'; </script>";
}
?>
```

View Pet Donor Table:

```
<?php
        include("config.php");
        $sql = "select * from donorregister";
        $result = mysqli_query($con,$sql);
        $count=mysqlI_num_rows($result);
   if($count>0)
        {
                while($row = mysqli_fetch_array($result))
                {
                        $did=$row[0];
                        $dname=$row['dname'];
                        $demail=$row['demail'];
                        $dquestion=$row['dquestion'];
                        $danswer=$row['danswer'];
                        $dphone=$row['dphone'];
                        $daddress=$row['daddress'];
                        $dpassword=$row['dpassword'];
                        ?>
                                        <tbody>
                                                <tr>
                                                        <td><?php echo $did; ?></td>
                                                        <td>
                                                        <h5 class="font-16"><?php echo
$dname; ?></h5>
                                                        </td>
                                                        <td><?php echo $demail; ?></td>
                                                        <td><?php echo $dquestion; ?></td>
                                                        <td><?php echo $danswer; ?></td>
```

```
                                        <td><?php echo $dphone; ?></td>
                                        <td><?php echo $daddress; ?></td>
                                        <td><?php echo $dpassword; ?></td>
                                        <td>
                                        <form method="post" action="">
                                        <input type="text" name="did[]"
value="<?php echo $did; ?>"
hidden="true">

                                        <button
name="delete"></i>Delete</button>

                                        </form>
                                        </td>
                                    </tr>
                                    <?php
                }
        }
?>
<?php
if(isset($_POST['delete']))
{
        foreach ($_POST['did'] as $key => $value)
        {
                $selectid=$value;
        }
        $query1 = "delete from donorregister where did='".$selectid."'";
        mysqli_query($con,$query1) or die(mysqli_error($con));
        echo "<script> location.href='donortable.php'; </script>";
}
?>
```

View Pet Shop Table:

```
<?php
	include("config.php");
	$sql = "select * from petshopregister";
	$result = mysqli_query($con,$sql);
	$count=mysqlI_num_rows($result);
if($count>0)
	{
	    while($row = mysqli_fetch_array($result))
	        {
	            $psid=$row[0];
	            $pname=$row['pname'];
	            $psname=$row['psname'];
	            $psemail=$row['psemail'];
	            $psphone1=$row['psphone1'];
	            $psphone2=$row['psphone2'];
	            $psaddress=$row['psaddress'];
	            $psquestion=$row['psquestion'];
	            $psanswer=$row['psanswer'];
	            $pspassword=$row['pspassword'];
	            ?>
	            <tbody>
	                    <tr>
	                        <td><?php echo $psid; ?></td>
	                        <td>
	                            <h5 class="font-16"><?php echo
$pname; ?></h5>
	                        </td>
	                        <td>
```

```
                                        <h5 class="font-16"><?php echo $psname;
?></h5>
                                        </td>
                                            <td><?php echo $psemail; ?></td>
                                            <td><?php echo $psphone1; ?></td>
                                            <td><?php echo $psphone2; ?></td>
                                            <td><?php echo $psaddress; ?></td>
                                            <td><?php echo $psquestion; ?></td>
                                            <td><?php echo $psanswer; ?></td>
                                            <td><?php echo $pspassword; ?></td>
                                            <td>
                                            <form method="post" action="">
                                            <input type="text" name="psid[]"
value="<?php                                                    echo $psid; ?>"
hidden="true">

                                            <button name="delete"><i class="dw
dw-delete-                                                      3"></i>Delete</button>
                                            </form>
                                            </td>
                                        </tr>
                                    </tbody>
                                    </table>
<?php
if(isset($_POST['delete']))
{
    foreach ($_POST['psid'] as $key => $value)
    {
        $selectid=$value;
    }
    $query1 = "delete from petshopregister where psid='".$selectid."'";
    mysqli_query($con,$query1) or die(mysqli_error($con));
```

```
        echo "<script> location.href='petshoptable.php'; </script>";
}
?>
```

User Side:

Adopt Pet:

```
<?php
session_start();
?>
<?php
include 'config.php';
$sql1 = "select * from addpet";
$result1 = mysqli_query($con,$sql1);
$num1=mysqlI_num_rows($result1);
$sl=0;
if($num1 > 0)
{
while($row1 = mysqli_fetch_array($result1))
{
$sl+=1;
$id=$row1[0];
$name=$row1['pname'];
$image=$row1['image1'];
?>
<?php
if(isset($_POST['view']))
{
foreach ($_POST['id'] as $key => $value)
```

```
{
$id=$value;
}
$_SESSION['pets']=$id;
echo "<script> location.href='singlepetdetail.php'; </script>";
}
?>
```

View Petcare Products:

```
<?php
include 'config.php';
$sql1 = "select * from products";
$result1 = mysqli_query($con,$sql1);
$num1=mysqlI_num_rows($result1);
$sl=0;
if($num1 > 0)
{
while($row1 = mysqli_fetch_array($result1))
{
$sl+=1;
$id=$row1[0];
$name=$row1['name'];
$pprice=$row1['pprice'];
$image=$row1['image'];
?>
<?php
if(isset($_POST['view']))
{
foreach ($_POST['id'] as $key => $value)
```

```
{
$id=$value;
}
$_SESSION['item']=$id;
echo "<script> location.href='singleproductdetails.php'; </script>";
}
?>
```

View My Orders:

```
<?php
include 'config.php';
$sql1 = "SELECT ordertb.oid, ordertb.pid, products.name, products.pprice, products.image, ordertb.quantity FROM ordertb INNER JOIN products ON ordertb.pid=products.pid where ordertb.userid='".$_SESSION['petuser']."'";
$result1 = mysqli_query($con,$sql1);
$num1=mysqlI_num_rows($result1);
$sl=0;
$total=0;
if($num1 > 0)
{
while($row1 = mysqli_fetch_array($result1))
{
$oid=$row1[0];
$pid=$row1[1];
$name=$row1[2];
$price=$row1[3];
$img=$row1[4];
$qty=$row1[5];
$tot=$qty*$price;
$total=$total+$tot;
```

```
?>
<?php
if(isset($_POST['delete']))
{
include 'config.php';
foreach ($_POST['oid'] as $key => $value)
{
$orderid=$value;
}
echo
$sql1 = "delete from ordertb where oid='".$orderid."'";
$result1 = mysqli_query($con,$sql1);
echo "<script>
alert('Order cancelled');
</script>";
echo "<script> location.href='myorder.php'; </script>";
}
?>
```

User Feedback:

```
<?php
session_start();
if(isset($_SESSION['petuser']))
{
include 'config.php';
$sql = "SELECT name FROM userregister where email='".$_SESSION['petuser']."'";
$result = mysqli_query($con,$sql);
$count=mysqlI_num_rows($result);
if ($count > 0)
```

```
{
while($row2 = mysqli_fetch_array($result))
{
$name=$row2[0];
}
$_SESSION['name']=$name;
}
}
?>
<?php
if(isset($_POST['submit']))
{
include 'config.php';
$msg=$_POST['message'];
$sql3 = "insert into
userfeedback(uname,uemail,umessage)values('".$_SESSION['name']."','".$_SESSION
['petuser']."','".$msg."')";
mysqli_query($con,$sql3) or die(mysqli_error($con));
echo "<script>
alert('Uploaded Successfully');
</script>";
echo "<script> location.href='index.php'; </script>";
}
?>
```

Pet Donor Side:

Add Pet Details:

```
<?php

session_start();

if(isset($_SESSION['donoruser']))

{

$donoruser=$_SESSION['donoruser'];

include 'config.php';

$sql1 = "select * from donorregister where demail='".$_SESSION['donoruser']."'";

$result1 = mysqli_query($con,$sql1);

$num1=mysqlI_num_rows($result1);

$sl=0;

if($num1 > 0)

{

while($row1 = mysqli_fetch_array($result1))

{

$dname=$row1['dname'];

}

}

?>

<?php

include 'config.php';

if(isset($_POST["addpet"]))

{

$name=$_POST['name'];

$nickname=$_POST['nickname'];

$age=$_POST['age'];

$health=$_POST['health'];
```

```
$description=$_POST['description'];

$fname = $_FILES["image"]["name"];

$filename = $name.$fname;

$tempname = $_FILES["image"]["tmp_name"];

$folder = "image/".$filename;

if (move_uploaded_file($tempname, $folder))  {

$sql = "INSERT INTO addpet
(pname,nickname,age,health,description,image1,donorid,donorname) VALUES
('".$name."','".$nickname."','".$age."','".$health."','".$description."','".$filename."','".$donorus
er."','".$dname."')";

mysqli_query($con, $sql);

echo "<script>

alert('Successfully Uploaded');

</script>";

echo "<script> location.href='viewpet.php'; </script>";

}else{

echo "<script>

alert('Upload Failed, size should be less than 2MB');

</script>";

}

}

?>

<?php

}

else

{

echo "<script> location.href='donorlogin.php'; </script>";

}

?>
```

View Pet Details:

```
<?php
include 'config.php';
$sql1 = "select * from addpet";
$result1 = mysqli_query($con,$sql1);
$num1=mysqlI_num_rows($result1);
$sl=0;
if($num1 > 0)
{
while($row1 = mysqli_fetch_array($result1))
{
$sl+=1;
$id=$row1[0];
$name=$row1['pname'];
$image=$row1['image1'];
?>
<?php
if(isset($_POST['view']))
{
foreach ($_POST['id'] as $key => $value)
{
$id=$value;
}
$_SESSION['pets']=$id;
echo "<script> location.href='singlepetdetail.php'; </script>";
}
?>
<?php
if(isset($_POST['delete']))
```

```
{
foreach ($_POST['id'] as $key => $value)
{
$id=$value;
}
$query6 = "delete from addpet where petid='".$id."'";
mysqli_query($con,$query6) or die(mysqli_error($con));
echo "<script> location.href='viewpet.php'; </script>";
}
?>
```

View Requests:

```
<?php
include 'config.php';
$sql1 = "SELECT adoptpet.adoptid, adoptpet.petid, adoptpet.userid, addpet.pname, addpet.image1, userregister.name, userregister.address, userregister.phone FROM adoptpet INNER JOIN addpet ON adoptpet.petid=addpet.petid INNER JOIN userregister on adoptpet.userid=userregister.email where addpet.donorid='".$_SESSION['donoruser']."'";
$result1 = mysqli_query($con,$sql1);
$num1=mysqlI_num_rows($result1);
$sl=0;
$total=0;
if($num1 > 0)
{
while($row1 = mysqli_fetch_array($result1))
{
$sl+=1;
$adoptid=$row1[0];
$petid=$row1[1];
$userid=$row1[2];
$pname=$row1[3];
```

```
$img=$row1[4];
$name=$row1[5];
$address=$row1[6];
$phone=$row1[7];
?>
```

Feedback:

```
<?php
session_start();
if(isset($_SESSION['donoruser']))
{
include 'config.php';
$sql = "SELECT dname FROM donorregister where
demail='".$_SESSION['donoruser']."'";
$result = mysqli_query($con,$sql);
$count=mysqlI_num_rows($result);
if ($count > 0)
{
while($row2 = mysqli_fetch_array($result))
{
$name=$row2[0];
}
$_SESSION['dname']=$name;
}
}
?>
```

Pet Shop Side

Add product details:

```
<?php

session_start();
if(isset($_SESSION['shopuser']))
{
$shopuser=$_SESSION['shopuser'];
include 'config.php';
$sql1 = "select * from petshopregister where psemail='".$_SESSION['shopuser']."'";
$result1 = mysqli_query($con,$sql1);
$num1=mysqlI_num_rows($result1);
$sl=0;
if($num1 > 0)
{
while($row1 = mysqli_fetch_array($result1))
{
$psname=$row1['psname'];
}
}
?>
<?php
include 'config.php';
if(isset($_POST["add"]))
{
        $name=$_POST['name'];
        $pprice=$_POST['pprice'];
        $oldprice=$_POST['oldprice'];
        $sstatus=$_POST['sstatus'];
```

```
        $quantity=$_POST['quantity'];
        $pdetails=$_POST['pdetails'];
        $fname = $_FILES["image"]["name"];
        $filename = $name.$fname;
        $tempname = $_FILES["image"]["tmp_name"];
        $folder = "image/".$filename;
        if (move_uploaded_file($tempname, $folder))  {
$sql = "INSERT INTO products (name,pprice,oldprice,sstatus,quantity,pdetails,image,shopuser,psname) VALUES ('".$name."','".$pprice."','".$oldprice."','".$sstatus."','".$quantity."','".$pdetails."','".$filename."','".$shopuser."','".$psname."')";
                mysqli_query($con, $sql);
        echo "<script>
                                alert('Upload Successful');
                            </script>";
      }else{
        echo "<script>
                                alert('Upload Failed, IMAGE should be less than 2GB');
                            </script>";
    }
}
?>
<?php
}
else
{
        echo "<script> location.href='shoplogin.php'; </script>";
}?>
```

View Products:

```
<?php
session_start();
?>
<?php
include 'config.php';
$sql1 = "select * from products";
$result1 = mysqli_query($con,$sql1);
$num1=mysqlI_num_rows($result1);
$sl=0;
if($num1 > 0)
{
while($row1 = mysqli_fetch_array($result1))
{
$sl+=1;
$id=$row1[0];
$name=$row1['name'];
$pprice=$row1['pprice'];
$image=$row1['image'];
?>
<?php
                if(isset($_POST['view']))
                        {
                                foreach ($_POST['id'] as $key => $value)
                                        {
                                                $id=$value;
                                        }
                                $_SESSION['item']=$id;
```

```
                                    echo "<script> location.href='singleproductdetails.php';
</script>";
                                }
?>
```

View Orders:

```
<?php

session_start();

if(isset($_SESSION['shopuser']))

{

?>

<?php

include 'config.php';

$sql1 = "SELECT ordertb.oid, ordertb.pid, products.name, products.pprice, products.image,
ordertb.quantity, ordertb.userid, address.address1, address.address2, address.city, address.pin,
address.contact FROM ordertb INNER JOIN products ON ordertb.pid=products.pid INNER
JOIN address on ordertb.userid=address.userid where
products.shopuser='".$_SESSION['shopuser']."'";

$result1 = mysqli_query($con,$sql1);

$num1=mysqlI_num_rows($result1);

$sl=0;

$total=0;

if($num1 > 0)

{

while($row1 = mysqli_fetch_array($result1))

{

$sl+=1;

$oid=$row1[0];

$pid=$row1[1];

$name=$row1[2];

$price=$row1[3];
```

```
$img=$row1[4];
$qty=$row1[5];
$Userid=$row1[6];
$addr1=$row1[7];
$addr2=$row1[8];
$city=$row1[9];
$pin=$row1[10];
$contact=$row1[11];
$tot=$qty*$price;
$total=$total+$tot;
?>
<?php
}
else
{
        echo "<script> location.href='user-login.php'; </script>";
}
?>
```

Feedback:

```
<?php
session_start();
if(isset($_SESSION['shopuser']))
{
   include 'config.php';
        $sql = "SELECT pname FROM petshopregister where
psemail='".$_SESSION['shopuser']."'";
        $result = mysqli_query($con,$sql);
        $count=mysqlI_num_rows($result);
        if ($count > 0)
```

```
        {
            while($row2 = mysqli_fetch_array($result))
              {
                $name=$row2[0];
              }
            $_SESSION['pname']=$name;
        }
}
?>
<?php
            if(isset($_POST['send']))
                {
                    include 'config.php';
                    $msg=$_POST['message'];
                    $sql3 = "insert into
                    userfeedback(uname,uemail,)values('".$_SESSION['pname']."','
                ".$_SESSI      ON['shopuser']."','".$msg."')";
                    mysqli_query($con,$sql3) or die(mysqli_error($con));
                    echo "<script>
                        alert('Upload Successful');
                        </script>";
                        echo "<script> location.href='feedback.php'; </script>";
                }
?>
```

CHAPTER-7

TESTING

7.1 Introduction

Testing is the major quality control measure used during software development it is a basic function to detect errors in the software during the requirement analysis and design, of the output of the document that is usually textual and non-executable after the coding phase. The computer programs are available and can be executed for testing purpose this implies that testing not only has to uncover errors introduced during the previous phase.

The goal of testing is done cover requirement design and coding errors in the program system's working according to the specifications. It is the phase where we try to break system system will use it. Implementation is a final important phase it involves user training system testing in order to ensure successful running of the proposed system. The user tests the system and changes are made according to their needs .The testing involves the testing of the developed system using various kinds of data. While testing errors are noted and correctness is maintained.

7.2 Testing Criteria

7.2.1 Objective of testing

- Finding defects which may get created by the programmer while developingthe software
- Gaining confidence in and providing information about the level of quality
- To prevent defects
- To make sure that the end results meets the business and user requirements.
- To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specification.

7.3 Testing Methodology

7.3.1 Unit Testing:

Unit testing focuses efforts on the smallest unit of software design this is known as module testing or white box testing, the modules are tested separately. The test is carried out during the programming stage itself. In this step, each module is found to be working satisfactorily as regards to the expected output from the module.

7.3.2 Integration Testing:

In integration testing the different units of the system are integrated together to form the complete system. This type of testing checks the system as a whole to ensure that it is doing what it's supposed to do the testing of an integrated system can be carried out top-down, bottom-up or Big-Bang. In this type of testing some parts are tested with white box testing and some with black box testing techniques. This type of testing plays a very important role in increasing systems productivity. We have checked the system by using integration testing techniques.

7.3.3 System Testing:

Apart from testing the system to validate the functionality of software against the requirements. It is also necessary to test the non functional aspect of the system, some examples of non functional tools include tests to check the performance data security usability volume load and stress that we have used in a project to test the various module. System testing consists of the following steps:

1. Program(s) testing
2. String testing
3. System Testing
4. System Documentation
5. User Acceptance test

7.3.4 Field Testing:

The special type of testing may be very important in some projects. Here the system is tested in actual operational surroundings the interfaces with other systems and the real world are checked. This type of testing is very rarely used so far our project is concerned we haven't tested a project using field testing.

7.3.5 Acceptance Testing:

After the developer has completed all rounds of testing and he is satisfied with the system, then the user takes over and retest system from his point of view to judge whether it is acceptable according to some previously identified criteria. This is almost always a tricky situation in the project because of the inherent conflict between the developer and the user in this project. It is the job of the developer of the planner to check the system that whether the system fulfills the goals or not.

7.4 Test Cases

Testing for valid user name:

Test case	Input	Test description	Output
1	User name starts with character	User name cannot start with character	Must Enter Characters
2	User name is left blank	User name cannot be left blank	Must Enter username

Testing for valid password:

Test case	Input	Test description	Output
1	Password is left blank	Password cannot be blank	Must Enter password
2	Invalid password entered	Valid password must be entered	Password mismatch

Testing for valid Email address:

Test case	Input	Test description	Output
1	Email address is not in Correct format	Email address Should have Correct format	Invalid Expression
2	Email address with space	Email address cannot have space	Invalid Expression
3	Email is left blank	Email cannot be blank	Must Enter Email ID

Testing for data insertion:

Test case	Input	Test description	Output
1	Mandatory fields left empty	Mandatory fields cannot be left empty	cannot be left empty Must enter data
2	Duplicate entry	Duplicate entry not allowed	Appropriate error message
3	Input without above faults	Valid input	Record inserted Successfully

Testing for phone number:

Test case	Input	Test description	Output
1	Phone number entered with alphabets	Phone number Cannot have alphabets	Enter only numbers
2	Phone number entered is more than 10 digits	Phone number with more than 10 digits cannot be entered	Invalid phone number

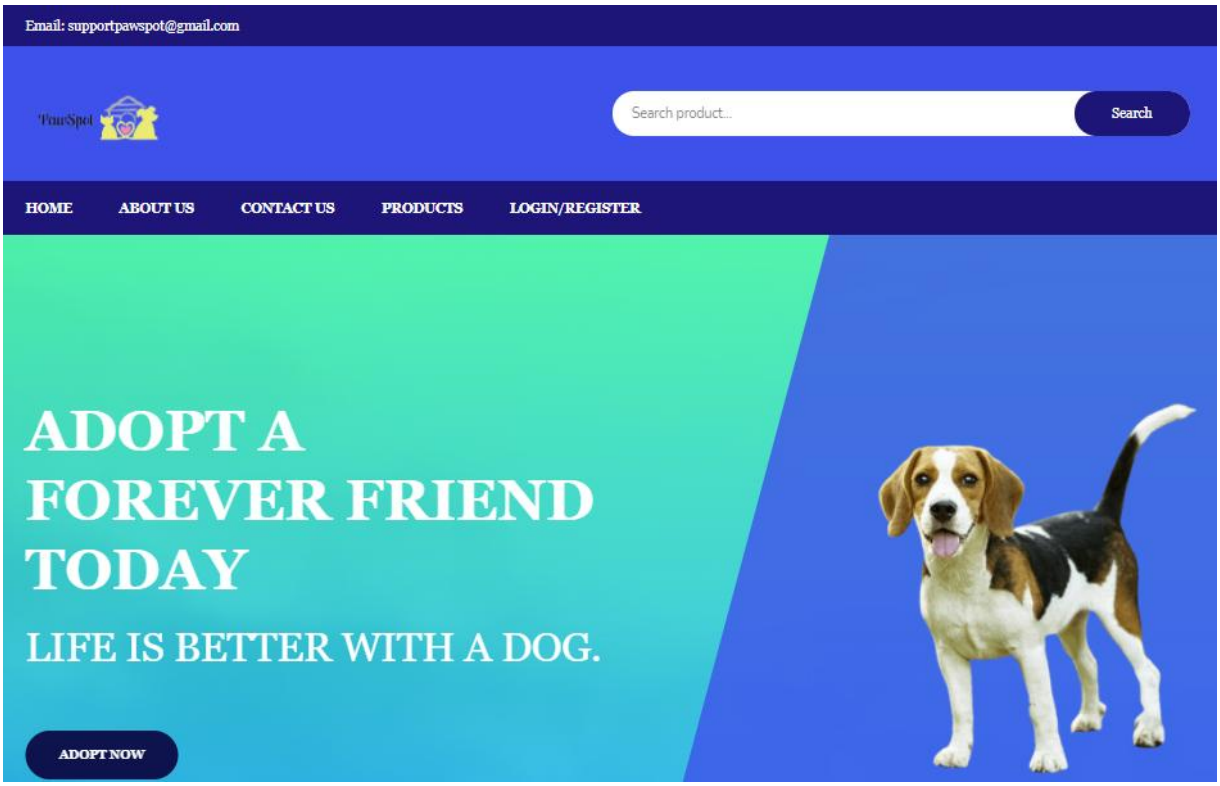
Testing for change password:

Test case	Input	Test description	Output
1	Any field left blank	All fields are compulsory	Must enter Password
2	Invalid password	Valid password must be entered	Enter correct password
3	Retyped password does not match	Retyped password must match	Password mismatch
4	Valid input	Valid input	Password changed successfully

CHAPTER-8

SNAPSHOTS

Home Page:



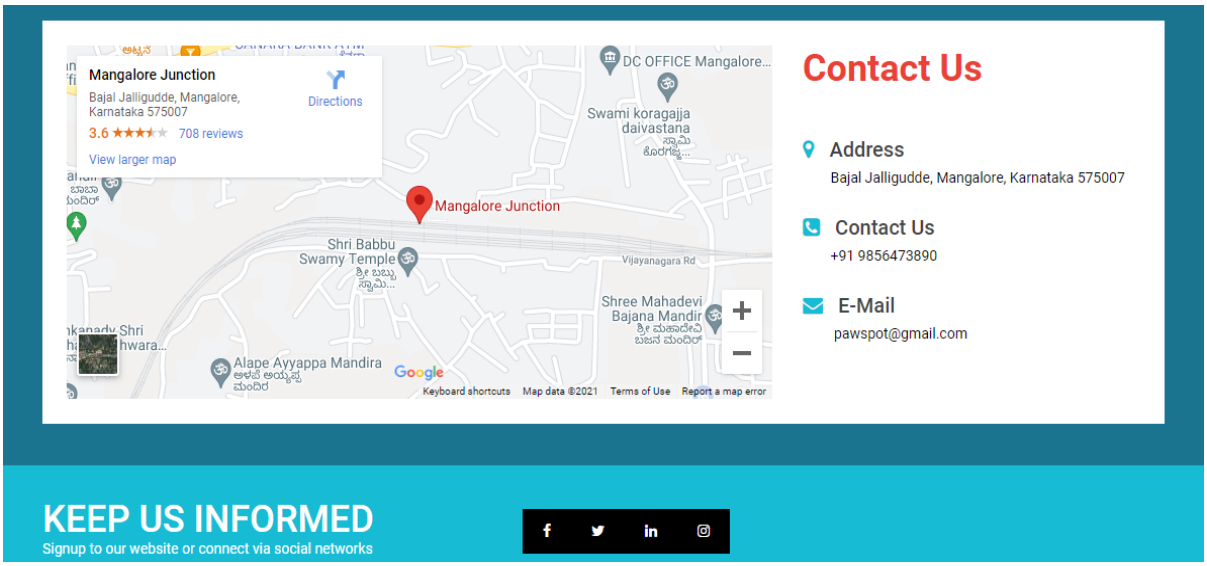
About Us:

ABOUT US

The main idea of PawSpot webapp is to provide a user friendly interface to automate the process of serving towards the welfare of the pets by giving the abandoned pets a place of shelter and care them with affection. The application also gives guidelines for caring towards the pets, adoption procedure of a pet. The user can do the adoption process through the application as this process is time consuming if done manually. The user can register themselves to the application and adopt a pet or purchase the products related to pets. For adoption, the user can choose the pets that are available, view their details and if they wish they can continue with the process of adoption.

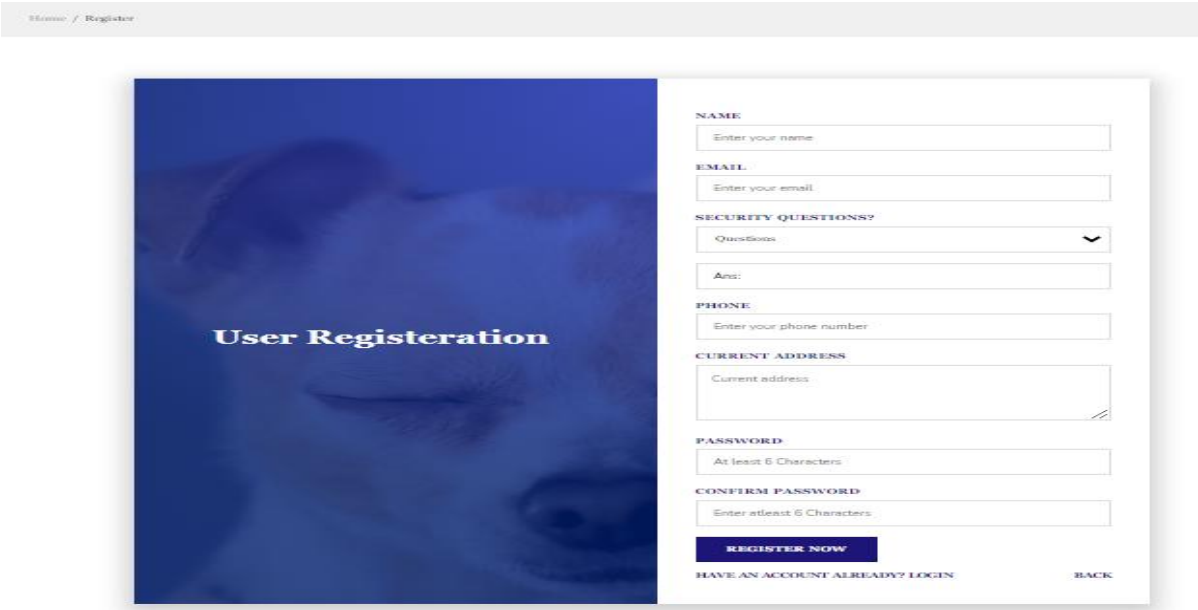
Read More

Contact Us:

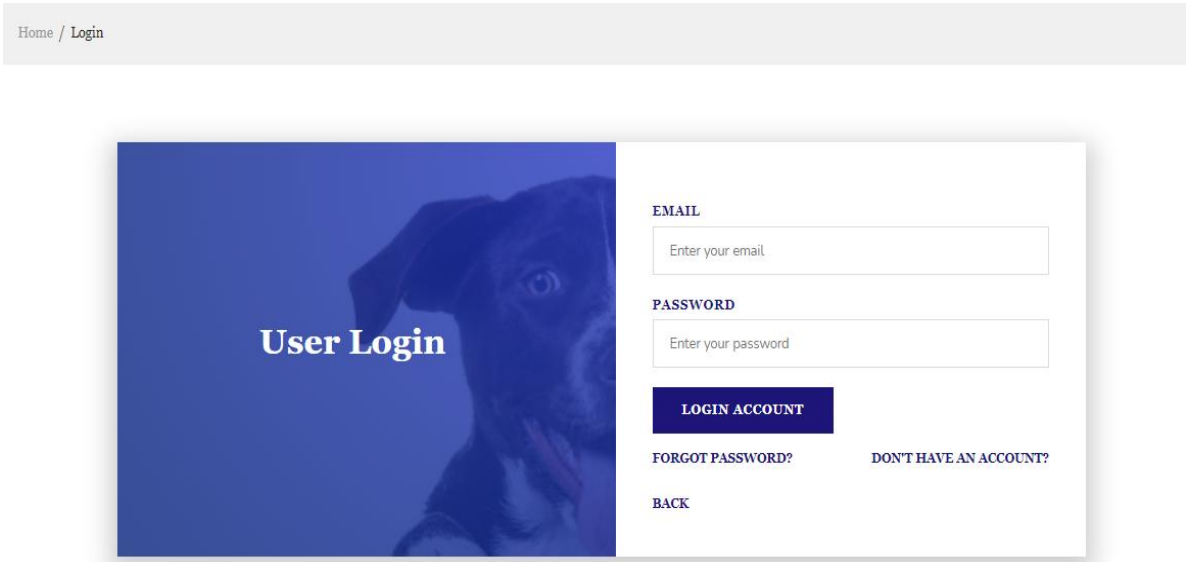


USER SIDE:

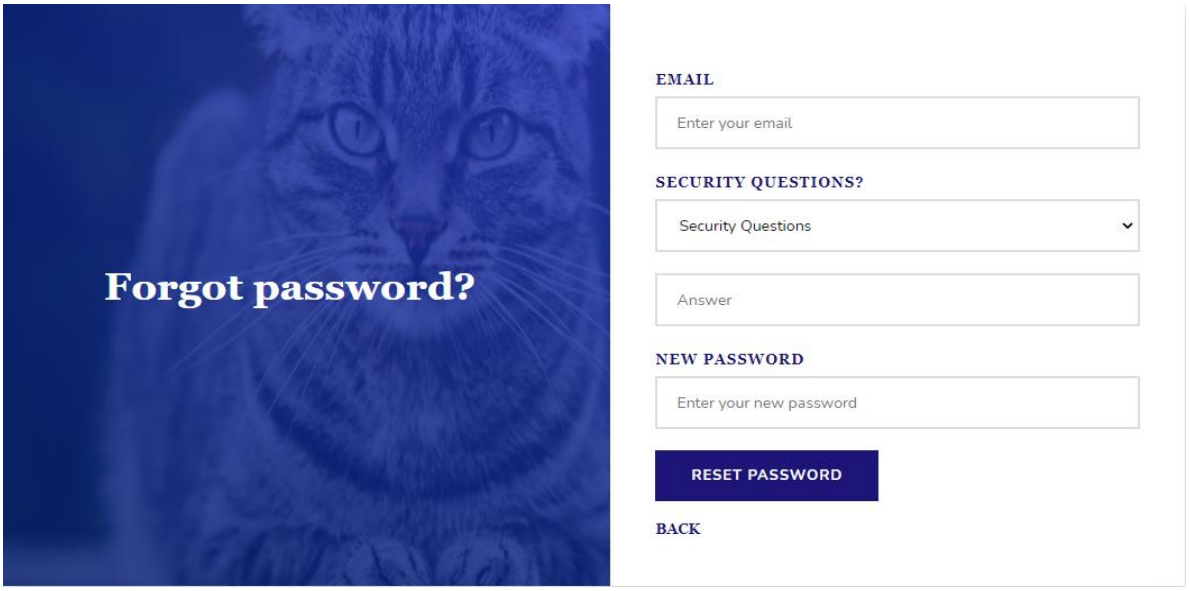
User Registration:



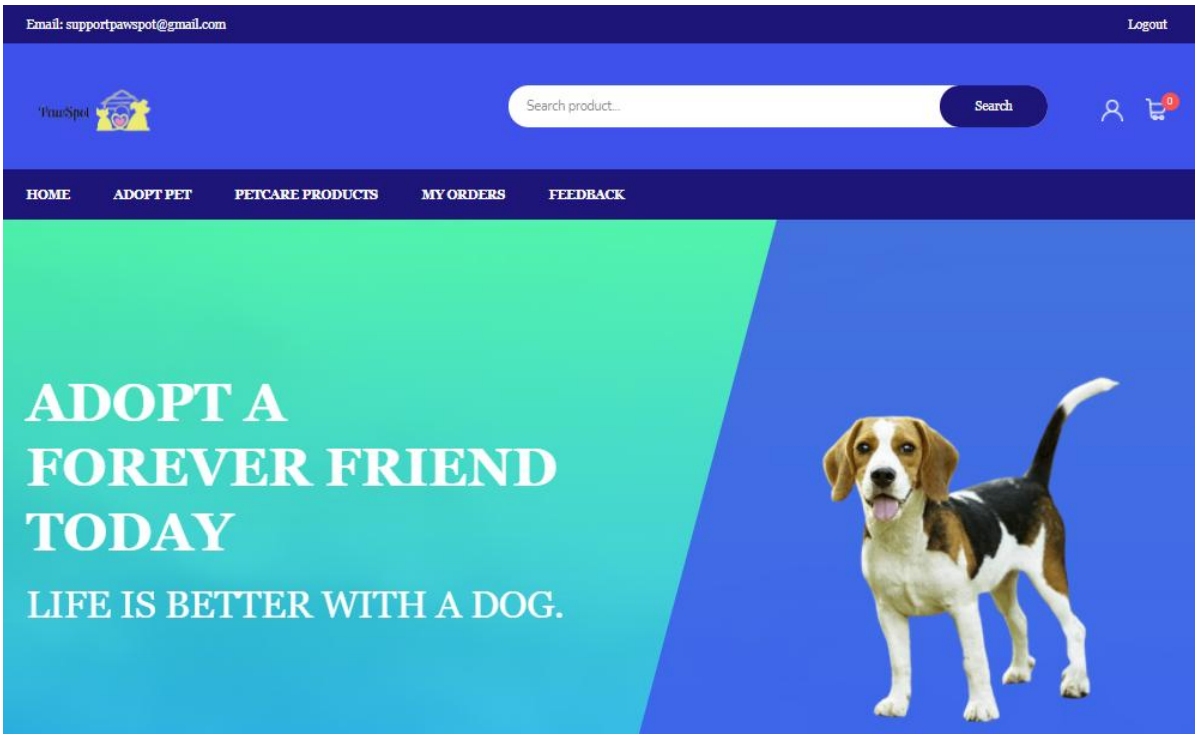
User Login:



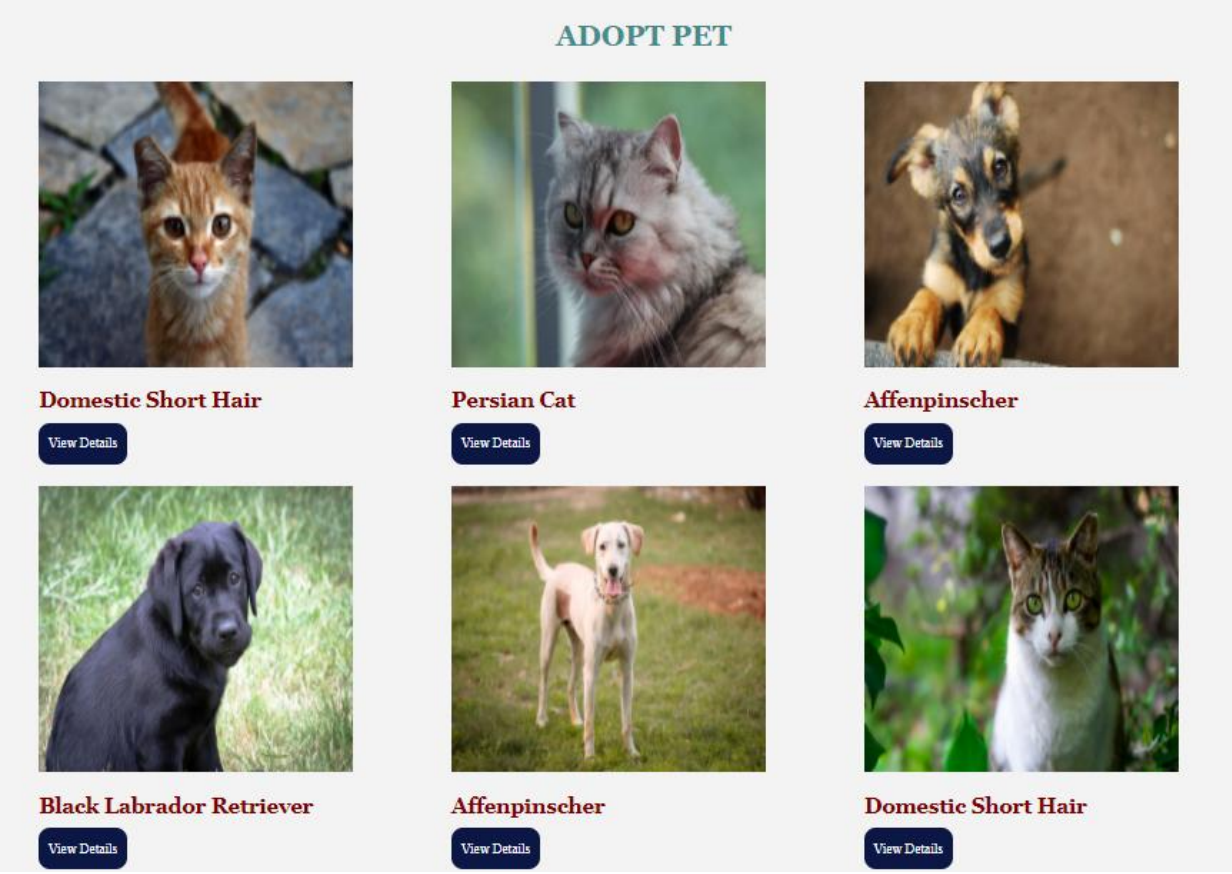
User Forgot Password:



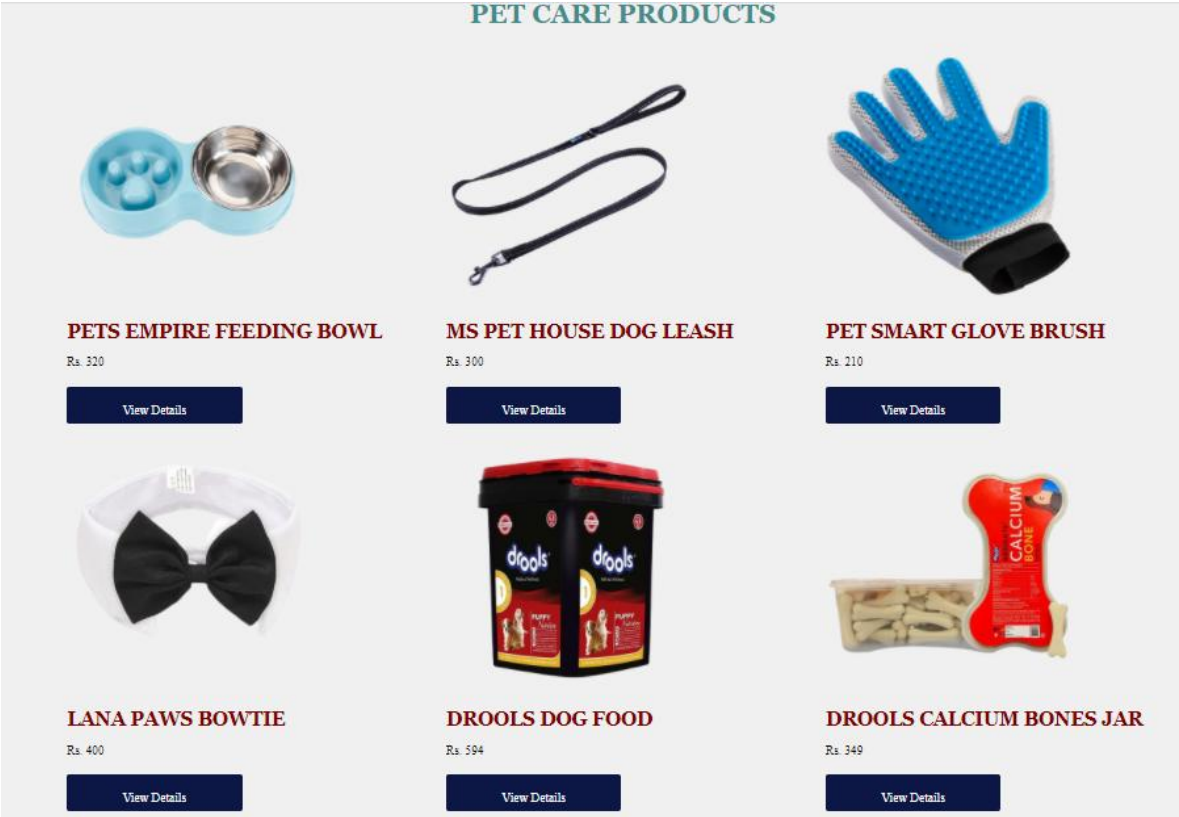
User Home Page:



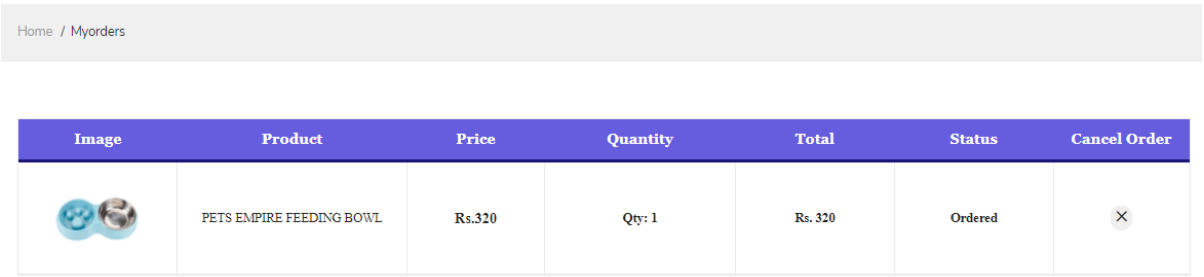
User Adopt Pet Page:



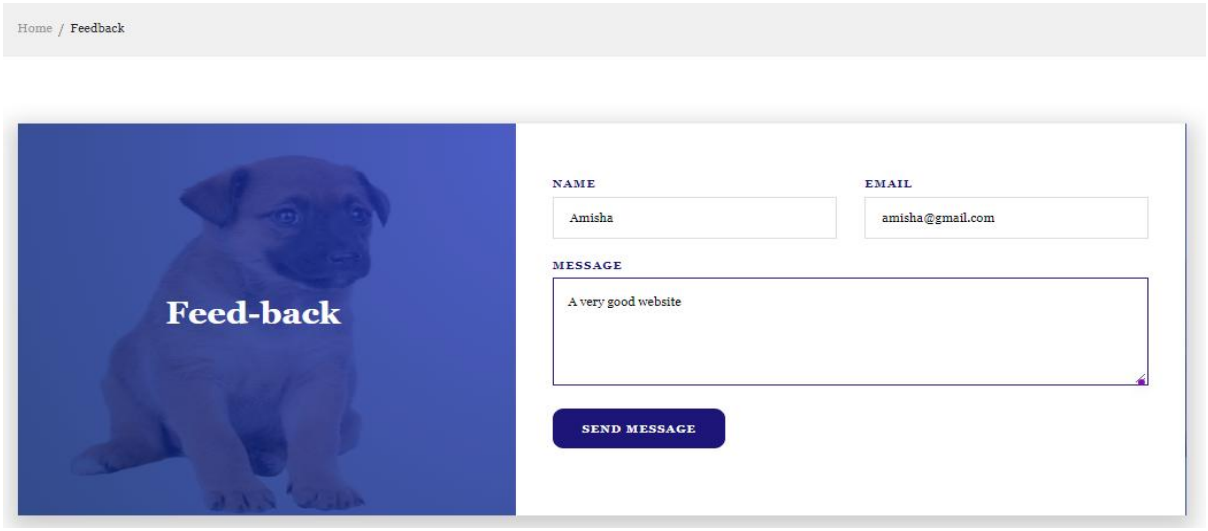
User Petcare Products Page:



User My Orders Page:



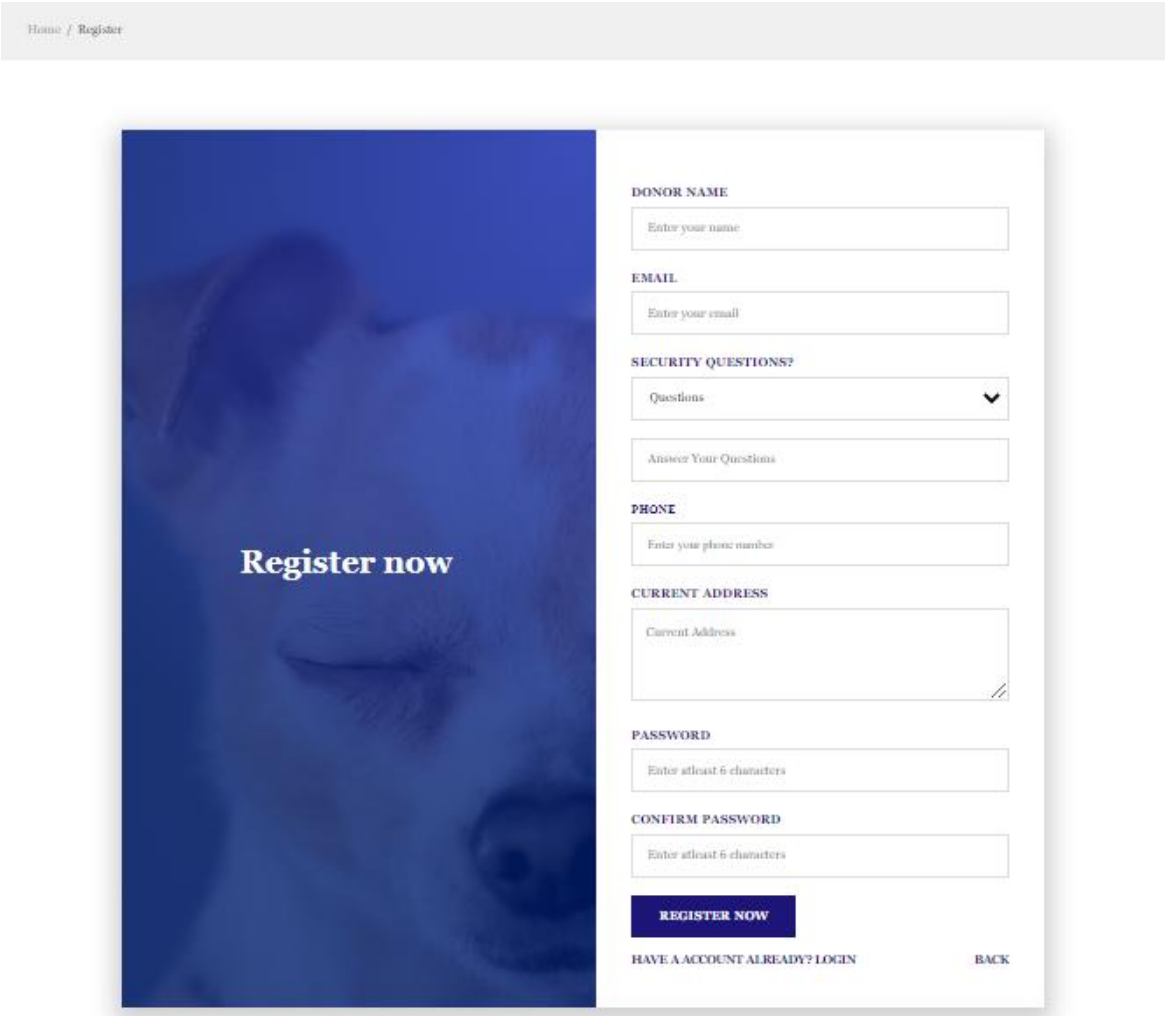
User Feedback Page:



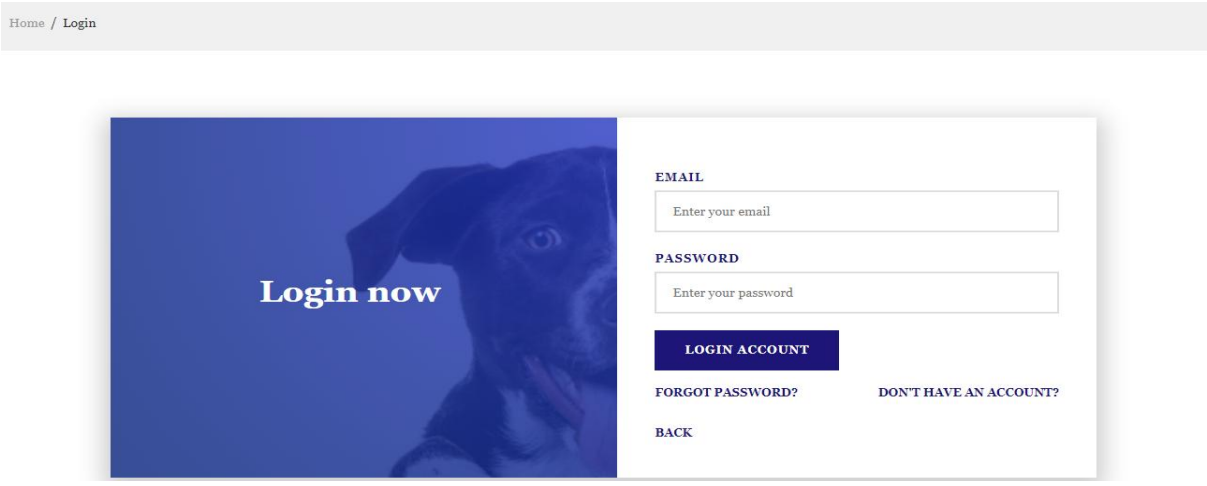
User Profile:

PET DONOR SIDE:

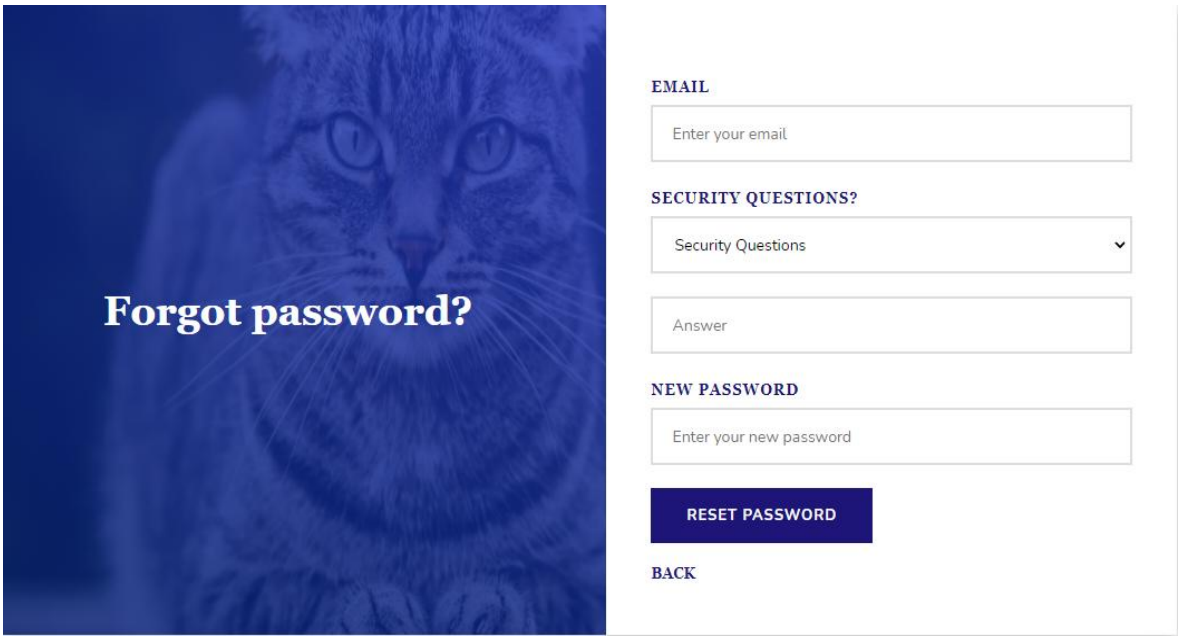
Pet Donor Registration:



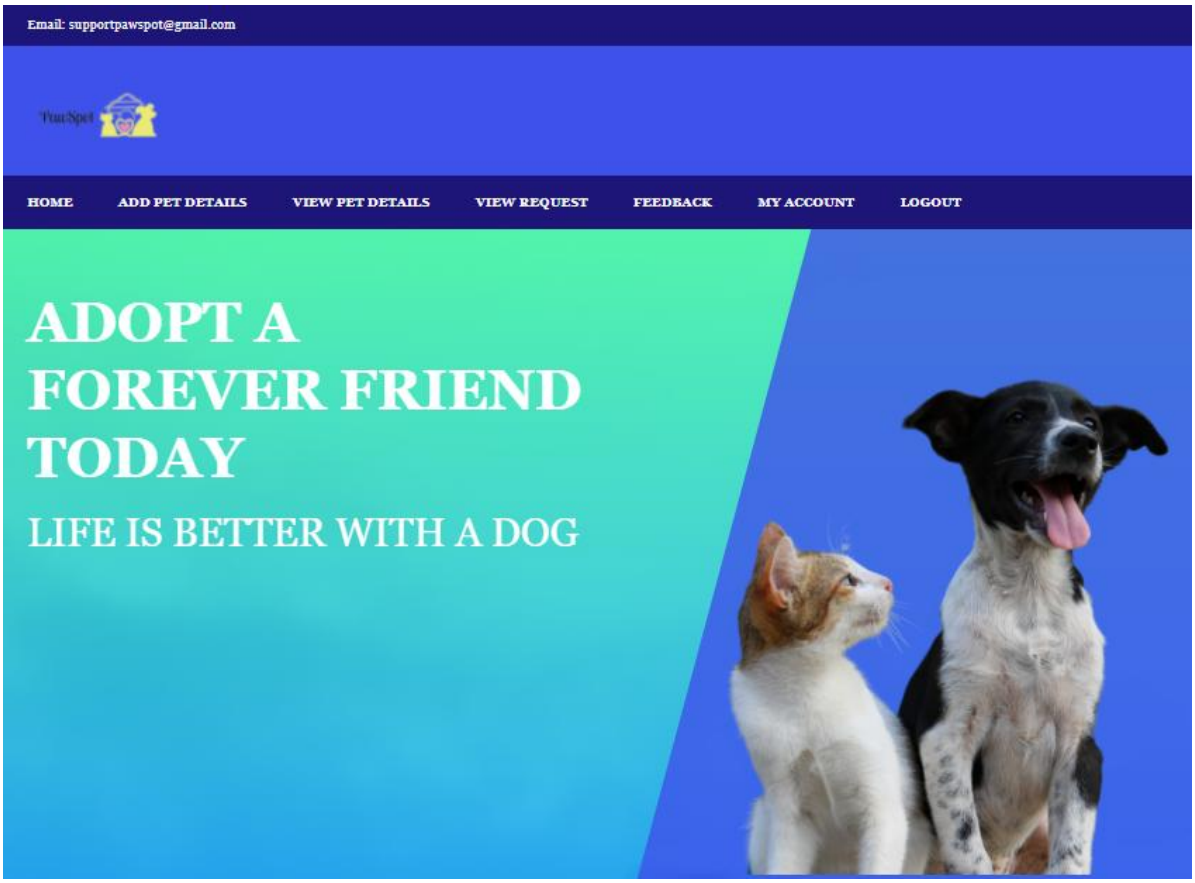
Pet Donor Login:



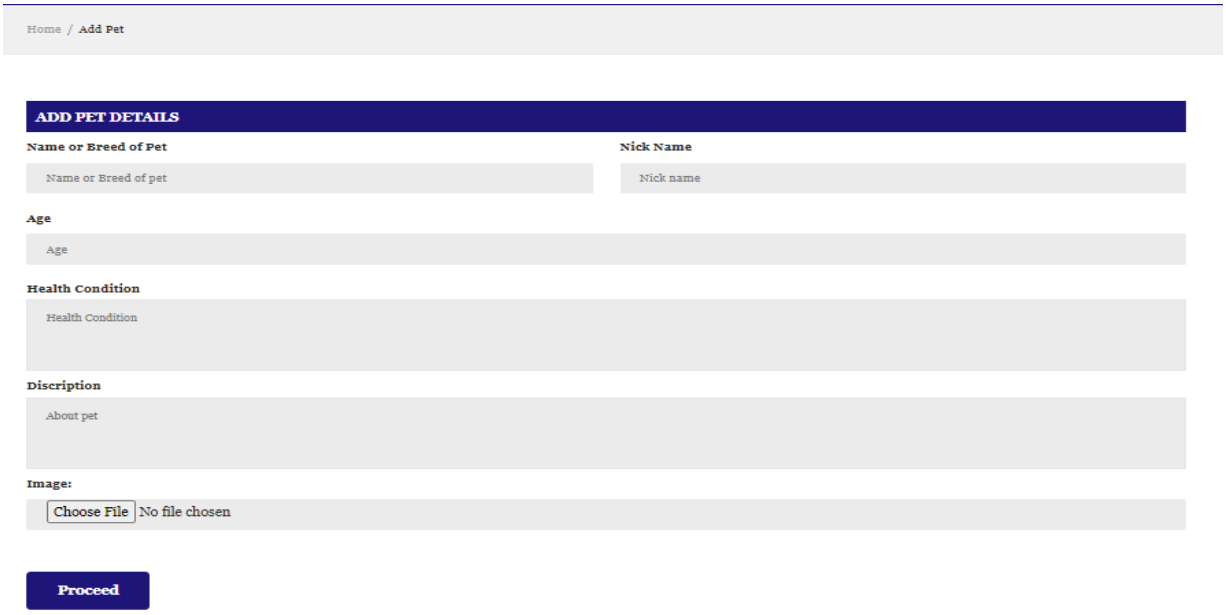
Pet Donor Forgot Password:



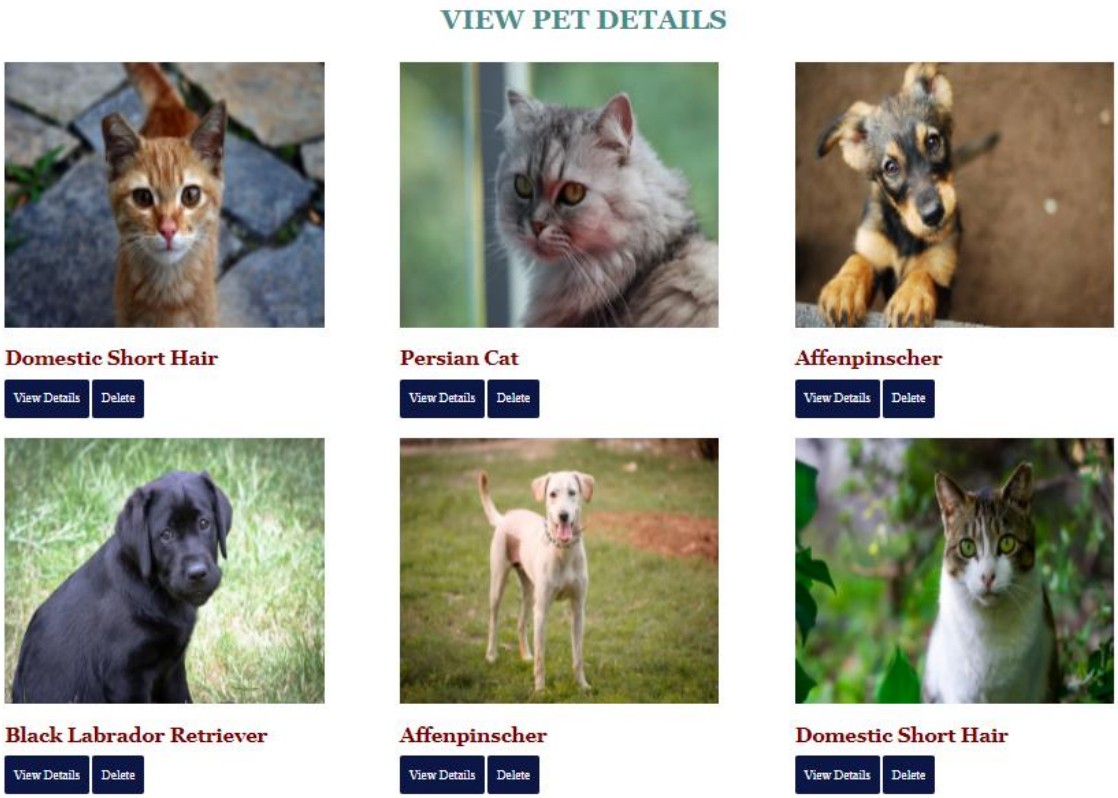
Pet Donor Home Page:



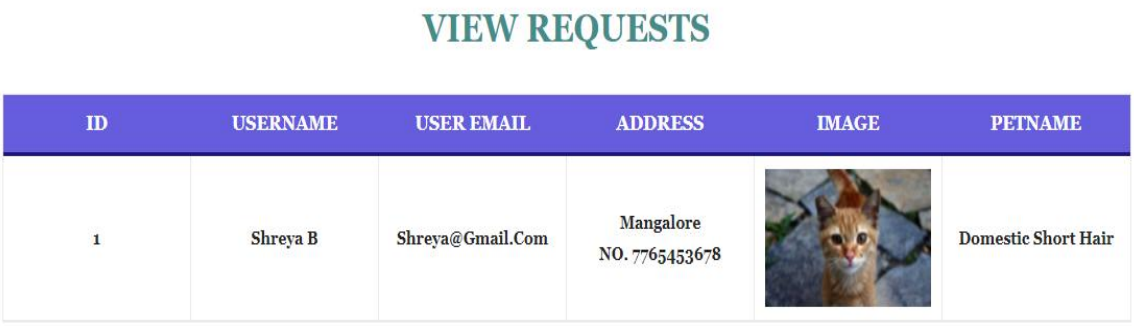
Pet Donor Add Pet Details Page:



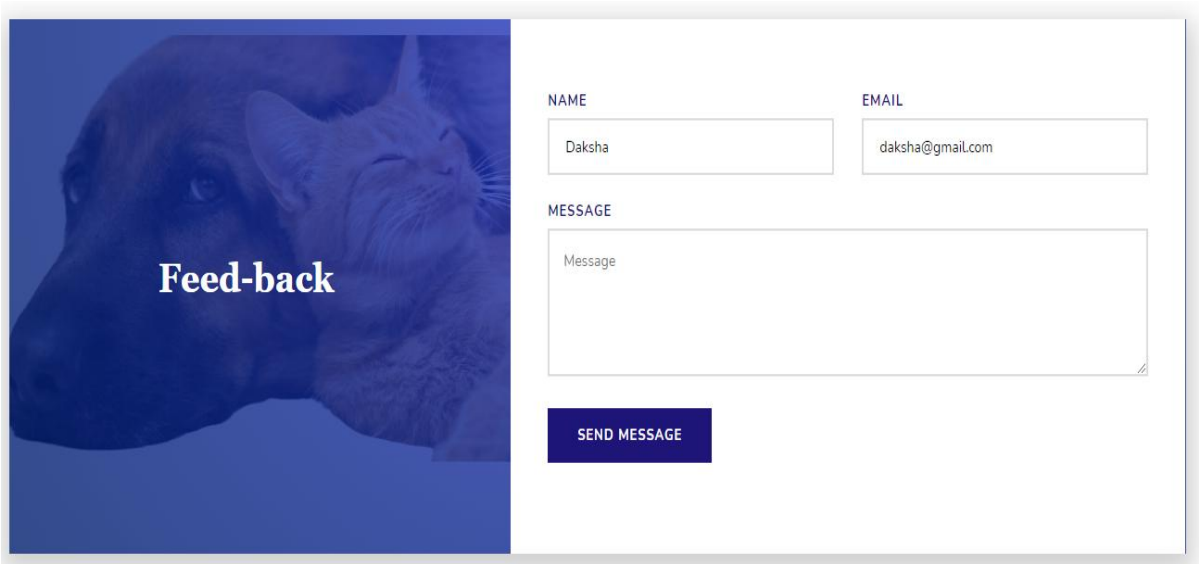
Pet Donor View Pet Details:



Pet Donor View Requests Page:



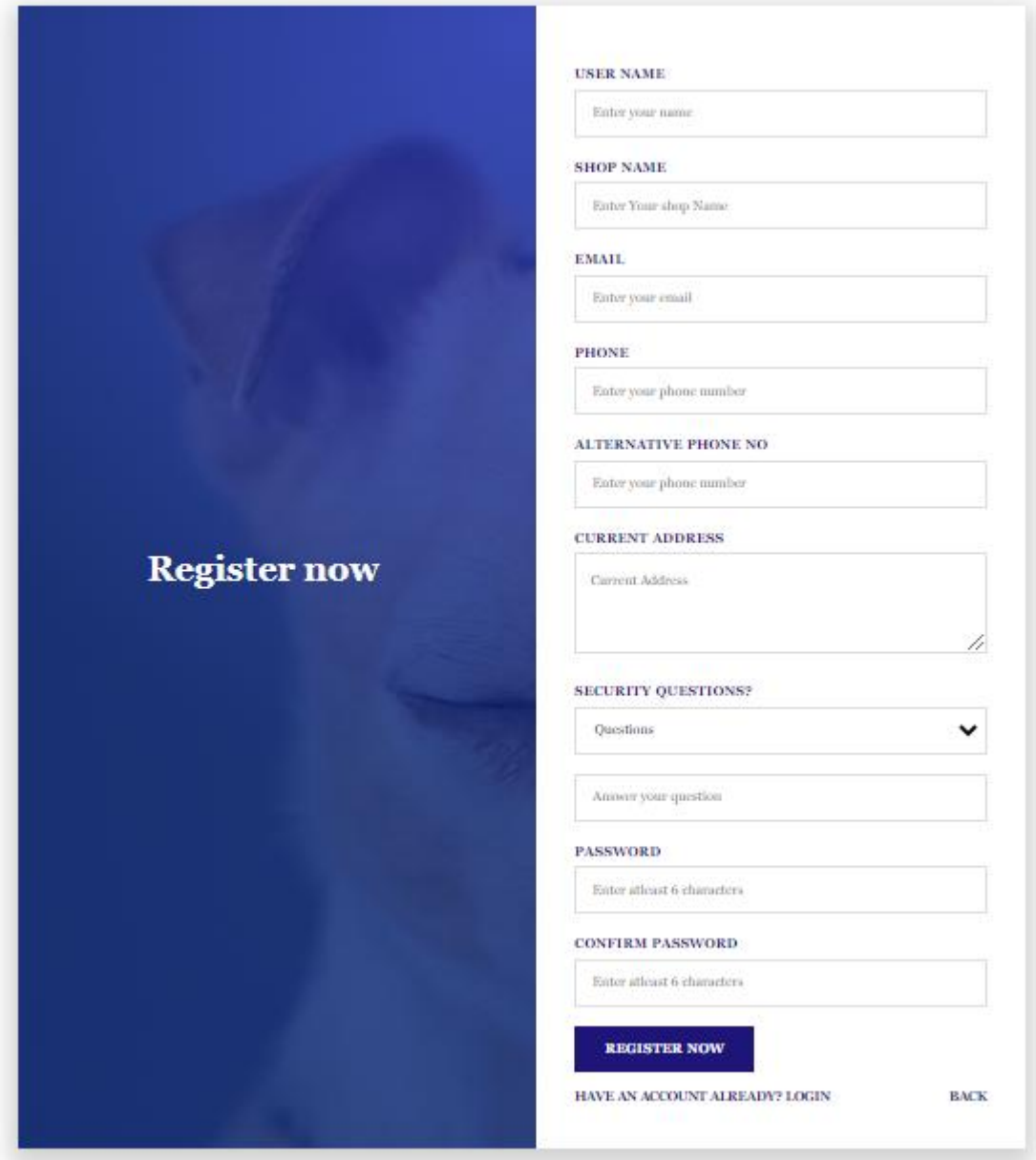
Pet Donor Feedback Page:



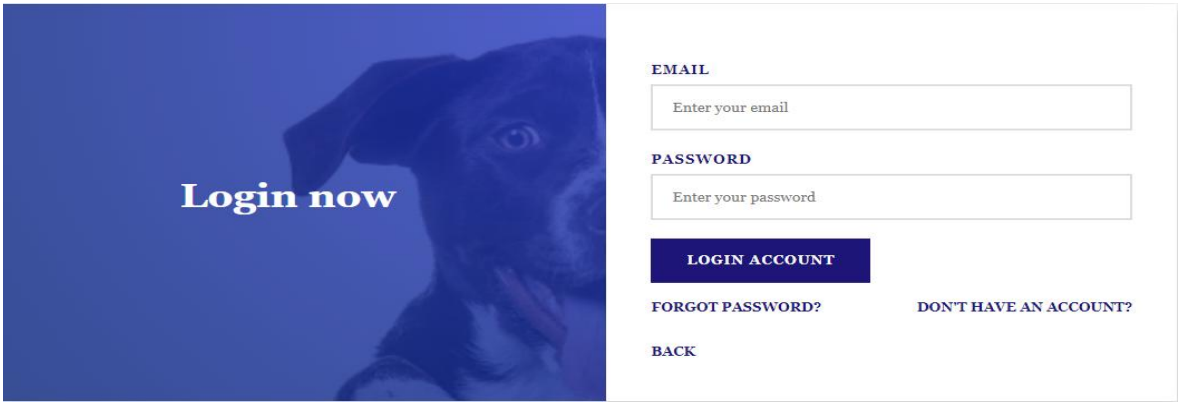
Pet Donor Profile:

PET SHOP SIDE:

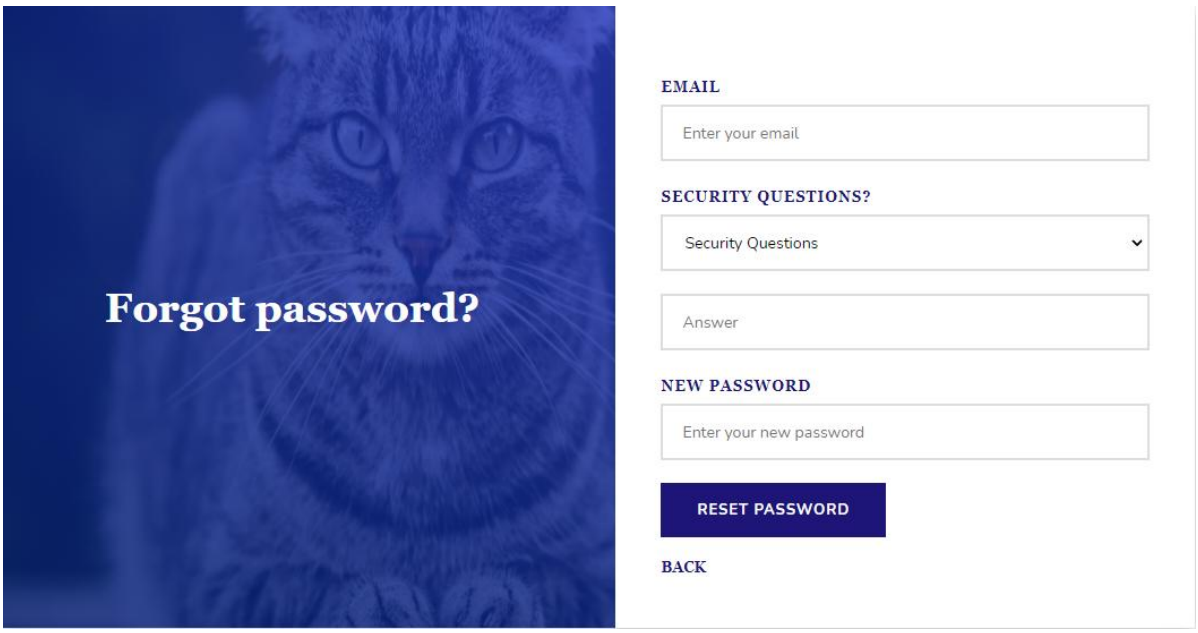
Pet Shop Registration:



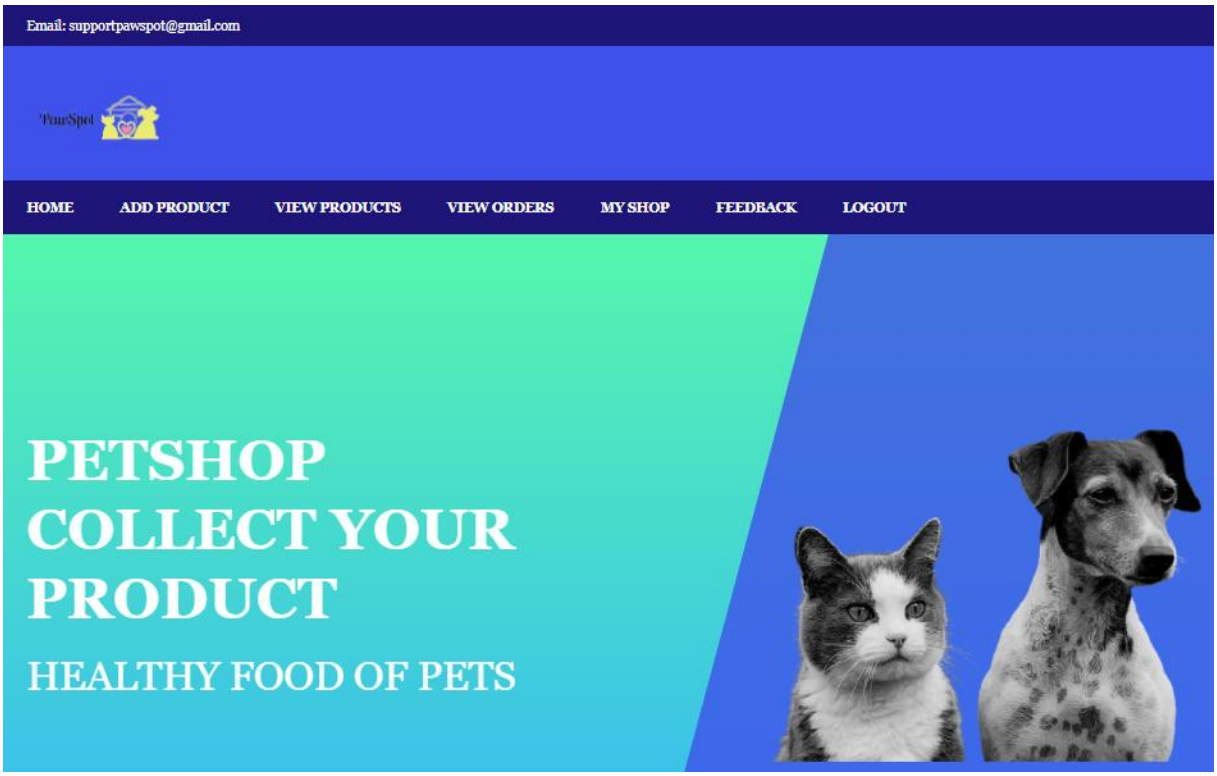
Pet Shop Login:



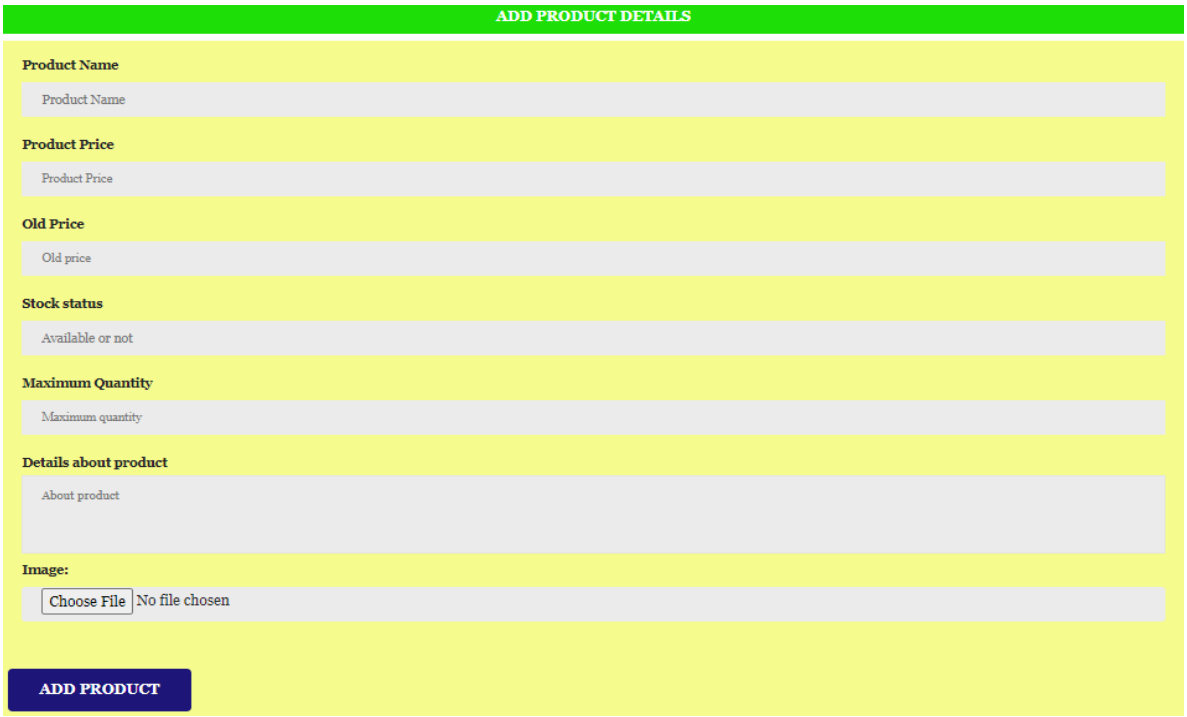
Pet Shop Forgot Password:



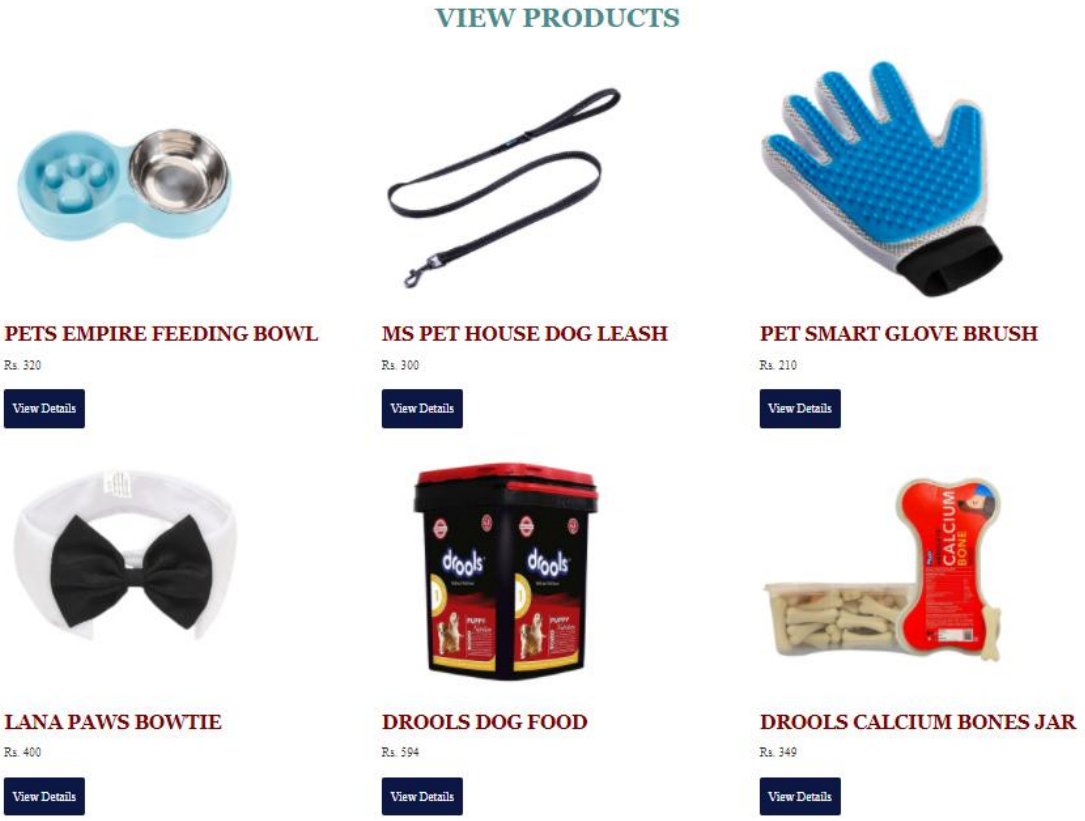
Pet Shop Home Page:



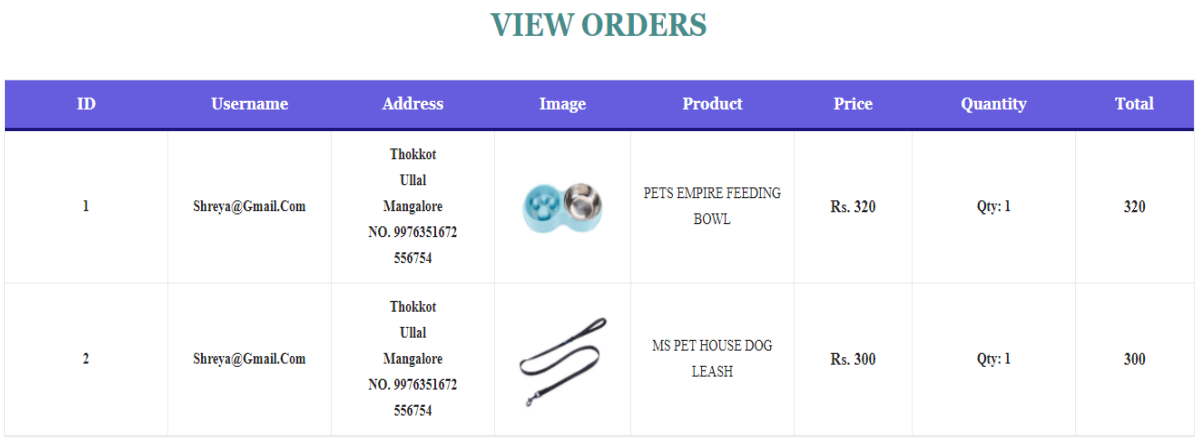
Pet Shop Add Product Page:



Pet Shop View Products Page:

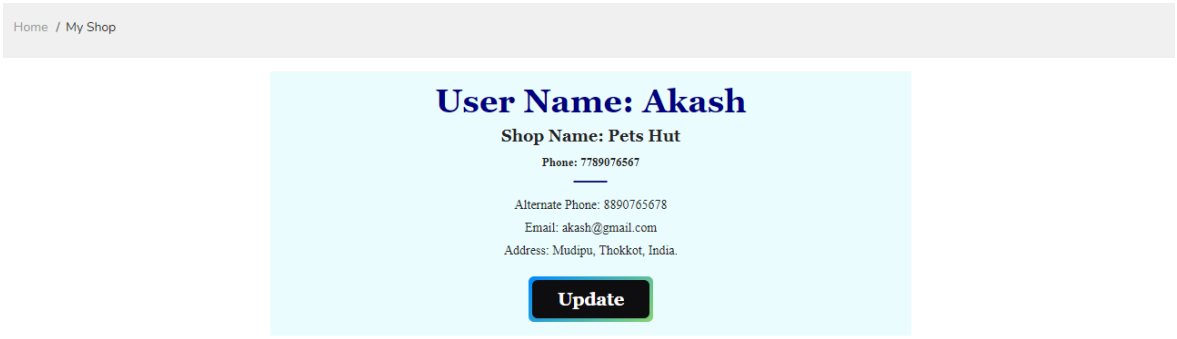


Pet Shop View Orders Page:

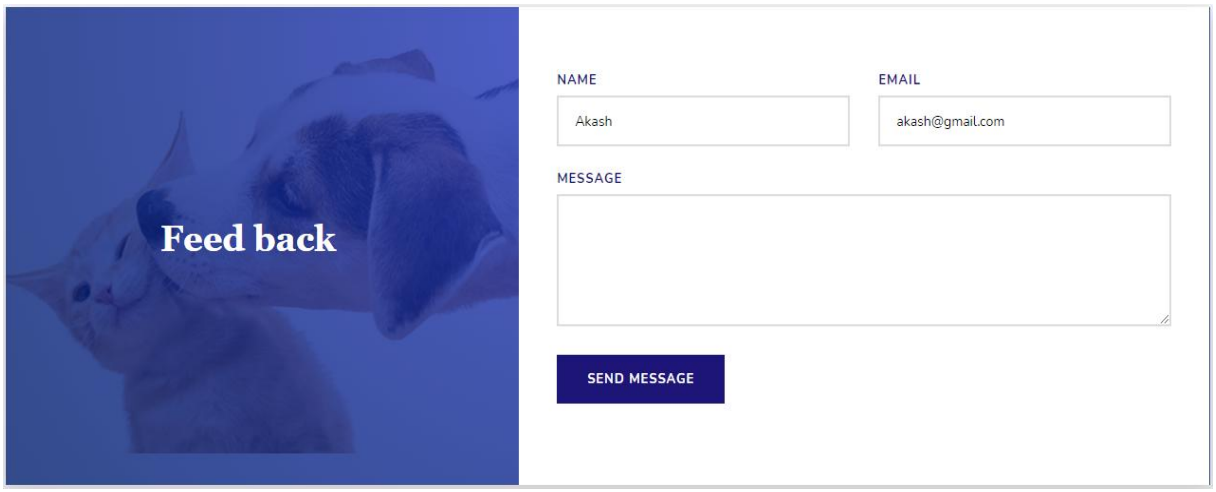


ID	Username	Address	Image	Product	Price	Quantity	Total
1	Shreya@Gmail.Com	Thokkot Ullal Mangalore NO. 9976351672 556754		PETS EMPIRE FEEDING BOWL	Rs. 320	Qty: 1	320
2	Shreya@Gmail.Com	Thokkot Ullal Mangalore NO. 9976351672 556754		MS PET HOUSE DOG LEASH	Rs. 300	Qty: 1	300

Pet Shop Profile:

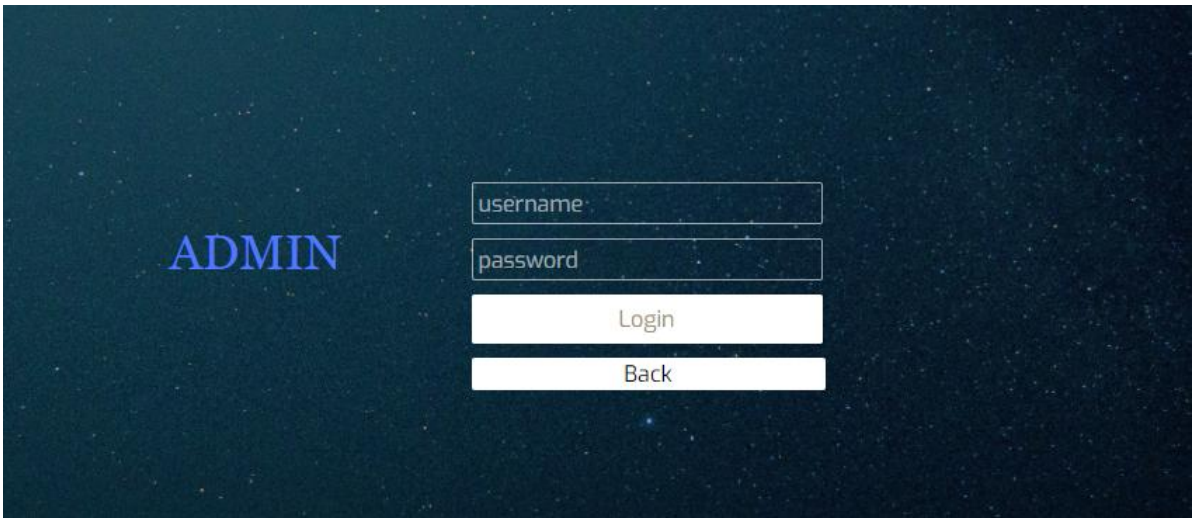


Pet Shop Feedback:

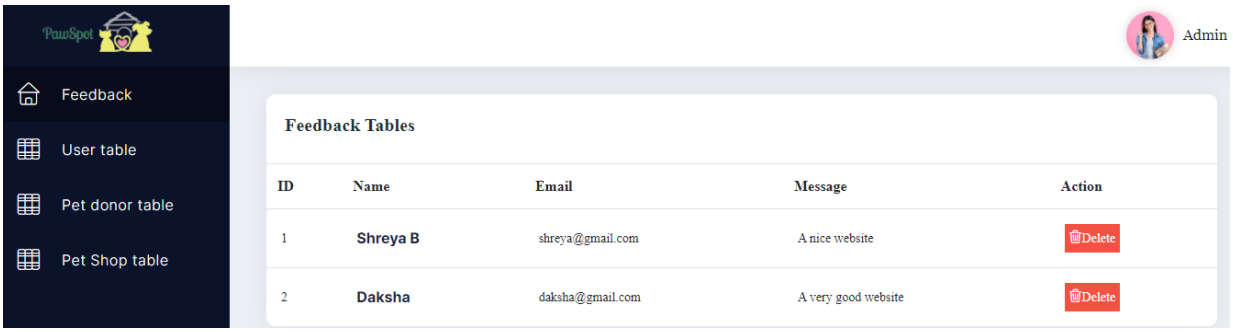


ADMIN SIDE:

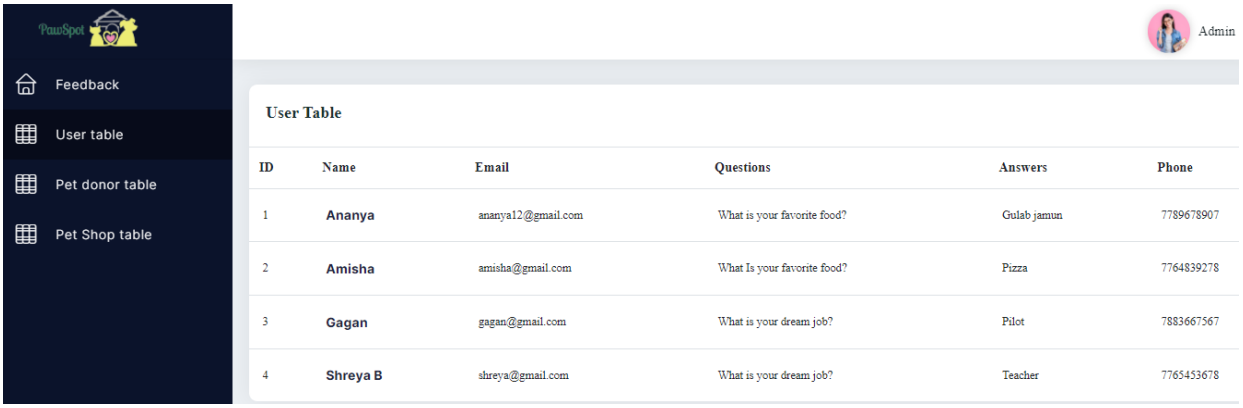
Login Page:



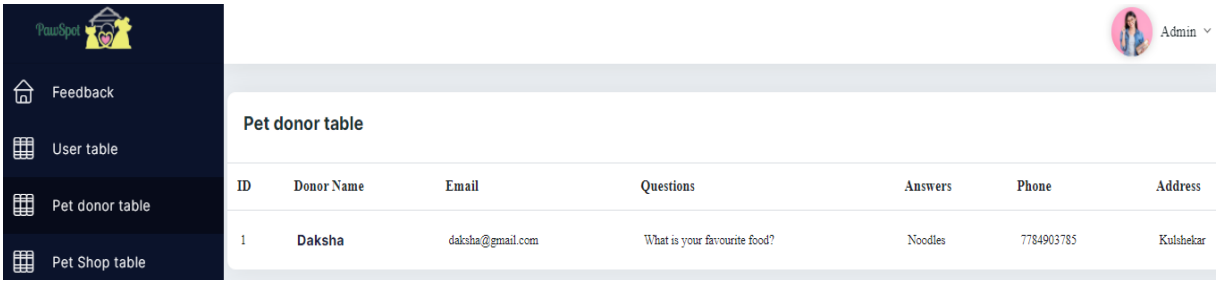
View Feedback Table:



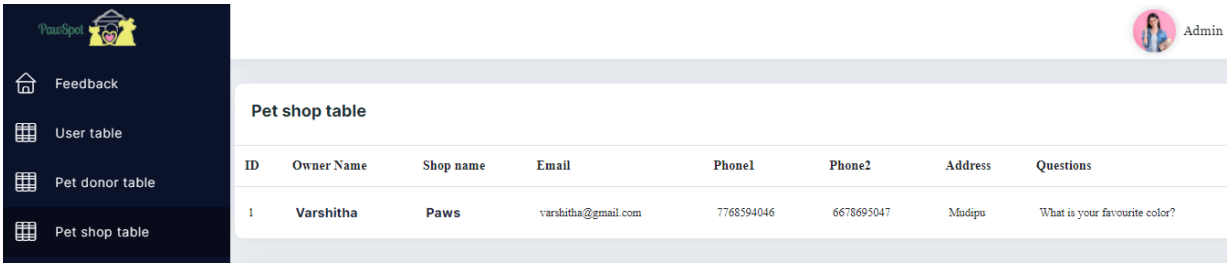
View User Table:



View Pet Donor Table:



View Pet Shop Table:



CHAPTER-9

CONCLUSION AND FUTURE ENHANCEMENTS

9.1 Conclusion

The "PAWSPOT" is the web application that aims to provide a user friendly interface to automate the process of serving towards the welfare of the pets by giving the abandoned pets a place of shelter and care them with affection. The application also gives guidelines for caring towards the pets, adoption procedure of a pet. The user can do the adoption process through the application as this process is time consuming if done manually.

9.2 Future Works

The present system is being developed as web application. Although the application can be accessed by portable devices like cell phones through browsers.

In future, we would like to develop as an android application with GPS tracking facility.

CHAPTER-10

BIBLIOGRAPHY

Websites:

1. http://stackoverflow.com
2. https://github.com/
3. https://www.w3schools.com/