VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi: 590 018

A Database Management Systems Mini Project report on

"PET STORE MANAGEMENT SYSTEM"

Submitted in partial fulfillment of the requirement for the award of Degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

By

SAGNIK SARKAR **1AY20CS131**
PRIYANSHU SINGH **1AY20CS114**

Under the guidance of

Prof. Swathy U

Prof. Sneha N P

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
ACHARYA INSTITUTE OF TECHNOLOGY
(Affiliated to Visvesvaraya Technological University, Belagavi)

2022-2023

ACHARYA INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi)
Soladevanahalli, Bangalore – 560090

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the Database Management Systems mini project entitled **"**PET STORE MANAGEMENT SYSTEM**"** is a bonafide work carried out by **SAGNIK SARKAR(1AY20CS131) &PRIYANSHU SINGH (1AY20CS114)** of 5th semester in partial fulfillment for the award of degree of **Bachelor of Engineering in Computer Science & Engineering of the Visvesvaraya Technological University**, **Belagavi**, during the year **2022-2023.** It is certified that all corrections/ suggestions indicated for internal assessments have been incorporated in the Report deposited in the departmental library. The Mini Project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for the **Bachelor of Engineering Degree**.

Signature of Guides **Signature of H.O.D**

Name of the examiners **Signature with date**

1.

2.

ABSTRACT

Pet Store Management System is a database-driven application that allows users to browse and purchase pets online, and allows admins to manage the store's inventory and process orders. The system includes several tables to store information about users, admins, pets, orders, and inventory. The users table stores information about the users of the system, including their name, email address, and password, as well as the date and time of their last login and the date and time that their account was created. The admins table stores similar information about the admins of the system. The pets table stores information about the pets that are available for sale in the store, including their species, breed, age, and price. The orders table stores information about orders placed by users, including the user's ID, the pet's ID, and the date of the order. The inventory table stores information about the store's inventory, including the pet's ID, the quantity in stock, and the minimum and maximum levels for each item. The system also includes various features and functionality to allow users to browse and search for pets, view pet details, add pets to their cart, and place orders. Admins can add, update, and delete pets, process orders, and view inventory levels and reports. Overall, the pet store management system provides a convenient and user-friendly platform for users to purchase pets and for admins to manage the store.

ACKNOWLEDGEMENT

I express my gratitude to our institution and management for providing us with good infrastructure, laboratory, facilities and inspiring staff whose gratitude was of immense help in completion of this seminar successfully.

I express my sincere gratitude to our principal, **Dr. Prakash M R** for providing required environment and valuable support for developing this mini project.

My sincere thanks to **Dr. AJITH PADYANA,** Head of the Department, Computer Science and Engineering, Acharya Institute of Technology for his valuable support and for rendering us resources for this mini project work.

I express my gratitude to **Prof. SNEHA N P, Prof. SWATHY U,** Dept. Computer Science and Engineering, Acharya Institute of Technology who guided me with valuable suggestions in completing this mini-project at every stage.

My gratitude thanks should be rendered to many people who helped me in all possible ways.

SAGNIK SARKAR

(1AY20CS131)

PRIYANSHU SINGH

(1AY20CS114)

CONTENTS

CHAPTER **PAGE NO.**

Chapter 1

Introduction

1.1 Description about Online Cab Management System

Chapter 2

SOFTWARE REQUIREMENT SPECIFICATION

2.1 **Hardware Requirements**

• Processor: Dual Core or better

• RAM: 4 GB or more

• Hard Drive: At least 1 TB of storage

• Network: High-speed wired or wireless connection

• Web Server: Apache or IIS

• Database Server: MySQL 5.6 or higher

FRONT-END:

- HTML
- Bootstrap

BACK-END:

- PHP

2.2 Software Requirements

• Operating System: Windows 7, 8, or 10

• Web Browser: Latest version of Chrome, Firefox, or Safari

• Software: Pet Store Management System

• Security: Firewall and antivirus protection

2.3 **Functional Requirements**

- New User Registration: The system should allow a new user to register by giving the required details.
- User Login: The system should allow to an existing user to login using the correct password to access the Site.
- Admin Login: The admin can login to his/her account and view the details of the bookings and also the users.
- Automatic update to database once a new user register. Also, if a user initiates a booking, it gets automatically updated to the database.

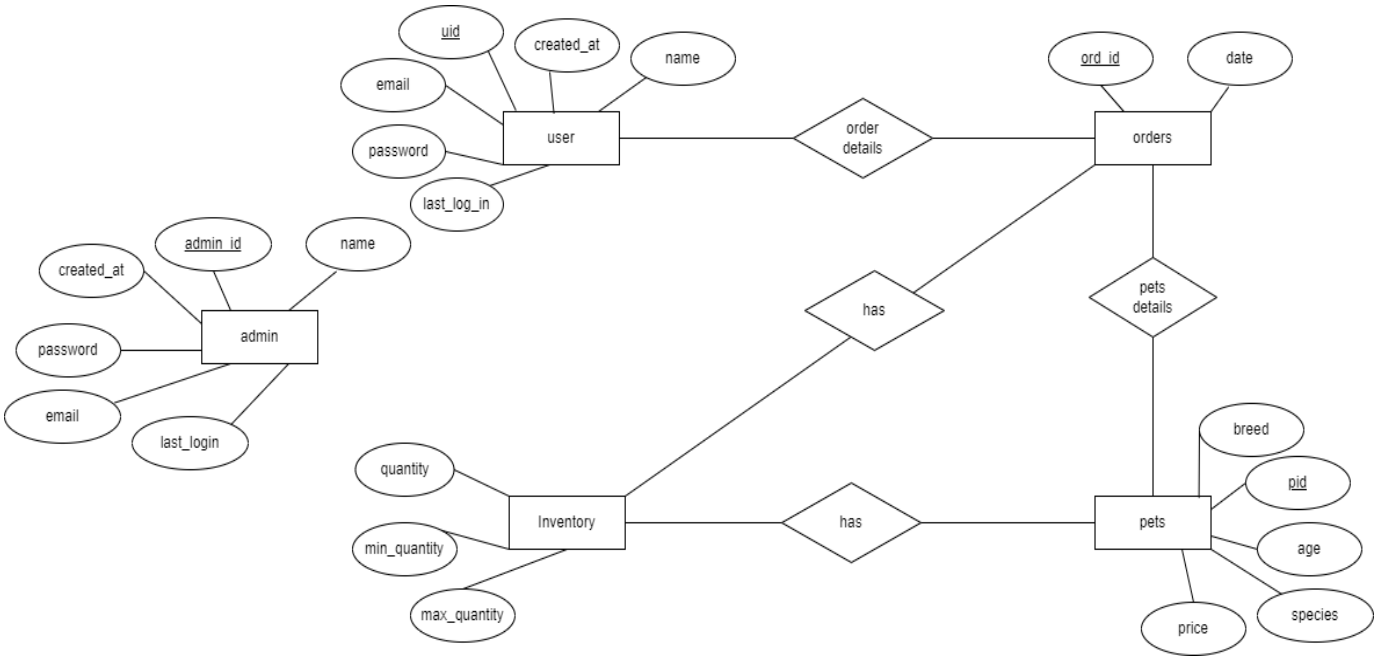
2.4 **Non-functional Requirements**

- Performance Requirements: Performance of the system should be fast and accurate. System shall handle expected and unexpected errors. Should be able to handle large amount of data.
- Safety Requirements: Must be two servers, one main server and one backup server.
- Security Requirements: User authentication and validation of members using their password. Proper accountability which includes not allowing a member to see other member's account. Only admin will see and manage all user's account. Proper user authentication should be provided.

Chapter 3

SYSTEM DESIGN

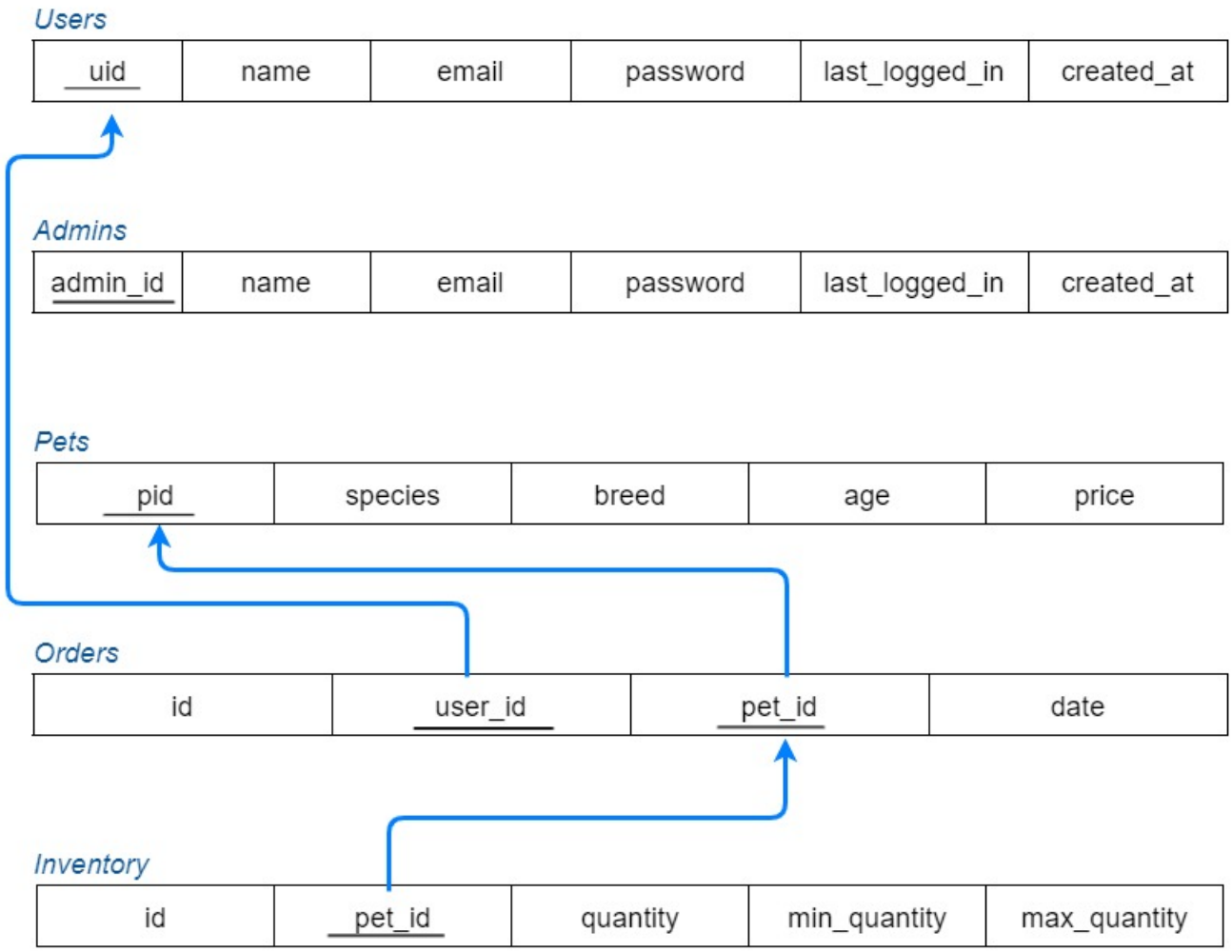
3.1 **ER Diagram**



3.2 **Schema Diagram**

A schema contains schema objects like table, primary key, foreign key, views, columns, data types, stored procedures. A database schema can be represented by using visual diagram. That diagram shows the database objects and relationships with each other. Database schema

defines its entities and the relationship among them. It contains a descriptive detail of the database which can be depicted by means of Schema diagram.



Chapter 4

IMPLEMENTATION

4.1 Creation of tables

SYNTAX:

Create table table_name (attribute_1 data_type_1, attribute_2 data_type_2,

attribute_n data_type_n,

primary key(attribute),

foreign key(attribute) references parent_table(attribute) on delete cascade);

4.1.1 USER TABLE

```
CREATE TABLE users (
  uid int(11) NOT NULL,
  name varchar(255) NOT NULL,
  email varchar(255) NOT NULL,
  password varchar(255) NOT NULL,
  last_logged_in datetime DEFAULT NULL,
  created_at datetime NOT NULL
)
```

	uid	name	email	password	last_logged_in	created_at
Edit Copy Delete	1	Michael Jordan	michael.jordan@example.com	p@$$w0rd	NULL	2023-01-15 12:33:05
Edit Copy Delete	2	Katy Perry	katy.perry@example.com	s3cureP@ss	NULL	2023-01-15 12:33:05
Edit Copy Delete	3	David Beckham	david.beckham@example.com	B0bj0hn$0n	NULL	2023-01-15 12:33:06
Edit Copy Delete	6	Priyanshu Singh	iampriyanshu20@gmail.com	MYPWD123	NULL	2023-01-15 13:09:31
Edit Copy Delete	7	Sagnik Sarkar	sagnik@gmail.com	sagnik123	NULL	2023-01-15 21:18:32

4.1.2 ADMIN TABLE

```
CREATE TABLE admins (
  admin_id int(11) NOT NULL,
  name varchar(255) NOT NULL,
  email varchar(255) NOT NULL,
  password varchar(255) NOT NULL,
  last_logged_in datetime DEFAULT NULL,
  created_at date NOT NULL DEFAULT curdate()
)
```

	admin_id	name	email	password	last_logged_in	created_at
Edit Copy Delete	1	John Smith	john.smith@example.com	p@$$w0rd	NULL	2023-01-15
Edit Copy Delete	2	Jane Doe	jane.doe@example.com	s3cureP@ss	NULL	2023-01-15
Edit Copy Delete	3	Bob Johnson	bob.johnson@example.com	B0bj0hn$0n	NULL	2023-01-15
Edit Copy Delete	4	Priyanshu Singh	iampriyanshu20@gmail.com	mypwd123	NULL	2023-01-15
Edit Copy Delete	5	Sagnik Sarkar	sagnik@gmail.com	admin@sagnik	NULL	0000-00-00

4.1.3 PETS TABLE

```
CREATE TABLE pets (
  pid int(11) NOT NULL,
  species varchar(255) NOT NULL,
  breed varchar(255) NOT NULL,
  age int(11) NOT NULL,
  price decimal(10,2) NOT NULL
)
```

	pid	species	breed	age	price
Edit Copy Delete	1	Dog	Golden Retriever	2	500.00
Edit Copy Delete	2	Dog	Labrador Retriever	1	400.00
Edit Copy Delete	3	Dog	German Shepherd	3	600.00
Edit Copy Delete	4	Cat	Siamese	2	350.00
Edit Copy Delete	5	Cat	Persian	1	300.00
Edit Copy Delete	6	Cat	Maine Coon	3	400.00
Edit Copy Delete	7	Fish	Goldfish	1	10.00
Edit Copy Delete	8	Fish	Koi	2	20.00
Edit Copy Delete	9	Fish	Tetra	3	5.00
Edit Copy Delete	10	Bird	Parakeet	1	50.00
Edit Copy Delete	11	Bird	Cockatiel	2	40.00

4.1.4 INVENTORY TABLE

```
CREATE TABLE inventory (
  id int(11) NOT NULL,
 pet_id int(11) NOT NULL,
  quantity int(11) NOT NULL,
  min_quantity int(11) NOT NULL,
  max_quantity int(11) NOT NULL
)
```

	id	pet_id	quantity	min_quantity	max_quantity
Edit Copy Delete	1	1	49	20	100
Edit Copy Delete	2	2	30	10	50
Edit Copy Delete	3	3	40	20	80
Edit Copy Delete	4	4	25	10	40
Edit Copy Delete	5	5	45	20	80
Edit Copy Delete	6	6	35	15	60
Edit Copy Delete	7	7	20	10	30
Edit Copy Delete	8	8	30	15	50
Edit Copy Delete	9	9	50	25	80
Edit Copy Delete	10	10	40	20	70
Edit Copy Delete	11	11	35	15	60

4.1.5 ORDERS TABLE

```
CREATE TABLE orders (
  oid int(11) NOT NULL,
  user_id int(11) NOT NULL,
  pet_id int(11) NOT NULL,
  quantity int(11) NOT NULL,
  total_price double NOT NULL,
  status varchar(10) NOT NULL DEFAULT LIVE,
  date date NOT NULL DEFAULT curdate()
)
```

	oid	user_id	pet_id	quantity	total_price	status	date
Edit Copy Delete	1	6	4	2	700	CANCELED	0000-00-00
Edit Copy Delete	2	3	3	5	3000	LIVE	2023-01-15
Edit Copy Delete	3	6	4	3	1050	CANCELED	2023-01-15
Edit Copy Delete	12	6	4	1	350	CANCELED	2023-01-15
Edit Copy Delete	13	6	4	1	350	LIVE	2023-01-15
Edit Copy Delete	14	6	7	1	10	LIVE	2023-01-15
Edit Copy Delete	15	6	7	1	10	LIVE	2023-01-15
Edit Copy Delete	16	6	7	3	30	LIVE	2023-01-15
Edit Copy Delete	17	7	1	1	500	CANCELED	2023-01-15

4.2 Insertion of values

SYNTAX:

Insert into table_name values(v1, v2, v3, … vn);

4.2.1 USERS TABLE

INSERT INTO users VALUES

(1, 'Michael Jordan', 'michael.jordan@example.com', 'p@$$w0rd', NULL, '2023-01-15 12:33:05'),

(2, 'Katy Perry', 'katy.perry@example.com', 's3cureP@ss', NULL, '2023-01-15 12:33:05'),

(3, 'David Beckham', 'david.beckham@example.com', 'B0bj0hn$0n', NULL, '2023-01-15 12:33:06'),

(6, 'Priyanshu Singh', 'iampriyanshu20@gmail.com', 'MYPWD123', NULL, '2023-01-15 13:09:31'),

(7, 'Sagnik Sarkar', 'sagnik@gmail.com', 'sagnik123', NULL, '2023-01-15 21:18:32');

select * from users;

	uid	name	email	password	last_logged_in	created_at
Edit Copy Delete	1	Michael Jordan	michael.jordan@example.com	p@$$w0rd	NULL	2023-01-15 12:33:05
Edit Copy Delete	2	Katy Perry	katy.perry@example.com	s3cureP@ss	NULL	2023-01-15 12:33:05
Edit Copy Delete	3	David Beckham	david.beckham@example.com	B0bj0hn$0n	NULL	2023-01-15 12:33:06
Edit Copy Delete	6	Priyanshu Singh	iampriyanshu20@gmail.com	MYPWD123	NULL	2023-01-15 13:09:31
Edit Copy Delete	7	Sagnik Sarkar	sagnik@gmail.com	sagnik123	NULL	2023-01-15 21:18:32

4.2.2 ADMIN TABLE

INSERT INTO admins VALUES

(1, 'John Smith', 'john.smith@example.com', 'p@$$w0rd', NULL, '2023-01-15'),

(2, 'Jane Doe', 'jane.doe@example.com', 's3cureP@ss', NULL, '2023-01-15'),

(3, 'Bob Johnson', 'bob.johnson@example.com', 'B0bj0hn$0n', NULL, '2023-01-15'),

(4, 'Priyanshu Singh', 'iampriyanshu20@gmail.com', 'mypwd123', NULL, '2023-01-15'),

(5, 'Sagnik Sarkar', 'sagnik@gmail.com', 'admin@sagnik', NULL, '0000-00-00');

select * from admin;

	admin_id	name	email	password	last_logged_in	created_at
Edit Copy Delete	1	John Smith	john.smith@example.com	p@$$w0rd	NULL	2023-01-15
Edit Copy Delete	2	Jane Doe	jane.doe@example.com	s3cureP@ss	NULL	2023-01-15
Edit Copy Delete	3	Bob Johnson	bob.johnson@example.com	B0bj0hn$0n	NULL	2023-01-15
Edit Copy Delete	4	Priyanshu Singh	iampriyanshu20@gmail.com	mypwd123	NULL	2023-01-15
Edit Copy Delete	5	Sagnik Sarkar	sagnik@gmail.com	admin@sagnik	NULL	0000-00-00

4.2.2 PETS TABLE

INSERT INTO pets VALUES

(1, 'Dog', 'Golden Retriever', 2, '500.00'),

(2, 'Dog', 'Labrador Retriever', 1, '400.00'),

(3, 'Dog', 'German Shepherd', 3, '600.00'),

(4, 'Cat', 'Siamese', 2, '350.00'),

(5, 'Cat', 'Persian', 1, '300.00'),

(6, 'Cat', 'Maine Coon', 3, '400.00'),

(7, 'Fish', 'Goldfish', 1, '10.00'),

(8, 'Fish', 'Koi', 2, '20.00'),

(9, 'Fish', 'Tetra', 3, '5.00'),

(10, 'Bird', 'Parakeet', 1, '50.00'),

(11, 'Bird', 'Cockatiel', 2, '40.00'),

(12, 'Bird', 'African Grey', 3, '300.00'),

(13, 'Dog', 'Pomeranian', 0, '600.00');

select * from pets;

	pid	species	breed	age	price
Edit Copy Delete	1	Dog	Golden Retriever	2	500.00
Edit Copy Delete	2	Dog	Labrador Retriever	1	400.00
Edit Copy Delete	3	Dog	German Shepherd	3	600.00
Edit Copy Delete	4	Cat	Siamese	2	350.00
Edit Copy Delete	5	Cat	Persian	1	300.00
Edit Copy Delete	6	Cat	Maine Coon	3	400.00
Edit Copy Delete	7	Fish	Goldfish	1	10.00
Edit Copy Delete	8	Fish	Koi	2	20.00
Edit Copy Delete	9	Fish	Tetra	3	5.00
Edit Copy Delete	10	Bird	Parakeet	1	50.00
Edit Copy Delete	11	Bird	Cockatiel	2	40.00

4.2.3 ORDERS TABLE

INSERT INTO orders VALUES

(1, 6, 4, 2, 700, 'CANCELED', '0000-00-00'),

(2, 3, 3, 5, 3000, 'LIVE', '2023-01-15'),

(3, 6, 4, 3, 1050, 'CANCELED', '2023-01-15'),

(12, 6, 4, 1, 350, 'CANCELED', '2023-01-15'),

(13, 6, 4, 1, 350, 'LIVE', '2023-01-15'),

(14, 6, 7, 1, 10, 'LIVE', '2023-01-15'),

(15, 6, 7, 1, 10, 'LIVE', '2023-01-15'),

(16, 6, 7, 3, 30, 'LIVE', '2023-01-15'),

(17, 7, 1, 1, 500, 'CANCELED', '2023-01-15');

select * from orders;

	oid	user_id	pet_id	quantity	total_price	status	date
Edit Copy Delete	1	6	4	2	700	CANCELED	0000-00-00
Edit Copy Delete	2	3	3	5	3000	LIVE	2023-01-15
Edit Copy Delete	3	6	4	3	1050	CANCELED	2023-01-15
Edit Copy Delete	12	6	4	1	350	CANCELED	2023-01-15
Edit Copy Delete	13	6	4	1	350	LIVE	2023-01-15
Edit Copy Delete	14	6	7	1	10	LIVE	2023-01-15
Edit Copy Delete	15	6	7	1	10	LIVE	2023-01-15
Edit Copy Delete	16	6	7	3	30	LIVE	2023-01-15
Edit Copy Delete	17	7	1	1	500	CANCELED	2023-01-15

4.2.3 INVENTORY TABLE

```
INSERT INTO inventory  VALUES
(1, 1, 49, 20, 100),
(2, 2, 30, 10, 50),
(3, 3, 40, 20, 80),
(4, 4, 25, 10, 40),
(5, 5, 45, 20, 80),
(6, 6, 35, 15, 60),
(7, 7, 20, 10, 30),
(8, 8, 30, 15, 50),
(9, 9, 50, 25, 80),
(10, 10, 40, 20, 70),
(11, 11, 35, 15, 60);
```

	id	pet_id	quantity	min_quantity	max_quantity
Edit Copy Delete	1	1	49	20	100
Edit Copy Delete	2	2	30	10	50
Edit Copy Delete	3	3	40	20	80
Edit Copy Delete	4	4	25	10	40
Edit Copy Delete	5	5	45	20	80
Edit Copy Delete	6	6	35	15	60
Edit Copy Delete	7	7	20	10	30
Edit Copy Delete	8	8	30	15	50
Edit Copy Delete	9	9	50	25	80
Edit Copy Delete	10	10	40	20	70
Edit Copy Delete	11	11	35	15	60

4.3 Queries

1. Display inventory details of those whoes pid is 3.
Ans: SELECT * FROM inventory WHERE pet_id = '3';

Output:

	id	pet_id	quantity	min_quantity	max_quantity
Edit Copy Delete	3	3	40	20	80

2.Update inventory in the database whose id 3.
Ans: UPDATE inventory SET quantity = quantity + $add_quantity WHERE id='$id';

Output:

Edit Copy Delete	3	3	50	20	80

3. Update order in the database to CANCEL whoes ord is 3
Ans:UPDATE orders SET status = 'CANCELLED' WHERE oid='3';

Output:

☐	Edit	Copy	Delete	3	6	4	3	1050	CANCELLED 2023-01-15

4. Delete the customer from the database
Ans: DELETE FROM users WHERE uid = '2';

Output:

	uid	name	email	password	last_logged_in	created_at
Edit Copy Delete	1	Michael Jordan	michael.jordan@example.com	p@$$w0rd	NULL	2023-01-15 12:33:05
Edit Copy Delete	3	David Beckham	david.beckham@example.com	B0bj0hn$0n	NULL	2023-01-15 12:33:06
Edit Copy Delete	6	Priyanshu Singh	iampriyanshu20@gmail.com	MYPWD123	NULL	2023-01-15 13:09:31
Edit Copy Delete	7	Sagnik Sarkar	sagnik@gmail.com	sagnik123	NULL	2023-01-15 21:18:32

5. Delete the inventory from the database
Ans: DELETE FROM inventory WHERE id = '4';

Output:

	id	pet_id	quantity	min_quantity	max_quantity
☐ Click the drop-down arrow to toggle column's visibility.	1	1	49	20	100
☐ Edit Copy Delete	2	2	30	10	50
☐ Edit Copy Delete	3	3	50	20	80
☐ Edit Copy Delete	5	5	45	20	80
☐ Edit Copy Delete	6	6	35	15	60
☐ Edit Copy Delete	7	7	20	10	30
☐ Edit Copy Delete	8	8	30	15	50
☐ Edit Copy Delete	9	9	50	25	80

6. Delete the pet from the database

Ans: `DELETE FROM pets WHERE pid = '2';`

	pid	species	breed	age	price
Edit Copy Delete	1	Dog	Golden Retriever	2	500.00
Edit Copy Delete	3	Dog	German Shepherd	3	600.00
Edit Copy Delete	4	Cat	Siamese	2	350.00
Edit Copy Delete	5	Cat	Persian	1	300.00
Edit Copy Delete	6	Cat	Maine Coon	3	400.00
Edit Copy Delete	7	Fish	Goldfish	1	10.00

';

Output:

7. Update inventory in the database

Ans: UPDATE inventory SET min_quantity='40', max_quantity='50' WHERE id='5';

Output:

	id	pet_id	quantity	min_quantity	max_quantity
Edit Copy Delete	1	1	49	20	100
Edit Copy Delete	3	3	50	20	80
Edit Copy Delete	5	5	45	40	50
Edit Copy Delete	6	6	35	15	60
Edit Copy Delete	7	7	20	10	30

8. Update pet in the database

Ans: UPDATE pets SET breed='$breed', species='$species', age='$age', price='$price' WHERE pid='$pid';

Output:

	pid	species	breed	age	price
Edit Copy Delete	1	Dog	Golden Retriever	2	500.00
Edit Copy Delete	3	Dog	German Shepherd	3	600.00
Edit Copy Delete	4	Cat	Siamese	2	350.00
Edit Copy Delete	5	Cat	Persian	1	300.00
Edit Copy Delete	6	Cat	Maine Coon	3	400.00
Edit Copy Delete	7	greman	Dog	10	100.00

4.4 Triggers

A database trigger is procedural code that is automatically executed in response to certain events on a particular table or view in a database. The trigger is mostly used for maintaining the integrity of the information on the database. A trigger is a stored procedure in database which automatically invokes whenever a special events in the database occurs.

For example, a trigger can be invoked when a row is inserted into a specified table or when certain table columns are being updated.

SYNTAX:

Create trigger [trigger_name]

[before | after] {insert | update | delete} on [table_name]

[for each row]

[trigger_body]

4.5 Stored Procedure

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 are useful in following circumstances:

- If a database program is needed by several applications, it can be stored at server and invoked by any of the application programs. This reduces duplication of effort and improves software modularity.
- Executing program at the server can reduce data transfer and communication cost between the client and server in certain situations.

These procedures can enhance the modelling power provided by views by allowing more complex types of derived data to be made available to the database users.

Chapter 5

RESULTS AND DISCUSSION

The following figure shows the first page of Pet Store Management System. A new user can click on the sign-up option and fill up the required fields to make a new account and after agreeing to the terms and conditions a new account is created .

If you are already a user then you can login by clicking on the Login option and entering the respective user name and password used at the time of sign up.

The data entered by a new user will automatically get stored in the database.

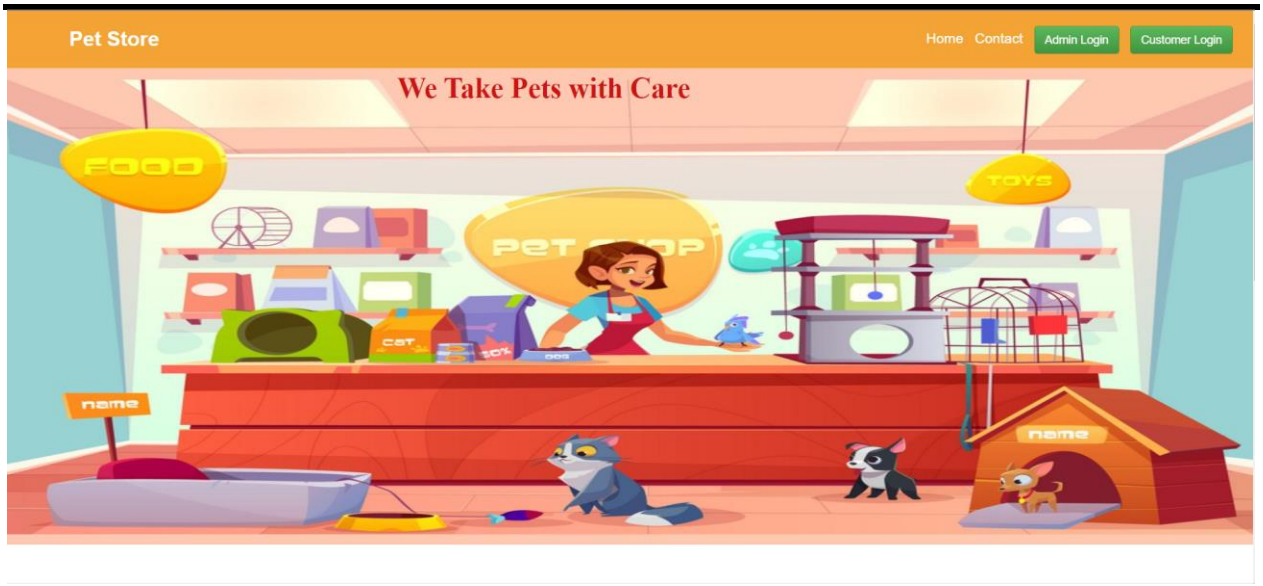


Fig:5.1

Shown below is the sign-up page (Fig 5.2). You have to provide proper details to sign up successfully.

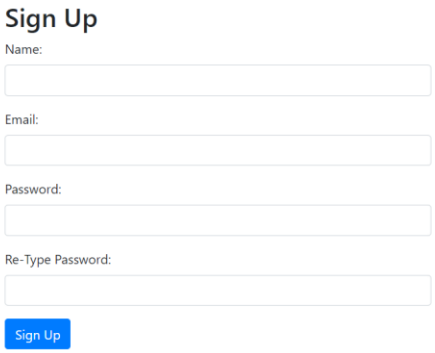


Fig: 5.2

Shown below is the login page (Fig 5.3). You have to give correct email id and password to login successfully.

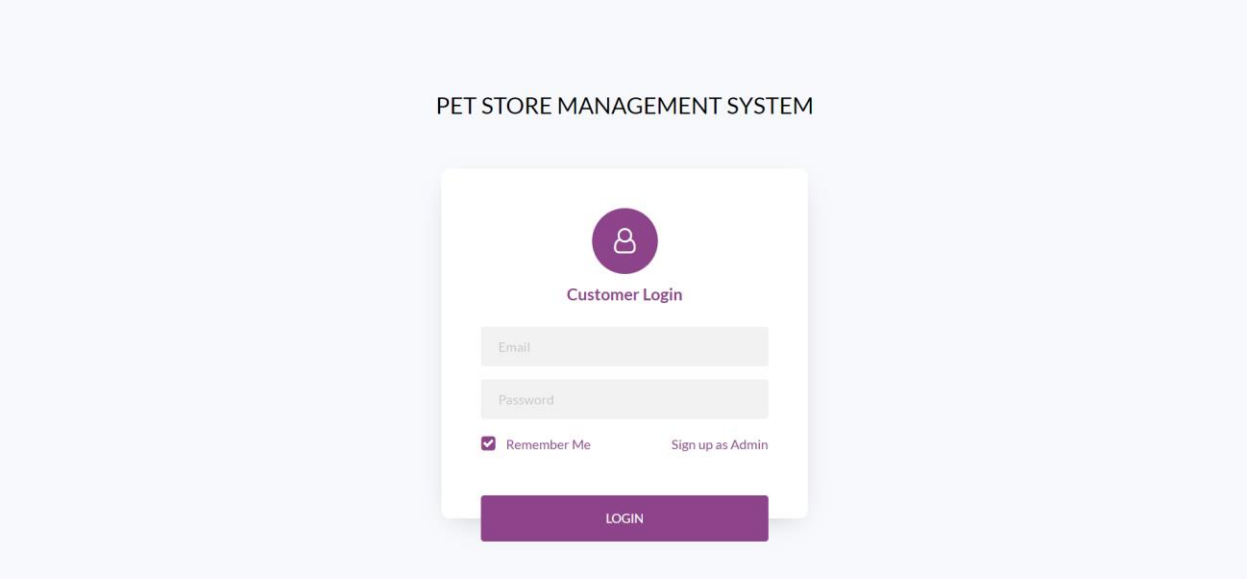


Fig: 5.3

Now if we login to a valid user. For example, I logged in customer account. I will see the following. (Fig 5.4).

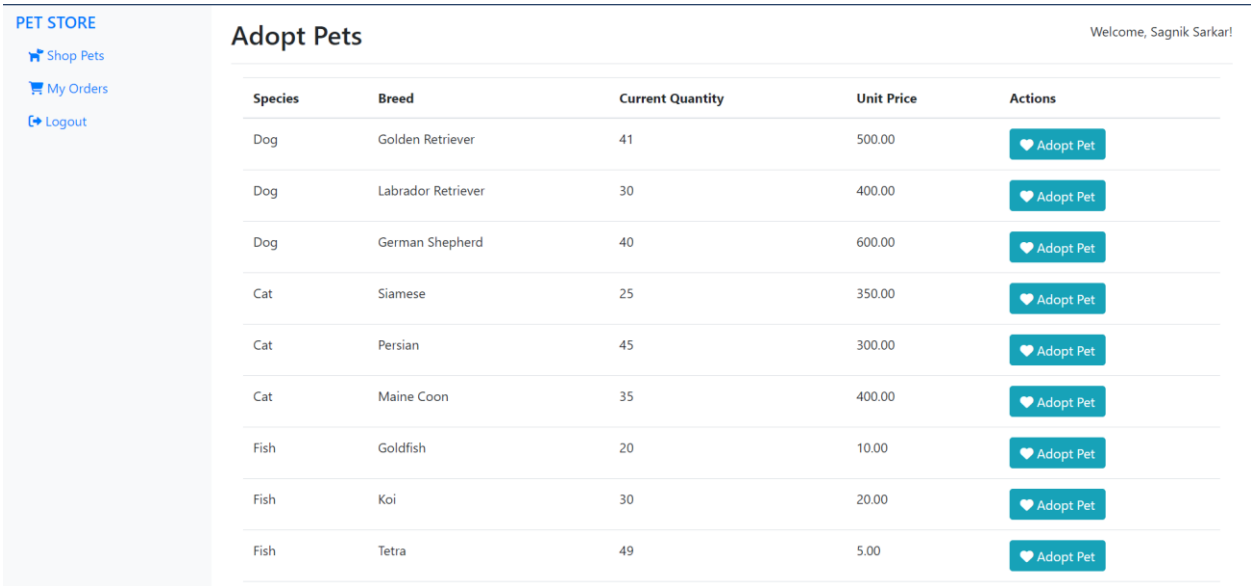


Fig: 5.4

Now, if the user wants to adopt pets then he/she can click on the Adopt Pet option and can adopt the pet . (Fig: 5.5)

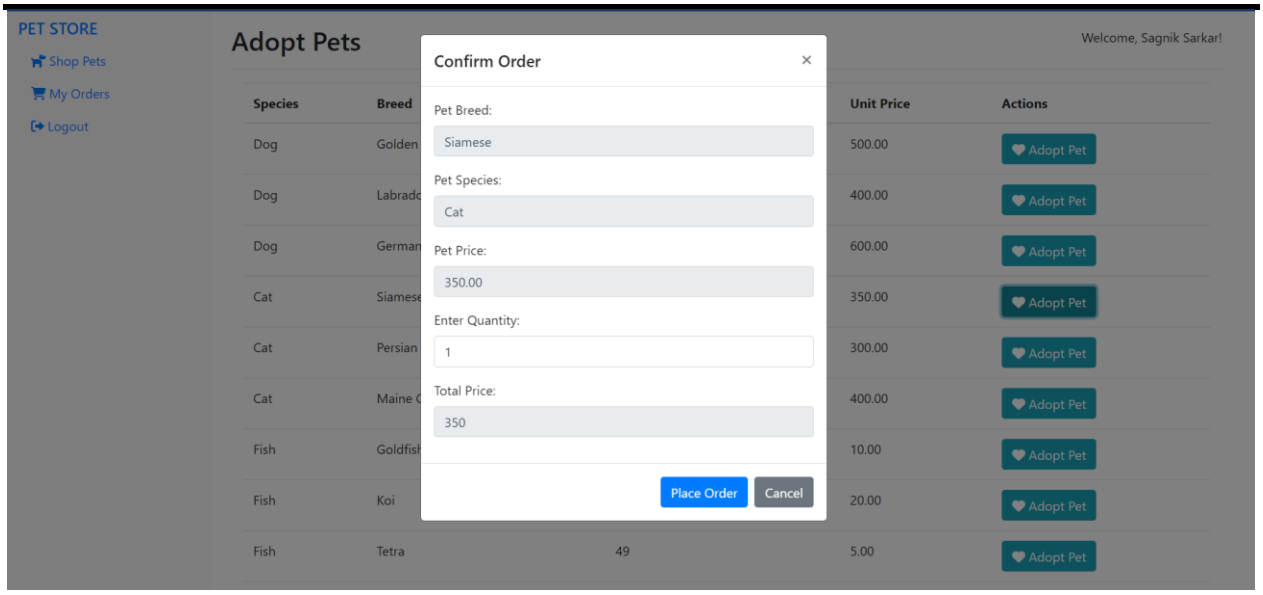


Fig: 5.5

Now, if the customer wants to see his/her order by clicking on My Orders he/she can view.(Fig: 5.6)

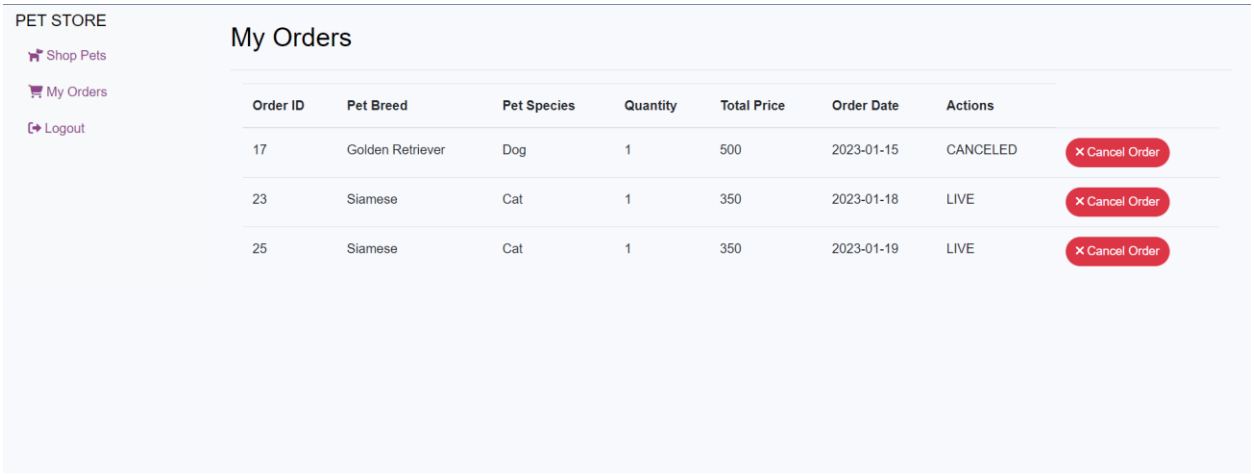


Fig: 5.6

If user wants to cancel the order he/she can click on cancel button and action will be updated to cancel from live(Fig: 5.7)

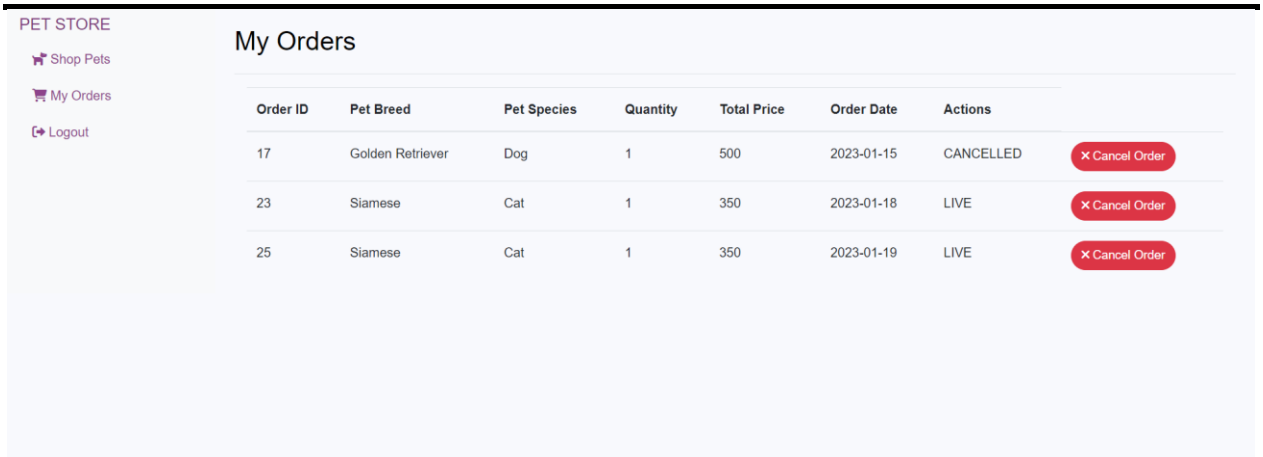


Fig: 5.7

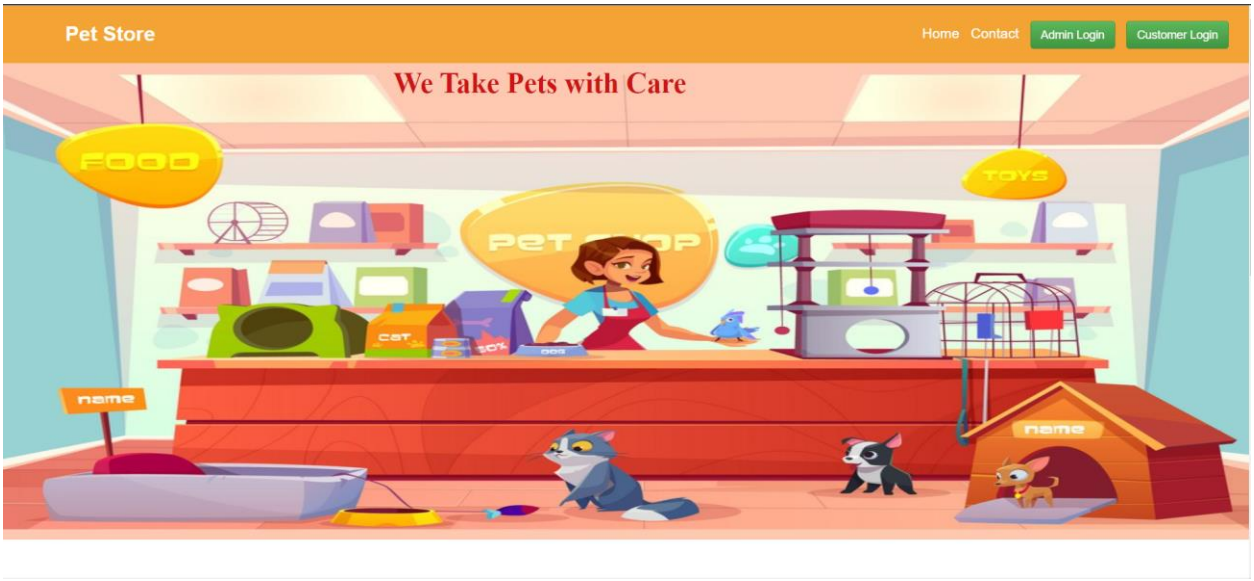


Fig:5.8

Finally, If the user clicks on log out, then he will be redirected to the first page.

Chapter 6

CONCLUSION AND FUTURE ENHANCEMENT

BIBLIOGRAPHY

The following are the details of sites and books referred which have details regarding the creation of front-end and back-end database:

- Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.
- Database Management System, Ramakrishnan and Gehrke, 3rd Edition , 2014, McGraw Hill
- Silberschatz Korth and Sudharshan, Database System Concepts, 6th Edition, McGraw Hill, 2013

Web References:

- Introduction, Conclusion, Definations , www.google.com
- Definations, Examples, www.wikipedia.org