SRI KRISHNA COLLEGE OF TECHNOLOGY
(An Autonomous Institution)
Approved by AICTE | Affiliated to Anna University Chennai|
Accredited by NBA - AICTE| Accredited by NAAC with ‘A’ Grade
KOVAIPUDUR, COIMBATORE 641042

PETS SELLING APPLICATION

A PROJECT REPORT

Submitted by

SACHIN VIPU M D	-	727822TUIT201
SANDHANA KRISHNAN S	-	727822TUIT202
SANGAMITHRA U	-	727822TUIT203
SANJAY KUMAR G	-	727822TUIT204

In partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

JULY 2024

BONAFIDE CERTIFICATE

Certified that this project report **PETS SELLING SYSTEM** is the bonafide work of **SACHIN VIPU M D 727822TUIT201, SANDHANA KRISHNAN S 727822TUIT202, SANGAMITHRA U 727822TUIT201, SANJAY KUMAR G 727822TUIT204** who carried out the project work under my supervision.

SIGNATURE	*SIGNATURE*
Dr. K . SURESH KUMAR	**Dr. RAJESH KUMAR**
SUPERVISOR	**HEAD OF THE DEPARTMENT**
Assistant Professor,	Associate Professor,
Department of Information Technology,	Department of Information Technology,
Sri Krishna College of Technology, Coimbatore-641042	Sri Krishna College of Technology, Coimbatore-641042

Certified that the candidate was examined by me in the Project Work Viva Voce examination held on__________________at Sri Krishna College of Technology, Coimbatore-641042.

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

First and foremost we thank the **Almighty** for being our light and for showering his gracious blessings throughout the course of this project.

We express our gratitude to our beloved Principal, **Dr. M.G. Sumithra,** for providing all facilities.

We are grateful to our beloved Head, Computing Sciences **Dr.T. Senthilnathan,** for her tireless and relentless support.

With the grateful heart, our sincere thanks to our Head of the Department **Dr. Rajesh Kumar**, Department of Information Technology for the motivation and all support to complete the project work.

We thank **Dr. K . Suresh Kumar**, Assistant Professor, Department of Information Technology, for his motivation and support.

We are thankful to all the **Teaching and Non-Teaching Staff** of Department of Information Technology and to all those who have directly and indirectly extended their help to us in completing this project work successfully.

We extend our sincere thanks to our **family members** and our beloved **friends**, who had been strongly supporting us in all our endeavour

ABSTRACT

ABSTRACT

The Furry Finds website stands as a robust and inclusive platform dedicated to the buying and selling of pets and pet-related products, bringing together a diverse community of pet owners, breeders, sellers, and animal enthusiasts. This online marketplace is thoughtfully designed to provide a secure, user-friendly environment where individuals can explore a broad array of pet categories, including dogs, cats, birds, reptiles, and small mammals. Each category is equipped with detailed profiles, encompassing essential information such as health records, lineage, and behavioral traits, enabling prospective pet owners to make informed decisions.

A standout feature of Furry Finds is its sophisticated search and filtering system, allowing users to easily navigate through thousands of listings to find the perfect pet or pet product tailored to their needs. The platform ensures secure transactions through integrated payment gateways, offering peace of mind to both buyers and sellers. User profiles on the platform include comprehensive reviews and ratings, fostering a trustworthy community where interactions are transparent and reliable.

Beyond transactions, Furry Finds is committed to promoting responsible pet ownership. Additionally, the website emphasizes the importance of pet adoption and rescue, partnering with shelters and rescue organizations to provide users with options to adopt or foster animals in need of homes.

The website's design is optimized for mobile devices, ensuring that users can enjoy a seamless and intuitive browsing experience whether they are on a smartphone, tablet, or desktop. With its comprehensive approach, Furry Finds not only facilitates the buying and selling of pets but also fosters a knowledgeable and caring community of pet lovers.

TABLE OF CONTENTS

TABLE OF CONTENT

LIST OF FIGURES

LIST OF FIGURES

LIST OF ABBREVIATIONS

LIST OF ABBREVIATIONS

ABBREVIATIONS	**ACRONYMS**
HTML	HYPERTEXT MARKUP LANGUAGE
CSS	CASCADING STYLESHEET
JS	JAVASCRIPT

INTRODUCTION

CHAPTER 1

INTRODUCTION

This project is a development of FurryFinds, a comprehensive online platform designed for pet enthusiasts and prospective pet owners. FurryFinds is dedicated to providing a seamless and engaging experience for users looking to adopt or purchase pets, as well as for sellers seeking to reach a broader audience. Our platform offers a wide range of services, from detailed pet profiles and seller information to an efficient and user-friendly payment system.

At FurryFinds, we understand that finding the perfect pet involves more than just browsing through listings. That's why we offer a variety of features to enhance the user experience, including detailed descriptions of pets, seller profiles, and interactive tools for managing cart items and making secure payments. Whether you’re interested in adopting a new furry friend or purchasing pet-related products, our platform ensures a smooth and enjoyable process from start to finish.

The growing trend of online pet adoption and purchasing has created a need for platforms that not only connect buyers and sellers but also provide a high level of transparency and security. With the integration of Stripe payment API, FurryFinds guarantees a secure transaction process, while our robust back-end system ensures efficient handling of user data and order management.

FurryFinds is designed to cater to the needs of both pet seekers and sellers. For potential pet owners, we offer detailed insights into each pet's background and the seller's information to make informed decisions. For sellers, our platform provides an opportunity to showcase their pets and products to a wider audience, backed by an easy-to-use interface and reliable support.As pet adoption and pet care continue to evolve, FurryFinds aims to stay at the forefront of these trends by continually enhancing our platform and services. Whether you’re looking to find your next pet or expand your reach as a seller, FurryFinds is here to support you every step of the way.

1.1 PROBLEM STATEMENT

How can we create a comprehensive platform that facilitates pet adoption and purchasing, ensuring a seamless experience for users while providing robust features for sellers and ensuring secure transactions?

1.2 OVERVIEW

This report outlines the development and implementation of the FurryFinds platform, a React-based e-commerce solution dedicated to pet adoption and purchasing. The platform aims to provide a unified space where users can easily browse, adopt, and purchase pets, while also offering a seamless experience for sellers to list their pets.

1.3 OBJECTIVE

The objective of the FurryFinds platform is to deliver a comprehensive online service for pet adoption and purchasing that caters to users' diverse needs. The platform aims to provide a wide selection of pets, each with detailed profiles and seller information, to help users find their ideal companions. It is designed to offer an intuitive browsing experience, allowing users to search and filter pets based on criteria such as type, age, breed, and location. Additionally, the platform will facilitate secure and efficient transactions, integrating payment processing and order management features to ensure a smooth purchase process. For sellers, FurryFinds provides tools to create effective listings with customizable profiles and detailed pet descriptions.

LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

2.1 RELATED WORKS

[1] An Analysis of E-Commerce Customization: A Literature Review by M.L. Moreno, M. Ramos, and P.A. García .

A comprehensive overview of the literature concerning e-commerce customization, discussing its challenges, benefits, trends, and issues. The authors review the theoretical and empirical evidence that describes the motivations and usage of customization, the challenges of its implementation, and the effects of customization on customer satisfaction, loyalty, and profitability.

[2] A Review of the Literature on E-Commerce Customization by A.B. de Oliveira, G.P. de Oliveira, and S.Q.M. de Oliveira.

The paper gives a comprehensive on e-commerce customization, discussing its challenges, benefits, trends, and issues. The authors analyze the theoretical and empirical evidence that describes the motivations and usage of customization, the challenges of its implementation, and the effects of customization on customer satisfaction, loyalty, and profitability.

[3] E-Commerce Customization: A Literature Review by P.J.H.M. van den Besselaar and M.P. van Leeuwen .

The comprehensive overview of the literature concerning e-commerce customization, discussing its challenges, benefits, trends, and issues. The authors analyze the theoretical and empirical evidence that describes the motivations and usage of customization, the challenges of its implementation, and the effects of customization on customer satisfaction, loyalty, and profitability.

[4] E-Commerce Customization: A Literature Review by S.G. Nandu and A.K. Nandu .

This paper provides a comprehensive overview of the literature concerning e-commerce customization, discussing its challenges, benefits, trends, and issues. The authors review the theoretical and empirical evidence that describes the motivations and usage of customization, the challenges of its implementation, and the effects of customization on customer satisfaction, loyalty, and profitability.

[5] The Impact of Customization on E-commerce: A Literature Review by M.W. Rahman and F. Rahman .

This paper reviews the literature on e-commerce customization, discussing its challenges, benefits, trends, and issues. The authors analyze the theoretical and empirical evidence that describes the motivations and usage of customization, the challenges of its implementation, and the effects of customization on customer satisfaction, loyalty, and profitability.

SYSTEM SPECIFICATION

CHAPTER 3
SYSTEM SPECIFICATION

In this chapter, we are gonna see the softwares that we have used to build the website. This chapter gives you a small description about the softwares used in the project.

3.1 VS CODE

Visual Studio Code is a source code editor developed by Microsoft for Windows, Linux, and macOS. It includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is also customizable, so users can change the editor's theme, keyboard shortcuts, and preferences.

VS Code is an excellent code editor for React projects. It is lightweight, customizable, and has a wide range of features that make it ideal for React development. It has built-in support for JavaScript, JSX, and TypeScript, and enables developers to quickly move between files and view detailed type definitions. It also has a built-in terminal for running tasks. Additionally, VS Code has an extensive library of extensions that allow developers to quickly add features like code snippets, debugging tools, and linting support to their projects.

3.2.REACT

React is a JavaScript library created by Facebook for building user interfaces. It is a component-based, declarative, and highly efficient library that is used to develop interactive UIs (user interfaces) for single page web applications. React uses a virtual DOM (Document Object Model) that makes it faster and easier to manipulate the DOM elements. It also provides declarative components that allow developers to write code that is easy to read and maintain. React also offers an extensive library of tools and components that make it easier to develop complex user interfaces.

3.3 ROUTERS IN REACT

Routers are important components in React applications. They provide the ability to navigate between different views or components of the application. React Router is the most popular library to handle routing in React applications. It provides the ability to define routes, set up links, and render components based on the current route. It also provides features like data fetching, code-splitting, and server-side rendering.

3.4 MySQL Workbench

MySQL Workbench is a unified visual development environment for MySQL databases, available for Windows, macOS, and Linux. It provides a suite of tools for database design, development, and administration. Users can visually design, model, generate, and manage databases, as well as perform SQL development and manage data.

MySQL Workbench is a valuable tool for managing databases in any project, including those involving complex data structures like those found in FurryFinds. It offers features such as an intuitive drag-and-drop interface for designing database schemas, powerful SQL query editors with syntax highlighting and auto-completion, and a visual query builder for constructing and executing SQL queries. Additionally, MySQL Workbench includes tools for database migration, performance monitoring, and data modeling, making it a comprehensive solution for managing and optimizing databases. Its robust set of features helps ensure data integrity and provides developers with the tools needed to efficiently handle database operations and schema changes.

3.5 LOCAL STORAGE

Local storage is a type of web storage for storing data on the client side of a web browser. It allows websites to store data on a user’s computer, which can then be accessed by the website again when the user returns. Local storage is a more secure alternative to cookies because it allows websites to store data without having to send it back and forth with each request. Local storage is a key-value pair storage mechanism, meaning it stores data in the form of a key and corresponding value. It is similar to a database table in that it stores data in columns and rows, except that local storage stores the data in the browser rather than in a database. Local storage is often used to store user information such as preferences and settings, or to store data that is not meant to be shared with other websites. It is also used to cache data to improve the performance of a website. Local storage is supported by all modern web browsers, including Chrome, Firefox, Safari, and Edge. It is accessible through the browser’s JavaScript API. Local storage is a powerful tool for websites to store data on the client side. It is secure, efficient, and can be used to store data that does not need to be shared with other websites.

Local Storage is a great way to improve the performance of a website by caching data. Local storage in web browsers allows website data to be stored locally on the user’s computer. It is a way of persistently storing data on the client side, which is not sent to the server with each request. This allows users to store data such as preferences, login information, and form data without needing to send it to a server. It is typically stored in a browser’s cookie file, but it can also be stored in other locations such as HTML5 Local Storage and IndexedDB. The data stored in local storage is persistent and can be accessed by the website even if the user closes the browser or navigates to another page. It is a great way for websites to store user-specific data, as it is secure, reliable, and fast. It is also a great way for developers to store data that does not need to be sent to the server with each request.One of the key benefits of using local storage is its reliability. Unlike server-side storage, which can be affected by network outages or other server issues, local storage is stored locally on the user’s machine, and so is not affected by these issues. Another advantage of local storage is its speed. Because the data is stored locally, it is accessed quickly, as there is no need to send requests to a server. This makes it ideal for storing data that needs to be accessed quickly, such as user preferences or session data. Local storage is also secure, as the data is stored on the user’s machine and not on a server. Thismeans that the data is not accessible by anyone other than the user, making it a good choice for storing sensitive information.

3.6 SPRINGBOOT

Spring Boot is a framework that simplifies the development of Java-based applications by providing auto-configuration, built-in tools, and default setups. It enables developers to create stand-alone, production-ready applications with minimal configuration by embedding web servers like Tomcat or Jetty. Spring Boot offers pre-configured starter projects for common functionalities, built-in monitoring and management features, and can be run easily from the command line or deployed with minimal setup.

PROPOSED SYSTEM

CHAPTER 4

PROPOSED SYSTEM

This chapter gives a small description about the proposed idea behind the development of our website

4.1 PROPOSED SYSTEM

The FurryFinds website is designed with a modern, user-friendly interface that is optimized for both desktop and mobile devices. The platform features a clear and intuitive structure, allowing users to easily navigate through various categories such as pet products, accessories, and seller information. The website includes user registration and login functionality to facilitate purchases and track orders.

Customers can browse through a range of customizable pet products and services, including options for pet accessories and delivery services. The site boasts a secure shopping cart system, enabling users to add items to their cart for immediate or future purchase. It supports multiple payment methods, including major credit cards, PayPal, and Apple Pay, ensuring a seamless checkout experience.

Additionally, the website integrates a robust review and rating system, allowing customers to provide feedback on products and services they have purchased. This feedback helps other users make informed decisions and contributes to overall product quality. For added convenience, the site includes installation services for pet-related products and ensures that all items are covered under a comprehensive warranty. In the event of defects or malfunctions, customers can reach out through the About Us page for support, where products will be repaired or replaced if they are not physically damaged.

4.2 ADVANTAGES

- Customization Options: Customers can select specific pet products and configurations to tailor their purchases to meet their unique needs and preferences. This includes choosing from a range of pet accessories, food, and other items to create the ideal package for their pets.

- Price Transparency: The website provides clear pricing for each item, allowing customers to see the cost of individual products as well as the total price of their selections. This transparency helps customers make informed purchasing decisions and manage their budgets effectively.

- Compatibility Assurance: FurryFinds ensures that all products and accessories are compatible with each other, reducing the risk of issues related to mismatched items. This feature is particularly useful for customers purchasing multiple items or accessories for their pets.

- Time-Saving: Customers can efficiently browse, select, and purchase pet products and accessories online, saving time compared to visiting multiple stores or websites. The streamlined shopping process allows for a quicker and more convenient purchasing experience.

- Quality Assurance: The website offers warranties and quality guarantees on all products, instilling confidence in customers regarding the reliability and durability of their purchases. This assurance helps build trust and customer satisfaction.

- Cost Savings: Purchasing pet products and accessories individually can often be more economical than buying pre-packaged or bundled items. By selecting items based on their specific needs, customers can potentially save money while still obtaining high-quality products.

METHODOLOGIES

CHAPTER 5

METHODOLOGIES

This chapter gives a small description about how our system works.

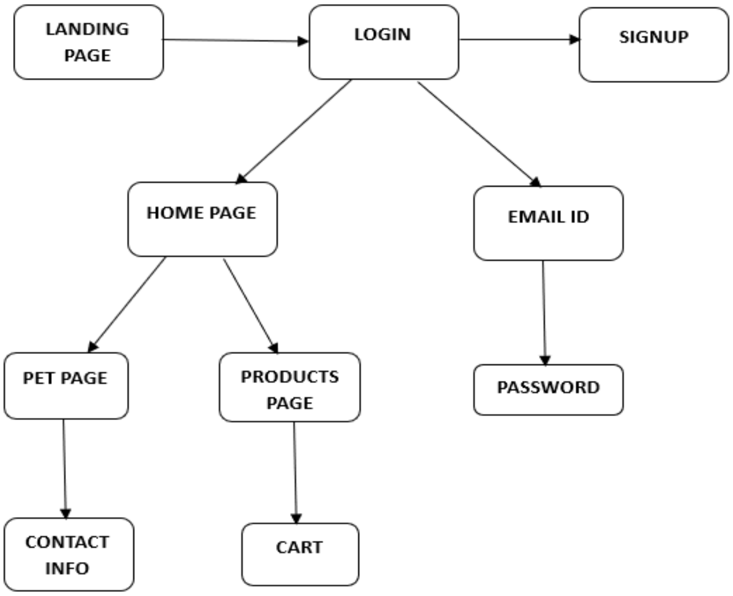


Fig 5.1 PROCESS FLOW DIAGRAM

5.1.LOGIN

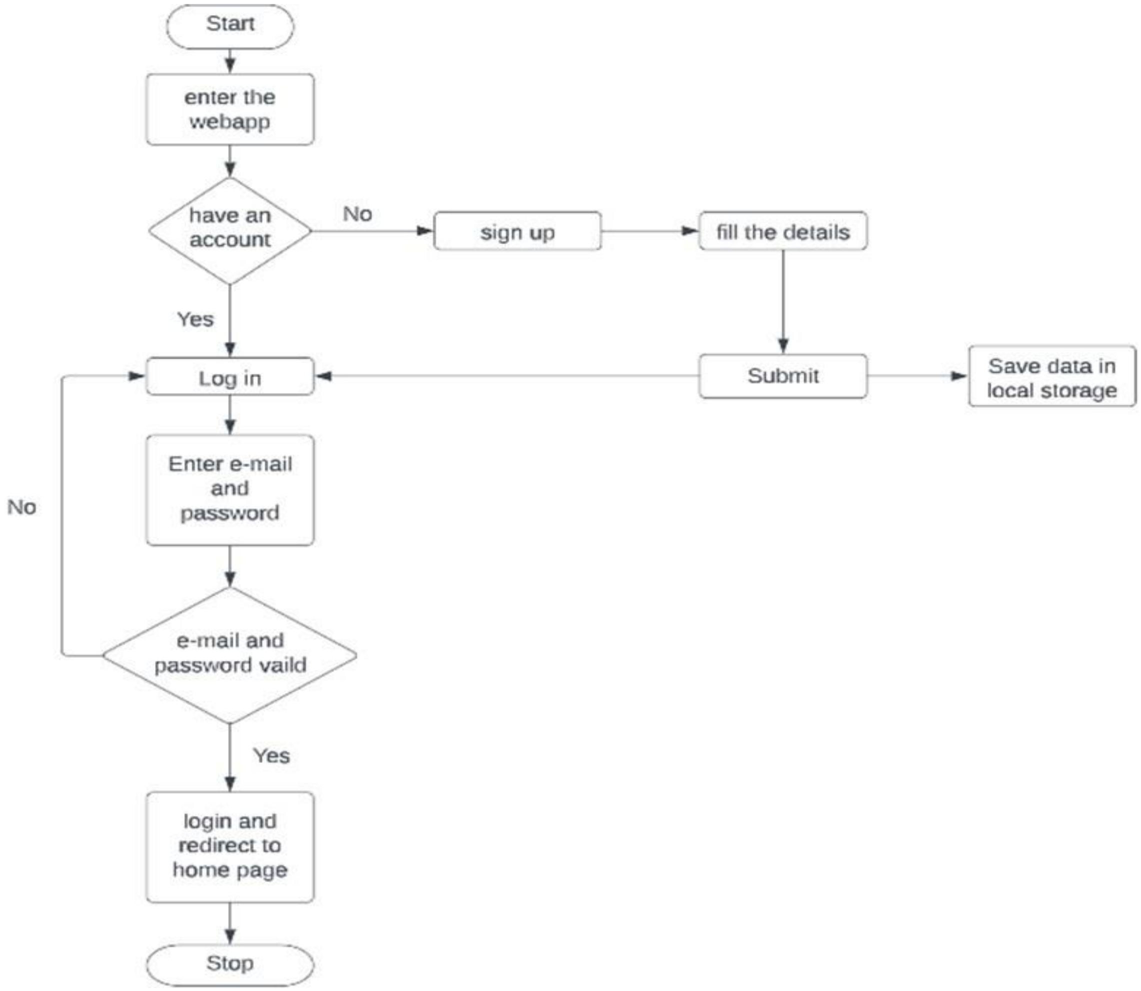

Start
enter the webapp
have an account
No
sign up
fill the details
Yes
Log in
Submit
Save data in local storage
No
Enter e-mail and password
e-mail and password vaild
Yes
login and redirect to home page
Stop

Fig 5.2. LOGIN PAGE FLOWCHART

In this page we will be asking about the username and password of the user. Firstly the website validates the user inputs. It verifies the username and password by checking it with the usernames and passwords stored in the local storage when the user creates an account in the website.

5.2 SIGNUP PAGE

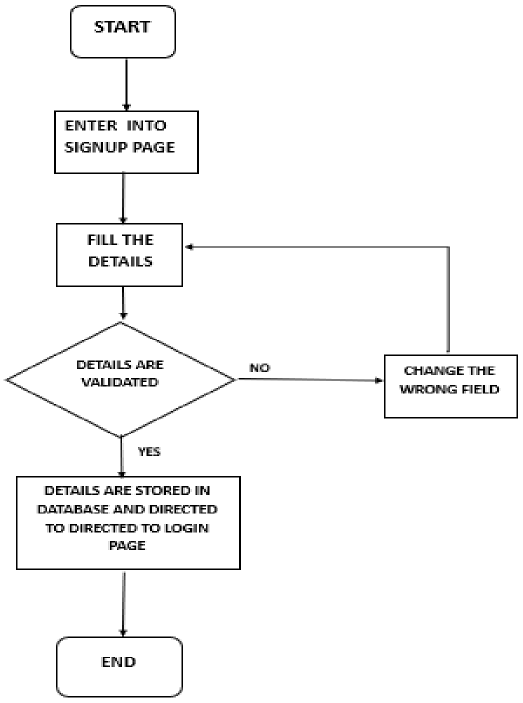


Fig 5.3 SIGNUP PAGE FLOWCHART

This page asks users about the basic details of the user to create an account. This page asks for details like username, password,email id, phone number. After the user enters the details , these details are then validated by our code . If all details are correct then the user is then directed to the login page

5.3. FORGOT PASSWORD

If an existing user forgets the password he can access a component named forgot password where he will be directed for a page that’s requesting their email id, after entering the existing email id the user will be redirected to the email OTP verification, then the user can change their password.

5.4. PRODUCTS PAGE

This page displays the list of products available for sale. These details have been displayed from the list of products available in the javascript file named data.js. So that if we want to add some products to a particular product we can easily do it by adding the details to the products js file.

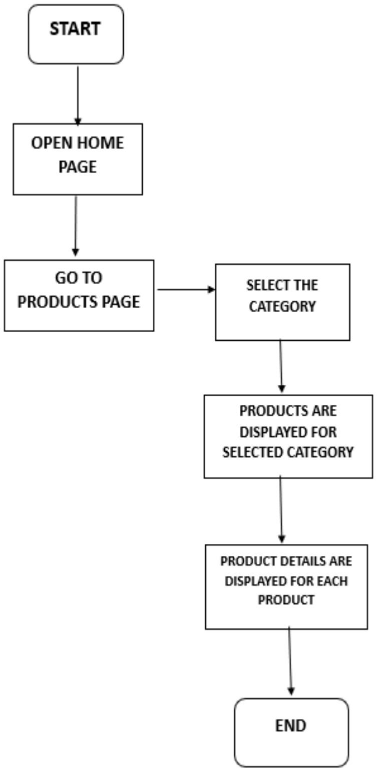

START
OPEN HOME PAGE
GO TO PRODUCTS PAGE
SELECT THE CATEGORY
PRODUCTS ARE DISPLAYED FOR SELECTED CATEGORY
PRODUCT DETAILS ARE DISPLAYED FOR EACH PRODUCT
END

Fig 5.4 PRODUCTS PAGE FLOWCHART

5.5 CART PAGE

This page displays the set of products that the user selected by clicking the button "Add to cart" available on the products page. It also displays the products with their names and its price with quantity. This also helps you to increase the quantity of the products you have added using a plus button which is available on the page after every product's name. You can also remove or reduce the quantity by using the minus symbol button which is also available below every product's name. If the cart doesn't contain anyitem it will display that the cart is empty.

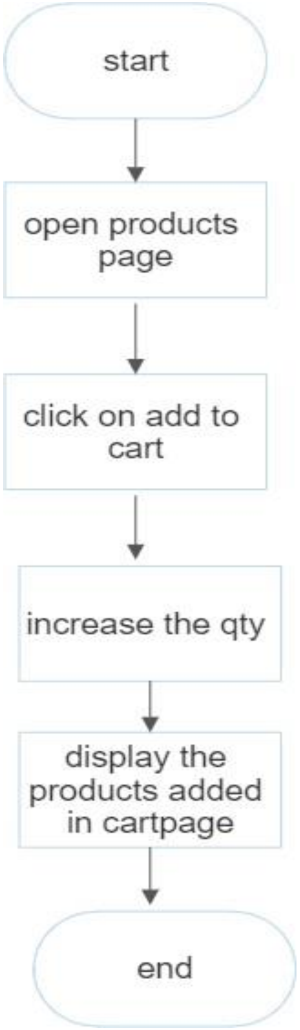


Fig 5.5 CART PAGE FLOWCHART

IMPLEMENTATION AND RESULT

CHAPTER 6

IMPLEMENTATION AND RESULT

This chapter gives a description about the output that we produced by developing the website of our idea.

6.1 LOGIN

When User enters our website he will be asked about his login details like email id and password. The login details will be verified with the details given while the user creates an account.

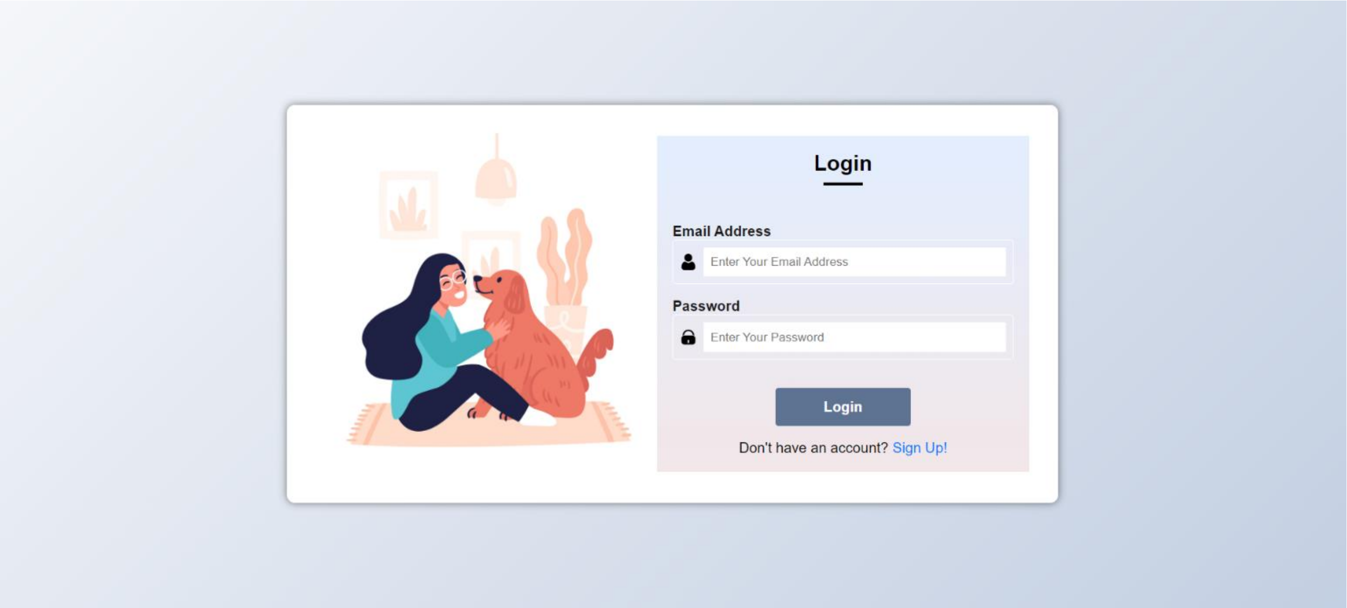
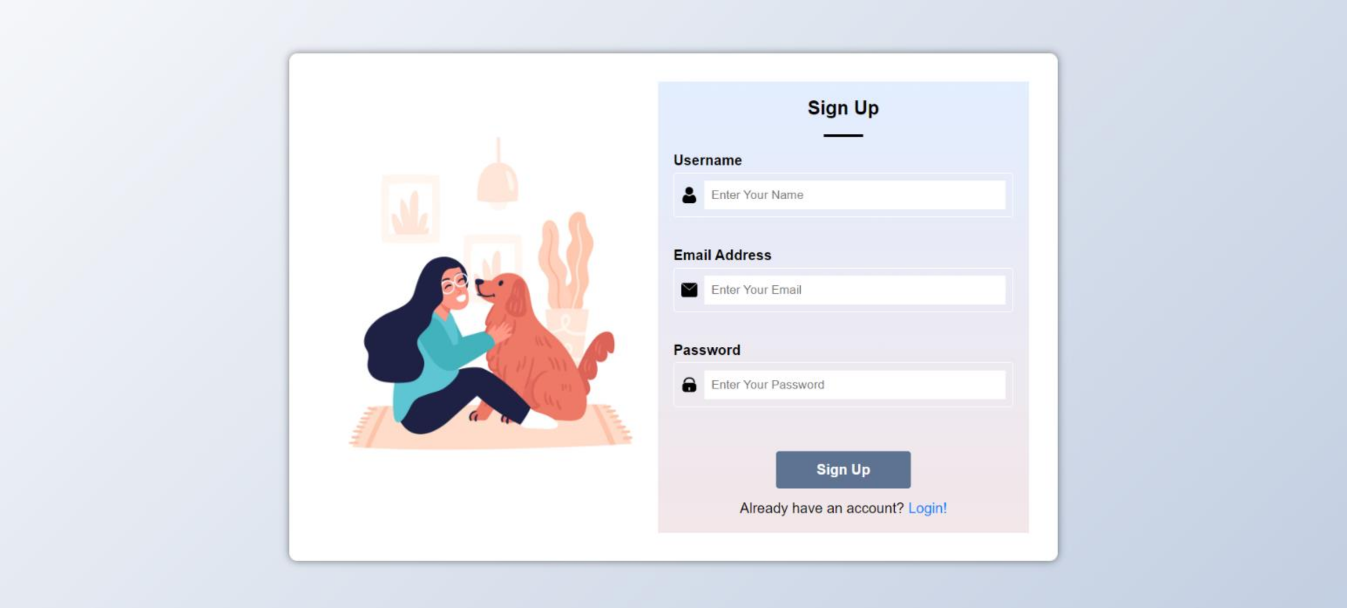


Fig 6.1 LOGIN PAGE

6.2.2 SIGN UP

If a user doesn’t have an account on the website, User can use a component named create new account available in the login page. When the user clicks on that he will be redirected to the signup page. In sign up he should fill up his email id, password and phone number. These inputs will be validated.

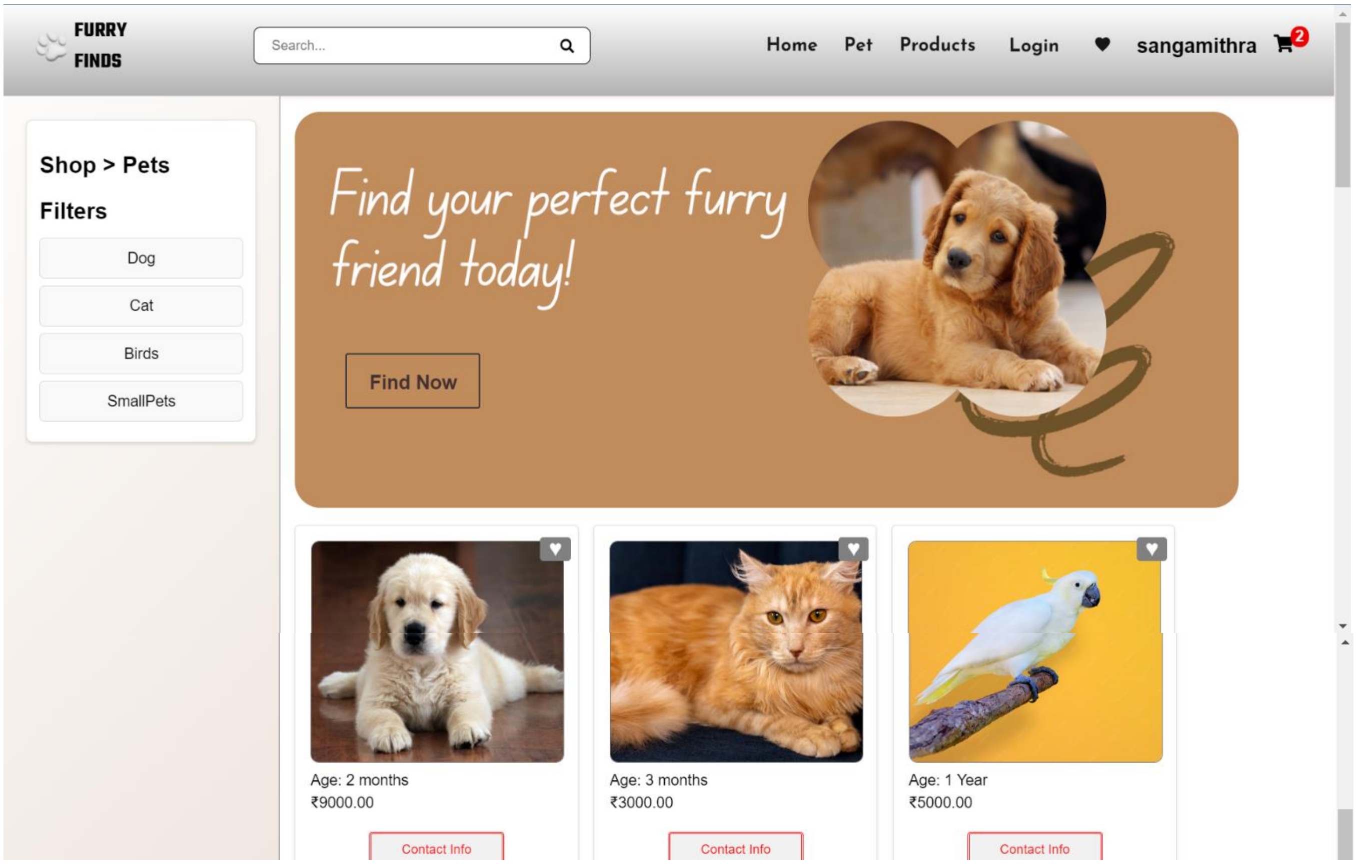


Already have an account? Login!

Fg 6.2 SIGNUP PAGE

6.2.3 HOME

This page features an overview of our offerings, including a curated list of products and pets available for sale. The content is dynamically sourced from the data.js file, allowing for easy updates and additions. Whether the user looking to explore new products or find their next pet, they can effortlessly manage and expand the inventory through this centralized file..



Age: 2 months
₹9000.00

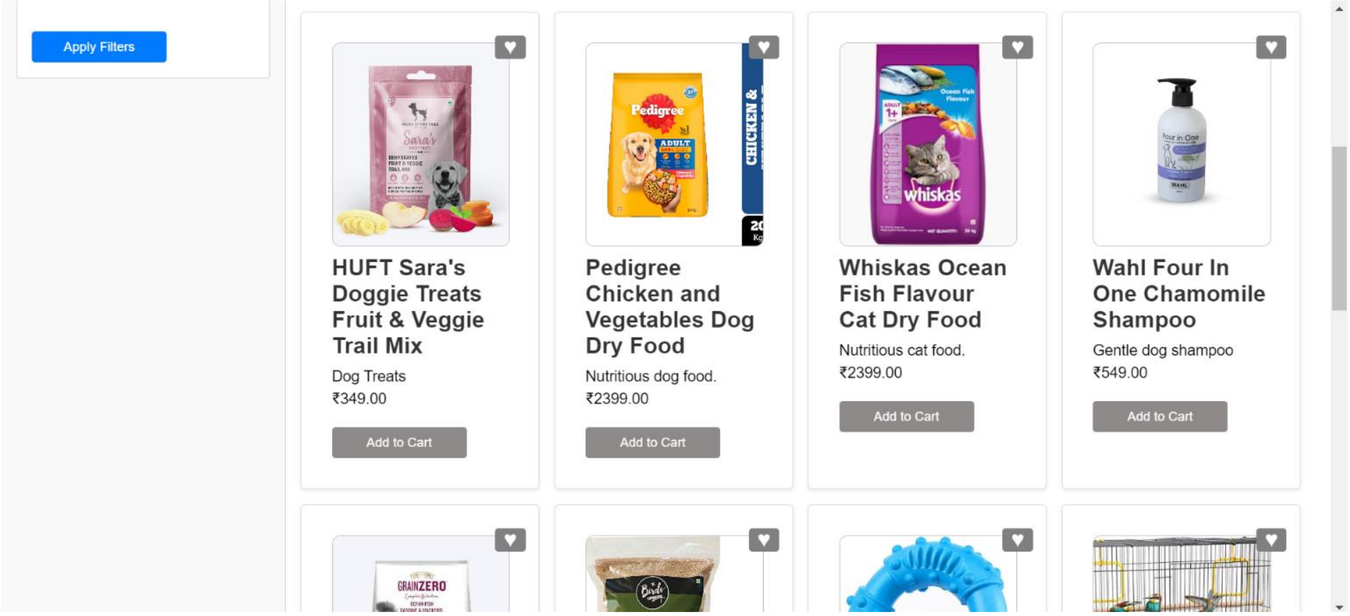
Age: 3 months
₹3000.00

Age: 1 Year
₹5000.00

Fig 6.3 HOME PAGE

6.2.4 LIST OF PRODUCTS

This page displays the list of products available for sale under a range of pet categories . These details have been displayed from the list of products available in the javascript file named data.js. So that if we want to add some products to the particular product we can easily do it by adding the details to the products js file.



HUFT Sara's Doggie Treats Fruit & Veggie Trail Mix

Dog Treats

₹349.00

Pedigree Chicken and Vegetables Dog Dry Food

Nutritious dog food.

₹2399.00

Whiskas Ocean Fish Flavour Cat Dry Food

Nutritious cat food.

₹2399.00

Wahl Four In One Chamomile Shampoo

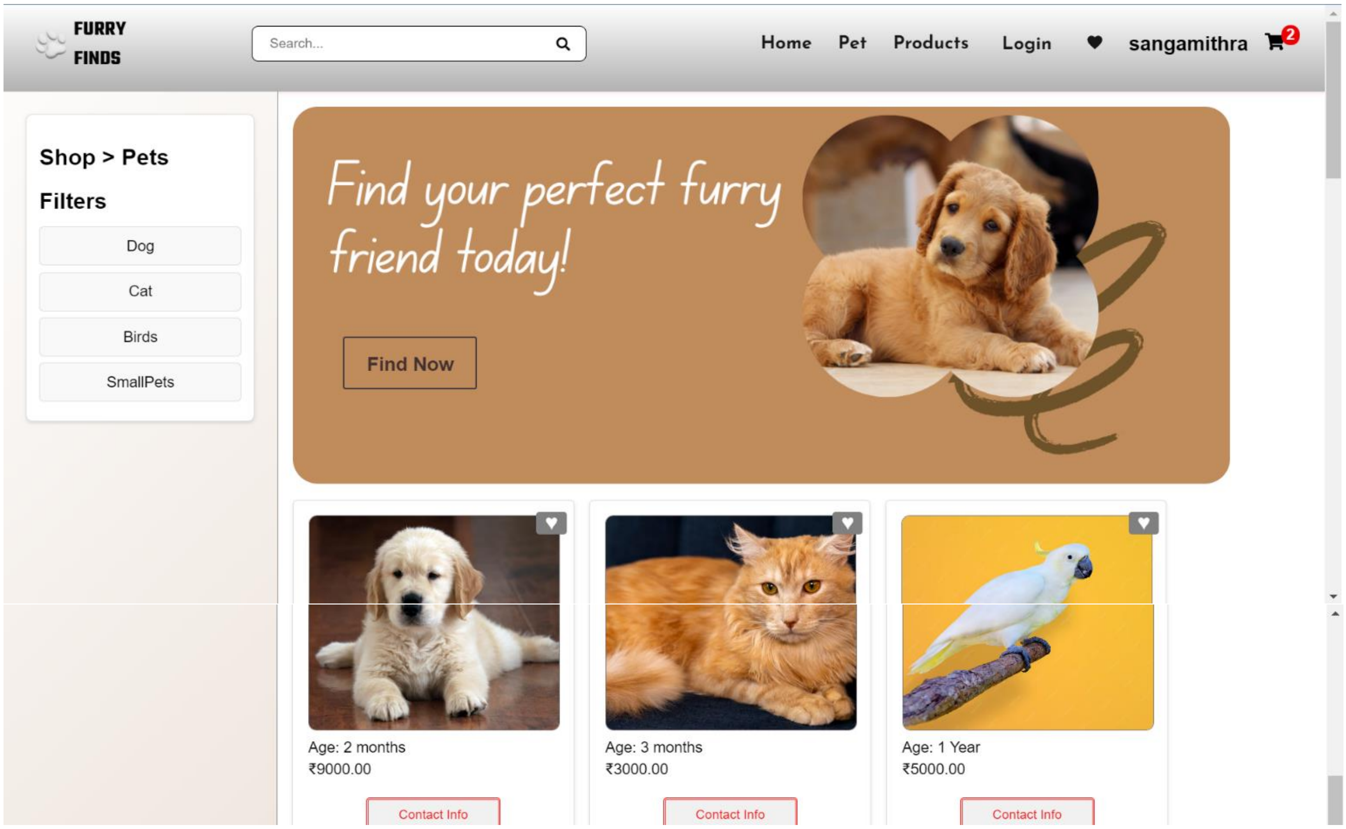
Gentle dog shampoo

₹549.00

Fig 6.4 PRODUCT PAGE

6.2.5 LIST OF PETS

This page displays the list of pets available for sale. The contact info of the seller can also be viewed. These details have been displayed from the list of pets available in the javascript file named data.js. So that if we want to add some details to the particular pet we can easily do it by adding the details to the pets data js file.



Find your perfect furry friend today!

Age: 2 months
₹9000.00

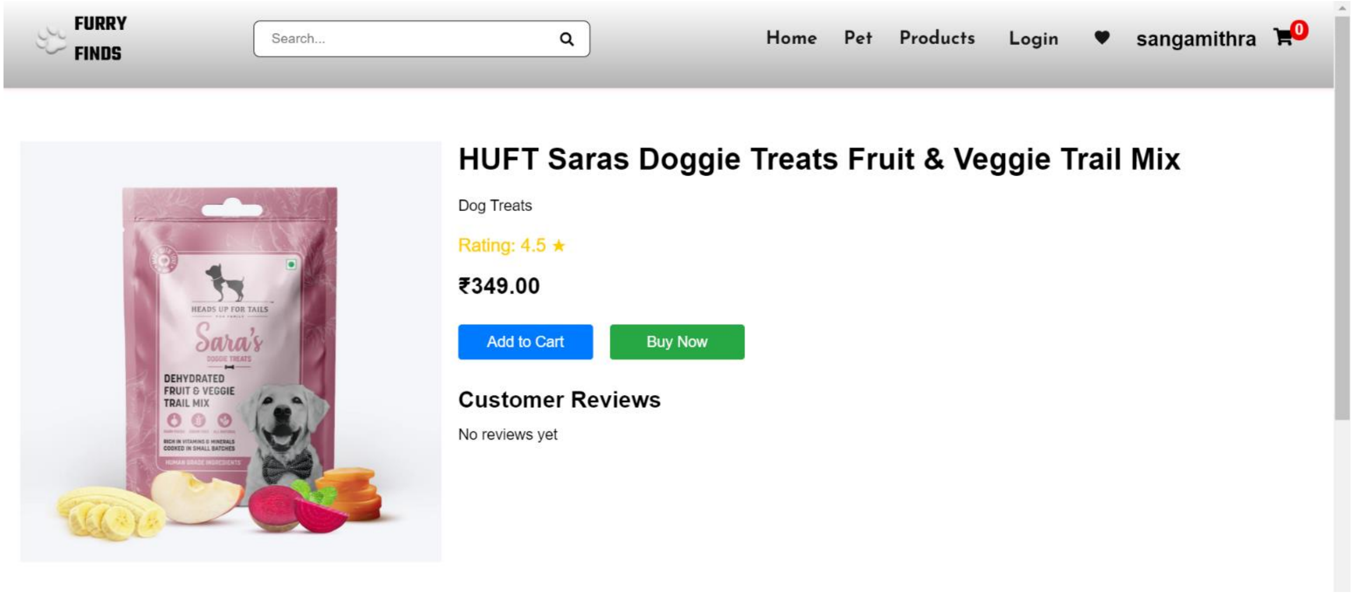
Age: 3 months
₹3000.00

Age: 1 Year
₹5000.00

Fig 6.5 PET PAGE

6.2.6 PRODUCT DISPLAY PAGE

The Product Display page on the "Furry Finds" website showcases a variety of pet products available for sale. This page dynamically displays products by fetching data from the back-end, allowing for a real-time inventory that reflects the latest additions and updates. Each product is presented with essential details such as the name, price, and description, making it easy for users to browse and select items for purchase. The user interface is designed for ease of use, providing sorting and filtering options to help users find specific products quickly. The page also includes options to view more details or add products directly to the cart.



HUFT Saras Doggie Treats Fruit & Veggie Trail Mix

Dog Treats

Rating: 4.5 ★

₹349.00

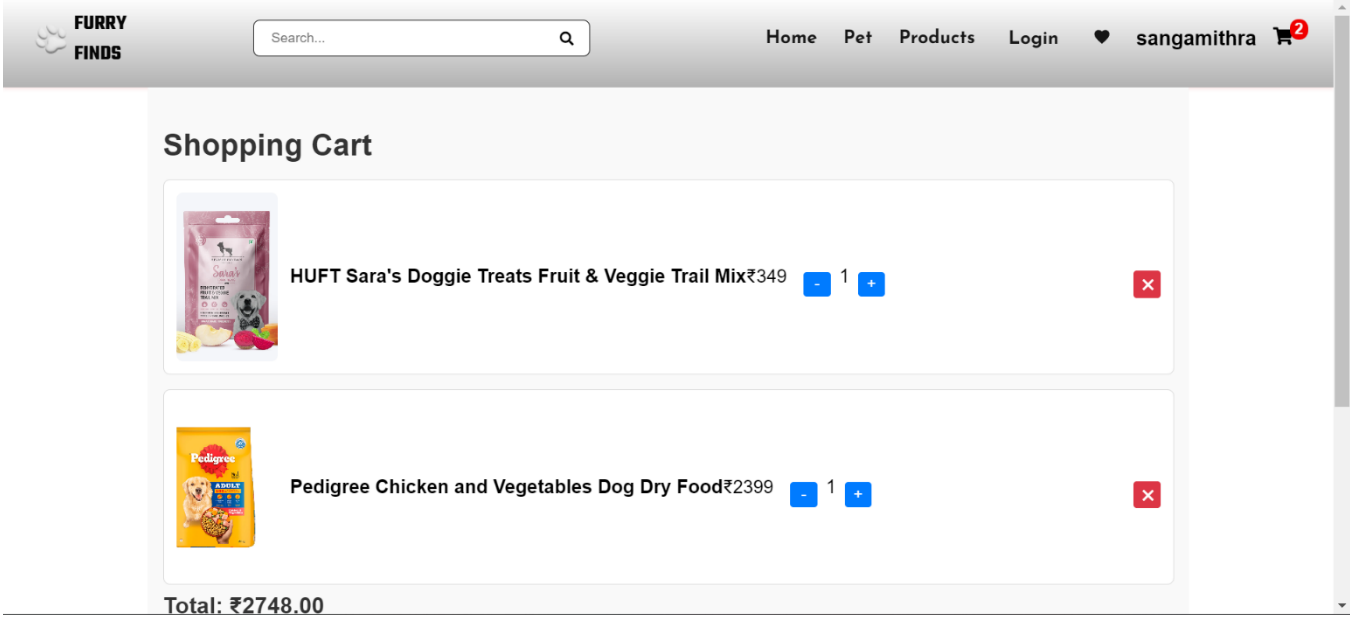
Customer Reviews

No reviews yet

Fig 6.6 PRODUCT DISPLAY PAGE

6.2.7 CART PAGE

This page displays the set of products that the user selected by clicking the button “Add to cart” available on the products page. It also displays the products with their names and its price with quantity. This also helps you to increase the quantity of the products you have added using a plus button which is available on the page after every product's name. You can also remove or reduce the quantity by using the minus symbol button which is also available below every product's name. If the cart doesn’t contain any item it will display that the cart is empty.



Shopping Cart

HUFT Sara's Doggie Treats Fruit & Veggie Trail Mix₹349 - 1 + ✕

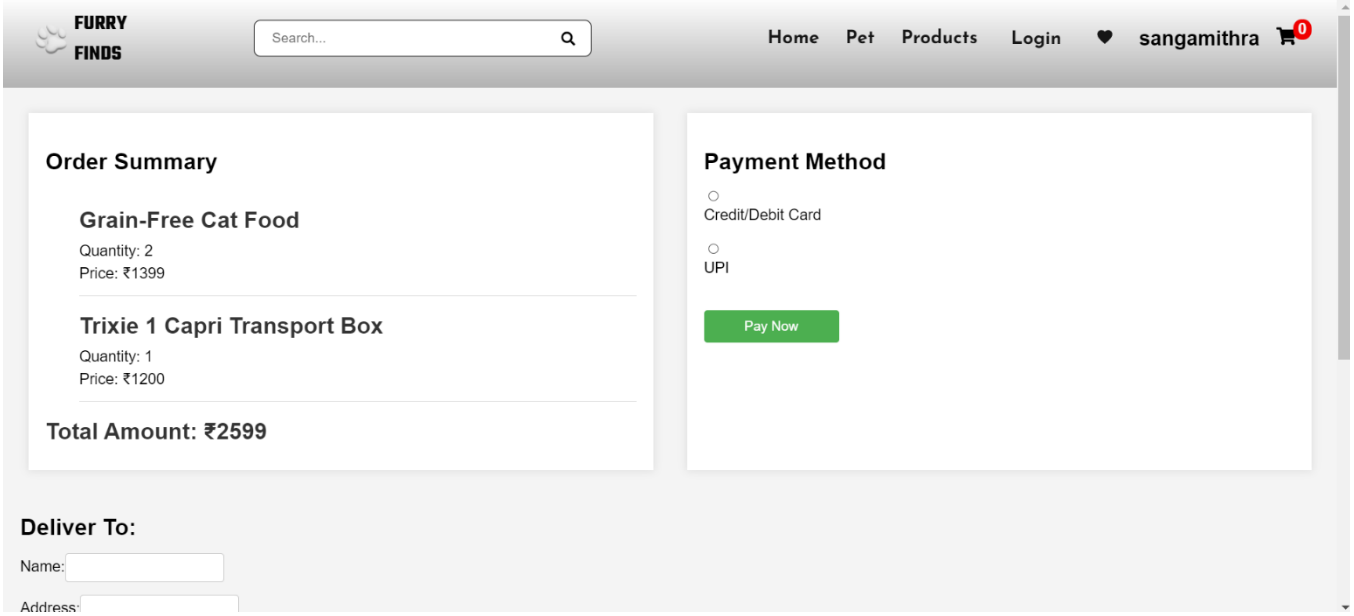
Pedigree Chicken and Vegetables Dog Dry Food₹2399 - 1 + ✕

Total: ₹2748.00

Fig 6.7 CART PAGE

6.2.8 CHECKOUT PAGE

The Checkout page on the "Furry Finds" website provides users with a streamlined and secure process to finalize their purchases. It summarizes the items in the user's cart, displaying product names, quantities, and total price. Users can review their order details, adjust quantities, or remove items directly from this page. Additionally, the checkout page features multiple payment options, allowing users to select their preferred method, whether it be credit card, PayPal, or UPI. To ensure a smooth transaction, users are prompted to input or confirm their shipping address and contact information, which is securely saved for future use. The page is designed with a clear, user-friendly interface that guides users through each step of the checkout process, from reviewing their order to completing the payment. Emphasizing security and ease of use, the checkout page helps ensure that users can complete their purchases with confidence, ultimately leading to a successful transaction and a satisfying shopping experience on Furry Finds.



Order Summary

Grain-Free Cat Food
Quantity: 2
Price: ₹1399

Trixie 1 Capri Transport Box
Quantity: 1
Price: ₹1200

Total Amount: ₹2599

Payment Method

○ Credit/Debit Card

○ UPI

Deliver To:

Name:

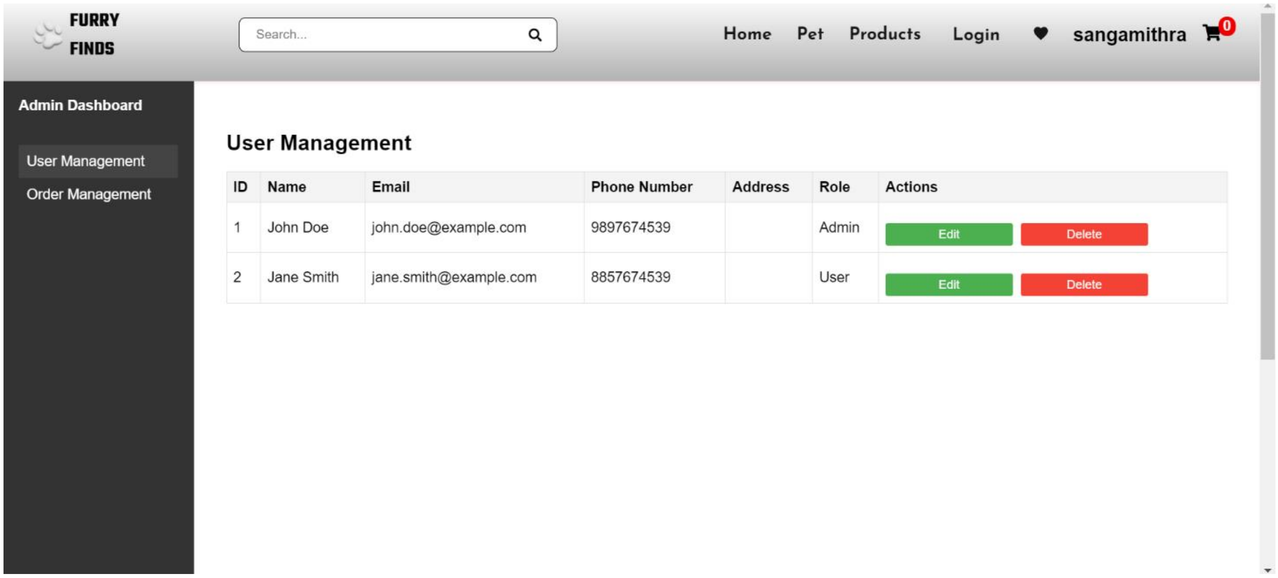
Address:

Fig 6.8 CHECKOUT PAGE

6.2.9 ADMIN PAGE

The Admin Page serves as the central hub for managing the various aspects of the Furry Finds website, providing the administrator with powerful tools to efficiently oversee orders, users, and products. In the Order Management section, the admin can view a detailed list of all orders placed by users, complete with order IDs, user names, product details, and the current status of each order. The admin can easily update the order status to reflect changes such as "Pending," "Shipped," or "Delivered," ensuring smooth processing and tracking of orders.

The User Management section allows the admin to manage user accounts, offering the ability to view user information, update details, and deactivate accounts if necessary. This feature ensures that only valid users can access the platform, maintaining the integrity of the user base.



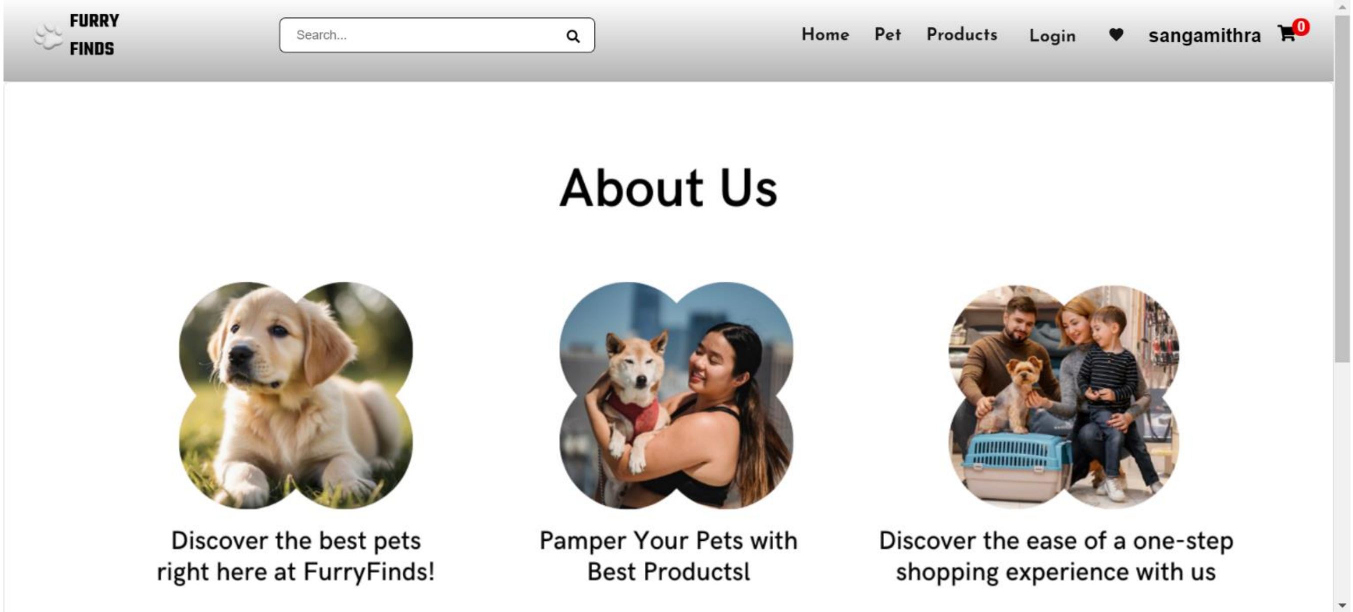
User Management

ID	Name	Email	Phone Number	Address	Role	Actions
1	John Doe	john.doe@example.com	9897674539		Admin	Edit Delete
2	Jane Smith	jane.smith@example.com	8857674539		User	Edit Delete

Fig 6.9 ADMIN PAGE

6.2.10 ABOUTUS PAGE

The About Us page on the "Furry Finds" website provides a comprehensive overview of the platform's mission, values, and the team behind it. This page highlights the website's dedication to connecting pet lovers with their ideal companions, whether through adoption, purchase, or access to pet essentials. It emphasizes the platform's commitment to promoting responsible pet ownership by offering educational resources, expert advice, and support for both new and experienced pet owners. The page also introduces the team of passionate animal lovers and industry professionals who work tirelessly to ensure a secure and user-friendly marketplace. It underscores the platform's partnerships with shelters and rescue organizations, reinforcing its role in advocating for animal welfare. Additionally, the "About Us" page conveys the website's vision of building a trusted community where pet owners, breeders, sellers, and animal enthusiasts can interact, share knowledge, and support one another in creating a safe and caring environment for pets.



About Us

Discover the best pets right here at FurryFinds!

Pamper Your Pets with Best Productsl

Discover the ease of a one-step shopping experience with us

Fig 6.10 ABOUT PAGE

6.7 CODING

Login:

```
import React, { useState, useContext } from 'react';
import { Link, useNavigate } from 'react-router-dom';
import { AuthContext } from '../Context/AuthContext';
import './CSS/LoginSignup.css';

const Login = () => {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const { login } = useContext(AuthContext);
  const navigate = useNavigate();

  const handleLogin = async () => {
    await login(email, password);
    navigate('/');
  };

  return (
    <div className='loginsignup'>
      <div className='loginsignup-container'>
        <h1>Login</h1>
        <div className='loginsignup-fields'>
          <input
            type='email'
            placeholder='Email Address'
            value={email}
            onChange={(e) => setEmail(e.target.value)}
          />
          <input
            type='password'
            placeholder='password'
            value={password}
            onChange={(e) => setPassword(e.target.value)}
          />
        </div>
        <button onClick={handleLogin}>Continue</button>
        <Link style={{ textDecoration: 'none' }} to='/signup'>
                  <p className='loginsignup-login'>Don't have an account? <span>Signup Here</span></p>
        </Link>
      </div>
    </div>
  );
}

export default Login;
```

Sign up:

```
import React, { useState, useContext } from 'react';
import { Link, useNavigate } from 'react-router-dom';
import { AuthContext } from '../Context/AuthContext';
import './CSS/LoginSignup.css';

const Signup = () => {
  const [name, setName] = useState('');
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const { register } = useContext(AuthContext);
  const navigate = useNavigate();

  const handleSignup = async () => {
    await register(name, email, password);
    navigate('/');
  };

  return (
    <div className='loginsignup'>
      <div className='loginsignup-container'>
        <h1>Sign Up</h1>
        <div className='loginsignup-fields'>
          <input
            type='text'
            placeholder='Your Name'
            value={name}
            onChange={(e) => setName(e.target.value)}
          />
          <input
            type='email'
            placeholder='Email Address'
            value={email}
            onChange={(e) => setEmail(e.target.value)}
          />
          <input
            type='password'
            placeholder='Password'
            value={password}
            onChange={(e) => setPassword(e.target.value)}
          />
        </div>
        <button onClick={handleSignup}>Continue</button>
        <Link style={{ textDecoration: 'none' }} to='/login'>
          <p className='loginsignup-login'>Already have an account? <span>Login Here</span></p>
        </Link>
        <div className='loginsignup-agree'>
          <input type='checkbox' name=" id="/>
          <p>By continuing, I agree to the terms of use and privacy policy.</p>
        </div>
      </div>
    </div>
  );
};

export default Signup;
```

Product page:

```
import React, { useContext, useState } from 'react';
import './CSS/ShopCategory.css';
import { ShopContext } from '../Context/ShopContext';
import Item from '../Components/Item/Item';
import Search from '../Components/Search/Search';
import noProductsImg from '../Components/Assets/no_matches.png';
import noMatch from '../Components/Assets/no_matches.png';
import Footer from '../Components/Footer/Footer';

const ShopCategory = (props) => {
  const { all_product } = useContext(ShopContext);
  const [searchVal, setSearchVal] = useState("");
  const [visibleProducts, setVisibleProducts] = useState(8); // Initial number of products to
display

  const handleSearch = () => {
    const filteredProducts = all_product.filter((item) => {
      return (
        item.category === props.category &&
        item.name.toLowerCase().includes(searchVal.toLowerCase())
      );
    });

    if (filteredProducts.length === 0) {
      <div className='img'>
      <img src={noMatch} alt="No matches found" />
    </div>
    }

    return filteredProducts;
  };

  const handleLoadMore = () => {
    setVisibleProducts(prevVisibleProducts => prevVisibleProducts + 10);
  };

  const filteredProducts = handleSearch();

  return (
    <div className='shop-category'>
      <img className='shopcategory-banner' src={props.banner} alt="/>
      <div className='shopcategory-indexSort'>
        <Search searchVal={searchVal} setSearchVal={setSearchVal}
handleSearch={handleSearch} />
      </div>
      <div className='shopcategory-products'>
        {filteredProducts.length > 0 ? (
          filteredProducts.slice(0, visibleProducts).map((item, i) => (
            <Item
              key={i}
              id={item.id}
              name={item.name}
              description={item.description}
              image={item.image}
              new_price={item.new_price}
              old_price={item.old_price}
            />
          ))
        ) : (
          <div className='no-products'>
            <img src={noProductsImg} alt="No products found" />
          </div>
        )}
      </div>

      {filteredProducts.length > visibleProducts && (
        <div className='shopcategory-loadmore'>
          <button className='' onClick={handleLoadMore}>
            Explore More
          </button>
        </div>
      )}
      <Footer />
    </div>
  );
}

export default ShopCategory;
```

Cart page:

```
import React, { useContext } from 'react';
import './CartItems.css';
import { ShopContext } from '../../Context/ShopContext';
import remove_icon from '../Assets/cart_cross_icon.png';
import { Link } from 'react-router-dom';

const CartItems = () => {
    const { getTotalCartAmount, all_product, cartItems, removeFromCart } = useContext(ShopContext);

  return (
    <div className='cartitems'>
      <div className='cartitems-format-main'>
        <p>Products</p>
        <p>Title</p>
        <p>Price</p>
        <p>Quantity</p>
        <p>Total</p>
        <p>Remove</p>
      </div>
      <hr/>

      {all_product.map((e) => {
        if (cartItems[e.id] > 0) {
          return (
            <div key={e.id}>
              <div className='cartitems-format cartitems-format-main'>
                <img src={e.image} alt='' className='carticon-product-icon'/>
                <p>{e.name}</p>
                <p>${e.new_price}</p>
                <button className='cartitems-quantity'>{cartItems[e.id]}</button>
                <p>${e.new_price * cartItems[e.id]}</p>
                  <img className='cartitems-remove-icon' src={remove_icon} onClick={() => removeFromCart(e.id)} alt=''/>
              </div>
              <hr/>
            </div>
          );
        }

        return null;
      })}
```

```
        <div className='cartitems-down'>
          <div className='cartitems-total'>
            <h1>Cart Totals</h1>
            <div>
              <div className='cartitems-total-item'>
                <p>Subtotal</p>
                <p>${getTotalCartAmount()}</p>
              </div>
              <hr/>
              <div className='cartitems-total-item'>
                <p>Delivery Fee</p>
                <p>Free</p>
              </div>
              <hr/>
              <div className='cartitems-total-item'>
                <h3>Total</h3>
                <h3>${getTotalCartAmount()}</h3>
              </div>
            </div>
            <Link to='/payment'>
              <button>PROCEED TO CHECKOUT</button>
            </Link>
          </div>
          <div className='cartitems-promocode'>
            <p>If you have a promo code, enter it here</p>
            <div className='cartitems-promobox'>
              <input type='text' placeholder='Promo Code'/>
              <button>Submit</button>
            </div>
          </div>
        </div>
     );
    }

export default CartItems;
```

CONCLUSION

CHAPTER 7

CONCLUSION

This chapter tells about the conclusion that anyone can drive from the project and the learning we learnt by taking over this project.

7.1 CONCLUSION

The completion of the FurryFinds website project represents a significant achievement for our team. We have dedicated extensive effort to develop a platform that is intuitive, secure, and tailored to meet the needs of pet owners. The website offers a streamlined, reliable, and secure way for customers to purchase pet products and accessories, with a diverse range of items available for selection.

We are proud of the user-friendly design and robust security features implemented on the website, ensuring a safe and seamless shopping experience. By integrating the latest technology and best practices, we have created a platform that not only simplifies the purchasing process but also enhances customer satisfaction. The addition of detailed product information and easy navigation helps users find and select the perfect items for their pets with ease.

Additionally, we have provided valuable resources such as guides and tutorials to assist customers throughout their shopping journey. This support further contributes to an enjoyable and efficient experience.

Overall, the FurryFinds website project has been a success, resulting in a platform that is both secure and user-centric. We anticipate a successful launch and are excited about the potential for continued growth and positive impact in the pet product market.

7.2 FUTURE SCOPE

The FurryFinds project has significant potential for future development and expansion. With advancements in technology and the growing demand for diverse pet products, the platform can evolve to offer even more value to its users. Enhancements could include advanced customization options, allowing users to select and tailor products to better meet their pets' specific needs. The platform can also stay ahead of industry trends by integrating the latest innovations in pet care and technology, such as virtual try-ons for accessories or AI-driven recommendations.

Moreover, FurryFinds can enrich its offerings by providing educational resources, including expert advice, training guides, and health tips, positioning itself as a comprehensive pet care resource. Implementing loyalty programs and personalized discounts can attract and retain customers, while expanding the product range and forming partnerships with new suppliers can cater to a broader audience. Building an online community where pet owners can share experiences and participate in discussions can foster user engagement and create a sense of belonging.

Continuous improvements to the website’s design and functionality will ensure a seamless and enjoyable shopping experience. These enhancements will not only contribute to the platform’s growth but also ensure its relevance and success in the competitive pet products market.

REFERENCES

REFERENCES

1. Spring Boot Documentation
"Spring Boot Reference Documentation,"
https://docs.spring.io/spring-boot/docs/current/reference/html/

2. MySQL Documentation
"MySQL 8.0 Reference Manual,"
https://dev.mysql.com/doc/refman/8.0/en/

3. React.js Documentation
"React – A JavaScript library for building user interfaces,"
https://reactjs.org/docs/getting-started.html

4. Spring Boot & MySQL Integration
Baeldung, "Spring Boot with MySQL: How to do it right,"
https://www.baeldung.com/spring-boot-mysql

5. Building a CRUD Application with React and Spring Boot
Spring.io, "Building a RESTful Web Service with Spring Boot and React,"
https://spring.io/guides/tutorials/react-and-spring-data-rest/

6. Pet Shop Application Example with Spring Boot
GitHub Repository, "Pet Clinic Sample Application using Spring Boot,"
https://github.com/spring-petclinic/spring-petclinic-rest

7. React.js and Redux for E-commerce Applications
FreeCodeCamp, "How to Build an E-commerce Site with React and Redux,"

https://www.freecodecamp.org/news/how-to-build-an-e-commerce-site-with-react-and-redux/

8. Using MySQL with Spring Boot in a Web Application
Java Guides, "Spring Boot, MySQL, JPA, Hibernate Restful CRUD API Tutorial,"
https://www.javaguides.net/2019/05/spring-boot-mysql-jpa-hibernate-restful-crud-api-tutorial.html

9. E-commerce Design with React and Spring Boot
Medium, "Building a Simple E-commerce Application with React, Spring Boot, and MySQL,"
https://medium.com/@daya628/building-a-simple-e-commerce-application-with-react-spring-boot-and-mysql-d71e2cbfd78e

10. Spring Boot & MySQL Example
DZone, "A Simple Web App With Spring Boot, Spring Security, and MySQL,"
https://dzone.com/articles/a-simple-web-app-with-spring-boot-spring-security