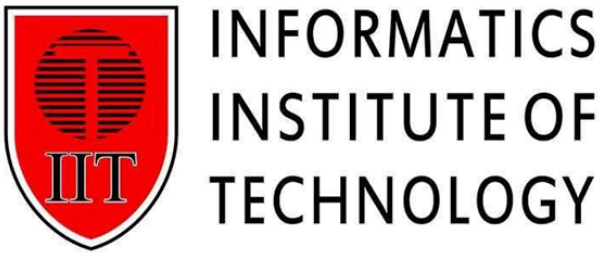


Informatics Institute of Technology

5COSC009C Software Development Group Project

Module Leader : Banu Athuraliya

Mentored by : Pumudu Fernando

Team name : Code Hustle - SE 53

Project name : WooHoo

Team Members

Name	IIT No.	UOW
T. Sujana	20210070	w1866994
S. Danishkar	20210779	w1867668
A.K.B.J. Perera	20210123	w1870576
P.M.C.N. Jayarathna	20210265	w1867053
R.A. Kumarawadu	20210413	w1867139

Declaration

We hereby certify that this project report and all the artifacts associated with it is our own work and it has not been submitted before nor is currently being submitted for any degree programme.

Name	IIT No.	UOW
T. Sujana	20210070	w1866994
S. Danishkar	20210779	w1867668
A.K.B.J. Perera	20210123	w1870576
P.M.C.N. Jayarathna	20210265	w1867053
R.A. Kumarawadu	20210413	w1867139

1. The work contained in the report is original and has been done by us under the mentorship of Mr.Pumudu Fernando.

2. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other university within this country or abroad.

3. We have followed the guidelines provided by the university in writing the report.

4. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and given their details in the references.

Team Code Hustle

IIT – SE - 53

Abstract

Skin diseases in animals can have a significant impact on their welfare and quality of life. Early detection and diagnosis of these conditions is crucial for effective treatment and management. This research aims to develop a skin disease detection system for animals using image processing and machine learning techniques. WooHoo provides a user-friendly interface for users to upload images of their pets' skin diseases and get the detected skin disease of the animal. The report includes four main chapters: Implementation, Testing, Evaluation, and Conclusion.

The application's development process is described in the Implementation part along with the technologies used, design factors, and features applied. The Testing part describes the various kinds of testing carried out on the application, including unit testing, performance testing, usability testing, and compatibility testing. This testing includes both functional and nonfunctional needs. The evaluation procedure, which included both quantitative and qualitative evaluation techniques like user surveys and comments, is presented in the evaluation part. The process of evaluation evaluated user satisfaction, efficiency, usability, compatibility, security, and constant development in addition to the accuracy and reliability of the diagnosis.

The Conclusion chapter summarizes the project's achievements and limitations and analyzes the possible effects of the application on the health and wellbeing of animals. The chapter comes to the conclusion that the application is a useful tool for pet owners and veterinarians in managing animal health, and that it could be made even more useful in the future by adding features like an animal adoption feature or a feature that recommends diet plans.

Overall, the web application for detecting animal skin diseases has the ability to increase the efficiency and precision of diagnosing skin conditions in animals and may have a major effect on the health and wellbeing of animals.

Keywords:

Machine learning, Image processing, Deep learning, Computer vision

Acknowledgement

We would like to express our great appreciation for the continuous support and encouragement to our module leader, **Mr. Banuka Athuraliya**, throughout this project. His insightful observations and feedback contributed greatly to the creation of this project report. We would like to express our great appreciation for the continuous support and encouragement to our mentor, **Mr. Pumudu Fernando**, throughout this project. His insightful observations and feedback contributed greatly to the creation of this project report.

We would also like to thank our SDGP module lecturers, for their valuable time in arranging the lectures needed to gain an overview and insight into the entire module concept. And through online feedback sessions they provided encouragement and gave ideas to try to achieve higher standards in our project Woohoo.

It was very important to solve our challenges. Our gratitude also goes to Dr. T. K. Pillai and the industrial experts, for their insight into various aspects of this domain and inspiring us when the difficulties seemed insurmountable.

All of this would not be possible without our parents. This would have been impossible. It was their strength that kept us going through the hardest times. It would be true to say that they have done only as much as they do. Finally, we are also indeed thankful for all our friends who have helped us constantly whenever we need a helping hand during this project.

Finally, we are also indeed thankful for all our friends who have helped us constantly whenever we need a helping hand during this project.

Table of Content

List of figures

List of tables

Abbreviations Table

HTML	Hypertext Markup Language
GUI	Graphical User Interface
CI	Continuous Integration
CD	Continuous Deployment
API	Application Programming Interface
JSON	JavaScript Object Notation
CNN	Convolutional Neural Network
ANN	Artificial Neural Network
GPS	Global Positioning System
UI	User Interface
UX	User Experience

Chapter 1: Implementation

1.1 Chapter Overview

This chapter will discuss about the overview of the prototype of the woohoo application giving a detailed discussion about the technology selected to successfully implement Woohoo application and also shows how the backend and frontend part of the application is coded with screenshots and justification of the git repository and finally it also details about the deployment pipeline.

1.2 Overview of the prototype

Hardware and software elements would presumably be used in this prototype of dog and cat skin disease detection application WooHoo. The hardware element could be a camera that can take clear pictures of the animal's skin.

The software component would develop algorithms to analyze the pictures and look for signs of skin disease. These algorithms are using machine learning techniques to learn from a large dataset of images of diseased skin of dogs and cats and identify patterns that are indicative of specific skin conditions.

The prototype includes a user UI that enables interaction with the system and displays the results of the analysis. This web-based interface enables users to upload animal skin images and get the detected skin disease of that animal. This prototype also contains other features like letting users find accommodations for stray dogs and cats.

Overall, this animal skin disease detection application WooHoo is accurate, reliable, and easy to use, when providing outputs to users and veterinarians for identifying and managing skin diseases in dogs and cats.

1.3 Technology selections

1.3.1 React

ReactJS is a JavaScript library that is used to create modular user interface elements. React is used to create modular user experiences. React can power native applications using React Native and can display on the server using Node.

ReactJS is used to build the interactive user interface of this WooHoo application. ReactJS helped to improve the development process of this skin disease detection web application by providing a modular, efficient, and reusable approach to building the UI.

1.3.2 Nodejs

JavaScript code can be executed outside of an online browser using Nodejs, an open-source, cross-platform runtime environment. Building server-side web apps, command-line tools, and real-time programs like chat programs and online sports typically use Nodejs.

Nodejs is used in this WooHoo application to handle the server-side logic of the animal skin disease detection web application, such as processing image uploads, running machine learning algorithms for skin disease detection, and managing user accounts and authentication, and accommodation functionality.

1.3.3 Tailwind CSS

A collection of pre-made, ready-to-use CSS classes are offered by the utility-first Tailwind CSS framework, which can be rapidly applied to HTML components to create unique user interfaces. It provides an extensive collection of low-level tool classes for designing components like color, font, spacing, and layout.

or this application's user interface, Tailwind CSS is used to create a simple and clear design. So using tailwind CSS, attractive elements like buttons and input fields are created. Additionally, Tailwind CSS is used to make specific color schemes that complement the application's identity and improve the user experience and makes it responsive so that it can be used across different platforms.

1.3.4 Python

TensorFlow

Building and developing machine learning models use the open-source library TensorFlow, which supports dataflow and differentiable computing. The most widely used computer language for machine learning is Python, and TensorFlow offers a high-level API that enables programmers to create and train sophisticated machine learning models using only a few lines of Python code. Numerous machine learning algorithms and methods, including deep learning, reinforcement learning, and neural networks, are supported by TensorFlow.

As TensorFlow is a powerful open-source software library for building and training machine learning models, it is used for the image recognition and object detection in this application. A model would need to be trained using a collection of images of diseased animal skin in order to be used in an online tool for TensorFlow-based disease identification in animals. So the pictures and the model are preprocessed and trained using TensorFlow's APIs.

Keras

Python-based open-source Keras is a neural network framework. It is a well-liked option for creating deep learning models because it is user-friendly, extensible, and flexible. Developers can rapidly create and test out various architectures and hyperparameters by Keras high-level API for creating and training neural networks. Convolutional neural networks (CNNs), recurrent neural networks (RNNs), and even bespoke models are supported, among many other kinds of neural networks. After preparing a dataset of

animal diseased skin images, Keras is used to define and train a deep learning model that can identify the skin disease of that animal.

1.3.5 MongoDB

MongoDB is a popular NoSQL database that can be used to store and manage data. MongoDB provides a flexible data model that can handle both structured and unstructured data. MongoDB is used to keep data of the user accounts like the user name, email and hashed password and also it is used to store the organization details that are displayed in the accommodation page. MongoDB is used to keep data of the user accounts like the username, email and hashed password and also it is used to store the organization details that are displayed in the accommodation page.

1.4 Implementation of the data science component

The data science component of the project is done using two different models for cats and dogs. The cat model can detect diseases like Ringworm and Earmites whereas the dog model can detect diseases like Mange and Ringworm.

This animal skin disease detection web application's core data science component usually implements machine learning algorithms and data processing techniques to recognize and categorize animal skin diseases based on input pictures.

Data collection and preprocessing linked to animal skin diseases is the first stage in putting the core data science component into practice. Getting a dataset of pictures of diseased skin of the dog or cat, cleaning the data, and getting it ready for use in machine learning methods are all necessary steps in this process and some of the dataset we got were already preprocessed.

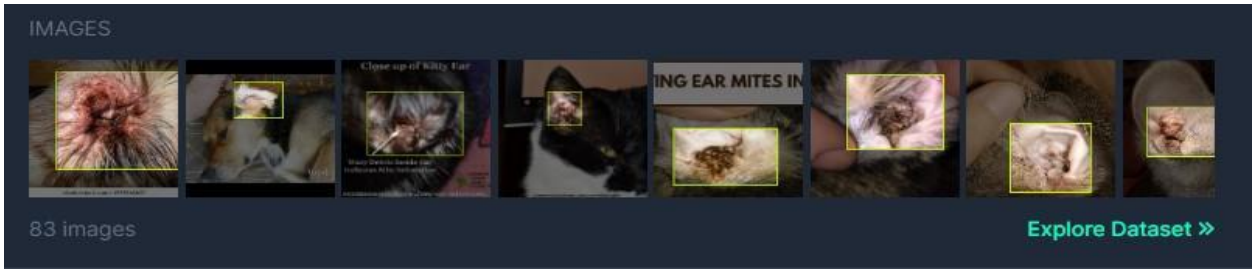


Figure 1: Dataset collection

The data set is then labeled according to the particular disease and their corresponding name.

```
DIRECTORY = r'D:\Desktop\Cat_model\dataset\train'
CATEGORIES = ["Earmites","Ringworm"]

IMG_SIZE = 130

data = []

for category in CATEGORIES:
    folder = os.path.join(DIRECTORY, category)
    label = CATEGORIES.index(category)
    for img in os.listdir(folder):
        img_path = os.path.join(folder, img)
        img_arr = cv2.imread(img_path)
        img_arr = cv2.resize(img_arr, (IMG_SIZE, IMG_SIZE))
        data.append([img_arr, label])
random.shuffle(data)


X=[]
y=[]

for features, labels in data:
    X.append(features)
    y.append(labels)
```

Figure 2: Cat model labeling dataset

```
DIRECTORY = r'D:\Downloads\Dog_model\Dog_model\dataset\train'
CATEGORIES = ["Mange","Ringworm"]

IMG_SIZE = 130

data = []

for category in CATEGORIES:
    folder = os.path.join(DIRECTORY, category)
    label = CATEGORIES.index(category)
    for img in os.listdir(folder):
        img_path = os.path.join(folder, img)
        img_arr = cv2.imread(img_path)
        img_arr = cv2.resize(img_arr, (IMG_SIZE, IMG_SIZE))
        data.append([img_arr, label])
random.shuffle(data)


X=[]
y=[]

for features, labels in data:
    X.append(features)
    y.append(labels)
```

Figure 3: Dog model labeling dataset

Training machine learning models that can precisely recognize and categorize various kinds of animal skin diseases is the next stage after preprocessing the data. Convolutional neural networks (CNNs) is used as a machine learning method for classifying images. The preprocessed dataset is used to train these models, and as part of the training process, the model parameters are usually optimized to reduce inaccuracy and increase precision.

```
X = np.array(X)
y = np.array(y)

# y = to_categorical(y)

model = Sequential()

# Add a convolutional layer with 32 filters, a 3x3 kernel, and ReLU activation
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(130, 130, 3)))

# Add a max pooling layer with a 2x2 pool size
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(256, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

# Add a flatten layer to convert the 2D feature maps to a 1D feature vector
model.add(Flatten())

model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))

# Add a dense layer with 128 neurons and ReLU activation
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))

# Add an output layer with 2 neurons and a softmax activation
model.add(Dense(2, activation='softmax'))

# Compile the model with binary cross-entropy loss and the Adam optimizer
opt = keras.optimizers.RMSprop(learning_rate=0.0001)

model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy'])
```

Figure 4: Cat model CNN code

```
X = np.array(X)
y = np.array(y)

# y = to_categorical(y)

model = Sequential()

# Add a convolutional layer with 32 filters, a 3x3 kernel, and ReLU activation
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(130, 130, 3)))

# Add a max pooling layer with a 2x2 pool size
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(256, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

# Add a flatten layer to convert the 2D feature maps to a 1D feature vector
model.add(Flatten())

model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))

# Add a dense layer with 128 neurons and ReLU activation
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))

# Add an output layer with 2 neurons and a softmax activation
model.add(Dense(2, activation='softmax'))

# Define the optimizer
opt = keras.optimizers.Adam(learning_rate=0.0001)

# Compile the model with binary cross-entropy loss and the Adam optimizer
model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy'])
```

Figure 5: Dog model CNN code

TensorFlow is used as the framework to combine machine learning models with the web application after they have been trained. This web application uses a skin picture of an animal to recognize and categorize the skin disease using a trained machine learning model. The web interface can then show the model's result, which will show the detected skin disease of the animal.

```python
from keras.models import load_model
import cv2
import numpy as np

# Load the saved model
model = load_model('cat.model(newOptimiser2)')

# Load the image you want to classify
img = cv2.imread(r'dataset\test\Ringworm\Ringworm-cat-Handsome-5680LF_1_jpg.rf.8f5acea6a261fc1720135f31c6f960cc.jpg')
img = cv2.resize(img, (130, 130))
img = np.reshape(img, [1, 130, 130, 3])

prediction = model.predict(img)

print(prediction)
if prediction[0][0] > prediction[0][1]:
    print("Earmites")
else:
    print("Ringworm")
```

```
[[1.5601117e-05 9.9998438e-01]]
Ringworm
```

Figure 6: Prediction from the cat model

```python
from keras.models import load_model
import cv2
import numpy as np

# Load the saved model
model = load_model('dog.model(newOptimiser2)')

# Load the image you want to classify
img = cv2.imread(r'D:\Downloads\Dog_model\Dog_model\dataset\train\Mange\IMG-20230214-WA0048.jpg')
img = cv2.resize(img, (130, 130))
img = np.reshape(img, [1, 130, 130, 3])

prediction = model.predict(img)

print(prediction)
if prediction[0][0] > prediction[0][1]:
    print("Manges")
else:
    print("Ringworm")
```

```
[[9.9999964e-01 3.9298968e-07]]
Manges
```

Figure 7: Prediction for the dog model

So, the core data science component of this animal skin disease detection application involves machine learning algorithms and data processing methods to recognize and categorize animal skin diseases based on input pictures. Veterinarians and other animal health experts are able to swiftly and reliably spot possible disease outbreaks and take appropriate action by

integrating this component with the web application and utilizing data visualization and analysis methods.

1.5 Implementation of the backend component

The backend of the web application requires a server to handle HTTP requests and responses. The server is implemented using Node.js and Python. The backend file structure is provided below.

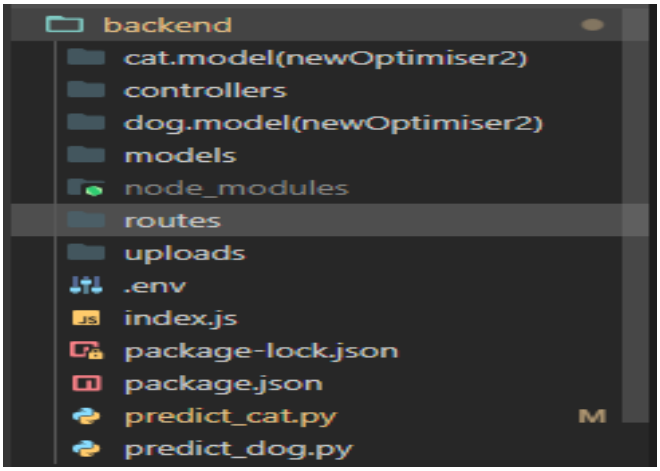


Figure 8: Backend file structure

1.5.1 The main API call routes

The below code is the main API routes dedicated for the users, upload images and the organization details,

```
// middlewares
app.use(cors())
app.use(express.json())

app.use((req,res,next)=>{
    console.log(req.path, req.method);
    next()
})
// routes
app.use("/api/user",userRoutes)
app.use("/api", uploadRoutes)
app.use("/api", organizationRoutes)
```

Figure 9: API routes

1.5.2 Disease prediction

The code below gets the image from the frontend and using the user selected animal type it decides what model to run. By using the package called child process we were able to run the python prediction file in Node.js and get the prediction result and send it as the response to the frontend.

```
const fs = require("fs");
const { spawn } = require('child_process');
const imageModel = require("../models/imageModel");


const uploadImageNPredict = async (req, res) =>{
    let output = '';
    let animalType = req.body.animalType
    console.log(animalType);
    const saveImage =  imageModel({
        name: req.body.name,
        img: {
            data: fs.readFileSync("uploads/" + req.file.filename),
            contentType: "image/png",
        },
    });
    saveImage
        .save()
        .then((response) => {
            console.log("image is saved");
            // res.send('image is saved')
        })
        .catch((err) => {
            console.log(err, "error has occur");
        });
    let pyScript = ""
    if(animalType === 'cat') {
        pyScript = spawn('python', ['predict_cat.py', './uploads/' + req.file.filename]);
        console.log("ran the cat model");
    }else{
        pyScript = spawn('python', ['predict_dog.py', './uploads/' + req.file.filename]);
        console.log("ran the dog model");
    }

    pyScript.stdout.on('data', (data) => {
        console.log(`stdout: ${data}`);
        output = data.toString();
    });
    pyScript.on('close', (code) => {
        console.log(`child process exited with code ${code}`);
        console.log(output)
        res.send(output)
    });
    console.log("output"+output)
}

module.exports = {uploadImageNPredict}
```

Figure 10: Disease prediction code

1.5.3 Accommodation page routes

Routes for functionality like retrieving organization, adding new organization, updating organization, and also deleting organizations are provided below.

```
const retreiveOrganization = async(req,res)=>{
    try{
        const organization = await Organization.find();
        res.status(200).json(organization);
    }catch(err){
        res.status(400).json({message:err.message});
        console.log(err);
    }
}

const addOrganization = async(req,res)=>{
    const {organizationName,noOfSlotsDogs,noOfSlotsCats,userEmail} = req.body;
    try{
        const organization = await Organization.create({organizationName, noOfSlotsDogs, noOfSlotsCats, userEmail})
        res.status(200).json(organization);
    }catch(err){
        res.status(400).json({message:err.message});
        console.log(err);
    }
}

const updateOrganization = async(req,res)=>{
    const { organizationName, noOfSlotsDogs, noOfSlotsCats, userEmail } = req.body;
    const id = req.params.id; // assuming the ID is passed as a parameter in the URL

    try {
    const organization = await Organization.findByIdAndUpdate(id, {
        organizationName,
        noOfSlotsDogs,
        noOfSlotsCats,
        userEmail
    }, { new: true });

    res.status(200).json(organization);
    } catch(err) {
    res.status(400).json({ message: err.message });
    console.log(err);
    }
}
```

Figure 11: Accommodation page code

```
const deleteOrganization = async(req,res)=>{
    const id = req.params.id;
    try{
        const organization = await Organization.findByIdAndDelete(id);
        res.status(200).json(organization);
    }catch(err){
        res.status(400).json({message:err.message});
        console.log(err);
    }
}

module.exports = {retreiveOrganization, addOrganization, updateOrganization, editOrganisation, deleteOrganization}
```

Figure 12: Accommodation page code

1.6 Implementation of the front-end component

Users engage with the user interface on the frontend to access the features of the application. It includes a navigation menu, buttons and other elements that allow users to perform actions such as uploading images or viewing results. The user interface is implemented using Tailwind CSS along with React frontend frameworks.

Users need to upload images of the affected areas to diagnose skin diseases in animals. Input forms are used to collect this information from the user. These input forms are created using React and styled with Tailwind CSS.

At last, the application provides the results of the detected skin disease to the user. This result can take the form of a diagnosis. This feedback mechanism is implemented using React, Tailwind CSS along with server-side programming language Python to handle the backend logic and API integration.

And also, there is an Accommodation Page where organizations can create accounts and help users to find shelters for the stray animals by booking an organization if there is any available slot and also the organization can update their available slots whenever they login to the web page and also delete their entire organization if needed.

Home Page

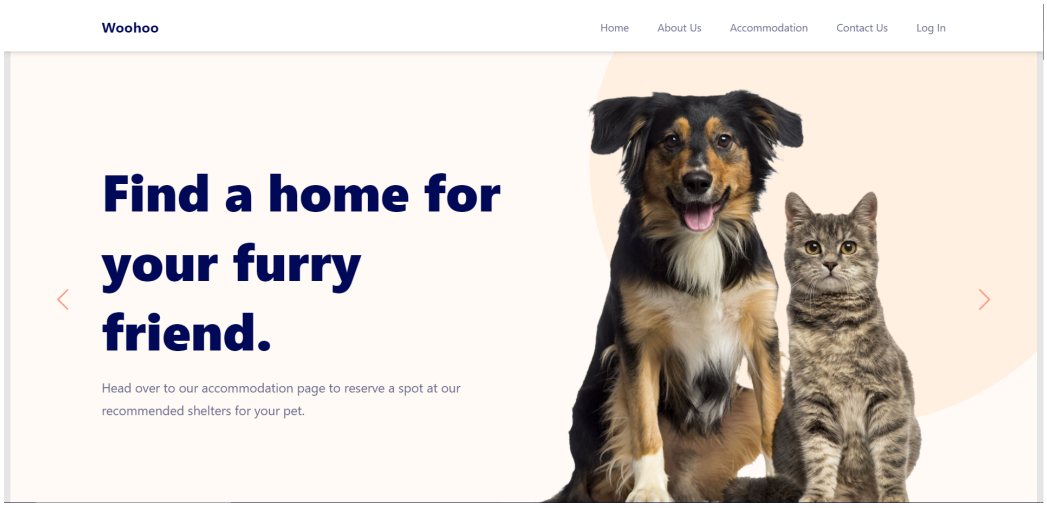


Figure 13: Home Page

Signup Page

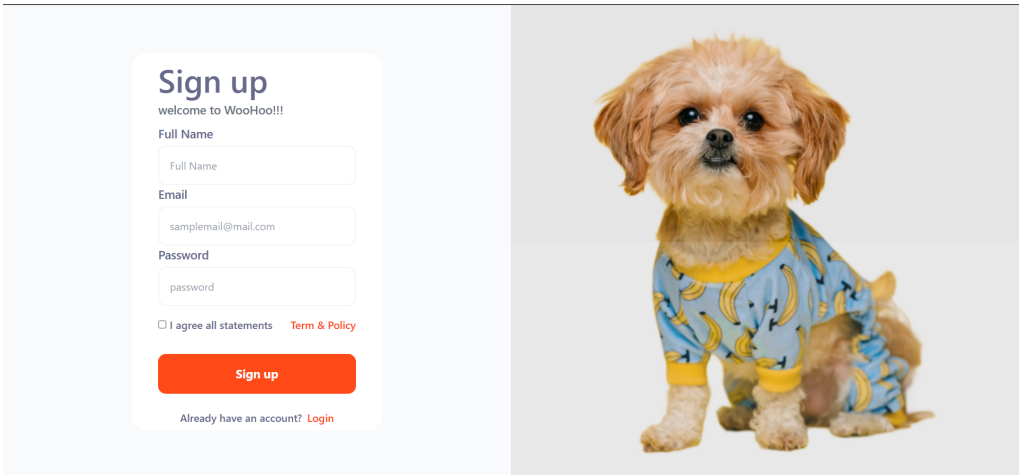


Figure 14: Signup Page

Login Page

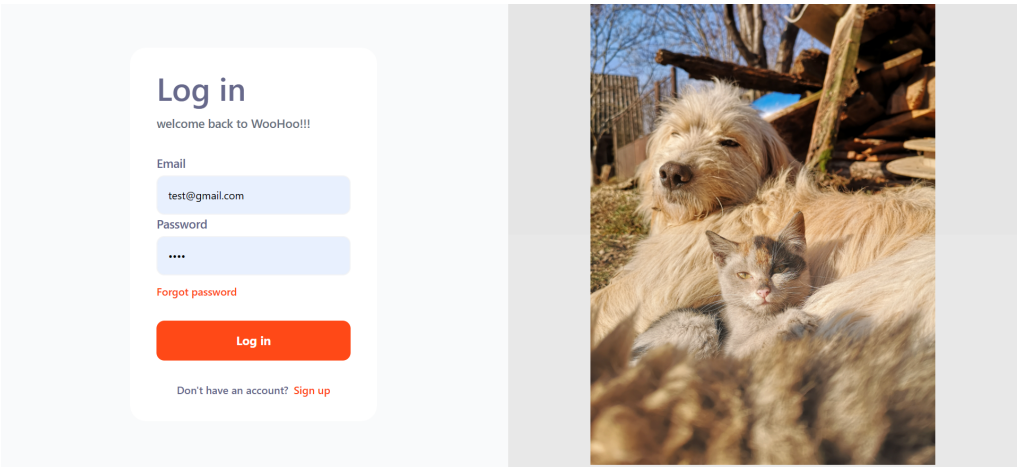


Figure 15: Login Page

Services Page

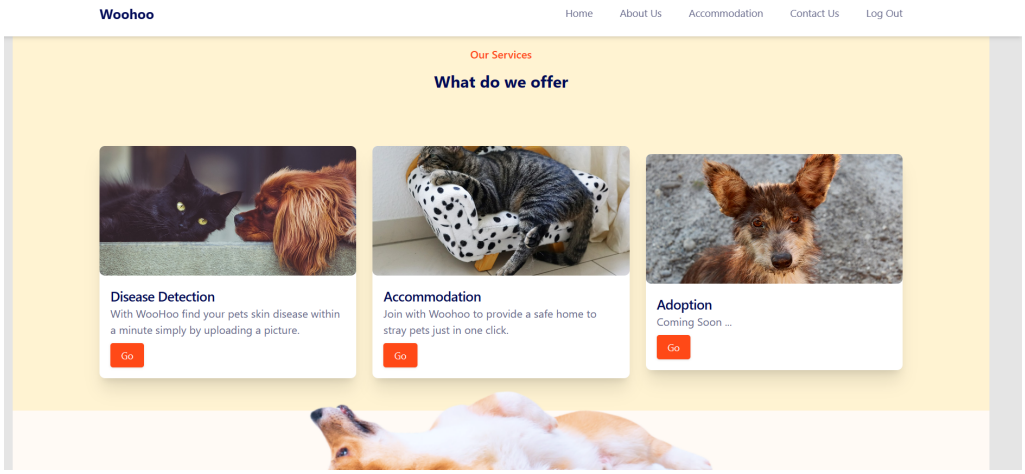


Figure 16: Service Page

Image Upload Page

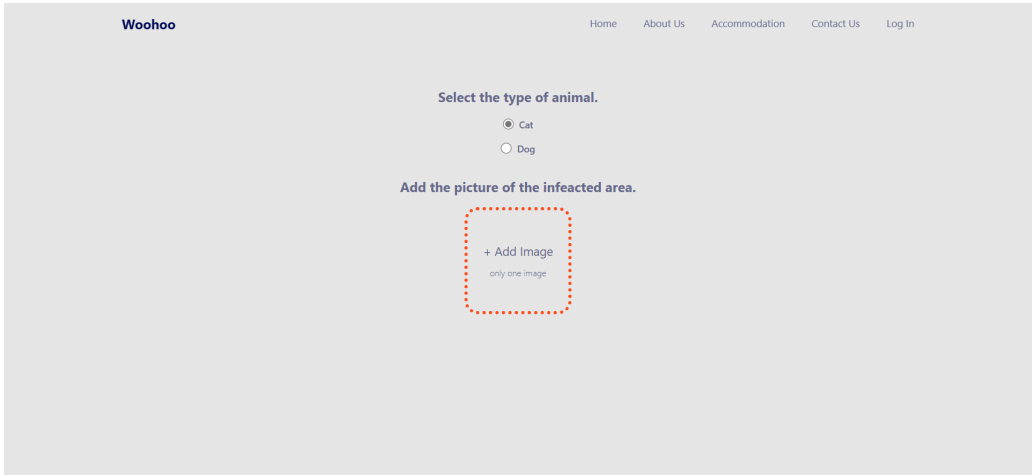


Figure 17: Image Upload Page

Detected Skin Disease Result Page

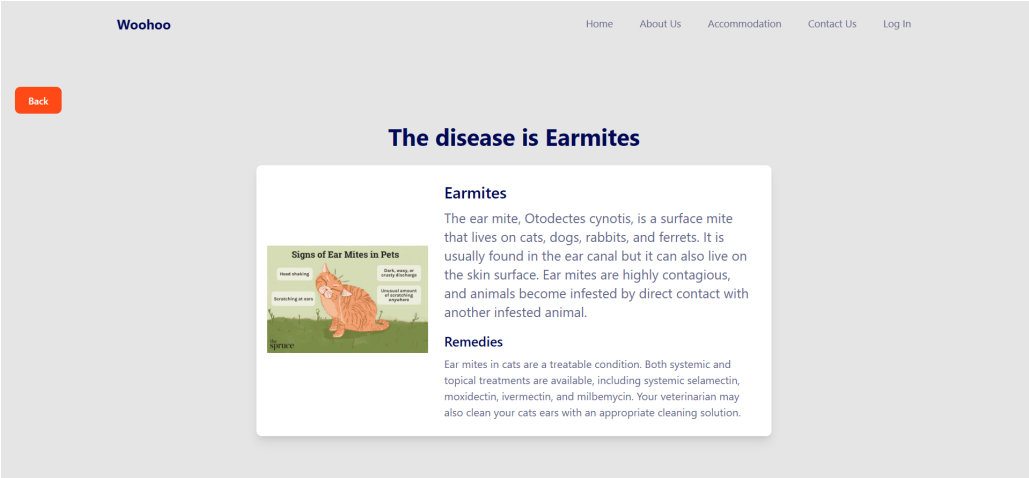
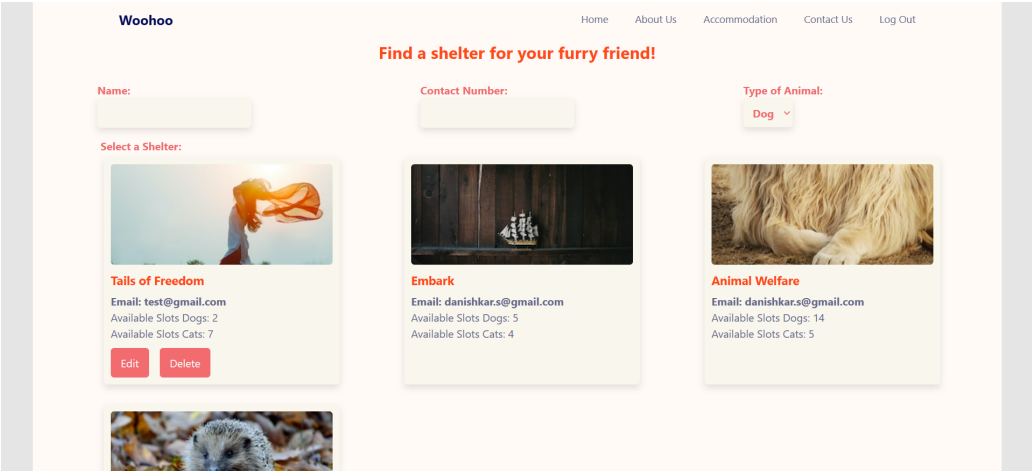


Figure 18: Detected Skin Disease Result Page

Accommodation Page



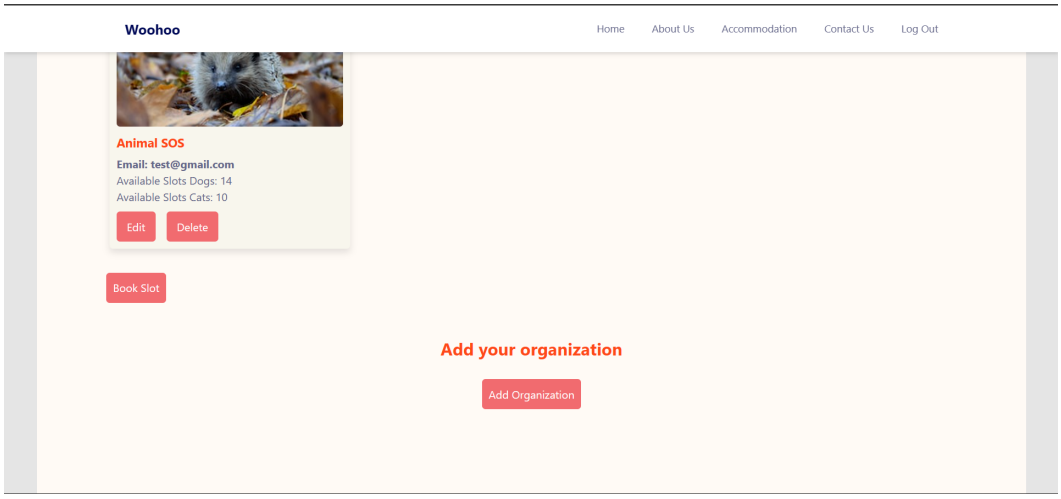


Figure 19: Accommodation Page

About Us Page

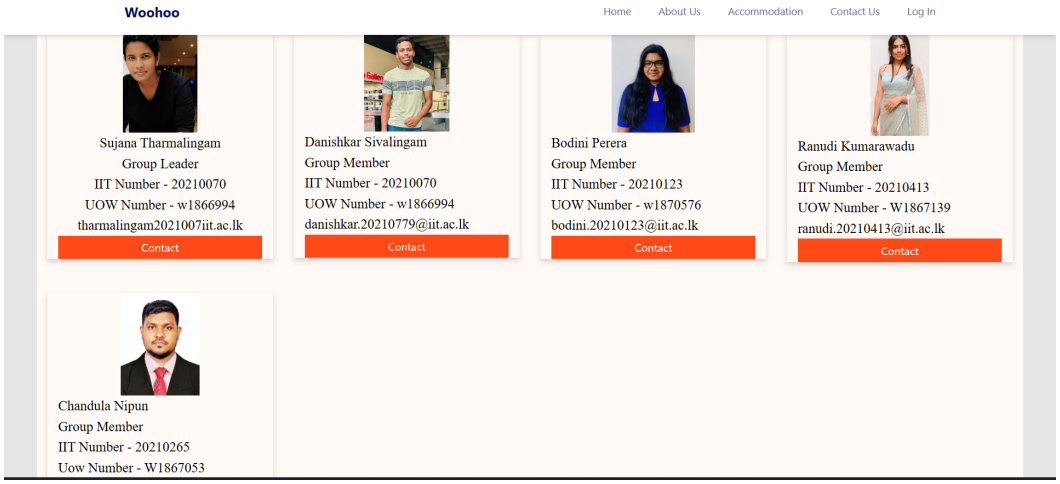


Figure 20: About Us Page

Contact Us

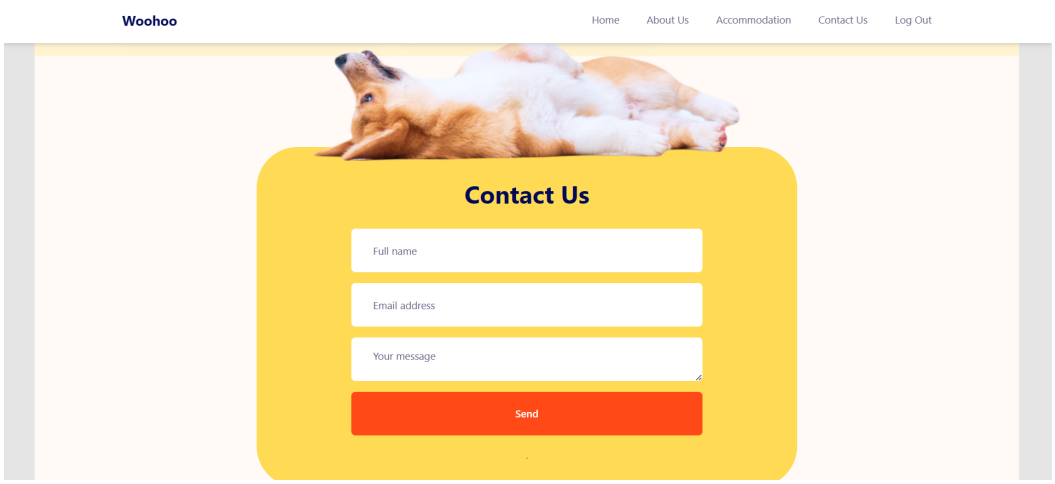


Figure 21: Contact Us Page

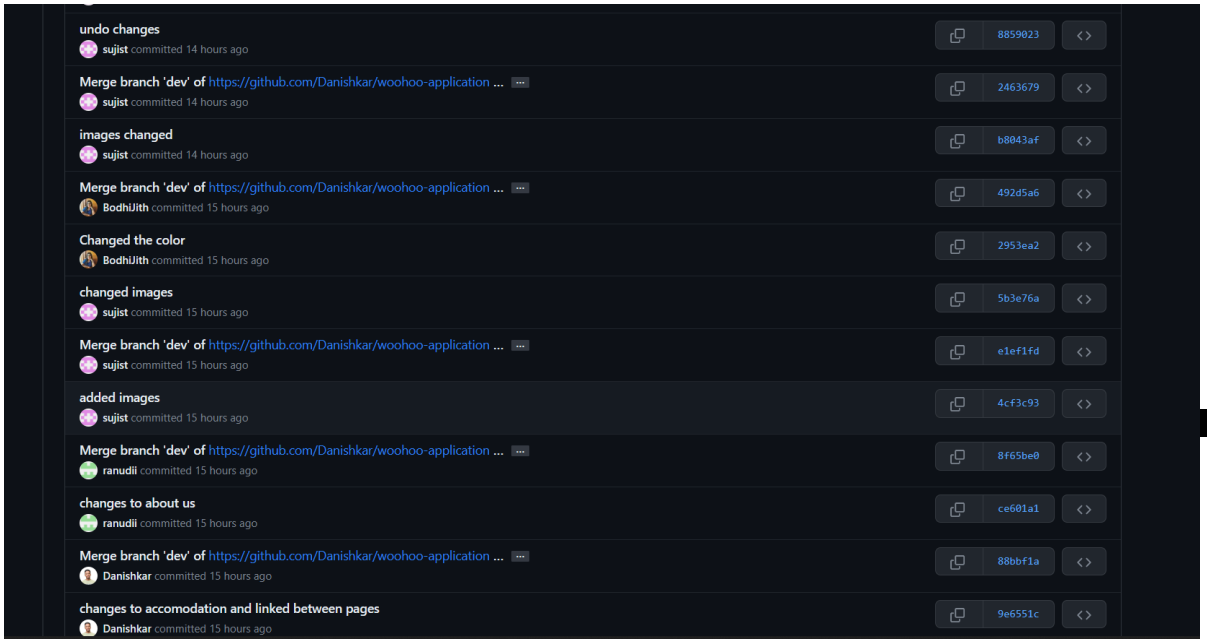
1.7 GIT Repository

Creating the GIT repository is done using a GitHub. After the repository is created, it is cloned to the personal computers using the Git command line or a Git GUI client.

After the repository is cloned, files are added to the project and the other changes are made to existing files. Then they are committed to the repository using the Git command line. A commit is a record of the code modifications performed at that particular point in time.

It was easy for team members to clone the repository, make changes, and then upload their changes back to the repository because Git allows developers to communicate with other team

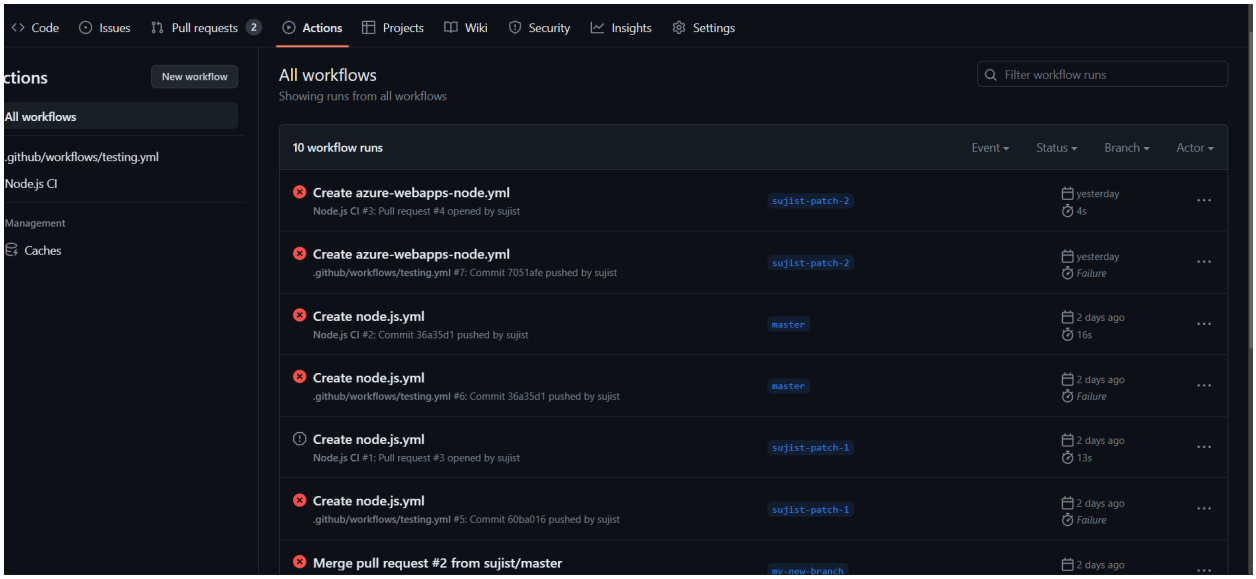
members by sharing their code through the Git hosting service. Git history provides the ability

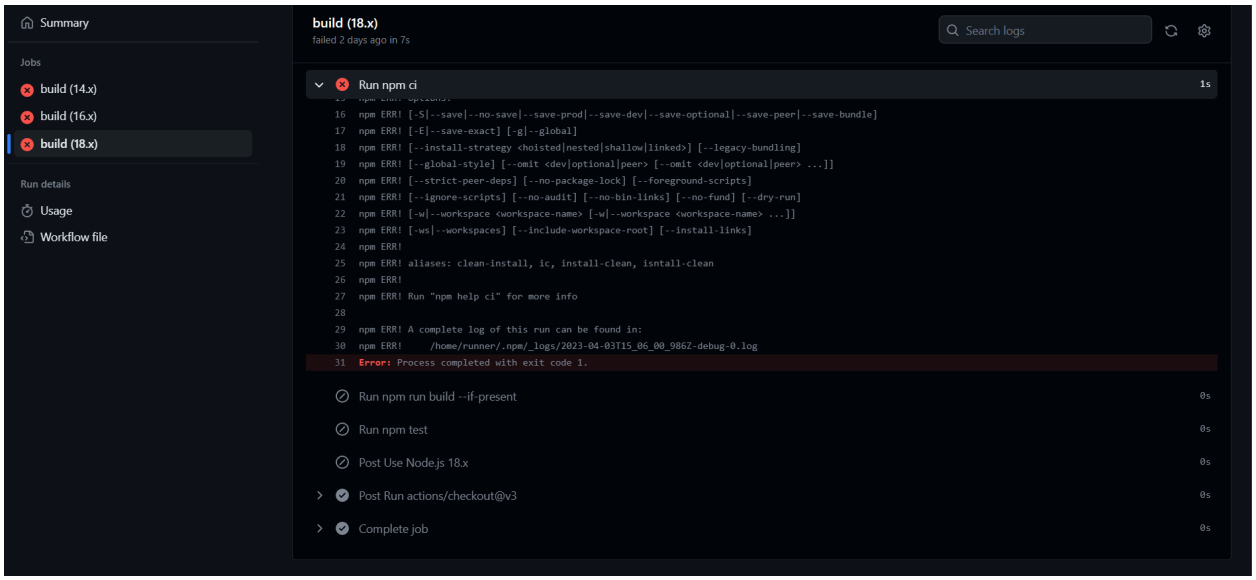


for reviewing code changes and resolving conflicts

Figure 22: GIT Repository

1.8 Deployments/CI-CD Pipeline





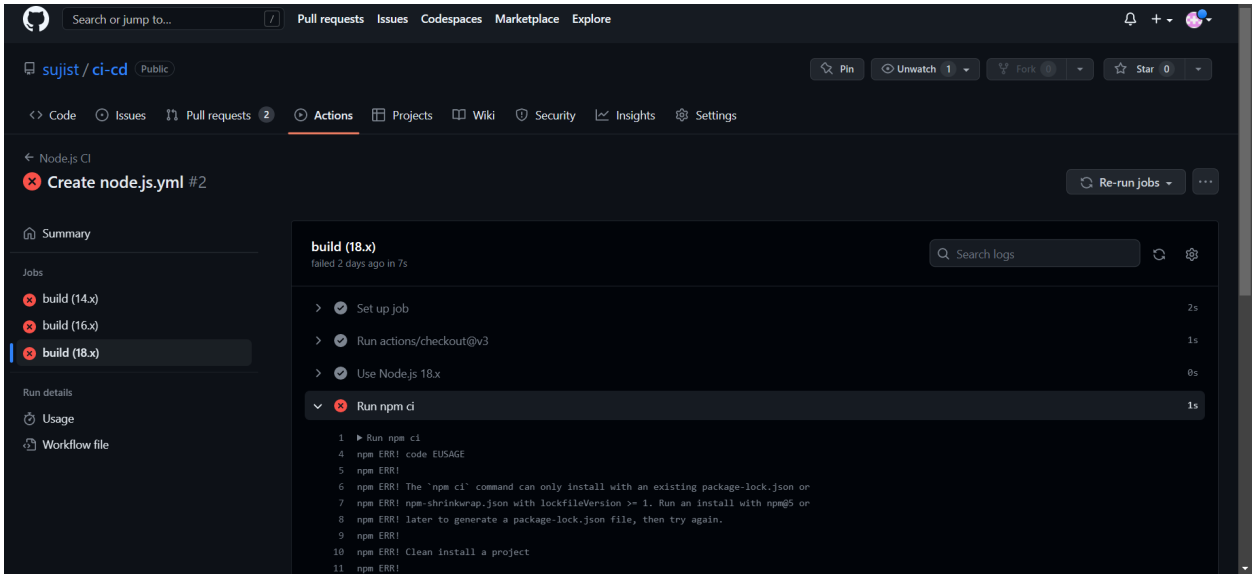


Figure 23: CI/CD Pipeline

The method of automating the integration of code changes from several developers into a single codebase is known as continuous integration, or CI. It is a software development technique in which the teams often commit the work to the main code repository (Github). Finally, upon integration, there are automatic tools that construct the just contributed code, perform a code review, etc.

The main objectives of continuous integration are to detect and fix issues more quickly, make it simpler for the team to integrate our code, increase the caliber of the product, and shorten the time it takes to ship new feature updates.

Even the team members tried to implement ci/cd pipeline to the git repository using Github action and make the process automatic but unfortunately anyone could not fully implement the ci cd pipeline to the repository.

1.9 Chapter Summary

The overview of the prototype, technologies that were selected when implementing this application, implementation of the data science component, implementation of the backend

component, implementation of the front end component, GIT repository and deployments/CI-CD pipeline that are used in this project are discussed in this chapter Methodology. React, Nodejs, Tailwind CSS, Python and MongoDB are the technologies that are used to implement this web application. And how data science components, frontend and backend are implemented are explained in this chapter. The next chapter is discussing how the testing was done for this application.

Chapter 2: Testing

.

2.1 Chapter Introduction

The previous chapter discussed the implementation of the WooHoo application. And this chapter will be discussing how the testing has done to this WooHoo application. The functional and nonfunctional testing requirements are discussed in detail in this chapter. Also the details about Unit testing, Performance testing, Usability testing and Compatibility testing that has been done for the application are discussed in this chapter.

2.2 Testing Criteria

- Accuracy

  This web application has been tested to ensure that it correctly identifies animal skin diseases. Results of this web application are compared to the results of knowledgeable vets to determine how accurate the system is.

- Sensitivity

  The web application should be capable of identifying any potential skin conditions in animals. Using a collection of animals with various skin diseases as dog and cat mange, ringworm, this has been verified.

- Specificity

The animal skin disease detection application must be able to distinguish between different kinds of skin diseases. This is tested by using datasets of dogs and cat skin images with various skin diseases.

- User-friendliness

  The web application should be easy to use and navigate.
- Speed

  The web application has to be quick in order to deliver results immediately. The length of time it takes for the system to return data can be used to verify this.

- Security

  The web application must be secure in order to protect information about users. In order to find system vulnerabilities, security tests can be used to verify this.

- Compatibility

  The web application needs to work with various kinds of devices and browsers. This can be verified by accessing this application on different devices and browsers.

- Reliability

  The web application should be reliable and provide consistent results. Multiple tests on the same dataset can be used to check this.

- Error Handling

  Web applications should efficiently manage errors and provide concise error messages. This can be verified by intentionally creating mistakes and examining the system's error notifications.

2.3 Testing functional requirements

Image upload: Users can upload pictures of animals' skin for the application to analyze.

Image processing: The application is able to process the uploaded photos and identify accurate skin disease that exists in the image.

Disease classification: Depending on the type, severity, and the skin disease location of the identified skin diseases, the application is able to categorize them.

2.4 Testing non-functional requirements

Performance: The application's performance and speed is satisfied with user requests, particularly when handling big image files.

Reliability: There is not much downtime or system problems, and the program is always accessible to users.

Scalability: The application has the capacity to manage a growing number of users and picture submissions, and is able to scale up or down in response to user demand.

Maintainability: The program is simple to update and keep, with clear instructions and help for further growth.

2.5 Unit testing

Functions		Description	Status
UT1	Username and Password (Login page)	Check the username and password validity.	Passed
UT2	Creating a new account	Take users input and save the username and password	Passed

UT3	Navigation bar	Navigate through the pages	Passed
UT4	Upload file compatibility (Upload page)	Does not allow files that are not .png, .jpg or .jpeg to be uploaded.	Passed
UT4	Detects disease (Upload Page)	Displays the disease and the disease card	Passed
UT5	Requires all details (Accommodation page)	Displays error message when user does not fill all the fields	Passed
UT6	Booking free shelter slot (Accommodation page)	Displays confirmation message when user makes a booking	Passed
UT7	Add an organization (Accommodation page)	New organizations can log in and update their available slots	Passed
UT8	Contact the WooHoo team (Home page)	Users can send an email to the WooHoo team directly through the website	Passed

Table 1: Unit Testing test cases

2.6 Performance testing

An animal skin disease detection application received performance testing to ensure that it can manage various loads and utilization levels and that it satisfies the stakeholders' performance expectations. This has made it easier to find and fix performance problems early on in the development process, resulting in a more solid and dependable program.

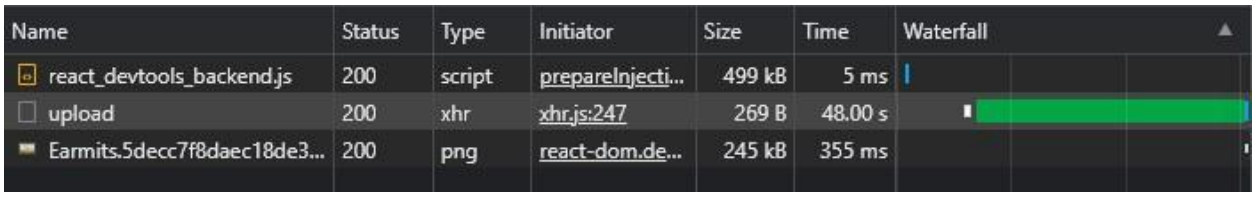


Figure 24: Performance Testing

2.7 Usability testing

Usability testing evaluates how users experience the application and how the user face should be designed. The design of "WooHoo" has been curated with keeping in mind user friendliness and a modern look.

2.7.1 Sign up page

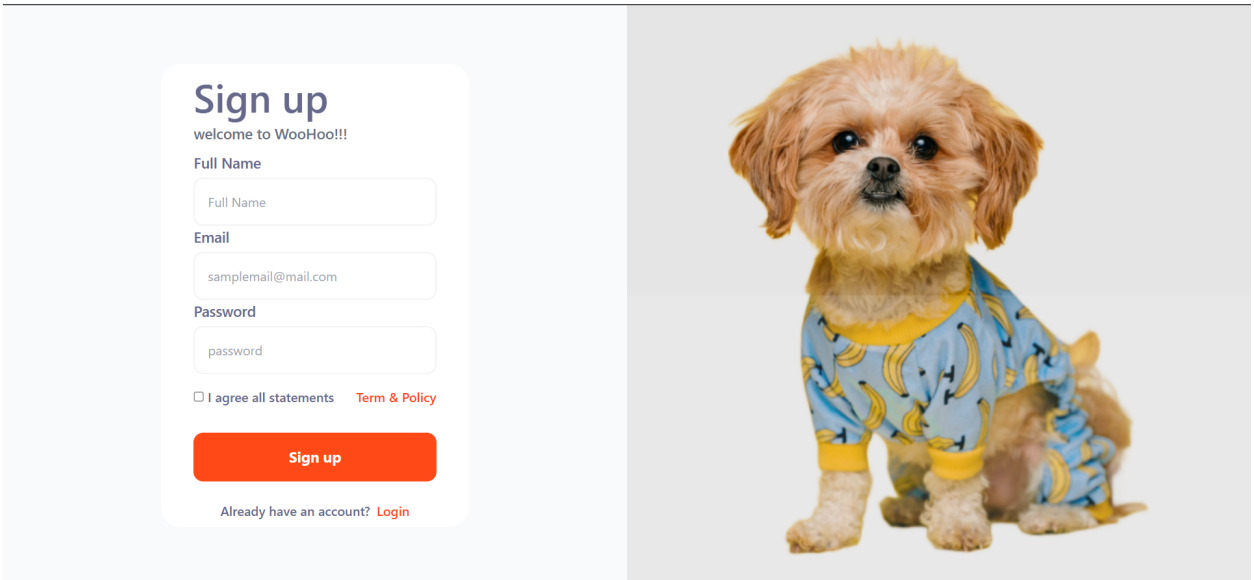


Figure 25: Sign Up page for usability testing

2.7.2 Login page

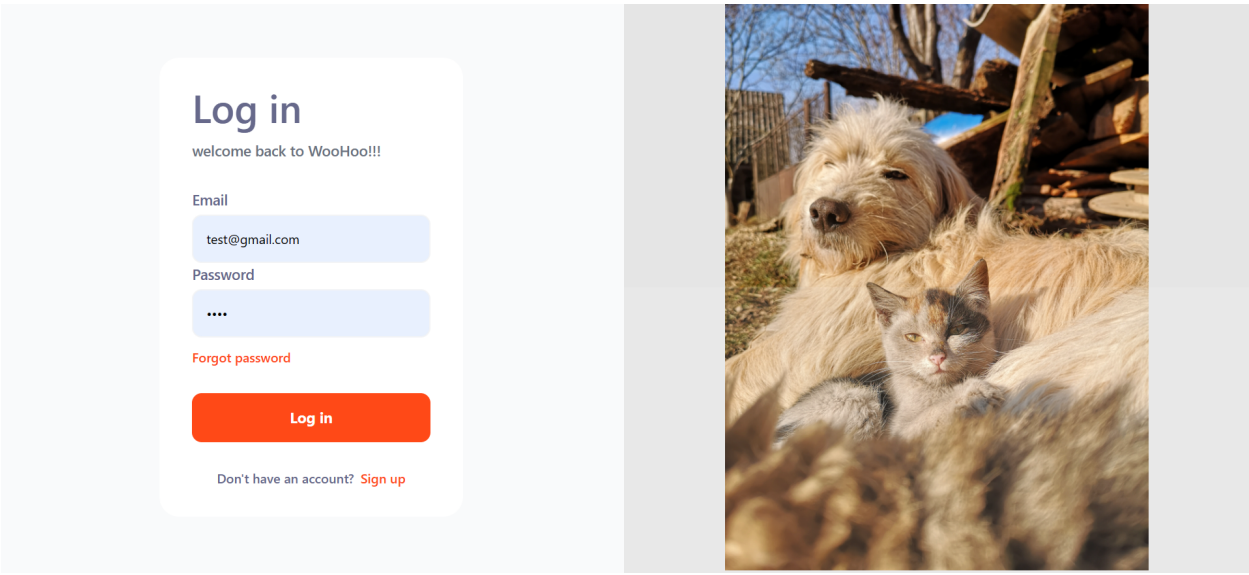


Figure 26: Login page for usability testing

2.7.3 Upload page

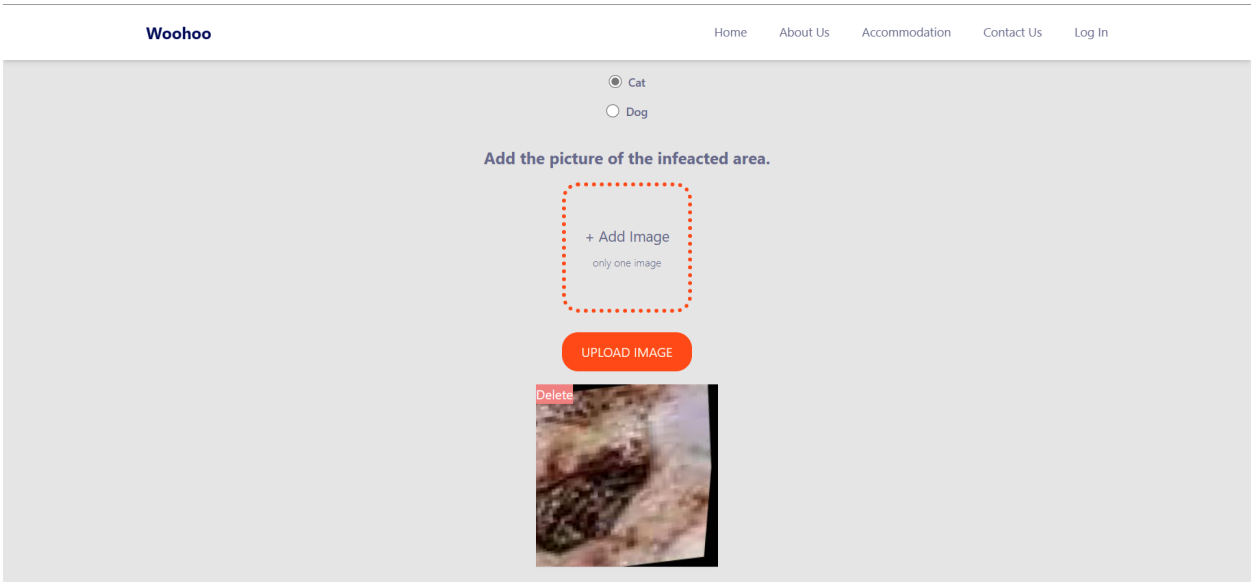


Figure 27: Image upload page for usability testing

2.7.4 Accommodation page

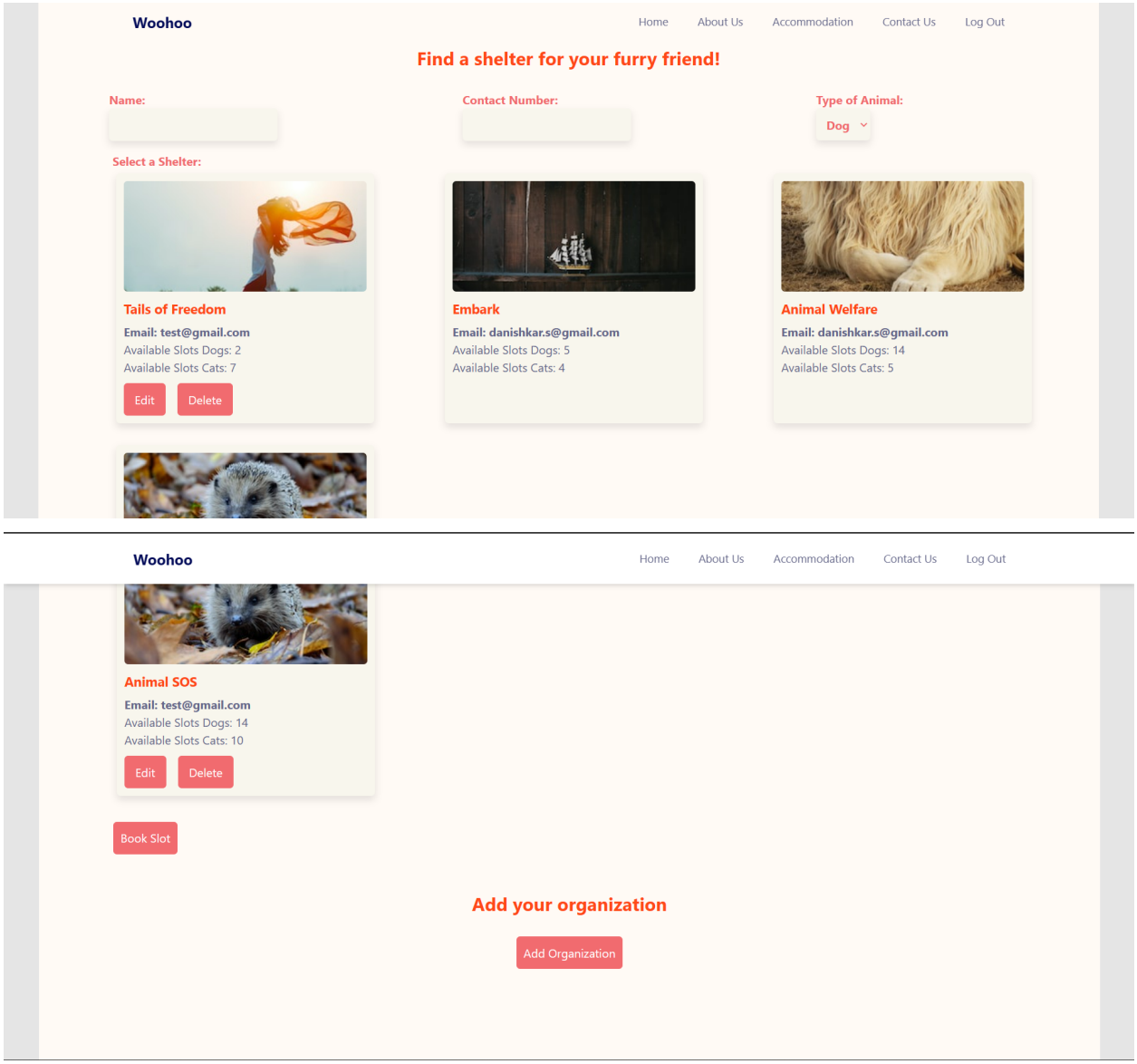


Figure 28: Accommodation page for usability testing

2.7.5 Contact us page

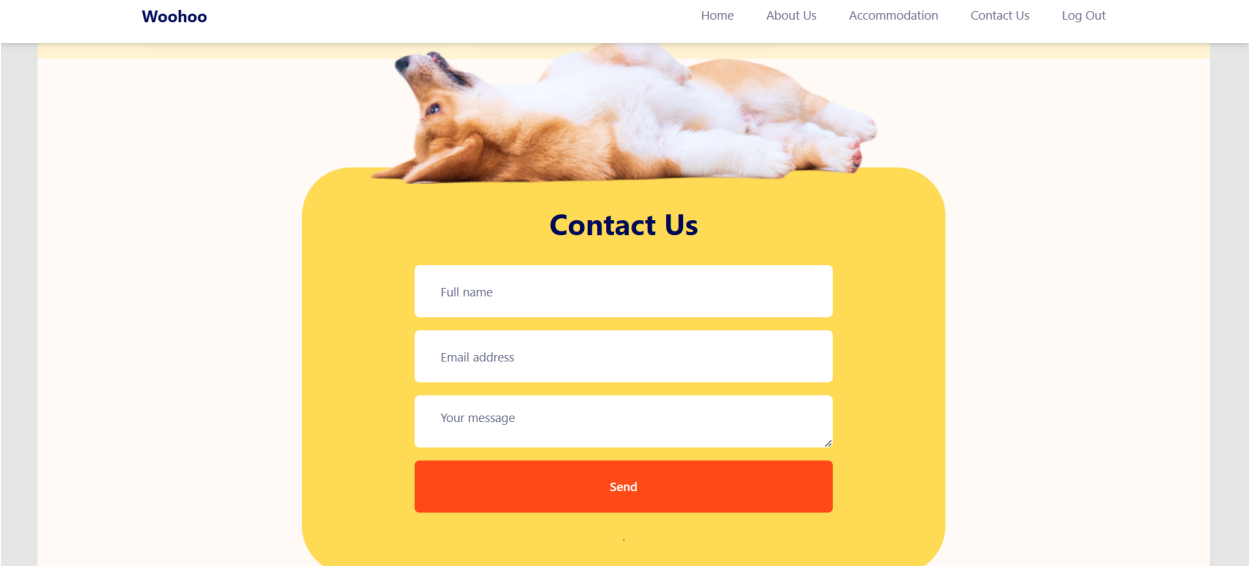


Figure 29: Contact us page for usability testing

2.8 Compatibility testing

This WooHoo application is tested for compatibility to make sure it works with various hardware, software, and network setups and can be used by a variety of users.

Browser	Version	Status	Comments
Microsoft Edge	111.0	pass	Shows all pages that it intends to display and Styles, Page's performance is not changed.
Google Chrome	111.0	pass	Shows all pages that it intends to display and Styles, Page's

			performance is not changed.
Safari	15.3	pass	Shows all pages that it intends to display and Styles, Page's performance is not changed.

Table 2: Compatibility Testing in browsers

OS	**Device**	**Status**	**Comments**
Android	Samsung	passed	Shows all pages that it intends to display and Styles, Page's performance is not changed.
iOS	iPhone	passed	Shows all pages that it intends to display and Styles, Page's performance is not changed.

Table 3: Compatibility Testing in Device

2.9 Chapter Summary

This testing chapter provides a comprehensive overview of the different types of testing that can be performed on the animal skin disease detection application. The chapter then covers the testing of functional and nonfunctional requirements. Also there is a discussion of how Unit, Performance, Usability and Compatibility testing has been done. The next chapter will discuss the Evaluation of the WooHoo application.

Chapter 3: Evaluation

3.1 Chapter Overview

This chapter will be discussing the evaluation methods that have been used in WooHoo application. Qualitative, quantitative and self evaluation that are done for the application is discussed in this chapter.

3.2 Evaluation methods

User surveys : Conducting user surveys are is one of the easiest ways to assess the application.

Expert evaluation : During expert evaluation, the application is reviewed by experts in the field, such as veterinarians or animal researchers, who comment on the application's accuracy, relevance, and general effectiveness.

3.3 Quantitative evaluation

Using numerical data or metrics to measure or evaluate something is known as quantitative evaluation. To quantify the performance, efficacy, or influence of a system, process, or product, it entails gathering and interpreting data. Often, the findings of a quantitative evaluation are presented in terms of numbers, statistics, or other quantifiable indicators.

We have quantitatively evaluated the accuracy of the cat and dog machine learning models using the test dataset.

```python
# Load test data
from tensorflow.keras.preprocessing import image_dataset_from_directory
from keras.models import load_model
base_dir = 'dataset'
model = load_model('cat.model(newOptimiser2)')
test_data = image_dataset_from_directory(base_dir+"/test",
                                         #   image_size=(130,130),
                                         subset='validation',
                                         seed = 1,
                                         validation_split=0.9,
                                         batch_size= 10,
                                         label_mode='categorical'
                                         )

# Evaluate the model on the test data
loss, accuracy = model.evaluate(test_data)
print("Test accuracy:", accuracy)
```

```
Found 112 files belonging to 2 classes.
Using 100 files for validation.
10/10 [==============================] - 1s 59ms/step - loss: 2.0770 - accuracy: 0.7700
Test accuracy: 0.7699999809265137
```

Figure 30: Test Accuracy for Cat Model

```python
# Load test data
from tensorflow.keras.preprocessing import image_dataset_from_directory
from keras.models import load_model
base_dir = 'dataset'
model = load_model('dog.model(newOptimiser2)')
test_data = image_dataset_from_directory(base_dir+"/test",
                                         image_size=(130,130),
                                         subset='validation',
                                         seed = 1,
                                         validation_split=0.9,
                                         batch_size= 10,
                                         label_mode='categorical'
                                         )

# Evaluate the model on the test data
loss, accuracy = model.evaluate(test_data)
print("Test accuracy:", accuracy)
```

```
Found 75 files belonging to 2 classes.
Using 67 files for validation.
7/7 [==============================] - 2s 46ms/step - loss: 1.5243 - accuracy: 0.4627
Test accuracy: 0.46268656849861145
```

Figure 31: Test Accuracy for Dog Model

3.4 Qualitative evaluation (Feedback from end users, domain experts and industry experts)

3.4.1 Feedback from end users

From the survey conducted with end users, the overall feedback for the application was positive. The navigation and simplicity of the application was appreciated. (More info has been included in Appendix A)

3.4.2 Feedback from Domain Experts

In order to collect data sets and expertise from the medical field, veterinarians were approached to fine tune the purpose of project WooHoo.

- Dr. T. K. Pillai (MBBS, Colombo Pet Care Centre)

  "The Code Hustle team group SE-53 of second year students from Informatic Institute of Technology has visited Colombo/Pet Care Centre and requested help regarding a project. They Explained to me about the main scope of the project. I had agreed to help them in the project. Accordingly, I explained to them about some common symptoms found in certain skin diseases of dogs and cats. These could be registered and used to diagnose diseases. These datasets could be used for their project. This type of project will help these students in their future studies."

3.4.3 Feedback from Industry experts

With the experience and advice of industry professionals, the project "WooHoo" has gathered a vast amount of expertise. Several years of expertise and work methods have assisted our team in developing the most recent technologies and procedures to improve the overall quality of this project.

- Mrs. M.S. Weerasinghe (Senior Software Engineer, :Different, Colombo)

  "The website was very user-friendly, and it is easy to navigate. Overall, I was very impressed with the project considering it has been done by undergraduate students. One suggestion for improvement would be to incorporate a feature that allows users to track their uploaded images and view the results of previous diagnoses, and the accuracy of each diagnosis."

3.5 Self evaluation

3.5.1. T. Sujana

As a team we equally divided the workload among the members where I was supposed to work on the frontend of the website and develop the dog model using CNN from the already developed cat model from our team member. I used figma to design the frontend parts (login , sign up)as i was completely new to this technology at the beginning i was helpless and confused what to do then later i took some time to self learn frontend tools like figma by watching tutorials online and some notes from online sites after getting familiar with the basics it was much easier to work on the tool and design the website. And in order to create a new machine learning model i have to learn a lot of theory and practical stuffs i used tools like tensorflow ,keras and i used many libraries such as numpy ,cv2 , matplot and so on, the most challenging part was the dataset since we did not have sufficient datasets online we started creating our own dataset to create our machine learning model.

3.5.2. S. Danishkar

I mainly worked on the integration of the frontend and backend for the upload page, the accommodation page, fully developed the home page and the cat model. For the model I used

keras and tensorflow to create the CNN architecture. Came up with design for the home page and also integrated the contact us page using Email.js where when the user fills out the form and submits the email will be sent to both WooHoo and the user. For the accommodation page added an edit and delete button for the organization which has the same email as the currently logged in user so that the user can update the number slots available and the email associated with the organization and also delete the organization if needed. The most challenging part was the upload page integration to the backend because the model was created using python and the backend routing is done using Node.js therefore I used an npm package called the "child_proceess" which allows to run python scripts in Node.js using this I was able to get the predict outcome for the image with the disease and display it back to the user.

3.5.3. A.K.B.J. Perera

When implementing this project WooHoo, I was supposed to implement the frontend of the image uploading page where the user can upload the image of an animals' skin disease. The frontend component was built using React and Nodejs that can be used for building user interfaces. Tailwind CSS is used to style the frontend components. The web page is made up of several different components, such as a form that lets users upload an image from a gallery, a preview that shows a thumbnail of the uploaded image and an upload icon that the image gets uploaded to the server. The form component is created using React on the frontend and has a file input area where users can pick the picture file. Tailwind CSS classes are used to design the image and container and the preview component shows a thumbnail of the user-uploaded picture. Since I didn't have any ideas about Nodejs, Tailwind CSS and React, I had to watch and go through a lot of videos and tutorials to get the knowledge of those technologies. Since we were lacking datasets to get the project done, we had to collect pictures of various animal skin diseases from websites and create the datasets.

3.5.4. P.M.C.N. Jayarathna

As a developer working on the front-end and back-end of the About Us and Sign Up pages, I am proud of the work that I have done. On the front-end, I have created a visually appealing and easy-to-use interface for the user. I have ensured that the user experience is seamless and that the pages are responsive and work well on all devices. On the back-end, I have created robust APIs that handle the data input and validation, ensuring that user data is handled securely and efficiently. I have also implemented error handling and logging to detect and resolve any issues that may arise. Overall, I feel that the pages are functional, user-friendly, and meet the requirements of the project. However, I am constantly looking for ways to improve my code and make the pages even better. I will continue to work on enhancing the pages and incorporating feedback to ensure that they meet the highest standards of quality and user experience.

3.5.5. R.A. Kumarawadu

For this project, I worked on the front-end development of our website using React, NodeJs and Tailwind CSS. One of the key challenges that I faced was learning React, as I had not worked with this technology before. However, I was able to quickly get up to speed by completing online tutorials and working closely with my team members who had more experience with React. I developed the accommodation page which consists of a form to take the users details and then show them the available shelter slots. An organization can log into the website and update their available slots in the accommodation page. Project WooHoo also gave me more insight into the world of machine learning and I found that working with Tensorflow and Keras was both challenging and rewarding. These libraries are incredibly powerful and versatile, but also require a deep understanding of machine learning concepts and best practices in order to use them effectively.

3.6 Chapter Summary

This chapter discussed the evaluation methods that have been done such as user surveys and expert evaluation. Also about the Qualitative evaluation, quantitative evaluation and self evaluation are also adequately outlined in this chapter. The achievements, future enhancements of this WooHoo application will be discussed in the next chapter, ‘Conclusion’.

Chapter 4: Conclusion

4.1 Chapter Overview

This chapter will be discussing how the aims and objectives of the SRS report have been achieved and also the limitations of the research and the areas that couldn't be completed. Also the future improvements that are planned to do for this WooHoo application are also stated in this chapter.

4.2 Achievements of aims and objectives

This project successfully accomplished the following steps within the allocated time.

1. Project Initiation - The initial goal was to identify an existing problem with a goal, scope, goals, and recommended remedy. The SRS report's chapter 1 describes all of the requirements.

2. Literature Review - The second objective is to locate all existing work linked to the specified topic after studying each and every facet of the probable issue domain. Chapter 2 of the SRS report contains a full report on how this has been accomplished.

3. Project Management- Following the analysis of the Literature Review, the next goal was to outline techniques in depth. To move on with the project, the Iterative model was chosen as the software development technique. The progress of team members was tracked using the project management software "ClickUp".

4. System Requirements Specification - The following stage of the project is to collect study results, such as questionnaires. Google form surveys were circulated to pet owners and university students in Sri Lanka. The data was used to identify the project's functional and nonfunctional needs. All of the specifics may be found in Chapter 4 of the SRS report.

5. This chapter describes the Legal, Social, Ethical, Professional Issues and the Data Set as to conclude the previous chapter. Detailed description is in Chapter 6 of SRS report.

6. Design - This phase is to design the architecture and Design (Class Diagram, Sequence Diagram, UI Design, Activity Diagram) of “WooHoo”. In the Chapter 6 of SRS report all the diagrams were provided.

7. Implementation - Within the phase every detail about Implementation Technologies that are used to build the project are being reviewed. Challenges that are faced through the implementation period.

8. Testing - Project “WooHoo” is tested within this chapter. Functional and nonfunctional requirements are tested. All tests are tested using Qualitative and Quantitative methods.

9. Evaluation – Quantitative and Qualitative methods are used in the phase of evaluation of the “WooHoo” web application to also get an evaluation from End-users, Domain experts and Industry experts by demonstrating the image classification system. All the details are provided in chapter 3 of the Implementation report domain.

4.3 Limitations of the research

Accuracy

Some skin conditions are harder to identify than others, and the algorithm's accuracy also is changing based on the complexity of the condition.

Additionally, there are differences in skin conditions depending on the breed, age, and other features of the animal, which can impact the algorithm's accuracy. The environment, lighting, and image clarity may also have an impact on the algorithm's precision.

Lack of Datasets

This WooHoo application is an animal skin disease detection application where there needs to be more information, skin disease picture datasets. With the lack of datasets, it was very challenging to begin the project and also update the database that carries the information. Because of this, it is difficult to recognize many skin diseases.

4.4 Future enhancements

Development of a mobile app for the WooHoo application : Since this animal skin disease detection application is a web application, the team is planning to develop a mobile application for the WooHoo web application. By creating a mobile application, users will be able to access it more conveniently and while they are on the go.

Adding an adoption feature : Adding an animal adoption feature to the WooHoo application would significantly extend its reach and objectives of the application's scope and purpose. It will be a valuable addition that could help to promote animal welfare and reduce animal homelessness. Also it would help to promote responsible pet ownership.

Recommending a diet plan to the pets of pet owners : WooHoo application could provide a more comprehensive approach to detecting animal skin diseases and promoting animal health by

developing a feature to recommend a diet plan for the pets. So that pet owners can plan their pet's meals more healthily and easily.

Ability to detect more skin diseases : Even though the current application can only detect some most common skin diseases, in the future the team is planning to expand the ability to detect more skin diseases that could provide valuable benefits for animals and pet owners.

4.5 Extra work (Competitions, research papers, etc).

Microsoft Imagine Cup 2023

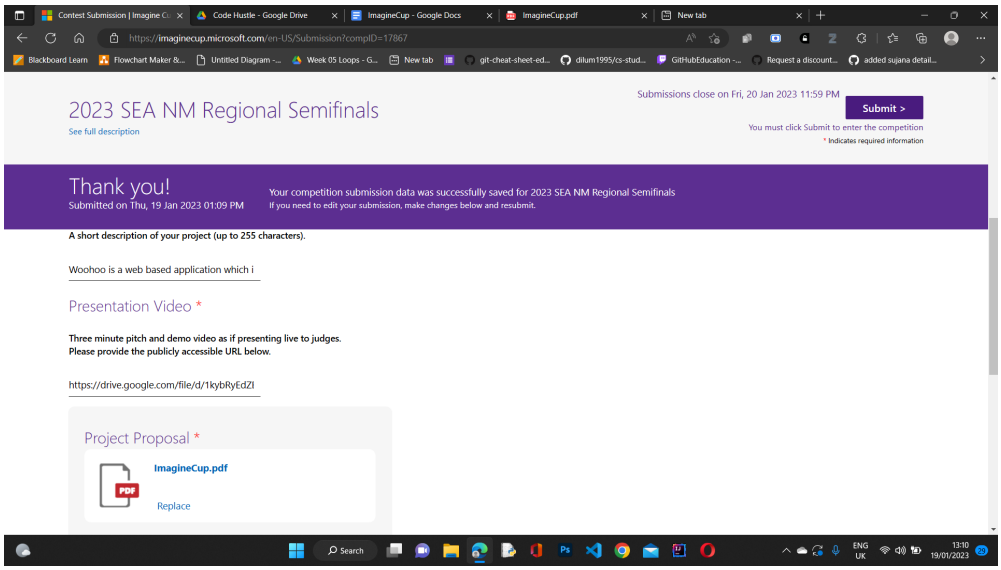


Figure 32: ImagineCup Submission

IEEE Codesprint 2023

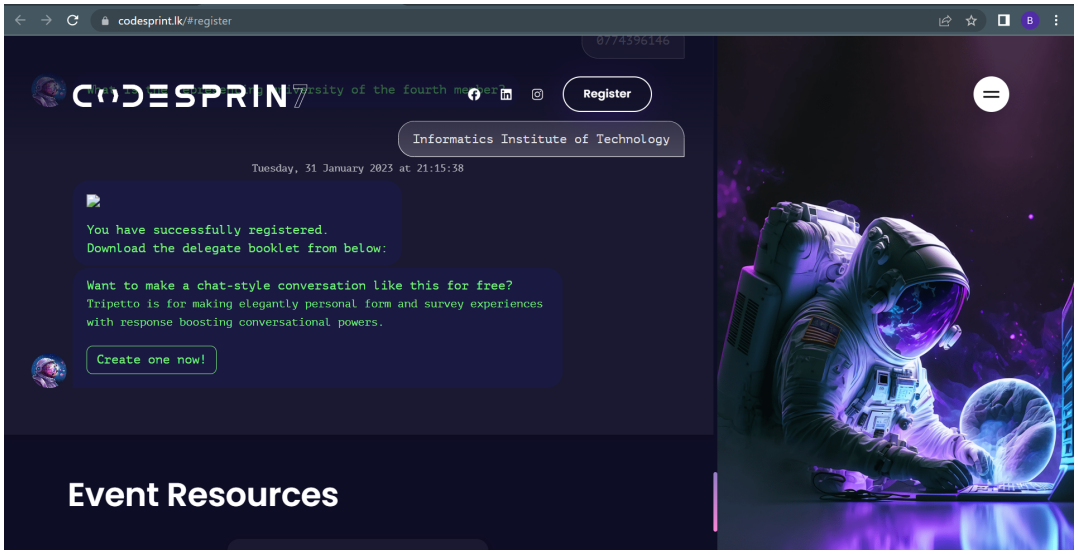


Figure 33: CodeSprint Submission

4.6 Concluding remarks

This project WooHoo for detecting animal skin diseases has been effectively developed, tested, and evaluated. Users can upload pictures of their dogs' or cats' skin disease conditions and get an automated diagnostic using the application's user-friendly UI. Functional and nonfunctional criteria, unit testing, performance testing, usability testing, and compatibility testing were all part of the testing procedure. Both quantitative and qualitative evaluation techniques, such as user questionnaires and comments, were used during the evaluation process.

Overall, the application proved to be accurate and reliable at identifying common skin diseases among animals. And also users can help stray animals to find a shelter through the Accommodation feature of this application However, there will be future enhancements that could include adding features such as an animal adoption feature or a diet plan recommendation feature. It would be important to collaborate with animal specialists and create an important image library to increase the application's capacity to identify more skin illnesses.

In conclusion, this web application for detecting animal skin diseases offers a helpful tool for veterinarians and pet owners to manage the health of their animals. Animal health and wellbeing may be significantly affected, and it may speed up and more accurately diagnose skin problems in animals.

References

- Hwang, S. *et al.* (2022) *Classification of dog skin diseases using deep learning with images captured from Multispectral Imaging device - molecular & cellular toxicology*, *SpringerLink*. Springer Nature Singapore. Available at: https://link.springer.com/article/10.1007/s13273-022-00249-7 [Accessed: March 2, 2023].

- Jusman, Y. and Nurkholid, M.A.F. (2020) *Expert system for detecting cat skin disease using certainty ... - umy*, *Expert System for Detecting Cat Skin Disease using Certainty Factor Method*. Available at: https://journal.umy.ac.id/index.php/jet/article/download/9458/5579 [Accessed: March 12, 2022].

- M. Y. Munirah, S. Suriawati and P. P. Teresa (2016) *Design and development of online dog diseases diagnosing system - ijiet*, *Design and Development of Online Dog Diseases*

Diagnosing System. Available at: http://www.ijiet.org/vol6/816-C1004.pdf [Accessed: March 17, 2022].

- Mellores, R.M.M. (2020) *[PDF] Android-based application utilizing image processing with artificial neural network for detecting ringworm and yeast infections for dogs using neuroph framework: Semantic scholar*, *International Journal of Advanced Trends in Computer Science and Engineering*. Available at: https://www.semanticscholar.org/paper/Android-based-Application-Utilizing-Image-with-for-Mellores/2dcfeb239e16bf9bf6fbb48990f818c431f60aca [Accessed: March 12, 2022].

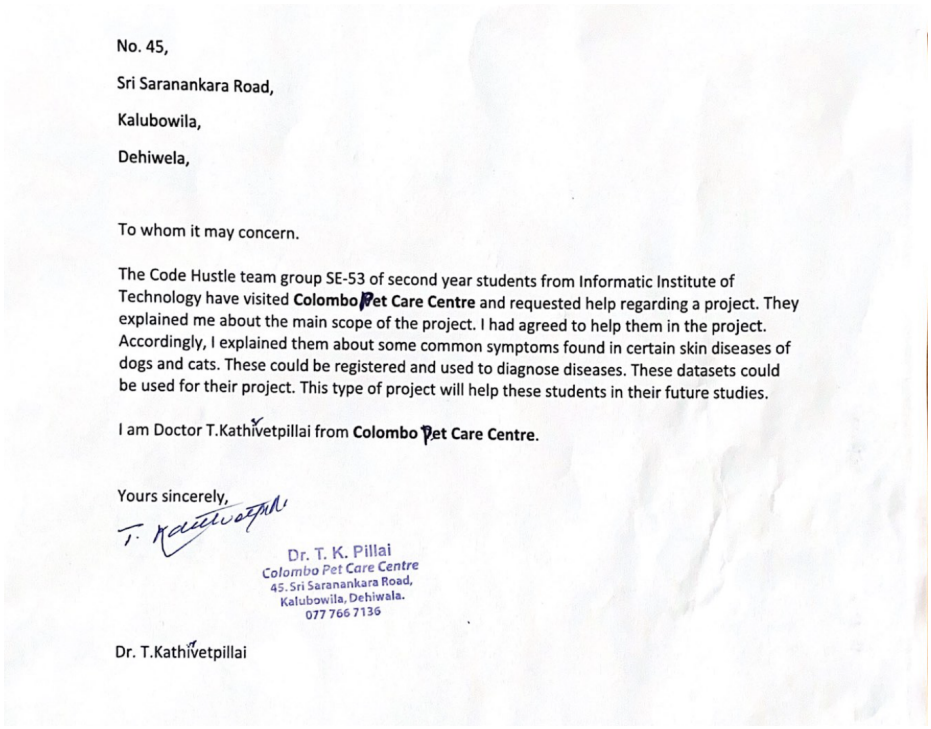
- Velasco, J. (2019). A Smartphone-Based Skin Disease Classification Using MobileNet CNN. *International Journal of Advanced Trends in Computer Science and Engineering*, pp.2632–2637. doi:10.30534/ijatcse/2019/116852019.[Accessed: March 12, 2022].

- Velasco, J. (2019). A Smartphone-Based Skin Disease Classification Using MobileNet CNN. *International Journal of Advanced Trends in Computer Science and Engineering*, pp.2632–2637. doi:10.30534/ijatcse/2019/116852019.[Accessed: March 12, 2022].

Appendix

Appendix Section A - Evaluation survey

A survey was sent to pet owners and shelter workers. This evaluation survey was conducted to get a better understanding of the feasibility of "WooHoo" and to analyze the expectations users have before developing our system. Screenshots of the survey results are given below.



No. 45,

Sri Saranankara Road,

Kalubowila,

Dehiwela,

To whom it may concern.

The Code Hustle team group SE-53 of second year students from Informatic Institute of Technology have visited **Colombo Pet Care Centre** and requested help regarding a project. They explained me about the main scope of the project. I had agreed to help them in the project. Accordingly, I explained them about some common symptoms found in certain skin diseases of dogs and cats. These could be registered and used to diagnose diseases. These datasets could be used for their project. This type of project will help these students in their future studies.

I am Doctor T.Kathivetpillai from **Colombo Pet Care Centre**.

Yours sincerely,

Dr. T. K. Pillai
Colombo Pet Care Centre
45, Sri Saranankara Road,
Kalubowila, Dehiwala.
077 766 7136

Dr. T.Kathivetpillai

Figure 34: Feedback from domain expert

Did the machine learning model accurately detect the skin diseases in the images you provided? Were there any instances where it incorrectly identified a disease or failed to identify a disease?

5 responses

- it detected correctly
- The result that got provided is accurate.
- The ml model accurately detected the skin disease.
- It correctly identified the skin disease.
- The skin disease was accurately got detected.

Is there anything else you would like to add or suggest for improvement?

3 responses

- no
- No
- Developing a mobile application for this website

How user-friendly was the website? Was it easy to navigate and use?

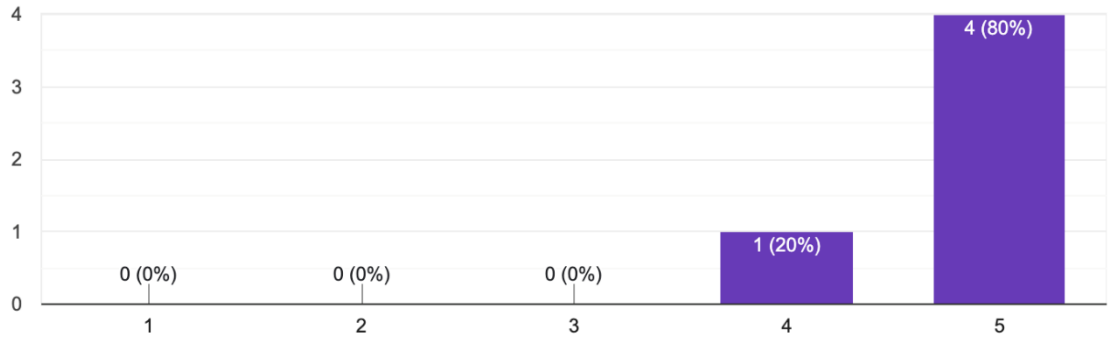
5 responses

- very easy
- It was so good
- It was easy to use.
- The website is user friendly
- The website was easy to use and it was user friendly

How would you rate the overall quality of the website?

Copy

5 responses



Were there any features or information that you think should be added to the website?

3 responses

add the adoption soon

Adding an adoption feature for pet lovers.

A diet plan recommendation for pets.