

CSE1901 - Technical Answers to Real World Problems (TARP)

Project Report

Woof: A Pet Care System

By

Sriya Nanduri : 19BCE1184
Tejas Vaichole : 19BCE1295
Vedang Sawarkar : 19BCE1303
Sanika Kulkarni : 19BCE1328
Divyanshi Thapa : 19BCE1367

B. Tech Computer Science and Engineering

Submitted to

Dr. Geetha S

School of Computer Science and Engineering



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

April 2022

DECLARATION

We hereby declare that the report titled “**Woof: A pet care system**” submitted by me to VIT Chennai is a record of bona-fide work undertaken by me under the supervision of **Dr. Geetha S**, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai.

Signature of the Candidates



Sriya Nanduri
Reg No. 19BCE1184



Tejas Vaichole
Reg No. 19BCE1295



Vedang Sawarkar
Reg No. 19BCE1303



Sanika Kulkarni
Reg No. 19BCE1328



Divyanshi Thapa
Reg. No. 19BCE1367

CERTIFICATE

Certified that this project report entitled “**Woof: A pet care system**” is a bonafide work of **Sriya Nanduri(19BCE1184)**, **Tejas Vaichole(19BCE1295)**, **Vedang Sawarkar(19BCE1303)**, **Sanika Kulkarni(19BCE1328)**, **Divyanshi Thapa(19BCE1367)** and they carried out the Project work under my supervision and guidance for CSE1901 - Technical Answers to Real World Problems (TARP).

Dr. Geetha S

SCOPE, VIT Chennai

ACKNOWLEDGEMENT

I, student of Vellore Institute of Technology, Chennai campus with name Divyanshi Thapa, registration number 19BCE1367 has carried out the Project work along with four others under the supervision and guidance of Dr. Geetha S for CSE1901 - Technical Answers to Real World Problems (TARP). I express my gratitude and acknowledge Dr. Nithyanandam P, Head of the Department (HoD), B.Tech Computer Science and Engineering .SCSE, VIT Chennai , Dr. Ganesan R, Dean of the School of Computer Science & Engineering, VIT Chennai , Dr. Geetha S, Associate Dean of the School of Computer Science & Engineering and my subject teacher, VIT Chennai for giving me this opportunity of implementing my theoretical knowledge to solve a real world problem. I also acknowledge my parents, friends and teammates for their constant support and guidance.

Signature of the Candidates



**Sriya Nanduri
Reg No. 19BCE1184**



**Tejas Vaichole
Reg No. 19BCE1295**



**Vedang Sawarkar
Reg No. 19BCE1303**



**Sanika Kulkarni
Reg No. 19BCE1328**



**Divyanshi Thapa
Reg. No. 19BCE1367**

ABSTRACT

Woof (A pet care system) is an alternative to ruthless shelter homes or expensive pet care centers, where the pet owner can scroll through dozens of pet lovers with whom he wishes to keep his pet. The main aim of our platform is to relieve pet owners from the guilt associated when traveling without their pet or leaving them in a boarding kennel. On our platform, the pet lover has to make an account and set up their profile mentioning where they stay, their experience with the types of pets, their cost to keep different kinds of pets, and whether or not they will provide food during the stay. Once a profile has been set up, it will be visible in the listings to the pet owner. Pet owners can make a decision based on ratings and reviews as to whom they want to keep their pet with.

The pet lovers can register themselves by depositing a security amount, identity proof, and other details. Their years of experience and ratings will move them on top of the search result of the pet owner.

CONTENTS

Declaration	i
Certificate	ii
Acknowledgement	iii
Abstract	iv
1 Introduction	1
1.1 Objective and goal of the project	1
1.2 Problem Statement.	2
1.3 Motivation	3
1.4 Challenges	5
2 Literature Survey	6
3 Requirements Specification	7
3.1 Hardware Requirements	8
3.2 Software Requirements	9
4 System Design	10
5 Implementation of System	
6 Results & Discussion	
7 Conclusion and Future Work	
8 References	
Appendix	

1. Introduction

Industry estimates show there are around 19 million pets in India (around 80 percent of these are dogs), and on average, 6,00,000 pets are adopted every year. Pets become a part of the family the moment they step into our house. Pets become very used to human company, unlike stray animals who usually live on their own. It's a sad fact for pet owners that vacation planning can be such a hassle. So, we came up with the idea of a pet care system to fill in the void between pet owners, pet lovers, and the pet. There are many people who are excellent pet caretakers but cannot own a pet due to their work lives. But such people are willing to take care of others' pets during weekends or during their free time. Connecting such people with pet owners and creating and expanding this community is the main goal of this solution.

1.1. Objective and goal of the project

Woof is a platform that connects pet lovers and pet owners where the pet owner will be able to choose a pet lover nearby his place who is willing to take care of his pet on his behalf. The main aim of our platform is to relieve pet owners from the guilt associated when traveling without their pet or leaving them in a boarding kennel. On our platform, the pet lover has to make an account and set up their profile mentioning where they stay, their experience with the types of pets, their cost to keep different kinds of pets and whether or not they will provide food during the stay. Once a profile has been set up, they will be visible in the listings to the pet owner. The pet owners can make a decision based on ratings and reviews as to whom they want to keep their pet with.

1.2 Problem Statement

Pets provide emotional support to many people around the world. But they need constant care from people they stay with. Rapid urbanization, coupled with rising disposable incomes and a shift to nuclear families, are driving more people to get pets. The pet and the pet owner kind of become dependent on one another. Owning a pet

comes with certain responsibilities and lifestyle impacts. Most people cannot go on a vacation or even someplace nearby unless there is someone there at home to take care of them. Owning a pet by a nuclear family brings in a lot of lifestyle restrictions to the family. The entire family cannot go out on vacation as someone has to stay with the pet. Nearly 23 percent out of 2000 surveyed take a vacation once every few years, some even less often than that.

1.3 Motivation

The key motivation to develop Woof was to help pet owners enjoy a peaceful vacation and the pet lovers who cannot own a pet for any reason to be able to own a pet just for a few days. The Woof application connects the pet owners to pet lovers or carers near them by using geolocation through a booking system. The nearest pet lovers are displayed to the pet owner based on both their geo locations. The amount is charged by the pet lover on a per day basis. One unique feature of Woof is the tracker. The pet owners can track their pet through their phone to ensure that their pet is safe.

1.4 Challenges

The key challenges that were faced while developing the Woof application are :

1.4.1 Software

Satisfying all the software and version requirements of node packages.

1.4.2 Hardware

The tracker is a unique feature of the Woof application, but it came with certain challenges.

- Difficulty in finding the GSM module online as it was mostly not deliverable or out of stock.
- The GPS module which would send and receive the trackers coordinates faced connectivity issues which were solved later.

- Placing the tracker on stable level ground due to bulkiness

2. Literature Survey

Design of handheld positioning tracker based on GPS/GSM

GPS positioning technology and GSM wireless communication technology have been widely used in the military field and urban transportation and other civilian areas. Aiming at the singularity of handheld positioner function on the market, this paper designs a combination device of handheld positioner and tracker. By extracting GPS navigation information frame parameters and GSM wireless data transmission. The realization of the tracker in the TFT LCD screen real-time display itself and the tracking side (i.e., the position of the relative position information) function, completed conversion from latitude and longitude coordinates to Cartesian coordinates. The device can be used for the loss of items, the elderly children tracking and geo-location data mapping, data accuracy, and easy operation.

Designing a Low-Cost Location Tracker for Use in IoT Applications

This paper reports on a new low-cost location tracker design, utilizing GPS/BeiDou and 2G rather than the relatively costly, and much more limited coverage, 4G cellular solution. The target retail cost of the tracker is 7 Euro or less. The target market is in IoT asset tracking applications. The tracker has been successfully demonstrated in a pilot test in China for tracking 50,000 auto parts boxes.

2.1 Our Survey

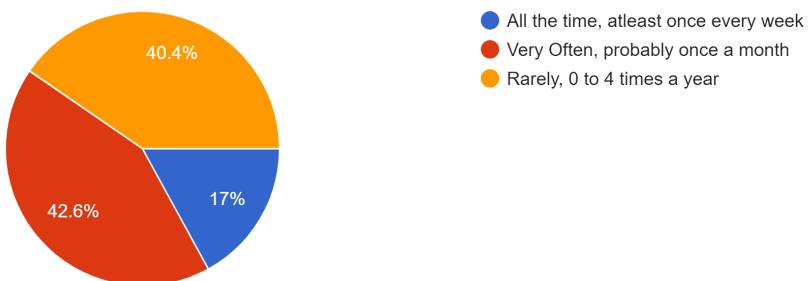
Would you be willing to take care of someone else's pet in your free time and earn some money from it ? If yes do you have a pet of your own as well?

49 responses



How often did you take a day off to explore places around your neighborhood or a vacation before the pandemic ?

47 responses



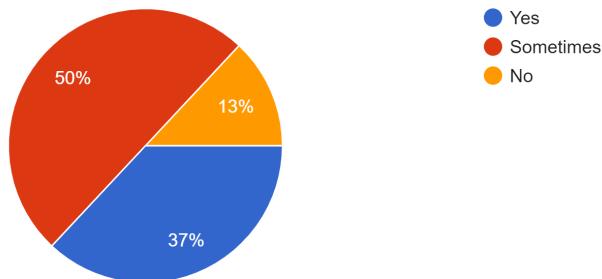
Did you take your pets along with you for a long vacation before the pandemic?

41 responses



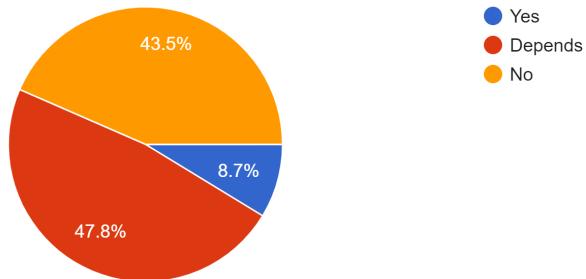
Does having a pet make it difficult to plan vacations?

46 responses



Do you trust your local pet care centers while you are away for a vacation?

46 responses



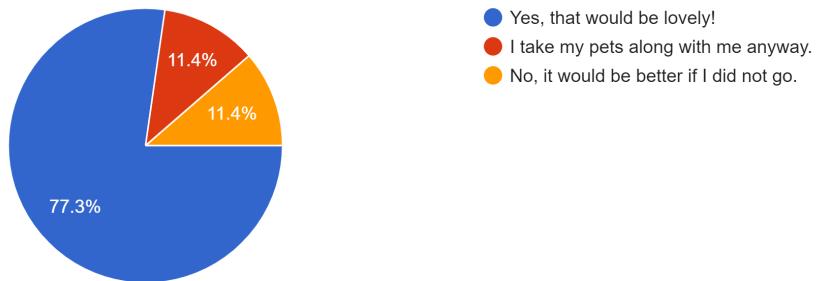
Does it bother you when you leave your pet behind with someone while you go for a vacation?

43 responses



Would you go on a vacation if there is some trustworthy, loving and experienced person to take care of your pets while you are away?

44 responses



3. Requirements Specification

3.1 Hardware Requirements

- Arduino Uno
- SIM808 GPS GSM module
- 9V Battery/ adapter
- Wires

3.2 Software Requirements

- Arduino IDE
- Visual Studio Code
- Node Package Manager
- Node packages

4. System Design

4.1 Software

The website has been hosted on Heroku platform. On visiting the website, the home page will be rendered displaying what WOOF is all about. This page will have separate buttons to login and signup as pet owner or as pet caretaker.

SignIn as Pet Owner-

Once signed in as pet owner, he/she will have options to view his/her profile, view the about page of WOOF, view the previous bookings and view the services offered by WOOF.

View Pet Owner's Profile-

On visiting the profile page, the pet owner will be able to view his/her profile as well as the pet's profile and change the password. On clicking on the edit button, the pet owner will be able to edit his/her profile along with the profile image as well as edit the pet's profile such as age.

View Bookings by Owner-

The owner will be able to view all his previous bookings along with the names of the pet caretakers, booking date, location,etc.

Pet Boarding Service-

Profiles of the pet care takers' living in the vicinity of 16-17 Km range to the pet owner, will be displayed in sorted fashion such that care takers nearest to the pet owners at top and those far away at the bottom. On clicking on the 'book' button of any one of the caretakers, the user will be directed to the profile page of the caretaker where the user will select the boarding date and the last date when he/she wants to collect back the pet. Based upon the number of days, the cost will be calculated automatically. Once the user clicks on the 'Confirm Booking' button, he/she will be redirected to the payment gateway.

Pet Tracking Service-

This service allows the owner to get the live location of the pet. The tracker connected to the pet will send the location coordinates to the server, using which an API will show the exact location on the map.

SignIn as Pet Caretaker-

On signing in as pet caretaker, the home page will have two options- view his/her profile and view booking made to him. The caretaker will also be able to view the about page of WOOF.

View Pet Caretaker's Profile-

On visiting the profile page, the pet caretaker will be able to view his/her profile and change the password. He/She will also have a button to disable his/her availability so that his/her profile won't be displayed to any of the pet owners while searching. On clicking on the edit button, the pet caretaker will be able to edit his/her profile along with the profile image.

Booking made to the Pet Caretaker-

On clicking on this button present on the home page by the pet caretaker, he/she will be able to view all the past and current bookings made to him.

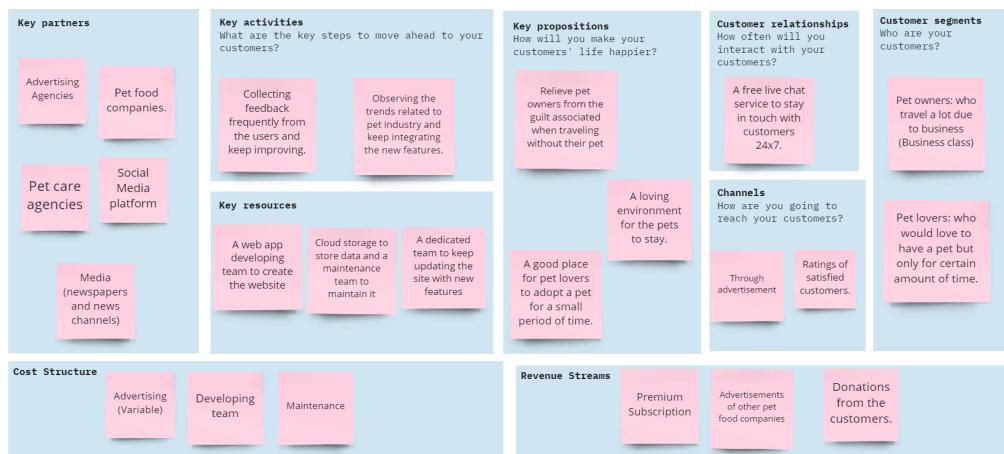
4.2 Hardware

For making the tracker part, the Arduino board is programmed using Arduino IDE. It is the master board which controls the incoming and outgoing signals of SIM808 GPS GSM module. SIM808 module is a complete Quad-Band GSM/GPRS module which combines GPS technology for satellite navigation. The compact design which integrated GPRS and GPS in a SMT package will significantly save both time and costs for customers to develop GPS enabled applications. Featuring an industry-standard interface and GPS function, it allows variable assets to be tracked seamlessly at any location and anytime with signal coverage. It features ultra-low

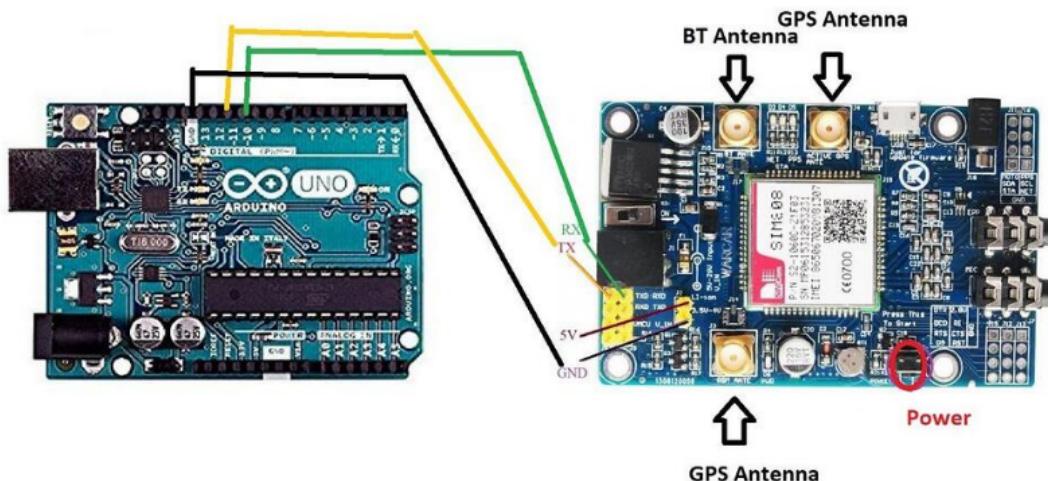
power consumption in sleep mode and integrated with a charging circuit for Li-Ion batteries, that make it get a super long standby time and is convenient for projects that use rechargeable Li-Ion batteries. It has high GPS receive sensitivity with 22 tracking and 66 acquisition receiver channels. Besides, it also supports A-GPS that is available for indoor localization. The module is controlled by AT command via UART.

5. Implementation of System

5.1 Planning:



5.2 Circuit diagram:



The RX pin of SIM808 is connected to the PIN 10 of Arduino UNO and the TX pin of SIM808 is connected to PIN 11 of the UNO board. The power supply of SIM808 should be 9v. This is given using an adapter.

The main idea of using SIM808 for tracking instead of NodeMCU is that it does not require the internet to send messages. As soon as the pet owner pings the sim which is inserted in the module, the location is sent to his mobile phone via the GSM module of SIM808. It uses satellite connection whereas in NodeMCU, we have to make sure that the pet is in the internet vicinity to get the position or send the message. The GPS module of SIM808 gets the information about the location and this is communicated to the owner via GSM.

6. Results and Discussion

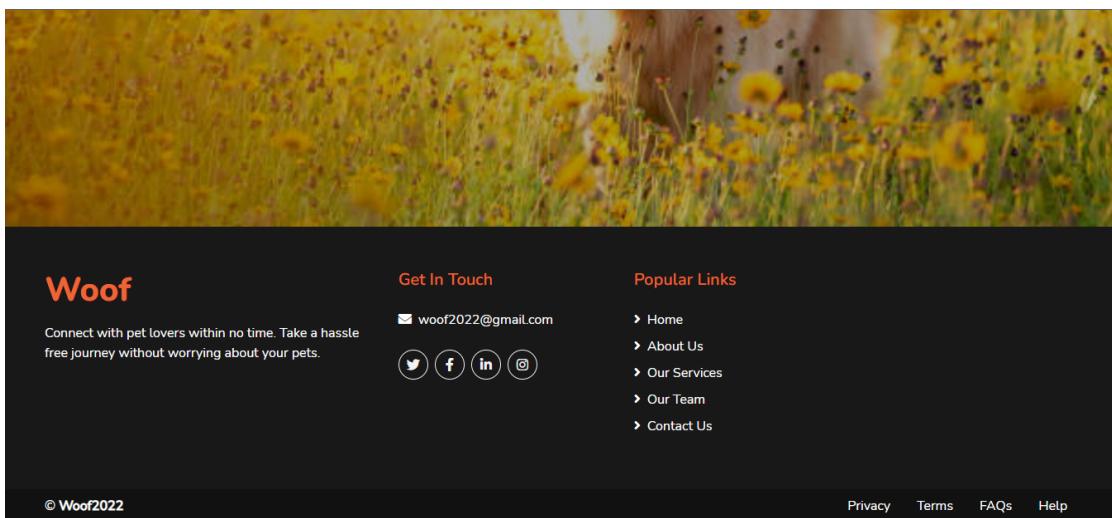
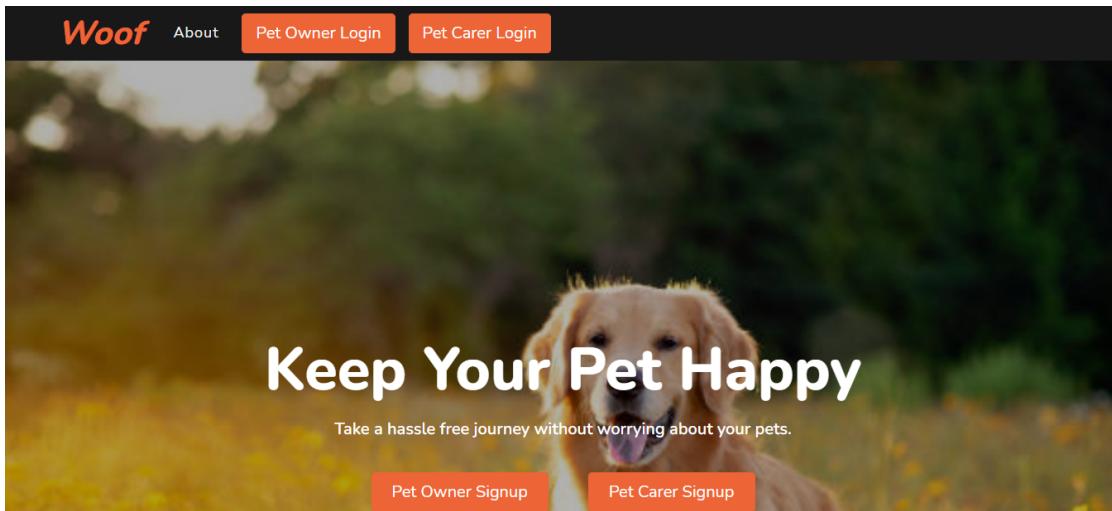
The Woof application successfully connects the pet owner and pet lover. Pet owners can book pet lovers by going to the services section in the application Top pet carers near them will be displayed to the pet owner. The pet owners and pet lovers can edit their profile at any time. Availability option present in the pet carer profile gives control to the pet carer as to when he would like to take bookings. The tracker system built with the gsm gps module successfully sends and receives messages to the owner of the pet. The pet owner has to just send a “Hi” and he/she will receive the reply message. The reply message contains the coordinates and a google map link of the tracker at that given time. This ensures safety of the pet and assurity to the pet owner that their pet is at the location he was dropped off to the pet lover.

7. Conclusion and Future Work

The Woof application is a complete system which can connect pet lovers and owners. The owner can track their pet through their phone by just sending a “Hi” message. Owners can enjoy their vacation while pet lovers can take care of these pets in their free time. The pet owners can review the pet carers making the system reliable.

In the further versions we wish to make the tracker smaller and provide a casing for the tracker to make it wearable and aesthetic. Connecting pet owners to pet shops and doctors is what we wish to integrate in the woof application.

Home Page before signing-



About Page-

The screenshot shows the 'About' section of the Woof platform. At the top, there is a navigation bar with the 'Woof' logo, a 'About' link, and two orange buttons for 'Pet Owner Login' and 'Pet Carer Login'. Below the navigation bar, the word 'About Us' is written in green. The main title 'Connecting Owners with Pet Lovers' is displayed in large, bold, orange and green text. A descriptive paragraph follows, explaining that Woof is an easy solution for pet owners to connect with nearby pet lovers for travel. It mentions a tracking device and video call feature for premium members. Below the text, there is a URL: <https://woof-pet-care.herokuapp.com/pets>, with a note that location can be tracked by owners remotely through the app. To the right of the text, there are three images: a dog sitting in a blue beanbag bed, a black and white dog standing next to another dog, and a group of cats sitting together.

Pet Owner SignUp-

The screenshot shows the 'Pet Owner SignUp' page of the Woof platform. The browser address bar shows 'localhost:3000/owner/signup'. The page has a dark header with the 'Woof' logo, a 'About' link, and two orange buttons for 'Pet Owner Login' and 'Pet Carer Login'. The main form is titled 'Sign Up' in large orange text. It contains several input fields: 'Name' (Rakesh), 'Email' (rakesh@gmail.com), 'Password' (*****), 'Age' (24), 'Phone Number' (7638253329), 'Address' (T nagar, Chennai), 'Pet Name' (Moti), 'Pet Type' (labrador), and 'Pet Description' (soft ball). To the right of the form, there is a section titled 'Going for a vacation?' with the heading 'Book For Your Pet' in orange. It includes links for 'Pet Boarding' and 'Pet Tracking' with their respective descriptions. The bottom of the screen shows a Windows taskbar with various icons and system status information.

Pet owner profile page:

The screenshot shows a web browser window for the 'Woof' application at localhost:3000/owner/profile. The interface includes a navigation bar with links for Woof, Services, Bookings, About, Profile, and Logout. The main content area is divided into two sections: 'Pet Owner Details' and 'Pet Details'.
Pet Owner Details:
Name : Rakesh
Email : rakesh@gmail.com
Phone Number : 7638253329
Age : 24
Address : T nagar, Chennai
Pet Details:
Name : Moti
Breed : labrador
Likes : soft ball
Dislikes : no food

The screenshot shows a web browser window for the 'Woof' application at localhost:3000/owner/profiledit. The interface includes a navigation bar with links for Woof, Services, Bookings, About, Profile, and Logout. The main content area displays a profile image of a dog and a form for updating the profile.
Profile Image:
A circular profile picture of a dog.
Profile Edit Form:
Choose File No file chosen
Name: Rakesh
Phone Number: 7638253329
Email: rakesh@gmail.com
Address: T nagar, Chennai

Pet carers near me display:

The screenshot shows a web browser window for the 'Woof' application. The title bar says 'Woof' and the URL is 'localhost:3000/owner/displaycarer'. The page has a dark header with 'Woof' logo, 'Services', 'Bookings', 'About', 'Profile', and 'Logout' buttons. The main content area is titled 'Available Pet Carers'. It displays two profiles: 'Raghu' (grey placeholder image) and 'Ashish' (image of a Shih Tzu dog). Each profile includes the name, email, phone number, cost per day, and a 'Book' button.

Name	Email	Phone Number	Cost
Raghu	raghu@gmail.com	7786593423	₹800 per day
Ashish	ashish@gmail.com	9678361182	₹1000 per day

Booking pet carer:

The screenshot shows a web browser window for the 'Woof' application. The title bar says 'Woof' and the URL is 'localhost:3000/owner/6268e4c2005eea4d31399769'. The page has a dark header with 'Woof' logo, 'Services', 'Bookings', 'About', 'Profile', and 'Logout' buttons. The main content area is titled 'Book Pet Carer'. On the left, there's a summary of the selected carer: Name: Ashish, Email: ashish@gmail.com, Phone Number: 9678361182, Age: 32, Address: Marina Mall, OMR, Chennai, Experience: 6 years, Cost: ₹1000 per day, and Total Cost: ₹1000. On the right, there are two date input fields (one for check-in and one for check-out), a bold 'Total Cost: 1000' label, and a large 'Confirm Booking' button.

Pet owner booking history:

The screenshot shows a web browser window for the 'Woof' application at localhost:3000/owner/history. The page title is 'Booking History'. On the left, there is a summary box containing the following information:

- Phone Number : 9678361182
- Address : Marina Mall, OMR, Chennai
- Experience : 6 Years
- Total Cost : ₹3000
- Address : Marina Mall, OMR, Chennai

A small image of a Shih Tzu dog is displayed next to the name 'Ashish' and his email 'ashish@gmail.com'. Below the summary box is a red 'Review' button. The main content area shows a list of bookings, which is currently empty. At the bottom of the page, there is a footer with the 'Woof' logo, social media links (Twitter, Facebook, LinkedIn, Instagram), and a search bar.

Pet carer SignIn:

The screenshot shows a web browser window for the 'Woof' application at localhost:3000/carer/login. The page title is 'Pet Carer Login'. It features a large orange login form with fields for email ('ashish@gmail.com') and password ('*****'). A black 'Login' button is at the bottom of the form. To the right of the form, there is a promotional message:

Have experience in taking care of Pets? Or you love pets but do not have one?

List yourself as a Pet Carer

Pet owners will choose you to take care of their pets

Below the login form is a footer with the 'Woof' logo, social media links, and a search bar. The footer also includes system status icons for temperature (37°C), battery level (15:51), and date/time (29-04-2022).

Pet carer booking history:

The screenshot shows a web browser window for the 'Woof' application at localhost:3000/carer/bookingdetails. The page title is 'Booking Details'. It displays two separate booking entries.

Booking 1:

- Name: Tejas Vaichole
- Email: tejasv515@gmail.com
- Phone Number: 8600230940
- Address: Shubhashree Residential Phase-2, D-501, Akurdi, Pune
- Start Date: 2022-04-27T00:00:00.000Z
- End Date: 2022-04-28T00:00:00.000Z

Booking 2:

- Name: Sanika Kulkarni
- Email: sanika.k.goa@gmail.com
- Phone Number: 9921950055
- Address: Model's Millennium Vista, 4E-S1
- Start Date: 2022-04-28T00:00:00.000Z
- End Date: 2022-04-29T00:00:00.000Z

The browser interface includes a search bar, taskbar icons, and system status indicators like temperature (37°C) and date (29-04-2022).

Pet carer profile page:

The screenshot shows a web browser window for the 'Woof' application at localhost:3000/carer/profile. The page title is 'Pet Carer Details'.

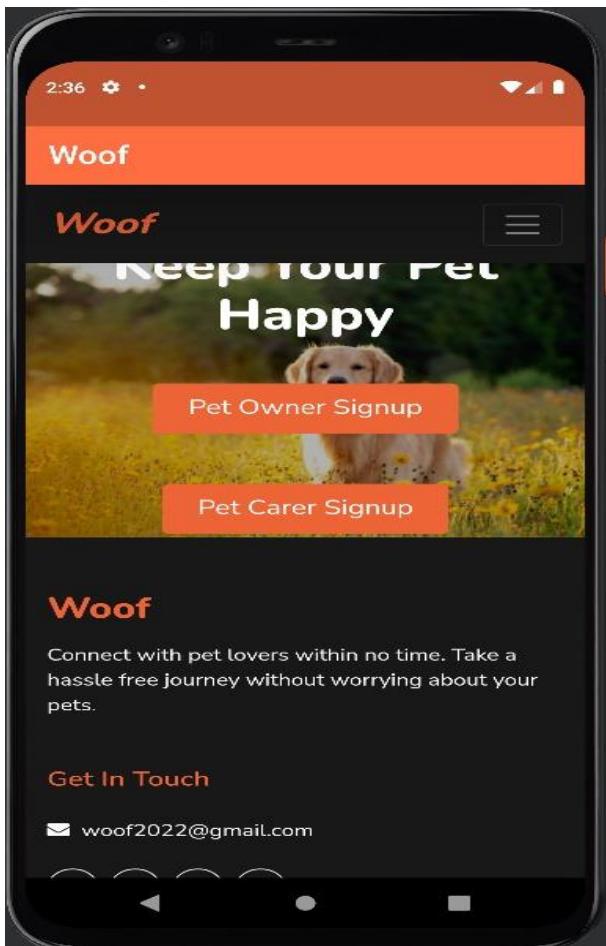
Pet Carer Details:

- Name: Ashish
- Email: ashish@gmail.com
- Phone Number: 9678361182
- Age: 32
- Address: Marina Mall, OMR, Chennai
- Experience: 6
- Availability: Available

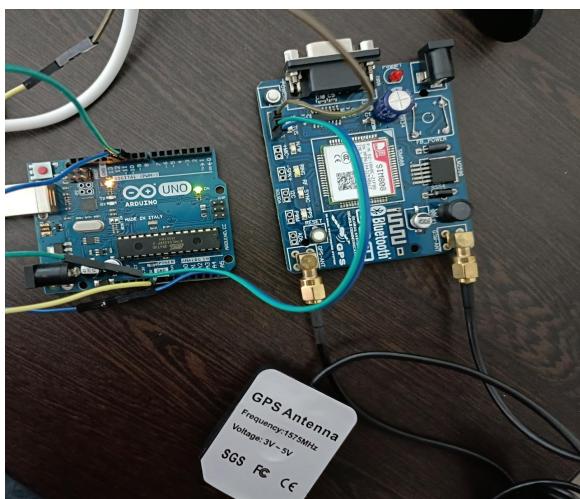
A small image of a white dog is displayed next to the profile information. An 'Edit' button is located below the profile details.

The browser interface includes a search bar, taskbar icons, and system status indicators like temperature (37°C) and date (29-04-2022).

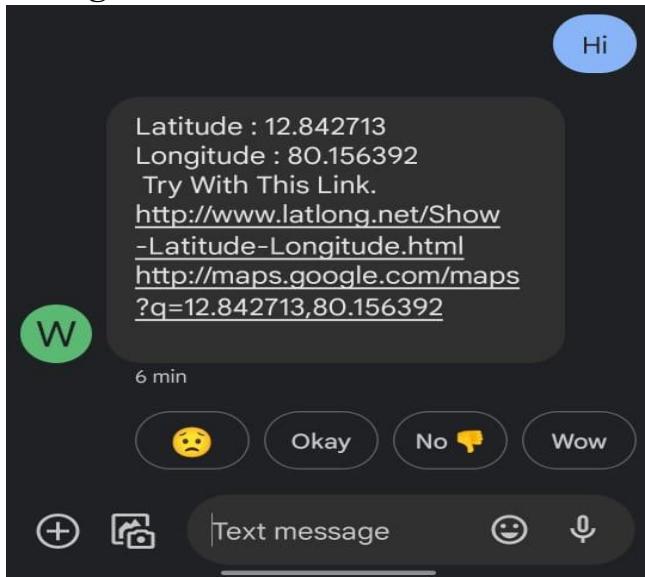
Mobile App



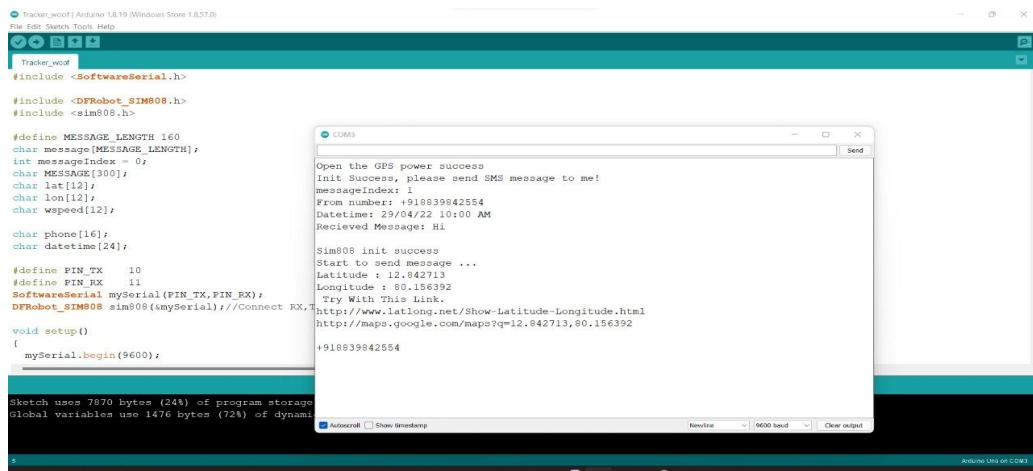
Tracker Setup-



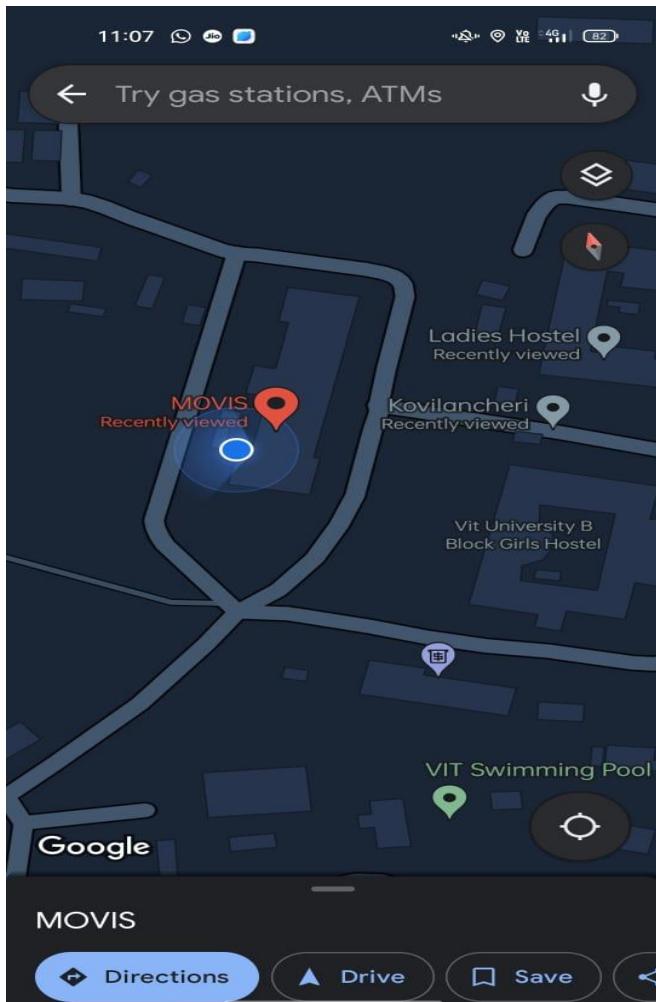
Getting the coordinates of pet by pinging the tracker with 'Hi' message-



Collecting the Coordinates on the IDE-



Viewing the exact location on map-



8. REFERENCES

X. Ge, R. Gu, Y. Lang and Y. Ding, "Design of handheld positioning tracker based on GPS/GSM," 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC), 2017, pp. 868-871, doi: 10.1109/ITOEC.2017.8122477.

I. Ganchev, Z. Ji and M. O'Droma, "Designing a Low-Cost Location Tracker for Use in IoT Applications," 2020 XXXIIrd General Assembly and Scientific Symposium of

the International Union of Radio Science, 2020, pp. 1-2, doi:
10.23919/URSIGASS49373.2020.9232023.

J. Zhou, H. Yang and K. Anderson, "SNPP ATMS On-Orbit Geolocation Error Evaluation and Correction Algorithm," in IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 6, pp. 3802-3812, June 2019, doi: 10.1109/TGRS.2018.2887407.

L. Jiang, M. Cheng and T. Matsumoto, "A TOA-DOA Hybrid Factor Graph-Based Technique for Multi-Target Geolocation and Tracking," in IEEE Access, vol. 9, pp. 14203-14215, 2021, doi: 10.1109/ACCESS.2021.3052233.

M. Cheng, M. R. K. Aziz and T. Matsumoto, "Integrated Factor Graph Algorithm for DOA-Based Geolocation and Tracking," in IEEE Access, vol. 8, pp. 49989-49998, 2020, doi: 10.1109/ACCESS.2020.2979510.

APPENDIX

CODE for GPS Tracker-

```
#include <SoftwareSerial.h>
#include <DFRobot_SIM808.h>
#include <sim808.h>

#define MESSAGE_LENGTH 160
char message[MESSAGE_LENGTH];
int messageIndex = 0;
char MESSAGE[300];
char lat[12], lon[12], wspeed[12];
char phone[16];
char datetime[24];

#define PIN_TX 10
#define PIN_RX 11
SoftwareSerial mySerial(PIN_TX,PIN_RX);
DFRobot_SIM808 sim808(&mySerial); //Connect RX,TX,PWR,
```

```

void setup()
{
    mySerial.begin(9600);
    Serial.begin(9600);

    //***** Initialize sim808 module *****
    while(!sim808.init())
    {
        Serial.print("Sim808 init error\r\n");
        delay(1000);
    }
    delay(3000);
    if( sim808.attachGPS())
        Serial.println("Open the GPS power success");
    else
        Serial.println("Open the GPS power failure");
    Serial.println("Init Success, please send SMS message to me!");
}

void loop()
{
    //***** Detecting unread SMS *****
    messageIndex = sim808.isSMSunread();
    //***** At least, there is one UNREAD SMS *****
    if(messageIndex > 0)
    {
        Serial.print("messageIndex: ");
        Serial.println(messageIndex);
        sim808.readSMS(messageIndex, message, MESSAGE_LENGTH, phone, datetime);

        //*****In order not to full SIM Memory, is better to delete it*****
        sim808.deleteSMS(messageIndex);
        Serial.print("From number: ");
        Serial.println(phone);
        Serial.print("Datetime: ");
        Serial.println(datetime);
        Serial.print("\nRecieved Message: ");
    }
}

```

```

Serial.println(message);

float la = sim808.GPSdata.lat;
float lo = sim808.GPSdata.lon;
dtostrf(la, 6, 2, lat); //put float value of la into char array of lat. 6 = number of digits before decimal sign. 2 = number of digits after the decimal sign.
dtostrf(lo, 6, 2, lon); //put float value of lo into char array of lon
sprintf(MESSAGE, "Latitude : %s\nLongitude : %s\nTry With This
Link.\nhttp://wwwlatlong.net>Show-Latitude-Latitude.html\nhttp://maps.google.com/maps?q=%s,%s\
n", lat, lon, lat, lon);

Serial.println("Sim808 init success");
Serial.println("Start to send message ... ");
Serial.println(MESSAGE);
Serial.println(phone);
sim808.sendSMS(phone,MESSAGE);
//***** Turn off the GPS power *****
//sim808.detachGPS();

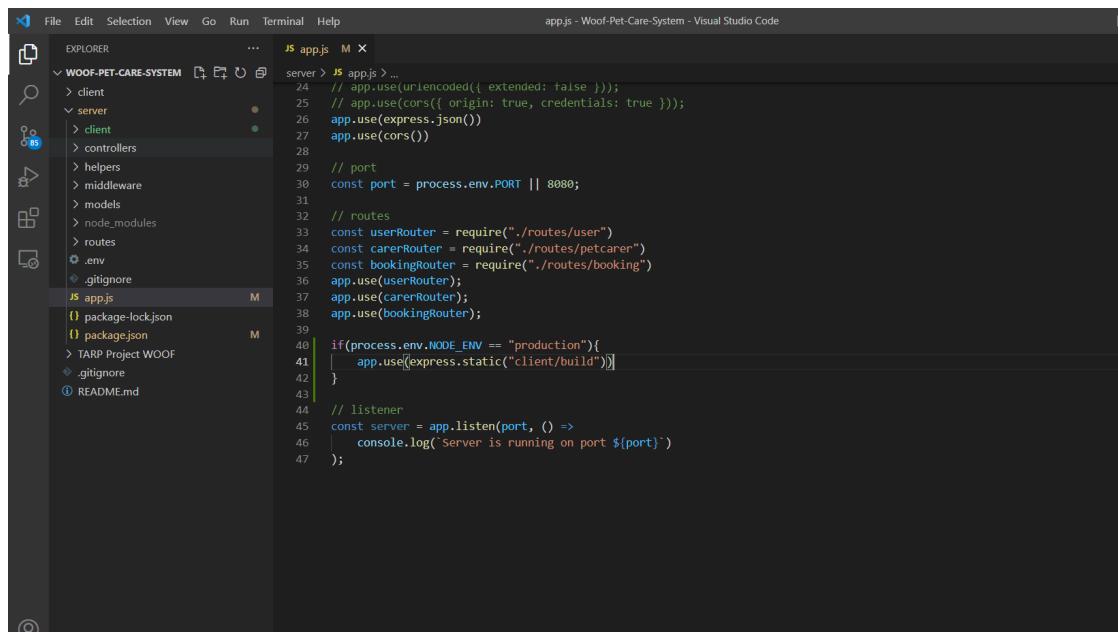
}

}

```

Software code:

Express



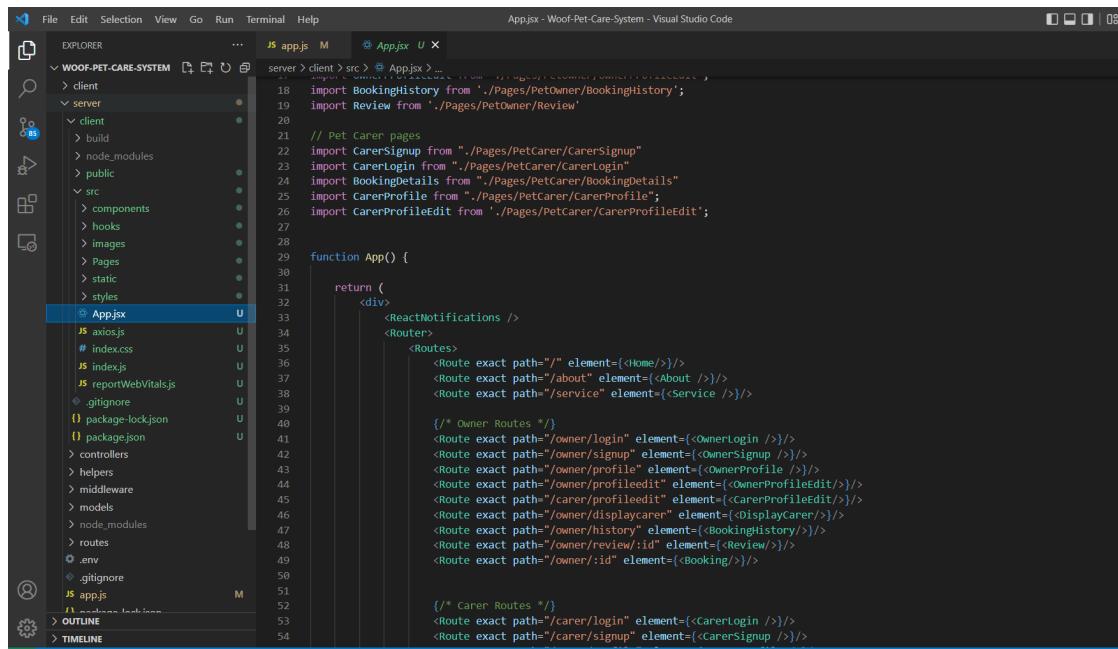
```

File Edit Selection View Go Run Terminal Help
WOOF-PET-CARE-SYSTEM app.js - Woof-Pet-Care-System - Visual Studio Code
EXPLORER
  > client
  > server
    > client
    > controllers
      > helpers
      > middleware
      > models
      > routes
      .env
      .gitignore
    > app.js M
  package-lock.json
  package.json M
  TARP Project WOOF
  .gitignore
  README.md

app.js
server > app.js > ...
24 // app.use(urlencoded({ extended: false }));
25 // app.use(cors({ origin: true, credentials: true }));
26 app.use(express.json())
27 app.use(cors())
28
29 // port
30 const port = process.env.PORT || 8080;
31
32 // routes
33 const userRouter = require("./routes/user")
34 const carerRouter = require("./routes/petcare")
35 const bookingRouter = require("./routes/booking")
36 app.use(userRouter);
37 app.use(carerRouter);
38 app.use(bookingRouter);
39
40 if(process.env.NODE_ENV == "production"){
41   [ app.use(express.static("client/build")) ]
42 }
43
44 // listener
45 const server = app.listen(port, () =>
46   [ console.log(`server is running on port ${port}`) ]
47 );

```

React:



The screenshot shows the Visual Studio Code interface with the file `App.jsx` open in the center editor pane. The code is a functional component named `App()` that returns a `<div>` element containing `<ReactNotifications />` and `<Router>`. The `<Router>` component contains several `<Route>` components for different pages like Home, About, Service, and various owner and carer login/signup routes. The left sidebar shows the project structure under the `WOOF-PET-CARE-SYSTEM` folder, including client and server branches, and the `src` directory which contains components, hooks, images, Pages, static, and styles.

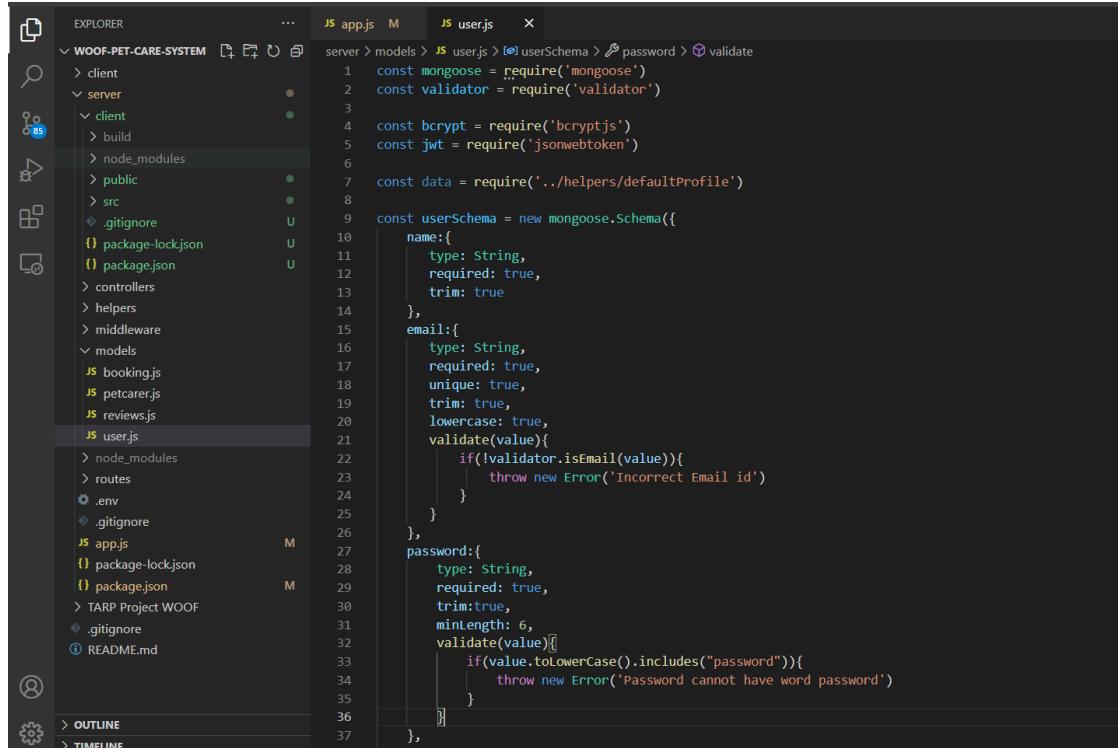
```
18 import BookingHistory from './Pages/PetOwner/BookingHistory';
19 import Review from './Pages/PetOwner/Review'
20
21 // Pet Carer pages
22 import CarerSignup from './Pages/PetCarer/CarerSignup'
23 import CarerLogin from './Pages/PetCarer/CarerLogin'
24 import BookingDetails from './Pages/PetCarer/BookingDetails'
25 import CarerProfile from './Pages/PetCarer/CarerProfile';
26 import CarerProfileEdit from './Pages/PetCarer/CarerProfileEdit';
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
```

```
function App() {
  return (
    <div>
      <ReactNotifications />
      <Router>
        <Route exact path="/" element={<Home />}>/</Route>
        <Route exact path="/about" element={<About />}>/</Route>
        <Route exact path="/service" element={<Service />}>/</Route>

        /* Owner Routes */
        <Route exact path="/owner/login" element={<OwnerLogin />}>/</Route>
        <Route exact path="/owner/signup" element={<OwnerSignup />}>/</Route>
        <Route exact path="/owner/profile" element={<OwnerProfile />}>/</Route>
        <Route exact path="/owner/profileedit" element={<OwnerProfileEdit />}>/</Route>
        <Route exact path="/carer/profileedit" element={<CarerProfileEdit />}>/</Route>
        <Route exact path="/owner/displaycarer" element={<DisplayCarer />}>/</Route>
        <Route exact path="/owner/history" element={<BookingHistory />}>/</Route>
        <Route exact path="/owner/review/:id" element={<Review />}>/</Route>
        <Route exact path="/owner/:id" element={<Booking />}>/</Route>

        /* Carer Routes */
        <Route exact path="/carer/login" element={<CarerLogin />}>/</Route>
        <Route exact path="/carer/signup" element={<CarerSignup />}>/</Route>
    </Router>
  );
}
```

Mongodb:



The screenshot shows the Visual Studio Code interface with the file `user.js` open in the center editor pane. It defines a Mongoose schema named `userSchema` with fields for name, email, and password. The `name` field is a required string that is trimmed. The `email` field is a required unique string in lowercase that must be a valid email ID. The `password` field is a required string of minimum length 6 that cannot contain the word "password". The left sidebar shows the project structure under the `WOOF-PET-CARE-SYSTEM` folder, including client and server branches, and the `src` directory which contains components, hooks, images, Pages, static, and styles.

```
1 const mongoose = require('mongoose')
2 const validator = require('validator')
3
4 const bcrypt = require('bcryptjs')
5 const jwt = require('jsonwebtoken')
6
7 const data = require('../helpers/defaultProfile')
8
9 const userSchema = new mongoose.Schema({
10   name:{
11     type: String,
12     required: true,
13     trim: true
14   },
15   email:{
16     type: String,
17     required: true,
18     unique: true,
19     trim: true,
20     lowercase: true,
21     validate(value){
22       if(!validator.isEmail(value)){
23         throw new Error('Incorrect Email id')
24       }
25     }
26   },
27   password:{
28     type: String,
29     required: true,
30     trim:true,
31     minLength: 6,
32     validate(value){
33       if(value.toLowerCase().includes("password")){
34         throw new Error('Password cannot have word password')
35       }
36     }
37   },
38 }
```

