WOOF- A DOG CARE SYSTEM

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

B.TECH. (COMPUTER ENGINEERING)

TO

RK UNIVERSITY, RAJKOT

SUBMITTED BY

Name of Student	**Enrollment No.**
Vishal Das	17SOECE11007

UNDER THE GUIDANCE OF

Internal Guide	**External Guide**
Dr. Paresh Tanna	Akash Kuamr Jadav
Associate Professor	Technical Lead
RK University	Aimdek Tech. Pvt. Ltd.
Rajkot, Gujarat	Ahmedabad - Gujarat

April 2021

SCHOOL OF ENGINEERING, RK UNIVERSITY, RAJKOT

DECLARATION

I hereby certify that I am the sole author of this project work and that neither any part of this project work nor the whole of the project work has been submitted for a degree to any other University or Institution. I certify that, to the best of my/our knowledge, my project work does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my project document, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. I declare that this is a true copy of my project work, including any final revisions, as approved by my project review committee.

Signature of Student

Vishal

VISHAL DAS

(17SOCE11007)

Date: 18/04/2021
Place: Ahmedabad, Gujarat

INDUSTRY CERTIFICATE

RK.UNIVERSITY

CERTIFICATE

This is to certify that the work which is being presented in the Project Report entitled **"Woof - A Dog Care System",** in partial fulfillment of the requirement for the award of the degree of **B.Tech. (Computer Engineering)** and submitted to the School of Engineering, RK University, is an authentic record of my own work carried out during a period from **February 2021 to April 2021.**

The matter presented in this Project Report has not been submitted by me for the award of any other degree elsewhere.

Signature of Student

Vishal

Vishal Das

(17SOCE11007)

This is to certify that the above statement made by the student(s) is correct to the best of my knowledge.

Internal Guide	**External Guide**	**Head of Department**
Dr. Paresh Tanna	Akash Kumar Jadav	Prof. Ashwin Raiyani
Associate Professor,TPO	Team Lead,	CE / IT / BCA / MCA
RK University ,	Aimdek Tech. Pvt. Ltd.,	School of Engineering,
Rajkot	Ahmedabad, Gujarat	RK University, Rajkot

April 2021

SCHOOL OF ENGINEERING
RK.UNIVERSITY

SCHOOL OF ENGINEERING, RK UNIVERSITY, RAJKOT

RK.UNIVERSITY

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many persons and I am extremely privileged to have this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I respect and thank Vishal Doshi Sir, for providing me an opportunity to do the project work in Ahmedabad and giving me all the support and guidance, which made it possible for me to complete the project duly. I am extremely thankful to him for providing such nice support and guidance.

I owe my deep gratitude and appreciation to my internal project guide Dr. Paresh Tanna Sir, who not only guided me throughout my project but also showed keen interest in my project work till the completion of my project. He also shared all the necessary information and his wisdom to help me develop an efficient system.

At last but not the least I would like to show my gratefulness towards my external guide Mr. Akash Kumar Jadav of Aimdek Technologies, for his timely support and guidance throughout the project work.

RK.UNIVERSITY

ABSTRACT

Woof - The Dog Care System endeavour to help every canine owner to take care of their canines and provide them all the necessary information and services online. It primary focus is to get the owner proper healthcare solutions for his/her pet. The modern solutions for healthcare, commerce and services has made it easier for us humans to have all that we require to take care of ourselves and make our live a bit more comfortable. No doubt newer technologies in the future will enhance it even further. The availability of internet connection almost everywhere has now made it possible for us to have all that we require literally in our finger tips. This system follows the same trends and makes it possible for the dog owners to get all the help they need in one click.The system also aims to connect different service and dog care providers to the customers and make them more easily and widely approachable.

TABLE OF CONTENT

LIST OF FIGURES

Figure No.	**Figure Name**	**Page No.**

LIST OF TABLES

COMPANY PROFILE

Founded in 2014, AIMDek Technologies is a team of professionals with engineering excellence commonly believing in the disruptive power of the technology. We are unified by our commitment to quality and innovation, where we ideate and meticulously craft value-based IT solutions for several businesses across the globe to power digital transformation across all major platforms and tech stacks with improved efficiency, productivity, and agility.
Our extensive knowledge expertise comprises but is not confined to service-oriented architecture, open source portal consulting, business intelligence, Enterprise Portals, Enterprise Content Management, PLM and ERP solutions to a varied industry including healthcare, sports, and fitness, education, manufacturing, insurance, and eCommerce.

Vision - "To deliver revolutionary solutions that empower business and inspires excellence."

Mission - "Partner with organizations to establish world-class IT infrastructure and solutions that improve conception, collaboration, and communication.
Our endeavor to explore and innovate ensures to strategically position our customers as industry leaders."

As an employee, I am expected to carry out my duties that I have been entrusted to me as an Liferay developer and improve my skills to be an full-stack developer. Communicate with the clients and complete project requirements in the given time period. Also as an employee I am also expected to keep growing my knowledge in the IT field and learn the new technologies.

Woof - A Dog Care System

CHAPTER 1
INTRODUCTION

1.1 **Project Summary**

Woof - A sound that fills every dog lover's ear with excitement and joy. Our dogs loves us unconditionally and even we love them to the moon and back. When it comes to taking care our dear beloved pet it could be a bit confusing and sometimes we need some extra help. The idea behind this particular venture is that every dog lover get the necessary information, assistance and best products for taking care of their best friend. This system aims to provide the user with medical assistance for dogs through informative blogs, articles, online healthcare advisors, provide a e-commerce portal for canine products, a portal for pet care and training and dog walkers where the user can ask for the respective services.

1.2 **Purpose**

They say, "Dogs do speak, but only to those know how to listen", and we all love our pets and want to take care them. Dog also called man's best friend is a being who loves unconditionally and remains loyal to the end. But sometimes due to our lack of knowledge, less accessibility and availability of required assistance and services we fail to provide the proper care to our canines. These issues can be from as simple as getting a dog collar to a full health check-up, the non-availability of these service can lead to some slow but serious health problems, and sometimes even cause early death. Sometimes we need extra help when we are required go out of station due to immediate reasons and its not always the case that our neighbour will be there for us, at these situations we can ask pet-care to take our canines into there care and we can focus on the work at hand, later we can get back our dear friend.

Dog care system is an initiative to help users better take of there beloved dogs and help them to maintain their health, training and also connect them to online services, so that our loyal canine can live a happy, healthy and longer life with us.

1.3 **Scope**

Dog care system is convenience tool that can help both user and the dog, the user can manage his/her dog needs and the dog will also get the required professional help and care. The system will also provide a platform where user can browse through various dog related products and choose the best product for them. It will also allow users to book services like dog care and training sessions, grooming, dog nanny and also ask for dog walkers if need be. It will also provide with all the necessary information that a dog lover need to have before adopting a dog. The system will also give health related advice and in needed case user can ask for professional help. These system will be fully online based and most of the services will be third party and some restricted to geographical regions.

1.4 **Technologies**

I have used MEAN stack technologies for development, as shown below

Front End	**Back End**	**Tools**
● Angular, ● Bootstrap	● Nodejs, ● ExpressJs ● Mongodb	● Visual Code ● Postman, ● Mongodb Compass, ● Canva ● Creatly

1.5 **Literature Review**

The current dog care systems and services available in the net are not fully able to solve the issues and also provides some individual solutions. The system that I endeavour to develop will provide the cumulative services so that the user can get all the needed services at on click and don’t have to surf through net for hours. Also plans are to improve the system to help provide vaccination system tracking and monitoring.

CHAPTER 2
PROJECT MANAGEMENT

2.1 **Project Planning**

2.1.1 Project Development Approach

After understanding the requirements and needs of the project, I have planned to approach the project in a modular and flexible manner. In this way, the different parts of the project can be managed easily and effectively. Project is developed in Nodejs and Angular. Following the software engineering standard as specified for Software Engineering. I am using Iterative Waterfall Model for the development of the system. This process model is explained in brief below.

- **Justification**

In the Software Development Life Cycle, there are different stages for requirement collection, feasibility study, requirement determination, design,

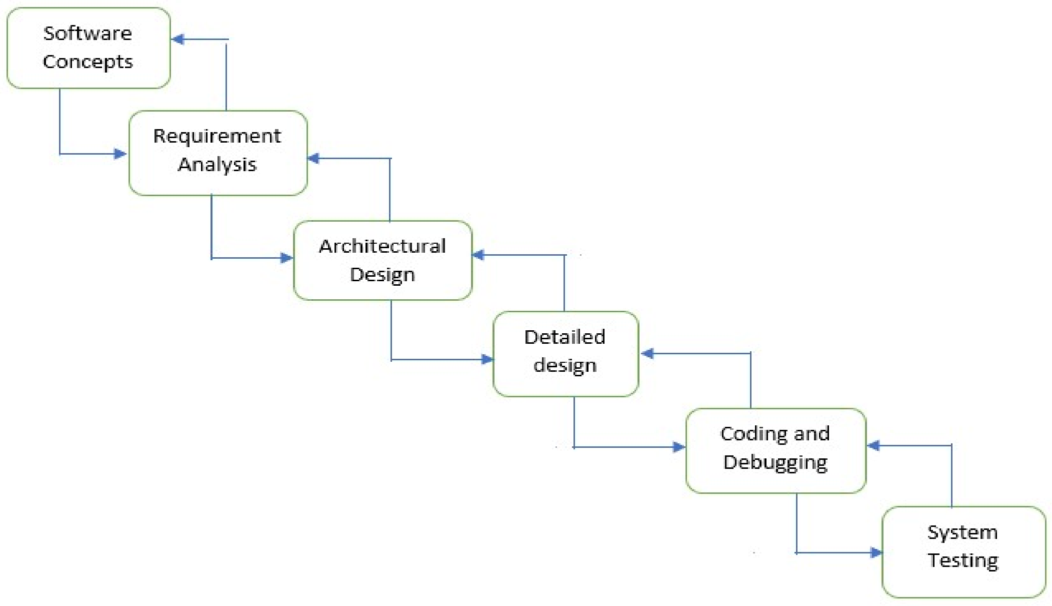


Fig. - Waterfall Model

coding and implementation and then testing and debugging so we can first identify requirements and we can do the feasibility study. Thus it is beneficial to first identify the requirements and then through feasibility study we can analyze these requirement and determine them for implementation. Then after gathering all necessary requirements we can easily design them and then the implementation becomes very easy and faster. The client requirements were quite fluctuating and that enforces us to choose a model that allows us to move back to any previous phase of the development life cycle, make changes over there, & again get it implemented in the next phase. This repeats until the satisfactory level is reached.

2.1.2 Project Plan

The road to the successful project development is the well planned strategy for the best and optimal use of resource available.

The step wise plan for the project is as follows:

1) Understand the system firstly.

2) Once familiar with the system, start to work on it.

3) Make tables in the database that holds values and their data types that are to be inserted.

4) First of all, design a page using various controls from the toolbox.

5) Create a relationship among all these tales, wherever needed.

6) Generate classes that cover the coding part and helps with the insertion operations.

7) Link the tables and classes with designed web page using objects.

8) Approval by testing

The back-end of the project is developed in NodeJS and for testing API Postman is used as an API client. ExpressJs is used as middleware which provides simple and efficient HTTP utility. Other dependencies used are Nodemon, Cors, Mongoose and Json formatter.

For front-end Angular v11 is used, with its features like MVC architecture, code generation, code splitting, etc makes it easier and faster to develop and also manage sites. Also angular CLI helps creating the components faster and testing and then deploy. Dependencies used along with angular are Bootstrap 4 and charts.js.

2.1.2 Schedule Representation

Scheduling the project task is an important project planning activity. It involves deciding which tasks would be taken up when. Based on the planned duration of required tests and collection of resources to complete those tasks projected completion date is calculated.

TASK NAME	START	END	DURATION
Requirement Gathering	20/01/21	24/01/21	5 days
Analysis of requirements	27/01/21	30/01/21	4 days
System Design and Coding	02/02/21	27/02/21	26 days
System design and testing	01/03/21	24/03/21	24 days
System Integration and Testing	26/03/21	03/04/21	9 days
Documentation	05/04/21	15/04/21	11 days

Table. - Schedule Plan

2.2 **Risk Management**

Risk identification and analysis of the project is done on the basis of the testing and reviews of similar products and accordingly best possible plans and strategies are taken into consideration for handling them.

2.2.1 Risk Identification

Some of the risk that were identified while developing the project includes

- User authentication
- User data security
- Secure Payment
- Request response

Since the project is still in development phase more risks are being identified and will be resolved accordingly.

2.2.2 Risk Analysis

The above risk shows that the user's validation and security are to be handled carefully and strongly so that future risk like fake request, data manipulation, spams, etc can be avoided. Also for the payments in the accessories and service section needs to be secure and fast, so that users can pay for the services with ease. The service providers are also needed to give proper confirmation and response to users for each requests and the fake requests and spams should be minimized.

2.2.3 Risk Planning

To achieve minimized risks in the final product some methods and planning are required to be satisfied, which includes

- User verification using number or email.
- Deploying strict password policies.
- Using secure and tested third party payment applications.
- Approval of each service request and delivery.

CHAPTER 3
SYSTEM REQUIREMENT STUDY

3.1 User Characteristics

This system will serve as a bridge among the dog owners and service providers. The types of users dealing with the system are,

- Dog owners
- Vets
- Dog walkers
- Pet care centers
- Admins

3.2 Hardware and Software requirements

The software requirements for the project is just basic browsers and internet connection from user perspective but for development purpose basic requirements include,

-nodejs with all the required dependencies,

-mongoDb or other noSQL,

-IDE for development purpose,

-angular CLI.

No specific hardware required, will work in all machines with at least above mentioned software requirements.

CHAPTER 4
SYSTEM ANALYSIS

4.1 Current Systems

The systems which are already available now related to pet management or pet services lack to provide the all the different functionalities at the same platform, they at best provides few services or most often individual services to the user. Some of the famous sites around the globe and in India are

Site	*Services Offered*	*Link*
Rover	provides pet related information, grooming, sitting, walking and shopping services.	https://www.rover.com/
BarkBox	Monthly goodies for dogs.	https://www.barkbox.com/
Muttorpolis	Provides pet accessories and custom products for pets.	https://muttropolis.com/
MSPCA	Pet adoption and care services.	https://www.mspca.org/
Only4Pets	Pet adoption service (India).	https://only4pets.com/
Marshallpetzone	Online Dog shop (India).	https://www.marshallspetzone.com/
PawsIndia	Dog accessories and care products (India).	https://pawsindia.com/
Pupkart	Custom dog collars and leashes (India)	https://www.pupkart.com/
Dogspot	Dog adoption (India).	https://www.dogspot.in/

Petsfolio	Dog walking, grooming, training (India).	https://www.petsfolio.com/

Table - Sites For Current Systems

4.2 **Problem with the current system**

Systems which are already available are divided and are providing individual services or at the very best few related services. Users or pet owners have to refer to different sites for different needs and this can be irritating and frustrating to them. Also, the some of the sites which provides multiple services are still improving there functionalities and trying to provide all the necessary services for better user experience and customer satisfaction.

4.3 **Requirements of the new system**

The new system aims to satisfy all the basic needs and functionalities that can be helpful for the users or pet owners. It will provide a common ground for both the pet owners, vets, walkers, accessories shops and dog care centers to interact together. This way it will provide the users what they need in the same place and help service providers to easily grow their business.

4.3.1 Functional Requirements

The functional requirements in system includes

I) User

- Login/Registration
- Dog profile management
- Online accessories
- Request walker/sitter
- Vet appointment booking
- Dog grooming and training

II) Vets

- Login/Registration
- Manage Appointments
- View dog details
- Add dog diagnostics

III) Walkers/Sitters

- Login/Registration
- Accept/Reject requests
- View Dog details

IV) Pet care centers

- Showcase services
- Accept appointments

4.3.2 Non - Functional Requirements

The non-functional requirements for the system includes

- Security
- Authentication
- Historical Data
- Adjustments and cancellations
- Availability
- Manageability

4.4 **Functions of the System**

Use Case diagram for the above system

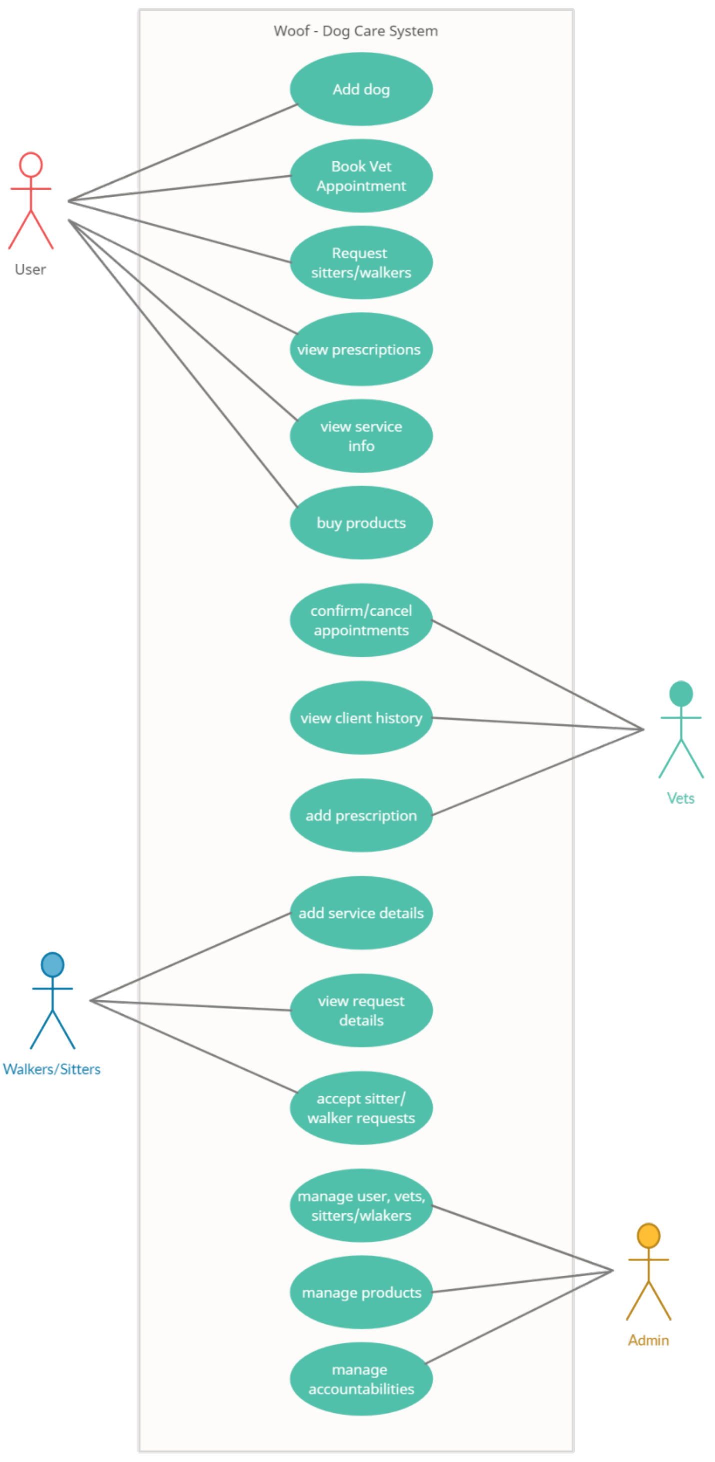


Fig. - Use Case Diagram

4.5 **Data Modeling**

4.5.1 Data Dictionary

All the system data is stored and managed in NoSql database MongoDb using compass GUI tool for both and testing of data persistence.

Field Names	**Null/ Not Null**	**Type**	**Key**
_id	Not null	Object	Primary
Name	Not null	String	
Email	Not null	String	
Password	Not null	String	
Number	Not null	Number	
Address	Not null	String	

Table. - User Data Dictionary

Field Names	**Null/ Not Null**	**Type**	**Key**
_id	Not null	Object	Primary
User_id	Not null	Object	Foreign
Dog_Name	Not null	String	
Age	Not null	Number	
Gender	Not null	String	

Table. - Dog Data Dictionary

Field Names	**Null/ Not Null**	**Type**	**Key**
_id	Not null	Object	Primary
Product_Name	Not null	String	
Price	Not null	Float	
Inventory	Not null	Long_int	
Description	Not null	String	

Quantity	Not null	Number	

Table. - Product Data Dictionary

Field Names	**Null/ Not Null**	**Type**	**Key**
_id	Not null	Object	Primary
Vet_Name	Not null	String	
Email	Not null	String	
Password	Not null	String	
Number	Not null	Number	
Address	Not null	String	
Charges	Not null	Float	
Schedules	Not null	Array	

Table. - Vet Data Dictionary

Field Names	**Null/ Not Null**	**Type**	**Key**
_id	Not null	Object	Primary
Walker_Name	Not null	String	
Email	Not null	String	
Password	Not null	String	
Number	Not null	Number	
Charges	Not null	Float	
Gender	Not null	String	

Table. - Walker Data Dictionary

Field Names	**Null/ Not Null**	**Type**	**Key**
_id	Not null	Object	Primary
User_id	Not null	Object	Foreign
Products	Not null	Object	

Date	Not null	Date	
Status	Not null	String	
Charges	Not null	Float	

Table. - Sales Data Dictionary

Field Names	**Null/ Not Null**	**Type**	**Key**
_id	Not null	Object	Primary
Dog_id	Not null	Object	Foreign
Vet_id	Not null	Object	Foreign
Date	Not Null	Date	
Status	Null	String	
Charges	Null	Number	

Table. - Request walker/siiter Data Dictionary

Field Names	**Null/ Not Null**	**Type**	**Key**
_id	Not null	Object	Primary
Dog_id	Not null	Object	Foreign
Date	Not Null	Date	
Status	Null	String	
Charges	Not null	Float	

Table. - Booking Data Dictionary

Field Names	**Null/ Not Null**	**Type**	**Key**
_id	Not null	Object	Primary
User_id	Not null	Object	Foreign
Vet_id	Not null	Object	Foreign
Date	Null	Date	
Status	Null	String	

Diagnostics	Null	String	
Medicine	Null	String	
Comments	Null	String	

Table. - Prescription Data Dictionary

4.5.2 ER Diagram

ER -diagram provides basic understanding of the system and how it relates the data in the database.

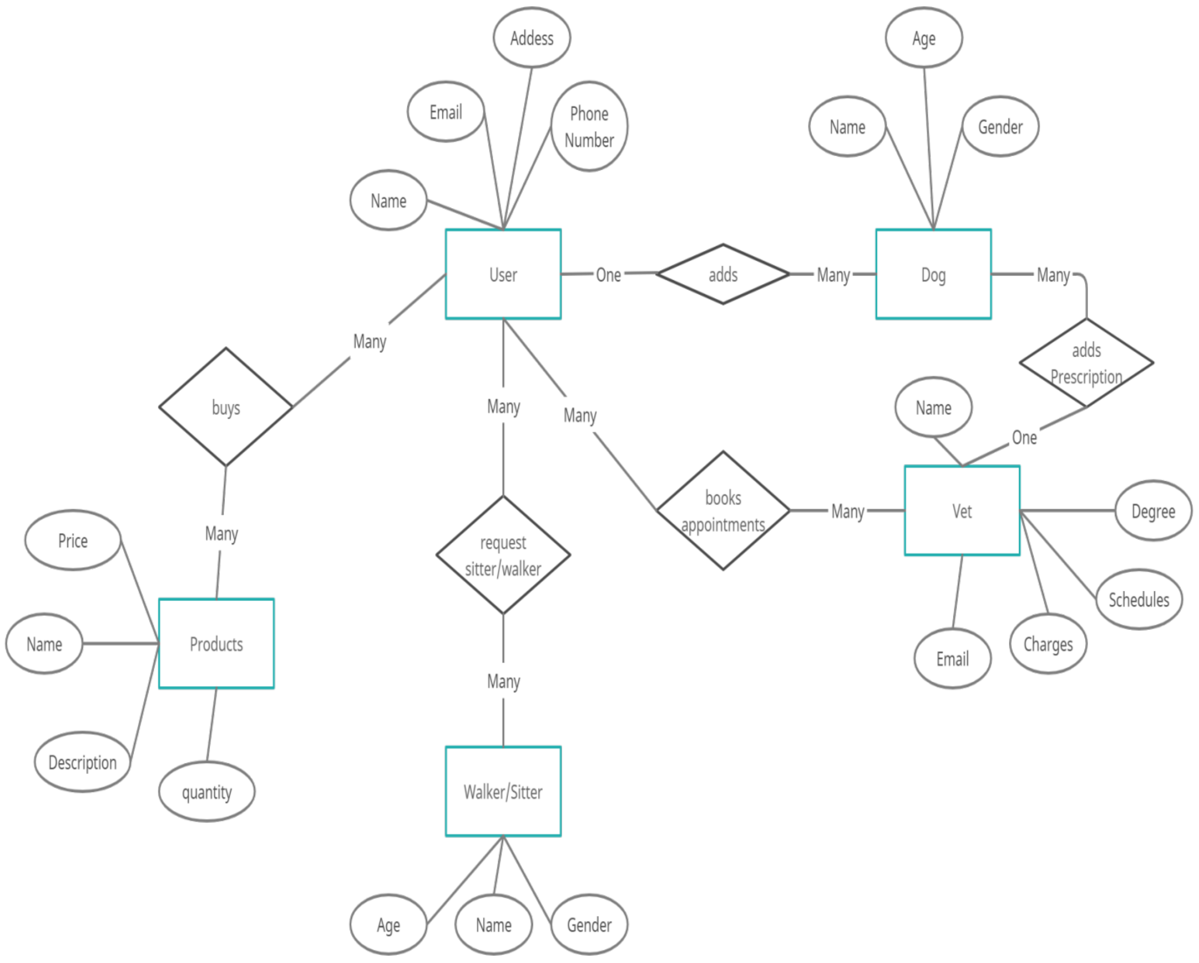


Fig. - ER Diagram

4.5.3 Class Diagram

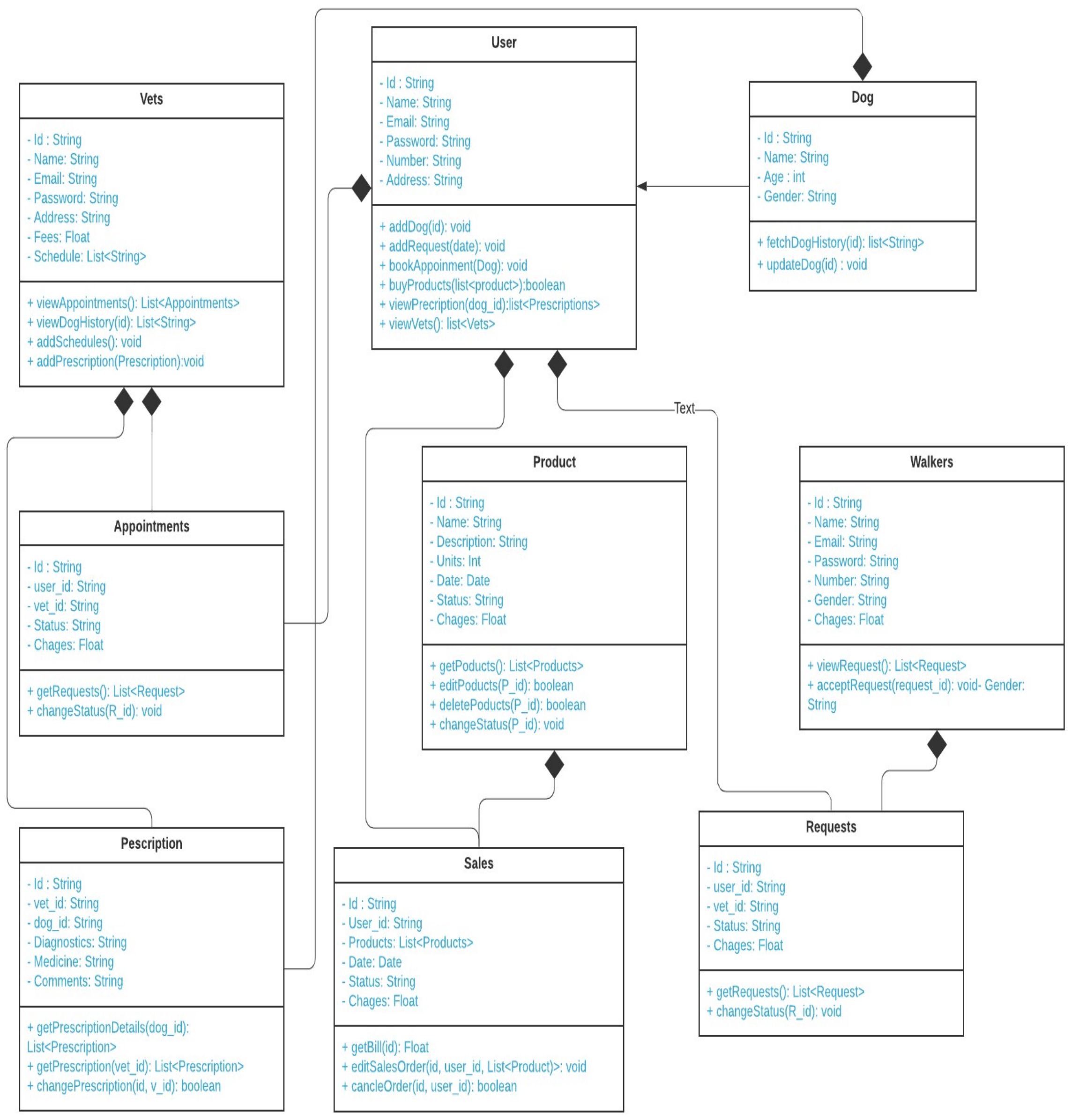


Fig. - Class Diagram

4.6 **Functional and Behavioral Modeling**

4.6.1 Data Flow Diagram

The data flow diagram shows how the flow of the data takes place from one part to other in the system. Below are the Level 0 DFD and,

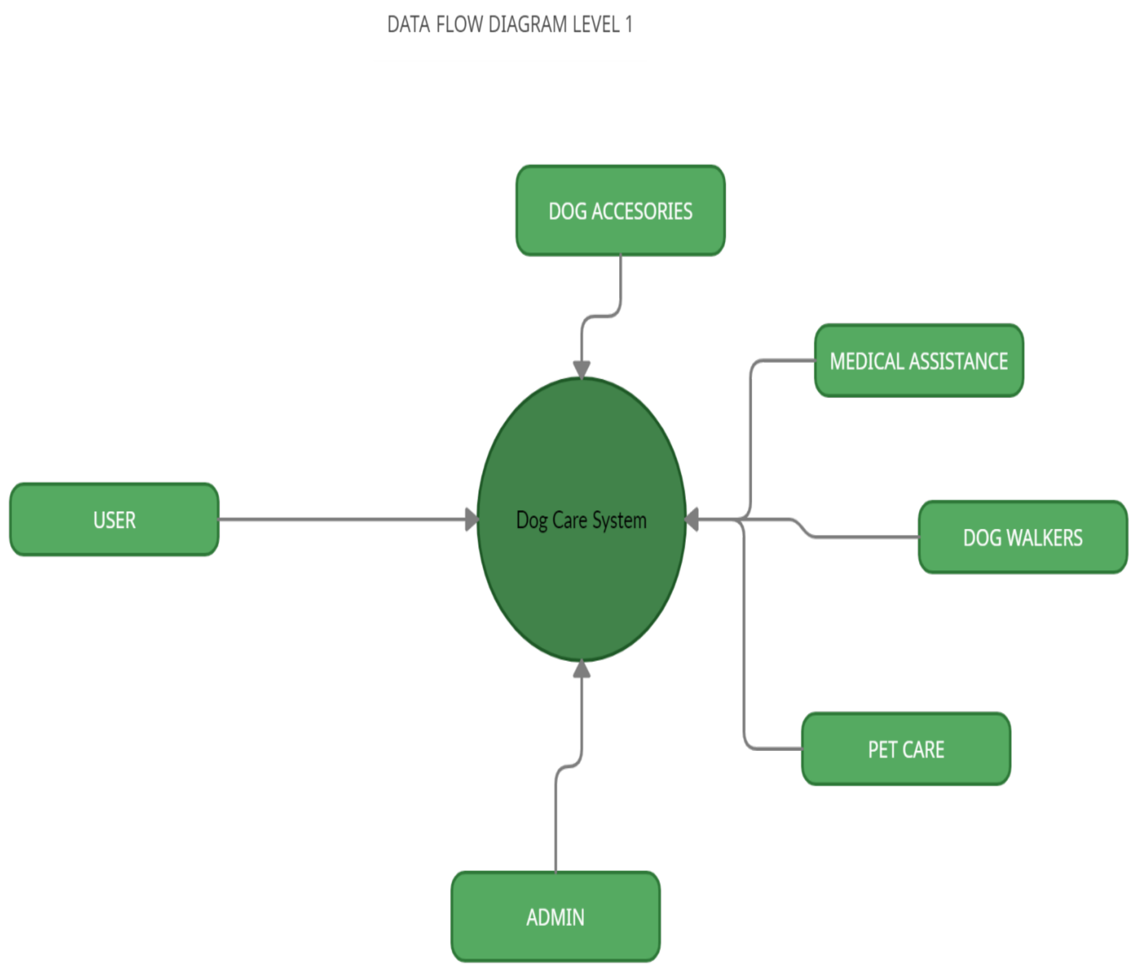


Fig. - Level 0 Data Flow Diagram

Level 1 Data flow diagrams for the dog care system.

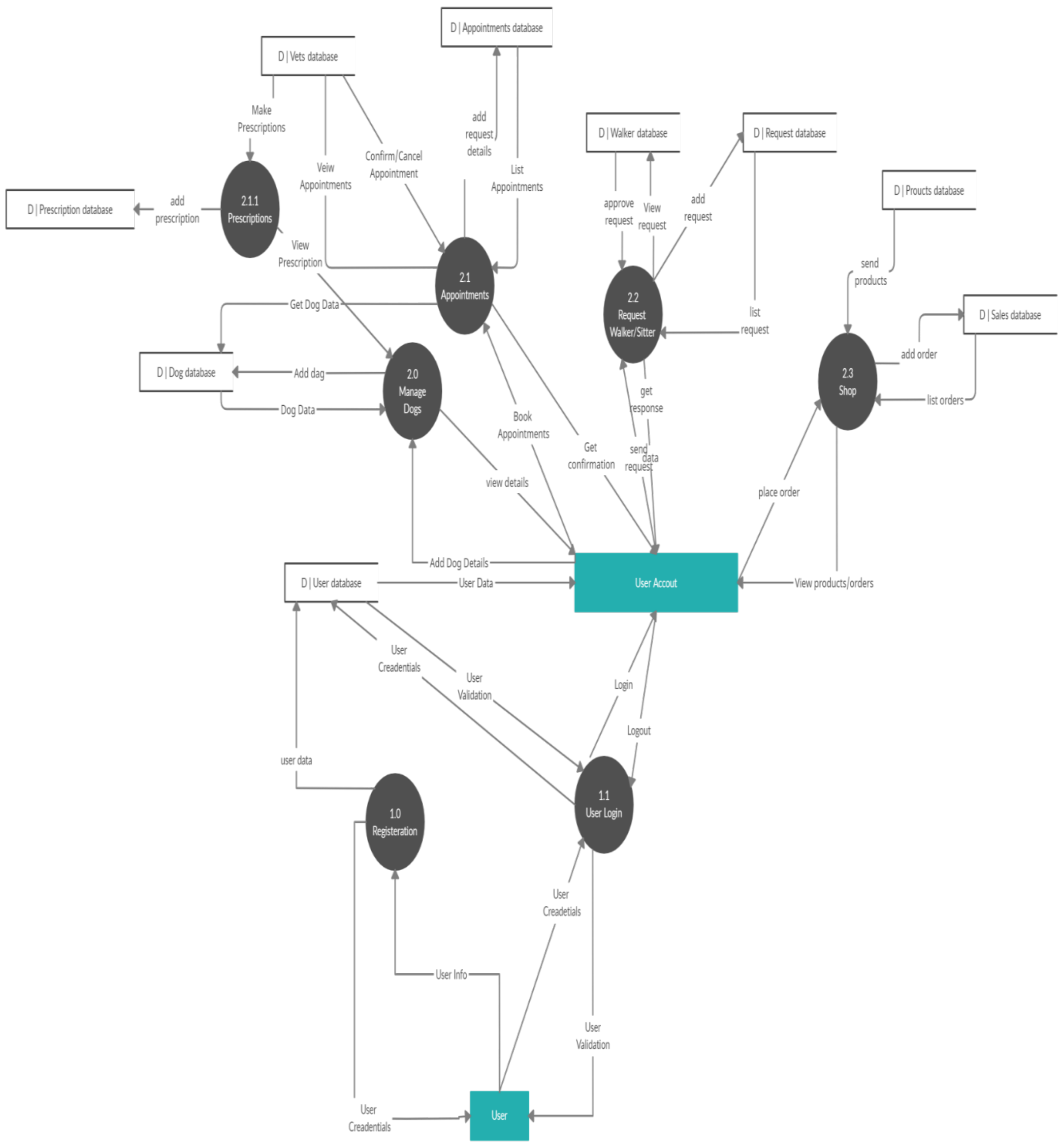


Fig. - Level 1 Data Flow Diagram

CHAPTER 5
SYSTEM DESIGN

5.1 **Database Design and Data Structure Design**

5.1.1 Tables and Relationship

Table 1	**Table 2**	**Relation Type**	**Foreign Keys**
User	Dog	One to Many	User_Id
User	Sales	Many to Many	User_Id
User	Request	Many to Many	User_Id
Dog	Booking	Many to One	Dog_Id
Product	Sales	Many to Many	Product_Id

Table - Tables and Relationships

5.1.2 Logical Description Of Data

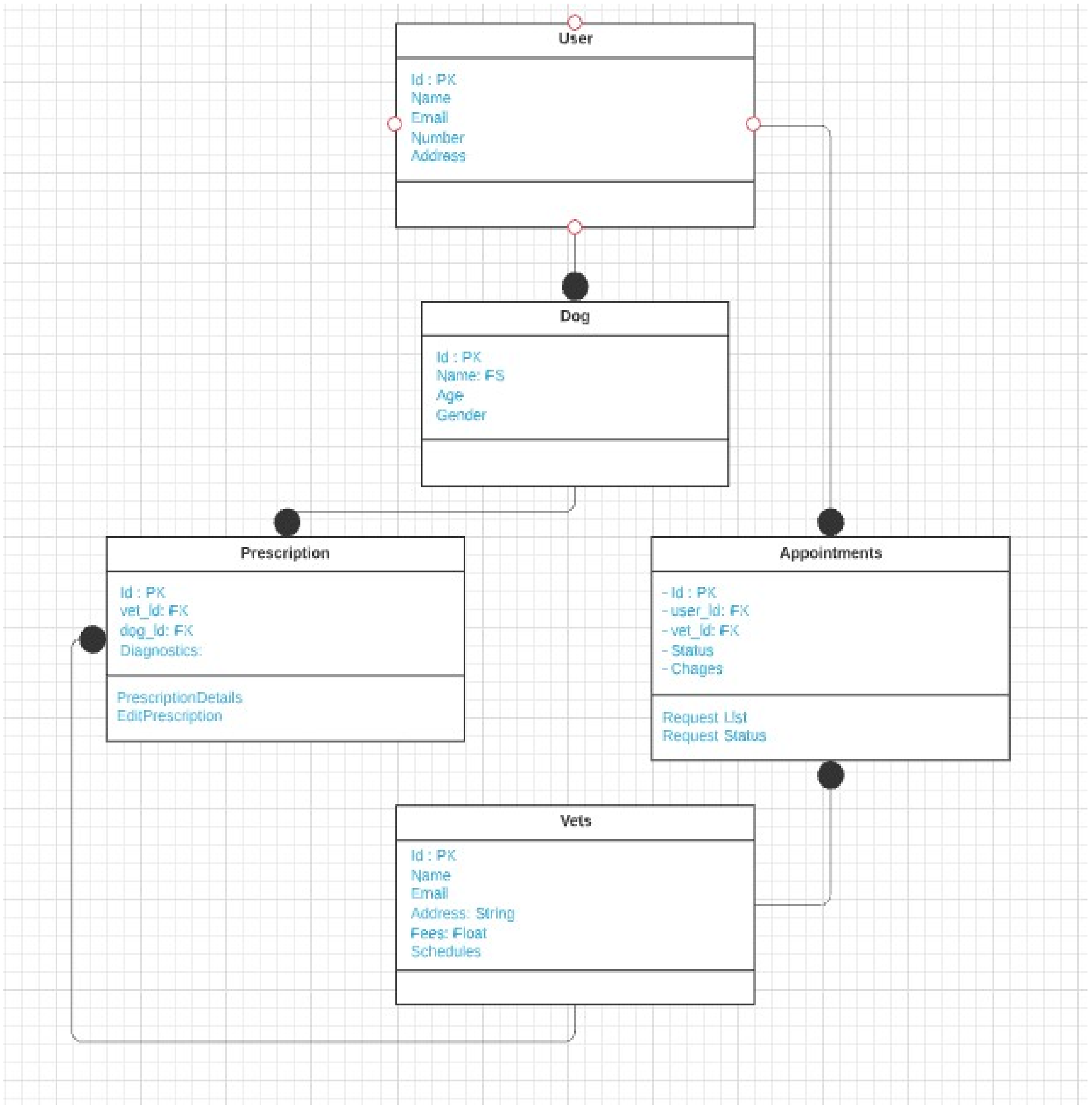


Fig. - Logical Flow of data 1

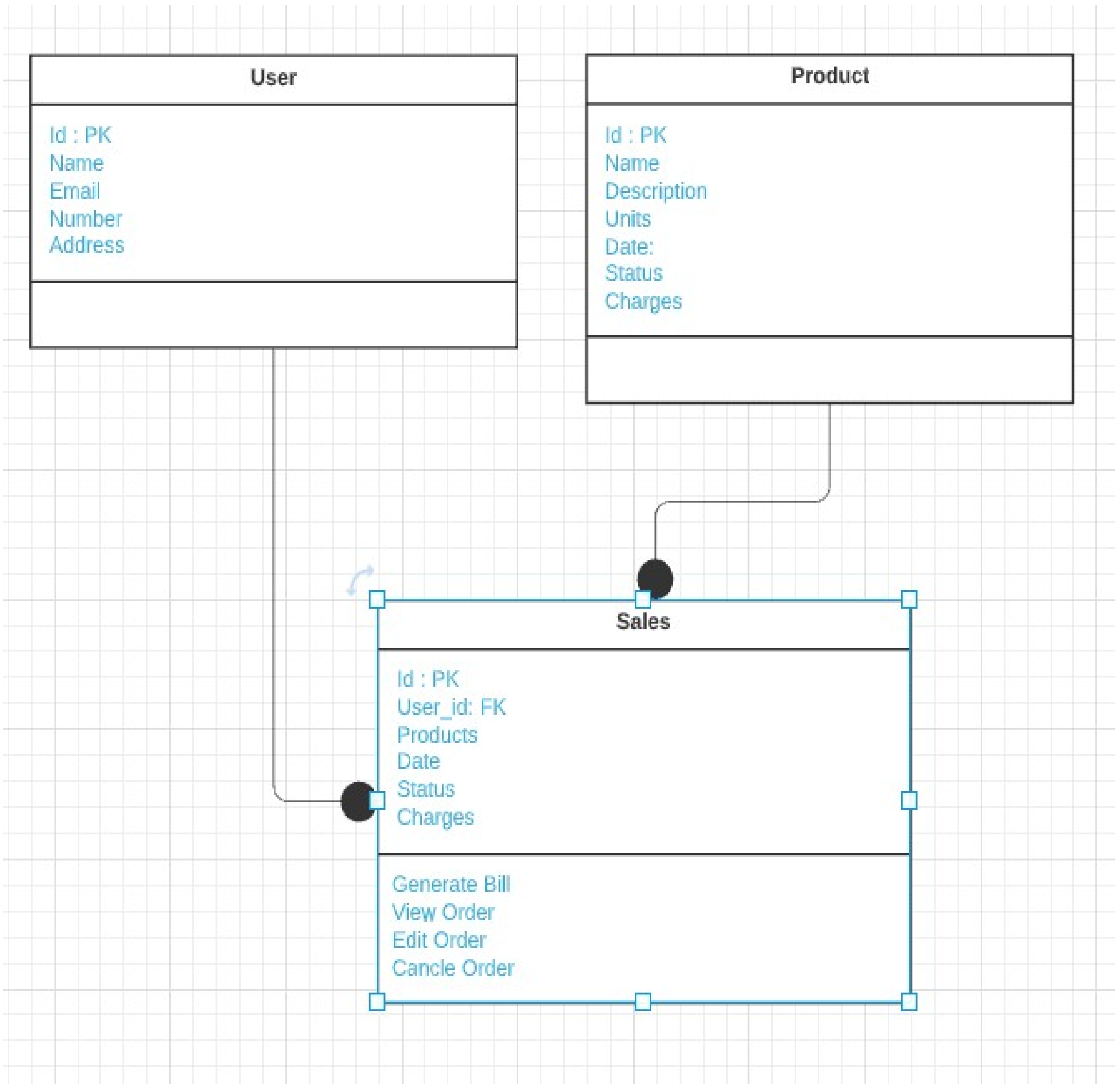


Fig. - Logical Flow of data 1

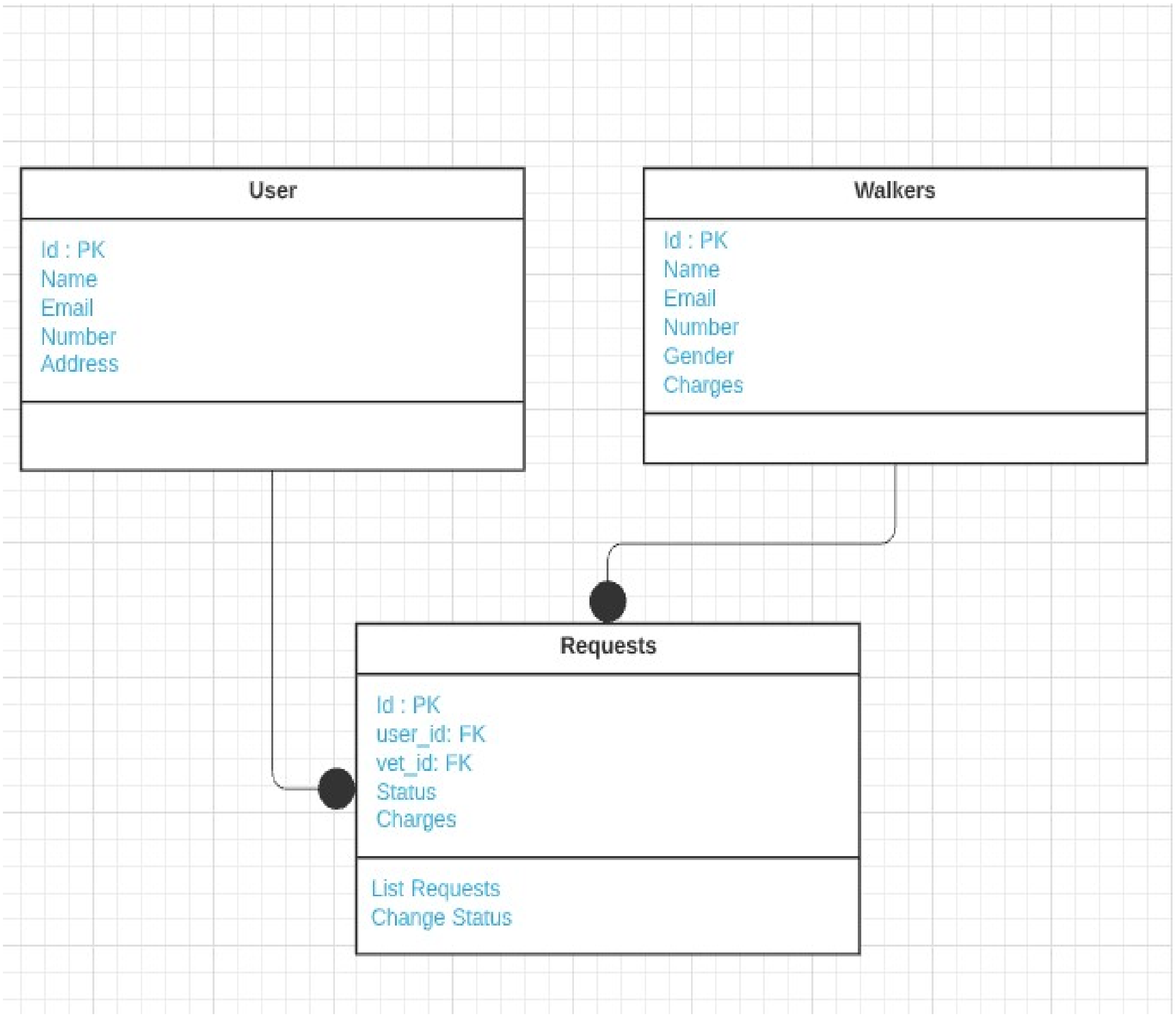


Fig. - Logical Flow of data 1

5.2 System Procedural Design(Flow chart)

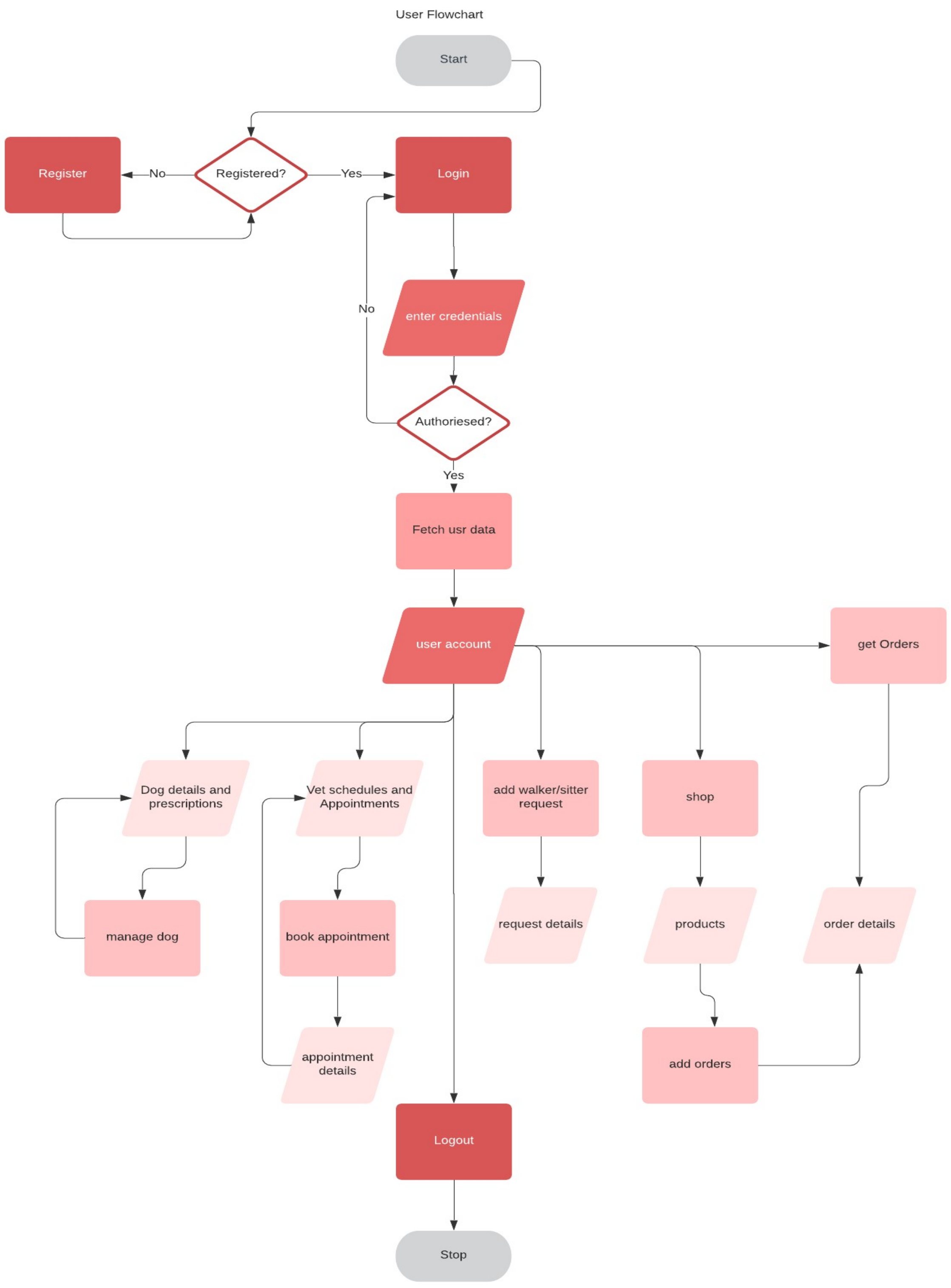


Fig.- User Flowchart

5.3 Input/Output and Interface Design

Here are some Sample of forms and UI used in the system.

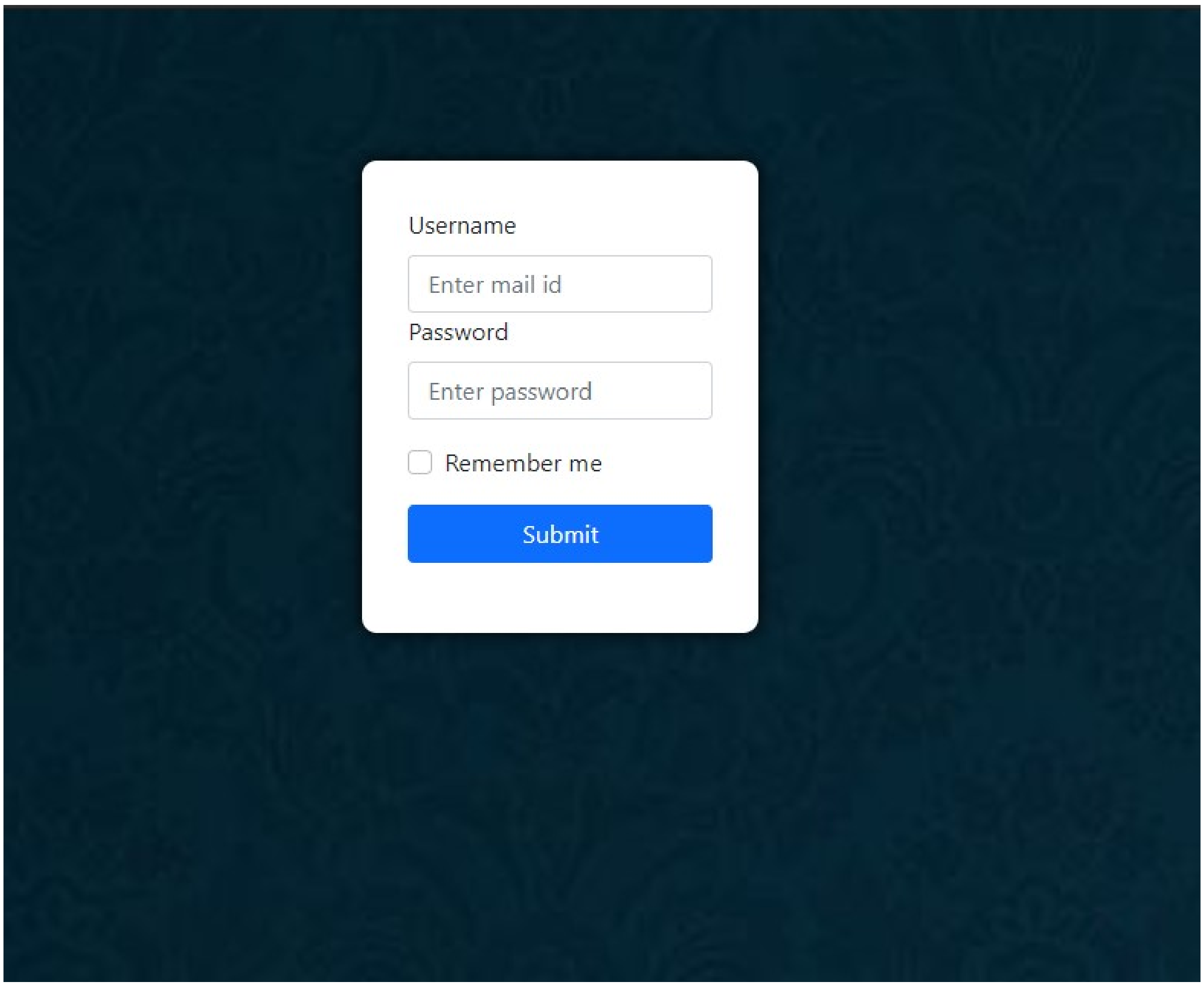
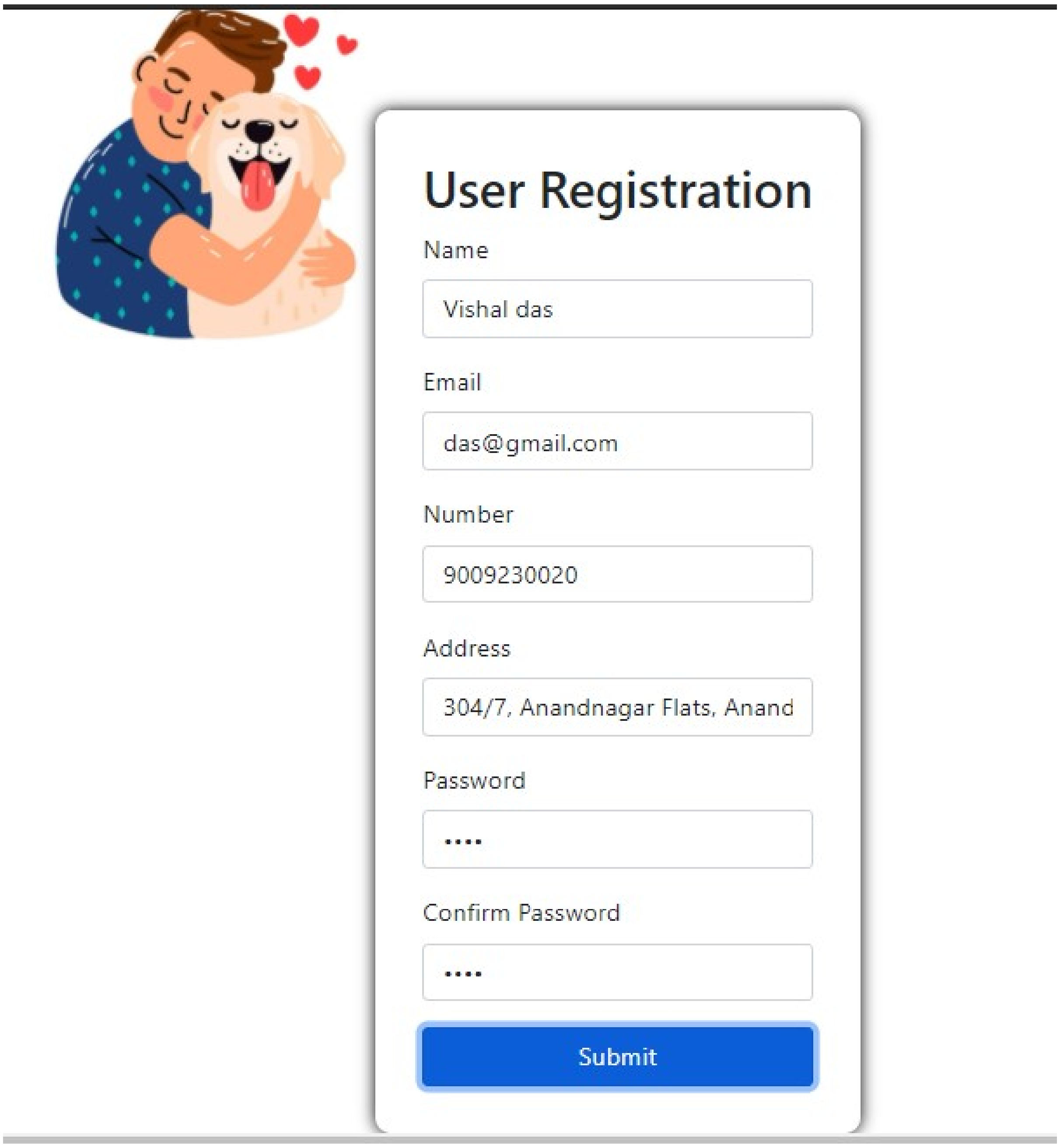


Fig.- Login Form



User Registration

Name	Vishal das
Email	das@gmail.com
Number	9009230020
Address	304/7, Anandnagar Flats, Anand
Password	••••
Confirm Password	••••

Fig.- User Registration Form

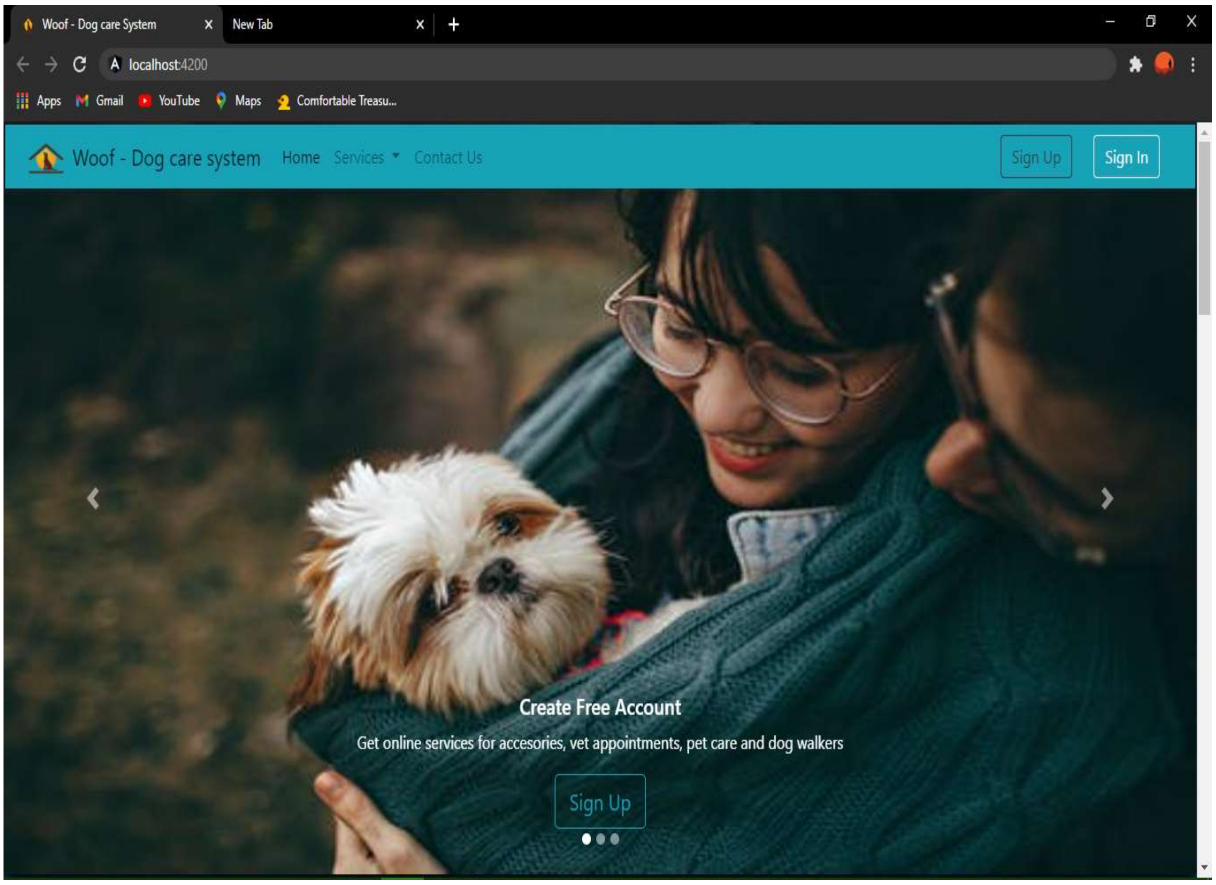


Fig.- Home Page

CHAPTER 6
IMPLEMENTATION AND PLANNING

6.1 **Environment Of Implementation**

The main goal of this system is to help users take better care of their dogs. It will provide the user 24/7 medical assistance for his/her dog(s) and also online consultancy experienced veterinarian. A portal from where user can get all the necessary products for dogs. Online connection to Dog care and training, grooming, dog nanny and dog walkers to help provide the necessary services. To implement such functionalities it needs reliable hosting server and efficient back-end services. To achieve this I am using NodeJS in back-end, Angular in front-end and currently using the Localhost for development but intend to use AWS services to test reliability and availability of the service.

6.2 **Program and Module Specification**

Dog Profile management.The major feature and functionalities of these project will basically provide platforms where the other third parties can sell there products and services, and also the system will provide key information related to canine health care and management systems to monitor and track their every need. It will includes some functionalities like

6.2.1 Dog management

Canine management system will help users to maintain profiles for their dogs which will include from their basic details to medical history and vaccinations. The system will also allow for other services and required help.

6.2.2 Accessories

This system will mainly focus on providing a portal from where the owners can buy products for their dogs online and get it delivered to them.

6.2.3 Get Medical Assistance

This one of the core functionality of the Dog care system, it will focus on providing the users with schedules of the vets near by or far, so that they book appointments and get the necessary medical assistance that their dog needs.

6.2.4 Request for walkers/sitters

This functionality aims to provide the user to ask for walkers and sitters for their loved pet when they can't. It is important that we take our pets, we can't leave them alone for days as they are dependent on us for their welfare. The walkers will accept the requests accordingly and dogs will get a playtime friend.

6.3 **Coding Standards**

Throughout the system development and even in future the practice is to follow the best coding standards and methods to improve the portability, debugging, readability and ease of understanding. The principles like DRY, encapsulation, open close design, etc along with the model, control and view in front-end and even the back-end is kept as modular as possible. Along naming conventions and comments are kept accordingly, so its easier to update code later.

6.3 **Sample Coding**

Here is sample code that is used to create the user API in the back-end using NodeJs.

index.js file

```
import express from 'express';

import bodyParser from 'body-parser';

import cors from 'cors';


import {_mong} from './db.js'

import router from './routes/usersRoutes.js';


const app = express();
const PORT = 8080;


const API_URL = `http://localhost:${PORT}/API/users`;
const API_URL_ECOM = `http://localhost:${PORT}/API/products`;
const API_URL_MDS = `http://localhost:${PORT}/API/doctors`;
const API_URL_WKS = `http://localhost:${PORT}/API/walkers`;



 //provide the CORS permission host to write request to this node server
let corsOption ={
   origin : 'http://localhost:4200',
   optionSuccessStatus : 200
}


app.use(cors(corsOption));
app.use(bodyParser.json());
app.get('/',(req,res)=>{


   const info = `<h3><i>Useful URLs for using Woof API .....</i></h3><br><hr>
            <b><h4>Users restAPI</h4></b>
            <b>GET request </b>: <a href="${API_URL}/list"> ${API_URL}/list </a> <br>
            <b>POST request </b>: <i> ${API_URL}/ </i><br>
            <b>PUT request </b>: <i>${API_URL}/{id} </i> <br>
            <b>DELETE request </b>: <i>${API_URL}/{id}</i><br>
            <b><h4>E-commerce restAPI</h4></b>
            <b>GET request </b>: <a href="${API_URL_ECOM}/list"> ${API_URL_ECOM}/list </a> <br>
            <b>POST request </b>: <i> ${API_URL_ECOM}/ </i><br>
```

```
                <b>PUT request </b>: <i>${API_URL_ECOM}/{id} </i> <br>
                <b>DELETE request </b>: <i>${API_URL_ECOM}/{id}</i><br>
                <b><h4>Medical-service restAPI</h4></b>
                <b>GET request </b>: <a href="${API_URL_MDS}/list"> ${API_URL_MDS}/list </a> <br>
                <b>POST request </b>: <i> ${API_URL_MDS}/ </i><br>
                <b>PUT request </b>: <i>${API_URL_MDS}/{id} </i> <br>
                <b>DELETE request </b>: <i>${API_URL_MDS}/{id}</i><br>
                <b><h4>Walkers restAPI</h4></b>
                <b>GET request </b>: <a href="${API_URL_WKS}/list"> ${API_URL_WKS}/list </a> <br>
                <b>POST request </b>: <i> ${API_URL_WKS}/ </i><br>
                <b>PUT request </b>: <i>${API_URL_WKS}/{id} </i> <br>
                <b>DELETE request </b>: <i>${API_URL_WKS}/{id}</i><br>
                `;
   res.send(info)

})


app.listen(PORT, () => console.log(`Server running on port: http://localhost:${PORT} `));


app.use('/API/users',router);
```

db.js file

```
import mong from 'mongoose';
```

```
export const _mong = mong.connect('mongodb://localhost:27017/Woof',
{useNewUrlParser : true,useUnifiedTopology : true})
.then( () => console.log('Mongodb connection successful...') )
.catch( (err) => console.log(err));
```

user.js file

```
import mong from 'mongoose';
const schema = new mong.Schema({
   name : 'string',
   email : 'string',
   password: 'string',
   phno : 'Number',
   address: 'string'});
export var User = mong.model('User',schema,'Users');
```

userRoute.js file

```
import express from 'express';
import {getUser,addUser,getUserById,updateUser,deleteUser} from
'../controllers/userController.js';

const router = express.Router();
//url for this requests starts with http://localhost:8080/API/users/
router.get('/list', getUser);
router.get('/:id', getUserById);
router.put('/:id', updateUser);
router.delete('/:id',deleteUser);
router.post('/', addUser);

export default router;
```

userController.js file

```
import { User } from '../models/user.js';
import mongoose from 'mongoose';

let ObjectId = mongoose.Types.ObjectId;

export const getUser = (req,res) => {
  User.find((err,data)=>{
    if(err){
      console.log('Error in retreiving mongodb data : ' +
JSON.stringify(err,null,2));
    }
    else{
      res.send(data);
    }
  });
};

export const getUserById = (req,res) => {
  if(!ObjectId.isValid(req.params.id))
    return res.status(400).send(`No record with given id ${req.params.id}`);

    User.findById(req.params.id, (err,data)=>{
      if(err)
```

```
            console.log('Error in retriving employee data : ' +
JSON.stringify(err,null,2));
         else
            res.send(data);
      });
};

export const addUser = (req,res) => {
   let user = new User({
      name : req.body.name,
      email : req.body.email,
      password : req.body.password,
      phno : req.body.phno,
      address : req.body.address
   });

   user.save((err,data)=>{
      if(err)
         console.log('Error in saving data : ' + JSON.stringify(err,null,2));
      else
         res.send(data);
   });
}

export const updateUser = (req,res) => {
   if(!ObjectId.isValid(req.params.id))
   return res.status(400).send(`No record with given id ${req.params.id}`);

   let user = {
        name : req.body.name,
        email : req.body.email,
        password : req.body.password,
        phno : req.body.phno,
        address : req.body.address
    };

    User.findByIdAndUpdate(req.params.id, {$set : user}, {new : true},
(err,data)=>{
        if(err)
            console.log(`Error in updating employee data : ` +
JSON.stringify(err,null,2));
        else
            res.send(data);
    });
};

export const deleteUser = (req,res) => {
    if(!ObjectId.isValid(req.params.id))
    return res.status(400).send(`No record with given id ${req.params.id}`);

    User.findByIdAndRemove(req.params.id, (err,data) => {
        if(err)
            console.log('Error in removing employee : ' +
JSON.stringify(err,null,2));
        else
            res.send(data);
    });
}
```

CHAPTER 7
TESTING

7.1 Testing Plan

Every developers nightmare but also very crucial part of development, testing. But for this system at the initial part of development testing section have been divided and is done using different tools. The main idea is to get the basic functionalities right and get rid of the bugs at the root so that it won't become a problem later. The standard testing strategies also starts from the individual testing then go up to merging and system testing.

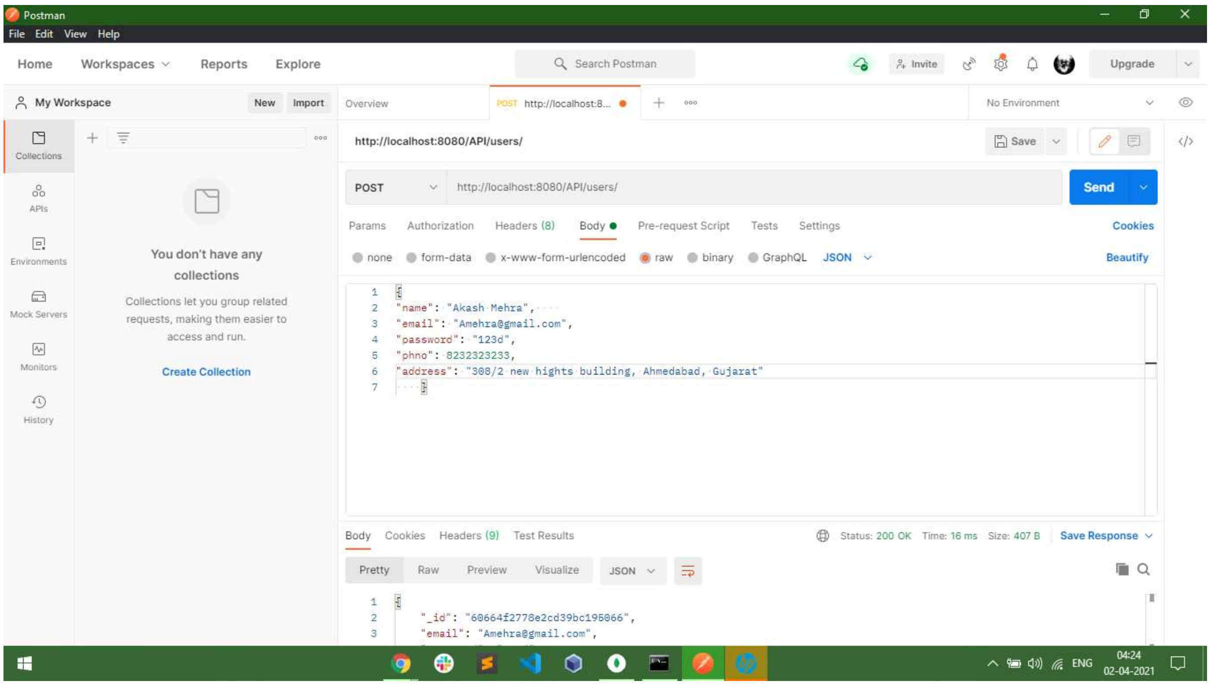
7.2 Testing Methods And Tools

As planed the system testing goes through multiple testings. Firstly the individual testing or unit testing which is done manual, no auto testing tools are used in this. Later when it comes to functional testing tools like Compass, Postman along with some manual testing which helps to test and review the functionalities. Lastly for integration testing, chai.http and Angular Karma is used.

7.3 Testing Cases

TEST NAME	**INPUT**	**OUTPUT**	**RESULT**	**COMMENT**
1)API test	Get request URLs	JSON format data from database	Success	CORS working
2) User authenti cation	Credential s	User details	Success	Validation working
3) Addi ng Request	Request and Request	Request confirmation and updates	Success	Requests are generated and showing in

	Details			db.
4)Add/remove/update user	Credentials	Operation Confirmation	Success	Admin and user can perform privileged actions



POST http://localhost:8080/API/users/

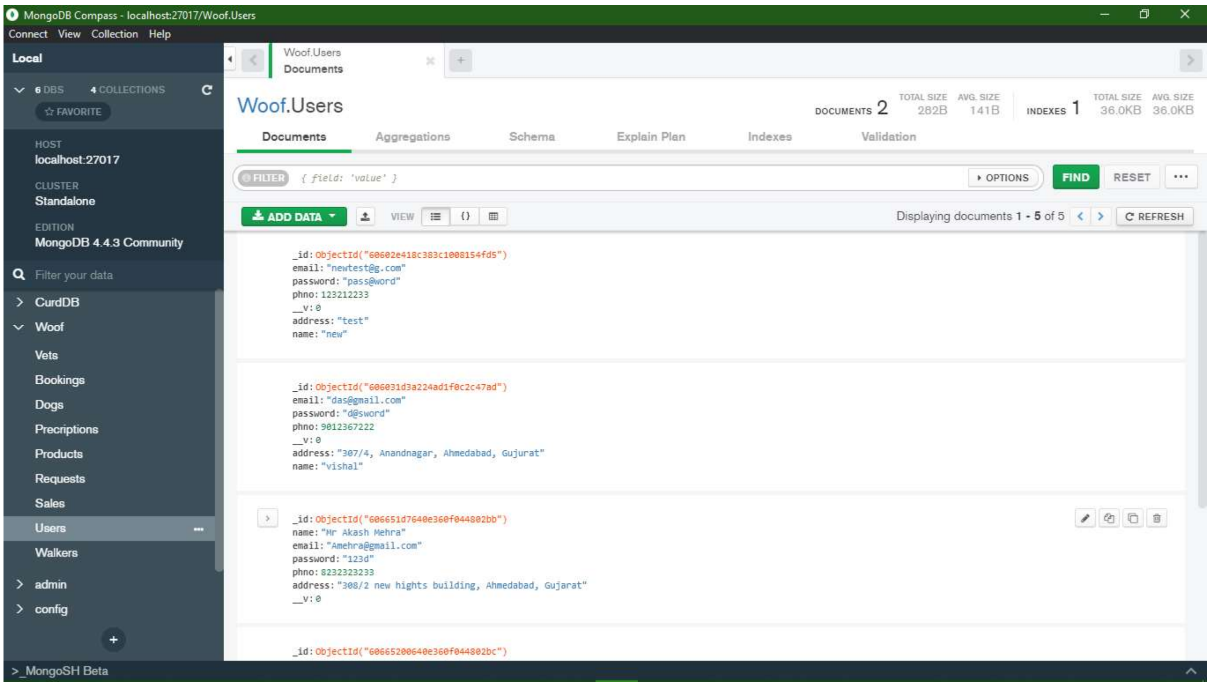
```
{
"name": "Akash Mehra",
"email": "Amehra@gmail.com",
"password": "123d",
"phno": 8232323233,
"address": "308/2 new hights building, Ahmedabad, Gujarat"
    }
```

Status: 200 OK Time: 16 ms Size: 407 B

```
{
    "_id": "60664f2778e2cd39bc195066",
    "email": "Amehra@gmail.com",
```

Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Invite Upgrade
My Workspace New Import
Collections APIs Environments Mock Servers Monitors History
You don't have any collections
Collections let you group related requests, making them easier to access and run.
Create Collection
Overview
No Environment
Save
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify
Body Cookies Headers (9) Test Results Save Response
Pretty Raw Preview Visualize JSON
ENG 04:24 02-04-2021

Fig.- Testing with Postman

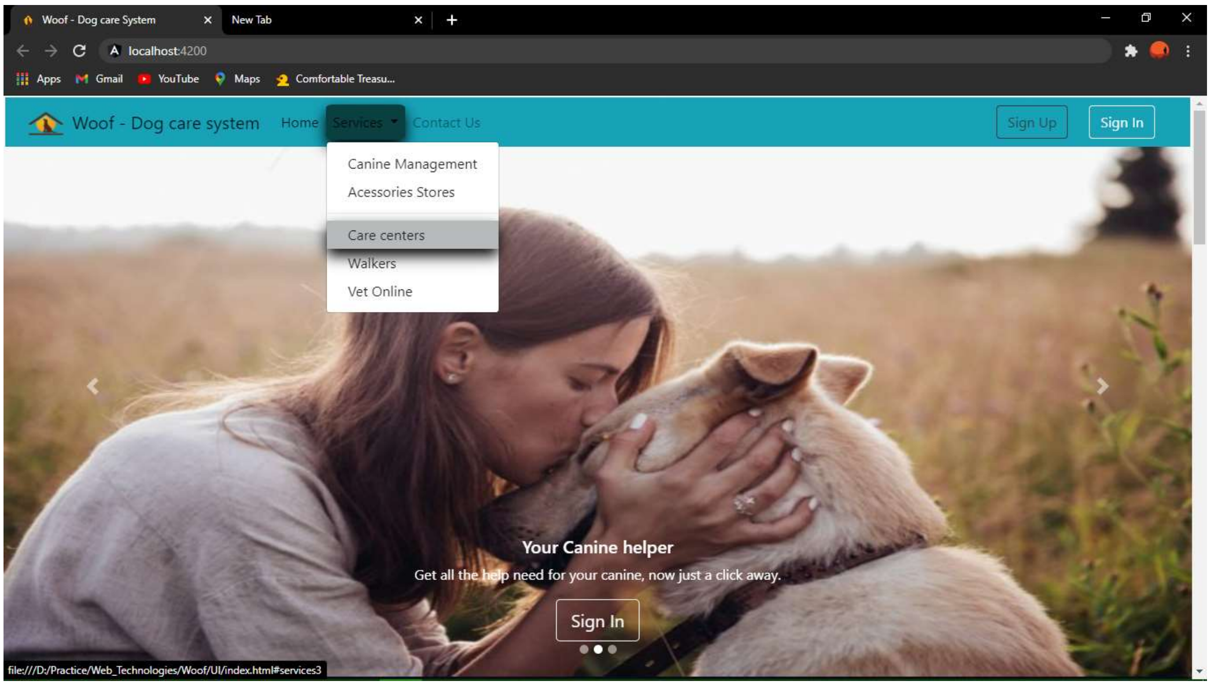

MongoDB Compass - localhost:27017/Woof.Users
Connect View Collection Help
Local
6 DBS 4 COLLECTIONS
FAVORITE
HOST localhost:27017
CLUSTER Standalone
EDITION MongoDB 4.4.3 Community
Filter your data
CurdDB
Woof
Vets
Bookings
Dogs
Precriptions
Products
Requests
Sales
Users
Walkers
admin
config
>_MongoSH Beta
Woof.Users Documents
Woof.Users
DOCUMENTS 2 TOTAL SIZE 282B AVG. SIZE 141B
INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB
Documents Aggregations Schema Explain Plan Indexes Validation
FILTER { field: 'value' }
OPTIONS FIND RESET
ADD DATA VIEW
Displaying documents 1 - 5 of 5
REFRESH

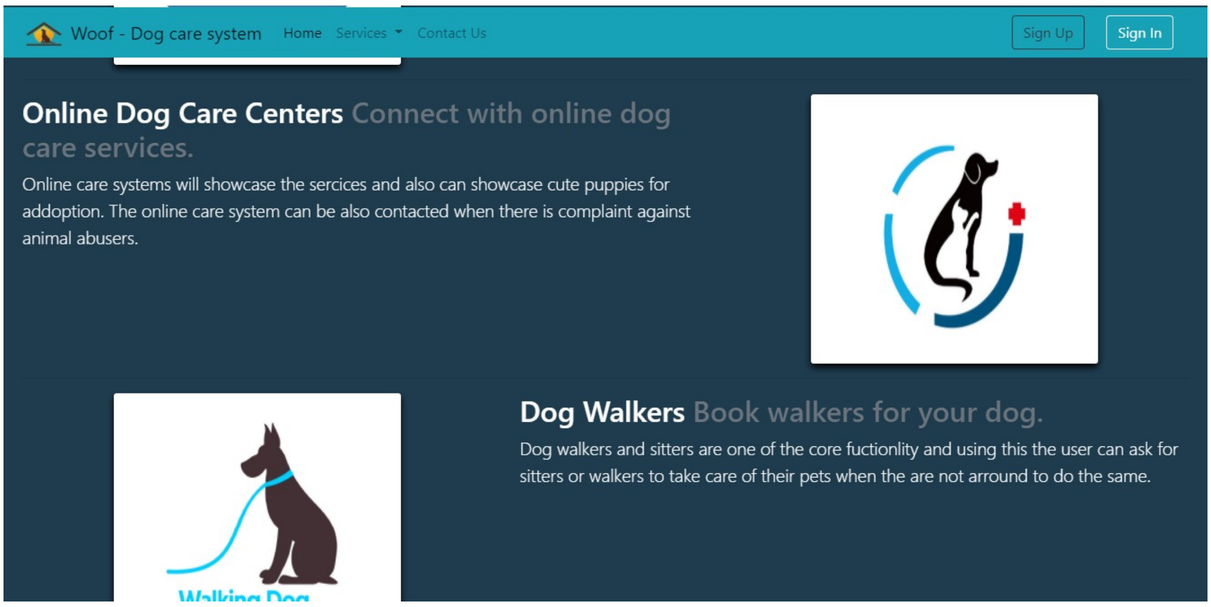
```
_id: ObjectId("60602e418c383c1008154fd5")
email: "newtest@g.com"
password: "pass@word"
phno: 123212233
__v: 0
address: "test"
name: "new"

_id: ObjectId("606031d3a224ad1f0c2c47ad")
email: "das@gmail.com"
password: "d@sword"
phno: 9012367222
__v: 0
address: "307/4, Anandnagar, Ahmedabad, Gujurat"
name: "vishal"

_id: ObjectId("606651d7640e360f044802bb")
name: "Mr Akash Mehra"
email: "Amehra@gmail.com"
password: "123d"
phno: 8232323233
address: "308/2 new hights building, Ahmedabad, Gujarat"
__v: 0

_id: ObjectId("60665200640e360f044802bc")
```

Fig.- Testing with MongoDbCompass

CHAPTER 8

SCREENSHOTS AND USER MANUAL



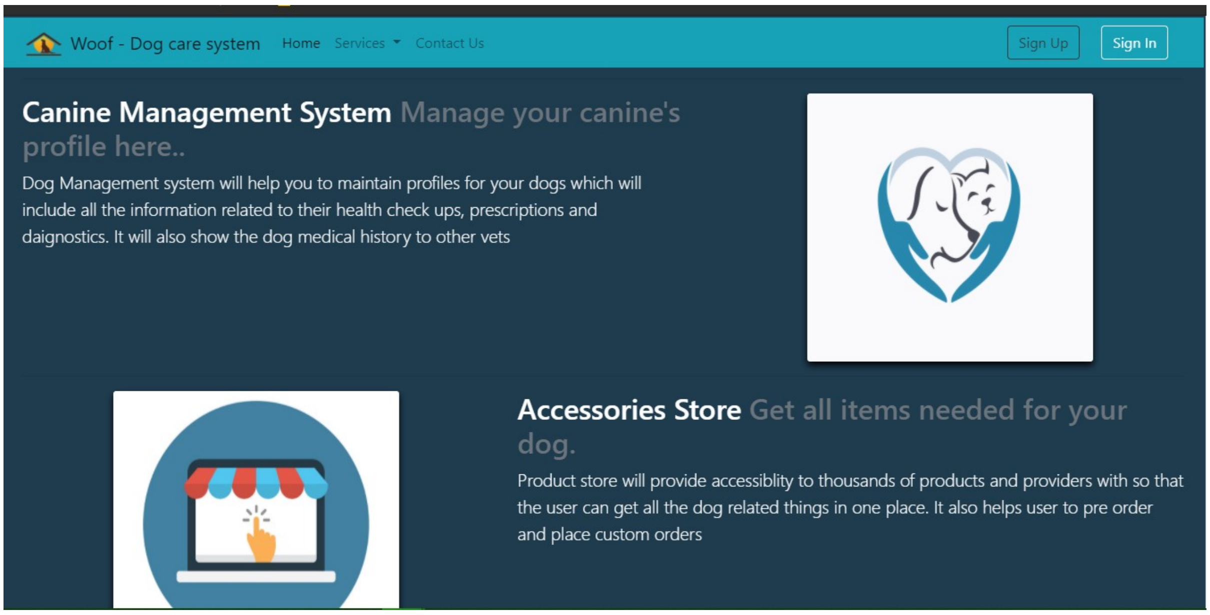


Online Dog Care Centers Connect with online dog care services.

Online care systems will showcase the sercices and also can showcase cute puppies for addoption. The online care system can be also contacted when there is complaint against animal abusers.

Dog Walkers Book walkers for your dog.

Dog walkers and sitters are one of the core fuctionlity and using this the user can ask for sitters or walkers to take care of their pets when the are not arround to do the same.

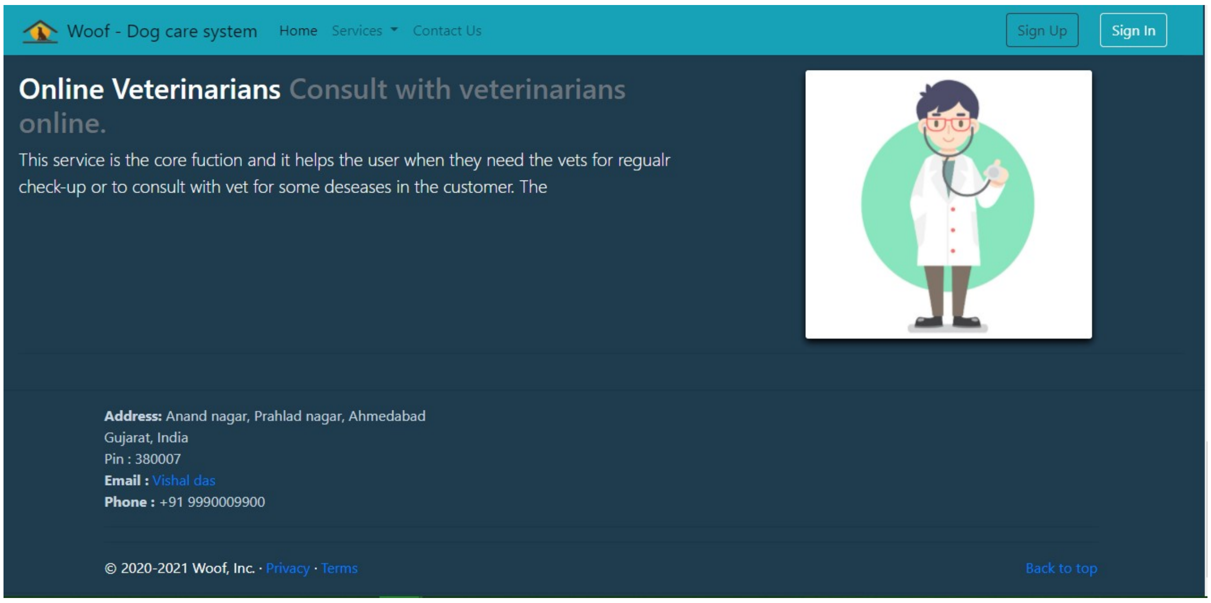


Canine Management System Manage your canine's profile here..

Dog Management system will help you to maintain profiles for your dogs which will include all the information related to their health check ups, prescriptions and daignostics. It will also show the dog medical history to other vets

Accessories Store Get all items needed for your dog.

Product store will provide accessiblity to thousands of products and providers with so that the user can get all the dog related things in one place. It also helps user to pre order and place custom orders



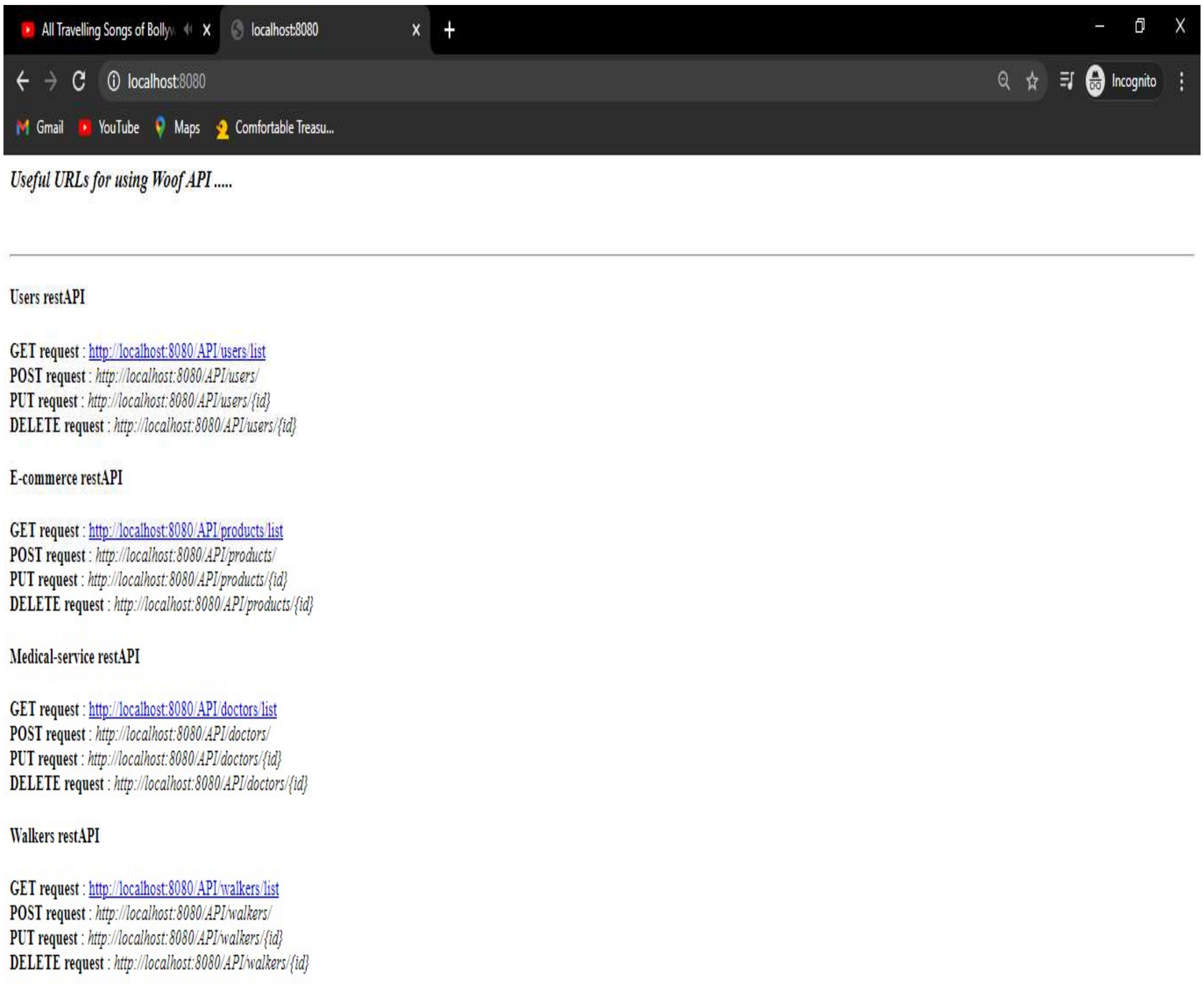
Online Veterinarians Consult with veterinarians online.

This service is the core fuction and it helps the user when they need the vets for regualr check-up or to consult with vet for some deseases in the customer. The

Address: Anand nagar, Prahlad nagar, Ahmedabad
Gujarat, India
Pin : 380007
Email : Vishal das
Phone : +91 9990009900

© 2020-2021 Woof, Inc. · Privacy · Terms

Back to top



Useful URLs for using Woof API .....

Users restAPI

GET request : http://localhost:8080/API/users/list
POST request : *http://localhost:8080/API/users/*
PUT request : *http://localhost:8080/API/users/{id}*
DELETE request : *http://localhost:8080/API/users/{id}*

E-commerce restAPI

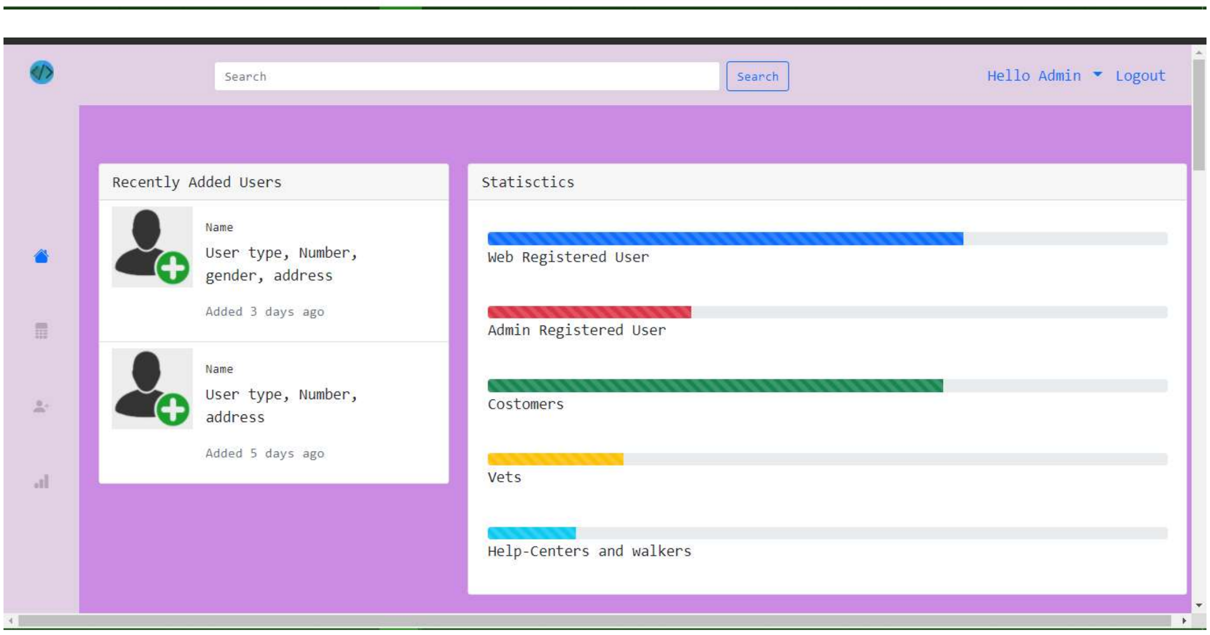
GET request : http://localhost:8080/API/products/list
POST request : *http://localhost:8080/API/products/*
PUT request : *http://localhost:8080/API/products/{id}*
DELETE request : *http://localhost:8080/API/products/{id}*

Medical-service restAPI

GET request : http://localhost:8080/API/doctors/list
POST request : *http://localhost:8080/API/doctors/*
PUT request : *http://localhost:8080/API/doctors/{id}*
DELETE request : *http://localhost:8080/API/doctors/{id}*

Walkers restAPI

GET request : http://localhost:8080/API/walkers/list
POST request : *http://localhost:8080/API/walkers/*
PUT request : *http://localhost:8080/API/walkers/{id}*
DELETE request : *http://localhost:8080/API/walkers/{id}*



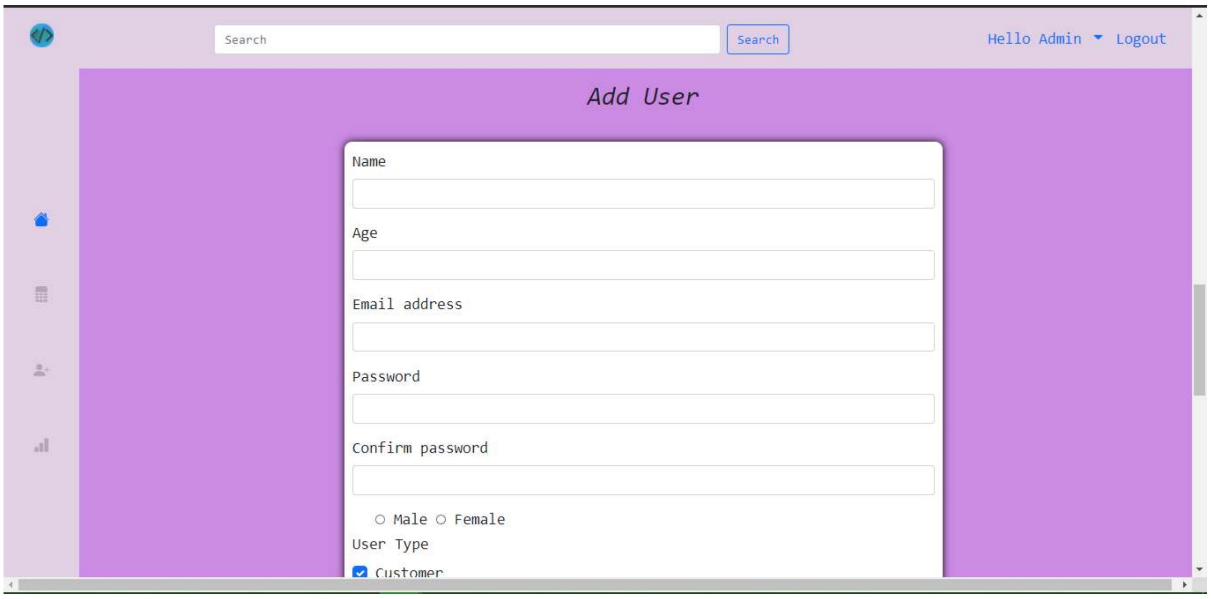
Recently Added Users

Name
User type, Number, gender, address
Added 3 days ago

Name
User type, Number, address
Added 5 days ago

Statisctics

Web Registered User
Admin Registered User
Costomers
Vets
Help-Centers and walkers



Add User

Name

Age

Email address

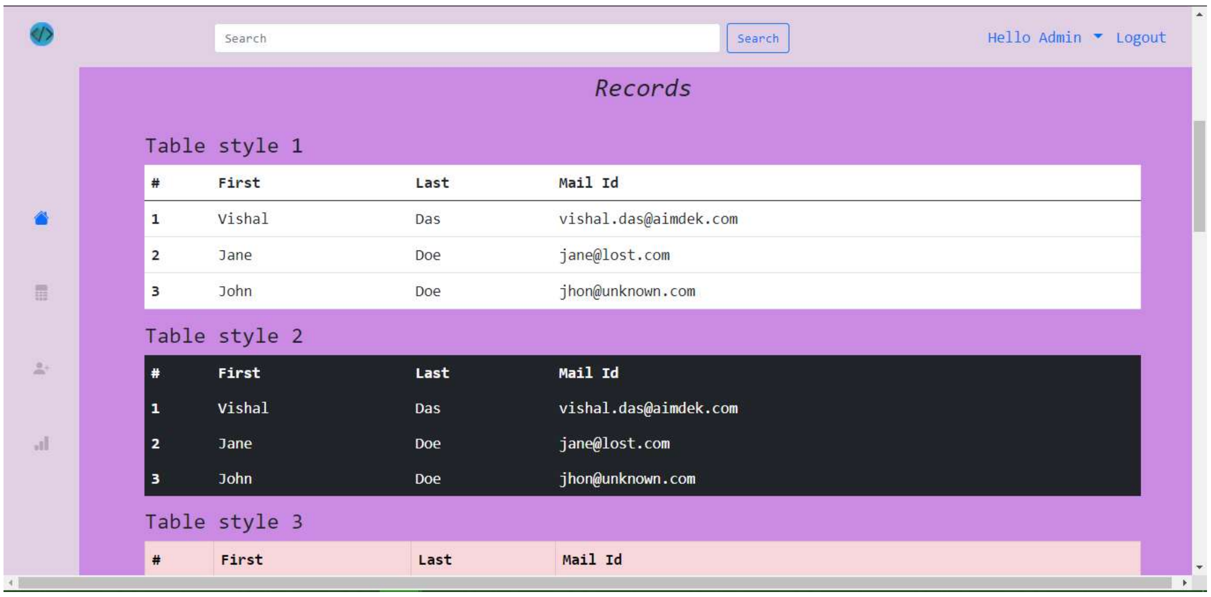
Password

Confirm password

○ Male ○ Female

User Type

[✓ Customer]



Records

Table style 1

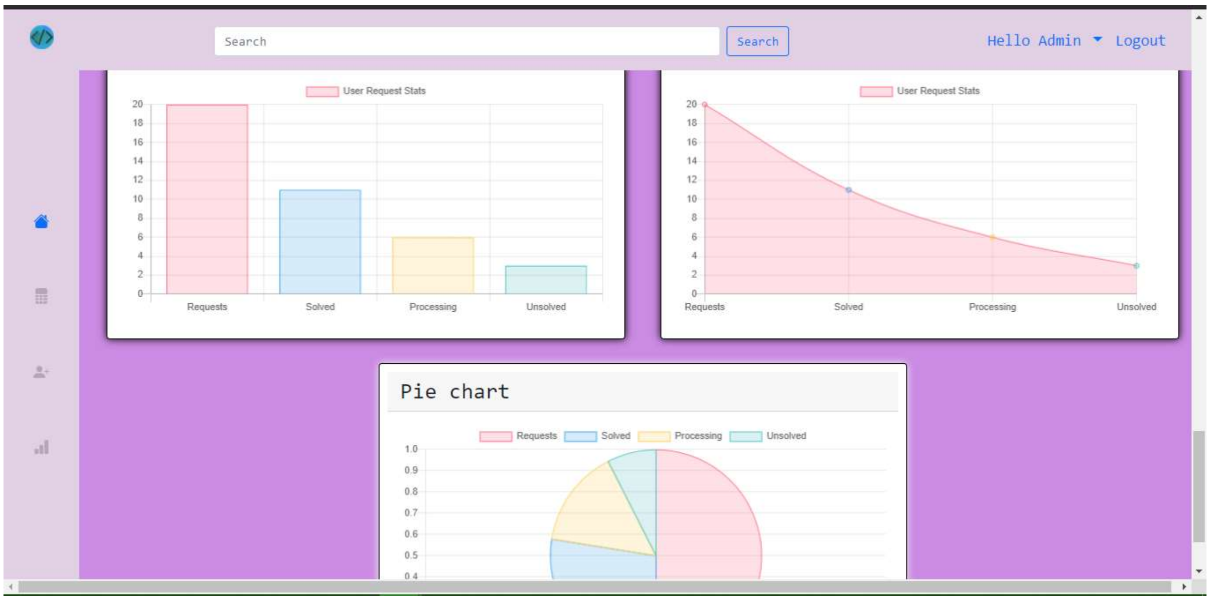
#	First	Last	Mail Id
1	Vishal	Das	vishal.das@aimdek.com
2	Jane	Doe	jane@lost.com
3	John	Doe	jhon@unknown.com

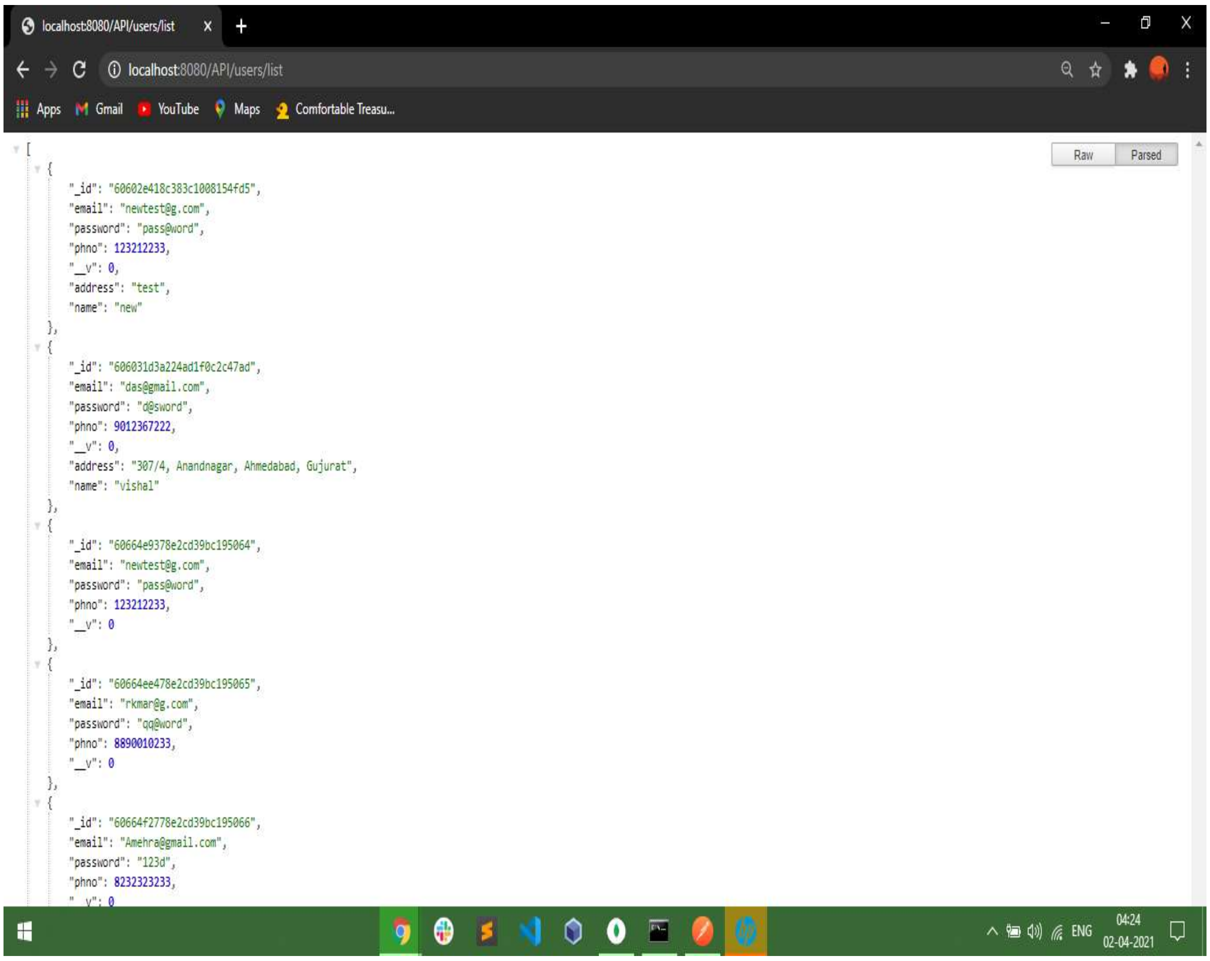
Table style 2

#	First	Last	Mail Id
1	Vishal	Das	vishal.das@aimdek.com
2	Jane	Doe	jane@lost.com
3	John	Doe	jhon@unknown.com

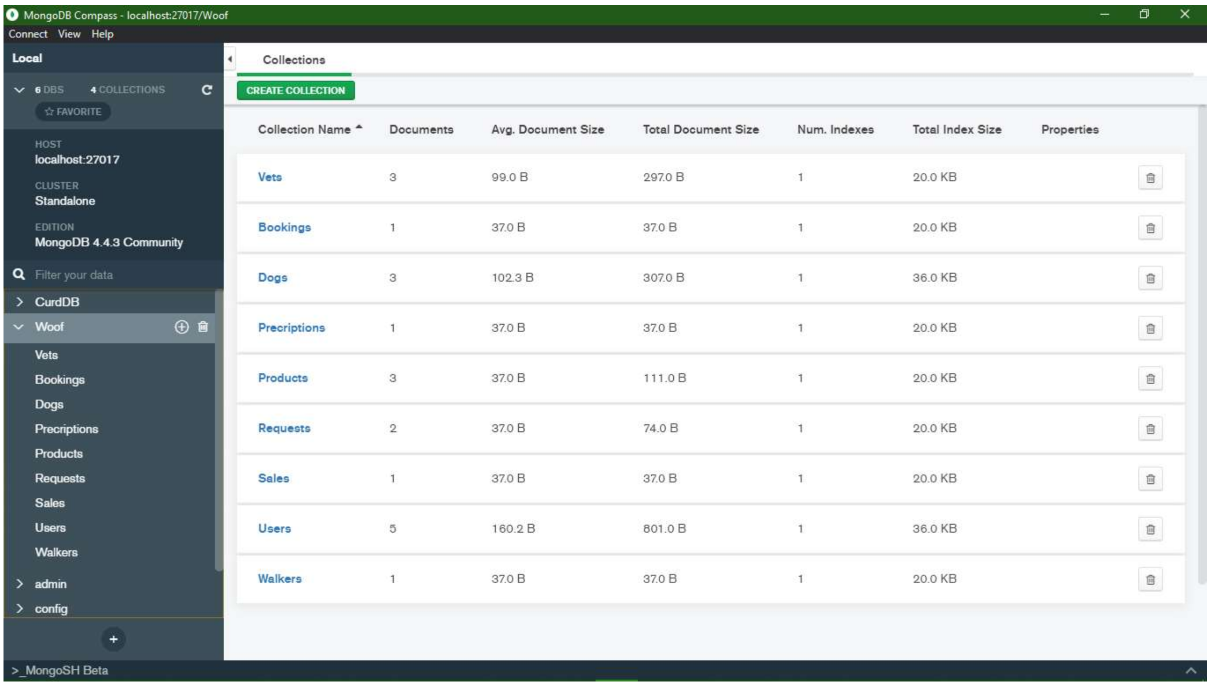
Table style 3

#	First	Last	Mail Id





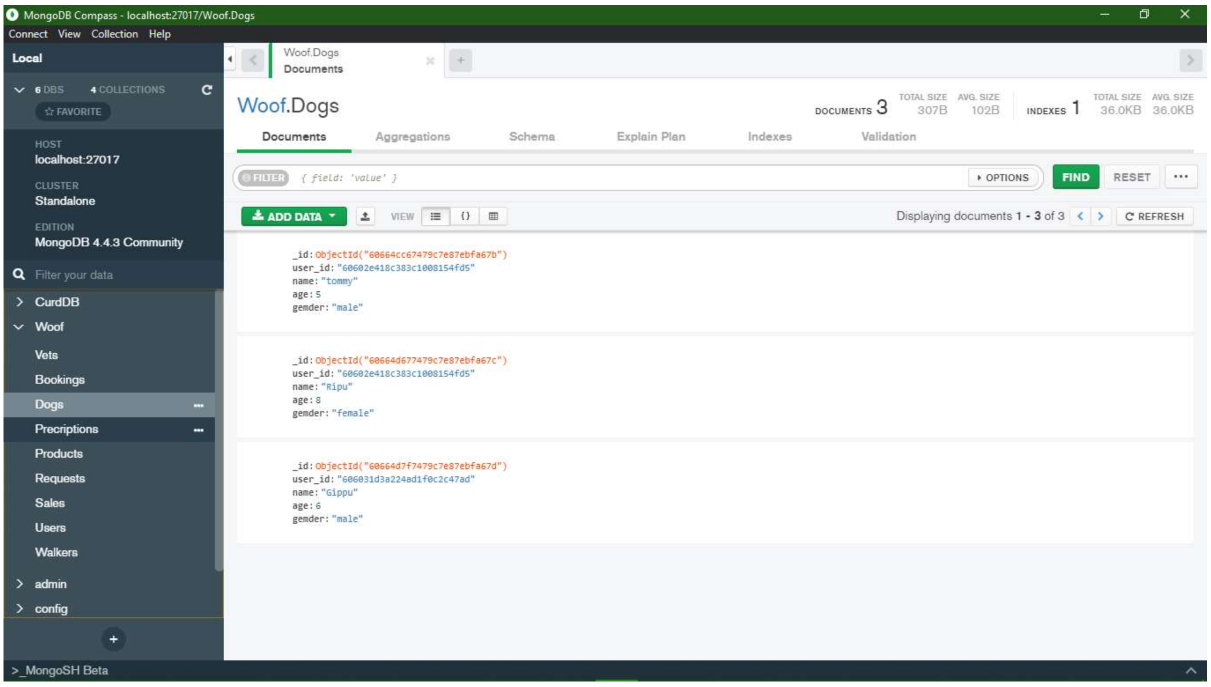
```
[
  {
    "_id": "60602e418c383c1008154fd5",
    "email": "newtest@g.com",
    "password": "pass@word",
    "phno": 123212233,
    "__v": 0,
    "address": "test",
    "name": "new"
  },
  {
    "_id": "606031d3a224ad1f0c2c47ad",
    "email": "das@gmail.com",
    "password": "d@sword",
    "phno": 9012367222,
    "__v": 0,
    "address": "307/4, Anandnagar, Ahmedabad, Gujurat",
    "name": "vishal"
  },
  {
    "_id": "60664e9378e2cd39bc195064",
    "email": "newtest@g.com",
    "password": "pass@word",
    "phno": 123212233,
    "__v": 0
  },
  {
    "_id": "60664ee478e2cd39bc195065",
    "email": "rkmar@g.com",
    "password": "qq@word",
    "phno": 8890010233,
    "__v": 0
  },
  {
    "_id": "60664f2778e2cd39bc195066",
    "email": "Amehra@gmail.com",
    "password": "123d",
    "phno": 8232323233,
    "__v": 0
```


MongoDB Compass - localhost:27017/Woof

Local — 6 DBS, 4 COLLECTIONS

HOST localhost:27017
CLUSTER Standalone
EDITION MongoDB 4.4.3 Community

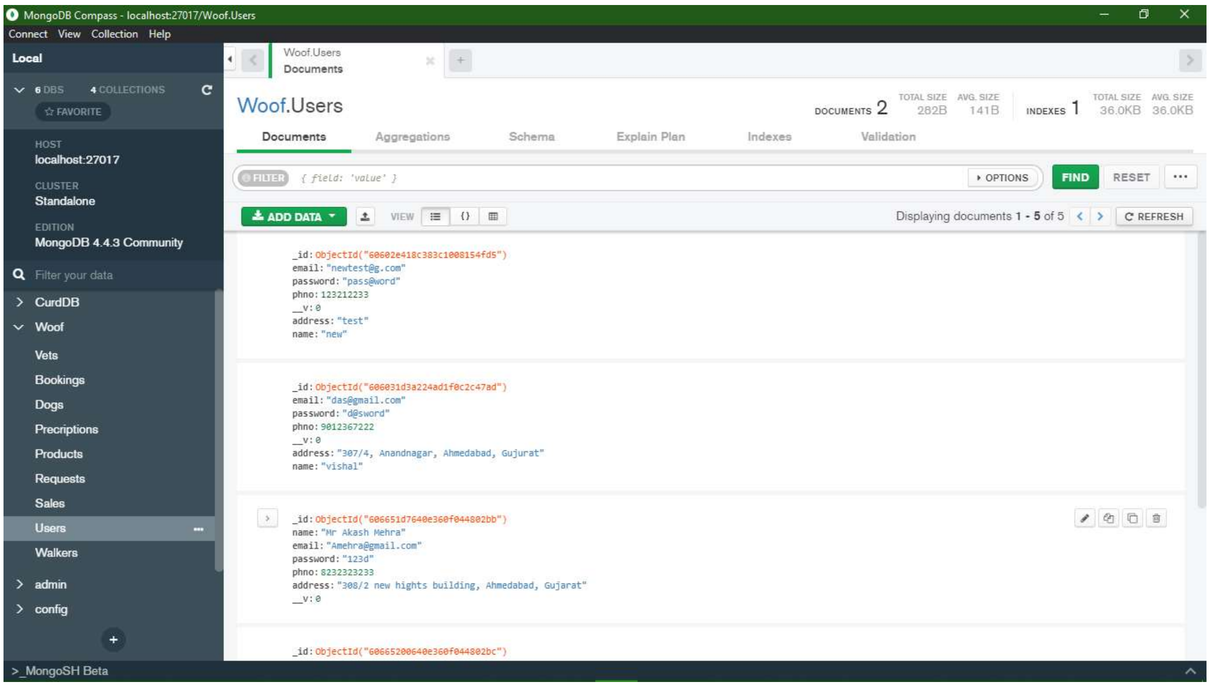
Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
Vets	3	99.0 B	297.0 B	1	20.0 KB	
Bookings	1	37.0 B	37.0 B	1	20.0 KB	
Dogs	3	102.3 B	307.0 B	1	36.0 KB	
Precriptions	1	37.0 B	37.0 B	1	20.0 KB	
Products	3	37.0 B	111.0 B	1	20.0 KB	
Requests	2	37.0 B	74.0 B	1	20.0 KB	
Sales	1	37.0 B	37.0 B	1	20.0 KB	
Users	5	160.2 B	801.0 B	1	36.0 KB	
Walkers	1	37.0 B	37.0 B	1	20.0 KB	


MongoDB Compass - localhost:27017/Woof.Dogs

Woof.Dogs — DOCUMENTS 3, TOTAL SIZE 307B, AVG. SIZE 102B; INDEXES 1, TOTAL SIZE 36.0KB, AVG. SIZE 36.0KB

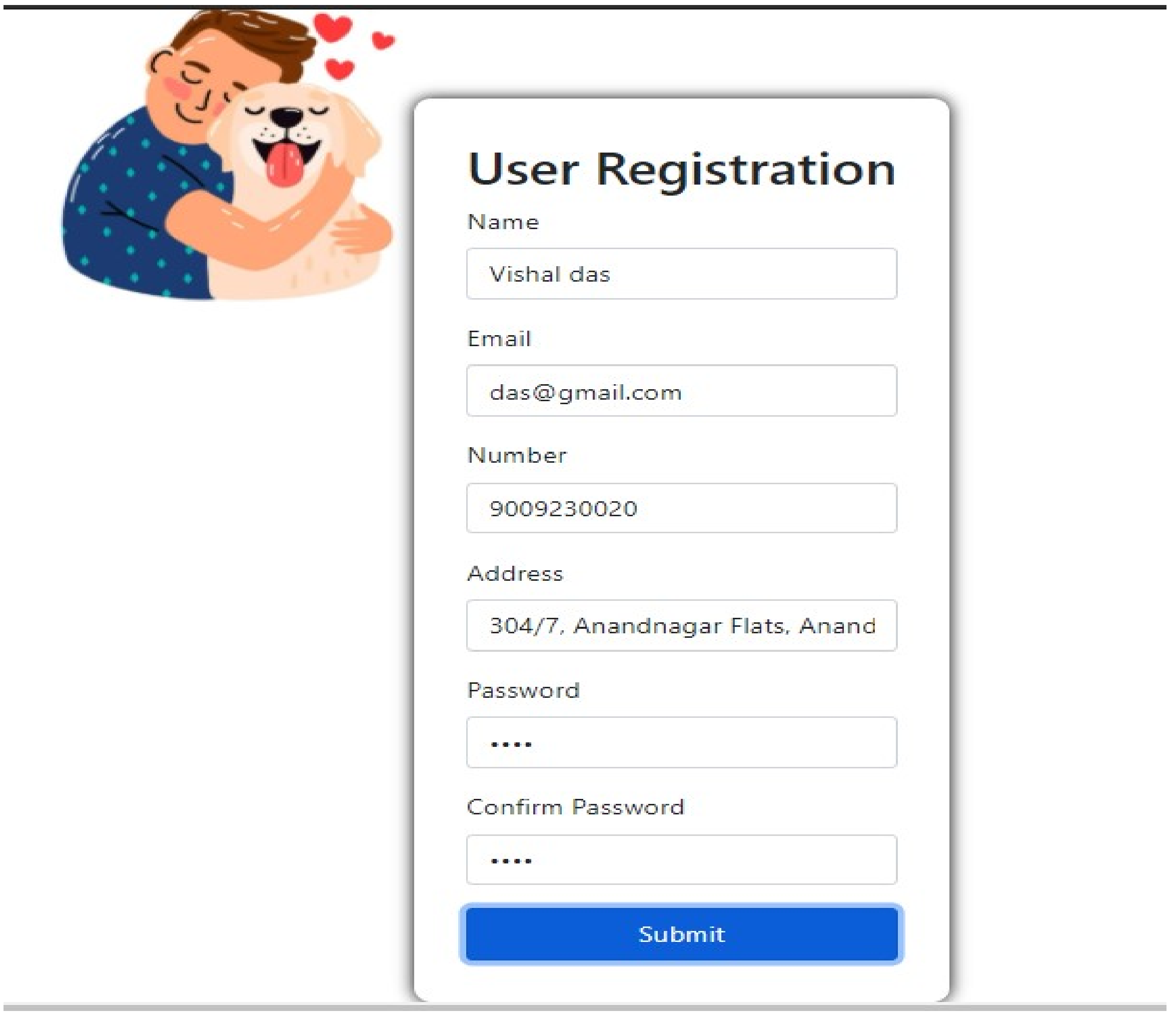
Displaying documents 1 - 3 of 3

```
_id: ObjectId("60664cc67479c7e87ebfa67b")
user_id: "60602e418c383c1008154fd5"
name: "tommy"
age: 5
gemder: "male"

_id: ObjectId("60664d677479c7e87ebfa67c")
user_id: "60602e418c383c1008154fd5"
name: "Ripu"
age: 8
gemder: "female"

_id: ObjectId("60664d7f7479c7e87ebfa67d")
user_id: "606031d3a224ad1f0c2c47ad"
name: "Gippu"
age: 6
gemder: "male"
```



```
{ field: 'value' }
```

```
_id: ObjectId("60602e418c383c1008154fd5")
email: "newtest@g.com"
password: "pass@word"
phno: 123212233
__v: 0
address: "test"
name: "new"

_id: ObjectId("606031d3a224ad1f0c2c47ad")
email: "das@gmail.com"
password: "d@sword"
phno: 9012367222
__v: 0
address: "307/4, Anandnagar, Ahmedabad, Gujurat"
name: "vishal"

_id: ObjectId("606651d7640e360f044802bb")
name: "Mr Akash Mehra"
email: "Amehra@gmail.com"
password: "123d"
phno: 8232323233
address: "308/2 new hights building, Ahmedabad, Gujarat"
__v: 0

_id: ObjectId("6066520064 0e360f044802bc")
```



User Registration

Name: Vishal das

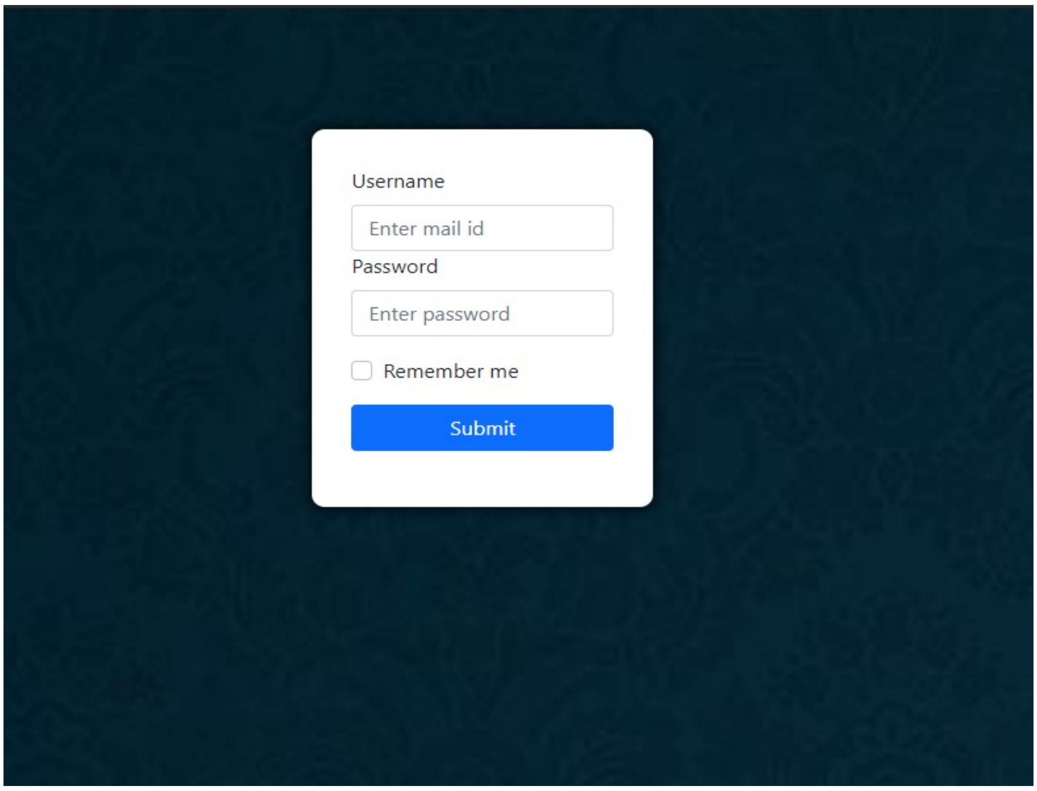
Email: das@gmail.com

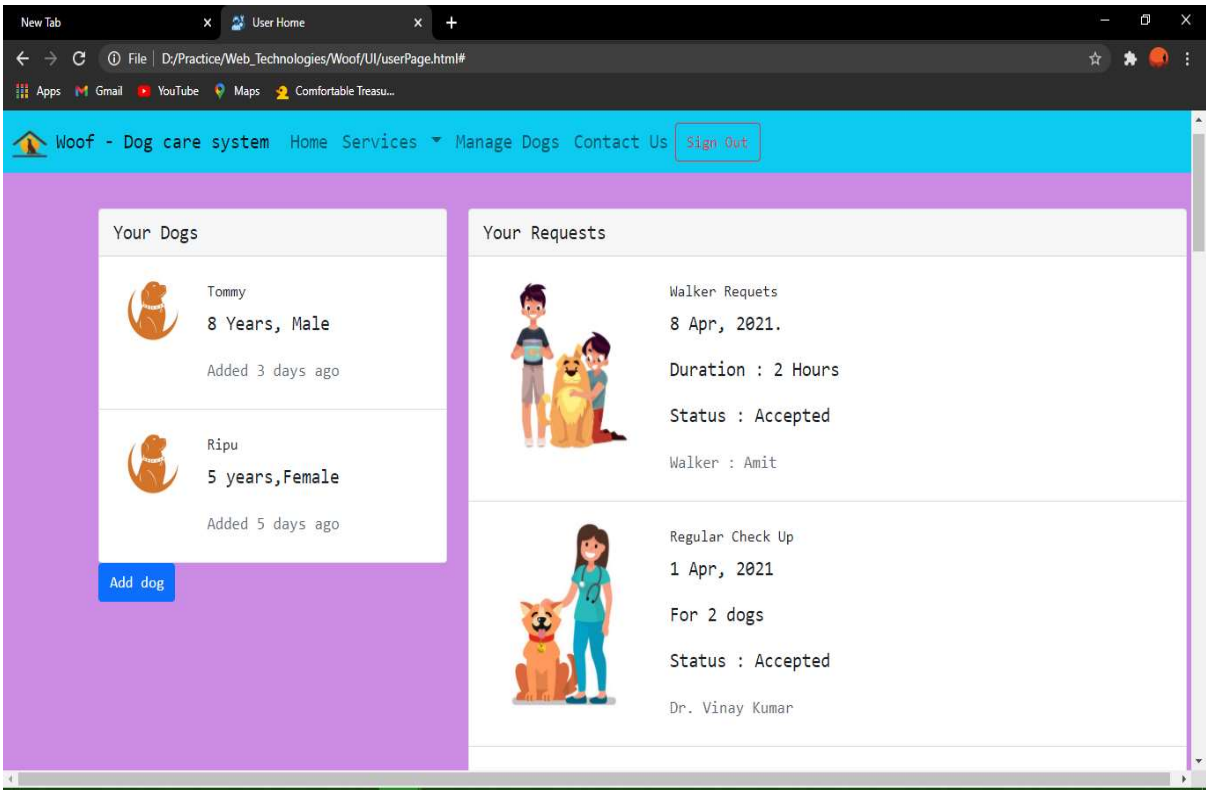
Number: 9009230020

Address: 304/7, Anandnagar Flats, Anand

Password: ••••

Confirm Password: ••••





Your Dogs

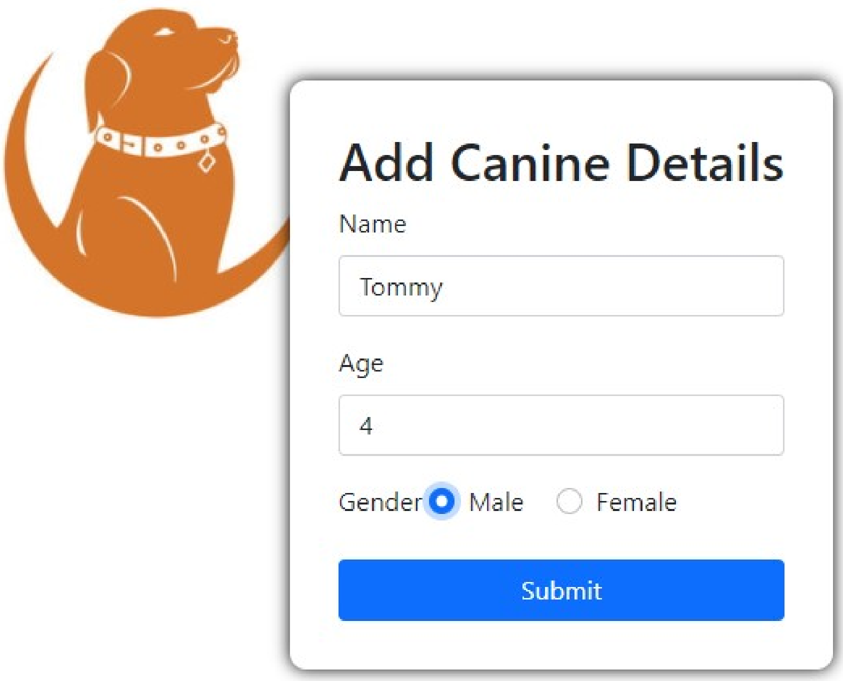
Tommy
8 Years, Male
Added 3 days ago

Ripu
5 years,Female
Added 5 days ago

Your Requests

Walker Requets
8 Apr, 2021.
Duration : 2 Hours
Status : Accepted
Walker : Amit

Regular Check Up
1 Apr, 2021
For 2 dogs
Status : Accepted
Dr. Vinay Kumar

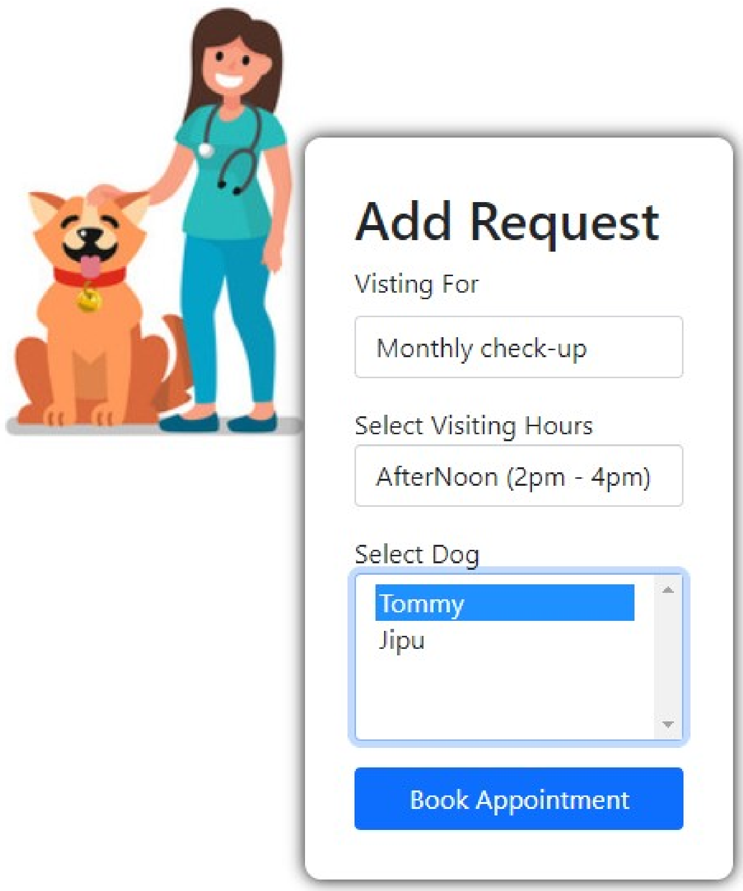


Add Canine Details

Name: Tommy

Age: 4

Gender [✓ Male] Female


Add Request

Visting For

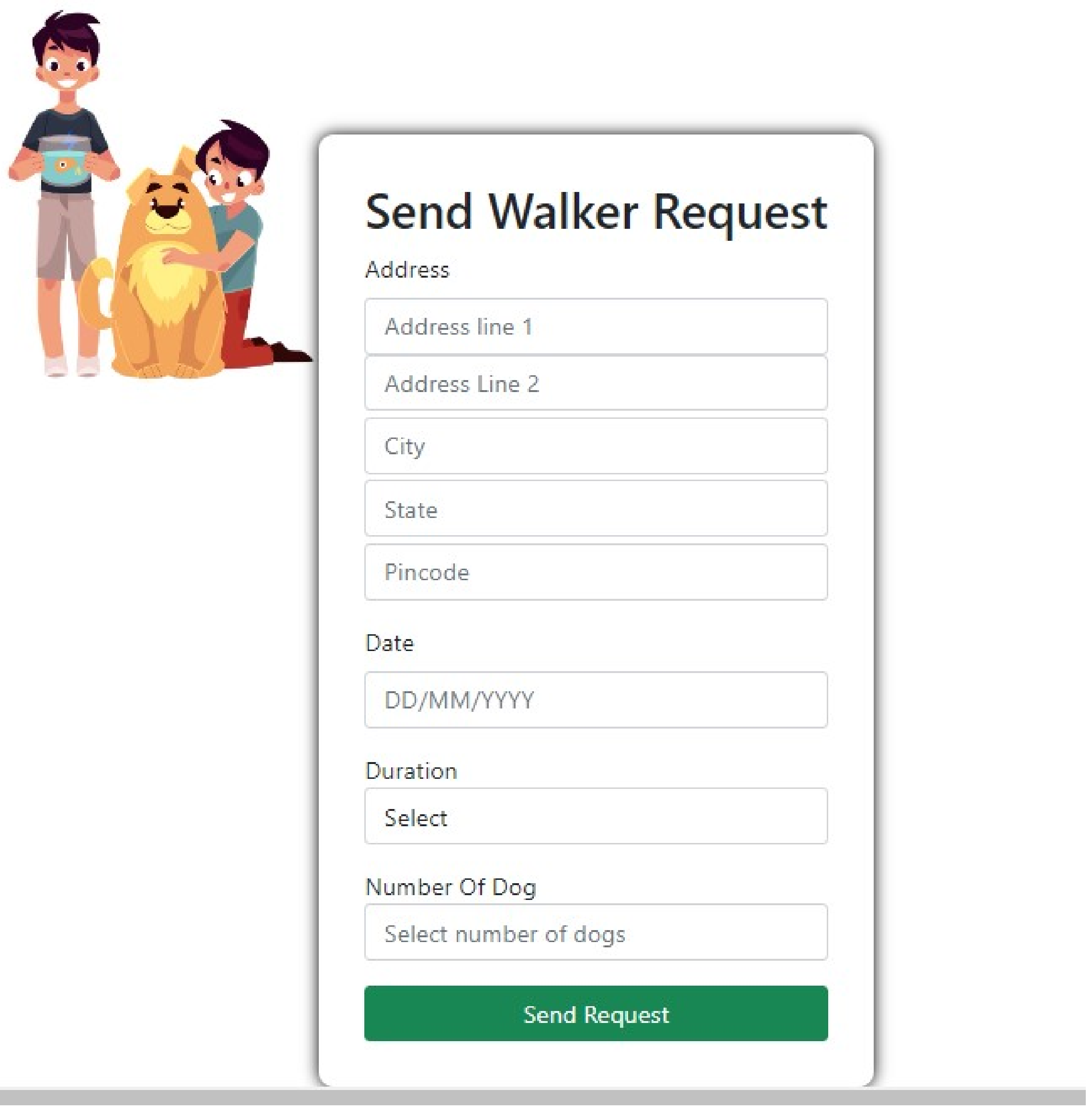
Monthly check-up

Select Visiting Hours

AfterNoon (2pm - 4pm)

Select Dog

Tommy
Jipu



Send Walker Request

Address

Address line 1

Address Line 2

City

State

Pincode

Date

DD/MM/YYYY

Duration

Select

Number Of Dog

Select number of dogs

Send Request

CHAPTER 9

CONCLUSION

To conclude, following the above explanations and results some points are very clear that there are systems that provide dog care and other related services which are working fine, but none of them are providing all the services that is intended by Woof The Dog Care System, and it also aims include other services so that every dog lover and their beloved canine get the best and all they need at the same place. Other than that we also intend to provide the service providers a platform where they can easily grow their business by serving as bridge between them and huge crowd of customers. These points suggest strongly that we need something like Woof and its services will serve many.

CHAPTER 10
REFERENCES

1) https://nodejs.org/en/

2) https://angular.io/

3) https://www.typescriptlang.org/

4) https://www.w3schools.com/js/

5) https://getbootstrap.com/docs/4.0/getting-started/introduction/

6) https://www.chartjs.org/

7) https://docs.mongodb.com/manual/

8) https://expressjs.com/

9) https://www.postman.com/

10) https://www.canva.com/

11) https://creately.com/

12) https://www.lucidchart.com/

13) https://www.youtube.com/watch?v=WBPrJSw7yQA

14) https://www.youtube.com/watch?v=l8WPWK9mS5M

15) https://www.youtube.com/watch?v=UYh6EvpQquw

16) https://www.npmjs.com/package/nodemon

17) https://www.npmjs.com/package/cors

18) https://mongoosejs.com/

19) https://www.youtube.com/watch?v=0eWrpsCLMJQ&list=PLC3y8-rFHvwhBRAgFinJR8KHIrCdTkZcZ

20) https://karma-runner.github.io/latest/index.html