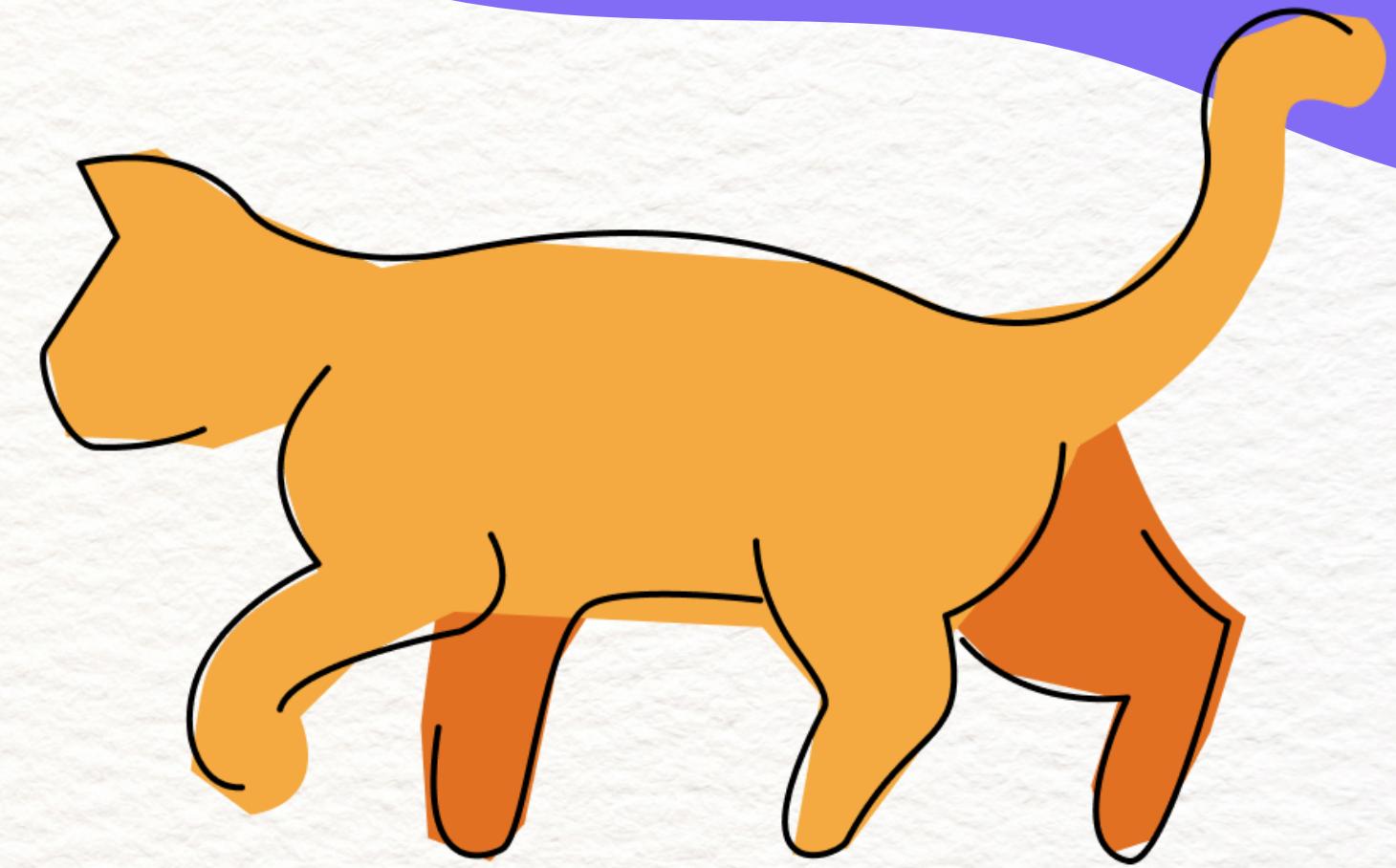


We'll help you
look after your
cat



Cat Care

Meet the Team



Jacob Brooks
Project Manager



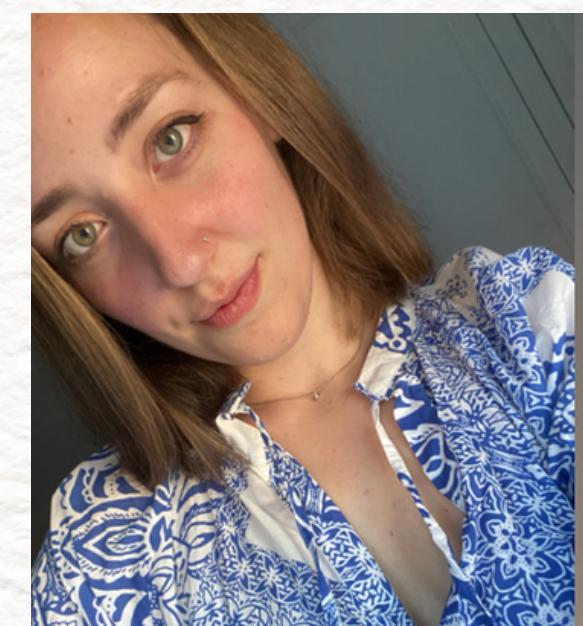
Alex Earle
FrontEnd



Harry Turner-Burns
FrontEnd



Lais Moraes
BackEnd



Roberta Capuano
BackEnd



Our Mission

Our aim is to make looking after your cat easier and stress-free.

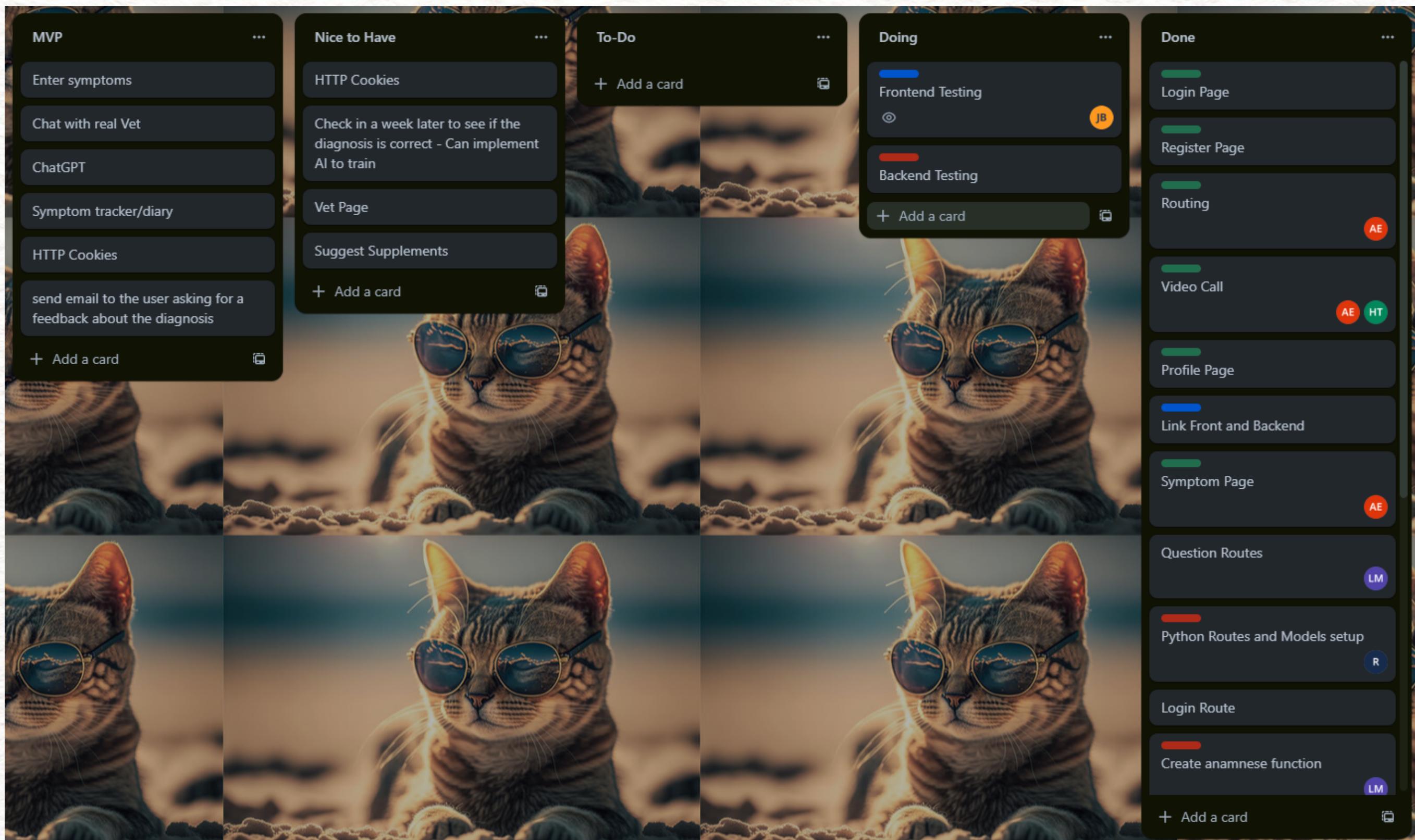
We have used real veterinary expertise to create a program that can diagnose your cat from information you provide.



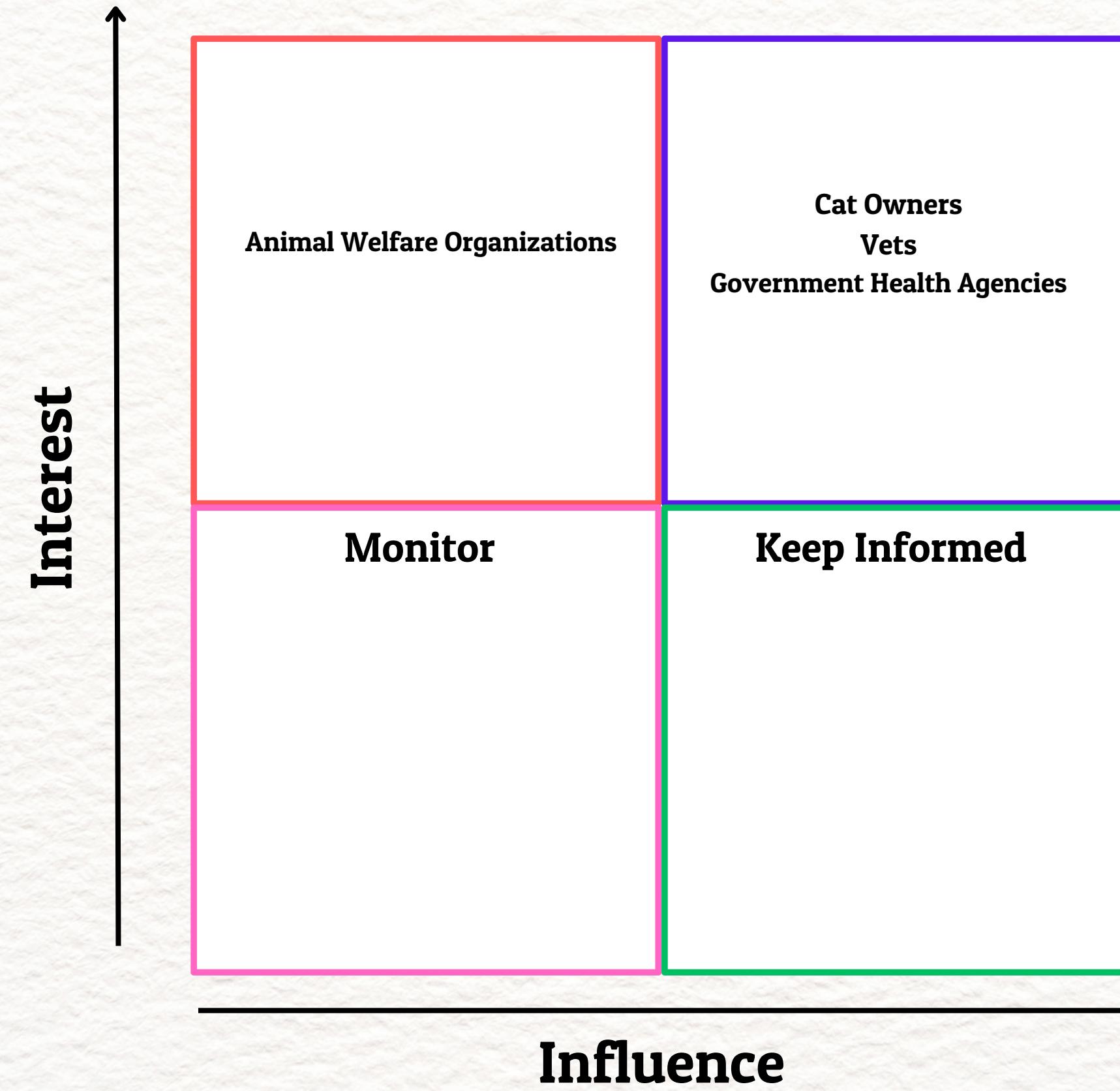
Planning And Delivery



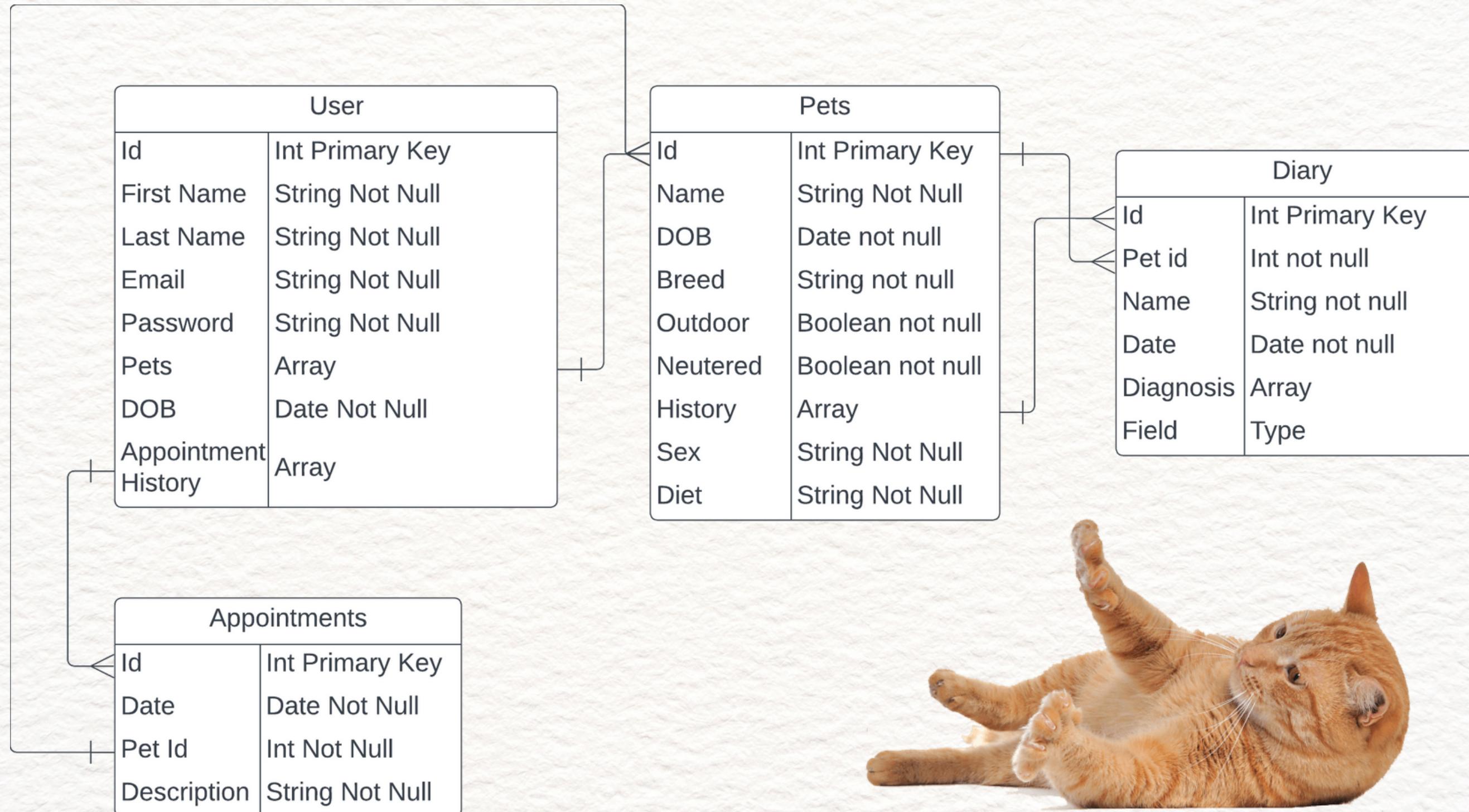
Trello



Stakeholders



Entity-Relationship-Diagram



Wireframes



Logo

Home Health Cat Location Menu

Gastrointestinal e.g. vomiting, diarrhoea

Respiratory e.g. coughing, short breath

Urinary e.g. incontinence, blood

Dermatological e.g. wounds, fur loss, itching

Musculoskeletal e.g. atrophy, limping

Neurological e.g. loss of balance, seizures

Ocular e.g. eye discharge, redness

I don't know

Logo

Home Health Cat Location Menu

Our Mission:
text text text

Check Symptoms

Chat to a Vet

Book a Video Appointment

Find Local Vets

Cat

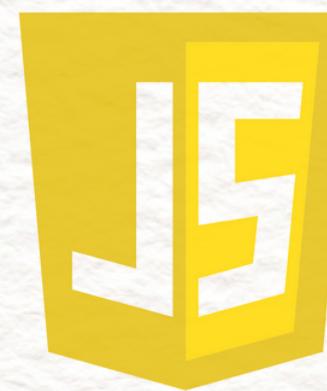
Moving Carousel of images

Technologies

HTML



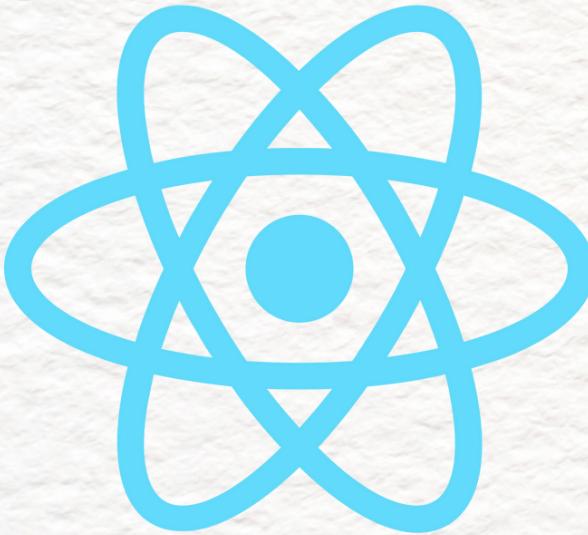
JS



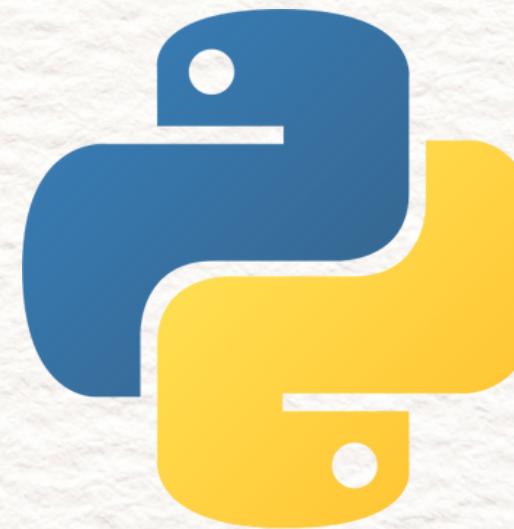
CSS



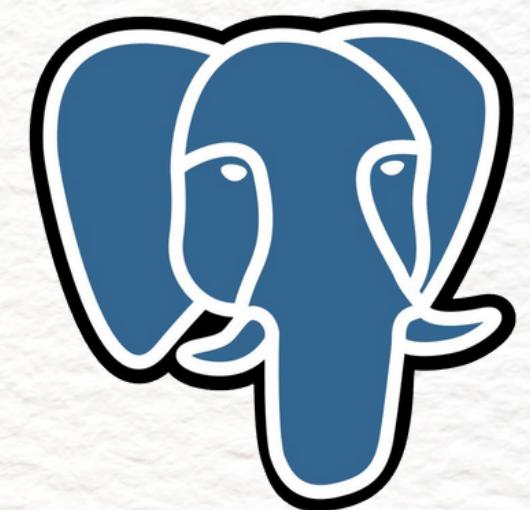
SQLAlchemy



OpenAI



pytest



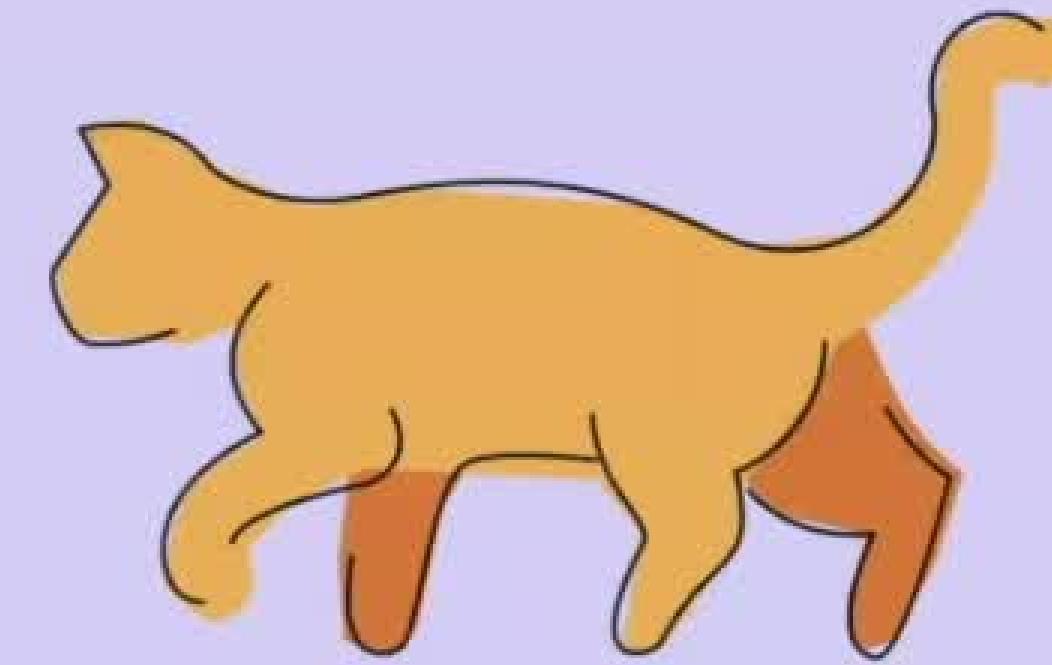
Flask

We'll help you look after your cat

Join us today to diagnose your cat
using our veterinary expertise.

[Join Us](#)

[Log In](#)



Cat Care

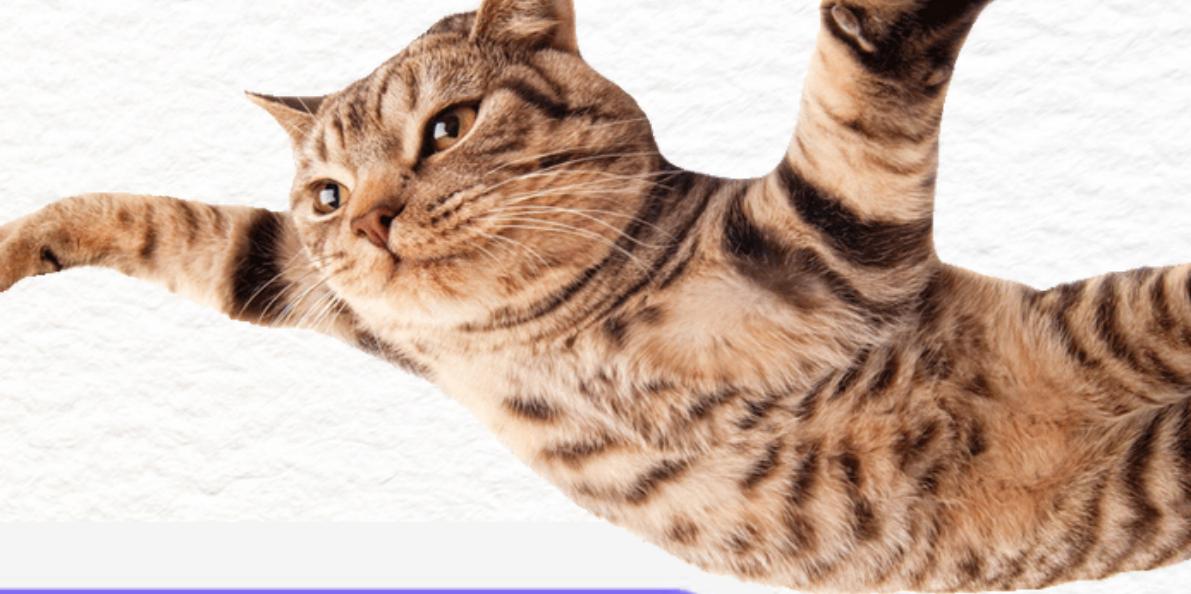
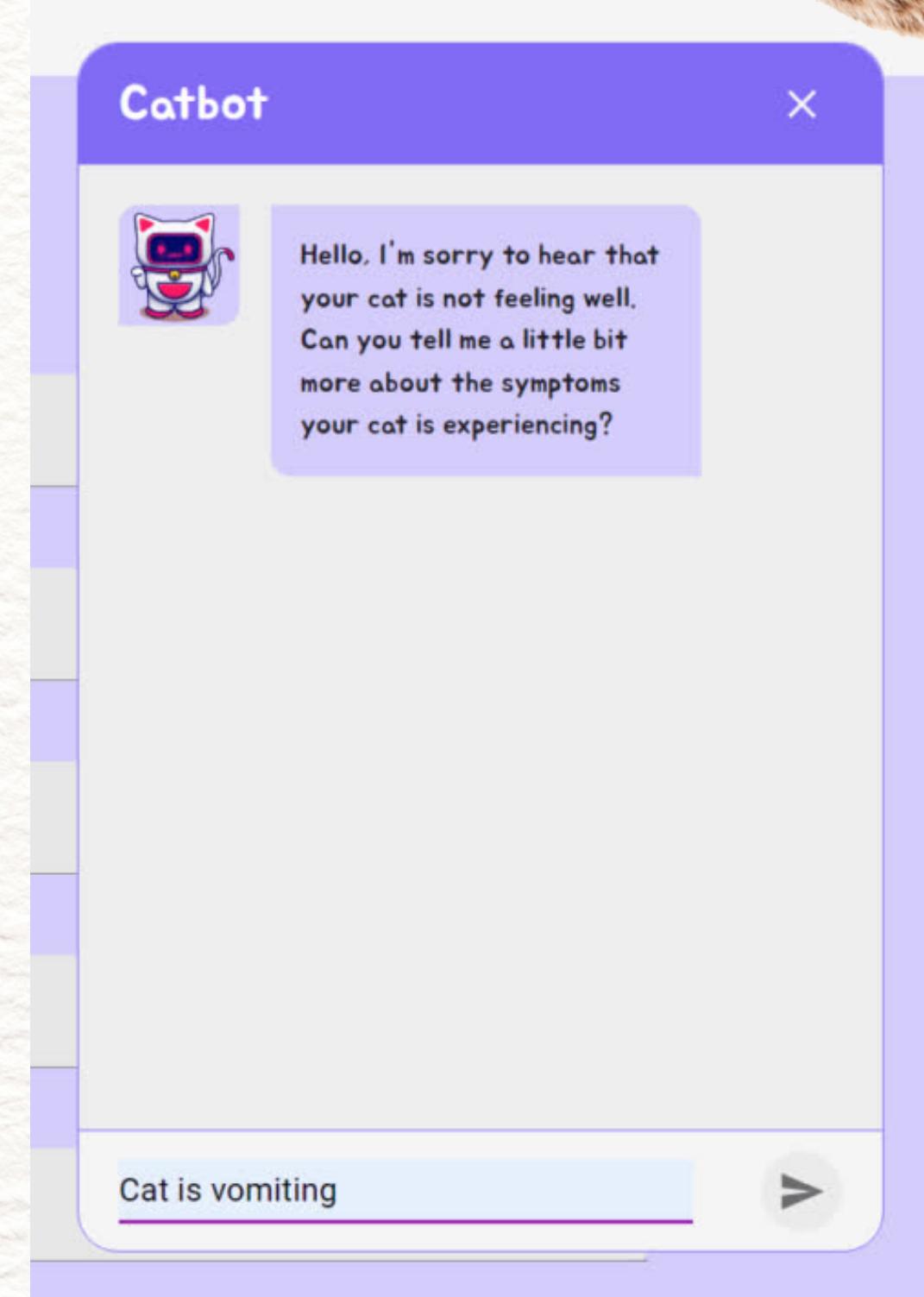
Join us for quick, convenient care for your cat

It is our commitment to ensure a professional and easy to use insight into your cat's health. If you want to join us in diagnosing your pet's problems before you see a vet, click the button below.

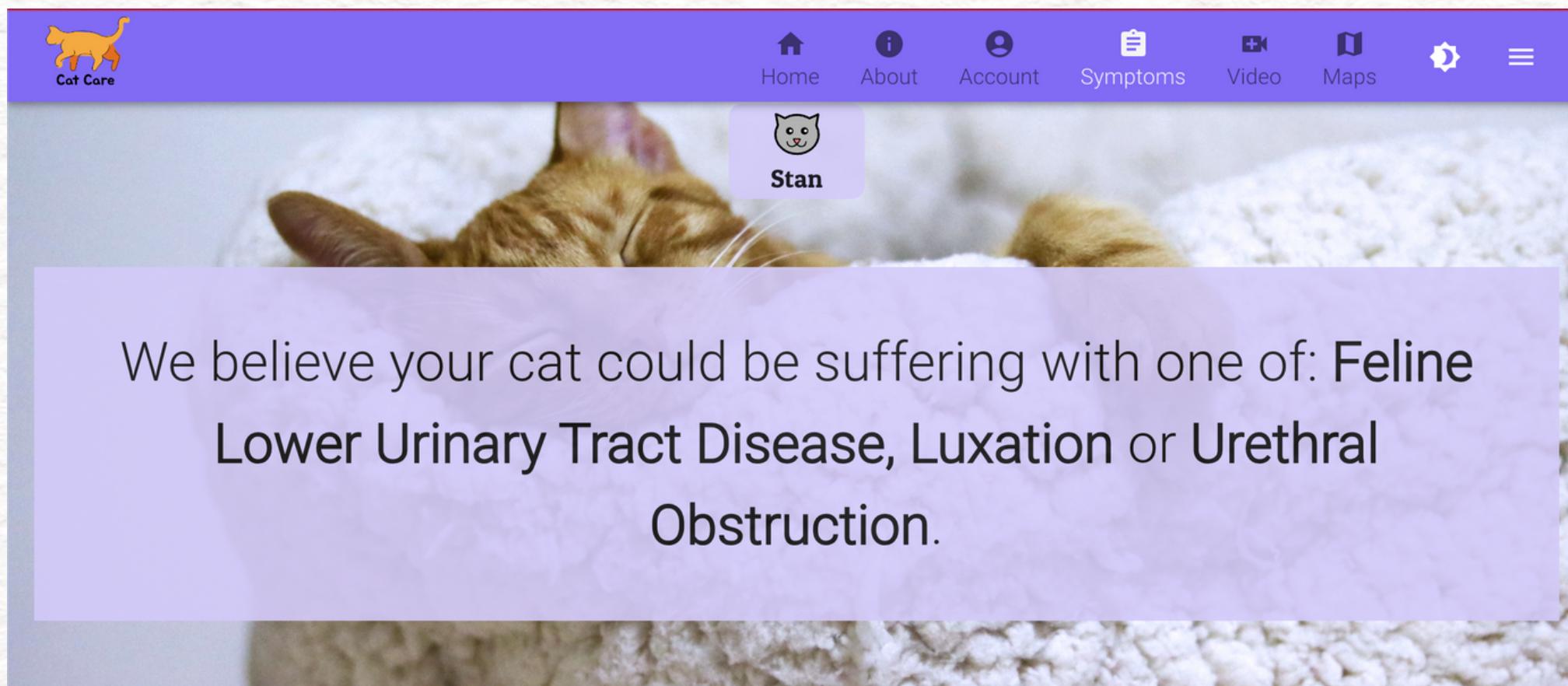
[Join Us](#)

Significant Code

```
22  const openAIFunc = async (value) => {
23    const bearer = "Bearer " + import.meta.env.VITE_OPENAI_KEY;
24    const options = {
25      method: "POST",
26      cors: "cors",
27      headers: {
28        Authorization: bearer,
29        "Content-Type": "application/json",
30      },
31      body: JSON.stringify({
32        model: "gpt-3.5-turbo-0301",
33        messages: [
34          {
35            role: "system",
36            content:
37              "You are a veterinarian whose job it is to cure my cat's illness",
38          },
39          {
40            role: "user",
41            content: value,
42          },
43        ],
44      }),
45    };
46
47    const response = await fetch(
48      "https://api.openai.com/v1/chat/completions",
49      options
50    );
51
```



```
27 def findDiagnosis(pet_id, questionsAnswered, answersUser):  
28     defaultVariableQuestions = getAllTrueDefaultVariablesIds()  
29     petDetails = getPetDetailsbyId(pet_id)  
30     answersTrueDefaultAnamnese = answerDefaultAnamnese(petDetails)  
31     BayesLibObj.setQuestionAnswer(defaultVariableQuestions, answersTrueDefaultAnamnese)  
32     answersRandomAnamnese = answerRandomAnamnese(answersUser)  
33     BayesLibObj.setQuestionAnswer(questionsAnswered, answersRandomAnamnese)  
34     print(BayesLibObj.as_dict())  
35     probabilities = BayesLibObj.Solve()  
36  
37     return probabilities
```



Q3: Is your cat experiencing hair loss or thinning fur?

No

Probably not

I don't know

Probably yes

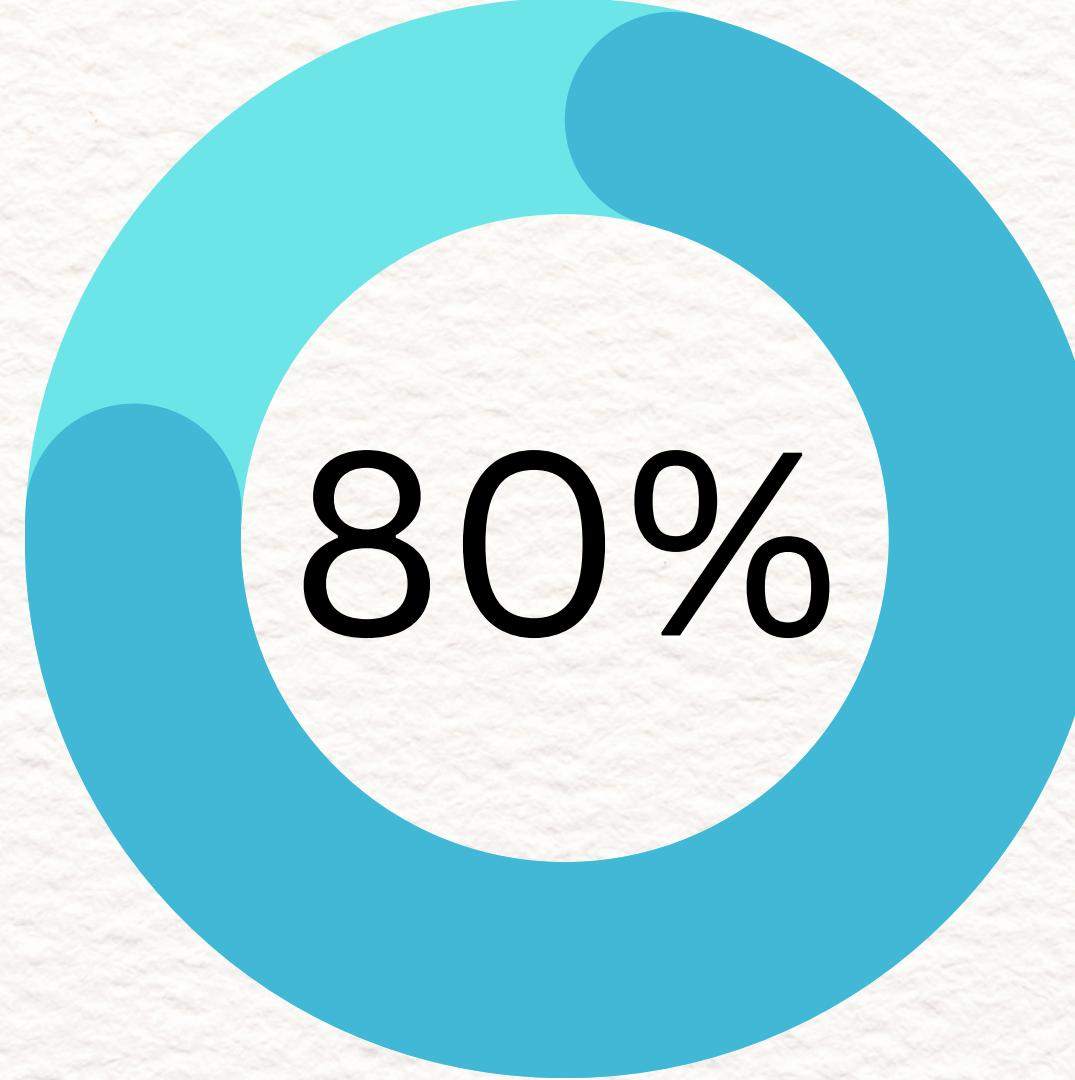
Yes



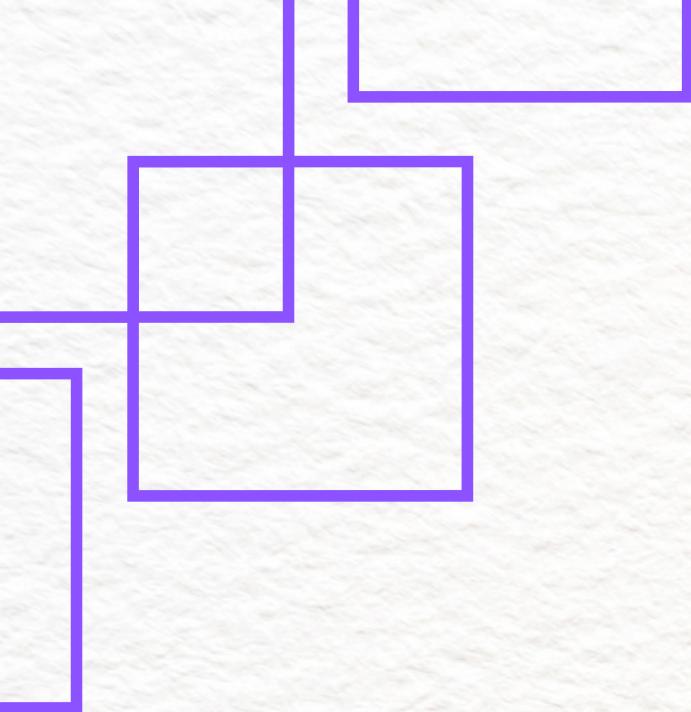
It is important that you seek the correct medical help as our calculations may not always be correct. You can book a video appointment in with one of our vets or see where local vets are in your area by clicking on

Testing

Coverage



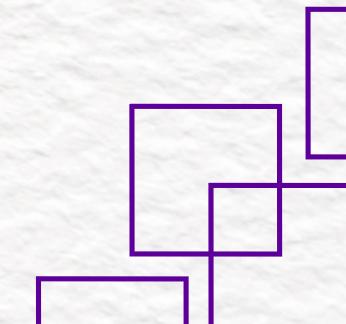
80%



Challenge



Considering how the frontend would interact with the backend to retrieve questions, submit answers, and receive diagnoses.



```
16     @diary.route("/diary", methods=["POST"])
17     def create_diary():
18         current_date = datetime.now().date()
19         data = request.json
20         pet_id = data['pet_id']
21         questions = data['questionsArray']
22         answers = data['answersArray']
23         result = findDiagnosis(pet_id, questions, answers)
24
25
26         listDiseases = []
27         for diagnosis in result:
28             id = int(diagnosis['disease_id'])
29             listDiseases.append(Diseases.query.get_or_404(id).as_dict()['name'])
30
31         try:
32             new_diary_entry = Diary(
33                 pet_id=pet_id,
34                 date=current_date,
35                 questions=questions,
36                 answers = answers,
37                 possibleDiagnosis= listDiseases
38                 #field=data["field"],
39             )
40             db.session.add(new_diary_entry)
41             db.session.commit()
42             return jsonify({"pet_id": new_diary_entry.pet_id, 'instance_id': new_diary_entry.id}), 201
43         except Exception as e:
44             print("ERROR!!!!!!!!!!", str(e))
```



Challenge



Structuring the database to enable training the algorithm using user feedback.



Solution



id	disease_id	diseasesVariables_id	no	probablyNot	iDontKnow	probablyYes	yes
1	1	1	0	0	0	0	1
2	1	2	0	0	0	0	1
3	1	3	0	0	0	0	1
4	1	4	0	0	0	0	1



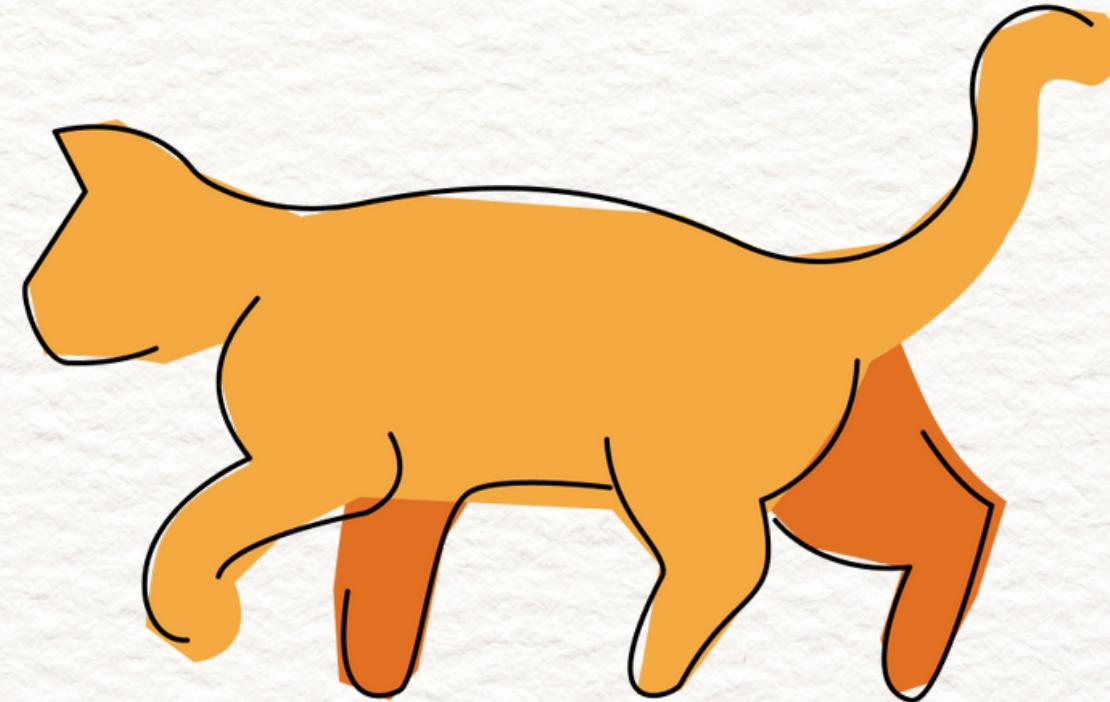
Future Features

- Admin page
- Better question selection
- Insert more diseases
- Suggest treatments or supplements



Thank you for listening!

Any questions?



Cat Care