# DJANGO- REACT PROJECT

**A drawing of a cat

Description automatically generated with low confidence**

**Site Name: *Wyld4Paws***

SheCodes Project – Animal Shelter / GoFund Website

Username: Admin

Password : Shecodes

# Overview

Life with a pet is a wonderful thing. Cats, dogs, and many other small mammals and creatures become our family members, friends and companions. Sadly, not all animals enjoy the luxury of having a loving home.

Every animal is an individual and has its own needs and just like humans also have their own personalities and medical needs. Unfortunately, not all humans know how to cope or can afford some of these needs and thousands of animals go into shelters needing help, support and a new forever home.

I wanted to build a shelter site where animals can be fostered, adopted or assisted with medical treatment! A subject close to my heart and whilst this is only for a project I wanted to try and make it as realistic as possible.

I’m just starting out on my coding journey, so I’ll start small and hopefully experiment and build over time.

# Part A submission

## MVP Plan

The Animal Shelter will eventually cater for many different types of animals, Onsite there is a kitten Nursery, Dog Runs, Bird Area, Horse paddocks.& wildlife enclosure.

There is a veterinary surgery onsite for the animals and where we can we assist families in need of financial support to offer a onetime emergency veterinary service or the option to fund raise.

Using Django-Rest and insomnia

Initially

1. Users can View the Animals
2. Create a User Account and Profile
3. Adopt an animal
4. Foster an Animal
5. Pledge to an Animal for Medical Care. (Support)

### Additional to MVP

1. Track own Pledges and animals being fostered or own adoptions
2. Comment with Donation or keep anonymous
3. Get Active thankyou messages for donations and notification email with tax invoice on submission of users donation
4. Pledge to individual fund or General Shelter support fund
5. They can donate gifts that will be used as a prize pool in a quarterly Raffle using their own time an skills ie, Paintings/drawing, Made Jewellery, Lawn Mowing, gift pack, vouchers or food and blankets to the shelter.
6. They can volunteer to work in the shelter, Reception, Dog Walking, Stables, Nursery etc.
7. Add own stories with support requests to assist with medical fees of adopted or fostered pets.

## **Colour Scheme:**

Silver / Teal / Navy and slash of Lime

## **Heading and body font(s)**

### Headings = Tangerine

### Body: Playfair Display

@import url('https://fonts.googleapis.com/css2?family=**Playfair+Display**&family=**Tangerine**&display=swap');

## **Links**

Link to DRF Github Repo

Link to DRF Deplayed Project

## **Stack**

### BackEnd

* Python
* Django
* Django Rest Framework (DRF)

### FrontEnd

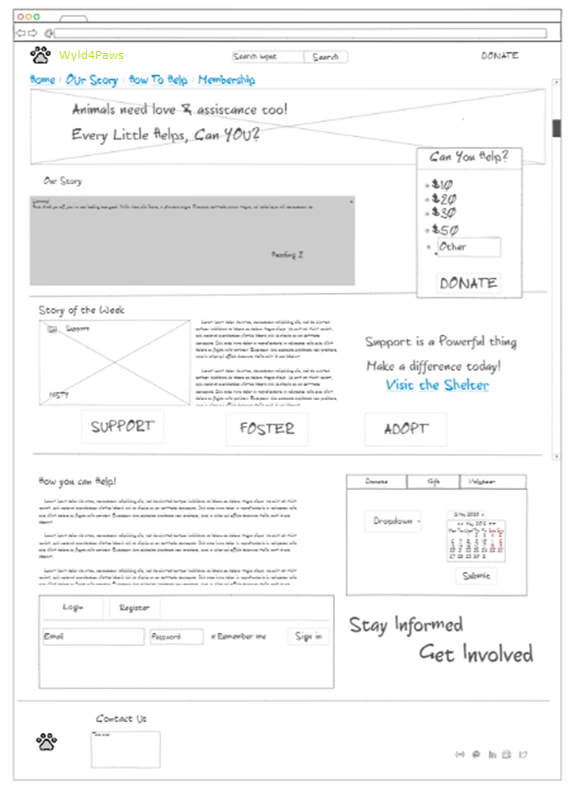
* Javascript
* React

### Tools

* VSCode
* Insomnia
* Lucid Charts

## **Wireframe: original ideas**

### LANDING PAGE



## Donation Page

Graphical user interface

Description automatically generated

How You Can Help Page

Diagram

Description automatically generated

Graphical user interface

Description automatically generatedSupport / Foster or Adopt Page

User Profile

A picture containing graphical user interface

Description automatically generated

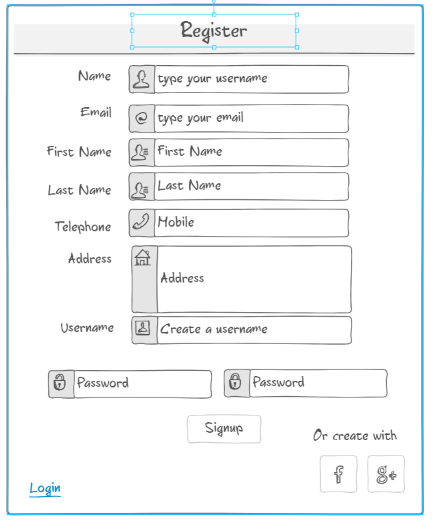
Graphical user interface, Word

Description automatically generatedCreate / Edit Story

Graphical user interface, text, application

Description automatically generatedAnimal Story

Window Popup Forms

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated

**Site Includes -** A database schema:

|  |  |  |  |
| --- | --- | --- | --- |
| **Schema** | **Animals at the shelter (Project)** | **Donations (Pledge)** | **User Accounts** |
| **1** | **Id Number:** | **One off** | **Username** |
| **2** | **Name:** | **Monthly** | **First Name** |
| **3** | **Type:** cat / dog / horse | **Gift** | **Last Name** |
| **4** | **Gender:**  M/F | **volunteer** | **email** |
| **5** | **Age:** | **amount** | **Mobile** |
| **6** | **Service Required: (Title)** Foster/Adopt/Donate | **anonymous** | **Address** |
| **7** | **Goal:** Amount | **Project** | **Password** |
| **8** | **Difficulty:** 1-5 | **Supporting** | **Image** |
| **9** | **Image** |  | **bio** |
| **10** | **Description** |  |  |
| **11** | **Date Created** |  |  |
| **12** | **Author / Owner** |  |  |
| **13** |  |  |  |

**API Specification:**

**Example API Spec -** (Hayley’s API SPEC to get started)

**User 1**

Username: cats

Password: catscats

**User 2**

Username: dogs

Password: dogsdogs

**Authentication**

Use /api-token-auth/ to get a token, then use Bearer auth with Token <token> on each endpoint requiring authorization.

**Endpoints**

|  |  |  |  |
| --- | --- | --- | --- |
| **Endpoint** | **Method** | **Body** | **Authorization/Authentication** |
| /projects/ | GET | List of Project objects. | None required |
| /projects/ | POST | Project object (without id or date\_created). | Need to be logged in. |
| /projects/?limit=3 | GET | Returns the number of projects specified by the limit query parameter. | None required. |
| /projects/?is\_open=True | GET | Returns the projects with the given status for is\_open as provided by is\_open query parameter. | None required. |
| /projects/?order\_by=date\_created | GET | Returns the projects in order of date created. | None required. |
| /projects/?order\_by=num\_pledges | GET | Returns the projects in order of the number of pledges. | None required. |
| /projects/?order\_by=recent\_pledges | GET | Returns the projects in order of most recently pledged to. | None required. |
| /projects/ | POST | Project object (without id). | Need to be logged in. |
| /projects/1/ | GET | Project with pledges field with list of Pledge objects. | None required |
| /projects/1/ | PUT | Project object (without id). | Need to be logged in.  Must be project owner. |
| /pledges/ | GET | List of Pledge objects. | None required |
| /pledges/?order\_by=date\_created | GET | List of Pledge objects in order of date created. | None required. |
| /pledges/ | POST | Pledge object (without id or date\_created). | Need to be logged in. |
| /api-token-auth/ | POST | User object. |  |

**Objects**

|  |  |
| --- | --- |
| Project | {      "id": 1,      "title": "",      "description": "",      "goal": 150,      "image": "",      "is\_open": false,      "date\_created": "",      "owner": "",      "total\_pledged": 246  } |
| Pledge | {      "id": 1,      "amount": 25,      "comment": "",      "anonymous": true,      "date\_created": "",      "supporter": "",      "project\_id": 1  } |
| User | {  "username": "admin",  "password": "password"  } |

**Objects**

|  |  |
| --- | --- |
| Project | {      "id": 1,      "title": "",      "description": "",      "goal": 150,      "image": "",      "is\_open": false,      "date\_created": "",      "owner": "",  “slug”  } |
| Pledge | {      "id": 1,      "amount": 25,      "comment": "",      "anonymous": true,      "supporter": "",      "project\_id": 1  “slug”  } |
| User | {      "id": 1,  user(models.Model):  title=models.CharField(max\_length=200)  link = models.CharField(max\_length=200)  subscriber = models.PositiveIntegerField()  address = models.CharField(max\_length=100)  email = models.CharField(max\_length=100)  description = models.TextField(max\_length=2000)  category = models.CharField(max\_length=100)  mobile = models.CharField(max\_length=100)  “slug”  ‘image’  } |
| Category | {  “id”:  “description’  “name”  “slug” |
|  |  |