**Pet.me**

**The problem and How it Solves it?**

Nowadays there are lots of stray animals that need a safe house or a safe environment to live in so many local organizations save this animals and offer them for adoption we try to facilitate this adoption process and find an easy way to connect the adopters with the people offering adoption.

Another problem is that people already have pets at home and they are moving to another country or are traveling for a long period of time and they want to find a caring home for their animals as soon as possible. or maybe even they are traveling for a short period of time and they need a temporary hosting house for the pets either for free or for or an affordable amount of money.

Our website is going to be like a craigslist but only for pets and pet owners. We do not aim for profit or any other gains.

**The Project schema:**

Multiple pages or Single Page App?

How much is processed in the back end?

Where to get dummy data?

**Project Requirements:**

User:

Properties

- First name

- Last name

- Email

- Gender

- Password

- Confirm password

- Mobile phone [validated against Egyptian phone numbers]

- Profile Picture

- Activation Email after registration

- Once the user register he should receive an email with the activation link. The user shouldn’t be able to login without activation. The activation link should expire after 24 hours.

- The user should be able to login after activation using his email and password

- Bonus: Allow users to login with facebook account

- Forgot Password (Bonus)

- The user should have an option to reset his password if he forgot it to receive a password reset link to his email

- User Profile, The user can view his profile which:

- He can view his profile

- He can view his pets

- He can view his posts

- He can edit all his data except for the email

- He can have extra optional info other than the info he added while registration (Birthdate, facebook profile, country)

- User can delete his account (Note that there must be a confirmation message before deleting)

- Bonus: User must enter his password to delete his account

====================

2 - Pets:

- The user can create a pet object which contains:

- Name

- Brief

- Gender

- Type: Dog/Cat

- Species

- Color

- Multiple pictures

- Set start/end time for the adoption/owning!

- Owner

Model :

Adoption

- User

- Pet

- Start at

- End at

======================

App: Social

- User can post about any of his pets

- A post is made automatically when adopting a new pet or offering a pet for adoption

- Comments

- Bonus: Comments can have replies

- Users can report inappropriate posts

- Users can report inappropriate comments

- post creator can delete the project if the donations are less than 25% of the target

- Project page should show the overall average rating of the project - Project page should show the project pictures in a slider

- Project page should show 4 other similar projects based on project tags

Model: Offer adoption

=======================================

App: Homepage should contains the following:

- A slider to show the highest five rated running projects to encourage users to donate

- List of the latest 5 projects

- List of latest 5 featured projects (which are selected by the admin)

- A list of the categories. User can open each category to view its projects

- Search bar that enables users to search projects by title or tag

=================

App: Social

4 - Comment

- user

- post

- content

5 - Report

- user

- user / comment / post

- reason

- status (seen / new / solved)

# Adoptor persona

## Demographics

Age from 18 years old to 50 years old

Gender both

Location all over Egypt but mainly cities not rural areas

 education a high school  or diploma or higher

## Background

Occupation housewives students mostly

 families status all

 goals and needs

Browse and pick from a number of pets with a certain characteristics

Get to know the pet and its background story

Trust that the pet is is not wild and is safe to adopt

See pictures and enough information about the pet and the owner

 challenges and pain points

 it's always a challenging to find a pet that is worthy of adopting and a safe at the same time and also you  you don't always have many pets to pick from  while choosing

Facebook groups that do the same purpose does not show you the history of the owner and the history of the pet and how they are treated

 how our product and service can help

 our products will facility the adoption process by building a profile for the pet and building a profile for the owner and having history for both and when the owner is offering his bed for adoption he will have to post photos and information about the pet thus the adopter will have many options and many pets to choose from

# the owner persona

 Demographics

 age from 18 to 50

 gender both

 location same as adopter

 education same as adopter

 Background

  occupation unemployed a frequent traveler a housewife

goals and needs

offer his pet for adoption to someone who is trustworthy and is not going to mistreat his pet or resell it - validity period for the ad

Can view the adopter profile and see is photo to ensure that he is a trustworthy person

Can follow the adopter after the adoption process and follow his Pet and see what he is up to

 challenges and pain points

 he needs a quick solution to give away his pet for adoption or hosting before traveling and such a short notice - priotirize by tags

 not being able to communicate with adopter and keep in touch with his pet

 how our product service will help

 it will keep him in contact with the adopters and adopters can still post photos and post feeds about the pet and what their status is up to - and chatting

Dedicated community of adopters and pet owners the adoption process can take place reliably and more quickly and with a report form we can ensure that there are no users with bad intentions - with reviews

Tips and advices for first-time adopters

1 - Frontend

Figma Design

https://www.figma.com/file/A7VAAWrlTSpdAobVObrbcX/Pet.me?type=design&node-id=16%3A69&mode=design&t=UQhpaqSVdWArzLLO-1

1 – Build user profile and basic buttons (login page, signup, forgot password page, success [email confirmation needed])

2 - pet page, profile page, )

3 – About us page - Explore page

- Add Email activation page and change the url in settings.py of DRF

- Add Social login redirection page and change urls in settings.py and in Google, FB, Github settings

- Add Password Reset Page and change url in settings.py

2- Backend

**The required APIs endpoints:**

|  |  |  |  |
| --- | --- | --- | --- |
| Endpoint | Method | Function | Permissions |
| **1 – Accounts App** | | | |
| /accounts/users/ | POST  body: [username – email - password] | Register a new accounts  The user gets an email containing a link for verification | - |
| GET | Gets all registered users on the system | Log in |
| /accounts/users/activation/ | POST  [uid – token] sent by email  frontend page should receive the link from email and send info by request | This is used to get the information sent by activation email – response 204 means that the user is ok to log in | - |
| /accounts/users/resend\_activation/ | POST [ email ]  Resend activation email | Resend activation email | - |
| /accounts/users/me/ | Get | Gets current user info | Log in |
| Put [Required fields] | Writes over old info |
| Patch [fields to update] | Updates info |
| DELETE [password] | Deletes user |
| /accounts/users/set\_email/ | POST [new\_email - current\_password] | Changes username email | Log in |
| /accounts/users/set\_password/ | POST [ new\_password,  current\_password] | Changes password | Log in |
| /accounts/users/reset\_password/ | POST [ email ] | Sends email with password reset link | - |
| /accounts/users/reset\_password\_confirm/ | POST [ uid – token – new\_password ]  This info is collected from reset password page from the email the user gets | Confirms the password change | - |
| /accounts /jwt/create/ | POST [ email – password ] | Logs in the user and response contains (access and refresh) tokens  The access token is user in header of each request like this:  Authentication: JWT “token”  Refresh token is used to regenerate access token when it expires | - |
| /accounts/jwt/refresh/ | POST [refresh] | Send the refresh token to get a new access token | - |
| /accounts/jwt/verify/ | POST [token] | Responds with 200 or 401 | - |
| accounts/o/<provider>/ | GET [redirect\_uri]  redirect\_uri is the same as the page where you would receive his social tokens |  | - |
| POST [code – state]  You should capture this from the redirect\_uri page  send as application/x-www-form-urlencoded not JSON | Logs user in and responds with access and refresh tokens | - |
| /accounts/<user\_id>/adoptions | GET | Get pet history of this users | - |
| **2 – Pets App** | | | |
| /pets/ | GET | Get list of all pets | - |
| POST [ name – brief – gender – pet\_type – species – color – birthdate – photos ] | Create a new pet | Log in |
| /pets/<pet\_id> | GET | Get specific pet details | - |
| POST | Overwrite old data | Owner |
| PATCH | Update and keep old data |
| DELETE | Delete pet |
| /pets/<pet\_id>/adoptions | GET | Get history of pet | - |
| /pets/<pet\_id>/offer | POST [ description ] | Offers this pet for adoption | Owner |
| **3 – Offers App** | | | |
| /offers/ | GET | Gets all offers | - |
| /offers/<offer\_id> | GET | A specific Offer | - |
| DELETE | Deletes the offer | Owner |
| /offers/requests/ | GET | Gets all requests on current user’s offers | Log in |
| /offers/<offer\_id>/requests/ | GET | Get all requests on this offer | Offer Owner |
| POST [ message ] | Send an adopt request | Log in |
| /offers/request/<request\_id> | DELETE | Deletes a request | Owner |
| /offers/request/<request\_id>/accept | GET | Accepts the request  This removes the pet from the old owner and adds it to the new owner, and it deletes the offer afterwards | Offer Owner |
| /offers/request/<request\_id>/reject | GET | Deletes the request | Offer Owner |
| **4 – Chat App** | | | |
| /chats/ | GET | Gets all messages associated with current user | Log in |
| /chats/<message\_id> | DELETE | Deletes specific message | Owner |
| /chats/user/<receiver\_id> | POST [ content ] | Creates a new message to this user from logged in user | Log in |
| **5 – Social App** | | | |
| posts/ | GET | Get all posts | - |
| POST [content - photos] | Create post | Log in |
| posts/ <post\_id> | GET |  | - |
| POST |  | Owner |
| PATCH |  |
| DELETE |  |
| posts/ reports/ | GET | Get all reports | Admin |
| posts/ reports/<report\_id> | GET | Get a report | Admin |
| DELETE | Delete a report |
| posts/ <post\_id>/reports/ | GET | Reports on this post (including those reporting comments) | Admin |
| POST [reason] | Create a report | Log in |
| posts/ comment/<comment\_id>/report/ | POST [reason] | Create a report | Log in |
| posts/ <post\_id>/comments/ | POST [content] | Create a comment on the post | Log in |
| posts/ comment/<comment\_id>/ | DELETE | Delete Comment | Owner |
| posts/ comment/<comment\_id>/replies/ | POST [content] | Create a Reply on the comment | Log in |
| posts/ comment/reply/<reply\_id> | DELETE | Delete Reply | Owner |

**How authentication works:**

See this https://djoser.readthedocs.io/en/latest/base\_endpoints.html

and this https://djoser.readthedocs.io/en/latest/jwt\_endpoints.html

1 – Send post data to this url to signup, en email is sent

http://127.0.0.1:8000/accounts/auth/users/

2 – After email verification, send login data to this url to login, you will receive a token [ Refresh + Access ]

[http://127.0.0.1:8000/accounts](http://127.0.0.1:8000/accounts/auth/users/)/auth/jwt/create/

3 – use token in Header for any request

'Authorization: Token b704c9fc3655635646356ac2950269f352ea1139'

4 - /auth/users/me

**For log in using social media**

**1 - send get request to this url:**

# change redirect url in google api settings and github oauth and facebook

with this get query ==> redirect\_uri=http://localhost:8000/accounts/auth/social/complete/github/

* http://127.0.0.1:8000/accounts/o/google-oauth2/
* [http://127.0.0.1:8000/accounts/o/facebook/](http://127.0.0.1:8000/accounts/auth/o/google-oauth2/)
* [http://127.0.0.1:8000/accounts/o/github/](http://127.0.0.1:8000/accounts/auth/o/google-oauth2/)

complete url::

[http://127.0.0.1:8000/accounts/o/google-oauth2/?redirect\_uri=http%3A%2F](http://127.0.0.1:8000/accounts/auth/o/google-oauth2/?redirect_uri=http%3A%2F) %2Flocalhost%3A8000%2Faccounts%2Fauth%2Fsocial%2Fcomplete%2Fgithub%2F

**You will get a response with a redirect url, redirect user to this url**

After the user authenticates the external application, it will redirect him to a page, where you should capture the {code and status} parameters in the url and make a post request →

to, issue a POST request to the endpoint /accounts/auth/o/{{ provider }}/ with the code and state arguments. You should use application/x-www-form-urlencoded not JSON. The user should be now authenticated in your application. It will respond with access and refresh tokens.