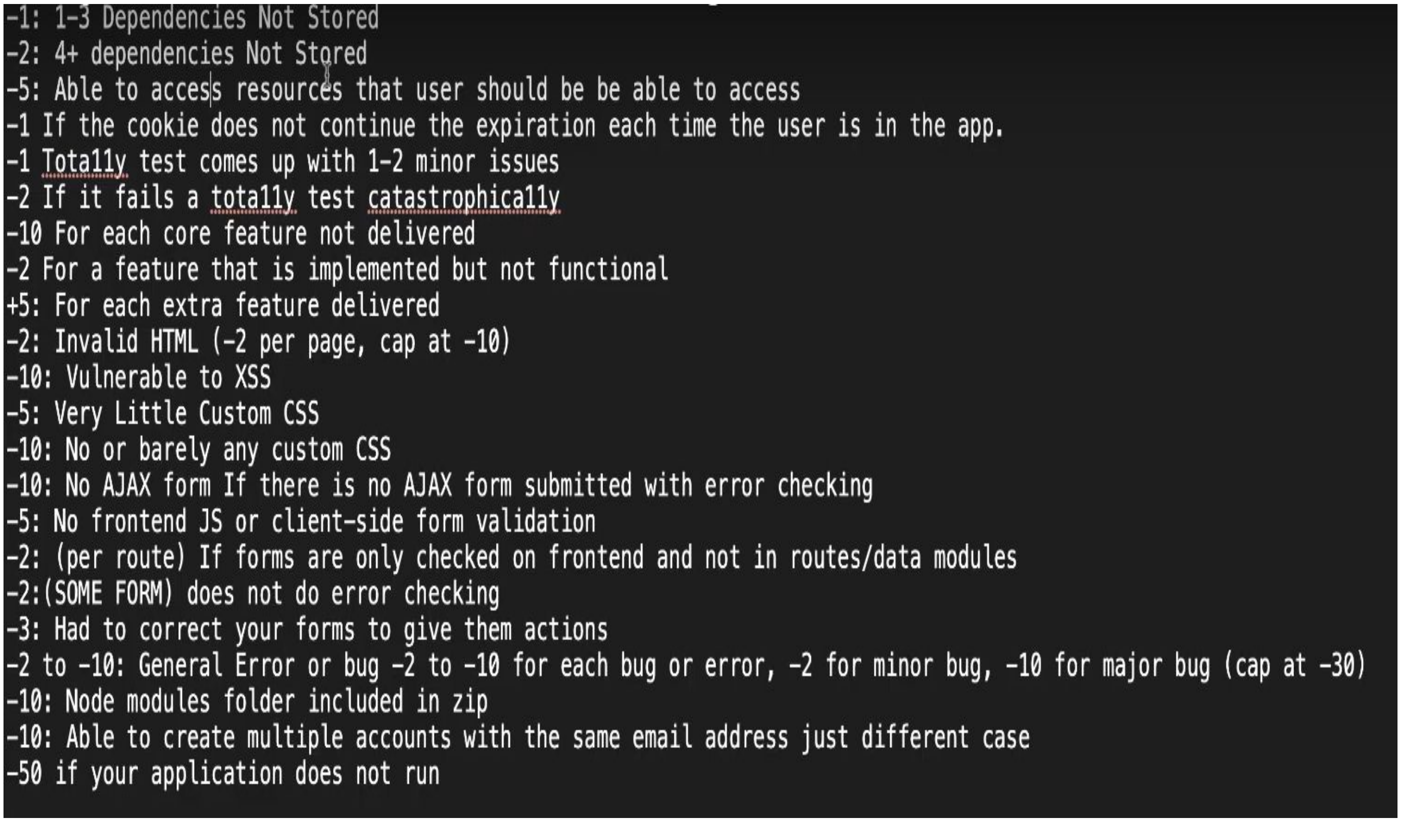
CS546 Project

NJ Dog Adoption Website

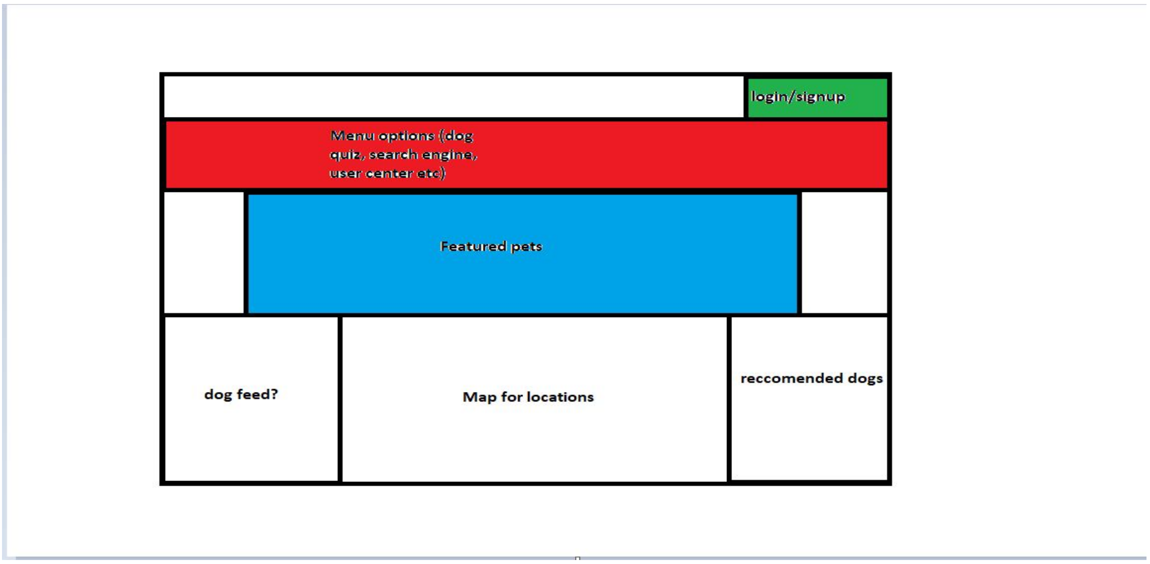
[GitHub]

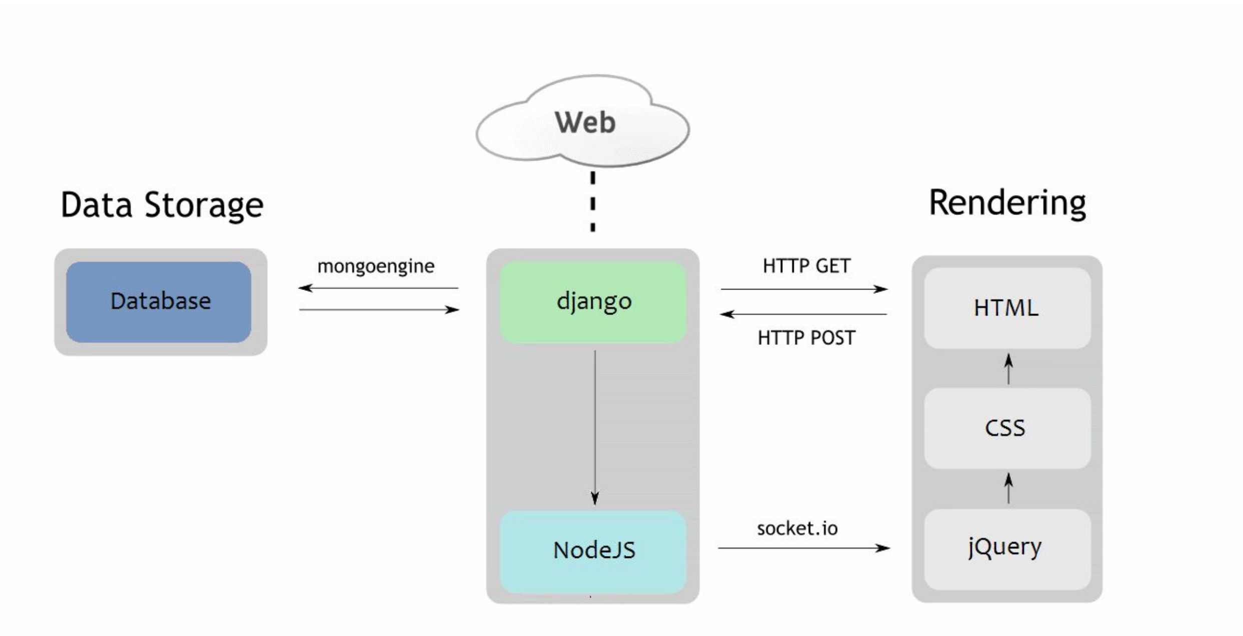
TEAM

- Zhixin Jiang
- Mohammed Qazi
- Pratim Patel
- Yuting Shao



```
-1: 1-3 Dependencies Not Stored
-2: 4+ dependencies Not Stored
-5: Able to access resources that user should be be able to access
-1 If the cookie does not continue the expiration each time the user is in the app.
-1 Tota11y test comes up with 1-2 minor issues
-2 If it fails a tota11y test catastrophica11y
-10 For each core feature not delivered
-2 For a feature that is implemented but not functional
+5: For each extra feature delivered
-2: Invalid HTML (-2 per page, cap at -10)
-10: Vulnerable to XSS
-5: Very Little Custom CSS
-10: No or barely any custom CSS
-10: No AJAX form If there is no AJAX form submitted with error checking
-5: No frontend JS or client-side form validation
-2: (per route) If forms are only checked on frontend and not in routes/data modules
-2:(SOME FORM) does not do error checking
-3: Had to correct your forms to give them actions
-2 to -10: General Error or bug -2 to -10 for each bug or error, -2 for minor bug, -10 for major bug (cap at -30)
-10: Node modules folder included in zip
-10: Able to create multiple accounts with the same email address just different case
-50 if your application does not run
```


login/signup
Menu options (dog quiz, search engine, user center etc)
Featured pets
dog feed?
Map for locations
reccomended dogs


Web
Data Storage
Rendering
Database
mongoengine
django
HTTP GET
HTTP POST
HTML
CSS
NodeJS
socket.io
jQuery

This website will be a dog adoption website, which will highlight local dogs that are being put up for adoption. It will emphasize accessibility and will make the initial steps of finding a pet to adopt easier for users. Users will be able to create accounts, and look through the different options of dogs on the website.

Core features:

- Main page

The main page after login will show a variety of different dogs that are able to be adopted. The user will be able to see all of the different features of the website from this page, and will also be able to pick an animal based on preferences. And in the main page, users can see their recent five views.

- Map

Allows the user to see a map of their location and the nearby places that they can adopt pets at.

- Dog Quiz

Quiz that allows potential pet owners to figure out what the perfect breed for their wants and needs is.

- View detailed information

Allow users to view the details of the dog, like picture, breed, age and location.

- Dog Search Engine

Allow users to search for the dog by breed, age and location.

- User Center

After the users log in, they can gain access to the User Center Page. This page will include users’ basic personal information and Favorites List.

- Admin System

Admins can login in the Admin System to manage all Foreground Data which includes dog posters data and register users data. Admins can do operations like add, delete, remove, update to the database. This system can be used when adding new dog posters and allows dog posters to update the description of the dog or delete it after the dog was adopted.

- Dog Feeding Guide

This Page includes two modules, one is Dog Problems, it provides solutions to common dog problems, such as “What To Do IF Your Dog Training Fails.” Another module is Dog Nutrition, it provides some recommendations about Dogs’ Food.

Extra Features:

- Feed aspect, where users can see a feed of the dogs that have been posted, kind of like a facebook feed.
- Similar dogs recommend, which will show several similar dogs based on the same key words when users see the detail page of one dog.
- Website Feed comment, users will be allowed to send the comments to the website owners. And the comments could be checked in the Admin System.

Technologies :

Node Js, MongoDB, HTML, CSS, Express,

DATABASE

The user collection will store all users registered in Dog Adoption Website, Users will be able to login via their username and password, update their details including email, gender, city, state, age, and users also can add favorite dogs into their favorite list.

```
{
"user_id": "6a7117a2-c1d2-4f8c-b27a-5b1d4b5b7233",
"user_name": "YutingShao”
"user_password": "Shao1010159322”
"user_email": "yshao13@stevens.edu",
"user_gender": "Femal",
"user_city": "Hoboken",
"user_state": "NJ",
"user_age": "25",
“favouriteList”: [7b7997a2-c0d2-4f8c-b27a-6a1d4b5b6310,
                  7b7997a2-c0d2-4f8c-b27a-6a1d4b5b6311,
                  7b7997a2-c0d2-4f8c-b27a-6a1d4b5b6312]
}
```

Name	**Type**	**Description**
user_id	String	A randomly generated unique identifier to represent the user.
user_name	String	A string that stores username, used to login.
user_password	String	A string stores the password of the user, used to login.
user_email	String	A string that stores the user’s email.
user_gender	String	A string that stores the gender of the user.
user_city & user_state	String	The address where the user lives in.
user_age	Int	An int that stores the user’s age.
favouriteList	Array	An array that stores the dog’s id which is added to favorite by users.

Dogs

The Dogs collection will store all the information created by advertiser or admin. It also stores comments that are related to the dog in the comments sub-document.

```
{     “dog_id”: “7b7997a2-c0d2-4f8c-b27a-6a1d4b5b6310”,

“dog_breed” : “Dobermann”,

“dog_name” : “Lokey”,

“dog_age” : “2 months”,

"dog_address": "8th Washington street Hoboken, NJ",

 “dog_description” : “He’s very active and affectionate . He loves a lot of attention and will let
you know by cuddling up to you. I prefer a loving single or couple who has time to give him the
attention he needs and deserves.”,

“dog_vaccine” : “[“rabies”, “distemper”, “parvo”]”,

“dog_picture”:
“https://upload.wikimedia.org/wikipedia/commons/c/c0/0Doberman-40172501920.jpg ",

“dog_postdate”: “2020-03-20”

“dog_color”: “Brown”,

“dog_size”: “(when grown) Med. 26-60 lbs (12-27 kg)”,

“dog_gender”: “Male”,

 "comments":[{

“comment_id”: “6a7117a2-c1d2-4f8c-b27a-5b1d4b5b7233”,

“user_name”: “Lokey”,

“comment”: “Cute doggie!”}]

}
```

Name	**Type**	**Description**
dog__id	String	Stores the dog id.
dog_breed	String	The breed of the dogs such as Dobermann, Pug, Boxer and etc.
dog_postdate	date	Store the date that the dogs advertisement created.
dog_name	String	Stores the name.
dog_age	int	Stores the age.
dog_address	String	Store the location of the dog.
dog_description	String	Stores the story of the dog.
dog_vaccine	String array	The type of vaccine the dog has injected.
dog_color	String	Stores the color.
dog_size	String	The size of the dog.
dog_gender	String	Stores the gender
comments	Array	Stores the text , username and id.

Comments

Comments is a subdocument of the dogs collection.

```
"comments":[{
“comment_id”: “7c1995a2-c1d2-4f8d-b11a-1d2c4b5b6133”,
“user_name”: “Lokey”,
“comment”: “Cute doggie!”
}]
```

Name	**Type**	**Description**
comment_id	string	A string represents the comment ID.
user_name	string	A string represents the name of the person who posted the comment.
comment	string	The comment text.

Dog Advertiser

This collection will store all of the information about the people who are advertising the dogs that can be adopted.

```
{
    "advertiser_id": "7b7997a2-c0d2-4f8c-b27a-6a1d4b5b6310",
    "advertiser_firstName": "Andrew”
    “advertiser_lastName”: “Carnegie",
    "advertiser_email": "acarnegie@gmail.com",
    “advertiser_cellphone”: “210-256373”,
    "advertiser_address": "8th Washington street Hoboken, NJ",
    "advertiser_password": "stevens24",
    “dogList”: [9c7997a2-c0d2-4f8c-b27a-6a1d4b5b6321,
                9c7997a2-c0d2-4f8c-b27a-6a1d4b5b6346]
}
```

Name	**Type**	**Description**
advertiser__id	String	A globally unique identifier to represent the user.
advertiser_firstname	String	A string that stores the user first name.
advertiser_lastname	String	A string that stores the user last name which they entered.
advertiser_email	String	The email information of the advertiser.
advertiser_userPassword	String	The password when users log in.
advertiser_cellphone	String	Stores the phone number of the advertiser.
advertiser_address	String	Stores the location of the adopter.
advertiser_dogList	Array	An array that stores dogs’ id which is posted by the user.

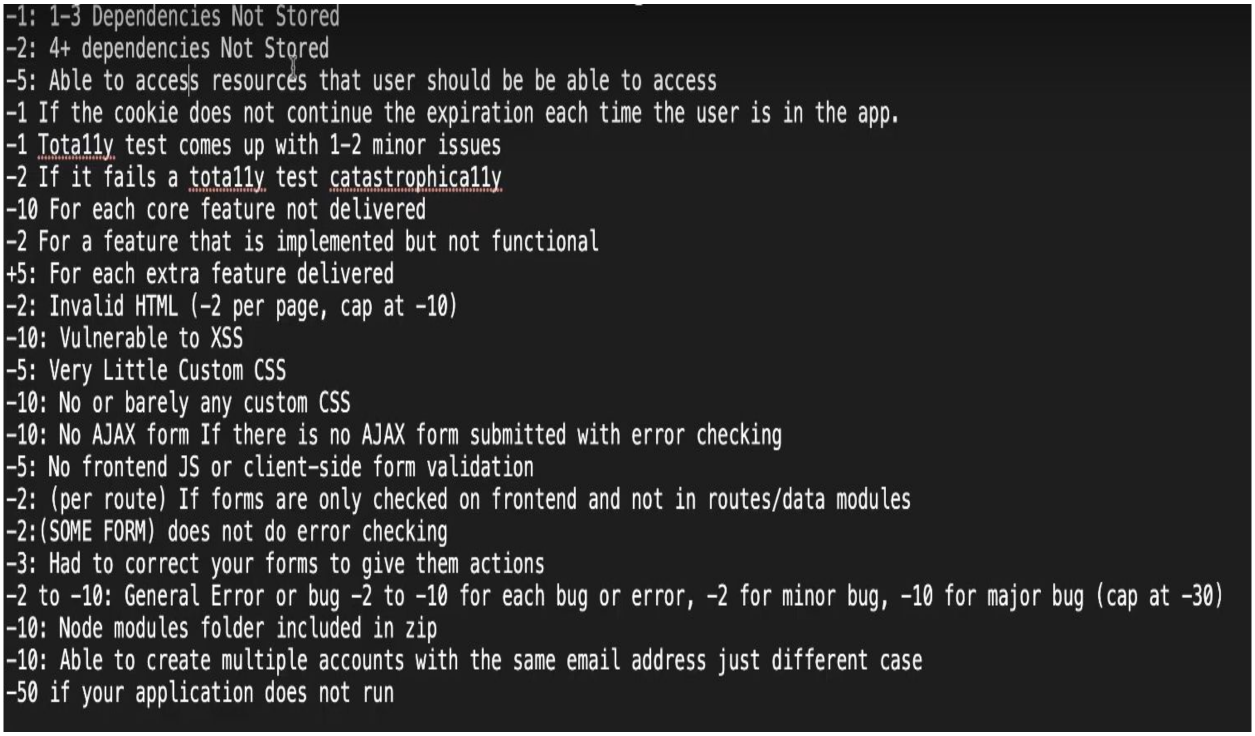
Admin User

The admin User collection will store the admin account information. Admin can login backstage database management interface and then do database management. It will be able to login via default username and password.

```
{
"admin_name": "Admin”,
"admin_password": "000000”
}
```

Name	**Type**	**Description**
admin_name	String	A string that stores admin name, used to login and gain access to backstage database management interface.
admin_password	String	A string stores the admin password, used to login and gain access to backstage database management interface.

Project Directory



```
-1: 1-3 Dependencies Not Stored
-2: 4+ dependencies Not Stored
-5: Able to access resources that user should be be able to access
-1 If the cookie does not continue the expiration each time the user is in the app.
-1 Tota11y test comes up with 1-2 minor issues
-2 If it fails a tota11y test catastrophica11y
-10 For each core feature not delivered
-2 For a feature that is implemented but not functional
+5: For each extra feature delivered
-2: Invalid HTML (-2 per page, cap at -10)
-10: Vulnerable to XSS
-5: Very Little Custom CSS
-10: No or barely any custom CSS
-10: No AJAX form If there is no AJAX form submitted with error checking
-5: No frontend JS or client-side form validation
-2: (per route) If forms are only checked on frontend and not in routes/data modules
-2:(SOME FORM) does not do error checking
-3: Had to correct your forms to give them actions
-2 to -10: General Error or bug -2 to -10 for each bug or error, -2 for minor bug, -10 for major bug (cap at -30)
-10: Node modules folder included in zip
-10: Able to create multiple accounts with the same email address just different case
-50 if your application does not run
```

- **Data**

  back_end

  - Back_advertisers.js **Mohammed**
  - Back_comments.js **Mohammed**
  - Back_dogs.js Jiang
  - Back_index.js Jiang
  - Back_longin.js **Pratim**
  - Back_users.js **Pratim**

- **Routes**

  advetisers.js **Mohammed**
  comments.js Jiang
  dogs.js Jiang
  index.js **Mohammed**
  login.js **Pratim**
  users.js **Pratim**

  front_favouriteList.js **Pratim**
  Front_index.js **Mohammed**
  Front_users.js **Pratim**
  Front_quiz.js **Jiang**

Front_map.js **Mohammed**

- **Config**
  mongoCollections.js
  mongoConnections.js
- **Interceptor**
  adminInterceptor.js
  userInterceptor.js

(**Yuting Shao**)

- **{Public**
  back_end
  - Css
  - Img
  - Js
  - Lib
  - admin_login.html
  - index.html
  - manage_advertisers.html
  - manage_comments.html
  - manage_dogs,html
  - manage_users.html

  front_end
  - Css
  - Img
  - Js
  - Lib
  - Index.html
  - Users_login.html
  - Advertisers_login.html
  - Users_register.html
  - Advertisers_register.html
  - Map.html
  - Quiz
    - Start
    - .
    - Result

  - user
    - User_center.html
      - Favoritelist
      - Profile

```
                        edit

            advertisers
                Advertisers_center.html
                Advertisers_dogsPost
                Advertisers_dogsGet
                Advertisers_dogsDeletet
                Advertisers_dogsUpdate
            Dogs_details
            Dog_Feeding_Guide
                Dog Nutrition(1-3 htmls)
                Dog Problem(1-3 htmls)
            .
            .
        404.html
        500.html
        400.html
        }
```

- **Views**
  .
  .
  .
- App.js
- package.json
- ...

Tentative Work-Flow

Write Code **(Two Weeks)**
Test Code **(2 Days)**
Final Deployment (Presentations, Additional Inputs) **(3-5 Days)**

Uploading this directory on our github repository

8 files for each backend developer,