

Wafer

LOOKING FOR A GOOD TIME? ADOPT A SHELTER DOG.





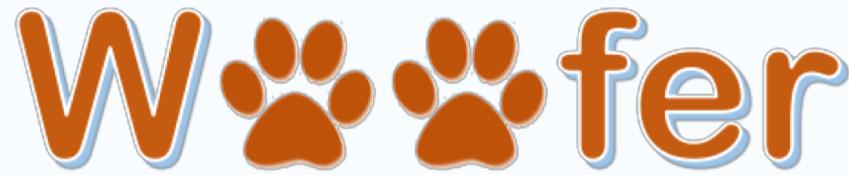
CREATE A WEBSITE THAT ALLOWS THE USER TO SEE A
SELECTION OF PHOTOS OF ADOPTABLE DOGS IN THEIR
AREA AND SAVE THEM TO A LIST WHERE THEY CAN
REVIEW MORE DETAILED INFORMATION AND CONTACT
THE RESCUE ORGANIZATION

Problem Domain



ACCESS THE PETFINDER API TO GET A LIST OF DOGS NEAR THE PROVIDED LOCATION. RENDER THE DOGS TO THE PAGE AND USE JTINDER TO ALLOW THE USER TO USE “TINDER-LIKE” SWIPE FUNCTIONALITY TO LIKE/DISLIKE VIEWED DOGS.

Solution Domain



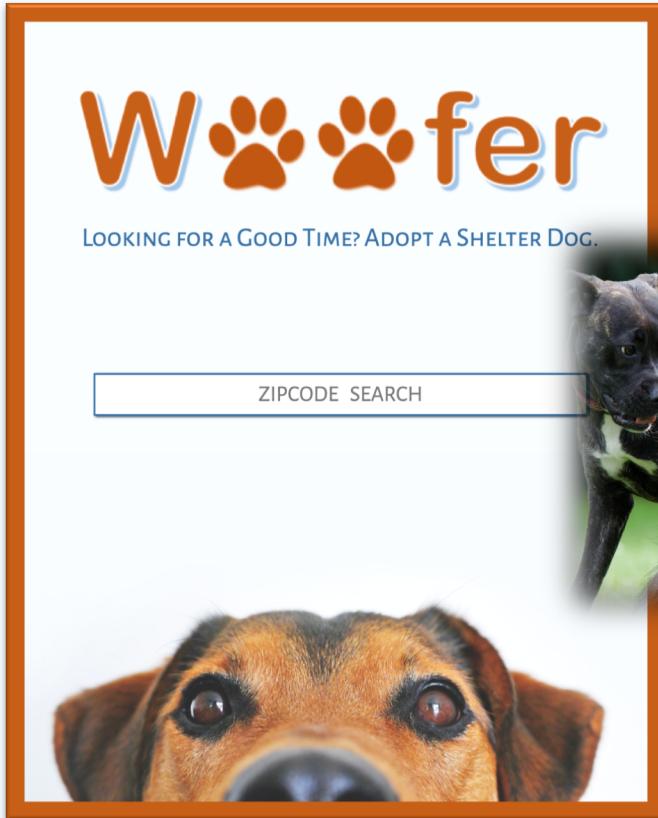
LOOKING FOR A GOOD TIME? ADOPT A SHELTER DOG.

APPLICATION DEMONSTRATION



Communication and Division of Work

FRONT END



BACK END

```
//-----WOOF LIST-----\n\nfunction woofList(request, response){\n    let likedDogs = [];\n    console.log(request.body);\n    let id = request.body.username;\n    let SQL='SELECT likes FROM users WHERE id = $1`;\n    client.query(SQL,[id])\n    .then(data=>{\n        let likes = JSON.parse(data.rows[0].likes)\n        let string = ''\n        for(let i=1 ; i<=likes.length; i++) {\n            string += '$' + i + ', '\n        }\n        let nstring = string.substring(0, string.length-2);\n        console.log(nstring);\n        let SQL2 = `SELECT * FROM dogs WHERE dog_id IN (${nstring})`;\n        client.query(SQL2, likes)\n        .then(result => {\n            result.rows.forEach(row => {\n                likedDogs.push(new DBDog(row))\n                console.log(`new dog created, ${likedDogs.length}`)\n            })\n            console.log(`rendering wooflist`);\n            response.render('pages/wooflist/listShow.ejs',{likedDogs});\n        }).catch((err)=>{console.log(err)});\n    }).catch((err)=>{console.log(err)});\n}\n\n
```

Project Organization

The image shows a Trello board titled "Woofern12" with the following columns and tasks:

- TODO**:
 - + Add a card
- IN PROGRESS**:
 - make our search hit 1000 dogs
 - phone, xl phone, desktop client
 - remove dogs from liked if the user decides(wooflist needs to have an unlike button)
- COMPLETE**:
 - setup: determine structure of SQL tables
 - frontend: setup index.ejs
 - frontend: researching swiping functionality
 - send a users list of liked dogs to their woof list
 - record which users have seen which dogs
 - make sure a user doesn't see the same dog twice
 - setup: update Trello board
 - individual dogs detail page
 - tap button functionality
 - setup: setup Trello board
- MVP**:
 - Wooflist mvp
 - dog detail page mvp
 - + Add another card
- STRETCH**:
 - create a client for actual computers
 - allow users to enter more settings than location'
 - give user keyboard accessibility
 - expand to cats/ other animals
 - cache invalidation
 - render icons based on an individual dogs information
 - + Add another card

Trello



Obstacles

- JTINDER LIBRARY UTILIZATION
- INTEGRATION OF MULTIPLE CSS SOURCES
- GETTING A USER ID FROM LOCAL STORAGE WITHOUT LOGIN FUNCTIONALITY

MERGE
CONFLICTS!!!



Accomplishments

- CLEAR AND LOGICAL WRRCC
- GREAT TEAM COMMUNICATION
- IMPLEMENTED THE JTINDER LIBRARY
- CREATED A DYNAMIC FRONT END USER INTERFACE





Future Functionality

- IMPROVE SPEED OF INTERACTIONS WITH DATABASE
- CACHE INVALIDATION AND UPDATE FOR SAVED DOGS
- EXPAND TO OTHER ANIMAL TYPES



- CREATE A USER PROFILE AND FILTER SEARCHES
- RESPONSIVE ON ALL DEVICES
- 100% KEYBOARD ACCESSIBLE APPLICATION





QUESTIONS?