

Making a simple Web Application:

I recently made a website for my own learning where you can make an account for your dog/pet & upload pictures of your dog. You can share their daily activities & see all other dog's profiles on the website. You can also schedule meetup for your dogs to make new friends.

Decomposition:

Breaking down the whole website into small problems and then sub categorize each problem into manageable parts.

- Creating a wireframe & design each component of the application.
- Break down the main webpage into small components including Navbar, Body, Footer
 - Navbar can further breakdown into application logo, profile picture, login/logout, forget password, settings etc.
 - Body could have a form to sign up for an account where user can put their information & if user already have an account, he/she will be able to sign in and can see their profile. User can interact with other accounts on the website.
 - Footer can have "Language preferences", "About the developers" and "Contact Us" information.

Pattern Recognition:

- Each user will have the same user data, which will include username, email, dog's name, password and zip code. We can use the same form for each user at time of sign in.
- Once user will login, every picture of dog will have title, & a story for their daily routine. So all the data will be stored in the same database with each user having their own column.
- When user will visit dog-A profile, all the pictures belongs to dog-A will come on the screen & this pattern will be same for each profile dog-B, dog-C ...
- Steps for editing profile information will be same for all the users
 - Edit profile picture
 - Go to settings -> edit profile -> edit profile picture -> upload picture -> attach picture -> upload.
 - Editing a password:
 - Go to settings -> edit profile -> change password -> check if password fulfil the complexity (includes special characters, numeric values etc.) -> update password in the database
- Pattern for editing password, username, email will be same for all the users

Abstraction:

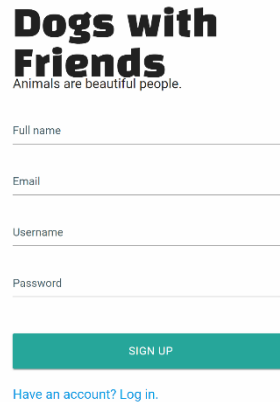
On each step, there will be some data, which we can filter out. It will make a problem easy to solve for example

- While signing up:
 - Important data:
 - User has a unique email address to which they have access to complete signup process
 - Unimportant details:
 - Which email account they have (google/yahoo/outlook)
- Uploading picture of dogs:
 - Important data:
 - Format of the picture should be .jpg or .png
 - Size of the picture is greater than 200x200 pixel (it's big enough for people to see it)
 - Unimportant details:
 - Maximum size of the picture
 - What dog is doing in the picture?

- When & where picture was taken?
- Who took the picture?

Algorithm Design:

Algorithm for signing in process will be as creating a main page with a form to sign up & a link to sign in if account already exist.



Dogs with Friends
Animals are beautiful people.

Full name

Email

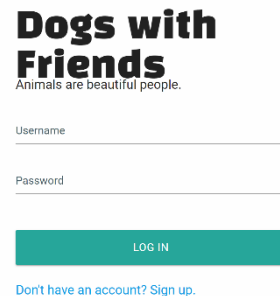
Username

Password

SIGN UP

[Have an account? Log in.](#)

- If “Have an account? Login” clicked
Take user to login page:



Dogs with Friends
Animals are beautiful people.

Username

Password

LOG IN

[Don't have an account? Sign up.](#)

Once “LOG IN” clicked

- If username exist:
 - Match password against that username in the database & if it is correct, let user login & route them to their profile page
 - If password is wrong, prompt an error & give them option to retry password & link to forget password.
- If username does not exist:
 - Route user to sign up page
- If signup clicked, take information provided in the form & create a profile for username. If email already exist in the database, route user to the sign in page with a prompt, account already exist & give them a link to forget password.