InclusiAbility: Phase 1

Group 05
Arul Tayal, Alexander Lee, Brandon Ling, Nick Favoriti, Ishaan Salhotra

Purposes/Motivation:

Our project strives to provide a detailed listing of disabilities and the diversity within this community. We also plan to include information about the various resources available for people in need, such as accessible places/events and technologies/services that can assist those with disabilities.

Website Architecture

For Phase 1, our architecture consists of the Frontend, which was built using React.

- Frontend (React)
    - Bootstrap:
        - Easy and consistent styling throughout
    - public folder:
        - contains images
        - people folder:
            - images of the team for about page
        - tools folder:
            - images of the tools we used for about page
    - Split src into:
        - components
            - contains the different custom components we used
            - i.e. navbar
        - data
            - contains json files with the website data
        - images
            - contains the images that are directly imported
        - pages
            - contains each page in the site
        - styles
            - contains the css styling for each page

User Stories

1. Number of Instances
    a. Customer - “I would like to see more instances as right now it seems you are planning to have 20, 30, 30. As a reminder, we need to have 50, 100, 150 instances.”
    b. We handled this Customer request by expanding to the state of Texas, rather than limiting ourselves to Austin. This allowed us to get more information for the things to do as well as resources. In regards to disabilities, we are going to scrape more specific disabilities, rather than looking at a higher level.
2. Media
    a. Customer - “Looking at the media you want to provide I see you are providing pictures of disabilities. I would like to see a graphical representation of how many people have that disability instead.”
    b. We handled this Customer request by implementing some basic graphical representations for now, but we will work on scraping better images in the next phase as we scrape APIs.
3. Resources
    a. Customer - “Right now it looks like you guys are leaning toward treatments as resources. Maybe also have help centers, hospitals, and organizations that provide support to disabled people as well.”
    b. We handled this Customer request by expanding our reach for treatments, which also helped us get more instances for resources. We will scrape these other data points as we begin web scraping.
4. What the website will display
    a. Customer - “You say you will provide an "overview of what disabilities are" in the project description but I would rather like to see detailed descriptions on different types of disabilities, like how it hinders the people affected and their struggles.”
    b. We handled this Customer request by making our descriptions more specific for each instance, handling the specific request given above. For our future instances, we will be sure to scrape specific information like those given above by our customer.
5. Sortable Topics
    a. Customer - “Have more ways to sort and filter your attributes. So like for resources, maybe a distance attribute so people can see the resources closest to them.”
    b. We handled this Customer request. We have manually sorted the instances, but as sorting features are not required to be implemented yet,

we do not have the drop down yet. However, manually they are sorted. In future phases, we will have the sort and filter options, in which we will include other attributes like the distance example given.

RESTful API Documentation

Our RESTful API documentation can be found here. The API will have endpoints to get ALL instances of disabilities, things to do, and resources stored in the database. Additionally, our API will have endpoints to get specific instances of disabilities, things to do, and resources based on various parameters.

For GET requests, we have not implemented authentication as there is no sensitive information stored in our database. However, for any PUT, POST, DEL requests, we will require higher privileges and keep them internal to the team of developers.

Models

Our project is utilizing three models: disabilities, things to do, and resources. Each model has at least 5 attributes associated with it. The disabilities model will contain the name of the disability, the category of the disability, causes, population statistics, and a list of famous people with such disability. The things to do model will contain the name of the place, its address, its zip code, disabilities it is accessible for, rating from google, and the average cost. Finally, the resources model will contain the name of the resource, disabilities it applies to, how to obtain it, how to use it, and its average price range. More information on each of the models (sorting and filtering options, connections to other models, and rich media features) can be found here.

In order to get the data for these attributes, we will scrape information from a series of API's. These scripts will upload data to our database after processing them. Then, we can access the information from our database and use it to regularly update our instances on our webpage.

Tools

We are utilizing the following tools in this phase:

AWS Amplify

- Allowed us to easily build, ship, and host our full-stack website

GitLab
- Allowed us to store files related to full-stack website in a repository, and allowed us to plan out out project using the issue tracker and milestones

React
- Allowed us to build an interactive UI for our website dynamically

Bootstrap
- Provided us with a framework and design templates for our website's UI

Zoom
- Allowed us to meet remotely to work on different aspects of the project

VS Code
- Used as our IDE for support with coding, debugging, and Git

Postman
- Allowed us to plan and document our API

Hosting

We are hosting the website at https://www.inclusiability.me/ via AWS Amplify.

Challenges

- The first challenge we faced was general communication. Figuring out a time and place to meet and work for all 5 of the team members proved to be harder than expected. This made it harder to find time to meet up with my group and get things started efficiently. However, we figured out times nonetheless and met up virtually when needed so we could all work together.
- Another challenge we faced was having to learn React for the front end. Most of us had little to no experience with React, so we had to spend extra time really learning how it works.
- A third challenge we faced was with using the Postman interface. Designing the interface meant we needed to very carefully plan every possible parameter and return value we would be getting in the future. It was also hard to figure out how to publish the documentation to be able to view it in a documenter.getpostman. However, with researching through help articles on Postman, we were able to figure it out.
- A final challenge we faced was with the entire set up process, but more specifically with Amplify and getting the Namecheap linked to it properly. However, two team members worked closely through it, and after researching how to connect, were able to figure it out.