CS373 Technical Report for RelievePoverty.me

Table of Contents

Motivation

The motivation behind our site was the fact that poverty is becoming a bigger issue within the United States. The cost of living in many cities is exponentially increasing while minimum wage requirements remain almost constant. There are typically negative stereotypes concerning those who are living under the wage gap, and many are reluctant to help them.

Our hope is that by bringing awareness to the issues concerning poverty and recommending related charities that we can encourage others to help contribute to this cause.

User Stories

1. **Phase 1: Get charity links**

   Customer Description: As a user, I would like to be able to get links to the websites of charities. This would facilitate donating and allow me to read more about the charities I care about. The link received from the API should be fully clickable/able to be pasted into a web browser.

   Estimation: We initially estimated that it would take about 3 minutes to use Charitynavigator's API to get data on 3 instances.

   Refinement: After enlarging the scope to account for the time to display the data on our site and add the functionality to our Postman API, we changed our estimation to 45 minutes.

   Actual Time: 50 minutes.

2. **Phase 1: Get state poverty rankings**

   Customer Description: As a user, I want to be able to get a ranked list of all 50 states from highest percent of population in poverty to lowest. This could easily be done by allowing users to get the rank of a given state. This information is useful in determining which areas need the most help.

   Estimation: After assuming that we would only be obtaining the ranks for the 3 instances we were using, we decided that it would take about 5 minutes to scrape the information.

   Refinement: After enlarging the scope to account for the time to display the data in sorted order and add the functionality to our Postman API, we changed our estimation to 30 minutes.

   Actual Time: 27 minutes.

3. **Phase 1: Get state poverty percentage**

   Customer Description: As a user, I would like to be able to get the poverty percentage of my state. Given a state, the api should return the percentage of its population that live in poverty. This information is useful in determining which areas need the most help.

   Estimation: After assuming that we would only be obtaining the percentages for the 3 instances we were using, we decided that it would take about 5 minutes to scrape the information.

   Refinement: After enlarging the scope to account for displaying the data on our website and adding the functionality to our Postman API, we changed our estimation to 25 minutes.

   Actual Time: 24 minutes.

4. **Phase 1: Get state poverty line**

   Customer Description: As a user, I would like to be able to query the average poverty line for each state via the api. Given a state, the api should return the quantitative poverty line (in USD) in that state. This is valuable information that can be used to inform salary decisions made by companies.

   Estimation: After assuming that we would only be obtaining the averages for the 3 instances we were using, we decided that it would take about 5 minutes to scrape the information.

   Refinement: After enlarging the scope to account for displaying the data on our website and adding the functionality to our Postman API, we changed our estimation to 25 minutes.

   Actual Time: 16 minutes.

5. **Phase 1: Get news**

Customer Description: As a user, I would like to be able to retrieve news on poverty that the site has to offer. It would be convenient to give the api a range of dates and get all the articles between them. This could be link to the articles, or the articles themselves.

Estimation: We estimated that it would take at most 10 minutes as the only requirement was to add a parameter to a single API call on Postman.

Refinement: Instead of adding just 1 example to filter by date, we decided to also have an example to filter by both date and state and changed the estimation to 15 minutes.

Actual Time: 7 minutes.

RESTful API

Background:

Our API will be hosted at https://api.relievepoverty.me and will be both designed and tested on Postman. Our current design and documentation is available here. Our approach was to have two API calls for each model; one to get a list of all instances with a small amount of information, then another to get all of the information about a single instance. We felt this approach was a good representation of our web-flow where a user wouldn’t need as much information to select a charity but would probably like more after viewing the instance itself.

Models

Articles

As the name suggests, each instance of this model is an article scraped through news sources such as the NYTimes. The model page has a list of all articles with a preview of the headline, a small summary, and a photo from the article. After clicking on an article the user is then sent to an instance of it that has the above information in addition to the names of the news organization, the author, the date published, a link to the actual article, the state the article is about, and a list of related charities. The article instances also have internal links to their respective state and related charities.

Charities

Each instance of this model is a charity scraped through the Charitynavigator API. The model page has a list of all charities with the charity name, mission statement, and photo. After clicking on a charity the user is sent to an instance of it that has the above information in addition to the charities' affiliation, tax classification, state, mailing address, website link, map, and list of related charities. The charity instances also have internal links to their respective state and related articles.

States

Each instance of this model is a state within the United States. The model page has a list of states with their names, flags, and poverty rankings. After selecting a state the user is then sent to an instance of it that has the above information in addition to the number and percentage of citizens below the poverty level across three different age distributions. The instance also has a list of counties with the highest poverty rate, and internal links to related articles and charities.

Tools

GitLab

For version control and continuous integration our group uses GitLab. We currently have two branches: a master and development branch. Our repository is located at https://gitlab.com/urielkugelmass/relievepoverty.

Postman

Currently we are using Postman to design our API's. Our current documentation is located at https://documenter.getpostman.com/view/5460449/RWgjY1qy.

AWS

We are currently using AWS to host our webapp. We will eventually also host our backend and database on it.

Bootstrap

We're using bootstrap, an open source CSS toolkit, for most of the designs on our website.

Hosting

We’re currently hosting our webapp on AWS. Within AWS, we have three buckets: relievepoverty.me which hosts our application, www.relievepoverty.me which redirects to the prior bucket, and api.relievepoverty.me which will eventually host our backend. All three are distributed through Cloudfront on AWS which provides our domains with SSL Certificates. The only downside is that Cloudfront’s cache has to be manually invalidated to display new versions of our buckets.