# Congress and Me

## Technical Report

Abu-Zayd Abdullah
Ben Burton
Calvin Dao
Robert Gutierrez
Ben Hazel
David Sikabwe

congressand.me

# Motivation

The six of us believe that a well-informed populace is crucial to a functioning democracy. We wanted a way for people to easily see what issues their representatives are and aren't talking about in their tweets and on the Congress floor, and we wanted to highlight which issues require attention on a nation-wide scale. To that end, we've built Congress and Me.

# User Stories (from Customer)

## Phase 1

- As a user, I would like the state's list of congress members to link to their specific web pages
- As a user, I would like to be able to view pictures of each and every congress member.
- As a user, I would like an about page that gives me more information about the developers and their individual contributions.
- As a user, I want a homepage that includes a basic description of what the website is about.
- As a user, I want to be able to navigate between pages at the top of the screen.

## Phase 2

- As a user, I would like my scroll height to be reset when changing pages.
- As a user, I would like an instance's picture on the model page to be clickable.
- As a user, I would like a footer on the bottom of (most) the pages.
- As a user, I would like different background photos for each model pages.
- As a user, I would like social media links to open in a new tab.

# User Stories (to Developer)

## Phase 1

- As a user, I'd like to see where a given language fits on a language cloud.
- As a user, I'd like to see the most spoken language in each country.
- As a linguist, I'd like to see the IPA transcription of a given language's autonym.
- As a user, I'd like to have a pretty splash page greet me upon visiting the site.
- As a user, I'd like to see the ISO language codes for a given language.

## Phase 2

- As a user, I'd like a short bio about and picture of each of the developers.
  - The about page looks a little... unfinished. It appears that you're pulling all the information from GitLab. You should ask the dev team to add a profile picture to their GitLab profiles so the about page shows their beautiful faces. Furthermore, you're all very interesting people and we'd like to know all about you! Give us a little information about yourselves!
- As a user, I'd like the country's flag to be on its country card.
  - The graphs on the Countries model page are very pretty. Problem is, I have no idea what they represent. I think it would be a much better idea to have the country's flag as the image. It would make countries easier to spot (my eye would see the Canadian flag a lot faster than the word Canada).
- As a user, I'd like the links to charities to open in a new tab.
  - Currently, the links open in the current tab. You want people to stay on your site for as long as possible (so they can see more ads 😉). We want as little interruption of the browsing experience as possible.
- As a user, I would like the About page cards to not be clickable.

    - Currently, the about page cards very much look like they do something or link to something. When you click on them, there is an animation, but nothing happens. They should either link to something or not be clickable.
- As a user with 30 tabs open, I'd like a non-default favicon on the tab.
    - Maybe put the little waving dude up there. It would help your web page appear more professional. Also, it would help the user quickly identify which tab is The Last Word.

# APIs

## Outsourced APIs

We gathered data on representatives, states, and issues from RESTful APIs exposed by three different websites. We used ProPublica to get information about members of Congress and issues being discussed. We used Sunlight Foundation to get information about states and who represent them in Congress. We used Google's Civic Information API to get more information about issues and Congress Members. We also used GitLab's API to dynamically display information about our team members' issues and commits.

## Our API

Our Restful API has four endpoints that can be queried: Representatives, States, Issues, and Mentions. We currently expose four methods: GET States, GET Issues, GET Representatives, and GET Mentions.

All four of them currently only take in one query parameter: page_number. The data is divided into pages (Representatives has 54 per page and States is 10 per page, for example), and if page_number is specified, then the API will return a page's worth of results for the given endpoint. Otherwise, all instances will be returned.

We plan to implement more query parameters in later phases.

# Models

Our three models are Representatives (members of the United States House and Senate), States, and Issues (abortion, guns, immigration, et cetera). Representatives are from states and discuss issues. States have representatives and have a most-discussed issue. Issues are discussed by representatives, and they have certain states in which they are hotly discussed.

- Issues
    - Name
    - National ranking in regards to other issues being discussed
    - State ranking in regards to other issues being discussed
    - States in which it is being discussed
    - Representatives who are talking about the topics
- State
    - Name
    - Congress Members
    - Issues being discussed in state
    - Ranking of issues being discussed
    - Issues not being discussed in state that are being discussed in other States
- Congress Members
    - Name

- Home State
- Party Affiliation
- Issues that they are discussing

# Tools

## React
Our front end uses React, a JavaScript library. We use Bootstrap, a CSS framework, for styling.

## Database
Our database has four tables: Representatives, Issues, States, and Mentions. The attributes for the first three tables are the attributes for the corresponding models. The final table Mentions is a grotesque jerry-rigged association table linking Representatives, States, and Issues. The attributes for this table include a representative's name, their state abbreviation, and a column for every issue. Each issue attribute currently lists whether or not that issue is on that representatives website. In future phases, we will work on getting more robust data on frequencies of discussion. We use the Flask-SQLAlchemy framework to manipulate our database with Python code (see `app.py`).

## Slack
Slack is our team collaboration tool. We have different channels for different topics like front end, back end, AWS, et cetera.

## GitLab
We use GitLab for version control. GitLab's issue boards are helpful for keeping track of which members of our group are assigned to what tasks as well as allowing us to connect with our customer group—they can provide issues and we can discuss the issues with them.

## Flask
We use Flask, a Python web application framework, to access our database with Python code. Flask-SQLAlchemy, an extension of Flask, lets us create API endpoints in our Python code.

## Docker
Docker creates a virtual environment on which to run code. Our EC2 instance uses our Docker image to run our backend.

# Hosting

## Amazon Web Services
Our front end is a React app that we store using S3, a cloud storage service within AWS. This S3 bucket is deployed through CloudFront, AWS's content delivery network.  Our backend is a Flask app running on an EC2 (Elastic Cloud Compute, also an AWS service) instance. Our PostrgreSQL database is stored on AWS's Relational Database Service (RDS).

## Domains
We acquired our pretty URL congressand.me from NameCheap (we used to be Congress Conversation, but Congress and Me is conducive to a free URL). This URL is used as a CNAME alias that points to our CloudFront endpoint. We obtained an SSL certificate through

the AWS Certificate Manager, which allows us to secure our site with HTTPS. Our API endpoint is registered as a CNAME through Elastic Beanstalk.