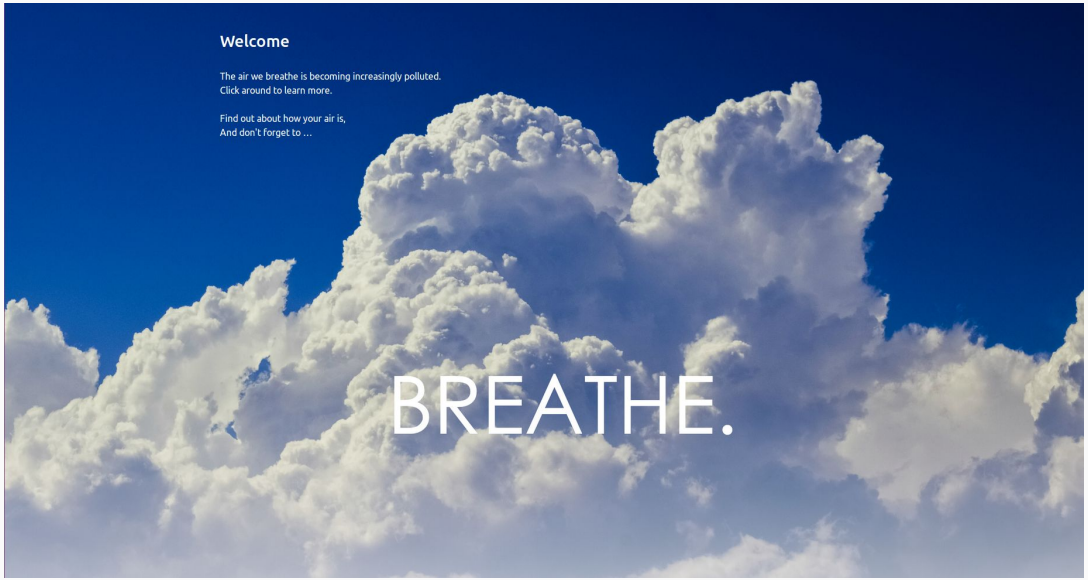
Engage Climate Change

- What they did do well?
  - Very clean UI, a lot of data on the instance pages
  - Social Media integration
  - Politicians' YouTube videos
- What did we learn from their website?
  - Taking notes on design and layout
  - We were able to learn information about politicians and their related legislature
- What can they do better?
  - Fix the alignment of their histogram visualizations.
  - Word clouds can be messy and not informative enough
- What puzzles us about the website?
  - Too much unavailable information for environmental legislation instances


HowsMyAir.me
CS 373 Group 3
David Bommersbach, Jason Lihuang,
Zelin Cui, Jamison Miles, Jason Sim

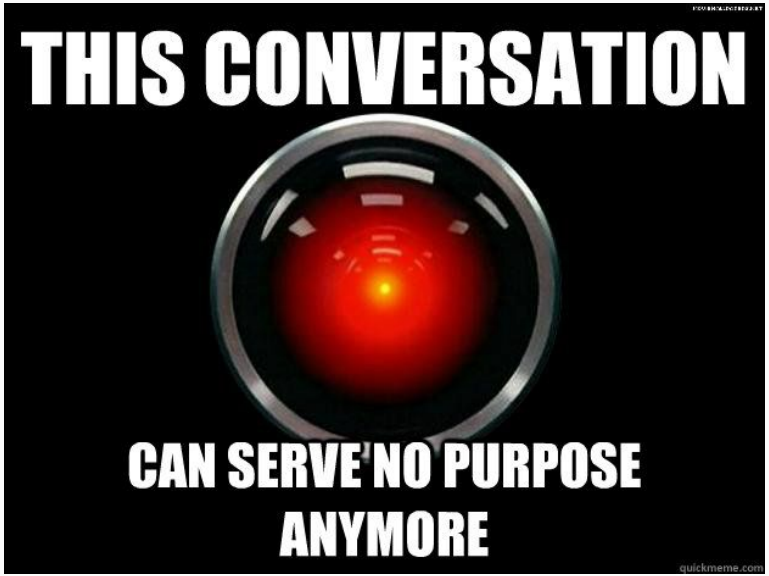
Overview

1. Introduction
2. Static Site
3. Dynamic Site
4. Added Functionality
5. Visualizations & Demo



How’s Your Air? 🤔

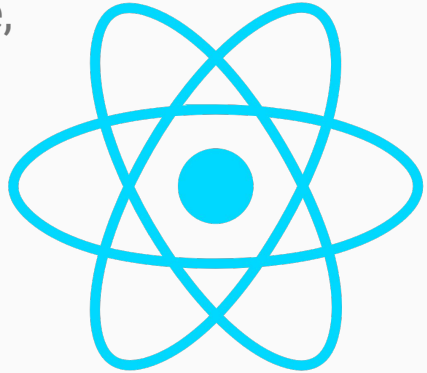
- **Purpose?**
  - **Overview of Air Quality**



Phase I: Static Website and Setup

React-Bootstrap & React Strap: Premade, reusable components

- Cards
- Rows, Columns
- ListGroups
- NavBar and Routing



Static Website Hosting

- Elastic Beanstalk Environment
- S3 used to host our website
- Domain Name provided by name cheap
- Cloudfront as Content Delivery Network
- Route 53 as DNS web service

Phase II: Dynamic Site Transition

- Using Component States
- Fetching Data
- Implementing Pagination

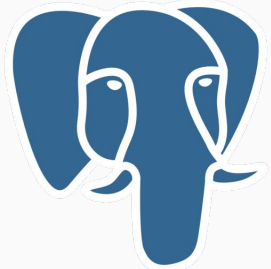
```
91      componentDidMount() {
92        console.log(this.getQuery())
93        fetch(this.getQuery())
94            .then(res => res.json())
95            .then(data => {
96            this.setState({
97              data : data,
98              num_results: data.num_results,
99              objects: data.objects,
100             page: data.page,
101             total_pages: data.total_pages,
102             isloaded : true,
103             states : this.getUniques(data.objects,"state_name"),
104             time_zones : this.getUniques(data.objects,"timezone")
105           })
106           if (this.props.setObjects != null){
107             this.props.setObjects(data.objects)
108           }
109       })
110     };
```

Backend Infrastructure

- Compute provided by EC2
- Web App API contained in Docker
- PostgreSQL on RDS

Flask, SQLAlchemy and Database

- Flask and SQLAlchemy are the tools we used to generate our Database
- SQLAlchemy set up the DB tables and connections necessary
- Flask provided the RESTful resources
- Database accessed and managed through PgAdmin



SQLAlchemy

Data Scraping and Formatting

- Large General Scrapers
  - Written in python
  - Decided it was better to get the most information we could
  - Stored the data in CSVs rather than JSON in files.
- Excel for data processing

Phase III: Additional Functionality

- Searching, Filtering, and Sorting
- Building JSON Query Strings

```
112     getQuery(){
113       let filterObjs = buildFilter(`${this.props.searchfield || 'city'}`,"ilike",`%25${this.props.searchstring || ''}%25`);
114       let orderObjs = buildOrder("city","asc")
115
116       if (this.state.current_state != "None") {
117         filterObjs += ","+buildFilter("state_name","eq",this.state.current_state)
118       }
119       if (this.state.current_timezone != "None") {
120         filterObjs += ","+buildFilter("timezone","eq",this.state.current_timezone)
121       }
122
```

Unit Tests

- Front-end:
  - Enzyme, Chai
  - Mocha
- Back-end:
  - Unittest
  - Postman





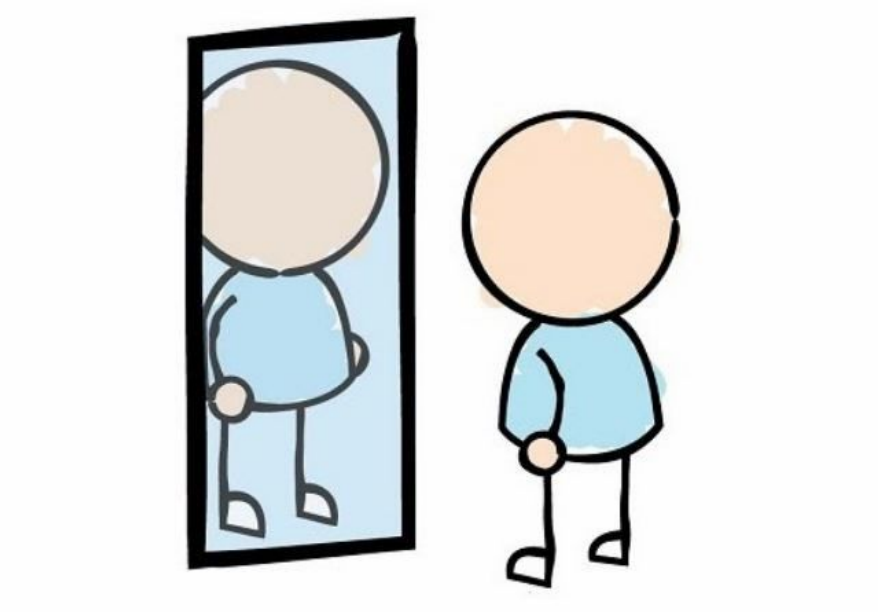
Acceptance Tests

Selenium:

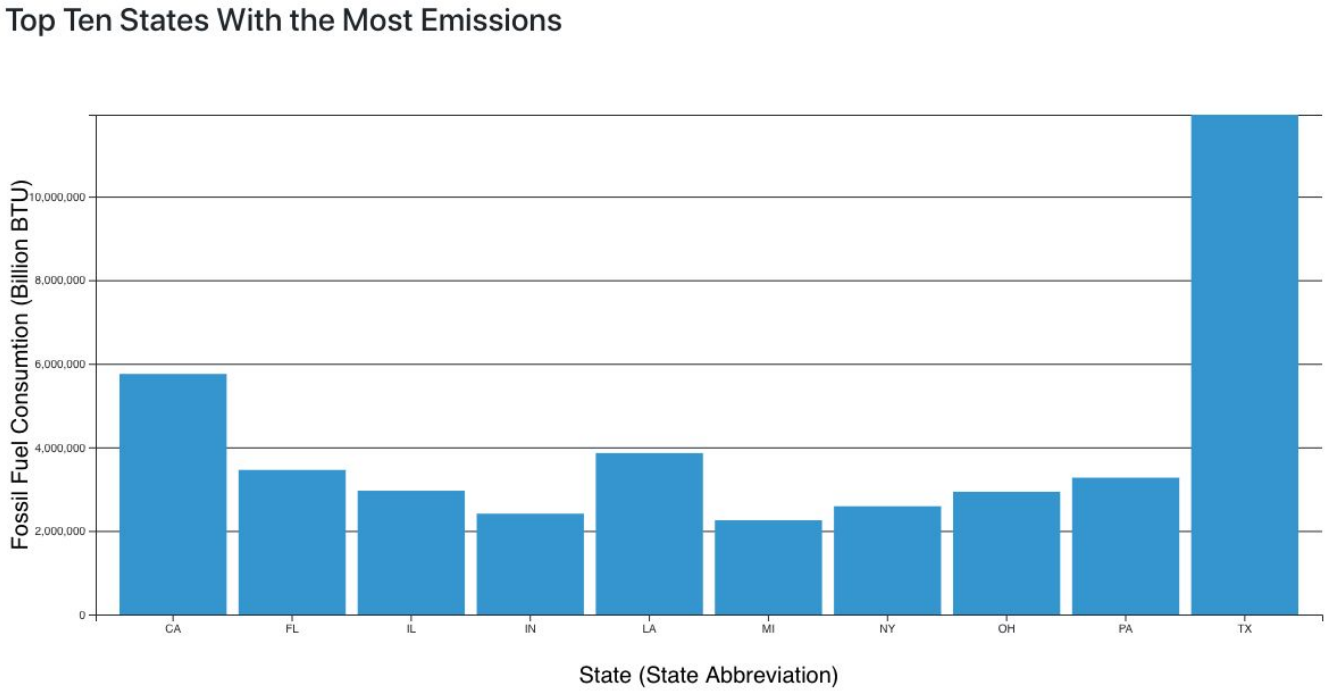
- Web automation framework
- Mimics real life interaction with the web browser

Self Critique

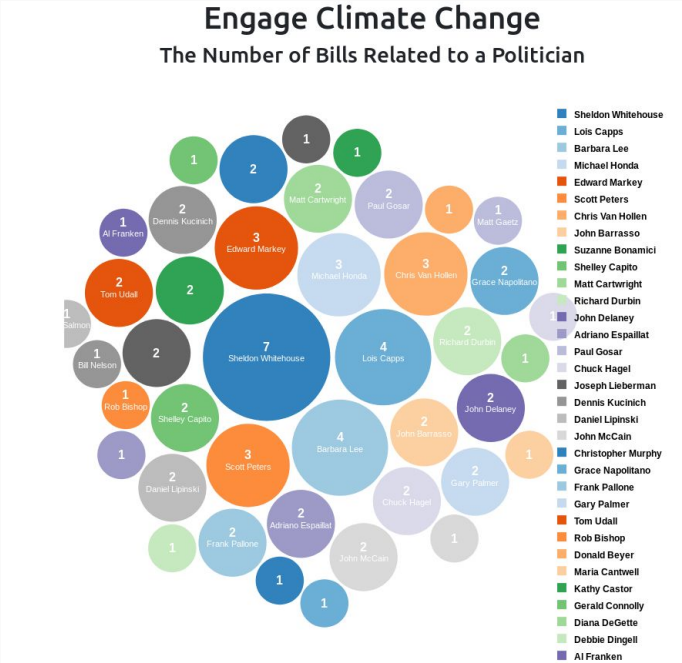
- What did we do well?
- What did we learn?
- What can we do better?
- What puzzles us?



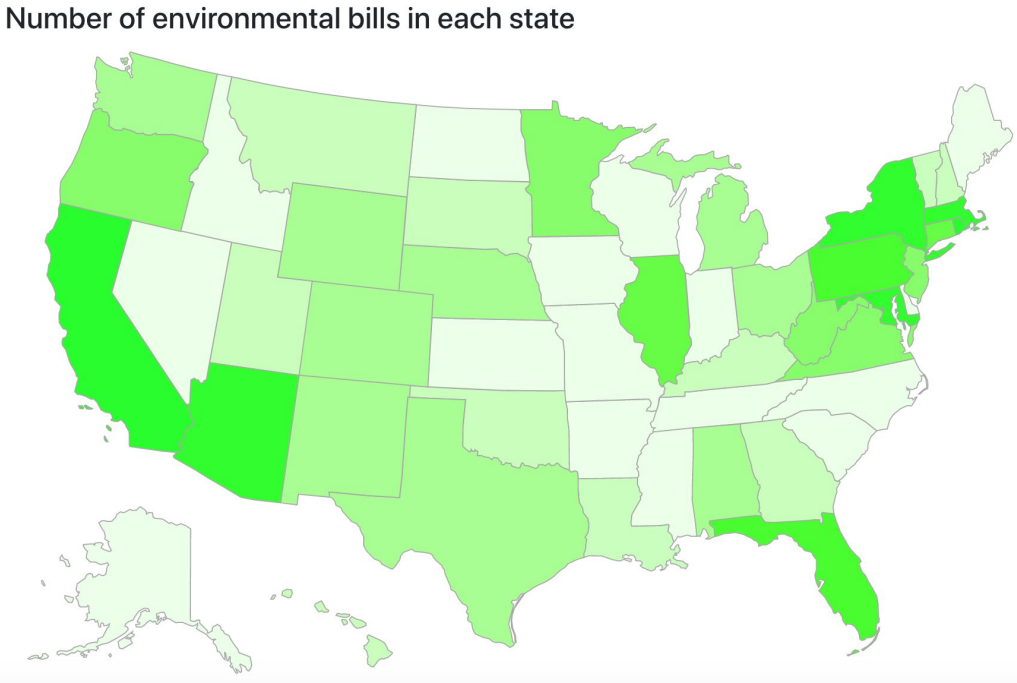
Engage Climate Change Visualization 1:



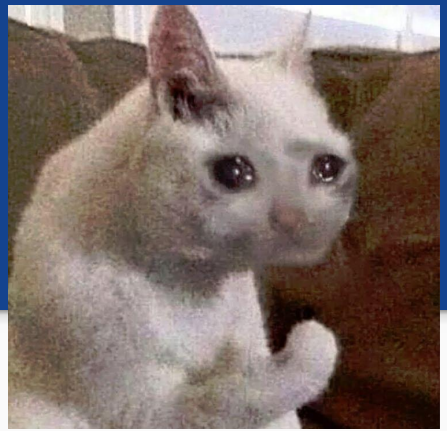
Engage Climate Change Visualization 2:

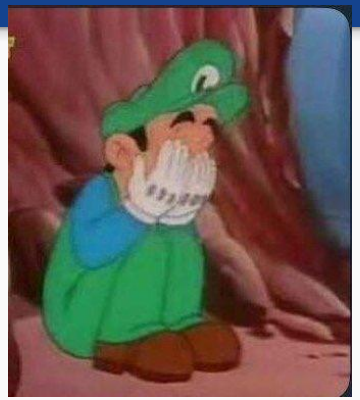


Engage Climate Change Visualization 3:



Demo





WEBSITE