|  |
| --- |
| It’s Just Politics |
| Specification Documentation |
| Functional Specifications |

|  |
| --- |
| Steph Michalopoulos [Team Steph]  January 2020 – March 2020 |

# Product Description/Overview

**Problem:** Staying politically informed requires a great deal of effort, which leaves many voters filling out ballots without sufficient knowledge. Sorting through internet articles for reliable sources is time-consuming and gathering accurate, factual candidate data requires diligent research.

**Proposal**: To develop a responsive web application that allows users to become and remain informed about political matters and officials. The product will provide users the ability to customize their own political news dashboard featuring articles from news sources of their choice, search detailed information about the activity and performance of our senators, obtain contact information for other state officials, and stay updated in the upcoming presidential elections with primary polls from various sources. “It’s Just Politics” will provide specific information on the congress history of any sitting United States senator including sponsored bills, committee and subcommittee memberships, DW-NOMINATE scores, and how a given senator’s votes align with party affiliation. In addition to the data on senators and state-level officials, the site will include a page displaying recent presidential primary polling data from multiple sources. This product will offer the convenience of information consolidation.

# Goals

## Minimum Viable Product

Users will need the ability to:

* Register/Log in
* Select and edit the news sources applied to their dashboard
* Add keywords/search terms to be applied to their news dashboard
* Save articles
* View senators and other state-officials by state search
* View congress history of senators
* Note senate seats up for grabs in 2020
* View polling data for the 2020 presidential election

# Product Requirements and Resources

## Technologies

### Backend

* Visual Studio 2015, 2017, or 2019
* .NET Framework 4.5.2 or higher
* Languages
  + C#

### Frontend

* Libraries/References
  + Bootstrap library
  + Font-awesome library
  + jQuery library
  + easy-pie-chart library
  + Google charts
  + “Anychart” library
* Languages
  + HTML5
  + jQuery
  + JavaScript

### Database Storage

* Microsoft SQL Server Management Studio 2014 or later
* SQL Server Native Client 10.0
* Database Privileges
  + db\_owner
* Languages
  + Transact SQL

### Deployment

* Web server FTP access
  + Host
  + Username
  + Password
  + Port number

### Developer Tools

* GitHub for source control
* Postman for API testing

## Data Sources

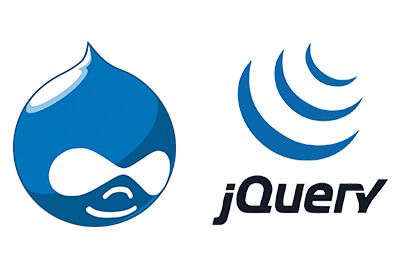
### API Documentation and Credentials For:

* Google Civic Information API
* ProPublica Congress API
* ProPublica Campaign Finance API
* News API

### Bulk Imports

* FiveThirtyEight polling .csv files
  + <https://github.com/fivethirtyeight/data/tree/master/polls>



A close up of a sign

Description automatically generated

# Out of Scope

## Additional Features to Consider Post-Deployment

* Incorporate congress activity in real time
* Gather detailed data on candidates running for open senate seats
* Include information and resources regarding how users can get more involved in supporting candidates

# Open Questions

* How will API server issues be handled?
* Will we need to request an increase in maximum daily requests?
* Can all likely errors be accounted for when relying on multiple API’s?

# Approach & Logic

## Organizing Data

### Dynamic

* Ex. Voting related data that changes frequently

### Static

* Ex. Names of United States Senators in the 116th Congress

### Why?

To limit the use of web service calls and furthermore rely less on technology that is beyond direct control.

## Retrieving Data

* Stored procedures

Absolutely not SQL queries will be embedded in source code to protect the application from SQL injection attacks.

Order of Development

1. Create classes
2. Dynamic data storage
   1. Create stored procedures
   2. Insert data into database
3. Determine best methods to make connections between API’s
4. Create preferences page
5. Create news dashboard
6. Create representatives
7. Import and test polling data
8. Create graphs for the representatives’ page
9. Create graphs for polling data

# Measuring Impact

## Accuracy Testing

I will confirm the accuracy of the data displayed on the webpage by debugging and comparing the data retrieved by the product to the data received in Postman. I will do this for all senators and state-level officials.

# Security and Privacy

## Database Security

* Execute all SQL queries by use of stored procedures
* Absolutely no SQL statements will be included in the source code

## User Security

* All passwords will be hashed
* No sensitive data will be collected
* Data stored – email address, hashed password, state registered if registered, political party affiliation

# Timeline/Milestones

## Database Preparation

Completed by the start of week three

## Preferences

Completed by the start of week four

## News Dashboard

Completed by the start of week five

## Representative Components

Completed by the start of week seven

## Presidential Primary Data

Completed by the start of week eight

## Fine Tuning and Testing

Weeks eight through ten

# Projected Challenges

Extensive data retrieval by use of API’s can become particularly difficult as data does not always arrive in the expected format. This project requires many calls to various API’s, with no common ID’s, which can make error handling challenging.