# ECE 595/495

High Level Design & User Stories

# Team Project: myVote

Fall 2016

Abigail Jacoby

Alexander Roessner

Karthick Krishnamurthy

Suresh Babu Panem

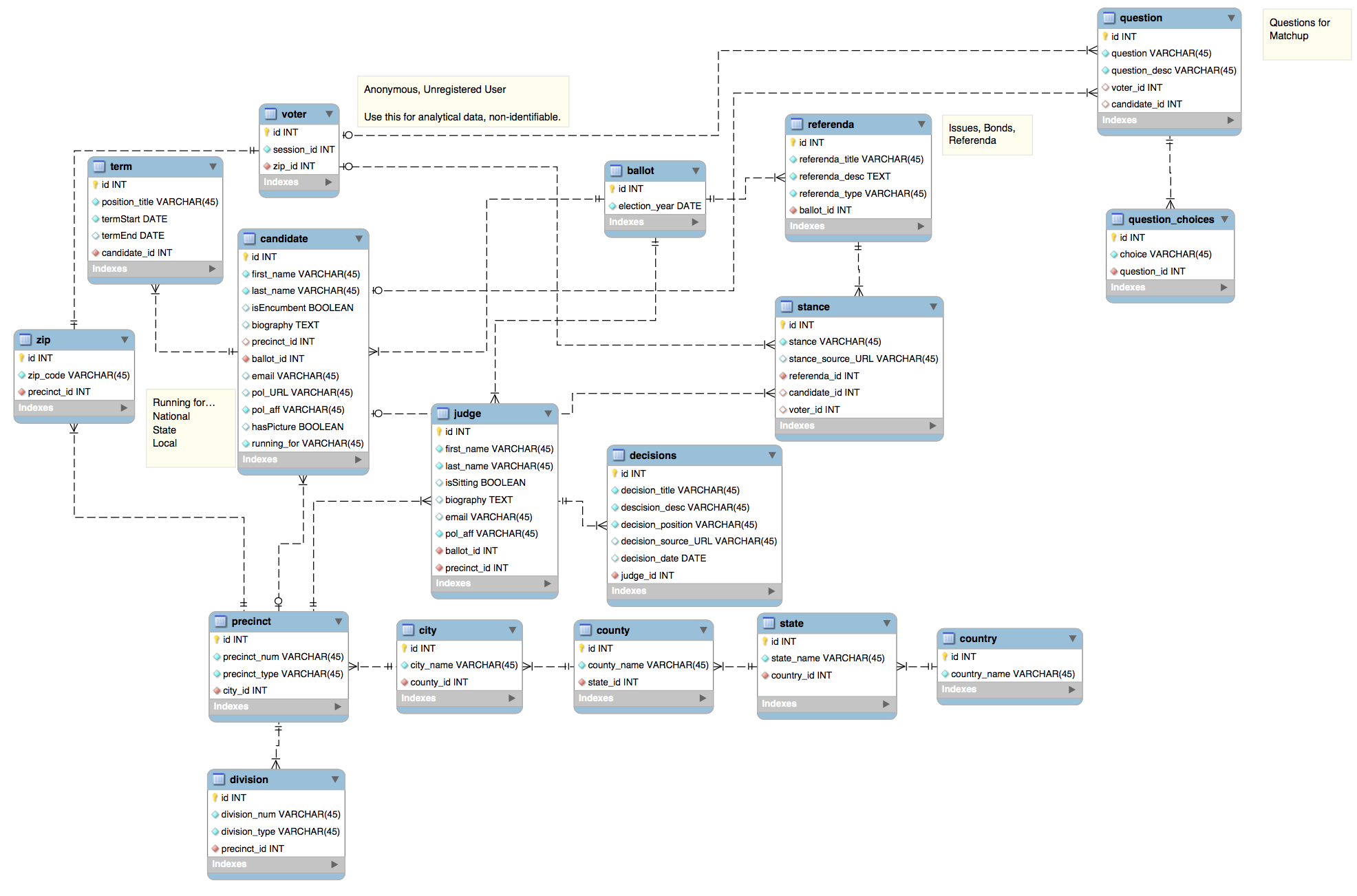
Bharath Kumar Reddy Anandigari

Ravi Teja Gorti

Shawn Chavez

# High Level Design: ERD

Below is the ERD that was developed for the myVote site. Changes can be expected, but the basic data organization and separation is set.



([PDF Link](https://github.com/roessnakhan/ece595-project/blob/master/artifacts/ERD_alt.pdf))

# High Level Design: ERD Breakdown

Below is a listing and explanation of any data listed in the above ERD.

## VOTER

* id – INT – ID for voter. (REQ, PK)
* session\_id – INT – This session variable is meant to track a user through the site without having to have them register, as the idea of having someone’s political opinions stored might actually deter people from using the site. (REQ)
* zip\_id – INT – ZIP Code Field. This connects to another table, but the idea would be that a user enters their ZIP code and then if a match is found in the ZIP table it’ll allow them to proceed (i.e. we’ll have data for that area). (REQ, FK)

## BALLOT

* id – INT – ID for ballot. (REQ, PK)
* election\_year – DATE – Year of election, could be adjusted to just be the election day. (REQ)

## CANDIDATE

* id – INT – ID for candidate. (REQ, PK)
* first\_name – STRING – First name of candidate. (REQ)
* last\_name – STRING – Last name of candidate. (REQ)
* isEncumbent – BOOL – Whether a candidate is currently in the position they’re running for.
* biography – TEXT – Candidate biography for their page. (OPT)
* precinct\_id – INT – If a candidate is running for a precinct office, their precinct ID. (OPT, FK)
* ballot\_id – INT – Ballot a candidate will appear on. (REQ, RK)
* email – STRING – E-mail address of candidate. (OPT)
* pol\_URL – STRING – URL for this person’s website. (OPT)
* pol\_aff – STRING – Political affiliation. (REQ)
* hasPicture – BOOL – Indicator that candidate has a picture on the server. Probably won’t implement this. (OPT)
* running\_for – STRING – What position they’re running for. (REQ)

## TERM

* id – INT – ID for Term. (REQ, PK)
* position\_title – Title of past/current position. (REQ)
* termStart – Start date of past/current position (REQ)
* termEnd – End date of past/current position (OPT)
* candidate\_id – INT – ID of candidate. (REQ, FK)

## ZIP

* id – INT – ID for ZIP Code (REQ, PK)
* zip\_code – STRING – ZIP Code (REQ)
* precinct\_id – INT – Precinct ID for given ZIP Code. (REQ, FK)

## PRECINCT

* id – INT – ID for Precinct (REQ, PK)
* precinct\_num – STRING – The number for the precinct. (REQ)
* precinct\_type – STRING – Explains what differentiates the precinct from the others. (REQ)

## DIVISION

* id – INT – ID for Division (REQ, PK)
* division\_num – STRING – The number for the division. (REQ)
* division\_type – STRING – Explains what differentiates the division from the others. (REQ)
* precinct\_id – INT – ID for precinct. (REQ, FK)

## CITY

* id – INT – ID for City (REQ, PK)
* city\_name – STRING – Name of city. (REQ)
* county\_id – INT – ID of county. (REQ)

## STATE

* id – INT – ID for State (REQ, PK)
* state\_name – STRING – Name of city. (REQ)
* country\_id – INT – ID of county. (REQ, FK)

## COUNTRY

* id – INT – ID for Country (REQ, PK)
* country\_name – STRING – Name of city. (REQ)

## JUDGE

* id – INT – ID for judge. (REQ, PK)
* first\_name – STRING – First name of judge. (REQ)
* last\_name – STRING – Last name of judge. (REQ)
* isSitting – BOOL – Whether a judge is currently in the position they’re running for.
* biography – TEXT – Judge biography for their page. (OPT)
* email – STRING – E-mail address of judge. (OPT)
* pol\_aff – STRING – Political affiliation. (REQ)
* ballot\_id – INT – Ballot a judge will appear on. (REQ, RK)
* precinct\_id – INT – If a judge is running for a precinct office, their precinct ID. (REQ, FK)

## DECISIONS

* id – INT – ID for Decision. (REQ, PK)
* decision\_title – STRING – Title of judgment or decision (REQ)
* decision\_desc – STRING – Description of judgment or decision. (REQ)
* decision\_position – STRING – How the judge ruled in decision. (REQ)
* decision\_source\_URL – STRING – If the decision can be referenced by a URL, that is available. (OPT)
* decision\_date – DATE – When a decision occurred. (OPT)
* judge\_id – INT – ID of judge. (REQ, FK)

## REFERENDA

* id – INT – ID for Referenda (REQ, PK)
* referenda\_title – STRING – Title of referenda (e.g. Bond C). (REQ)
* referenda\_desc – STRING – Description of referenda. (REQ)
* referenda\_type – STRING – Type of referenda (bond, issue, referenda). (REQ)
* ballot\_id – INT – ID for Ballot. (REQ, FK)

## STANCE

* id – INT – Stance ID (REQ, PK)
* stance – STRING – Position (for, against, neutral, undecided, etc.) (REQ)
* stance\_source\_URL – STRING – For candidates, this would be to source where their stance came from. (OPT)
* referenda\_id – ID for referenda. (REQ, FK)
* candidate\_id – ID for candidate, if this relates to them. (OPT, FK)
* voter\_id – ID for voter, if this relates to them. (OPT, FK)

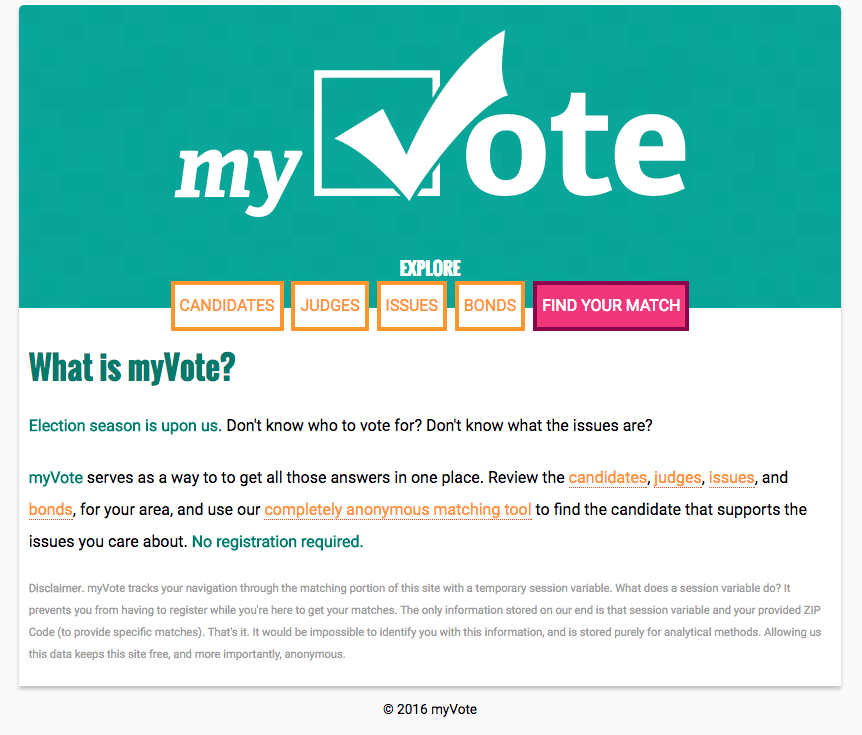
## QUESTION

* id – INT – ID for matching questions. (REQ, PK)
* question – STRING – Basic question (REQ)
* question\_desc – STRING – Question description. Quick overview. (REQ)
* voter\_id – ID of voter if this question is being asked to them. (OPT, FK)
* candidate\_id – ID of candidate if this question is being asked to them. (OPT, FK)

## QUESTION CHOICES

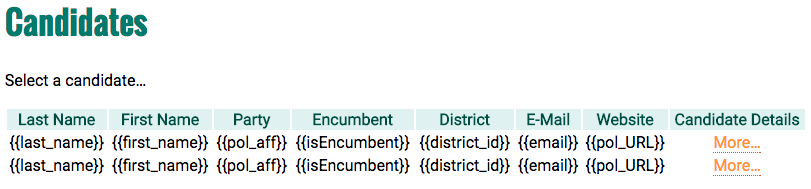
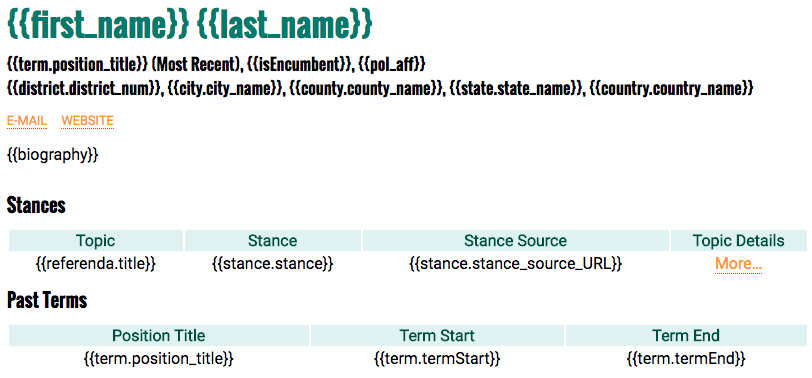
* id – INT – ID for question choice. (REQ, PK)
* choice – STRING – Question choice. (REQ)
* question\_id – INT – ID of question choice is being assigned to. (REQ, FK)

# User Stories



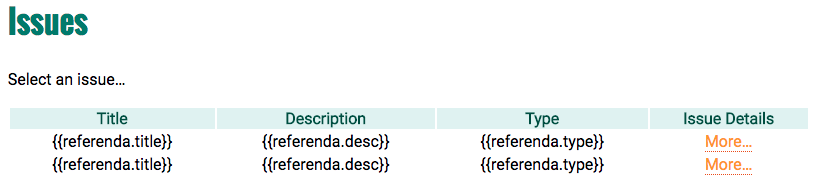
## I’m a User and I want to see where a Candidate stands on an issue.

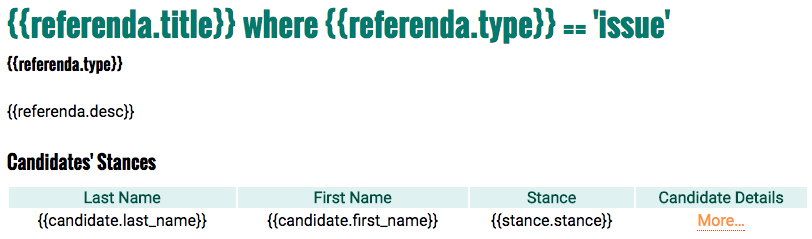
A user (visitor) can just navigate to Candidates to get a breakdown of all candidates and they can drill down further to a specific candidate.



## Likewise…

## …a user can also go to an issue page to see a Candidate’s stance on that issue.

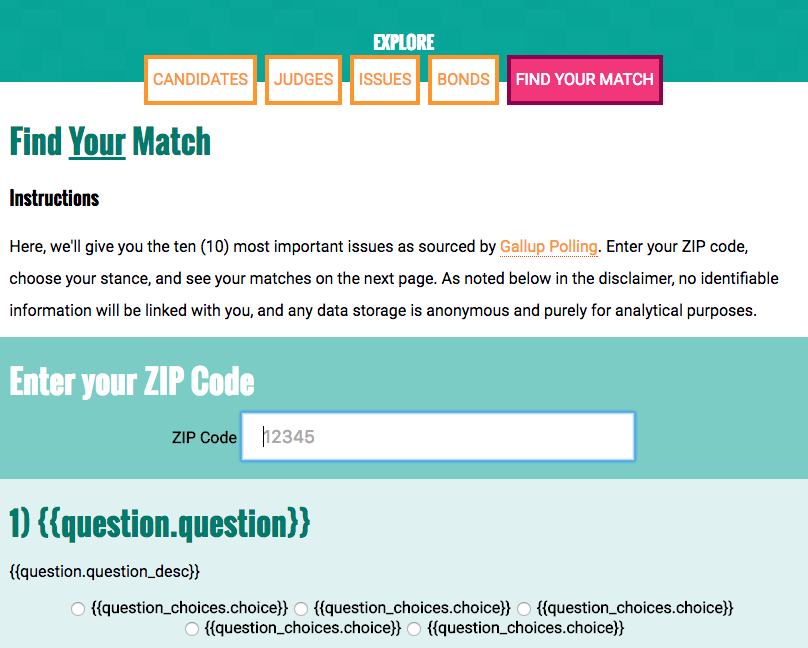
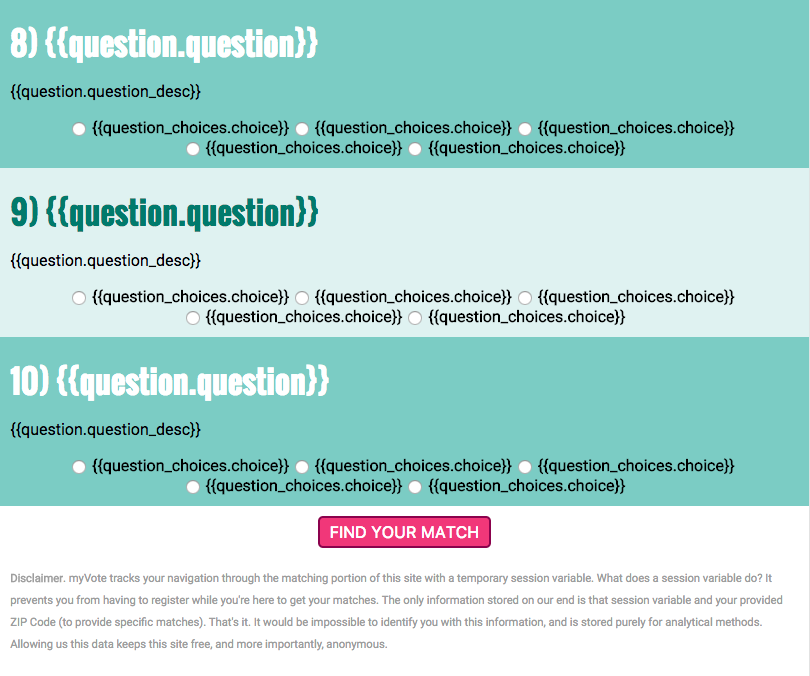
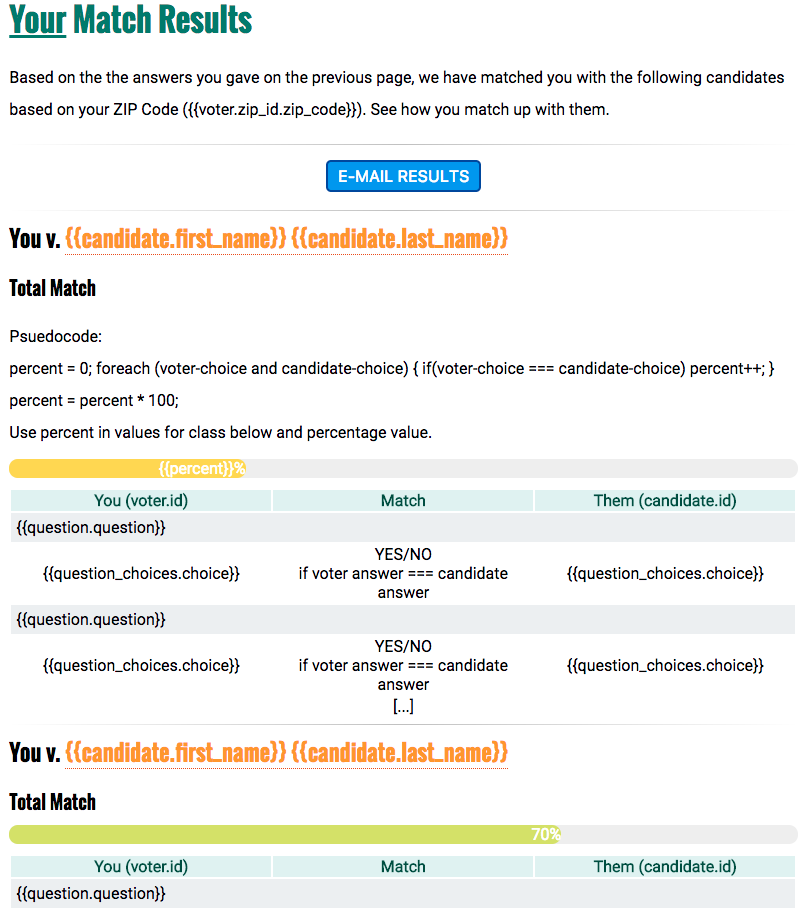




## Similarly for Judges and their recent decisions…

## I’m a User and I want to match my opinions with a Candidate.

The primary purpose of this website is to inform a user’s (voter’s) choices based on their feelings on ten (10) key issues as determined by popular polls like Gallup. When navigating to the *Find Your Match* page initially, the user should be taken through a redirect to create a new session variable (no user interaction), which is stored in the Voter table along with the ZIP code they enter. The user enters their ZIP code, answers questions, and is delivered match results, which they can e-mail to themselves.



## As a User, I am able to see Candidates, Judges, Issues, Bonds, Find Your Match buttons on home page.

Criteria:

1. When User clicks on Candidates, Judges, Issues, Bonds button, user should be able to navigate to separate individual pages
2. When User clicks on Find your Match button page scrolls down highlighting zip-code text field.

## As a User I am able to see zip-code functionality working fine on home page

Criteria:

1. User should see zip-code text field on home page, showing validations if user enters alphabets or numbers more than 5 digits.
2. As user enters zip-code and click enter following division top ten issues will be displayed under the page.

## As a user, I am able to answer the questions related to the present issues in county.

Criteria:

1. Questions related to issues should be displayed with radio button options under zip-code text field.
2. User should be able to answer the questions and following matches with the candidates should be displayed after completing the quiz.

## As a User, I am able to see judges profiles page on clicking Judges button on Home page

Criteria:

1. When user clicks on Judges button user should be traversed to Judges profile page having their decisions related to issues as well.
2. First page will be collapsed and on clicking More button page expands and respective judge profile will appears.

# Testing Plan

We will be using Rails’ internal testing in lieu of *rspec* for any testing we do. Since the site doesn’t require a lot of user input, testing will be limited to the matching portion of the site. If time permits, we’ll build an admin backend to edit entries in the database, and generate appropriate tests, though the need is not that big as this data isn’t as dynamic as other applications might be.

# Gantt Chart

