Zoe Gershuny
OSU CS 406 Projects
Project Plan

Project Title: E-Vote

Introduction:

Due to the pandemic that requires people to stay at home for everyone's safety, it is harder to maintain safety and social guidelines to conduct voting for elections. Not only requiring people to go to their polling location would put them and others at risk of contracting COVID-19, but also, they would have to wait on longer lines, which are already extremely long in certain polling districts. Long lines could impact the voter's ability to allocate their portion of the day to vote and increase the risk of contracting COVID-19. It is seen that in some districts, the wait time has increased because the polling staff had to clean the machines, private booths, tables, pens, and any equipment and supplies used after every person utilizes it.

The US Congress has spoken about absentee and mail-in ballots to provide an alternate option. However, the they could take one step further to make voting accessible for everyone regardless of their location in the United States and in the world. It would be to have a single website for all elections from local to national. It is an opportunity for the country to move forward to take advantage of the power of technology

The project will be an interactive and dynamic database where people will be able to enter their personal information and vote based on their residency for local, state, and national elections for greater participation from American citizens.

On the personal level, it will allow me to make something that is related to the field I work in – government and politics. In addition, I would be able to learn how to implement JavaScript functions and AJAX in an ideal way a for databases to be more modular.

Purpose:

Having a single database would solve the following problems:

For the Voters

1. Online registration to register to vote and change parties and personal information
2. See all the elections they can vote for based on their residency
3. Fairer elections because voting is more accessible from anywhere as long as internet is provided – no need for absentee or mail in ballots which then could be phased out

For the US Government

1. Achieve greater participation from US citizens for those who have…
    a. inflexible hours to go to the polling location
    b. not yet decided to vote early for early voting, absentee ballot
    c. forgotten to put in a request for absentee or mail-in ballot
    d. forgotten about the election(s)
2. Quicker turnaround to announce the results

3. Increase participation from younger and more tech savvy voters

For the scope of this project for CS 406 Summer 2020 Term, not all functionalities for a realistic database will be implemented (i.e. login, create account, user's access, etc but maybe for the next term!).

Tools:

- OSU Server PhpMyAdmin, MySQL
- IDE
    - Option 1: PhpStorm if able to set it up
    - Option 2: Sublime Text
- JavaScript/Node JS (compile on flip with forever)
- AJAX
- Web Design
    - HTML
    - Bootstrap
    - Handlebars
- CS 340 Project Code to rewrite the code for this project's topic and this time implement functions (and AJAX if time remains)
- US Cities Database data dump (modified for attributes' naming consistency): https://github.com/kelvins/US-Cities-Database

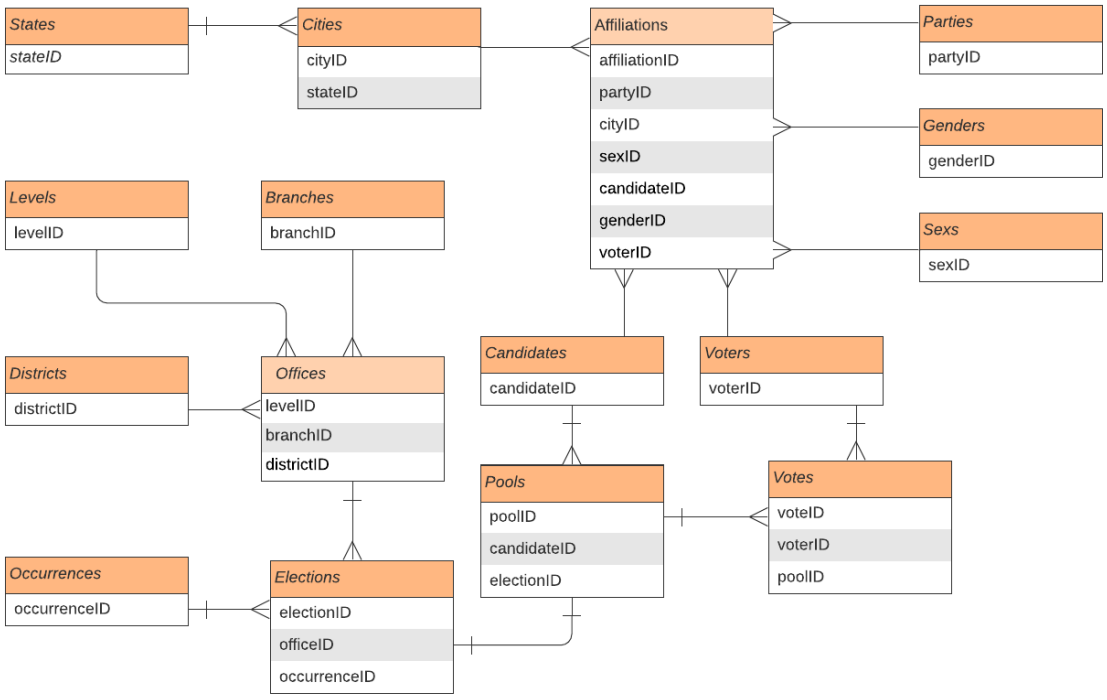
Graphics:

Entities' Description

1. States: list of all the states
2. Cities: list of all the cities
3. Parties: list of all the official political parties
4. Genders: list of all genders that the Candidates and Voters to identify themselves
5. Sexs: list of all genders that the Candidates and Voters were born with
6. Voters: list of all voters who can vote and their personal information
    a. ssn – unique key, each voter has their own social security number that will distinguish them from the rest
    b. it is possible that the voter is also a candidate
7. Candidates: list of all candidates who are running for an office
    a. ssn – unique key, each voter has their own social security number that will distinguish them from the rest
    b. incumbent – Boolean to mark true or false whether or not the person is running for a reelection
    c. it is possible that the candidate is also a voter
8. Affiliations: list of all known Voters and Candidates' affiliations with Cities (and States), Parties, Genders, Sexs
    a. Nullable attributes: candidateID and voterID. Both should not be nullable, but at least one could be nullable
9. Levels: the three levels of government (Federal, State, and Local)
10. Branches: the three branches of government (Legislative, Judicial, and Executive)
11. Districts: holds numbers from 1 to 150 (currently known as the state/city has the most districts)

12. Offices: information about the office position
    a. branchID is nullable – some positions are elected and not within the three branches (i.e. sheriff)
13. Occurrences: the type of the Election (primary, general, special, runoff)
14. Elections: information about the election
15. Pools: information about who is running in which election
    a. choiceID: a number designated to the candidate for that specific election to be used in counting votes
16. Votes: information about which election and who did the person vote for

ERD



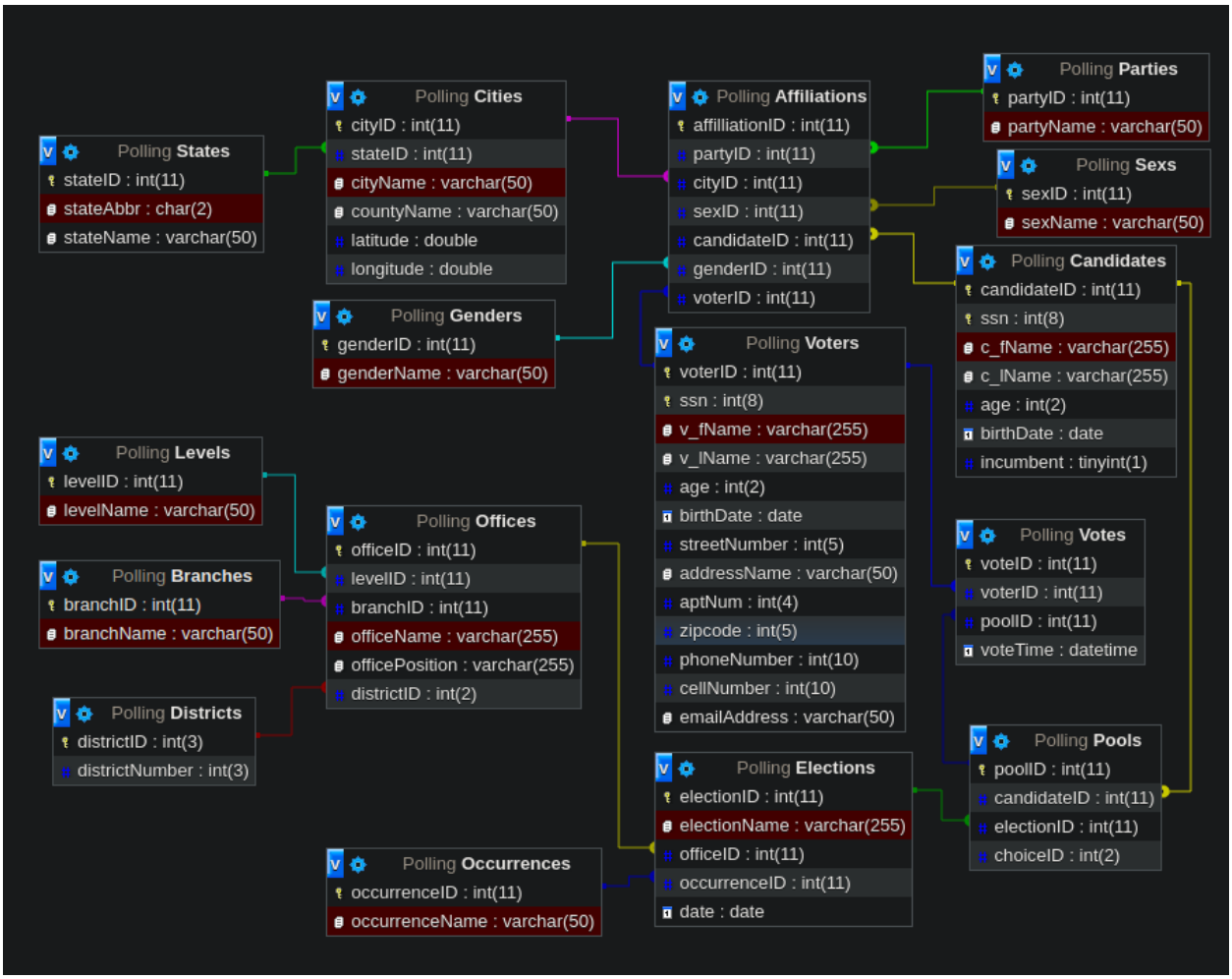
The image above is the Entity-Relationship Diagram that shows the relationship between entities.

1. There are many Cities in a State
2. Affiliations entity is a junction table for M:M relationships for the following tables
    a. Cities
    b. Parties
    c. Genders
    d. Sexs
    e. Voters
    f. Candidates
3. A Candidate can be in one to many Pools
4. A Voter can have zero to many Votes
5. A Pool will have zero to many Votes
6. Offices entity is a junction table for M:M relationships for the following tables

a. Levels
b. Districts
c. Branches

7. Each office has many Elections
8. Each election has one Pool (of candidates)

SCHEMA



The image above is the schema is the relationship diagram showing all the attributes, keys, and relationships between tables.

Entity	PK	FK	Relationship(s)
States	stateID	None	- 1:M stateID as a FK inside Cities
Cities	cityID	stateID	- 1:M with cityID as a FK inside Affiliations - M:1 with States with stateID as a FK inside Cities
Genders	genderID	None	- 1:M with genderID as a FK inside Affiliations
Sexs	sexID	None	- 1:M with sexID as a FK inside Affiliations
Candidates	candidateID	None	- 1:M with candidateID as a FK inside Affiliations
Parties	partyID	None	- 1:M with partyID as a FK inside Affiliations

Affiliations	affiliationID	partyID cityID sexID candidateID genderID voterID	- M:1 with Cities, Genders, Parties, Sexs, Candidates, and Voters with cityID, genderID, partyID, sexID, candidateID, and voterID respectively as FKs in Affiliations
Voters	voterID	None	- 1:M voterID as a FK inside Affiliations and Votes
Levels	levelID	None	- 1:M with levelID as a FK inside Offices
Branches	branchID	None	- 1:M with branchID as a FK inside Offices
Districts	districtID	None	- 1:M with districtID as a FK inside Offices
Offices	officeID	branchID levelID districtID	- M:1 with Levels, Branches, and Districts as levelID, branchID, and districtID respectively as FKs in Offices - 1:M with officeID in Elections
Occurrences	occurrenceID	None	- 1:M occurrenceID as a FK inside Elections
Elections	electionID	officeID occurrenceID	- M:1 with Offices and Occurrences with officeID and occurrenceID as FKs respectively in Elections - 1:1 with electionID as a FK in Pools
Pools	poolID	candidateID electionID	- 1:1 with Elections with electionID as a FK in Pools - 1:M with Candidates with candidateID as a FK in Pools - 1:M with Votes with a poolID in Votes
Votes	voteID	voterID poolID	- M:1 with Voters with a voterID in Votes - M:1 with Pool with poolID in Votes

Work Schedule

Due Date	Tasks
July 14th	- Set up the database backend, including filling with sample data - Write SQL queries to be used - Design the static pages. When completed, create the handlebars template - Create files for JavaScript functions (i.e. delete, update)
July 21st	- Implement (connect front-back) the home page - Implement display results and dynamic dropdown functions
July 28th	- Implement add/delete voter/candidate with validations - implement a search page to see affiliations
August 4th	- Implement search with a portion of a text (ie “Josh” will return with all results that has Josh in it like “Joshua”) - Implement any functionalities if time available
August 15th	- Finish up what was not completed before

Conclusion:

The E-Vote database will help to solve three issues: 1) wait time at the polls, 2) safety and social distancing guidelines and 3) increase voter's participation. Not only that, the government could move forward to streamline their system. Not only it would help to protect everyone's health but also it would move the country forward. It also helps me as I create a project that is in the area I work in and to build onto my JavaScript knowledge on best implementing it for databases.