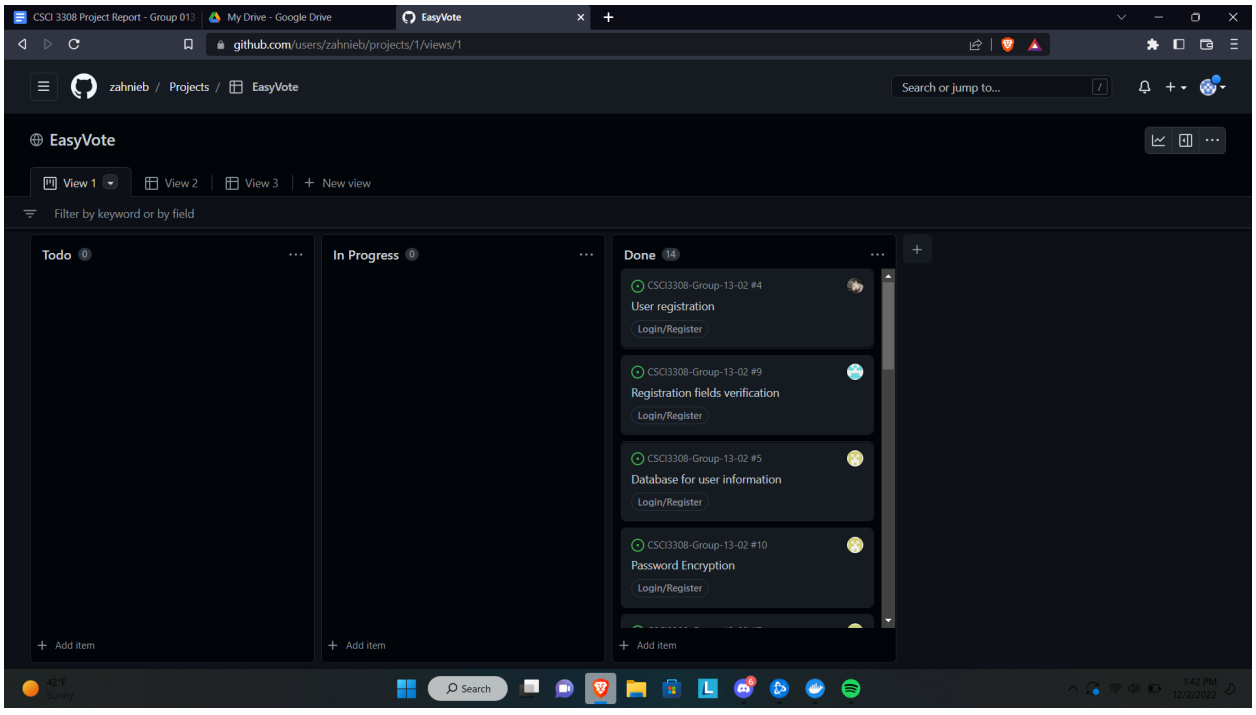
CSCI 3308 Project Report - Group 013-2

Title: EasyVote

Who: Tome Dudanov, Zahir Nieblas Bedolla, Matthew Etter, Jason Swartz

Project Description: Democracy - why should I vote?
Our group is passionate about Democracy and the right to vote in the United States. EasyVote helps you find the nearest polls to you. But, how? When you register an account on EasyVote, you will be asked for a username, a password, an address, state, and most importantly, ZIP code. You must input these to be able to access EasyVote. The app then utilizes Google's Civic Information API to find all the nearest voting polls to you. Whatever results are returned, are displayed in the "Where Can I Vote?" page, sorted by distance from nearest to farthest in your state. This filtering is done primarily by the ZIP code, which is why that's the most important thing we require when logging in.

Project Tracker: https://github.com/users/zahnieb/projects/1



Video:
https://www.youtube.com/watch?v=QzWQF9fgPDA

VCS:

https://github.com/zahnieb/CSCI3308-Group-13-02

Contributions:

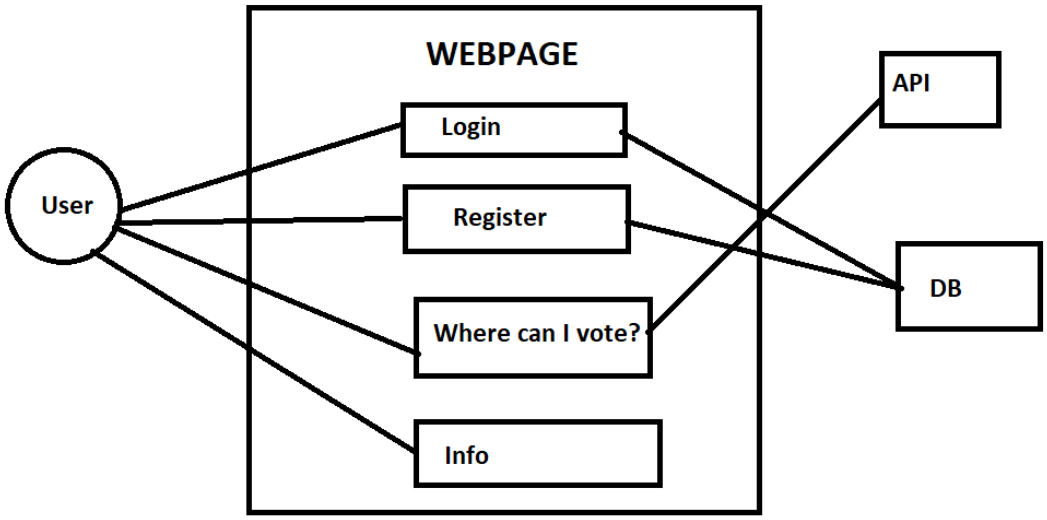
Tome Dudanov: Tome was our front-end developer. He was in charge of what the finished product looked like. He made all of the EJS files, and contributed heavily to the styling of the webpage. After that was done, he contributed to some back-end functionality, like displaying error messages when errors occurred during login and register, and rendering the information page with a variable that keeps track of whether or not the user is logged in, so that we could have a dynamic navbar.

Zahir Nieblas Bedolla: Zahir was in charge of back-end API calls to the Google Civics API and getting that data from the back end to the front-end successfully. Implemented PostgreSQL database design as well as set-up and initialization of database and environment for docker to easily use the environment for development of our web application. Also assisted in redirects and rendering of pages.

Jason Swartz: Jason's main role was implementing the register functionality and the settings page, where an user can change their address or password. He also designed the password encryption functionality and helped with the login design.

Matthew Etter: Matt's main role was support through helping work out the login page backend, updating the readme file, organizing files into the correct places, and taking meeting notes.

Use Case Diagram:



Test Results:

Team testing:

- Sign-in Recognizes username
    - Integration Testing
    - User Testing
    - Test Cases:
        - If an user inputs an existing username and matching existing password.
        - If an user inputs an existing username but incorrect password
        - If an user inputs a username that does not exist, regardless of the password
    - Test Data
        - Usernames, Passwords from SQL Database
        - Username and password entered in from user
    - Test Environment
        - Docker/PostgreSQL for database holding usernames and passwords
        - Localhost
    - Test Results
        - User will successfully be logged in and redirected to the Polls page.
        - User will not be signed in, rendering a message of incorrect username/password.
    - User Acceptance Testers

        - Other students in CSCI3308

*All aspects of the sign-in page worked as intended.

- Registration
    - User Testing
    - Test Cases:
        - If an user enters all valid credentials
        - If an username is taken
        - If a password is not long enough, too long, or does not contain the correct amount of caps/numbers/special characters
        - If an address is invalid
    - Integration Testing:
        - Password,username & address filled out.
        - Username & Password set criteria of minimum 8 characters
        - Username not in use, then successful registration
        - Username in use, then message of username in use, try again.
    - Testing Data:
        - Username, Password & address entered in by user
    - Test Environment
        - Docker/PostgreSQL for database holding usernames and passwords
        - Localhost
    - Test Results
        - Successful registration
        - User is not able to register
    - User Acceptance Testers
        - Other students in CSCI3308
- Showing Nearby Polls
    - Testing for successful API calls dependent on user address input.
    - Testing Environment
        - Mocha/Chai
        - Docker
        - PostgreSQL
        - Localhost
    - Test Data
        - User address, API call
    - Test Results
        - 200 return api call, successful
    - User Acceptance Testers

- Other students in CSCI3308

*All aspects of the Registration page worked as intended

Outside Testing:

Overall, the testers really enjoyed our website and found the functionality very useful. They said they would definitely use our website for a future election. When using the website users first clicked on the register link below the sign in card on the first page that is shown (login). The users then registered with a username and password as well as their home address. After registering the users both logged in and were brought to the “Where can I vote page” where they viewed all nearby polling stations. Due to the issue of there not being an election at the moment I did have to explain that the addresses and hours of operation would not appear because in the eyes of the google API the polls did not exist. They were very understanding that in future elections the website will operate with full functionality during the voting season. One user who was very intrigued used our settings tab and tried changing the address to see if the website would work in Los Angeles as well and was very surprised to see that it does. They stated that, “it’s nice to have a poll finder that works anywhere because a lot of times you have to use a government site that can only be used for your area.” The same user also used the password changing feature and liked how simple it was but did bring up the question of security stating, “I wish there was two way authentication through email or text so someone can’t just change my password without me having to approve it.” Both users also tested if incorrect passwords or old passwords let them in and were happy to see that they don’t. The users enjoyed the website overall but there were many critiques that have been noted and if there were a future update would definitely be included. For simplicity I will be listing out the likes and dislikes of users below:

Tester J:

Likes:

- Really enjoy the idea of finding all poll locations.
- Very seamless, easy to use.
- Tabs look great and are very easy to use.

Dislikes:

- Did not enjoy the font because it reminded them of scam sites.
- Wants better security, two way authentication preferred.
- Wants a better sized background image to fill the page and not look as pixelated.

- Too much white on the page.
- Says we should put separators between each result.

Tester K:

Likes:
- Found functionality very helpful.
- Enjoyed the changing addresses feature.
- Overall the page looked nice and clean.
- Liked that it shows the different things to vote for within the election.
- Liked how it lists the districts to make sure she goes to the correct one.

Dislikes:
- Did not like the image as the background of the website.
- Wishes the settings page had the information inline with the category, for example (Username: TesterK) opposed to the TesterK being below Username.

*Both testers are not CSCI students and have no CSCI backgrounds.

<u>Deployment:</u>
http://csci3308.int.colorado.edu:49164/login