The Youth Vote Project

Alexandra Lavin | Jemilla Scotten | Joanne Monaghan | Rana Khoury | Susan Bailey

What are you building?

We are building a website that will allow young people to take part in voting and elections. They will be able to cast their ballot and see the results of the youth vote nationally, as well as by their school or region. There will be two main users and focuses; educators wanting to register their establishment to take part in the project, and young people using the platform to vote.

The educator's user journey will consist of: navigating to the website, choosing to register a school/register an interest in taking part, selecting their school, selecting how many participants they will have from their school, confirm all of the details, and then they will be issued with a list of unique poll card numbers to give out to their students.

The young person's journey will be: navigate to the website, select 'I want to vote', enter their poll card number, enter their age, select the party they want to vote for, confirm their choice, and cast their ballot.

Anyone will be able to access the website to see the results and use the educational/external resources, but a user must have a unique poll card number in order to be able to vote.

What does it do or what kind of problem does it solve?

Voting is habit forming and voter turnout between 18-26 year olds is the lowest compared to all other age categories. In places where the voting age is 16, 16-18 year olds vote more than their older peers (18-24 and 25-34) showing that if given the opportunity young people want to be involved in democracy. If we can give 11-17 year olds the opportunity to vote, discuss, and see the value of their vote, the hope is this will inspire and engage them in democracy and they will continue to vote in subsequent elections.

References:
https://www.electoral-reform.org.uk/wp-content/uploads/2017/06/Votes-at-16-briefing.pdf
http://ukandeu.ac.uk/votes-at-16-what-effect-would-it-have/

What are the key features of your system?

Front End

Home page
Brief information about the Youth Vote project with two main user journeys to choose from; school registration page and a voting page.

School registration / Poll card request page:
Use JSON file or API to have a drop down/searchable list of schools. Confirmation of choice and then the option to enter the number of poll card numbers (poll card id) they require with an algorithm generated in the back end. A confirmation step before submission will appear.

A printable/downloadable list:
Once the number of poll cards is confirmed, a printable/downloadable list will be generated for the person requesting the information. The poll card ID can be made up of a school ID, constituency ID or postcode ID, plus the unique voter ID.

Voter page:
Use the poll card ID number to login and enter age (for results after the election). They then can vote for one of the main parties. No other personal info needed with confirmation before.

Election info page:
Find your local poll station, your local candidates. (via API democracy club). Information about the main parties. This is not part of the MVP (minimum viable product).

Election results page:
National cumulative results are available to view. With an extension of restricting results by school/age/area.

Back End

Algorithm:
Algorithm to create a unique poll card ID made up by using school ID/constituency or postcode ID, and a unique voter ID.

JSON file / API:
JSON file, populated from the department of Education with a list of schools and their addresses or an API to retrieve schools information.

Potential APIs:
https://democracyclub.org.uk/projects/data/ : shows local candidates, local poll station
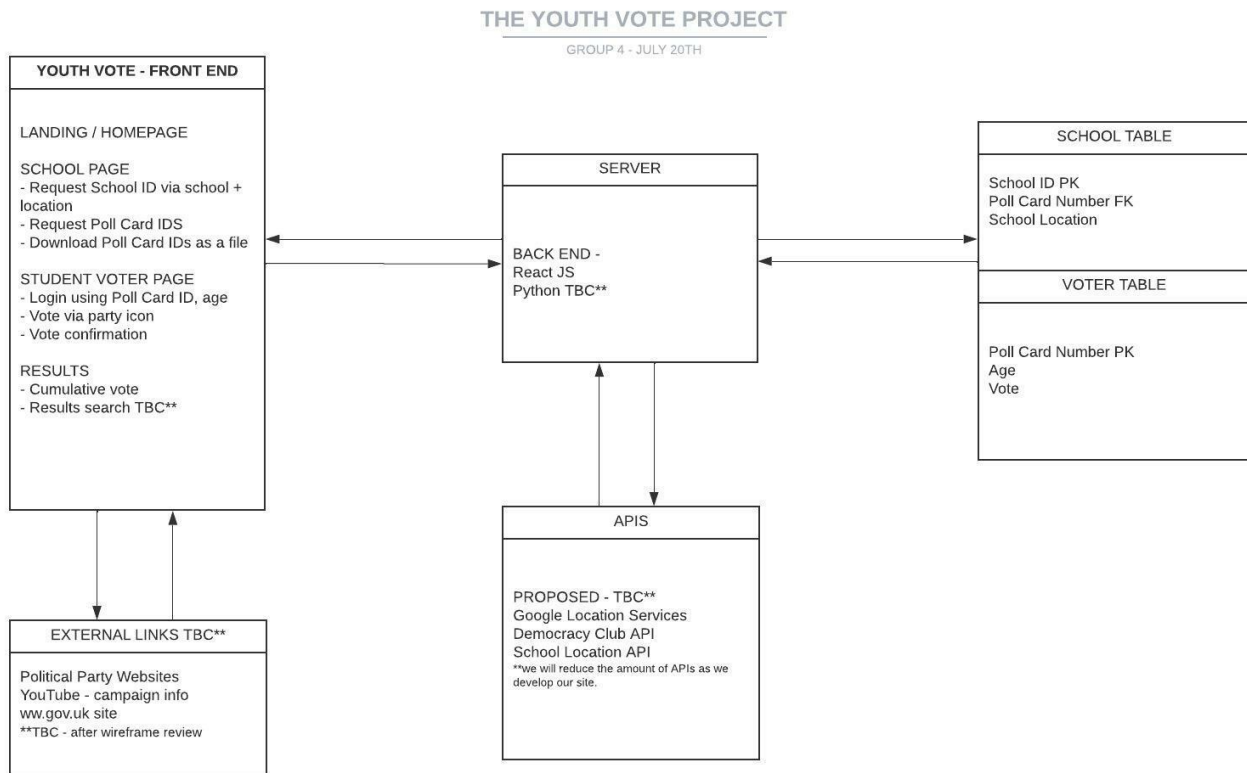https://www.theyworkforyou.com/: get constituency/mp
https://members-api.parliament.uk/: election results, candidates, mp interests/information

Databases
- Schools table with requested poll card numbers (based on the above), postcode/constituency
- Voter table with poll card ID, party voted for, constituency, school

Provide a sample architecture diagram of your system

THE YOUTH VOTE PROJECT

GROUP 4 - JULY 20TH

YOUTH VOTE - FRONT END

LANDING / HOMEPAGE

SCHOOL PAGE
- Request School ID via school + location
- Request Poll Card IDS
- Download Poll Card IDs as a file

STUDENT VOTER PAGE
- Login using Poll Card ID, age
- Vote via party icon
- Vote confirmation

RESULTS
- Cumulative vote
- Results search TBC**

SERVER

BACK END -
React JS
Python TBC**

SCHOOL TABLE

School ID PK
Poll Card Number FK
School Location

VOTER TABLE

Poll Card Number PK
Age
Vote

APIS

PROPOSED - TBC**
Google Location Services
Democracy Club API
School Location API
**we will reduce the amount of APIs as we develop our site.

EXTERNAL LINKS TBC

Political Party Websites
YouTube - campaign info
ww.gov.uk site
**TBC - after wireframe review

Describe the team approach to the project work: how are you planning to distribute the workload, how are you managing your code, how are you planning to test your system.

The workload will be divided **between front-end tasks** (HTML, CS, React…) and **back-end tasks** (server, API, database…), according to the team members strengths.

- 2 team members on the front-end.
- 2 team members on the back-end.
- 1 team member on both front-end and back-end (linking both)

Several weekly meetings are being organised to monitor each team member progress of the project, discuss the next steps and support any team member encountering challenges.

We share a **Google Drive** folder where project's documents are saved and accessible to every team member and we also have a **Trello board** that helps us track everyone's progress.

Once the project has been fully designed (features, architecture and wireframing), **all main tasks of the project will be distributed to each team member** as well as a **project timeline** to help us keep on track (meeting to deliver the main tasks for front-end/back-end taking place on the 23rd of July).

Finally, once we have a good code base, we will **share our codes on a GitHub repository** and start **linking the codes together** before some of the testing.

Regarding testing, **unit testing** will be done alongside the code being created and once the codes are merged together, we will proceed with a **systems integration testing**. The last testing phase will be done with a **user acceptance testing**.