THE YOUTH VOTE | PROJECT WORK DOCUMENT
GROUP 4

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

INTRODUCTION:

The Youth Vote: teaching and engaging 11-18 year olds to get involved in elections and democracy.

Young people aged 18 - 24 vote in a lower proportion than all other UK age groups according to the Electoral Reform Society. Statistically, if a young person does not vote in their first election they are significantly less likely to vote in future elections, meaning the issues and concerns of young people are not being heard.

Politics is not part of the national curriculum in schools, and a lack of knowledge contributes to why young people don't vote. The Youth Vote project gives educators a platform to introduce voting concepts to young people via a simple and accessible site, offering students opportunities to discuss, debate and become active citizens.

REPORT ROADMAP:

- **Background**
- **Objective**
- **Project Roadmap**
- **Specifications and Website Architecture**
- **Wireframes and Design**
- **Implementation and Execution**
- **Testing and Evaluation**
- **Conclusion**

BACKGROUND:

Our website allows young people to take part in voting and elections in a simple, easy to use format. They are able to cast their ballot and see the results of the youth vote nationally. There are 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 is able to search and select their school from a list, select how many poll cards they will need for their school, confirm their details, and then be issued with a list of unique poll card numbers to give out to their students.

The young person is able to select on the website '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.

Finally, all users may access the site to immediately see the national cumulative vote. Furthermore, the results are accessible and can be viewed by anyone (non-educators and non-young-voters), but a user must have a unique poll card number in order to be able to vote.

OBJECTIVE:

Voting is habit-forming: when young people learn the voting process and vote, they are more likely to do so when older.

Our objective was to produce an attractive, age-appropriate site to enable users to conduct a simple mock election, giving educators a tool to inform young people about voting, and young people a place to learn voting fundamentals. In turn, we aim to spark interest and discussion about politics. We hope to inform and inspire young people to engage with further elections in real life, developing life-long skills.

PROJECT ROADMAP:

The below project roadmap shows our division of each epic, allocating one week per team for each task. Stories were divided between the teams according to team availability and skillset.

JULY 18TH	JULY 25TH	AUGUST 1ST	AUGUST 8TH
Wireframes			
Planning DB strategy	Architecture + UI		
	Build DB, local API		
	React HTML / CSS Skeleton	API, Results Algorithm	
		Build & Integrate Components	Integration Unit Testing
			Front-End Integration user / UI testing

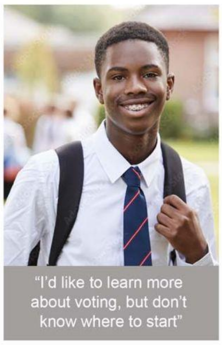
SPECIFICATIONS AND DESIGN:

User personas:
Our aim is to capture a broad demographic of educators and young people, with two key considerations:

- Educator participation is likely to be out of school, or compressed within the PSHRSCE curriculum.
- Disenfranchised young people may be dealing with other factors such as digital poverty, educational or special needs.

Our site needs to be simple enough to understand and use in time-conscious settings, easy to use on both mobile and desktop formats and age-appropriate. It should be presented in a simple, fun way and give immediate feedback at every stage.

YOUTH VOTER:


"I'd like to learn more about voting, but don't know where to start"

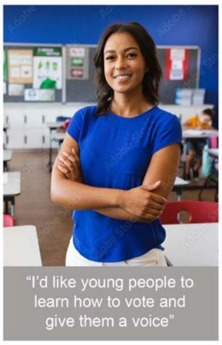
SECONDARY SCHOOL STUDENT 13-17

Goals :
To be introduced to voting and some of the terminology around voting e.g.pollcards and names of political parties.

Frustrations
"There are too many special terms, I don't understand it."
"We haven't really learned about politics in school."
"Politics and religion cause a lot of arguments."
"I don't feel informed enough and when I try there's too much to take in."

Motivations
"I will be able to vote in a few years, and I don't want to feel uninformed."

YOUTH VOTE ORGANISER:


"I'd like young people to learn how to vote and give them a voice"

SECONDARY SCHOOL TEACHER

Goals :
Our educator would like to get more young people engaged in politics, and for them to see how their vote gives them a voice. If more younger people vote, politicians will give more consideration to the challenges facing young people.

Frustrations:
"When I discuss politics with young people, they believe politicians don't care about the issues young people face. Politics and the political system is overcomplicated."
"Many young people start adult life lacking basic knowledge about how they can participate in politics."

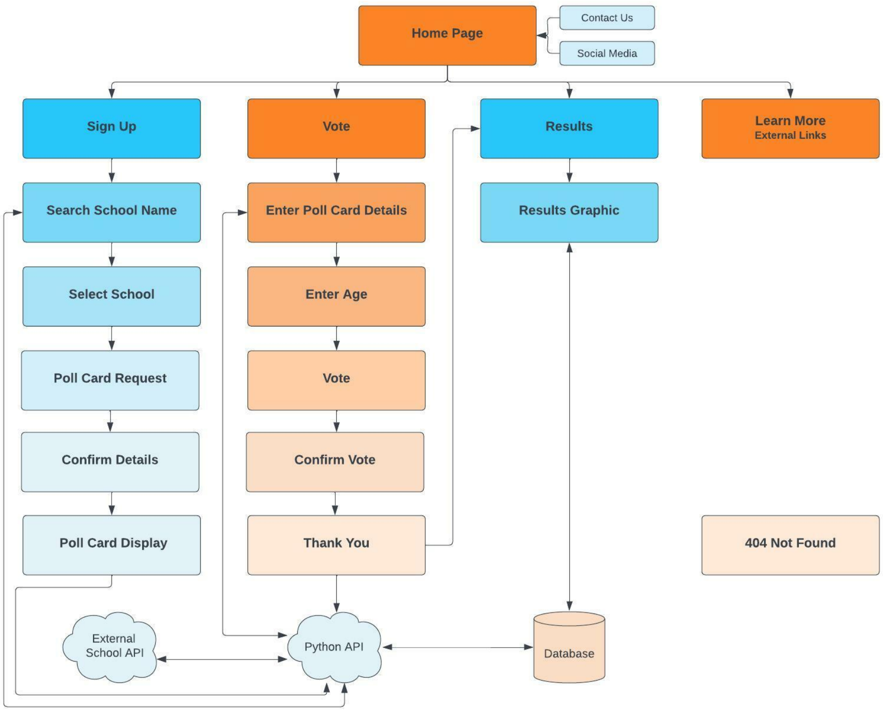
Motivations:
"I want to teach my students basic voting terminology and conduct a simple election with their classmates to set them on a path to become regular voters in adult life."

User Stories:
With the above research we can create two user stories -

- Youth Voter Elijah - *"As a young person I want a simple way of seeing how voting works, so I can go through the process and see the results when my class or school votes."*

- Youth Vote Organiser Miss Kenny - *"As head of year 9 I'd like to organise a simple election, giving me an opportunity to discuss voting terminology so that students can participate and vote. I'd then encourage the students to follow the cumulative results so they can observe how their votes affect the outcome."*

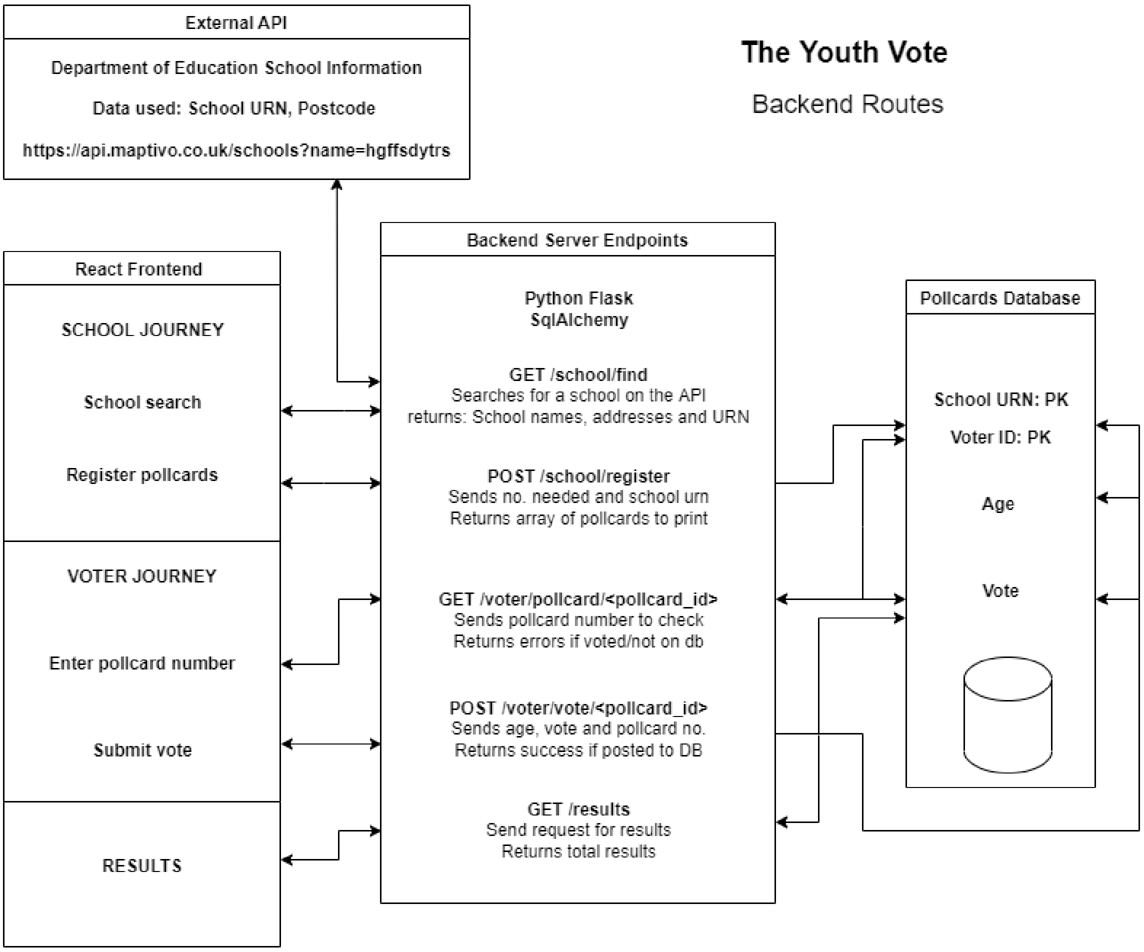
Website Architecture:



The site is made up of 5 keys pages: the **home page**, the **school sign up page**, the **voting page**, the **results page** and the **learn more page** for external resources where people can learn more about voting and democracy.

The main objective of the website is to encourage young people to vote, therefore the key navigation will be directed towards the young person's user journey - voting, checking the results and learning more.

System Architecture:



External API
Department of Education School Information
Data used: School URN, Postcode
https://api.maptivo.co.uk/schools?name=hgffsdytrs
The Youth Vote
Backend Routes
React Frontend
SCHOOL JOURNEY
School search
Register pollcards
VOTER JOURNEY
Enter pollcard number
Submit vote
RESULTS
Backend Server Endpoints
Python Flask
SqlAlchemy
GET /school/find
Searches for a school on the API
returns: School names, addresses and URN
POST /school/register
Sends no. needed and school urn
Returns array of pollcards to print
GET /voter/pollcard/<pollcard_id>
Sends pollcard number to check
Returns errors if voted/not on db
POST /voter/vote/<pollcard_id>
Sends age, vote and pollcard no.
Returns success if posted to DB
GET /results
Send request for results
Returns total results
Pollcards Database
School URN: PK
Voter ID: PK
Age
Vote

Database, Server & API Connection - We used Flask and SQLalchemy to connect the database with the server and the server to the external API.

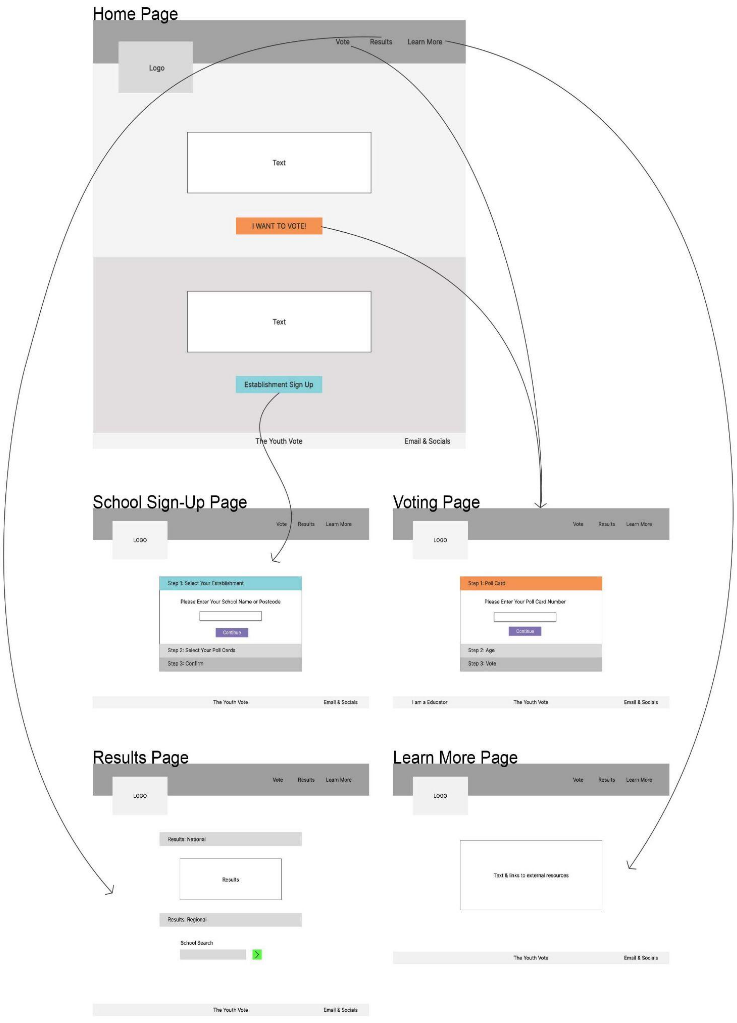
Frontend & Backend Connection - We used the Axios library in order to send requests between the backend and the frontend of the website.

APIs - The website data will be generated by two APIs: the Department of Education API (DoE) and a local Flask/Python API. The localFlask/ Python API is connected to the database..

API endpoints - We have several endpoints:

1. GET school/find - The DoE API is requested to search for a school via postcode or name and return the right list of schools according to the user input.
2. POST school/register - Once the right school is selected and the number of poll cards are added on the frontend (user input), the Flask API takes the school urn, the number of pollcards requested, to then generate the right number of voters ID, to finally register school urn and voters IDs in the database accordingly. When the data is added to the database, the Flask API then merges school urn with voters IDs to create unique poll cards numbers that are being returned on the frontend as a list that the school can copy and save.
3. GET vote/pollcard/<pollcard_id> - Young voters will input their unique poll card number on the website to vote. The Flask API checks if the poll card number exists in the database and also if the poll card number has already been used for a vote.
4. POST voter/vote/<pollcard_id> - If the poll card number does exists and has no vote registered yet, the user can enter their age and vote on the frontend (user input) and then the poll card number, the age and the vote are sent to the Flask API, in order to register the vote into the database.
5. GET results - Finally to display the national results on the website, a request is made to the database, however in the future, the way the database has been built, we could display results for individual schools.

WIREFRAMES:



To the left is the site overview wireframe. Please refer to Week 2 Homework for a more comprehensive overview of the site.

DESIGN:

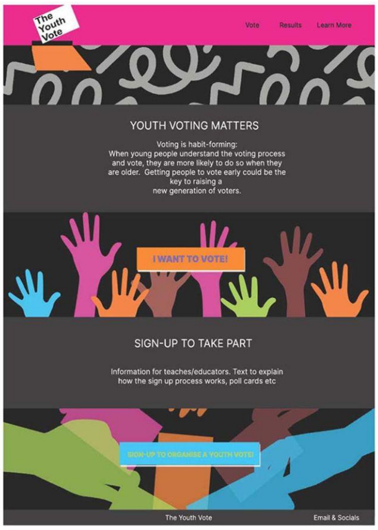
Below is the site design. We have chosen a dark colour palette contrasted with bright colours, this gives a youthful and fun feel to the site which is aligned with our project aims.

We have also considered how the site will look on both desktop and mobile screen sizes, using features such as burger menu to save space while maintaining our design on a smaller screen.

We have chosen a simple sans-serif font for AAA accessibility.

We have taken into consideration the Nielsen 10 general principles for interaction design.

By adding drop shadows to buttons we are indicating to the user this is a button offering 'visibility of system status’ and ‘matching the real world system’. User control is offered by clear destinations and CTA throughout, showing the user alternative journeys and back button at each stage. For further heuristic analysis, please refer to our Week 2 Homework document.

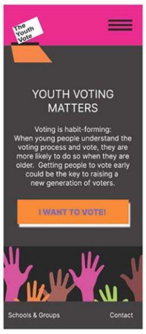


YOUTH VOTING MATTERS

Voting is habit-forming: When young people understand the voting process and vote, they are more likely to do so when they are older. Getting people to vote early could be the key to raising a new generation of voters.

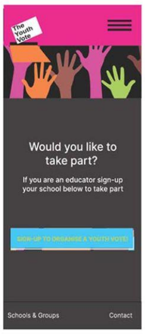
SIGN-UP TO TAKE PART

Information for teaches/educators. Text to explain how the sign up process works, poll cards etc



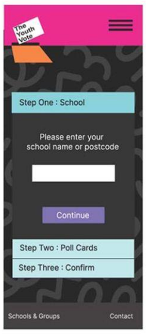
YOUTH VOTING MATTERS

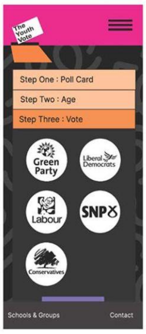
Voting is habit-forming: When young people understand the voting process and vote, they are more likely to do so when they are older. Getting people to vote early could be the key to raising a new generation of voters.

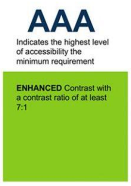


Would you like to take part?

If you are an educator sign-up your school below to take part

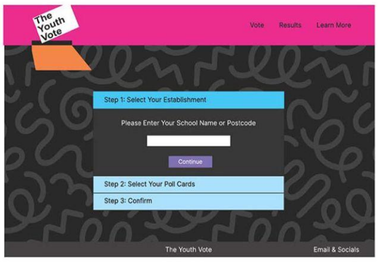


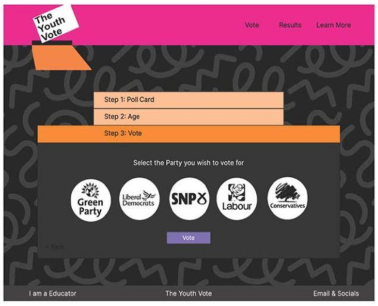


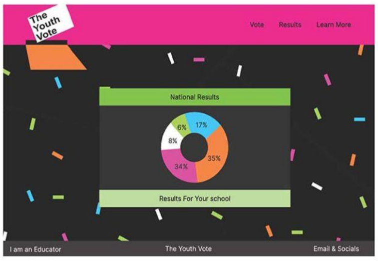

AAA
Indicates the highest level of accessibility the minimum requirement
ENHANCED Contrast with a contrast ratio of at least 7:1

Text colour : #000000 Hex Black
#F5F5F5 Whitesmoke

Background: #000000 Hex Black
#282C34 Grey
#88702 Orange
#00CCFF Blue
#ED028C Pink


The Youth Vote
Vote
Results
Learn More
Step 1: Select Your Establishment
Please Enter Your School Name or Postcode
Continue
Step 2: Select Your Poll Cards
Step 3: Confirm
The Youth Vote
Email & Socials





TECHNICAL AND NON-TECHNICAL REQUIREMENTS:

Non-functional requirements:

We have taken into consideration web accessibility, making sure our design is a minimum of AA compliant, and in places where text is descriptive and important to the user journey aiming for AAA compliance.

We have adopted a ‘Mobile First’ design strategy, anticipating many users will access the site via their mobile and adjusting our design to include burger menus and adding breakpoints to our css to control how we want our site to appear. Our site supports all major browsers.

Authentication and authorisation:

We have a separate pathway for schools to request poll cards. The present MVP iteration is designed for an educational setting rather than a public one therefore at the present stage authentication is low priority. We will review this for future iterations, should the app become a national campaign.

When authorising voters, we have ensured the correct poll card number has to be entered to vote and also that the poll card number can only be used once in the backend.

Availability:

The website will be available in its MVP form continuously, the timing of elections and voting can be decided by the user.

Future iterations of this site should create 'realistic' voting conditions, i.e having limited windows of availability when our ‘polling station’ is open, in line with national / local elections as they happen in real life.

Data Quality:

The database is as small as possible, only retaining essential information. We are only storing the school URN (the unique reference number given to each school by the Department of Education) and using our external API to retrieve any other information. The external API returns information from the Department of Education so is likely to be accurate and up-to-date. We are also generating a voter ID which is then merged with the schoolURN to create a unique poll card number for each voter. Finally we have a column for the voter’s age and vote.

Privacy:
For security reasons and database size we are not taking any information from individuals other than age and their individual vote. This allows the vote to be anonymous, while being able to display results per school and per region using the school URN.

Human Error:
Inputs in the front end have catches for human error and will not allow the user to proceed without entering data in the format we require. Error messages can appear or pop up to alert users they are using the wrong format, and request they enter the information again. API results have inbuilt error codes and messages to alert they are working correctly.

Maintainability:
The site uses widely available frameworks and software, with clearly structured files and code commentary (where required). The team believes the site can be readily maintained by both internal and external teams.

Performance:
The longest anticipated load time on the site is when requesting from the DoE API, especially if it generates many matching results. In this scenario the rest of the page is rendered and a loading message is displayed. This allows the user to know that it is working and will be able to check details shortly.

IMPLEMENTATION AND EXECUTION:

Team roles:
Rana Khoury - Scrum Master / Integration Front End & Back End
Jemilla Scotton - Product Owner / Front End Lead Dev
Joanne Monaghan - Back End Lead Dev / Unit Testing
Alexandra Lavin - Back End Dev
Susan Bailey - Front End Dev / Design / User testing

Tools and Libraries:
Python, Flask, flask_sqlalchemy, SQL alchemy, flask_cors, requests, json, Unittest
React (including hooks useState and useEffect), Axios, Material UI, Bootstrap, React Router DOM, Framer Motion.

Implementation Process - Challenges:
It has been a challenge to retrieve information from the database and to connect the backend server to React. Adding pollcards to the database required several re-works - we had to find a way of adding poll cards on the users request. We initially had two tables and we were auto incrementing them within SQL but this proved a challenge when implementing this with the combination keys.

We made the decision to simplify the database to one table with five columns and create the rows in Python. We then had the challenge of allowing schools to request poll cards multiple times, so the new poll cards numbers would add up to those previously created.

Displaying the responses from the API proved to be a challenge on the front end and we needed to include asynchronous loading as part of our Axios request in order to be able to store and display data.

Implementation Process - Achievements:
Roles were allocated according to each team member's strengths which meant we were able to start the project very quickly, allowing each team member to focus on their allocated story without overlap of roles.
The biggest achievement is also the biggest challenge: getting each part of the site connected was the hardest but most satisfying task.

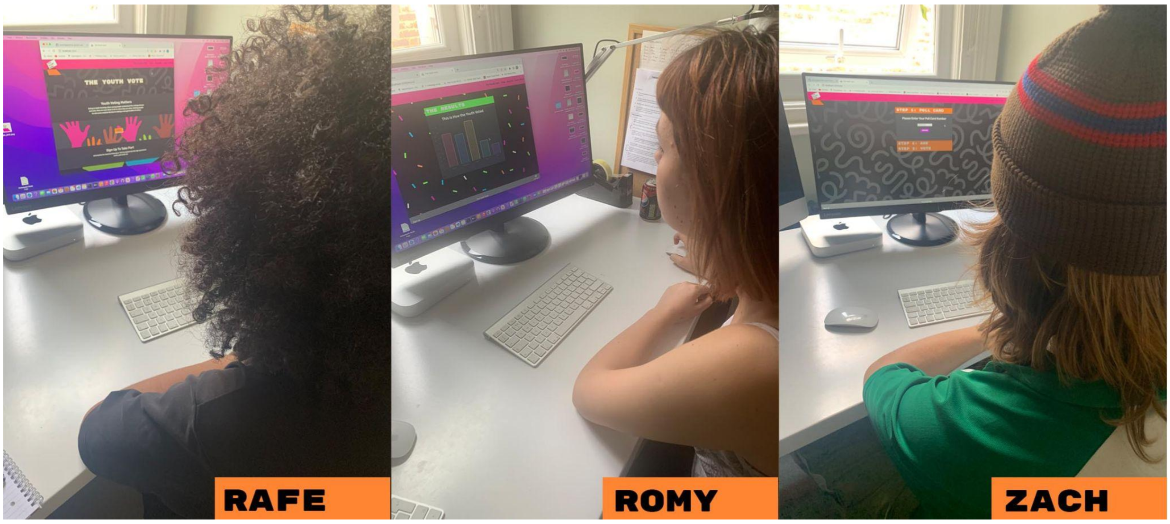
Agile Development:
We divided the team and Epics at an early stage and then met for stand ups a minimum of 4 times a week to decide distribution of stories. We used Trello for stories, a shared Google Drive for notes, and Slack for day-to-day communication. Code reviews were conducted during pull requests.

TESTING AND EVALUATION:

Testing strategy:
We have conducted unit testing on all the back end functions, and functional testing on the site as a whole. The website was tested to check every user scenario and make sure errors were handled properly by the backend and the frontend.
For future iterations we would like to conduct database functional testing. We would also like to check the school URN from the DoE API to flag any errors occurring because the external URN changed. When the new school year starts in September we would like to conduct user testing to inform our next iteration.

System Limitations:
At present our site offers a limited number of results options. Future iterations should offer breakdowns of results by school, constituency, local and national elections. Currently it is possible to vote at any time, and we believe the open status of the vote does not completely deliver a true voting experience. We would like to open and close our ballot box to introduce the time-conscious nature of voting to young people.

User Testing:
We tested our prototype on three young people aged 15-16.



All three users were given the following context and questions: “Your teacher has asked you to participate in a mock election, and given you poll cards to vote. Please give feedback on how easy you find the site to navigate and if you can achieve your aim of voting.”

All users gave enthusiastic feedback about the design and site appearance. They found it easy to navigate and locate the vote page, and liked being directed straight to the results page. All gave positive feedback on site navigation and felt it was simple and appealing to use. All were keen to see cumulative results by school - they wanted to see how their peers would vote.
All users requested larger ‘vote’ and ‘sign up’ buttons on the homepage for additional clarity for their user journeys.
All users requested more information on the results page e.g. to make the party names larger and wanted context on the numbers on the chart.
More detailed feedback is available within our user feedback document.

CONCLUSION:

We believe we have delivered an excellent minimum viable product that has potential to be rolled out across the UK - subject to user feedback. It is our opinion this would be possible within one further sprint.

We have ambitions to add further features, such as timed elections and more results search options, but will need to undertake extensive user testing to shape further iterations.

The project has been a steep learning curve for the whole team but it has left us keen to pursue both further iterations for The Youth Vote site and to start new projects.