



Election leaflets, 2024 project

Background

Electionleaflets.org collects uploaded leaflets from around the UK and stores them in an archive.

It's designed to track agendas and promises made in ground campaigns. This is so that voters can see what their candidate's position on issues is before elections, but also so that journalists and researchers can understand what's being said across the country.

It's been running since late 2009, although it's not been maintained or widely used since about 2016.

It used to be called 'The Straight Choice' after a particularly obnoxious leaflet sent by former MP Simon Hughes. We renamed it at some point between 2010 and 2015 to 'Election Leaflets'.

Who we're looking for

We're looking for help with Python/Django development as well as some JavaScript / frontend work. The JavaScript work is quite self-contained, and most of the work to be done is in Python/Django.

Because this project is fairly unmaintained, a lot of the work will be in triaging and discovering issues. We need someone who is comfortable with some level of project management (although the rest of the DC team will be on hand to help with this).

This project is unlikely to have a lot of maintenance in the long run, so we're looking for stable, well documented work that will help keep the maintenance easy in future. Hopefully it can keep going for another 10 years, and we want to achieve this by keeping the tech choices simple and standards based (e.g using stable libraries rather than experimental projects).

The code is here: <https://github.com/DemocracyClub/electionleaflets>

If you think you're interested, please email hello@democracyclub.org.uk

Current status

There is a mostly working website at <https://electionleaflets.org/>. The site is working in that it is possible to upload leaflets to it, and later browse those leaflets.

There is an admin interface and a basic set of tools for administering images (rotating, cropping, redacting, etc).

There are some bugs, outdated packages and features are lacking. We suspect there is a failure rate in uploads, but we don't know what this rate is. There are some errors with showing uploaded images and making thumbnails, but we're not sure what causes this yet.

Deployment and tech stack

We deploy this project to AWS Lambda via API Gateway. This is to keep running costs low and allow scaling with very little (or no) effort.

It does cause some problems. The upload limit for Lambda is 5mb. Most phones take photos that are bigger than this, so we need to upload to S3 directly (using pre-signed URLs) using JavaScript.

Thumbnails are processed using a Lambda trigger.

The project itself is written in Django and uses PostgreSQL. We use Django templates for the frontend rather than splitting the project into an API backend and JavaScript frontend.

The HTML is defined according to the Democracy Club [design system](#).

Considerations and complexities

The data model for the project needs some consideration. At the core, is a Leaflet model, that can contain one or more LeafletImages.

Against the Leaflet we want to attach some more information, such as political party, individual candidates, date received. In almost every case a UK postcode can be attached. For this phase of work, we should assume that we always get a postcode.

There is scope to add more metadata, like the type of leaflet (newspaper-like leaflets are of political interest at the moment, so we'd like to be able to identify them).

In general, we want to hang data off a Leaflet.

Trying to fit leaflets into our election model requires some tradeoffs. Not all leaflets relate to an election or a candidate (more on this below). We can't use parliamentary boundaries because 1. These change and 2. It might be that local boundaries are better. And vice versa.

There's also complications about boundary changes: boundaries only come into force on the day of election, but leaflets are sent before elections.

Because of this, we should be able to store the postcode and later filter by the best election ID or boundary.

We also need to define what a leaflet is and isn't. There are campaigning leaflets sent by 3rd party (non-regulated) groups, for example letters to neighbours about local issues. Because these aren't sent by a political party, we could exclude them, but they still tell a story about the local issues.

New features

Aside from the core upload feature, we want to allow better exploration of the archive. This means adding more filters and browsing interfaces to the frontend, but also allowing a tagging and data cleaning interface for logged in users to use. It should be possible for anyone to make an account and start cleaning data or tagging it.

This work is now partially funded by JRRT. [See the funding pitch](#). We also want to work with partners like FullFact and media organisations.

Technical backlog

We should work in GitHub mainly, but it's much easier to write the higher level tasks here for now.

“Back to square one”

- **Issue triage.** A load of the existing issues will be old and not relevant. Some will be duplicates, some will still need working on.
- **Upgrades.** Move to Django 4.2 (latest LTS that everything else DC is on), Python 3.10, latest postgres (upgrade the RDS). Upgrade all pip-installed packages to the latest versions.
- **Remove dead code.** From various refactors in the past, we have a load of dead code. Identify and remove this.
- **Update documentation.** To allow us to onboard people better in future. The docs are really old and need to be striped back. MVP is better than old.

Uploader improvements

- Test in various browsers to make sure the camera actually works. We have a browserstack account so we can manually do this. We might be able to automate these tests
- Find out why some uploads fail. Log errors better (frontend logging into sentry?)
- Ensure Lambda resizes properly and that paths are unique to the upload / leaflet

Browsing interface

- Allow browsing by various properties. TBC

Tagging interface

- Allow user log in. Use email OTT like we do on devs.DC, WDIV.
- Present tagging interface to users.
- Show list of leaflets that need tagging
- Various filters, chances are we can use the same browsing interface

OCR / text analysis

- Low priority, but we might be able to shove everything into AWS Textract and allow text searching for keywords at least
- Something something LLM something...profit?! Donno, there might be some fun to be had.