

Mini Project

You guys will be creating a sources internal tool! We actually already have a [sources internal tool](#). This will be a simplified clone of it.

What is a Source

A source is a person who we can contact to gather information for our articles. For example, a person in the UCLA administration might be a source we can talk to for an article about future tuition costs.

Requirements

1. There should be a page that lists all sources.
2. There should be a way to create new sources
3. There should be a way to modify existing sources
4. There should be a way to delete sources
5. It should store the sources in the backend's database
 - a. If you stop the server, close the browser and restart, the data should still be there.

It does not need to look pretty. All of this functionality can be on one single web page. Each source will have a name, organization, phone numbers, emails and notes (just like our existing sources internal tool)

Help Getting Started

1. You can start by cloning the workshop starter code: `git clone https://github.com/dailybruin/internal.tools.django.workshop`
2. You'll need to create a new folder in backend/. This will be the django app for sources
 - a. Make sure to add this folder in the INSTALLED_APPS in backend/settings.py
3. Figure out what data you need to store and how to create 1 or more models for the data
 - a. You can create a models.py in the folder you created in step 2. Then you can define your models there
 - b. You can test out your model(s) by opening the ``docker-compose run web folder_name_of_django_project/manage.py shell``, importing the model(s) and then running any models code
 - c. Feel free to use the [second workshop](#) as a reference

4. Create the endpoints. I would start with the endpoint for creating sources
5. Built the frontend to use the endpoint.

Frontend

I know we didn't really cover React in much depth at all. I would recommend learning with [this tutorial](#). This is the same one I used it and it was helpful.

If you are running into any issue or if you want any help, feel free to reach out to me!

Challenges

1. Make an Organization model. Then use django foreign keys to link organizations to sources. Here is a good [overview](#) of foreign keys (which are many to one relationships) and here is some [documentation](#) for django foreign keys
 - a. The idea behind a foreign key is that each source is associated with 1 organization
 - b. However, in the other direction, each organization can be associated with any number of sources. Imagine we know 2 different people in UCLA's administration.
 - c. Next, add some extra information to the organization model. Maybe something like an alternative name for the organization or a notes section for the organization.
2. Make a sign-in page using Django's built in User model.
 - a. We usually use react router to have multiple pages in our frontend
 - b. Only allow signed in users to view sources
3. Make pagination for the listing of sources.
 - a. Pagination means sources are grouped into pages. Each page has a small amount of sources (maybe 10). In order to see more sources, you have to go to the next page
 - b. Ideally, you should make the backend endpoint accept a parameter for the page number. Then the backend should only send the sources from that page.
 - c. Pagination is useful when your database has lots of entries and it would take too long to send all that data in one request.
4. Make a search bar which queries the backend.