

Final Project Deliverable #3
Completed Software: The Reproductive Rights Rangers
CAPP 30122 - Computer Science with Applications 2

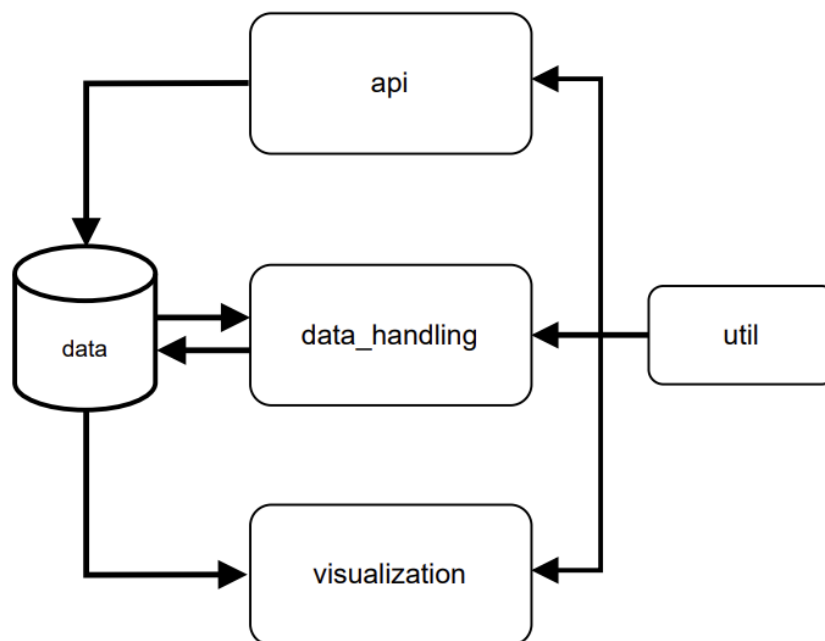
Aïcha Camara (aichacamara), Kate Habich (ehabich), Chanteria Milner (milnerca), Michael Plunkett (michplunkett)

Project Abstract

For our project, we addressed U.S. abortion policies and access as our topic. We used the state-level policy data available through the [Abortion Policy API](#) to conduct analyses on national abortion policies. We also isolated the ANSIRH Database and the I Need An A database as resources for gaining health clinic location data. With this data, we developed visualizations of abortion access, clinic location, and services provided within a state and on the zip code level. These three core datasets enabled us to create a large U.S. map in which individuals can hover over a state to gain a state-wide understanding of abortion and other reproductive health care laws. Additionally, we assembled tabular representations of abortion data by state and zip code to provide more information to users searching for reproductive health care centers near their zip code. Finally, we added a visualization that informs users of the top 20 cities with the most abortion clinics.

Software Structure

Main Software Architecture



Our project is structured around three main modules: API, Data Handling, and Visualization. The API module calls data from the Abortion Policy API database (abortionpolicyapi.com). The resulting JSON file is structured with state names as the key of the top level dictionary, whose

values are set to dictionaries of state abortion policies based on four main topics: gestational limits, insurance coverage, abortion access for minors, and abortion waiting periods. This data is then feeded into the visualizations package to develop country-level abortion data visualizations.

The Data Handling module cleans and restructures data from the ANSIRH Abortion Facility Database (<https://abortionfacilitydatabase-ucsf.hub.arcgis.com/>), which contains information on abortion clinics across the United States. The resulting JSON file is structured as a nested dictionary with two layers. The first layer contains zip code dictionaries keyed to their respective states. The second layer contains data of abortion clinics keyed to the zip code those clinics reside in. This data is then feeded into the visualization package to develop a country map and state/zip code level tables.

Finally, the Visualization module takes the resulting data and creates a Plotly dashboard containing tables and interactive maps to visualize the collected data. The util package stores auxiliary functions and constants used throughout the application, and the tests package contains tests for each of the functions used throughout the application.

Description of Responsibilities

1. Project Architecture
 - a. *Makefile* (Michael Plunkett, Kate Habich)
 - b. Poetry environment setup (Michael Plunkett)
 - c. *README* (Michael Plunkett)
 - d. *.gitignore* (Michael Plunkett)
 - e. *Black* (Michael Plunkett)
 - f. *Pylint* and *pylintrc* file sourcing (Michael Plunkett)
 - g. *Pytest* structure and configuration (Michael Plunkett)
 - h. *CODEOWNERS* (Michael Plunkett)
 - i. *__main__.py* (Michael Plunkett, Kate Habich)
 - j. Program directory structure (Michael Plunkett)
 - k. *pull_request_template.md* (Michael Plunkett)
 - l. GitHub repository, branch, and branch protection configuration (Michael Plunkett)
 - m. *__init__.py* files (Michael Plunkett)
 - n. Coding oversight and general project management (Michael Plunkett)
2. Data collection
 - a. Identifying abortion policy API dataset (Chanteria Milner)
 - b. Gaining access to the ANSIRH database (Chanteria Milner)
3. API
 - a. Abortion Policy API (Chanteria Milner, Michael Plunkett)
 - b. OpenDataSE GeoJSON data (Michael Plunkett)

- c. *service.py* (Michael Plunkett)
 - d. API tests (Michael Plunkett, Chanteria Milner)
 - 4. Data Parsing
 - a. ANSIRH functions (Kate Habich)
 - b. *service.py* (Michael Plunkett)
 - c. Tests (Chanteria Milner)
 - 5. Data Visualization (Aïcha Camara, Michael Plunkett)
 - a. *abstract_visualization.py* (Kate Habich, Michael Plunkett)
 - b. *service.py* (Michael Plunkett, Aïcha Camara)
 - c. *util.py* (Chanteria Milner)
 - d. *state_summary.py* (Aïcha Camara, Chanteria Milner, Michael Plunkett)
 - e. *zip_code.py* (Aïcha Camara, Chanteria Milner)
 - f. *city.py* (Aïcha Camara, Chanteria Milner)
 - g. *usa_country.py* (Aïcha Camara, Michael Plunkett)
 - h. *app.py* (Aïcha Camara, Chanteria Milner, Michael Plunkett)
 - 6. Util
 - a. Constants (Michael Plunkett, Chanteria Milner)
 - b. Functions (Chanteria Milner, Kate Habich, Michael Plunkett)
 - c. Tests (Chanteria Milner, Michael Plunkett)

Application Guide

Any information not found in the below two images, can be found [in the project's README file](#).

Standard Commands

- `make format` : Formats the python files within the project using the Python formatter `Black`
- `make lint` : Runs `pylint` on the codebase
- `make test` : Runs test cases in the `test` directory
- `make api` : Runs the `api` portion of the codebase
- `make parse-data` : Parses the data output of the `api` layer
- `make visualize` : Takes the data produced from the `parse-data` layer and creates the project's visualizations
- `make run` : Runs the whole application by running `make api`, `make parse-data`, and `make visualize` in serial

How do we run this thing?

There are two ways that you can run this application, one of them is to run all components of it at once and the other is to run each component individually. I will give you the instructions for both methods below.

Run with one command

1. After you have installed [Poetry](#), run the command from the base repository directory: `poetry shell`
2. Run the `poetry install` command to install the package dependencies within the project.
3. Run the `make run` command to run the application. This will spin up a web page you can access via the URL `localhost:8005`.

Run each package individually

1. After you have installed [Poetry](#), run the command from the base repository directory: `poetry shell`
2. Run the command `poetry install` to install the package dependencies within the project.
3. Run the `make api` command to get the data from the *Abortion Policy API*.
4. Run the `make parse-data` command to parse the data so that we can have our data in the format needed for the visualizations.
5. Run the `make visualize` command to start the [Flask](#) server, accessible via the URL `localhost:8005`, so that we can visualize the data we have pulled and parsed!
 - For the `make visualize` command, you must have called the two commands that are referenced before it for it to run successfully.
 - Without the data from `make api` and `make parse-data`, `make visualize` has nothing to act on.

Aspired versus Achieved Accomplishments

The original project goals included creating a searchable dashboard and database mapping U.S. reproductive laws nationally, as well as by state, with an interactive dropdown that updated the state map and information based on user inputs. We were able to accomplish most of the goals that we set out to achieve including locally hosting a Plotly Dash “Reproductive Rights Mapping” dashboard that visualized current U.S. reproductive rights data nationwide, with a chart that held the same information for ease of accessibility and readability. Additionally the dashboard also has tabular data represented in charts of reproductive healthcare clinics by zipcode and a barchart ordered by the top twenty cities with the most clinics.

The project had a reach goal of creating a database based on the U.S. abortion access laws and clinic information with which users could conduct search queries based on location. The project also aimed to create map visualization by individual state, where users could interact with the map to see zip-code level abortion clinic information, including count of clinics and services provided. We weren’t able to achieve these goals in the timeframe of the project, but we hope to continue developing it after the course ends and maintain a database of current U.S. reproductive healthcare data that we can host online for use by the general public.