Group Members: Olivia Pinney(opinney), Piper Kurtz(kurtzp), Lauren Quattrocchi (laurenq), Ellie Jackson(ecjackson)

This project has two components and serves as a proof of concept for ECCSC – a Chicago non-profit offering first responders as an alternative to the police. Most violence intervention groups primarily utilize Chicago Police data; however, ECCSC would like to use Citizen data to evaluate the accuracy of the Chicago Police Data, and consider using both data sources to compare their own first responder activity against. Their primary concern is that the Chicago Police data may not always categorize incident type correctly.

1. **Record Link:** Project EZ runs an analysis on two primary sources used to track crime incident data for Chicago. We apply a record linking algorithm to match crime data extracted from the Chicago Data Portal's API to reported incidences of crime pulled from crowd-sourced application Citizen. We match incidences based on geo coordinates, date, and time. This enables ECCSC to evaluate Citizen reports of crime compared with Chicago's Police response to crime and assess any severity differences in reporting.

2. **First Response Activity:** Pulls down data from a google survey which ECCSC uses to log first response activity. Cleans data and generates latitude and longitude coordinates to map into a map of Chicago using Dash. ECCSC can visually assess their first response activity for internal operations and to demonstrate their value proposition to prospective funders.

Project Structure
The project is modularized into

Responsible Member	Primary Modules	Primary Tasks
Piper Kurtz	Map Refresh Data	• Chicago data API pull • Visualization of ECCSC data using Dash • Create command line interaction to select which component of project to run
Olivia Pinney	Refresh Data	• Citizen scrape • Built automated daily citizen pull (on local machine) • Storage system for preserving csv files aggregating by month of crime incident and key search terms
Ellie Jackson	Link Records Refresh Data	• Pull down google sheet ECCSC data • Find latitude and longitudes based on inputted survey data • Record link algorithm
Lauren Quattrocchi	Link Records Data	• Establish SQL database connection for each data source • Clean incoming data and data types • Record link algorithm

Interact with EZ Project
To run our project from command line:

```
cd ~/proj-ez
```

```
sh install.sh
```

(if local machine requires different command line arguments for virtual environment, then enter following commands to run program)

```
cd ez
bash project-requirements
```

Expected output:

Interaction:

- Asks for user input, if they would like to run record link analysis, map data, or exit program
- Once user inputs, echos user choice and runs corresponding command
- If still in script after command is run, reminds user of possible inputs

From Link Records Analysis:

- Prints the start and end dates captured by Chicago Crime data and Citizen. Prints number of days captured between the two datasets which overlap and can generate a match
- Writes all matches to a csv file called "match_file.csv"

From ECCSC Map visualization:

- Displays map of first response activity from ECCSC data generated from google forms survey. Click cursor on marker to display detail of crime type.

Project Accomplishments

Since our project worked with a Chicago organization, our project has a continued roadmap and vision beyond the scope of this project. For the project deadline, our objective was to have a proof-of-concept model to demonstrate to ECCSC the potential analysis and data collection visualizations. We will be presenting our work to ECCSC after the break.

Presenting these visualizations to the ECCSC team will help increase buy-in for the data collection efforts currently in development. At ECCSC's headquarters, there is a large TV display. Displaying the map generated by our project at ECCSC's headquarters will allow ECCSC to have a visual presence of their work and value provided to Chicago on a daily basis. This will increase a shared understanding of the organization's reach, and demonstrate the importance of data collection efforts for people across all levels of the organization.

It's important to highlight this project captures functionality beyond the scope of this project. The project considers functionality for ECCSC by automating data pulls from Citizen and creating a storage record process based on the incident types ECCSC cares about. ECCSC does not yet have in-house tech capabilities, so this functionality will be critical to the program's continued use. We will be able to expand upon this with our coursework in databases next quarter.

Project Structure

Below is the structure of our software program ez. We modularized our program in the following format: Our top-level package is ez. Ez contains our project-interaction batch file which allows you to decide which component of the project to run, linking records or mapping data.

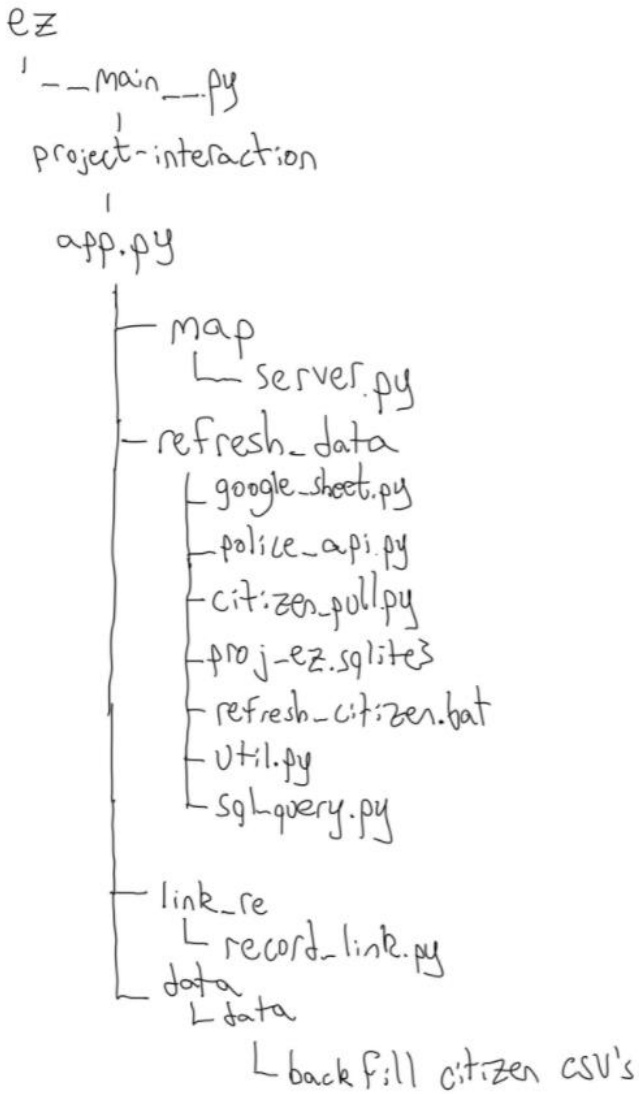
Ez has three sub-packages map, refresh_data, and link_records:

The purpose of the map package is to generate a map of ECCSC's responses. In map, server.py is the main file which calls the dash server to create the map. Server.py imports the sql_query module from our refresh_data package to generate the data frame of ECCSC's data.

The purpose of refresh_data package is to refresh our sql database with the most recent data from our sources, the citizen app (citizen_pull.py), and the Chicago crime data portal(police_api.py.). It also contains our sqlite3 database proj-ez. Both the citizen_pull and the police_api call the util.py file to import functions to use for sql and csv connections. The refresh_citizen.bat is a batch file that is referenced locally on Olivia's computer to refresh the citizen data daily. The google_sheet.py file pulls data from the ECCSC survey data into the sql database.

The purpose of the link records package is to find matches between the Chicago Crime Data Portal and Citizen App data. The main file is the record_link.py file which matches event based on location and time. Record_link.py imports the sql_query module from our refresh_data package to generate the citizen and Chicago data frames.

Additionally, there is a data directory which holds historical citizen data. This historical record is important because our method of scaping Citizen only allows access to the last 1,000 records. These csvs ensure a backup of data in case the sql table is corrupted. The sql table can be backfilled from these files as needed.


ez
__main__.py
project-interaction
app.py
map
server.py
refresh_data
google_sheet.py
police_api.py
citizen_pull.py
proj_ez.sqlite3
refresh_citizen.bat
util.py
sqlquery.py
link_re
record_link.py
data
data
backfill citizen CSV's