Judd.world

Easy access to New York State's conviction and sentencing data

Contents of this template

You can delete this slide when you're done editing the presentation

Fonts	To view this template correctly in PowerPoint, download and install the fonts we used
Used and **alternative resources**	An assortment of graphic resources that are suitable for use in this presentation
Thanks slide	You must keep it so that proper credits for our design are given
Colors	All the colors used in this presentation
Icons and **infographic resources**	These can be used in the template, and their size and color can be edited
Editable presentation theme	You can edit the master slides easily. For more info, click **here**

For more info:
SLIDESGO | **BLOG** | **FAQs**

You can visit our sister projects:
FREEPIK | **FLATICON** | **STORYSET** | **WEPIK** | **VIDEVO**

Table of contents

01

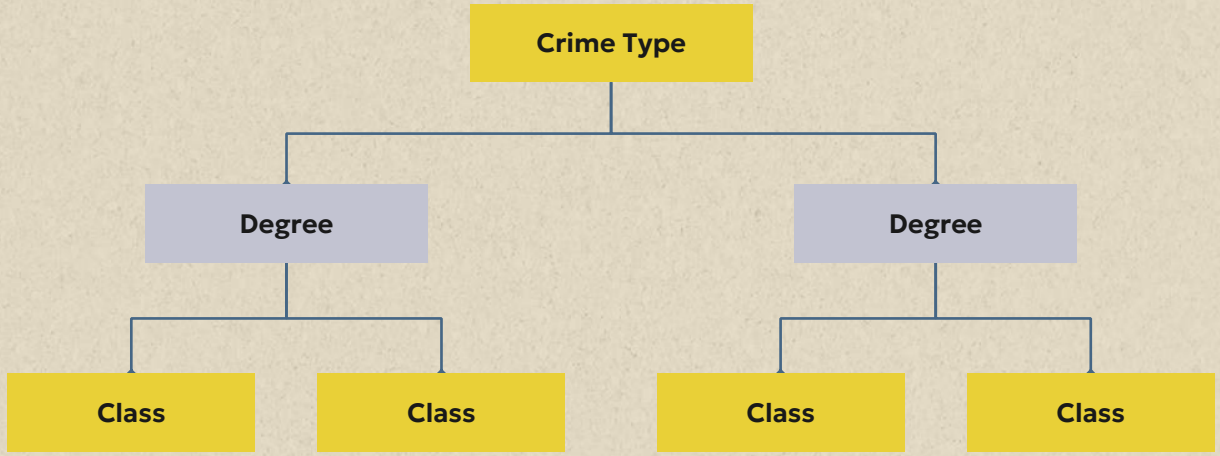
Introduction

Mission

Judd's mission is to provide the public with easy access to accurate and real local conviction and sentencing data. By simplifying complex data, Judd intends to promote transparency and accountability in the criminal justice system, helping individuals grasp its nuances and identify biases and disparities.

It also aims to empower the public by informing them, enabling members of the public to make informed decisions with regards to the criminal justice system. As a result, Judd will play a crucial role in shaping a more just and equitable society, fostering a culture of transparency, accountability, and inclusivity.

Crimes, Degree, Classes



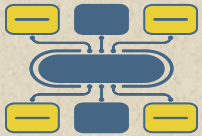
- Crimes: Refer to the specific categories of offenses, such as robbery, assault, or burglary, which define the nature of the criminal act
- Degrees: Refer to the level of severity or culpability within a specific crime type, such as first-degree murder, which determines the applicable penalties and sentencing ranges
- Classes: Refer to the category of crimes based on their seriousness and potential punishment, such as felonies or misdemeanors.

Note: Because crime types in the same degrees have the same classes, classes are not included in the forms.

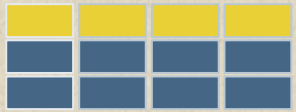
Stack

Data Collection and Scraping: Python, Selenium, Pandas

Data Preprocessing and Processing: JavaScript, Excel



Database and Data Storage: Supabase



Web Stack: React, Node, Vite, Bootstrap

Data Visualization: React-chartjs-2

Data Collection

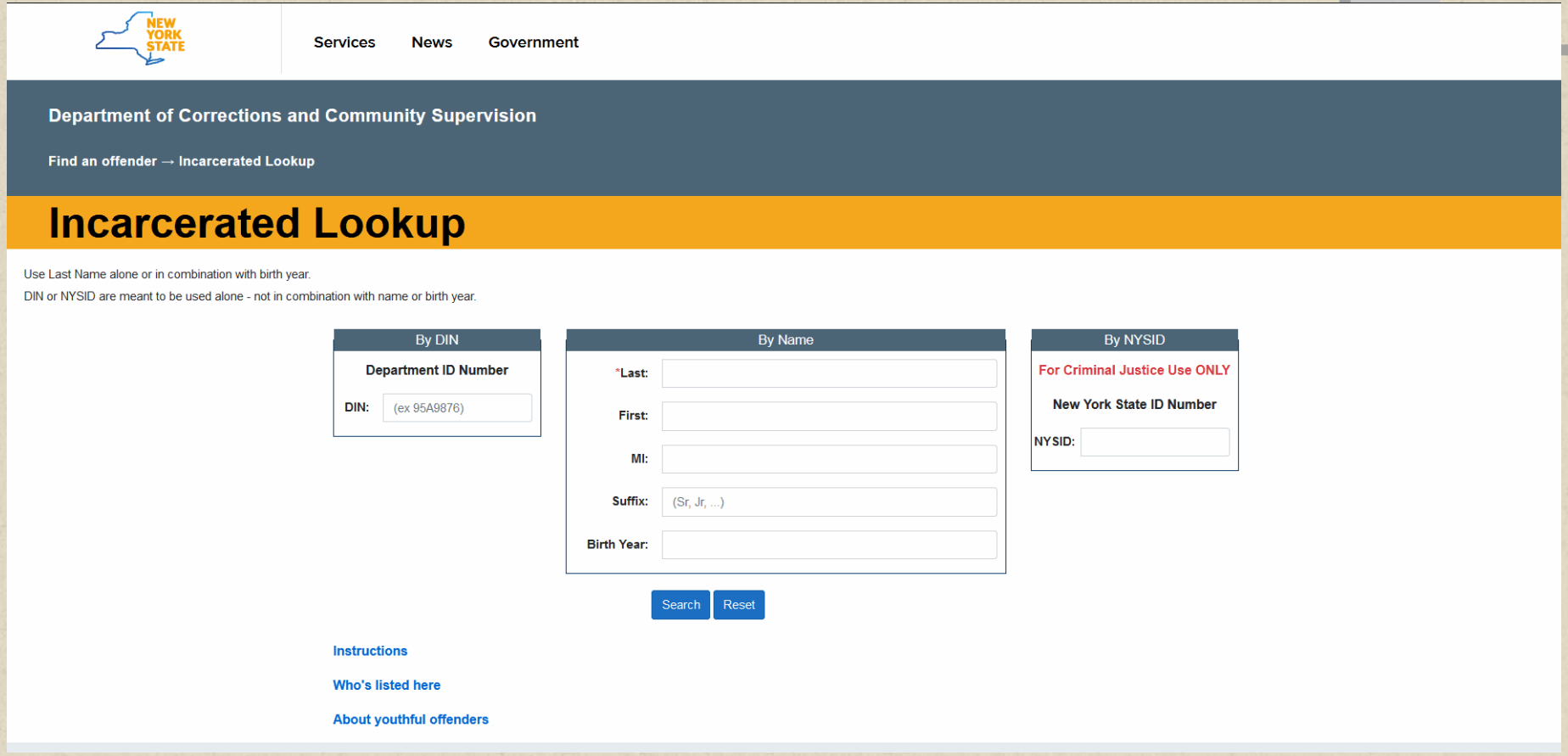
To build Judd, I needed the real sentencing and conviction data for New York State, and the website that houses this data is New York State's Department of Corrections and Community Supervision website at: https://nysdoccslookup.doccs.ny.gov/ .

Due to the structure of the website, where each page presents only 10 convicts at a time and clicking on a link to get the complete data for a convict, I had to interact with elements on the web page to be able to scrape the data. Doing this manually was not feasible, so I had to use a dynamic scraper to interact with elements on the page, then scrape the data, and then handle navigating to the next page. And I used Python's Selenium package for this.

When building the script, I had to handle timeout exceptions and events where an element was not visible, so I used selenium methods to implement the necessary waits and exception handlers.

After getting access to a page's data, I created a pandas dataframe with the data and appended the dataframe to a local csv file. After scraping into the csv file, I removed all duplicates and formed a final csv file. The data scraper is accessible at: https://github.com/mao-99/NYS-DOCCS-Data-Scraper/

Nysdoccs.lookup.ny.gov



Database Schema

column_name	data_type
id	bigint
din	character varying
firstName	text
lastName	text
dob	date
age	smallint
race	character varying
multipleOffense	boolean
crimeArray	jsonb
degreeArray	jsonb
status	character varying
prison	character varying
county	character varying
aggMinSentence	smallint
aggMaxSentence	smallint
incarcerationDate	date
sentenceStart	date
releaseDate	date
sentenceDuration	smallint
created_at	timestamp with time zone
classArray	jsonb

Data Preprocessing, Cleaning & Database Population

After scraping the data into a csv file, I needed to import the data into my database and so I wrote a NodeJS script to clean up the data, and handle entries with incomplete or improperly formatted entries. I then separated the entries into the proper format and columns according to my database schema and wrote each new properly formatted entry into a final csv file. The csv file is called dataScrap and it is accessible at: https://github.com/mao-99/judd.world/tree/main/src/backend

I then used Microsoft Excel to further normalize the dataset and clean it up and make formatting changes to be compatible with my Supabase table. After cleaning and preprocessing, I imported the csv file into my Supabase table.

The supabase table currently has 337,244 entries.

Note: Due to supabase restrictions, the app is not able to utilize all 337,244 entries because of a timeout error. So anytime the application runs, it only runs on a random 10,000 out of the total entries.

Features

- Simple Average: A user can find the average sentence for a specific profile. A user can specify an age, race, and county which they want to check the average for. They then select at least one crime type and a corresponding degree. If a user does not specify a degree, then the degree defaults to the "1ST" degree.
- Simple Average Plots: Here, a user can select a maximum of four crime types and their corresponding degrees. A user can also specify a filter which they want to filter the plot with. A user can get the average sentence for the selected crime type(s) sorted by age, county, or race. If a user chooses to sort by age, then they can select the age range which they want each bar to represent. This feature presents the averages to user as histograms.
- Simple Average Comparisons: Here, a user can select a maximum of four selections, where each selection is an input group of demographics (i.e. age, county, and race). Based on the user's input, a chart then presents the average sentences of all crimes that satisfy the selected inputs. If a user fills in multiple selections, the chart only returns crimes that those two demographic profiles have in common. For example, if a user selects "Queens" and "Bronx" for two selections respectively, then the chart returns the averages of all crimes that "Queens" and "Bronx" share.

Note: If a user's input does not have any corresponding entries, then the simple average returns 0, and the simple average plots either don't show any data or they do not show at all

Demo

Home

Simple Averages

Simple Average Plots

Simple Average Comparisons

Average Sentences

Crime(s) Selection

Note that unspecified degree(s) will be defaulted to 1ST

Note that an unspecified filter will default to AGE

Crime 1

Select Crime(s) | Select a Degree

Crime 2 (optional)

Select Crime(s) | Select a Degree

Crime 3 (optional)

Select Crime(s) | Select a Degree

Crime 4 (optional)

Select Crime(s) | Select a Degree

Select Filter

Select Filter

Next Steps

- Work on design
- Transition from supabase to postgres database
- Bug fixes
- Include an additional comparison feature where users are able to compare the average of a particular demographic info to the average across all demographics. For users to see how well/worse off demographics are compared to the average/fair convice.
- Include a standardized value to tell a user how disperse the sentencing for a particular demographic is, called a judd index

Questions

Thanks

Do you have any questions?
LinkedIn: Mubarak Odufade
GitHub: mao-99
Discord: mao.nda