**CMPE 280 Project**

*SafeLink*

*Visualization of Crime Dataset from Chicago Police Department*

***Done By:***

**Team CAGNA**

Gurleen Dhillon

Devashish Nyati

Chaithra Lakshmi Sathyanarayana

Anjali Vijayaraghavan

Table of Contents

[**1. Application Description**](#_jom2jn4gkyoz) **2**

[Users and Story](#_tedzb54ee4iw) 2

[Application KPIs](#_pogs0vcvoy30) 3

[User Decisions](#_6mb9besnjh3z) 3

[**2. Application Overview**](#_pbpf5rb5b2gz) **3**

[Design Principles](#_fegda3pjoct6) 3

[Architecture](#_w0mumma3m2k3) 4

[Technologies Used](#_xpjcyosffe6b) 5

[Client-side](#_bk2klum3k7uj) 5

[Server-side](#_5thd7c20pznp) 5

[Database](#_nuq2uwn8blfi) 5

[Tableau](#_pb0ehsozfeql) 5

[**3. Data - Extraction, Transformation, Loading (ETL)**](#_3pt7zdehshyc) **5**

[Data Extraction](#_bz3lat8j6o0y) 5

[Crime Details](#_57ade0yo3lix) 5

[Strategic Subject List (SSL)](#_n79ec2g4r69) 8

[Fire Station Details](#_tsn6vi1jg92n) 9

[Police Station Details](#_tg0w25tw4zbe) 9

[Data Transformation](#_y1nvacopakeu) 9

[Loading Data](#_81h27cvujyb9) 10

[**4. Application Design**](#_458xh8ln26p6) **11**

[Visualizations](#_z9timcgcakve) 11

[Dashboard](#_mdfa2vxmvwyf) 11

[Community Area - Zip](#_6pkvds2alpt5) 14

[2019](#_2iqwvb7ok8t2) 15

[SSL (Strategic Subject List)](#_gbj67ows3n59) 16

[Crime Incident Report](#_og36kifimzzq) 18

[About](#_brraefmgd4dh) 18

[Pagination](#_ef02off10o3b) 18

[Search](#_swvcysfu2vnc) 18

[Data](#_alowghcuu9m4) 19

[EJS](#_12hlm0aj20k3) 20

[Contribution to Project Story](#_8sxdccwbgwto) 21

[Design Patterns](#_ke0pdxkdq08g) 22

[Running the Application](#_bu0kiy19hv88) 24

# 1. Application Description

SafeLink is a tool to visualize reported crime in Chicago. The application has a dashboard with the monthly crime stats for the current month as the landing page. There are various tabs on the left that provide navigation. SafeLink is a third party tool that is leveraging the data provided by the City of Chicago Data Portal to provide visualizations and correlations of crime trends. The goal of the application is to communicate the safest zip codes/community areas within the City of Chicago:

* Safety measured as crime per capita
* Highlight frequency of crime and type of crime
* Nearest fire and police stations
* Transparency of Crime Incident Reports:
  + Build trust with community
  + Raise awareness with the public

## Users and Story

* 1. Citizens of Chicago
  2. Citizens moving to Chicago (New to Chicago)
  3. Chicago Police Department

Residents of Chicago, if crime troubles you, and you want details about crimes like where crimes are taking place, what type of crimes are the most frequent ones, what is your risk assessment score, safety score of your community, nearest police stations and fire stations in your area, and much more details, SafeLink is the one to go for.

If you are planning to move to Chicago, and crime worries you, SafeLink can help you find a safe place to stay in Chicago. Our dashboard covers the crime statistics for Chicago on a whole, giving the user a clearer picture of how things currently are in the city. You can search for community areas and their safety scores.

SafeLink is also for police department and law makers. They can check what types of crimes are happening, predictions of crime for the coming days, how many arrests are successful and a lot more.

## Application KPIs

The KPIs for the application are:

* Number of Crimes
* Number of Arrests
* Homicide Rate
* Rate change in crime from month to month
* Community Area with highest reported incidents for the month

## User Decisions

The decisions that can be made from the dashboard are:

* Users can choose to focus on a community area and explore the crime trends
* Users can determine how safe a particular zip code is
* Users can choose to navigate to the SSL page to learn more about the Risk Assessment Score
  + Can interact with the visualizations and filter based on age
  + Users can get domestic violence trends through 2002 to 2019

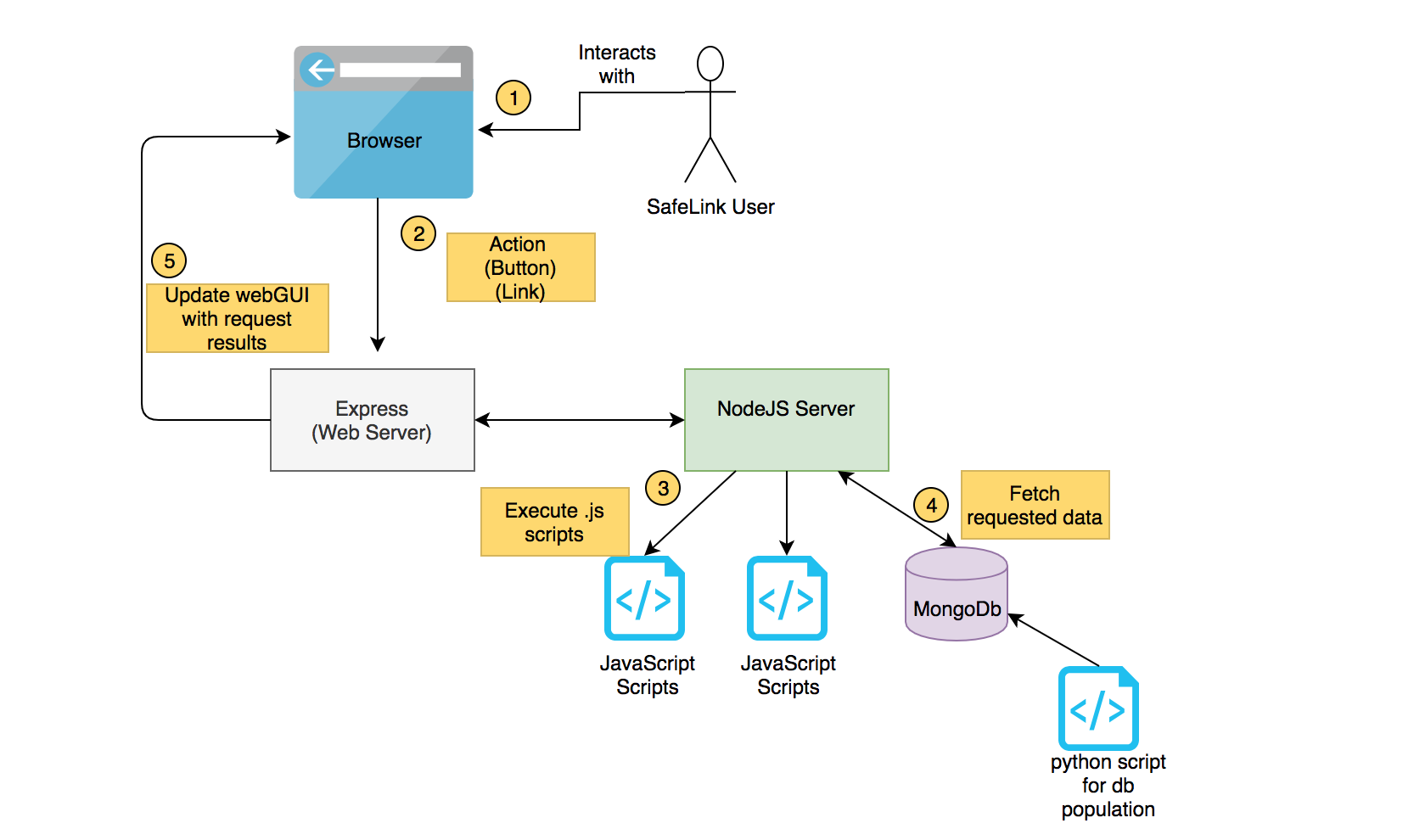
# 2. Application Overview

## Design Principles

The design principles enacted for the application:

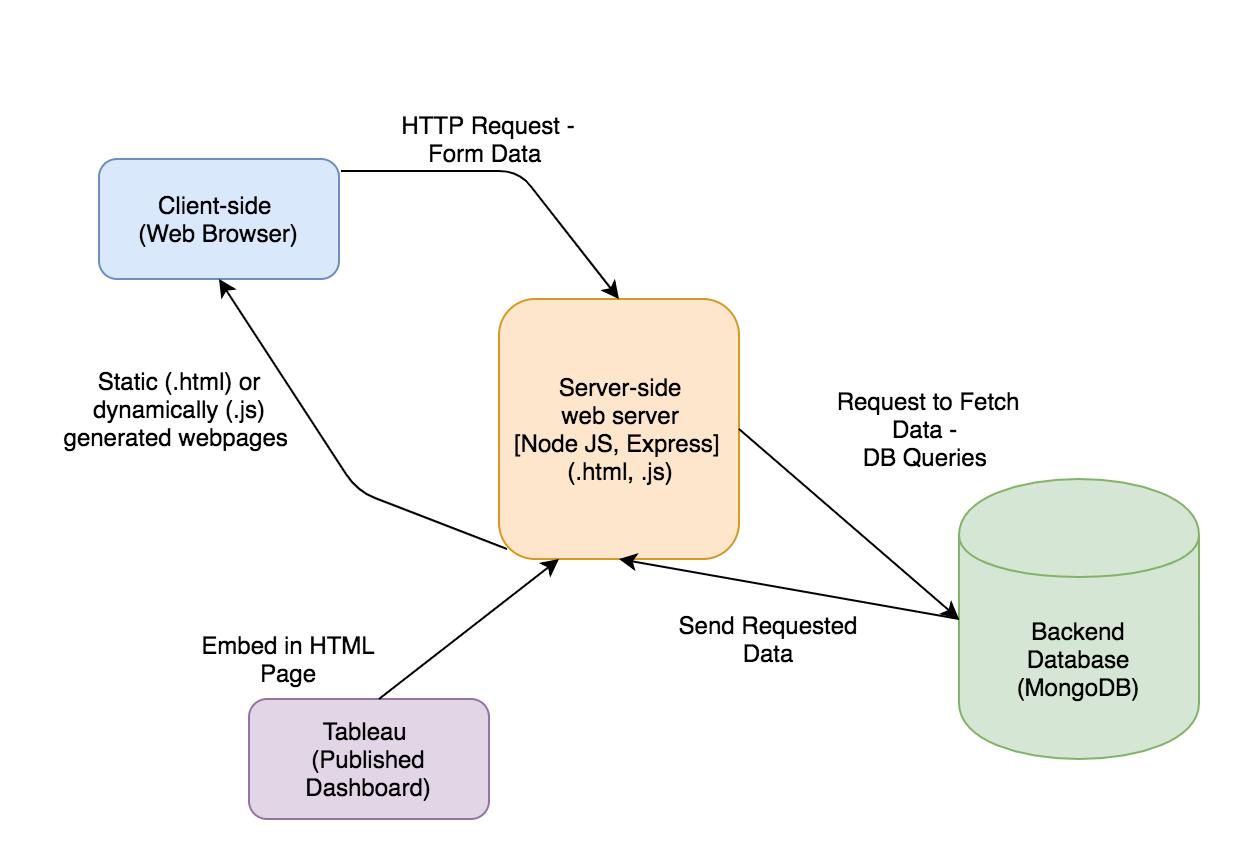
* Consistency
  + The page layout is consistent across the application
  + Navigation menu retains the same order
* Gestalt's Principle
  + Data points and visualizations that were similar were kept in close proximity
  + There is continuity between the visualizations
* Simplicity is Key
  + Minimal text
  + Strategic use of color
  + Elements have similar look and feel throughout application
* Data-ink ratio
  + Each of the graphs and tables eliminate the non-essential pixels

## Architecture



**Figure 1. The Architecture Diagram for SafeLink**

The application’s landing page is the dashboard and from there the user can navigate to various pages using the left hand navigation bar. The tabs can be clicked from any page and any of the various pages are accessible from any point in the application.



**Figure 2. The Application Architecture**

The application follows the standard three-tiered web application architecture. As mentioned below, there are three distinct components or “sides” to the application: client-side, server-side and backend. These components will be discussed in detail below with respect to what specific technologies were used.

In addition to the standard three-tier architecture there is a fourth component which is Tableau. Tableau was used to create the visualizations and thus the published Tableau dashboards needed to be embedded directly in the static html pages.

## Technologies Used

### Client-side

* For the client-side we relied on html, css, javascript.
* The html and css was from a bootstrap template that is open source.
* Javascript was implemented in addition within the html pages.
* Ajax was also used in some pages to fetch and render information either at page load or enter of a field. For instance on the zip code page the user can search a specific zip code and once the user enters the value ajax does an api call to gather the information.

### Server-side

* Express and nodejs are used for the server side.

### Database

* MongoDB was used to store the data. Mongodb is hosted on cloud using mlab.

### Tableau

* Tableau was used for visualizations

# 3. Data - Extraction, Transformation, Loading (ETL)

## Data Extraction

### Crime Details

General details about the crime, regarding its location, type of crime etc. is covered by the dataset retrieved from the URL - [*https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2/data*](https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2/data)

The columns under the above dataset are as follows -

|  |  |
| --- | --- |
| **Column** | **Details** |
| ID | Unique identifier for the record. |
| Case Number | The Chicago Police Department RD Number (Records Division Number), which is unique to the incident. |
| Date | Date when the incident occurred. this is sometimes a best estimate. |
| Block | The partially redacted address where the incident occurred, placing it on the same block as the actual address. |
| IUCR | The Illinois Uniform Crime Reporting code. This is directly linked to the Primary Type and Description. See the list of IUCR codes at <https://data.cityofchicago.org/d/c7ck-438e>. |
| Primary Type | The primary description of the IUCR code. |
| Description | The secondary description of the IUCR code, a subcategory of the primary description. |
| Location Description | Description of the location where the incident occurred. |
| Arrest | Indicates whether an arrest was made. |
| Domestic | Indicates whether the incident was domestic-related as defined by the Illinois Domestic Violence Act |
| Beat | Indicates the beat where the incident occurred. A beat is the smallest police geographic area – each beat has a dedicated police beat car. Three to five beats make up a police sector, and three sectors make up a police district. The Chicago Police Department has 22 police districts. See the beats at <https://data.cityofchicago.org/d/aerh-rz74>. |
| District | Indicates the police district where the incident occurred. See the districts at <https://data.cityofchicago.org/d/fthy-xz3r>. |
| Ward | The ward (City Council district) where the incident occurred. See the wards at <https://data.cityofchicago.org/d/sp34-6z76>. |
| Community Area | Indicates the community area where the incident occurred. Chicago has 77 community areas. See the community areas at <https://data.cityofchicago.org/d/cauq-8yn6>. |
| FBI Code | Indicates the crime classification as outlined in the FBI's National Incident-Based Reporting System (NIBRS). See the Chicago Police Department listing of these classifications at [http://gis.chicagopolice.org/clearmap\_crime\_sums/crime\_typ](http://gis.chicagopolice.org/clearmap_crime_sums/crime_types.html) |
| X Coordinate | The x coordinate of the location where the incident occurred in State Plane Illinois East NAD 1983 projection. This location is shifted from the actual location for partial redaction but falls on the same block. |
| Y Coordinate | The y coordinate of the location where the incident occurred in State Plane Illinois East NAD 1983 projection. This location is shifted from the actual location for partial redaction but falls on the same block. |
| Year | Year the incident occurred. |
| Updated On | Date and time the record was last updated. |
| Latitude | The latitude of the location where the incident occurred. This location is shifted from the actual location for partial redaction but falls on the same block. |
| Longitude | The longitude of the location where the incident occurred. This location is shifted from the actual location for partial redaction but falls on the same block. |
| Location | The location where the incident occurred in a format that allows for creation of maps and other geographic operations on this data portal. This location is shifted from the actual location for partial redaction but falls on the same block. |

The dataset covers data from 2001 to present year. But for our application, we are have used the dataset from the years 2016 till 2019 and it is downloaded as a CSV file.

### Strategic Subject List (SSL)

SSL scores reflect an individual’s probability of being involved in a shooting incident either as a victim or an offender. Scores are calculated and placed on a scale ranging from 0 (extremely low risk) to 500 (extremely high risk). Similar to Crime Details, the dataset is from the years 2016 to 2019 and is downloaded as a CSV file. A few of the columns from this dataset are as follows -

|  |  |
| --- | --- |
| **Column** | **Details** |
| SSL SCORE | Strategic Subject List Score |
| PREDICTOR RAT AGE AT LATEST ARREST | Attribute RAW Score; The raw score calculated by the model; The individual's age at the time of their most recent arrest. |
| PREDICTOR RAT VICTIM SHOOTING INCIDENTS | Attribute RAW Score; The raw score calculated by the model; The number of times an individual has been the victim of a shooting. |
| PREDICTOR RAT VICTIM BATTERY OR ASSAULT | Attribute RAW Score; The raw score calculated by the model; The number of times an individual has been the victim of an aggravated battery or aggravated assault |
| PREDICTOR RAT ARRESTS VIOLENT OFFENSES | Attribute RAW Score; The raw score calculated by the model; The number of times the individual has been arrested for a violent offense. |
| PREDICTOR RAT GANG AFFILIATION | Attribute RAW Score; The raw score calculated by the model; Indicator if an individual has been confirmed to be a member of a criminal street gang. 1=Yes; 0=No |
| PREDICTOR RAT NARCOTIC ARRESTS | Attribute RAW Score; The raw score calculated by the model; The number of times the individual has been arrested for a narcotics offense. |

More fields can be found from the URL - <https://data.cityofchicago.org/Public-Safety/Strategic-Subject-List/4aki-r3np>

### Fire Station Details

Using dataset in CSV format from the URL - <https://data.cityofchicago.org/Public-Safety/Strategic-Subject-List/4aki-r3np>

We are capturing information regarding the closest fire stations in a certain locality by using the zip code as the search criteria. Fields for this dataset are *NAME, ADDRESS, CITY, STATE, ZIP, ENGINE* and *LOCATION* which by the names themselves clearly indicate what each stand for.

### Police Station Details

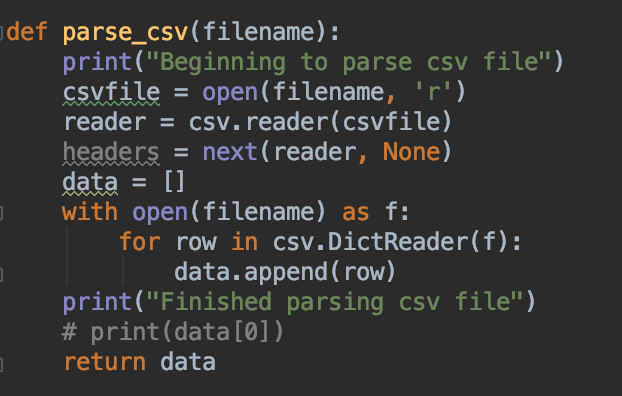
Using dataset in CSV format from the URL - <https://data.cityofchicago.org/Public-Safety/Police-Stations/z8bn-74gv>

We are capturing contact along with some basic information of police stations in a certain locality by using the zip code as the search criteria. Fields for this dataset are - *DISTRICT, DISTRICT NAME, ADDRESS, CITY, STATE, ZIP, WEBSITE* which by the names themselves clearly indicate what each stand for.

## Data Transformation

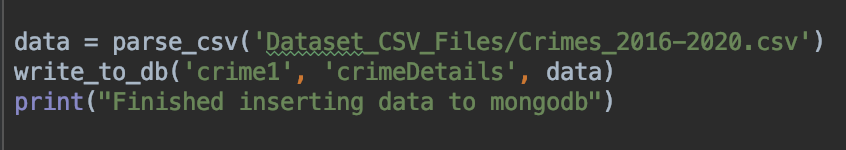
A Python script is written to gather the records in each of the individual CSV files and transform the data to a dictionary before loading the information into MongoDB.

In order to transform the content of the CSV file, it has to be first read and parsed as shown below:



**Figure 3. Code Snippet of Parsing Data**

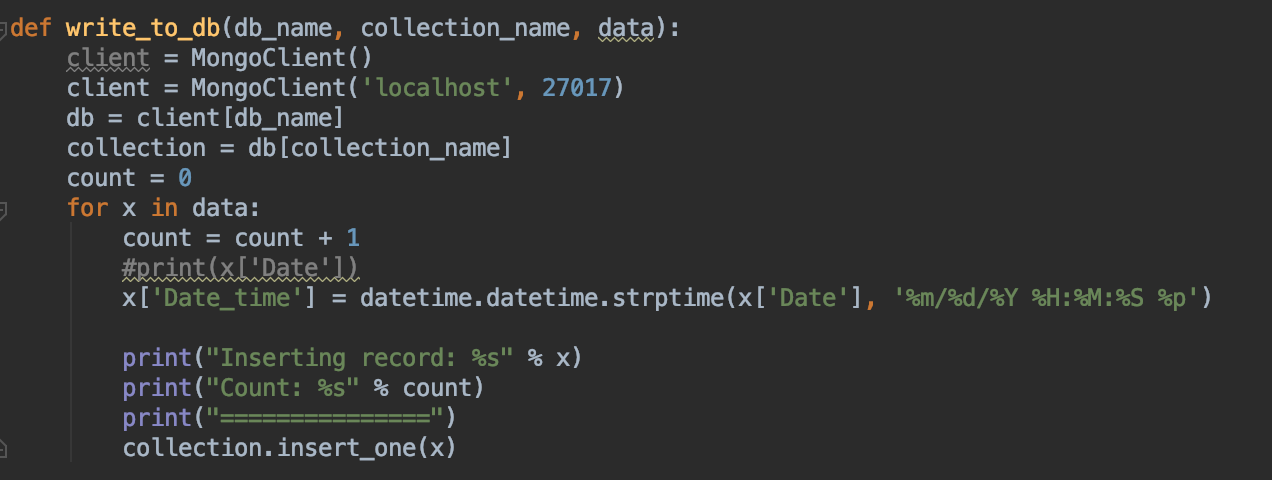
In the code snippet shown above, information is printed on the log stating parsing has begun. After opening the the CSV file in *r* (read) format, we go through each row of content and read it in dictionary format and then append it to *data* list. The function *parse\_csv()* is called from the main as shown below:



**Figure 4. Calling *parse\_csv()* function**

## Loading Data

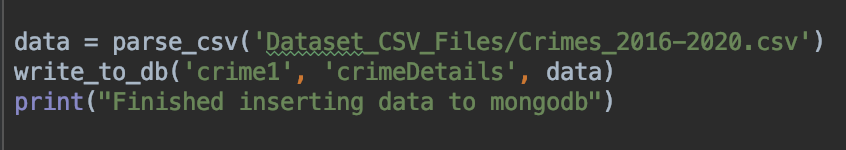
In order to write into MongoDB, connection string is written using MongoClient, as shown below:



**Figure 5. Code Snippet of Writing Data to MongoDB**

After creating the connection to MongoDB, using the collection name passed through the variable *collection\_name* each row of content is inserted into this collection using *insert\_one()* function.

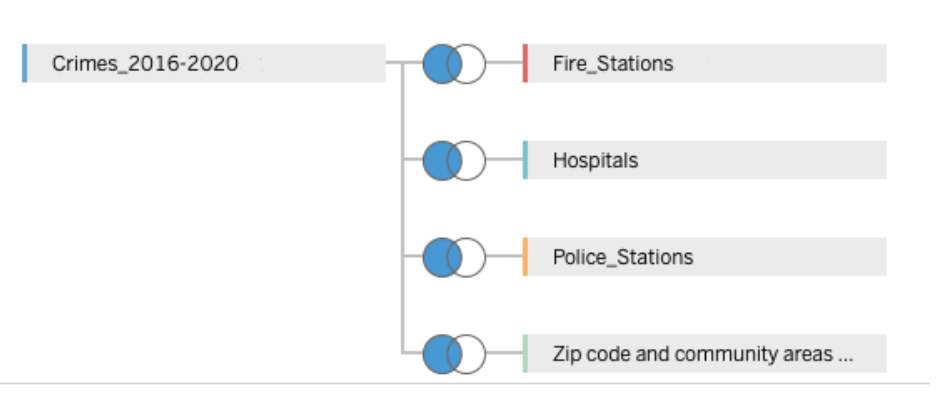
The function *write\_to\_db()* is called once the CSV file has been parsed as shown below:



**Figure 6. Calling *write\_to\_db()* function**

Loading data to Tableau:

To load data to Tableau from MongoDB, MongoDB BI Connector was used. Different collections and csv files were joined. Left inner join was used to connect crimes collection with police, fire station and zipcode to community area collections.



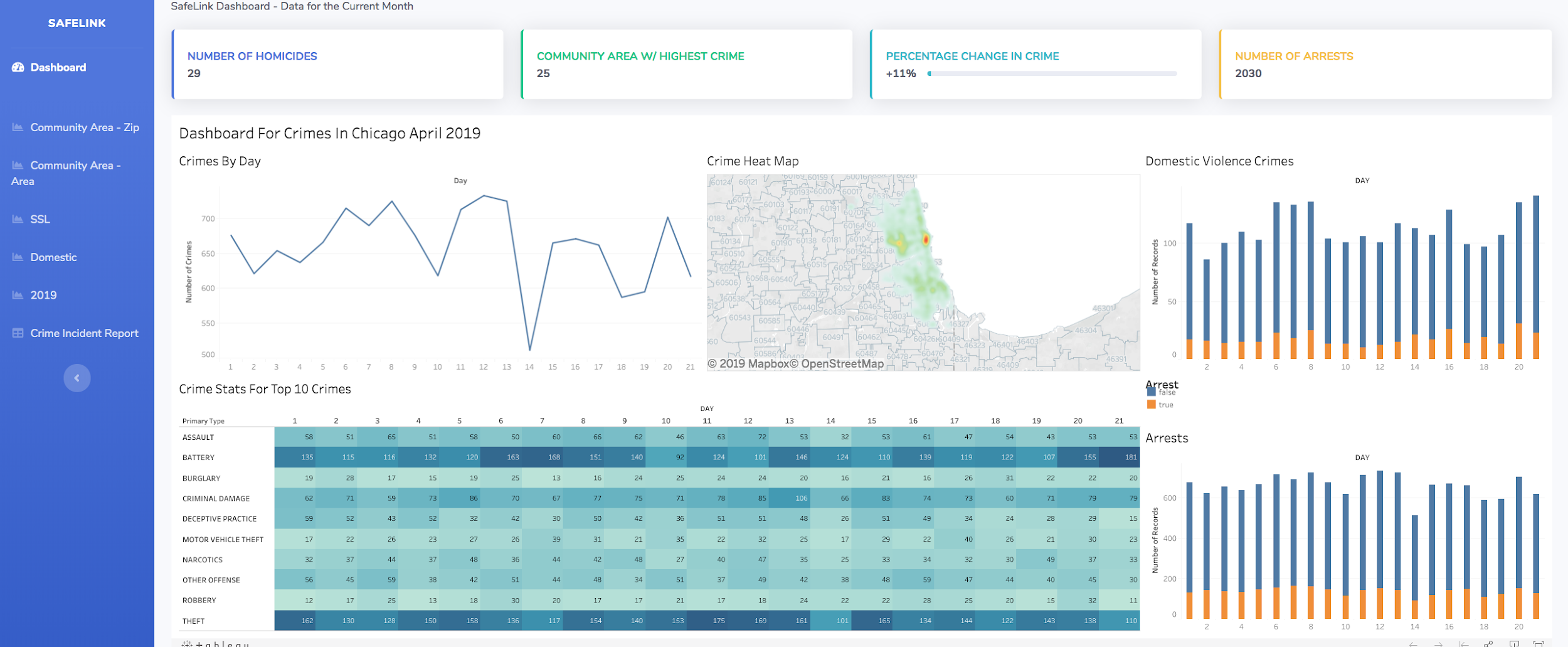
**Figure 7. Tableau Data Source**

# 4. Application Design

## Visualizations

### Dashboard

The below pictures shows the screenshot of dashboard. We have used tableau to generate the visualizations. The four modules at top use data from mongodb. These modules are updated via ajax so that when the query returns only the data in the box is refreshed and everything else is not. For the top 4 modules, since the data is being fetched from mongodb, if the data is not loaded, then it displays the state as “Loading”.

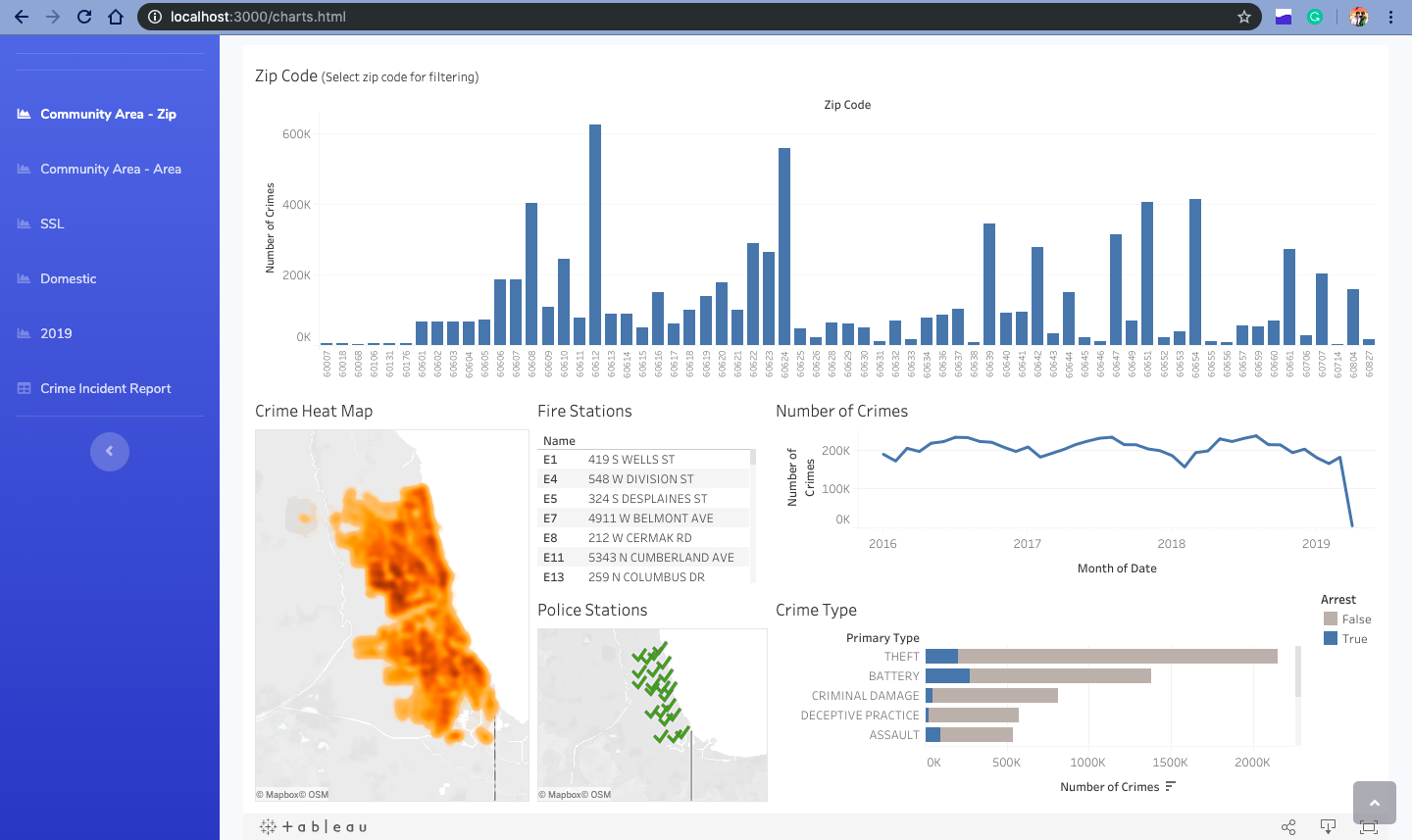
****

**Figure 8. Screenshot of Dashboard page for the month of April 2019**

1. The dashboard displays the graphs and statistics for the current month. The above screenshot shows visualizations for the month of April 2019
2. There are 4 KPIs at the top of the dashboard page.
   1. Number of homicides - This gives the number of homicides for the current month
   2. Community area with highest crime - Based on the crime statistics, this gives the community area with the highest number of crimes for the current month. This helps the user know whether the community they are looking into is the most dangerous area of Chicago.
   3. Percentage change in crime - This shows the percentage change in crime over the from the two months ago to the last month. This helps the user know if the crime rates in the city are increasing or decreasing. From the dashboard, we can see that there is a 11% increase in crime rate from the month of February 2019 to March 2019.
   4. Number of arrests - This shows the number of arrests for the current month - April 2019. Since arrests happen in crimes that are considered quite serious, the users can get an idea about the crimes and people involved in serious crimes from this KPI.
3. Crimes by day - Crime by day is a line chart that shows the total number of crimes every day for the current month. This helps the user know what to expect about the count of crimes on an everyday basis.
4. Crime Heat Map - The crime heat map helps to user get an estimate of which areas are considered dangerous. The color varies from green to yellow to red corresponding to low number of crimes, medium number of crimes and high number of crimes respectively. This helps the user know whether the area they are going though is safe or not. They might carry a self defense weapon like pepper spray or stun gun if they are venturing into places that could be dangerous.
5. Domestic violence crimes - This is a stacked bar graph, giving the breakdown of domestic violence crimes and other crimes. Since, the total crime statistics includes the domestic violence crimes, if the user wants to know about the number of crimes unrelated to domestic violence, they can use this chart. Domestic violence is family specific, and if the user’s family does not face the issue of domestic violence, they might be interested in non-domestic violence crimes and use this chart for the same. Consequently, if the user is a domestic violence victim or interested in working for organizations involved in working with domestic violence victims, this helps them know the domestic violence statistics for the current month.
6. Crime stats for top 10 crimes - Although there are varieties of crimes, there are a few crimes that often frequently. Therefore the user can get an overview of the top ten crimes each day for the current month. To make it easier for the user, the color changes from light blue to dark blue based on the number of crimes. Light blue indicates lower number of crimes and dark blue indicates high number of crimes. This helps the user know what kind of crime they might want to be prepared against. They might carry a self defense weapon like pepper spray or stun gun to protect themselves against crime like assault, and battery.
7. Arrests - This is a stacked bar chart that shows the ratio of number of arrests to non-arrests for the crimes everyday. The general assumption is that arrests are carried out for serious crimes than for small misdemeanors. This helps the users get an idea about how many serious crimes might happen on an everyday basis.

### Community Area - Zip

To know the crime statistics by community area or zip code, the user can navigate to community area - zip or community area - area link. Below is a screenshot of community area by zip code dashboard. This page has crime details filtered by zip codes and safety score for each zip code.

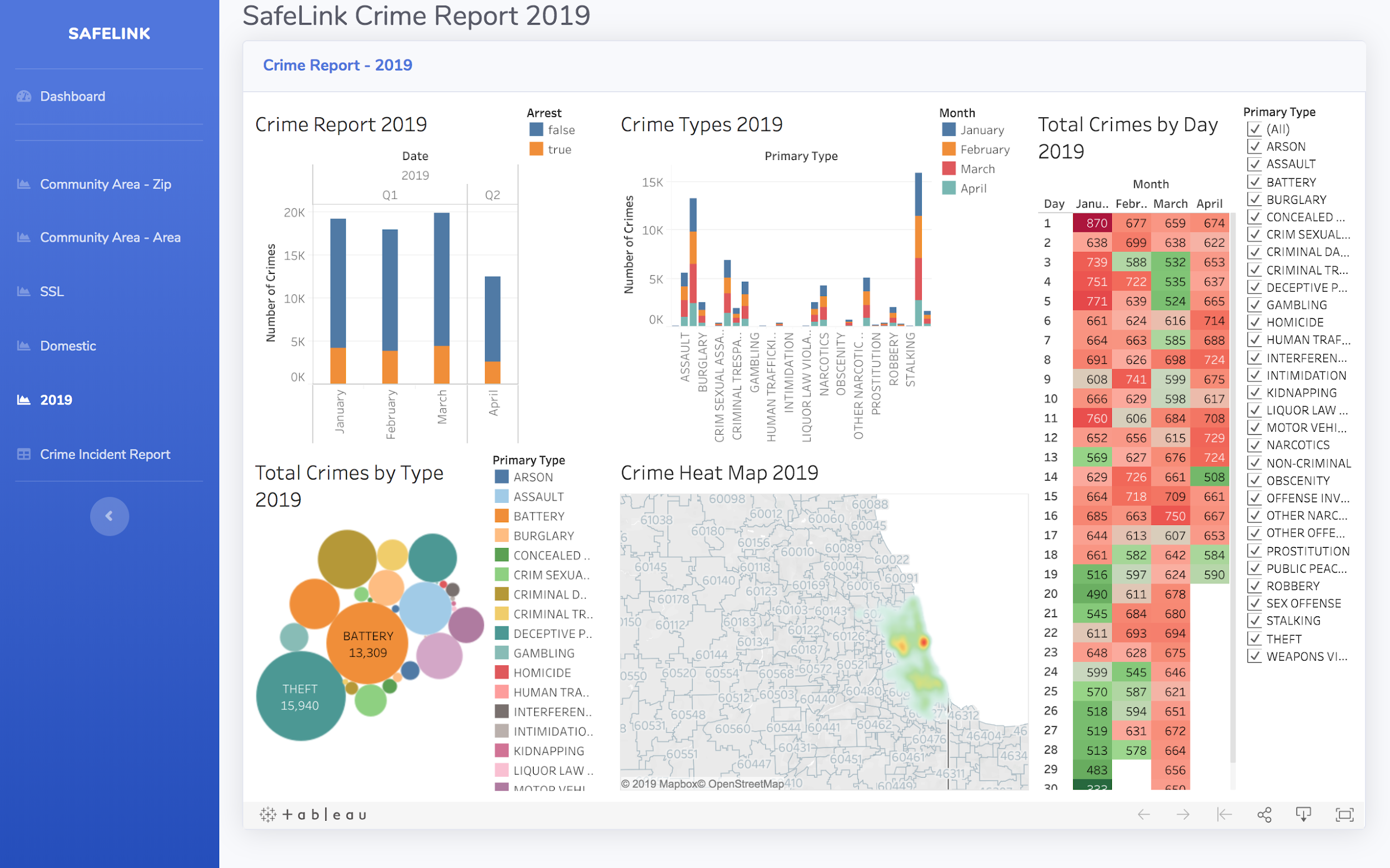


**Figure 9. Screenshot of Community Area by Zip Code**

1. Safety Score by Zip Code: This is a search box where the user can enter the zip code for which he/she wants to check the safety score. SafeLink has an algorithm which takes different parameters like number of crimes, arrests etc and calculates the safety score for that zip code.
2. Zip Code: This is a bar graph which has number of crimes on the y-axis and zip code on the x-axis. The user can select any zip code from this chart and the rest of the dashboard will be dynamically generated accordingly. This helps the user compare all the zip codes with number of crimes at a glance and see more details about that any particular zip code.
3. Crime Heat Map: This is a crime heat map.Yellow indicates the area with low number of crimes and red indicates the areas with higher number of crimes. When a user selects a zip code, the heat map is generated for that zip code. This helps the user to visualize the crime rate on the map of Chicago.
4. Fire Stations and Police Stations: This shows a list of fire stations and police stations are plotted on the map of Chicago. When the user selects any zip code, it shows the details about that zip code. This helps the user to find fire stations and police stations in their zip codes.
5. Number of Crimes with Year: This is a line chart with number of crimes on y-axis and year on x-axis. This helps user see the crime trend with year for complete Chicago and any particular zip code.
6. Crime Type with Number and Arrests: This is a bar chart which has primary type of crime on the y-axis and number of crimes on the x-axis. The crimes are filtered by if the arrest was made or not. This help the user to be aware of the crimes that happen in their areas and stay safe.

### 2019

To know the crime statistics from the beginning of the year 2019, the user can go to the navigation link 2019. The below picture shows the screenshot of crime report of 2019. This page gives month wise crime statistics of Chicago for the current year 2019.

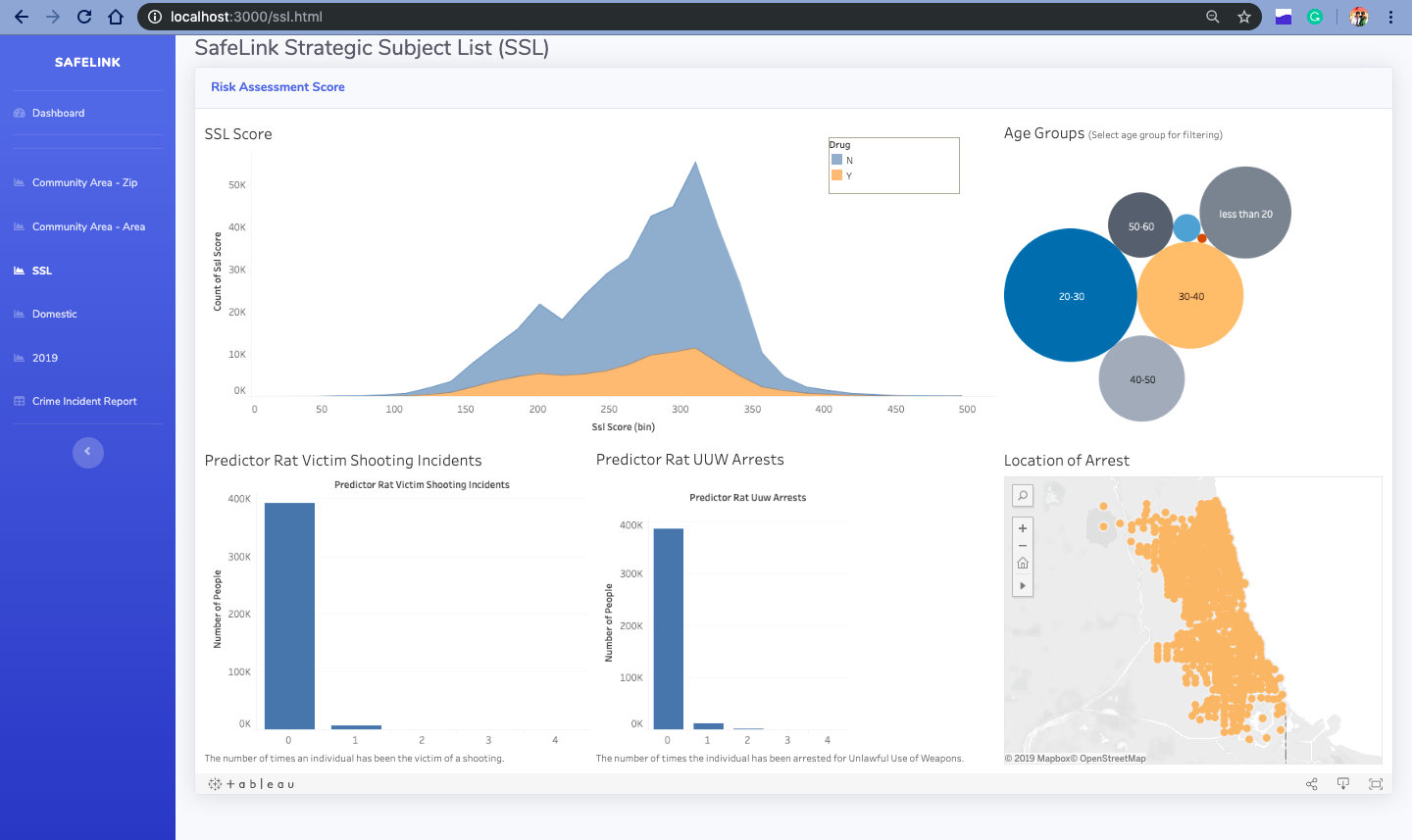
****

**Figure 10. Screenshot of “Crime Statistics for 2019” page**

1. Crime Report 2019 - This is a stacked bar graph that shows the total number of crimes month wise for 2019. The stacks indicate whether arrests were carried out on those crimes. The user will get a better picture of the crime variations by month for the current year.
2. Crime Types 2019 - This is a stacked bar graph and shows the ratio of crimes every month. The bars indicate the type of crime and different colors are used to represent the months. In the above graph january, february, march, april are represented by blue, orange, red, and green respectively. This helps the user get an overview of the number of crimes by type and month.
3. Total Crimes By Day 2019 - This chart shows the total number of crimes for every day of the current year. The color varies from dark green, light green, light red, dark red to indicate lowest, medium-low, medium-high, highest crime numbers. From the chart, users can infer that crime rates tend to be the highest on holidays like New years day on January 1st. The user can get the number of crimes based on crime type by using the filter to filter based on crime type.
4. Total Crimes By Types 2019 - This is a bubble chart and shows the total number of crimes by crime type from the beginning of the current year-2019. The size of the bubbles varies based on the number of crimes. Bigger the bubble, higher is the number of crimes. From this graph the user can infer about the most often committed crimes this year. The user can get the number of crimes by crime type by using the filter.
5. Crime Heat Map 2019 - This is crime heat map for the current year. The green indicates areas with lower crime rates, yellow indicates areas with medium crime rates, and red indicates areas with highest crime rates. This is based on the total crimes for the current year 2019. The user gets an overall picture of the dangerous and safe areas as per the crime rates for the current year.

### SSL (Strategic Subject List)

This dashboard help the users to know their risk of getting involved in a shooting. The SSL scores reflect an individual’s probability of being involved in a shooting incident either as a victim or an offender. The scale for the scores is from 0 (extremely low risk) to 500 (extremely high risk). The picture below shows a screenshot of SSL dashboard. This dashboard is useful for police, lawmakers and citizens of Chicago. New laws can be made to prevent crimes and make Chicago safe.



**Figure 11. Screenshot of “SSL” page**

1. SSL Score: This is an area chart which shows Strategic Subject List Score. It is labeled by drugs yes and drugs no. It has number of people on the y-axis and SSL Score on the x-axis. It shows the number of people who can be involved in shootings.
2. Age Groups: This a bubble chart age and number of people. The number of people is represented by the size of bubble. User can filter particular age groups from this chart. This can help lawmakers to make laws for age groups who are more likely to get involved in shooting incidents.
3. Predictor Victim Shooting Incidents: This bar chart shows number of people by predictor victim shooting incidents. It tells the number of times an individual has been involved in a shooting incident. This can help police that particular people are getting involved in shooting and find patterns.
4. Predictor Rat UUW Arrests: This is bar chart. It shows the number of times an individual has been arrested for Unlawful Use of Weapons. This chart can help the lawmakers change laws if people are getting arrested more number of times for using weapons.
5. Location of Arrest: This is a map which shows the location of arrests. This can tell where more number of arrests take place.

## Crime Incident Report

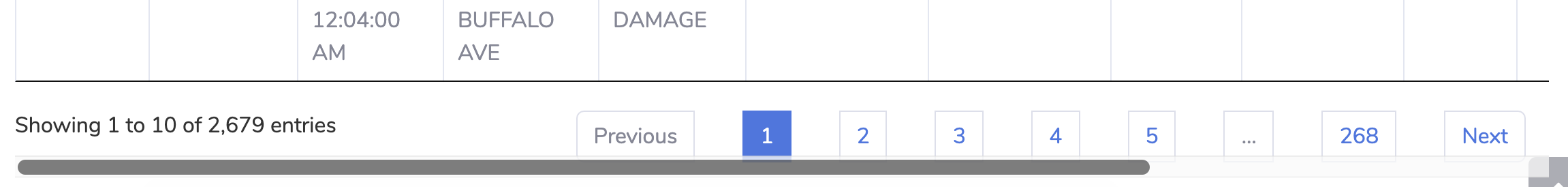
### About

Data is being extracted from multiple data sources and at the end of it all, if the user wants to view details on crimes, the tables page takes care of that. As an individual who wants to know what exactly the police database has details regarding crime such as Homicide etc. will give the user an idea of how a crime is categorized as Homicide and how severe the crime can be.

### 

### Pagination

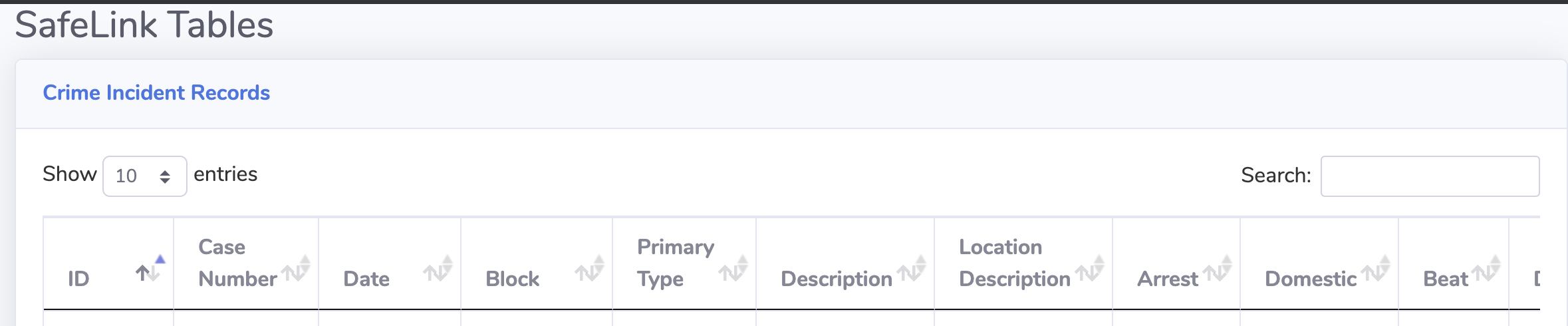
The Crime Incident Report page (SafeLink Tables) was built using DataTables a free tool to make HTML tables interactive. The table was designed to have a pagination feature on the bottom that provides the page number along with so users find it easier to skip pages if needed or use the PREV and NEXT buttons to navigate through the content.



**Figure 12. Pagination**

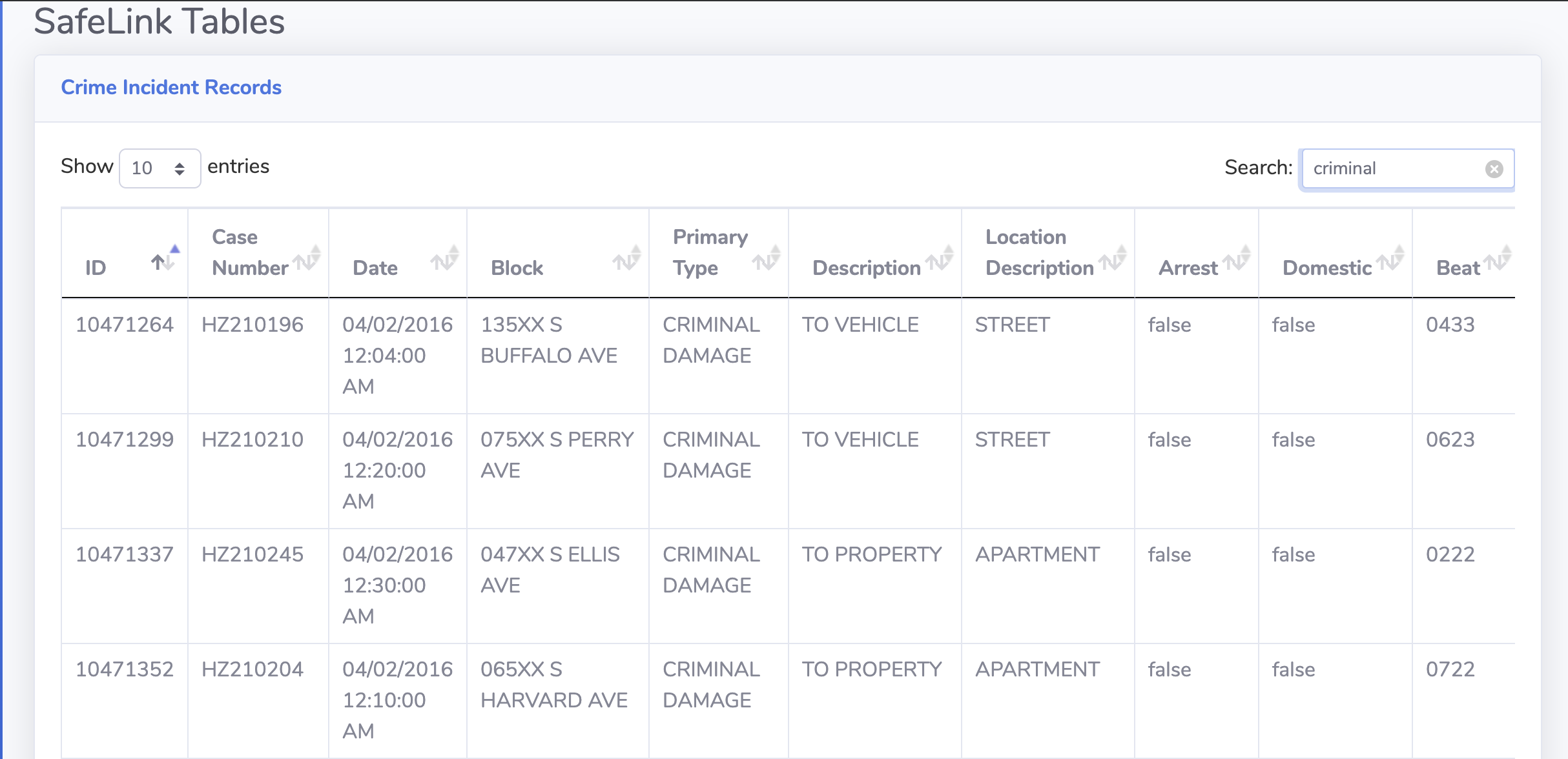
### Search

Crime Incident Report Page has a Search feature that appears on the top right corner.



**Figure 13. Search**

This search feature is built such that based on the entry entered in the Search text box, the entire table is scanned for it and those rows that make that text box entry present will be displayed, as shown below:



**Figure 14. Search Result**

Based on the search, wherever the text *criminal* appears, the table is updated to display all those rows that has the entry *criminal*. The search feature is flexible and is not case sensitive.

### Data

Content for the table is extracted based on the controller code written using the MVC architecture of Node.JS. The controller code as shown below:



**Figure 15. Controller Code for Data Retrieval**

The controller code is written to first connect to MongoDB, retrieve the data for a given time period which in this case is for the month of April and the details are then from the result object, the entries are sent as a response with the intention to render *table.ejs* with the content of *entries.*

In *table.ejs* (located under *views* folder of *app\_server*) the object is read and displayed in HTML table format as shown below:

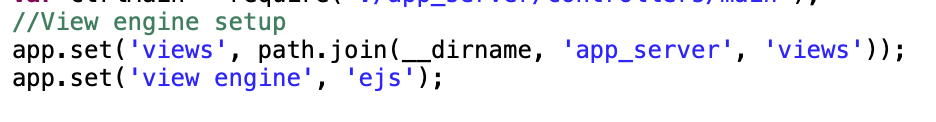


**Figure 16. Code in table.ejs**

### EJS

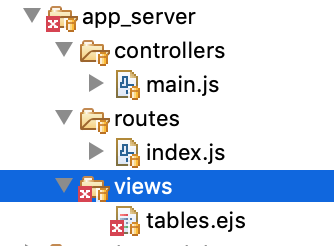
EJS is a templating engine that makes rendering templates easier. In most applications, HTML templates are reusable, but due to the limitations of JavaScript in most cases creating a new template to display as a result seems like the easier option. With the introduction of templating engines such as Jade, EJS, Pug reusing templates is possible.

To use EJS for our project, we defined the view engine to EJS as shown below:



**Figure 17. View Engine Setup**

By setting the view engine to EJS, in *app.js* and providing the folder path for views it sets the application to retrieve the EJS templates from the *views* folder. Given below is the folder structure:



**Figure 18. Folder Structure**

### Contribution to Project Story

After the viewing the charts under the different tabs, if users want to see what are the details of the reported crimes, the *Crime Incident Report* displays the relevant information. Since our application retrieves data from different sources, sometimes we need a resource that holds all the information together making it easier to view the content.

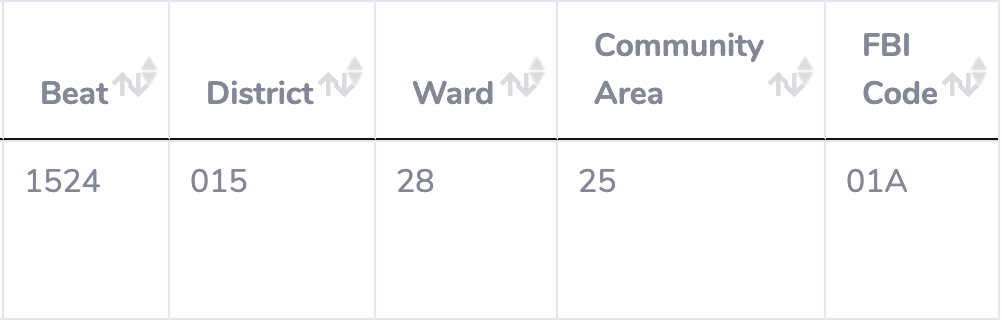
Taking a row as an example, each row contains information that is simple and straightforward to understand irrespective of the user’s profile. If the user was one from the general public, then the following content will make more sense to that user:



**Figure 19. Content from Single Row**

It clearly indicates the date and time of the reported crime, along with the location and other details such as whether it was considered as a *Domestic* crime and if any arrests (*Arrest)* were made for it. This gives the user of general public a simple yet detailed report of the crime.

If the user is member from the police force, then the other details such as:



**Figure 20. Police Related Information**

*District* code, *Ward* number, *Community Area, FBI Code* are useful information. It helps the member of the police force to use the related information by either looking for other crimes within the same *Ward* and so on.

## Design Patterns

The following design patterns were implemented in the application:

Organization

* Dashboard
  + What
    - The first page, or the landing page, of the application is the dashboard that has aggregated data of the top most popular crimes for the month
  + When
    - Incoming flow of data both from MongoDB and Tableau
  + Why
    - User gains insights to the current state of the month
    - Provide details of the state of crime

Layout and List

* Visual Framework
  + What
    - The navigation bar is fixed on the left it does not change from page to page to provide a sense of continuity and familiarity
  + When
    - The content of the page is what changes but the layout remains consistent
  + Why
    - Helps page content stand out
    - The eye does not have to scan the whole page over again
* Center Stage
  + What
    - The content is in the center of the page so the eye goes there first
    - The visualizations are the most important component of the application
  + When
    - When the page is loaded it is the center
  + Why
    - The user is guided to the most important part of the page
* Tiled Sections
  + What
    - On the dashboard there are four modules at the top that provide textual information regarding the monthly stats
  + When
    - Makes the stats for the month easily visible once the page is loaded
  + Why
    - Provide information in a different form than just relying on graphs
* Striped Rows
  + What
    - When the user goes to the crime incident report page they are presented with a table of the records
  + When
    - To provide visual distinction from one record to the next
  + Why
    - Easier for the eye to read at a glance

Actions

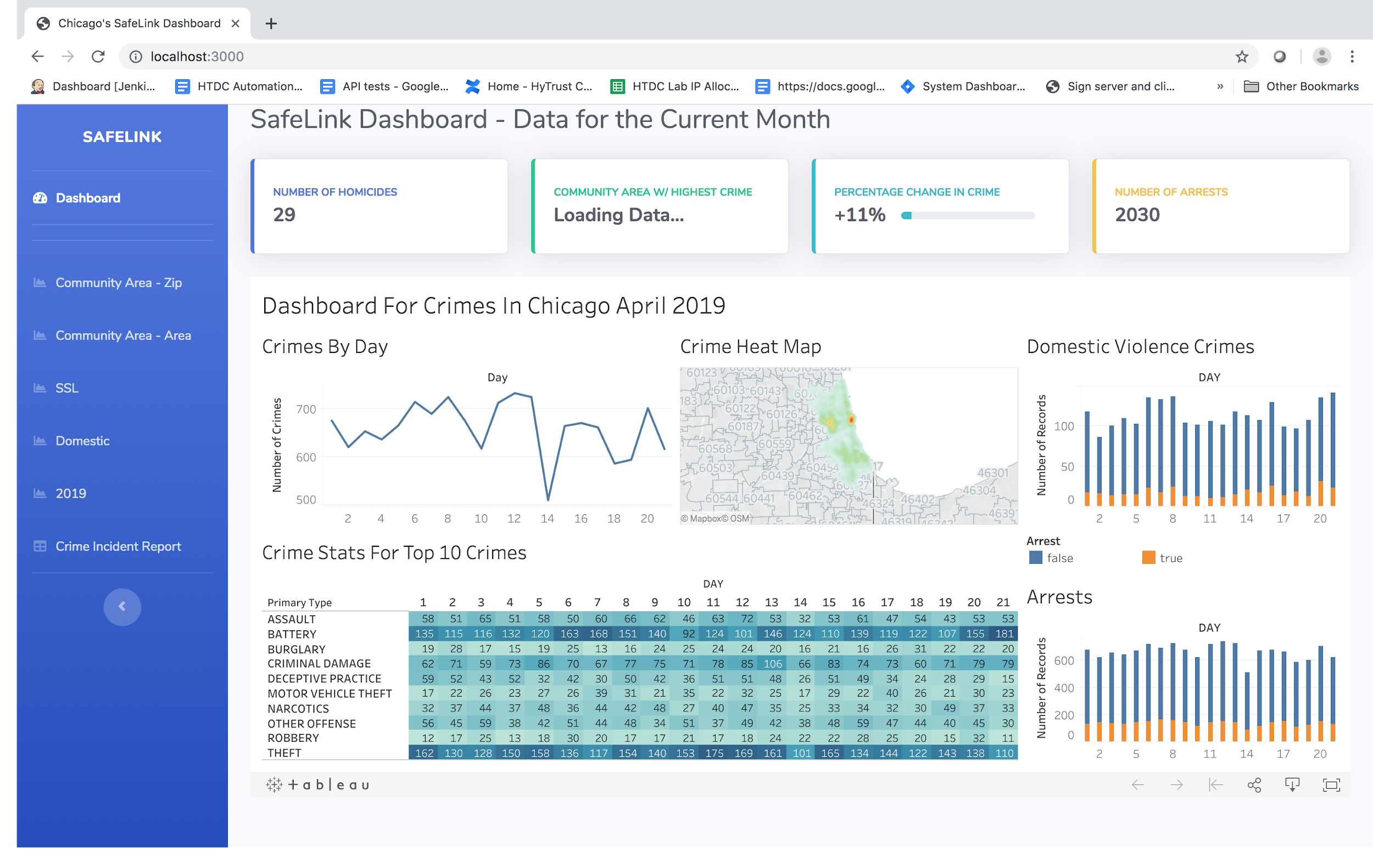
* Buttons
  + The buttons are made clear of that they are buttons by rounding the edges and keeping the look of them consistent with industry standard
  + An example of a button can be found on the Community Area - Zip Code page. The user must enter a zip code and then hit the blue search button to complete the action
  + What
    - The sole button in the application is given a distinct color so it is easily recognizable as clickable
  + When
    - The user must enter the button when they are done entering a zip code
  + Why
    - To complete the action
* Hover Tools
  + What
    - The user is able to hover over the various data points in the graph to filter and choose what they want to focus on
  + When
    - Allows data to be highlighted and brought to the forefront
  + Why
    - Allowing this gives the user a greater degree of freedom to gather what they need from the data
* Progress
  + What
    - On the Community Area - Zip Code page when the data is being loaded from the database the progress is communicated to the user
  + When
    - Data is being transferred as per user request
  + Why
    - To allow for better UX, the user understands the state of the system

User Input

* Text Input Controls - Structured Format
  + What
    - When entering a zip code the length is limited to 5
  + When
    - Input data must be a valid zip code
  + Why
    - Reduces data entry
* Input - Same-Page Error Message
  + What
    - When the user enters an invalid zip code (5 letters instead of 5 numbers)
  + When
    - User might enter a bad zip code
  + Why
    - Provides immediate feedback to the user that the entered text was not in the right format

## Running the Application

1. Navigate to **/project**
2. Install the packages through package.json using “npm install”
3. Start the application by running the following command: ‘node app.js’
4. Open up a browser and navigate to “<http://localhost:3000/>”
5. Click the “dashboard” tab on the sidebar or navigate to <http://localhost:3000/index.html>.
6. The following screen will be visible. The top 4 modules will take some time to load data as the data involves aggregation queries on mongodb. It shows as “Loading data” as shown below.
7. View various modules by clicking on the link in navigation bar.
8. Please wait for a minute, after clicking on *Crime Incident Report* tab on the navigation bar. The table takes a while to update the content along with the interactive features.



**Figure 21. Dashboard screenshot**