I) Introduction:

In the wake of the COVID-19 pandemic, there has been a remarkable resurgence in people's passion for travel, which shows no signs of reducing. Thanks to the rapid advancements in transportation and technology, individuals now have the opportunity to explore previously less explored territories. However, the allure of the unknown comes with its risks. Every area can present potential dangers, from minor inconveniences like pickpocketing to more serious threats like violent crimes. That's why we propose to create a web application that doesn't just cater to tourists but also serves as an invaluable resource for residents. We aim to empower individuals with vital information about their neighborhoods' safety and crime trends.

The Crime and Accident Database is a comprehensive repository that supports research on various aspects of crime, including its causes, consequences, legal and societal implications, litigation, and evolving trends. It houses a wealth of resources, from correctional and law enforcement trade publications to crime reports and insightful blogs, all relevant to our initiative.

Our project is centered around developing an intuitive interface within the BSafe app, designed to offer residents and tourists alike a means of staying informed about potential hazards and, more importantly, equipping them with the knowledge to mitigate risks in the most secure manner possible. BSafe provides access to a wealth of data on crime and accidents spanning various regions. Users can significantly enhance their safety by utilizing BSafe for real-time information, minimizing unforeseen mishaps during their travels or daily lives.

Theory for BSafe:

To benefit from the full range of features offered, users must register with BSafe. This registration allows users to retrieve comprehensive data on criminal and accident-related incidents within specific geographic areas. Each geographical region is color-coded to reflect its crime or accident tendencies. Areas prone to accidents are highlighted in yellow, while parts with less severe crimes are marked in orange. Locations with potentially life-threatening crimes, such as shootings, are prominently displayed in red.

Users must grant consent to share their location to access live information and receive timely notifications. This enables the app to notify users when they enter areas with potential safety concerns, suggesting nearby safe havens like police stations, fire stations, and hospitals. Furthermore, users can access transportation details, including bus, tram, and ferry schedules, helping them plan their journeys more effectively.

Users also have the option to contribute feedback and ratings based on their experiences, both within specific areas and with the BSafe application itself. BSafe will securely store this feedback and usage data to enhance future trend analysis.

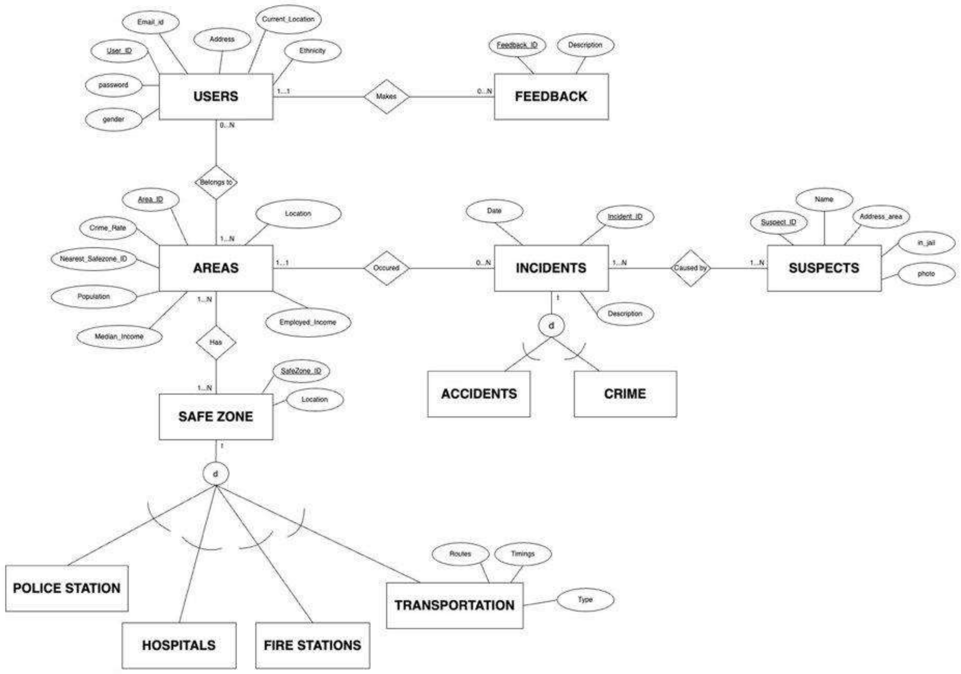
The data collected from users will include visit timings, user pathways, weather conditions, and other relevant factors, all contributing to a more comprehensive understanding of crime and accident patterns. The company will categorize and store user data in various segments. Basic user information will be retained, including UserID, name, address, age, gender, contact number, email address, location, and ethnicity. The collection of age, gender, and ethnicity data aids in identifying and cautioning users about specific crime trends that may target individuals based on these demographic factors.

Other Requirements:

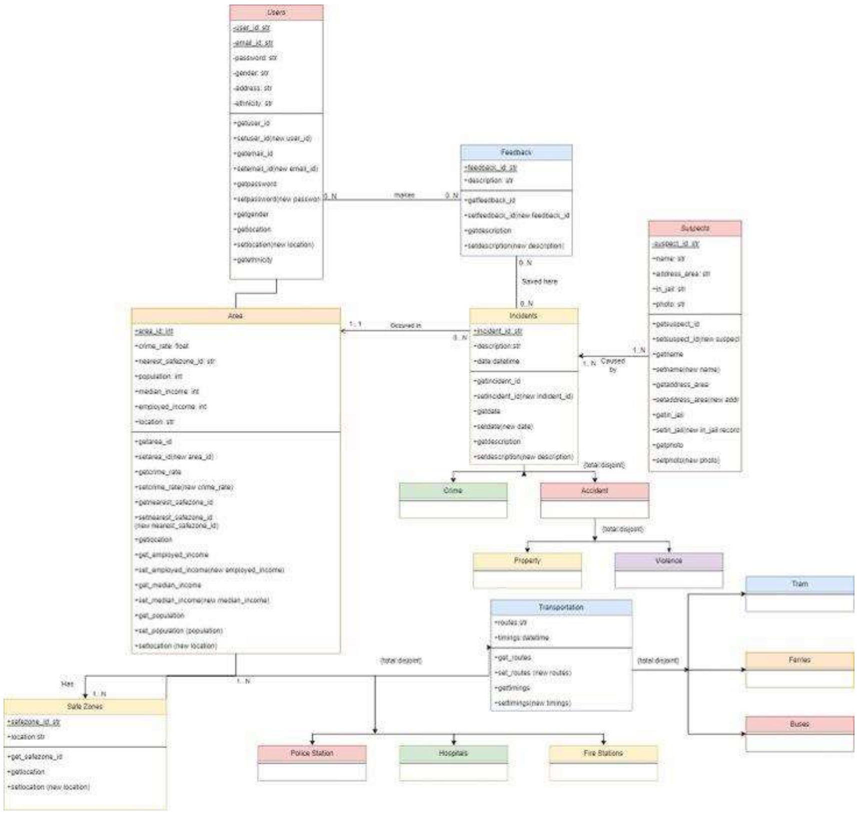
1. An individual can hold 1 account in the application; an application can have many users.
2. Users can access zero to many crimes and accident information.
3. The user can access the safe zones based on one to many locations (Area table); areas must have at least one safe zone. (Type of safe zone)
4. Areas can have one to many crimes and accident information, and crime and accident information should have one area under it
5. Areas can have zero to many transportation services; different transportation can be available for many areas.
6. Users can have one to many criminal details based on location and also based on type of crime for the betterment.

II) Conceptual Data Modelling

1. EER (Enhanced Entity Relationship) Diagram



2. UML Diagram



III) Mapping Conceptual Model to Relational Model

Primary Key- Underlined Foreign Key- *Italicized and Underlined*

- **Users** (User_ID, password, Email_ID, gender, Address, Current_location, Ethnicity)

User_ID is PRIMARY KEY and Email ID is Unique.

- **Feedback** (Feedback_ID, Description, *User_ID*)

Feedback_ID is PRIMARY KEY

FOREIGN KEY User_ID refers to Users; NOT NULL

- **Belong_to**(*User_ID*, *Area_ID*)

FOREIGN KEY User_ID refers to Users; NOT NULL

FOREIGN KEY Area_ID refers to Areas; NOT NULL

User_ID and Area_ID is PRIMARY KEY together

- **Areas** (Area_ID, Crime_Rate, Location, Nearest_Safezone_ID, Population, Median_Income,Employed_Income)

Area_ID is PRIMARY KEY

- **Safezone** (Safezone_ID, Location)

Safezone_ID is the PRIMARY KEY

- **PoliceStation** (*Safezone_ID*)

FOREIGN KEY Safezone_ID refers to Safezone; NOT NULL

- **Hospitals** (*Safezone_ID*)

FOREIGN KEY Safezone_ID refers to Safezone; NOT NULL

- **FireStation** (*Safezone_ID*)

FOREIGN KEY Safezone_ID refers to Safezone; NOT NULL

- **Transportation** (*Safezone_ID*, Routes, Timings, Type)

FOREIGN KEY Safezone_ID refers to Safezone; NOT NULL

- **Has** (*Area_ID*, *Safezone_ID*)

FOREIGN KEY Area_ID refers to Areas; NOT NULL

FOREIGN KEY Safezone_ID refers to Safezone; NOT NULL

Safezone_ID and Area_ID is PRIMARY KEY together

- **Incidents** (Incident_ID, Description, Incident_type, Incident_datetime, Area_ID)

Incident_ID is the PRIMARY KEY

FOREIGN KEY Area_ID refers to Areas; NOT NULL

- **Suspects** (Suspect_ID, Name, Address_Area, In_jail, Photo)

Suspect_ID is PRIMARY KEY

- **Caused_by** (Incident_ID, Suspect_ID)

FOREIGN KEY Incident_ID refers to Incidents; NOT NULL
FOREIGN KEY Suspect_ID refers to Incidents; NOT NULL
Incident_ID and Suspect_ID is PRIMARY KEY together

IV) Implementation in MYSQL

<u>Creation of Tables using Create Command</u>:

```
-- Table: Users
CREATE TABLE Users (
    User_ID INT PRIMARY KEY,
    Password VARCHAR(255), -- assuming a reasonable length
    Email_ID VARCHAR(255) UNIQUE,
    Gender VARCHAR(10), -- assuming 'Male' or 'Female'
    Address VARCHAR(255),
    Current_location VARCHAR(255),
    Ethnicity VARCHAR(50)
);

-- Table: Feedback
CREATE TABLE Feedback (
    Feedback_ID INT PRIMARY KEY,
    Description TEXT,
    User_ID INT NOT NULL,
    FOREIGN KEY (User_ID) REFERENCES Users(User_ID),
    FOREIGN KEY (Area_ID) REFERENCES Areas(Area_ID)
);

-- Table: Belong_to
CREATE TABLE Belong_to (
    User_ID INT,
    Area_ID INT,
    PRIMARY KEY (User_ID, Area_ID),
    FOREIGN KEY (User_ID) REFERENCES Users(User_ID),
    FOREIGN KEY (Area_ID) REFERENCES Areas(Area_ID)
```

```
);

-- Table: Areas
CREATE TABLE Areas (
    Area_ID INT PRIMARY KEY,
    Crime_Rate FLOAT,
    Nearest_Safezone_ID INT,
    Population INT,
    Median_Income INT,
    Employed_Income INT,
    Location VARCHAR(255)
    FOREIGN KEY (Nearest_Safezone_ID) REFERENCES Safezone(Safezone_ID)
);

-- Table: Safezone
CREATE TABLE Safezone (
    Safezone_ID INT PRIMARY KEY,
    Location VARCHAR(255)
);

-- Table: PoliceStation
CREATE TABLE PoliceStation (
    Safezone_ID INT PRIMARY KEY,
    FOREIGN KEY (Safezone_ID) REFERENCES Safezone(Safezone_ID)
);

-- Table: Hospitals
CREATE TABLE Hospitals (
    Safezone_ID INT PRIMARY KEY,
    FOREIGN KEY (Safezone_ID) REFERENCES Safezone(Safezone_ID)
);

-- Table: FireStation
CREATE TABLE FireStation (
    Safezone_ID INT PRIMARY KEY,
    FOREIGN KEY (Safezone_ID) REFERENCES Safezone(Safezone_ID)
);

-- Table: Transportation
CREATE TABLE Transportation (
    Safezone_ID INT PRIMARY KEY,
    Routes VARCHAR(255),
    Timings VARCHAR(50),
    Type VARCHAR(50),
    FOREIGN KEY (Safezone_ID) REFERENCES Safezone(Safezone_ID)
);

-- Table: Has
CREATE TABLE Has (
    Area_ID INT,
    Safezone_ID INT,
    PRIMARY KEY (Area_ID, Safezone_ID),
    FOREIGN KEY (Area_ID) REFERENCES Areas(Area_ID),
    FOREIGN KEY (Safezone_ID) REFERENCES Safezone(Safezone_ID)
);

-- Table: Incidents
CREATE TABLE Incidents (
    Incident_ID INT PRIMARY KEY,
    Description TEXT,
    Incident_datetime DATETIME,
    Incident_type VARCHAR(50)
);
```

-- Table: Suspects

```
CREATE TABLE Suspects (
    Suspect_ID INT PRIMARY KEY,
    Name VARCHAR(255),
    Address_Area VARCHAR(255),
    In_jail BOOLEAN,
    Photo BLOB -- assuming binary large object for storing images
);
```

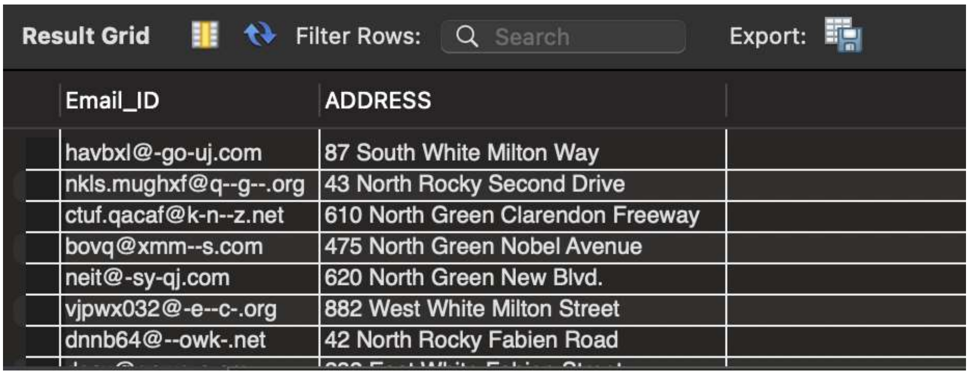
-- Table: Caused_by

```
CREATE TABLE Caused_by (
    Incident_ID INT,
    Suspect_ID INT,
    PRIMARY KEY (Incident_ID, Suspect_ID),
    FOREIGN KEY (Incident_ID) REFERENCES Incidents(Incident_ID),
    FOREIGN KEY (Suspect_ID) REFERENCES Suspects(Suspect_ID)
);
```

Queries:

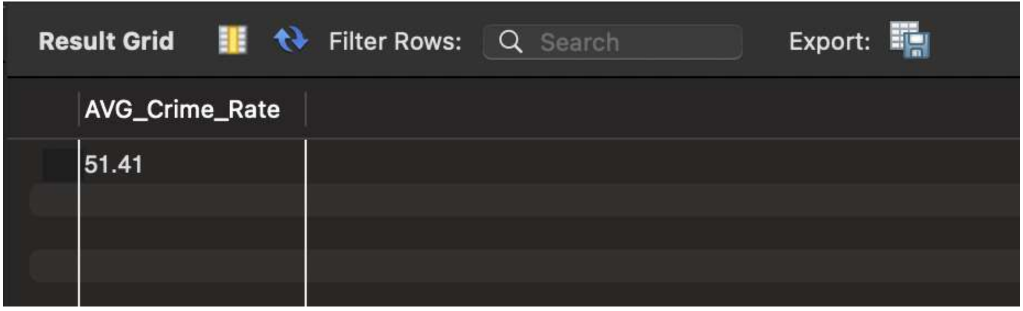
-- Simple Query:
1. Retrieve the names and addresses of all users.

```
SELECT Email_ID, ADDRESS FROM USERS;
```



Email_ID	ADDRESS
havbxl@-go-uj.com	87 South White Milton Way
nkls.mughxf@q--g--.org	43 North Rocky Second Drive
ctuf.qacaf@k-n--z.net	610 North Green Clarendon Freeway
bovq@xmm--s.com	475 North Green Nobel Avenue
neit@-sy-qj.com	620 North Green New Blvd.
vjpwx032@-e--c-.org	882 West White Milton Street
dnnb64@--owk-.net	42 North Rocky Fabien Road

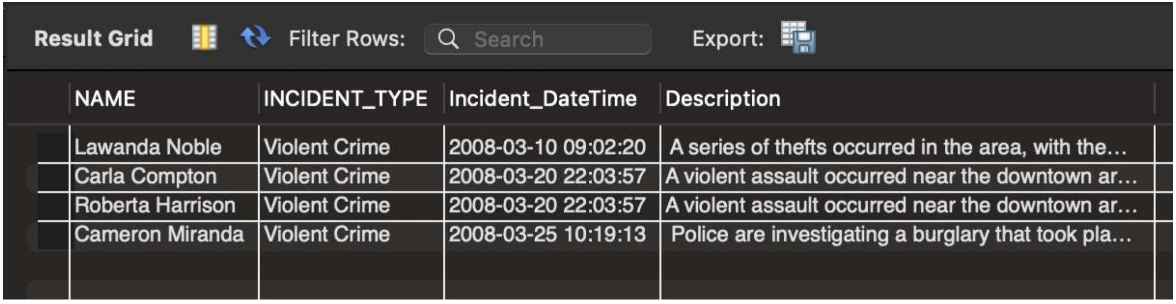
-- Aggregate:
2. Find the average crime rate in all areas.
SELECT Round(AVG(Crime_Rate),2) AS AVG_Crime_Rate FROM AREAS;



AVG_Crime_Rate
51.41

-- Inner Join/Outer Join:
3. List the incidents along with the corresponding suspect names and descriptions.

```
SELECT
   S.NAME, I.INCIDENT_TYPE, I.Incident_DateTime, I.Description
FROM
   CAUSED_BY C
      INNER JOIN
   INCIDENTS I ON I.INCIDENT_ID = C.INCIDENT_ID
      INNER JOIN
   SUSPECTS S ON C.SUSPECT_ID = S.SUSPECT_ID
WHERE
   MONTH(Incident_DateTime) = 03
ORDER BY Incident_DateTime;
```

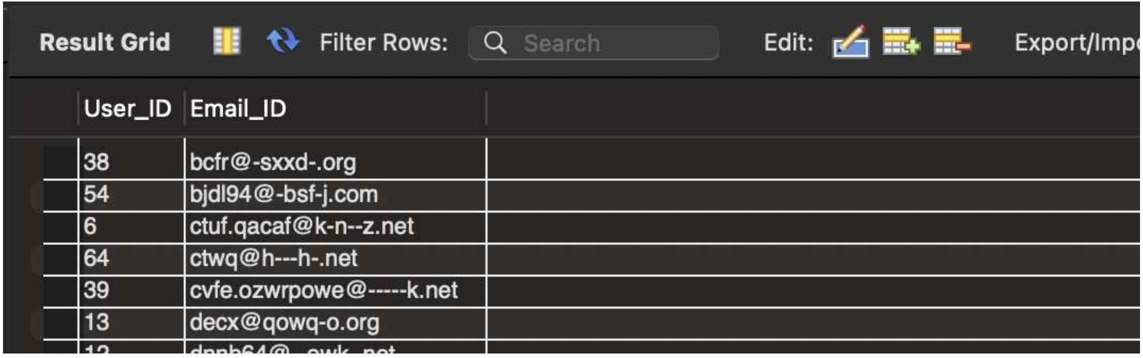


NAME	INCIDENT_TYPE	Incident_DateTime	Description
Lawanda Noble	Violent Crime	2008-03-10 09:02:20	A series of thefts occurred in the area, with the...
Carla Compton	Violent Crime	2008-03-20 22:03:57	A violent assault occurred near the downtown ar...
Roberta Harrison	Violent Crime	2008-03-20 22:03:57	A violent assault occurred near the downtown ar...
Cameron Miranda	Violent Crime	2008-03-25 10:19:13	Police are investigating a burglary that took pla...

-- Nested Query:

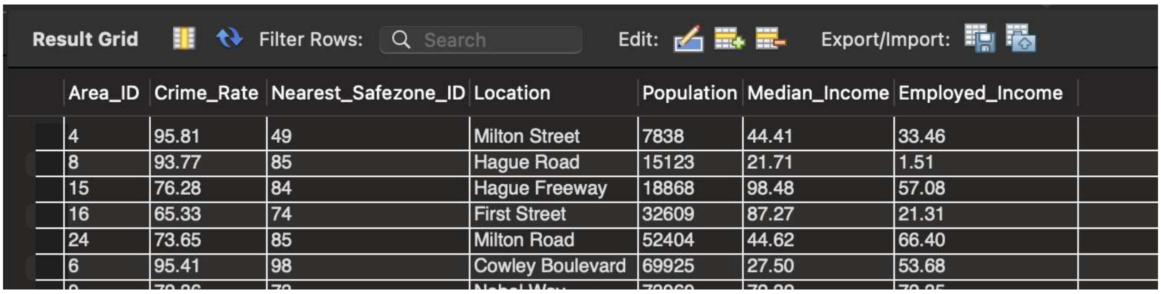
4. Retrieve the user IDs and email addresses of users who have not provided feedback.

SELECT User_ID,Email_ID
FROM USERS
WHERE User_ID not in (SELECT User_ID from Feedback);



User_ID	Email_ID
38	bcfr@-sxxd-.org
54	bjdl94@-bsf-j.com
6	ctuf.qacaf@k-n--z.net
64	ctwq@h---h-.net
39	cvfe.ozwrpowe@-----k.net
13	decx@qowq-o.org

5. Identify all areas where the crime rate is higher than the average crime rate across all areas.

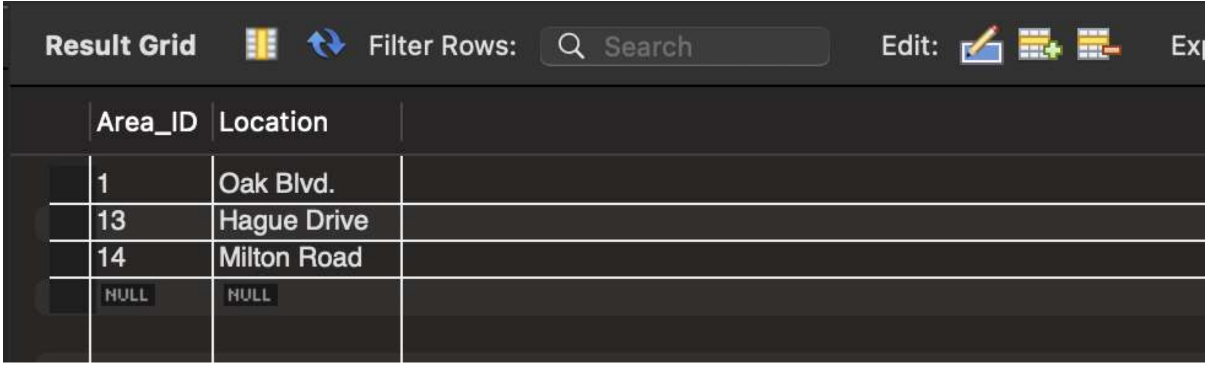


Area_ID	Crime_Rate	Nearest_Safezone_ID	Location	Population	Median_Income	Employed_Income
4	95.81	49	Milton Street	7838	44.41	33.46
8	93.77	85	Hague Road	15123	21.71	1.51
15	76.28	84	Hague Freeway	18868	98.48	57.08
16	65.33	74	First Street	32609	87.27	21.31
24	73.65	85	Milton Road	52404	44.62	66.40
6	95.41	98	Cowley Boulevard	69925	27.50	53.68

-- Correlated Queries:

6. Retrieve the Areas with at least 5 Incidents.

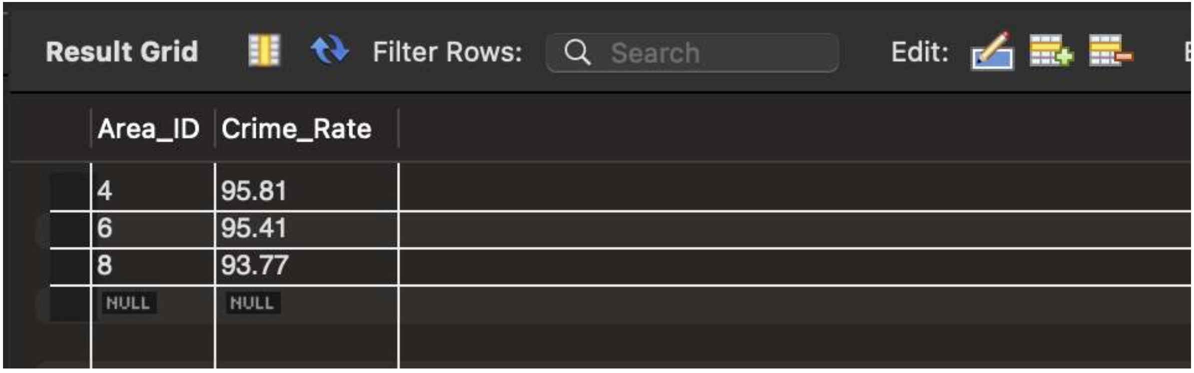
SELECT A.Area_ID, A.Location
FROM AREAS A WHERE 5<=
(SELECT COUNT(*)
FROM Incidents I
WHERE I.Area_ID = A.Area_ID);



Area_ID	Location
1	Oak Blvd.
13	Hague Drive
14	Milton Road
NULL	NULL

7. Retreive the three highest Crime_Rate areas

SELECT A1.Area_ID, A1.Crime_Rate FROM areas A1 WHERE 3 >
(SELECT COUNT(*)
FROM areas A2
WHERE A1.Crime_Rate < A2.Crime_Rate);

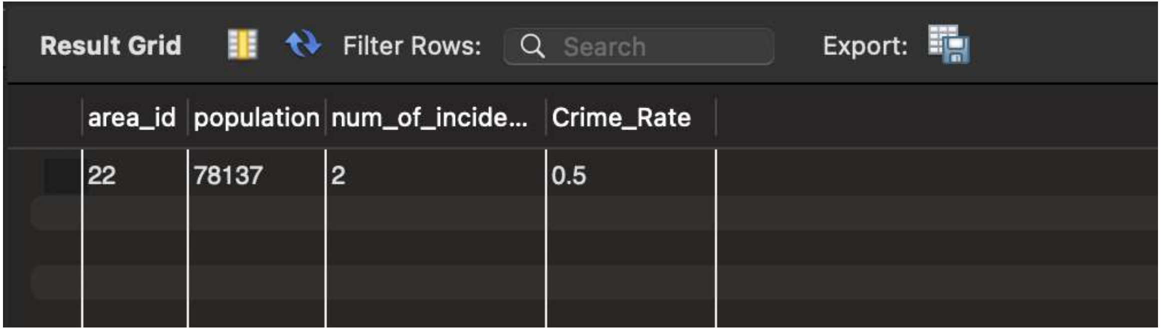


Area_ID	Crime_Rate
4	95.81
6	95.41
8	93.77
NULL	NULL

-- ALL

8. Retreive the name, population and number of Incidents of all the Areas with lowest crime_rate.

select a.area_id, a.population, count(i.incident_id) as num_of_incidents, a.Crime_Rate
from areas a
inner join incidents i on i.area_id=a.area_id
where a.Crime_Rate<=ALL(select Crime_Rate from areas)
group by a.area_id;

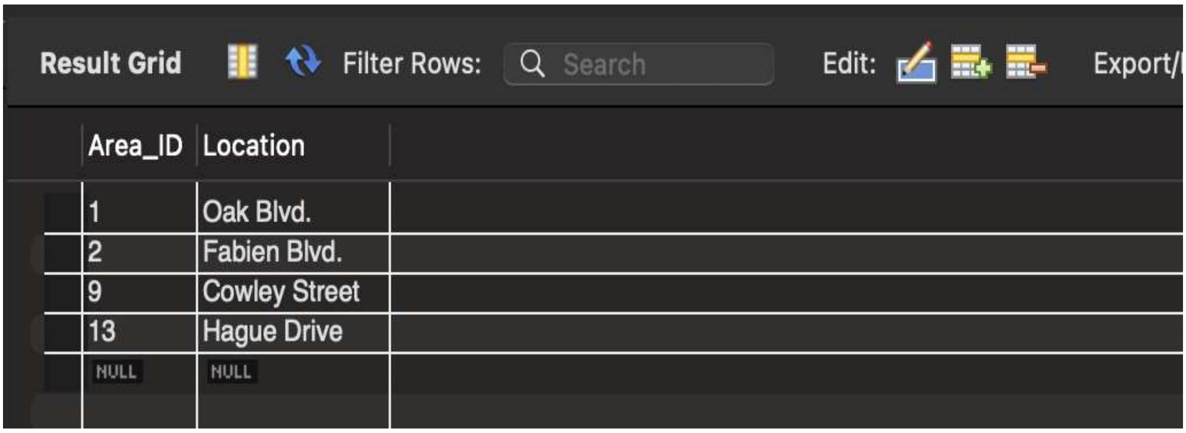


area_id	population	num_of_incide...	Crime_Rate
22	78137	2	0.5

-- ANY

9. retreive the names of the areas who do not have the highest crime rate and population > 90000

SELECT Area_ID, Location
FROM Areas
WHERE population>90000 and Crime_Rate <
ANY(SELECT Crime_Rate FROM Areas);

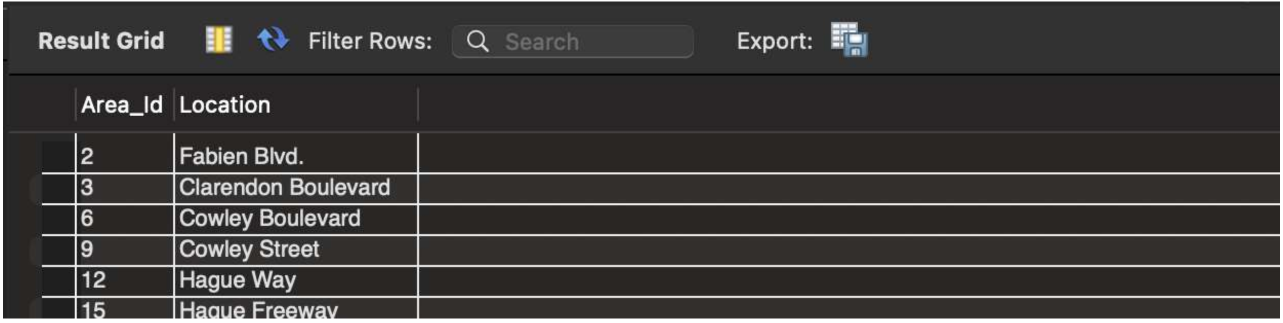


Area_ID	Location
1	Oak Blvd.
2	Fabien Blvd.
9	Cowley Street
13	Hague Drive
NULL	NULL

-- EXISTS/ NOT EXISTS

10. retrieve areas which do not have hospitals.

```
Select Area_Id, Location
from Areas A
Where NOT EXISTS (select * from Has H, Hospitals hp
where h.safezone_id=hp.safezone_id and h.area_id=a.area_id);
```

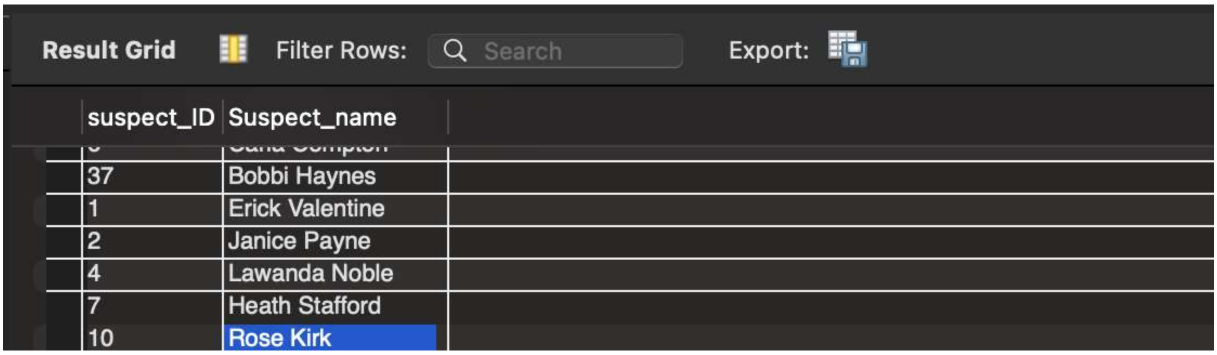


Area_Id	Location
2	Fabien Blvd.
3	Clarendon Boulevard
6	Cowley Boulevard
9	Cowley Street
12	Hague Way
15	Hague Freeway

-- UNION

11. select suspects who are either in jail or who have greater than 3 incidents

```
(select c.suspect_ID, s.name as Suspect_name
from caused_by c, suspects s
where c.suspect_ID=s.suspect_ID
group by c.suspect_ID
having count(Incident_ID)>=3)
union all
(select suspect_ID, name as Suspect_name
from suspects
where in_jail=1);
```

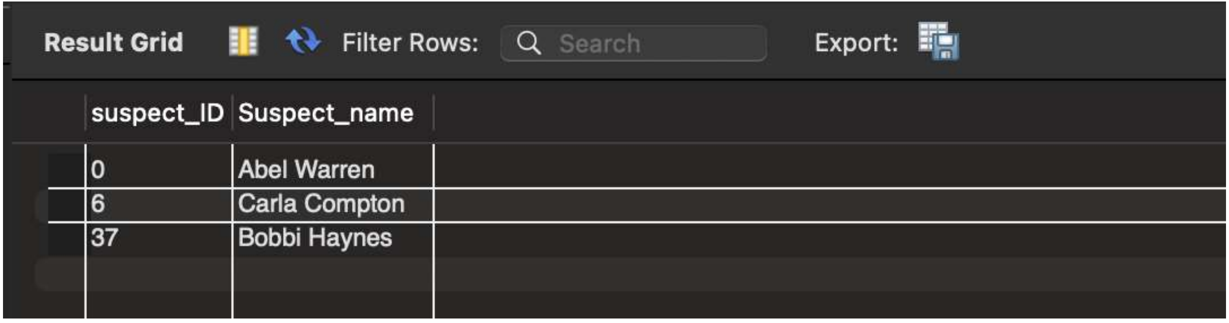


suspect_ID	Suspect_name
6	Carla Compton
37	Bobbi Haynes
1	Erick Valentine
2	Janice Payne
4	Lawanda Noble
7	Heath Stafford
10	Rose Kirk

-- Subqueries in select and from

12. select the suspects with 3 or more incidents

```
select c.suspect_ID, (select name from suspects s where s.suspect_ID=c.suspect_ID) as
Suspect_name
from (select suspect_ID, count(Incident_ID) as num_of_Incidents
from caused_by
group by suspect_ID) c
where num_of_Incidents>=3;
```



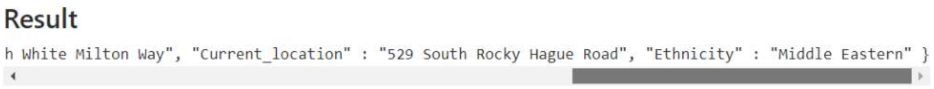
suspect_ID	Suspect_name
0	Abel Warren
6	Carla Compton
37	Bobbi Haynes

V) Implementation in NoSQL

Tables namely Areas, Users, Suspects and Incidents have been created in MongoDB playground. The following MongoDB Queries were executed:

1.Query to Update User 0

```
// updating current location of user
db.users.update(
{"User_ID": 0},
{
$set: {
"Current_location": "529 South Rocky Hague Road"
}
}
);
db.users.find({"User_ID": 0});
```

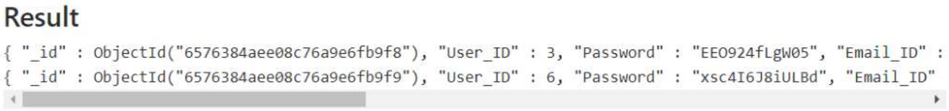


Result

```
h White Milton Way", "Current_location" : "529 South Rocky Hague Road", "Ethnicity" : "Middle Eastern" }
```

2.Query to Display Users Who are Female and Asian

```
db.users.find({
"Gender": "Female",
$and: [
{"Ethnicity":"Asian"}]
})
```

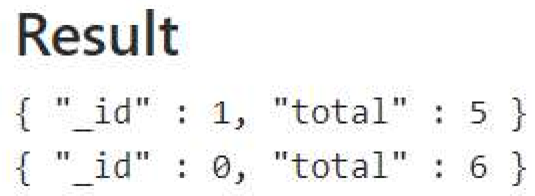


Result

```
{ "_id" : ObjectId("6576384aee08c76a9e6fb9f8"), "User_ID" : 3, "Password" : "EEO924fLgW05", "Email_ID" :
{ "_id" : ObjectId("6576384aee08c76a9e6fb9f9"), "User_ID" : 6, "Password" : "xsc4I6J8iULBd", "Email_ID"
```

3.Query using aggregate to get count of suspects in jail

```
db.suspects.aggregate(
[{ $group: { _id: "$In_jail", total: { $sum: 1 } } },
{ $sort: { total: 1 } } ]);
```

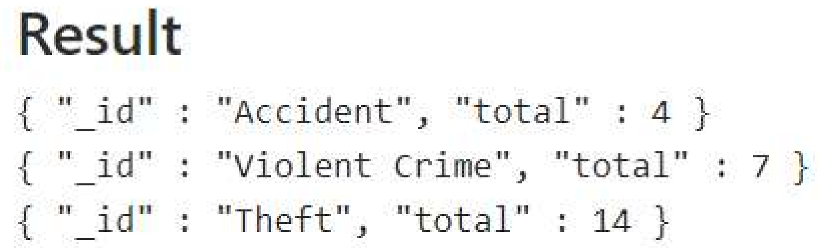


Result

```
{ "_id" : 1, "total" : 5 }
{ "_id" : 0, "total" : 6 }
```

4.Query using aggregate to get frequencies of incidents based on Incident Types

```
db.incidents.aggregate(
[{ $group: { _id: "$Incident_type", total: { $sum: 1 } } },
{ $sort: { total: 1 } } // -1 ==> desc
]);
```

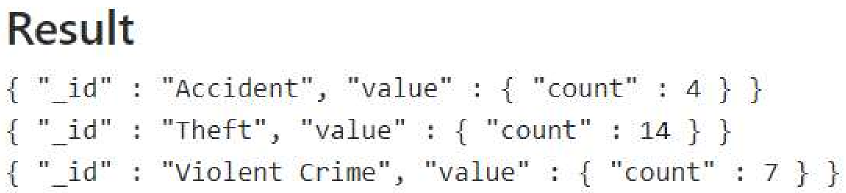


Result

```
{ "_id" : "Accident", "total" : 4 }
{ "_id" : "Violent Crime", "total" : 7 }
{ "_id" : "Theft", "total" : 14 }
```

5.Query using a map-reduce pipeline to get frequencies of incidents on Incident Types

```
db.incidents.mapReduce(
  // Map function
  function() {
    var INCIDENT_TYPE = this.Incident_type;
    emit(INCIDENT_TYPE, {count: 1});
  },
  // Reduce function
  function(INCIDENT_TYPE, values) {
    var count = 0;
    values.forEach(function (value) {
        count += value.count;
    });
    return { count: count};
},
  // Options
{
    out: "incident_counts", // Output collection name
    finalize: function(key, reducedValue) {
        return { count: reducedValue.count};
    }
} ). find();
```



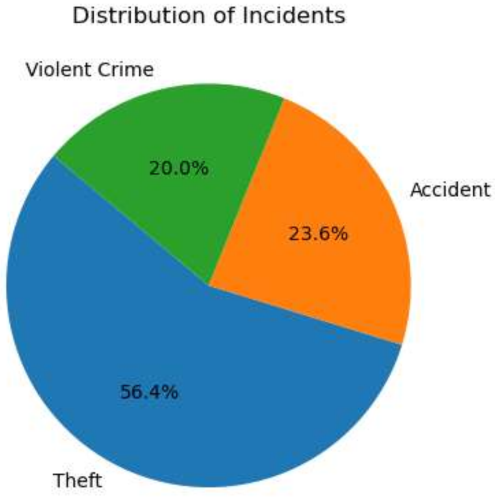
Result

```
{ "_id" : "Accident", "value" : { "count" : 4 } }
{ "_id" : "Theft", "value" : { "count" : 14 } }
{ "_id" : "Violent Crime", "value" : { "count" : 7 } }
```

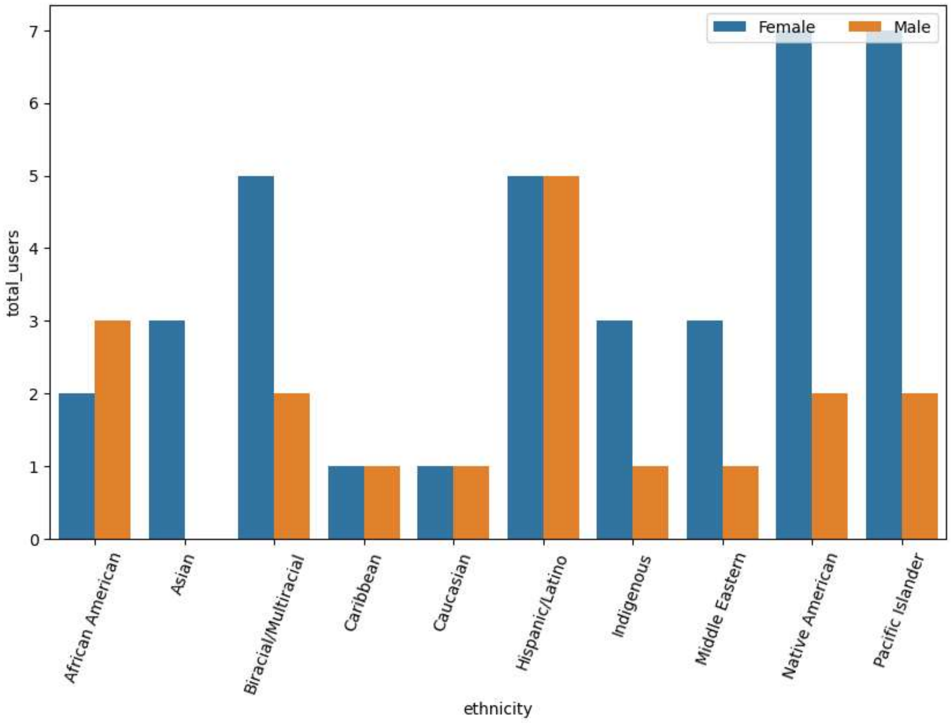
VI) Application in Python

The MySQL Connector allows the Jupyter Notebook to connect to the database. Cursor.connection() was used to run a variety of SQL queries, and the resulting query outputs were transformed into dataframes for additional analysis. Visualizations aimed at providing significant insights into the data were produced using libraries such as matplotlib and seaborn.

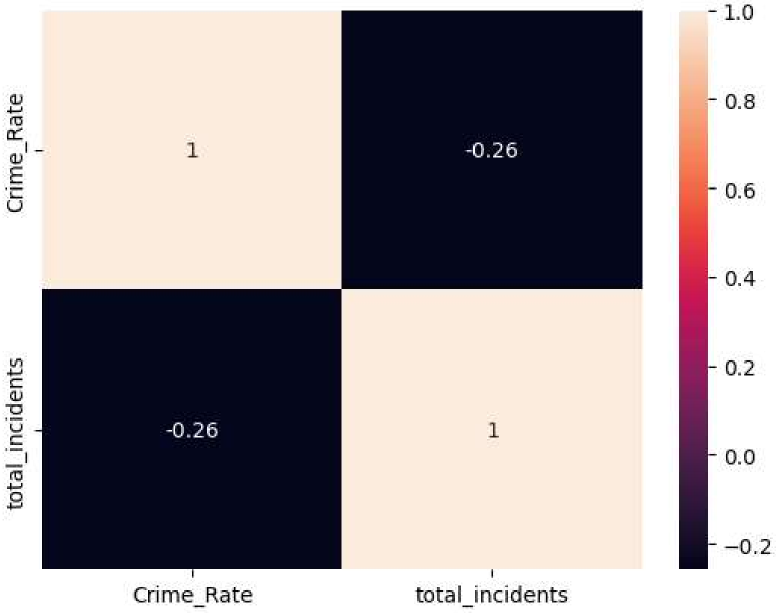
Graph 1. Types and frequencies of incidents reported



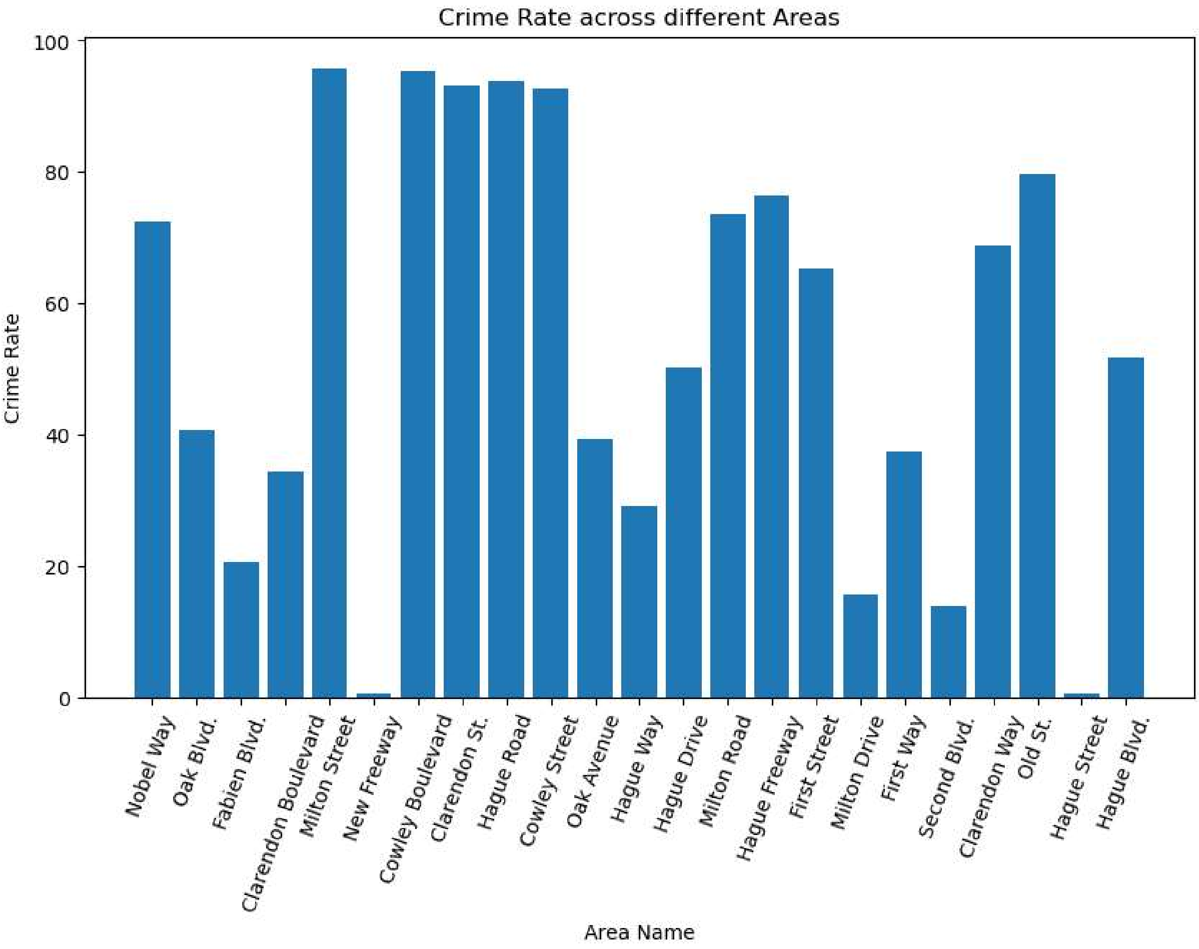
Graph 2. Distribution of users based on Gender and Ethnicity



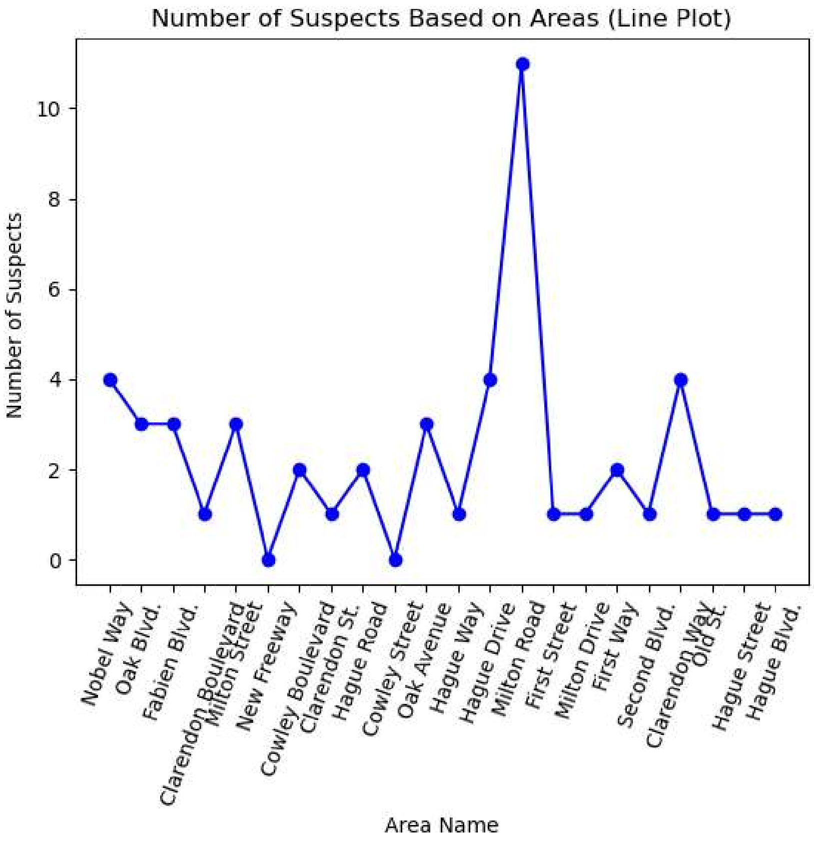
Graph 3. Correlation between crime rates and the occurrence of incidents in specific areas



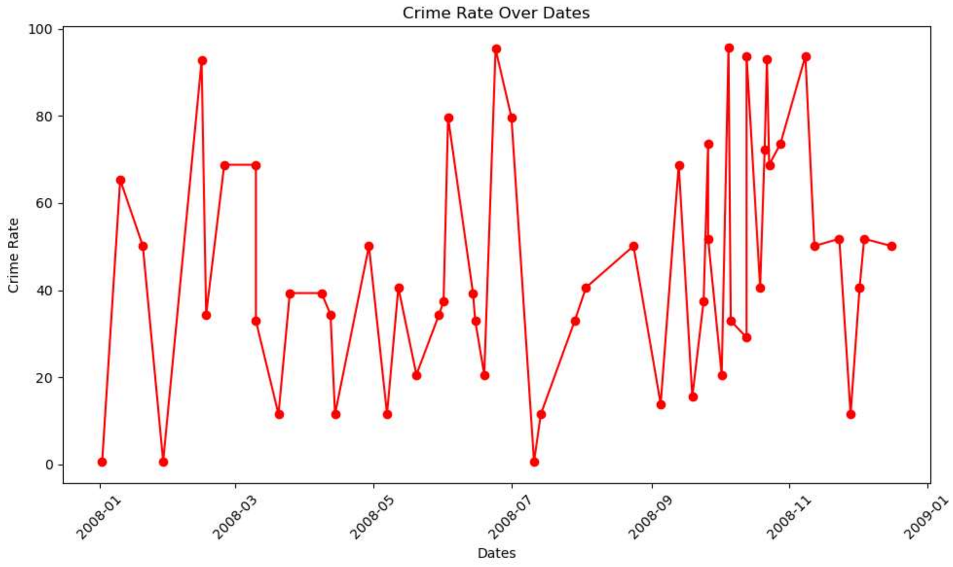
Graph 4. Crime rate across different areas



Graph 5. Number of suspects are there, and the distribution of their locations



Graph 6. Overall trend in crime rates over the observed time-period



VII) Summary and Improvement:

In response to the increased interest in travel post-COVID-19, our proposed Crime and Accident Database, integrated into the BSafe app, addresses the need for a comprehensive safety resource for both residents and tourists. The database contains a wealth of crime-related information, providing users with valuable insights into neighborhood safety and crime trends.

Recommendations for Improvement:

1. **Enhanced User Registration:** To optimize user engagement, consider enhancing the user registration process, ensuring it remains user-friendly while capturing essential information. This step is crucial for delivering personalized safety recommendations based on individual profiles.

2. **Expansion of Safety Notifications:** Improve the notification system to include not only potential safety concerns but also real-time updates on local events, weather conditions, and other factors that might impact users' safety. This enhancement would provide users with a holistic view of their surroundings.

Concerns and Further Study:

1. **Privacy and Data Security:** Given the sensitive nature of the data collected, it's imperative to prioritize and communicate robust privacy measures. Conduct further studies on data security protocols to ensure user information remains confidential and secure, addressing potential concerns regarding the collection and storage of personal data.

2. **User Experience Optimization:** Regularly assess and optimize the user interface and experience within the BSafe app to ensure seamless navigation and user satisfaction. This includes ongoing usability testing and feedback incorporation to refine the app's features and functionalities.

3. **Community Engagement:** Encourage active user participation by fostering a sense of community within the app. Implement features that allow users to connect, share safety tips, and engage in discussions about their experiences. This could further enhance the platform's effectiveness in promoting safety awareness.

In conclusion, the integration of the Crime and Accident Database into the BSafe app is a significant step towards providing users with a valuable safety resource. Further improvements, such as enhanced user registration and expanded safety notifications, coupled with addressing privacy concerns, will contribute to the overall success and user satisfaction of the BSafe app. Regular updates and community engagement efforts will ensure the platform remains relevant and effective in promoting safety.