Thomas McLinden and Victor Liu

12/3/24

CMPSC 463, Section 1

Professor Janghoon Yang

Project 2: Crime Tracker

Goal

The goal of this project was to utilize the skills and knowledge of this course to create an application to help solve a real-world problem. In this case, we created a crime tracker application, where users could enter any zip code in Philadelphia County to view its crime statistics.

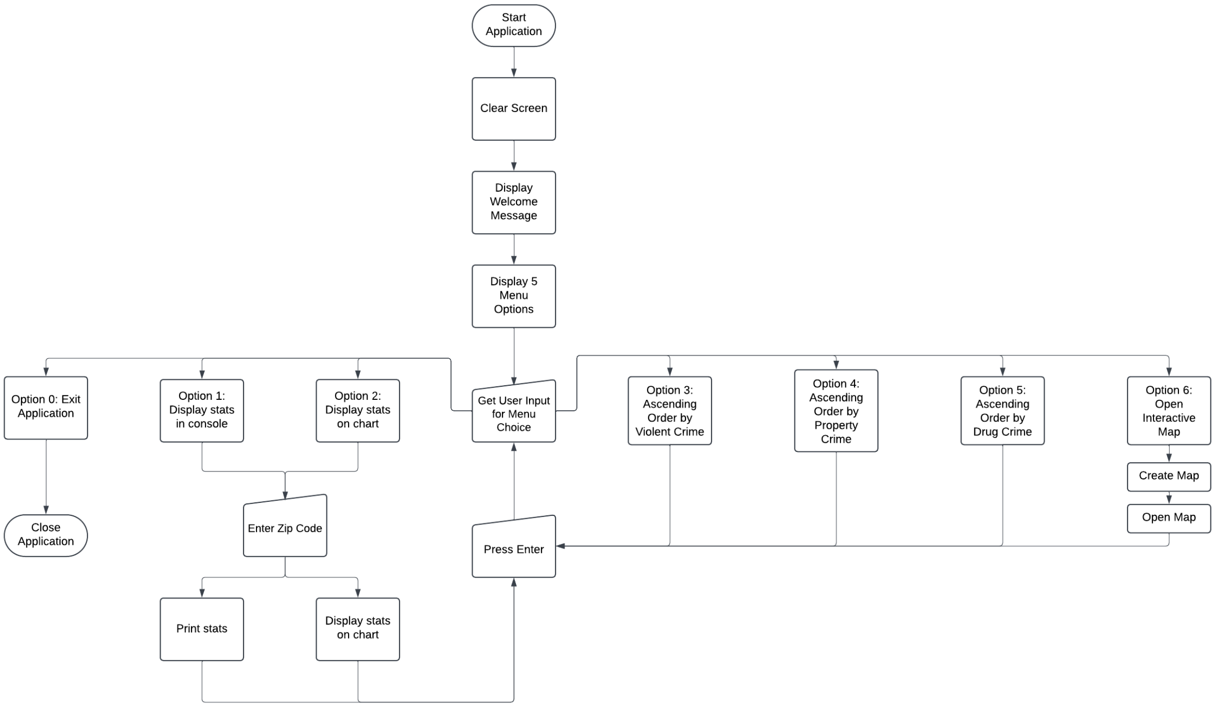
Significance

This application is primarily intended for residents, students, and employees of Philadelphia. By allowing users to obtain the crime statistics of any Philadelphia zip code, users can make decisions on areas they deem unsafe and plan accordingly.

Installation and Instruction to Use:

The latest version of python will need to be installed with the *matplotlib*, *pandas*, *folium*, and *geopy* libraries. The GitHub repository can then be cloned to use the application.

Structure of the Code:

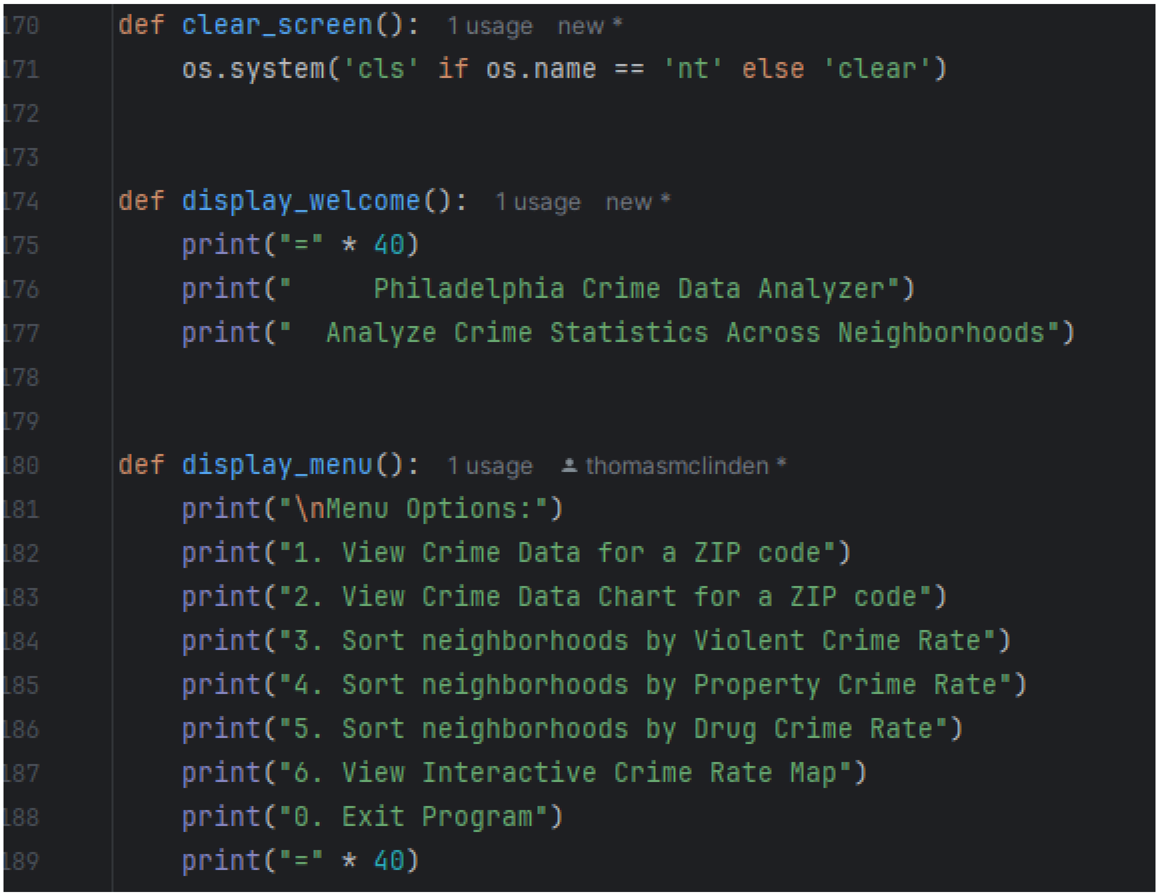


Start Application
Clear Screen
Display Welcome Message
Display 5 Menu Options
Option 0: Exit Application
Option 1: Display stats in console
Option 2: Display stats on chart
Get User Input for Menu Choice
Option 3: Ascending Order by Violent Crime
Option 4: Ascending Order by Property Crime
Option 5: Ascending Order by Drug Crime
Option 6: Open Interactive Map
Create Map
Open Map
Close Application
Enter Zip Code
Press Enter
Print stats
Display stats on chart

The application begins by initializing the lists containing the zip codes, town names, and their crime statistics. The options are printed for the user on the console and a while loop is activated, where users select options on what functions to run. The user also has the option to close the program, which will break the loop.

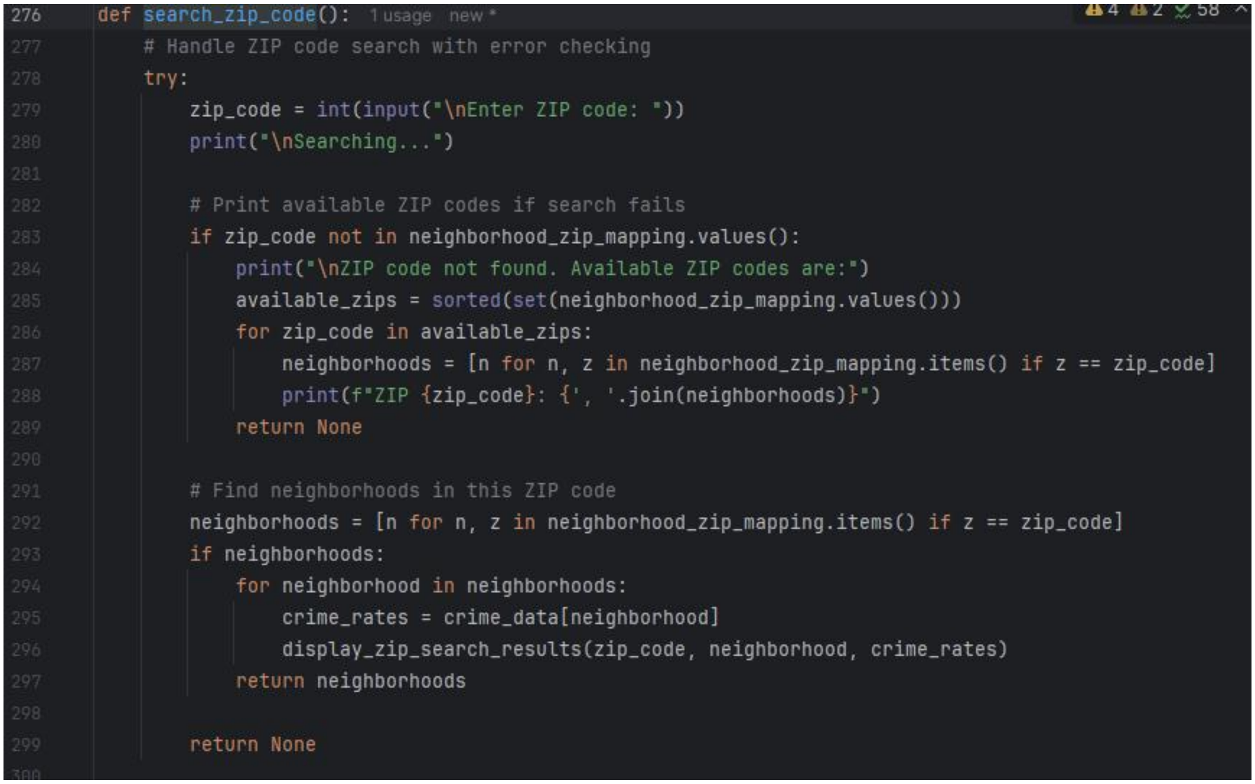
List of Functionalities and Verification Results:

- The console prints the options available to the user



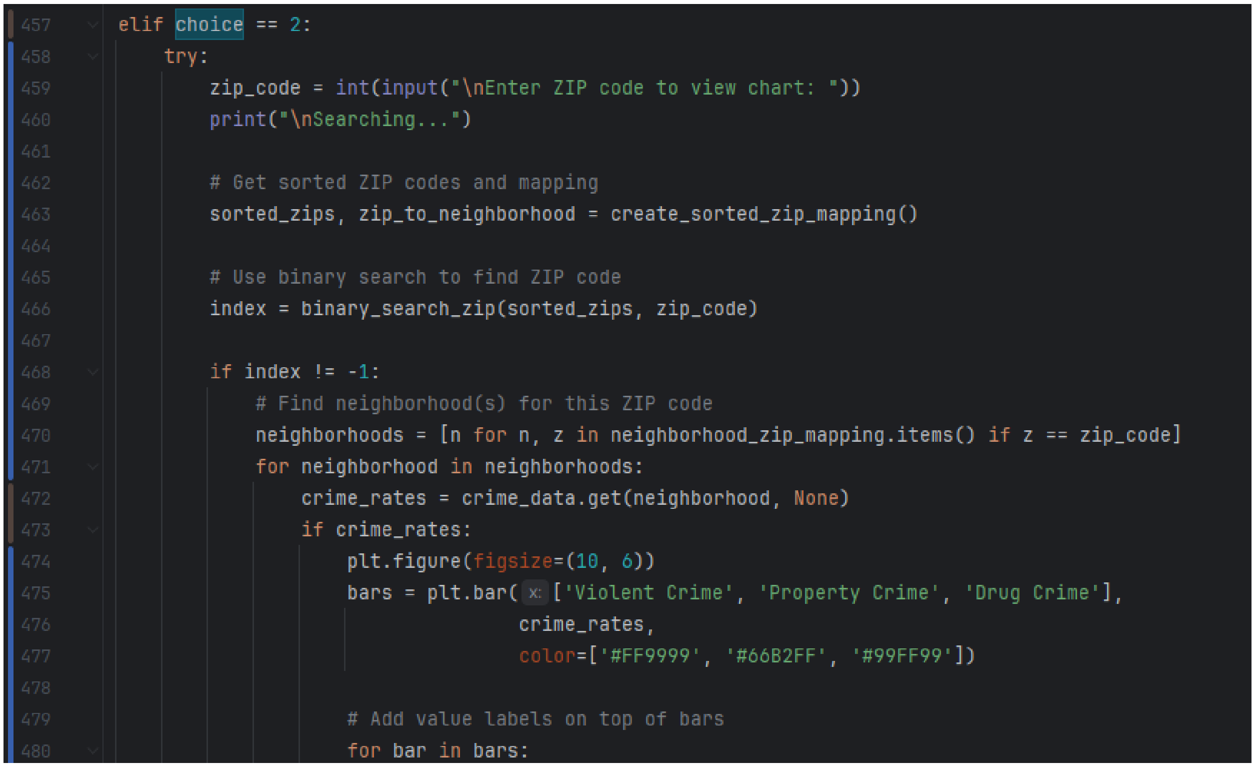
```python
def clear_screen():
    os.system('cls' if os.name == 'nt' else 'clear')


def display_welcome():
    print("=" * 40)
    print("     Philadelphia Crime Data Analyzer")
    print("  Analyze Crime Statistics Across Neighborhoods")


def display_menu():
    print("\nMenu Options:")
    print("1. View Crime Data for a ZIP code")
    print("2. View Crime Data Chart for a ZIP code")
    print("3. Sort neighborhoods by Violent Crime Rate")
    print("4. Sort neighborhoods by Property Crime Rate")
    print("5. Sort neighborhoods by Drug Crime Rate")
    print("6. View Interactive Crime Rate Map")
    print("0. Exit Program")
    print("=" * 40)
```

- Print statistics in the console:



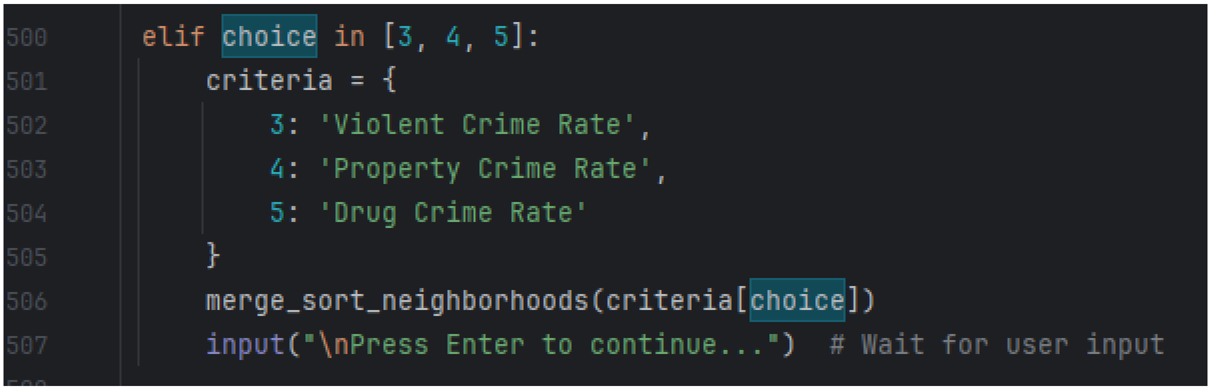
```python
def search_zip_code():
    # Handle ZIP code search with error checking
    try:
        zip_code = int(input("\nEnter ZIP code: "))
        print("\nSearching...")

        # Print available ZIP codes if search fails
        if zip_code not in neighborhood_zip_mapping.values():
            print("\nZIP code not found. Available ZIP codes are:")
            available_zips = sorted(set(neighborhood_zip_mapping.values()))
            for zip_code in available_zips:
                neighborhoods = [n for n, z in neighborhood_zip_mapping.items() if z == zip_code]
                print(f"ZIP {zip_code}: {', '.join(neighborhoods)}")
            return None

        # Find neighborhoods in this ZIP code
        neighborhoods = [n for n, z in neighborhood_zip_mapping.items() if z == zip_code]
        if neighborhoods:
            for neighborhood in neighborhoods:
                crime_rates = crime_data[neighborhood]
                display_zip_search_results(zip_code, neighborhood, crime_rates)
            return neighborhoods

        return None
```

- Display statistics on a chart:

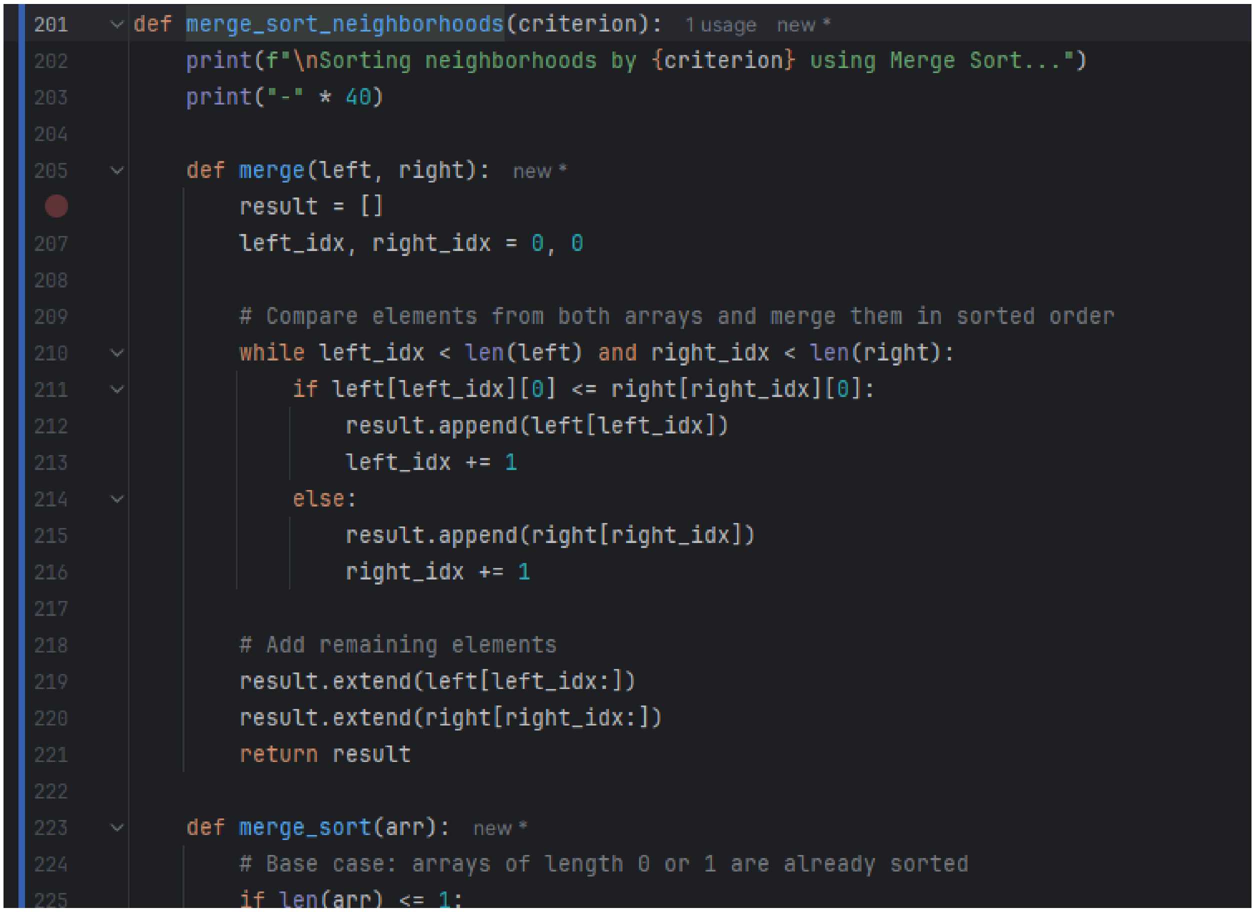


```python
elif choice == 2:
    try:
        zip_code = int(input("\nEnter ZIP code to view chart: "))
        print("\nSearching...")

        # Get sorted ZIP codes and mapping
        sorted_zips, zip_to_neighborhood = create_sorted_zip_mapping()

        # Use binary search to find ZIP code
        index = binary_search_zip(sorted_zips, zip_code)

        if index != -1:
            # Find neighborhood(s) for this ZIP code
            neighborhoods = [n for n, z in neighborhood_zip_mapping.items() if z == zip_code]
            for neighborhood in neighborhoods:
                crime_rates = crime_data.get(neighborhood, None)
                if crime_rates:
                    plt.figure(figsize=(10, 6))
                    bars = plt.bar(['Violent Crime', 'Property Crime', 'Drug Crime'],
                                   crime_rates,
                                   color=['#FF9999', '#66B2FF', '#99FF99'])

                    # Add value labels on top of bars
                    for bar in bars:
```

- Sort by Violent Crime, Property Crime, or Drug Crime:

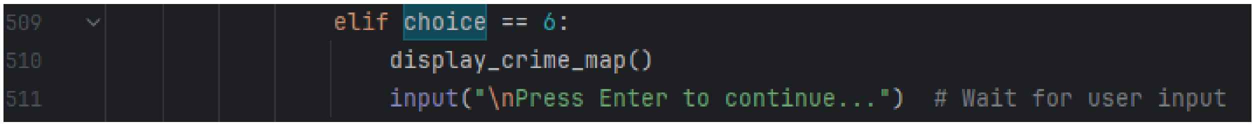


```python
    elif choice in [3, 4, 5]:
        criteria = {
            3: 'Violent Crime Rate',
            4: 'Property Crime Rate',
            5: 'Drug Crime Rate'
        }
        merge_sort_neighborhoods(criteria[choice])
        input("\nPress Enter to continue...")  # Wait for user input
```

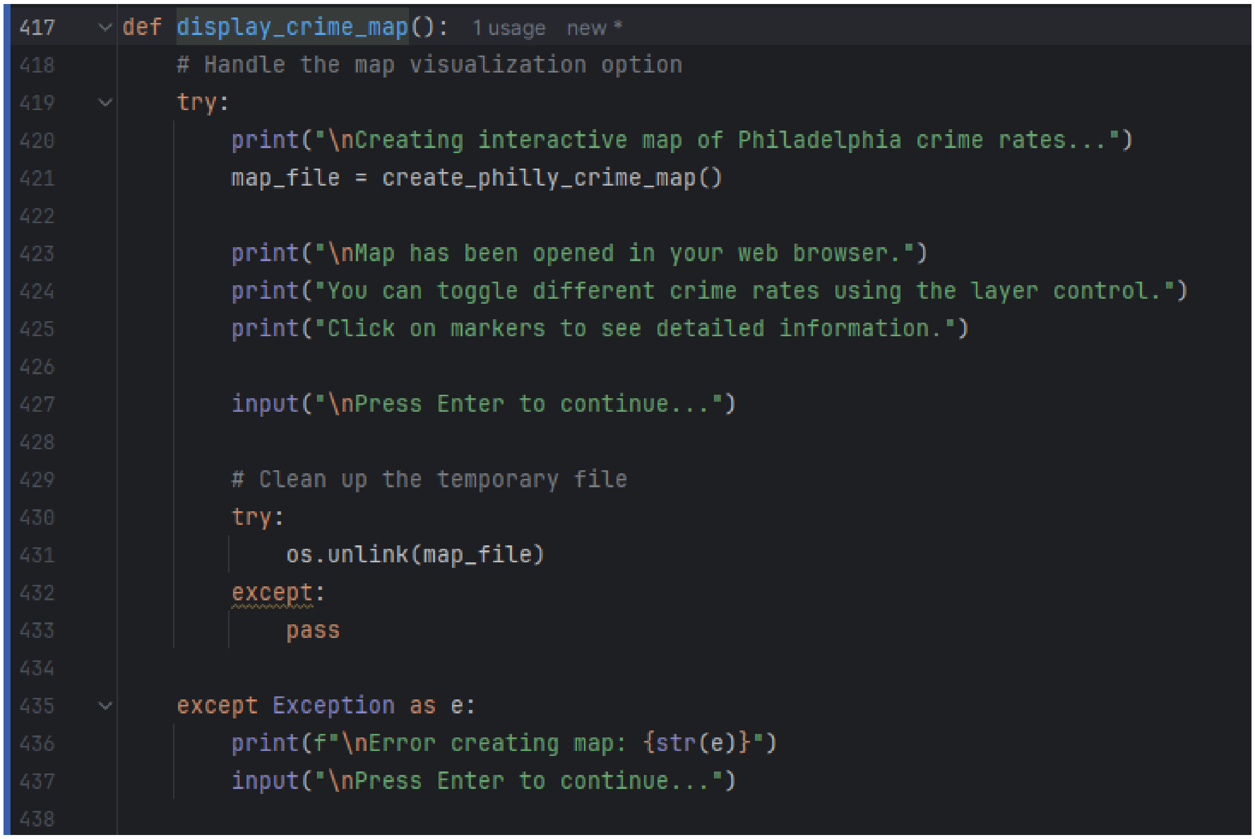


```python
def merge_sort_neighborhoods(criterion):
    print(f"\nSorting neighborhoods by {criterion} using Merge Sort...")
    print("-" * 40)

    def merge(left, right):
        result = []
        left_idx, right_idx = 0, 0

        # Compare elements from both arrays and merge them in sorted order
        while left_idx < len(left) and right_idx < len(right):
            if left[left_idx][0] <= right[right_idx][0]:
                result.append(left[left_idx])
                left_idx += 1
            else:
                result.append(right[right_idx])
                right_idx += 1

        # Add remaining elements
        result.extend(left[left_idx:])
        result.extend(right[right_idx:])
        return result

    def merge_sort(arr):
        # Base case: arrays of length 0 or 1 are already sorted
        if len(arr) <= 1:
```

- Display the Crime Map:

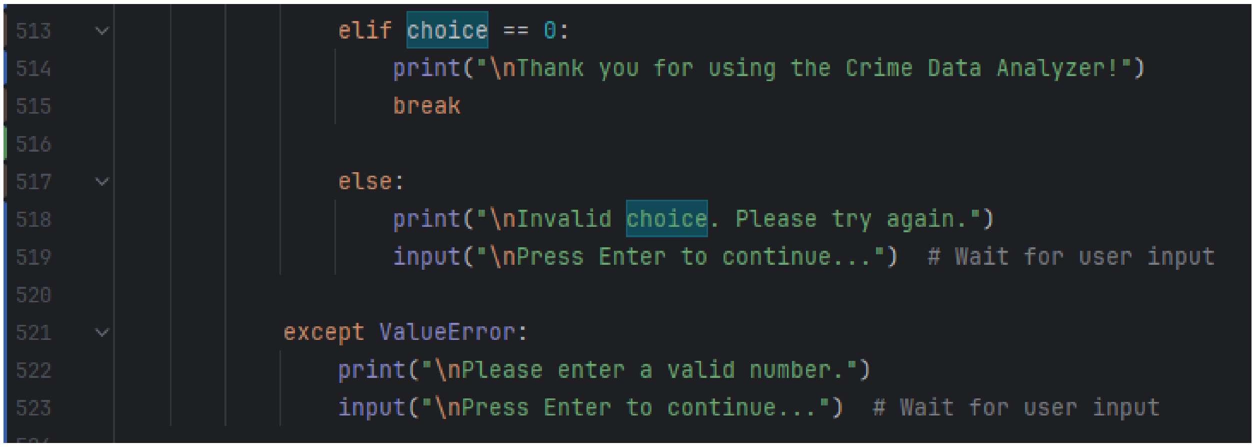


```
                elif choice == 6:
                    display_crime_map()
                    input("\nPress Enter to continue...")  # Wait for user input
```



```
def display_crime_map():  1 usage  new *
    # Handle the map visualization option
    try:
        print("\nCreating interactive map of Philadelphia crime rates...")
        map_file = create_philly_crime_map()

        print("\nMap has been opened in your web browser.")
        print("You can toggle different crime rates using the layer control.")
        print("Click on markers to see detailed information.")

        input("\nPress Enter to continue...")

        # Clean up the temporary file
        try:
            os.unlink(map_file)
        except:
            pass

    except Exception as e:
        print(f"\nError creating map: {str(e)}")
        input("\nPress Enter to continue...")
```

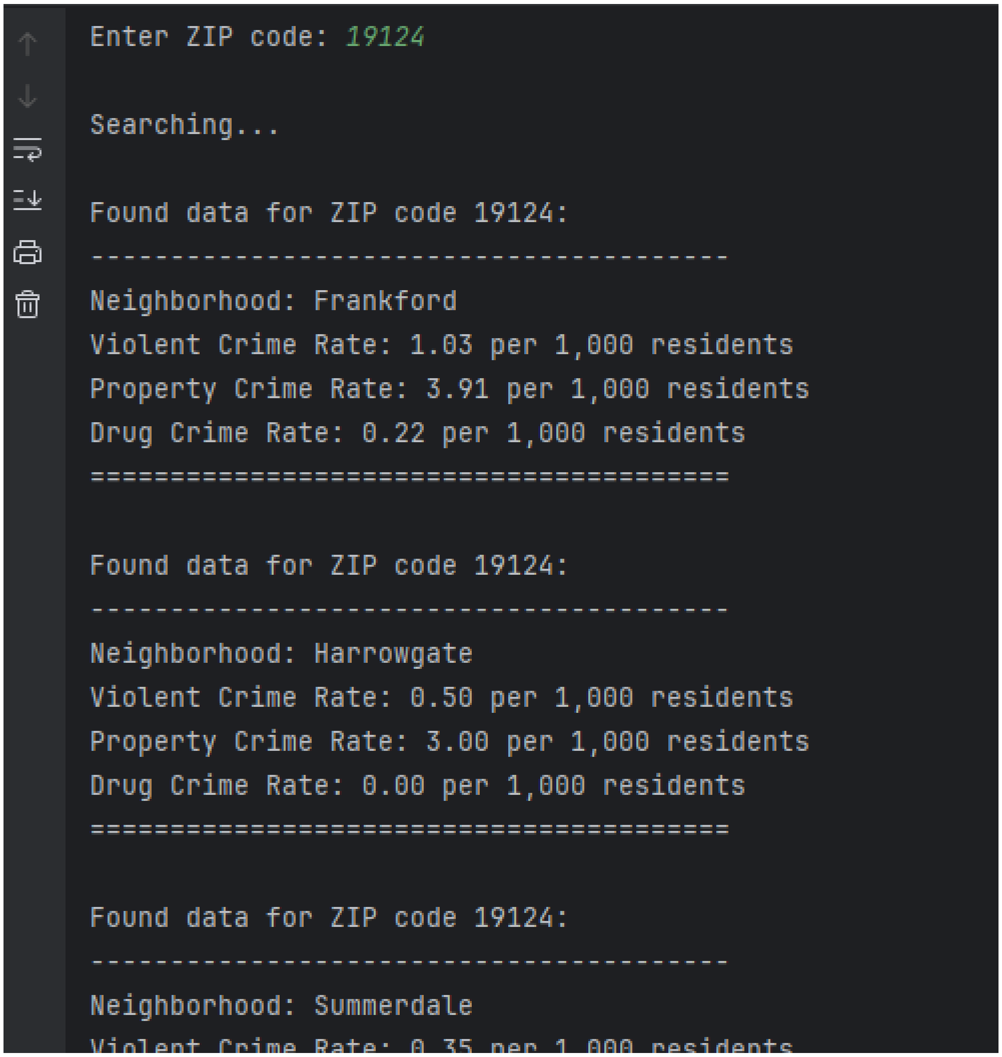
- Exit the application:



```
                elif choice == 0:
                    print("\nThank you for using the Crime Data Analyzer!")
                    break

                else:
                    print("\nInvalid choice. Please try again.")
                    input("\nPress Enter to continue...")  # Wait for user input

            except ValueError:
                print("\nPlease enter a valid number.")
                input("\nPress Enter to continue...")  # Wait for user input
```

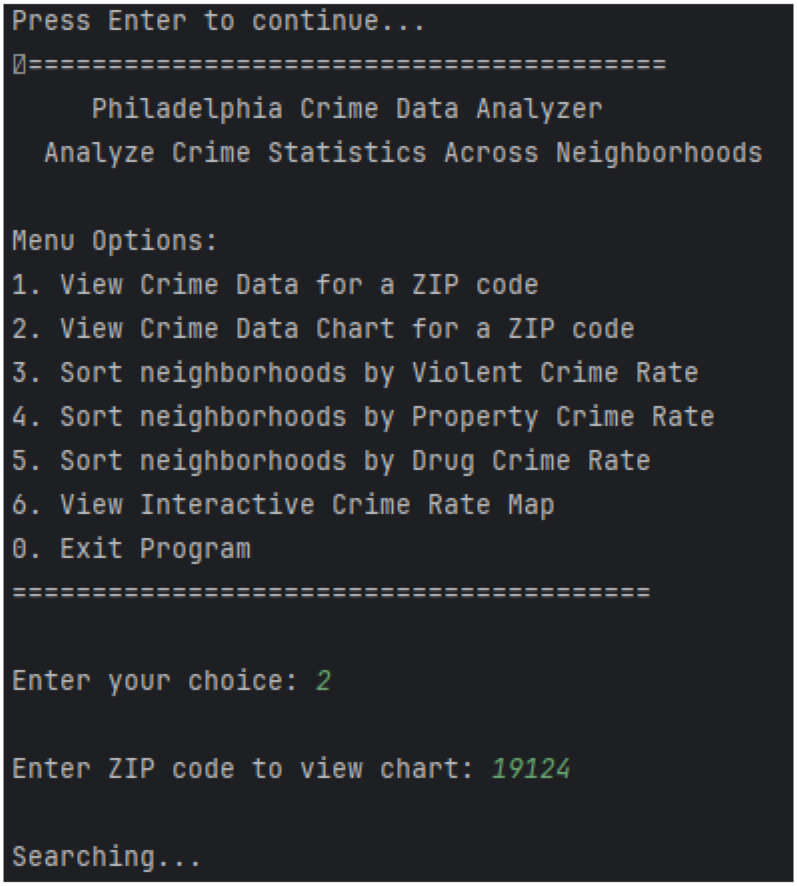
Showcasing the Achievement of Project Goals:

- Print statistics in the console:

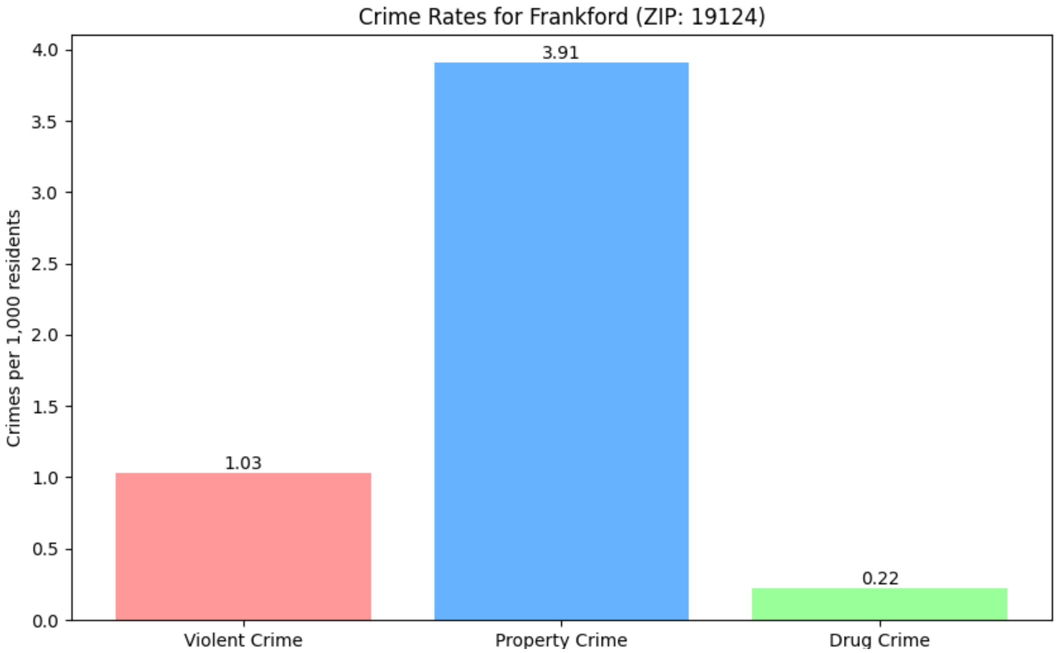


```
Enter ZIP code: 19124

Searching...

Found data for ZIP code 19124:
----------------------------------------
Neighborhood: Frankford
Violent Crime Rate: 1.03 per 1,000 residents
Property Crime Rate: 3.91 per 1,000 residents
Drug Crime Rate: 0.22 per 1,000 residents
========================================

Found data for ZIP code 19124:
----------------------------------------
Neighborhood: Harrowgate
Violent Crime Rate: 0.50 per 1,000 residents
Property Crime Rate: 3.00 per 1,000 residents
Drug Crime Rate: 0.00 per 1,000 residents
========================================

Found data for ZIP code 19124:
----------------------------------------
Neighborhood: Summerdale
Violent Crime Rate: 0.35 per 1,000 residents
```

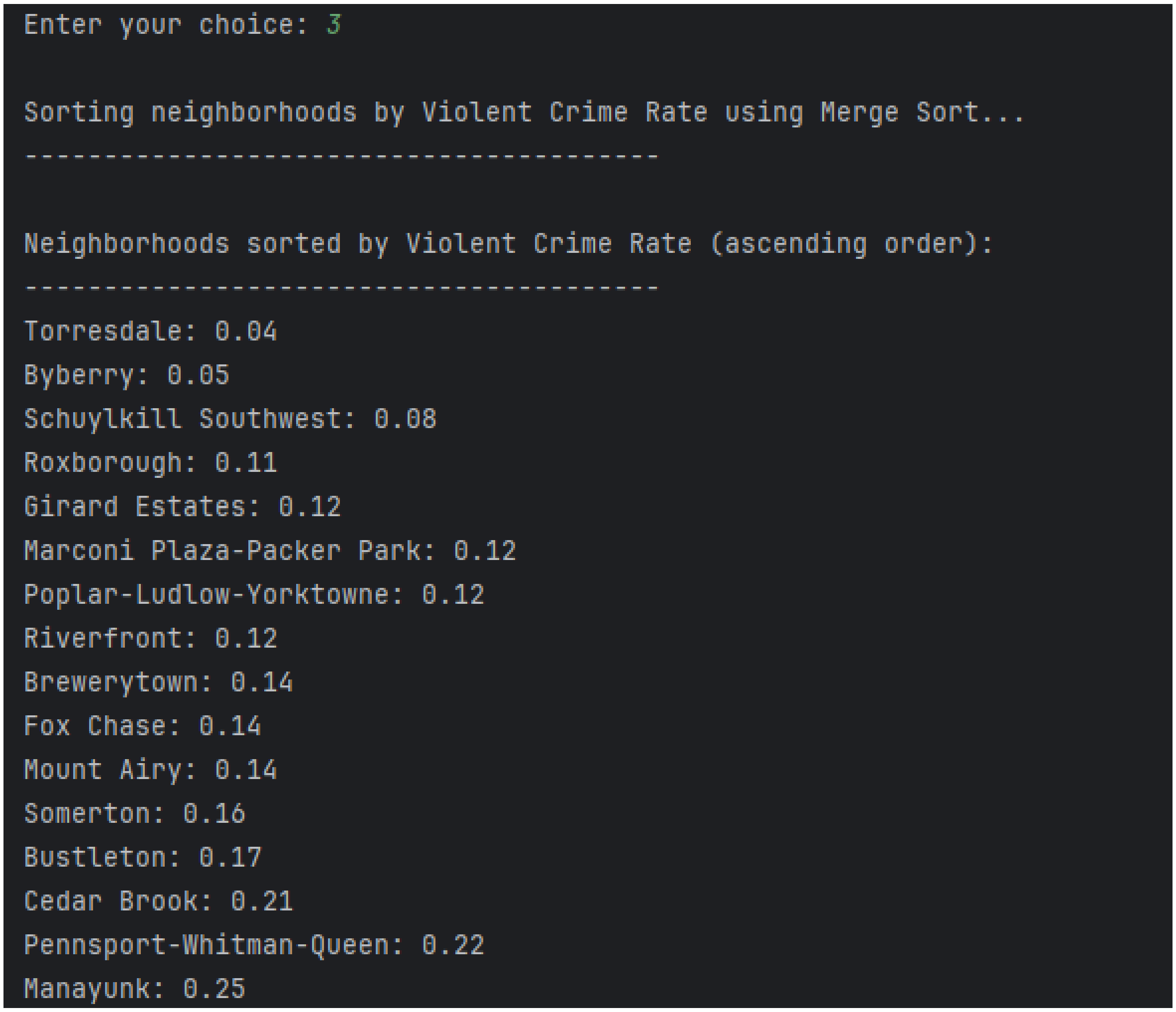
- Display Statistics on the Chart:



```
Press Enter to continue...
▯========================================
     Philadelphia Crime Data Analyzer
  Analyze Crime Statistics Across Neighborhoods

Menu Options:
1. View Crime Data for a ZIP code
2. View Crime Data Chart for a ZIP code
3. Sort neighborhoods by Violent Crime Rate
4. Sort neighborhoods by Property Crime Rate
5. Sort neighborhoods by Drug Crime Rate
6. View Interactive Crime Rate Map
0. Exit Program
========================================

Enter your choice: 2

Enter ZIP code to view chart: 19124

Searching...
```

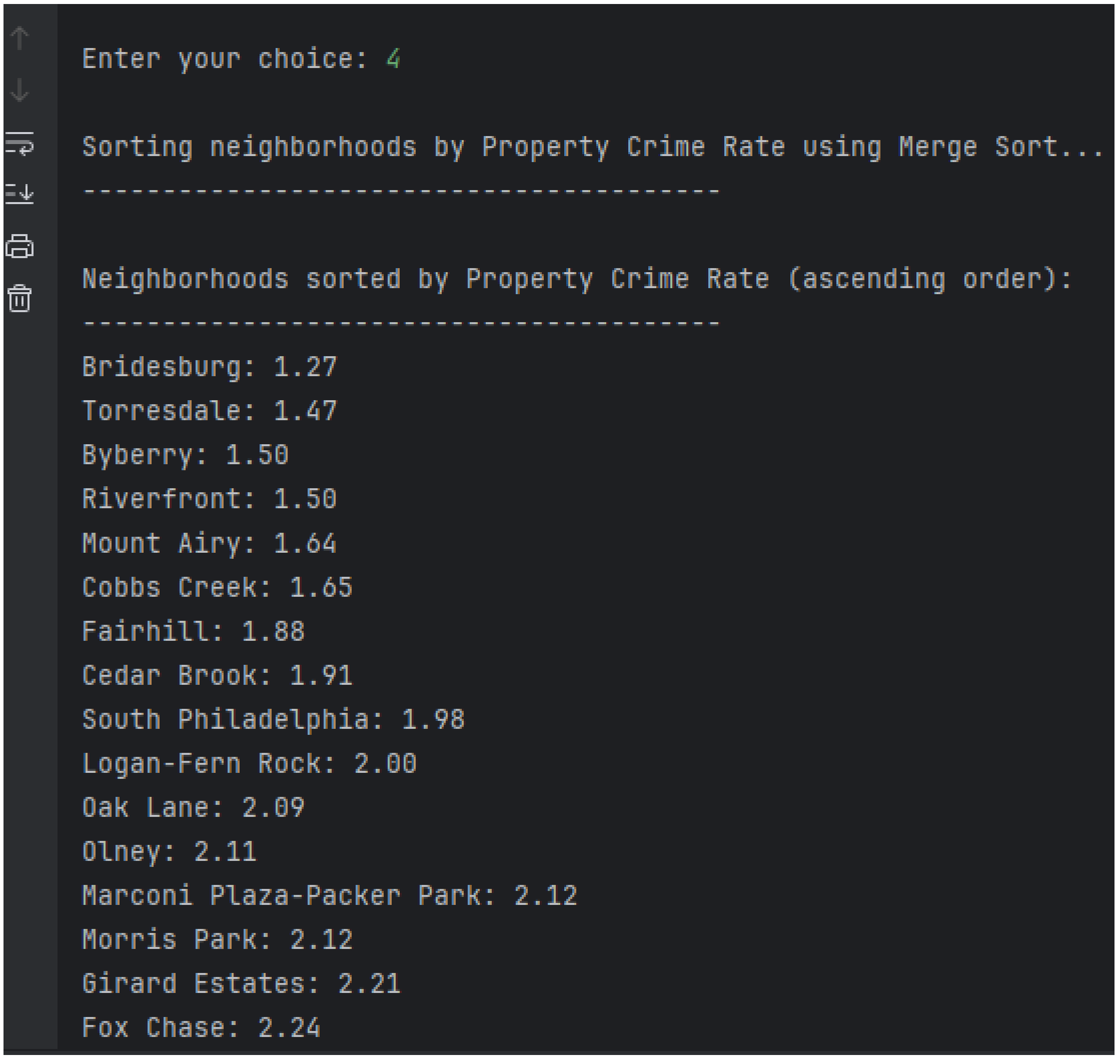


- Sort by Violent Crime:



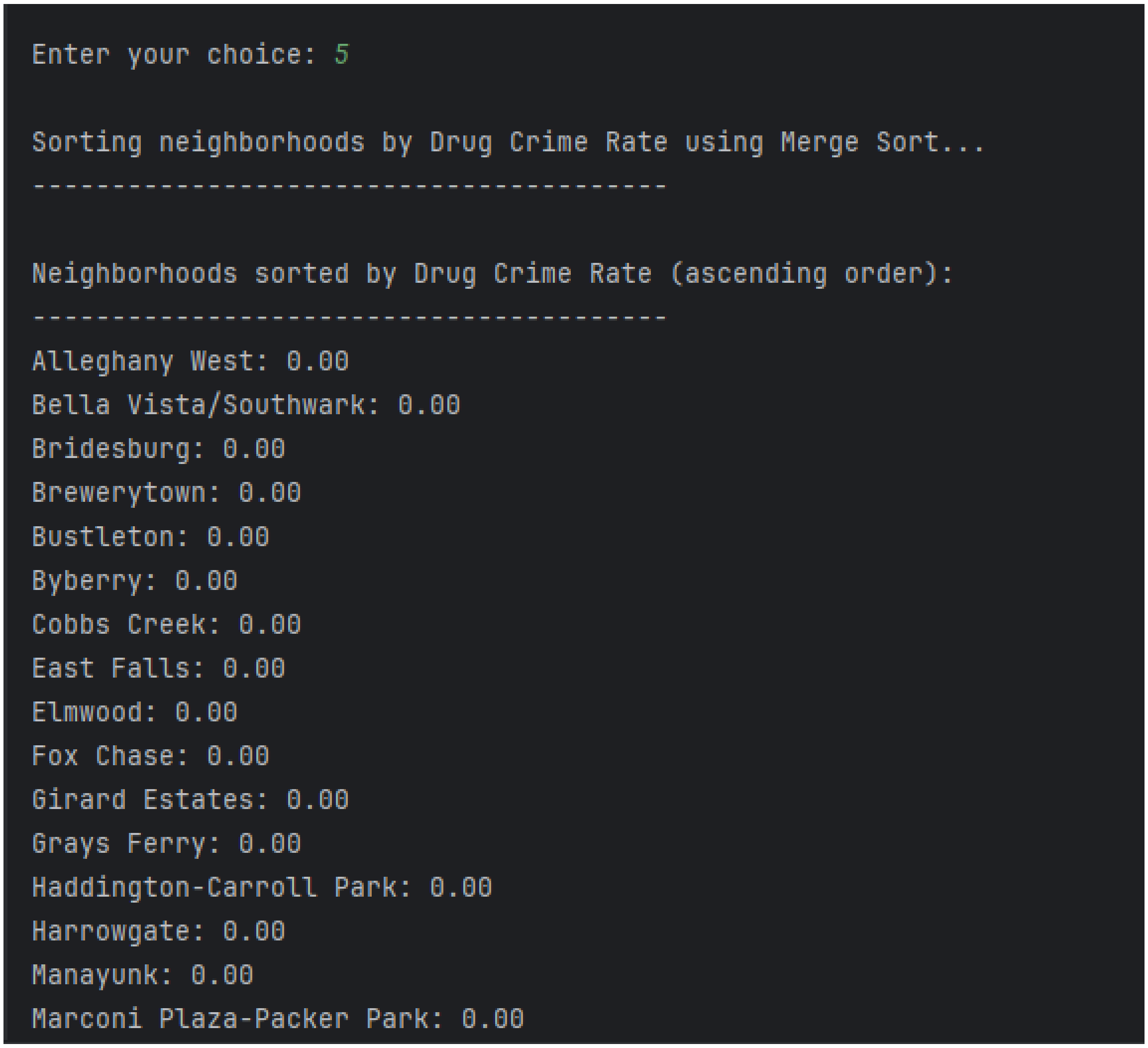
```
Enter your choice: 3

Sorting neighborhoods by Violent Crime Rate using Merge Sort...
----------------------------------------

Neighborhoods sorted by Violent Crime Rate (ascending order):
----------------------------------------
Torresdale: 0.04
Byberry: 0.05
Schuylkill Southwest: 0.08
Roxborough: 0.11
Girard Estates: 0.12
Marconi Plaza-Packer Park: 0.12
Poplar-Ludlow-Yorktowne: 0.12
Riverfront: 0.12
Brewerytown: 0.14
Fox Chase: 0.14
Mount Airy: 0.14
Somerton: 0.16
Bustleton: 0.17
Cedar Brook: 0.21
Pennsport-Whitman-Queen: 0.22
Manayunk: 0.25
```

- Sort by Property Crime:



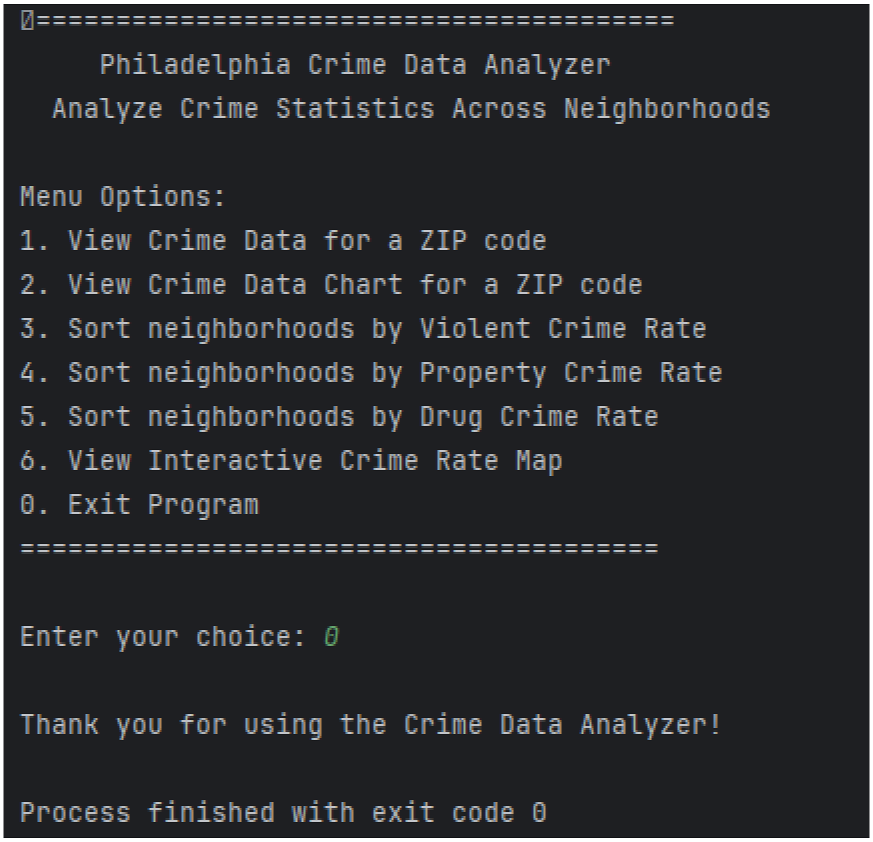
```
Enter your choice: 4

Sorting neighborhoods by Property Crime Rate using Merge Sort...
----------------------------------------

Neighborhoods sorted by Property Crime Rate (ascending order):
----------------------------------------
Bridesburg: 1.27
Torresdale: 1.47
Byberry: 1.50
Riverfront: 1.50
Mount Airy: 1.64
Cobbs Creek: 1.65
Fairhill: 1.88
Cedar Brook: 1.91
South Philadelphia: 1.98
Logan-Fern Rock: 2.00
Oak Lane: 2.09
Olney: 2.11
Marconi Plaza-Packer Park: 2.12
Morris Park: 2.12
Girard Estates: 2.21
Fox Chase: 2.24
```

- Sort by Drug Crime:



```
Enter your choice: 5

Sorting neighborhoods by Drug Crime Rate using Merge Sort...
----------------------------------------

Neighborhoods sorted by Drug Crime Rate (ascending order):
----------------------------------------
Alleghany West: 0.00
Bella Vista/Southwark: 0.00
Bridesburg: 0.00
Brewerytown: 0.00
Bustleton: 0.00
Byberry: 0.00
Cobbs Creek: 0.00
East Falls: 0.00
Elmwood: 0.00
Fox Chase: 0.00
Girard Estates: 0.00
Grays Ferry: 0.00
Haddington-Carroll Park: 0.00
Harrowgate: 0.00
Manayunk: 0.00
Marconi Plaza-Packer Park: 0.00
```

- Exit the Application:



```
▯========================================
     Philadelphia Crime Data Analyzer
  Analyze Crime Statistics Across Neighborhoods

Menu Options:
1. View Crime Data for a ZIP code
2. View Crime Data Chart for a ZIP code
3. Sort neighborhoods by Violent Crime Rate
4. Sort neighborhoods by Property Crime Rate
5. Sort neighborhoods by Drug Crime Rate
6. View Interactive Crime Rate Map
0. Exit Program
========================================

Enter your choice: 0

Thank you for using the Crime Data Analyzer!

Process finished with exit code 0
```

Discussion and Conclusions:

While the application serves its main purpose, by providing users with crime-related data in an easily accessible manner, it is not without its limitations. The application does not have access to a database to update these statistics. Each value is hard coded into the application itself, which would require developers to manually edit these values for them to be updated. Furthermore, The binary search function is only effective if the ZIP codes are sorted. If the data is large and not sorted, the performance could degrade. The *create_sorted_zip_mapping* function creates a new sorted list every time it is called, which could be inefficient. In terms of classroom materials implemented into this project, the application utilizes *matplotlibs*, a library frequently used during homework assignments. When displaying statistics in terms of specific crime rates, the application uses *mergesort* to sort the neighborhoods and *binary search* when searching for a specific zip code. In conclusion, the application serves its primary function of informing residents about the safety of Philadelphia neighborhoods.