SCHOOL OF COMPUTING, ENGINEERING AND DIGITAL TECHNOLOGIES

CIS4044-N-BF1-2021

Software for Digital Innovation

JANUARY 18, 2021

Prepared by

B1326237 Gowthami Nagappan

ABSTRACT

The goal of the project is to design and implement a graphical user interface (GUI) programme that can process and visualise Covid-19 confirmed cases. The source of the covid data is from the website administered by the UK government and the other data 'Stop and Search' is retrieved from publicly available police data on the web portal (https://data.police.uk/docs/method/stops-force/). The application is written in Python and the GUI is created with the Tkinter Python module. The Pandas and Matplotlib modules are used to process and visualize the data about the Total number of cases reported each day over a given date range. The Total cases reported each month over a particular period are identified. Additionally, the areas with the highest number of cases on a particular day and areas with the largest positive change of cases are pulled for the particular period. The data of two or more areas are compared and the daily change is shown as a visual on the graph.

INTRODUCTION

Python has become one of the most popular interpreted scripting languages with applications in data science, machine learning, and general software development. It is a programming language which has a robust data-oriented library ecosystem and capabilities for processing, analysing and visualising data. In this project, I used Python modules to process and analyse data as well as create a user interface. The commonly used libraries are Tkinter, Pandas and Matplotlib.

Pandas is used to read data in csv format and transform it to a legible pandas data frame allowing us to alter the data as required. The library we used to visualise the processed data as bar plots, line plots and pie charts was Matplotlib. Finally, we'll need a user interface to enter parameters and browse between panels for our final plots.

METHODS

The goal is to create a GUI programme that can read and handle a csv file which contains information on the proven Covid-19 cases as well as to collect Stop and Search data from the internet via an API request. We used Tkinter for our user interface and when we run our app, the UI will prompt us to select whether we want to view the graphs in Tkinter frames or not.

Here we have processed and visualized two different data sets. For plotting the covid-19 data, we have a csv file that contains all the data we require. Using pandas, we can easily load the csv from our data directory using the pandas function 'read_csv'. But in the case of Stop and Search by police force, the data is not available as a csv file. We are provided with an API route, from which we can directly fetch the data using a get request.

Integrated Development Environment (IDE)

IDE is tool that connects programmer with the application. We used Anaconda - Spyder IDE which is a freely available open-source environment for the programmers and Anaconda CMD.exe prompts for loading all the necessary python packages and libraries with its dependency.

PYTHON LIBRARIES AND USAGES

Tkinter

The user interface is created with Tkinter. The user interface is made up of a class called 'App' that is inherited from the parent tkinter and acts as our root window as well as a class called 'MainFrame' which acts as a parent frame. It is made up of a list of frames that allows us to go back and forth between the frames we constructed.

Pandas

The information we have is in.csv format. However, we must read and convert it to a pandas data frame in order to process it according to our requirements. We use the pandas function ‘read_csv' to read the csv file, and then append the 'newCasesBySpecimenDate-0 59' and 'newCasesBySpecimenDate-0 60+' columns to make a 'total cases' column. To gather our essential data sets for our final plots, we apply single and repeated queries to this data frame.

Matplotlib

We utilise the matplotlib library to make visualisations out of the data which we analysed using pandas. We used the library to create bar plots, line plots, and pie charts. Using the class 'FigCanvasTkAgg,' we can easily incorporate the plots into our tkinter frame, allowing us to see the plots without having to open a new matplotlib window.

Tkcalendar

We use tkcalendar to input time periods with fewer lines of code instead of using separate entries for month, day and year.

Autocomplete Combobox

The library is used for the convenience of the users. When we use this method, we only need to input the first few letters in the data entry and the rest will appear automatically.

Additional loaded Python libraries

Babel , certify , charset-normalizer , cycler , fonttools , idna , kiwisolver , numpy , packaging , Pillow , Pyparsing , python-dateutil , pytz , requests , six , tk , ttkwidgets , urllib3 are additional libraries saved in text file version to load using pip install -r filename.text command.

SECURITY RISKS OF OPEN-SOURCE LIBRARIES

We're going to use open-source python packages to analyse the data and plot our graphs. As it is an open-source library, it may pose number of security issues. The contributors themselves make vulnerabilities publicly available. This means that if the software contains a vulnerable issue, anyone can exploit it which could have an impact on our project if we utilise that library. These flaws can be exploited by cyber criminals. The second issue is a lack of safety. There are no legal requirements for security in open-source libraries. Furthermore, these softwares are not covered by a warranty and the creators and supporters can discontinue support at any moment without warning.

RESULT

Module 1: Covid-19 and Stop & Search Homepage

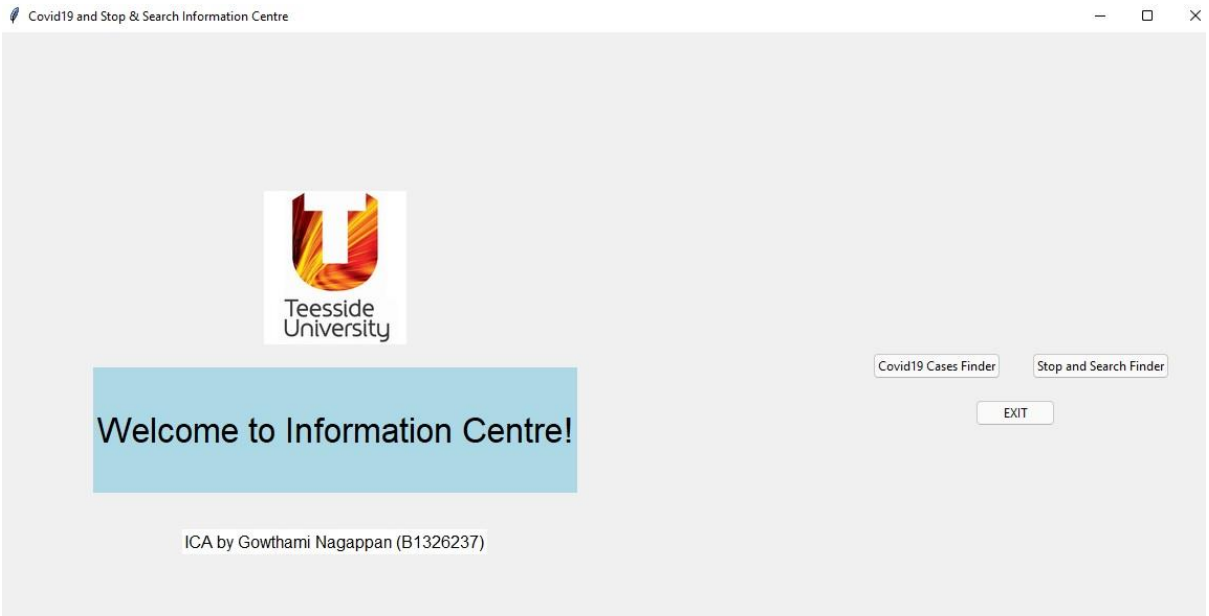

Fig 1: Tkinter home page

Fig 1 shows the home window of the application. Here we can see the left section contains the university logo and the covid-19 and ‘stop and search’ query button. The right section contains the app description and the developer’s name.

Module 2: Covid-19 Finder Query Page

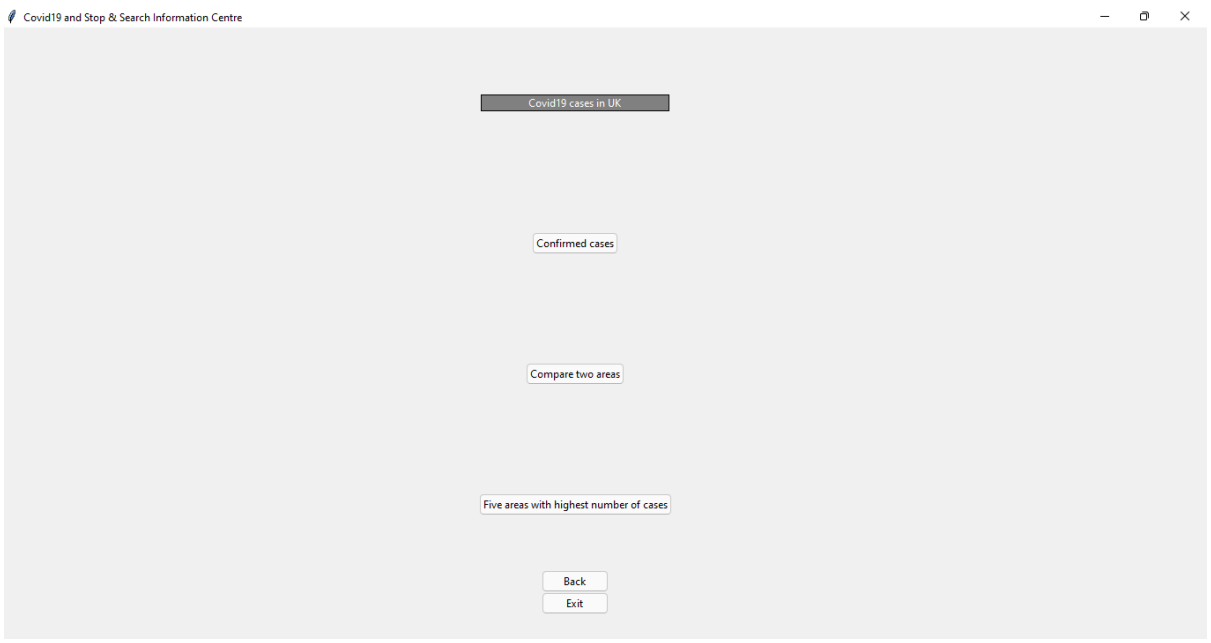

Fig 2: Covid-19 Case Finder Page

Fig 2 shows the query page which consists of three query buttons. They are confirmed cases in a given time period, compare cases confirmed in two areas and 5 areas which has the highest number of confirmed cases

Module 3: Covid-19 Confirmed Case on Daily Basis

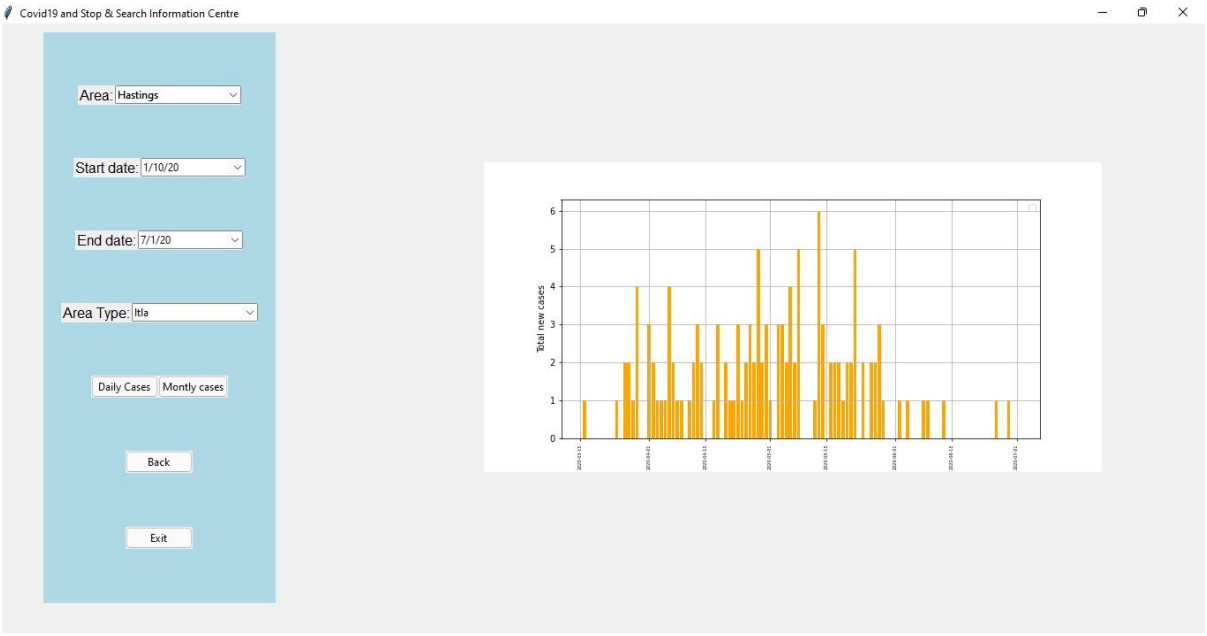

Fig 3: Daily plot for covid-19 cases over a given time in a given area

Module 4: Covid-19 Confirmed Case on Monthly Basis

From Fig 2, we can see that the confirmed cases are at peak during the month of April to May in Hastings. In the plot we have selected the time period from 10th of January to 1st of July.

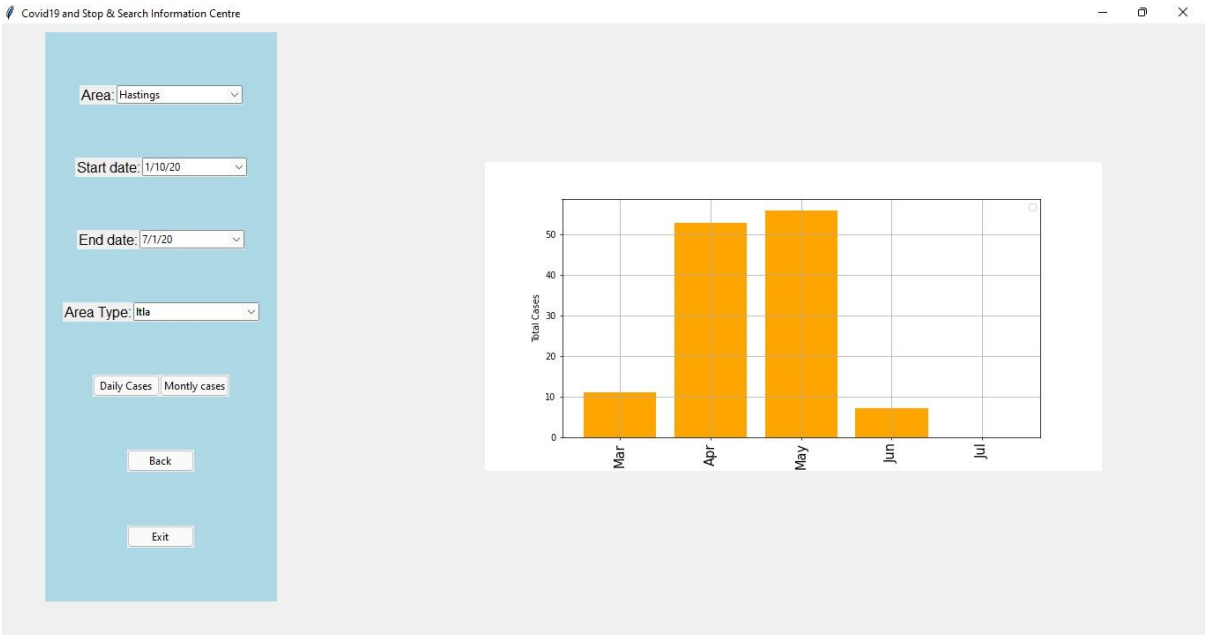

Fig 4: Monthly plot for covid-19 cases over a given time in a given area

Fig 4 gives us a clear understanding about the number of covid cases in each month during a time period, we can see that the confirmed cases are at a peak in May in the area Hastings

Module 5: Compare Two areas Covid cases on Monthly Basis

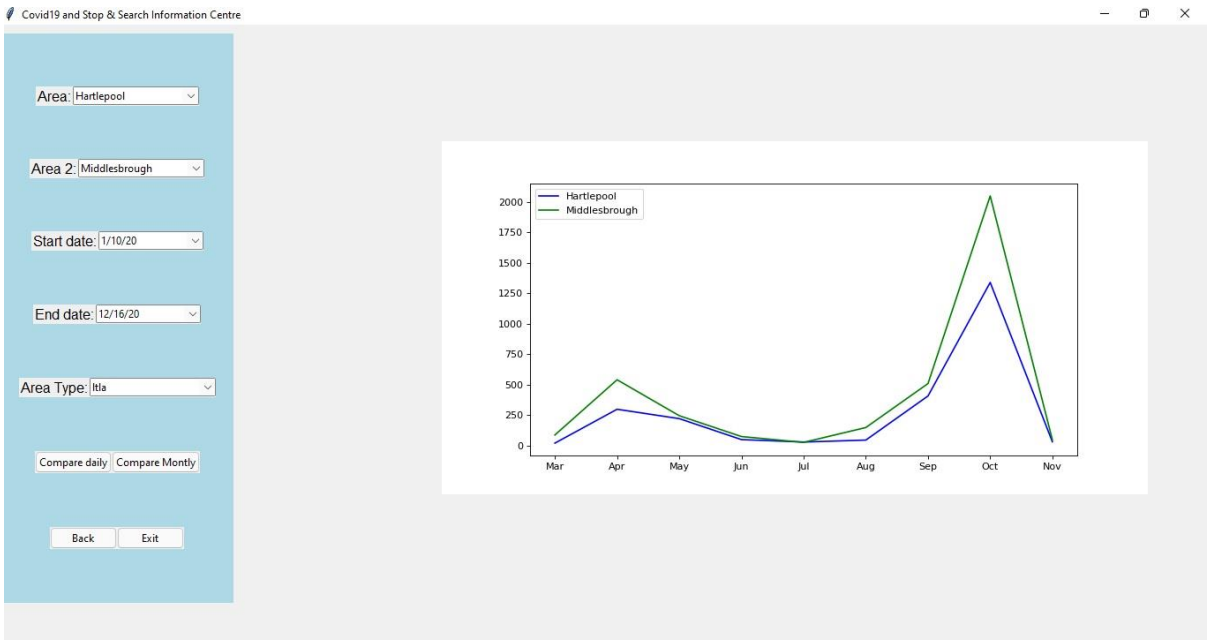

Fig 5: Comparison of monthly cases in Hastings and Middlesbrough

Module 6: Compare Two areas Covid cases on Daily Basis

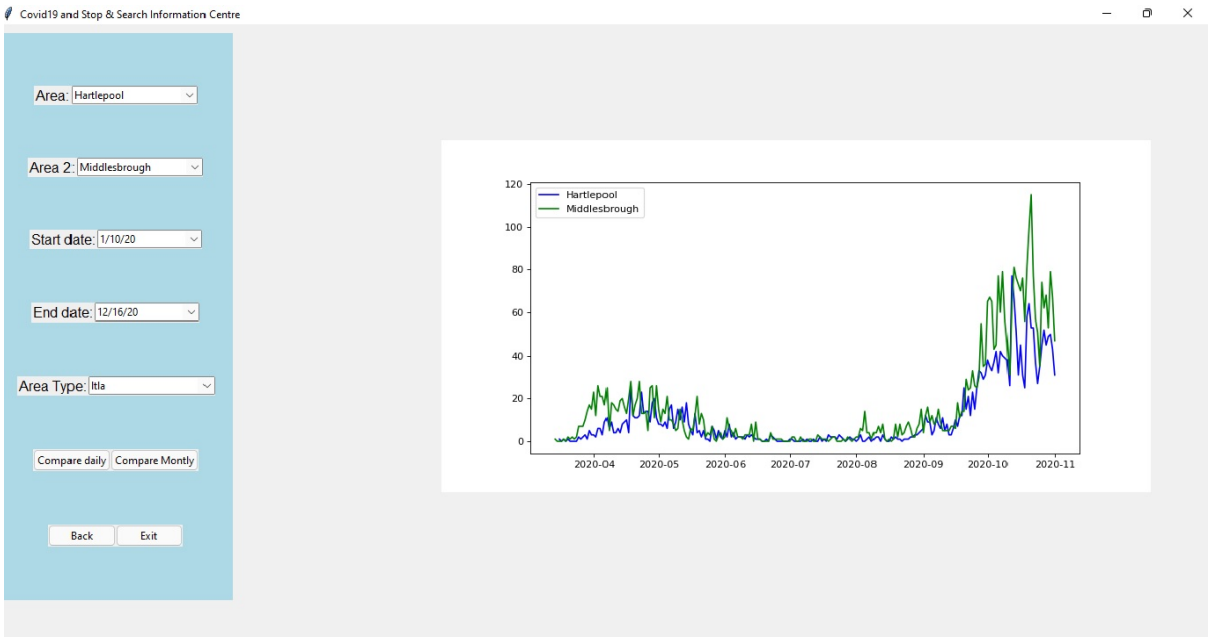

Fig 6: Comparison of daily cases in areas Hastings and Middlesbrough

Fig 5 and Fig 6 gives us an understanding about the covid-19 cases in the areas, Middlesbrough and Hastings. From both graphs, it is clear that in Middlesbrough the confirmed covid-19 cases are higher when compared to Hastings from January 2020 to December 2020

Module 7: Compare Five Areas Covid cases on Particular Day

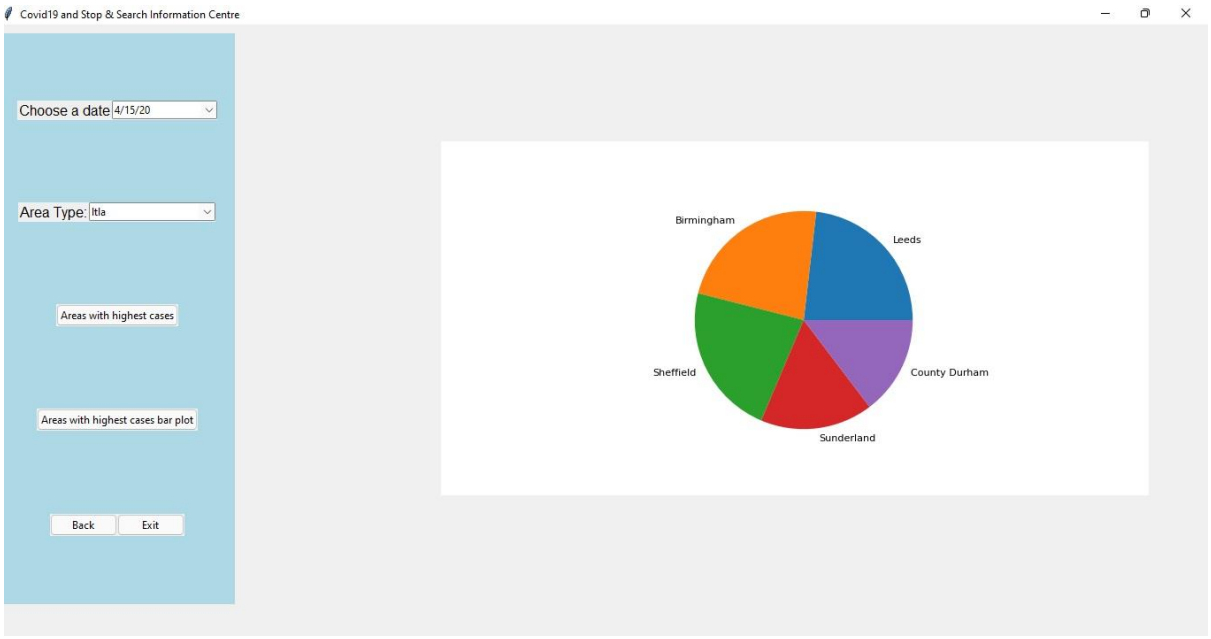

Fig 7: 5 areas with highest confirmed cases in a given day

Fig 7 shows a pie plot which consist of 5 areas that comes in the top 5 highest confirmed covid-19 cases in the UK. Here we can see that the chosen date is 15th of April 2020 and area type is LTLA. We can inference that Birmingham, Sheffield and Leeds has the highest number of confirmed cases.

STOP AND SEARCH

We use the data from the internet provided by the police to analyse and visualise the Stop and Search data. 'https://data.police.uk/api/stops-force?force=force>&date=year-month>' was the API route which was used to get the data. We can select a force name and date from the retrieved data to obtain the total number of teens stopped and searched by the force on a given day.

Module 8: Stop and Search Force on Particular Month

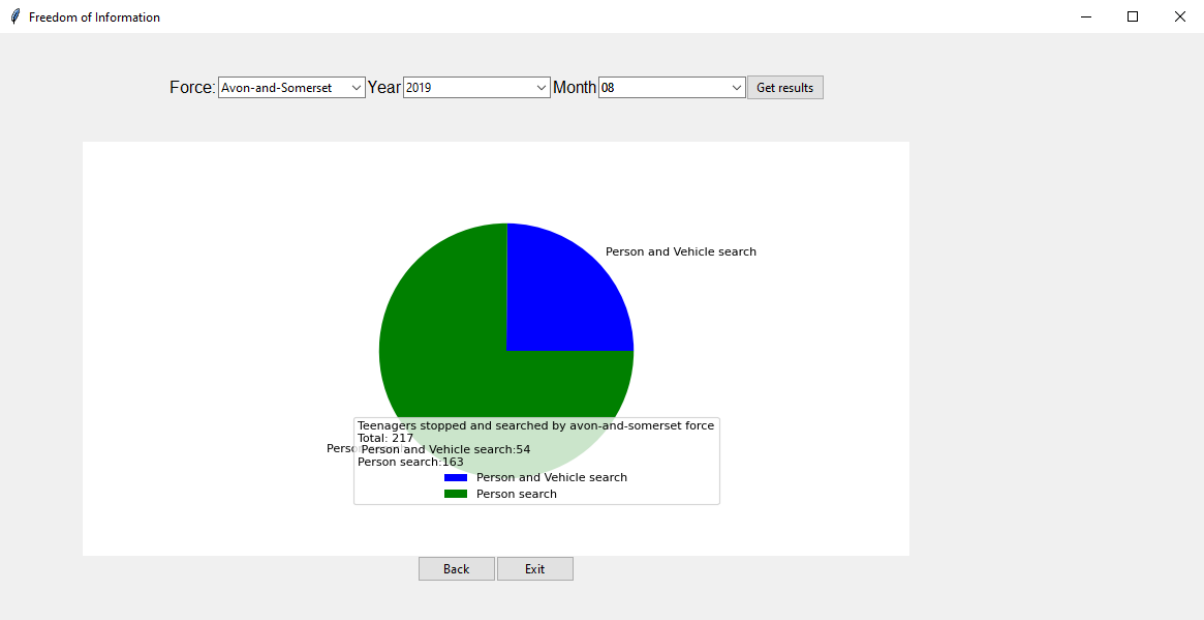

Fig 8: Pie chart for the total number of teenagers who have been stopped and searched

From Fig 8, it was evident that Cleveland as the police force. So in the year August 2019, 217 teenagers were stopped and searched by Cleveland police in which 163 of them were stopped for person search and 54 of them was stopped for both person and vehicle search.

Scenario ID	Test case ID	Test case description	Test step	Test description	Expected result	Actual result	Status	comment
TS01	TC01	Launch the freedom of information application	Pre-condition	1. Activate virtual environment and install required libraries	Virtual environment activation and libraries installation success	Virtual environment activation and libraries installation success	Pass	
			Step 1	Launch the application	The freedom of information application launched succesfully	The app launched succesfully	Pass	
TS02	TC02	Plot daily and montly covid cases	Pre-condition	1. Read csv file 2. Add every cases for different age groups and create a column named 'total_cases'	Csv file read succesfully and created a new column	Csv file read succesfully and created a new column named 'total_cases'	Pass	
			Step 1	Choose a time period and an area to track daily cases in.	Plot a bar graph displaying the changes in confirmed cases throughout the time period you've chosen.	Bar plot launced succesfully	Pass	
			Step 2	Choose a time period and an area to track monthly cases	Plot a bar graph showing the variations in confirmed cases over the chosen time period	Bar plot launced succesfully	Pass	
TS03	TC03	compare total cases between two areas on daily and monthly basis	Step 1	Choose a time period and two areas to compare daily cases	Plot line graphs for two indivudual areas showing the variations	Line plot launched succesfully	Pass	
			Step 2	Choose a time period and two areas to compare monthly cases	Plot line graphs for two indivudual areas showing the variations on a monthly basis	Line plot launched succesfully	Pass	
TS04	TC04	Evaluate the five areas which have the highest number of covid-19 cases	Step 1	Choose a date and plot a pie chart showing the ten areas	Plot pie chart showing the 5 areas with highest number of cases	Pie chart plotted succesfully	Pass	
			Step 2	Choose a date and plot a bar plot showing the ten areas	Plot a bar graph chart showing the 10 areas with highest number of cases	Bar plot launched successfuly	Pass	
TS05	TC05	Examine the police Stop and Search data from internet and visualize the results	Pre-condition	1. Fetch data from the website using an api route 2. Process the data	Succesfully fetch data and process for visualizaton	Data fetched succesfully and processed	Pass	
			Step 1	Choose a force name, year and month and plot a chart	Plot a pie chart showing the person search and vehicle search	Pie chart launched succesfully	Pass	

Fig 9. Test case

CONCLUSION

Any firm or business system cannot function without data. In order to uncover the information relevant for decision making it is necessary to gather, process and analyse data flow in a fast and accurate manner. Python is becoming increasingly used in scientific computing. It comes with several data-oriented libraries that can help to speed up and simplify data processing. Pandas and Matplotlib are two of the most widely used data processing and visualisation frameworks. We can read a large amount of data and process it quickly and easily using these libraries. Furthermore when it comes to the user side, the Tkinter library's application is very handy. The user may quickly navigate through the windows and provide the information needed for the final representation.

REFERENCE

Ceder, N.R. 2017, The quick Python book, Third edn, Manning, Greenwich

Oliphant, T.E. 2007, "Python for Scientific Computing", Computing in science & engineering, vol. 9, no. 3, pp. 10-20.

Mckinney, W. 2017, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, O'Reilly Media, Incorporated, Sebastopol.

Idris, I. 2014, Python data analysis: learn how to apply powerful data analysis techniques with popular open source Python modules, 1st edn, PACKT Publishing.

Yim, A., Chung, C. & Yu, A. 2018, Matplotlib for Python Developers: Effective Techniques for Data Visualization with Python, 2nd Edition, Packt Publishing, Limited, Birmingham.

Tosi, S. 2009, Matplotlib for Python developers: build remarkable publication quality plots the easy way, 1st edn, PACKT Publishing, Olton.