The Orchid School, Pune

A Project Report
On

COVID-19 Vaccination Progress Management System

For
AISSCE 2022 Examination
[As a part of the Informatics Practices Course (065)]
TERM-2

SUBMITTED BY:

Abhilasha Saxena
Aparna Iyer
Arya Ganu

Under the Guidance of:
Mrs. Jayashri Samudra

CERTIFICATE

This is to certify that the Project entitled 'COVID-19 VACCINATION PROGRESS MANAGEMENT SYSTEM' is a bonafide work done by **Aparna Iyer** of class XII, Session 2021-22 in partial fulfilment of CBSE's Examination 2022 for the subject Informatics Practices (065) and has been carried out under my guidance.

This report or a similar report on the topic has not been submitted for any other examination and does not form a part of any other course undergone by the candidate.

…………………………	**…………………………**
Signature of Student	**Signature of Teacher/Guide**
Name: Aparna Iyer	**Name: Mrs. Jayashri Samudra**
…………………………	**…………………………**
Signature of Unit Head	**Signature of Principal**
Name: Ms. Ashwini Shah	**Name: Ms. Sangeeta Kapoor**

…………………………..

Signature of External Examiner

Place: The Orchid School, Pune

Date: ……..………..

ACKNOWLEDGEMENT

I undertook this project work, as the part of my std XII Informatics Practices(065) course. I tried to apply my best of knowledge and experience, gained during the study and class work experience. However, developing a software system is generally a quite complex and time-consuming process. It requires a systematic study, insight vision and professional approach during the design and development.

Moreover, the developer always feels the need, the help and good wishes of the people near you, who have considerable experience and ideas. I would like to extend my sincere thanks and gratitude to my teacher Ms. Jayashri Samudra for guiding us throughout the life cycle of Software Development**.**

I would like to take the opportunity to extend my sincere thanks and gratitude to my Unit Head Ms. Ashwini Shah for giving valuable time and moral support to develop this software.

I am very much thankful to our Principal Ms. Sangeeta Kapoor for being a source of inspiration.

Class XII A
Aparna Iyer

C O N T E N T S

1. Introduction

1.1. Purpose of the System

This project aims to provide a system to visualise and analyse world vaccination data, in order to understand the trends in the rate of vaccine-administration, the relative progress of vaccination in various countries, the percentage of population vaccinated, once or twice, as well as the interdependence of these.

The project furthermore provides the analyst or expert a user interactive experience by giving the opportunity to update date-wise world vaccination records with live data entered. It creates dynamic visualizations of live world vaccination data. This helps the analyst, researcher or user to gain a visual understanding of world-wide vaccination status, in various forms. It generates predictions based on existent data, thus aiding experts and analysts in making decisions and drawing meaningful conclusions.

1.2. Advantages of the System

The **COVID-19 Vaccination Management System** integrates the analysis and visualisation of current world vaccination data, with the records entered by the user, in order to maintain a live database and generate graphs and predictions in real-time.

The advantages of the system are as follows:

1. It betters the understanding of the trends in the rate of vaccine-administration, the relative progress of vaccination in various countries; the people vaccinated, once or twice, the start-date of the vaccination programme across countries, as well as the interdependence of these, through effective data visualisation and analysis.
2. It provides an interface for the user or analyst to enter data relevant to their own vaccination status and integrates the user’s records into the main database to maintain a real-time database.
3. It can make predictions about the vaccination timeline based on the current and average rates and percentage of vaccination.
4. It provides qualitative insights specific to the countries selected which can help experts to decide the future steps of vaccination.
5. It generates a visual understanding of vaccination progress across the world.

1.3. Software and Datasets

Front-end:

Front-End is the development environment where we write the program code to develop the interface so that the user can communicate with the system. I have used Python to develop the project. The programs were developed and tested using Spyder Integrated Development Environment.

Back-End:

Back-End refers to the database where data is stored. I have used CSV Files as the back-end to store the vaccination records dataset.

Both the softwares used for developing project work are specified by CBSE and freely available as they are open source software.

Dataset:

The project uses an open-source COVID-19 vaccination progress dataset from Kaggle, 'COVID World Vaccination Progress' (https://www.kaggle.com/gpreda/covid-world-vaccination-progress), which contains country-wise information on the absolute number of total vaccinations, the total number of people vaccinated at least once, the total number of people fully vaccinated, daily vaccinations, the total number of people fully vaccinated per hundred citizens, the total vaccinations per hundred citizens and the vaccines used in the country, by date. It also contains the country vaccinations grouped by manufacturer.

Reason for selection of the dataset:

I chose this dataset because it reflects the progress of COVID-19 vaccination across the world, which is a highly important insight for citizens, researchers and medical professionals to have against the scenario of the pandemic. It is useful in making predictions about the percentage completion of vaccination by a certain date, as well as in understanding which vaccine is predominantly being administered and the relative standing of countries in terms of the number of people being vaccinated once or twice.

1.4. Analytics done by the System

This project analyses world-wide vaccination data, generating the following visualisations:

1. Comparison between vaccination progress in 6 countries selected by the user- once-vaccinated and fully-vaccinated, through a double bar graph.
2. Trend of daily vaccination in India from the start till date, through a line chart.
3. Comparison of the use of popular vaccine brands across countries by means of a pie chart.

The project also generates a predictive timeline of percentage completion of vaccination based on the current percentage and rate of vaccination.

2. Objective and Scope of the Project

The objective of the software project is to develop a Vaccination Progress Management System (VPMS) that will allow users, analysts and researchers to update and analyse world-wide vaccination records. This project is aimed at enhancing the understanding of the vaccination progress in the selected countries; generating predictions to help official bodies to devise informed strategies in the vaccination programme, as well as helping common citizens understand the situation of vaccination better.

2.1. Functionality of the System

The COVID-19 VPMS is designed to provide the following functionality:

a. To provide a user-friendly, centralized, dynamic and integrated environment for visualisation and analysis of live world-wide vaccination data.
b. The proposed system should maintain the existing records from the imported dataset and update the user-entered data into the existing fields.
c. To maintain data integrity between the existing dataset and the data provided by the user.
d. To ensure the dynamism of the analysis by providing the user with the facility of customising the graphs with their choice of countries and topic of visualisation.

In its current scope, the software enables the user to retrieve and update the information from a centralised dataset stored in a CSV File.

2.2. Limitations of the System

1. While the user can append new records to the dataset, no changes can be made to the existing records in the dataset, that is, the user cannot modify the original data in the dataset.
2. The records entered by the user to export into the CSV database cannot be deleted or modified.

3. Problem Definition and Analysis

A well-defined problem is the focal point of a project. This critical step helps ensure and mobilize the right resources. Writing a well-defined problem statement is the first step towards a successful project because a problem well-defined is a problem half solved.

The hardest part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirement. Defining and applying good, complete requirements are hard to work, and success in this endeavour has eluded many of us. Yet, we continue to make progress.

Without defining a problem, developers do not know what to build, customers do not know what to expect, and there is no way to validate that the built system satisfies the requirement. Problem definition describes the ‘what’ of a system, not the ‘how’. The quality of a software product is only as good as the process that creates it.

Problem Definition and Analysis is the activity that encompasses learning about the problem to be solved, understanding the needs of the users. It includes all activities related to the following:

1. Identification and Documentation of user’s needs.
2. Creation of a document that describes the external behaviour and the association constraints that will satisfy those needs.
3. Analysis and Validation of the requirements documents to ensure consistency, completeness, and feasibility
4. Evolution of needs.

Problem Statement for the System:

Vaccination administration being recorded on a daily basis across the world needs to be visualised graphically so as to provide the user with a comprehensive understanding of vaccination progress. This information is absent and the user finds it hard to draw inferences from the raw data. In addition, a user with reliable vaccination records should be able to add them to the existing vaccination data, so as to create an expanding dataset and dynamic, live visualisations.

An analyst, researcher or planner should be able to gain a visual understanding of vaccination status in the desired countries and obtain predictions based on existent data. This would be useful to experts and analysts in making decisions, drawing conclusions and planning vaccinations.

The proposed system is expected to do the following:

1. To provide a user-friendly, integrated and centralized environment for storage and updating of date-wise vaccination records against country.
2. The proposed system should maintain all the records and transactions and should generate the required graphs, reports and information when required.
3. To provide an efficient and secure information storage, flow and retrieval system, ensuring the integrity and validity of records.
4. To provide a graphical and user-friendly interface to interact with a centralized database based on client-server architecture.
5. To generate customisable visualisations as well as meaningful insights and predictions.

4. System Design & Development

4.1. Database Design

An important aspect of system design is the design of data storage structure. To begin with, a logical model of data structure is developed first.

The VPMS Program uses a Master Data File named **country_vaccinations.csv''** which is downloaded from https://www.kaggle.com/gpreda/covid-world-vaccination-progress. This is in the.csv format.

Context

Data is collected daily from the 'Our World in Data' GitHub repository for covid-19, merged and uploaded. Country level vaccination data is gathered and assembled in one single file. Then, this data file is merged with locations data file to include vaccination sources information.

Content

The data (country vaccinations) contains the following information:

- Country- this is the country for which the vaccination information is provided;
- Country ISO Code - ISO code for the country;
- Date - date for the data entry; for some of the dates we have only the daily vaccinations, for others, only the (cumulative) total;
- Total number of vaccinations - this is the absolute number of total immunizations in the country;
- Total number of people vaccinated - a person, depending on the immunization scheme, will receive one or more (typically 2) vaccines; at a certain moment, the number of vaccination might be larger than the number of people;
- Total number of people fully vaccinated - this is the number of people that received the entire set of immunization according to the immunization scheme (typically 2); at a certain moment in time, there might be a certain number of people that received one vaccine and another number (smaller) of people that received all vaccines in the scheme;
- Daily vaccinations (raw) - for a certain data entry, the number of vaccination for that date/country;
- Daily vaccinations - for a certain data entry, the number of vaccination for that date/country;

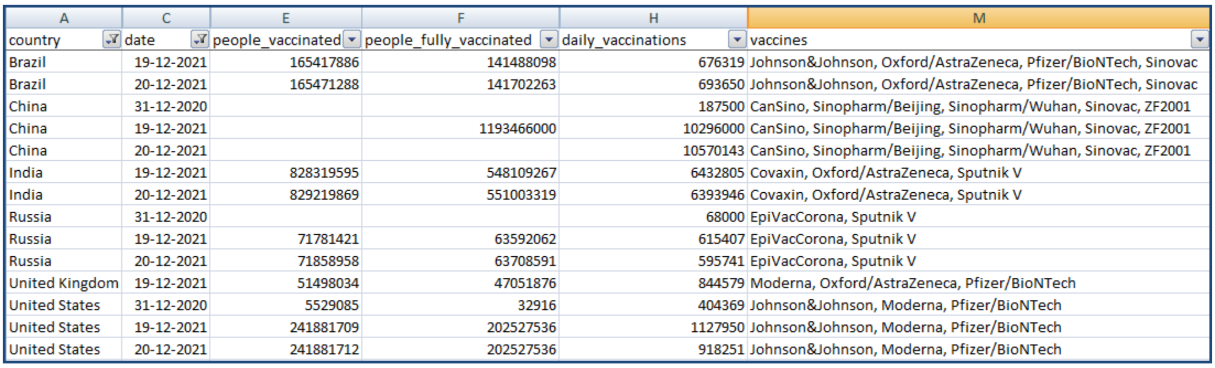
- Total vaccinations per hundred - ratio (in percent) between vaccination number and total population up to the date in the country;
- Total number of people vaccinated per hundred - ratio (in percent) between population immunized and total population up to the date in the country;
- Total number of people fully vaccinated per hundred - ratio (in percent) between population fully immunized and total population up to the date in the country;
- Number of vaccinations per day - number of daily vaccination for that day and country;
- Daily vaccinations per million - ratio (in ppm) between vaccination number and total population for the current date in the country;
- Vaccines used in the country - total number of vaccines used in the country (up to date);
- Source name - source of the information (national authority, international organization, local organization etc.);
- Source website - website of the source of information;

This data file is used as a base to extract multiple 'Datasets' which can then be separately used for Analysis and Display to the User of the VPMS System.

4.2 Datasets Design:

The Database of the VPMS Contains 4 Datasets or Data Tables. The tables are designed to provide the information required by the Administrator and User of the VPMS .The tables and their structure are given below.

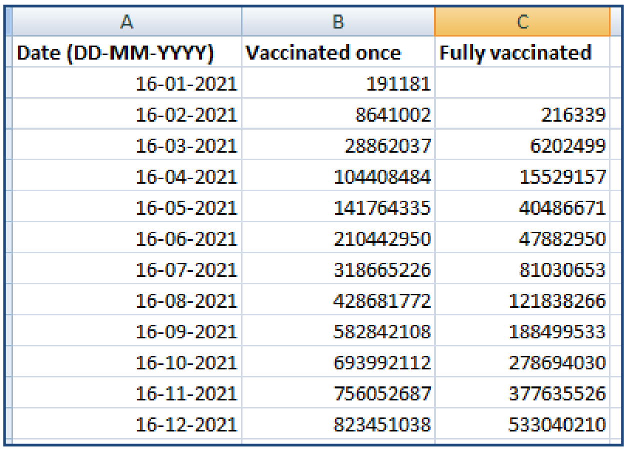
A. Complete Data Set : **'country_vaccinations.csv''**(Described in Section above)



A	C	E	F	H	M
country	date	people_vaccinated	people_fully_vaccinated	daily_vaccinations	vaccines
Brazil	19-12-2021	165417886	141488098	676319	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac
Brazil	20-12-2021	165471288	141702263	693650	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac
China	31-12-2020			187500	CanSino, Sinopharm/Beijing, Sinopharm/Wuhan, Sinovac, ZF2001
China	19-12-2021		1193466000	10296000	CanSino, Sinopharm/Beijing, Sinopharm/Wuhan, Sinovac, ZF2001
China	20-12-2021			10570143	CanSino, Sinopharm/Beijing, Sinopharm/Wuhan, Sinovac, ZF2001
India	19-12-2021	828319595	548109267	6432805	Covaxin, Oxford/AstraZeneca, Sputnik V
India	20-12-2021	829219869	551003319	6393946	Covaxin, Oxford/AstraZeneca, Sputnik V
Russia	31-12-2020			68000	EpiVacCorona, Sputnik V
Russia	19-12-2021	71781421	63592062	615407	EpiVacCorona, Sputnik V
Russia	20-12-2021	71858958	63708591	595741	EpiVacCorona, Sputnik V
United Kingdom	19-12-2021	51498034	47051876	844579	Moderna, Oxford/AstraZeneca, Pfizer/BioNTech
United States	31-12-2020	5529085	32916	404369	Johnson&Johnson, Moderna, Pfizer/BioNTech
United States	19-12-2021	241881709	202527536	1127950	Johnson&Johnson, Moderna, Pfizer/BioNTech
United States	20-12-2021	241881712	202527536	918251	Johnson&Johnson, Moderna, Pfizer/BioNTech

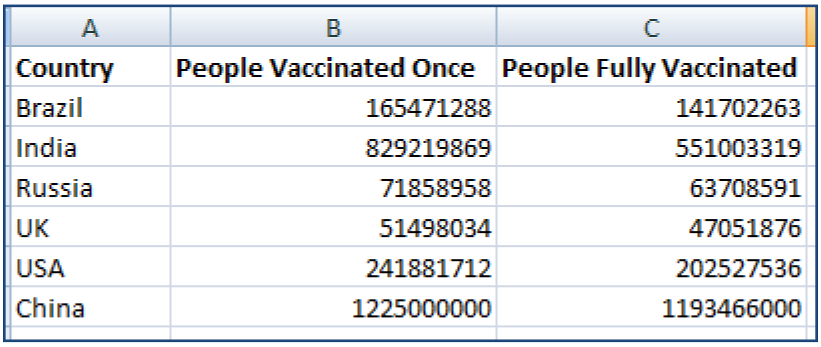
Note: This is a representative image as the file contains over 60,000 records of data from open sources. Columns relevant to the project have been displayed.

B. Data Set 1: Monthly Vaccinations in India ('Monthly vaccinations.csv')



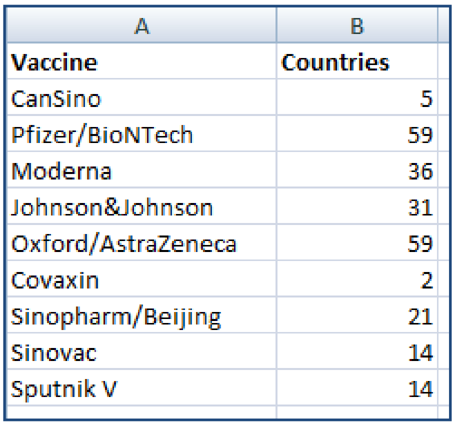
Date (DD-MM-YYYY)	Vaccinated once	Fully vaccinated
16-01-2021	191181	
16-02-2021	8641002	216339
16-03-2021	28862037	6202499
16-04-2021	104408484	15529157
16-05-2021	141764335	40486671
16-06-2021	210442950	47882950
16-07-2021	318665226	81030653
16-08-2021	428681772	121838266
16-09-2021	582842108	188499533
16-10-2021	693992112	278694030
16-11-2021	756052687	377635526
16-12-2021	823451038	533040210

C. Data Set 2: Total number of vaccinated people across different countries ('Countries.csv')



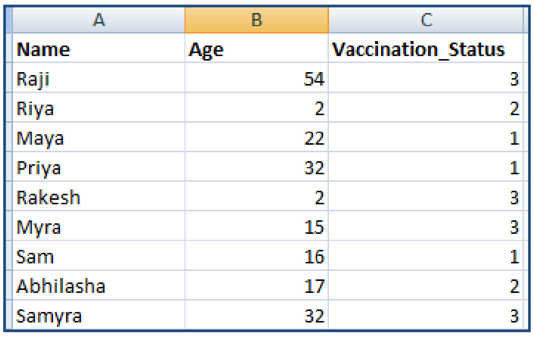
Country	People Vaccinated Once	People Fully Vaccinated
Brazil	165471288	141702263
India	829219869	551003319
Russia	71858958	63708591
UK	51498034	47051876
USA	241881712	202527536
China	1225000000	1193466000

D. Data Set 3:Number of Countries Using different Vaccine Brands (‘VacVsCountries.csv’):



Vaccine	Countries
CanSino	5
Pfizer/BioNTech	59
Moderna	36
Johnson&Johnson	31
Oxford/AstraZeneca	59
Covaxin	2
Sinopharm/Beijing	21
Sinovac	14
Sputnik V	14

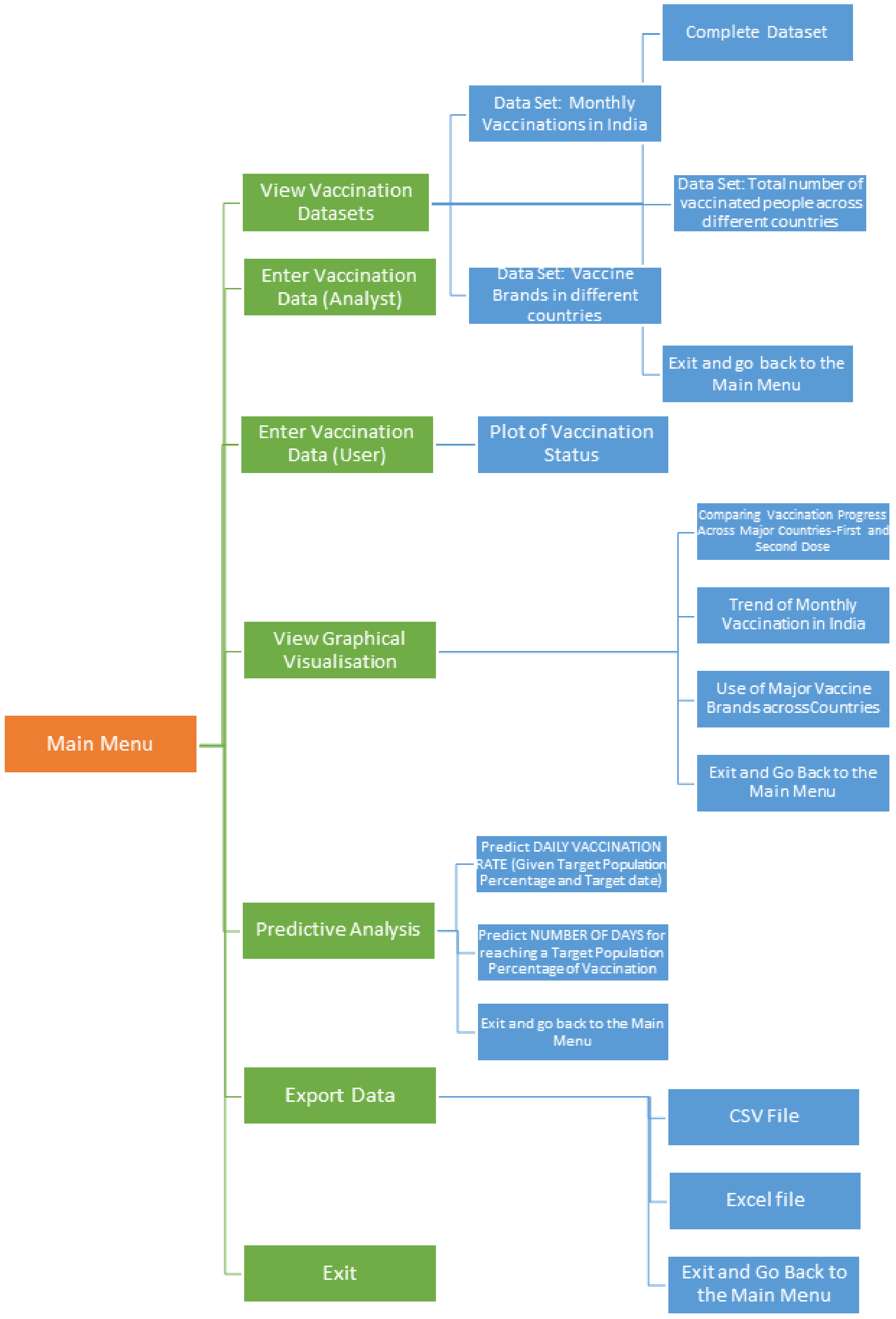
E. Data Table : User Data (‘UserDataFile.csv')



Name	Age	Vaccination_Status
Raji	54	3
Riya	2	2
Maya	22	1
Priya	32	1
Rakesh	2	3
Myra	15	3
Sam	16	1
Abhilasha	17	2
Samyra	32	3

4.3. Software Design:

System Design:


Main Menu
View Vaccination Datasets
Enter Vaccination Data (Analyst)
Enter Vaccination Data (User)
View Graphical Visualisation
Predictive Analysis
Export Data
Exit
Complete Dataset
Data Set: Monthly Vaccinations in India
Data Set: Total number of vaccinated people across different countries
Data Set: Vaccine Brands in different countries
Exit and go back to the Main Menu
Plot of Vaccination Status
Comparing Vaccination Progress Across Major Countries-First and Second Dose
Trend of Monthly Vaccination in India
Use of Major Vaccine Brands acrossCountries
Exit and Go Back to the Main Menu
Predict DAILY VACCINATION RATE (Given Target Population Percentage and Target date)
Predict NUMBER OF DAYS for reaching a Target Population Percentage of Vaccination
Exit and go back to the Main Menu
CSV File
Excel file
Exit and Go Back to the Main Menu

Prediction Algorithms:

A. *Predict Daily Vaccination Rate:*

To predict daily vaccination rate required to vaccinate a Target population in a given country by a Target Date;

Data from Master Vaccination Database:

- ***Daily Vaccination Rate***
- ***Population Vaccinated***
- ***Country Population**(from general sources)*

User Inputs:

- ***Target Population Percentage***
- ***Target Date***

Algorithm:

1. *Read **Target Population Percentage** and **Target Date** from User*
2. *Filter Master data on country and Get **Daily Vaccination Rate** from last 2 cumulative daily Vaccination numbers for the country*
3. *Get **Last Vaccination Data Entry Date** from **file***
4. *Calculate **No of days** from **Last Vaccination Data Entry Date** to **Target Date***
5. *Calculate **Population to be Vaccinated***

   Population to be Vaccinated = Country Population Target Population Percentage – Population Vaccinated*
6. *Calculate **Predicted Daily Vaccine Rate***

   Predicted Daily Vaccine Rate = Population to be Vaccinated /No.of Days

B. *Predict No of Days:*

To predict Number of days to vaccinate a Target population in a given country at the current Vaccination Rate;

Data from Master Vaccination Database:

- ***Daily Vaccination Rate***
- ***Population Vaccinated***
- ***Country Population**(from general sources)*

User Inputs:

- ***Target Population Percentage***

Algorithm:

1. *Read **Target Population Percentage** from User*
2. *Filter Master data on country and Get **Daily Vaccination Rate** from last 2 cumulative daily Vaccination numbers for the country*
3. *Get **Last Vaccination Data Entry Date** from **file***
4. *Calculate **Population to be Vaccinated***

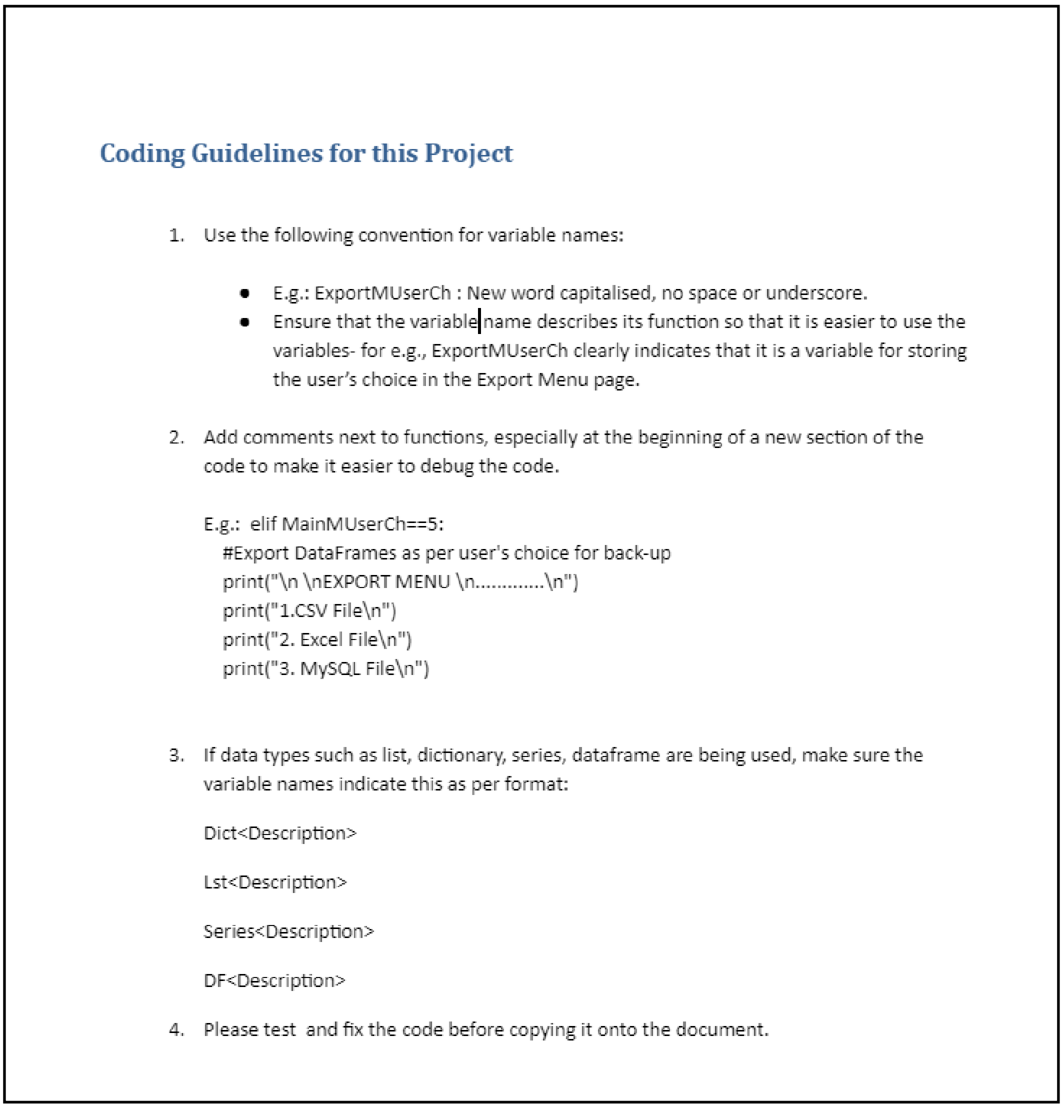
   Population to be Vaccinated = Country Population Target Population Percentage – Population Vaccinated*
5. *Calculate **Predicted Daily Vaccine Rate***

   Predicted Number of Days = Population to be Vaccinated /Daily Vaccination date

4.4 Design Considerations:

The following design considerations were planned and implemented while coding:

1. Modularity; the use of functions reduces run-time and increases readability and maintainability of the code.
2. Error Handling: the use of exception and error handling, in case invalid choices are entered.
3. Ease of Use: Readable and easy-to-use menu-based User Interface.
4. Visualisations: The use of Python’s Matplotlib Library for appealing and tangible graphs.
5. Maintainability: The use of coding guidelines for ease of maintenance.



Coding Guidelines for this Project

1. Use the following convention for variable names:

   - E.g.: ExportMUserCh : New word capitalised, no space or underscore.
   - Ensure that the variable name describes its function so that it is easier to use the variables- for e.g., ExportMUserCh clearly indicates that it is a variable for storing the user's choice in the Export Menu page.

2. Add comments next to functions, especially at the beginning of a new section of the code to make it easier to debug the code.

   ```
   E.g.:  elif MainMUserCh==5:
     #Export DataFrames as per user's choice for back-up
     print("\n \nEXPORT MENU \n............\n")
     print("1.CSV File\n")
     print("2. Excel File\n")
     print("3. MySQL File\n")
   ```

3. If data types such as list, dictionary, series, dataframe are being used, make sure the variable names indicate this as per format:

   Dict<Description>

   Lst<Description>

   Series<Description>

   DF<Description>

4. Please test and fix the code before copying it onto the document.

5. System Implementation

5.1. The Hardware used:

While developing the system, the hardware used was:

HP Pavilion 23 Core i5 4570T CPU (4GB RAM)

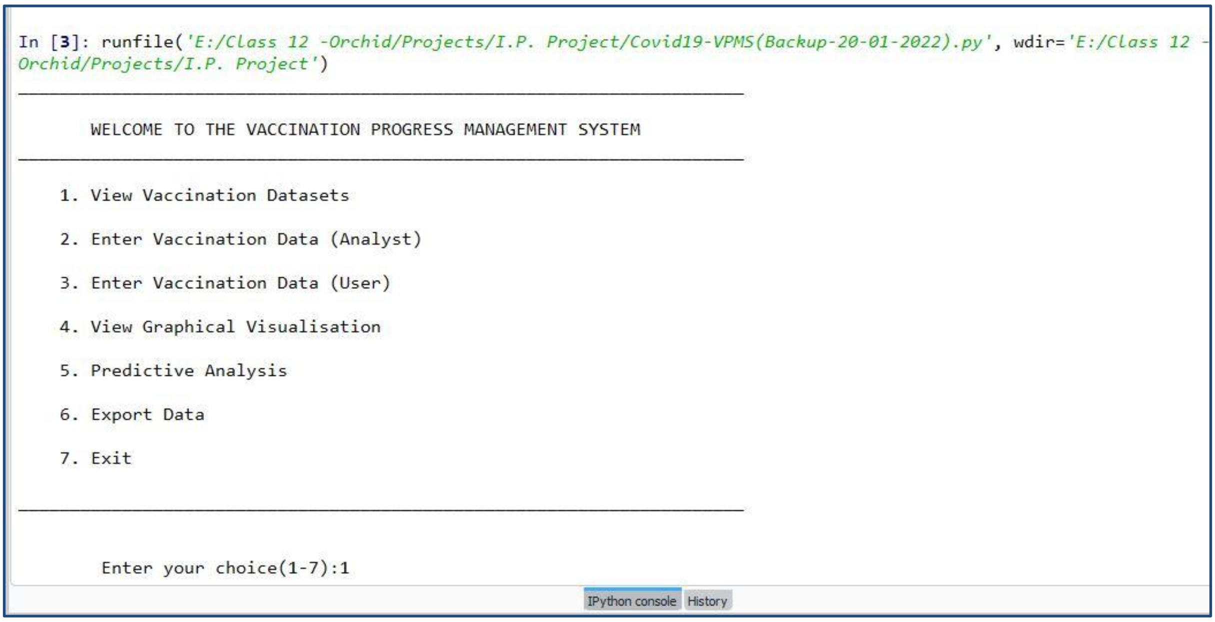
5.2. The Software used:

- Microsoft Windows® 8.1 as Operating System.
- Spyder (Python 3.8) as Front-end Development environment.
- MS-Word for documentation.
- MS-Excel for CSV Data Files.

5.3 Input, Output Screens and Coding

Menu System

Main Menu

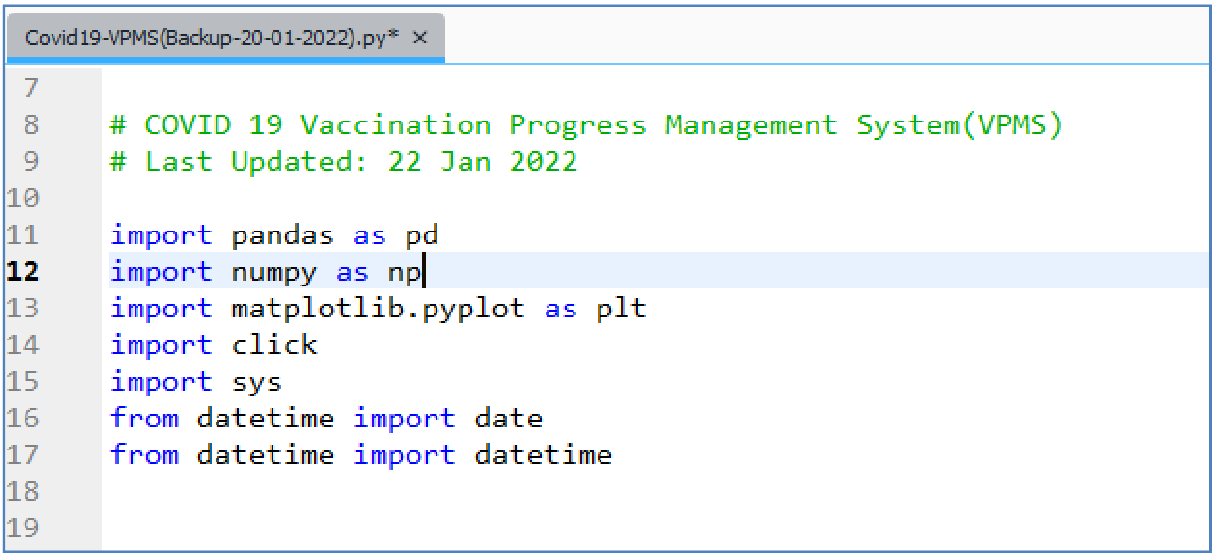


```
In [3]: runfile('E:/Class 12 -Orchid/Projects/I.P. Project/Covid19-VPMS(Backup-20-01-2022).py', wdir='E:/Class 12 -
Orchid/Projects/I.P. Project')
______________________________________________________________________

       WELCOME TO THE VACCINATION PROGRESS MANAGEMENT SYSTEM
______________________________________________________________________

    1. View Vaccination Datasets

    2. Enter Vaccination Data (Analyst)

    3. Enter Vaccination Data (User)

    4. View Graphical Visualisation

    5. Predictive Analysis

    6. Export Data

    7. Exit

______________________________________________________________________


        Enter your choice(1-7):1
```

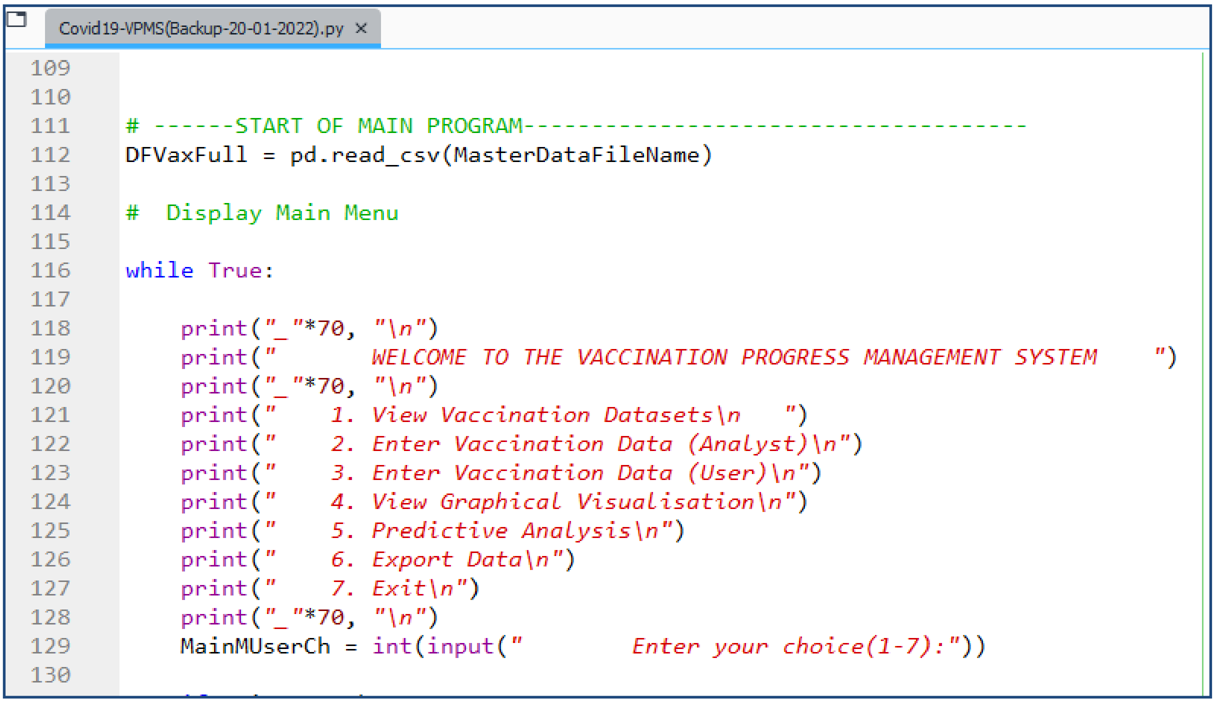
Upon installation of the VPMS Software, the Main Menu features on the user's screen, as shown above. It provides the user with seven options, as indicated.

The first six choices lead the user to the respective sub-menus. The final ‘Exit’ option allows the user to exit the program.

Code for Main Menu



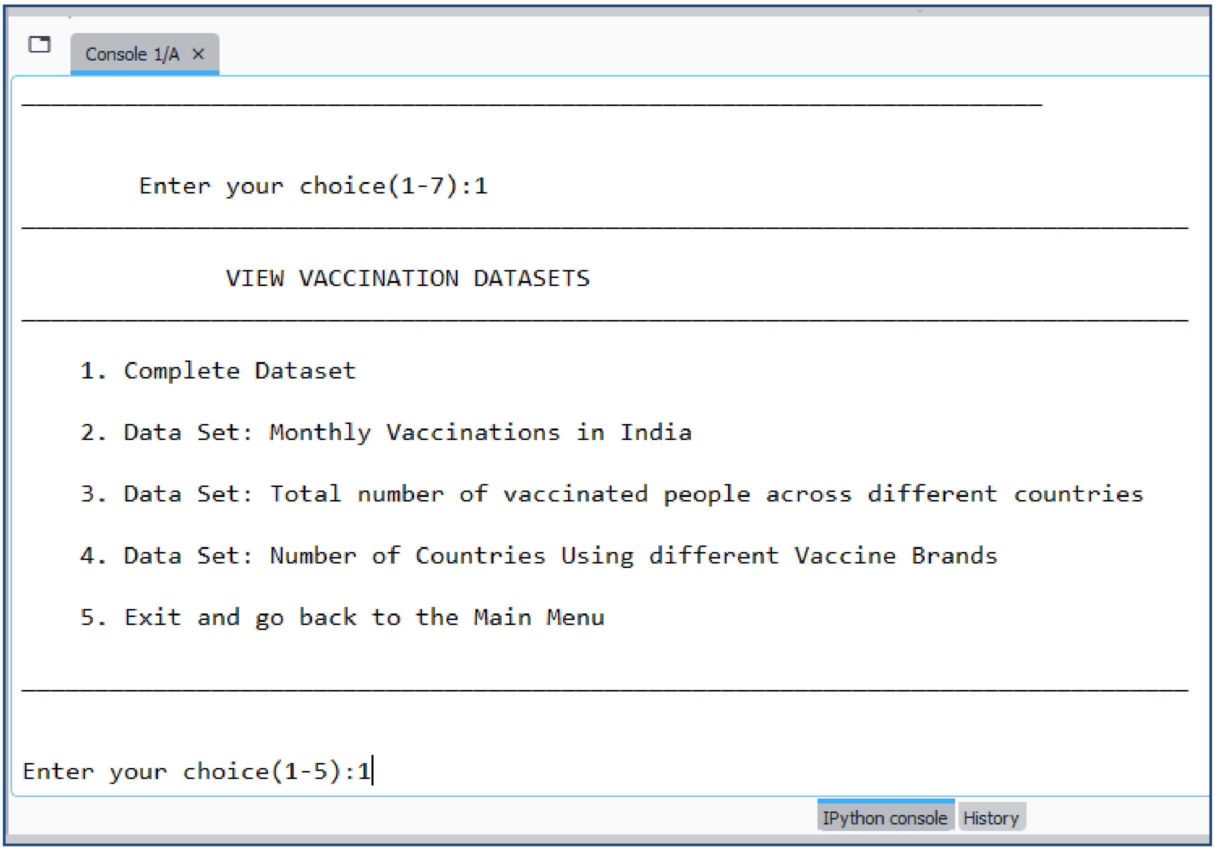
```python

# COVID 19 Vaccination Progress Management System(VPMS)
# Last Updated: 22 Jan 2022

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import click
import sys
from datetime import date
from datetime import datetime
```



```python


# ------START OF MAIN PROGRAM------------------------------------
DFVaxFull = pd.read_csv(MasterDataFileName)

#  Display Main Menu

while True:

    print("_"*70, "\n")
    print("       WELCOME TO THE VACCINATION PROGRESS MANAGEMENT SYSTEM    ")
    print("_"*70, "\n")
    print("    1. View Vaccination Datasets\n   ")
    print("    2. Enter Vaccination Data (Analyst)\n")
    print("    3. Enter Vaccination Data (User)\n")
    print("    4. View Graphical Visualisation\n")
    print("    5. Predictive Analysis\n")
    print("    6. Export Data\n")
    print("    7. Exit\n")
    print("_"*70, "\n")
    MainMUserCh = int(input("        Enter your choice(1-7):"))

```

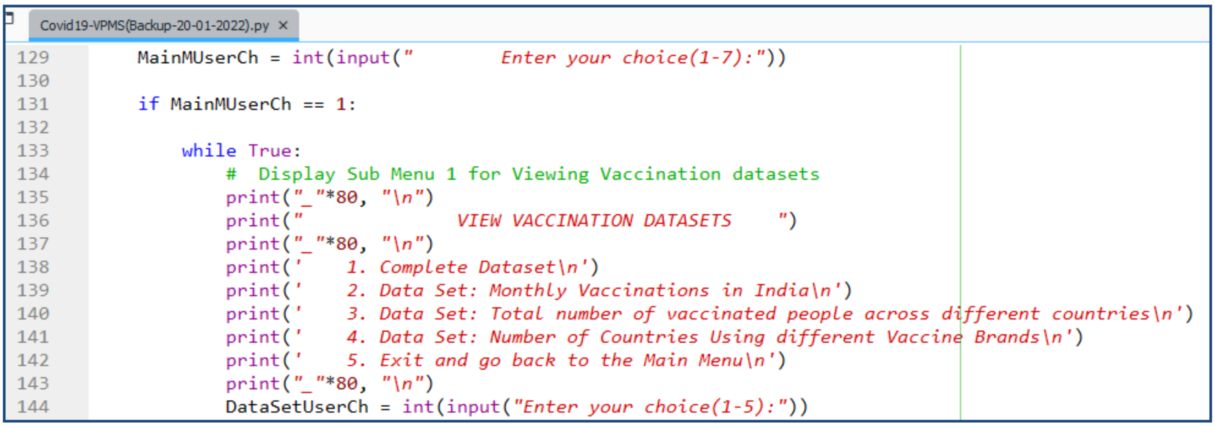
Sub-Menus

1.'View Data Sets' Sub-Menu



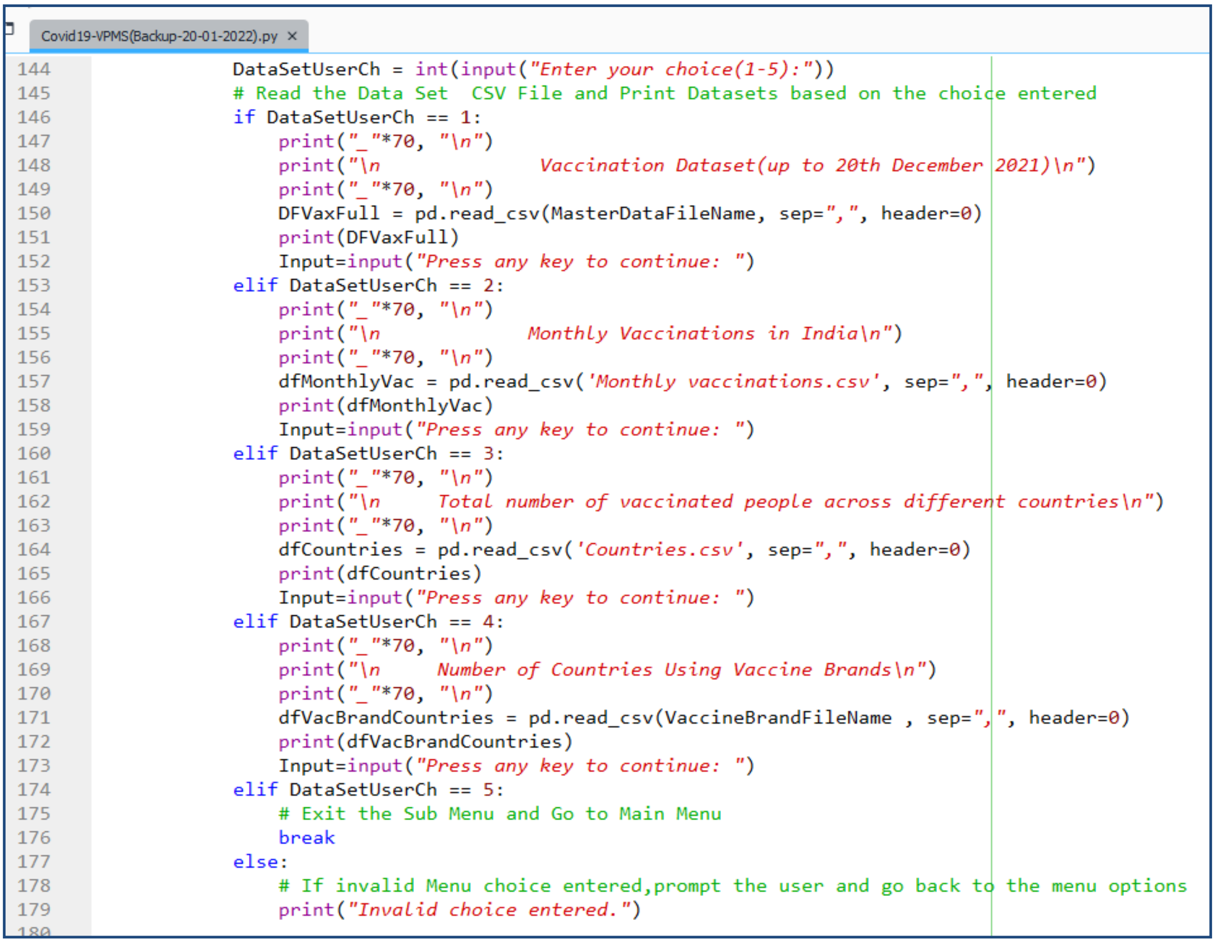
```
____________________________________________________________________________


        Enter your choice(1-7):1
________________________________________________________________________________

              VIEW VACCINATION DATASETS
________________________________________________________________________________

    1. Complete Dataset

    2. Data Set: Monthly Vaccinations in India

    3. Data Set: Total number of vaccinated people across different countries

    4. Data Set: Number of Countries Using different Vaccine Brands

    5. Exit and go back to the Main Menu

________________________________________________________________________________


Enter your choice(1-5):1
```

Code for 'View Vaccination Data Sets' Sub Menu



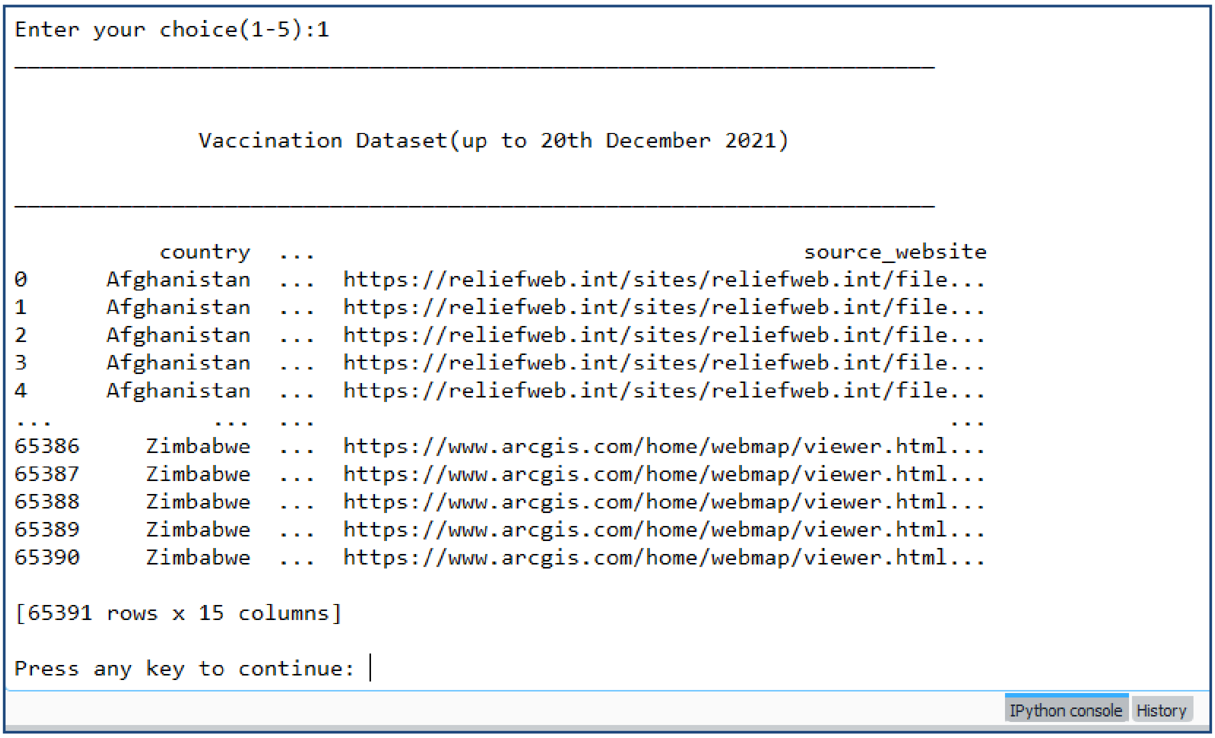
```python
    MainMUserCh = int(input("        Enter your choice(1-7):"))

    if MainMUserCh == 1:

        while True:
            #  Display Sub Menu 1 for Viewing Vaccination datasets
            print("_"*80, "\n")
            print("              VIEW VACCINATION DATASETS    ")
            print("_"*80, "\n")
            print('    1. Complete Dataset\n')
            print('    2. Data Set: Monthly Vaccinations in India\n')
            print('    3. Data Set: Total number of vaccinated people across different countries\n')
            print('    4. Data Set: Number of Countries Using different Vaccine Brands\n')
            print('    5. Exit and go back to the Main Menu\n')
            print("_"*80, "\n")
            DataSetUserCh = int(input("Enter your choice(1-5):"))
```

Based on the user's choice of Data Set (Options 1 to 4), the code for the display of the selected Dataset is invoked and the output is displayed.



```python
            DataSetUserCh = int(input("Enter your choice(1-5):"))
            # Read the Data Set  CSV File and Print Datasets based on the choice entered
            if DataSetUserCh == 1:
                print("_"*70, "\n")
                print("\n              Vaccination Dataset(up to 20th December 2021)\n")
                print("_"*70, "\n")
                DFVaxFull = pd.read_csv(MasterDataFileName, sep=",", header=0)
                print(DFVaxFull)
                Input=input("Press any key to continue: ")
            elif DataSetUserCh == 2:
                print("_"*70, "\n")
                print("\n             Monthly Vaccinations in India\n")
                print("_"*70, "\n")
                dfMonthlyVac = pd.read_csv('Monthly vaccinations.csv', sep=",", header=0)
                print(dfMonthlyVac)
                Input=input("Press any key to continue: ")
            elif DataSetUserCh == 3:
                print("_"*70, "\n")
                print("\n     Total number of vaccinated people across different countries\n")
                print("_"*70, "\n")
                dfCountries = pd.read_csv('Countries.csv', sep=",", header=0)
                print(dfCountries)
                Input=input("Press any key to continue: ")
            elif DataSetUserCh == 4:
                print("_"*70, "\n")
                print("\n     Number of Countries Using Vaccine Brands\n")
                print("_"*70, "\n")
                dfVacBrandCountries = pd.read_csv(VaccineBrandFileName , sep=",", header=0)
                print(dfVacBrandCountries)
                Input=input("Press any key to continue: ")
            elif DataSetUserCh == 5:
                # Exit the Sub Menu and Go to Main Menu
                break
            else:
                # If invalid Menu choice entered,prompt the user and go back to the menu options
                print("Invalid choice entered.")
```

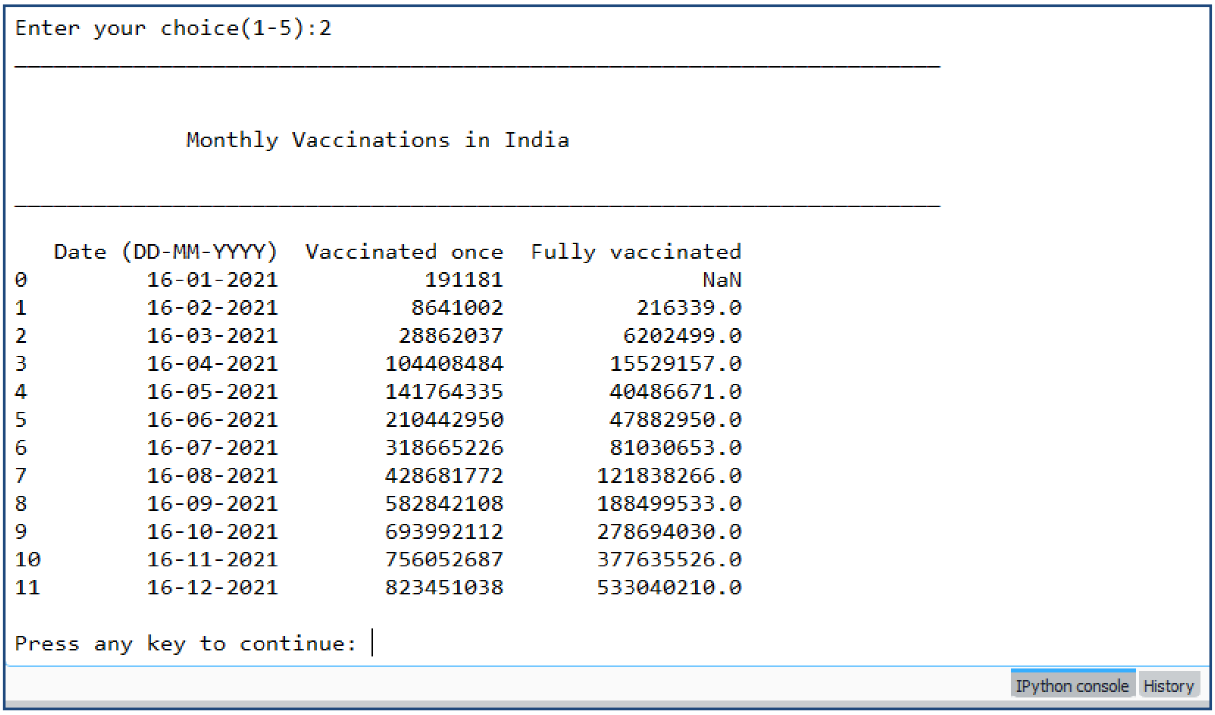
Option 1 Selected: Output Screen



```
Enter your choice(1-5):1
______________________________________________________________________ 


              Vaccination Dataset(up to 20th December 2021)

______________________________________________________________________ 

           country  ...                                     source_website
0      Afghanistan  ...  https://reliefweb.int/sites/reliefweb.int/file...
1      Afghanistan  ...  https://reliefweb.int/sites/reliefweb.int/file...
2      Afghanistan  ...  https://reliefweb.int/sites/reliefweb.int/file...
3      Afghanistan  ...  https://reliefweb.int/sites/reliefweb.int/file...
4      Afghanistan  ...  https://reliefweb.int/sites/reliefweb.int/file...
...            ...  ...                                                ...
65386     Zimbabwe  ...  https://www.arcgis.com/home/webmap/viewer.html...
65387     Zimbabwe  ...  https://www.arcgis.com/home/webmap/viewer.html...
65388     Zimbabwe  ...  https://www.arcgis.com/home/webmap/viewer.html...
65389     Zimbabwe  ...  https://www.arcgis.com/home/webmap/viewer.html...
65390     Zimbabwe  ...  https://www.arcgis.com/home/webmap/viewer.html...

[65391 rows x 15 columns]

Press any key to continue: 
```

Option 1 displays the original dataset imported from Kaggle as a DataFrame with 65,391 rows and 15 columns.

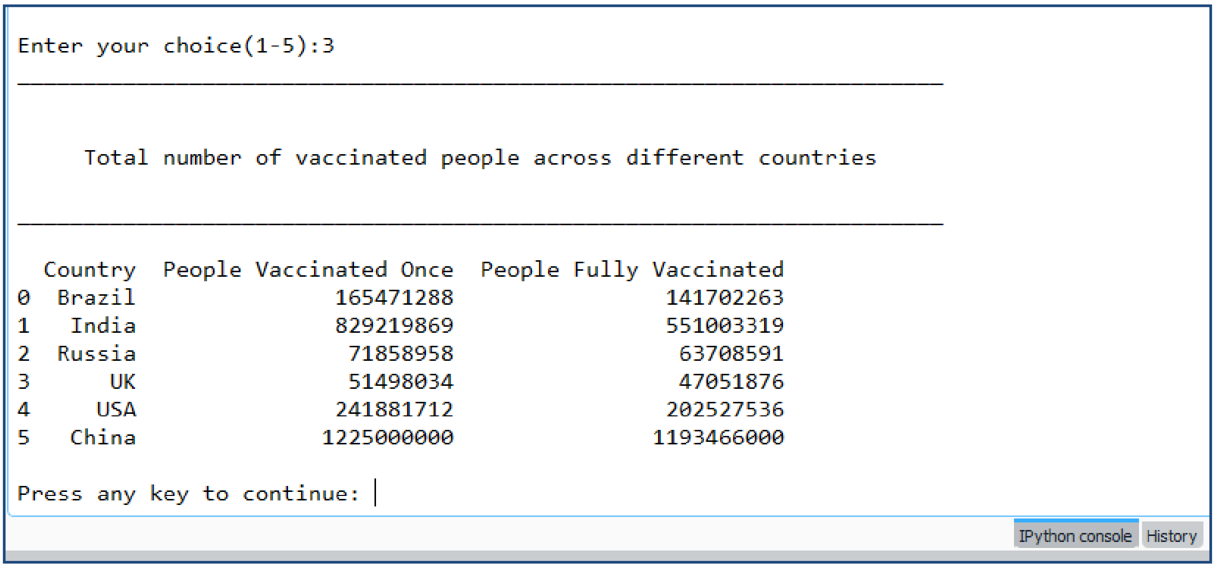
Option 2: Selected: Output Screen



```
Enter your choice(1-5):2
______________________________________________________________________


             Monthly Vaccinations in India


______________________________________________________________________

   Date (DD-MM-YYYY)  Vaccinated once  Fully vaccinated
0         16-01-2021           191181               NaN
1         16-02-2021          8641002          216339.0
2         16-03-2021         28862037         6202499.0
3         16-04-2021        104408484        15529157.0
4         16-05-2021        141764335        40486671.0
5         16-06-2021        210442950        47882950.0
6         16-07-2021        318665226        81030653.0
7         16-08-2021        428681772       121838266.0
8         16-09-2021        582842108       188499533.0
9         16-10-2021        693992112       278694030.0
10        16-11-2021        756052687       377635526.0
11        16-12-2021        823451038       533040210.0

Press any key to continue: |
```

Option 2 displays a subset of the original dataset, showing the number of citizens once vaccinated and fully vaccinated, month-wise.

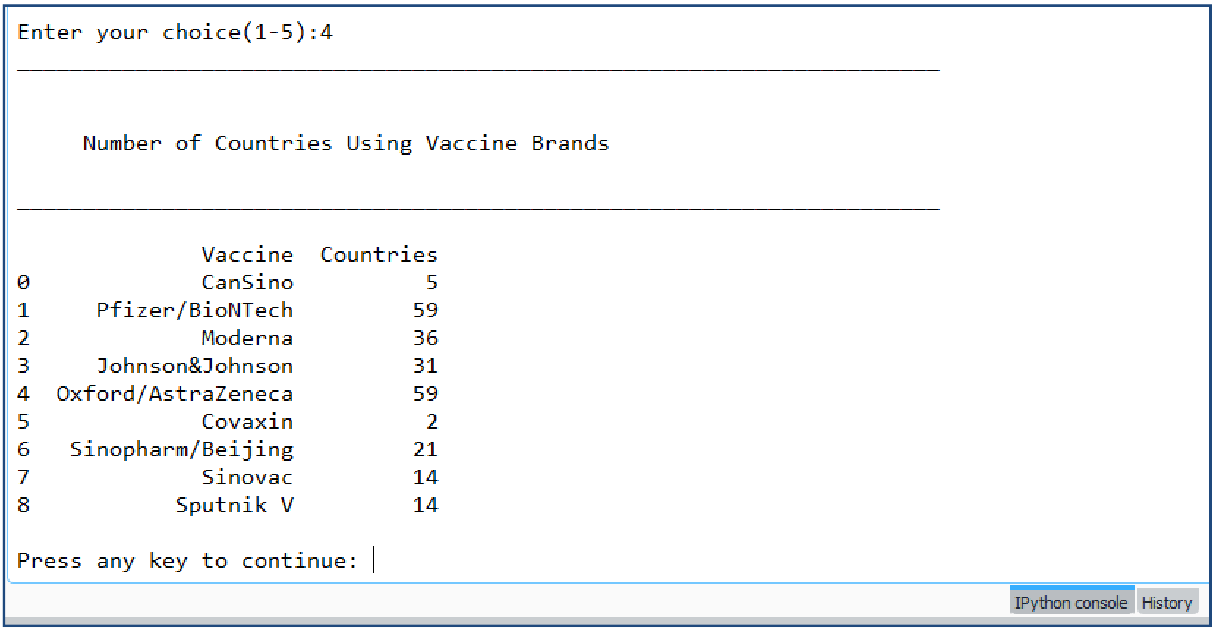
Option 3: Selected: Output Screen



```
Enter your choice(1-5):3
______________________________________________________________________


     Total number of vaccinated people across different countries


______________________________________________________________________

  Country  People Vaccinated Once  People Fully Vaccinated
0  Brazil               165471288                141702263
1   India               829219869                551003319
2  Russia                71858958                 63708591
3      UK                51498034                 47051876
4     USA               241881712                202527536
5   China              1225000000               1193466000

Press any key to continue: |
```

Option 3 displays a subset of the original dataset, showing the total number of people vaccinated once and the number of people fully vaccinated across six countries.

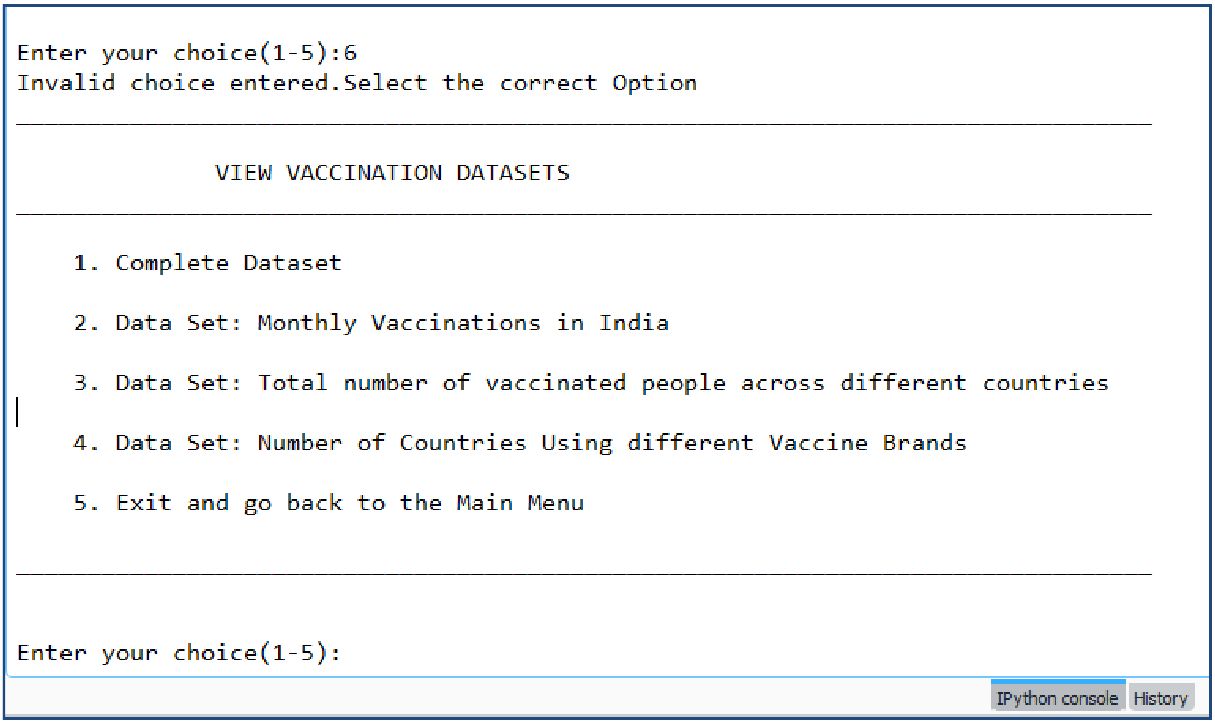
Option 4: Selected: Output Screen



```
Enter your choice(1-5):4
______________________________________________________________________


     Number of Countries Using Vaccine Brands

______________________________________________________________________

              Vaccine  Countries
0             CanSino          5
1      Pfizer/BioNTech         59
2              Moderna         36
3      Johnson&Johnson         31
4   Oxford/AstraZeneca         59
5              Covaxin          2
6    Sinopharm/Beijing         21
7              Sinovac         14
8            Sputnik V         14

Press any key to continue: 
```

Option 4 displays a subset of the original dataset, showing the various vaccine brands against the corresponding number of countries in which they are being used.

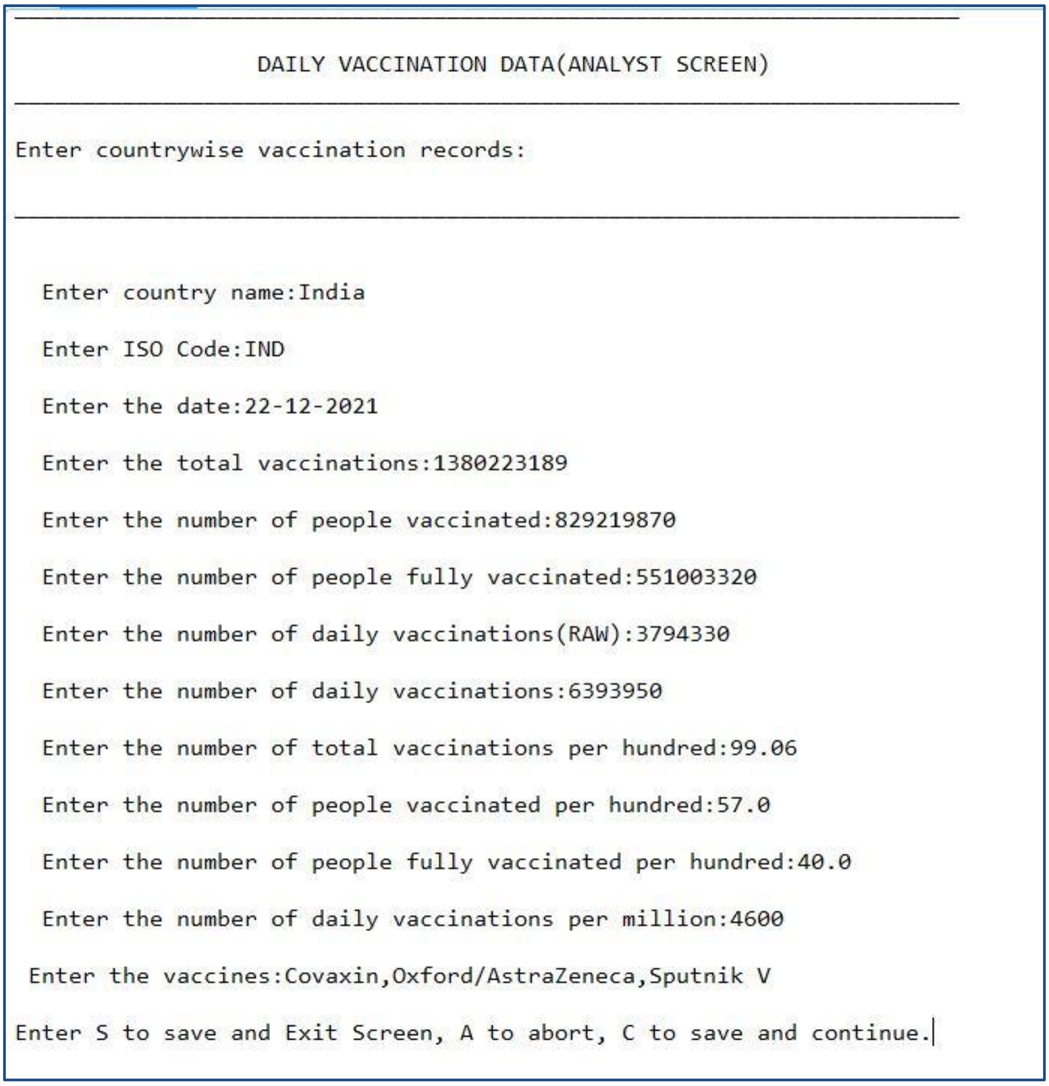
Invalid Option Selected-Output



```
Enter your choice(1-5):6
Invalid choice entered.Select the correct Option
________________________________________________________________

              VIEW VACCINATION DATASETS
________________________________________________________________

    1. Complete Dataset

    2. Data Set: Monthly Vaccinations in India

    3. Data Set: Total number of vaccinated people across different countries

    4. Data Set: Number of Countries Using different Vaccine Brands

    5. Exit and go back to the Main Menu

________________________________________________________________


Enter your choice(1-5):
```

When a number other than 1 to 5 is entered, the user is informed that their choice is invalid.

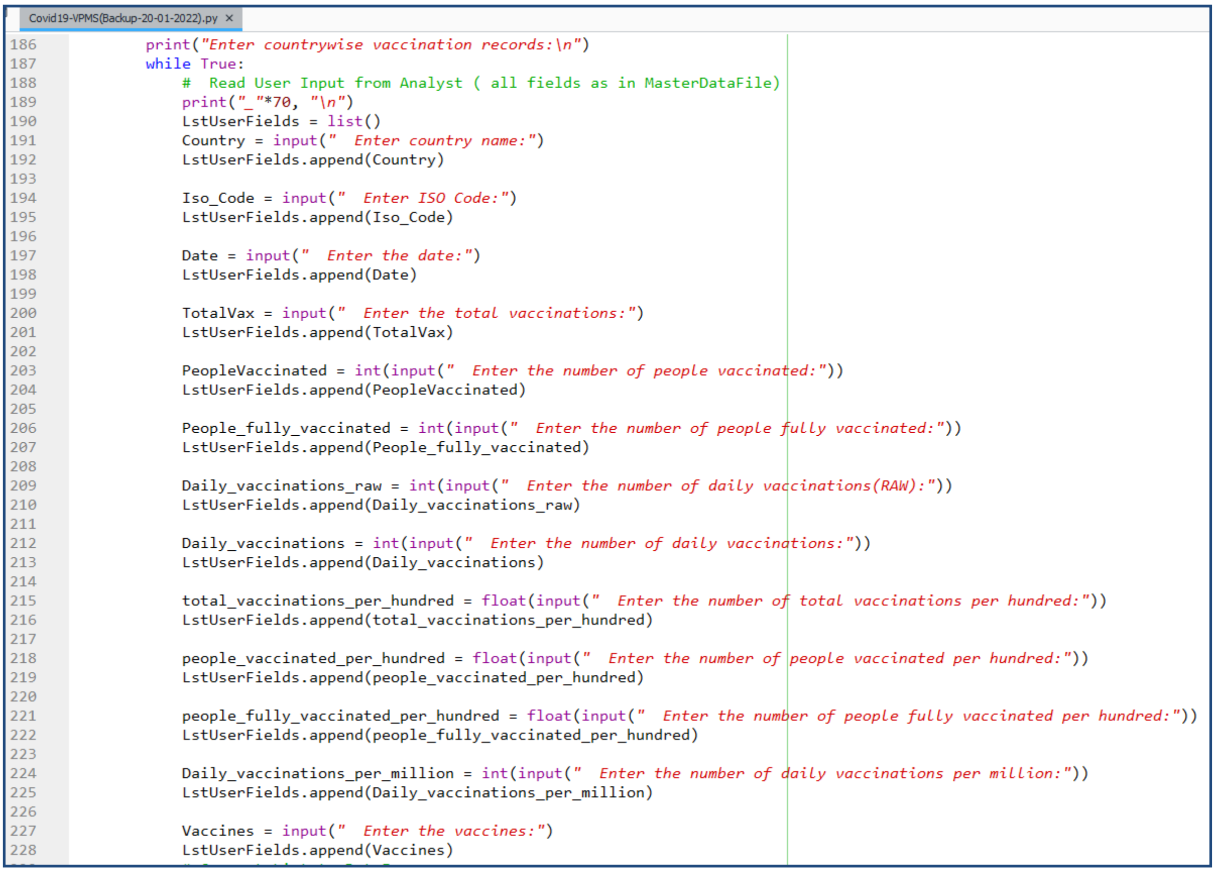
2. Enter Vaccination Data (Analyst) Input Screen



```
______________________________________________________________________

                  DAILY VACCINATION DATA(ANALYST SCREEN)
______________________________________________________________________

Enter countrywise vaccination records:

______________________________________________________________________


  Enter country name:India

  Enter ISO Code:IND

  Enter the date:22-12-2021

  Enter the total vaccinations:1380223189

  Enter the number of people vaccinated:829219870

  Enter the number of people fully vaccinated:551003320

  Enter the number of daily vaccinations(RAW):3794330

  Enter the number of daily vaccinations:6393950

  Enter the number of total vaccinations per hundred:99.06

  Enter the number of people vaccinated per hundred:57.0

  Enter the number of people fully vaccinated per hundred:40.0

  Enter the number of daily vaccinations per million:4600

 Enter the vaccines:Covaxin,Oxford/AstraZeneca,Sputnik V

Enter S to save and Exit Screen, A to abort, C to save and continue.
```

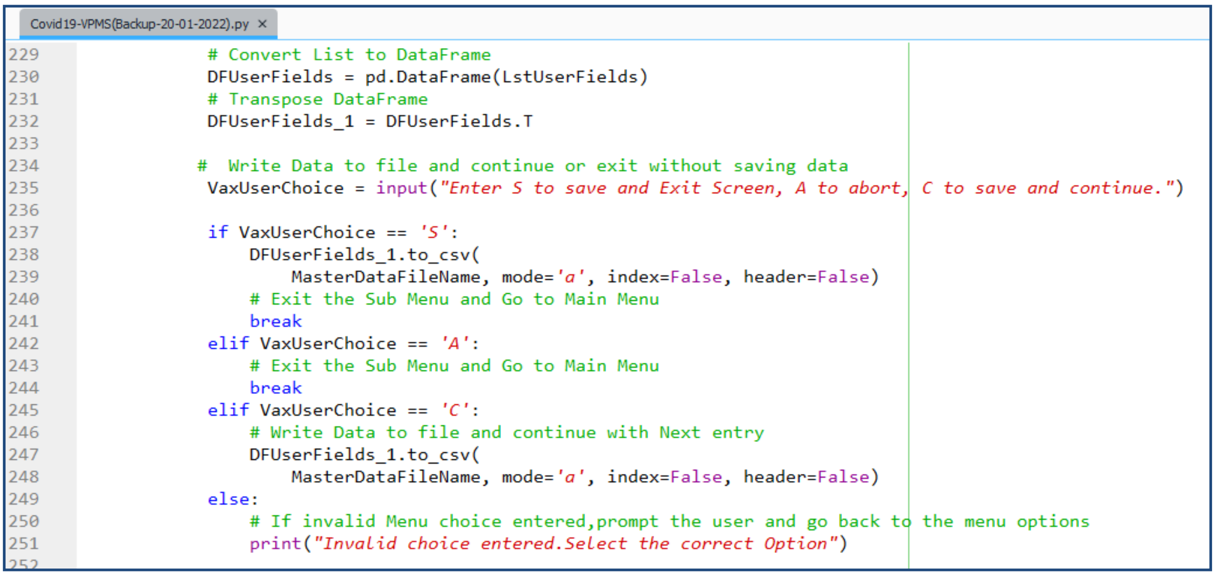
The second choice in the Main Menu is meant for an analyst or administrator who has access to the latest vaccination records.

Enter Vaccination Data (Analyst) Code



```
        print("Enter countrywise vaccination records:\n")
        while True:
            #  Read User Input from Analyst ( all fields as in MasterDataFile)
            print("_"*70, "\n")
            LstUserFields = list()
            Country = input("  Enter country name:")
            LstUserFields.append(Country)

            Iso_Code = input("  Enter ISO Code:")
            LstUserFields.append(Iso_Code)

            Date = input("  Enter the date:")
            LstUserFields.append(Date)

            TotalVax = input("  Enter the total vaccinations:")
            LstUserFields.append(TotalVax)

            PeopleVaccinated = int(input("  Enter the number of people vaccinated:"))
            LstUserFields.append(PeopleVaccinated)

            People_fully_vaccinated = int(input("  Enter the number of people fully vaccinated:"))
            LstUserFields.append(People_fully_vaccinated)

            Daily_vaccinations_raw = int(input("  Enter the number of daily vaccinations(RAW):"))
            LstUserFields.append(Daily_vaccinations_raw)

            Daily_vaccinations = int(input("  Enter the number of daily vaccinations:"))
            LstUserFields.append(Daily_vaccinations)

            total_vaccinations_per_hundred = float(input("  Enter the number of total vaccinations per hundred:"))
            LstUserFields.append(total_vaccinations_per_hundred)

            people_vaccinated_per_hundred = float(input("  Enter the number of people vaccinated per hundred:"))
            LstUserFields.append(people_vaccinated_per_hundred)

            people_fully_vaccinated_per_hundred = float(input("  Enter the number of people fully vaccinated per hundred:"))
            LstUserFields.append(people_fully_vaccinated_per_hundred)

            Daily_vaccinations_per_million = int(input("  Enter the number of daily vaccinations per million:"))
            LstUserFields.append(Daily_vaccinations_per_million)

            Vaccines = input("  Enter the vaccines:")
            LstUserFields.append(Vaccines)
```

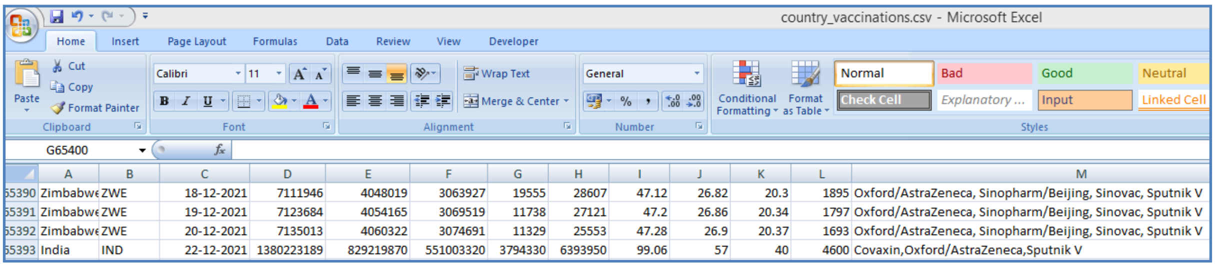
Based on the user's choice, the Daily vaccine record for a country can be saved to the Master data file (country_vaccinations.csv).



```
            # Convert List to DataFrame
            DFUserFields = pd.DataFrame(LstUserFields)
            # Transpose DataFrame
            DFUserFields_1 = DFUserFields.T

           #  Write Data to file and continue or exit without saving data
            VaxUserChoice = input("Enter S to save and Exit Screen, A to abort, C to save and continue.")

            if VaxUserChoice == 'S':
                DFUserFields_1.to_csv(
                    MasterDataFileName, mode='a', index=False, header=False)
                # Exit the Sub Menu and Go to Main Menu
                break
            elif VaxUserChoice == 'A':
                # Exit the Sub Menu and Go to Main Menu
                break
            elif VaxUserChoice == 'C':
                # Write Data to file and continue with Next entry
                DFUserFields_1.to_csv(
                    MasterDataFileName, mode='a', index=False, header=False)
            else:
                # If invalid Menu choice entered,prompt the user and go back to the menu options
                print("Invalid choice entered.Select the correct Option")
```

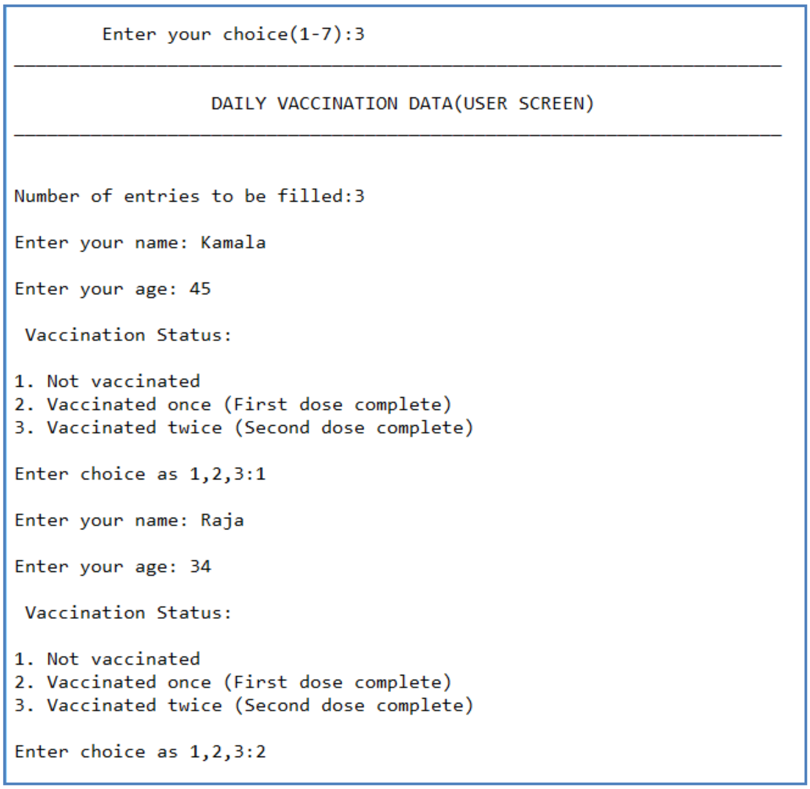
Output:

On clicking ‘S’, the daily vaccination data is appended to the end of the Master Data CSV File. (country_vaccinations.csv)

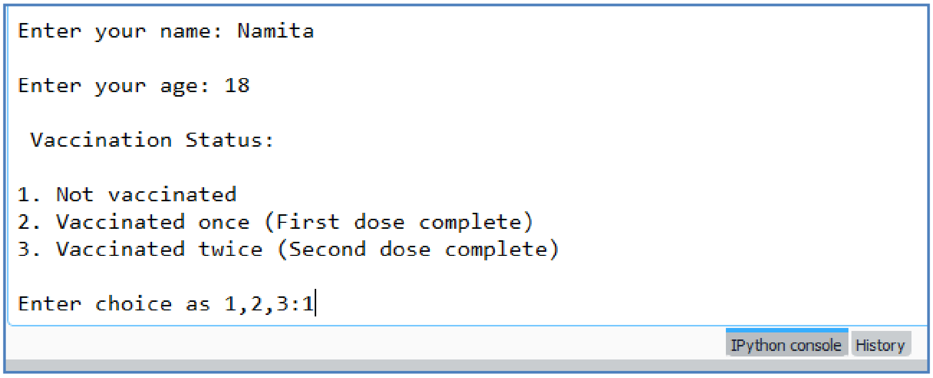


	A	B	C	D	E	F	G	H	I	J	K	L	M
65390	Zimbabwe	ZWE	18-12-2021	7111946	4048019	3063927	19555	28607	47.12	26.82	20.3	1895	Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V
65391	Zimbabwe	ZWE	19-12-2021	7123684	4054165	3069519	11738	27121	47.2	26.86	20.34	1797	Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V
65392	Zimbabwe	ZWE	20-12-2021	7135013	4060322	3074691	11329	25553	47.28	26.9	20.37	1693	Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V
65393	India	IND	22-12-2021	1380223189	829219870	551003320	3794330	6393950	99.06	57	40	4600	Covaxin,Oxford/AstraZeneca,Sputnik V

3. Enter Vaccination Data (User) Input Screen



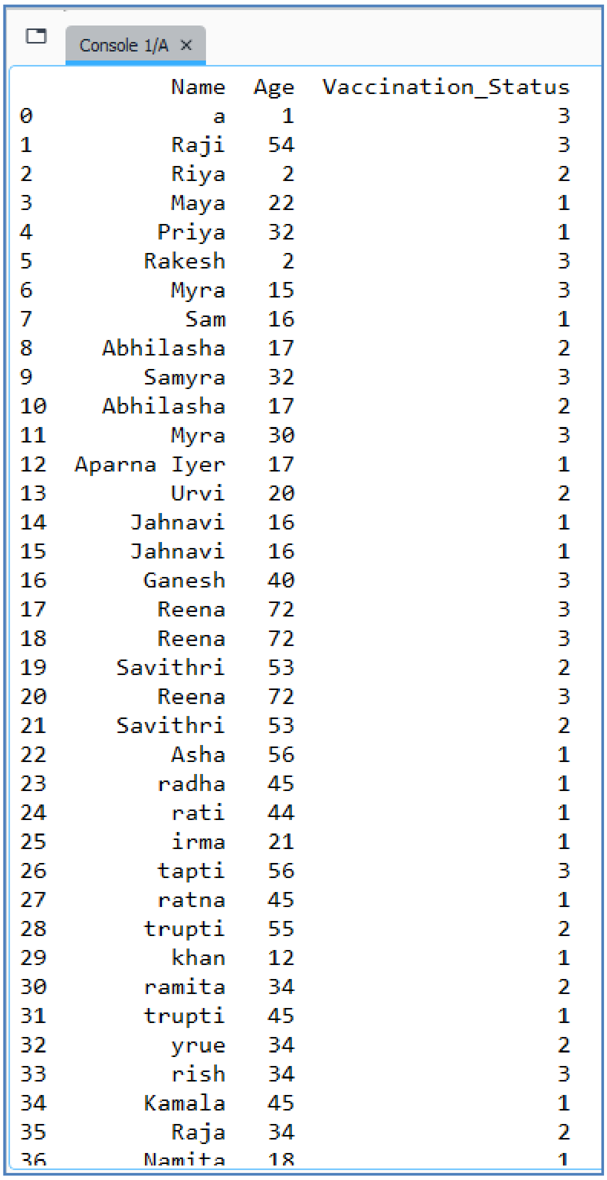
```
        Enter your choice(1-7):3
_______________________________________________________________________

                    DAILY VACCINATION DATA(USER SCREEN)
_______________________________________________________________________


Number of entries to be filled:3

Enter your name: Kamala

Enter your age: 45

 Vaccination Status:

1. Not vaccinated
2. Vaccinated once (First dose complete)
3. Vaccinated twice (Second dose complete)

Enter choice as 1,2,3:1

Enter your name: Raja

Enter your age: 34

 Vaccination Status:

1. Not vaccinated
2. Vaccinated once (First dose complete)
3. Vaccinated twice (Second dose complete)

Enter choice as 1,2,3:2
```



```
Enter your name: Namita

Enter your age: 18

 Vaccination Status:

1. Not vaccinated
2. Vaccinated once (First dose complete)
3. Vaccinated twice (Second dose complete)

Enter choice as 1,2,3:1
```

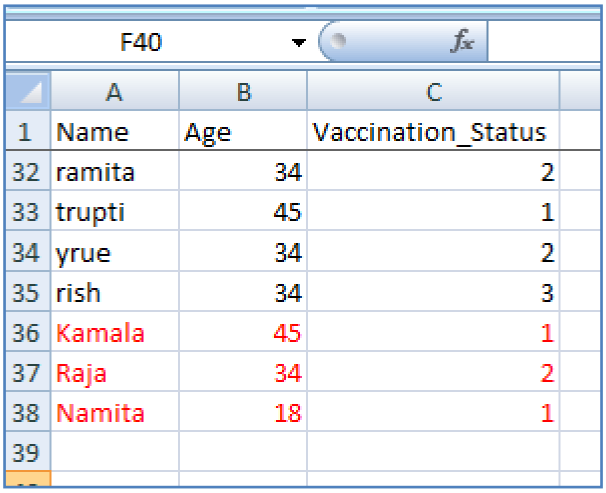
The third choice is meant for the user to enter data that they have access to, that is, vaccination status- whether they have not been vaccinated, have been vaccinated once or have been vaccinated twice. Any other choice would redirect the user to the same screen.

Output: User Data on Screen



```
           Name  Age  Vaccination_Status
0             a    1                   3
1          Raji   54                   3
2          Riya    2                   2
3          Maya   22                   1
4         Priya   32                   1
5        Rakesh    2                   3
6          Myra   15                   3
7           Sam   16                   1
8     Abhilasha   17                   2
9        Samyra   32                   3
10    Abhilasha   17                   2
11         Myra   30                   3
12  Aparna Iyer   17                   1
13         Urvi   20                   2
14      Jahnavi   16                   1
15      Jahnavi   16                   1
16       Ganesh   40                   3
17        Reena   72                   3
18        Reena   72                   3
19     Savithri   53                   2
20        Reena   72                   3
21     Savithri   53                   2
22         Asha   56                   1
23        radha   45                   1
24         rati   44                   1
25         irma   21                   1
26        tapti   56                   3
27        ratna   45                   1
28      trupti   55                   2
29         khan   12                   1
30       ramita   34                   2
31       trupti   45                   1
32         yrue   34                   2
33         rish   34                   3
34       Kamala   45                   1
35         Raja   34                   2
36       Namita   18                   1
```

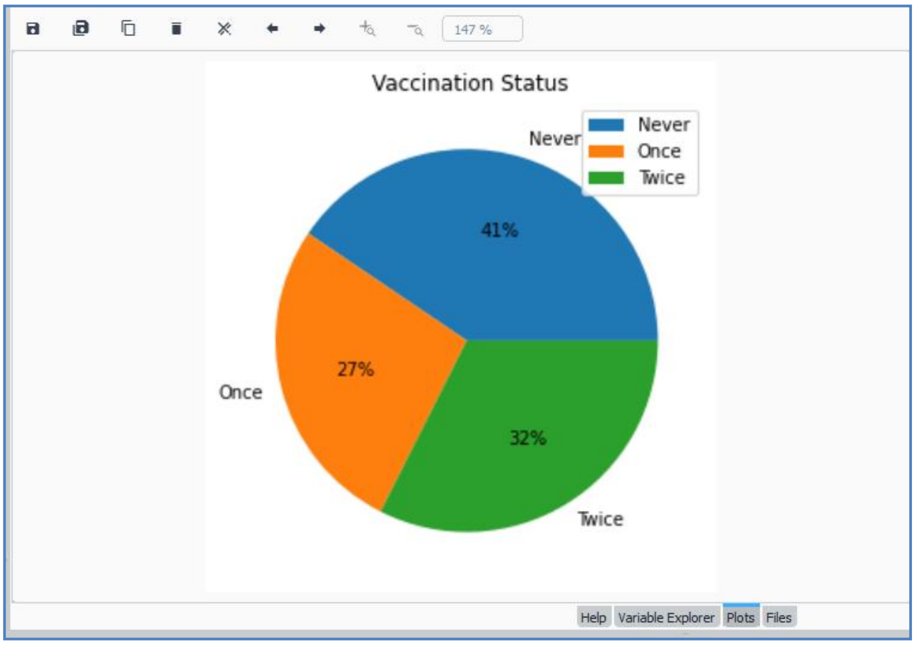
Output: UserDataFile.csv



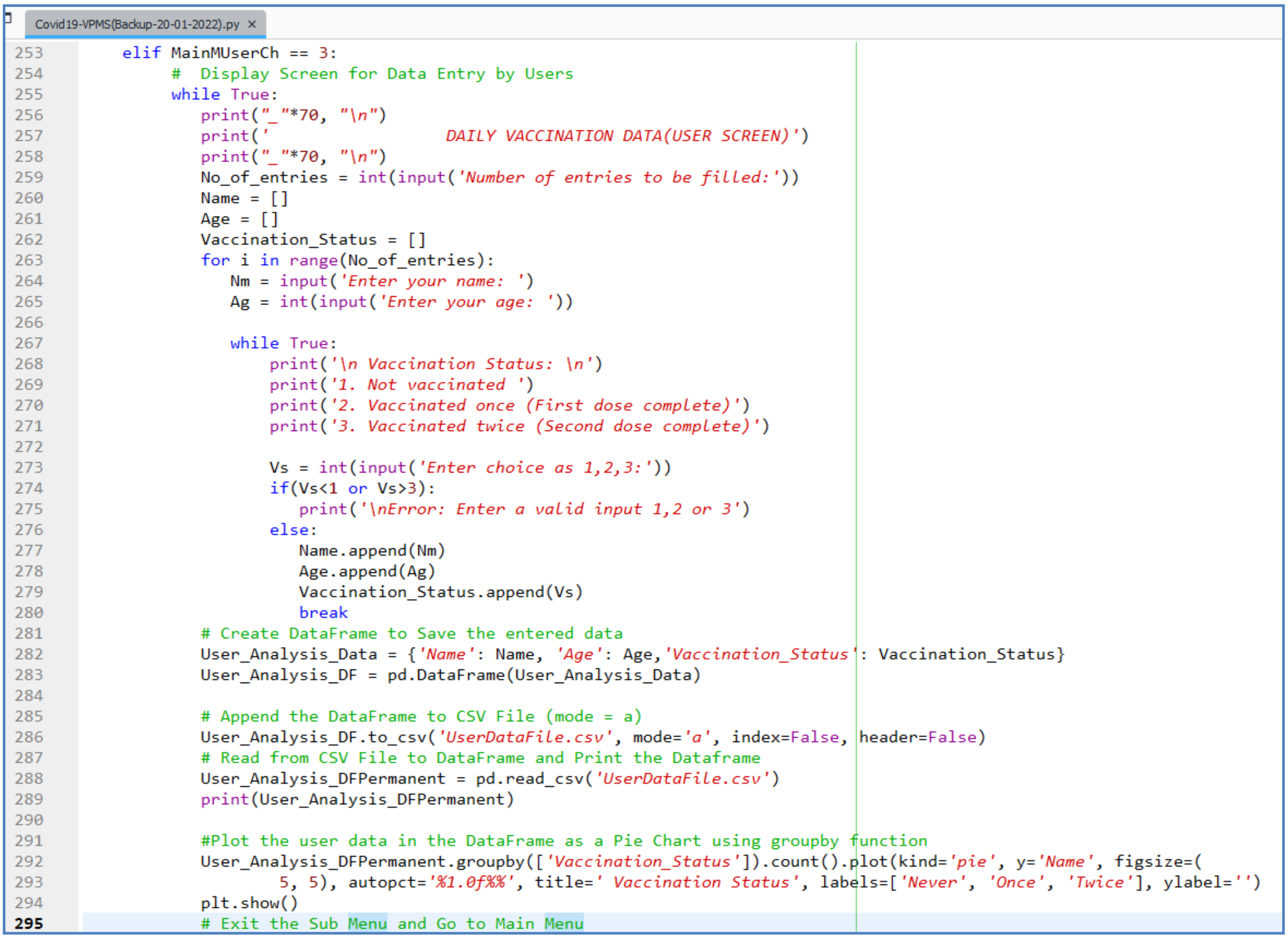
F40

	A	B	C
1	Name	Age	Vaccination_Status
32	ramita	34	2
33	trupti	45	1
34	yrue	34	2
35	rish	34	3
36	Kamala	45	1
37	Raja	34	2
38	Namita	18	1
39			

Plot of User Vaccination Status (Never, Once, Twice)



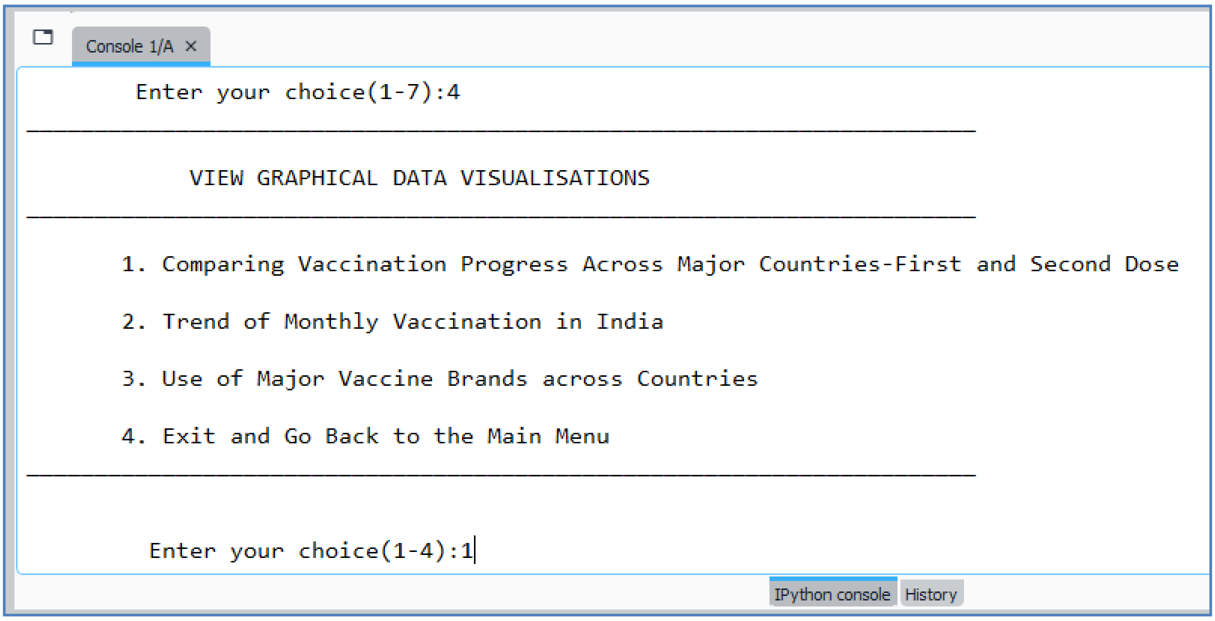
Enter Vaccination Data (User) Code:



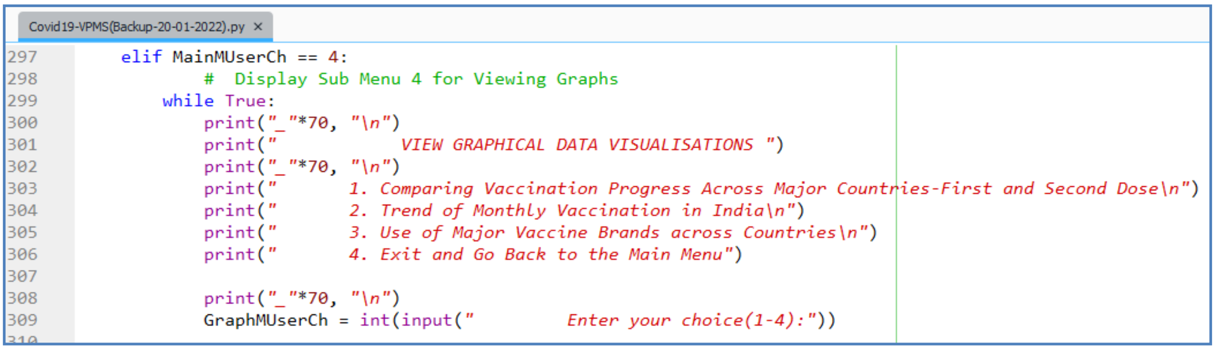
Covid19-VPMS(Backup-20-01-2022).py

```python
    elif MainMUserCh == 3:
         #  Display Screen for Data Entry by Users
         while True:
            print("_"*70, "\n")
            print('                        DAILY VACCINATION DATA(USER SCREEN)')
            print("_"*70, "\n")
            No_of_entries = int(input('Number of entries to be filled:'))
            Name = []
            Age = []
            Vaccination_Status = []
            for i in range(No_of_entries):
               Nm = input('Enter your name:  ')
               Ag = int(input('Enter your age:  '))

               while True:
                   print('\n Vaccination Status:  \n')
                   print('1.  Not vaccinated  ')
                   print('2.  Vaccinated once (First dose complete)')
                   print('3.  Vaccinated twice (Second dose complete)')

                   Vs = int(input('Enter choice as 1,2,3:'))
                   if(Vs<1 or Vs>3):
                      print('\nError: Enter a valid input 1,2 or 3')
                   else:
                      Name.append(Nm)
                      Age.append(Ag)
                      Vaccination_Status.append(Vs)
                      break
            # Create DataFrame to Save the entered data
            User_Analysis_Data = {'Name': Name, 'Age': Age,'Vaccination_Status': Vaccination_Status}
            User_Analysis_DF = pd.DataFrame(User_Analysis_Data)

            # Append the DataFrame to CSV File (mode = a)
            User_Analysis_DF.to_csv('UserDataFile.csv', mode='a', index=False, header=False)
            # Read from CSV File to DataFrame and Print the Dataframe
            User_Analysis_DFPermanent = pd.read_csv('UserDataFile.csv')
            print(User_Analysis_DFPermanent)

            #Plot the user data in the DataFrame as a Pie Chart using groupby function
            User_Analysis_DFPermanent.groupby(['Vaccination_Status']).count().plot(kind='pie', y='Name', figsize=(
                    5, 5), autopct='%1.0f%%', title=' Vaccination Status', labels=['Never', 'Once', 'Twice'], ylabel='')
            plt.show()
            # Exit the Sub Menu and Go to Main Menu
```

4.View Graphical Visualisation



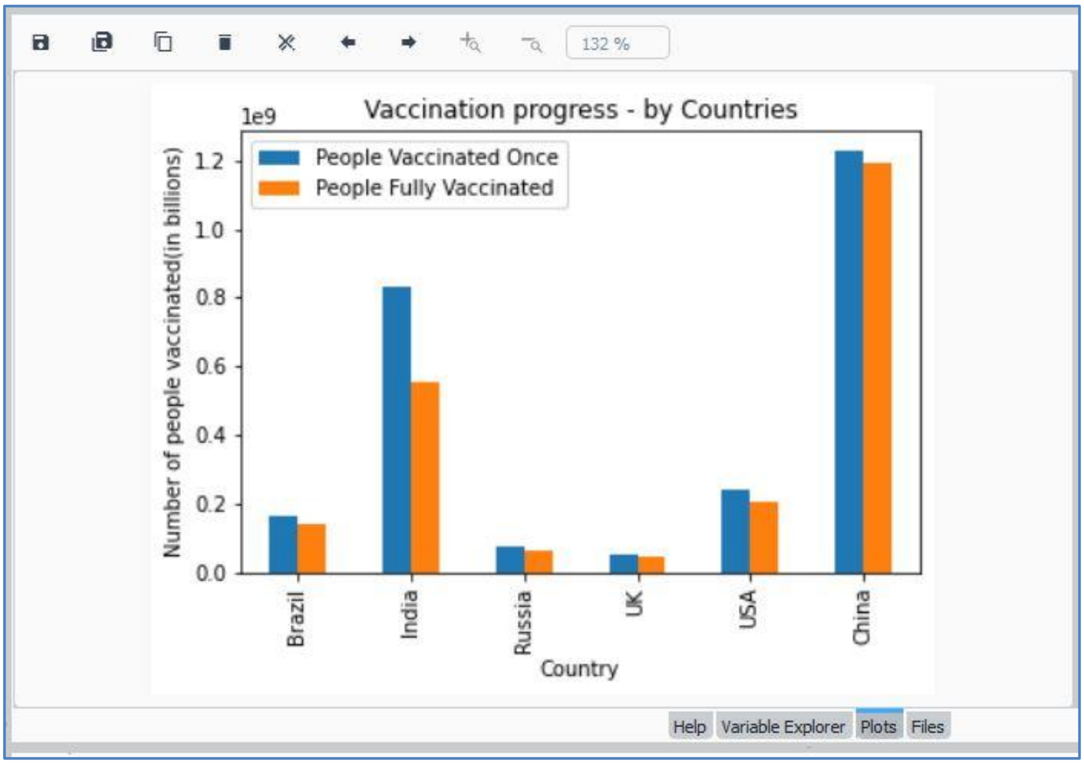
```
           Enter your choice(1-7):4
________________________________________________________________________________

                VIEW GRAPHICAL DATA VISUALISATIONS
________________________________________________________________________________

         1. Comparing Vaccination Progress Across Major Countries-First and Second Dose

         2. Trend of Monthly Vaccination in India

         3. Use of Major Vaccine Brands across Countries

         4. Exit and Go Back to the Main Menu
________________________________________________________________________________


          Enter your choice(1-4):1
```



```
    elif MainMUserCh == 4:
            #  Display Sub Menu 4 for Viewing Graphs
        while True:
            print("_"*70, "\n")
            print("            VIEW GRAPHICAL DATA VISUALISATIONS ")
            print("_"*70, "\n")
            print("       1. Comparing Vaccination Progress Across Major Countries-First and Second Dose\n")
            print("       2. Trend of Monthly Vaccination in India\n")
            print("       3. Use of Major Vaccine Brands across Countries\n")
            print("       4. Exit and Go Back to the Main Menu")

            print("_"*70, "\n")
            GraphMUserCh = int(input("         Enter your choice(1-4):"))
```

Based on the user’s choice, 3 different Graphs are displayed using the Datasets created and displayed for user to view the trends.

4.1. Comparing Vaccination Progress across Major Countries-First and Second Dose)

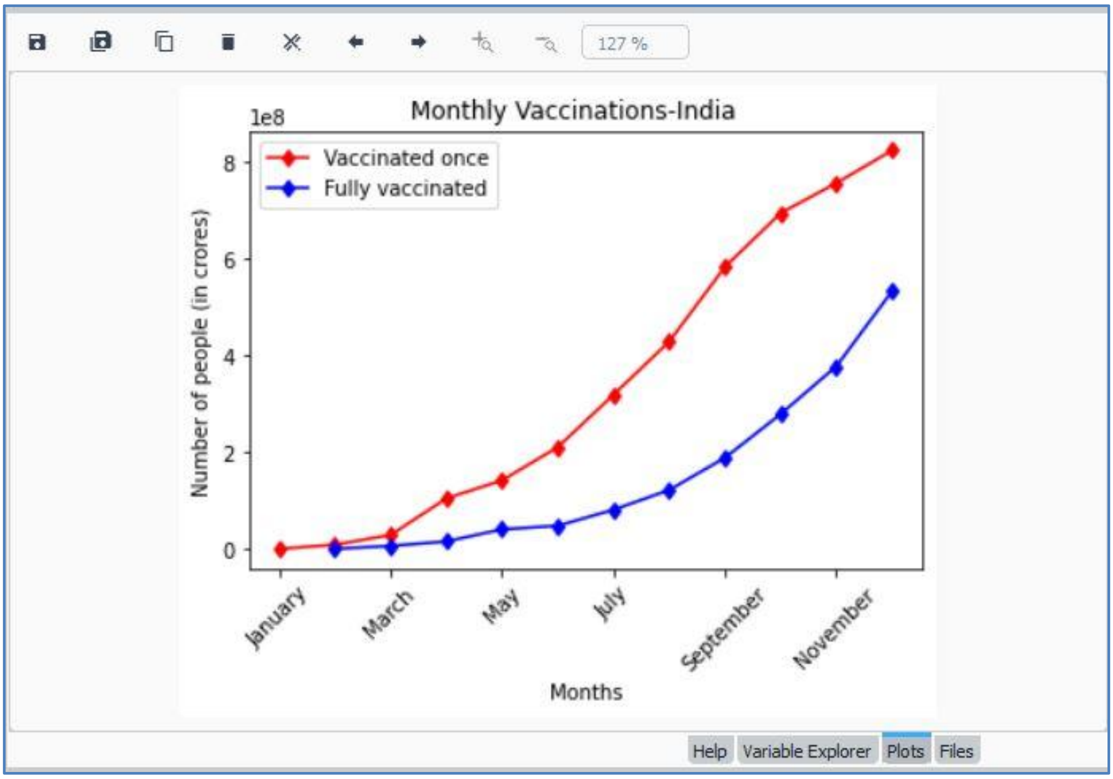


The user can infer various data from the graphs. For example:

From this graph, it can be inferred that the number of doses that have been administered by India is much greater than the number of doses that have been administered by Brazil, Russia, UK and U.S.A.

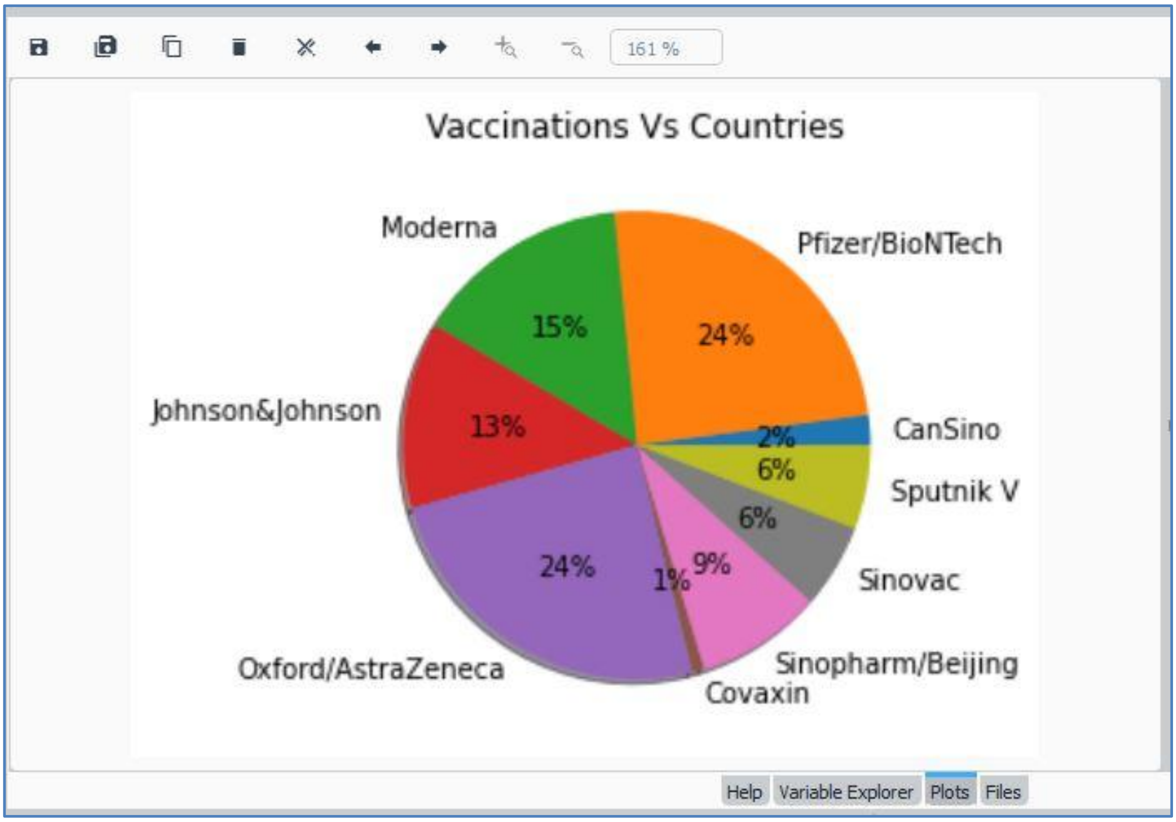
This shows us that India has vaccinated a large number of people in a short period of time.

4.2. Trend of Monthly Vaccination in India



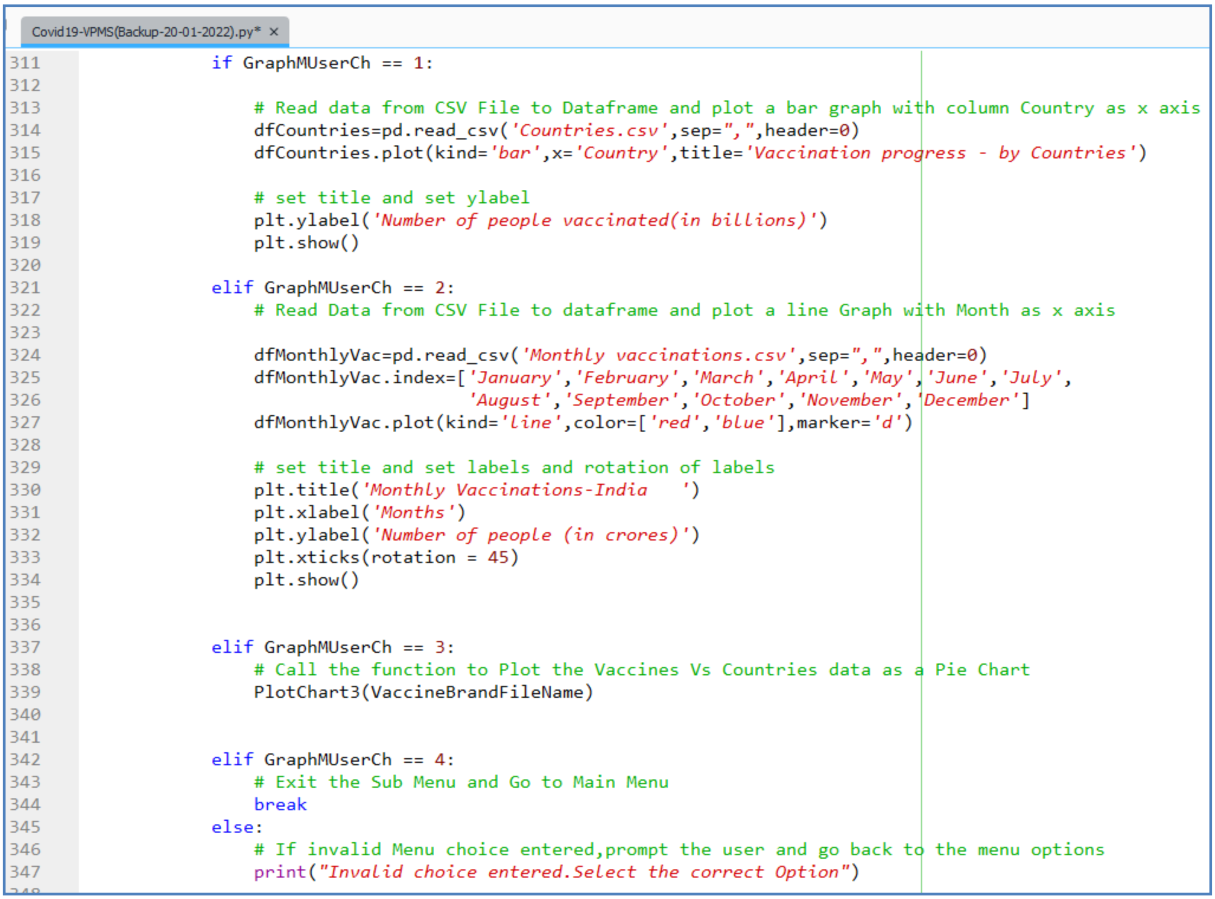
From this graph, the month-wise progress of vaccinations (first and second dose) can be viewed. The decline in the rate of people vaccinated once, between the months of October to December, (visible from the gradual slope) indicates a slow-down in the vaccination rate in these months.

4.3 Use of Major Vaccine Brands across Countries

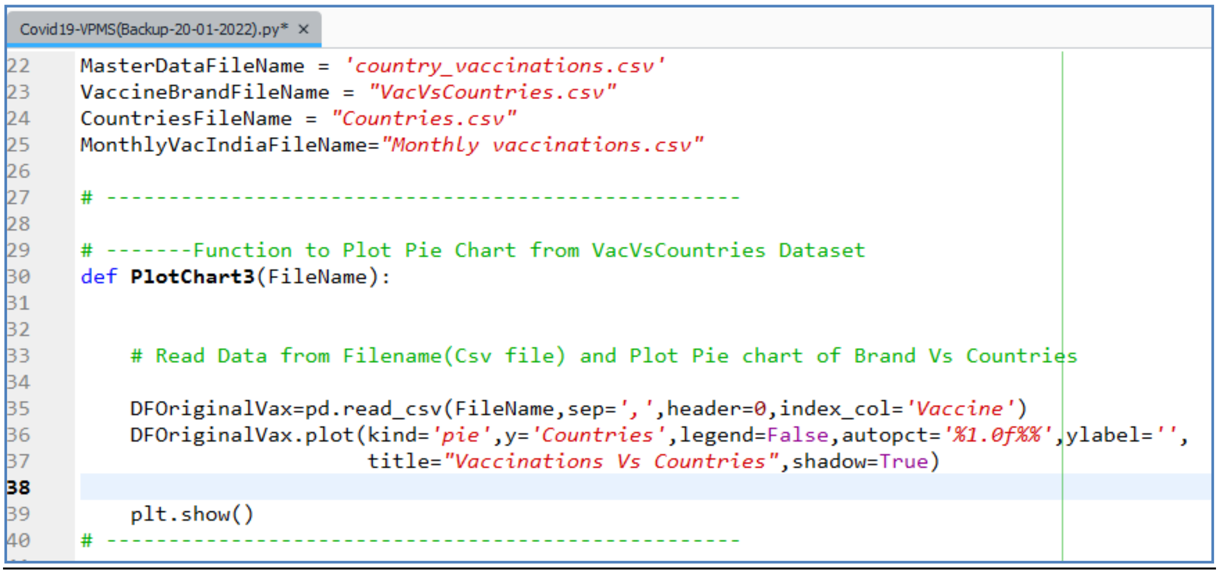


From this graph, the vaccine brands being used by all the countries and the percentage of countries using each vaccine brand can be viewed. The vaccine brands being used most commonly are Oxford/Astra Zeneca and Moderna, which are being used by nearly 50% of the population. The analyst can decide to recommend the purchase of certain vaccinations for their country. (This is known as vaccine donation.)

Code to Display Data Visualisations

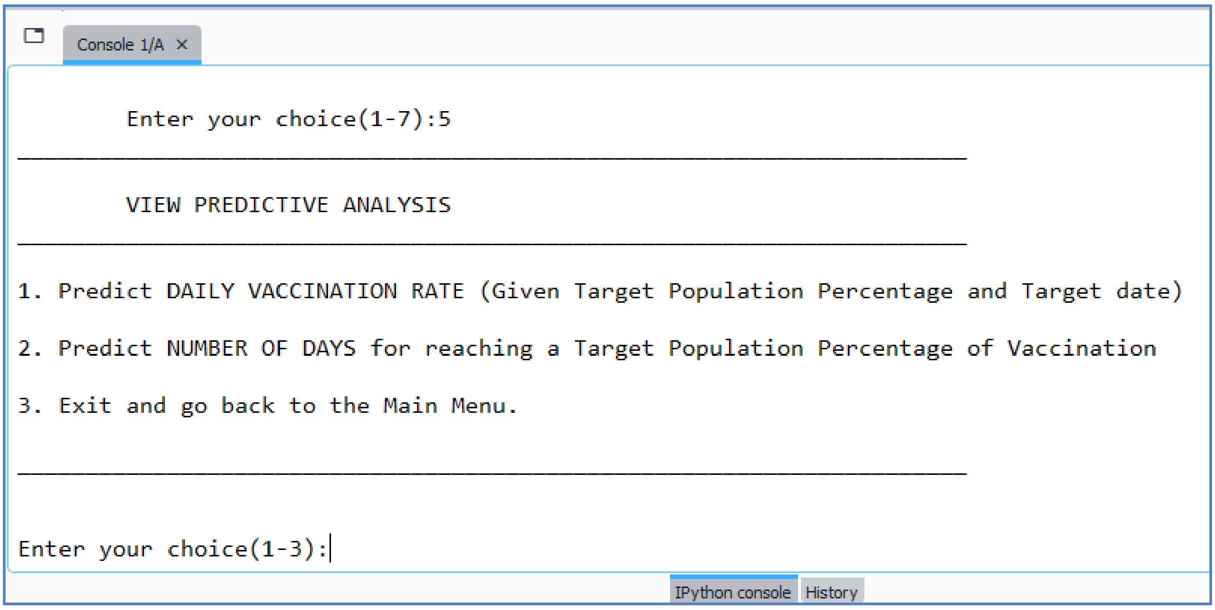


```python
        if GraphMUserCh == 1:

            # Read data from CSV File to Dataframe and plot a bar graph with column Country as x axis
            dfCountries=pd.read_csv('Countries.csv',sep=",",header=0)
            dfCountries.plot(kind='bar',x='Country',title='Vaccination progress - by Countries')

            # set title and set ylabel
            plt.ylabel('Number of people vaccinated(in billions)')
            plt.show()

        elif GraphMUserCh == 2:
            # Read Data from CSV File to dataframe and plot a line Graph with Month as x axis

            dfMonthlyVac=pd.read_csv('Monthly vaccinations.csv',sep=",",header=0)
            dfMonthlyVac.index=['January','February','March','April','May','June','July',
                                'August','September','October','November','December']
            dfMonthlyVac.plot(kind='line',color=['red','blue'],marker='d')

            # set title and set labels and rotation of labels
            plt.title('Monthly Vaccinations-India   ')
            plt.xlabel('Months')
            plt.ylabel('Number of people (in crores)')
            plt.xticks(rotation = 45)
            plt.show()


        elif GraphMUserCh == 3:
            # Call the function to Plot the Vaccines Vs Countries data as a Pie Chart
            PlotChart3(VaccineBrandFileName)


        elif GraphMUserCh == 4:
            # Exit the Sub Menu and Go to Main Menu
            break
        else:
            # If invalid Menu choice entered,prompt the user and go back to the menu options
            print("Invalid choice entered.Select the correct Option")
```



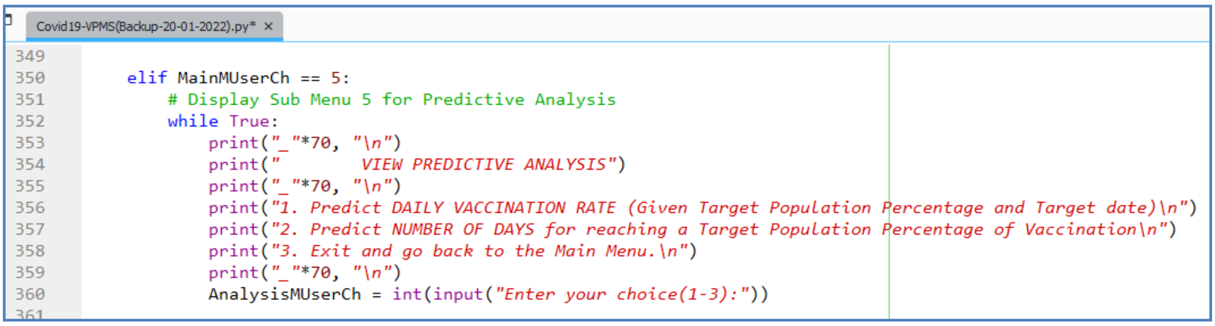
```python
MasterDataFileName = 'country_vaccinations.csv'
VaccineBrandFileName = "VacVsCountries.csv"
CountriesFileName = "Countries.csv"
MonthlyVacIndiaFileName="Monthly vaccinations.csv"

# --------------------------------------------------

# -------Function to Plot Pie Chart from VacVsCountries Dataset
def PlotChart3(FileName):


    # Read Data from Filename(Csv file) and Plot Pie chart of Brand Vs Countries

    DFOriginalVax=pd.read_csv(FileName,sep=',',header=0,index_col='Vaccine')
    DFOriginalVax.plot(kind='pie',y='Countries',legend=False,autopct='%1.0f%%',ylabel='',
                       title="Vaccinations Vs Countries",shadow=True)

    plt.show()
# --------------------------------------------------
```

5. Predictive Analysis Menu

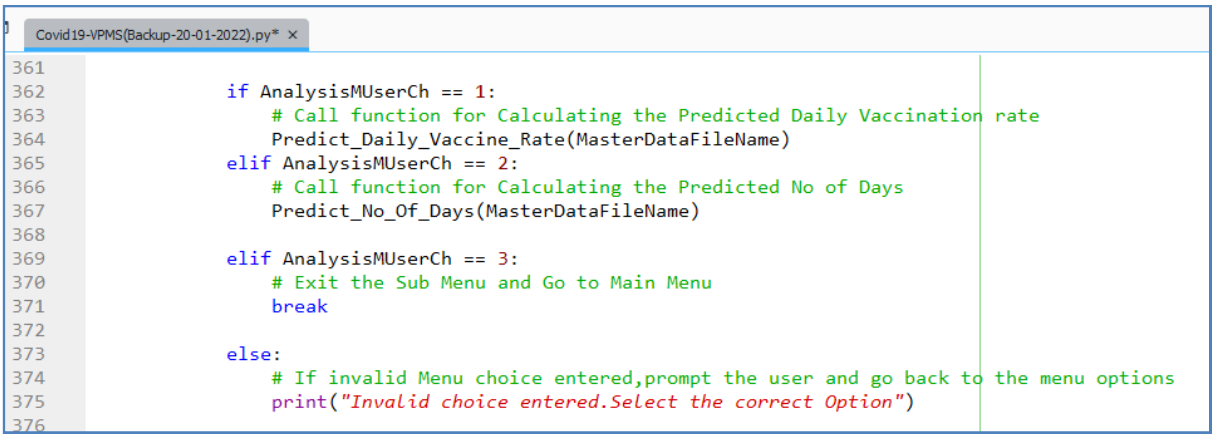


```
        Enter your choice(1-7):5
______________________________________________________________________

        VIEW PREDICTIVE ANALYSIS
______________________________________________________________________

1. Predict DAILY VACCINATION RATE (Given Target Population Percentage and Target date)

2. Predict NUMBER OF DAYS for reaching a Target Population Percentage of Vaccination

3. Exit and go back to the Main Menu.

______________________________________________________________________

Enter your choice(1-3):
```

Predictive Analysis Menu: Code

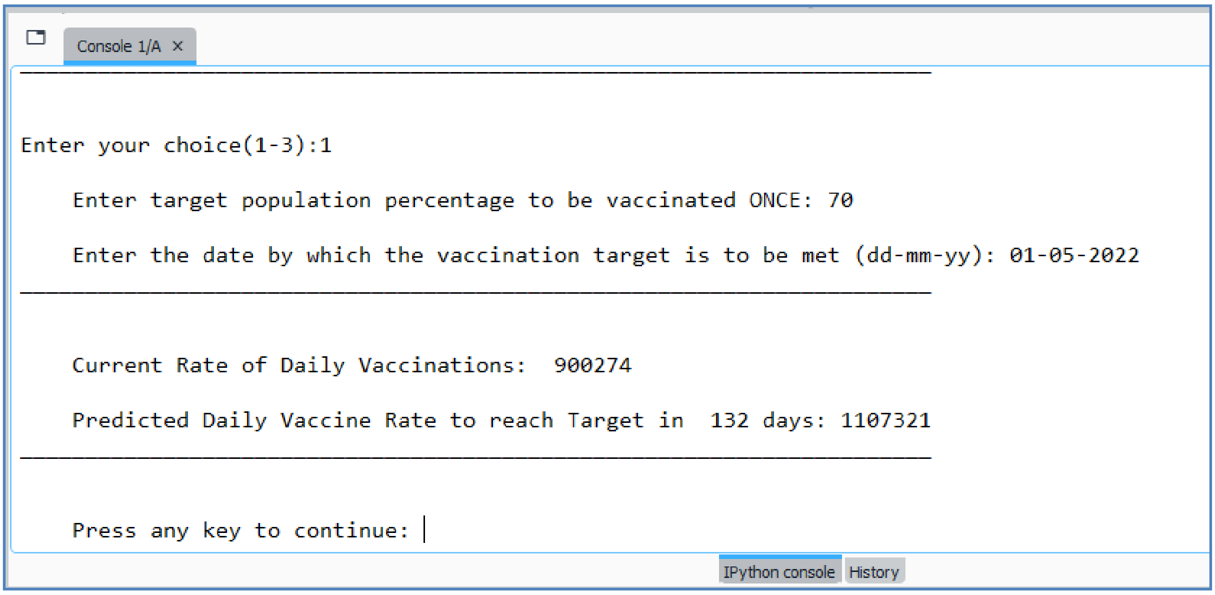


```python

    elif MainMUserCh == 5:
        # Display Sub Menu 5 for Predictive Analysis
        while True:
            print("_"*70, "\n")
            print("        VIEW PREDICTIVE ANALYSIS")
            print("_"*70, "\n")
            print("1. Predict DAILY VACCINATION RATE (Given Target Population Percentage and Target date)\n")
            print("2. Predict NUMBER OF DAYS for reaching a Target Population Percentage of Vaccination\n")
            print("3. Exit and go back to the Main Menu.\n")
            print("_"*70, "\n")
            AnalysisMUserCh = int(input("Enter your choice(1-3):"))
```



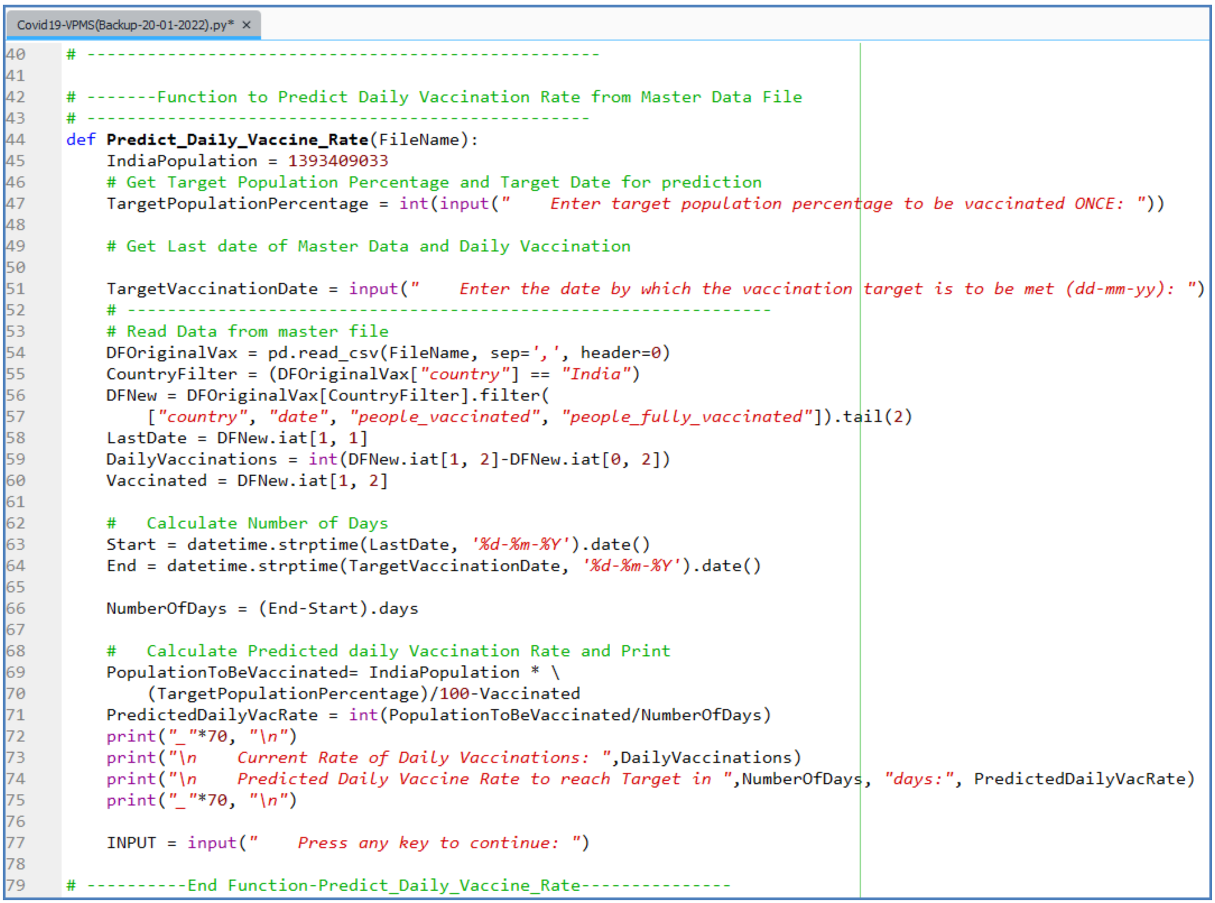
```python

            if AnalysisMUserCh == 1:
                # Call function for Calculating the Predicted Daily Vaccination rate
                Predict_Daily_Vaccine_Rate(MasterDataFileName)
            elif AnalysisMUserCh == 2:
                # Call function for Calculating the Predicted No of Days
                Predict_No_Of_Days(MasterDataFileName)

            elif AnalysisMUserCh == 3:
                # Exit the Sub Menu and Go to Main Menu
                break

            else:
                # If invalid Menu choice entered,prompt the user and go back to the menu options
                print("Invalid choice entered.Select the correct Option")
```

Output



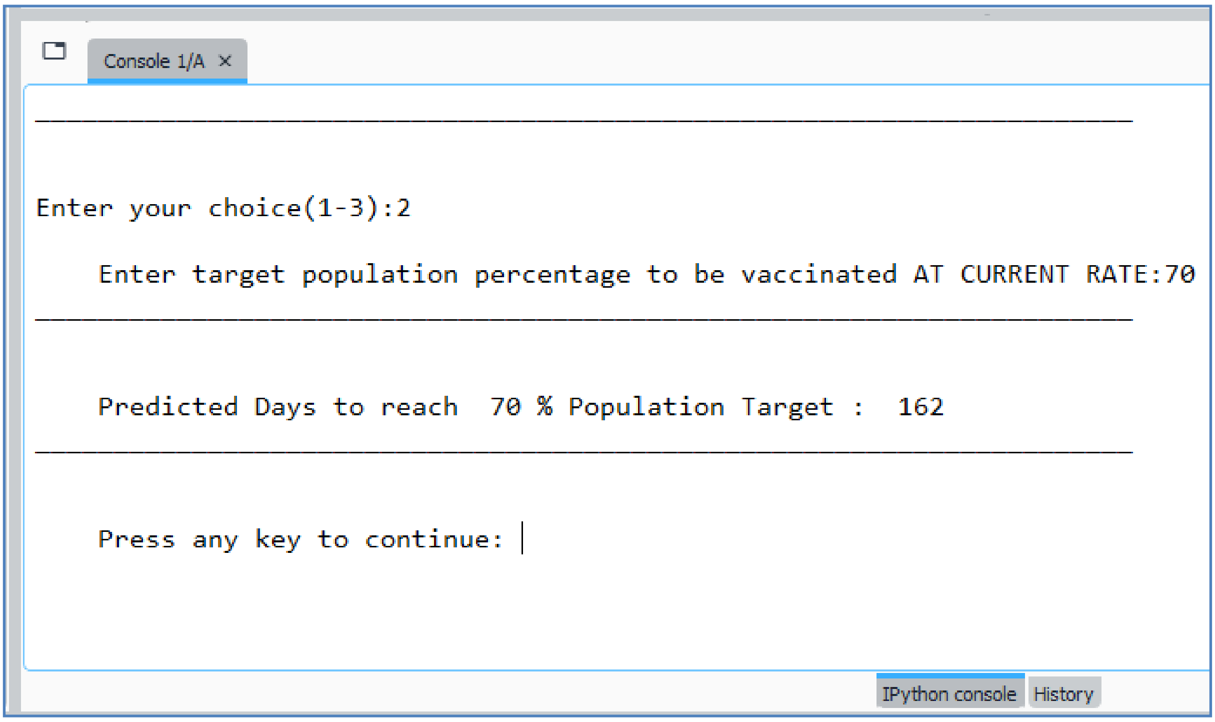
```
______________________________________________________________________

Enter your choice(1-3):1

    Enter target population percentage to be vaccinated ONCE: 70

    Enter the date by which the vaccination target is to be met (dd-mm-yy): 01-05-2022
______________________________________________________________________

    Current Rate of Daily Vaccinations:  900274

    Predicted Daily Vaccine Rate to reach Target in  132 days: 1107321
______________________________________________________________________

    Press any key to continue: 
```

Predictive Analysis –Daily Vaccination Rate: Code



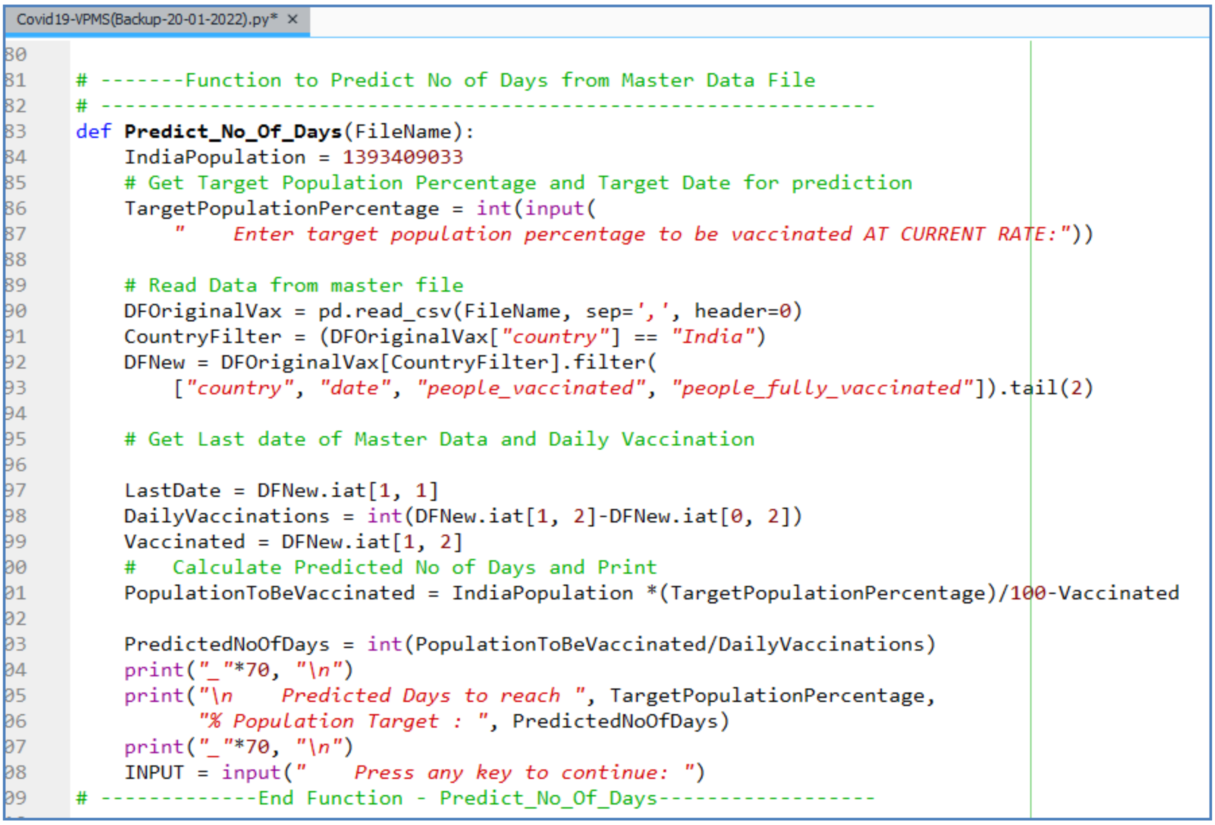
```python
# --------------------------------------------------

# -------Function to Predict Daily Vaccination Rate from Master Data File
# -------------------------------------------------
def Predict_Daily_Vaccine_Rate(FileName):
    IndiaPopulation = 1393409033
    # Get Target Population Percentage and Target Date for prediction
    TargetPopulationPercentage = int(input("    Enter target population percentage to be vaccinated ONCE: "))

    # Get Last date of Master Data and Daily Vaccination

    TargetVaccinationDate = input("    Enter the date by which the vaccination target is to be met (dd-mm-yy): ")
    # ---------------------------------------------------------------
    # Read Data from master file
    DFOriginalVax = pd.read_csv(FileName, sep=',', header=0)
    CountryFilter = (DFOriginalVax["country"] == "India")
    DFNew = DFOriginalVax[CountryFilter].filter(
        ["country", "date", "people_vaccinated", "people_fully_vaccinated"]).tail(2)
    LastDate = DFNew.iat[1, 1]
    DailyVaccinations = int(DFNew.iat[1, 2]-DFNew.iat[0, 2])
    Vaccinated = DFNew.iat[1, 2]

    #   Calculate Number of Days
    Start = datetime.strptime(LastDate, '%d-%m-%Y').date()
    End = datetime.strptime(TargetVaccinationDate, '%d-%m-%Y').date()

    NumberOfDays = (End-Start).days

    #   Calculate Predicted daily Vaccination Rate and Print
    PopulationToBeVaccinated= IndiaPopulation * \
        (TargetPopulationPercentage)/100-Vaccinated
    PredictedDailyVacRate = int(PopulationToBeVaccinated/NumberOfDays)
    print("_"*70, "\n")
    print("\n    Current Rate of Daily Vaccinations: ",DailyVaccinations)
    print("\n    Predicted Daily Vaccine Rate to reach Target in ",NumberOfDays, "days:", PredictedDailyVacRate)
    print("_"*70, "\n")

    INPUT = input("    Press any key to continue: ")

# ----------End Function-Predict_Daily_Vaccine_Rate---------------
```

Output



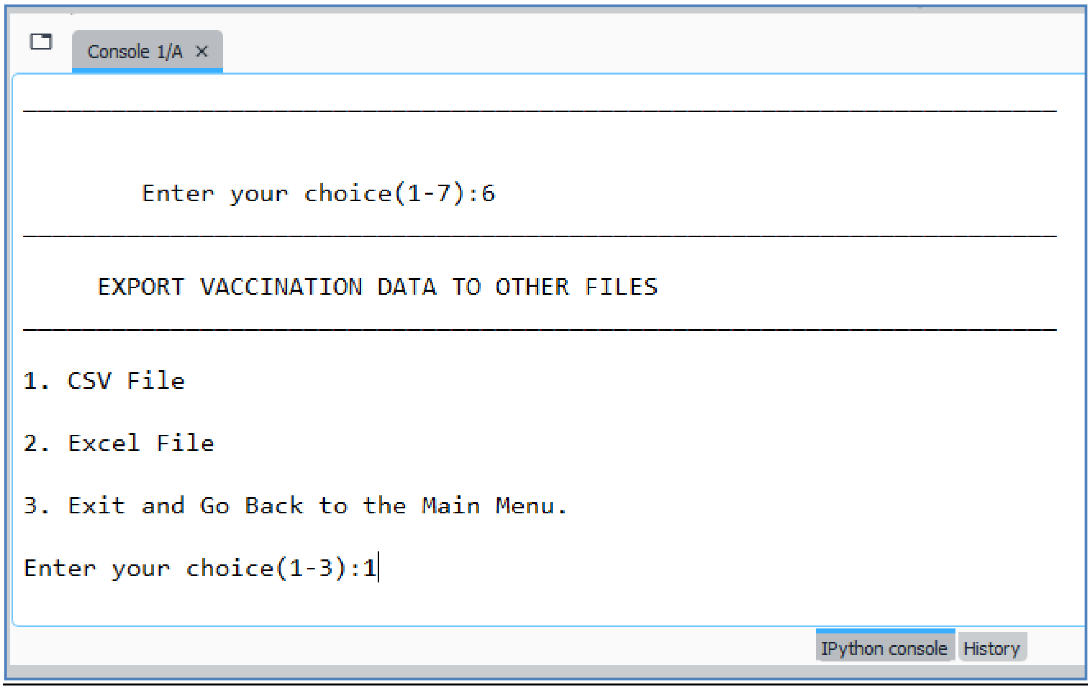
```
______________________________________________________________________

Enter your choice(1-3):2

    Enter target population percentage to be vaccinated AT CURRENT RATE:70
______________________________________________________________________

    Predicted Days to reach  70 % Population Target :  162
______________________________________________________________________

    Press any key to continue: 
```

Predictive Analysis –No of Days: Code



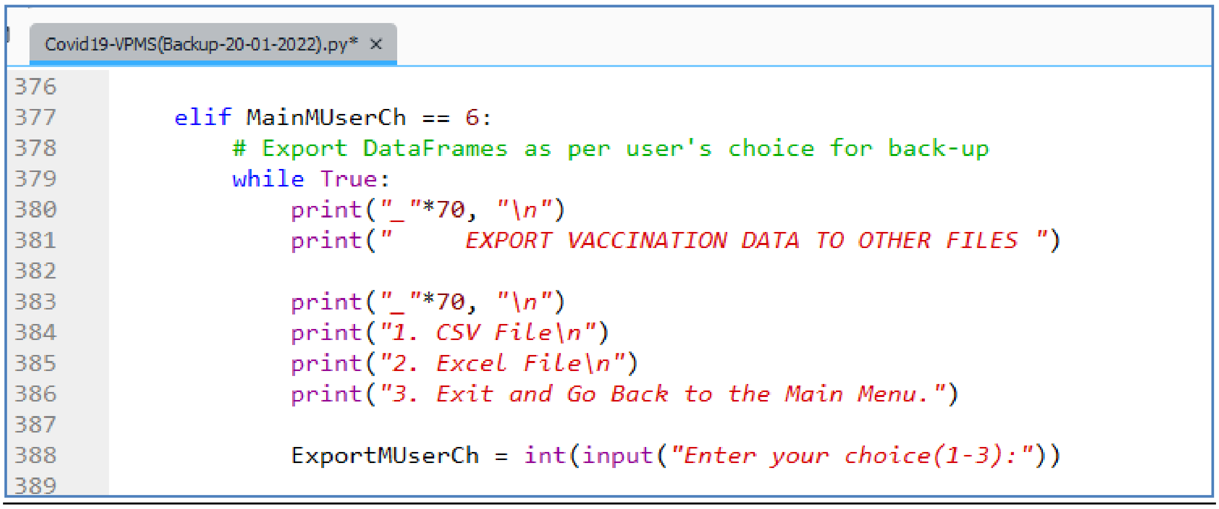
```python

# -------Function to Predict No of Days from Master Data File
# -----------------------------------------------------------------
def Predict_No_Of_Days(FileName):
    IndiaPopulation = 1393409033
    # Get Target Population Percentage and Target Date for prediction
    TargetPopulationPercentage = int(input(
        "    Enter target population percentage to be vaccinated AT CURRENT RATE:"))

    # Read Data from master file
    DFOriginalVax = pd.read_csv(FileName, sep=',', header=0)
    CountryFilter = (DFOriginalVax["country"] == "India")
    DFNew = DFOriginalVax[CountryFilter].filter(
        ["country", "date", "people_vaccinated", "people_fully_vaccinated"]).tail(2)

    # Get Last date of Master Data and Daily Vaccination

    LastDate = DFNew.iat[1, 1]
    DailyVaccinations = int(DFNew.iat[1, 2]-DFNew.iat[0, 2])
    Vaccinated = DFNew.iat[1, 2]
    #   Calculate Predicted No of Days and Print
    PopulationToBeVaccinated = IndiaPopulation *(TargetPopulationPercentage)/100-Vaccinated

    PredictedNoOfDays = int(PopulationToBeVaccinated/DailyVaccinations)
    print("_"*70, "\n")
    print("\n    Predicted Days to reach ", TargetPopulationPercentage,
          "% Population Target : ", PredictedNoOfDays)
    print("_"*70, "\n")
    INPUT = input("    Press any key to continue: ")
# ------------End Function - Predict_No_Of_Days-----------------
```

6. Export Data Sub-Menu

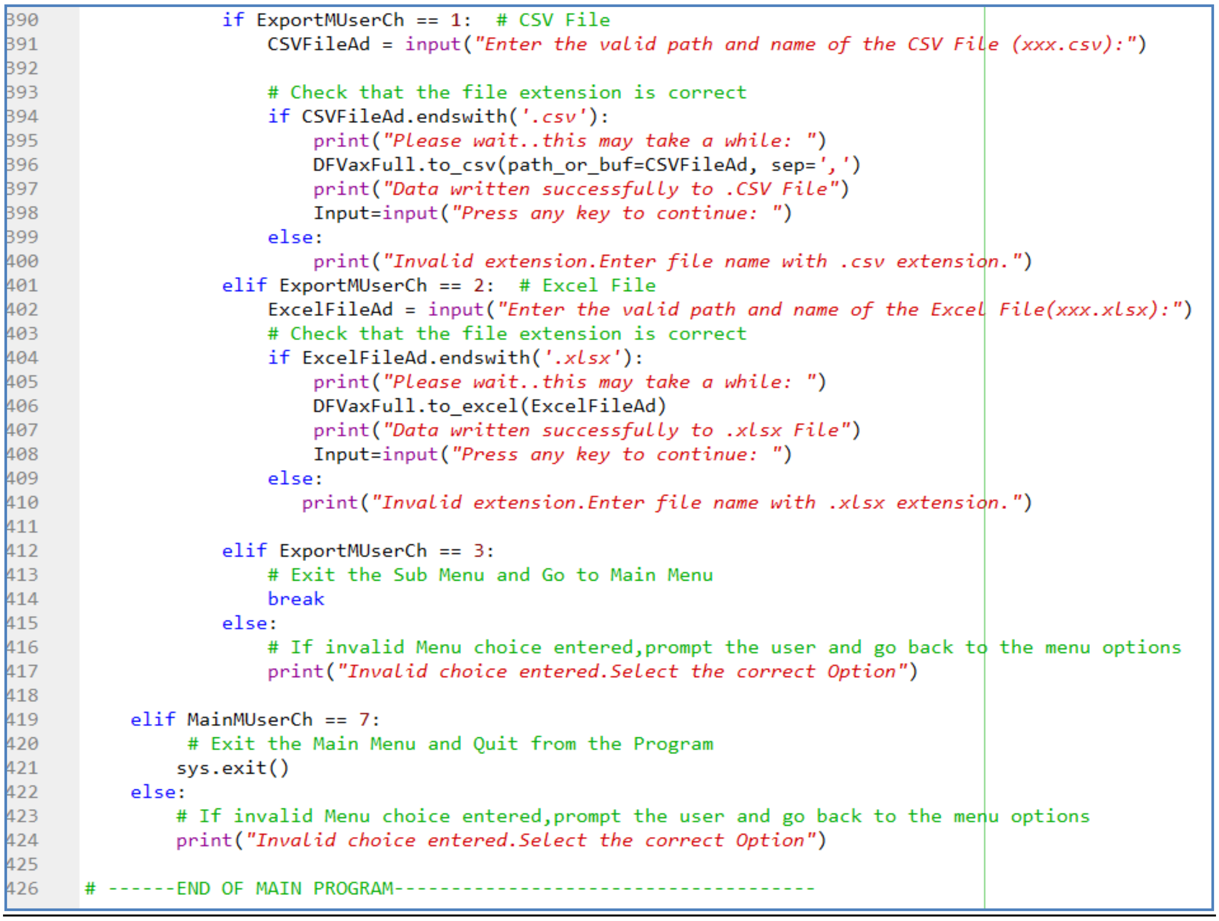


```
______________________________________________________________________


        Enter your choice(1-7):6
______________________________________________________________________

     EXPORT VACCINATION DATA TO OTHER FILES
______________________________________________________________________

1. CSV File

2. Excel File

3. Exit and Go Back to the Main Menu.

Enter your choice(1-3):1
```

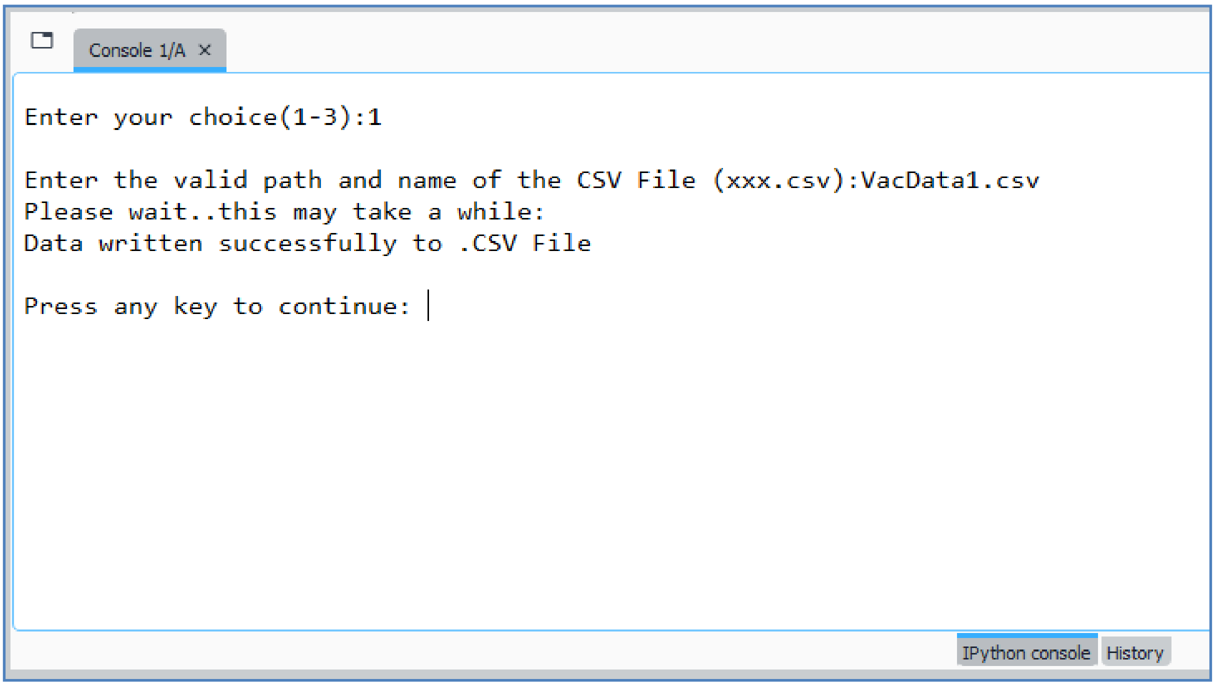
Export Data Sub-Menu-Code



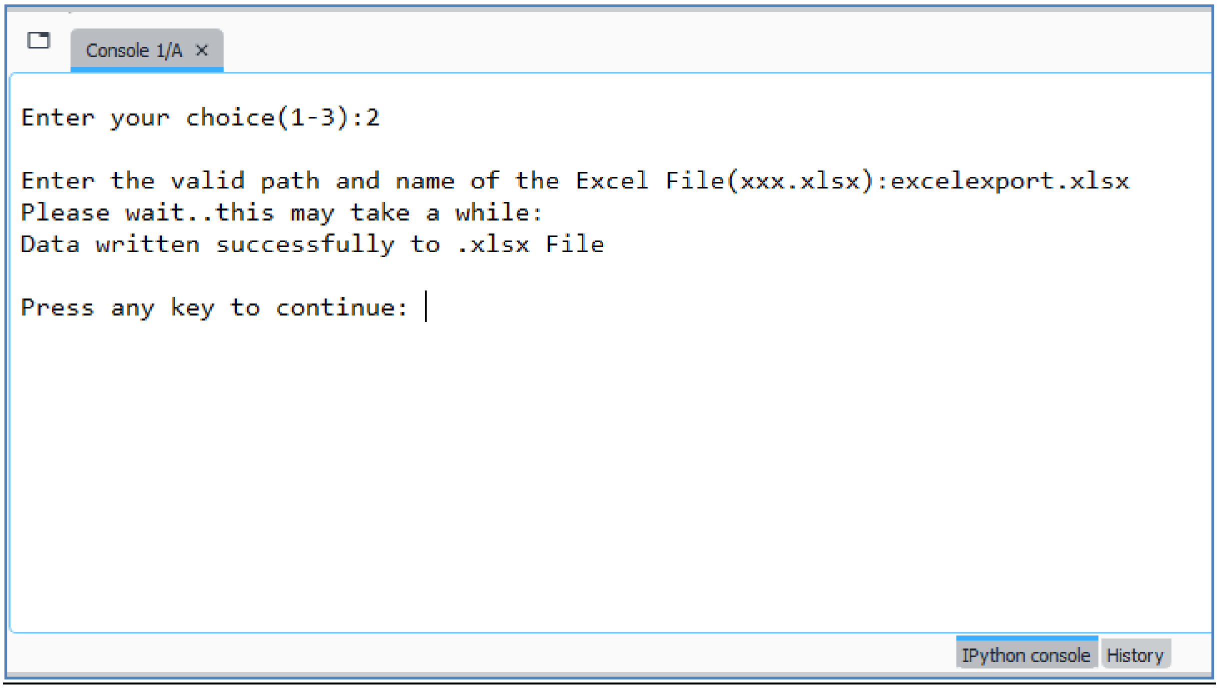
```python

    elif MainMUserCh == 6:
        # Export DataFrames as per user's choice for back-up
        while True:
            print("_"*70, "\n")
            print("     EXPORT VACCINATION DATA TO OTHER FILES ")

            print("_"*70, "\n")
            print("1. CSV File\n")
            print("2. Excel File\n")
            print("3. Exit and Go Back to the Main Menu.")

            ExportMUserCh = int(input("Enter your choice(1-3):"))

```



```
            if ExportMUserCh == 1:  # CSV File
                CSVFileAd = input("Enter the valid path and name of the CSV File (xxx.csv):")

                # Check that the file extension is correct
                if CSVFileAd.endswith('.csv'):
                    print("Please wait..this may take a while: ")
                    DFVaxFull.to_csv(path_or_buf=CSVFileAd, sep=',')
                    print("Data written successfully to .CSV File")
                    Input=input("Press any key to continue: ")
                else:
                    print("Invalid extension.Enter file name with .csv extension.")
            elif ExportMUserCh == 2:  # Excel File
                ExcelFileAd = input("Enter the valid path and name of the Excel File(xxx.xlsx):")
                # Check that the file extension is correct
                if ExcelFileAd.endswith('.xlsx'):
                    print("Please wait..this may take a while: ")
                    DFVaxFull.to_excel(ExcelFileAd)
                    print("Data written successfully to .xlsx File")
                    Input=input("Press any key to continue: ")
                else:
                   print("Invalid extension.Enter file name with .xlsx extension.")

            elif ExportMUserCh == 3:
                # Exit the Sub Menu and Go to Main Menu
                break
            else:
                # If invalid Menu choice entered,prompt the user and go back to the menu options
                print("Invalid choice entered.Select the correct Option")

    elif MainMUserCh == 7:
         # Exit the Main Menu and Quit from the Program
        sys.exit()
    else:
        # If invalid Menu choice entered,prompt the user and go back to the menu options
        print("Invalid choice entered.Select the correct Option")

# ------END OF MAIN PROGRAM------------------------------------
```

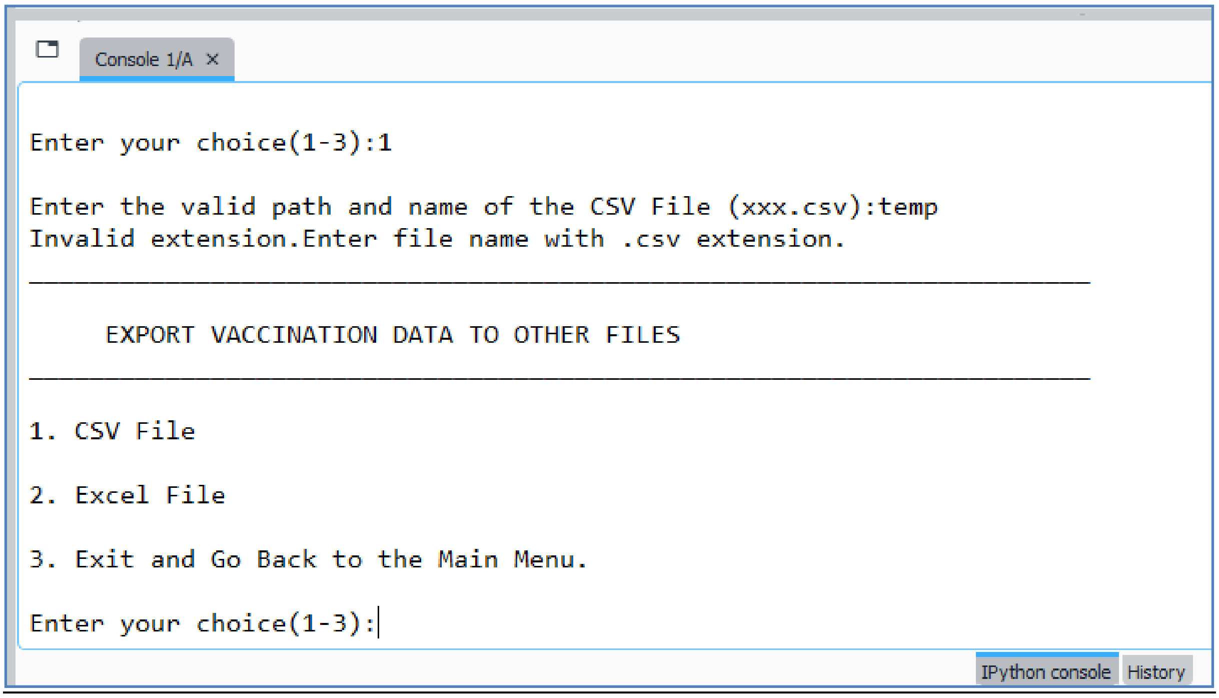


```
Enter your choice(1-3):1

Enter the valid path and name of the CSV File (xxx.csv):VacData1.csv
Please wait..this may take a while:
Data written successfully to .CSV File

Press any key to continue:
```



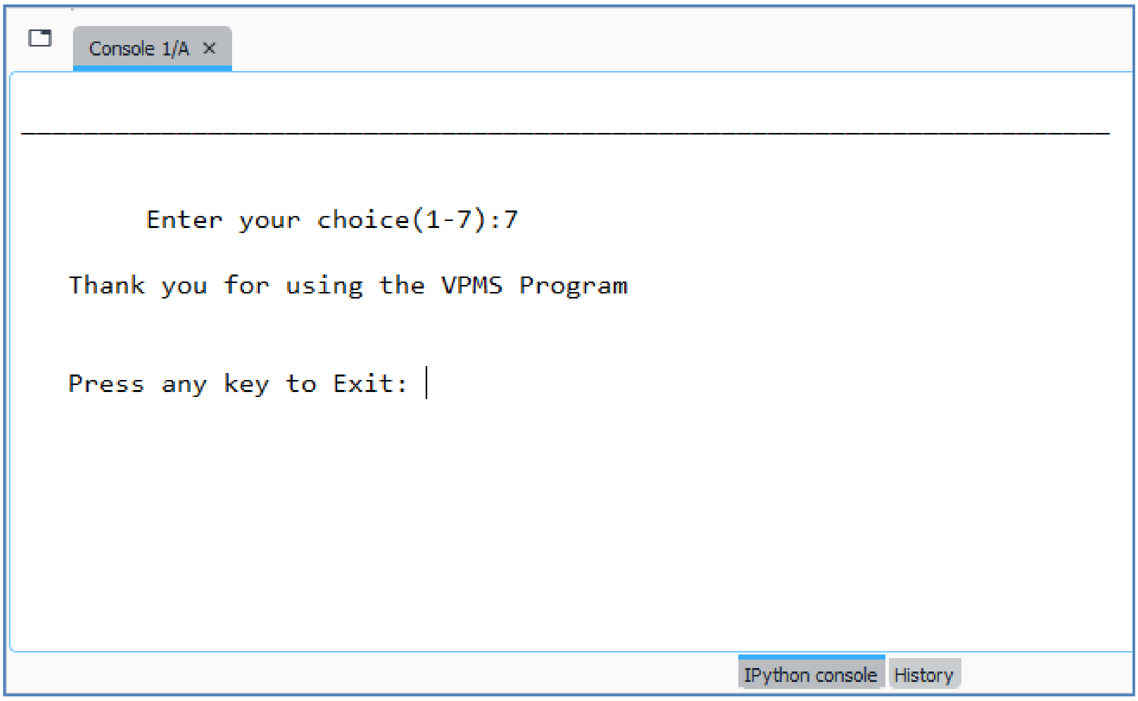
```
Enter your choice(1-3):2

Enter the valid path and name of the Excel File(xxx.xlsx):excelexport.xlsx
Please wait..this may take a while:
Data written successfully to .xlsx File

Press any key to continue:
```

Error Handling : Invalid file extension

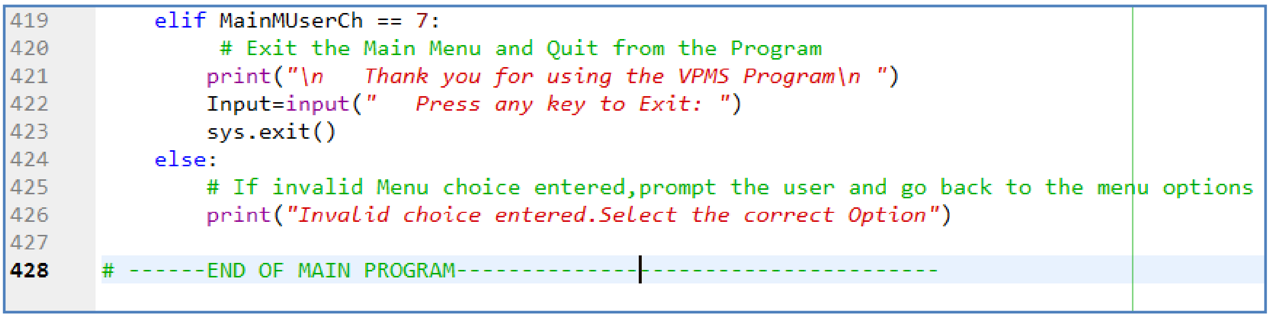


```
Enter your choice(1-3):1

Enter the valid path and name of the CSV File (xxx.csv):temp
Invalid extension.Enter file name with .csv extension.
______________________________________________________________________

     EXPORT VACCINATION DATA TO OTHER FILES
______________________________________________________________________

1. CSV File

2. Excel File

3. Exit and Go Back to the Main Menu.

Enter your choice(1-3):
```

7. Exit



```
__________________________________________________________________________________

      Enter your choice(1-7):7

  Thank you for using the VPMS Program


  Press any key to Exit:
```



```
    elif MainMUserCh == 7:
         # Exit the Main Menu and Quit from the Program
        print("\n    Thank you for using the VPMS Program\n ")
        Input=input("    Press any key to Exit: ")
        sys.exit()
    else:
        # If invalid Menu choice entered,prompt the user and go back to the menu options
        print("Invalid choice entered.Select the correct Option")

# ------END OF MAIN PROGRAM-------------------------------------
```

6. User Manual

6.1 How to install Software:

Hardware Requirement-

- Latest Intel processor-based PC/Laptop at Client/Server end.
- 1 GB RAM and 4GB HDD space (for Database) is desirable.
- Standard I/O devices like Keyboard and Mouse etc.

Software Requirement-

- Windows 2000/XP OS is desirable.
- Python Version 3.0 or higher should be installed with Pandas and MatPlotLib Library
- Spyder 3(Python 3.8) IDE.

Database / Software Installation

Installation of System Software:

Step 1: **Python Installation:** To execute the program for analysis of Covid-19 Vaccination data, we must install Pandas. It is a python library used for data analysis and can be installed from www.python.org. The Pandas installation is dependent on the operating system. For Windows, we can use the command "pip install pandas" .

Step 2: **Spyder IDE Installation:** Spyder is a free and open source scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It features a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package.
Install Spyder from https://www.spyder-ide.org/

Installation of VPMS Software:

The software project is distributed as a folder ‘VPMS’ containing a set of 6 files:1 python (.py) file and 5 datasets (.CSV files.).

1. Covid19-VPMS.py
2. Countries.csv
3. country_vaccinations.csv
4. Monthly vaccinations.csv
5. UserDataFile.csv
6. VacVsCountries.csv

Note: The CSV files must be saved on the user's computer in the same folder as the .py file.

To install the software, follow the steps given below:

Step 1: Create a folder VPMS on any drive
Step 2: Copy the source code (.py) and data files in the distribution
Step 3: Open Spyder IDE by running the SPYDER.exe file
Step 4: Open the Covid-19 VPMS Python file using the File→Open option
Step 4: Choose Run→ Run option to start the program.

Follow the prompts on the output console to operate the program.

7. References

In order to work on this project titled –**COVID-19 Vaccination Progress Management System,** I have referred to the following books and websites during the various phases of development of the project.

(1) http://kaggle.com/gpreda/covid-world-vaccination-progress

(2) Informatics Practices for class XII -by Sumita Arora

(3) https://www.geeksforgeeks.org/python-sys-module/

(4) www.python.org

Other than the above-mentioned books and websites, the suggestions and supervision of my teacher and my class experience also helped me to develop this software project.

-----xxxxx-----