**Covid Analysis Dashboard Using Python**

*An Industry Oriented Mini Project report*

*Submitted in partial fulfillment of requirements*

*For the award of degree of*

**BACHELOR OF TECHNOLOGY**

**In**

**INFORMATION TECHNOLOGY**

**By**

**P.Savanth 19131A1280**

**R.vivek 19131A1294**

**T.Mohana Bhavani 19131A12A9**

**V.Vyshnavi 19131A12B5**



**GAYATRI VIDHYA PARISHAD**

**College of Engineering**

**(Autonomous)**

Under the esteemed guidance of

I.V.S.Venugopal

Department of Information Technology

**GAYATRI VIDYA PARISHAD COLLEGE OF**

**ENGINEERING(AUTONOMOUS)**

(Affiliated to JNTU-K)



**GAYATRI VIDYA PARISHAD**

**College of Engineering**

**(Autonomous**)

**CERTIFICATE**

This report “**Covid data analysis dashboard using python**” is a bonafide record of the industry oriented mini project work submitted by

**P.Savanth 19131A1280**

**R.Vivek 19131A1294**

**T.Mohana Bhavani 19131A12A9**

**V.Vyshnavi 19131A12B5**

in their V semester and fulfilment of the requirements for the Award of Degree of

**BACHELOR OF TECHNOLOGY**

**In**

**INFORMATION TECHNOLOGY**

**DECLARATION**

we, hereby declare that this Project Work entitled “ **Covid data analysis dashboard using python”** is a bonafide work done by us and submitted to the Department of **Information Technology, G.V.P. College of Engineering (Autonomous), Visakhapatnam,** in fulfilment for the award of the degree of B.Tech is our own work and it is not submitted to any other university or has been published any time before.

PLACE: **Visakhapatnam**

DATE:

**ACKNOWLEDGEMENT**

We would like to express our deep sense of gratitude to our esteemed institute Gayatri Vidya

Parishad College of Engineering (Autonomous), which has provided us an opportunity to fulfil our cherished desire.

We express our sincere thanks to our Principal Prof. **A. B. KOTESWARA RAO** for this encouragement to us during the course of the project, giving us a chance to explore and learn new technologies in the form of mini projects.

We express our deep sense of Gratitude to Prof. **Dr. K. B. MADHURI** head of the department, Department of Information Technology.

We express our profound Gratitude and our deep indebtedness to our guide **I.V.S venugopal** whose valuable suggestions, guidance and comprehensive assistance helped us a lot in realizing my present project.

We also thank **Mr. Srinu Bevara**, Assistant Professor Project Co-ordinator, for guiding us throughout the project and leading us in completing our project efficiently.

We would also like to thank all the members of the teaching and non-teaching staff of the Information Technology department for all their support in completion of our project.

**P.Savanth 19131A1280**

**R.vivek 19131A1294**

**T.Mohana Bhavani 19131A12A9**

**V.Vyshnavi 19131A12B5**

**Abstract**

A trillion MB’s of data got generated every second,out of which there is useful Data and unuseful data.Inorder to get our required information from this huge data ,is difficult with normal techniques .But using python,we can get data in simple way and we can also represent data in graphical forms to make us understand very easily.

And observing the patterns hidden in the data set is difficult normally but we can observe easily using the graphical forms like(bar chart,pie chart etc..).

**Index**

**­­**

**1.Introduction**

1.1 Objective.

1.2 Purpose.

**2. SOFTWARE REQUIREMENT SPECIFICATION**

2.1 Functional Requirements.

2.2 Non-Functional Requirements.

2.3 System Requirements.

2.4 Software Requirements.

2.5 Hardware Requirements.\

**3. Project Description.**

3.1 Problem Definition.

3.2 Project Overview.

**4.System Design.**

4.1 Software Description.

4.1.1 Python IDE.

4.1.2 Numpy.

4.1.3 pandas.

4.1.4 streamlit.

4.1.5 mysql.connector.

**5.Implementation.**

5.1 Importing modules of python.

5.2 Collecting and cleaning of dataset.

5.3 Overall analysis.

5.4 Country wise analysis.

5.5 Admin authentication.

5.6 Data Updation.

**6.Testing.**

6.1 Output

**7.Development.**

7.1 Code.

**\**

**1.INTRODUCTION**

* 1. **Objective**

Data analysis is defined as the process of evaluating data using analytical and logical thinking to study each component of research data. This analysis is just one of the many steps that must be completed when performing a search experiment.Data analysisis the organization and arrangement of data, in order to produce and highlight data in the form of information used to answer specific questions. There are many different ways to collect data, so data collection depends on the type of research an individual conducts. One or more of the following methods can be used:

(a) Observations; they rely on observing something or someone

(b) Interviews based on talking to and interviewing people.

Data analysis includes data mining, text analysis, business intelligence, and general data visualization.

**1.2 Purpose**

# The main purpose of data analysis is to understand the data and gain useful information so that the derived knowledge can help in decision making. Even though data is abundant nowadays, it’s available in different forms and scattered over various sources. Data analysis helps to clean and transform all this data into a consistent form so it can be effectively studied. **Data analysis is important in research, it makes studying data a lot simpler and more accurate. It helps the researchers straightforwardly interpret the data so that researchers don’t leave anything out that could help them derive insights from it.**

**2. SOFTWARE REQUIREMENT SPECIFICATION**

**2.1 FUNCTIONAL REQUIREMENTS**

**Functional requirements** in an Software Requirement Specification indicate what a software must do and how it must function; they are product features that focus on user needs. And it is also useful for defining the needs of the software.

***Input:*** user need to enter the name of the country of which he wants the analysis.

***Output:*** User gets the analysis of the country according to his input.

**2.2 NON-FUNCTIONAL REQUIREMENTS**

***Non-Functional Requirements*** (NFR) specifies the quality attribute of a software . They judge the software based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software.

***Reliability***: Reliability depends on the evaluation result of the system by correct identification of the words entered by the user in the query. ***Easy to use***: The software is simple, user-friendly so any one can use this system without any difficulties.***Processing Time***: Our model processes the answer in a couple of seconds.

**2.3 SYSTEM REQUIREMENTS**

System requirements are the required specifications a device must have in order to use certain hardware or software. For example, a computer may require a specific I/O port to work with a peripheral device. A smartphone may need a specific operating system to run a particular app.

Typical system requirements for a software program include: Operating system, Programming Interface, Web Interface and different kinds of packages included.

System requirements listed for a hardware device may include: Processor Specifications, Memory Specifications etc

**2.4 SOFTWARE** **REQUIREMENTS**

The software requirements are description of features and functionalities of the target system.

***Operating System*:** WINDOWS 10***Programming Interface*:** Python 3.9***Modules:*** Numpy, pandas, streamlit.

**2.5 HARDWARE REQUIREMENTS**

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware requirements. It is generally associated with Hardware Compatibility List

***Processor Specification:*** i5 Processor Based Computer***Memory Specifications:*** 4GB-RAM, 320GB Hard Disk Monitor

**3.PROJECT DESCRIPTION**

**3.1 PROBLEM DEFINITION**

The outbreak of coronavirus disease (COVID) has been declared a Public Health Emergency of International Concern (PHEIC) and the virus has now spread to many countries and territories. While a lot is still unknown about the virus that causes COVID .People who are eagerly searching the statistics of covid in different countries and to observe different patterns hidden in these statistics ,can be done through out web application .And our dashboard also makes the users to understand the data and statistics easily through graphical representations.

**3.2 PROJECT OVERVIEW**

Our project is to show the overall and country wise covid analysis of countries through tables and through graphical forms(like bar chart, pie chart etc..).Based on the data which is present in the data set ,dashboard will show analysis(like top 10 countries with highest number of deaths ,top 10 countries with lowest number of deaths etc..) using different modules of python.

**4.SYSTEM DESIGN**

**4.1 SOFTWARE DESCRIPTION**

**4.1.1 Python IDE**

IDE stands for Integrated Development Environment. It’s a coding tool which allows you to write, test and debug your code in an easier way, as they typically offer code completion or code insight by highlighting, resource management, debugging tools. And even though the IDE is a strictly defined concept, it’s starting to be redefined as other tools such as notebooks start gaining more and more features that traditionally belong to IDEs. For example, debugging your code is also possible in Jupyter Notebook.

**4.1.2 NumPy**

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

**4.1.3 pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures.Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Key Features of Pandas:

* Fast and efficient DataFrame object with default and customized indexing.
* Tools for loading data into in-memory data objects from different file formats.
* Data alignment and integrated handling of missing data.
* Reshaping and pivoting of date sets.
* Label-based slicing, indexing and subsetting of large data sets.
* Columns from a data structure can be deleted or inserted.
* Group by data for aggregation and transformations.
* High performance merging and joining of data.
* Time Series functionality.

**4.1.4 Streamlit**

**S**treamlit is an open-source python library that is useful to create and share data web apps. It is slowly gaining a lot of momentum in the data science community. Because of the ease with which one can develop a data science web app, many developers use it in their daily workflow.

**Advantages:**   
1. It embraces python-scripting.   
2. Less code is needed to create amazing web-apps.   
3. No call backs are needed since widgets are treated as variables.   
4. Data caching simplifies and speeds up computation pipelines.

**4.1.5 mysql.connector**

MySQL Connector/Python enables Python programs to access MySQL databases, using an API that is compliant with the Python Database API Specification v2.0 (PEP 249). It is written in pure Python and does not have any dependencies except for the Python Standard Library.

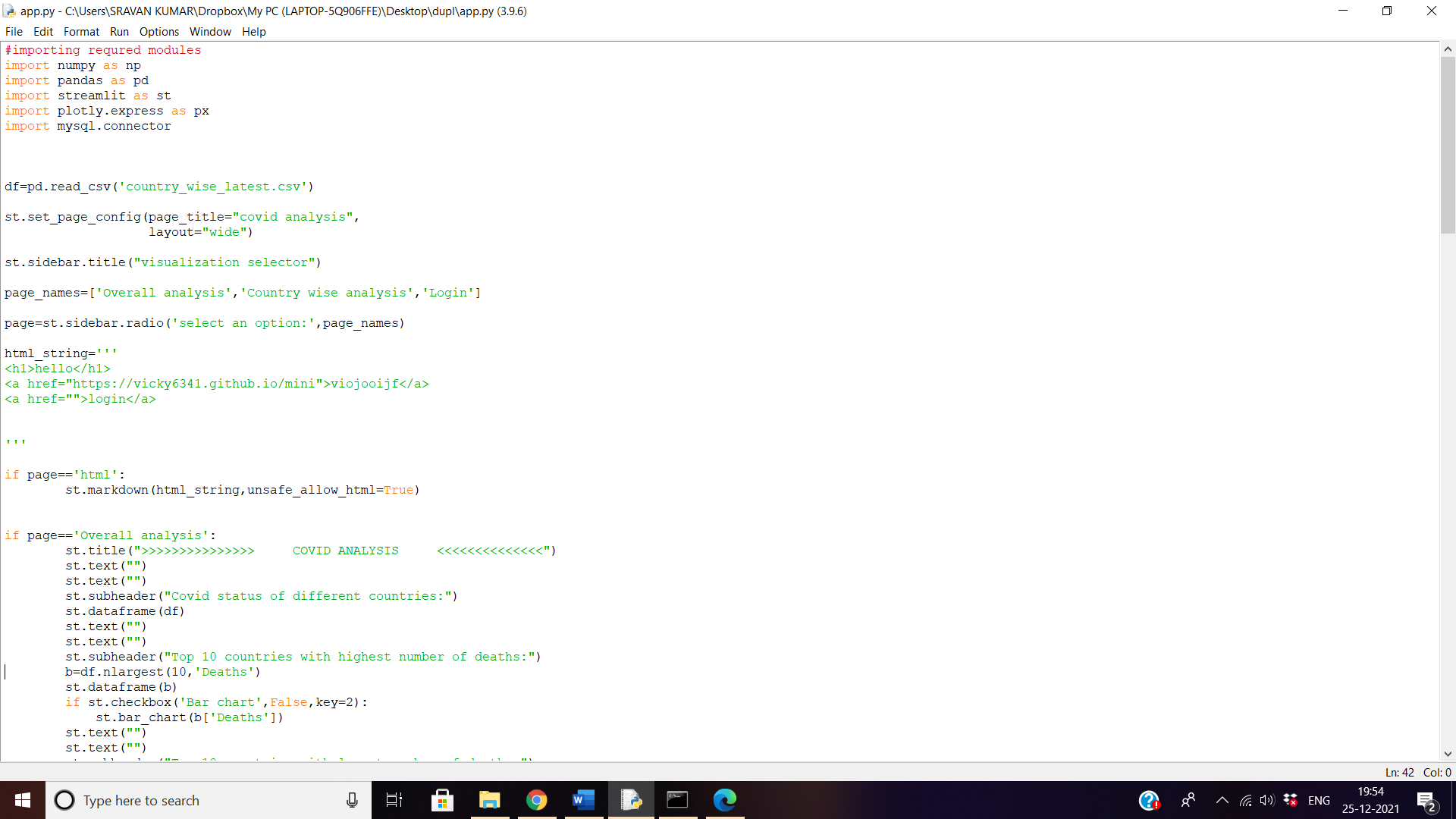
**There are various versions of MySQL Connector/Python available:**

| Connector/Python Version | MySQL Server Versions | Python Versions |
| --- | --- | --- |
| 8.0 | 8.0, 5.7, 5.6, 5.5 | 3.8, 3.7, 3.6, 3.5, 3.4, 2.7 |
| 2.2 (continues as 8.0) | 5.7, 5.6, 5.5 | 3.5, 3.4, 2.7 |
| 2.1 | 5.7, 5.6, 5.5 | 3.5, 3.4, 2.7, 2.6 |
| 2.0 | 5.7, 5.6, 5.5 | 3.5, 3.4, 2.7, 2.6 |
| 1.2 | 5.7, 5.6, 5.5 (5.1, 5.0, 4.1) | 3.4, 3.3, 3.2, 3.1, 2.7, 2.6 |

**5 Implementation:**

**5.1 Import modules of python:**

At first, we have to import different modules of python ,in which each one has specific functionality like numpy (for data manipulation),pandas(data analysis and data visualization), streamlit(to create a simple web application without using front end and back end technologies like html,css,javascript etc..).

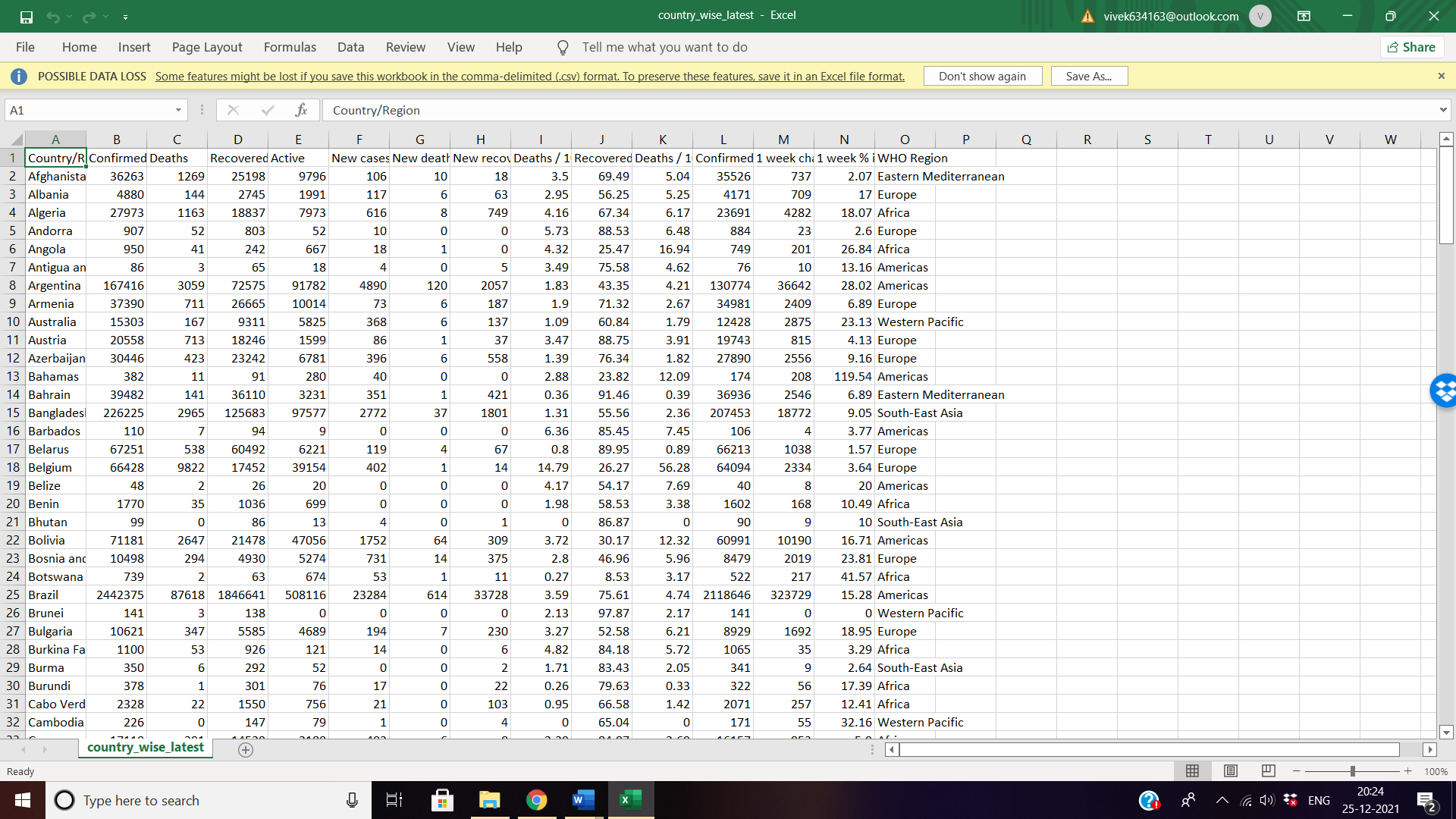


**5.2 Collecting and cleaning of dataset:**

In order to analyse the data we have to first collect the data ,for that purpose we have downloaded a covid dataset from (<https://www.kaggle.com/datasets>)

Kaggle website.and the dataset consists of values of various attributes of different countries.(like number of deaths,number of recovered cases,number

of confirmed cases etc..)which are used to analyse the data accurately.



After downloading the dataset,we have to import the whole data which is present in a csv file into a dataframe data structure which is present in pandas module of python.The imported data consists of unnecessary information which is not useful for our analysis,so we have to remove unnecessary data present in the dataframe.

**5.3 Overall Analysis:**

In our dashboard home page consists of overall covid analysis of countries in which we have show the countries with top deaths and lowest deaths and we have to represent them in graphical forms like barcharts.And to select top ‘n’

Rows with high ‘x’ attribute values we will use nlargest(n,’x’) function ,which is applied on dataframe object and similarly we will use nsmallest(n,’x’) function to get top ‘n’ rows with smallest ‘x’ attribute values.

Fig 1.1 shows data and bar charts of top 10 countries with highest number of deaths.

Fig 1.2 shows data and bar charts of top 10 countries with lowest number of deaths.

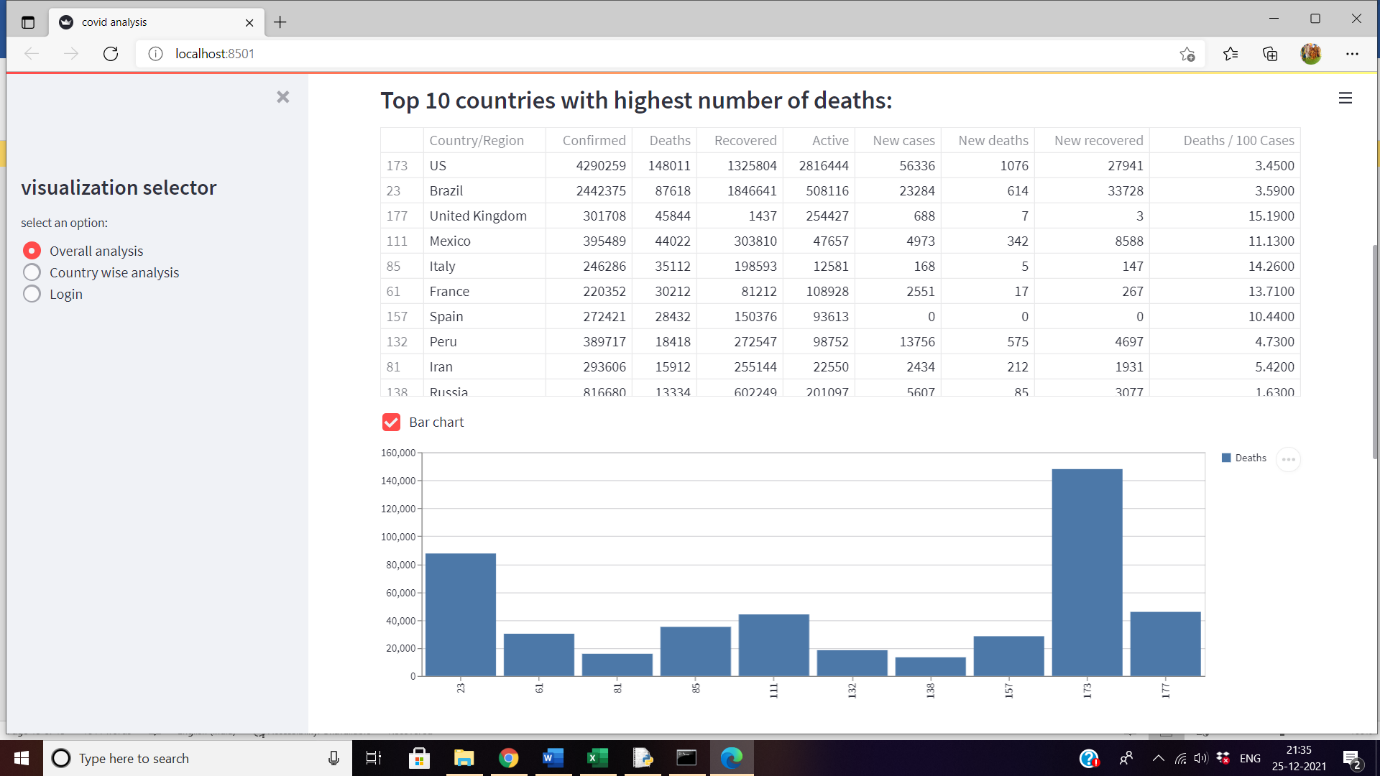
****

Fig-1.1 top 10 countries with highest no. of deaths.

****

**Fig-1.2 top 10 countries with lowest no. of deaths.**

**5.4 Country wise analysis:**

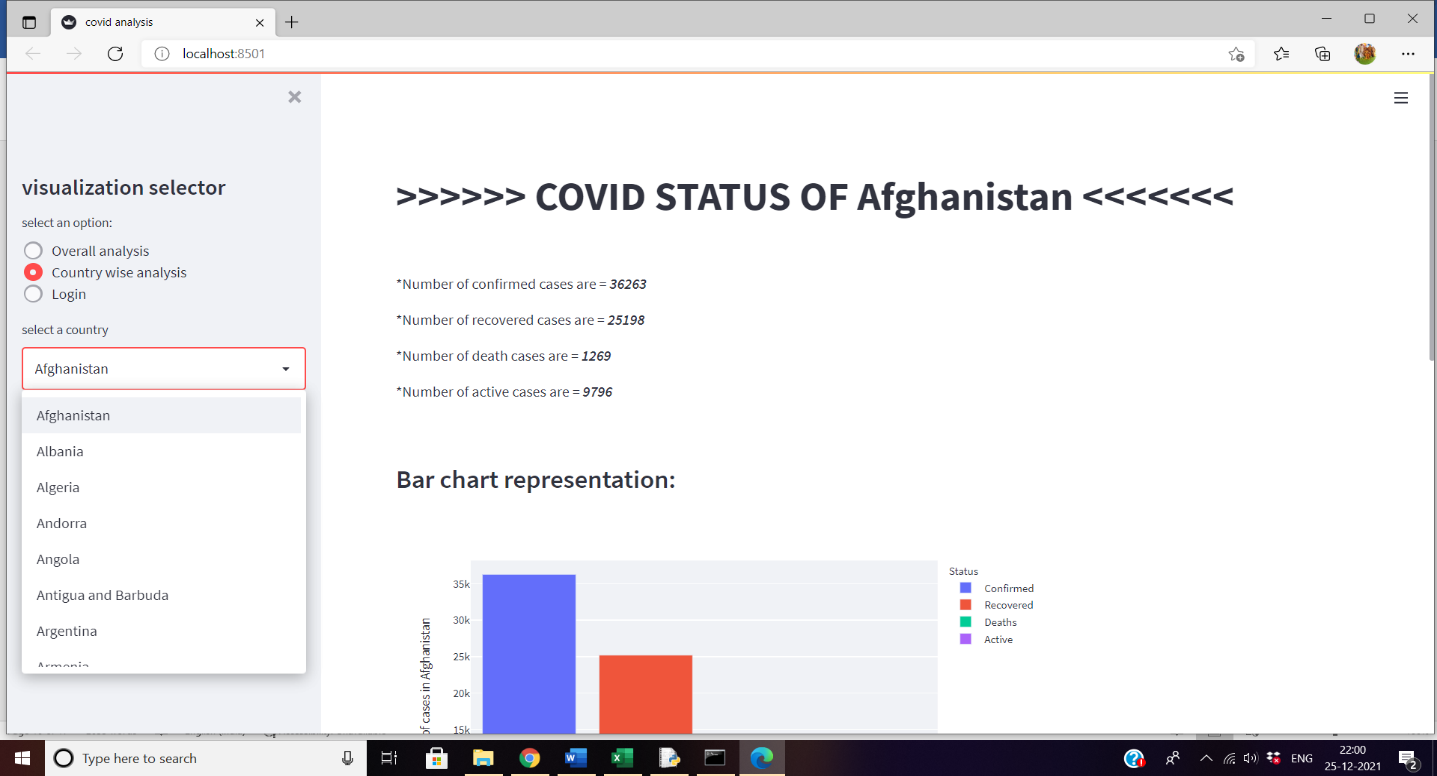
In country wise analysis, dashboard will show about each country data like number of confirmed cases,number of deaths,number of active cases.And the end user should select the country of which ,he should want the data. Along with data the dashboard also displays respective bar chart and pie chart representation of selected country.And in order to represent the data in pie chart we use bar() function.

**Syntax:**plotly.express.bar(data\_frame=None, x=None, y=None, color=None, facet\_row=None, facet\_col=None, facet\_col\_wrap=0, hover\_name=None, hover\_data=None, custom\_data=None, text=None, error\_x=None, error\_x\_minus=None, error\_y=None, error\_y\_minus=None, animation\_frame=None, animation\_group=None, category\_orders={}, labels={}, color\_discrete\_sequence=None, color\_discrete\_map={}, color\_continuous\_scale=None, range\_color=None, color\_continuous\_midpoint=None, opacity=None, orientation=None, barmode=’relative’, log\_x=False, log\_y=False, range\_x=None, range\_y=None, title=None, template=None, width=None, height=None)

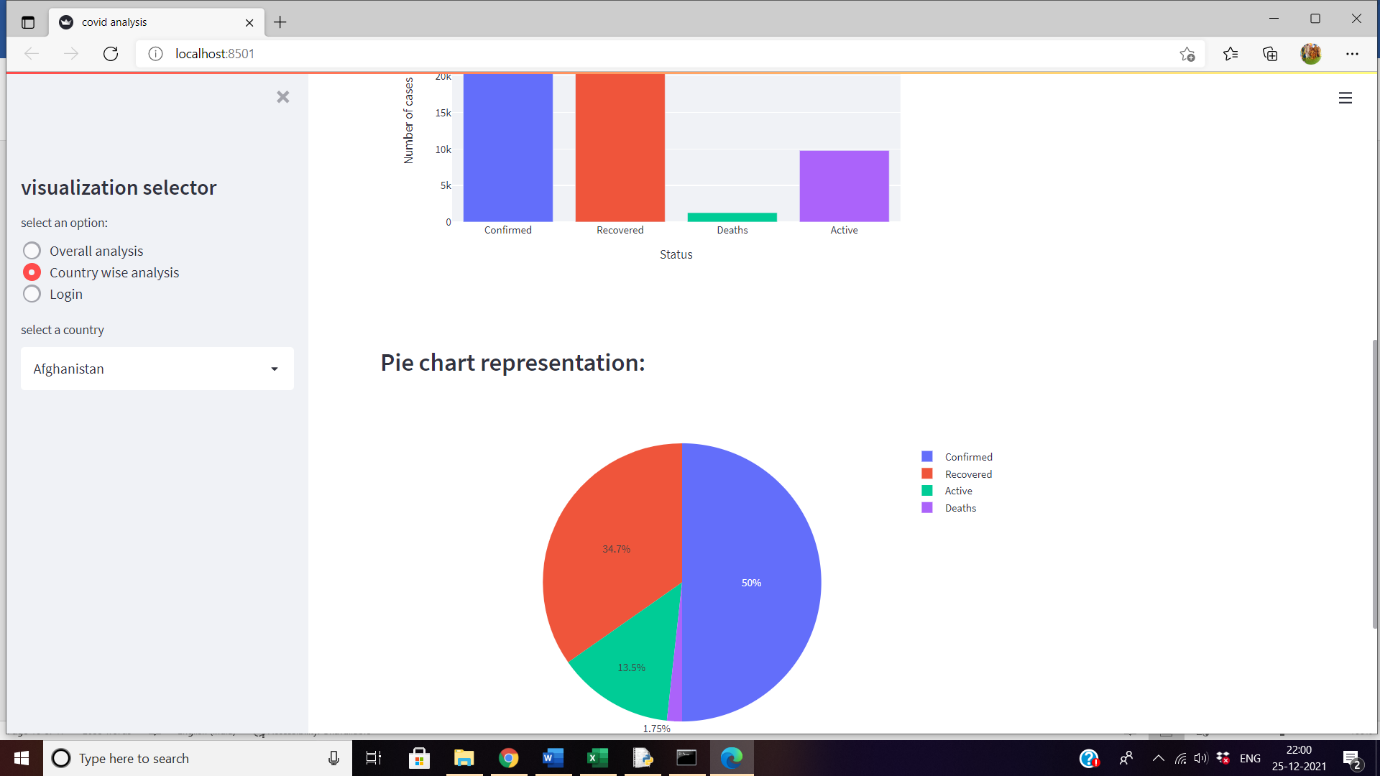
*And to represent data in pie chart we use pie() function.*

**Syntax:**plotly.express.pie(data\_frame=None, names=None, values=None, color=None, color\_discrete\_sequence=None, color\_discrete\_map={}, hover\_name=None, hover\_data=None, custom\_data=None, labels={}, title=None, template=None, width=None, height=None, opacity=None, hole=None)

*Fig-1.3,1.4 shows the country wise analysis of selected country.*

**

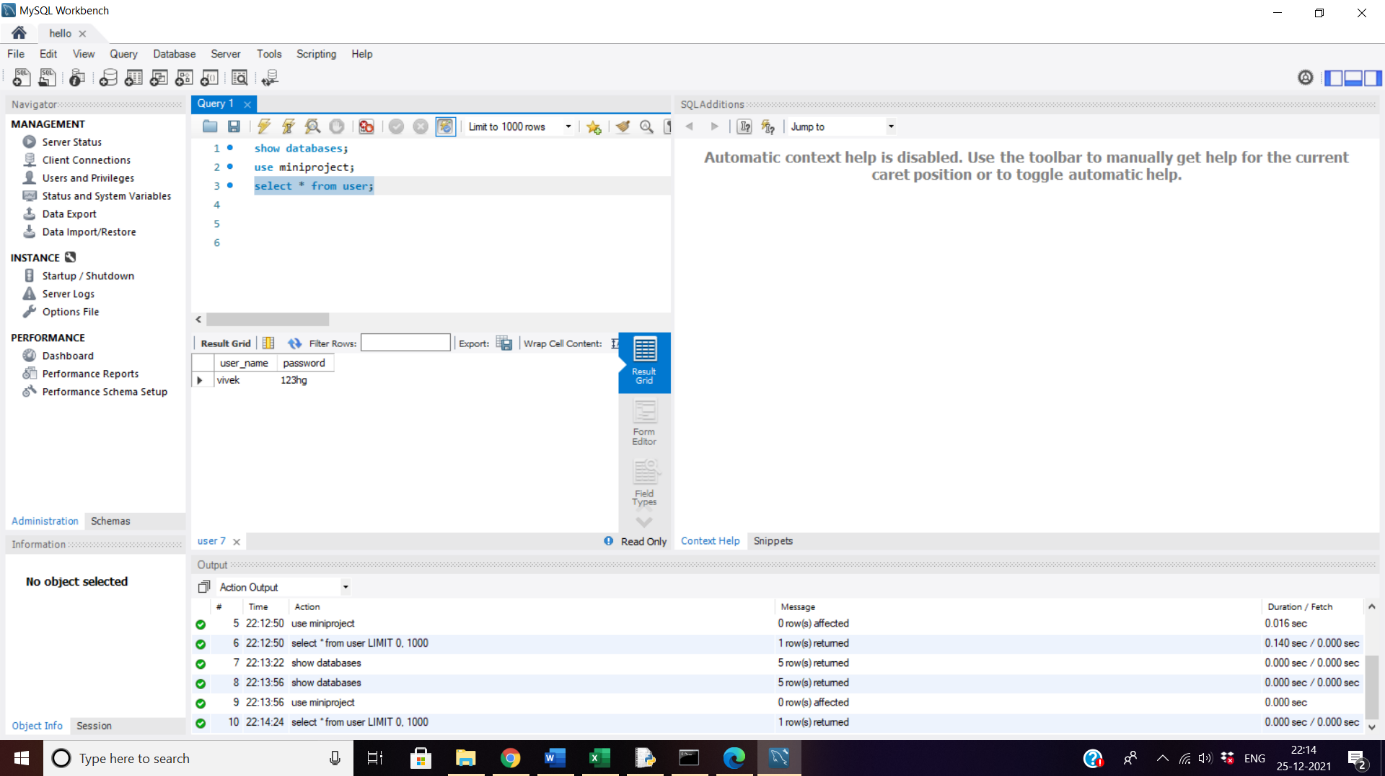
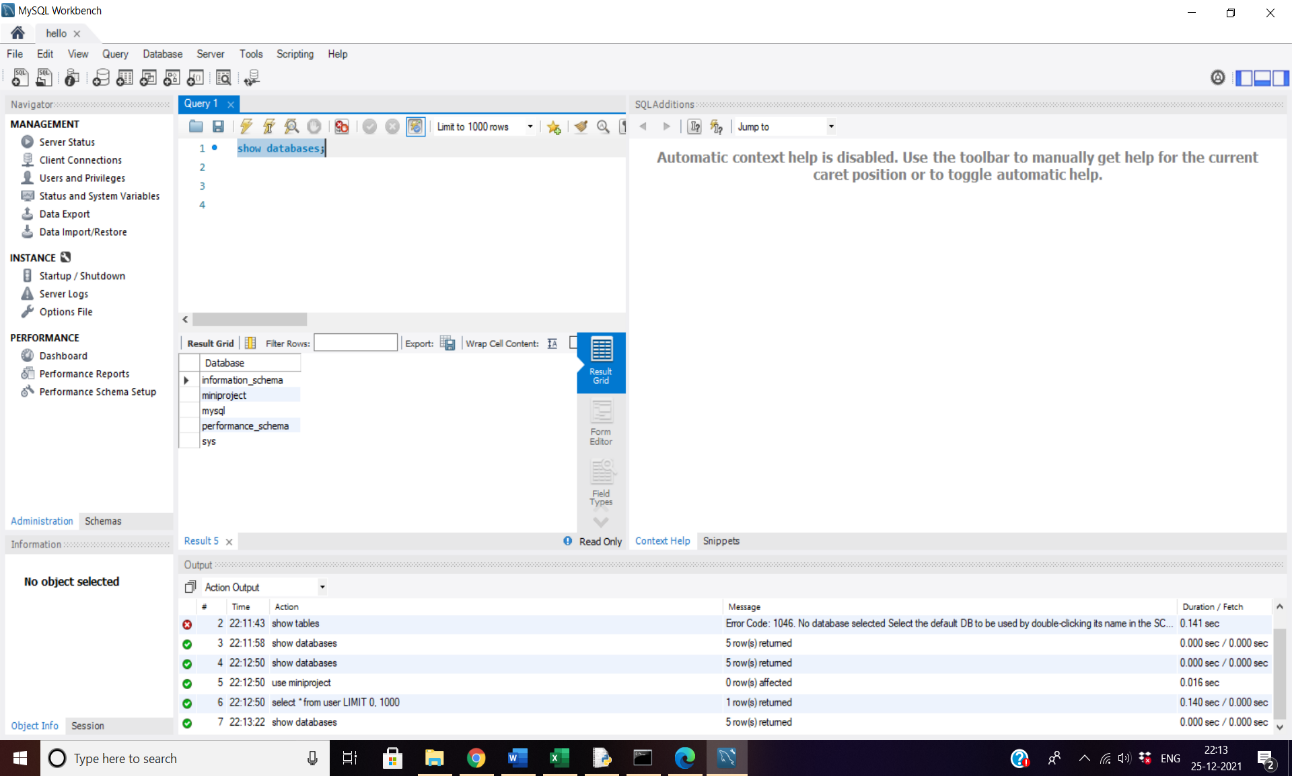
*Figure-1.3*

**

*Figure-1.4*

**5.5 Admin authentication:**

Our web application allows only admin to update data of covid dataset, For that we have connected our python program to mysql database.In mysql database,we have created a separate database called ‘miniproject’ in which there is a relation or table ‘user’ which stores the admin login credentials through which only admin can able to update the dataset after successful authentication.



*Figure-(1.5) figure-(1.6)*

In order to connect and communicate our python program with mysql database we have to follow the below steps:

1)Load and register driver.

2)Establish connection between python application and database.

3)create statement object.

4)send and execute sql queries.

5)process results from resultset.

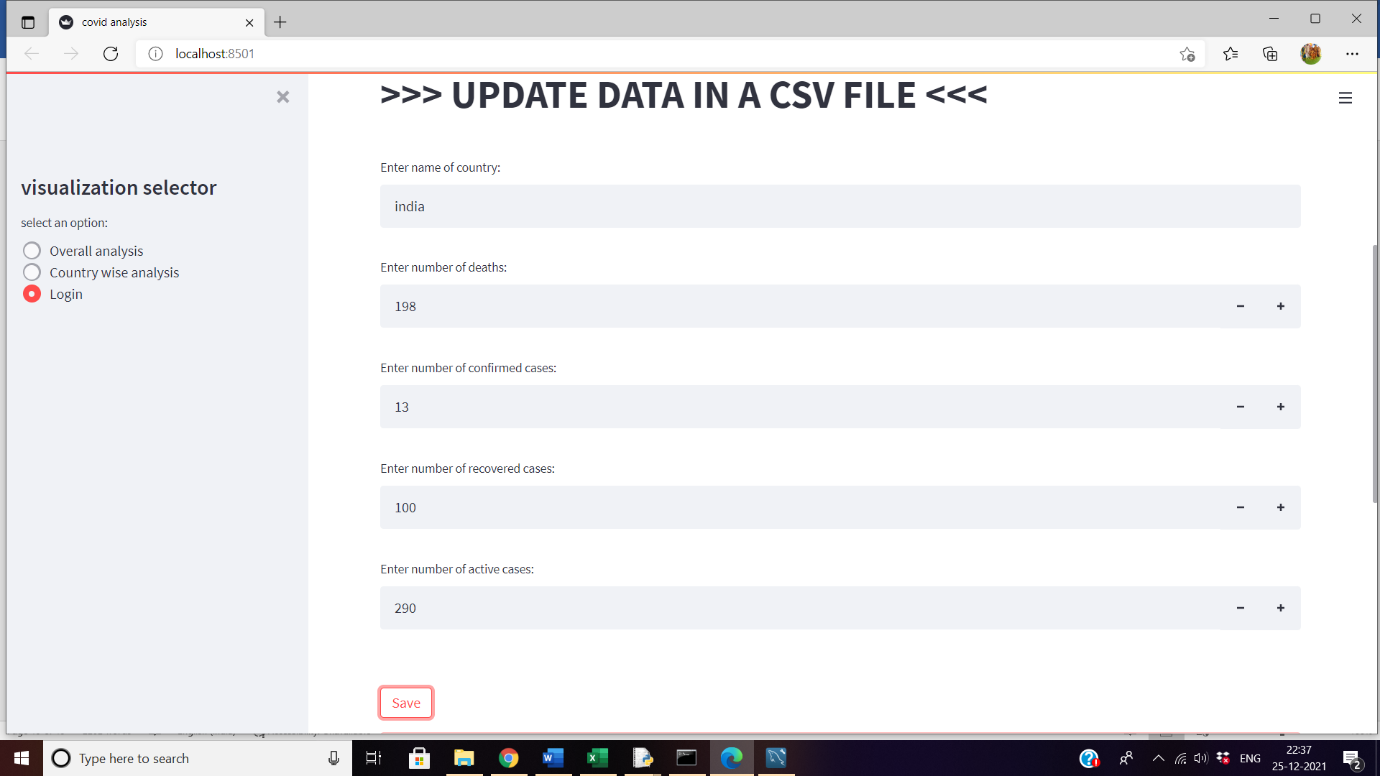
6)close connection.

The above all steps can be implemented in the form of code by following way:



**5.6 Data Updation:**

In this page,Admin have to give the correct credentials to update the database, otherwise a warning saying ‘give valid credentials’ will be displayed on the screen.And after giving correct login credentials admin can update the data in the following way:



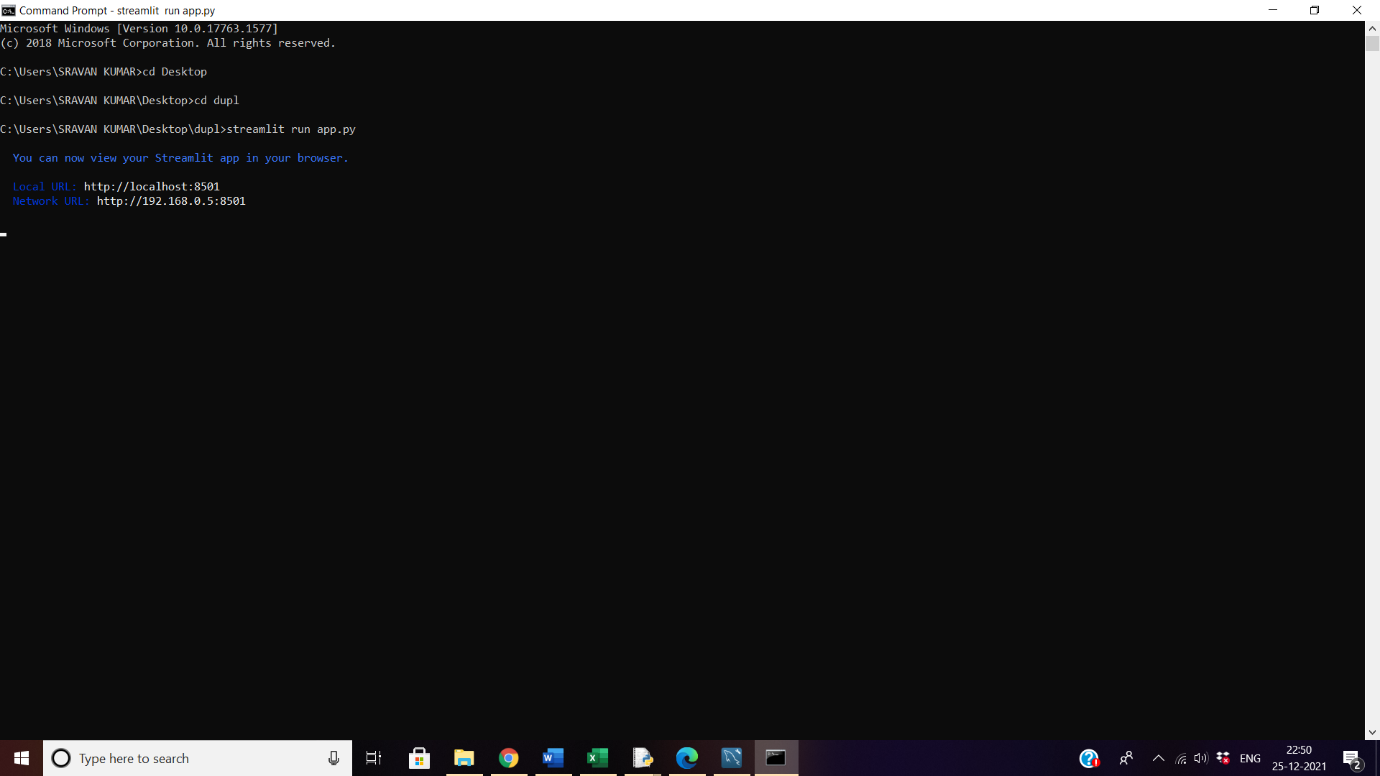
After updating the data admin has to click on save button. So,the data will be updated in csv file and all the data and graphs in overall analysis page,country wise analysis page will be changed according to the changes made by the admin.

**6.Testing**

Finally our web application got created and In order to run out streamlit web app we have to run the following command in our python IDE or command prompt:

Streamlit run app.py

We have to run the above command after going to respective folder in which app.py has saved.otherwise we will get ‘NO SUCH DIRECTORY/FILE’.And after getting execution successful two links will be created which are local url and network url.



**7.Development**

**Code:**

**import numpy as np**

**import pandas as pd**

**import streamlit as st**

**import plotly.express as px**

**import mysql.connector**

**df=pd.read\_csv('country\_wise\_latest.csv')**

**st.set\_page\_config(page\_title="covid analysis", layout="wide")**

**st.sidebar.title("visualization selector")**

**page\_names=['Overall analysis','Country wise analysis','Login']**

**page=st.sidebar.radio('select an option:',page\_names)**

**if page=='html':**

**st.markdown(html\_string,unsafe\_allow\_html=True)**

**if page=='Overall analysis':**

**st.title(">>>>>>>>>>>>>>> COVID ANALYSIS <<<<<<<<<<<<<<")**

**st.text("")**

**st.text("")**

**st.subheader("Covid status of different countries:")**

**st.dataframe(df)**

**st.text("")**

**st.text("")**

**st.subheader("Top 10 countries with highest number of deaths:")**

**b=df.nlargest(10,'Deaths')**

**st.dataframe(b)**

**if st.checkbox('Bar chart',False,key=2):**

**st.bar\_chart(b['Deaths'])**

**st.text("")**

**st.text("")**

**st.subheader("Top 10 countries with lowest number of deaths:")**

**b=df.nsmallest(10,'Deaths')**

**st.dataframe(b)**

**b\_cou\_list=b["Country/Region"].values.tolist()**

**b\_active\_list=b["Active"].values.tolist()**

**b=pd.DataFrame({**

**'index':b\_cou\_list,**

**'Active':b\_active\_list,**

**}).set\_index('index')**

**if st.checkbox('Bar chart',False,key=3):**

**st.bar\_chart(b)**

**def get\_total\_dataframe(dataset):**

**total\_dataframe=pd.DataFrame({'Status':['Confirmed','Recovered','Deaths','Active'],**

**'Number of cases':(dataset.iloc[0]['Confirmed'],dataset.iloc[0]['Recovered'],dataset.iloc[0]['Deaths'],dataset.iloc[0]['Active'])})**

**return total\_dataframe**

**country\_total=get\_total\_dataframe(df)**

**if page=='Country wise analysis':**

**select=st.sidebar.selectbox('select a country',df['Country/Region'])**

**country\_data=df[df['Country/Region']==select]**

**confirm=country\_data.iloc[0]['Confirmed']**

**recovered=country\_data.iloc[0]['Recovered']**

**deaths=country\_data.iloc[0]['Deaths']**

**active=country\_data.iloc[0]['Active']**

**st.title(" >>>>>> COVID STATUS OF %s <<<<<<<"%(select))**

**st.text("")**

**st.text("")**

**st.text("")**

**st.markdown("\*Number of confirmed cases are = \*\*\_%d\_\*\*"%(confirm))**

**st.markdown("\*Number of recovered cases are = \*\*\_%d\_\*\*"%(recovered))**

**st.markdown("\*Number of death cases are = \*\*\_%d\_\*\*"%(deaths))**

**st.markdown("\*Number of active cases are = \*\*\_%d\_\*\*"%(active))**

**st.text("")**

**st.text("")**

**st.text("")**

**st.subheader("Bar chart representation:")**

**c\_t\_g=px.bar(country\_total,x='Status',y='Number of cases',labels={'Number of cases':'Number of cases in %s'% (select)},color='Status')**

**st.plotly\_chart(c\_t\_g)**

**st.text("")**

**st.text("")**

**st.subheader("Pie chart representation:")**

**fig=px.pie(country\_data,values=[confirm,recovered,deaths,active],names=['Confirmed','Recovered','Deaths','Active'])**

**st.plotly\_chart(fig)**

**if page=='Login':**

**st.title(">>> Login(only admin can update data)<<<")**

**user=st.text\_input("Enter user name:","")**

**st.text("")**

**st.text("")**

**password=st.text\_input("Enter password:","")**

**st.text("")**

**st.text("")**

**if 'counter' not in st.session\_state:**

**st.session\_state.counter=False**

**if st.button("login") or st.session\_state.counter:**

**st.session\_state.counter=True**

**mydb=mysql.connector.connect(host='localhost',user='root',passwd='Vivek@634163#',database='miniproject',auth\_plugin='mysql\_native\_password',port='3306')**

**cursor=mydb.cursor(buffered=True)**

**cursor.execute("select \* from user where user\_name='vivek' ")**

**data=cursor.fetchall()**

**if len(data)!=0:**

**if(data[0][0]==user and data[0][1]==password):**

**st.title(">>> UPDATE DATA IN A CSV FILE <<<")**

**st.text("")**

**st.text("")**

**name=st.text\_input("Enter name of country:","")**

**st.text("")**

**deaths=st.number\_input("Enter number of deaths:",min\_value=0,max\_value=10000000,step=1)**

**st.text("")**

**confirmed=st.number\_input("Enter number of confirmed cases:",min\_value=0,max\_value=10000000,step=1)**

**st.text("")**

**recovered=st.number\_input("Enter number of recovered cases:",min\_value=0,max\_value=10000000,step=1)**

**st.text("")**

**active=st.number\_input("Enter number of active cases:",min\_value=0,max\_value=10000000,step=1)**

**st.text("")**

**st.text("")**

**st.text("")**

**if st.button("Save"):**

**name=name[0].upper()+name[1:]**

**df.loc[df['Country/Region']==name,'Confirmed']=confirmed**

**df.loc[df['Country/Region']==name,'Deaths']=deaths**

**df.loc[df['Country/Region']==name,'Recovered']=recovered**

**df.loc[df['Country/Region']==name,'Active']=active**

**df.to\_csv("country\_wise\_latest.csv",index=False)**

**st.markdown("successfully saved")**

**else:**

**st.markdown("give valid credentials")**