|  |
| --- |
|  |
| Fighting CORONA  The Pandemic of 2020 |
| |  |  |  |  | | --- | --- | --- | --- | | [Name] | XII F | [Rollno] | Bal Bhavan International School | |

## **CERTIFICATE**

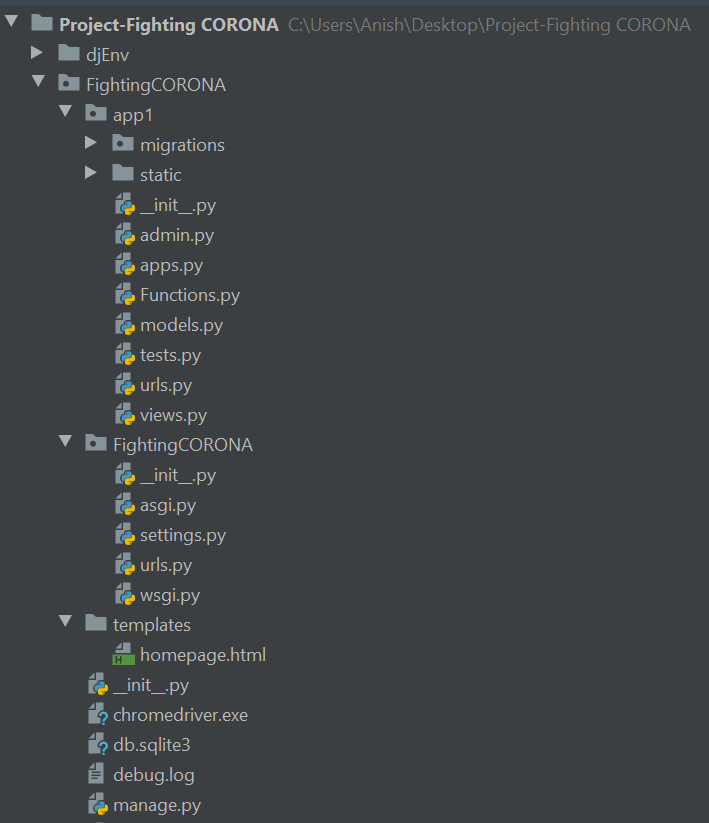
**This is to certify that the content of this project entitled, ‘FightingCORONA’ has been completed successfully by [Name] of class XII Science of Bal Bhavan International School under my guidance in the academic year 2020-21. The project is a result of his efforts and endeavors. The project is found worthy of acceptance as final project report for the subject Computer Science of Class XII.**

**Reema Chaudhary**

**P.G.T. Computer Science**

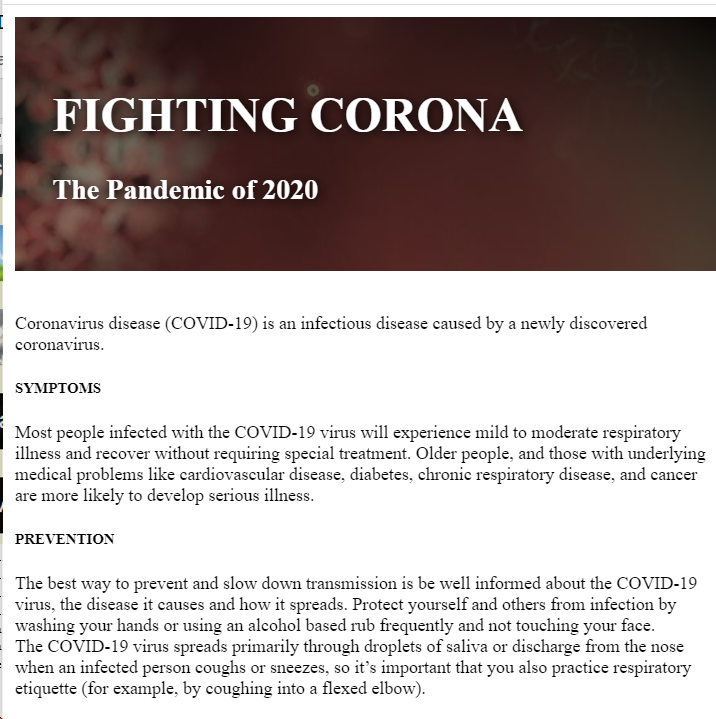
**Bal Bhavan International School**

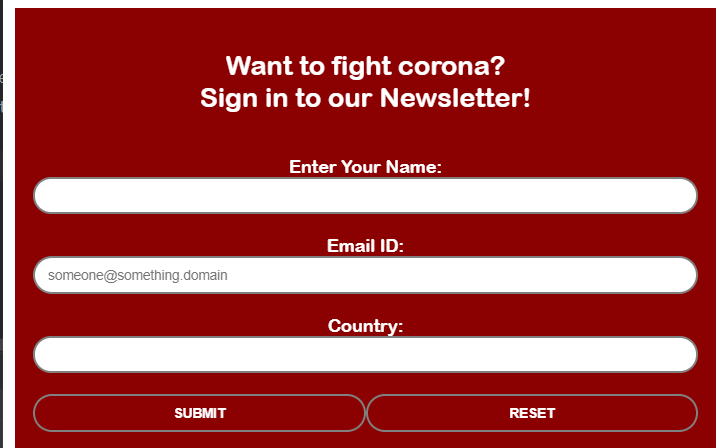
**Directory structure**

****

****

# HTML and CSS





## **Home page**

## Source Code:

## {% load static %}

<!DOCTYPE html>

<html>

<head>

<title>FIGHTING CORONA</title>

<link rel="stylesheet" href="{% static 'general.css' %}">

</head>

<body>

<header style="background-image:url('{% static 'corona back 2.jpg' %}'); height:100%">

<h1>FIGHTING CORONA</h1>

<h3>The Pandemic of 2020</h3>

</header>

<content style="margin:1em;">

<br><br>

Coronavirus disease (COVID-19) is an infectious disease caused by a newly discovered coronavirus.

<br>

<h5>SYMPTOMS</h5>

Most people infected with the COVID-19 virus will experience mild to moderate respiratory illness and recover without requiring special treatment. Older people, and those with underlying medical problems like cardiovascular disease, diabetes, chronic respiratory disease, and cancer are more likely to develop serious illness.

<br>

<h5>PREVENTION</h5>

The best way to prevent and slow down transmission is be well informed about the COVID-19 virus, the disease it causes and how it spreads. Protect yourself and others from infection by washing your hands or using an alcohol based rub frequently and not touching your face.

<br>

The COVID-19 virus spreads primarily through droplets of saliva or discharge from the nose when an infected person coughs or sneezes, so it’s important that you also practice respiratory etiquette (for example, by coughing into a flexed elbow).

<br><br>

</content>

<div id=form>

<h2>Want to fight corona?<br>

Sign in to our Newsletter!

</h2>

<br>

<form action=# method="post">

{% csrf\_token %}

<label>Enter Your Name:</label><br>

<input type=text name="Name:" required><br><br>

<label>Email ID:</label><br>

<input type= email name="Email:" title="someone@something.com" required pattern="[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,}$" placeholder="someone@something.domain"><br><br>

<label>Country:</label><br>

<input list="Country:" name="Country:" id="Country" required>

<datalist id="Country:" name="Country:">

<option value= "Afghanistan" > Afghanistan</option>

<option value= "Albania" > Albania</option>

<option value= "Algeria" > Algeria</option>

<option value= "Andorra" > Andorra</option>

<option value= "Angola" > Angola</option>

<option value= "Anguilla" > Anguilla</option>

<option value= "Antigua and Barbuda" > Antigua and Barbuda</option>

<option value= "Argentina" > Argentina</option>

<option value= "Armenia" > Armenia</option>

<option value= "Aruba" > Aruba</option>

<option value= "Australia" > Australia</option>

<option value= "Austria" > Austria</option>

<option value= "Azerbaijan" > Azerbaijan</option>

<option value= "Bahamas" > Bahamas</option>

<option value= "Bahrain" > Bahrain</option>

<option value= "Bangladesh" > Bangladesh</option>

<option value= "Barbados" > Barbados</option>

<option value= "Belarus" > Belarus</option>

<option value= "Belgium" > Belgium</option>

<option value= "Belize" > Belize</option>

<option value= "Benin" > Benin</option>

<option value= "Bermuda" > Bermuda</option>

<option value= "Bhutan" > Bhutan</option>

<option value= "Bolivia" > Bolivia</option>

<option value= "Bosnia and Herzegovina" > Bosnia and Herzegovina</option>

<option value= "Botswana" > Botswana</option>

<option value= "Brazil" > Brazil</option>

<option value= "British Virgin Islands" > British Virgin Islands</option>

<option value= "Brunei" > Brunei</option>

<option value= "Bulgaria" > Bulgaria</option>

<option value= "Burkina Faso" > Burkina Faso</option>

<option value= "Burundi" > Burundi</option>

<option value= "CAR" > CAR</option>

<option value= "Cabo Verde" > Cabo Verde</option>

<option value= "Cambodia" > Cambodia</option>

<option value= "Cameroon" > Cameroon</option>

<option value= "Canada" > Canada</option>

<option value= "Caribbean Netherlands" > Caribbean Netherlands</option>

<option value= "Cayman Islands" > Cayman Islands</option>

<option value= "Chad" > Chad</option>

<option value= "Channel Islands" > Channel Islands</option>

<option value= "Chile" > Chile</option>

<option value= "China" > China</option>

<option value= "Colombia" > Colombia</option>

<option value= "Comoros" > Comoros</option>

<option value= "Congo" > Congo</option>

<option value= "Costa Rica" > Costa Rica</option>

<option value= "Croatia" > Croatia</option>

<option value= "Cuba" > Cuba</option>

<option value= "Curaçao" > Curaçao</option>

<option value= "Cyprus" > Cyprus</option>

<option value= "Czechia" > Czechia</option>

<option value= "DRC" > DRC</option>

<option value= "Denmark" > Denmark</option>

<option value= "Diamond Princess" > Diamond Princess</option>

<option value= "Djibouti" > Djibouti</option>

<option value= "Dominica" > Dominica</option>

<option value= "Dominican Republic" > Dominican Republic</option>

<option value= "Ecuador" > Ecuador</option>

<option value= "Egypt" > Egypt</option>

<option value= "El Salvador" > El Salvador</option>

<option value= "Equatorial Guinea" > Equatorial Guinea</option>

<option value= "Eritrea" > Eritrea</option>

<option value= "Estonia" > Estonia</option>

<option value= "Eswatini" > Eswatini</option>

<option value= "Ethiopia" > Ethiopia</option>

<option value= "Faeroe Islands" > Faeroe Islands</option>

<option value= "Falkland Islands" > Falkland Islands</option>

<option value= "Fiji" > Fiji</option>

<option value= "Finland" > Finland</option>

<option value= "France" > France</option>

<option value= "French Guiana" > French Guiana</option>

<option value= "French Polynesia" > French Polynesia</option>

<option value= "Gabon" > Gabon</option>

<option value= "Gambia" > Gambia</option>

<option value= "Georgia" > Georgia</option>

<option value= "Germany" > Germany</option>

<option value= "Ghana" > Ghana</option>

<option value= "Gibraltar" > Gibraltar</option>

<option value= "Greece" > Greece</option>

<option value= "Greenland" > Greenland</option>

<option value= "Grenada" > Grenada</option>

<option value= "Guadeloupe" > Guadeloupe</option>

<option value= "Guatemala" > Guatemala</option>

<option value= "Guinea" > Guinea</option>

<option value= "Guinea-Bissau" > Guinea-Bissau</option>

<option value= "Guyana" > Guyana</option>

<option value= "Haiti" > Haiti</option>

<option value= "Honduras" > Honduras</option>

<option value= "Hong Kong" > Hong Kong</option>

<option value= "Hungary" > Hungary</option>

<option value= "Iceland" > Iceland</option>

<option value= "India" > India</option>

<option value= "Indonesia" > Indonesia</option>

<option value= "Iran" > Iran</option>

<option value= "Iraq" > Iraq</option>

<option value= "Ireland" > Ireland</option>

<option value= "Isle of Man" > Isle of Man</option>

<option value= "Israel" > Israel</option>

<option value= "Italy" > Italy</option>

<option value= "Ivory Coast" > Ivory Coast</option>

<option value= "Jamaica" > Jamaica</option>

<option value= "Japan" > Japan</option>

<option value= "Jordan" > Jordan</option>

<option value= "Kazakhstan" > Kazakhstan</option>

<option value= "Kenya" > Kenya</option>

<option value= "Kuwait" > Kuwait</option>

<option value= "Kyrgyzstan" > Kyrgyzstan</option>

<option value= "Laos" > Laos</option>

<option value= "Latvia" > Latvia</option>

<option value= "Lebanon" > Lebanon</option>

<option value= "Lesotho" > Lesotho</option>

<option value= "Liberia" > Liberia</option>

<option value= "Libya" > Libya</option>

<option value= "Liechtenstein" > Liechtenstein</option>

<option value= "Lithuania" > Lithuania</option>

<option value= "Luxembourg" > Luxembourg</option>

<option value= "MS Zaandam" > MS Zaandam</option>

<option value= "Macao" > Macao</option>

<option value= "Madagascar" > Madagascar</option>

<option value= "Malawi" > Malawi</option>

<option value= "Malaysia" > Malaysia</option>

<option value= "Maldives" > Maldives</option>

<option value= "Mali" > Mali</option>

<option value= "Malta" > Malta</option>

<option value= "Martinique" > Martinique</option>

<option value= "Mauritania" > Mauritania</option>

<option value= "Mauritius" > Mauritius</option>

<option value= "Mayotte" > Mayotte</option>

<option value= "Mexico" > Mexico</option>

<option value= "Moldova" > Moldova</option>

<option value= "Monaco" > Monaco</option>

<option value= "Mongolia" > Mongolia</option>

<option value= "Montenegro" > Montenegro</option>

<option value= "Montserrat" > Montserrat</option>

<option value= "Morocco" > Morocco</option>

<option value= "Mozambique" > Mozambique</option>

<option value= "Myanmar" > Myanmar</option>

<option value= "Namibia" > Namibia</option>

<option value= "Nepal" > Nepal</option>

<option value= "Netherlands" > Netherlands</option>

<option value= "New Caledonia" > New Caledonia</option>

<option value= "New Zealand" > New Zealand</option>

<option value= "Nicaragua" > Nicaragua</option>

<option value= "Niger" > Niger</option>

<option value= "Nigeria" > Nigeria</option>

<option value= "North Macedonia" > North Macedonia</option>

<option value= "Norway" > Norway</option>

<option value= "Oman" > Oman</option>

<option value= "Pakistan" > Pakistan</option>

<option value= "Palestine" > Palestine</option>

<option value= "Panama" > Panama</option>

<option value= "Papua New Guinea" > Papua New Guinea</option>

<option value= "Paraguay" > Paraguay</option>

<option value= "Peru" > Peru</option>

<option value= "Philippines" > Philippines</option>

<option value= "Poland" > Poland</option>

<option value= "Portugal" > Portugal</option>

<option value= "Qatar" > Qatar</option>

<option value= "Romania" > Romania</option>

<option value= "Russia" > Russia</option>

<option value= "Rwanda" > Rwanda</option>

<option value= "Réunion" > Réunion</option>

<option value= "S. Korea" > S. Korea</option>

<option value= "Saint Kitts and Nevis" > Saint Kitts and Nevis</option>

<option value= "Saint Lucia" > Saint Lucia</option>

<option value= "Saint Martin" > Saint Martin</option>

<option value= "Saint Pierre Miquelon" > Saint Pierre Miquelon</option>

<option value= "San Marino" > San Marino</option>

<option value= "Sao Tome and Principe" > Sao Tome and Principe</option>

<option value= "Saudi Arabia" > Saudi Arabia</option>

<option value= "Senegal" > Senegal</option>

<option value= "Serbia" > Serbia</option>

<option value= "Seychelles" > Seychelles</option>

<option value= "Sierra Leone" > Sierra Leone</option>

<option value= "Singapore" > Singapore</option>

<option value= "Sint Maarten" > Sint Maarten</option>

<option value= "Slovakia" > Slovakia</option>

<option value= "Slovenia" > Slovenia</option>

<option value= "Somalia" > Somalia</option>

<option value= "South Africa" > South Africa</option>

<option value= "South Sudan" > South Sudan</option>

<option value= "Spain" > Spain</option>

<option value= "Sri Lanka" > Sri Lanka</option>

<option value= "St. Barth" > St. Barth</option>

<option value= "St. Vincent Grenadines" > St. Vincent Grenadines</option>

<option value= "Sudan" > Sudan</option>

<option value= "Suriname" > Suriname</option>

<option value= "Sweden" > Sweden</option>

<option value= "Switzerland" > Switzerland</option>

<option value= "Syria" > Syria</option>

<option value= "Taiwan" > Taiwan</option>

<option value= "Tajikistan" > Tajikistan</option>

<option value= "Tanzania" > Tanzania</option>

<option value= "Thailand" > Thailand</option>

<option value= "Timor-Leste" > Timor-Leste</option>

<option value= "Togo" > Togo</option>

<option value= "Trinidad and Tobago" > Trinidad and Tobago</option>

<option value= "Tunisia" > Tunisia</option>

<option value= "Turkey" > Turkey</option>

<option value= "Turks and Caicos" > Turks and Caicos</option>

<option value= "UAE" > UAE</option>

<option value= "UK" > UK</option>

<option value= "USA" > USA</option>

<option value= "Uganda" > Uganda</option>

<option value= "Ukraine" > Ukraine</option>

<option value= "Uruguay" > Uruguay</option>

<option value= "Uzbekistan" > Uzbekistan</option>

<option value= "Vatican City" > Vatican City</option>

<option value= "Venezuela" > Venezuela</option>

<option value= "Vietnam" > Vietnam</option>

<option value= "Western Sahara" > Western Sahara</option>

<option value= "World" > World</option>

<option value= "Yemen" > Yemen</option>

<option value= "Zambia" > Zambia</option>

<option value= "Zimbabwe" > Zimbabwe</option>

</datalist>

<br><br>

<input type=submit value=SUBMIT><input type=reset value=RESET>

</form>

</div>

</body>

</html>

## CSS

\*{

box-sizing:border-box;

}

body{

margin:1em;

padding:0;

}

header{

height:100%;

margin:auto;

width:fill-parent;

background-attachment: fixed;

background-repeat: no-repeat;

background-position:right top;

background-size: cover;

background-color: #760202;

color:white;

text-shadow: 2px 2px 15px black;

font-size: 2em;

padding:50px;

}

content{

margin:2em;

padding:1em;

font-size:1.5em;

}

#form{

width: fill-parent;

padding:1em;

text-align:center;

color:white;

font-weight: 1000;

font-family: "Arial Rounded MT",sans-serif;

font-size: 1.5em;

}

#form:h2{font-size:2em;}

input{

margin:auto;

padding:1em;

width:100%;

height:50px;

border-radius:50px;

border:2px solid gray;

font-size: 0.75em;

}

input:hover{border-color:blue}

input[type=submit],input[type=reset]{

padding:1px;

width:50%;

background-color:#e00000;

color:white;

border-color:white:

font-size:1em;

font-weight: bold;

}

input[type=submit]:hover,input[type=reset]:hover{

background-color:#ff4500;

color:darkred;

border-color:darkred:

font-size:1em;

**DJANGO Web framework**

## Functions.py

import smtplib  
import ssl  
import random  
  
  
def mail(receiver\_email, message):  
 port = 465 # For SSL  
 smtp\_server = "smtp.gmail.com"  
 sender\_email = "fightingcorona2021@gmail.com" # Enter your address  
 password = "pxfvucsampcosost"  
 context = ssl.create\_default\_context()  
 with smtplib.SMTP\_SSL(smtp\_server, port, context=context) as server:  
 server.login(sender\_email, password)  
 server.sendmail(sender\_email, receiver\_email, message)  
  
# define a function to get tips  
def getTip():  
 tips = [  
 "Wash milk bags the moment we take it & wash your hands while you are at it.",  
 "Consider cancelling newspapers",  
 "Keep a separate tray for couriers. Courier person can place the envelope/pkg in the tray and courier may be left untouched for at least 24 hours.",  
 "Instruct maids not to touch main door. On entering the home, she has to immediatly wash hands thoroughly, before touching other things . After that, wipe the calling-bell switch with a cleaning fluid :)",  
 "Avoid getting swiggy, amazon etc as far as possible.",  
 "Wash all fruits and vegitables once we bring them home"  
 "Remote, phone and keyboards are the most highly contaminated elements in our house. Clean them at least once a day using cleaning fluid.",  
 "Wash hands frequently when in house or in office. Once every hour at least.",  
 "Avoid public transport as far as possible. Even Ola and Uber may be used when absolutely unavoidable.",  
 "Avoid gyms, swimming pool and other exercise areas, where surface contact or air - borne contamination is inevitable",  
 "Cancel tuitions, dance/music classes, etc.",  
 "When you return home from office, shopping, etc. discard your clothes and wash your hands and feet thoroughly.",  
 "Most importantly do not touch hands anywhere on face. Inform children and parents.",  
 "Ask senior citizens to stop going for the routine walking exercise."  
 ]  
 return (tips[random.randint(0, len(tips) - 1)])

## views.py

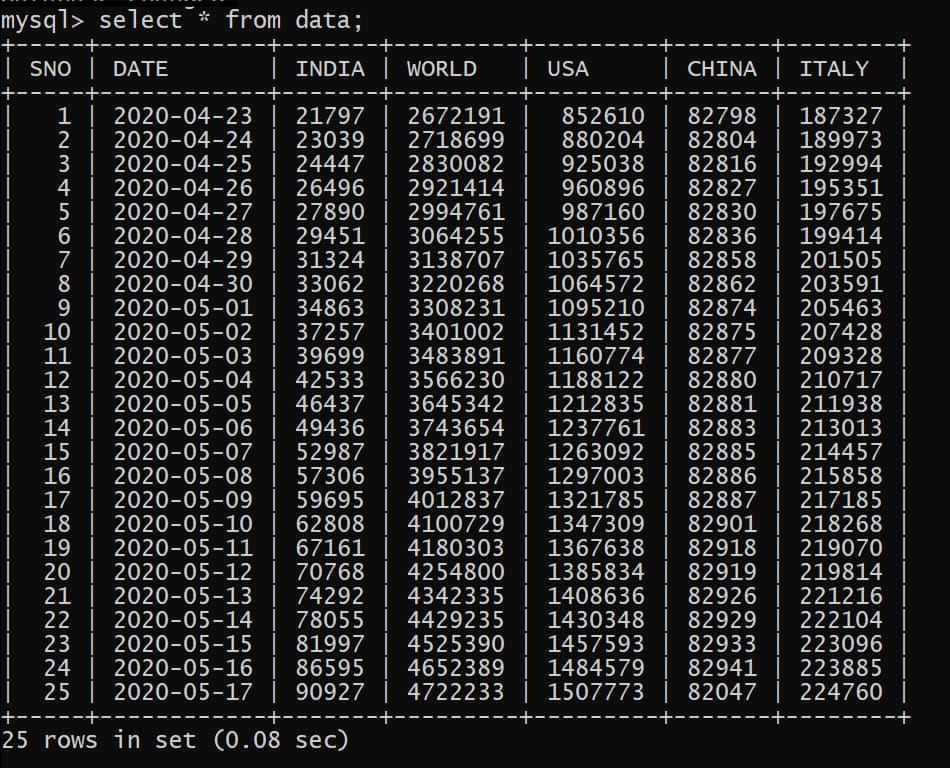
from django.shortcuts import render  
from django.core.files import File  
from . import Functions  
import smtplib,ssl  
  
def home(request):  
 if request.method =="POST":  
 port = 465 # For SSL  
 smtp\_server = "smtp.gmail.com"  
 sender\_email = "fightingcorona2021@gmail.com" # Enter your address  
 receiver\_email = "anishbanerjee2002@gmail.com" # Enter receiver address  
 password = "pxfvucsampcosost"  
 mydata = request.POST  
 with open('users.txt','w') as f:  
 wrt = File(f)  
 for key,value in mydata.items():  
 wrt.write(key)  
 wrt.write(value)  
 wrt.write("\n")  
 f=open("users.txt","r")  
 strg=f.read()  
 message = f"Subject: New login\n\nNew login!\n Details:{strg}"  
 f.close()  
 context = ssl.create\_default\_context()  
 with smtplib.SMTP\_SSL(smtp\_server, port, context=context) as server:  
 server.login(sender\_email, password)  
 server.sendmail(sender\_email, receiver\_email, message)  
  
  
 return render(request,"homepage.html")  
  
# Create your views here.

## urls.py

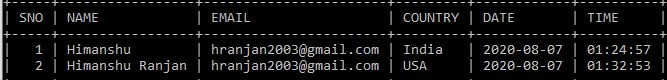
from django.contrib import admin  
from django.urls import path, include  
from app1 import views  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('home/', include('app1.urls'))  
]

**MySQL Tables**

## SQL table: data



## SQL table: record



## **Web Scrapping with Selenium and MySQL Connectivity**

## Scrapping and Updating.py

from selenium import webdriver  
import unittest  
import mysql.connector as mycon  
import datetime  
  
#Establish connection with database  
  
con = mycon.connect(  
 host="localhost",  
 user="root",  
 passwd="1234",  
 database="users"  
)  
cursor = con.cursor()  
cursor.execute("Select \* from data")  
  
data = cursor.fetchall()  
try:sno=data[len(data)-1][0]+1  
except:sno=1  
  
#Get current date  
  
def get\_day():  
 a=datetime.datetime.now()  
 day=a.day  
 month=a.month  
 year=a.year  
 day=f"{str(year)}-{str(month)}-{str(day)}"  
 return day  
  
#Scrapping  
  
  
class WebTable:  
 def \_\_init\_\_(self, webtable):  
 self.table = webtable  
  
 def column\_data(self, column\_number):  
 col = self.table.find\_elements\_by\_xpath("//tbody[1]/tr/td[" + str(column\_number) + "]")  
 rData = []  
 for webElement in col:  
 rData.append(webElement.text)  
 return rData

class Test(unittest.TestCase):  
   
 def test\_web\_table(self):  
 driver = webdriver.Chrome(executable\_path=r".\chromedriver.exe")  
 driver.implicitly\_wait(30)  
  
 driver.get("https://www.worldometers.info/coronavirus/")  
 w = WebTable(driver.find\_element\_by\_xpath('//table[@id="main\_table\_countries\_today"]/tbody[1]'))  
  
 country=w.column\_data(2)  
 n\_coun=country.count('')  
 for i in range(n\_coun):  
 country.remove('')  
  
   
 total\_cases=w.column\_data(3)  
 n\_total\_cases=total\_cases.count('')  
 for i in range(n\_total\_cases):  
 total\_cases.remove('')  
  
  
 # Scrap the required details  
   
 b=[]  
 for k in ["India","World","USA","China","Italy"]:  
   
 index\_coun = country.index(k)  
  
 cases = total\_cases[index\_coun]  
 temp=cases.split(',')  
 j=""  
 for i in temp:  
 j+=i  
 cases=int(j)  
 b.append(cases)  
  
 #Insert into database  
  
 cursor.execute(f'insert into data values({sno},"{get\_day()}",{b[0]},{b[1]},{b[2]},{b[3]},{b[4]});')  
 con.commit()  
 con.close()  
 print("work successful")  
  
 driver.close()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 unittest.main()

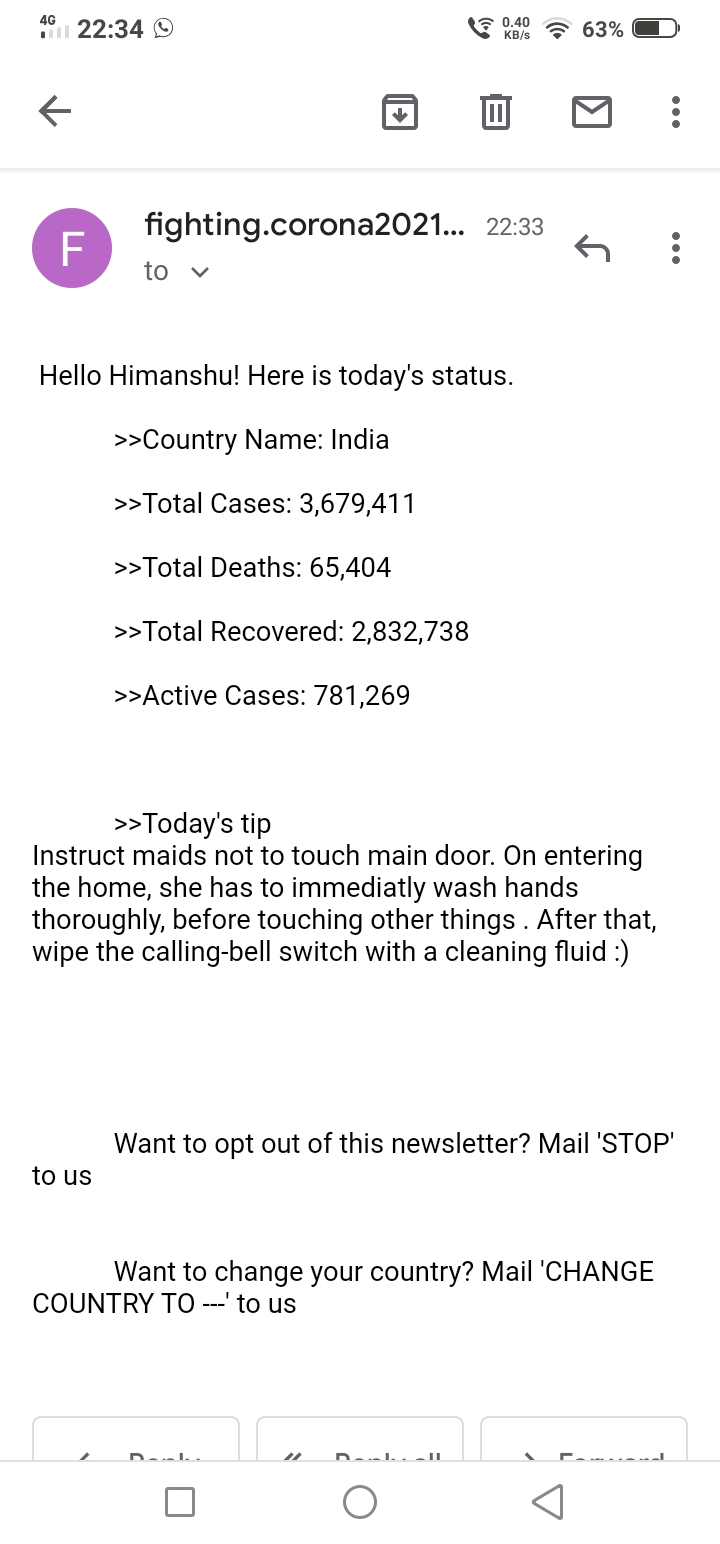
## UpdatingSQL.py

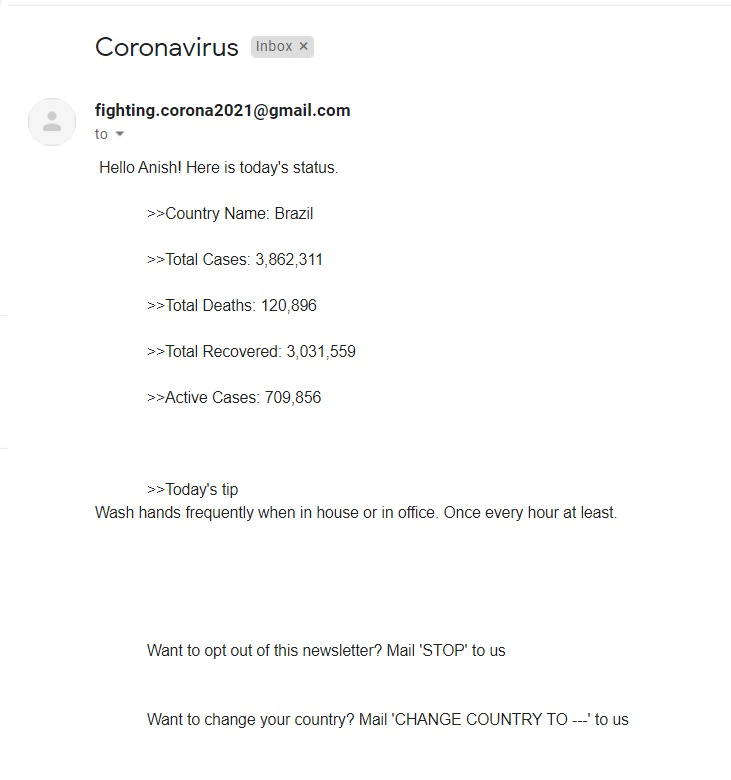
import mysql.connector as mycon  
import datetime  
#Get current date  
  
def get\_day():  
 a=datetime.datetime.now()  
 day=a.day  
 month=a.month  
 year=a.year  
 day1=f"{str(year)}-{str(month)}-{str(day)}"  
 return day1  
def get\_time():  
 a=datetime.datetime.now()  
 hour=a.hour  
 minute=a.minute  
 second=a.second  
 time1=f"{str(hour)}:{str(minute)}:{str(second)}"  
 return time1  
  
#file  
def upload():  
 try:  
 with open("users.txt","r") as f:  
 l=f.readlines()  
 l.pop(0)  
   
 name=l[0][5:-1]  
 email=l[1][6:-1]  
 country=l[2][8:-1]  
 print(name,email,country)  
   
 #connect  
   
 con=mycon.connect(  
 host="localhost",  
 user="root",  
 passwd="1234",  
 database="users"  
 )  
  
  
 if con.is\_connected():  
 print("Connected")  
   
 cursor=con.cursor()  
 cursor.execute("Select \* from record")  
  
 data=cursor.fetchall()  
  
 try:sno=data[len(data)-1][0]+1  
 except:sno=1  
 print(f'insert into record values({sno}, "{name}" ,"{email}","{country}","{get\_day()}","{get\_time()}");')  
 cursor.execute(f"insert into record values({sno},'{name}' ,'{email}','{country}','{get\_day()}','{get\_time()}');")  
 con.commit()  
 con.close()  
 print("Work Successful")  
  
 with open("users.txt","w") as f:  
 f.write("")  
   
   
 except Exception as e:  
 print(e)  
 print("Nothing to update")  
  
upload()

## WebScrapping.py

from selenium import webdriver  
import unittest  
from app1 import Functions  
import mysql.connector as mycon  
import updatingsql as SQL  
SQL.upload()  
  
#Establish connection with database  
  
con = mycon.connect(  
 host="localhost",  
 user="root",  
 passwd="1234",  
 database="users"  
)  
cursor = con.cursor()  
cursor.execute("Select \* from record ORDER BY SNO DESC LIMIT 1")  
  
data = cursor.fetchall()  
n = len(data)  
countries= []  
for i in range(n):  
 countries.append(data[i][3])  
  
  
  
class WebTable:  
 def \_\_init\_\_(self, webtable):  
 self.table = webtable  
  
 def column\_data(self, column\_number):  
 col = self.table.find\_elements\_by\_xpath("//tbody[1]/tr/td[" + str(column\_number) + "]")  
 rData = []  
 for webElement in col:  
 rData.append(webElement.text)  
 return rData  
  
  
class Test(unittest.TestCase):  
 def test\_web\_table(self):  
 driver = webdriver.Chrome(executable\_path=r".\chromedriver.exe")  
 driver.implicitly\_wait(30)  
  
 driver.get("https://www.worldometers.info/coronavirus/")  
 w = WebTable(driver.find\_element\_by\_xpath('//table[@id="main\_table\_countries\_today"]/tbody[1]'))  
  
 country = w.column\_data(2)  
 n\_coun = country.count('')  
 for i in range(n\_coun):  
 country.remove('')  
  
 total\_cases = w.column\_data(3)  
 n\_total\_cases = total\_cases.count('')  
 for i in range(n\_total\_cases):  
 total\_cases.remove('')  
  
 total\_deaths = w.column\_data(5)  
 n\_total\_deaths = total\_deaths.count('')  
 for i in range(n\_total\_deaths):  
 total\_deaths.remove('')  
  
 total\_recover = w.column\_data(7)  
 n\_total\_recover = total\_recover.count('')  
 for i in range(n\_total\_recover):  
 total\_recover.remove('')  
  
 total\_active = w.column\_data(8)  
 n\_total\_active = total\_active.count('')  
 for i in range(n\_total\_active):  
 total\_active.remove('')  
  
 for k in countries:  
  
 index\_coun = country.index(k)  
  
 # Variables:  
  
 cases = total\_cases[index\_coun]  
 deaths = total\_deaths[index\_coun]  
 recovered = total\_recover[index\_coun]  
 active = total\_active[index\_coun]  
 print(cases,deaths,recovered,active)  
  
  
 msg = f'''Subject: Coronavirus\n\n Hello {data[countries.index(k)][1]}! Here is today's status.\n  
 >>Country Name: {k}\n  
 >>Total Cases: {cases}\n  
 >>Total Deaths: {deaths}\n  
 >>Total Recovered: {recovered}\n  
 >>Active Cases: {active}\n\n\n  
 >>Today's tip\n{Functions.getTip()}\n  
 '''  
 print(msg)  
  
 Functions.mail(data[countries.index(k)][2], msg)  
  
  
 driver.close()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 unittest.main()

**Some Output Emails**





## Acknowledgements

I would like to take this opportunity to express my deep sense of gratitude to all those people without whom this project could never have been completed. First and foremost, I would like to thank my parents for their inexhaustible source of inspiration.

I would like to extend my gratitude to Mr. Kunal Gupta, Principal, Bal Bhavan International School, Dwarka for his constant encouragement and moral support, without which I would have never been able to give in my best.

I would like to thank Ms. Reema Chaudhary, PGT Computer Science for her keen interest in the work, ever useful practical knowledge and her kind supervision.

Their guidance and supervision were very helpful in bringing this work to conclusion.

[Name]