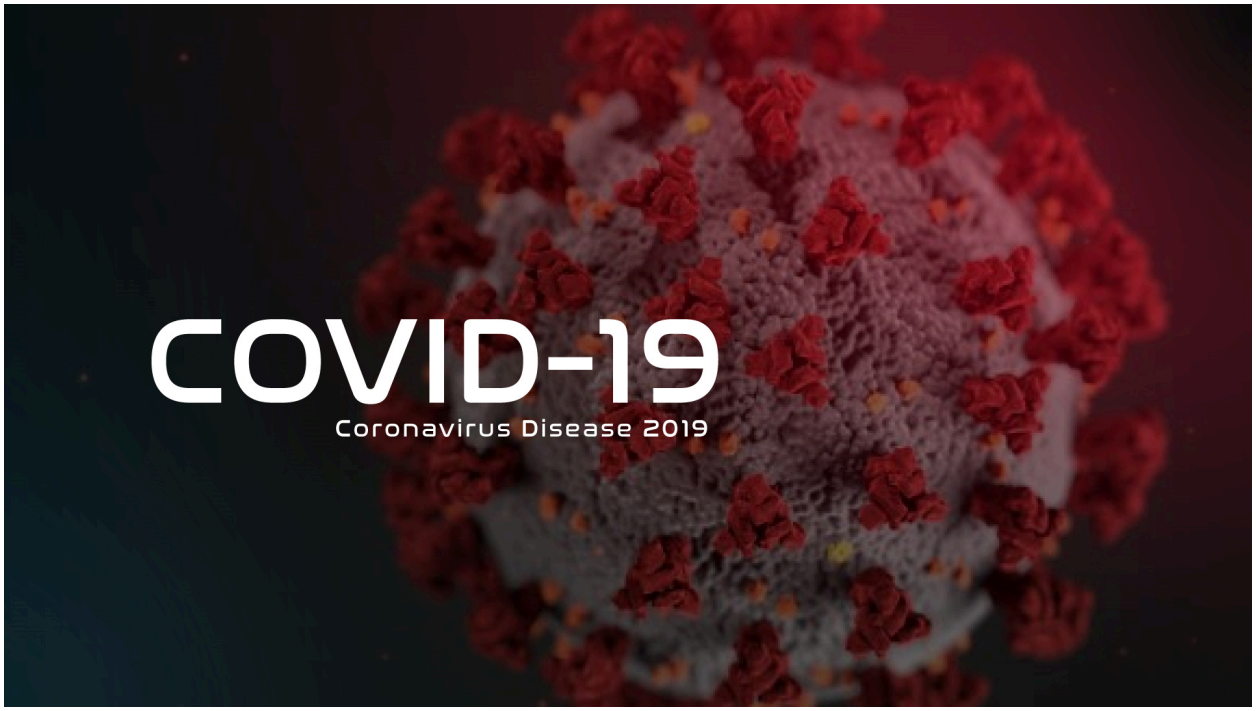


C-TAP:

COVID-19 Tracking and Analysing Platform

School name : Indira national school

Student name : Saniya Jain

CBSE Rollno. :

Class : XII Sci Alpha

Academic year : 2020 - 2021

Indira National School, Tathawade

CERTIFICATE

This is to certify that the project prepared by

Miss Saniya Jain

of **Class XII** entitled 'C-TAP: COVID-19 Tracking and Analysing Platform' for CBSE Board, All India Senior School Certificate Examination for the academic year **2017 – 2018,** for **Computer Science** subject at Indira National School, Tathawade has been examined and the report is found worthy of acceptance.

Principal's Signature	External Examiner	Internal Examiner

Date: School Stamp:

ACKNOWLEDGEMENT

I express my special thanks of gratitude to my teacher Ms. Aradhana Mundhe ma'am who gave me a golden opportunity to make such an interesting program and work on this wonderful project on the topic C-TAP: COVID - 19 Tracking And Analysing Platform.

It has given me a reason to research and dwell on the effect of COVID - 19 and dwell on its statistics.

I would like to thank Indira National School for encouraging me to gain such knowledge by providing an appropriate chance and infrastructure.

I would also like to thank the CBSE board for allowing me to act on my interests.

Thank you.

Saniya Jain

Indira National School

TABLE OF CONTENTS

WORKING DESCRIPTION

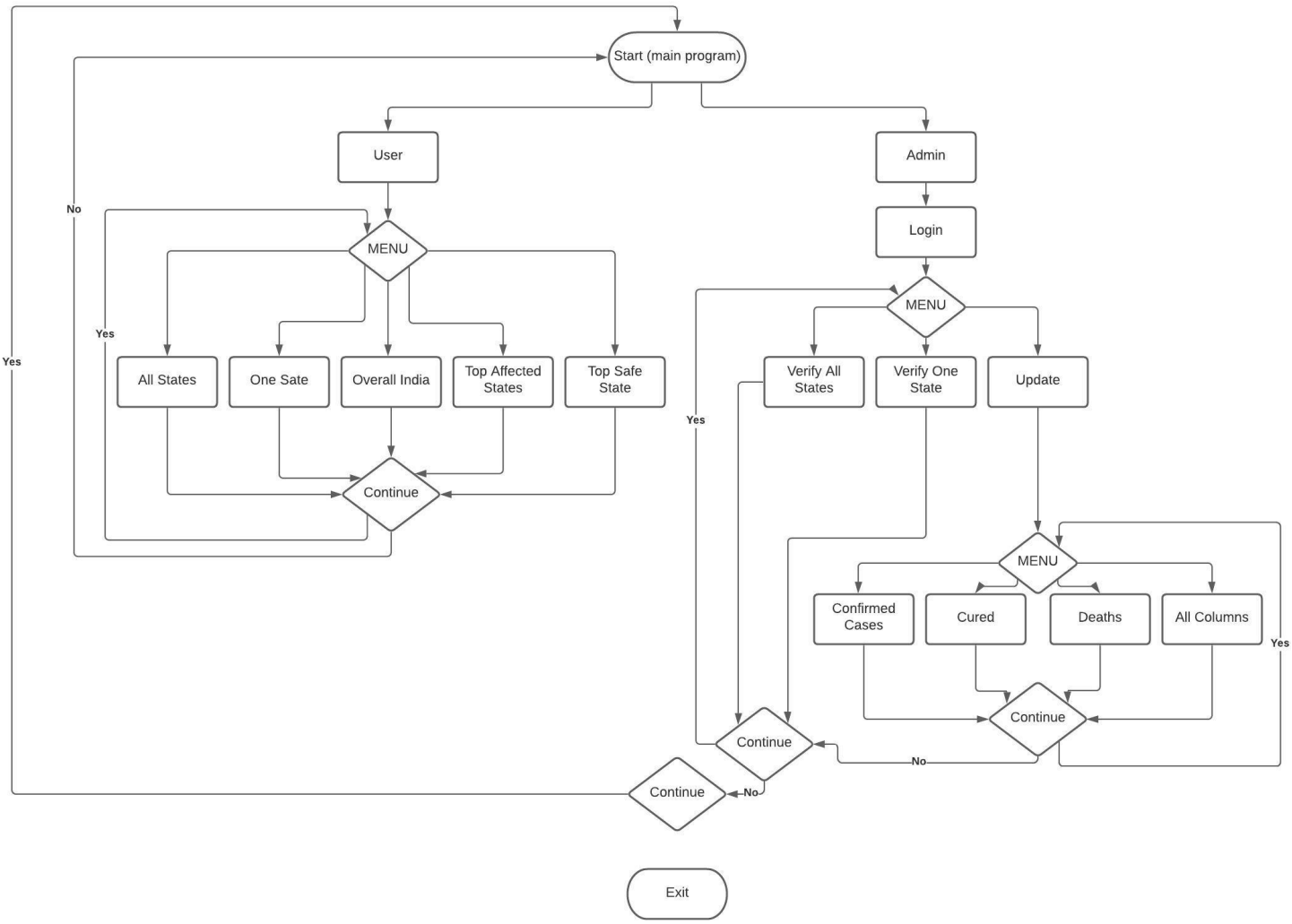
In a time like the COVID-19 Pandemic, it is important that people have a platform to keep a track of it.

This project is a software that provides a COVID-19 tracking and analyzing platform. It has two types of logins, user login, and admin login.

In the user login, there is no Id and password necessary, it is available to all in a formatted form. One can access the data that is available in my database using the menu. The functionalities can be used for as long as the user wishes.

The admin perspective program is secured by password. It can be used to verify the information, of the data can be updated. The functionalities can be used for as long as the admin wishes.

FLOWCHARTS


Start (main program)
User
Admin
Login
MENU
All States
One Sate
Overall India
Top Affected States
Top Safe State
Continue
No
Yes
MENU
Verify All States
Verify One State
Update
Yes
MENU
Confirmed Cases
Cured
Deaths
All Columns
Yes
Continue
Continue
No
Continue
No
Exit

LIBRARIES IMPORTED

1. Prettytable: PrettyTable function is imported from this module for the creation of tables to make the printing of substitutions allotment presentable

2. Mysql.connector : Mysql.connector used to have database connectivity of python with MySql .cursor() function used to create cursor objects to help execute queries and report the changes made to the database.

DATABASE USED

Database name: saniya

Tables present:-

1. Admin
2. COVID

DATABASE TABLE CODE

```
CREATE TABLE admin
(ID VARCHAR(50),
PASSWORD VARCHAR(50));

INSERT INTO admin
VALUES ('Arohi Gupta','abc123'),
('Saniya Jain','abcd1234');

CREATE TABLE COVID
(SNO INT(2),
STATE VARCHAR(50),
CONFIRMED_CASES INT(10),
CURED INT(10),
DEATH INT(10),
DATETIME TIMESTAMP DEFAULT CURRENT_TIMESTAMP);

INSERT INTO COVID
VALUES ( 1 ,' Andaman and Nicobar Islands ', 149 , 77 , 0 ),
( 2 ,' Andhra Pradesh ', 22259 , 11101 , 264 ),
( 3 ,' Arunachal Pradesh ', 287 , 109 , 2 ),
( 4 ,' Assam ', 13336 , 8729 , 16 ),
( 5 ,' Bihar ', 13944 , 9816 , 115 ),
( 6 ,' Chandigarh ', 513 , 402 , 7 ),
( 7 ,' Chhattisgarh ', 3525 , 2835 , 14 ),
( 8 ,' Dadra and Nagar Haveli and Daman and Diu ', 408 , 184 , 0 ),
( 9 ,' Delhi ', 107051 , 82226 , 15 ),
( 10 ,' Goa ', 2151 , 1273 , 9 ),
```

(11 ,' Gujarat ', 38333 , 27289 , 1993),
(12 ,' Haryana ', 18690 , 14106 , 282),
(13 ,' Himachal Pradesh ', 1101 , 833 , 11),
(14 ,' Jammu and Kashmir ', 9261 , 5567 , 149),
(15 ,' Jharkhand ', 3096 , 2170 , 22),
(16 ,' Karnataka ', 28877 , 11876 , 470),
(17 ,' Kerala ', 6195 , 3559 , 27),
(18 ,' Ladakh ', 1041 , 836 , 1),
(19 ,' Madhya Pradesh ', 16036 , 11987 , 629),
(20 ,' Maharashtra ', 230599 , 127259 , 9667),
(21 ,' Manipur ', 1435 , 793 , 0),
(22 ,' Meghalaya ', 80 , 43 , 1),
(23 ,' Mizoram ', 197 , 133 , 0),
(24 ,' Nagaland ', 657 , 304 , 0),
(25 ,' Odisha ', 10624 , 7006 , 48),
(26 ,' Puducherry ', 1008 , 480 , 14),
(27 ,' Punjab ', 6907 , 4828 , 178),
(28 ,' Rajasthan ', 22063 , 16866 , 482),
(29 ,' Sikkim ', 133 , 71 , 0),
(30 ,' Tamil Nadu ', 122350 , 74167 , 1700),
(31 ,' Telangana ', 29536 , 17279 , 324),
(32 ,' Tripura ', 1761 , 1324 , 1),
(33 ,' Uttarakhand ', 3258 , 2650 , 46),
(34 ,' Uttar Pradesh ', 31156 , 20331 , 845),
(35 ,' West Bengal ', 24823 , 16291 , 827),
(36 ,' Lakshadweep ', 0 , 0 , 0)

DATABASE TABLES

TABLE: Admin

ID	PASSWORD
Arohi Gupta	abc123
Saniya Jain	abcd1234

TABLE: COVID

SNO	STATE	CONFIRMED_CASES	CURED	DEATH	DATETIME
1	Andaman and Nicobar Islands	4773	4631	61	2020-12-08 21:49:50
2	Andhra Pradesh	872288	859624	7038	2020-12-08 21:50:19
3	Arunachal Pradesh	16395	15605	55	2020-12-07 20:27:49
4	Assam	213759	209214	993	2020-12-07 20:29:12
5	Bihar	238648	231884	1297	2020-12-08 21:29:17
6	Chandigarh	18113	16899	293	2020-12-08 21:30:11
7	Chhattisgarh	248232	225633	3010	2020-12-08 21:31:22
8	Dadra and Nagar Haveli and Daman and Diu	3346	3328	2	2020-12-08 21:31:56
9	Delhi	593924	561732	9706	2020-12-08 21:32:23
10	Goa	48776	46778	701	2020-12-08 21:33:03
11	Gujarat	220168	201580	4095	2020-12-08 21:34:15
12	Haryana	245288	230551	2611	2020-12-08 21:34:38
13	Himachal Pradesh	45697	37063	739	2020-12-08 21:35:04
14	Jammu and Kashmir	113568	106758	1755	2020-12-08 21:35:40
15	Jharkhand	110457	107710	988	2020-12-08 21:36:01
16	Karnataka	894004	857351	11867	2020-12-08 21:36:28
17	Kerala	639664	577616	2441	2020-12-08 21:36:50
18	Ladakh	8896	8014	121	2020-12-08 21:37:15

19	Madhya Pradesh	215957	199167	3347	2020-12-08 21:39:47
20	Maharashtra	1855341	1730715	47774	2020-12-08 21:37:57
21	Manipur	26225	22997	309	2020-12-08 21:38:19
22	Meghalaya	12314	11573	120	2020-12-08 21:38:47
23	Mizoram	3968	3757	6	2020-12-08 21:39:26
24	Nagaland	11418	10773	66	2020-12-08 21:40:08
25	Odisha	321564	316447	1778	2020-12-08 21:40:34
26	Puducherry	37270	36263	615	2020-12-08 21:41:00
27	Punjab	156839	144301	4934	2020-12-08 21:41:22
28	Rajasthan	282512	258393	2448	2020-12-08 21:41:43
29	Sikkim	5203	4715	117	2020-12-08 21:42:10
30	Tamil Nadu	791552	769048	11809	2020-12-08 21:42:36
31	Telangana	29536	17279	324	2020-07-09 21:34:20
32	Tripura	32925	32125	373	2020-12-08 21:43:18
33	Uttarakhand	78509	71980	1295	2020-12-08 21:44:03
34	Uttar Pradesh	31156	20331	845	2020-07-09 21:35:27
35	West Bengal	505054	472454	8771	2020-12-08 21:44:41
36	Lakshadweep	0	0	0	2020-12-08 21:37:33

SOURCE CODE

Main Program

```
import mysql.connector as msc

mycon = msc.connect(host ='localhost',user='root',passwd = 'root0509',database = 'saniya')

if mycon.is_connected():
```

```
    print('Connected to C-TAP: COVID-19 TRACKING AND ANALYSING PLATFORM',"\n')
else:
    print('Error')

mycur = mycon.cursor()

#-----------------------------------------MENU-----------------------------------------------

cch = 'y'
while cch == 'y':

     print('================================================================================
     ===',"\n')
    print('----------MENU---------')
    print('Login as User    :   1')
    print('Login as Admin   :   2')
    print('Exit             :   3')
    print('-----------------------')
    ch = int(input('Enter your choice : '))
    print()

#-------------------------------------LOGIN AS USER-------------------------------------------
```

```
    if ch == 1:
        from PROJECT import user_project    # The library project has the module User_project

#-------------------------------------LOGIN AS ADMIN-------------------------------------

    elif ch == 2:
        query = 'SELECT * FROM ADMIN'
        mycur.execute(query)
        res = mycur.fetchall()              # fetching all login details

      #LOGIN ID
        id_ = input('Enter your Login ID: ')
        l = []
        for i in res:
            l.append(i[0])
        while id_ not in l :
            print('Incorrect Login ID! ')
            id_ = input('Enter your Login ID: ')

      #LOGON PASSWORD
        pswd = input('Enter your login password: ')
        query = 'SELECT PASSWORD FROM ADMIN WHERE ID = "{}"'.format(id_)
```

```
        mycur.execute(query)
        res = mycur.fetchone()
        n = 2

        if res[0] == pswd:
            from PROJECT import admin_project   # library PROJECT contains module admin_project

        while res[0] != pswd and n!=0:          # condition for only 3 password attempts
            print('Incorrect Password! You will get',n,'more attempt(s).')
            pswd = input('Enter your login password: ')
            n -= 1
        else:
            if res[0] != pswd and n == 0:
                print('Too many attempts! Acess Denied.',"\n')



#-----------------------------------------EXIT-----------------------------------------

    elif ch == 3:
        cch = 'n'
```

```
        print('Thank You for using C-TAP!')
        print('Quitting!')
```

User_Project

```
#-------------------------------------DATE FUNCTION-----------------------------------
def date1(month):
    query = 'SELECT DATETIME FROM COVID WHERE STATE = "{}"'.format(month)
    mycur.execute(query)
    res = mycur.fetchone()
    print()
```

```
    print ('Last updated: ',res[0] , '\n')
    return

#----------------------------------MORTALITY RATE FUNCTION------------------------------
def mortality_rate(i, j):
    if i == 0:
        return 0
    else:
        rate = ( j / i) * 100
        return rate

#-----------------------------------RECOVERY RATE FUNCTION--------------------------------
def recovery_rate(i,j):
    if i == 0:
        return 0
    else:
        rate = ( j / i ) * 100
        return rate

#---------------------------------------STATE FUNCTION-----------------------------------
def info_state(name):
    query = 'SELECT * FROM COVID WHERE STATE  = "{}"'.format(name)
```

```
    mycur.execute(query)
    res = mycur.fetchone()
    return res

#----------------------------------------SUM FUNCTION-----------------------------------
def sum1(a)  :
    query = ' SELECT SUM({}) FROM COVID'.format(a)
    mycur.execute(query)
    res = mycur.fetchone()
    return res

#----------------------------------TOP AFFECTED FUNCTION--------------------------------
def max1(res):
    l1 = []
    l2 = []
    for i in range(1,4):
        max1 = max(res)
        l1.append(int(max(res)[0]))
        res.remove(max1)
        query = 'SELECT STATE FROM COVID WHERE CONFIRMED_CASES = {}'.format(max1[0])
        mycur.execute(query)
        a = mycur.fetchone()
```

```
        l2.append(a[0])
        print(i," , a[0])
    graph(l1,l2)
    return

#------------------------------TOP SAFEST FUNCTION--------------------------
def min1(res):
    l1 = []
    l2 = []
    for i in range(1,4):
        min1 = min(res)
        l1.append(int(min(res)[0]))
        res.remove(min1)
        query = 'SELECT STATE FROM COVID WHERE CONFIRMED_CASES = {}'.format(min1[0])
        mycur.execute(query)
        a = mycur.fetchone()
        l2.append(a[0])
        print(i," , a[0])
    graph(l1,l2)
    return

#---------------------------------------SPACING---------------------------------------
```

```
def spacing(i,n,x):
    b = n - len(str(i[x]))
    s = ' '
    for j in range(1,b + 1):
      s+=' '
    return s

#----------------------------------------GRAPH---------------------------------------
def graph(l1,l2):
    plt.pie(l1,labels = l2,explode = [0.025,0.025,0.025])
    plt.show()
    return

#---------------------------------------MAIN FUNCTION-------------------------------


#__main__
import matplotlib.pyplot as plt
from prettytable import PrettyTable
import mysql.connector as msc
mycon = msc.connect(host ='localhost',user='root',passwd = 'root0509',database = 'saniya')
if mycon.is_connected():
```

```
    print('==========================================================================
    ===','\n')
  print('WELCOME USER','\n')
else:
  print('Error')

mycur = mycon.cursor()

#-----------------------------------------MENU-------------------------------------


cch = 'y'
while cch == 'y':

    print('==========================================================================
    ===','\n')
  print('--------------MENU---------------')
  print('Information About All States : 1')
  print('Information About One State  : 2')
  print('Overall India                : 3')
  print('Top Affected States          : 4')
  print('Top Safe States              : 5')
```

```
    print('--------------------------------')
    ch = int(input('Enter your choice (1 to 5): '))
    print()

#---------------------------------------VALIDATION---------------------------------------
    while ch >= 6 or ch <= 0:                   # condition for invalid choice
        print('Error! Enter valid choice.')
        ch = int(input('Enter your choice: '))
        print()

#-------------------------------------ALL STATES-----------------------------------------
    if ch == 1:
        query = 'SELECT * FROM COVID'
        mycur.execute(query)
        res = mycur.fetchall()
        t = PrettyTable(['Sno', 'State','Confirmed_cases' , 'Cured ' , 'Death' , 'Last_updated'])
        for Sno, State,Confirmed_cases , Cured  , Death , Datetime in res:
            t.add_row([Sno, State,Confirmed_cases , Cured  , Death , Datetime])
        print(t)

#-------------------------------------ONE STATE------------------------------------------
    elif ch == 2:
```

```
        name = input('Enter name of state: ')
        date1(name)
        res = info_state(name)
        mr = mortality_rate(res[2],res[4])
        rr = recovery_rate(res[2],res[3])
        print('State           :' , res[1])
        print('Total Cases     :',res[2])
        print('Cured           :',res[3] )
        print('Deaths          :', res[4] )
        print('Mortality Rate  :' , round(mr , 2) , '%' )
        print('Recovery Rate   :' ,round(rr , 2) , '%' )

#-------------------------------------OVERALL INDIA-------------------------------------
    elif ch == 3:
        l = []
        for a in ('Confirmed_cases' , 'Cured', 'Death'):
            res = sum1(a)
            s = spacing(a.split(), 15 , 0)
            print(a,s,': ' , res[0])
            l.append(res)
        mr = mortality_rate(l[0][0],l[2][0])
        rr = recovery_rate(l[0][0],l[1][0])
```

```
        print('Mortality Rate    : ' , round(mr , 2) , '%' )
        print('Recovery Rate     : ' ,round(rr , 2) , '%' )

#-------------------------------------TOP AFFECTED-------------------------------------
    elif ch == 4:
        query = 'SELECT CONFIRMED_CASES FROM COVID'
        mycur.execute(query)
        res = mycur.fetchall()
        max1(res)

#-------------------------------------TOP SAFEST---------------------------------------
    elif ch == 5:
        query = 'SELECT CONFIRMED_CASES FROM COVID'
        mycur.execute(query)
        res = mycur.fetchall()
        min1(res)

#-------------------------------------CONTINUING---------------------------------------
    print()
    cch = input('Do you wish to continue as a user? (y/n): ')
    print()
    if cch != 'y':
```

```
    print()
    print('Exiting')
    print('Thank You for using our services!'"\n')
```

Admin_Project

```
#------------------------------DATE FUNCTION-------------------------------------
def date1(state):
    query = 'SELECT DATETIME FROM COVID WHERE STATE = "{}"'.format(state)
    mycur.execute(query)
    res = mycur.fetchone()
```

```
    print()
    print ('Last updated: ',res[0] , '\n')
    return

#-----------------------------------MORTALITY RATE FUNCTION-------------------------------
def mortality_rate(i, j):
    if i == 0:
        return 0
    else:
        rate = ( j / i) * 100
        return rate

#-----------------------------------RECOVERY RATE FUNCTION--------------------------------
def recovery_rate(i,j):
    if i == 0:
        return 0
    else:
        rate = ( j / i ) * 100
    return rate

#-----------------------------------------SPACING-----------------------------------------
def spacing(i,n,x):
```

```
    b = n - len(str(i[x]))
    s = ' '
    for j in range(1,b + 1):
      s+=' '
    return s

#---------------------------------------STATE FUNCTION----------------------------------
def info_state(name):
    query = 'SELECT * FROM COVID WHERE STATE  = "{}"'.format(name)
    mycur.execute(query)
    res = mycur.fetchone()
    return res

#---------------------------------------UPDATE FUNCTION----------------------------------
def update(name , clm):
    n = int(input('Enter new info: '))
    query1 = 'UPDATE COVID SET {} = {} WHERE STATE = "{}"'.format(clm , n , name)
    query2 = 'UPDATE COVID SET DATETIME = CURRENT_TIMESTAMP WHERE STATE =
      "{}"'.format(name)
    mycur.execute(query1)
    mycon.commit()
    mycur.execute(query2)
```

```
    mycon.commit()
    return 'Done'

#-----------------------------------UPDATE ALL FUNCTION-------------------------------
def update_all(name , clm):
    for a in clm:
        print(a)
        n = int(input('Enter new info: '))
        query1 = 'UPDATE COVID SET {} = {} WHERE STATE = "{}"'.format(a , n , name)
        query2 = 'UPDATE COVID SET DATETIME = CURRENT_TIMESTAMP WHERE STATE =
      "{}"'.format(name)
        mycur.execute(query1)
        mycon.commit()
        mycur.execute(query2)
        mycon.commit()
    return 'Done'

#---------------------------------------MAIN FUNCTION-----------------------------------
#__main__
import mysql.connector as msc
from prettytable import PrettyTable
mycon = msc.connect(host ='localhost',user='root',passwd = 'root0509',database = 'saniya')
```

```
if mycon.is_connected():

    print('================================================================================
    ===','\n')
  print('WELCOME ADMIN!','\n')
else:
  print('Error')


mycur = mycon.cursor()

#-----------------------------------------MENU---------------------------------------
cch = 'y'
while cch == 'y':

    print('================================================================================
    ===','\n')
  print('----------------MENU----------------')
  print('Verify Information of All States : 1')
  print('Verify Information of One State  : 2')
  print('Update Information               : 3')
  print('----------------------------------')
  ch = int(input('Enter your choice (1 to 3): '))
  print()
```

```
#---------------------------------------VALIDATION---------------------------------------
    while ch >= 4 or ch <= 0:                  # condition for invalid choice
        print('Error! Enter valid choice.')
        ch = int(input('Enter your choice: '))
        print()

#-------------------------------------VERIFY ALL STATES-----------------------------------------
    if ch == 1:
        query = 'SELECT * FROM COVID'
        mycur.execute(query)
        res = mycur.fetchall()
        t = PrettyTable(['Sno', 'State','Confirmed_cases' , 'Cured ' , 'Death' , 'Last_updated'])
        for Sno, State,Confirmed_cases , Cured  , Death , Datetime in res:
            t.add_row([Sno, State,Confirmed_cases , Cured  , Death , Datetime])
        print(t)

#-------------------------------------VERIFY ONE STATE-----------------------------------------
    elif ch == 2:
        name = input('Enter name of state: ')
        date1(name)
        res = info_state(name)
```

```
        mr = mortality_rate(res[2],res[4])
        rr = recovery_rate(res[2],res[3])
        print('State           :' , res[1])
        print('Total Cases     :',res[2])
        print('Cured           :',res[3] )
        print('Deaths          :', res[4] )
        print('Mortality Rate  :' , round(mr , 2) , '%' )
        print('Recovery Rate   :' ,round(rr , 2) , '%' )

#-----------------------------------------UPDATE-----------------------------------------
    elif ch == 3:
        uch = 'y'
        while uch == 'y':

     print('================================================================================
     ===','\n')
            name = input('Enter name of state: ')
            print()
            print('---------------MENU---------------')
            print('Update No. of Confirmed Cases : 1')
            print('Update No. of Cured           : 2')
            print('Update No. of Deaths          : 3')
            print('Update All Columns            : 4')
```

```
print('----------------------------------','\n')
ch = int(input('Enter your choice (1 to 4): '))
print()

while ch >= 5 or ch <= 0:                # condition for invalid choice
    print('Error! Enter valid choice.','\n')
    ch = int(input('Enter your choice: '))
    print()
if ch == 1:
    clm = 'Confirmed_Cases'
    res = update(name , clm)
    print(res)
elif ch == 2:
    clm = 'Cured'
    res = update(name , clm)
    print(res)
elif ch == 3:
    clm = 'Death'
    res = update(name , clm)
    print(res)
elif ch == 4:
    clm = ('Confirmed_Cases','Cured','Death')
```

```
            res = update_all(name , clm)


        print()
        uch = input('Do you wish to continue updating? (y/n): ')
        print()

#---------------------------------------CONTINUING---------------------------------------
    print()
    cch = input('Do you wish continue as an admin? (y/n): ')
    print()
    if cch != 'y':
        print('Exiting')
        print('Thank You for using our services!')
```

OUTPUT

```
Connected to C-TAP: COVID-19 TRACKING AND ANALYSING PLATFORM

==========================================================================
```

----------MENU---------

Login as User : 1

Login as Admin : 2

Exit : 3

Enter your choice : 1

==========

WELCOME USER

==========

----------MENU----------

Information About All States : 1

Information About One State : 2

Overall India : 3

Top Affected States : 4

Top Safe States : 5

Enter your choice (1 to 5): 1

Sno	State	Confirmed_cases	Cured	Death	Last_updated
1	Andaman and Nicobar Islands	4758	4624	61	2020-12-07 20:25:23
2	Andhra Pradesh	871972	859029	7033	2020-12-07 20:21:13
3	Arunachal Pradesh	16395	15605	55	2020-12-07 20:27:49
4	Assam	213759	209214	993	2020-12-07 20:29:12
5	Bihar	238648	231884	1297	2020-12-08 21:29:17
6	Chandigarh	18113	16899	293	2020-12-08 21:30:11
7	Chhattisgarh	248232	225633	3010	2020-12-08 21:31:22
8	Dadra and Nagar Haveli and Daman and Diu	3346	3328	2	2020-12-08 21:31:56
9	Delhi	593924	561732	9706	2020-12-08 21:32:23
10	Goa	48776	46778	701	2020-12-08 21:33:03
11	Gujarat	220168	201580	4095	2020-12-08 21:34:15
12	Haryana	245288	230551	2611	2020-12-08 21:34:38
13	Himachal Pradesh	45697	37063	739	2020-12-08 21:35:04
14	Jammu and Kashmir	113568	106758	1755	2020-12-08 21:35:40
15	Jharkhand	110457	107710	988	2020-12-08 21:36:01
16	Karnataka	894004	857351	11867	2020-12-08 21:36:28
17	Kerala	639664	577616	2441	2020-12-08 21:36:50
18	Ladakh	8896	8014	121	2020-12-08 21:37:15
19	Madhya Pradesh	215957	199167	3347	2020-12-08 21:39:47
20	Maharashtra	1855341	1730715	47774	2020-12-08 21:37:57
21	Manipur	26225	22997	309	2020-12-08 21:38:19
22	Meghalaya	12314	11573	120	2020-12-08 21:38:47
23	Mizoram	3968	3757	6	2020-12-08 21:39:26
24	Nagaland	11418	10773	66	2020-12-08 21:40:08

25	Odisha	321564	316447	1778	2020-12-08 21:40:34
26	Puducherry	37270	36263	615	2020-12-08 21:41:00
27	Punjab	156839	144301	4934	2020-12-08 21:41:22
28	Rajasthan	282512	258393	2448	2020-12-08 21:41:43
29	Sikkim	5203	4715	117	2020-12-08 21:42:10
30	Tamil Nadu	791552	769048	11809	2020-12-08 21:42:36
31	Telangana	29536	17279	324	2020-07-09 21:34:20
32	Tripura	32925	32125	373	2020-12-08 21:43:18
33	Uttarakhand	78509	71980	1295	2020-12-08 21:44:03
34	Uttar Pradesh	31156	20331	845	2020-07-09 21:35:27
35	West Bengal	505054	472454	8771	2020-12-08 21:44:41
36	Lakshadweep	0	0	0	2020-12-08 21:37:33

Do you wish to continue as a user? (y/n): y

==========

----------MENU----------

Information About All States : 1

Information About One State : 2

Overall India : 3

Top Affected States : 4

Top Safe States : 5

Enter your choice (1 to 5): 2

Enter name of state: Maharashtra

Last updated: 2020-12-08 21:37:57

State : Maharashtra

Total Cases : 1855341

Cured : 1730715

Deaths : 47774

Mortality Rate : 2.57 %

Recovery Rate : 93.28 %

Do you wish to continue as a user? (y/n): y

==========

----------MENU----------

Information About All States : 1

Information About One State : 2

Overall India : 3

Top Affected States : 4

```
Top Safe States             : 5
--------------------------------
Enter your choice (1 to 5): 3

Confirmed_cases   :  8933008
Cured             :  8423687
Death             :  132699
Mortality Rate    :  1.49 %
Recovery Rate     :  94.30 %

Do you wish to continue as a user? (y/n): y

================================================================================

----------MENU----------
Information About All States : 1
Information About One State  : 2
Overall India                : 3
Top Affected States          : 4
Top Safe States              : 5
--------------------------------
Enter your choice (1 to 5): 4
```

1 . Maharashtra

2 . Karnataka

3 . Andhra Pradesh

Do you wish to continue as a user? (y/n): y

==========

----------MENU----------

Information About All States : 1

Information About One State : 2

Overall India : 3

Top Affected States : 4

Top Safe States : 5

Enter your choice (1 to 5): 5

1 . Lakshadweep

2 . Dadra and Nagar Haveli and Daman and Diu

3 . Mizoram

Do you wish to continue as a user? (y/n): n

Exiting

Thank You for using our services!

==========

----------MENU---------

Login as User : 1

Login as Admin : 2

Exit : 3

Enter your choice : 2

Enter your Login ID: Saniya Jain

Enter your login password: abcd1234

==========

WELCOME ADMIN!

==========

----------MENU----------

Verify Information of All States : 1

Verify Information of One State : 2

Update Information : 3

Enter your choice (1 to 3): 1

Sno	State	Confirmed_cases	Cured	Death	Last_updated
1	Andaman and Nicobar Islands	4758	4624	61	2020-12-07 20:25:23
2	Andhra Pradesh	871972	859029	7033	2020-12-07 20:21:13
3	Arunachal Pradesh	16395	15605	55	2020-12-07 20:27:49
4	Assam	213759	209214	993	2020-12-07 20:29:12
5	Bihar	238648	231884	1297	2020-12-08 21:29:17
6	Chandigarh	18113	16899	293	2020-12-08 21:30:11
7	Chhattisgarh	248232	225633	3010	2020-12-08 21:31:22
8	Dadra and Nagar Haveli and Daman and Diu	3346	3328	2	2020-12-08 21:31:56
9	Delhi	593924	561732	9706	2020-12-08 21:32:23
10	Goa	48776	46778	701	2020-12-08 21:33:03
11	Gujarat	220168	201580	4095	2020-12-08 21:34:15
12	Haryana	245288	230551	2611	2020-12-08 21:34:38

13	Himachal Pradesh	45697	37063	739	2020-12-08 21:35:04
14	Jammu and Kashmir	113568	106758	1755	2020-12-08 21:35:40
15	Jharkhand	110457	107710	988	2020-12-08 21:36:01
16	Karnataka	894004	857351	11867	2020-12-08 21:36:28
17	Kerala	639664	577616	2441	2020-12-08 21:36:50
18	Ladakh	8896	8014	121	2020-12-08 21:37:15
19	Madhya Pradesh	215957	199167	3347	2020-12-08 21:39:47
20	Maharashtra	1855341	1730715	47774	2020-12-08 21:37:57
21	Manipur	26225	22997	309	2020-12-08 21:38:19
22	Meghalaya	12314	11573	120	2020-12-08 21:38:47
23	Mizoram	3968	3757	6	2020-12-08 21:39:26
24	Nagaland	11418	10773	66	2020-12-08 21:40:08
25	Odisha	321564	316447	1778	2020-12-08 21:40:34
26	Puducherry	37270	36263	615	2020-12-08 21:41:00
27	Punjab	156839	144301	4934	2020-12-08 21:41:22
28	Rajasthan	282512	258393	2448	2020-12-08 21:41:43
29	Sikkim	5203	4715	117	2020-12-08 21:42:10
30	Tamil Nadu	791552	769048	11809	2020-12-08 21:42:36
31	Telangana	29536	17279	324	2020-07-09 21:34:20
32	Tripura	32925	32125	373	2020-12-08 21:43:18
33	Uttarakhand	78509	71980	1295	2020-12-08 21:44:03
34	Uttar Pradesh	31156	20331	845	2020-07-09 21:35:27
35	West Bengal	505054	472454	8771	2020-12-08 21:44:41

```
|  36 |              Lakshadweep               |        0        |    0    |   0    | 2020-12-08
    21:37:33 |
+-----+----------------------------------------+-----------------+---------+--------+-------------------
    --+
```

Do you wish continue as an admin? (y/n): y

==========

----------MENU----------

Verify Information of All States : 1

Verify Information of One State : 2

Update Information   : 3

Enter your choice (1 to 3): 2

Enter name of state: Goa

Last updated: 2020-12-08 21:33:03

State   : Goa

Total Cases   : 48776

Cured   : 46778

Deaths   : 701

Mortality Rate : 1.44 %

Recovery Rate   : 95.9 %

Do you wish continue as an admin? (y/n): y

==========

----------MENU----------

Verify Information of All States : 1

Verify Information of One State : 2

Update Information   : 3

Enter your choice (1 to 3): 3

==========

Enter name of state: Andaman and Nicobar Islands

----------MENU----------

Update No. of Confirmed Cases : 1

Update No. of Cured   : 2

Update No. of Deaths   : 3

Update All Columns : 4

Enter your choice (1 to 4): 1

Enter new info: 4773

Done

Do you wish to continue updating? (y/n): y

==========

Enter name of state: Andaman and Nicobar Islands

----------MENU----------

Update No. of Confirmed Cases : 1

Update No. of Cured : 2

Update No. of Deaths : 3

Update All Columns : 4

Enter your choice (1 to 4): 2

Enter new info: 4631

Done

Do you wish to continue updating? (y/n): y

==========

Enter name of state: Andaman and Nicobar Islands

----------MENU----------

Update No. of Confirmed Cases : 1

Update No. of Cured : 2

Update No. of Deaths : 3

Update All Columns : 4

Enter your choice (1 to 4): 3

Enter new info: 61

Done

Do you wish to continue updating? (y/n): y

==========

Enter name of state: Andhra Pradesh

----------MENU----------

Update No. of Confirmed Cases : 1

Update No. of Cured : 2

Update No. of Deaths : 3

Update All Columns : 4

Enter your choice (1 to 4): 4

Confirmed_Cases

Enter new info: 872288

Cured

Enter new info: 859624

Death

Enter new info: 7038

Do you wish to continue updating? (y/n): n

Do you wish to continue as an admin? (y/n): y

==========

----------MENU----------

Verify Information of All States : 1

Verify Information of One State : 2

Update Information : 3

Enter your choice (1 to 3): 1

Sno	State	Confirmed_cases	Cured	Death	Last_updated
1	Andaman and Nicobar Islands	4773	4631	61	2020-12-08 21:49:50
2	Andhra Pradesh	872288	859624	7038	2020-12-08 21:50:19
3	Arunachal Pradesh	16395	15605	55	2020-12-07 20:27:49
4	Assam	213759	209214	993	2020-12-07 20:29:12
5	Bihar	238648	231884	1297	2020-12-08 21:29:17
6	Chandigarh	18113	16899	293	2020-12-08 21:30:11

7	Chhattisgarh	248232	225633	3010	2020-12-08 21:31:22
8	Dadra and Nagar Haveli and Daman and Diu	3346	3328	2	2020-12-08 21:31:56
9	Delhi	593924	561732	9706	2020-12-08 21:32:23
10	Goa	48776	46778	701	2020-12-08 21:33:03
11	Gujarat	220168	201580	4095	2020-12-08 21:34:15
12	Haryana	245288	230551	2611	2020-12-08 21:34:38
13	Himachal Pradesh	45697	37063	739	2020-12-08 21:35:04
14	Jammu and Kashmir	113568	106758	1755	2020-12-08 21:35:40
15	Jharkhand	110457	107710	988	2020-12-08 21:36:01
16	Karnataka	894004	857351	11867	2020-12-08 21:36:28
17	Kerala	639664	577616	2441	2020-12-08 21:36:50
18	Ladakh	8896	8014	121	2020-12-08 21:37:15
19	Madhya Pradesh	215957	199167	3347	2020-12-08 21:39:47
20	Maharashtra	1855341	1730715	47774	2020-12-08 21:37:57
21	Manipur	26225	22997	309	2020-12-08 21:38:19
22	Meghalaya	12314	11573	120	2020-12-08 21:38:47
23	Mizoram	3968	3757	6	2020-12-08 21:39:26
24	Nagaland	11418	10773	66	2020-12-08 21:40:08
25	Odisha	321564	316447	1778	2020-12-08 21:40:34
26	Puducherry	37270	36263	615	2020-12-08 21:41:00
27	Punjab	156839	144301	4934	2020-12-08 21:41:22
28	Rajasthan	282512	258393	2448	2020-12-08 21:41:43
29	Sikkim	5203	4715	117	2020-12-08 21:42:10

```
|  30 |               Tamil Nadu               |      791552     |  769048 | 11809 | 2020-12-08
    21:42:36 |
|  31 |               Telangana                |      29536      |  17279  |  324  | 2020-07-09
    21:34:20 |
|  32 |                Tripura                 |      32925      |  32125  |  373  | 2020-12-08
    21:43:18 |
|  33 |              Uttarakhand               |      78509      |  71980  |  1295 | 2020-12-08
    21:44:03 |
|  34 |             Uttar Pradesh              |      31156      |  20331  |  845  | 2020-07-09
    21:35:27 |
|  35 |               West Bengal              |      505054     |  472454 |  8771 | 2020-12-08
    21:44:41 |
|  36 |               Lakshadweep              |        0        |    0    |   0   | 2020-12-08
    21:37:33 |
+-----+----------------------------------------+-----------------+---------+-------+------------------
    --+
```

Do you wish continue as an admin? (y/n): n

Exiting

Thank You for using our services!

==========

----------MENU---------

Login as User : 1

Login as Admin : 2

Exit : 3

Enter your choice : 3

Thank You for using C-TAP!

Quitting!

BIBLIOGRAPHY

1. Zedcode.com
2. Geeksforgeeks.org
3. Sumita Arora computer science textbook for class 11
4. Sumita Arora computer science textbook for class 12
5. python.org