****

**--------------Details Removed---------------**

****

**Abstract: -**

Health Issue Management System is a computerized system which stores and retrieves data associated with medical treatment. This project is aimed toward explaining the relevance and importance of DBMS in this system. it's projected towards enhancing the connection between doctors/patients and different sorts of health issues through the utilization of DBMS, and thereby making it convenient for the doctors to see differing types of health issues starting from diseases to injuries as once they require such, they will utilize this software.

This software has two parts. First is the doctor and the other is the health issue part. The doctor part is employed as a face and therefore the disease part is that the rear of this technique. This software utilises DBMS which allows the prevailing doctors to access database and allows new doctors to check in for online access.

This system allows the doctors to look for a patient’s medical record and to see about the various sorts of diseases and their symptoms and cures. This also allows the doctors to update the disease table whenever a new disease is discovered. This software also allows the doctors to update a patient’s medical record or add a new patient.

The main purpose of this software is to scale back the manual errors involved during this process and to make it convenient for the doctors to access the patient’s medical record and other details associated with various sorts of diseases.

**Introduction: -**

We believe that simplicity and power are the best combination. Keeping this in mind we have designed our Health Issue Management System.

During the present novel coronavirus outbreak, doctors and healthcare workers have become the saviours for humanity. They risk their lives every day by going to hospital and directly interacting with the patients. If they stop working humanity would cease to exists. That’s why we decided to create a Health Issue Management System which might help the doctors interact, diagnose and keep a record of their patients in an exceedingly contactless manner.

This Heath Issue Management System includes registration of patients, details of doctors and storing their details into the system. The software has the power to offer a singular id for each patient and stores the details of each patient and the staff automatically and can be accessed by doctors through their accounts.

It can be accessed employing a username and password by the doctor. They will also have the power to view and modify data into the database. The information is often accessed easily. The interface is extremely user-friendly, speedy and comfortable to use. The information is well protected for private use with encryption systems and backups.

This software will be often be helpful for both, multi - speciality hospitals, as to cover a good range of hospital administration and management processes, as well for clinics for safer interaction. It is powerful, flexible, and straightforward to use and is designed and developed to deliver real conceivable benefits to health care departments.

We believe that this will be a small step towards helping those who are the backbone of this human race and who risk their lives daily so that we stay safe.

**Project motive: -**

During the recent pandemic, it's important to limit oneself from manual pen and paper work and adopt a contactless way like a digital record of all patients. The central purpose of this software is to scale back the manual errors involved during this process and to make it convenient for the doctors to access the patient’s medical record and other details associated with various sorts of diseases. And as we all know doctor are working hard to continuously refine and advance their processes so as to enhance patient care, reduce costs, and improve the patient’s overall experience and satisfaction. Healthcare project management skills have become increasingly important to businesses, including the healthcare industry, because they assist to control costs, manage risk, and improve project outcomes. By applying project management techniques, organizations can systematically plan, organize, and execute a gaggle of tasks maximizing resources and achieving specific goals.

**Tables with Normalisation: -**

**Doctor table: -**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| d\_id | dnamedfirst | dnamedlast | password | speciality | shift | phone |

**Patient table: -**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| p\_id | Pfirst | pdlast | City | DOB | age | DOA | number |

**Virus table: -**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| p\_id | dname | vname | treatment | symptoms |

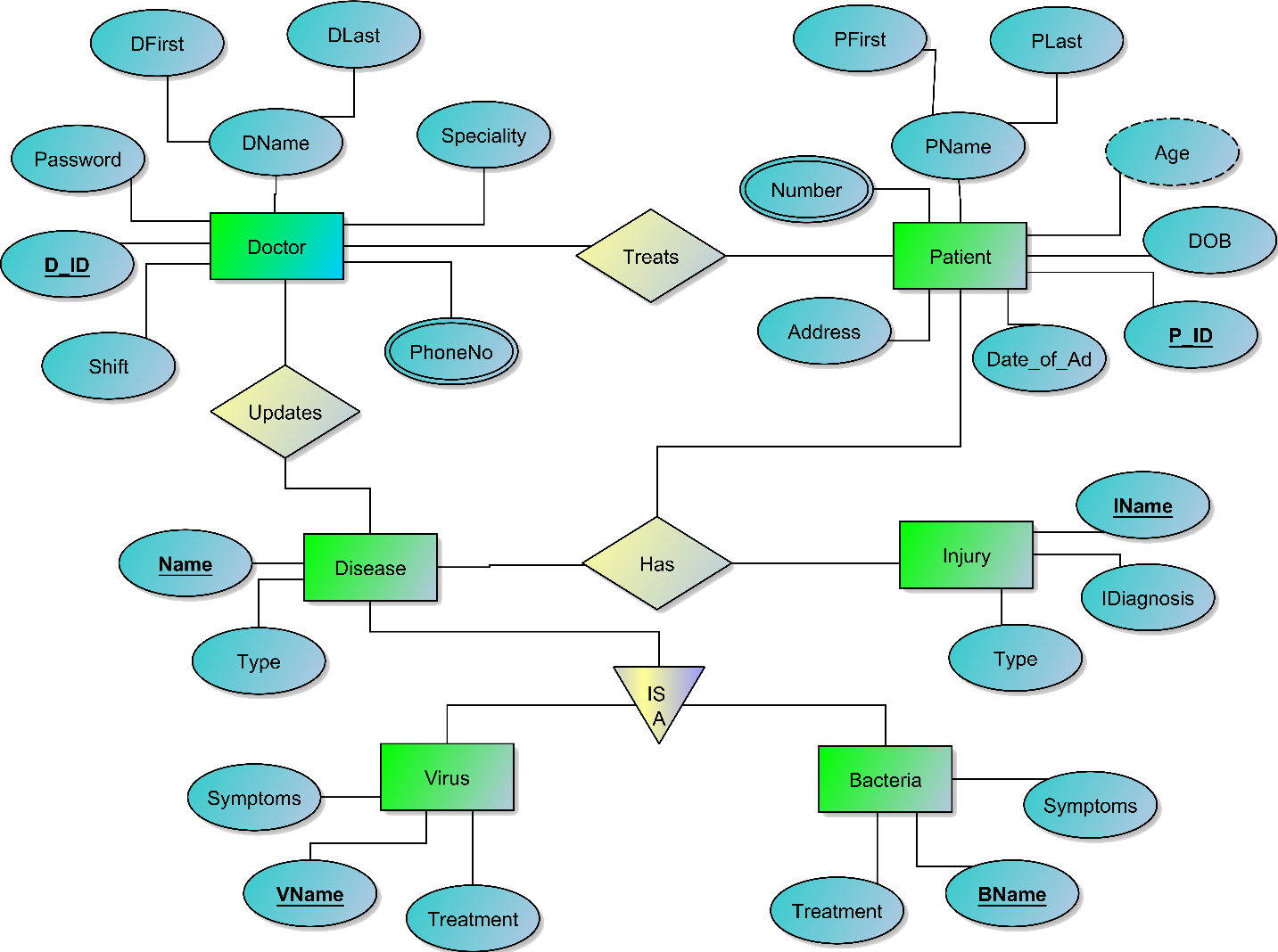
**Bacteria table: -**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| p\_id | dname | bname | treatment | symptoms |

**Injury table: -**

|  |  |  |  |
| --- | --- | --- | --- |
| p\_id | iname | idiagnosis | type |

**Database Design: -**



M

M

M

M

M

M

M

**Methodology with Modules Description: -**

1. **Registration module -**

In the registration module if the doctor already has an account then they log in into their account alternatively if it's a new doctor then they're asked to make a new account and to enter their details. After registering the doctors can view the medical record of existing patients or add new ones or delete old ones.

The doctors also can view diseases, their symptoms and diagnosis. The doctor either signs in into his/her existing account or he/she creates a new account.

1. **Patient module –**

The patient module consists of the medical record of all the patients. This will be accessed by the doctors and therefore the doctors also can add or delete records from this module.

Once they have entered the system, they will either view, add or delete a patient record by simply selecting their desired option and entering the patient’s ID.

1. **Disease/Injury module -**

This module is supposed for the doctors additionally. It allows them to ascertain different sorts of diseases, injuries, their symptoms and cures and the patient details with each of the diseases/injuries. This module also can be updated by the doctors.

1. **Administrative Module –**

This module is supposed for the executive officers wherein they will enter the override code and consider all of the info stored within the database.

Once they're through with their work and they press exit them the system automatically backups the entire data in two different locations. Thus, ensuring that the info isn't compromised.

**Minimum System Requirements: -**

1. **Hardware –**
2. Linux: RHEL 6/7, 64/32-bit (almost all libraries also work in Ubuntu)
3. 64/32-bit CPU (Intel / AMD architecture)
4. 2 GB RAM
5. 2 GB free disk space

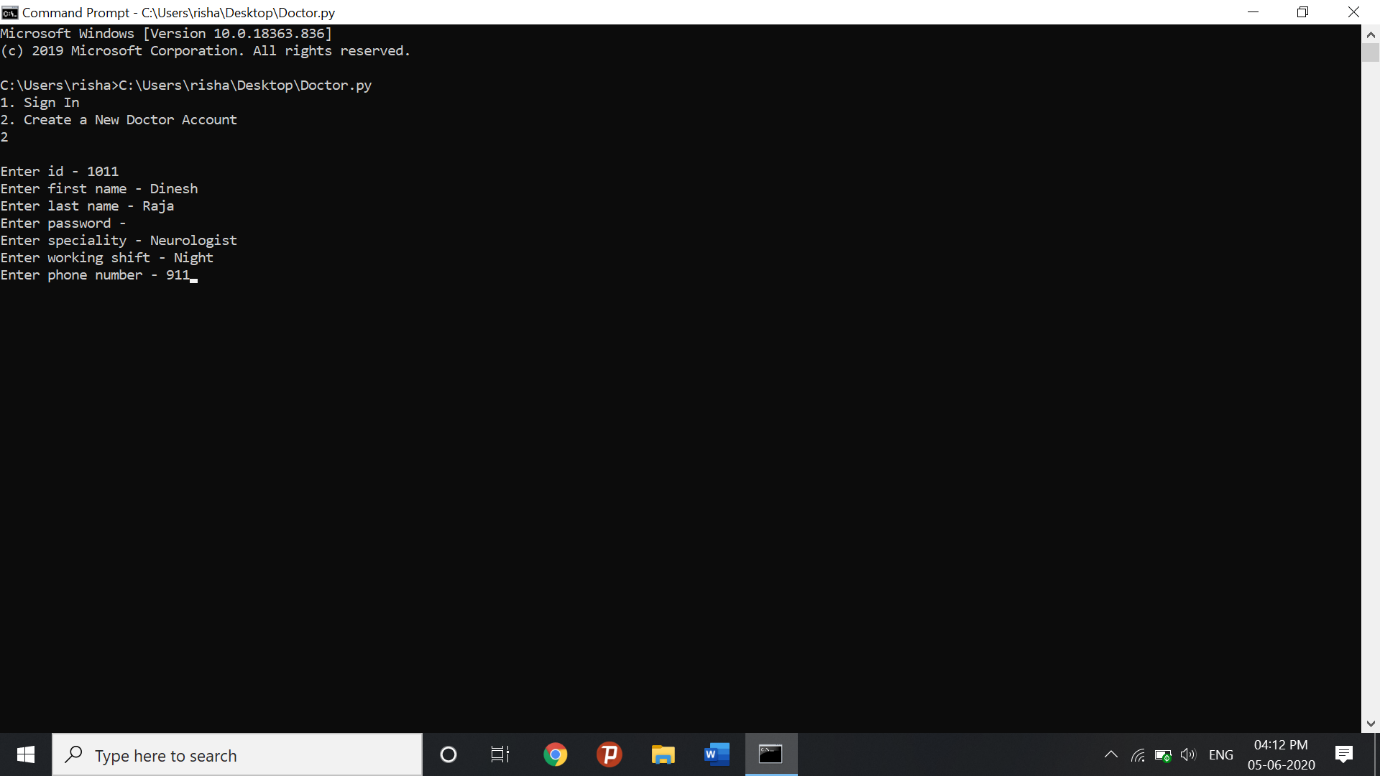
2. Software -

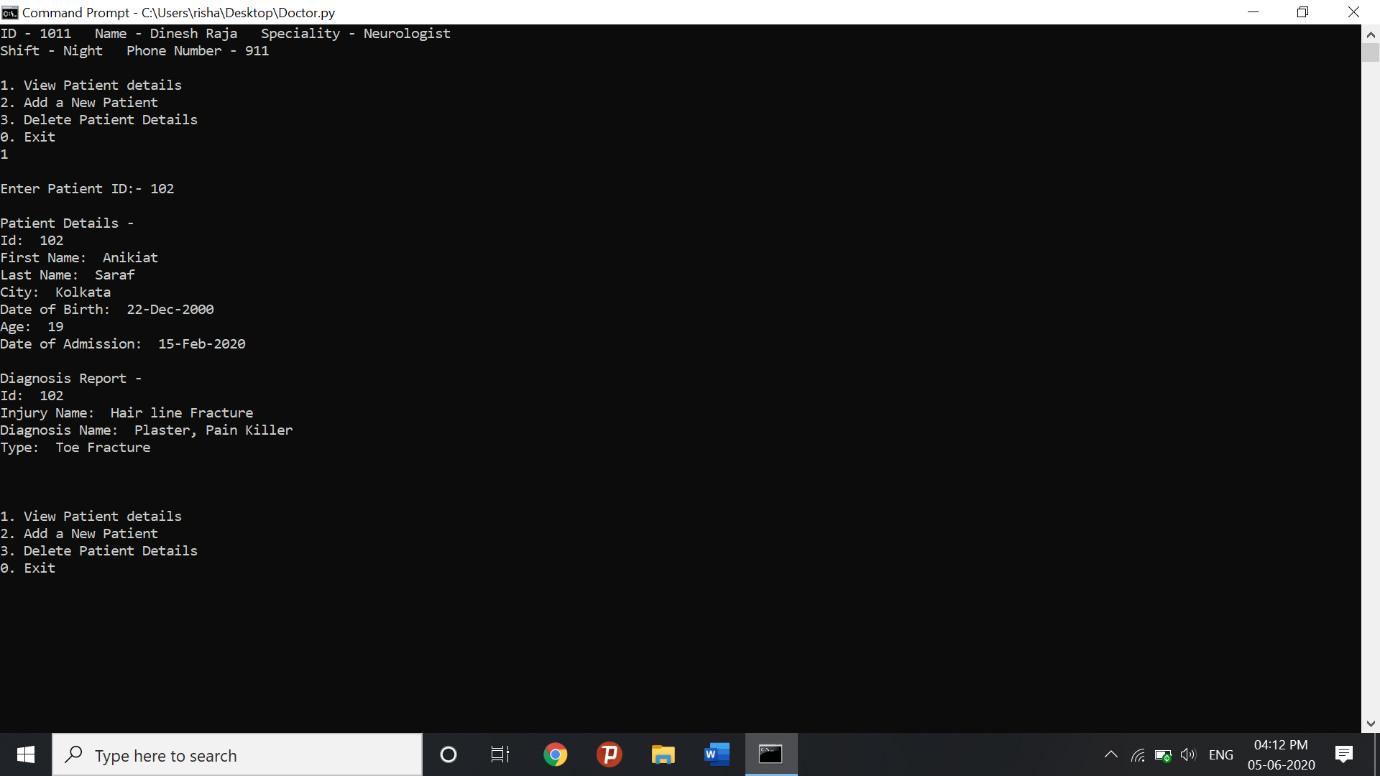
1. Windows 7 or later

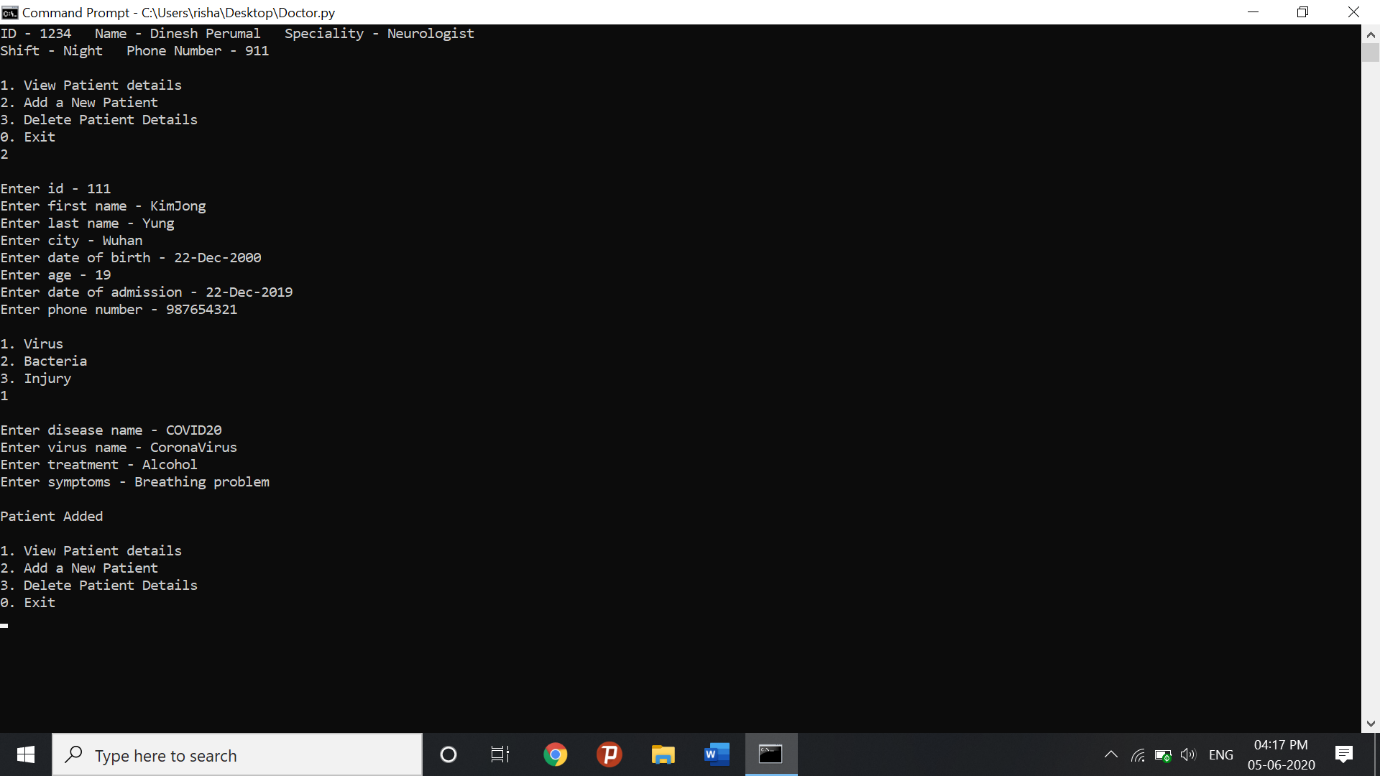
2. Mac OS X 10.11 or higher

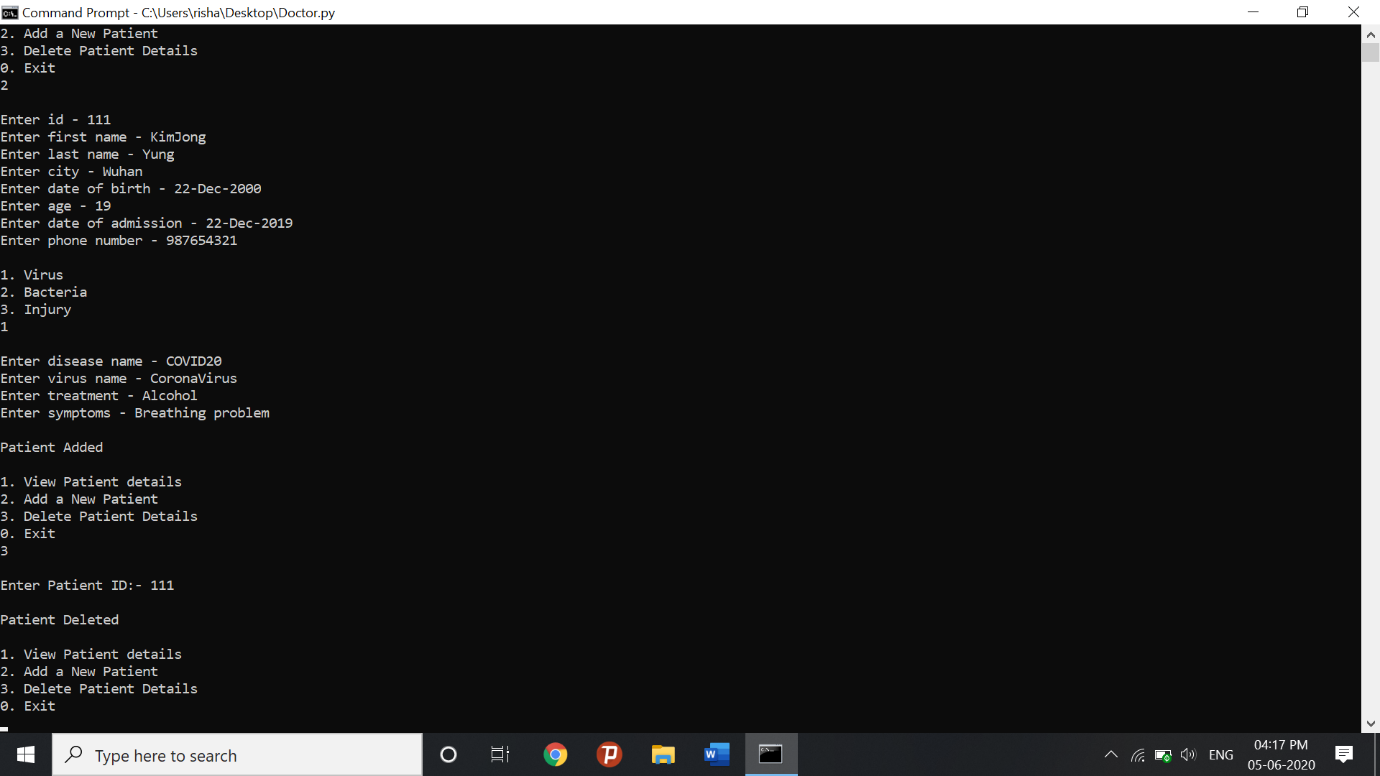
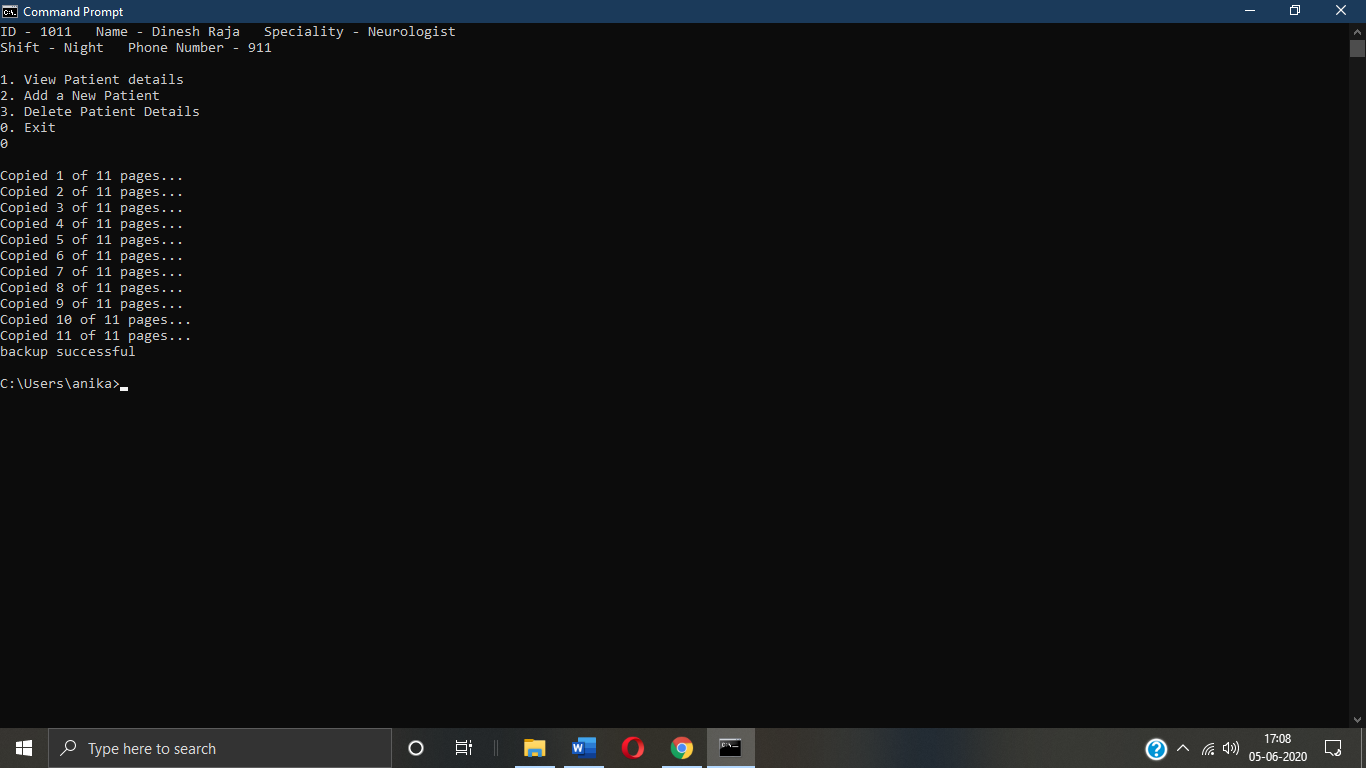
3. Microsoft .NET Framework 4.6

**Results: -**









**Conclusion: -**

Health Issue Management System may be a computerized system which stores and retrieves data associated with medical treatment. But this system also empowers the doctors to access a patient’s medical records and to examine them with respect to the various sorts of diseases and injuries. This gives the doctors the power to update the disease table whenever a new disease is discovered. This system enables a medical organization to improve its effectiveness, quality of labour and reduce manual errors. Also, all the current information is available for the doctors and officials wherever they require it. The information is well protected for private use and makes the information processing in no time. Thus, the computerization of the Health Centre will help greatly in maintaining of proper information about the patients, drug information and diagnosis.

We believe that this may be a little step towards helping those that are the backbone of this humanity and who risk their lives daily in order that we stay safe.

**Appendix: -**

* 1. **Code -**

**import** getpass

**import** sqlite3

connection**=**sqlite3**.**connect**(**'hospital.db'**)**

cursor**=**connection**.**cursor**()**

error**=**1

**from** os **import** system**,** name

**def** screen\_clear**():**

**if** name **==** 'nt'**:**

\_ **=** system**(**'cls'**)**

**else:**

\_ **=** system**(**'clear'**)**

cursor**.**execute**(**"""select count(name) from sqlite\_master where type='table' and name='doctor'"""**)**

**if** cursor**.**fetchone**()[**0**]==**0**:**

cursor**.**execute**(**"""CREATE TABLE doctor (

d\_id number primary key,

dnamedfirst VARCHAR2(20),

dnamedlast VARCHAR2(30),

password varchar2(20) not null,

speciality varchar2(40) not null,

shift varchar2(10) not null,

phone number(10) not null);"""**)**

cursor**.**execute**(**"""select count(name) from sqlite\_master where type='table' and name='patient'"""**)**

**if** cursor**.**fetchone**()[**0**]==**0**:**

cursor**.**execute**(**"""CREATE TABLE patient (

p\_id number primary key,

pfirst VARCHAR2(20),

pdlast VARCHAR2(30),

City varchar2(20) not null,

DOB date not null,

age number not null,

DOA date not null,

number number(10) not null);"""**)**

cursor**.**execute**(**"""CREATE TABLE virus (

p\_id number not null,

dname VARCHAR2(20) primary key,

vname VARCHAR2(20),

treatment VARCHAR2(50),

symptoms varchar2(50) not null);"""**)**

cursor**.**execute**(**"""CREATE TABLE bacteria (

p\_id number not null,

dname VARCHAR2(20) primary key,

bname VARCHAR2(20),

treatment VARCHAR2(50),

symptoms varchar2(50) not null);"""**)**

cursor**.**execute**(**"""CREATE TABLE injury (

p\_id number not null,

iname VARCHAR2(20) primary key,

idiagnosis VARCHAR2(50),

type varchar2(50) not null);"""**)**

cursor**.**execute**(**"""insert into patient values(101,'Mohit','Nayak','Bangalore','15-March-2001',18,'08-March-2020',9078435952)"""**)**

cursor**.**execute**(**"""insert into patient values(102,'Anikiat','Saraf','Kolkata','22-Dec-2000','19','15-Feb-2020',9674825476)"""**)**

cursor**.**execute**(**"""insert into patient values(103,'Rishank','Pratik','Orissa','22-Dec-2001','18','19-Nov-2015',9117854569)"""**)**

cursor**.**execute**(**"""insert into patient values(104,'Risav','Jana','Nepal','06-Jan-2001',18,'25-Oct-2010',7854963284)"""**)**

cursor**.**execute**(**"""insert into patient values(105,'Wilson','Vidyut','Mumbai','23-Nov-2001',18,'23-Nov-2005',7854129645)"""**)**

cursor**.**execute**(**"""insert into patient values(106,'Dinesh','Sharma','Rajasthan','23-Feb-2000',20,'23-Feb-2000',8476423858)"""**)**

cursor**.**execute**(**"""insert into virus values(103,'Ebola','Ebov','Oxygen Therapy, IV Fluids','Muscle Pain, Fever, Bleeding')"""**)**

cursor**.**execute**(**"""insert into virus values(105,'Measles','Paramyxo','Vitamin A','Cough, Skin Rash')"""**)**

cursor**.**execute**(**"""insert into bacteria values(101,'TB','Mycobacterium','Antibiotics','Cough and Sneezes')"""**)**

cursor**.**execute**(**"""insert into bacteria values(106,'Cholera','Vibrio','IV Fluids, Antibiotics','Seizures, Diarrhoea')"""**)**

cursor**.**execute**(**"""insert into injury values(102,'Hair line Fracture','Plaster, Pain Killer','Toe Fracture')"""**)**

cursor**.**execute**(**"""insert into injury values(104,'bullet wound','Removal of Bullet','Wound')"""**)**

**print(**"Databse created successfully"**)**

**else:**

e**=**1

**while** e**!=**0**:**

e**=int(input(**"1. Sign In\n2. Create a New Doctor Account\n"**))**

**if** e**==**2**:**

did**=int(input(**'\nEnter id - '**))**

dnf**=input(**'Enter first name - '**)**

dnl**=input(**'Enter last name - '**)**

pas**=**getpass**.**getpass**(**'Enter password - '**)**

spec**=input(**'Enter speciality - '**)**

shf**=input(**'Enter working shift - '**)**

ph**=int(input(**'Enter phone number - '**))**

cursor**.**execute**(**"""insert into doctor values(?,?,?,?,?,?,?)"""**,(**did**,**dnf**,**dnl**,**pas**,**spec**,**shf**,**ph**))**

screen\_clear**()**

e**=**1

**elif** e**==**1**:**

**while** error**==**1**:**

i**=input(**"\nEnter your ID - "**)**

p**=**getpass**.**getpass**(**"Enter your Password - "**)**

cursor**.**execute**(**"""select count(d\_id) from doctor where d\_id=(?)"""**,(**i**,))**

**if** cursor**.**fetchone**()[**0**]==**1**:**

cursor**.**execute**(**"""select count(password) from doctor where password=?"""**,(**p**,))**

**if** cursor**.**fetchone**()[**0**]==**1**:**

**print(**"\nSign in successful!"**)**

screen\_clear**()**

error**=**0

e**=**0

r**=**1

cursor**.**execute**(**"""select d\_id,dnamedfirst,dnamedlast,speciality,shift,phone from doctor where d\_id=(?)"""**,(**i**,))**

**for** row **in** cursor**.**fetchall**():**

**print(**"ID -"**,**row**[**0**],**" Name -"**,**row**[**1**],** row**[**2**],**" Speciality -"**,**row**[**3**],**"\nShift -"**,**row**[**4**],**" Phone Number -"**,**row**[**5**])**

**while** r**!=**0**:**

**print(**"\n1. View Patient details\n2. Add a New Patient\n3. Delete Patient Details\n0. Exit"**)**

r**=int(input())**

**if** r**==**1**:**

access**=input(**"\nEnter Patient ID:- "**)**

cursor**.**execute**(**"""select count(\*) from patient where p\_id=(?)"""**,(**access**,))**

**if** cursor**.**fetchone**()[**0**]!=**0**:**

cursor**.**execute**(**"""select \* from patient where p\_id=(?)"""**,(**access**,))**

**print(**"\nPatient Details - "**)**

**for** row **in** cursor**.**fetchall**():**

**print(**"Id: "**,** row**[**0**])**

**print(**"First Name: "**,** row**[**1**])**

**print(**"Last Name: "**,** row**[**2**])**

**print(**"City: "**,** row**[**3**])**

**print(**"Date of Birth: "**,** row**[**4**])**

**print(**"Age: "**,** row**[**5**])**

**print(**"Date of Admission: "**,** row**[**6**])**

**print(**"\nDiagnosis Report - "**)**

cursor**.**execute**(**"""select count(\*) from virus where p\_id=(?)"""**,(**access**,))**

**if** cursor**.**fetchone**()[**0**]!=**0**:**

cursor**.**execute**(**"""select \* from virus where p\_id=(?)"""**,(**access**,))**

**for** row **in** cursor**.**fetchall**():**

**print(**"Id: "**,** row**[**0**])**

**print(**"Disease Name: "**,** row**[**1**])**

**print(**"Virus Name: "**,** row**[**2**])**

**print(**"Treatment: "**,** row**[**3**])**

**print(**"Symptoms: "**,** row**[**4**])**

**print(**"\n"**)**

cursor**.**execute**(**"""select count(\*) from bacteria where p\_id=(?)"""**,(**access**,))**

**if** cursor**.**fetchone**()[**0**]!=**0**:**

cursor**.**execute**(**"""select \* from bacteria where p\_id=(?)"""**,(**access**,))**

**for** row **in** cursor**.**fetchall**():**

**print(**"Id: "**,** row**[**0**])**

**print(**"Disease Name: "**,** row**[**1**])**

**print(**"Bacteria Name: "**,** row**[**2**])**

**print(**"Treatment: "**,** row**[**3**])**

**print(**"Symptoms: "**,** row**[**4**])**

**print(**"\n"**)**

cursor**.**execute**(**"""select count(\*) from injury where p\_id=(?)"""**,(**access**,))**

**if** cursor**.**fetchone**()[**0**]!=**0**:**

cursor**.**execute**(**"""select \* from injury where p\_id=(?)"""**,(**access**,))**

**for** row **in** cursor**.**fetchall**():**

**print(**"Id: "**,** row**[**0**])**

**print(**"Injury Name: "**,** row**[**1**])**

**print(**"Diagnosis Name: "**,** row**[**2**])**

**print(**"Type: "**,** row**[**3**])**

**print(**"\n"**)**

**else:**

**print(**"Incorrect Patient id"**)**

**elif** r**==**2**:**

pid**=int(input(**'\nEnter id - '**))**

pnf**=input(**'Enter first name - '**)**

pnl**=input(**'Enter last name - '**)**

pcity**=input(**'Enter city - '**)**

pdob**=input(**'Enter date of birth - '**)**

page**=int(input(**'Enter age - '**))**

pdoa**=input(**'Enter date of admission - '**)**

pnum**=int(input(**'Enter phone number - '**))**

cursor**.**execute**(**"""insert into patient values(?,?,?,?,?,?,?,?)"""**,(**pid**,**pnf**,**pnl**,**pcity**,**pdob**,**page**,**pdoa**,**pnum**))**

**print(**"\n1. Virus\n2. Bacteria\n3. Injury"**)**

m**=int(input())**

**if** m**==**1**:**

dname**=input(**"\nEnter disease name - "**)**

bname**=input(**"Enter virus name - "**)**

treatment**=input(**"Enter treatment - "**)**

symptoms**=input(**"Enter symptoms - "**)**

cursor**.**execute**(**"""insert into virus values(?,?,?,?,?)"""**,(**pid**,**dname**,**bname**,**treatment**,**symptoms**))**

**elif** m**==**2**:**

dname**=input(**"\nEnter disease name - "**)**

bname**=input(**"Enter bacteria name - "**)**

treatment**=input(**"Enter treatment - "**)**

symptoms**=input(**"Enter symptoms - "**)**

cursor**.**execute**(**"""insert into bacteria values(?,?,?,?,?)"""**,(**pid**,**dname**,**bname**,**treatment**,**symptoms**))**

**elif** m**==**3**:**

iname**=input(**"\nEnter injury name - "**)**

idiag**=input(**"Enter diagnosis - "**)**

itype**=input(**"Enter injury type - "**)**

cursor**.**execute**(**"""insert into injury values(?,?,?,?)"""**,(**pid**,**iname**,**idiag**,**itype**))**

**print(**"\nPatient Added"**)**

connection**.**commit**()**

**elif** r**==**3**:**

access**=input(**"\nEnter Patient ID:- "**)**

cursor**.**execute**(**"""select count(\*) from patient where p\_id=(?)"""**,(**access**,))**

**if** cursor**.**fetchone**()[**0**]!=**0**:**

cursor**.**execute**(**"""delete from patient where p\_id=(?)"""**,(**access**,))**

cursor**.**execute**(**"""select count(\*) from virus where p\_id=(?)"""**,(**access**,))**

**if** cursor**.**fetchone**()[**0**]!=**0**:**

cursor**.**execute**(**"""delete from virus where p\_id=(?)"""**,(**access**,))**

cursor**.**execute**(**"""select count(\*) from bacteria where p\_id=(?)"""**,(**access**,))**

**if** cursor**.**fetchone**()[**0**]!=**0**:**

cursor**.**execute**(**"""delete from bacteria where p\_id=(?)"""**,(**access**,))**

cursor**.**execute**(**"""select count(\*) from injury where p\_id=(?)"""**,(**access**,))**

**if** cursor**.**fetchone**()[**0**]!=**0**:**

cursor**.**execute**(**"""delete from injury where p\_id=(?)"""**,(**access**,))**

**else:**

**print(**"Incorrect Patient id Patient does not exist"**)**

**print(**"\nPatient Deleted"**)**

connection**.**commit**()**

**elif** r**==**0**:**

**exit()**

**else:**

**print(**"Incorrect passoword. Please retry "**)**

**else:**

**print(**"Incorrect User ID. Please retry "**)**

**break**

**elif** e**==**2212**:**

cursor**.**execute**(**"""select \* from doctor"""**)**

**print(**cursor**.**fetchall**())**

cursor**.**execute**(**"""select \* from virus"""**)**

**print(**cursor**.**fetchall**())**

cursor**.**execute**(**"""select \* from bacteria"""**)**

**print(**cursor**.**fetchall**())**

cursor**.**execute**(**"""select \* from injury"""**)**

**print(**cursor**.**fetchall**())**

**break**

connection**.**commit**()**

connection**.**close**()**

**print**("")

**def** progress(status, remaining, total):

**print**(f'Copied {total-remaining} of {total} pages...')

**try:**

sqliteCon = sqlite3.connect('hospital.db')

backupCon = sqlite3.connect('hospital\_backup.db')

with backupCon:

sqliteCon.backup(backupCon, pages=1, progress=progress)

**print**("backup successful")

**except** sqlite3.Error as error:

print("Error while taking backup: ", error)

**finally:**

**if**(backupCon):

backupCon.close()

sqliteCon.close()

* 1. **Queries –**
     1. connection=sqlite3.connect('hospital.db')
     2. cursor=connection.cursor()
     3. select count(name) from sqlite\_master where type='table' and name='doctor’
     4. connection.backup()
     5. CREATE TABLE doctor (

d\_id number primary key,

dnamedfirst VARCHAR2(20),

dnamedlast VARCHAR2(30),

password varchar2(20) not null,

speciality varchar2(40) not null,

shift varchar2(10) not null,

phone number(10) not null);

* + 1. select d-id,p\_id,pnamefirst from doctor d cross join patient p on d.d\_id = p.d\_id;
    2. not null
    3. cursor.fetchone()[0]==0
    4. insert into virus values(105,'Measles','Paramyxo','Vitamin A','Cough, Skin Rash')
    5. cursor.execute()
    6. insert into doctor values(?, ?, ?, ?, ?, ?, ?), (did,dnf,dnl,pas,spec,shf,ph)
    7. cursor.fetchall()
    8. delete from virus where p\_id=(?), (access,)
    9. connection.commit()
    10. auto increment
    11. connection.close()
    12. create view "Details" as "select \* from patient";
    13. default 'kolkata'
    14. select d-id,dnamedfirst,p\_id,pnamefirst from doctor d inner join patient p on d.d\_id=p.d\_id;
    15. select pnamefirst as patient\_name from patient;
    16. update patient set diagnosis=’pneumonia’ where p\_id=(?),(access,);
    17. foreign key (p\_id) references virus(p\_id)
    18. def progress(status, remaining, total):

print(f'Copied {total-remaining} of {total} pages...')

try:

sqliteCon = sqlite3.connect('hospital.db')

backupCon = sqlite3.connect('hospital\_backup.db')

with backupCon:

sqliteCon.backup(backupCon, pages=1, progress=progress)

print("backup successful")

except sqlite3.Error as error:

print("Error while taking backup: ", error)

finally:

if(backupCon):

backupCon.close()

sqliteCon.close()