DELHI PUBLIC SCHOOL PANIPAT REFINERY

(DEPARTMENT OF COMPUTER SCIENCE)

ACADEMIC YEAR: 2021-22

TOPIC OF THE PROJECT

Hospital Management System

ACKNOWLEDGEMENTS

It's a great pleasure for me to express my gratitude to our honorable pioneer and coordinator (as well as teacher), Mrs. Sabina Garg for giving us the prestigious opportunity and guidance to accomplish the project-based laboratory report and providing constant motivation throughout the semester.

I express my sincere gratitude to our principal Mr. Vinod Sharma for his administration towards our academic growth and intellectual excellence.

Finally, it is pleased to acknowledge the indebtedness to all those who devoted them directly or indirectly to make this project report a success. A big thanks to all of them for making this project a success.

TABLE OF CONTENTS

OBJECTIVES OF THE PROJECT

The objectives of the project are to incorporate the theoretical world of Computer Science into the practical world. It aims at the application of classroom knowledge to real-world situations by providing them a means to utilize their programmatic skills for developing good and reliable software.

The project will encourage students for additional learning of concepts of Computer Science. It intends to prepare students for the upcoming future where the application of gained knowledge to real-life problems has utmost significance.

1. Write programs utilizing modern software tools.
2. Apply object-oriented programming principles effectively when developing small to medium-sized projects.
3. Write effective procedural code to solve small to medium-sized problems.
4. Students will demonstrate a breadth of knowledge in computer science, as exemplified in the areas of systems, theory, and software development.
5. Students will demonstrate the ability to conduct research or applied Computer Science projects, requiring writing and presentation skills that exemplify scholarly style in computer science.

INTRODUCTION AND PREFACE

The health of citizens is the wealth of the nation. The field has witnessed a rapid metamorphosis in all of its sections. Hospital Management System is designed to improve quality as well as management in areas of clinical process & analysis and activity-based costing.

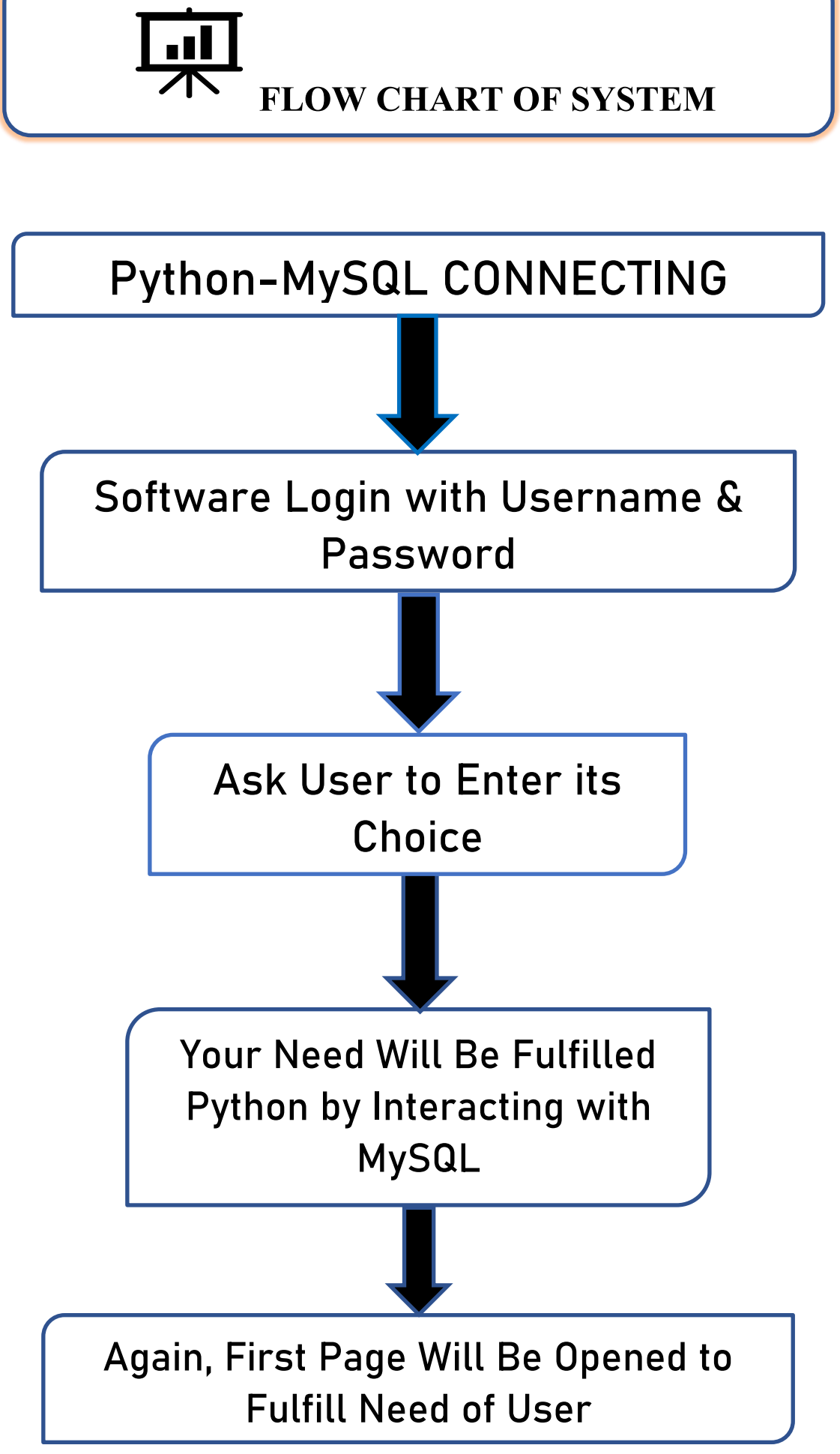
The Hospital Management System can be created taking username and password. It's accessible by only admin, doctor & receptionist and data can be retrieved easily. HMS is powerful, flexible & easy to use.

The project HMS includes registration of patients, storing their details into the system by using any database.

The software has the facility to give a unique ID for every patient & stores detail of every patient manually. Admin can view the availability of doctors & details of a patient using the name & ID.

HMS is designed for multi-specialist hospitals, to cover a wide range of health administration & administration processes. It also aims at providing low-cost reliable automation of existing systems. The system also provides excellent security of data at every level of user-system interaction and provides a robust & reliable storage facility.

FLOW CHART OF SYSTEM

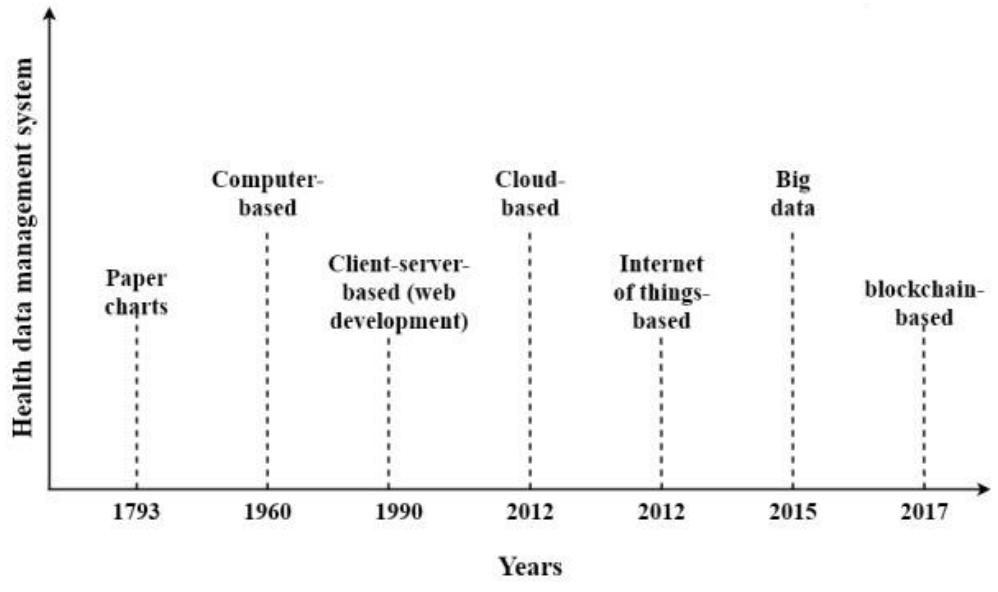


THE EXISTING SYSTEM

The manual and human-based medical recording systems have always been an evolving process. Still, the existing system suffers from various challenges such as

- distrust among hospitals and patients
- poor data quality
- security issues
- lack of proper online infrastructure
- data redundancy
- inaccessible data by patients
- lack of data integrity
- and high risk of errors and inconsistency

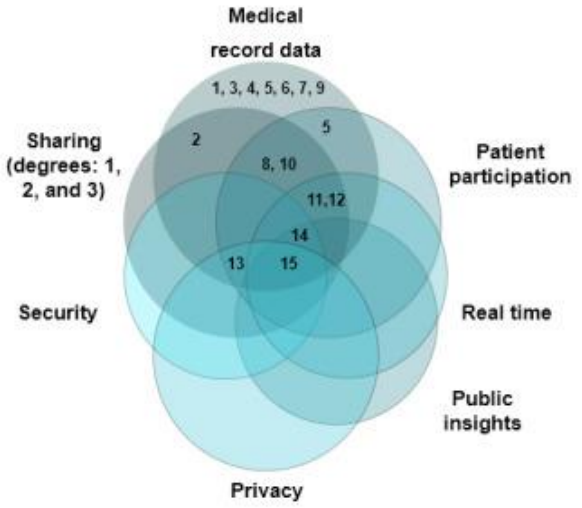
Evolution of health data management system:



Limitations of health data management systems

Health Data Management Systems	Limitations
Paper Charts	Illegible handwriting resulting in incorrect treatments and deaths. Requires physical storage and is susceptible to unplanned destruction such as flood, fire, rodents, and degradation. Physically cumbersome to read, understand, and search for specific information. The cost and time required for paper charts to be requested for duplication and then delivered are unacceptably high.
Computer-based	Medical records are managed by the physicians and cannot be accessed by the patients. Physicians visiting a patient have to note down or memorize the patient's medical data to return to the hospital and record it digitally, which may lead to errors.
Client-server-based	A patient has no traceability on how his or her data are used. The issues of security, privacy, and single point of failure. In addition, a cohesive view of a patient's medical data from multiple hospitals is difficult. Requires repeating medical tests at times, which results in more time, cost, and effect on health conditions.
Cloud-based	Single point of failure, loss of data control and stewardship, a requirement of steady internet connection, and data reliability.
IoT-based	Data security and patient privacy are major concerns.
Big-data–based	The process of data aggregation from different storage sites is time-consuming, complex, and expensive. The data are stored using different formats and require pre-processing. In addition, preserving the security of the data and privacy of the patient identity while maintaining the usefulness of data for analysis and studies is quite challenging.
Block chain-based	The process of ledger update on multiple nodes is energy consuming and suffers from the issue of low throughput.

REQUIREMENTS OF A HEALTH DATA MANAGEMENT SYSTEM



(1) medical record data,

(2) real-time data access,

(3) patient participation,

(4) data sharing,

(5) data security,

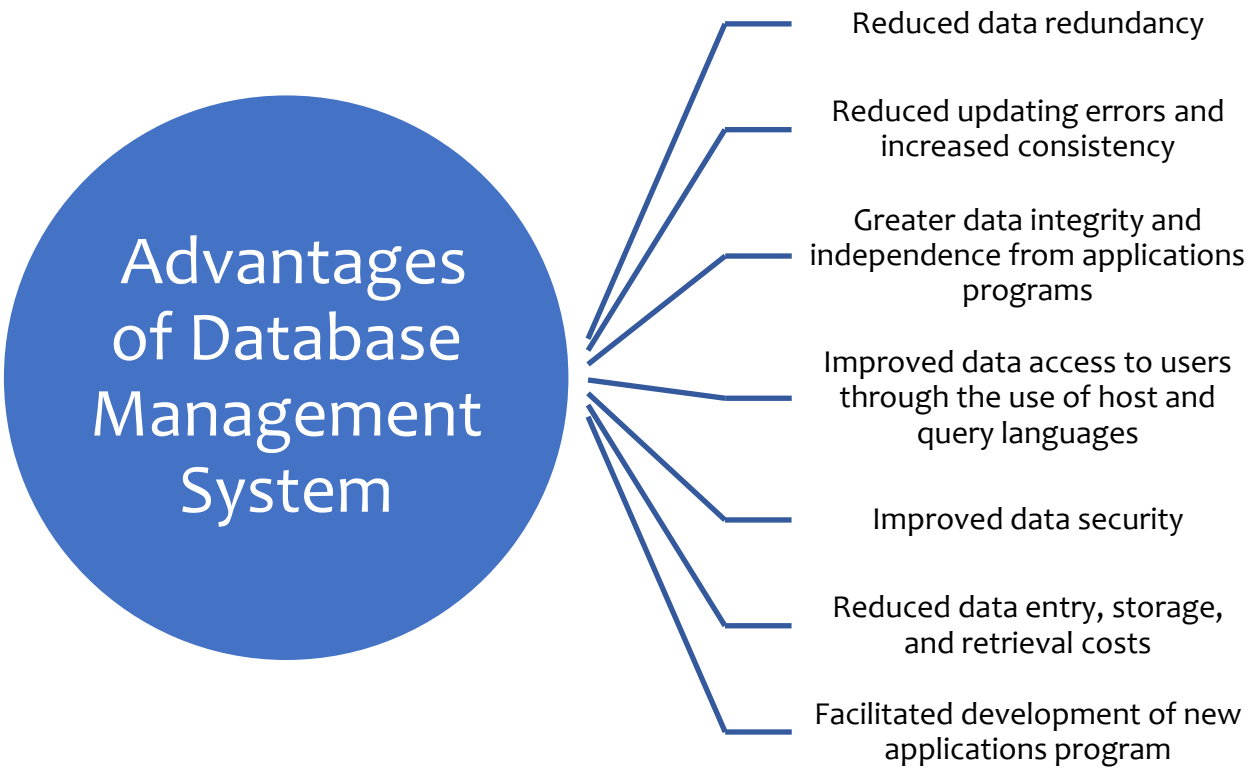
(6) patient identity privacy, and

(7) public insights

THE PROPOSED SYSTEM

The present world is the world of information technology.

With digitization at its height, one cannot afford manual and human-based work where computers and machines can be more powerful and precise. Working environments such as hospitals need infallible methods of data management.



Our model is based on MySQL-Python connectivity.

It features a Hospital Database Management System (HDBMS) along with special emphasis for Covid-19 patients.

In the period of the ongoing Covid-19 pandemic, our HDBMS will provide fast and efficient solutions for modern hospitals. It not only will serve the purpose of efficient, secure, and reliable patient registration in the hospital but will also help the patients and hospital for Covid-19 specific needs in the epidemic time. Thus, it would result in faster and better treatment of patients.

The system will make sure that the full-fledged focus of the hospital will be on treating their patients and serving them without the worries of maintaining proper records of patients.

The patients will be able to securely access their data which the hospital would possess and would have complete control over it.

The software would provide two-way login. Patients and Hospital staff can log in with their specific credentials which in turn would provide a secure methodology of accessing and viewing data.

With patients having control over their data, the challenge of distrust among hospitals and patients can be resolved.

FUNCTIONALITY

The software is expected to deliver high-quality output with reliable accuracy and speed. By default, the software comes preloaded with all the features and functionalities that a hospital would require. Following are a brief overview of what the software would be able to achieve with python-MySQL connectivity:

1. **Two-way login** –

   (i) Hospital Staff login

   - Registering Patient, Doctors, Worker details
   - Total Patient, Doctor, Worker details
   - Details of Patients, Doctors, Workers at an individual level
   - Updation of previous records
   - Covid-19 Section
   - Prescription for patients

   (ii) Patient login

   - View its details
   - Request update for its details
   - Register as a new patient
   - View past prescriptions
   - Covid-19 patient login
   - Feedback

2. **Additional info**:

   - Pandemic Do's & Don'ts
   - Tips about healthy living
   - Detailed information about particular food items
   - New Feature request for the software

With the help of different databases, tables of MySQL, and its connectivity with python all the features of the software would be made possible.

MySQL Databases Used:

- Hospital Staff
- Patient
- Additional info

Code Editor Used: Python IDLE

Source Code:

```
import time
from sys import exit
import mysql.connector as sql

con = sql.connect(host = "localhost", user = "root", password =
"ENG249/94MySQL", database = "Project")  #Open a connection to MySQL
Database
if con.is_connected():  #Checks connection
    print("Successfully connected.")
cursor = con.cursor()  #Creates cursor instance

print("""|_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-
_-_-_-_|
                         HOSPITAL MANAGEMENT SYSTEM
|_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_|
\n""")

print("""
         |_-_-_-_-_-_________________________________-_-_-_-_-_-|
         |                   Login/Sign-up Page                  |
         ||_-_-_-_-_-_________________________________-_-_-_-_-_||


                     1. Login
                     2. Sign-up as a new user
                     3. Update User-credentials
""")

def signup():
    signchoice = input("\nWould you like to sign-up for a new
account?...(Yes/No): ")
    if signchoice == "Yes" or signchoice == "YES" or signchoice == "yes":
        print("\n\n\t\tNew User Sign-up\n")
        New_user = input("\nEnter Name: ")
        New_Password = input("\nCreate Password: ")
        New_Password_confm = input("Confirm your Password: ")
        while New_Password_confm != New_Password:  #Prompts the user to
re-enter correct password when confirmed password and password entered are
not the same
            print("\nPlease ReEnter your password again:  ")
            New_Password = input("Create Password: ")
            New_Password_confm = input("Confirm your Password: ")
        u_signup = "INSERT INTO login(User, Password) Values ('{}',
'{}')".format(New_user, New_Password)  #Inserts User's username and
password in login table of MySQL
        cursor.execute(u_signup)  #Executes the SQL query u_signup
        con.commit()  #Commits the changes to MySQL database
        print("\nSucessfully Registered!🔒")
        return("Welcome!")
    else:
        print("\n\t\tThanks for visiting!\n")
        return(exit())  #Exits the system when user don't want to sign in
```

```
def login():
    print("\n\t\tExisting User Login")
    count = 0
    User = input("\nEnter Username: ")
    Password = input("\nEnter Password: ")
    cursor.execute("select * from login")  #SQL retrieves all the data of
table login
    u_total = cursor.fetchall()  #Fetches all the records in the resultset
    u_list = []
    for row in u_total:  #Processing u_total tuple one row at a time
        u_list.append(row)  #Appends one row in u_list at a time
    for i in u_list:
        if i[0] == User:
            if i[1] == Password:
                return("Welcome!")  #Username & Password found correct
            elif i[1] != Password:  #Username is correct but Password is
incorrect
                print("Wrong Password! Please try again.")
                for j in range(3):  #Repeats the entering password and
waiting for 60 seconds process 3 times before exiting the system
completely
                    while count <= 3: #Four chances to enter correct
password
                        Password = input("Enter Password: ")
                        if i[1] == Password:
                            return("\nWecome!")  #Exits the fuction
login() when password found correct
                        print("Wrong Password! Please try again.")
                        count += 1
                    if j == 2:  #Condition for exiting the system
                        break
                    print("\nYou have entered your password incorrect 4
times. Please wait 60 seconds to enter password again.\n")
                    time.sleep(60)  #Waits the user for 60 seconds before
entering password again
                    count = 0  #Sets count value back to zero
                print("\nYou have entered wrong password multiple times.
Try Again later!\n")
                print("\n\t\tThanks for visiting!\n")
                return(exit())  #Exits the system
            break  #Breaks the loop when Username & Password found correct
    else:
        print("\nNo user found with this username!")
        return signup()  #Prompts the user to signup() function if the
username entered is incorrect or not in the database


def cred_check():
    print("\n\t\tUser Old Credentials check")
    count = 0
    u_old = input("\nEnter old username: ")
    passwd_old = input("\nEnter old password: ")
```

```
    cursor.execute("select * from login")
    u_total = cursor.fetchall()
    u_list = []
    for row in u_total:
        u_list.append(row)
    for i in u_list:
        if i[0] == u_old:
            if i[1] == passwd_old:
                return(update_credentials(u_old)) # Returns the u_old
variable to update_credentials function when Username & Password found
correct
            elif i[1] != passwd_old:  #Username is correct but Password is
incorrect
                print("Wrong Password! Please try again.")
                for j in range(3):  #Repeats the entering password and
waiting for 60 seconds process 3 times before exiting the system
completely
                    while count <= 3: #Four chances to enter correct
password
                        passwd_old = input("Enter Password: ")
                        if i[1] == passwd_old:
                            return(update_credentials(u_old))  #Returns
the u_old variable to update_credentials function when password found
correct
                        print("Wrong Password! Please try again.")
                        count += 1
                    if j == 2:  #Condition for exiting the system
                        break
                    print("\nYou have entered your password incorrect 4
times. Please wait 60 seconds to enter password again.\n")
                    time.sleep(60)  #Waits the user for 60 seconds before
entering password again
                    count = 0  #Sets count value back to zero
                print("\nYou have entered wrong password multiple times.
Try Again later!\n")
                print("\n\t\tThanks for visiting!\n")
                return(exit())  #Exits the system
            break  #Breaks the loop when Username & Password found correct
    else:
        print("\nNo user found with this username!")
        return signup()  #Prompts the user to signup() function if the
username entered is incorrect or not in the database


def update_credentials(u_old):
    print("\n\t\tUser Credentials Updater\n")
    u_update = input("\nEnter new Username: ")
    passwd_update = input("\nEnter new Password: ")
    passwd_update_cnfm = input("Confirm Password: ")
    while passwd_update_cnfm != passwd_update:  #Prompts the user to re-
enter correct password when confirmed password and new password entered
are not the same
            print("Please ReEnter your password again:  ")
```

```
            passwd_update = input("\nEnter new Password: ")
            passwd_update_cnfm = input("\nConfirm Password: ")
    update_cmd = ("UPDATE login SET User = ('{}'), Password = ('{}') where
User = ('{}')".format(u_update, passwd_update, u_old))  #Updates User's
username and password in login table of MySQL
    cursor.execute(update_cmd)  #Executes the SQL query update_cmd
    con.commit()  #Commits the changes to MySQL database
    return ("\nSuccessfully Updated!🔐")

#Choice picker
while True:
    choose = int(input("Enter your choice: "))
    if choose == 1:
        task = login()
        print(task)
        break
    elif choose == 2:
        task = signup()
        print(task)
        break
    elif choose == 3:
        task = cred_check()
        print(task)
        break
    else:
        print("\nPlease enter your Choice between (1-3)\n")
        continue  #Prompts user back to enter the choice of choice not
between (1-3)

print("""
                ||__________ENTER YOUR CHOICE_____________||

                    ------------------------------------
                    | 1. Register Patient's details    |
                    | 2. Register Doctor's details     |
                    | 3. Register Worker's details     |
                    |----------------------------------|
                    | 4. Total Patient's details       |
                    | 5. Total Doctor's details        |
                    | 6. Total Worker's details        |
                    |----------------------------------|
                    | 7. Particular Patient's detail   |
                    | 8. Particular Doctor's details   |
                    | 9. Particular Worker's details   |
                    |----------------------------------|
                    | 10. Update Patient's Status      |
                    | 11. Delete Patient's Record      |
                    |----------------------------------|
                    | 12. Feedback                     |
                    | 13. Exit the System              |
                    ------------------------------------""")
```

```
def patient_register():
    print("\nPATIENT REGISTERATION\n")
    p_name = input("Enter Patient's name: ")
    p_gender = input("Enter Patient's gender: ")
    p_age = int(input("Enter Patient's age: "))
    p_ailment = input("Enter Patient's ailment: ")
    p_phone_no = int(input("Enter Patient's phone number: "))
    p_status = input("Enter your current Status.....(Ill/Well): ")

    p_insert = "INSERT INTO patient(Name, Gender, Age, Ailment,
Phone_Number, Current_Status) Values ('{}', '{}', {}, '{}', {},
'{}')".format(p_name, p_gender, p_age, p_ailment, p_phone_no, p_status)
#Inserts patient details in patient table of MySQL
    cursor.execute(p_insert)  #Executes the SQL query p_insert
    con.commit()  #Commits the changes to MySQL databse
    return "\nSucessfully Registered!😃😃😁"

def doctor_register():
    print("\nDOCTOR REGISTERATION\n")
    d_name = input("Enter Doctor's name: ")
    d_gender = input("Enter Doctor's gender: ")
    d_dept = input("Enter Doctor's department: ")
    d_phone_no = int(input("Enter Doctor's phone number: "))
    d_email = input("Enter Doctor's email address: ")

    d_insert = "INSERT INTO doctor(Name, Gender, Department, Phone_Number,
Email_Address) Values ('{}', '{}', '{}', {}, '{}')".format(d_name,
d_gender, d_dept, d_phone_no, d_email)
    cursor.execute(d_insert)
    con.commit()
    return "\nSucessfully Registered!😃😃😁"

def worker_register():
    print("\nWORKER REGISTERATION\n")
    w_name = input("Enter Worker name: ")
    w_gender = input("Enter Worker's gender: ")
    w_age = int(input("Enter Worker's age: "))
    w_worktype = input("Enter type of Work: ")
    w_phone_no = int(input("Enter Worker's phone number: "))

    w_insert = "INSERT INTO worker(Name, Gender, Age, WorkType,
Phone_Number) VALUES ('{}', '{}', {}, '{}', '{}')".format(w_name,
w_gender, w_age, w_worktype, w_phone_no)
    cursor.execute(w_insert)
    con.commit()
    return "\nSucessfully Registered!😃😃😁"

def patient_total():
    cursor.execute("select * from patient")  #SQL retrieves all the data
of table login
    p_total = cursor.fetchall()  #Fetches all the records in the resultset
```

```
    print("\nTotal patient details: \n")
    p_list = []  #Creates an empty list
    for row in p_total:  #Processing p_total tuple one row at a time
        p_list.append(row)  #Appends one row in u_list at a time
    for i in p_list:  #Transverse the p_list list one row stored as tuple
in p_list at a time
        print(i, end = "\n\n")  #Prints one row at a time
    return ""

def doctor_total():
    cursor.execute("select * from doctor")
    d_total = cursor.fetchall()
    print("\nTotal doctor details: \n")
    d_list = []
    for row in d_total:
        d_list.append(row)
    for i in d_list:
        print(i, end = "\n\n")
    return ""

def worker_total():
    cursor.execute("select * from worker")
    w_total = cursor.fetchall()
    print("\nTotal worker details: \n")
    w_list = []
    for row in w_total:
        w_list.append(row)
    for i in w_list:
        print(i, end = "\n\n")
    return ""

def patient_particular():
    cursor.execute("select * from patient")  #SQL retrieves all the data
of table login
    p_total = cursor.fetchall()  #Fetches all the records in the resultset
    p_list = []
    for row in p_total:
        p_list.append(row)
    p_desired = input("Enter the name of the patient whose details you
want to view: ")
    p_desired_cmd = "select * from patient where Name =
('{}')".format(p_desired)  #SQL retrieves the data of the desired patient
from the table patient
    cursor.execute(p_desired_cmd)  #Executes the query p_desired_cmd
    p_data = cursor.fetchall()  #Fetches the record of the desired patient
from the resultset
    for i in p_list:
        if i[1] == p_desired:  #Checks if the desired patient is in the
p_list list
            for row in p_data:
                return(row)
    else:
```

```
        return("Sorry, patient does not exist.")  #No patient found in the
table patient of name p_desired

def doctor_particular():
    cursor.execute("select * from doctor")
    d_total = cursor.fetchall()
    d_list = []
    for row in d_total:
        d_list.append(row)
    d_desired = input("\nEnter the name of the doctor whose details you
want to know: ")
    print()     #empty line
    d_desired_cmd = "select * from doctor where Name =
('{}')".format(d_desired)
    cursor.execute(d_desired_cmd)
    d_data = cursor.fetchall()
    for i in d_list:
        if i[1] == d_desired:
            for row in d_data:
                return(row)
    else:
        return("Sorry, the Doctor does not exist.")

def worker_particular():
    cursor.execute("select * from worker")
    w_total = cursor.fetchall()
    w_list = []
    for row in w_total:
        w_list.append(row)
    w_desired = input("Enter the name of the worker whose details you want
to know: ")
    w_desired_cmd = "select * from worker where Name =
('{}')".format(w_desired)
    cursor.execute(w_desired_cmd)
    w_data = cursor.fetchall()
    for i in w_list:
        if i[1] == w_desired:
            for row in w_data:
                return(row)
    else:
        return("Sorry, the Worker does not exist.")


def patient_update():
    p_name = input("\nEnter the name of the patient whose record you want
to update: ")
    cursor.execute("select * from patient")
    p_total = cursor.fetchall()
    p_list = []
    for row in p_total:  #Processing p_total tuple one row at a time
        p_list.append(row)   #Appends one row in u_list at a time
    for i in p_list:
```

```
        if i[1] == p_name:   #Checks if the desired patient is in the
p_list list
            p_ailment = input("\nDo you want to update details about
Patient's ailment...(Yes/No): ")  #Asks the user if user wants to update
patients's ailment
            if p_ailment == "yes" or p_ailment == "YES" or p_ailment ==
"Yes":
                p_ailment_new = input("\nEnter patient's current ailment:
")
                p_status = input("\nEnter the patient's current
status.....(Ill/Well): ")
                p_cmd = ("UPDATE patient SET Ailment = ('{}'),
Current_Status = ('{}') where Name = ('{}')".format(p_ailment_new,
p_status, p_name))  #Updates patient's ailment, current status with name
p_name in patient table of MySQL
            else:  #If user does not want to update patient's ailment
                p_status = input("\nEnter the patient's current status: ")
                p_cmd = ("UPDATE patient SET Current_Status = ('{}') where
Name = ('{}')".format(p_status, p_name))  #Updates patient's current
status with name p_name in patient table of MySQL
            cursor.execute(p_cmd)  #Executes the query p_cmd
            con.commit()  #Commits changes to the MySQL database
            return("Succesfully Updated Patient's Current Status!")
    else:
        return("No Patient with this name exists.")  #No patient found in
the table patient of name p_name

def patient_delete():
    p_name = input("\nEnter the name of the patient whose record you want
to delete: ")
    cursor.execute("select * from patient")
    p_total = cursor.fetchall()
    p_list = []
    for row in p_total:
        p_list.append(row)
    for i in p_list:
        if i[1] == p_name:
            p_cmd = ("DELETE from patient where Name =
('{}')".format(p_name))  #Deletes the patient's record with name p_name in
patient table of MySQL
            cursor.execute(p_cmd)  #Executes the query p_cmd
            con.commit()  #Commits the changes to MySQL database
            return("Successfully deleted Patient's Record!")
    else:
        return("No Patient with this name exists. Could not delete any
record!")  #No patient found in the table patient of name p_name

def feedback():
    print("\n\t\tFeedack")
    f_name = input("\nEnter your name: ")
    f_rate = int(input("\nPlease rate our system(0-10): "))
    f_feedback = input("\nTell us about you experience: ")
```

```
    f_insert = "INSERT into feedback(Name, Rating, Feedback) Values ('{}',
{}, '{}')".format(f_name, f_rate, f_feedback)  #Inserts users name,rating
and feedback in feedback table of MySQL
    cursor.execute(f_insert)  #Executes the quert f_inser
    con.commit()  #Commits the changes to My SQL databasess
    return ("\n\t\tThank You for you Valuable Feedback!")


ans = "Yes"
# Choice picker
while ans =="Yes" or ans == "yes" or ans == "YES":
    Choice = int(input("\nEnter your choice: "))

    if Choice == 1:
        task = patient_register()
        print(task)
    elif Choice == 2:
        task = doctor_register()
        print(task)
    elif Choice == 3:
        task = worker_register()
        print(task)
    elif Choice == 4:
        task = patient_total()
        print(task)
    elif Choice == 5:
        task = doctor_total()
        print(task)
    elif Choice == 6:
        task = worker_total()
        print(task)
    elif Choice == 7:
        task = patient_particular()
        print(task)
    elif Choice == 8:
        task = doctor_particular()
        print(task)
    elif Choice == 9:
        task = worker_particular()
        print(task)
    elif Choice == 10:
        task = patient_update()
        print(task)
    elif Choice == 11:
        task = patient_delete()
        print(task)
    elif Choice == 12:
        task = feedback()
        print(task)
    elif Choice == 13:
        print("\n\t\tBye! Stay Happy & Healthy 😀\n")
        exit()  #Exits the system
    else:
```

```
        print("Please enter your Choice between (1-13)")
        continue  #Prompts user back to enter the choice of choice not
between (1-13)
    ans = input("\nWant to get any other information?......(Yes/No): ")
else:
    print("\n\t\tBye! Stay Happy & Healthy 😀\n")

cursor = con.close()  #Closes the connection with MySQL
```

Output:

Initial Output Screen:

IDLE Shell 3.9.7

File Edit Shell Debug Options Window Help

```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\rajmo\OneDrive - Aakash Educational Services Ltd\computer Science XII\Project_Code
Successfully connected.
|_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_|
                         HOSPITAL MANAGEMENT SYSTEM
|_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_|


         |_-_-_-_-_-_-_---------------------------------_-_-_-_-_-_-_|
         |                   Login/Sign-up Page                      |
         ||_-_-_-_-_-_-_---------------------------------_-_-_-_-_-_-||

                     1. Login
                     2. Sign-up as a new user
                     3. Update User-credentials

Enter your choice: 
```

New User when Tries to Log-in:

```
Successfully connected.
|_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_|
                         HOSPITAL MANAGEMENT SYSTEM
|_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_|


         |_-_-_-_-_-_-_---------------------------------_-_-_-_-_-_-_|
         |                   Login/Sign-up Page                      |
         ||_-_-_-_-_-_-_---------------------------------_-_-_-_-_-_-||

                     1. Login
                     2. Sign-up as a new user
                     3. Update User-credentials

Enter your choice: 1

                Existing User Login

Enter Username: Ashwani

Enter Password: ENG12345

No user found with this username!

Would you like to sign-up for a new account?...(Yes/No): Yes


                New User Sign-up


Enter Name: Ashwani

Create Password: ENG12345
Confirm your Password: ENG12345

Sucessfully Registered!🔒
Welcome!
```

Existing User when tries to Log-in:

```
Successfully connected.
|_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-|
                        HOSPITAL MANAGEMENT SYSTEM
|_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-|


         |_-_-_-_-_-_-_________________________________-_-_-_-_-_-_|
         |                    Login/Sign-up Page                    |
         ||_-_-_-_-_-_---------------------------------_-_-_-_-_-_||

                    1. Login
                    2. Sign-up as a new user
                    3. Update User-credentials

Enter your choice: 1

                Existing User Login

Enter Username: Ashwani

Enter Password: ENG12345
Welcome!
```

Sign-up for New Account:

Entering different password in “Create Password” and “Confirm your Password” section prompts the user to Re-Enter the passwords again.

```
Successfully connected.
|_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-|
                        HOSPITAL MANAGEMENT SYSTEM
|_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-|


         |_-_-_-_-_-_-_________________________________-_-_-_-_-_-_|
         |                    Login/Sign-up Page                    |
         ||_-_-_-_-_-_---------------------------------_-_-_-_-_-_||

                    1. Login
                    2. Sign-up as a new user
                    3. Update User-credentials

Enter your choice: 2

Would you like to sign-up for a new account?...(Yes/No): Yes


                New User Sign-up


Enter Name: Harsh

Create Password: CS12345
Confirm your Password: Cs12345

Please ReEnter your password again:
Create Password: CS12345
Confirm your Password: CS12345

Sucessfully Registered!
Welcome!
```

Update User-credentials when user is already having an account:

```
Successfully connected.
|_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_|
                         HOSPITAL MANAGEMENT SYSTEM
|_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_|


         |_-_-_-_-_-_---------------------------------_-_-_-_-_-_-_|
         |                   Login/Sign-up Page                    |
         ||_-_-_-_-_-_--------------------------------_-_-_-_-_-_-||

                     1. Login
                     2. Sign-up as a new user
                     3. Update User-credentials

Enter your choice: 3

                User Old Credentials check

Enter old username: Harsh

Enter old password: CS12345

                User Credentials Updater


Enter new Username: Harsh

Enter new Password: ENG54321
Confirm Password: ENG54321

Successfully Updated!
```

Update User-credentials when user is not having an account:

```
Successfully connected.
|_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_|
                         HOSPITAL MANAGEMENT SYSTEM
|_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_|


         |_-_-_-_-_-_---------------------------------_-_-_-_-_-_-_|
         |                   Login/Sign-up Page                    |
         ||_-_-_-_-_-_--------------------------------_-_-_-_-_-_-||

                     1. Login
                     2. Sign-up as a new user
                     3. Update User-credentials

Enter your choice: 3

                User Old Credentials check

Enter old username: Jagdeesh

Enter old password: PY12345

No user found with this username!

Would you like to sign-up for a new account?...(Yes/No): No

                Thanks for visiting!
```

User Tries to Log-in but enters wrong password multiple times:

```
Successfully connected.
|_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-|
                         HOSPITAL MANAGEMENT SYSTEM
|_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-|


         |_-_-_-_-_-_---------------------------------_-_-_-_-_-_-|
         |                   Login/Sign-up Page                    |
         ||_-_-_-_-_-_---------------------------------_-_-_-_-_-_||

                     1. Login
                     2. Sign-up as a new user
                     3. Update User-credentials

Enter your choice: 1

                Existing User Login

Enter Username: Ashwani

Enter Password: sc
Wrong Password! Please try again.
Enter Password: cs
Wrong Password! Please try again.
Enter Password: ENG
Wrong Password! Please try again.
Enter Password: EN123
Wrong Password! Please try again.
Enter Password: ENG12321
Wrong Password! Please try again.

You have entered your password incorrect 4 times. Please wait 60 seconds to enter password again.

Enter Password: ENG
Wrong Password! Please try again.
Enter Password: ENG
Wrong Password! Please try again.
Enter Password: weew
Wrong Password! Please try again.
Enter Password: python
Wrong Password! Please try again.

You have entered your password incorrect 4 times. Please wait 60 seconds to enter password again.

Enter Password: project
Wrong Password! Please try again.
Enter Password: signin
Wrong Password! Please try again.
Enter Password: ENG
Wrong Password! Please try again.
Enter Password: ENG123
Wrong Password! Please try again.

You have entered wrong password multiple times. Try Again later!


                Thanks for visiting!

>>> |
```

Patient Registration:

```
Successfully connected.
|_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_|
                         HOSPITAL MANAGEMENT SYSTEM
|_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_|


         |_-_-_-_-_-_-_________________________________-_-_-_-_-_-_|
         |                     Login/Sign-up Page                  |
         ||_-_-_-_-_-_________________________________-_-_-_-_-_||

                     1. Login
                     2. Sign-up as a new user
                     3. Update User-credentials

Enter your choice: 1

                Existing User Login

Enter Username: Ashwani

Enter Password: ENG12345
Welcome!

                 ||___________ENTER YOUR CHOICE______________||

                     ------------------------------------
                     | 1. Register Patient's details    |
                     | 2. Register Doctor's details     |
                     | 3. Register Worker's details     |
                     |----------------------------------|
                     | 4. Total Patient's details       |
                     | 5. Total Doctor's details        |
                     | 6. Total Worker's details        |
                     |----------------------------------|
                     | 7. Particular Patient's detail   |
                     | 8. Particular Doctor's details   |
                     | 9. Particular Worker's details   |
                     |----------------------------------|
                     | 10. Update Patient's Status      |
                     | 11. Delete Patient's Record      |
                     |----------------------------------|
                     | 12. Feedback                     |
                     | 13. Exit the System              |
                     ------------------------------------

Enter your choice: 1

PATIENT REGISTERATION

Enter Patient's name: John Mishra
Enter Patient's gender: M
Enter Patient's age: 25
Enter Patient's ailment: Diabetes
Enter Patient's phone number: 989878987
Enter your current Status.....(Ill/Well): Ill

Sucessfully Registered!☺☺☺

Want to get any other information?......(Yes/No): No

                Bye! Stay Happy & Healthy ☺

>>> 
```

Doctor’s Registration: Login Section has been omitted in the below given output to avoid redundancy.

HARWARE/SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS: -

I. **Operating system**: - Windows 10 or above

II. **Processor**: - Pentium(any) or AMD
ATHALON (3800+-4200+dual core)

III. **Motherboard**: - 1.845 or 915,955 for Pentium or MSI
k9mm- via k8m800+8237r plus chipset for AMD ATHALON

IV. **RAM**: - 512 MB+

v. **Hard disk**: - 40 SATA GB or above

vi. **CD/DVD r/w multi drive combo**: (if back up required)

vii. **Floppy drive 1.44 MB**: (if backup required)

viii. **Monitor:** 14.1 or 15 -17 inch

ix. **Key board and mouse**

x. **Printer** [(if print is required – {hard copy}]

SOFTWARE REQUIREMENTS: -

I. **Windows OS**
II. **Python**

BIBLIOGRAPHY

1. Computer science With Python – Class XI By: Sumita Arora

2. A Project Report on Blood Bank Management System (BBMS) By: Praveen M Jigjinni

3. Websites: https://www.w3resource.com

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7380987/