C.V. RAMAN GLOBAL UNIVERSITY BHUBANESHWAR

PYTHON PROGRAMMING

CASE STUDY REPORT ON

HOSPITAL MANAGEMENT SYSTEM

SUBMITTED BY:-	
SAMIR PARIDA	2201020119
SATYAKAM ACHARYA	2201020124
S. CHANDRASHEKHAR	2201020125
SOUMYADEEP ROY	2201020131
SUBHAM KUMAR SAHOO	2201020135
SUBRAT SAHOO	2201020137
LIPSHA PRIYADARSHINI SAHOO	2201020149
RAJSHREE TANMAYEE SWAIN	2201020155
KAUSHIK BISWAL	2201020158
PRIYANSHU BARNWAL	2201020159

UNDER THE GUIDANCE OF:

SUPRIYA PANIGRAHY

ASST. PROFFESSOR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DETAILS OF TEAM MEMBERS

SL.NO.	NAME	REG NO.	CONTACT NO.	EMAIL ID
1.	SAMIR PARIDA	2201020119		2201020119@cgu-odisha.ac.in
2.	SATYAKAM ACHARYA	2201020124		2201020124@cgu-odisha.ac.in
3.	S. CHANDRASHEKHAR	2201020125		2201020125@cgu-odisha.ac.in
4.	SOUMYADEEP ROY	2201020131		2201020131@cgu-odisha.ac.in
5.	SUBHAM KUMAR SAHOO	2201020135		2201020135@cgu-odisha.ac.in
6.	SUBRAT SAHOO	2201020137		2201020137@cgu-odisha.ac.in
7.	LIPSHA PRIYADARSHINI SAHOO	2201020149		2201020149@cgu-odisha.ac.in
8.	RAJSHREE TANMAYEE SWAIN	2201020155		2201020155@cgu-odisha.ac.in
9.	KAUSHIK BISWAL	2201020158		2201020158@cgu-odisha.ac.in
10	PRIYANSHU BARNWAL	2201020159		2201020159@cgu-odisha.ac.in

INDEX PAGE

INTRODUCTION

In today’s fast-paced world, hospitals and medical organizations are finding it increasingly difficult to manage all the patients’ data, medical records, and treatment plans efficiently. To simplify these processes, a software solution is needed that can automate these tasks and help hospital staff manage patients’ information, keep track of medical appointments, and handle financial transactions.

This project aims to create a Hospital Management System using Python code that will help medical organizations manage all their patients’ data efficiently and easily. The system will allow users to perform several operations, including registering patients, retrieving patient details, providing with a discharge report and generating bills.

OBJECTIVES

The main objectives of this project are:

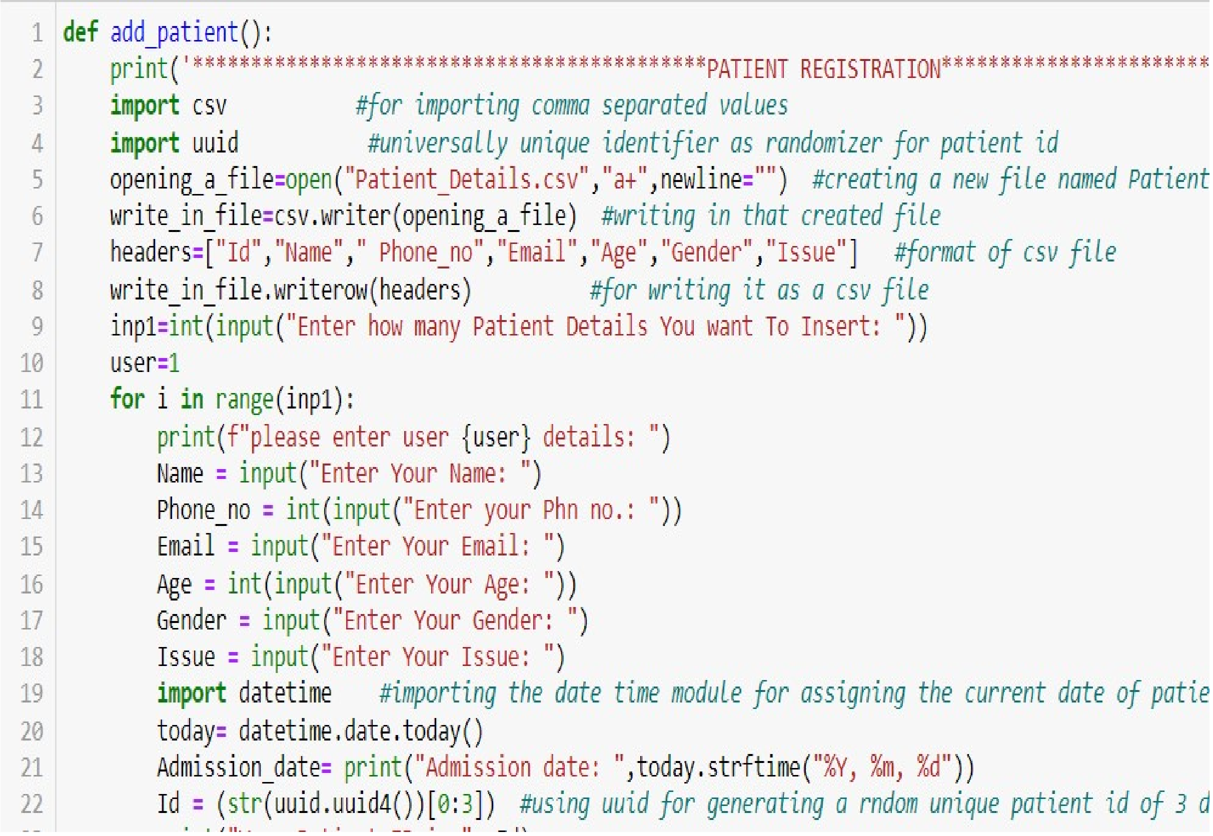
- To create a centralized database of patients' information that can be easily accessed by hospital staff.
- To provide a user-friendly interface that will allow medical staff to perform various operations such as registering patients, retrieving patient details, providing with a discharge report and generating bills.
- To reduce manual errors in data management and increase accuracy in patient information.
- To reduce the time spent on managing patients' data, thereby increasing productivity.
- To improve the overall efficiency and speed of medical processes.

METHODOLOGY

This project was developed using Python programming language, which is a high-level programming language that is easy to learn and understand. Python is known for its user-friendly interface, powerful data structures, and dynamic typing, making it a great choice for this project.

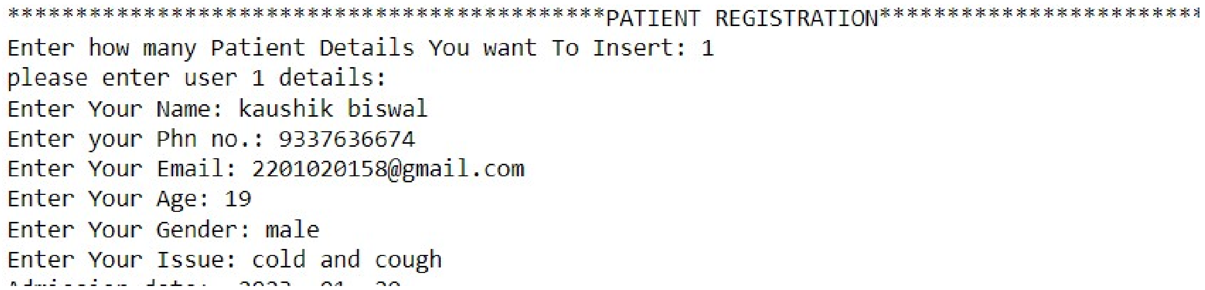
To develop the system, several modules were created, including:

Patient registration module: This module allows users to register new patients and store their personal and medical information in the database.



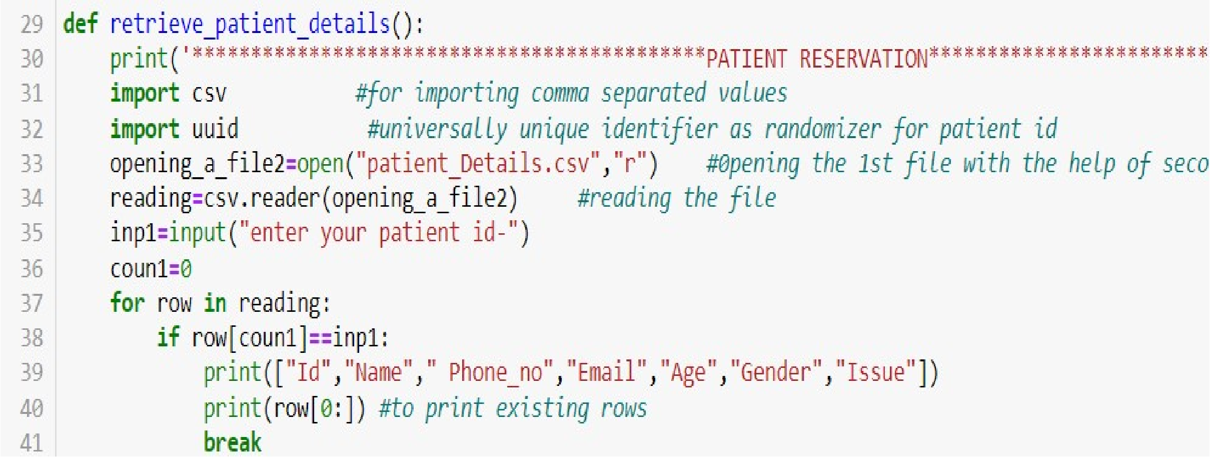
```
def add_patient():
    print('********************************************PATIENT REGISTRATION***********************
    import csv          #for importing comma separated values
    import uuid          #universally unique identifier as randomizer for patient id
    opening_a_file=open("Patient_Details.csv","a+",newline="")  #creating a new file named Patient
    write_in_file=csv.writer(opening_a_file)  #writing in that created file
    headers=["Id","Name"," Phone_no","Email","Age","Gender","Issue"]   #format of csv file
    write_in_file.writerow(headers)          #for writing it as a csv file
    inp1=int(input("Enter how many Patient Details You want To Insert: "))
    user=1
    for i in range(inp1):
        print(f"please enter user {user} details: ")
        Name = input("Enter Your Name: ")
        Phone_no = int(input("Enter your Phn no.: "))
        Email = input("Enter Your Email: ")
        Age = int(input("Enter Your Age: "))
        Gender = input("Enter Your Gender: ")
        Issue = input("Enter Your Issue: ")
        import datetime    #importing the date time module for assigning the current date of patie
        today= datetime.date.today()
        Admission_date= print("Admission date: ",today.strftime("%Y, %m, %d"))
        Id = (str(uuid.uuid4())[0:3])  #using uuid for generating a rndom unique patient id of 3 d
```

Firstly, we have defined a function which allows us to add details of a patient. We have imported CSV file for storing comma-separated value data and UUID for generating unique random identity number. Then we have opened a file naming “Patient_details.csv” with append function which allows us to add data to the csv file. Then we have added headers to distinguish which data goes to which element. Then using loops and conditions we have entered data for given patient(s). In the middle of the code we have also imported datetime module which allows us to record the admission date of the patient. After formatting the data according to our wish we have at last closed the file.



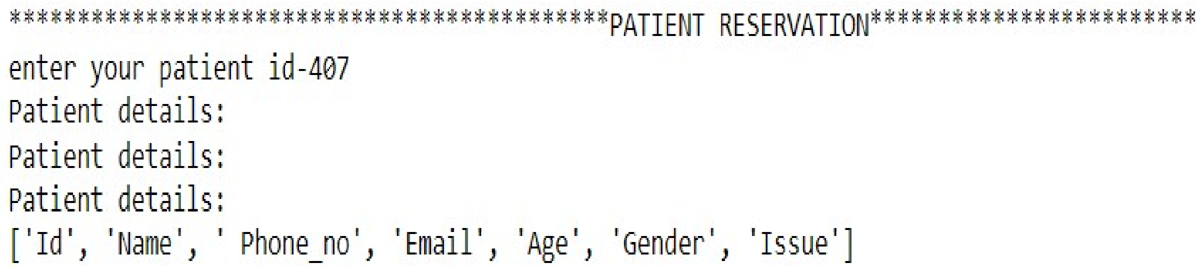
```
********************************************PATIENT REGISTRATION*************************
Enter how many Patient Details You want To Insert: 1
please enter user 1 details:
Enter Your Name: kaushik biswal
Enter your Phn no.: 9337636674
Enter Your Email: 2201020158@gmail.com
Enter Your Age: 19
Enter Your Gender: male
Enter Your Issue: cold and cough
Admission date:  2023 -01 -30
```

Patient details retrieving module: This module retrieves the patient details stored earlier in the system's database.



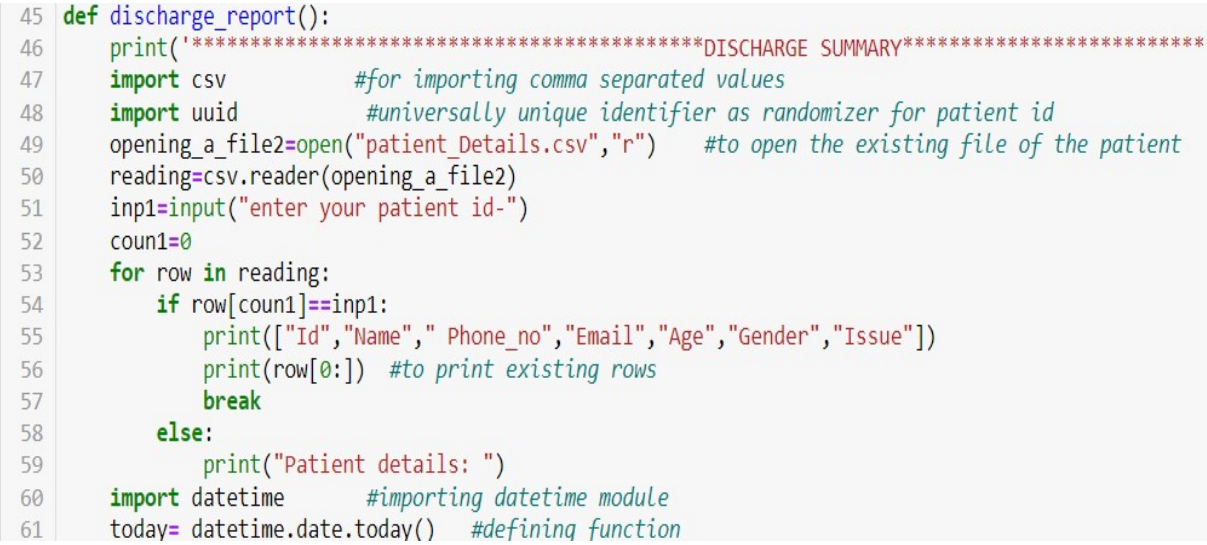
```
def retrieve_patient_details():
    print('********************************************PATIENT RESERVATION************************
    import csv          #for importing comma separated values
    import uuid          #universally unique identifier as randomizer for patient id
    opening_a_file2=open("patient_Details.csv","r")    #0pening the 1st file with the help of seco
    reading=csv.reader(opening_a_file2)     #reading the file
    inp1=input("enter your patient id-")
    coun1=0
    for row in reading:
        if row[coun1]==inp1:
            print(["Id","Name"," Phone_no","Email","Age","Gender","Issue"])
            print(row[0:]) #to print existing rows
            break
```

We have defined another function which allows us to retrieve the details of an existing patient. Again we have imported CSV file for accessing comma-separated value data and UUID for generating unique random identity number. We have the opened "Patient_details" file with read mode which allows us to read the accessed data from the csv file. Then we have taken a user input in which we will enter the patient id of an existing patient. Then using loops and conditions we get the patient details.



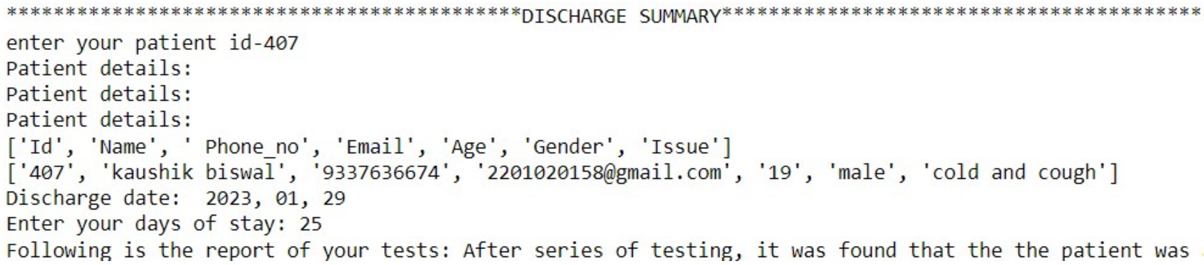
```
********************************************PATIENT RESERVATION************************
enter your patient id-407
Patient details:
Patient details:
Patient details:
['Id', 'Name', ' Phone_no', 'Email', 'Age', 'Gender', 'Issue']
```

Discharge report module: This module allows users tocommunicate vital information of the patient with the next healthcare provider.



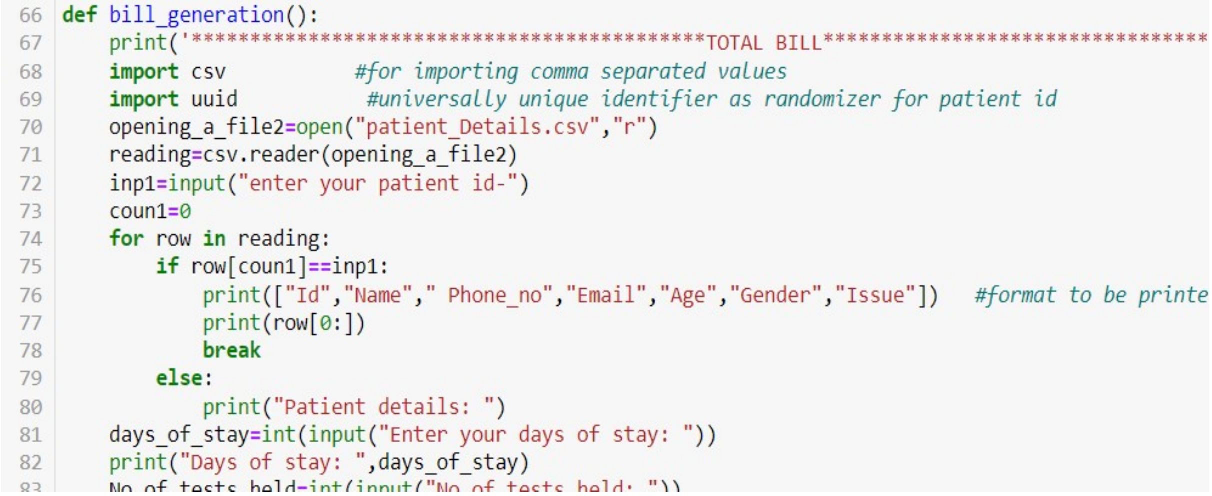
```
def discharge_report():
    print('*********************************************DISCHARGE SUMMARY*************************
    import csv           #for importing comma separated values
    import uuid           #universally unique identifier as randomizer for patient id
    opening_a_file2=open("patient_Details.csv","r")    #to open the existing file of the patient
    reading=csv.reader(opening_a_file2)
    inp1=input("enter your patient id-")
    coun1=0
    for row in reading:
        if row[coun1]==inp1:
            print(["Id","Name"," Phone_no","Email","Age","Gender","Issue"])
            print(row[0:])  #to print existing rows
            break
        else:
            print("Patient details: ")
    import datetime       #importing datetime module
    today= datetime.date.today()   #defining function
```

For making a discharge report we have followed the same previous steps to retrieve information of the patient. Then we have again imported datetime module for printing the discharge date. We have taken user input to enter the days of stay of the patient under the care of the hospital. Again we have taken user input in which the physician/doctor, under whose the patient was treated, will enter the report of the treatment and details about discharge of the patient.



```
********************************************DISCHARGE SUMMARY*****************************************
enter your patient id-407
Patient details:
Patient details:
Patient details:
['Id', 'Name', ' Phone_no', 'Email', 'Age', 'Gender', 'Issue']
['407', 'kaushik biswal', '9337636674', '2201020158@gmail.com', '19', 'male', 'cold and cough']
Discharge date:  2023, 01, 29
Enter your days of stay: 25
Following is the report of your tests: After series of testing, it was found that the the patient was
```

Financial transactions module: This module handles financial transactions such as bill report.

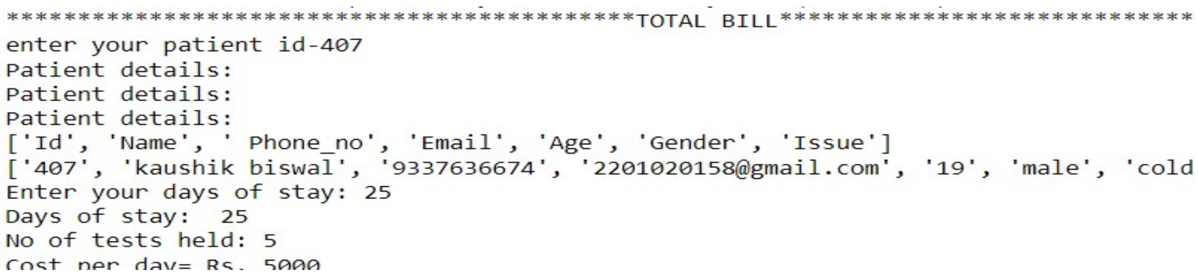


```
def bill_generation():
    print('********************************************TOTAL BILL*********************************
    import csv           #for importing comma separated values
    import uuid           #universally unique identifier as randomizer for patient id
    opening_a_file2=open("patient_Details.csv","r")
    reading=csv.reader(opening_a_file2)
    inp1=input("enter your patient id-")
    coun1=0
    for row in reading:
        if row[coun1]==inp1:
            print(["Id","Name"," Phone_no","Email","Age","Gender","Issue"])   #format to be printe
            print(row[0:])
            break
        else:
            print("Patient details: ")
    days_of_stay=int(input("Enter your days of stay: "))
    print("Days of stay: ",days_of_stay)
    No_of_tests_held=int(input("No of tests held: "))
```

For generating bill, we have followed the same previous steps to retrieve information of the patient. We have taken user input to enter the days of stay of the patient under the care of the hospital. Again we have taken user input to enter the number of tests that have been conducted on the patients. Then we have inputted hospital-defined reasonable cost per day of stay and cost of each tests conducted. Then we have calculated the total amount charged using the formula:

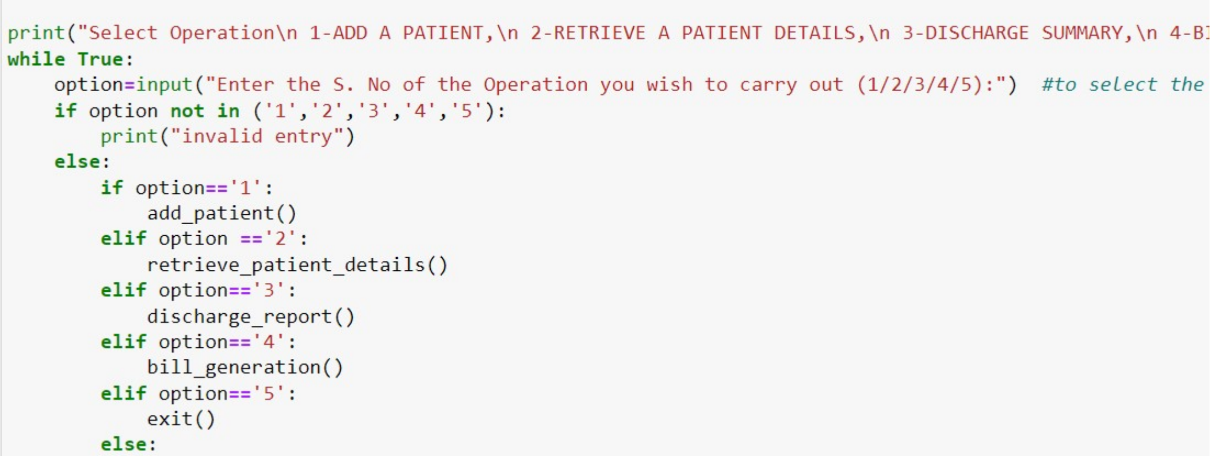
(days_of_stay*cost_per_day)+(No_of_tests_held*cost_per_test)

Which prints the amount charged on the patient.

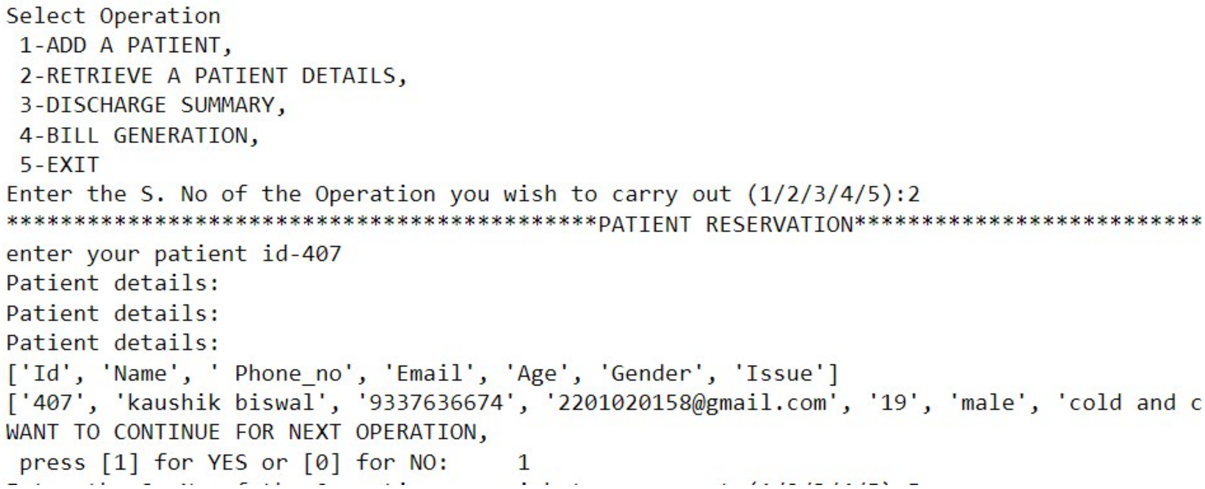


```
*******************************************TOTAL BILL******************************
enter your patient id-407
Patient details:
Patient details:
Patient details:
['Id', 'Name', ' Phone_no', 'Email', 'Age', 'Gender', 'Issue']
['407', 'kaushik biswal', '9337636674', '2201020158@gmail.com', '19', 'male', 'cold
Enter your days of stay: 25
Days of stay:  25
No of tests held: 5
Cost per day= Rs. 5000
```

We have included more codes to make the program work more efficiently, user-friendly and errorless.



```
print("Select Operation\n 1-ADD A PATIENT,\n 2-RETRIEVE A PATIENT DETAILS,\n 3-DISCHARGE SUMMARY,\n 4-BI
while True:
    option=input("Enter the S. No of the Operation you wish to carry out (1/2/3/4/5):")  #to select the
    if option not in ('1','2','3','4','5'):
        print("invalid entry")
    else:
        if option=='1':
            add_patient()
        elif option =='2':
            retrieve_patient_details()
        elif option=='3':
            discharge_report()
        elif option=='4':
            bill_generation()
        elif option=='5':
            exit()
        else:
```



```
Select Operation
 1-ADD A PATIENT,
 2-RETRIEVE A PATIENT DETAILS,
 3-DISCHARGE SUMMARY,
 4-BILL GENERATION,
 5-EXIT
Enter the S. No of the Operation you wish to carry out (1/2/3/4/5):2
*********************************************PATIENT RESERVATION**************************
enter your patient id-407
Patient details:
Patient details:
Patient details:
['Id', 'Name', ' Phone_no', 'Email', 'Age', 'Gender', 'Issue']
['407', 'kaushik biswal', '9337636674', '2201020158@gmail.com', '19', 'male', 'cold and c
WANT TO CONTINUE FOR NEXT OPERATION,
 press [1] for YES or [0] for NO:     1
```

Thus, the system was tested using various test cases to ensure that it was functioning correctly. The test cases included registering new patients, retrieving patient details, providing with a discharge report and generating bills.

RESULTS

The results of this project showed that the Hospital Management System using Python code was able to manage patients’ data efficiently and accurately. The system was able to reduce manual errors in data management and increase accuracy in patient information. Additionally, the user-friendly interface made it easy for medical staff to perform various operations such as registering patients, retrieving patient details, providing with a discharge report and generating bills.

CONCLUSION

In conclusion, the Hospital Management System using Python code was a success, meeting all the objectives of the project. The system was able to manage patients’ data efficiently, reduce manual errors in data management, and increase accuracy in patient information. The user-friendly interface and the various modules created made it easy for medical staff to perform various operations, increasing productivity and improving the overall efficiency and speed of medical processes.

REFERENCE

- https://www.geeksforgeeks.org
- https://chat.openai.com/chat
- https://www.youtube.com/
- https://www.google.com/