ADVANCED JAVA

AMITY INSTITUTE OF INFORMATION TECHNOLOGY

LAB-2

Name : Priya Kumari
Course : Advanced java
Program/Semester : BCA – 6 ‘B’
Enrollment Number : A45304821056

Submitted to :-

Dr. Naveen Kumar Singh

Department:- Amity Institude Of Information Technology
Session:- 2021-2024

CRUD OPERATIONS

Problem description :

The problem description for JDBC CRUD operations typically involves creating, reading, updating, and deleting records in a relational database using Java Database Connectivity (JDBC).The application should:

1. Provide options to perform CRUD operations including inserting new records into the database table, retrieving existing records from the table based on specified criteria, updating records in the table and deleting records from the table.

2. Implement error handling to manage connection failures and database operation exceptions gracefully.

3. SQL Syntax Error: Double check your SQL queries for syntax errors,Incorrect queries can lead to unexpected results or failures.

The application should focus on simplicity and functionality, serving as a basic template for JDBC usage in CRUD operations

DESIGN

The design of the problem statement for creating a simple Java application that establishes JDBC connection and performs CRUD operations involves several key components and considerations:

1. User Interface Design :

Upon running the application, users will be presented with a menu containing 5 options, with 4 of them representing crud operations(“Add New Patients”,”Display all Patient”,”Update Name of Patients,”Delete an Patients”)
and the last option for exiting the application gracefully. Based on the user's choice, the application will invoke the appropriate method from the Patient class to perform the CRUD operation.

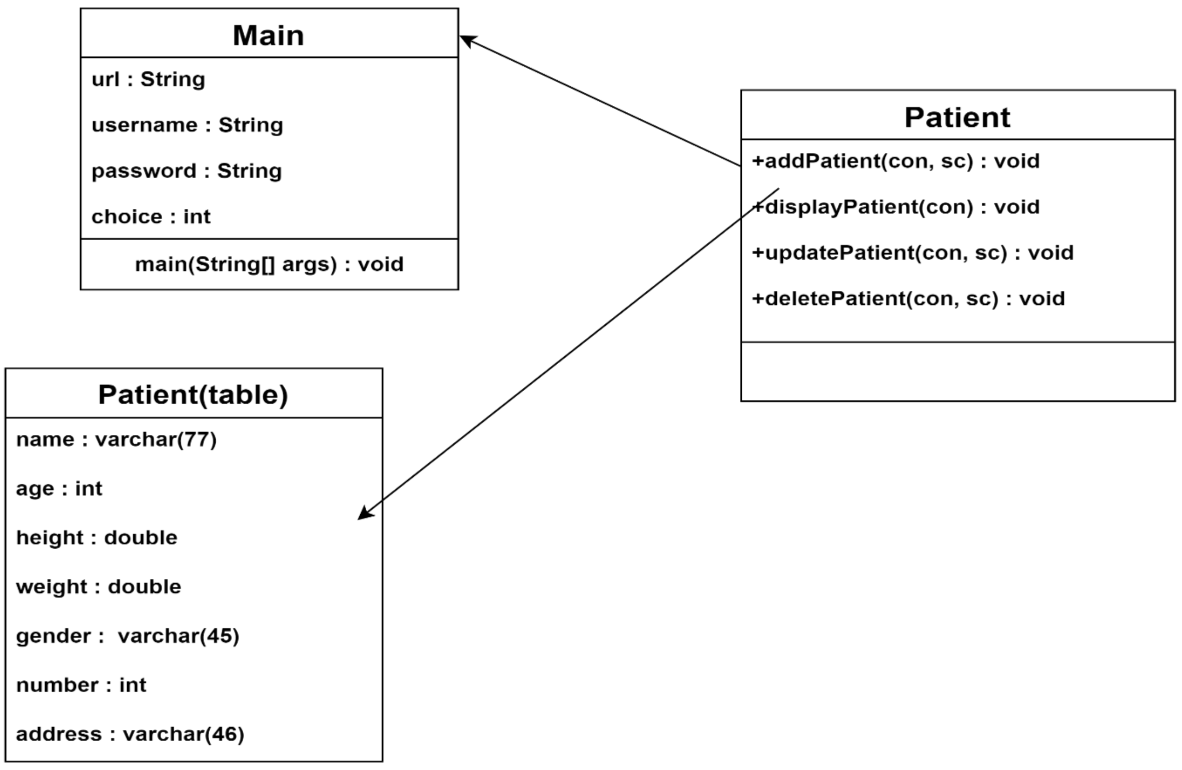
2. Database Connection Management:

The application needs to establish a JDBC connection with the relational database system using the correct connection details.

3. Error Handling:
Error handling should be implemented to manage exceptions during database operations.

4. Code Modularity and Maintainability:
The application's code should be modular and well-organized, following best practices in software design and development. It should be easy to maintain and extend, allowing for future enhancements or modifications without significant refactoring.
5.Class Diagram:

A class diagram is crucial for design purposes as it visually illustrates the structure, relationships, and behavior of classes within a system. It aids in organizing and conceptualizing software components, facilitating communication among developers, guiding implementation, and ensuring consistency and scalability throughout the design process. Here's a class diagram demonstrating our problem statement -



CODE

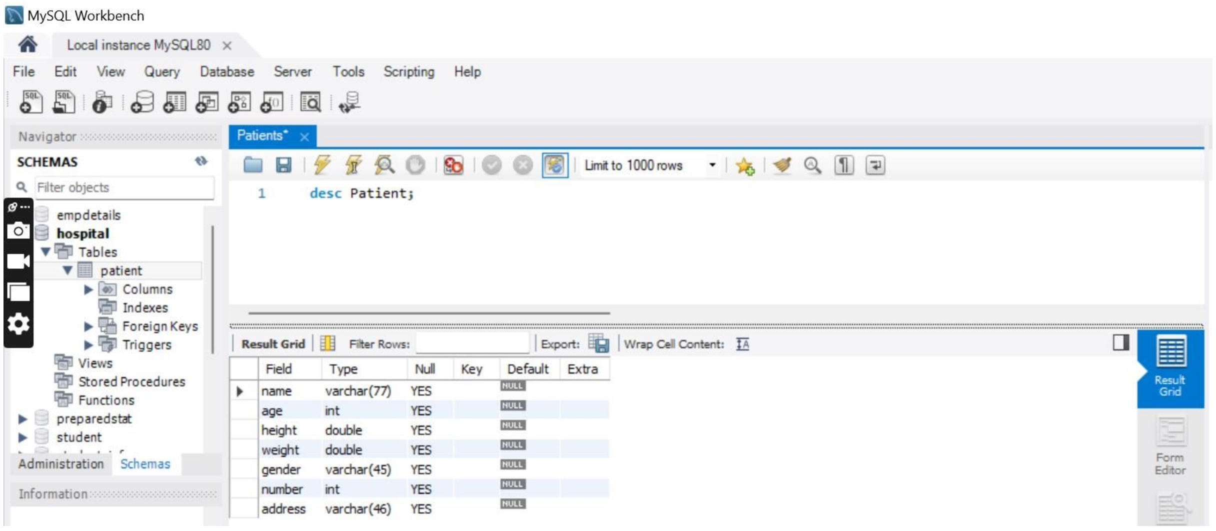
Patient.java

```
package patientdetail;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

public class Patient {

            // TODO Auto-generated constructor stub



      public void addPatient(Connection con, Scanner sc) throws
SQLException
      {
            Statement st = con.createStatement();

            //read Patient details
            System.out.println("Enter Patient Name");
            String name = sc.next();

            System.out.println("Enter Patient Age");
            int age = sc.nextInt();

            System.out.println("Enter Patient height");
            double height = sc.nextDouble();

            System.out.println("Enter Patient weight");
            double weight = sc.nextDouble();

            System.out.println("Enter Patient Gender");
            String gender = sc.next();

            System.out.println("Enter Patient Phone no.");
            int number = sc.nextInt();

            System.out.println("Enter Patient Address");
            String address = sc.next();


            //create sql squery string
            String query = String.format("Insert Into Patient values('%s',
'%d', '%f', '%f',  '%s', '%d', '%s') ", name, age,height,weight, gender,
number, address);

            //execute sql query
            int rows = st.executeUpdate(query);

            System.out.println(rows + " record inserted!!!");
      }

      public void displayPatient(Connection con) throws SQLException
```

```
        {
                Statement st = con.createStatement();

                ResultSet rs = st.executeQuery("select * from Patient");
                System.out.println("Displaying all Patient info");

                while(rs.next()) {
                        System.out.println(rs.getString(1)+ "\t"+rs.getInt(2)+
"\t"+
rs.getDouble(3)+"\t"+rs.getDouble(4)+"\t"+rs.getString(5)+"\t"+rs.getInt(6)
+"\t"+rs.getString(7));
                }
        }

        public void updatePatient(Connection con, Scanner sc) throws
SQLException {
                Statement st = con.createStatement();
                System.out.println("Enter Patient Name: ");
                String name = sc.next();
                System.out.println("Enter Patient Age: ");
                int age = sc.nextInt();
                System.out.println("Enter Patient New Address: ");
                String address = sc.next();

                String query = String.format("UPDATE Patient SET address='%s'
WHERE Name='%s' AND Age=%d", address, name, age);
                int rowsAffected = st.executeUpdate(query);

                System.out.println(rowsAffected+" recored updated!!!");

        }


        public void deletePatient(Connection con, Scanner sc) throws
SQLException {
                Statement st = con.createStatement();
                System.out.println("Enter Patient Name: ");
                String name = sc.next();

                String query = String.format("delete from Patient where
name='%s'", name);
              int rowsAffected = st.executeUpdate(query);

                System.out.println(rowsAffected + " recored deleted!!!");
        }


        public static void main(String[] args) throws ClassNotFoundException,
SQLException {
                // TODO Auto-generated method stub
                Class.forName("com.mysql.cj.jdbc.Driver");

                String url = "jdbc:mysql://localhost:3306/hospital";
                String username = "root";
                String pwd = "Root@123";
                Connection con = DriverManager.getConnection(url, username,
pwd);

                Scanner sc = new Scanner(System.in);

                Patient pi = new Patient();
        while(true) {

                        menu();
                        int choice = sc.nextInt();
                        switch(choice) {
                        case 1: pi.addPatient(con, sc);
                              break;

                        case 2: pi.displayPatient(con);
                              break;

                        case 3: pi.updatePatient(con, sc);
                              break;

                        case 4: pi.deletePatient(con, sc);
                              break;

                        case 5:
                              System.out.println("Bye Bye ...");
                              System.exit(0);

                        default:
                              System.out.println("Wrong Choice...");
                        }

                }
                }



        public static void menu() {
                System.out.println("------------Operation perform-----------");
                System.out.println("1. Add New Patient");
                System.out.println("2. Display All Patients");
                System.out.println("3. Update Name of Patient");
                System.out.println("4. Delete a Patient details");
                System.out.println("5. Exit");
                System.out.println("Your Choice...");
        }
                }
```

INPUT/OUTPUT

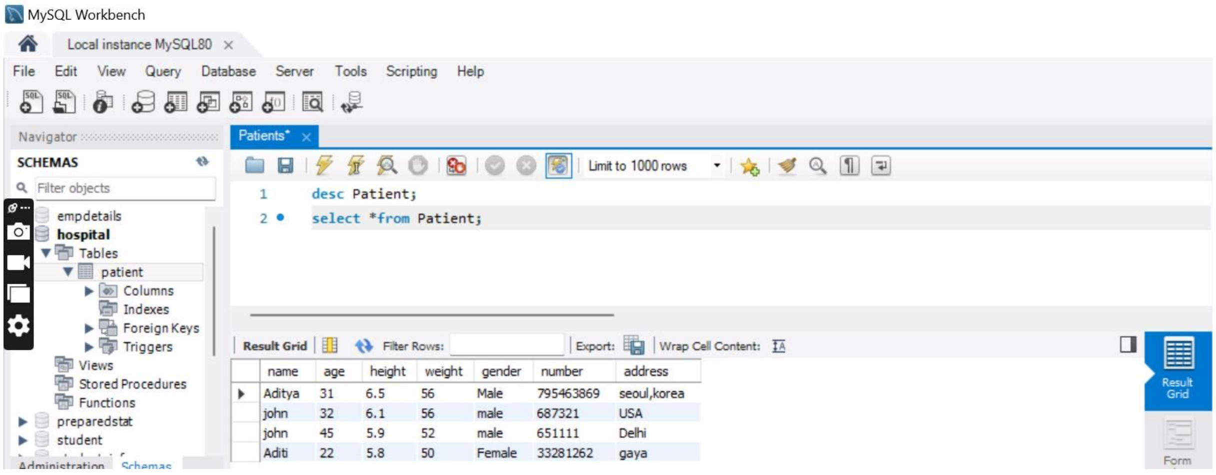
Describe The Table –



```
desc Patient;
```

	Field	Type	Null	Key	Default	Extra
▶	name	varchar(77)	YES		NULL	
	age	int	YES		NULL	
	height	double	YES		NULL	
	weight	double	YES		NULL	
	gender	varchar(45)	YES		NULL	
	number	int	YES		NULL	
	address	varchar(46)	YES		NULL	

Select The Table –

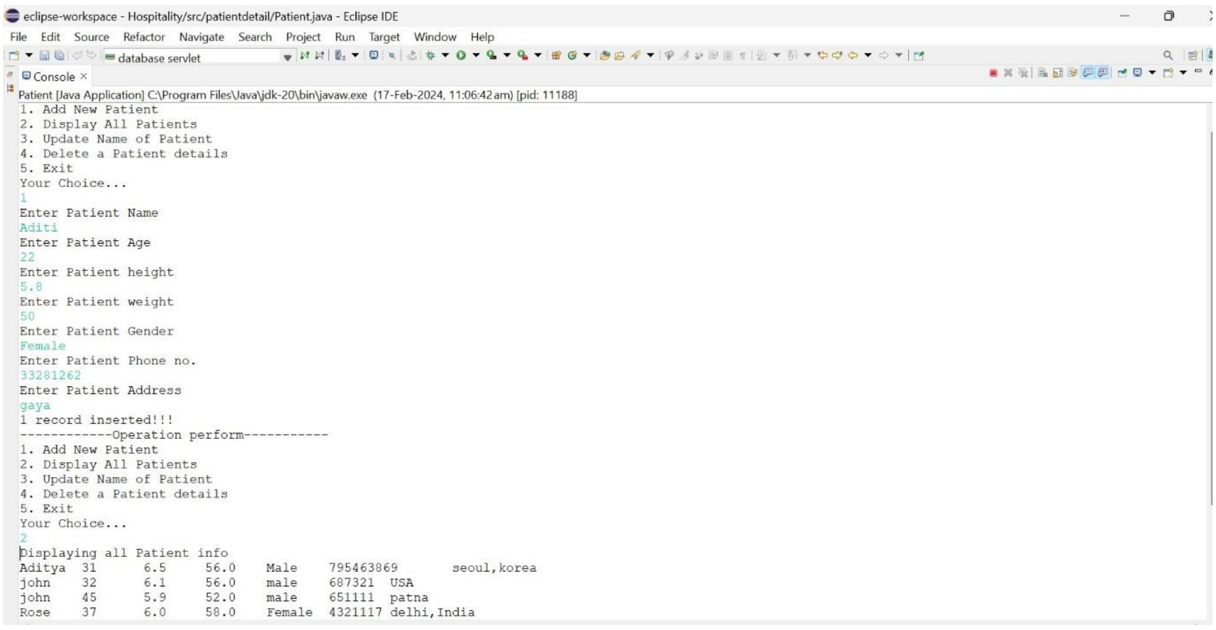


```
desc Patient;
select *from Patient;
```

	name	age	height	weight	gender	number	address
▶	Aditya	31	6.5	56	Male	795463869	seoul,korea
	john	32	6.1	56	male	687321	USA
	john	45	5.9	52	male	651111	Delhi
	Aditi	22	5.8	50	Female	33281262	gaya

CRUD Operation perform

Inserting operation

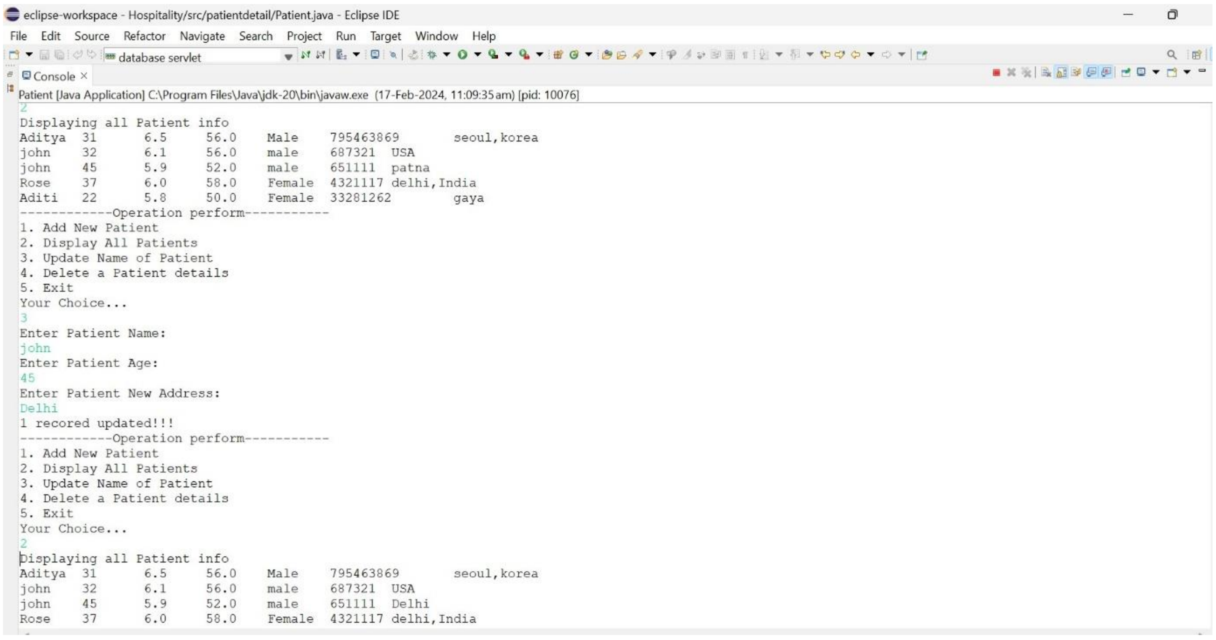


eclipse-workspace - Hospitality/src/patientdetail/Patient.java - Eclipse IDE

Patient [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (17-Feb-2024, 11:06:42 am) [pid: 11188]

```
1. Add New Patient
2. Display All Patients
3. Update Name of Patient
4. Delete a Patient details
5. Exit
Your Choice...
1
Enter Patient Name
Aditi
Enter Patient Age
22
Enter Patient height
5.8
Enter Patient weight
50
Enter Patient Gender
Female
Enter Patient Phone no.
33281262
Enter Patient Address
gaya
1 record inserted!!!
------------Operation perform-----------
1. Add New Patient
2. Display All Patients
3. Update Name of Patient
4. Delete a Patient details
5. Exit
Your Choice...
2
Displaying all Patient info
Aditya  31      6.5     56.0    Male    795463869       seoul,korea
john    32      6.1     56.0    male    687321  USA
john    45      5.9     52.0    male    651111  patna
Rose    37      6.0     58.0    Female  4321117 delhi,India
```

Display

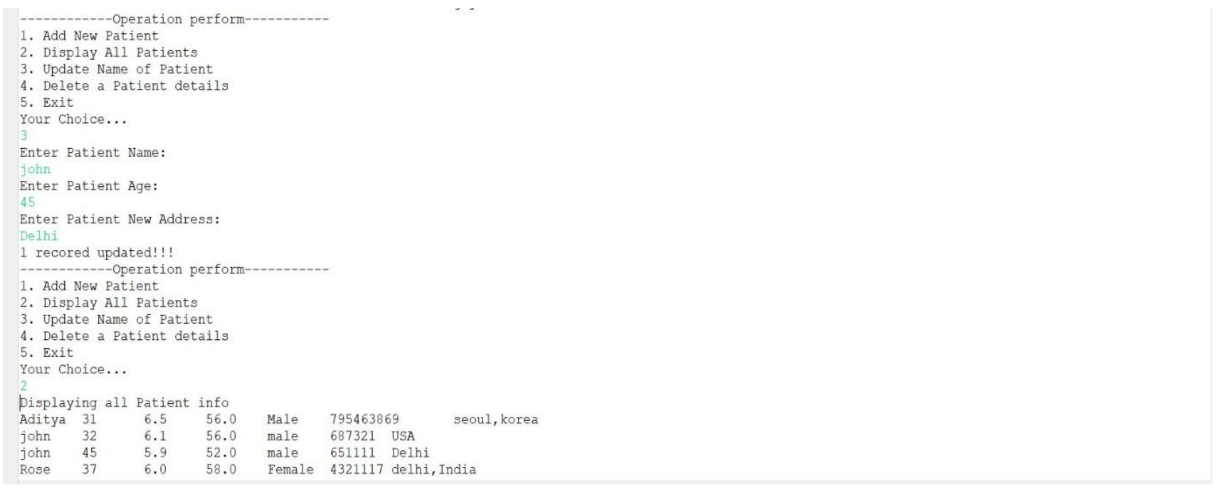


eclipse-workspace - Hospitality/src/patientdetail/Patient.java - Eclipse IDE

Patient [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (17-Feb-2024, 11:09:35 am) [pid: 10076]

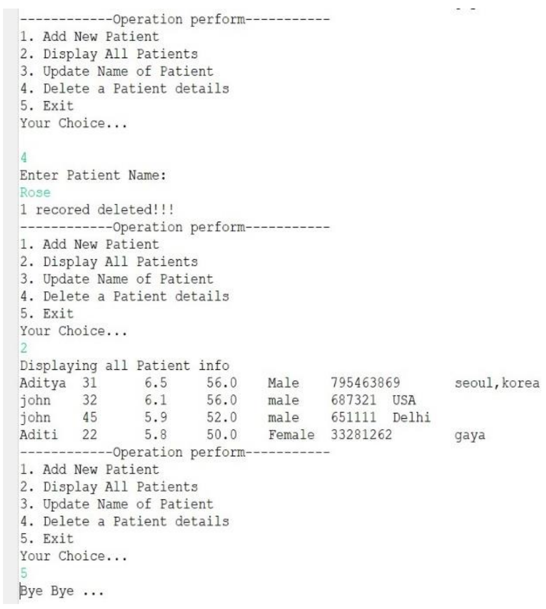
```
2
Displaying all Patient info
Aditya  31      6.5     56.0    Male    795463869       seoul,korea
john    32      6.1     56.0    male    687321  USA
john    45      5.9     52.0    male    651111  patna
Rose    37      6.0     58.0    Female  4321117 delhi,India
Aditi   22      5.8     50.0    Female  33281262        gaya
------------Operation perform-----------
1. Add New Patient
2. Display All Patients
3. Update Name of Patient
4. Delete a Patient details
5. Exit
Your Choice...
3
Enter Patient Name:
john
Enter Patient Age:
45
Enter Patient New Address:
Delhi
1 recored updated!!!
------------Operation perform-----------
1. Add New Patient
2. Display All Patients
3. Update Name of Patient
4. Delete a Patient details
5. Exit
Your Choice...
2
Displaying all Patient info
Aditya  31      6.5     56.0    Male    795463869       seoul,korea
john    32      6.1     56.0    male    687321  USA
john    45      5.9     52.0    male    651111  Delhi
Rose    37      6.0     58.0    Female  4321117 delhi,India
```

Update Operation



```
------------Operation perform-----------
1. Add New Patient
2. Display All Patients
3. Update Name of Patient
4. Delete a Patient details
5. Exit
Your Choice...
3
Enter Patient Name:
john
Enter Patient Age:
45
Enter Patient New Address:
Delhi
1 recored updated!!!
------------Operation perform-----------
1. Add New Patient
2. Display All Patients
3. Update Name of Patient
4. Delete a Patient details
5. Exit
Your Choice...
2
Displaying all Patient info
Aditya  31      6.5     56.0    Male    795463869       seoul,korea
john    32      6.1     56.0    male    687321  USA
john    45      5.9     52.0    male    651111  Delhi
Rose    37      6.0     58.0    Female  4321117 delhi,India
```

Delete Operation



```
------------Operation perform-----------
1. Add New Patient
2. Display All Patients
3. Update Name of Patient
4. Delete a Patient details
5. Exit
Your Choice...

4
Enter Patient Name:
Rose
1 recored deleted!!!
------------Operation perform-----------
1. Add New Patient
2. Display All Patients
3. Update Name of Patient
4. Delete a Patient details
5. Exit
Your Choice...
2
Displaying all Patient info
Aditya  31      6.5     56.0    Male    795463869       seoul,korea
john    32      6.1     56.0    male    687321  USA
john    45      5.9     52.0    male    651111  Delhi
Aditi   22      5.8     50.0    Female  33281262        gaya
------------Operation perform-----------
1. Add New Patient
2. Display All Patients
3. Update Name of Patient
4. Delete a Patient details
5. Exit
Your Choice...
5
Bye Bye ...
```