HOSPITAL MANAGEMENT SYSTEM

19I420 DATABASE MANAGEMENT SYSTEMS LABORATORY

R.HARRISH	22I461
K.RAM KUMAR	21I346
ASHWIN KUMAR SS	21I305
NIRAINTHAN J	21I338
KEVIN CARROLL DENIS T	21I323

DEPARTMENT OF INFORMATION TECHNOLOGY

PSG COLLEGE OF TECHNOLOGY

COIMBATORE-641 004

HOSPITAL MANAGEMENT SYSTEM

➢ Introduction

Hospital management system is a software application designed to manage the day-to-day operations of a hospital or healthcare facility. It is used to automate and streamline various tasks such as patient registration, appointment scheduling, billing and invoicing and medical record keeping.

In this project, we will be using Java Swing to create a user-friendly graphical user interface (GUI) for the hospital management system. The system will be backed by a MySQL database to store and retrieve data.

The main objectives of the system are to improve the efficiency of the hospital's operations, enhance patient care, and reduce errors and redundancies. The system will provide various modules for different departments such as the room, wards available, doctor details, patient details, billing and invoice and users for login.

The reception module will be used to register patients, schedule appointments, and manage patient records. The medical staff module will be used to view patient records. The billing and accounts module will be used to generate invoices and process payments..

NEED FOR HOSPITAL MANAGEMENT SYSTEM:

- User-friendly interface
- Patient management
- Appointment scheduling
- Electronic Medical Records (EMR)
- Billing and invoicing
- Reporting and analytics

ADVANTAGES OF HOSPITAL MANAGEMENT SYSTEM:

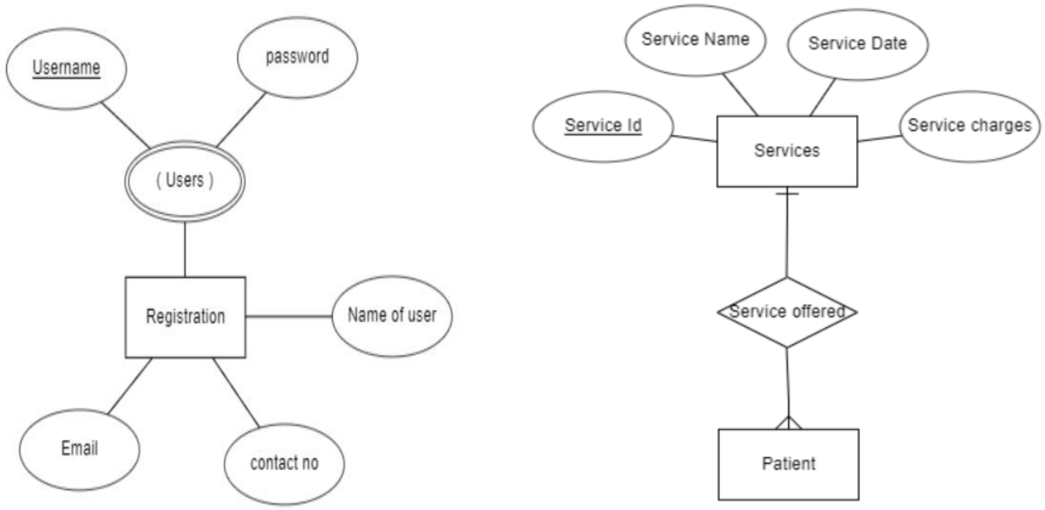
- Improved patient care
- Enhanced accuracy
- Improved communication
- Cost-effective
- Scalable

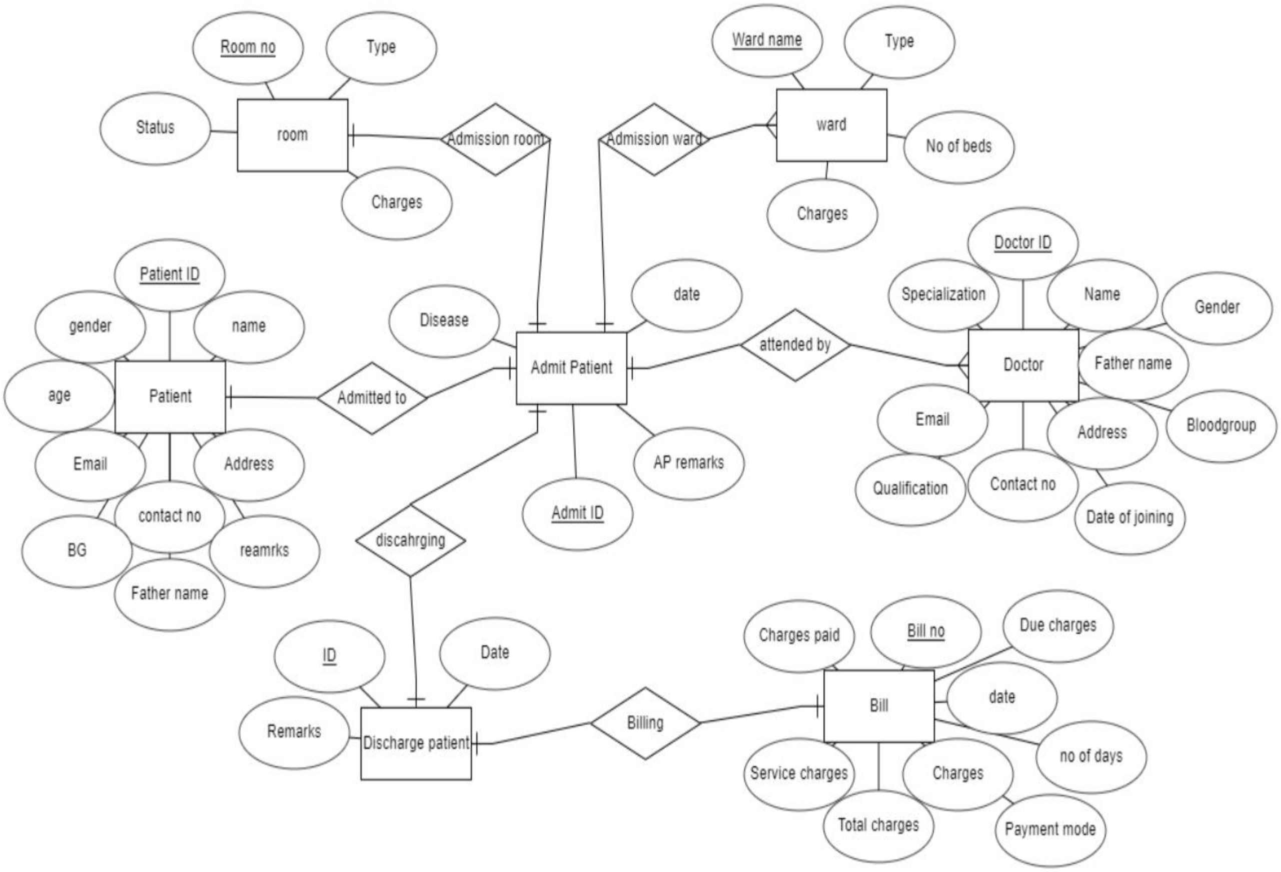
LANGUAGE USED FOR DESIGN:

FRONT-END: JAVA-SWING

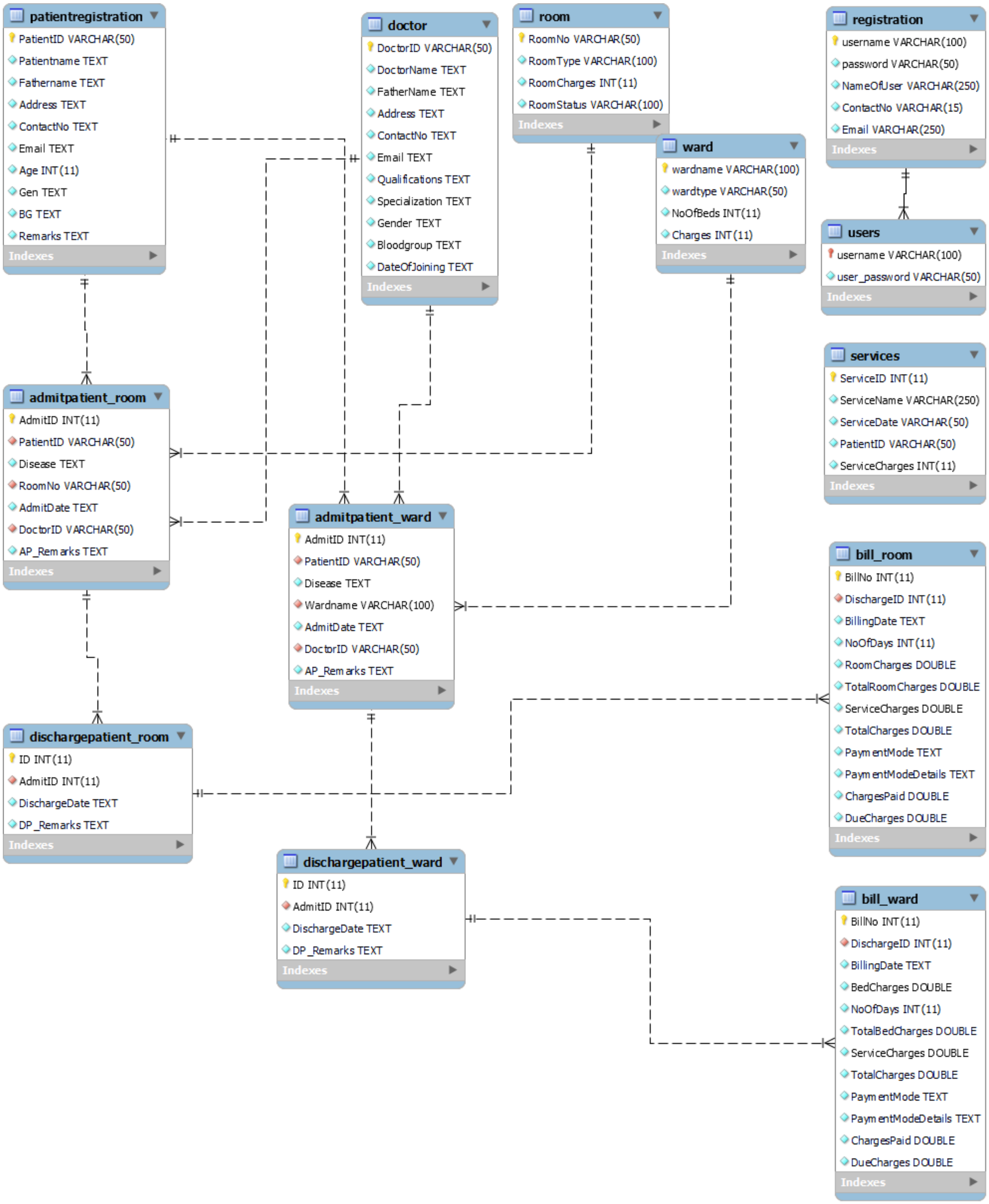
BACK-END: MYSQL DATABASE

➢ ER Diagram for Hospital Management System:





➢ RELATION SCHEMA MODEL:



➢ NORMALIZATION:

The Tables given are:

- Admitpatient_Room
- Admitpatient_Ward
- Bill_Woom
- Bill_Ward
- Dischargepatient_Room
- Dischargepatient_Ward
- Doctor
- PatientRegistration
- User
- Registration

Admitpatient_Room and Admitpatient_Ward:

Both the tables Admitpatient_Room and Admitpatient_Ward have the same structure and contain data related to patients who are admitted to a hospital. These tables are in the third normal form (3NF). The reasons are as follows:

1. First Normal Form (1NF): Each table has a primary key and no column contains multiple values.
2. Second Normal Form (2NF): Both tables are in 2NF because there is no partial dependency between the non-key columns and the primary key. All the non-key columns depend on the whole primary key.
3. Third Normal Form (3NF): Both tables are in 3NF because there is no transitive dependency between the non-key columns. All the non-key columns depend only on the primary key, and not on any other non-key column.

Bill_Room and Bill_Ward:

Both the tables Bill_room and Bill_ward have the same structure and contain data related to the billing information of patients who have been discharged from a hospital. These tables are in the third normal form (3NF). The reasons are as follows:

1. First Normal Form (1NF): Each table has a primary key and no column contains multiple values.
2. Second Normal Form (2NF): Both tables are in 2NF because there is no partial dependency between the non-key columns and the primary key. All the non-key columns depend on the whole primary key.
3. Third Normal Form (3NF): Both tables are in 3NF because there is no transitive dependency between the non-key columns. All the non-key columns depend only on the primary key, and not on any other non-key column.

Dischargepatient_Room and Dischargepatient_Ward:

Both the tables Dischargepatient_Room and Dischargepatient_Ward have the same structure and contain data related to the discharge information of patients who were admitted to a hospital. These tables are in the third normal form (3NF). The reasons are as follows:

1. First Normal Form (1NF): Each table has a primary key and no column contains multiple values.
2. Second Normal Form (2NF): Both tables are in 2NF because there is no partial dependency between the non-key columns and the primary key. All the non-key columns depend on the whole primary key.
3. Third Normal Form (3NF): Both tables are in 3NF because there is no transitive dependency between the non-key columns. All the non-key columns depend only on the primary key, and not on any other non-key column.

Doctor and PatientRegistration:

Both the tables Doctor and PatientRegistration contain data related to doctors and patients respectively. These tables are in the third normal form (3NF). The reasons are as follows:

1. First Normal Form (1NF): Each table has a primary key and no column contains multiple values.
2. Second Normal Form (2NF): Both tables are in 2NF because there is no partial dependency between the non-key columns and the primary key. All the non-key columns depend on the whole primary key.
3. Third Normal Form (3NF): Both tables are in 3NF because there is no transitive dependency between the non-key columns. All the non-key columns depend only on the primary key, and not on any other non-key column.

User and Registration:

Both the tables contain data related to user for login for control the management respectively. These tables are in the third normal form (3NF). The reasons are as follows:

1. First Normal Form (1NF): Both tables are in 1NF, as each column contains only atomic values, and there are no repeating groups of columns.
2. Second Normal Form (2NF): Both tables are in 2NF, as there are no partial dependencies. In other words, all non-key columns depend on the entire primary key.
3. Third Normal Form (3NF): Both tables are also in 3NF, as there are no transitive dependencies. In other words, all non-key columns depend only on the primary key, and not on any other non-key column.

➢ Code:

CONNECTING DATABASE:

```
import java.sql.*;
import javax.swing.*;
public class Connect {
    Connection con=null;

      public static Connection ConnectDB(){
          try{

              Class.forName("com.mysql.jdbc.Driver");
              String s = "jdbc:mysql://localhost:3306/hms_db?useSSL=false";
              Connection con = DriverManager.getConnection(s, "root", "root");
        return con;

      }catch(ClassNotFoundException | SQLException e){
          JOptionPane.showMessageDialog(null, e);
          return null;
   }}}
```

LOGIN:

```
String sql= "select * from users where UserName= '" + txtUserName.getText() + "' and user_Password ='" +
txtPassword.getText() + "'";
try
{
   pst=con.prepareStatement(sql);
   rs= pst.executeQuery();
   if (rs.next()){
      this.hide();
      MainMenu frm=new MainMenu();
      frm.setVisible(true);
   }
   else{

     JOptionPane.showMessageDialog(null, "Login Failed..Try again !","Access
denied",JOptionPane.ERROR_MESSAGE);
   }}
```

CRUD ON ROOM:

SELECT:

```
Statement stmt;
stmt = con.createStatement();
String sql1 = "Select RoomNo from Room where RoomNo= '" + txtRoomNo.getText() + "'";
rs = stmt.executeQuery(sql1);
if (rs.next()) {
   JOptionPane.showMessageDialog(this, "Room No. already exists", "Error",
JOptionPane.ERROR_MESSAGE);
   txtRoomNo.setText("");
   txtRoomNo.requestDefaultFocus();
   return;
}
```

INSERT:

```
String sql = "insert into Room(RoomNo,RoomType,RoomCharges,RoomStatus)values('" +
txtRoomNo.getText() + "','" + cmbRoomType.getSelectedItem() + "'," + txtRoomCharges.getText() +
",'Vacant')";
pst = con.prepareStatement(sql);
pst.execute();

JOptionPane.showMessageDialog(this, "Successfully saved", "Room Record",
JOptionPane.INFORMATION_MESSAGE);
btnSave.setEnabled(false);
Get_Data();
```

UPDATE:

```
con=Connect.ConnectDB();
 String sql= "update Room set Roomtype='"+ cmbRoomType.getSelectedItem() + "',RoomCharges=" +
txtRoomCharges.getText() + " where RoomNo='" + txtRoomNo.getText() + "'";
 pst=con.prepareStatement(sql);
 pst.execute();
 JOptionPane.showMessageDialog(this,"Successfully updated","Room
Record",JOptionPane.INFORMATION_MESSAGE);
 btnUpdate.setEnabled(false);
```

DELETE:

```
   int P = JOptionPane.showConfirmDialog(null," Are you sure want to delete
?","Confirmation",JOptionPane.YES_NO_OPTION);
if (P==0)
{
   con=Connect.ConnectDB();

   String sql= "delete from Room where RoomNo = '" + txtRoomNo.getText() + "'";
   pst=con.prepareStatement(sql);
   pst.execute();
   JOptionPane.showMessageDialog(this,"Successfully
deleted","Record",JOptionPane.INFORMATION_MESSAGE);
   Reset();
     }
```

CRUD ON WARD:

INSERT:

```
String sql= "insert into ward(Wardname,wardType,NoOfBeds,Charges)values('"+ txtWardName.getText() +
"','"+ cmbWardType.getSelectedItem() + "'," + txtNoOfbeds.getText() + "," + txtCharges.getText() + ")";
```

UPDATE:

```
String sql= "update Ward set Wardtype='"+ cmbWardType.getSelectedItem() + "',NoOfBeds=" +
txtNoOfbeds.getText() + ",Charges=" + txtCharges.getText() + " where Wardname='" +
txtWardName.getText() + "'";
```

DELETE:

```
String sql= "delete from ward where wardname = '" + txtWardName.getText() + "'";
```

SELECT:

```
String sql= "select * from ward where wardname = '" + table_click + "'";
```

CRUD ON REGISTRATION AND USER:

INSERTION IN BOTH TABLE:

```
String Password1= String.valueOf(txtPassword.getPassword());
 String sql= "insert into Registration(username,password,nameofuser,Email,ContactNo)values('"+
txtUserName.getText() + "','" + Password1 + "','" + txtName.getText() + "','" + txtEmailID.getText() + "','" +
txtContactNo.getText() + "')";

pst=con.prepareStatement(sql);
pst.execute();
 String sql2= "insert into Users(username,user_password)values('" + txtUserName.getText() + "','" +
Password1 + "')";

pst=con.prepareStatement(sql2);
pst.execute();
JOptionPane.showMessageDialog(this,"Successfully
Registered","User",JOptionPane.INFORMATION_MESSAGE);
Save.setEnabled(false);
```

DELETION IN BOTH TABLE:

```
String sql= "delete from Registration where Username = '" + txtUserName.getText() + "'";
pst=con.prepareStatement(sql);
pst.execute();
 String sql1= "delete from Users where Username = '" + txtUserName.getText() + "'";
pst=con.prepareStatement(sql1);
pst.execute();
JOptionPane.showMessageDialog(this,"Successfully
deleted","Record",JOptionPane.INFORMATION_MESSAGE);
Reset();
```

UPDATION IN BOTH TABLE:

```
con=Connect.ConnectDB();
String Password1= String.valueOf(txtPassword.getPassword());
  String sql= "update Registration set password='" + Password1 + "',nameofuser='" + txtName.getText() +
"',Email='" + txtEmailID.getText() + "',ContactNo='" + txtContactNo.getText() + "' where Username='" +
txtUserName.getText() + "'";

 pst=con.prepareStatement(sql);
 pst.execute();
  String sql2= "update Users set user_password='" + Password1 + "' where username='" +
txtUserName.getText() + "'";

 pst=con.prepareStatement(sql2);
 pst.execute();
 JOptionPane.showMessageDialog(this,"Successfully updated","User
info",JOptionPane.INFORMATION_MESSAGE);
 Update.setEnabled(false);
```

SELECT:

```
String sql="select NameOfUser as 'Name', UserName as 'User Name',Password,ContactNo as 'Contact
No',Email as 'Email ID' from Registration";
```

CRUD ON PATIENT REGISTRATION:

INSERT:

```
String sql = "insert into
PatientRegistration(PatientID,Patientname,FatherName,Email,ContactNo,Age,Remarks,Gen,BG,Address)va
lues('" + txtPatientID.getText() + "','" + txtPatientName.getText() + "','" + txtFathername.getText() + "','" +
txtEmailID.getText() + "','" + txtContactNo.getText() + "'," + txtAge.getText() + ",'" + txtRemarks.getText() +
"','" + cmbGender.getSelectedItem() + "','" + cmbBloodGroup.getSelectedItem() + "','" + txtAddress.getText()
+ "')";
```

DELETE:

```
String sql= "delete from PatientRegistration where PatientID = '" + txtPatientID.getText() + "'";
```

UPDATE:

```
String sql= "update PatientRegistration set Patientname='"+ txtPatientName.getText() + "',Fathername='"+
txtFathername.getText() + "',Email='"+ txtEmailID.getText() + "',ContactNo='"+ txtContactNo.getText() +
"',Age=" + txtAge.getText() + ",Remarks='"+ txtRemarks.getText() + "',Gen='" +
cmbGender.getSelectedItem() + "',BG='"+ cmbBloodGroup.getSelectedItem() + "',Address='" +
txtAddress.getText() + "' where PatientID='" + txtPatientID.getText() + "'";
```

SELECT:

```
String sql="select PatientID as 'Patient ID', PatientName as 'Patient Name',FatherName as 'Father
Name',Address,ContactNo as 'Contact No',Email as 'Email ID',Age,Gen as 'Gender',BG as 'Blood
Group',Remarks from Patientregistration";
```

CRUD ON PATIENT ADMISSION:

SELECT FROM MANY TABLES:

```
String sql="Select AdmitID as 'Admit ID',PatientRegistration.PatientID as 'Patient
ID',PatientRegistration.PatientName as 'Patient Name',PatientRegistration.Gen as
'Gender',PatientRegistration.BG as 'Blood Group',Disease,AdmitDate as 'Admit Date',Ward.Wardname as
'Ward Name',Doctor.DoctorID as 'Doctor ID',DoctorName as 'Doctor
Name',AdmitPatient_Ward.AP_Remarks as 'Remarks' from
Ward,Doctor,PatientRegistration,AdmitPatient_Ward where
Ward.Wardname=AdmitPatient_Ward.Wardname and Doctor.DoctorID=AdmitPatient_Ward.DoctorID
and PatientRegistration.PatientID=AdmitPatient_Ward.PatientID order by admitdate";
```

INSERTION WILL UPDATE ROOM (BOOKED):

```
    String sql = "insert into
AdmitPatient_Room(PatientID,Disease,AdmitDate,RoomNo,DoctorID,AP_Remarks)values('" +
PatientID.getText() + "','" + txtDisease.getText() + "','" + txtAdmitDate.getText() + "','" +
cmbRoomNo.getSelectedItem() + "','" + txtDoctorID.getText() + "','" + txtRemarks.getText() + "')";

    pst = con.prepareStatement(sql);
```

```
        String sql3 = "update room set RoomStatus='Booked' where RoomNo='" +
cmbRoomNo.getSelectedItem() + "'";
        pst = con.prepareStatement(sql3);
        pst.execute();
        JOptionPane.showMessageDialog(this, "Successfully admitted", "Patient",
JOptionPane.INFORMATION_MESSAGE);
        btnSave.setEnabled(false);
```

DELETE:

```
String sql= "delete from AdmitPatient_Room where AdmitID = " + txtAdmitID.getText() + "";
```

CRUD ON PATIENT DISCHARGE:

INSERTION WILL UPDATE ROOM (VACANT):

```
String sql= "insert into DischargePatient_Room(AdmitID,DischargeDate,DP_Remarks)values("+
txtAdmitID.getText() + ",'"+ txtDischargeDate.getText() + "','"+ txtRemarks.getText() + "')";

String sql3= "update room set RoomStatus='Vacant' where RoomNo='" + txtRoomNo.getText() + "'";
```

DELETE:

```
String sql= "delete from DischargePatient_Room where ID = " + txtDischargeID.getText() + "";
```

SELECT:

```
String sql="Select ID as 'Discharge ID', AdmitPatient_Room.AdmitID as 'Admit
ID',PatientRegistration.PatientID as 'Patient ID',PatientRegistration.PatientName as 'Patient
Name',PatientRegistration.Gen as 'Gender',PatientRegistration.BG as 'Blood Group',Disease,AdmitDate as
'Admit Date',Room.RoomNo as 'Room No',Doctor.DoctorID as 'Doctor ID',DoctorName as 'Doctor
Name',DischargeDate as 'Discharge Date',DP_Remarks as 'Remarks' from
Room,Doctor,PatientRegistration,AdmitPatient_Room,DischargePatient_Room where
Room.RoomNo=AdmitPatient_Room.RoomNo and Doctor.DoctorID=AdmitPatient_Room.DoctorID and
PatientRegistration.PatientID=AdmitPatient_Room.PatientID  and AdmitPatient_Room.admitID=
DischargePatient_Room.admitID order by Dischargedate";
```

BILLING :[ALIAS AND RETREIVING FROM DIFFERENT TABLE]

```
String sql="Select BillNo as 'Bill No.',DisChargePatient_Room.ID as 'Discharge ID',
AdmitPatient_Room.AdmitID as 'Admit ID',PatientRegistration.PatientID as 'Patient
ID',PatientRegistration.PatientName as 'Patient Name',PatientRegistration.Gen as
'Gender',PatientRegistration.BG as 'Blood Group',Disease,AdmitDate as 'Admit Date',Room.RoomNo as
'Room No',Doctor.DoctorID as 'Doctor ID',DoctorName as 'Doctor Name',DischargeDate as 'Discharge
Date',Bill_Room.RoomCharges as 'Room Charges',Bill_Room.ServiceCharges as 'Service
Charges',Bill_Room.BillingDate as 'Billing Date',PaymentMode as 'Payement Mode',PaymentModeDetails as
'Payment Mode Details',TotalCharges as 'Total Charges',ChargesPaid as 'Charges Paid',DueCharges as 'Due
Charges',NoOfDays as 'No. Of Days',TotalRoomCharges as 'Total Room Charges' from
Room,Doctor,PatientRegistration,AdmitPatient_Room,DischargePatient_Room,Bill_Room where
Room.RoomNo=AdmitPatient_Room.RoomNo and Doctor.DoctorID=AdmitPatient_Room.DoctorID and
PatientRegistration.PatientID=AdmitPatient_Room.PatientID  and AdmitPatient_Room.admitID=
DischargePatient_Room.admitID and Bill_Room.DischargeID=DischargePatient_Room.ID  order by
Billingdate";
```

GROUP BY, AGGREGATE, ORDER BY, JOIN:

```
    private void Get_Data1(){
      try{
        con=Connect.ConnectDB();
       String sql="select PatientRegistration.PatientID as 'Patient ID', PatientName as 'Patient
Name',sum(serviceCharges) as 'Service Charges' from Services join PatientRegistration where
Services.PatientID=PatientRegistration.PatientID group by PatientRegistration.PatientID,PatientName order
by PatientName";
         pst=con.prepareStatement(sql);
         rs= pst.executeQuery();
         jTable1.setModel(DbUtils.resultSetToTableModel(rs));
         }catch(Exception e){
            JOptionPane.showMessageDialog(null, e);}
```

INDEX:

```
try {
    String sql = "create index rno on Room(RoomNo)";
    pst = con.prepareStatement(sql);
    pst.execute();
} catch (Exception ex) {
    String sql = "insert into Room(RoomNo,RoomType,RoomCharges,RoomStatus)values('" +
txtRoomNo.getText() + "','" + cmbRoomType.getSelectedItem() + "'," + txtRoomCharges.getText() +
",'Vacant')";
    pst = con.prepareStatement(sql);
    pst.execute();

    JOptionPane.showMessageDialog(this, "Successfully saved", "Room Record",
JOptionPane.INFORMATION_MESSAGE);
    btnSave.setEnabled(false);
    Get_Data();}
```

VIEW:

```
con = Connect.ConnectDB();
String sql2 = "create view doc as select * from Doctor";
try {
    pst = con.prepareStatement(sql2);
    pst.execute();
} catch (Exception e) {
Statement stmt;
stmt= con.createStatement();
String sql1="Select DoctorID from doc where DoctorID= '" + txtDoctorID.getText() + "'";
rs=stmt.executeQuery(sql1);
if(rs.next()) {
    JOptionPane.showMessageDialog(this, "Doctor ID already exists", "Error",
JOptionPane.ERROR_MESSAGE);
    txtDoctorID.setText("");
    txtDoctorID.requestDefaultFocus();
    return;}
```

VIEW INSERT:

```
        String sql= "insert into
Doc(DoctorID,Doctorname,FatherName,Email,ContactNo,Qualifications,Specialization,Gender,BloodGroup,
DateOfJoining,Address)values('"+ txtDoctorID.getText() + "','"+ txtDoctorName.getText() + "','"+
txtFathername.getText() + "','"+ txtEmailID.getText() + "','"+ txtContactNo.getText() + "','"+
txtQualifications.getText() + "','"+ txtSpecialisation.getText() + "','" + cmbGender.getSelectedItem() + "','"+
cmbBloodGroup.getSelectedItem() + "','" + txtDateOfJoining.getText() + "','" + txtAddress.getText() + "')";

        pst=con.prepareStatement(sql);
        pst.execute();
        JOptionPane.showMessageDialog(this,"Successfully saved","Doctor
Record",JOptionPane.INFORMATION_MESSAGE);
        btnSave.setEnabled(false);

    }catch(HeadlessException | SQLException ex){
        JOptionPane.showMessageDialog(this,ex);}}}
```

VIEW UPDATE:

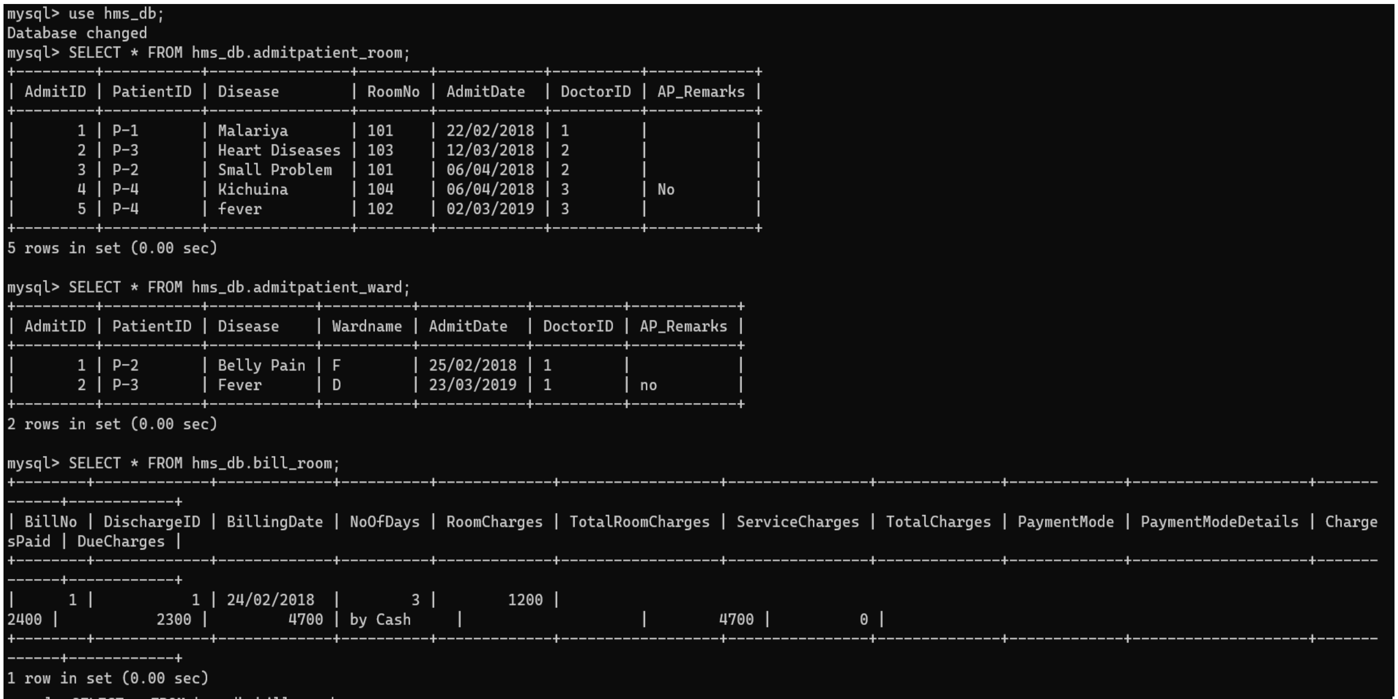
```
private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_btnUpdateActionPerformed
    try{
        con=Connect.ConnectDB();
        String sql= "update doc set Doctorname='"+ txtDoctorName.getText() + "',FatherName='"+
txtFathername.getText() + "',Email='"+ txtEmailID.getText() + "',ContactNo='"+ txtContactNo.getText() +
"',Qualifications='"+ txtQualifications.getText() + "',Specialization='"+ txtSpecialisation.getText() +
"',Gender='" + cmbGender.getSelectedItem() + "',BloodGroup='"+ cmbBloodGroup.getSelectedItem() +
"',DateOfJoining='" + txtDateOfJoining.getText() + "',Address='" + txtAddress.getText() + "' where
DoctorID='" + txtDoctorID.getText() + "'";

        pst=con.prepareStatement(sql);
        pst.execute();
        JOptionPane.showMessageDialog(this,"Successfully updated","Doctor
Record",JOptionPane.INFORMATION_MESSAGE);
        btnUpdate.setEnabled(false);

}}
```

VIEW DELETE:

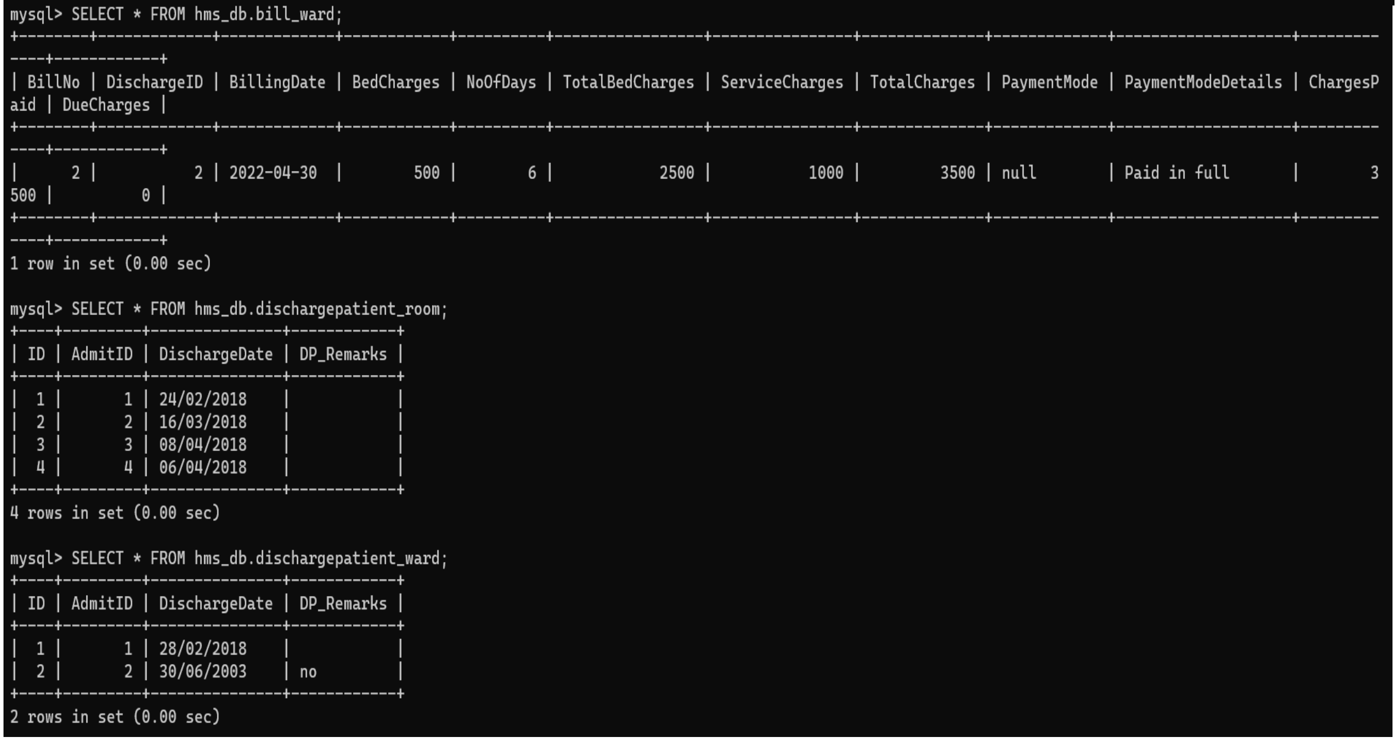
```
private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
        int P = JOptionPane.showConfirmDialog(null," Are you sure want to delete
?","Confirmation",JOptionPane.YES_NO_OPTION);
        if (P==0) {
                String sql = "delete from doc where DoctorID = '" + txtDoctorID.getText() + "'";
                pst = con.prepareStatement(sql);
                pst.execute();
                JOptionPane.showMessageDialog(this, "Successfully deleted", "Record",
JOptionPane.INFORMATION_MESSAGE);

                Reset();
        }}catch(HeadlessException | SQLException ex){ JOptionPane.showMessageDialog(this,ex);
    }
}
```

➢ Screenshot of Output:

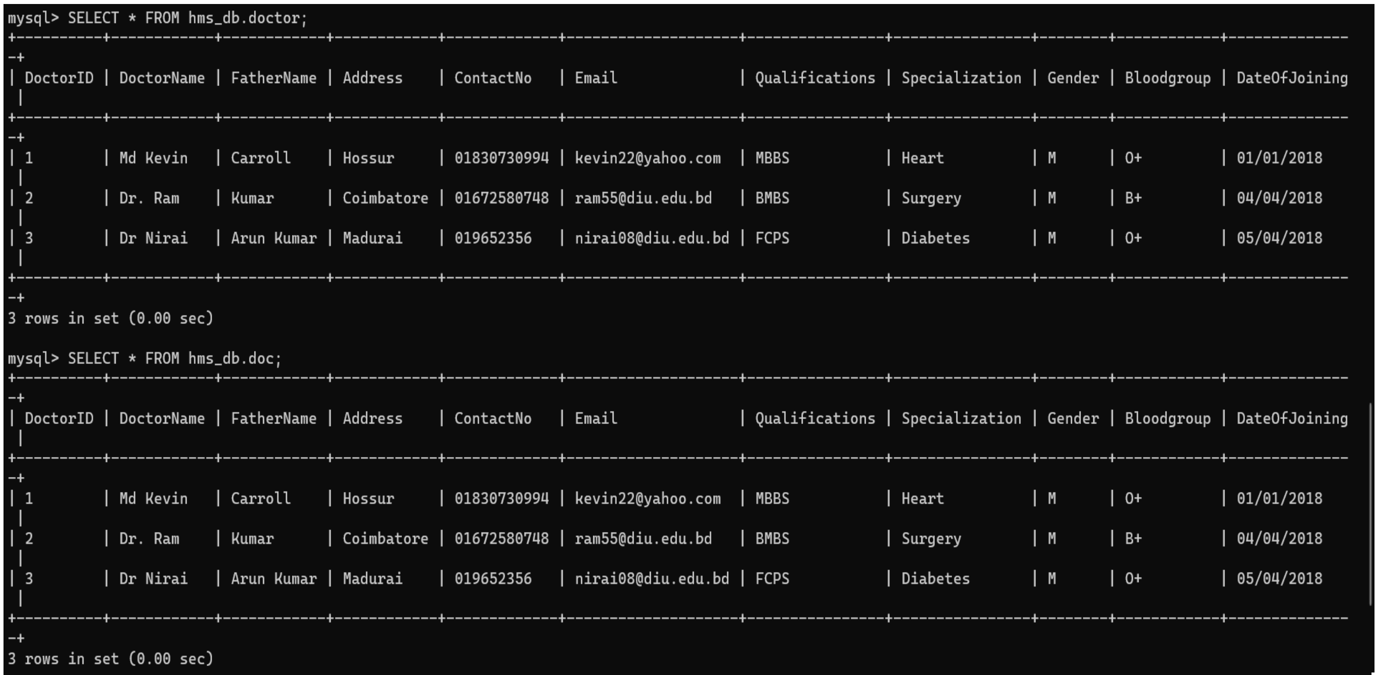
RECORDS IN TABLE:



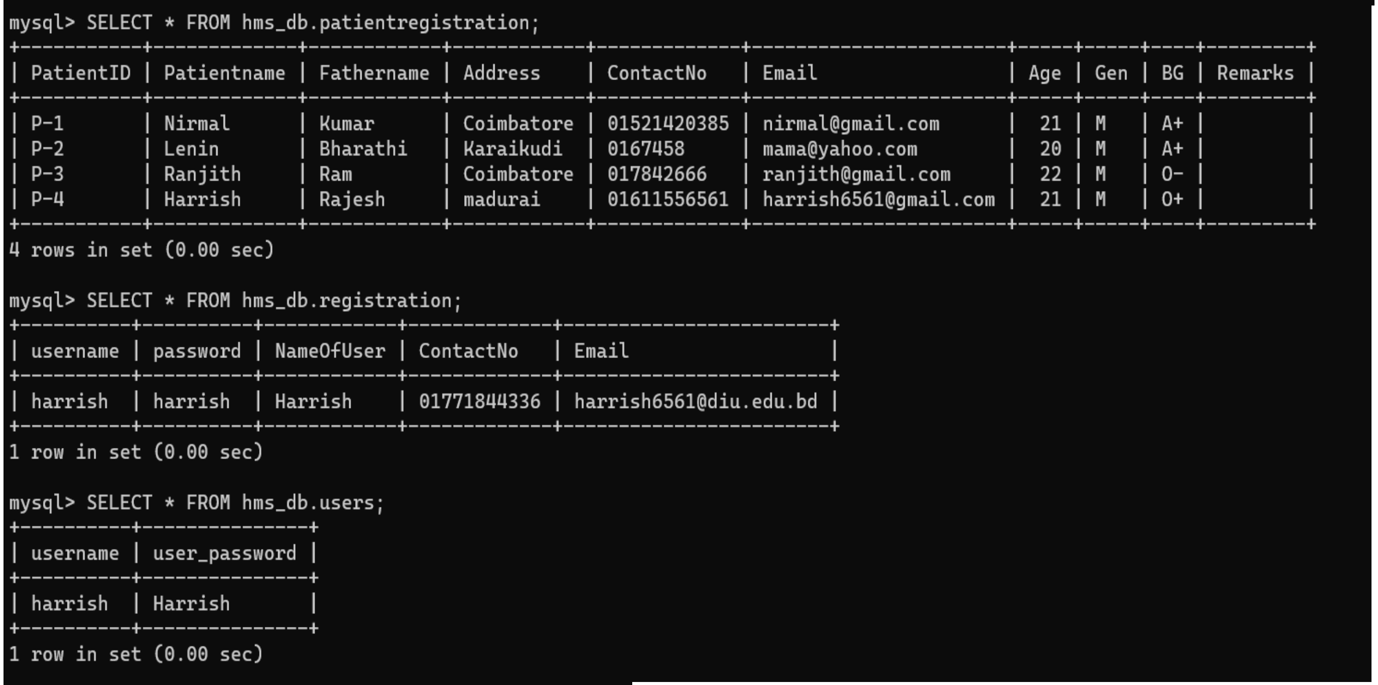
```
mysql> use hms_db;
Database changed
mysql> SELECT * FROM hms_db.admitpatient_room;
+---------+-----------+----------------+--------+------------+----------+------------+
| AdmitID | PatientID | Disease        | RoomNo | AdmitDate  | DoctorID | AP_Remarks |
+---------+-----------+----------------+--------+------------+----------+------------+
|       1 | P-1       | Malariya       | 101    | 22/02/2018 | 1        |            |
|       2 | P-3       | Heart Diseases | 103    | 12/03/2018 | 2        |            |
|       3 | P-2       | Small Problem  | 101    | 06/04/2018 | 2        |            |
|       4 | P-4       | Kichuina       | 104    | 06/04/2018 | 3        | No         |
|       5 | P-4       | fever          | 102    | 02/03/2019 | 3        |            |
+---------+-----------+----------------+--------+------------+----------+------------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM hms_db.admitpatient_ward;
+---------+-----------+------------+----------+------------+----------+------------+
| AdmitID | PatientID | Disease    | Wardname | AdmitDate  | DoctorID | AP_Remarks |
+---------+-----------+------------+----------+------------+----------+------------+
|       1 | P-2       | Belly Pain | F        | 25/02/2018 | 1        |            |
|       2 | P-3       | Fever      | D        | 23/03/2019 | 1        | no         |
+---------+-----------+------------+----------+------------+----------+------------+
2 rows in set (0.00 sec)

mysql> SELECT * FROM hms_db.bill_room;
+--------+-------------+-------------+----------+-------------+------------------+----------------+--------------+-------------+--------------------+-------
------+------------+
| BillNo | DischargeID | BillingDate | NoOfDays | RoomCharges | TotalRoomCharges | ServiceCharges | TotalCharges | PaymentMode | PaymentModeDetails | Charge
sPaid | DueCharges |
+--------+-------------+-------------+----------+-------------+------------------+----------------+--------------+-------------+--------------------+-------
------+------------+
|      1 |           1 | 24/02/2018  |        3 |        1200 |
2400 |          2300 |         4700 | by Cash     |                    |        4700 |          0 |
+--------+-------------+-------------+----------+-------------+------------------+----------------+--------------+-------------+--------------------+-------
------+------------+
1 row in set (0.00 sec)
```



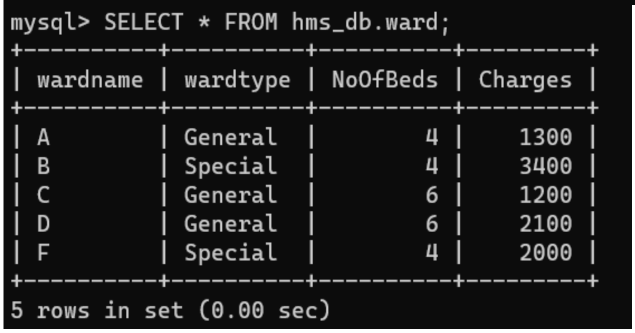
```
mysql> SELECT * FROM hms_db.bill_ward;
+--------+-------------+-------------+------------+----------+-----------------+----------------+--------------+-------------+--------------------+---------
----+------------+
| BillNo | DischargeID | BillingDate | BedCharges | NoOfDays | TotalBedCharges | ServiceCharges | TotalCharges | PaymentMode | PaymentModeDetails | ChargesP
aid | DueCharges |
+--------+-------------+-------------+------------+----------+-----------------+----------------+--------------+-------------+--------------------+---------
----+------------+
|      2 |           2 | 2022-04-30  |        500 |        6 |            2500 |           1000 |         3500 | null        | Paid in full       |        3
500 |          0 |
+--------+-------------+-------------+------------+----------+-----------------+----------------+--------------+-------------+--------------------+---------
----+------------+
1 row in set (0.00 sec)

mysql> SELECT * FROM hms_db.dischargepatient_room;
+----+---------+---------------+------------+
| ID | AdmitID | DischargeDate | DP_Remarks |
+----+---------+---------------+------------+
|  1 |       1 | 24/02/2018    |            |
|  2 |       2 | 16/03/2018    |            |
|  3 |       3 | 08/04/2018    |            |
|  4 |       4 | 06/04/2018    |            |
+----+---------+---------------+------------+
4 rows in set (0.00 sec)

mysql> SELECT * FROM hms_db.dischargepatient_ward;
+----+---------+---------------+------------+
| ID | AdmitID | DischargeDate | DP_Remarks |
+----+---------+---------------+------------+
|  1 |       1 | 28/02/2018    |            |
|  2 |       2 | 30/06/2003    | no         |
+----+---------+---------------+------------+
2 rows in set (0.00 sec)
```



```
mysql> SELECT * FROM hms_db.doctor;
+----------+------------+------------+------------+-------------+--------------------+----------------+----------------+--------+------------+--------------
-+
| DoctorID | DoctorName | FatherName | Address    | ContactNo   | Email              | Qualifications | Specialization | Gender | Bloodgroup | DateOfJoining
 |
+----------+------------+------------+------------+-------------+--------------------+----------------+----------------+--------+------------+--------------
-+
| 1        | Md Kevin   | Carroll    | Hossur     | 01830730994 | kevin22@yahoo.com  | MBBS           | Heart          | M      | O+         | 01/01/2018
 |
| 2        | Dr. Ram    | Kumar      | Coimbatore | 01672580748 | ram55@diu.edu.bd   | BMBS           | Surgery        | M      | B+         | 04/04/2018
 |
| 3        | Dr Nirai   | Arun Kumar | Madurai    | 019652356   | nirai08@diu.edu.bd | FCPS           | Diabetes       | M      | O+         | 05/04/2018
 |
+----------+------------+------------+------------+-------------+--------------------+----------------+----------------+--------+------------+--------------
-+
3 rows in set (0.00 sec)

mysql> SELECT * FROM hms_db.doc;
+----------+------------+------------+------------+-------------+--------------------+----------------+----------------+--------+------------+--------------
-+
| DoctorID | DoctorName | FatherName | Address    | ContactNo   | Email              | Qualifications | Specialization | Gender | Bloodgroup | DateOfJoining
 |
+----------+------------+------------+------------+-------------+--------------------+----------------+----------------+--------+------------+--------------
-+
| 1        | Md Kevin   | Carroll    | Hossur     | 01830730994 | kevin22@yahoo.com  | MBBS           | Heart          | M      | O+         | 01/01/2018
 |
| 2        | Dr. Ram    | Kumar      | Coimbatore | 01672580748 | ram55@diu.edu.bd   | BMBS           | Surgery        | M      | B+         | 04/04/2018
 |
| 3        | Dr Nirai   | Arun Kumar | Madurai    | 019652356   | nirai08@diu.edu.bd | FCPS           | Diabetes       | M      | O+         | 05/04/2018
 |
+----------+------------+------------+------------+-------------+--------------------+----------------+----------------+--------+------------+--------------
-+
3 rows in set (0.00 sec)
```



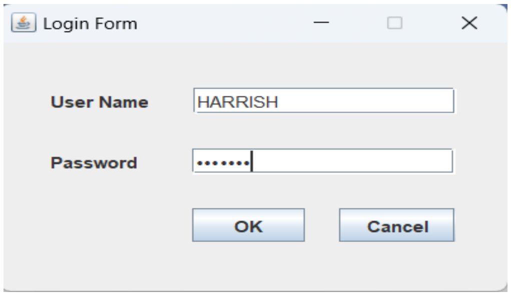
```
mysql> SELECT * FROM hms_db.patientregistration;
+-----------+-------------+------------+------------+-------------+------------------------+-----+-----+----+---------+
| PatientID | Patientname | Fathername | Address    | ContactNo   | Email                  | Age | Gen | BG | Remarks |
+-----------+-------------+------------+------------+-------------+------------------------+-----+-----+----+---------+
| P-1       | Nirmal      | Kumar      | Coimbatore | 01521420385 | nirmal@gmail.com       |  21 | M   | A+ |         |
| P-2       | Lenin       | Bharathi   | Karaikudi  | 0167458     | mama@yahoo.com         |  20 | M   | A+ |         |
| P-3       | Ranjith     | Ram        | Coimbatore | 017842666   | ranjith@gmail.com      |  22 | M   | O- |         |
| P-4       | Harrish     | Rajesh     | madurai    | 01611556561 | harrish6561@gmail.com  |  21 | M   | O+ |         |
+-----------+-------------+------------+------------+-------------+------------------------+-----+-----+----+---------+
4 rows in set (0.00 sec)

mysql> SELECT * FROM hms_db.registration;
+----------+----------+------------+-------------+-----------------------+
| username | password | NameOfUser | ContactNo   | Email                 |
+----------+----------+------------+-------------+-----------------------+
| harrish  | harrish  | Harrish    | 01771844336 | harrish6561@diu.edu.bd |
+----------+----------+------------+-------------+-----------------------+
1 row in set (0.00 sec)

mysql> SELECT * FROM hms_db.users;
+----------+---------------+
| username | user_password |
+----------+---------------+
| harrish  | Harrish       |
+----------+---------------+
1 row in set (0.00 sec)
```



```
mysql> SELECT * FROM hms_db.ward;
+----------+----------+----------+---------+
| wardname | wardtype | NoOfBeds | Charges |
+----------+----------+----------+---------+
| A        | General  |        4 |    1300 |
| B        | Special  |        4 |    3400 |
| C        | General  |        6 |    1200 |
| D        | General  |        6 |    2100 |
| F        | Special  |        4 |    2000 |
+----------+----------+----------+---------+
5 rows in set (0.00 sec)
```

APPLICATION INTERFACES:

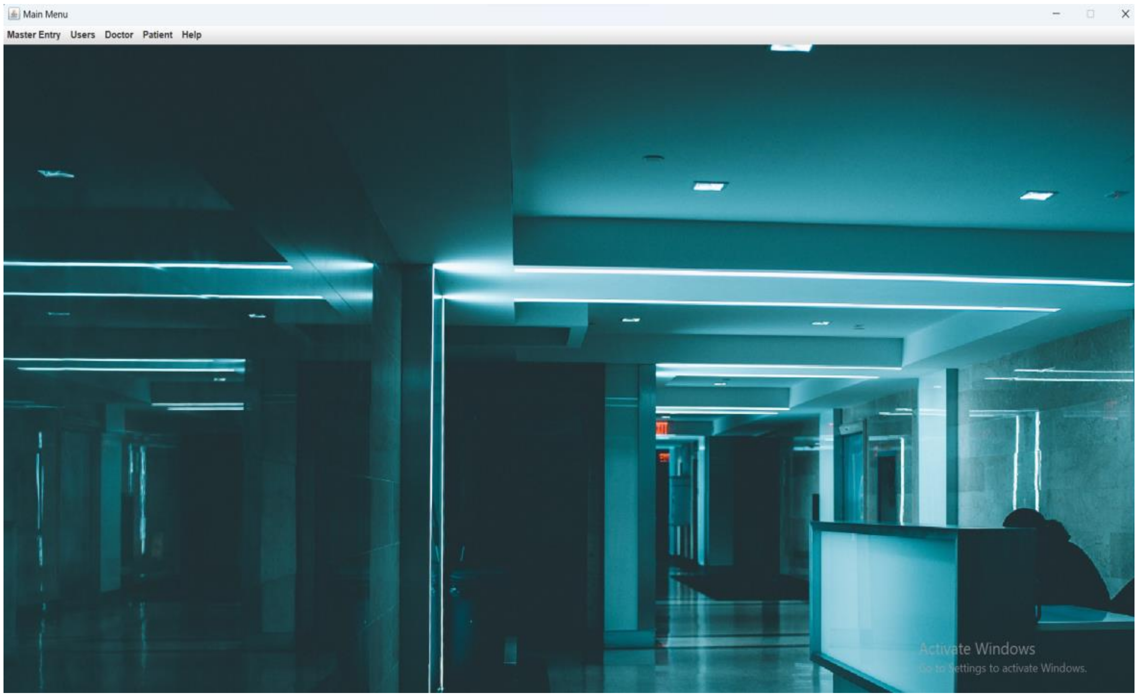
LOGIN:



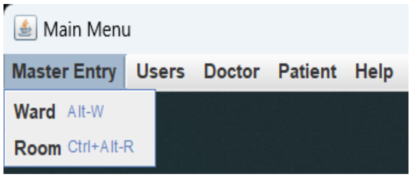
User Name HARRISH

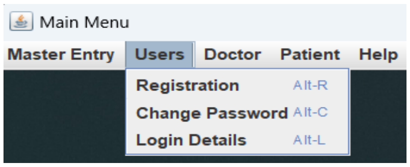
Password •••••••

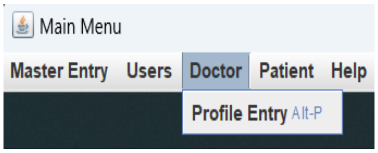
MAIN MENU:

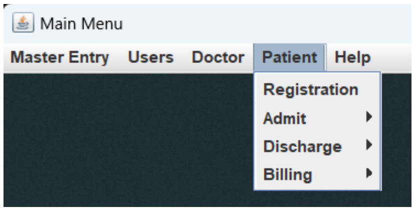


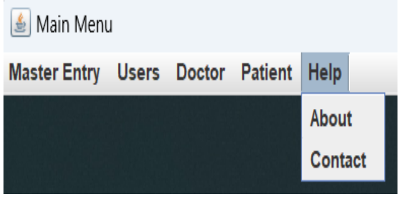
MENU ITMES:



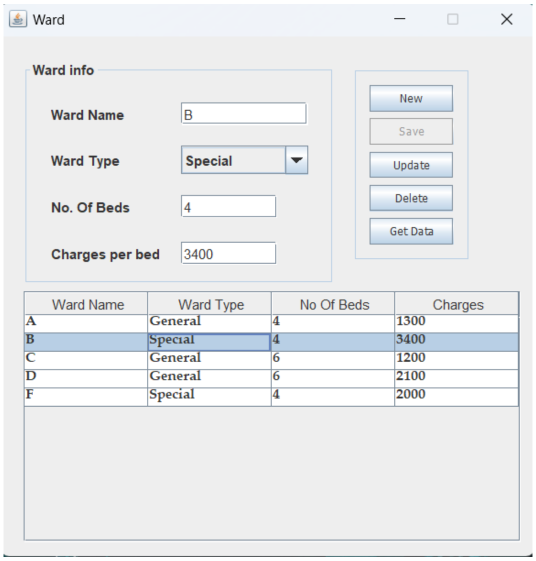








CRUD ON WARD AND ROOM:



Ward info

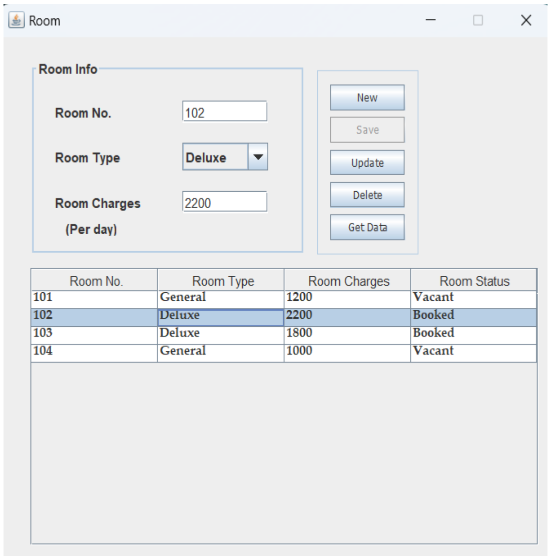
Ward Name: B

Ward Type: Special

No. Of Beds: 4

Charges per bed: 3400

Ward Name	Ward Type	No Of Beds	Charges
A	General	4	1300
B	Special	4	3400
C	General	6	1200
D	General	6	2100
F	Special	4	2000



Room Info

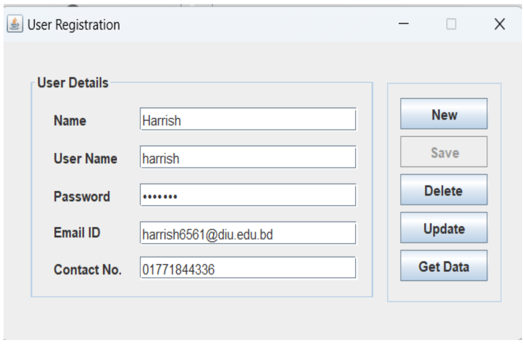
Room No.: 102

Room Type: Deluxe

Room Charges (Per day): 2200

Room No.	Room Type	Room Charges	Room Status
101	General	1200	Vacant
102	Deluxe	2200	Booked
103	Deluxe	1800	Booked
104	General	1000	Vacant

CRUD ON USER AND LOGIN CREATION:



User Details

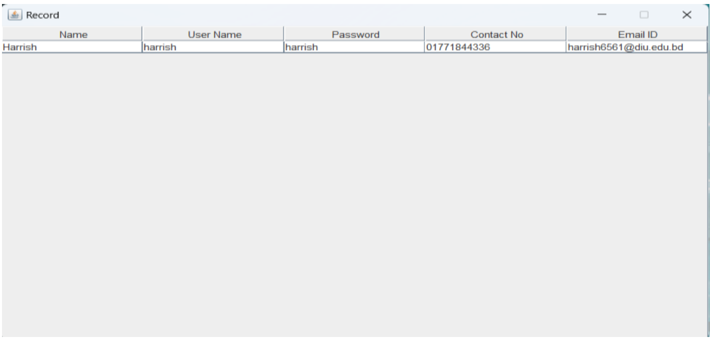
Name: Harrish

User Name: harrish

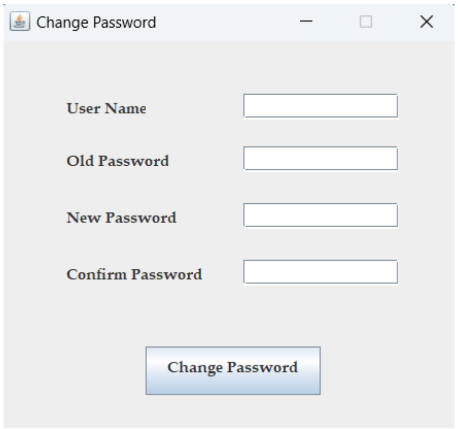
Password: •••••••

Email ID: harrish6561@diu.edu.bd

Contact No.: 01771844336



Name	User Name	Password	Contact No	Email ID
Harrish	harrish	harrish	01771844336	harrish6561@diu.edu.bd

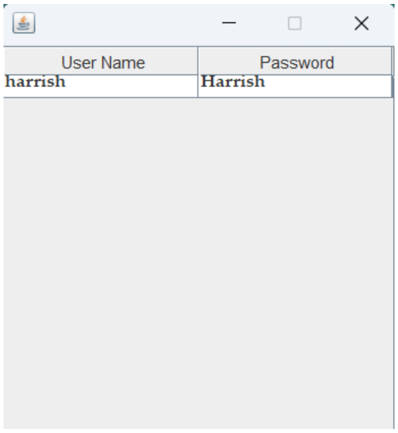


User Name:

Old Password:

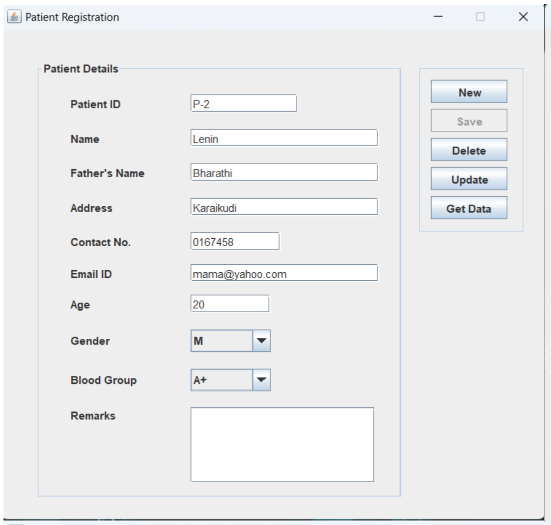
New Password:

Confirm Password:



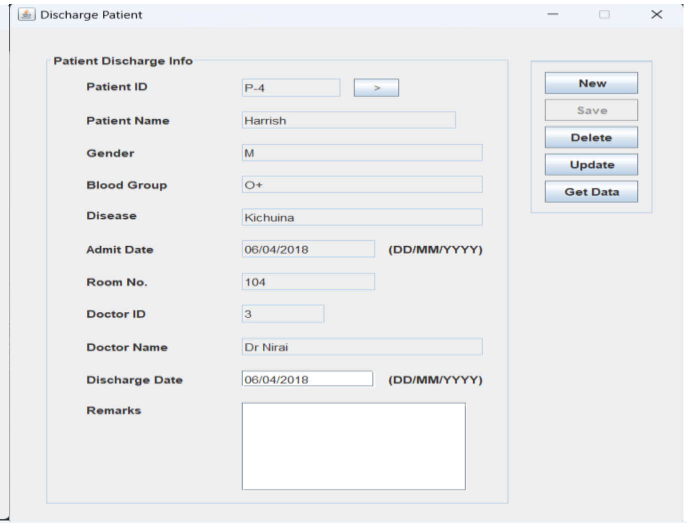
User Name	Password
harrish	Harrish

CRUD ON PATIENT REGISTRATION,ADMINSSION,DISCHARGE:



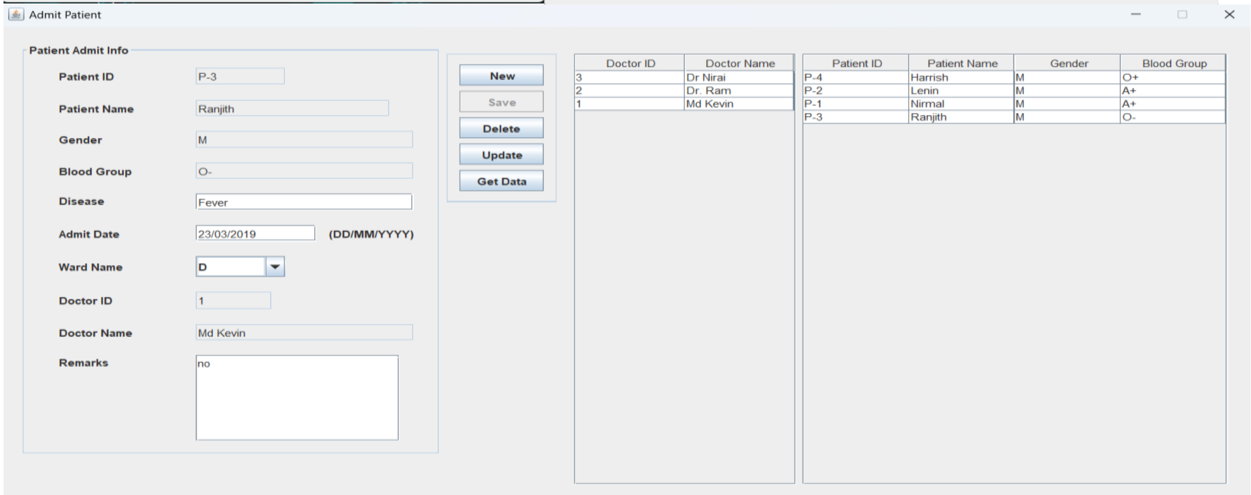
Patient Details

Patient ID	P-2
Name	Lenin
Father's Name	Bharathi
Address	Karaikudi
Contact No.	0167458
Email ID	mama@yahoo.com
Age	20
Gender	M
Blood Group	A+
Remarks	



Patient Discharge Info

Patient ID	P-4	
Patient Name	Harrish	
Gender	M	
Blood Group	O+	
Disease	Kichuina	
Admit Date	06/04/2018	(DD/MM/YYYY)
Room No.	104	
Doctor ID	3	
Doctor Name	Dr Nirai	
Discharge Date	06/04/2018	(DD/MM/YYYY)
Remarks		

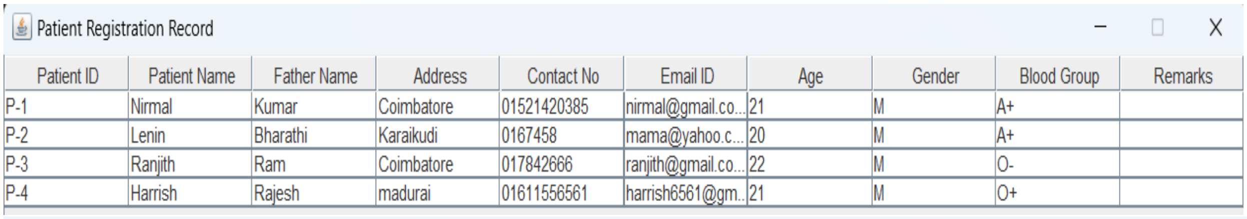


Patient Admit Info

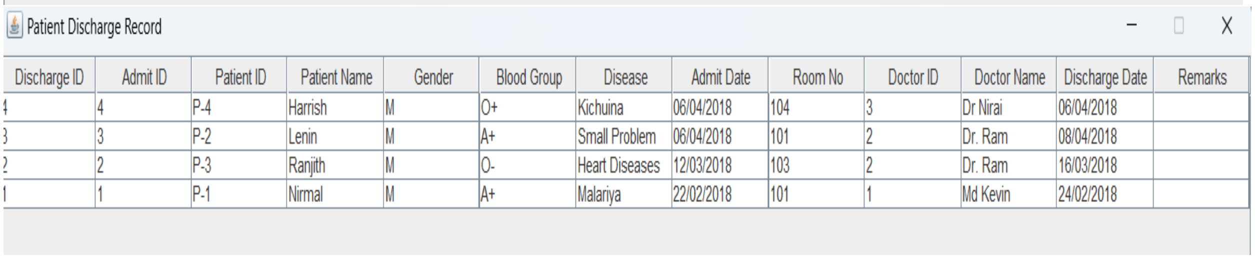
Patient ID	P-3	
Patient Name	Ranjith	
Gender	M	
Blood Group	O-	
Disease	Fever	
Admit Date	23/03/2019	(DD/MM/YYYY)
Ward Name	D	
Doctor ID	1	
Doctor Name	Md Kevin	
Remarks	no	

Doctor ID	Doctor Name
3	Dr Nirai
2	Dr. Ram
1	Md Kevin

Patient ID	Patient Name	Gender	Blood Group
P-4	Harrish	M	O+
P-2	Lenin	M	A+
P-1	Nirmal	M	A+
P-3	Ranjith	M	O-



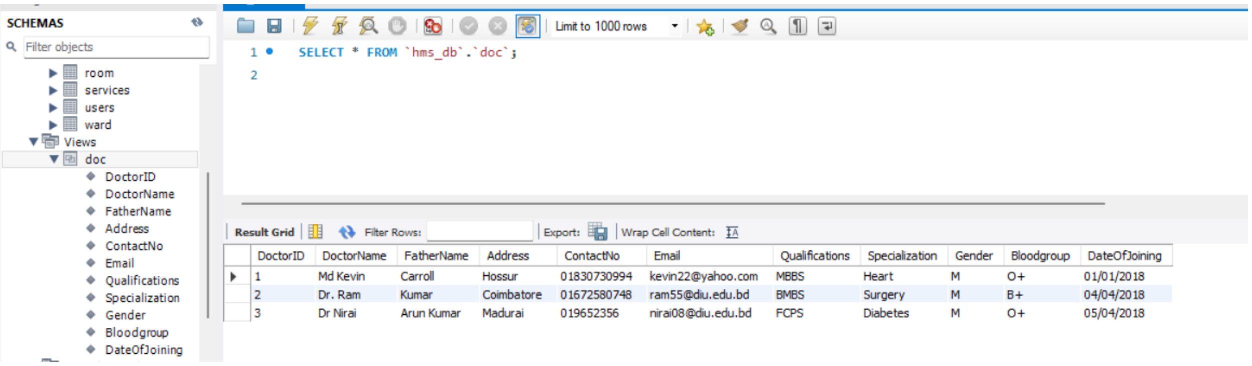
Patient ID	Patient Name	Father Name	Address	Contact No	Email ID	Age	Gender	Blood Group	Remarks
P-1	Nirmal	Kumar	Coimbatore	01521420385	nirmal@gmail.co...	21	M	A+	
P-2	Lenin	Bharathi	Karaikudi	0167458	mama@yahoo.c...	20	M	A+	
P-3	Ranjith	Ram	Coimbatore	017842666	ranjith@gmail.co...	22	M	O-	
P-4	Harrish	Rajesh	madurai	01611556561	harrish6561@gm...	21	M	O+	



Discharge ID	Admit ID	Patient ID	Patient Name	Gender	Blood Group	Disease	Admit Date	Room No	Doctor ID	Doctor Name	Discharge Date	Remarks
4	4	P-4	Harrish	M	O+	Kichuina	06/04/2018	104	3	Dr Nirai	06/04/2018	
3	3	P-2	Lenin	M	A+	Small Problem	06/04/2018	101	2	Dr. Ram	08/04/2018	
2	2	P-3	Ranjith	M	O-	Heart Diseases	12/03/2018	103	2	Dr. Ram	16/03/2018	
1	1	P-1	Nirmal	M	A+	Malariya	22/02/2018	101	1	Md Kevin	24/02/2018	

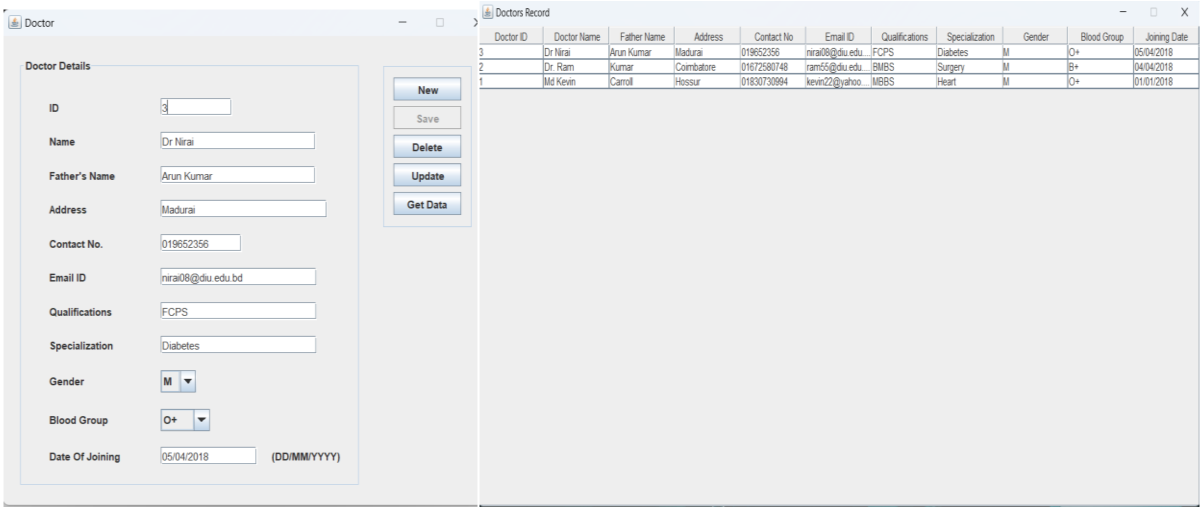
➢ Screenshot of the output to demonstrate any of the SQL operation [Joins, Aggregations, Index, View, Order by, Alias]:

CRUD ON VIEW:



```
SELECT * FROM `hms_db`.`doc`;
```

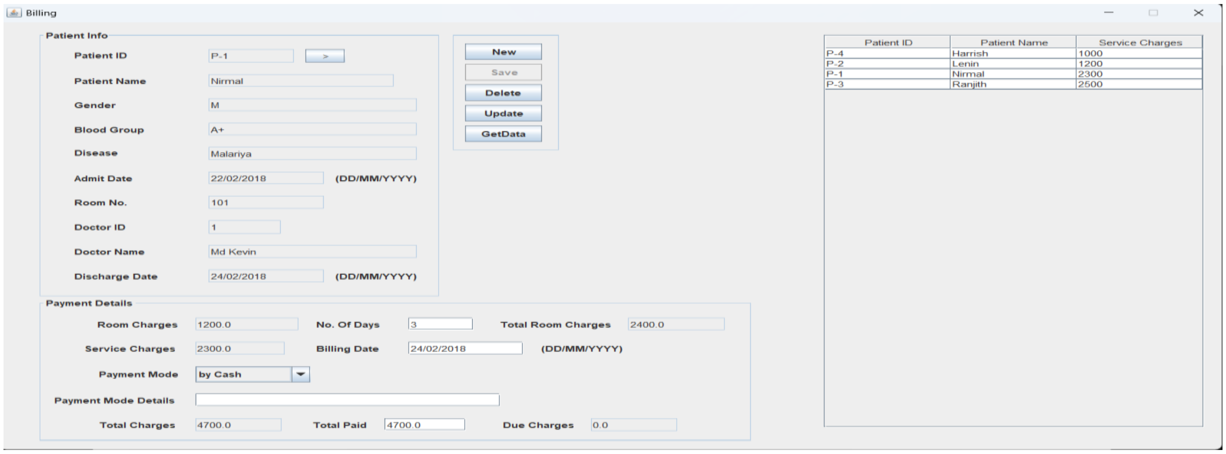
DoctorID	DoctorName	FatherName	Address	ContactNo	Email	Qualifications	Specialization	Gender	Bloodgroup	DateOfJoining
1	Md Kevin	Carroll	Hossur	01830730994	kevin22@yahoo.com	MBBS	Heart	M	O+	01/01/2018
2	Dr. Ram	Kumar	Coimbatore	01672580748	ram55@diu.edu.bd	BMBS	Surgery	M	B+	04/04/2018
3	Dr Nirai	Arun Kumar	Madurai	019652356	nirai08@diu.edu.bd	FCPS	Diabetes	M	O+	05/04/2018



ID	3
Name	Dr Nirai
Father's Name	Arun Kumar
Address	Madurai
Contact No.	019652356
Email ID	nirai08@diu.edu.bd
Qualifications	FCPS
Specialization	Diabetes
Gender	M
Blood Group	O+
Date Of Joining	05/04/2018 (DD/MM/YYYY)

Doctor ID	Doctor Name	Father Name	Address	Contact No	Email ID	Qualifications	Specialization	Gender	Blood Group	Joining Date
3	Dr Nirai	Arun Kumar	Madurai	019652356	nirai08@diu.edu....	FCPS	Diabetes	M	O+	05/04/2018
2	Dr. Ram	Kumar	Coimbatore	01672580748	ram55@diu.edu....	BMBS	Surgery	M	B+	04/04/2018
1	Md Kevin	Carroll	Hossur	01830730994	kevin22@yahoo....	MBBS	Heart	M	O+	01/01/2018

GROUP BY, AGGREGATE, JOIN, ORDER BY:



Patient Info

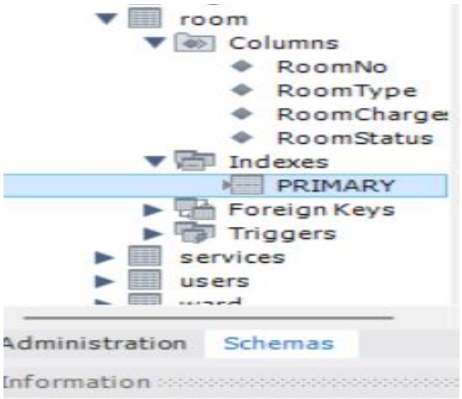
Patient ID	P-1
Patient Name	Nirmal
Gender	M
Blood Group	A+
Disease	Malariya
Admit Date	22/02/2018 (DD/MM/YYYY)
Room No.	101
Doctor ID	1
Doctor Name	Md Kevin
Discharge Date	24/02/2018 (DD/MM/YYYY)

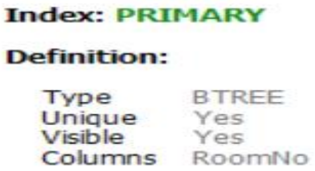
Payment Details

Room Charges	1200.0	No. Of Days	3	Total Room Charges	2400.0
Service Charges	2300.0	Billing Date	24/02/2018 (DD/MM/YYYY)		
Payment Mode	by Cash				
Payment Mode Details					
Total Charges	4700.0	Total Paid	4700.0	Due Charges	0.0

Patient ID	Patient Name	Service Charges
P-4	Harrish	1000
P-2	Lenin	1200
P-1	Nirmal	2300
P-3	Ranjith	2500

INDEX:



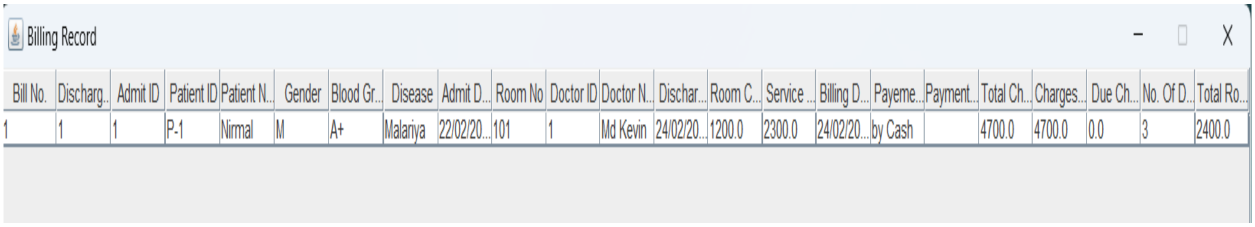


Index: PRIMARY

Definition:

Type	BTREE
Unique	Yes
Visible	Yes
Columns	RoomNo

BIILLING THE TOTAL FEES(ALIAS):



Bill No.	Discharg...	Admit ID	Patient ID	Patient N...	Gender	Blood Gr...	Disease	Admit D...	Room No	Doctor ID	Doctor N...	Dischar...	Room C...	Service ...	Billing D...	Payeme...	Payment...	Total Ch...	Charges...	Due Ch...	No. Of D...	Total Ro...
1	1	1	P-1	Nirmal	M	A+	Malariya	22/02/20...	101	1	Md Kevin	24/02/20...	1200.0	2300.0	24/02/20...	by Cash		4700.0	4700.0	0.0	3	2400.0

➢ References

https://datamateindia.com/a-complete-overview-of-hospital-management-system/#:~:text=A%20Hospital%20Management%20System%20(HMS)%20is%20a%20software%20designed%20to,provide%20better%20care%20and%20services.

https://www.javaguides.net/2019/07/java-swing-application-with-database connection.html

https://docs.oracle.com/javase/tutorial/uiswing/

https://www.javatpoint.com/mysql-tutorial

https://www.javaguides.net/2019/07/java-swing-jdbc-mysql-example.html