**INDEX**

* Introduction to
* Front-end and back-end.
* Java net Beans.
* MySql.
* Snapshots of Java form.
* Coding of Java Net-Beans.
* Entry
* Discharge
* Report
* Coding of MySQL
* Creating database
* Value Insertion
* Conclusions

Bibliography and References.

**Java Front-End overview**

The Front End Application controls between end-users, the Translator and possibly other applications (for e.g. ,a form repository , workflow application, and so on) you can also use the Front End Application to populate a form with data before presenting the form to a user, and to extract data from a form after the user submits it.

1. For software applications, front end is the same as user interface.
2. In client/server applications, the client part of the program is often called the front end and the server part is called the back end.

User interface design is the design of software applications, and websites with the focus on the user’s experience and interaction as simple and efficient as possible, in terms of accomplishing user goals- what is often called user-centered design(e.g, mental model) to create a system that is not only operational but also usable and adaptable to changing user needs

A good user interface needs to have following features:

1>**CLEAR :** Clarity is the most important element of user interface design. Indeed, the whole purpose of user interface design is to enable people to interact with your system by communicating meaning and function. If people can’t figure out how your application works or where to go on your website they will get confused and frustrated.

2>**CONCISE:** Clarity in a user interface is to keep things clear but also keep things concise. When you can explain a feature in a sentence instead of three, do it. When you can label an item with one word instead of two, do it. Save the valuable time of users by keeping things concise.

3>**FAMILIAR:** Familiar is just that : Something which appears like something else you have encountered before. When you are familiar with something, you know how it behaves\_ you know what to expect. Identify things that are familiar to your users and integrate that into your user interface.

4>**RESPONSIVE:** Responsive means a couple of things. First of all , responsive means fast. The interface , if not the software behind it, should work fast. Waiting for things to load and using laggy and slow interface is frustrating. Seeing things load quickly, or at the very least an interface that loads quickly (even if the content is yet to catch up) improves the user experience . Responsive also means the interface provides some form of feedback.

**5>CONSISTENT:** Consistent interfaces allow users to develop usage patterns-they will learn what the different buttons, tabs, icons and other interface elements look like and will recognize them and realize what they do min different contexts. They will also learn how certain things work, will be able to work out how to operate new features quicker, extrapolating from those extra experiences.

**6>ATTRACTIVE:** A good interface should be attractive. Attractive in a sense that it makes the use of that interface enjoyable when your software is pleasant to use, your costumer or staff will not simply be using it they will look forward to using it. There are of course many different types of software and websites, all produced for different markets and audiences. What looks ‘good’ for anyone particular audience will vary. This means that you should fashion the look and feel if your interface for your audience.

**7>EFFICIENT:** A user interface is vehicle that takes you places. Those places are the different functions of the software application or websites. A good interface should allow you to perform those functions faster and less effort. Now, ‘efficient’ sounds like a fairly vague attribute-if you combine all of the other things on this list, surely the interface will end up being efficient. what you really need to do make an interface efficient is to figure out what exactly the user is trying to achieve, and then let them do exactly that without any fuss.

**BACK END Overview**

A back-end is a database that is accessed by users indirectly through an external application rather than by application programming stored within the database itself or by low level manipulation of the data (e.g. through MySQL command ).

A back-end is a database stores data but does not include end-user application elements such as stored queries, forms, macros or reports.

A database which is fronted by a web server and can be accessed by browser that connected by browsers that connects into the server. Such database are usually employed in e-commerce applications:

For examples, online bookstores details of books on a back end database which can be browsed by users looking for a specific book. Increasingly mainstream database management system provide facilities for database to be easily connected to a Web server.

**Introduction to Java net beans**

Java is a popular 3rd-Generation programming language, which can be used to perform any of the thousands of things that a computer can do. With the features it offers, Java has become the language of choice for Internet and Intranet applications. Java plays an important role for the proper functioning of many software-based devices attached to a network. The kind of functionality the Java offers, has contributed a lot towards the popularity of Java.

A computer is a set of instructions given to computer. These instructions initiate some action and hence sometimes called executable instructions. In java programs, the executable instructions are specified through methods or a function is a sequence of some declaration statements and executable statements. In other programming languages, methods are known as functions, sometimes procedures, subprograms or subroutines.

In Java, which is strictly object-oriented any action can take place through methods and methods have to exist as a part of a class.

**Introduction to MySql**

A database system is basically a computer based record keeping system. The collection of data, usually referred to as the database, contains ion about one particular enterprise. In a typical file-processing system, permanent records stored in various files. A number of different application programs are written to extract records from files and add records to the appropriate files. But this scheme has a number of major limitations and disadvantages, such as data redundancy (duplication of data), data inconsistency, unsharable data unstandardized data insecure data, incorrect data etc. A database management system is answer to all these problems

As it provides a centralized control of the data.

Various advantages of database systems are:

* Database systems reduce data redundancy (data duplication) to a large extent.
* Database systems control data inconsistency to a large extent.
* Database facilitate sharing of data.
* Database enforce standards.
* Centralized database can ensure data security.
* Integrity can be maintained through database.

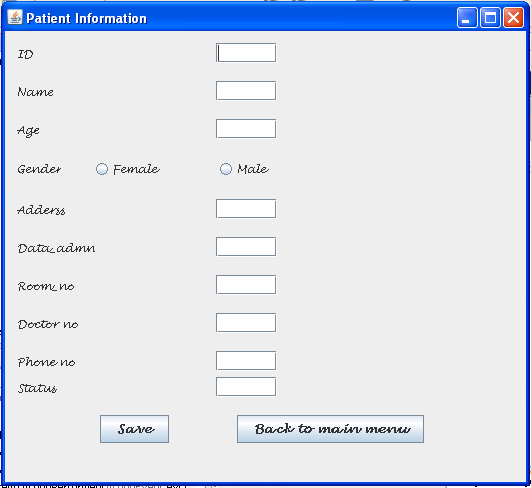
**Snapshots of Java form**

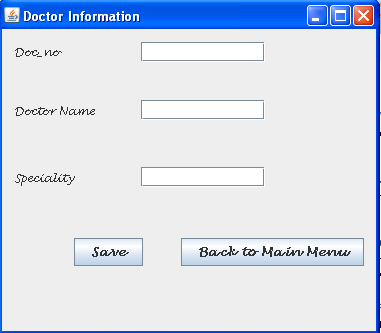
Hospital Management

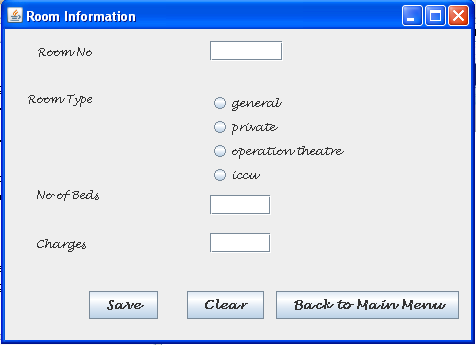


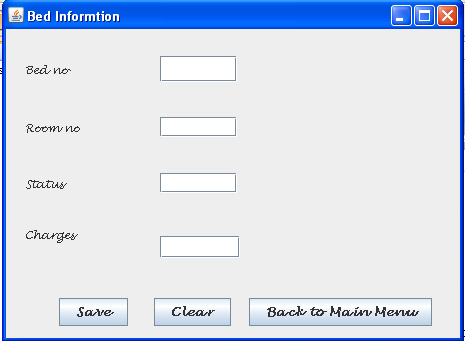
**Entry Form:**

Patient’s Information Form :

****

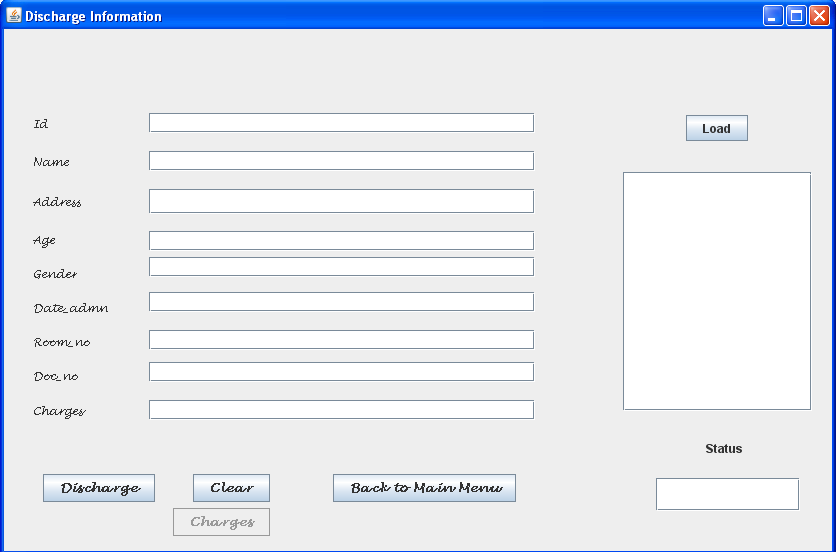
Doctor’s Information Form:

Room’s Information Form :

Bed’s Information Form:

**Discharge Form:**

Discharge Information Form :

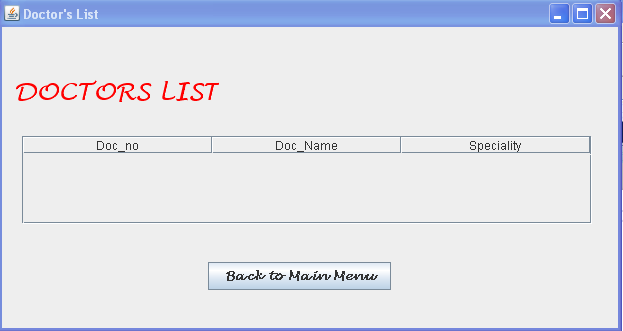


**Report Form:**

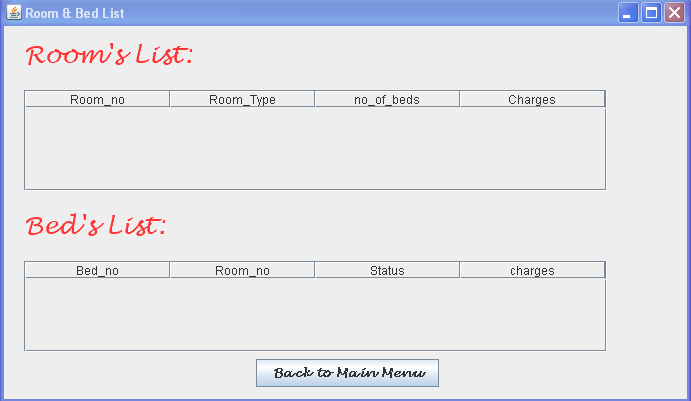
Patient’s List Form:

****

Doctor’s List Form:



Room & Bed list Form:



**Coding of Java Net-Beans**

**Entry:**

Patient Information Form:

import java.sql.\*;

import java.sql.Connection;

import javax.swing.JOptionPane;

**SAVE Button:**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/hospital","root","avs");

Statement stmt=con.createStatement();

ResultSet rs=null;

int id=Integer.parseInt(jtf.getText());

String name =jtf1.getText();

int age=Integer.parseInt(jtf2.getText());

String address =jtf3.getText();

String admn =jtf4.getText();

String status =t9.getText();

String gen=null;

if (jrb2.isSelected())

gen="M";

if (jrb1.isSelected())

{

gen="F";

}

String proomno=jtf5.getText();

int dno=Integer.parseInt(jtf6.getText());

int phno=Integer.parseInt(jtf7.getText());

String stsql="insert into patient values("+id+",'"+name+"','"+gen+"',"+age+",'"+address+"','"+admn+"',"+dno+",'"+proomno+"','"+status+"',"+phno+")";

int rowsEffect=stmt.executeUpdate(stsql);

JOptionPane.showMessageDialog(this,"Record added");

}

catch(Exception e){

JOptionPane.showMessageDialog(this,e.getMessage());

}

}

**BACK TO MAIN MENU Button:**

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

new mai().setVisible(true);

this.dispose();

}

Doctor Information Form:

import com.mysql.jdbc.Connection;

import com.mysql.jdbc.Statement;

import java.sql.\*;

import javax.swing.JOptionPane;

**SAVE Button:**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/hospital","root","avs");

Statement stmt=(Statement) con.createStatement();

ResultSet rs=null;

int doc\_no=Integer.parseInt(t1.getText());

String name =t2.getText();

String speciality =t3.getText();

String stsql="insert into doctor values("+doc\_no+",'"+name+"','"+speciality+"')";

int rowsEffect=stmt.executeUpdate(stsql);

JOptionPane.showMessageDialog(this,"Record added");

}

catch(Exception e){

JOptionPane.showMessageDialog(this,e.getMessage());

}

}

**BACK TO MAIN MENU Button:**

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

new mai().setVisible(true);

this.dispose();

}

Room Information Form:

import com.mysql.jdbc.Connection;

import com.mysql.jdbc.Statement;

import java.sql.\*;

import javax.swing.JOptionPane;

**SAVE Button:**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/hospital","root","avs");

Statement stmt=(Statement) con.createStatement();

ResultSet rs=null;

String roomtype=null;

if(gen.isSelected()){

roomtype="general";

}

if(pri.isSelected())

{

roomtype="private";

}

if(ot.isSelected())

{

roomtype="operation theatre";

}

if(ic.isSelected())

{

roomtype="ICCU";

}

int noofbeds=Integer.parseInt(t2.getText());

int charges=Integer.parseInt(t3.getText());

String roomno =t1.getText();

String stsql="insert into room values('"+roomno+"','"+roomtype+"',"+noofbeds+","+charges+")";

int rowsEffect=stmt.executeUpdate(stsql); JOptionPane.showMessageDialog(this,"Record added");

}

catch(Exception e){

JOptionPane.showMessageDialog(this,e.getMessage()) }

}

**CLEAR Button:**

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

t1.setText("");

t2.setText(null);

t3.setText(null);

**BACK TO MAIN MENU Button:**

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

new mai().setVisible(true);

this.dispose();

}

Bed Information Form:

import com.mysql.jdbc.Statement;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import javax.swing.JOptionPane;

**SAVE Button:**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/hospital","root","avs");

Statement stmt=(Statement) con.createStatement();

ResultSet rs=null;

int bed\_no=Integer.parseInt(t1.getText());

String room\_no =t2.getText();

String status =t3.getText();

int charges=Integer.parseInt(t4.getText());

String stsql="insert into bed values("+bed\_no+",'"+room\_no+"','"+status+"',"+charges+")";

int rowsEffect=stmt.executeUpdate(stsql);

JOptionPane.showMessageDialog(this,"Record added");

}

catch(Exception e){

JOptionPane.showMessageDialog(this,e.getMessage());

}

}

**CLEAR Button:**

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

t1.setText(null);

t2.setText(null);

t3.setText(null);

t4.setText(null);

}

**BACK TO MAIN MENU Button:**

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

new mai().setVisible(true);

this.dispose();

}

**Discharge:**

Discharge Information Form:

**DISCHARGE Button:**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try { Class.forName("com.mysql.jdbc.Driver");

Connection con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/hospital","root","avs");

Statement stmt=(Statement) con.createStatement();

ResultSet rs=null;

int r=Integer.parseInt(t6.getText());

String stsql="select \* from bed where status='available' && room\_no="+(r)+" ";

stmt= (Statement) con.createStatement();

rs=stmt.executeQuery(stsql);

while(rs.next()){

int charges =rs.getInt("charges");

String status=rs.getString("status");

t10.setText(""+charges);

t0.setText(status);

}

}

catch (Exception e){

}

}

**LOAD Button:**

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

try{

DefaultListModel model=(DefaultListModel)l1.getModel();

Class.forName("com.mysql.jdbc.Driver");

Connection con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/hospital","root","avs");

Statement stmt=(Statement) con.createStatement();

ResultSet rs=null;

String stsql="select \* from patient where status='available'";

rs=stmt.executeQuery(stsql);

while (rs.next()){

int a=rs.getInt(1);

String b=rs.getString(2);

model.addElement(a+"-"+b);

}

l1.setModel(model);

rs.close();

stmt.close();

con.close();

}

catch(Exception e){

JOptionPane.showMessageDialog(this,e.getMessage());

}

}

**CLEAR Button:**

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

t1.setText(null);

t2.setText(null);

t3.setText(null);

t4.setText(null);

t5.setText(null);

t6.setText(null);

t8.setText(null);

}

**BACK TO MAIN MENU Button:**

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

new mai().setVisible(true);

this.dispose();

}

**Report:**

Patients List:

import java.sql.\*;

import java.sql.Connection;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

/\*\*

**DISPLAY Button:**

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

try{

Class.forName("com.mysql.jdbc.Driver");

DefaultTableModel t=(DefaultTableModel)t1.getModel();

Connection con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/hospital","root","avs");

Statement st=con.createStatement();

String query="select \* from patient";

ResultSet r=st.executeQuery(query);

while(r.next())

{

int a=r.getInt("p\_id");

String b=r.getString(2);

String c=r.getString(3);

int d=r.getInt(4);

String f=r.getString(5);

String g=r.getString(6);

int h=r.getInt(7);

String i=r.getString(8);

String j=r.getString(9);

int k=r.getInt(10);

t.addRow(new Object[]{a,b,c,d,f,g,h,i,j,k});

}

t1.setModel(t);

r.close();

st.close();

con.close();

}

catch(Exception e){

JOptionPane.showMessageDialog(null,"Error in connection");

}

}

**BACK TO MAIN MENU Button:**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

new mai().setVisible(true);

this.dispose();

Doctors List:

private void formWindowGainedFocus(java.awt.event.WindowEvent evt) {

try{

Class.forName("com.mysql.jdbc.Driver");

DefaultTableModel t=(DefaultTableModel)t1.getModel();

Connection con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/hospital","root","avs");

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

String query="select \* from doctor";

ResultSet r=st.executeQuery(query);

while(r.next())

{

int a=r.getInt(1);

String b=r.getString(2);

String c=r.getString(3);

t.addRow(new Object[]{a,b,c});

}

t1.setModel(t);

r.close();

st.close();

con.close();

}

catch(Exception e){

JOptionPane.showMessageDialog(null,"Error in connection");

}

}

**BACK TO MAIN MENU Button:**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

new mai().setVisible(true);

this.dispose();

}

Room List & Bed List:

private void formWindowGainedFocus(java.awt.event.WindowEvent evt) {

try{

Class.forName("com.mysql.jdbc.Driver");

DefaultTableModel t=(DefaultTableModel)t1.getModel();

Connection con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/hospital","root","avs");

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

String query="select \* from room";

ResultSet r=st.executeQuery(query);

while(r.next())

{

String a=r.getString(1);

String b=r.getString(2);

int c=r.getInt(3);

int d=r.getInt(4);

t.addRow(new Object[]{a,b,c,d});

}

t1.setModel(t);

r.close();

st.close();

con.close();

}

catch(Exception e){

JOptionPane.showMessageDialog(null,"Error in connection");

}

try{

Class.forName("com.mysql.jdbc.Driver");

DefaultTableModel t=(DefaultTableModel)t2.getModel();

Connection con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/hospital","root","avs");

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

String query="select \* from bed";

ResultSet r=st.executeQuery(query);

while(r.next())

{

int a=r.getInt(1);

int b=r.getInt(2);

String c=r.getString(3);

int d=r.getInt(4);

t.addRow(new Object[ ]{a,b,c,d});

}

t2.setModel(t);

r.close();

st.close();

con.close();

}

catch(Exception e){

JOptionPane.showMessageDialog(null,"Error in connection");

}

}

**BACK TO MAIN MENU Button:**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

new mai().setVisible(true);

this.dispose();

**Coding of MY SQL**

create database hospital;

use hospital;

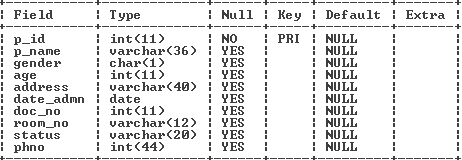
create table patient(p\_id int primary key, p\_name varchar(36), gender char(1), age int, address varchar(40), date\_admn date, doc\_no int, room\_no varchar(12), status varchar(20), phno int(44));

create table doctor(doc\_no int primary key, doc\_name varchar(36), speciality varchar(36));

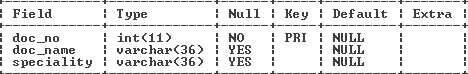
create table room(room\_no varchar(12), room\_type varchar(36), no\_of\_bed int, charges int);

create table bed(bed\_no int, room\_no varchar(12), status char(12), charges int);

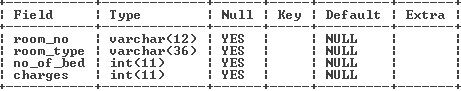
desc patient;



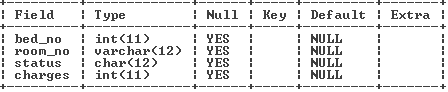
desc doctor;



desc room;



desc bed;



insert into room values('gen1', 'general', 40, 300);

insert into room values('gen2', 'general', 30, 300);

insert into room values('gen3', 'general', 35, 300);

insert into room values('priv1', 'private', 1, 550);

insert into room values('priv2', 'private', 1, 550);

insert into room values('priv3', 'private', 1, 550);

insert into room values('priv4', 'private', 1, 550);

insert into room values('ot1', 'operation theatre', 1, 800);

insert into room values('ot2', 'operation theatre', 1, 800);

insert into room values('ot3', 'operation theatre', 1, 800);

insert into room values('iccu1', 'ICCU', 1, 1200);

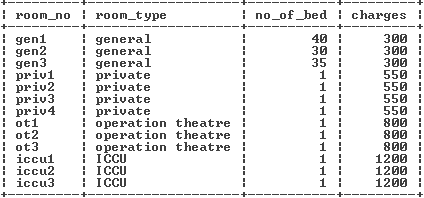
insert into room values('iccu2', 'ICCU', 1, 1200);

insert into room values('iccu3', 'ICCU', 1, 1200);

select \* from patient;

select \* from doctor;

select \* from room;



select \* from bed;

**Conclusion**

The making of this project is a quite interesting and acknowledgeable approach to Java Database Connectivity for us. This project can be instrumental in Hospital Management.

The project **Hospital Management System (HMS)** is for computerizing the working in a hospital. The software takes care of all the requirements of an hospital and is capable to provide easy and effective storage of information related to patients that come up to the hospital.

This project deals in entry of the patient in the hospital, allotment of duties to the doctors, room service, bed allotment. This project even tells the discharge report & charges to be taken by the patients. The last function of this project is to tell about the report of patients, doctors,rooms and beds.

Thus this SOFTWARE is extremely useful in saving the time and efforts of users.