An Intra Disciplinary Project report on

HOSPITAL MANAGEMENT SYSTEM

*Y.PAVAN KUMAR (Reg. No: 191FA04057)*

*G.JAYANTH (Reg.No : 191FA04380)*

*V.AVINASH (Reg. No: 191FA04053)*

*M.SAATHWIK(Reg. No: 191FA04042)*

*Under the guidance of*

Mr. NARENDRA

Assistant Professor

Dept Of Cse

Mr. VEERABRAHMAM

Assistant Professor

Dept Of Cse



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**VIGNAN’S FOUNDATION FOR SCIENCE AND TECHNOLOGY RESEARCH**

(Deemed to be University)

**Vadlamudi, Guntur - 522213, INDIA.**

**FEBRUARY 2021**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEEING**

###### BONAFIDE CERTIFICATE

This is to certify that the report entitled “**Hospital Management System**” is submitted by **k.jayanth(191fa04057),v.avinash(191fa04053),m.saathwik(191fa04042),**

**y.pavankumar(191fa04057)”** in the partial fulfilment of course work of intra disciplinary project, carried out in the department of CSE, VFSTR Deemed to be University.

|  |  |  |
| --- | --- | --- |
| **SIGNATURE** | **SIGNATURE** | **SIGNATURE** |
| **Dr. Venkatesulu D,** | **Mr.Narendra** | **Mr.veerabrahmam** |
| **HEAD OF THE DEPARTMENT**  Professor | **Oops through java Faculty.**  Assistant Professor | **DBMS Faculty**  Assistant Professor |
| Department of CSE | Department of CSE | Department of CSE |

Submitted for the External Review held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Internal Examiner External Examiner**

|  |  |  |
| --- | --- | --- |
|  | **TABLE OF CONTENT** | **PAGE. No** |
|  | **ABSTRACT** | 4 |
|  | INTRODUCTION   * 1. **Problem Statement** | 4 |
|  | PROJECT DESCRIPTION   * 1. **Features and Functions**   2. **Users** | 6-8 |
|  | REQUIREMENT ANALYSIS  * 1. **Functional Requirements**   2. **Software Requirement**   3. **Hardware Requirements** | 4 |
|  | **DESIGN AND DEVELOPMENT**   * 1. **Database Design**   2. **User Interface Design**   3. **Implementation**   4. **Source code** | 9-52 |
|  | **CONCLUSION** | 53 |

ABSTRACT:

This project Hospital Management system includes registration of patients, storing their details into the system, and also computerized billing . The software has the facility to give a unique id for every patient and stores the details of every patient and hospital tests done automatically. User can search details of a patient using the id. The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or staff or doctor. Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

PROBLEM STATEMENT:

* A system to manage the activities in a hospital

Patients request for appointment for any doctor. The details of the existing patients are retrieved by the system. New patients details Are updated in the system before they request for appointment with the help of receptionist. The receptionist confirms the appointment based on the availability of free slots for the respective doctors and the patient is informed. receptionist may cancel the appointment of tomorrow scheduled patient.

FEATURES AND FUNCTIONS:

* Auto generation of patient id
* Auto detection of user id
* Searching of patient’s details with id
* Storing of patient data
* Storing medicines quantity
* Remembering of patient having appointment tomorrow
* Display list of medicines having expiry date within 5 days

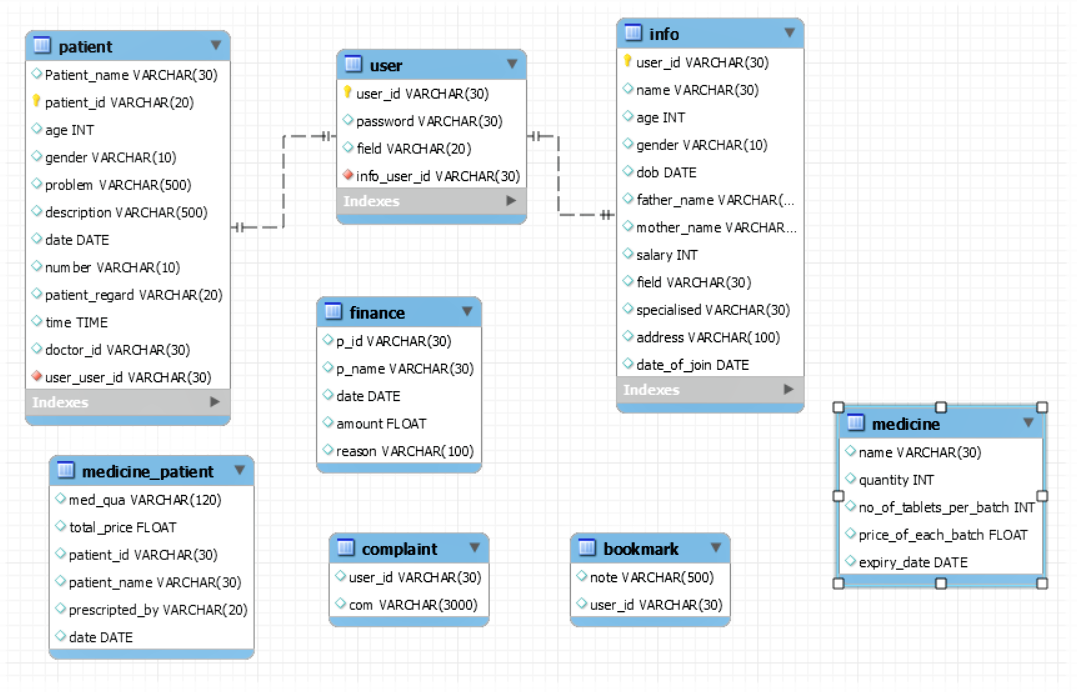
HARDWARE AND SOFTWARE REQUIREMENTS:

* Application Developing : NetBeans
* Data Storage : mysql
* Java version: jdk 15
* Pc with given specifications:
* Ram:8 gb
* Storage:1tb hdd
* Processor : i5 9th Gen
* Os : Windows

Users:

* Admin
* Doctor
* Receptionist
* Financer
* Medicine staff

DATABASE DESIGN:



DESCRIPTION:

My project Hospital Management system includes registration of patients, storing their details into the system. It will also contain doctor’s information .My software has the facility to give a unique id for every patient and stores the details of every patient and staff automatically. It includes a search facility. receptionist can search availability of a doctor and the details of a patient using the id.

The Hospital Management System can be used by entering their respective user id and password. It is accessible either by an administrator or receptionist or doctor. Only the respective person can add data in the database. The data can be retrieved easily. The interface is very user-friendly.

FACILITIES PROVIDED BY INTERFACE:

* The pages are having error tags which responsible for user friendly interface.
* Whenever doctor creating an account he kneed to give an unique user id the doctor knows when the user id is typed.
* There is no need of giving patient id the patient id is auto generated so that we can search the details of him with the help of the id.
* Whenever there is a work or operation the Receptionist get an alert.
* The patients who are having appointment tomorrow will be shown .
* The finance reports will be saved in the database to ensure the correctness.
* The medicines reports will be saved in the database to ensure the correctness.
* Expiry medicines will be shown before five days.
* Admin can add remove or see statistics of the staff.

ADVANTAGES OF HOSPITAL MANAGEMENT SYSTEM:

* Time-saving Technology
* Improved Efficiency by avoiding human errors
* Reduces scope for Error
* Data security and correct data retrieval made possible
* Cost effective and easily manageable
* Easy access to patient data with correct patient history
* Improved patient care made possible
* Easy monitoring of supplies in inventory
* Reduces the work of documentation
* Better Audit controls and policy compliance.

Features of[Hospital Management System](https://mocdoc.in/util/hospital-management-system):

* Appointment Management
* Billing Management
* Prescription Management
* Discharge Summary(future scope)
* Operation Management
* Pharmacy Management
* Lab Management
* Master Information System
* Manage Multiple Locations
* Easy Patient data retrieval
* Increased Data security
* Improve Visibility and Transparency
* Improved Quality Control
* Improved Management Visibility
* Ease to Access System Facilities
* Cost Effective

List of modules and tasks:

Admin:

1to add the Staff

2To delete staff

3to see statistics of doctor

4to see the complaints given by staff

Doctor:

1to see the patients he need to treat on that particular day

2to see his statistics

Receptionist:

1to give appointments to patient

2to show the people who have appointment tomorrow with phone number

Finance:

1to store the bill paid by every patient

2to search for the data

Medicine:

1to add and store the info of medicine

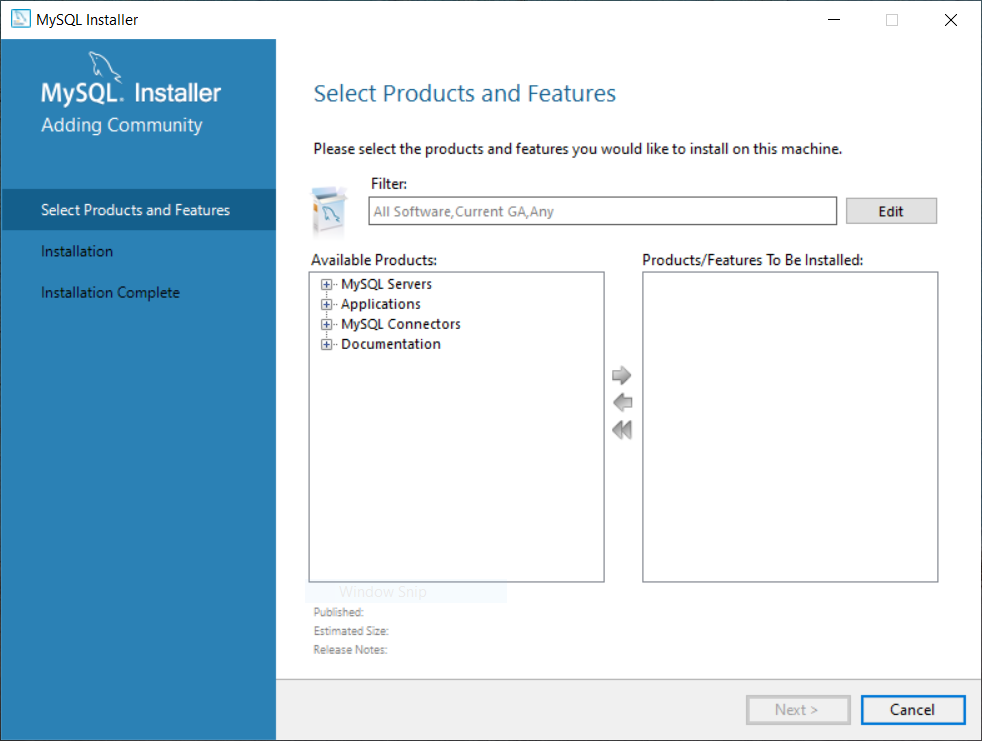
2to give medicines to patients and store details

3to show the list of medicines expires in 5 days

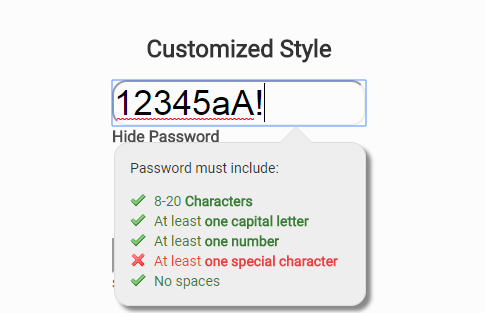
4to search for details of patient/medicine

OTHER DESCRIPTION ABOUT PROJECT.

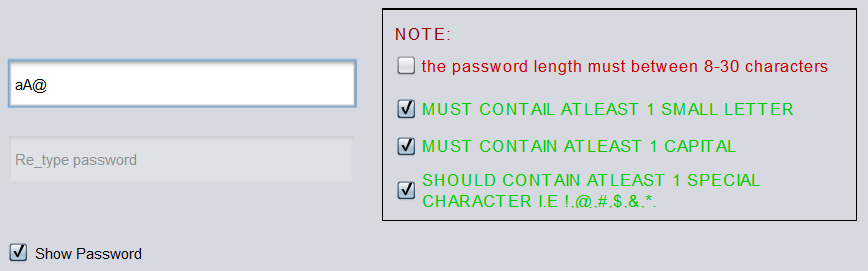
* The design of the project is taken from mysql design which is combination of white and blue.
* Most of the design are taken from applications.



* Another interesting thing is password checker.



(\* Google reference picture)



(\* Project picture)

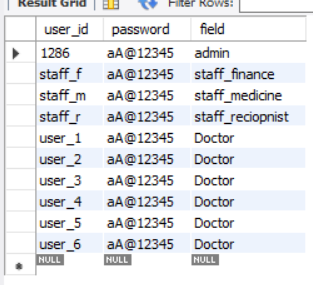
USER INTERFACE DESIGN:

LOGIN:

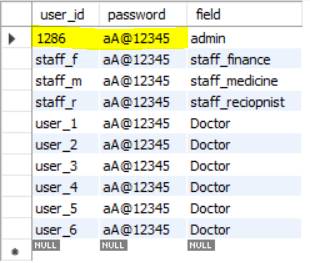
* Every user need to login using login page as shown in the figure.
* Every user has a user id and their respective password.



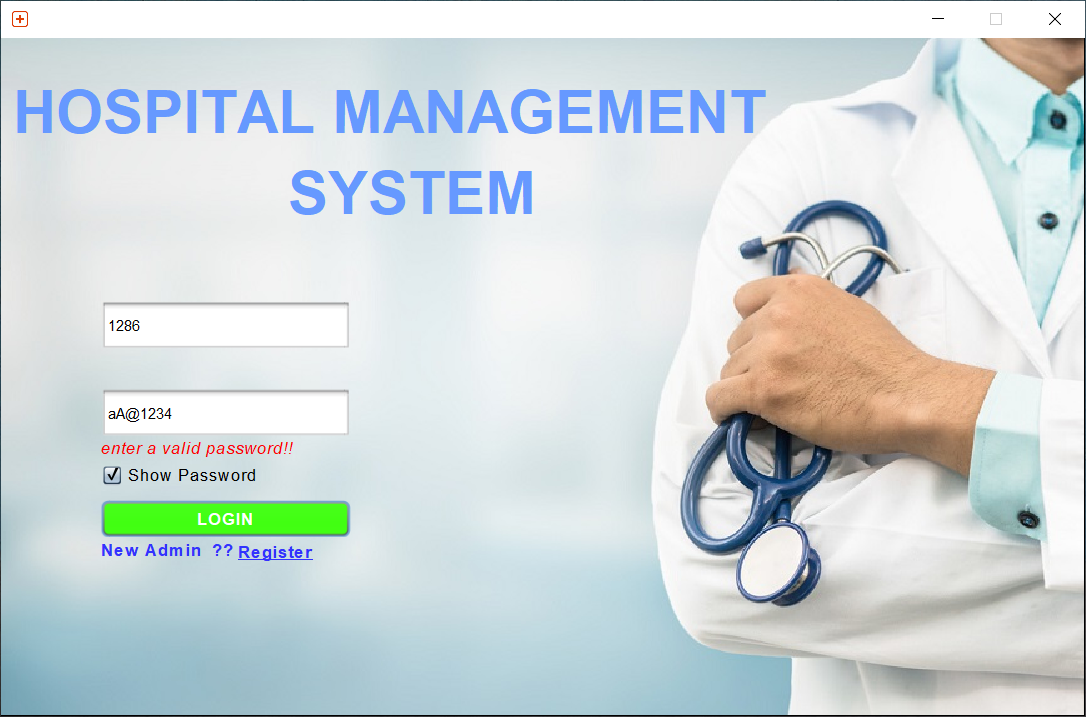
* Username , password and field of users will be saved in user table.



* If userid or password is wrong then a red color error tag will be displayed that entered user id or password is wrong as shown below.



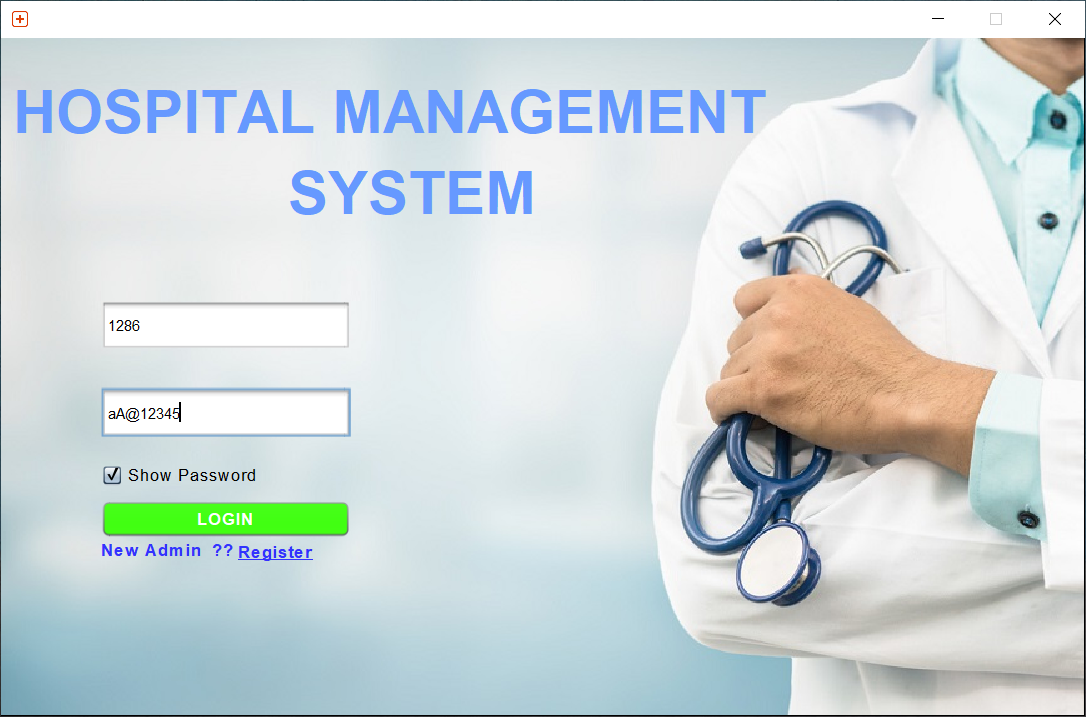
* the user id is 1286 and password is aA@12345 for admin.



* here we entered a wrong password actual password is aA@12345.
* If we enter a wrong user name then it show entered wrong user id.

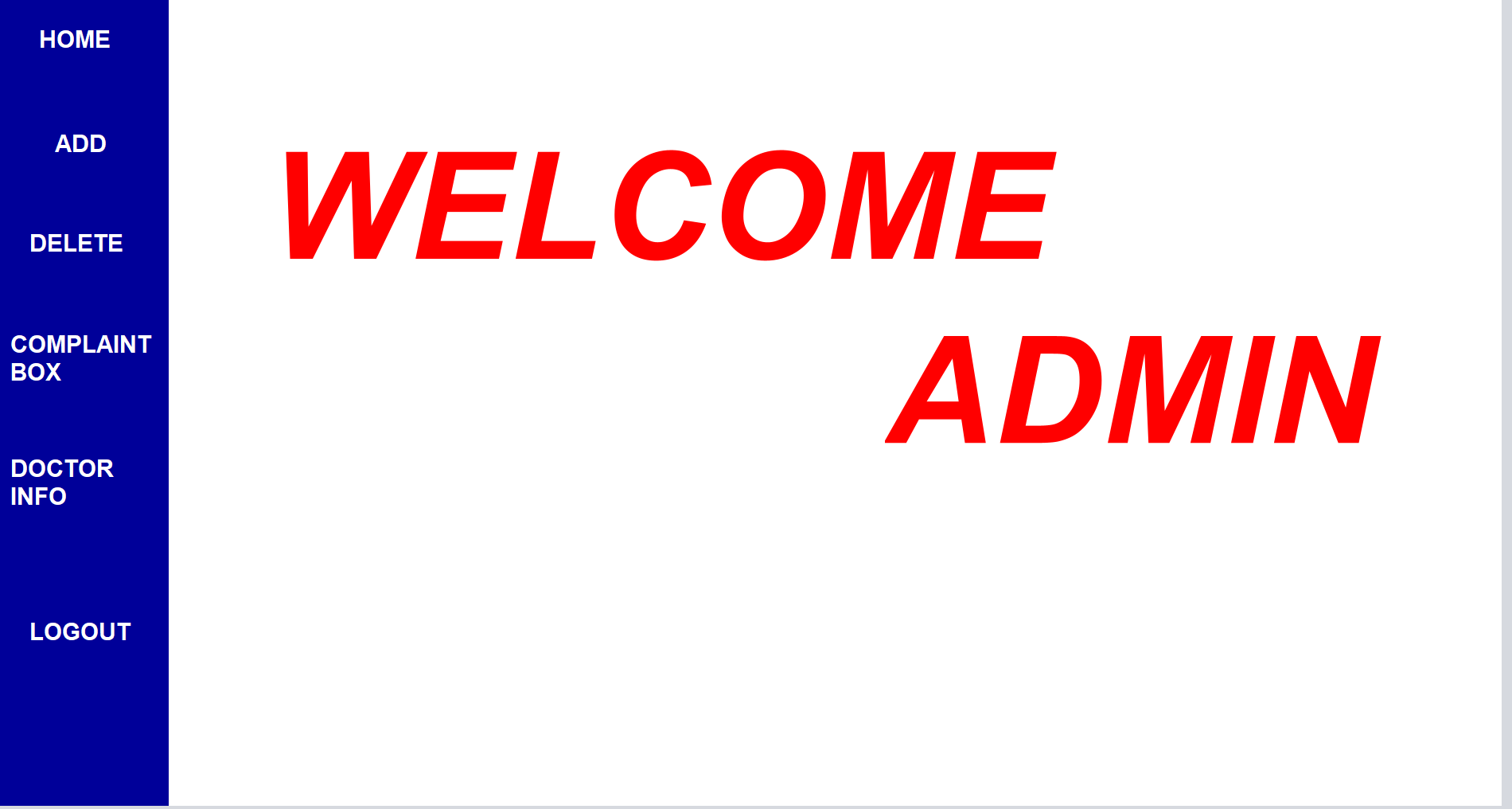


* We need to enter a valid user id and password in order to login to respective field page.



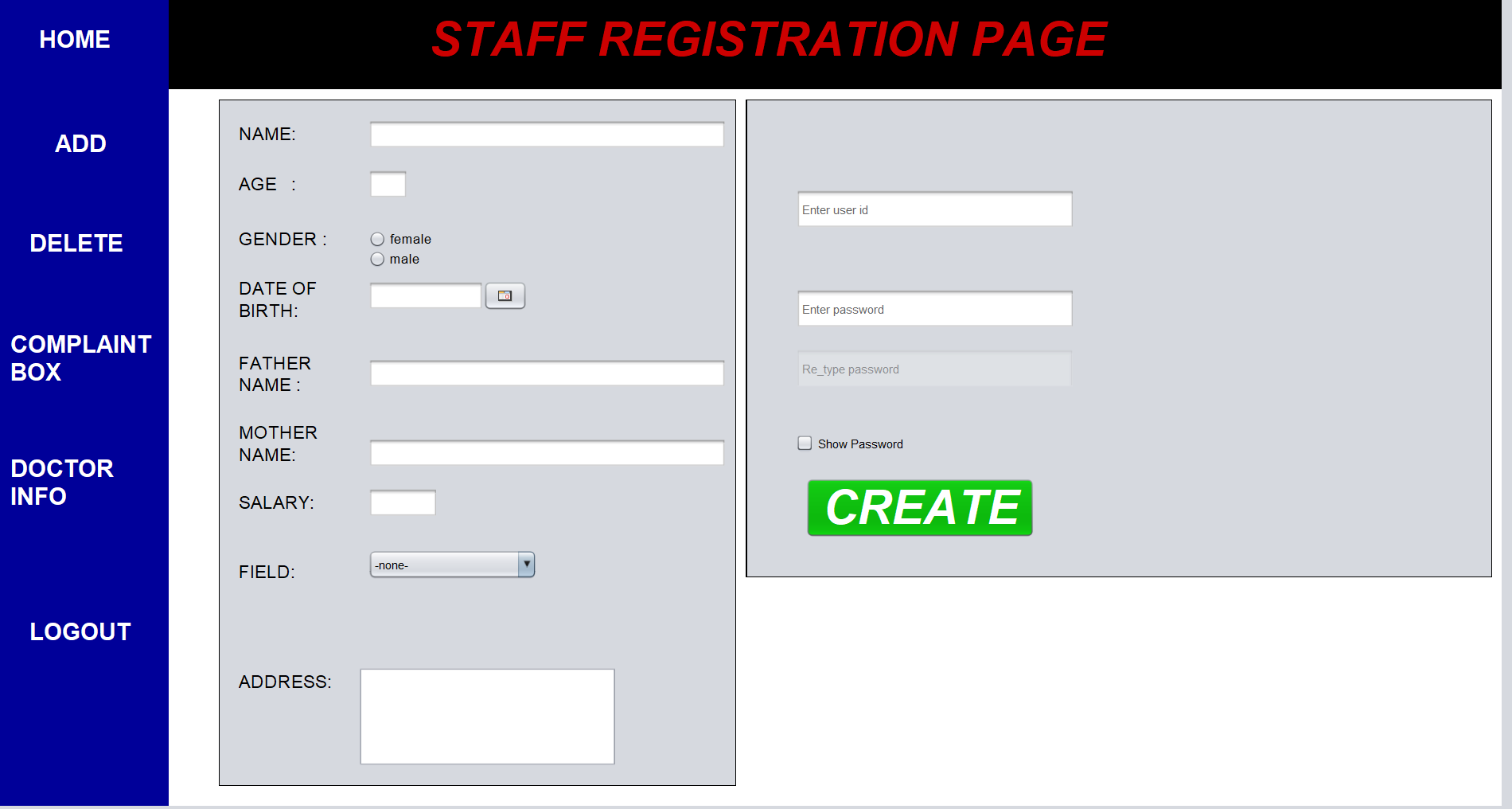
* If we enter a valid user id and password then their respective page will be opened.

ADMIN:

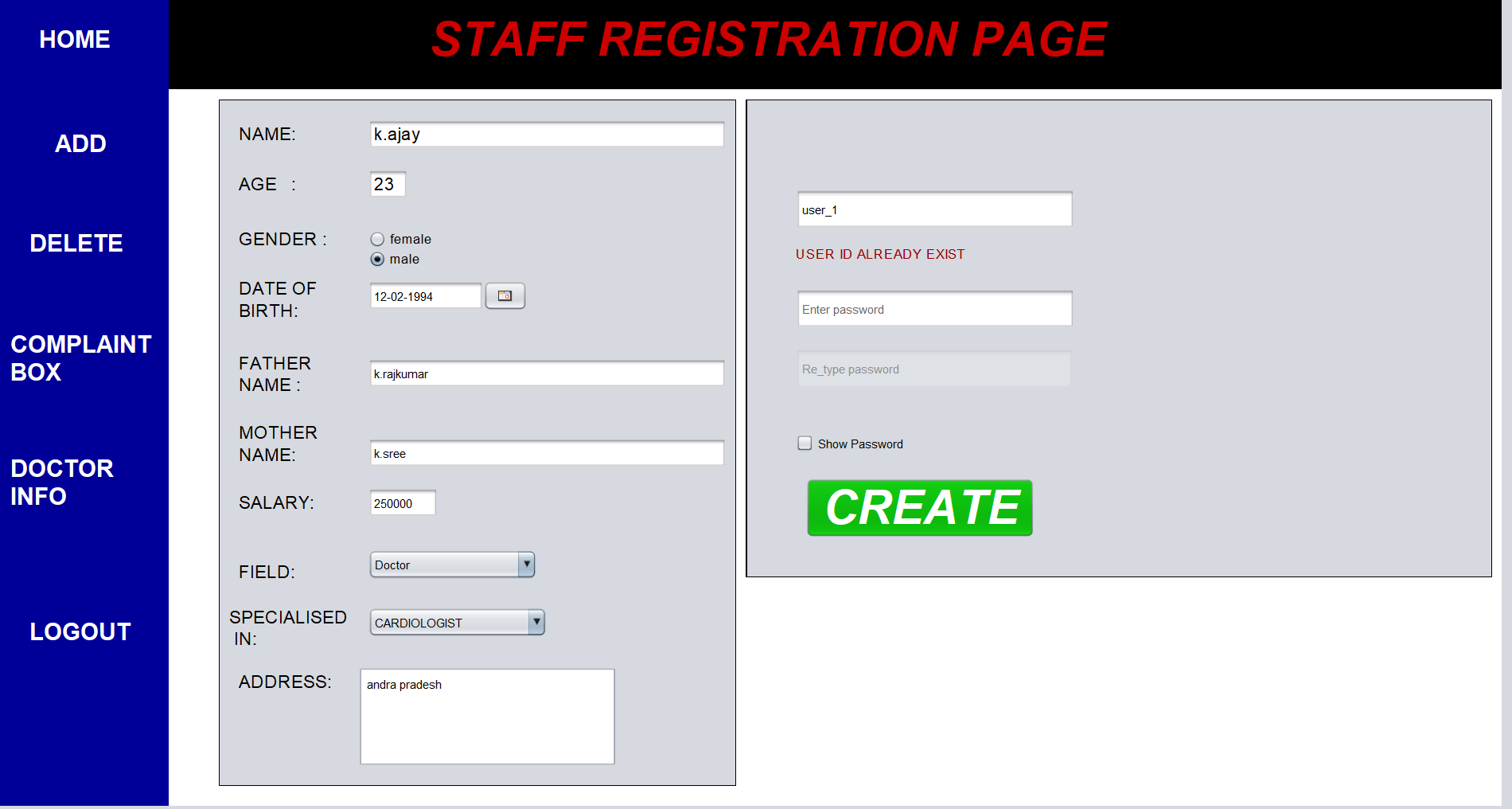


* 1286 is the user id of admin so admin page will be opened.
* In admin page there are 4 pages.
* 1st page is Add which is used to add the staff.
* 2nd page is delete which is used to delete the staff.
* 3rd page is complaint box which is used to get the complaints from the users.
* 4th page is doctors info which is used to see the statistics of doctor.
* Finally logout button is used to goto login page by closing the admin page.

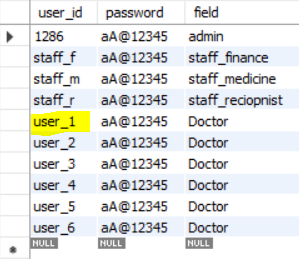
ADDPAGE:



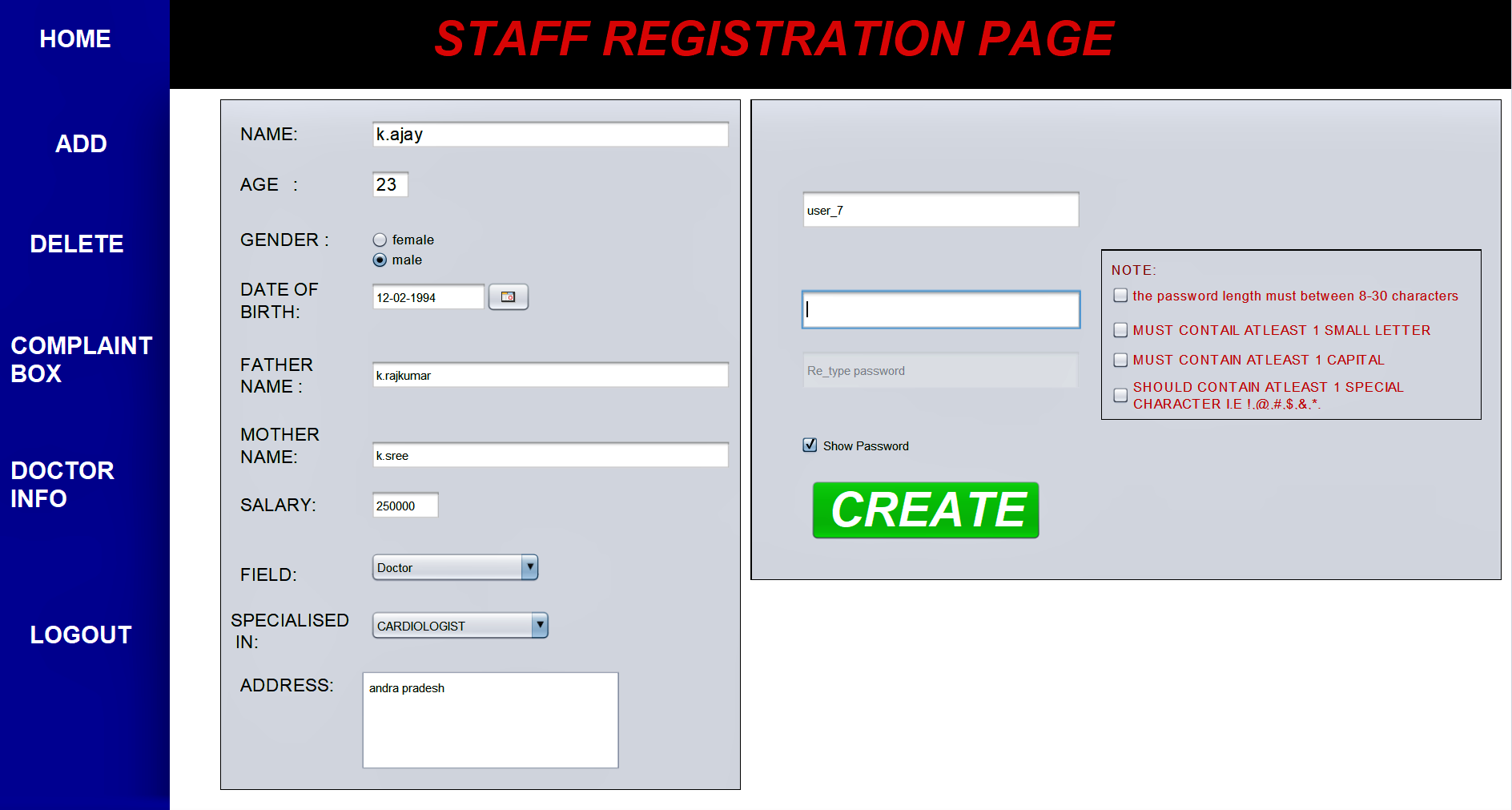
* The add registration page consist of some fields we need to fill them all correctly in order to add the staff.

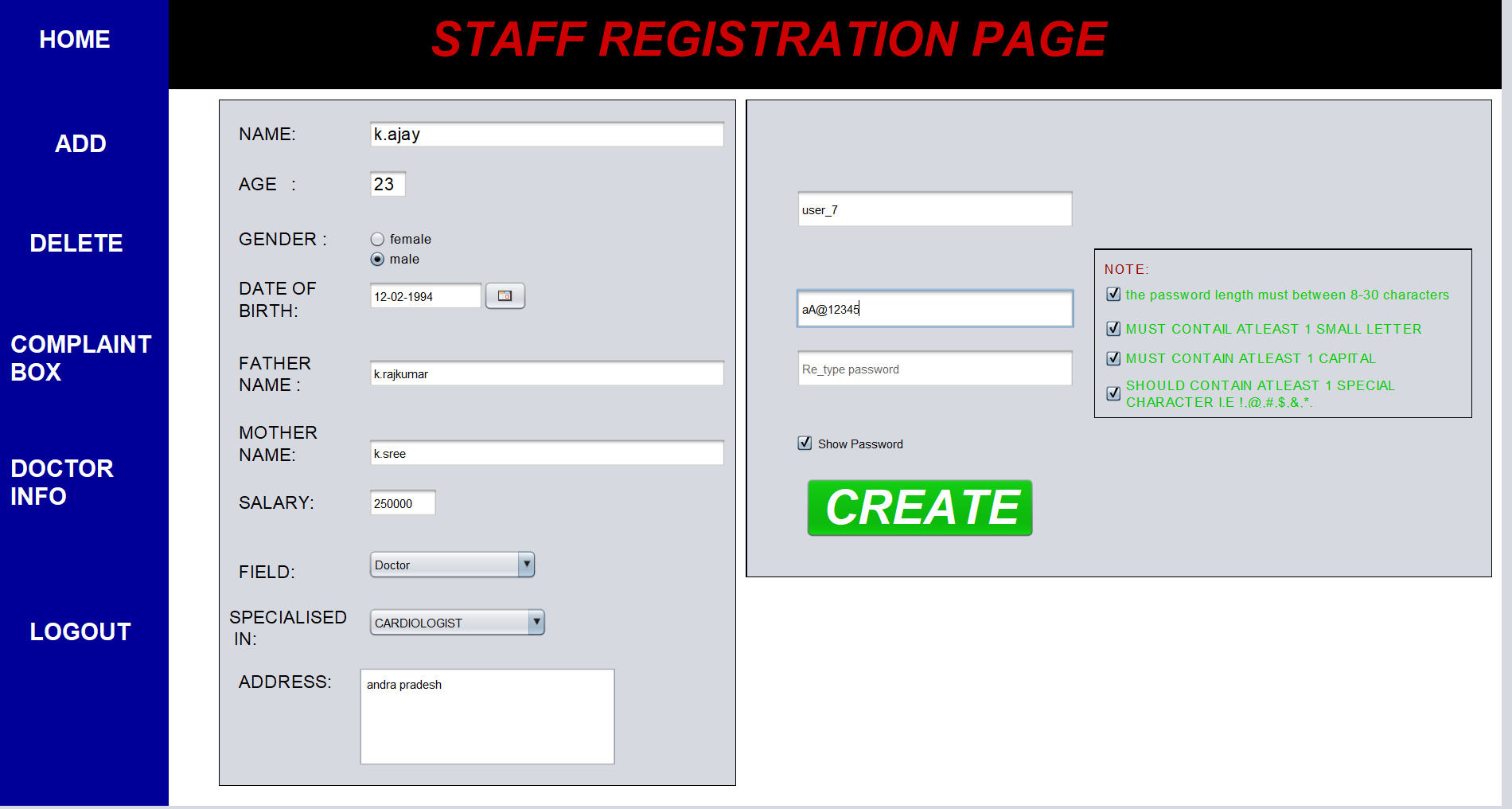


* if we enter a user id that is already exist in the database it show an error tag tha user id already exist.
* We need to enter a user id that is not in the database.



* Here user\_1 is already exists in the database so it asks us to use another userid.

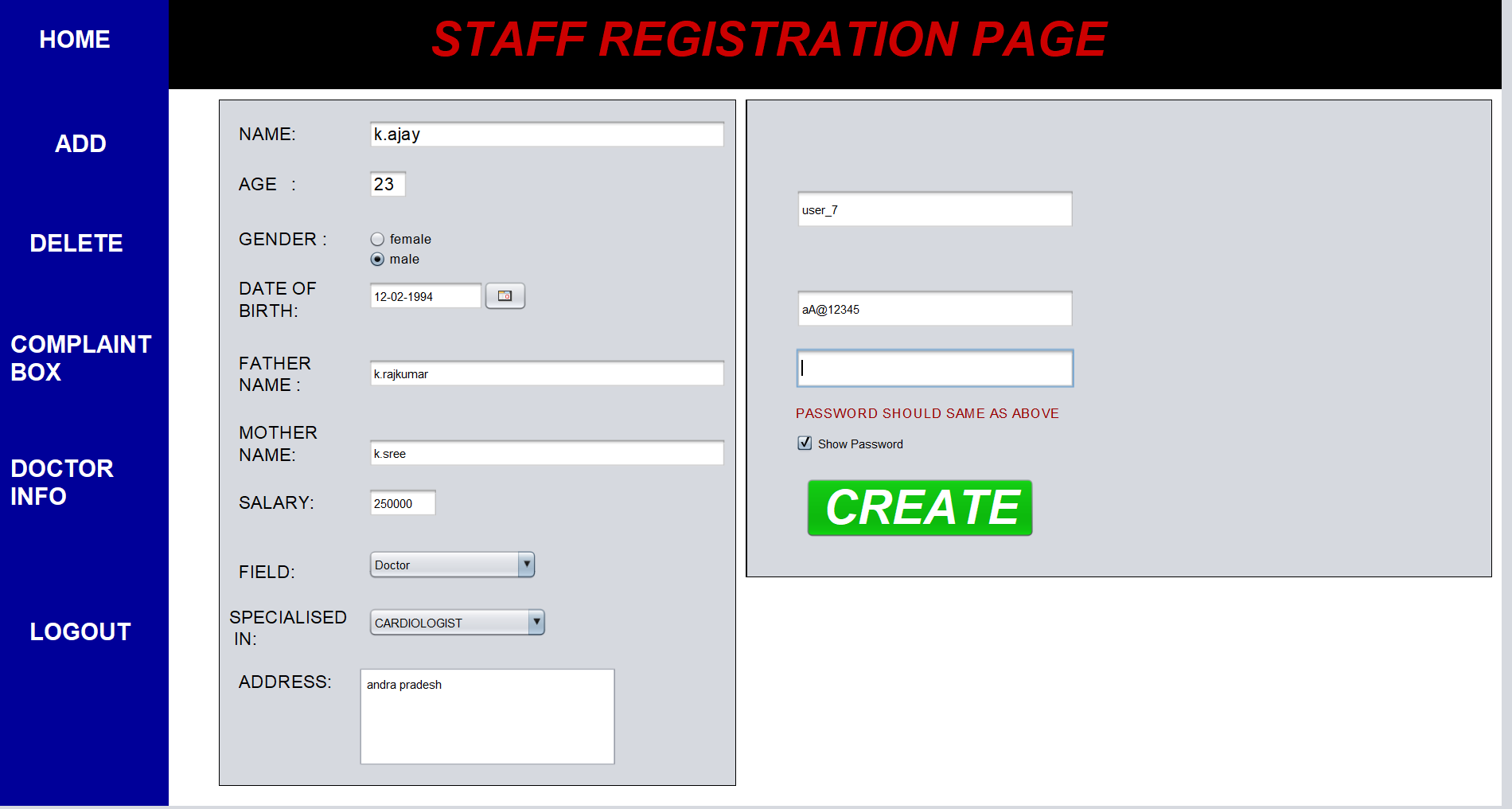




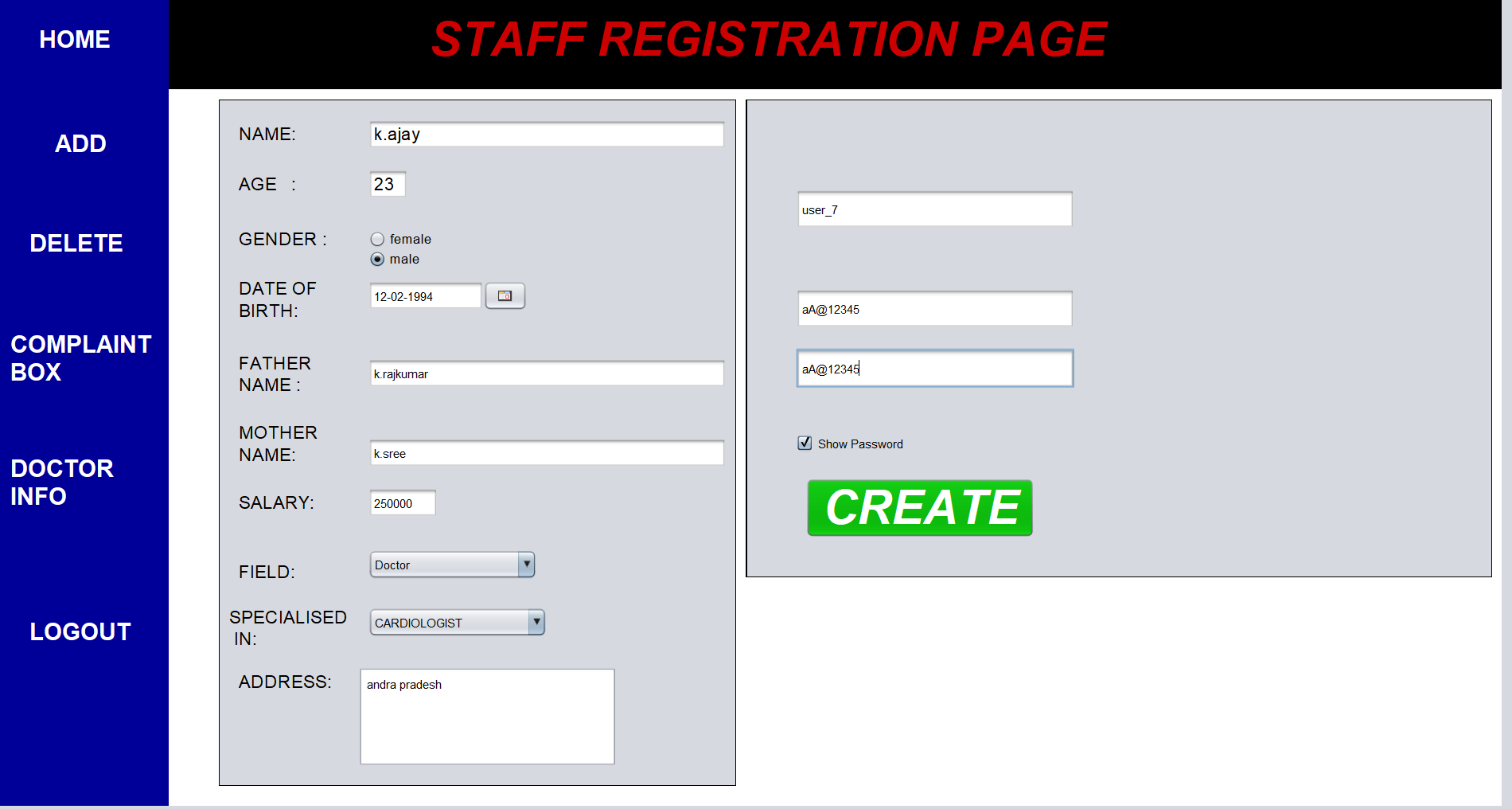
* Here we need to enter a password which satisfy the following conditions to unlock retype password field.

1. At least 1 small letter
2. At least 1 capital letter
3. 1 special symbol i.e !,@,#,$,&,\*.
4. The length of password must be greater than 8.

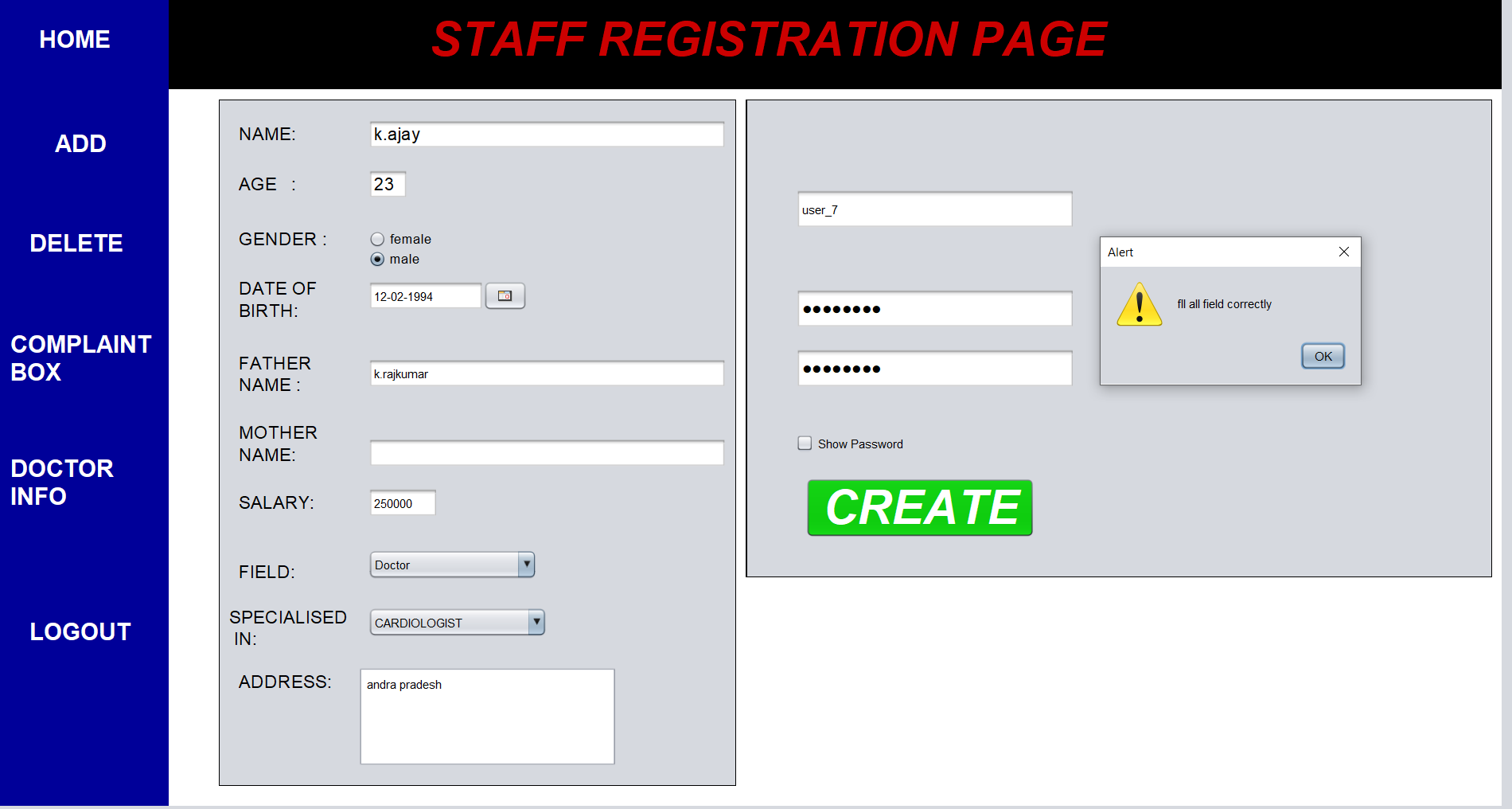
* In retype password field we need to enter same password as in password field else it shows an error tag password should same as above.



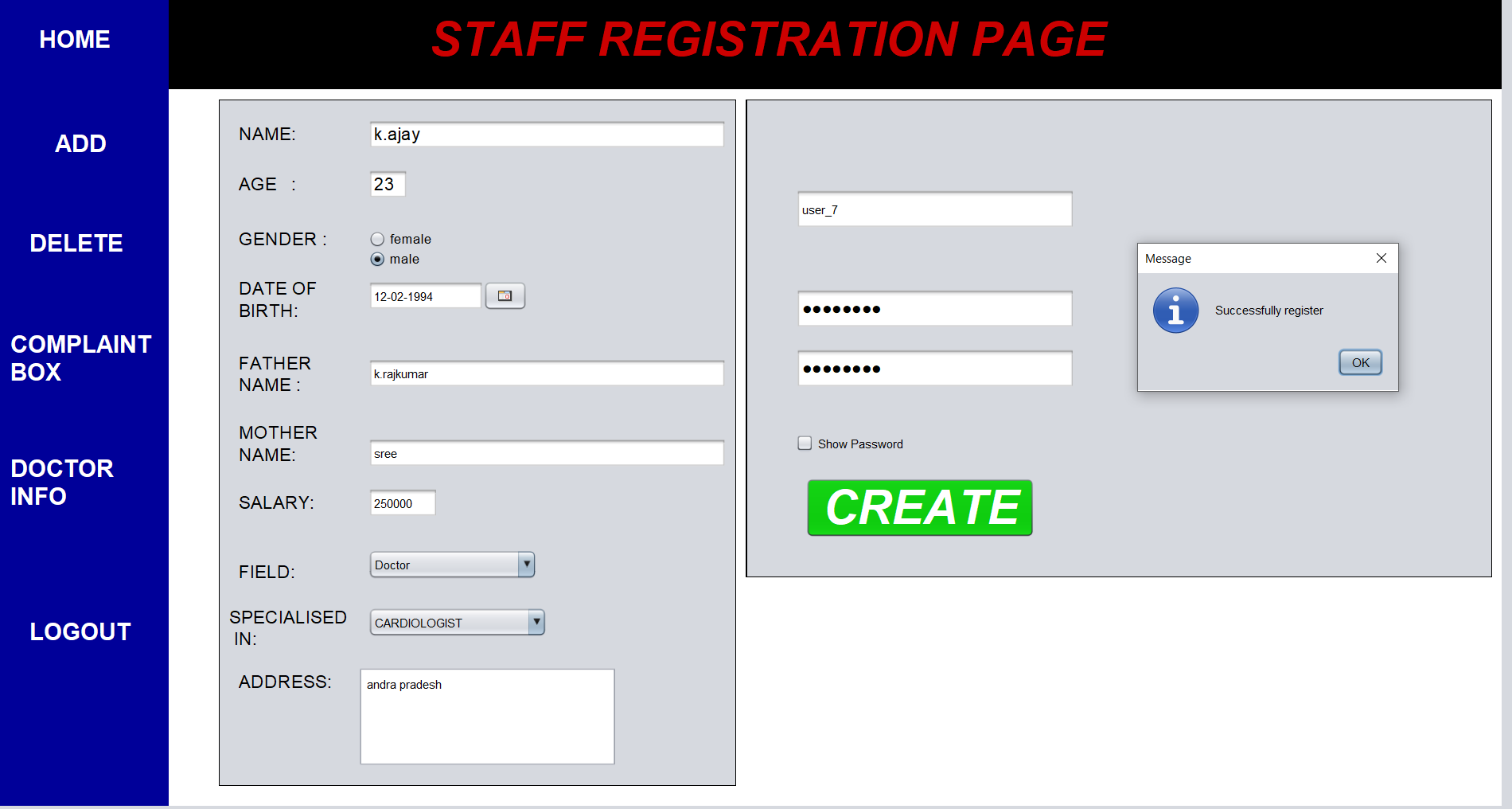
* If we enter password same as in password field then the error tag will be gone.



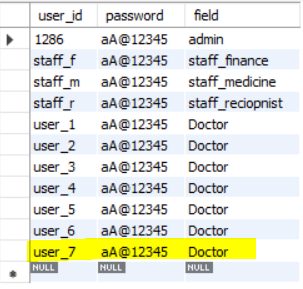
* We need to enter the all the fields in order to create account and also there should be no error tags in the page.

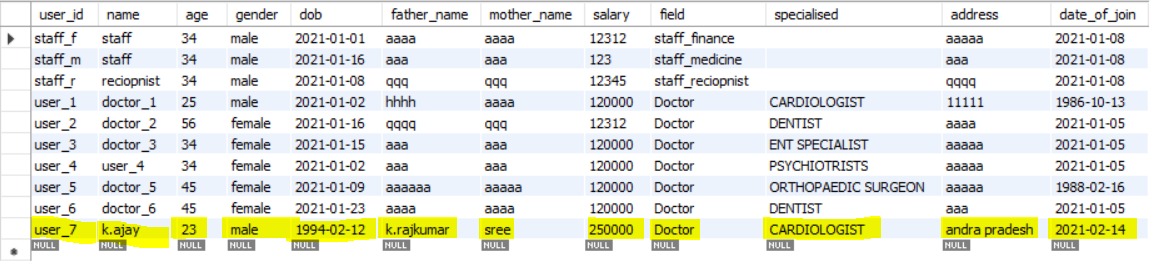


* In the above page the mother name field is not filled so that’s why it is showing an warning message to fill all the fields correctly.



* If the registration is successful it will be show registration is success full and save the details in the database:

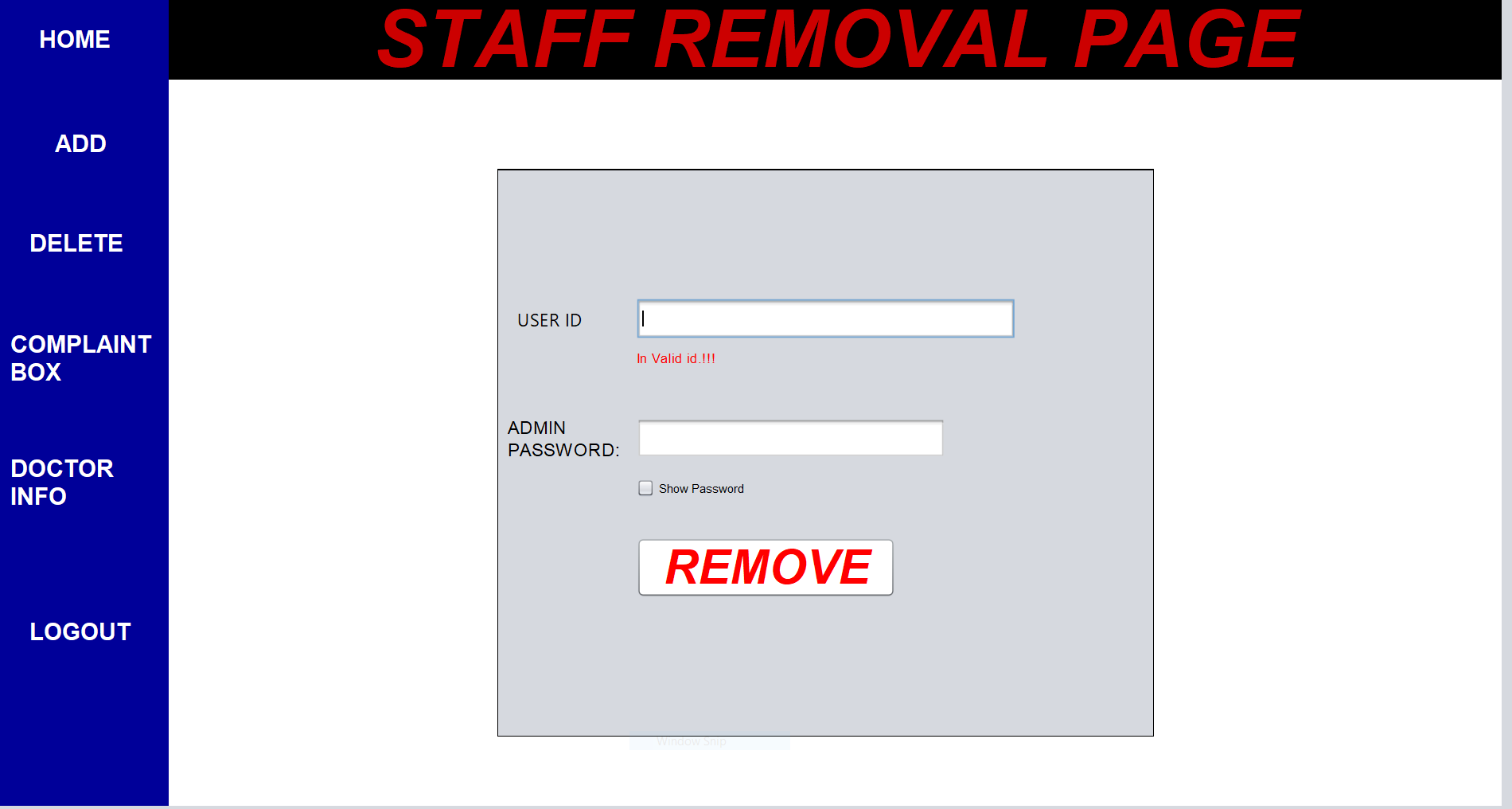




* User\_7 will be saved in both user table and info table.

DELETE:

* Delete page is used to delete a user from database.



* We need to enter a valid user id in order to remove in valid id tag.
* Here we need to enter a valid password else it show that enter a valid password.

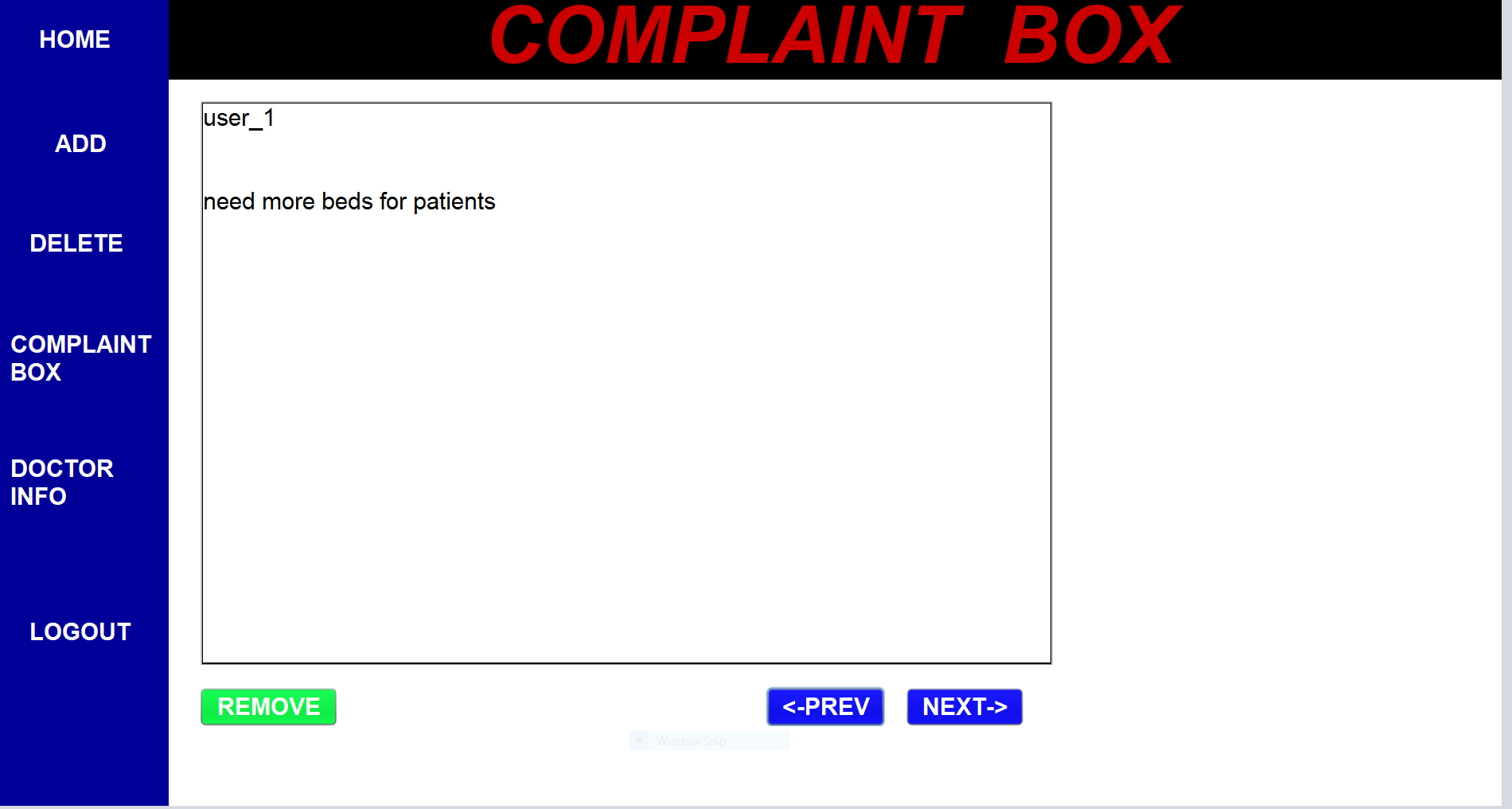


* we entered password as aA@12345 but the actual password is aA@12345.

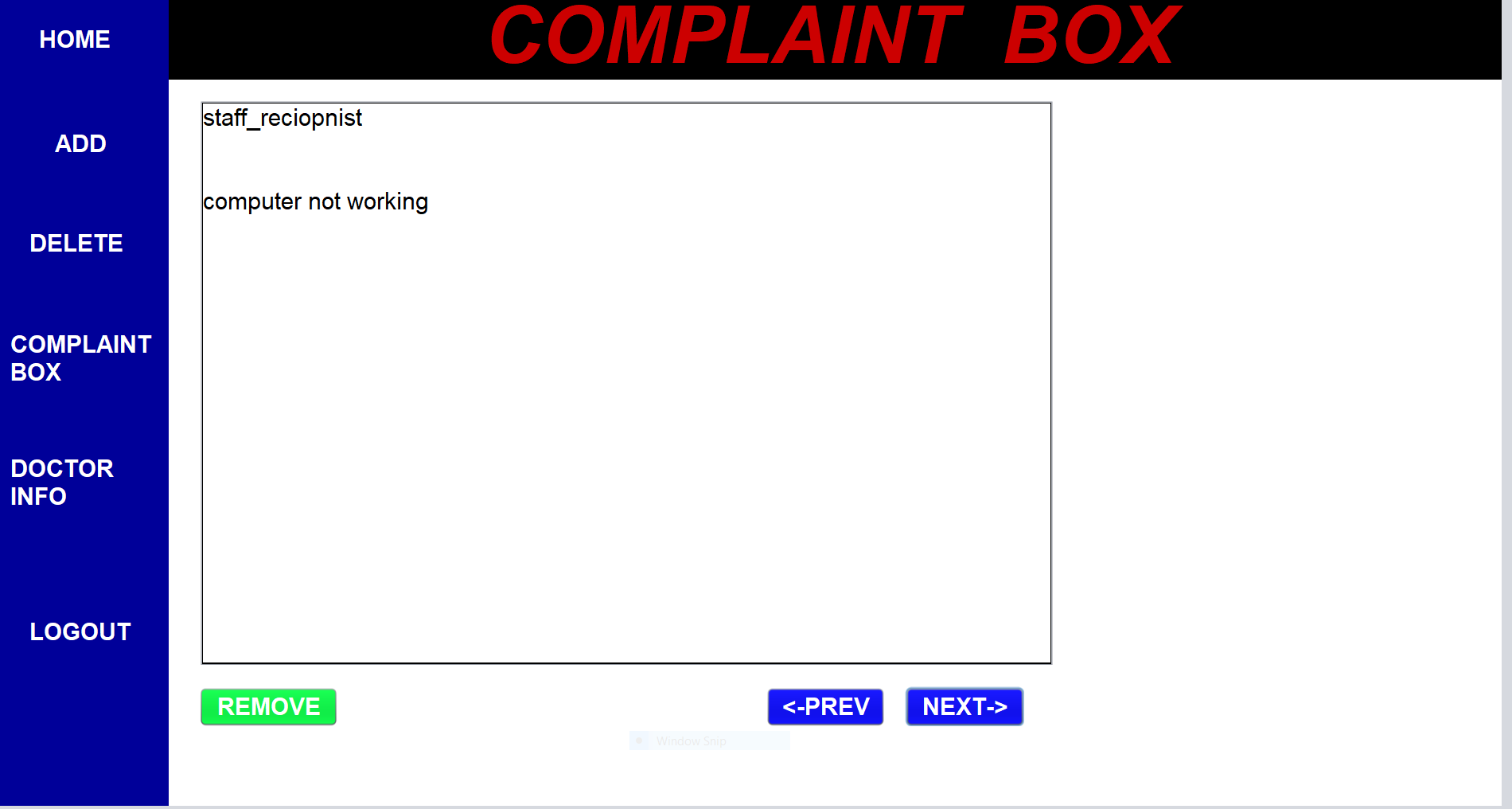


* if we enter a valid password then the user removed from the database.

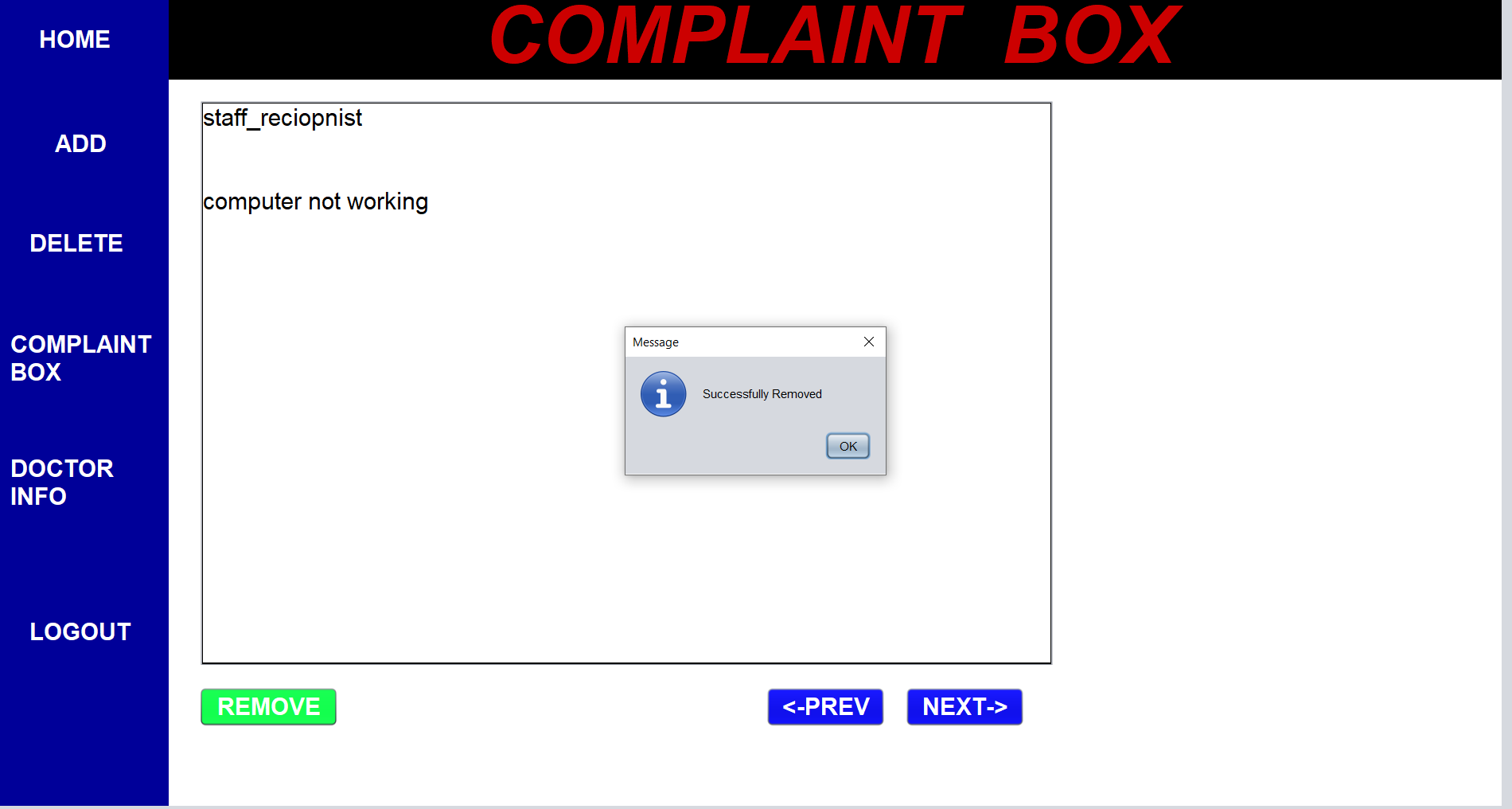
COMPLAINT BOX:



* complaint box is shown as user name followed by complain as show in figure

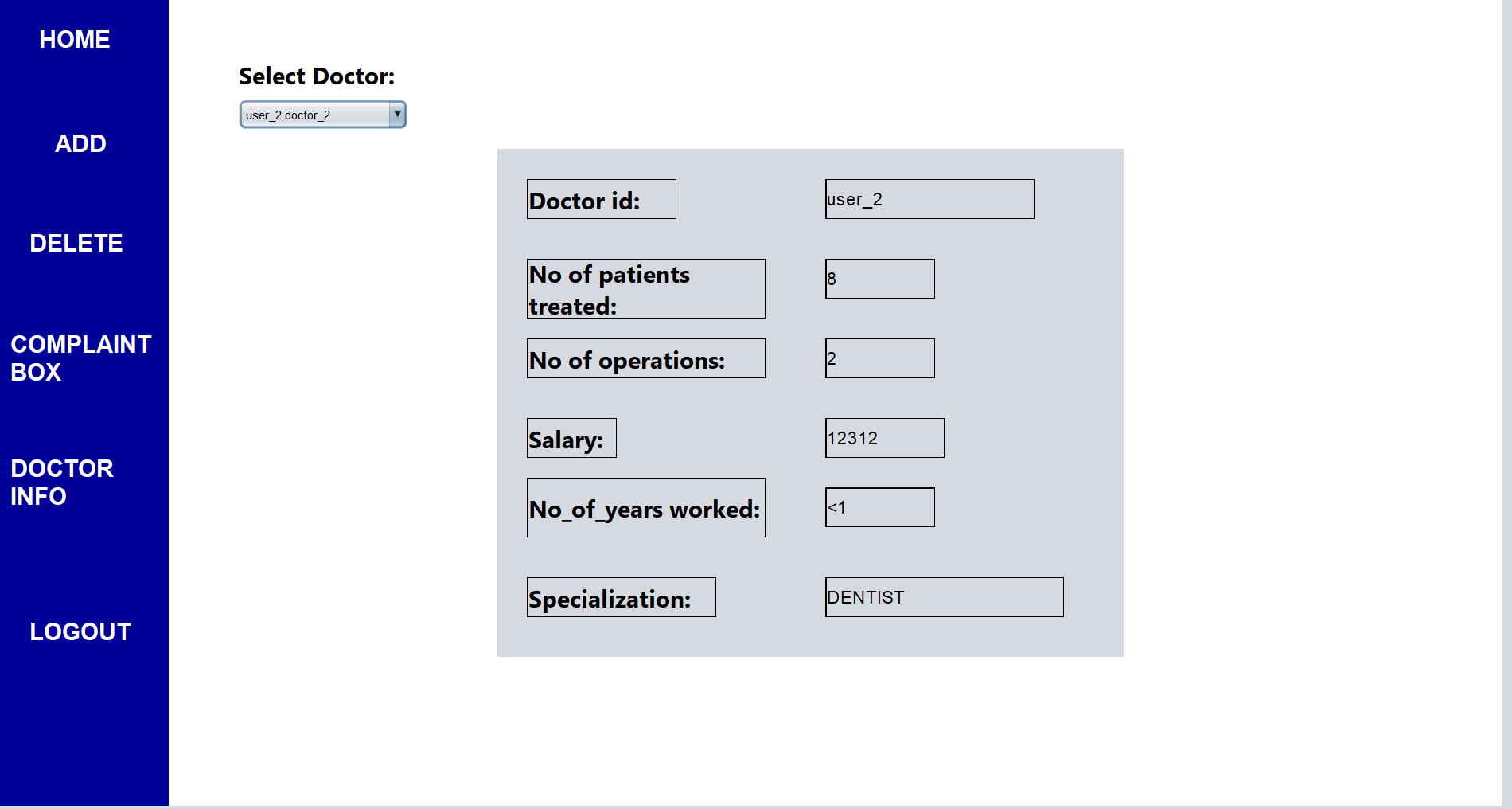


* if we want to go to next or previous complaint we can use the buttons as shown in figure.
* If the complaint is resolved by the user then he can delete it by using remove button



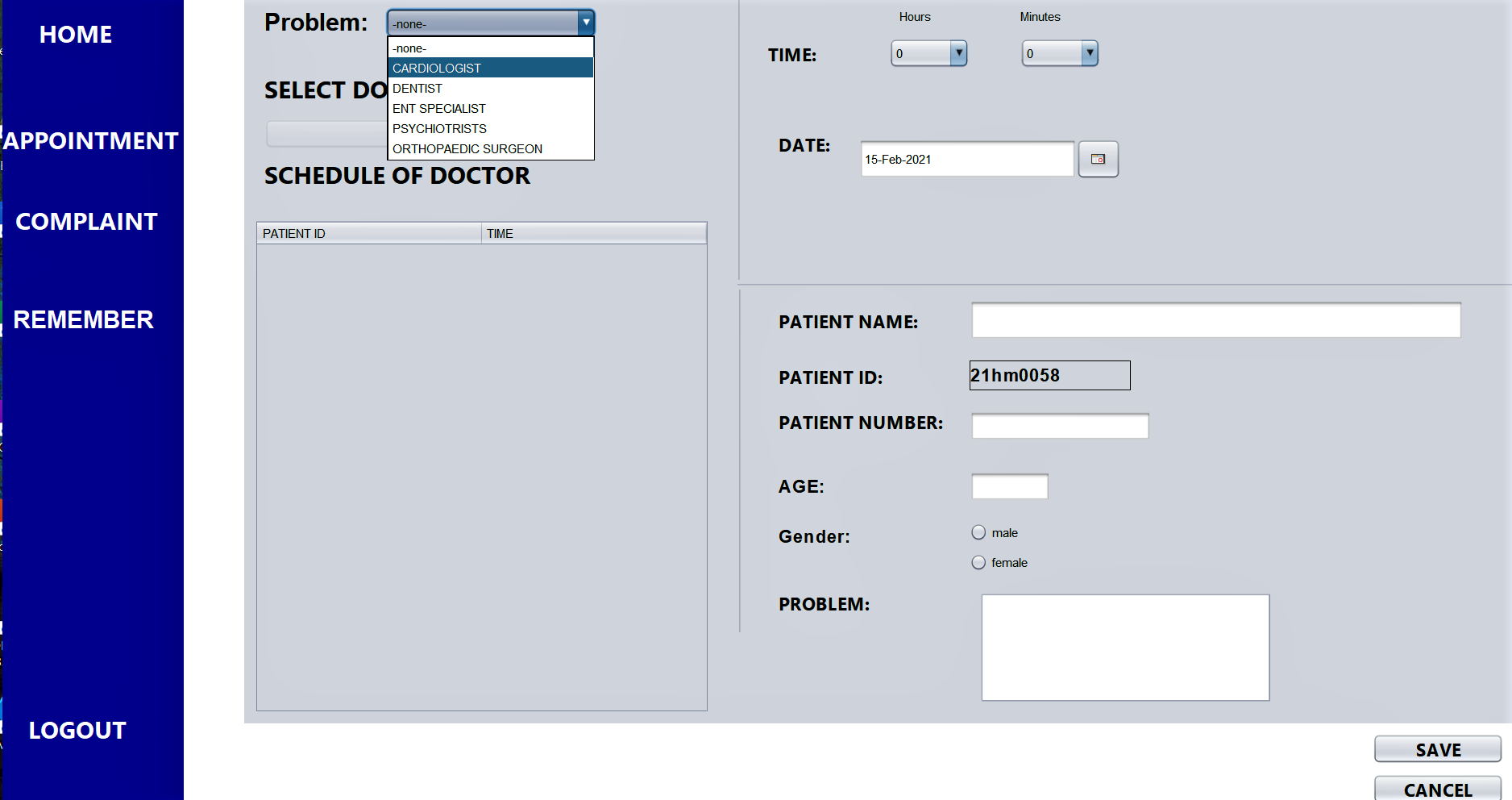
DOCTOR INFO:

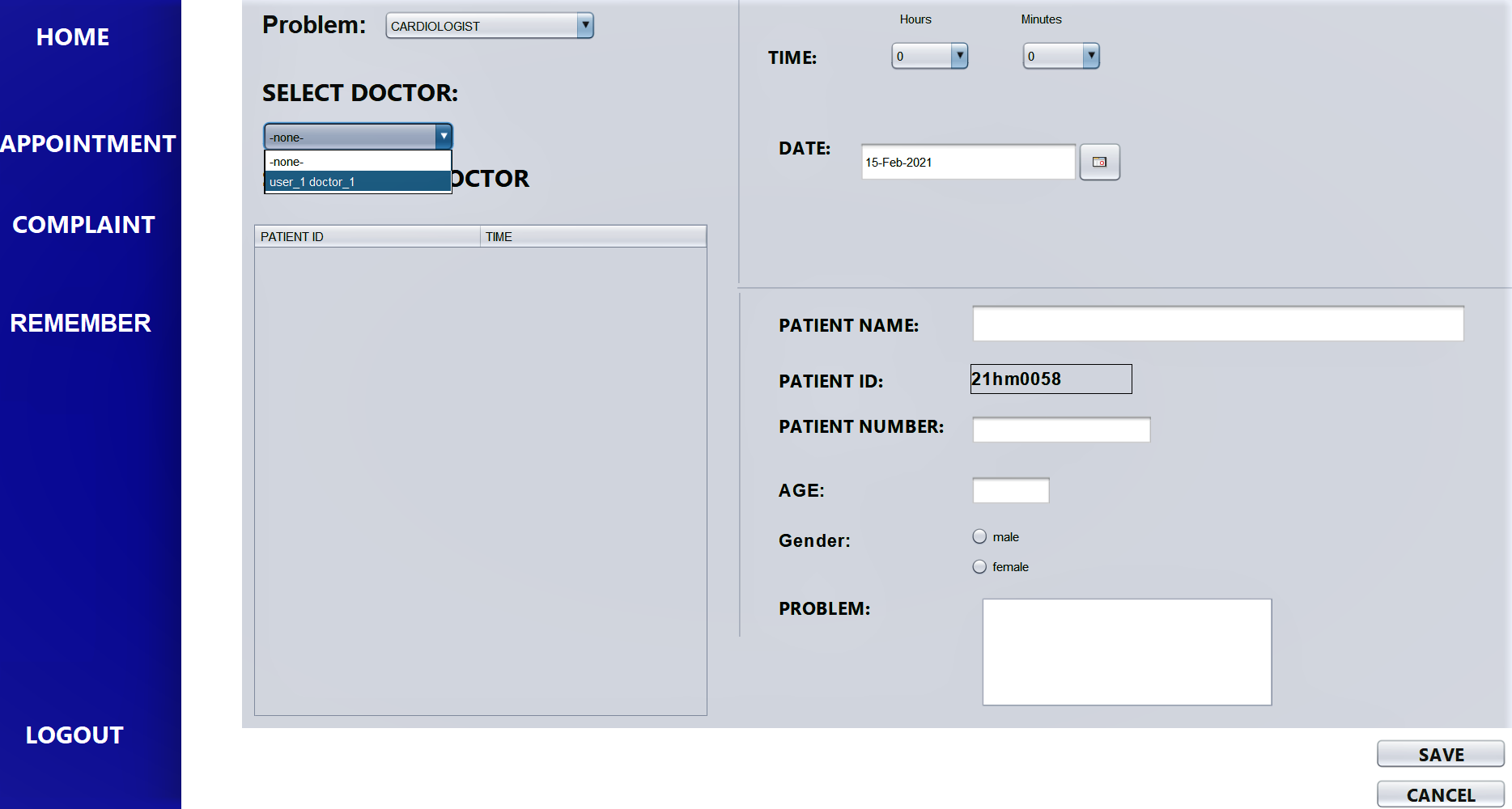




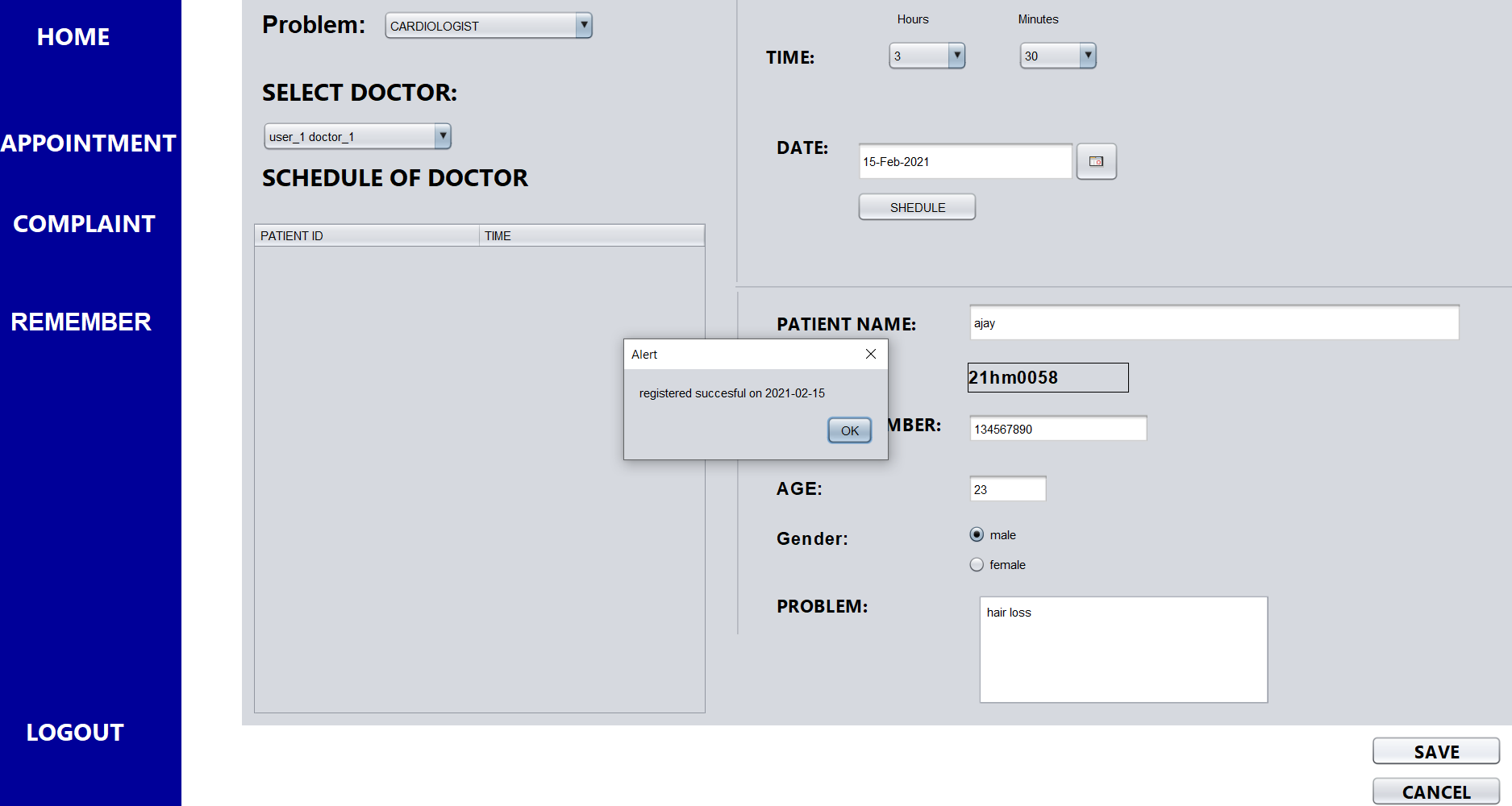
* We need to select a particular doctor to see their statistics.

RECEPTIONIST:

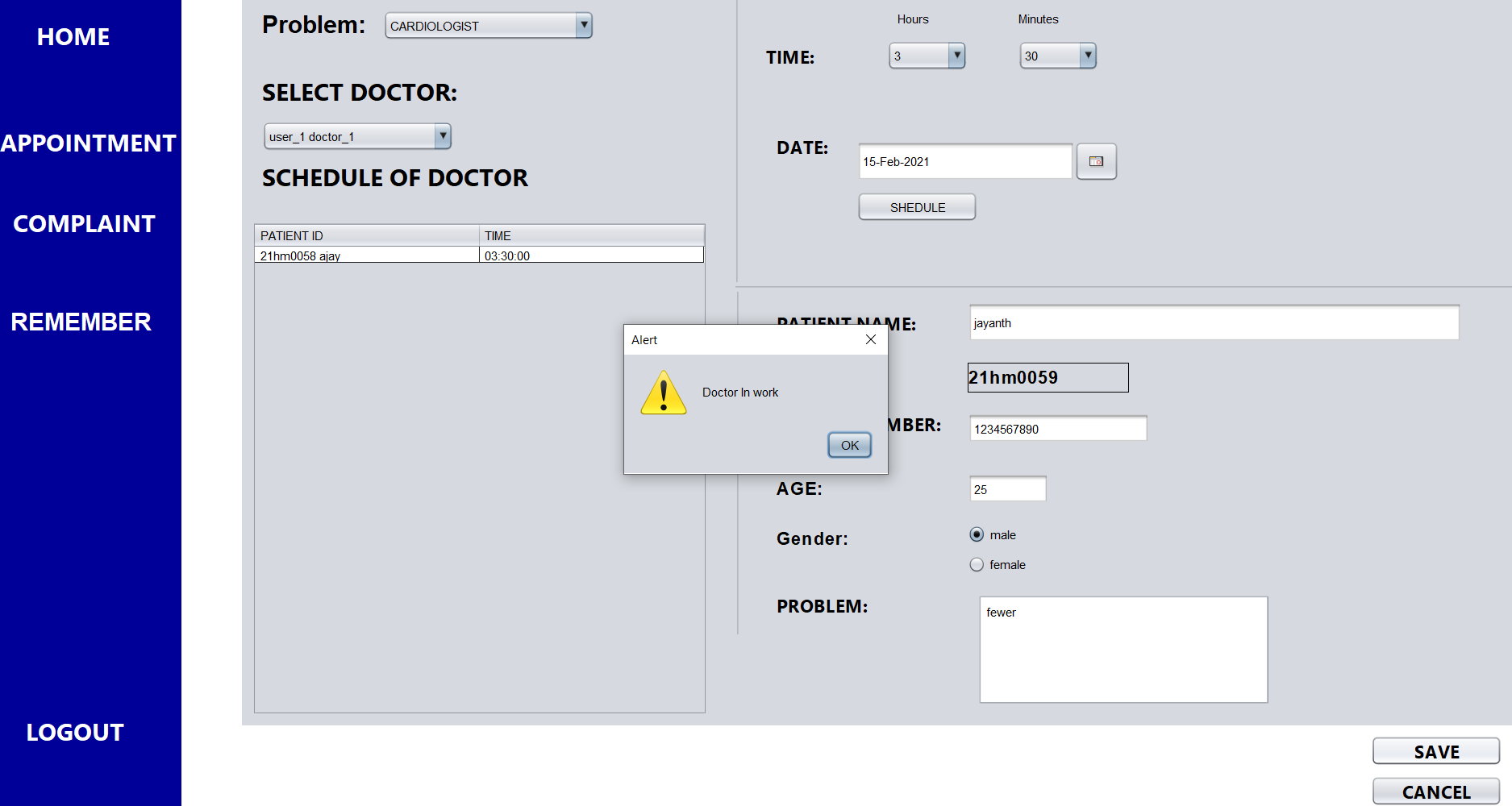




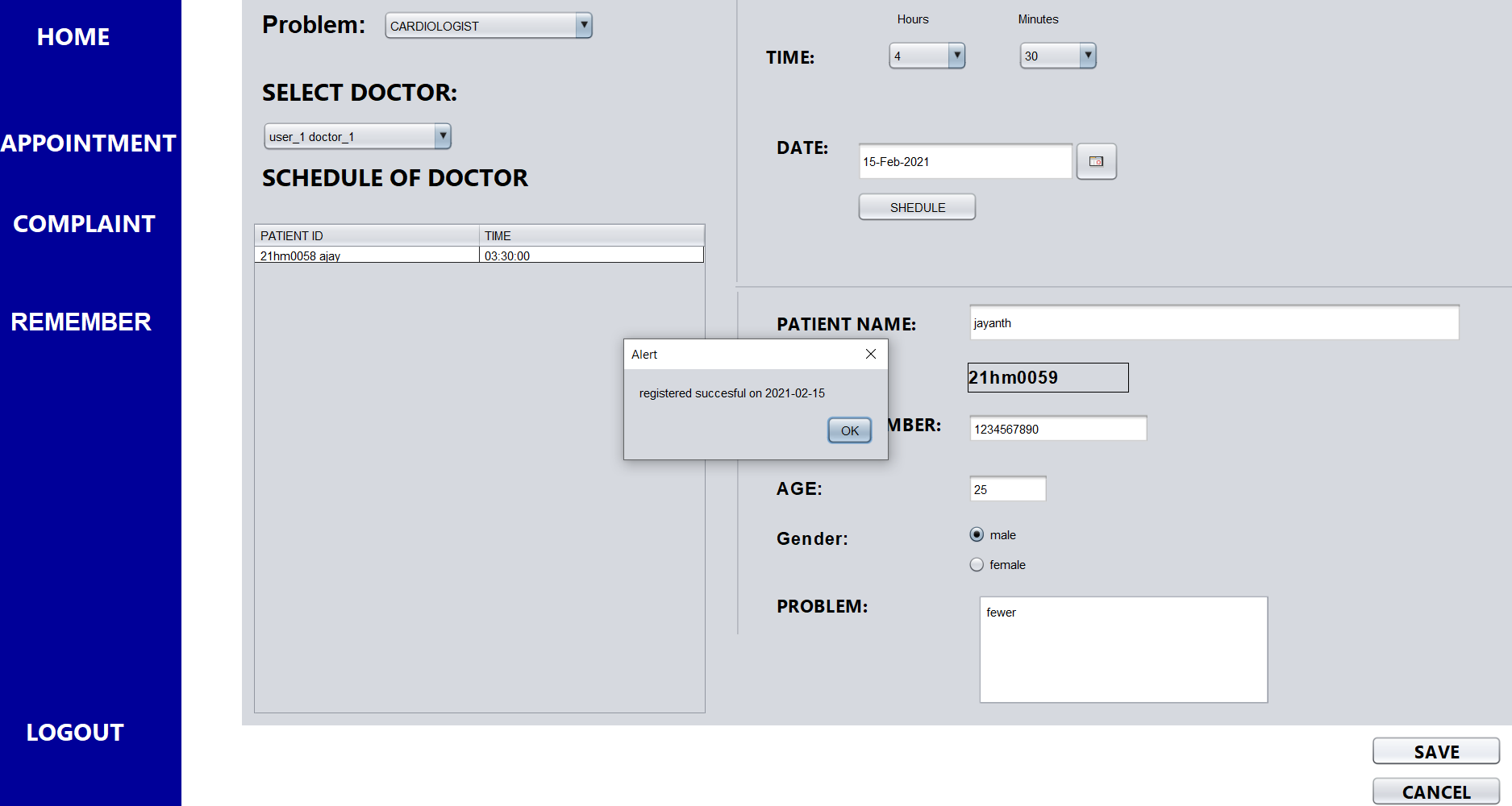
* The receptionist need to select the particular doctor so that the list of available doctors of particular categeory so that they can select the particular doctor.
* When ever a particular doctor is selected the schedule of the doctor will be shown in the below table
* The patient id will be automatically generated.
* We need to fill every field to save.



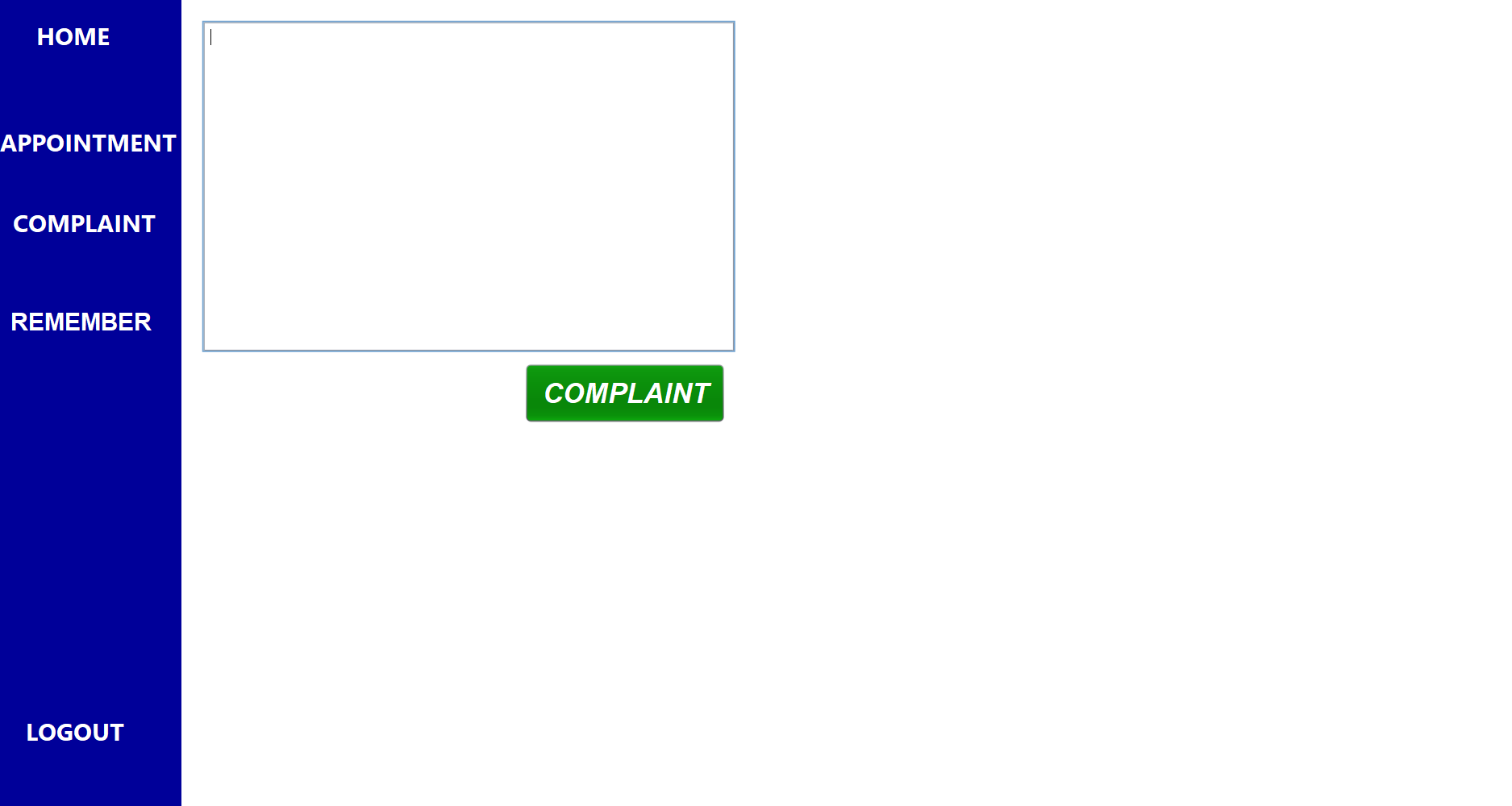
* If we filled every field then it will be shown as above that registration successful on some particular date.
* And it will be saved in the database permenantly.



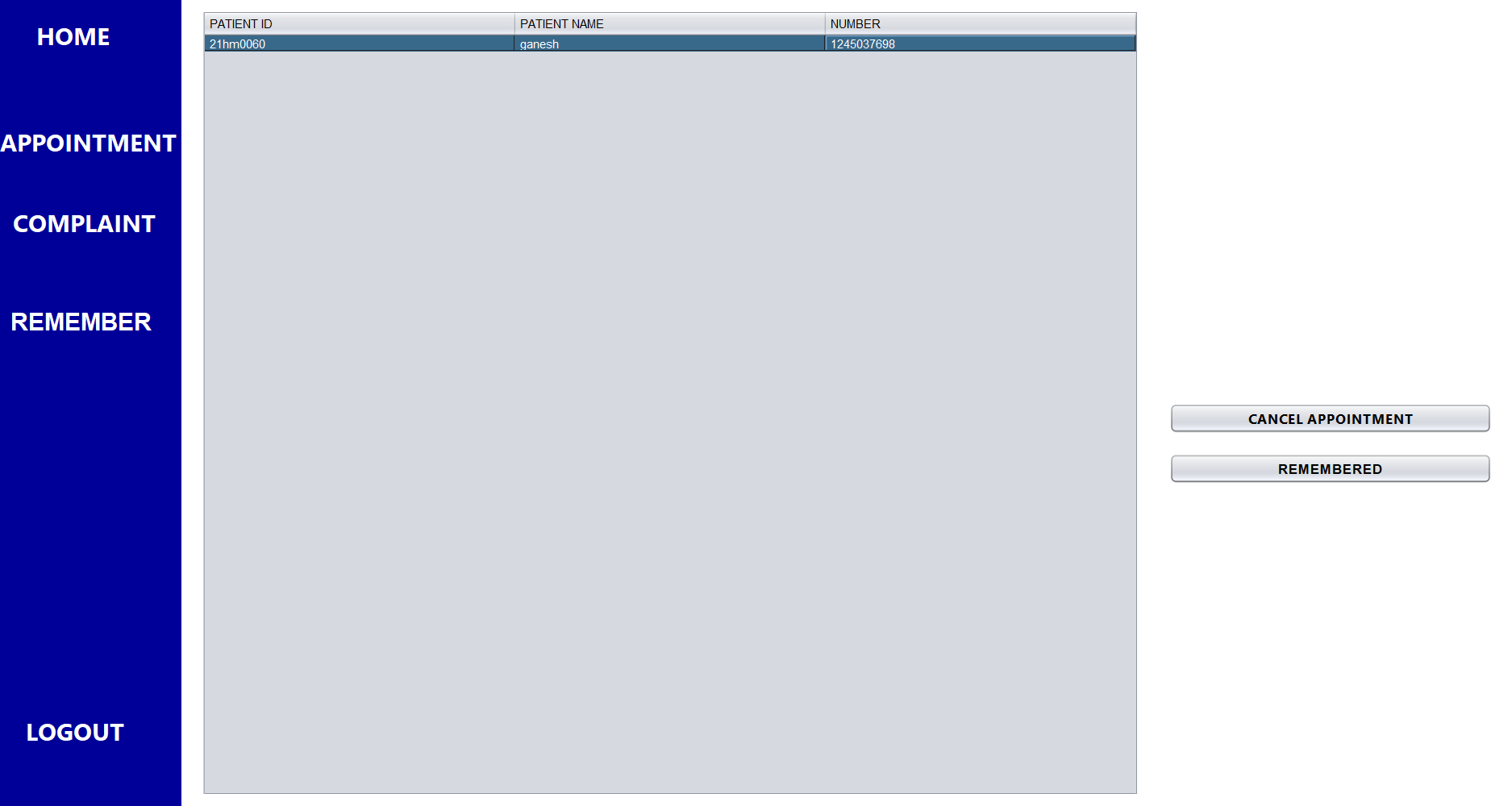
* if we give the work on same time then it will show a warning message doctor in work.



* if there is no work then it will be saved as registered successful on some date.

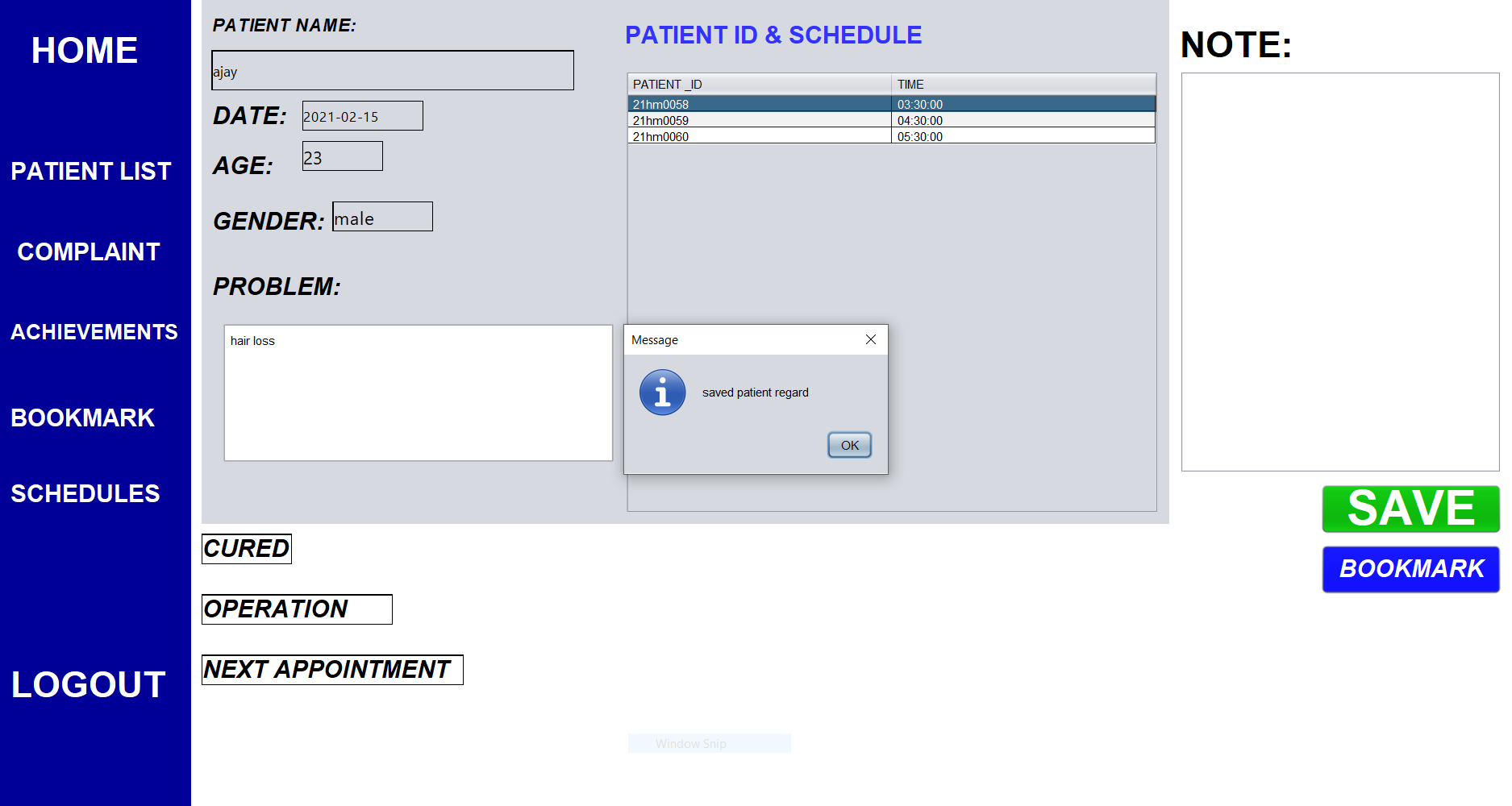


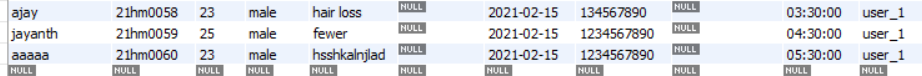
* The complaint box used to give complaint to the admin.



* The list of patients who have appointment tomorrow will be diaplayed in remembered page.
* They can cancel the appointment or they can remember them .if they cancel the appointment the details will be removed from the database.
* If they were remembered they just removed from the table. Not from the database

DOCTOR:

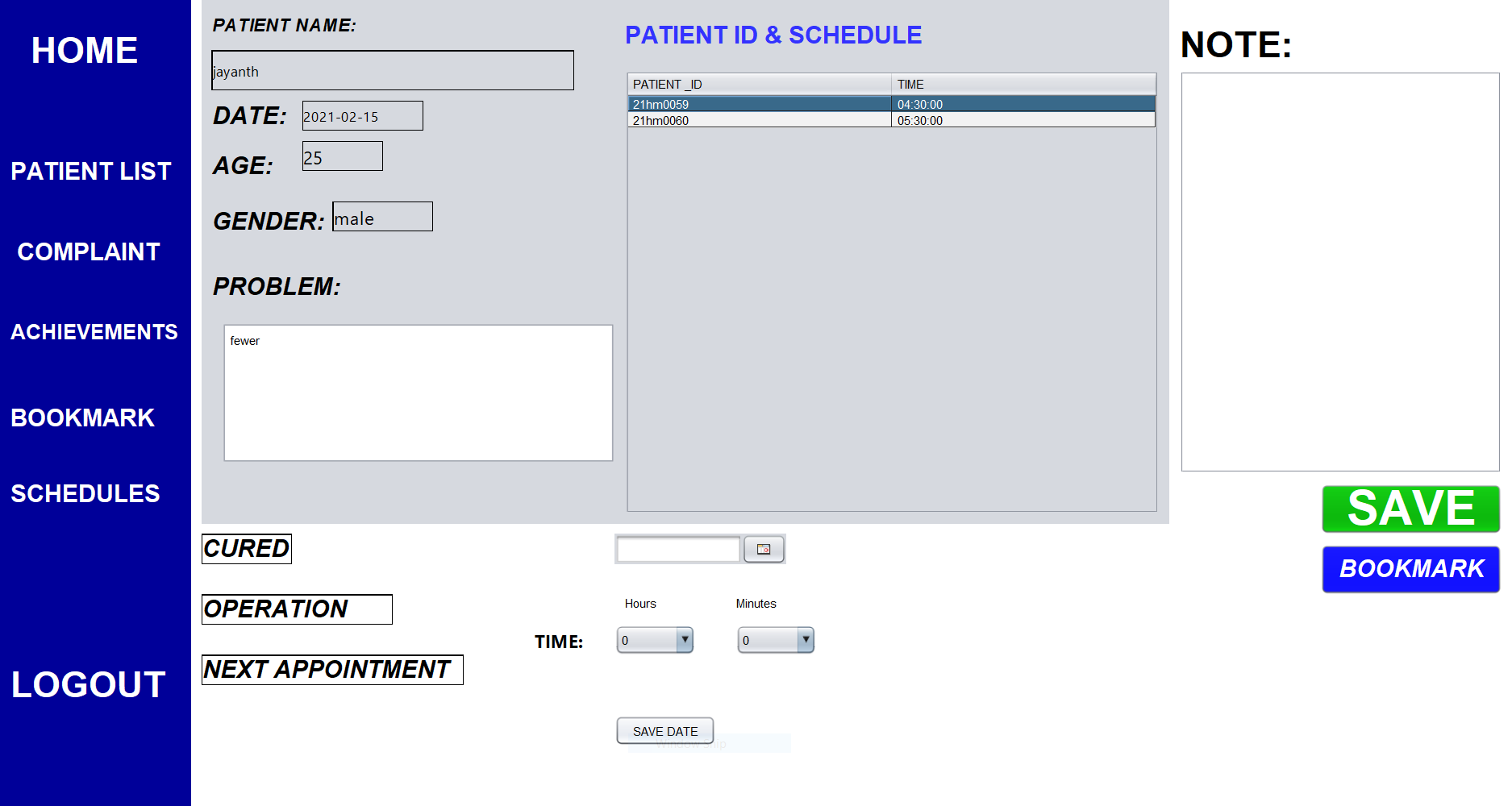


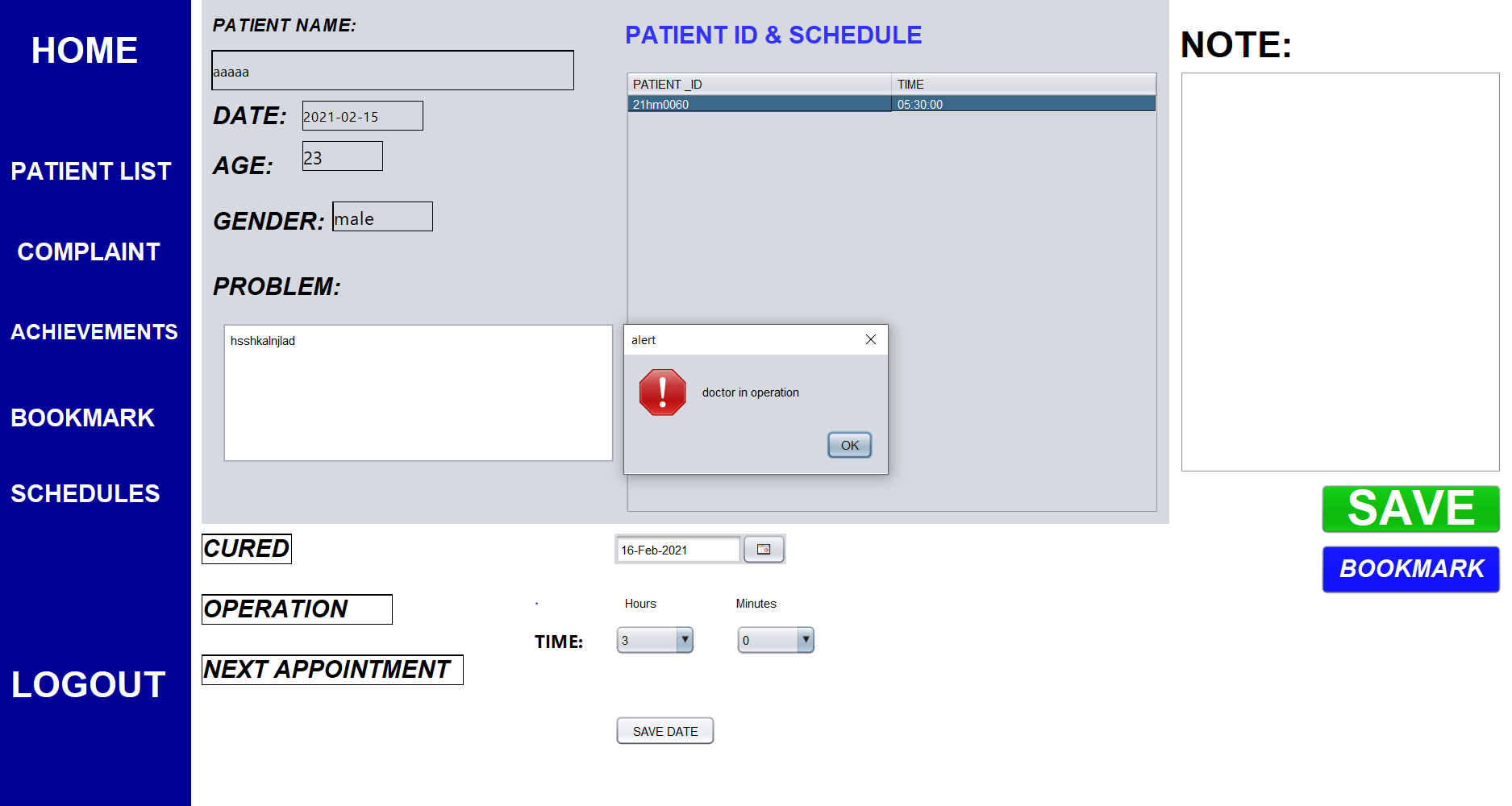


* The list of patients will be displayed to the doctor whenever doctor select a particular patient .
* There will be three options .

1. Cured
2. Operation
3. Next appointment.

* The operation and next appointment enables time and date to save.

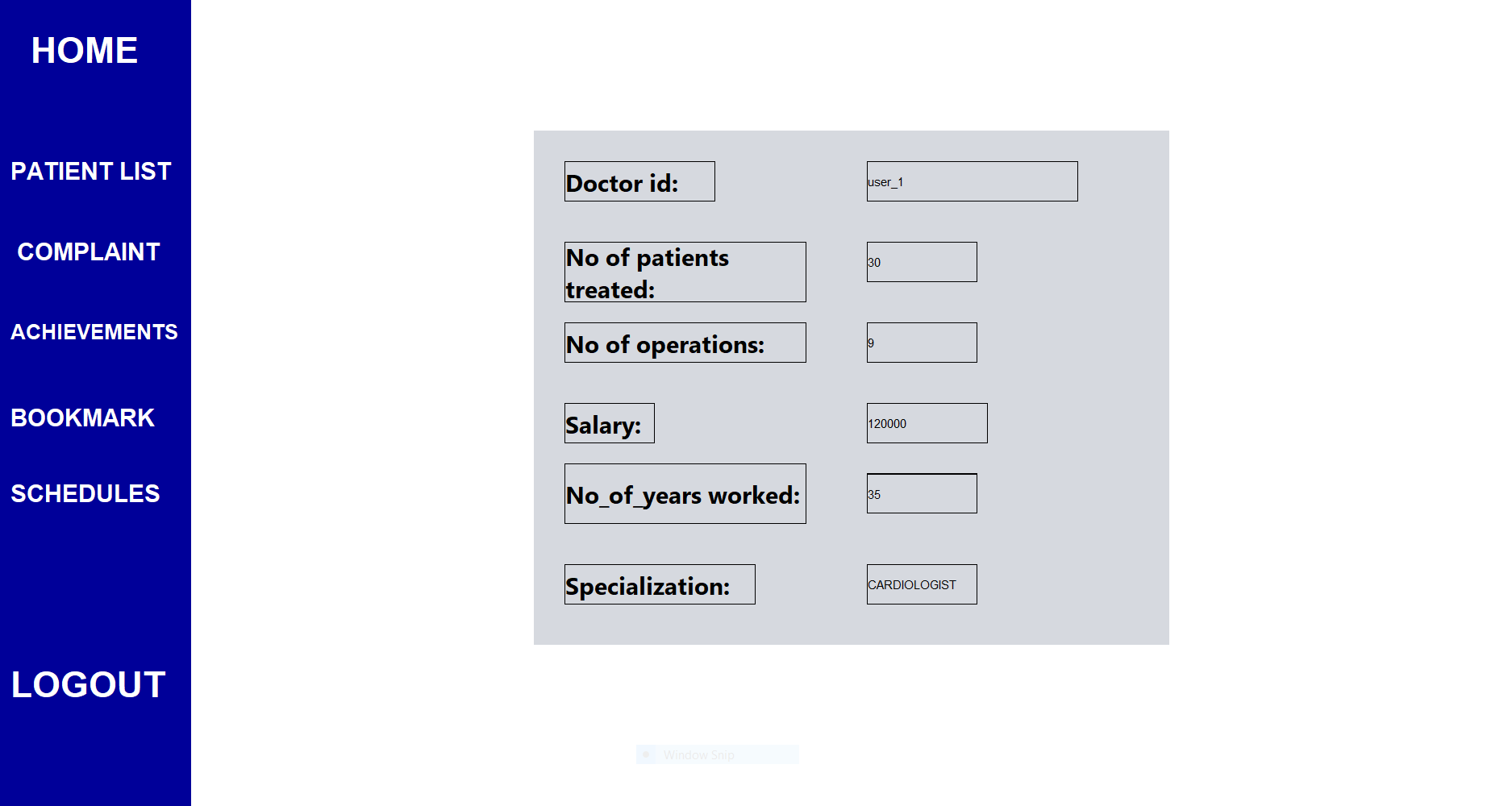




* Here if there is any operation then we need to give on another day.in an assumption doctor do only one operation with no appointment per day.



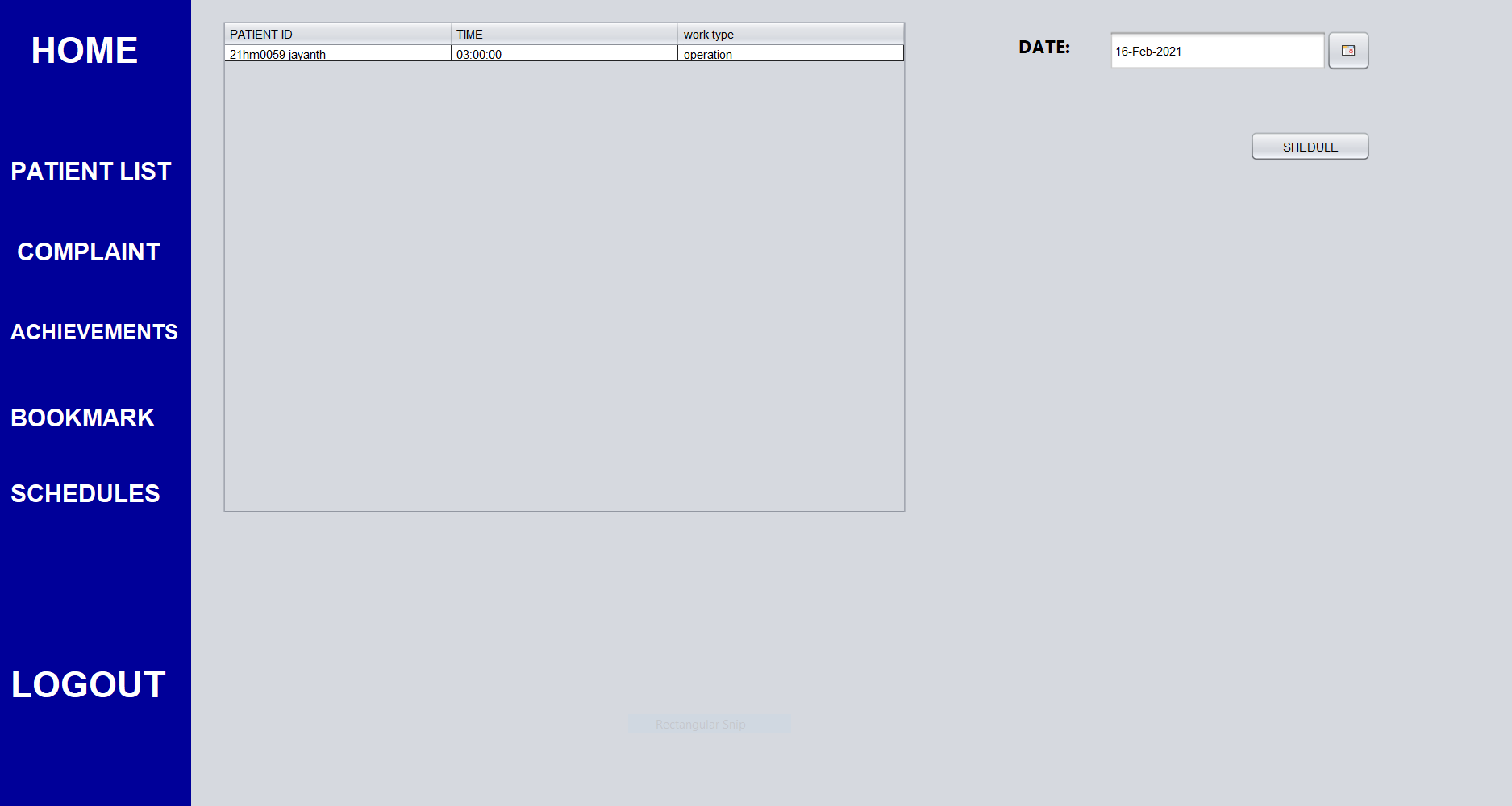
* All the patients regard will be saved in the database permenantly.



* The doctors can see the achievements he had means no of operation he had done no of patients he had treated and so on

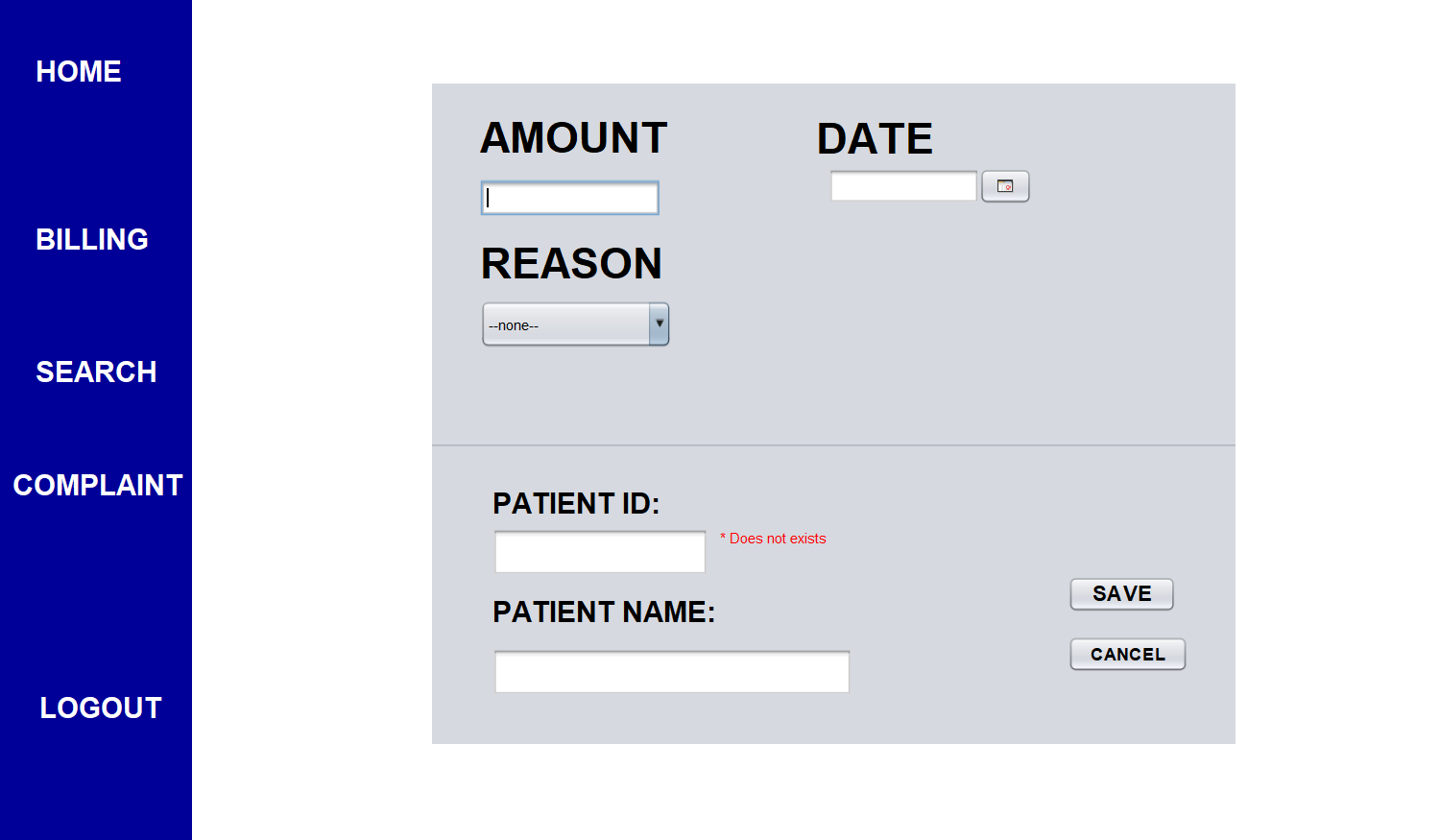


* He can see all his book marks. he can see all the bookmarks by moving front and back with the help of next and previous buttons
* If he want to delete a particular bookmark he can with the help of delete button.

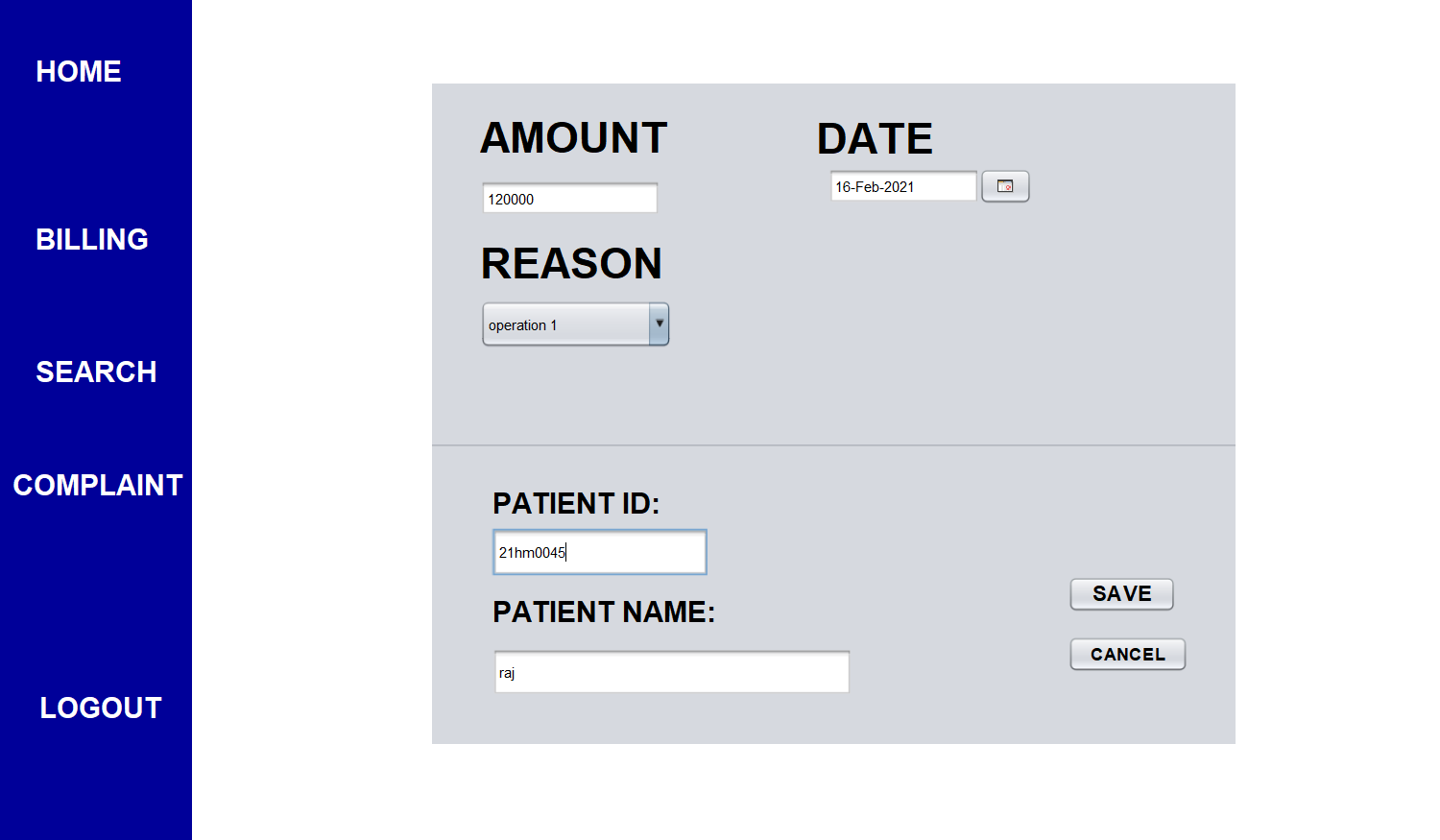


* The schedule of a particular doctor can be seen by his schedule page by selecting a specific date and clicking on schedule button.

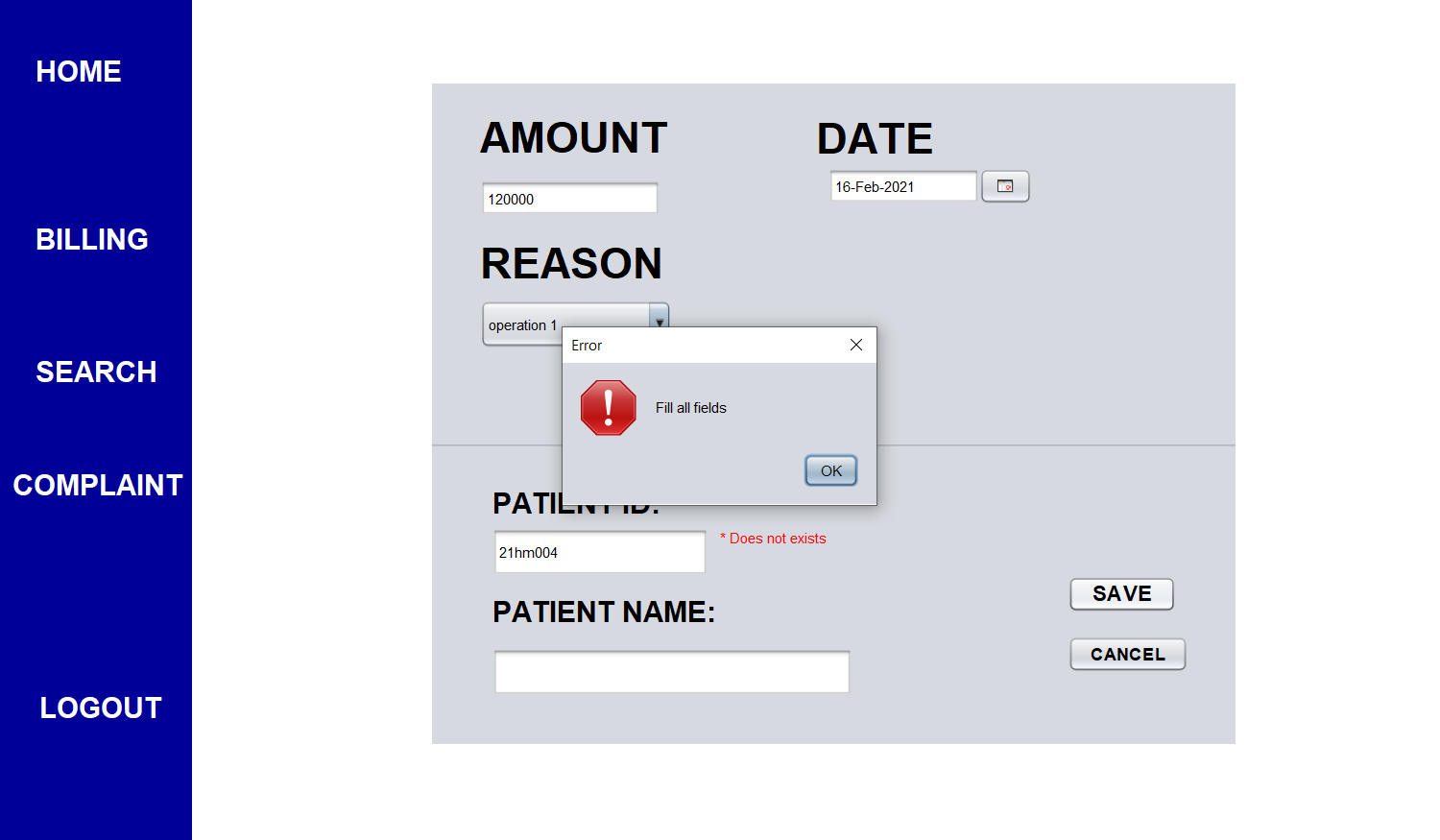
FINANCE:



* Finance page is used to store and search the details of patient regard about how much a patient paid.
* The values must be filled inorder to save in the database.



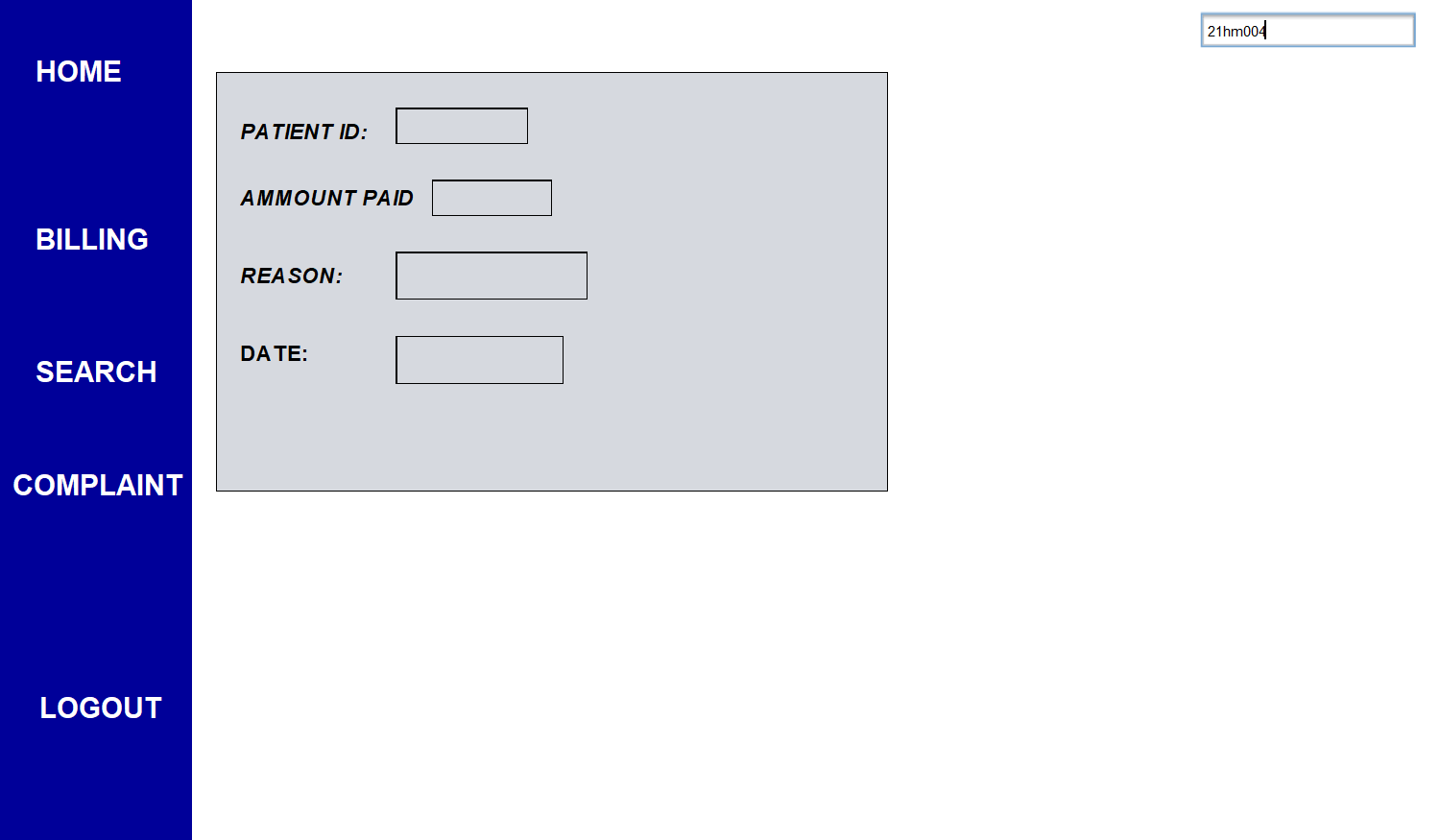
* The name is autodetected by typing patient id that is exciting in the database. else it is shown an error tag.

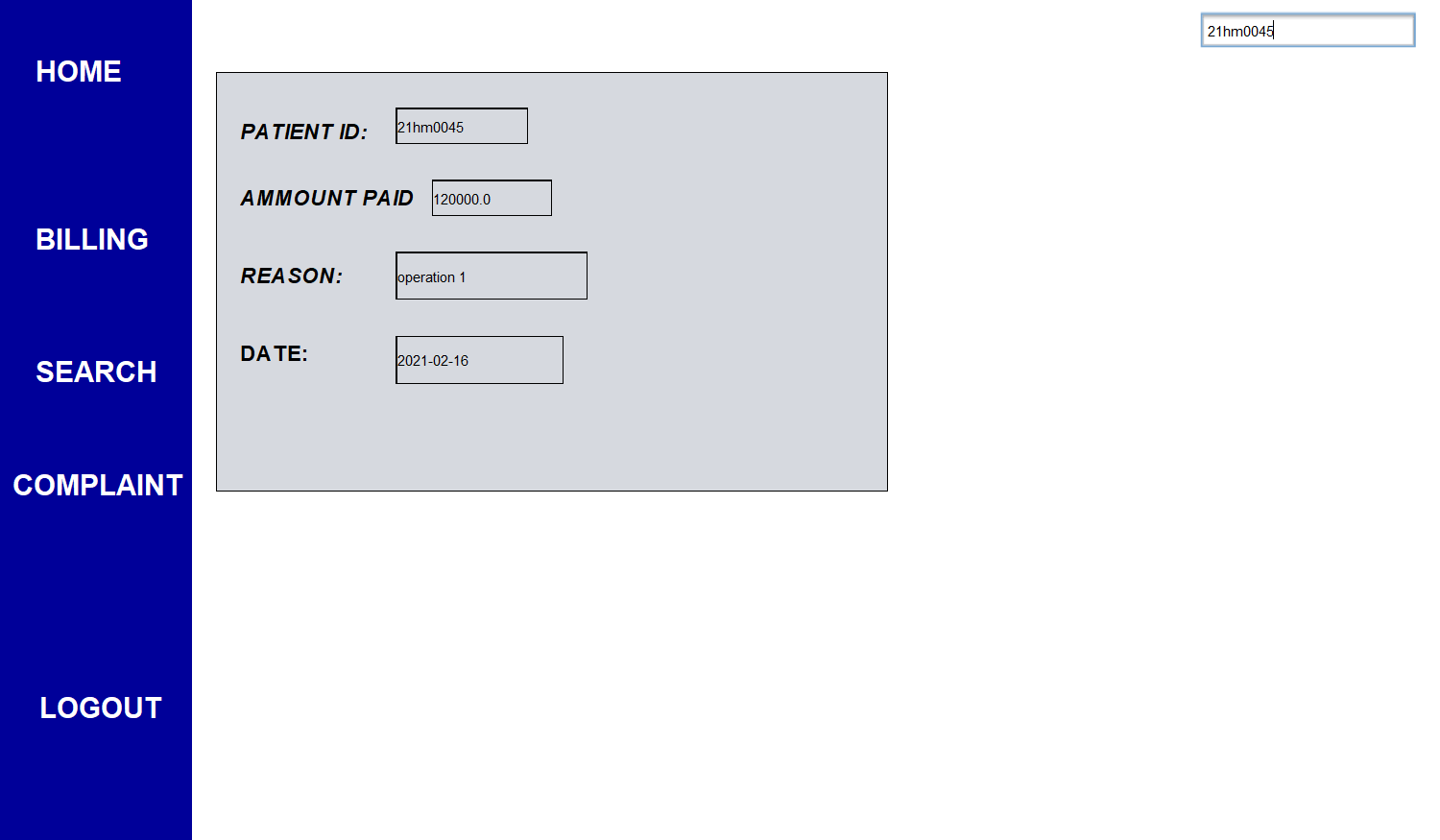


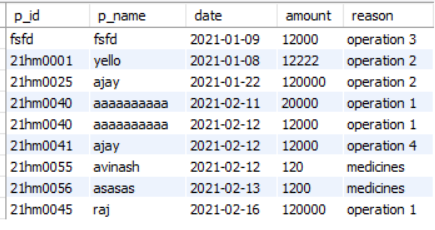
* If we enter an invalid id then name won’t be generated then if we try to save it in the database then it is shown an error message.



* If all the fields are correct then it will be shown stored successfully.



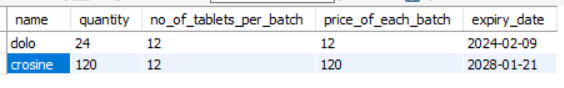


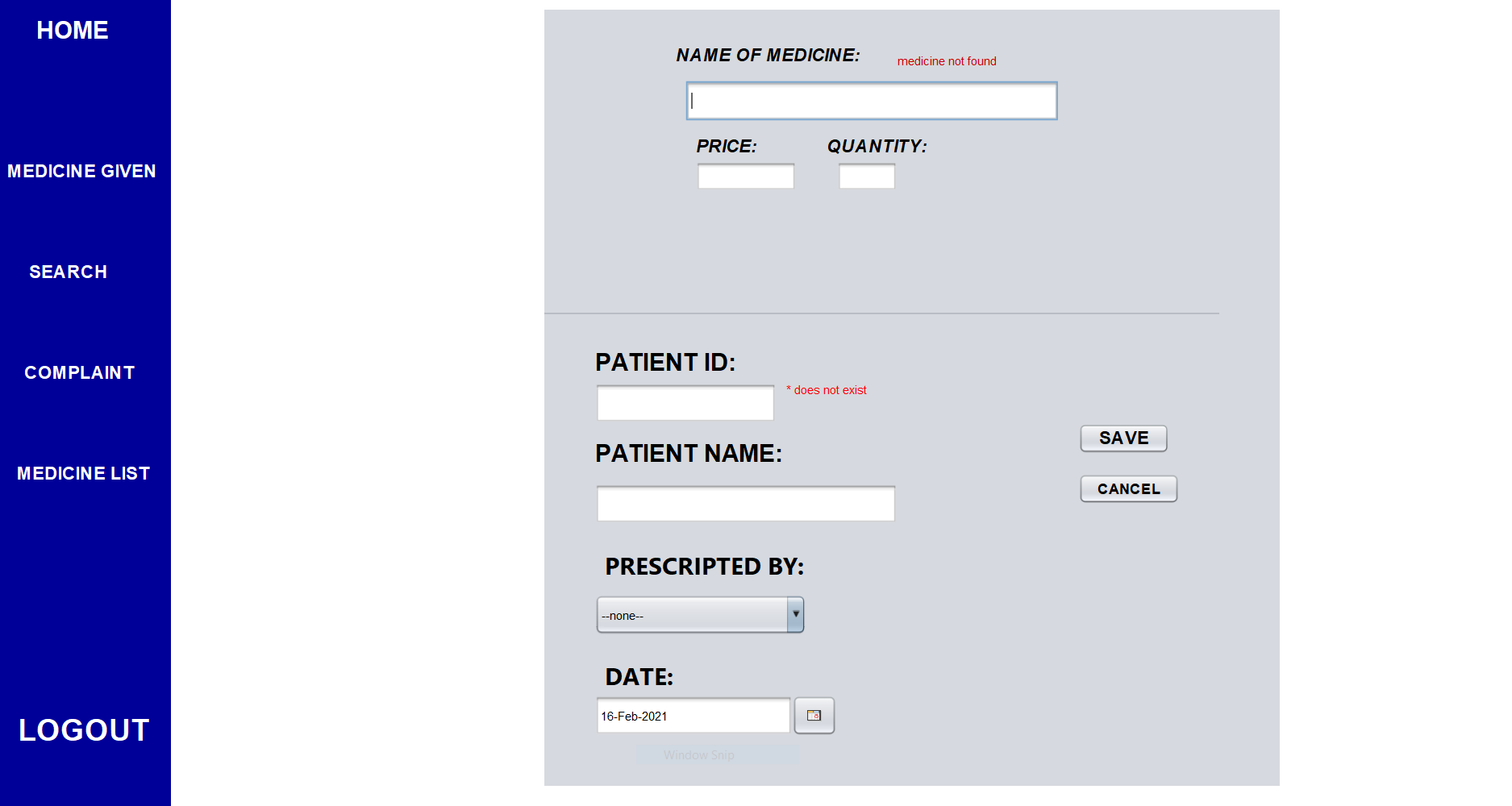


* We can also search for a particular user details as shown in figure.

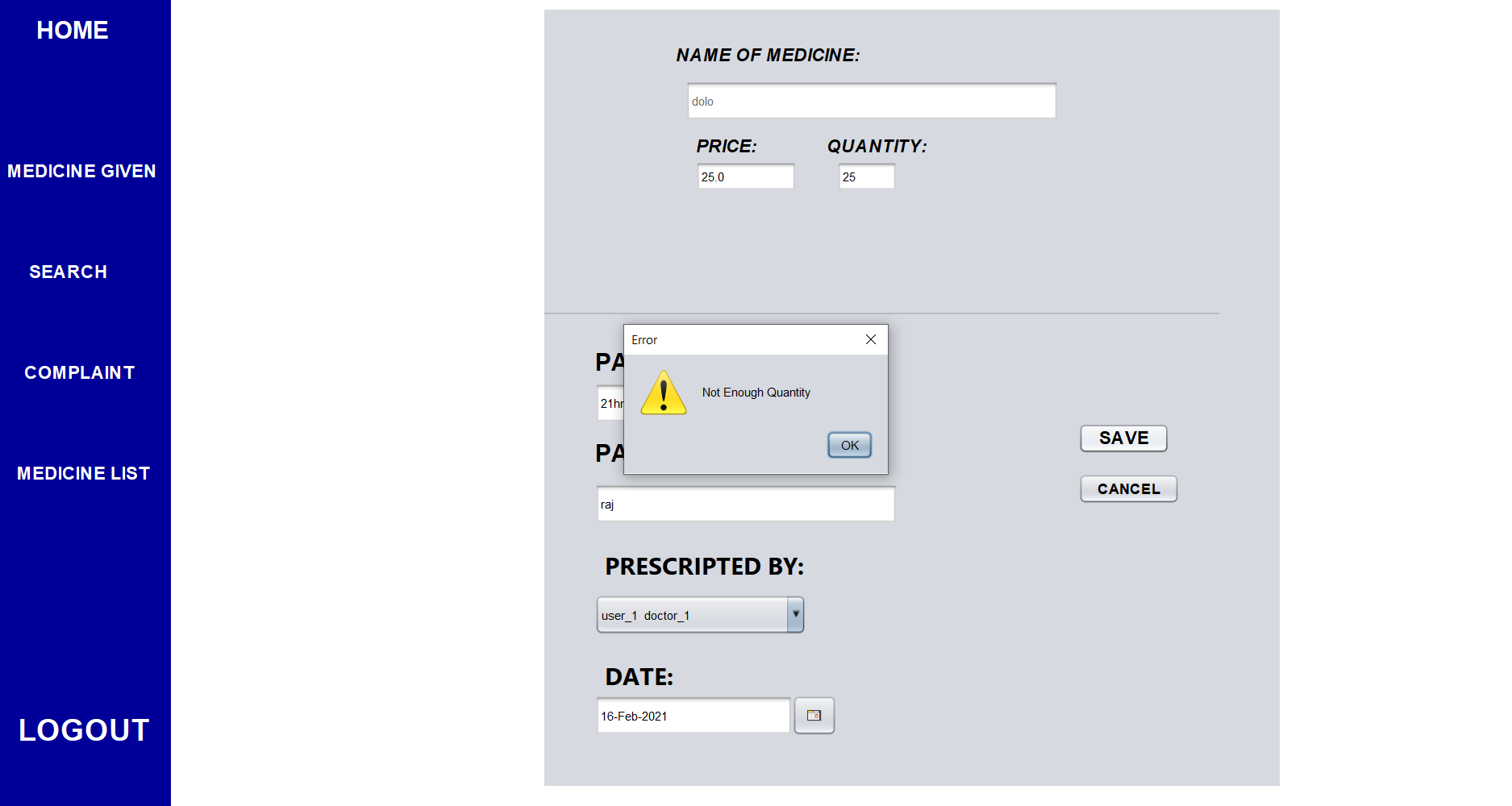
MEDICINE:

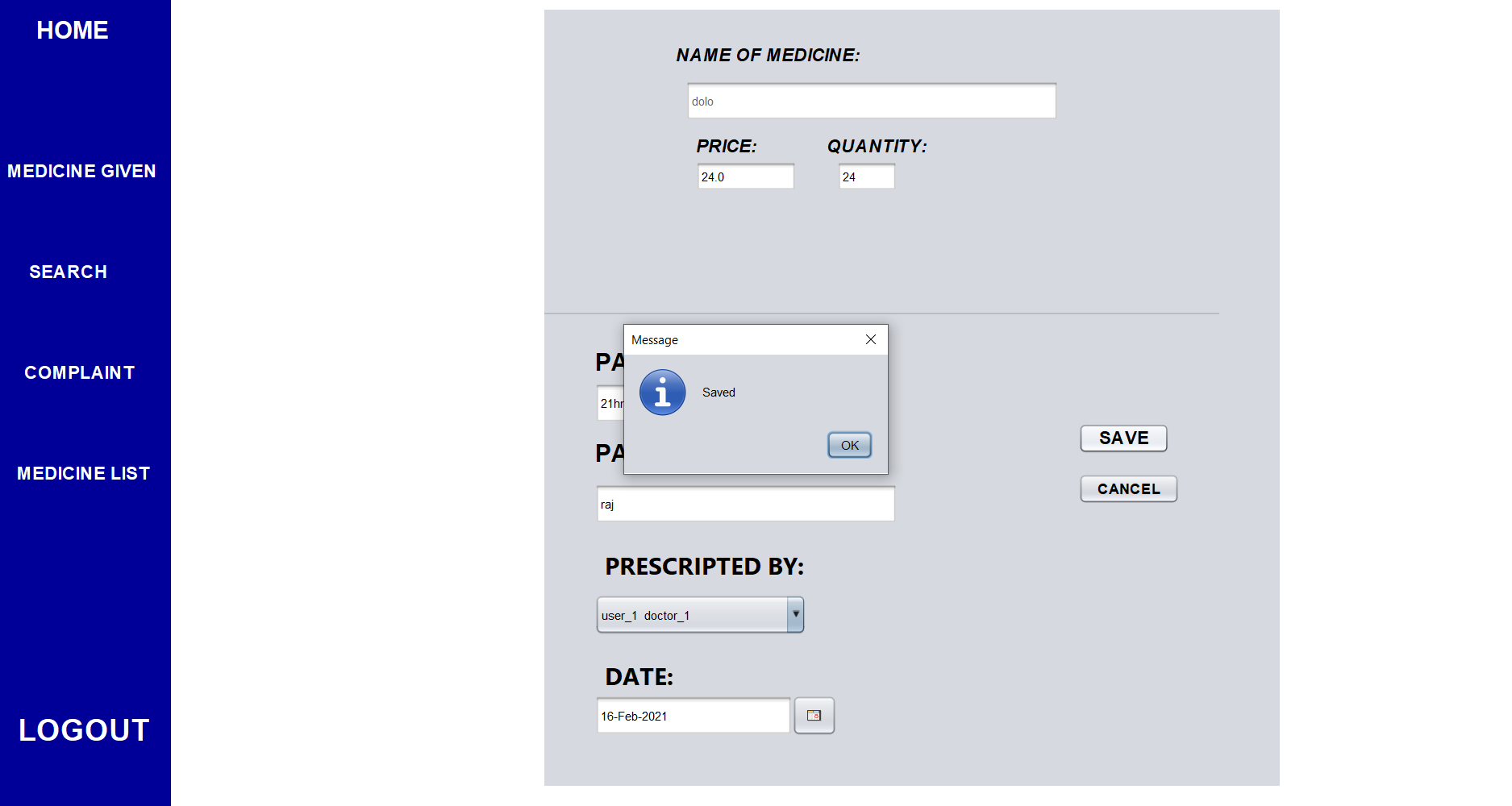
* Generally medicine staff store details of both medicines and as well as students.
* Medicine details required to know the quantity of each medicine and to know the list of medicines that expires within 5 days.
* Patient details are to search about a particular patient regard about medicines purchases.



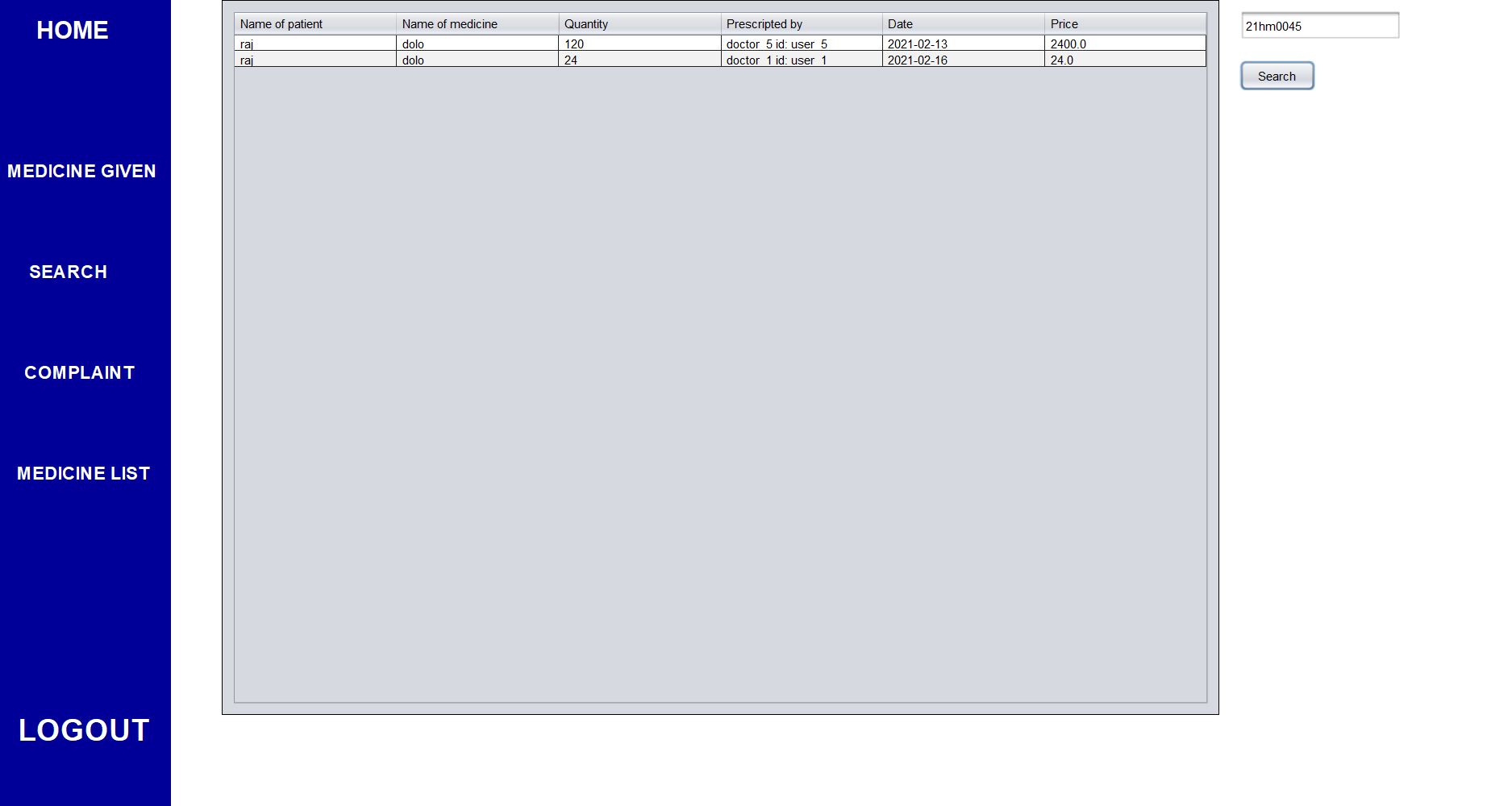


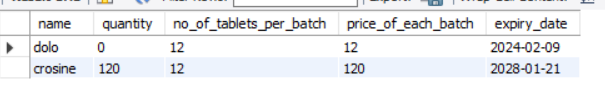
* We need to fill all the field correctly there should be no error tags in order to save data In the database.
* If the medicines quantity is not sufficient then it will show medicines are in sufficient quantity.



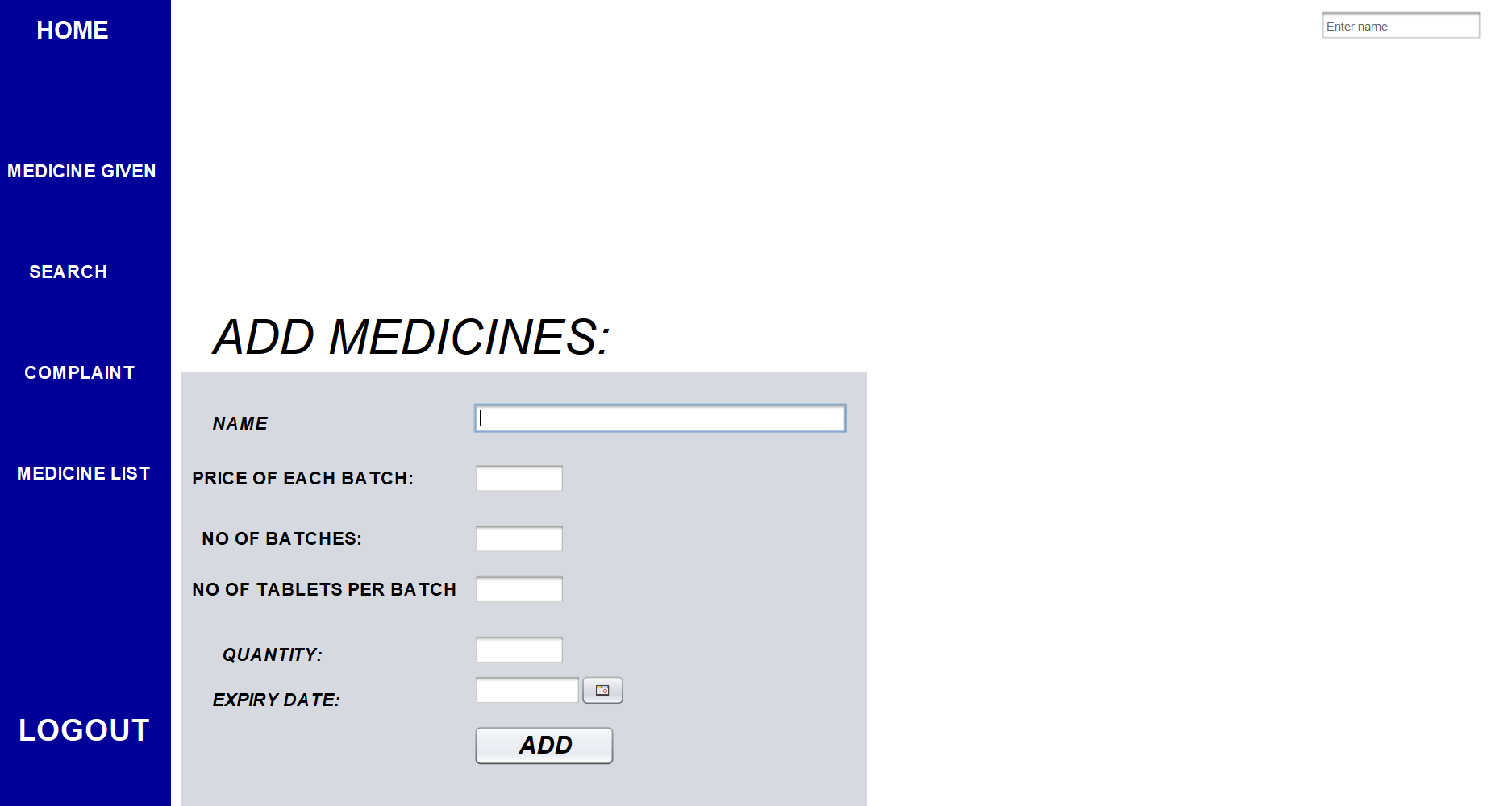


* As quantity is not sufficient so it shown warning when quantity<=available then it will be saved.





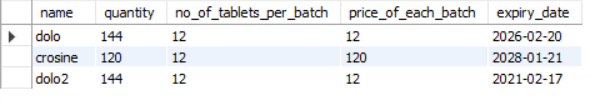
* We can search the details of a particular user also so that all the list of medicines taken by user will be shown.



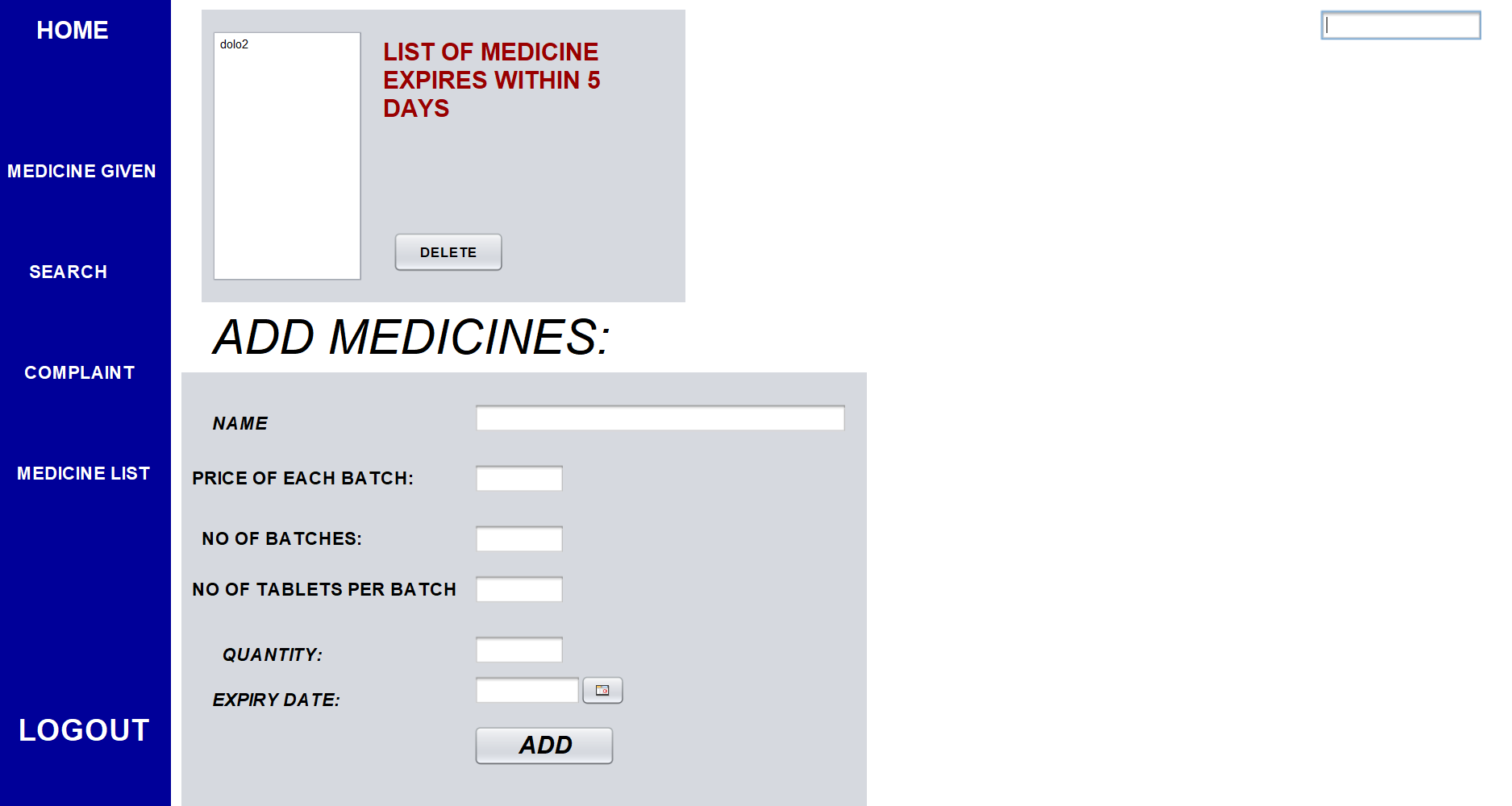


* If the name is same as in the database then the value get updated in the database.
* Else it will be saved in the database

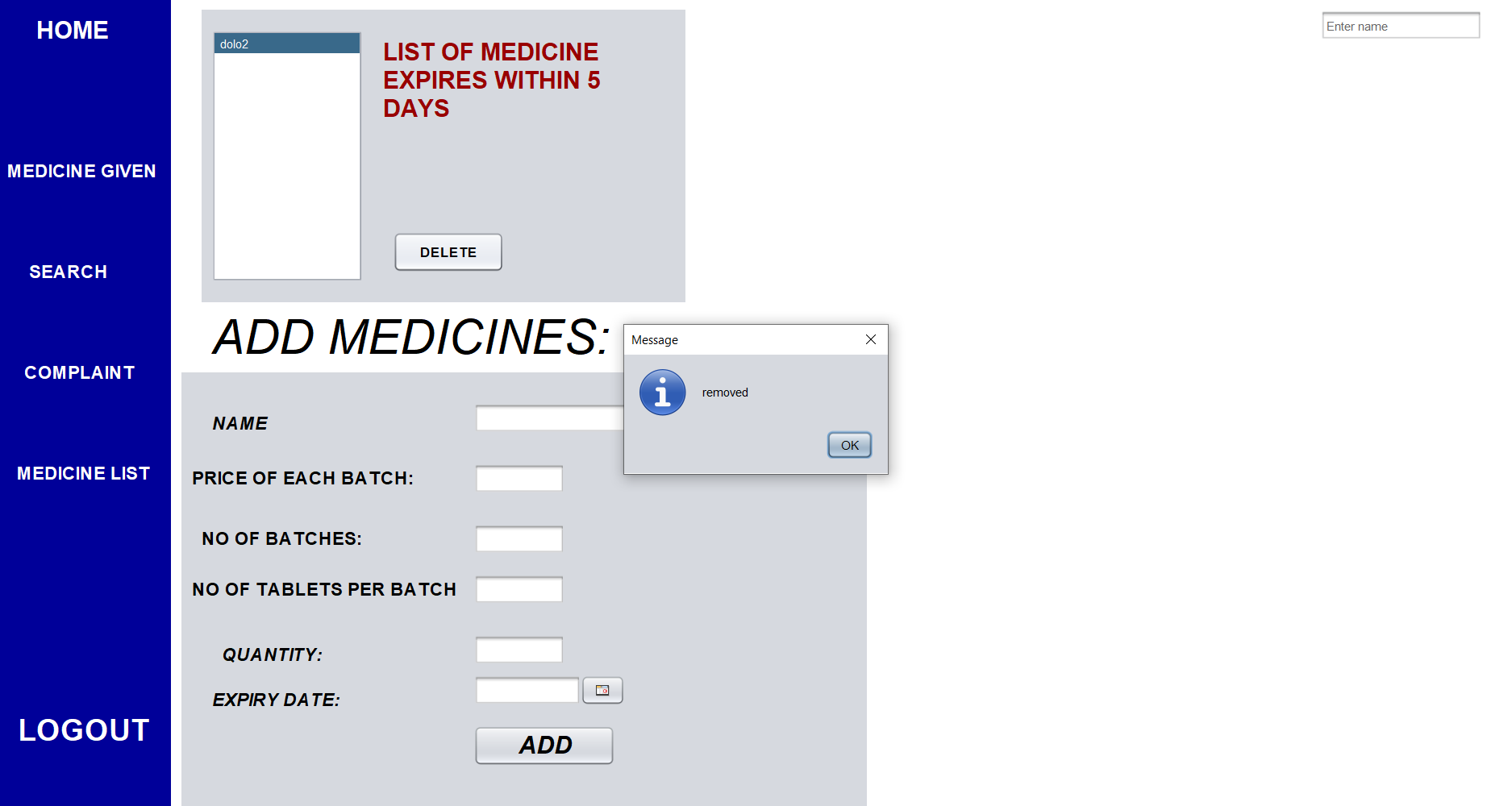


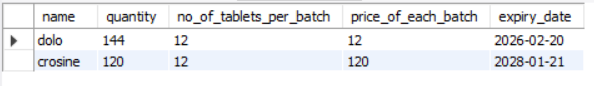


* As dolo name is already there in the database so it will be updated.
* Where as dolo 2 is not available so it is saved.

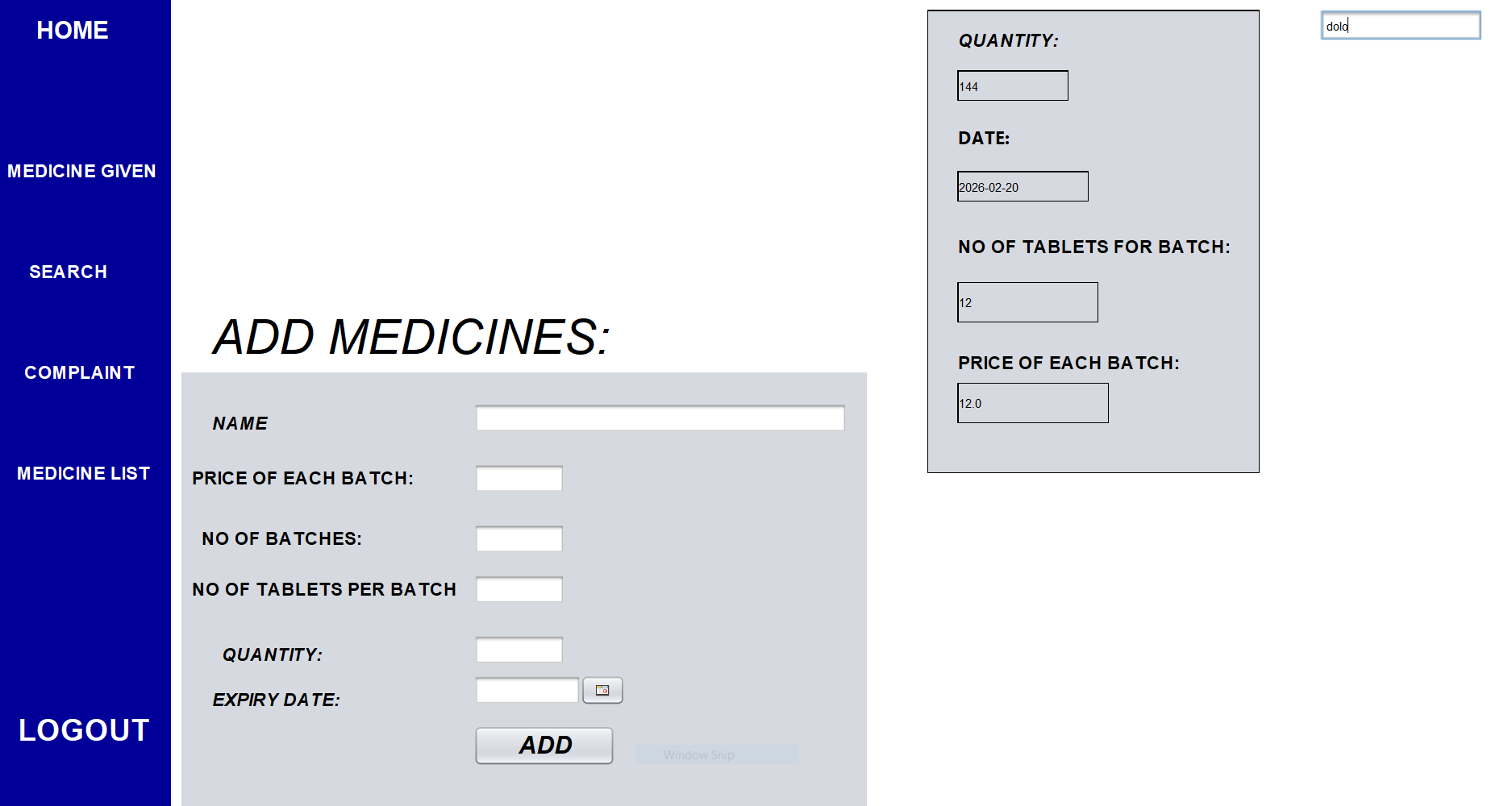


* As dolo2 expiry date is within 2 days it will be shown.





* We can also delete if we want to remove it from the database.



* We can also see details of a particular medicine so that we know the quantity before.

SAMPLE CODE:

LOGIN PAGE:

package mainpackage;

import java.awt.\*;

import java.sql.\*;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

public class Loginpage extends javax.swing.JFrame {

Connection c;

Statement stmt;

ResultSet rs1,rs2,rs;

public Loginpage() {

initComponents();

setIconImage();

try{

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

c=DriverManager.getConnection("jdbc:mysql://localhost:3306/hospital?autoReconnect=true&useSSL=false","root","1234");

stmt=c.createStatement();

}

catch(Exception e)

{

new Register().setVisible(true);

System.exit(0);

}

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

title1 = new javax.swing.JLabel();

title2 = new javax.swing.JLabel();

password = new javax.swing.JPasswordField();

userid = new javax.swing.JTextField();

user\_error = new javax.swing.JLabel();

password\_error = new javax.swing.JLabel();

jCheckBox1 = new javax.swing.JCheckBox();

login = new javax.swing.JButton();

jLabel4 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel1 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

setMinimumSize(new java.awt.Dimension(882, 580));

setResizable(false);

getContentPane().setLayout(null);

title1.setFont(new java.awt.Font("Arial", 1, 48)); // NOI18N

title1.setForeground(new java.awt.Color(102, 153, 255));

title1.setFocusable(true);

title1.setText("HOSPITAL MANAGEMENT");

getContentPane().add(title1);

title1.setBounds(10, 20, 620, 80);

title2.setFont(new java.awt.Font("Arial", 1, 48)); // NOI18N

title2.setForeground(new java.awt.Color(102, 153, 255));

title2.setText("SYSTEM");

getContentPane().add(title2);

title2.setBounds(230, 90, 210, 70);

password.setEchoChar((char)0);

password.setBackground(new java.awt.Color(255, 255, 255));

password.setText("Enter password");

password.setForeground(new java.awt.Color(102, 102, 102));

password.addFocusListener(new java.awt.event.FocusAdapter() {

public void focusGained(java.awt.event.FocusEvent evt) {

passwordFocusGained(evt);

}

public void focusLost(java.awt.event.FocusEvent evt) {

passwordFocusLost(evt);

}

});

password.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

passwordMouseClicked(evt);

}

});

getContentPane().add(password);

password.setBounds(80, 280, 200, 40);

userid.setText("Enter user id");

userid.setForeground(new java.awt.Color(102, 102, 102));

userid.setBackground(new java.awt.Color(255, 255, 255));

userid.addFocusListener(new java.awt.event.FocusAdapter() {

public void focusGained(java.awt.event.FocusEvent evt) {

useridFocusGained(evt);

}

public void focusLost(java.awt.event.FocusEvent evt) {

useridFocusLost(evt);

}

});

userid.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

useridMouseClicked(evt);

}

});

getContentPane().add(userid);

userid.setBounds(80, 210, 200, 40);

user\_error.setFont(new java.awt.Font("Arial", 2, 14)); // NOI18N

user\_error.setForeground(new java.awt.Color(255, 0, 0));

user\_error.setText("enter valid user id..!");

user\_error.setToolTipText("");

getContentPane().add(user\_error);

user\_error.setBounds(80, 250, 200, 20);

user\_error.setVisible(false);

password\_error.setFont(new java.awt.Font("Arial", 2, 14)); // NOI18N

password\_error.setForeground(new java.awt.Color(255, 0, 0));

password\_error.setText("enter a valid password!!");

getContentPane().add(password\_error);

password\_error.setBounds(80, 320, 200, 17);

password\_error.setVisible(false);

jCheckBox1.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N

jCheckBox1.setForeground(new java.awt.Color(0, 0, 0));

jCheckBox1.setText("Show Password");

jCheckBox1.addChangeListener(new javax.swing.event.ChangeListener() {

public void stateChanged(javax.swing.event.ChangeEvent evt) {

jCheckBox1StateChanged(evt);

}

});

getContentPane().add(jCheckBox1);

jCheckBox1.setBounds(80, 340, 160, 21);

login.setBackground(new java.awt.Color(51, 255, 0));

login.setFont(new java.awt.Font("Arial", 1, 14)); // NOI18N

login.setForeground(new java.awt.Color(255, 255, 255));

login.setText("LOGIN");

login.setBorder(null);

login.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

loginMouseClicked(evt);

}

});

getContentPane().add(login);

login.setBounds(80, 370, 200, 30);

jLabel4.setFont(new java.awt.Font("Arial", 1, 14)); // NOI18N

jLabel4.setForeground(new java.awt.Color(51, 51, 255));

jLabel4.setText("New Admin ?? ");

getContentPane().add(jLabel4);

jLabel4.setBounds(80, 400, 120, 20);

jLabel2.setFont(new java.awt.Font("Arial", 1, 14)); // NOI18N

jLabel2.setForeground(new java.awt.Color(51, 51, 255));

jLabel2.setText("<html><u>Register</u></html>");

jLabel2.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

jLabel2MouseClicked(evt);

}

public void mouseEntered(java.awt.event.MouseEvent evt) {

jLabel2MouseEntered(evt);

}

public void mouseExited(java.awt.event.MouseEvent evt) {

jLabel2MouseExited(evt);

}

});

getContentPane().add(jLabel2);

jLabel2.setBounds(190, 400, 70, 20);

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/mainpackage/iStock-949812160.jpg"))); // NOI18N

getContentPane().add(jLabel1);

jLabel1.setBounds(0, 0, 870, 560);

pack();

setLocationRelativeTo(null);

}// </editor-fold>

private void loginMouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

String field1="0",Password1,id1;

String password2="0",id2="0";

Password1=new String(password.getPassword());

id1=userid.getText();

if(id1.equals("Enter user id")==true)

user\_error.setVisible(true);

if(Password1.equals("Enter password")==true)

password\_error.setVisible(true);

if(id1.equals("Enter user id")==false&&Password1.equals("Enter password")==false)

{

int count=0;

try{

rs=stmt.executeQuery("select count(user\_id) from user where user\_id='"+id1+"'");

if(rs.next())

count=Integer.parseInt(rs.getString(1));

}

catch(Exception er)

{

System.out.println(er+"1");

}

if(count==0)

{

user\_error.setVisible(true);

password\_error.setVisible(true);

}

else

{

try{

rs=stmt.executeQuery("select \* from user where user\_id='"+id1+"'");

if(rs.next()==true)

{

field1=rs.getString("field");

password2=rs.getString("password");

id2=rs.getString("user\_id");

}

}

catch(Exception e)

{

System.out.println(e+"2");

}

if(id1.equalsIgnoreCase(id2)==false)

user\_error.setVisible(true);

if(Password1.equals(password2)==false)

password\_error.setVisible(true);

else

{

if(field1.equalsIgnoreCase("doctor")==true)

{

new doctor(userid.getText()).setVisible(true);

dispose();

try{

}

catch(Exception e)

{

System.out.println(e);

}

}

if(field1.equalsIgnoreCase("Admin")==true)

{

new Admin().setVisible(true);

dispose();

try{

}

catch(Exception e)

{

System.out.println(e);

}

}

if(field1.equalsIgnoreCase("staff\_finance")==true)

{

new staff\_finance().setVisible(true);

dispose();

try{

}

catch(Exception e)

{

System.out.println(e);

}

}

if(field1.equalsIgnoreCase("staff\_reciopnist")==true)

{

new staff\_reciopnist().setVisible(true);

dispose();

try{

}

catch(Exception e)

{

System.out.println(e);

}

}

if(field1.equalsIgnoreCase("staff\_medicine")==true)

{

new staff\_medicine().setVisible(true);

dispose();

try{

}

catch(Exception e)

{

System.out.println(e);

}

}

}

}

}

}

private void useridFocusGained(java.awt.event.FocusEvent evt) {

// TODO add your handling code here:

if(userid.getText().equals("Enter user id")==true)

{

userid.setText("");

userid.setForeground(new java.awt.Color(0,0,0));

}

}

private void useridFocusLost(java.awt.event.FocusEvent evt) {

// TODO add your handling code here:

if(userid.getText().equals("")==true)

{

userid.setText("Enter user id");

userid.setForeground(new java.awt.Color(102,102,102));

}

}

private void jCheckBox1StateChanged(javax.swing.event.ChangeEvent evt) {

// TODO add your handling code here:

if(jCheckBox1.isSelected()==true)

password.setEchoChar((char)0);

if(jCheckBox1.isSelected()==false&&(new String(password.getPassword())).equals("Enter password")==false)

password.setEchoChar('\u26AB');

}

private void passwordFocusGained(java.awt.event.FocusEvent evt) {

// TODO add your handling code here:

if((new String(password.getPassword())).equals("Enter password")==true&&jCheckBox1.isSelected()==false)

{

password.setText("");

password.setEchoChar('\u26AB');

password.setForeground(new java.awt.Color(0,0,0));

}

if((new String(password.getPassword())).equals("Enter password")==true&&jCheckBox1.isSelected()==true)

{

password.setText("");

password.setEchoChar((char)0);

password.setForeground(new java.awt.Color(0,0,0));

}

}

private void passwordFocusLost(java.awt.event.FocusEvent evt) {

// TODO add your handling code here:

if((new String(password.getPassword())).equals("")==true)

{

password.setText("Enter password");

password.setEchoChar((char)0);

password.setForeground(new java.awt.Color(102,102,102));

}

}

private void jLabel2MouseEntered(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

jLabel2.setForeground(new java.awt.Color(204,0,0));

}

private void jLabel2MouseExited(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

jLabel2.setForeground(new java.awt.Color(51,51,255));

}

private void jLabel2MouseClicked(java.awt.event.MouseEvent evt) {

try {

// TODO add your handling code here:

rs1=stmt.executeQuery("select count(\*) from user");

int i=0,j=0;

if(rs1.next())

i=Integer.parseInt(rs1.getString(1));

rs2=stmt.executeQuery("select count(\*) from info");

if(rs2.next())

j=Integer.parseInt(rs2.getString(1));

if(i==j)

new Register().setVisible(true);

else

JOptionPane.showMessageDialog(this,"Admin Already Exist", "Error", JOptionPane.ERROR\_MESSAGE);

} catch (SQLException ex) {

Logger.getLogger(Loginpage.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void useridMouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

user\_error.setVisible(false);

}

private void passwordMouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

password\_error.setVisible(false);

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Loginpage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Loginpage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Loginpage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Loginpage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new Loginpage().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JCheckBox jCheckBox1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel4;

public javax.swing.JButton login;

public javax.swing.JPasswordField password;

public javax.swing.JLabel password\_error;

public javax.swing.JLabel title1;

public javax.swing.JLabel title2;

public javax.swing.JLabel user\_error;

public javax.swing.JTextField userid;

// End of variables declaration

private void setIconImage() {

setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("logo.png")));

}

}

FUTURE SCOPE:

* Store patient tests reports
* Add payment methods
* Message the patients who have appointment tomorrow
* Make application very familiar to real world

CONCLUSION:

* Hospital management system is all about the modernizing a hospital through use of technology. Computers helps in it and take over the manual system for quick and easy functioning. This hospital management system is a quite the reliable and is proven on many stages. All the basic requirements of the hospital are provided in the hospital in order to manage it perfectly and large amount of data can also be stored . It gives many facilities like searching for the detail of patient , billing facilities as well as the creation of test reports. So it is a important system for modern days.

REFERENCES:

Youtube: <https://www.youtube.com/channel/UCnsZyWgc6v5uY251MkCHkvw>

Website: <https://docs.oracle.com/en/java/>

<https://www.javatpoint.com/>