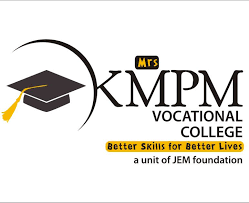
**Mrs. KMPM Vocational**

**COLLEGE**



 **Mrs. Poonam Mishra**

**Head Of Department**

**(BCA)**

**Mrs. KMPM Vocational College, JSR**

**2023**

**NURSING HOME**

**Management System**

****

A project report submitted in partial fulfillment of the requirement for the award of the degree of

**“BACHELOR OF COMPUTER APPLICATIONS”**

**SUBMITTED BY: Sourabh Sinha**

**Reg.No – KU2000291**

**Under Supervision Of**

**Mrs Poonam Mishra**

**(Internal Supervisor) (External Supervisor)**

**Department Of Computer Applications**

Mrs. Kmpm vocational College

(A Unit Of JEM Foundation)

(Affiliated To Kolhan University, Chaibasa )

Jamshedpur -831001

**DEPARTMENT OF BACHELOR OF COMPUTER APPLICATIONS**



Certificate

This is to certify that the project report Entitled “**Nursing Home Management System**“ has been submitted by **Sourabh Sinha (Reg. no:: KU2000291)**. In partial fulfillment of the Degree Of Bachelor of Computer Applications award to Kolhan University, Chaibasa is a record of bonafide work carried out by my guidance and supervision.

**Mrs Poonam Mishra**

**(Internal Supervisor)**

**HOD, Department Of Computer Application**

**DECLARATION**

I hereby declare that my Project Report titled \_\_\_\_\_\_\_\_\_\_is a bonafide record of

the project work which I have submitted to the Institute of Distance Education and

Online Learning, Chandigarh University in partial fulfillment of the credit

requirements for the degree of Master of Business Administration is my authentic

work. This project report has not been copied, duplicated, or plagiarised from any

other paper, journal, document, or book and has not been submitted to any

educational institute or otherwise for the award of any certificate, diploma, degree

or recognition.

This is an authentic piece of work and in case there is any query regarding the

same, I shall be held responsible for answering any queries in this regard

I hereby declare that my Project Report titled **“NURSING HOME MANAGEMENT SYSTEM”** is a bonafide record of the project work which I have submitted **to Mrs. KMPM Vocational College** in partial fulfillment of the credit requirements for the degree of Bachelor of Computer Applications is my authentic work. This project report has not been copied, duplicated, or plagiarised from any other paper, journal, document, or book and has not been submitted to any educational institute or otherwise for the award of any certificate, diploma, degree, or recognition. I also declare that wherever we have used material such as data, theoretical analysis, and text from other sources, I have given due credit to them by citing the source of work in this thesis. This is an authentic piece of work and in case there is any query regarding the same, I shall be held responsible for answering any queries in this regard

Sourabh Sinha

Reg.no- KU2000291

(Department Of Computer Application)

**Mrs. KMPM Vocational College**

**ACKNOWLEDGEMENT**

I would like to extend my gratitude to my guide Mrs. Poonam Mishra for her help. She helped me with all the technicalities and other details regarding the system.

I also appreciate the effort my project guide Mr. Manohar Singh showed in helping me throughout the project as and when required. I was new in the field of development and was unaware of the ambiance of a software development organization. It was he who guided me to work in a fashion of meeting the target time. With his encouragement, I completed this project.

I extend my heartfelt thanks to my parents and friends because without their help it would not be possible for me to complete this project.

Last but not least I would like to thank the almighty because this project would be impossible without his blessing.

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **S.no** | **CONTENT OF PROJECT** | **Page.no** |
|  | **Introduction** | **1-2** |
|  | **Objective** | **3** |
|  | **System Analysis** | **4** |
|  | **Preliminary Investigation** | **5** |
|  | **Interviews** | **6** |
|  | **Questionnaires** | **7-8** |
|  | **On-Site Observation** | **9** |
|  | **Document Reviewing** | **10** |
|  | **Feasibility Study** | **11-12** |
|  | **Project Planning** | **13** |
|  | **Project Scheduling** | **14-16** |
|  | **Hardware and Software Requirements** | **17** |
|  | **ER Diagram** | **19** |
|  | **Data Flow Diagram** | **20-25** |
|  | **Data Dictionary** | **26-27** |
|  | **Database Design** | **28-33** |
|  | **Coding** |  |
|  | **Input Screen** |  |
|  | **Validation and Security** |  |
|  | **Limitation** |  |
|  | **Future Scope** |  |
|  | **Bibliography** |  |

**INTRODUCTION**

The above system includes all the monitoring activities done in the nursing home. In recent years the trend of private institutions taking up health care to serve people has taken up a pace. This increases facilities regarding health care such as treatment of disease, operation facilities, and general consultation to doctors, which have been considered more serious. People have become more health conscious and thereby consider every sickness seriously.

In this competitive service sector, the requirement arises to maintain records, monitor each activity, and report different events with greater concern. Due to the above reason the system of nursing home automation has picked up a space.

Different other related to life insurance have also benefited from this system because people can maintain their proper health records. In case of any claims made by the concerned person, this record maintenance helps the claimer to get easy claims. Other health-related organizations such as Apollo Health Clinic, Relegate, etc. offer health packages. These health organizations are responsible for retaining the patients' health reports for around 10 years. It is not necessary that the patients only have to go for health check-ups as the deformities start to come with age. Previously once the person got well forgot about the disease and its ailments, but this should not be done because one disease can give rise to another disease and may show its effect over time. Every treatment requires a reference to the previous medical history of the patient. This helps the practitioner treat the patient well. Since the system requires a lot of documents to be maintained. The task becomes redundant and compulsory, hence it cannot be avoided. Health organizations are now considered businesses and are managed at the corporate level. The nursing homes are divided into different sections such as reception, cabins, wards, diagnostic sections, etc. The management of the nursing home desires to make it a more systematic and automated center. Every record monitoring, record referencing, resource management, doctor referencing, treatment, etc. are computerized so that the efficiency increases and management becomes more efficient and simpler.

OBJECTIVE

This project is being developed to maintain the Nursing Home in a more precise way. It helps in the automatic monitoring of the function performed by the nursing home as a daily routine.

All the details of the patient such as patient entry, admission, discharge, or change of ward/bed of the patient, are recorded by this system. This system also keeps records of different reports such as reports at the time of admission, discharge reports, bed/ward reports, test reports, and existing patient reports. Details of the medicine could also maintained by this system if at all the centers demand the same in future development. It is not treated in the project because, in the analysis period, I came to know that the medicine maintenance could be treated as an inventory process and it is given to others on a contractual basis.

The project developed as a part of the academic course, but for me, it is more than that. During the development of this project, the daily procedure of a nursing home is being kept in mind.

**SYSTEM ANALYSIS**

**NEED FOR THE SYSTEM:-**

A nursing home management system is an important system since it is designed regarding the working of a nursing home. In a nursing home, various events take place that are related to the health with the health of the of the people. Nowadays more than 80% of people are ill due to some or other disease. Some of them have some serious diseases while others have casual diseases. Now the number of nursing homes and pathology centers is increasing day by day.

For monitoring those nursing homes, the systems are required so that the events that occur there should be performed quickly and systematically. The use of such type of project consumes less time and effort while performing the daily sequence of the nursing home. In nursing, many events such as patient admission, daily treatment, tests performed, bed/ward change, discharge report, patient list, and doctor details are being kept. New entries are added daily and the system is growing day by day. So, it is very difficult to maintain these records without using a system.

**PRELIMINARY INVESTIGATION:-**

It is the determination of whether or not a project is worth doing. Once it has been determined that a project is feasible, the analyst can process it as a project, so different nursing homes were visited by the frequent visits to the nursing homes and from friends and relatives who were admitted to nursing homes at any point in time.

**FACT-FINDING:-**

After the above evaluation of the project, the facts related to the system are studied thoroughly. Different methods were applied to find out the facts and details of the system.

The details gathering about the system was done in a very systematic order which included frequent visits to the nursing home. Random questionnaires were put forward to the concerned person such as the receptionist of the nursing home employee in the queries which were done during fact fact-gathering process.

**On-Site Observation**

For gathering details regarding this project, various nursing homes and pathology centers are being visited. While gathering information regarding the project meeting with various doctors and pathologists helped me a lot in gathering information regarding the system.

While visiting different nursing homes we were able to learn details of various tests and also other works occurring in the nursing home. There I observed how all the activities occur in the nursing home. There we took how the treatment is going on. The timing of different doctors in the nursing home was also known. We put many arguments in front of receptionists regarding details of the treatment of various types of diseases.

**DOCUMENT REVIEWING**

In a nursing home documents regarding a patient and many other details are maintained. Several registers and files are being maintained for keeping the details of the patient in a nursing home, for example-

When a patient is admitted to a nursing home records are registered in more than one file such as an admission file, bed allocation file, etc. While conducting these works manually takes a lot of time, it also causes redundancy of data.

By the use of this project redundancy of data is reduced. It also reduces the consumption of time. With the use of this project, data are being maintained in a systematic manner that can be used easily, if we want to make any change in the created data it will be easy for us to do it. In case we have to see the record of a patient who got treatment in the nursing home a few months ago. It will be easy for us to recognize it but without the use of the system, it would be very difficult for us to search. Hence this system also reduces human effort

**FEASIBILITY STUDY OF THIS PROJECT**

**ECONOMIC FEASIBILITY:-**

It is the most frequently used method for evaluating the effectiveness of a new system. The benefits and savings are expected from the system. If the benefit outward the cost, the decision is made to design and implement the system, otherwise alternate system is considered.

The study that is to be developed requires a front-end software Visual Basic and a back-end database and MS- Access. Hence on the Software ground, the system costs around 20,000 to 25,000 excluding the Operating system which is arranged by the user and is easily affordable by the nursing home.

**TECHNICAL FEASIBILITY:-**

In technical feasibility, the financial condition is to be considered. If the budget is seriously constant, then the system is not feasible.

At the initial level, for small nursing homes, a single machine can be used for monitoring all the activities. A machine with a lower configuration, any Windows Operating System, or a normal printer can used with the system to be developed. Hence this machine is considered to be technically feasible

. **OPERATIONAL FEASIBILITY:-**

the proposed system is beneficial if the system meets the operating requirements of the organization. In this feasibility, the following points are considered:-

* Is there sufficient support for the project from the management or not?
* Are, the current business methods acceptable to the user or not?

Since the system is being developed with Visual Basic and a similar environment window is given, it can be easily used by a computer-savvy person

**PROJECT PLANNING**

Planning for a smaller project also requires effective management control to realize the desired result. Project planning for any company follows the following four main steps:

* Organizing the resources available for the project.
* Scheduling the events of the project.
* Evaluating the progress.
* Establishing standards for the project.

Effective management can complete the project on time, within budget, and with satisfactory results. To achieve a goal, there should be a strong determination and it should always be in one’s mind. Along with defining objectives, corporate management must assign priorities to various projects that:-  
Underway and clarify the relationship interaction between system projects and existing systems. A system project requires extensive interaction between developers and users. Many systems are designed and implemented through project teams headed by project teams headed by a project leader. To organize a project, the project leader must determine who is required for the project. Plans must be made to utilize overtime or/and shift personnel, to free the key systems users for participation in the project. The project leader is solely responsible for the completion of a project. Responsibilities avoided.

**PROJECT SCHEDULING**

The charting techniques are the scheduling tools of the project planner. Even the simplest project should be charted so that progress can be measured. The Gantt chart is effective for simple projects, especially when interrelationships among events are not too complex. Complicated scheduling usually requires a Pert chart. Projects are organized into modules, or segments, of related tasks.

Modular planning has the following advantages: -

It facilitates assigning responsibility and measuring processes and further allows systems analysts to work in concentrated areas of projects so they can master every aspect of the system. The key steps in a schedule are called milestones, or checkpoints. As the progresses, the date on which each milestone is completed is compared with the date for which it was projected. Enforcement is a normal managerial duty. The following are a variety of options that are open to the project leader: -

* Increase the budget.
* Increase manpower in the form of overtime or additional people.
* Add equipment.
* Change priorities.
* Replace the individual responsible.

Projects have many target dates, but few deadlines. Schedules are highly

dependent upon priorities and should be planned accordingly.

**SCHEDULING TECHNIQUE**

The following are various types of scheduling techniques in software

engineering are -

* **Work Breakdown Structure:-** the project is scheduled in various phases following a bottom-up or top-down approach. A tree-like structure is followed without any loops. At each phase or step, milestones and deliverables are mentioned concerning requirements. The work breakdown structure shows the overall breakup flow of the project and doesn't indicate any flow.

**• Flow graph:-** various modules are represented as nodes with edges connecting nodes. Dependency between nodes is shown by the flow of data between nodes. Nodes indicate milestones and deliverables with the corresponding module implemented.

* **Gantt charts or timeline chart: -** A Gantt chart can be developed for the entire project or a separate chart can be developed for each function. A tabular form is maintained where rows indicate the tasks with milestones and columns indicate duration. The horizontal bars that span across columns indicate the duration of the tasks.

The Gantt chart is a graphical representation that uses horizontal bars to

show the duration of the action or task.

A Gantt chart can be developed for the entire project or a separate chart can be developed for each function. A tabular form is maintained where rows with milestones and columns indicate the duration indicate the tasks (weeks/months). The horizontal bars that span across columns indicate the duration of the task. The circles indicate the milestones.

In the Gantt chart, the left side indicates the beginning of the task and the right side indicates the end of the task.

The Gantt Chart is used for scheduling the project work.

I have used the Gantt chart for scheduling this project for this system.

It has helped me to finish this system. Whenever I found that I was lagging behind the pace was picked up due to the scheduling scheme. It acted as a checkpoint for me.

**PERT Chart**

Pert chart stands for "Project Evaluation and Review Techniques". The main objective of this part is to find out the completion of a particular event within specified data. The pert approach text into account the uncertainty in the system. In this approach three time values are associated with each activity; the optimistic value, the pessimistic value, and the most likely value.

Pert Chart is used to project the Maximum duration the project will take to finish.

**Hardware and Software Requirement Specification**

**Hardware and Software Environment**

**Hardware Platform**

* Central Processing Unit(CPU) : Intel i3
* Random Access Memory(RAM): 4 GB
* Secondary Memory: 156 GB
* Pointing device: Mouse
* Monitor: 14’’ monitor VGA

**Software Platform**

* Operating system: Windows 10
* Front End Application Development: JAVA Applet
* Backend (Database): SQL
* Documentation design: MS Word 2016

**SOFTWARE ENGINEERING PARADIGM**

**Prototypic Model**

In this model, a working model of actual software is developed initially. The prototyping begins with requirements gathering. Develop meet and define the overall objective of the software, identify whatever requirements are known, and outline areas where further definition is mandatory. A quick design then occurs which focuses on a representation of those aspects of the software that will be visible to the customer or user. This quick design leads to the construction of a prototype. The prototype is evaluated by the customer or user and used to refine requirements for the software to be developed.

Iteration occurs as the prototype is tuned to satisfy the needs of the customer, while at the same time enabling the developer to better understand what needs to be done. Ideally, the prototype serves as a mechanism for identifying software requirements. The prototype can serve as a first system that is light by both the customer and developers because the user gets a feel of the actual system and developers get to build something immediately. The prototype model has the following features: -

* It helps in determining user requirements more deeply.
* At the time of actual product development, customer feedback is available.
* It does consider any types of risks at the initial level.

**ENTITY RELATIONSHIP DIAGRAM**

**ENTITY:-**

It is a source of input to the system and a destination of output produced by the system. An entity is an object that is of interest to an organization, objects of similar types are characterized by the same set of attributes or properties. Two objects are mutually distinguished and this fact is represented is represented in the entity set by giving them unique identifiers. Entities are the basic units used in modeling classes of concrete abstract objects.

**RELATIONSHIP:-**

A relationship set is used in data modeling to represent an association between entity sets. This association could have certain properties represented by the attributes of the relationship set. Each relationship set is named.

**DISCRIPTION OF DIFFERENT LEVEL OF DFD**

phone

age

DOA

ward

**PATIENT**

PID

**TEST**

**TAKES**

**PROCESS:-**

A process represents function in the system. It is a task that transforms incoming data to outgoing data by the logic of process. Each process is identified by unique name and number. It is represented by a circle.

**ENTITY:-**

It is a person or an organization that is outside the boundary of the system. It is a source of input to the system and destination of output produced by the system. It is represented by a rectangle.

**DATA STORE:-**

It is a file that stores a data in a system. It may be an area of disk or a manual index file. It is represented by open ended rectangle.

**0-Level DFD**

This level of DFD is basically drawn to identify the main system and the entities involved in it. It is not explanatory.

PATIENT

RECEPTION

DOCTOR

PATHOLOGY

REGISTER/REQUEST REPORT

RESPONSE

REGISTER/REQUEST PATIENT/

DOCTOR

**1-LEVEL DFD**

This level of data flow diagram displays all the main processes involved in the system, and the data store in use

PATHOLOGY

RECEPTIONIST

REPORT

PATIENT

DOCTOR

**Doc\_dbf**

**Pat\_dbf Test**

**Rep\_dbf**

**2nd level DFD(for all the process involving the entities Doctor and Patient)**

**This level if DFD includes all sub-process involved in the system.**

Doctor

**w-change**

**Patient Admission**

**P-details**

**Doc-dbf**

Patient

Receptionist

**2nd level DFD(for report process)**

Rceptionist

Patient p-test Dis-dbf

Pathology

Patient

Wchange-dbf

**2nd level DFD(for process bed change and test payment)**

reception

Patient

**Bill-det Test-dwt**

**DATA DICTIONARY**

It is a data structure that stores information about all the fields used in a system. The data dictionary is an organized listing of all data elements that are pertinent to the system, with precise decisions so that both the user and system analyst will have a common understanding of inputs, outputs, and components of stores. It consists of a set of rigorous definitions of old data elements and data structure. The data dictionary should describe the following.

* **Name:-** The primary name of data or control item, data store or an external entity.
* **Alias:**- Other name used for first entry.
* **Existing:**- Where used/how used?
* **Description:**- It involves the description of the name.
* **Future reference** or combinational fields might not be understood after a certain period of time, so a dictionary helps them to understand them.

The notifications used to develop a context description are as follows:-

**Notation meaning**

* **+ 🡪 and**
* **= 🡪 Comprised of**
* **[] 🡪 Either/or**
* **() 🡪 Operational entry**

**DATABASE DESIGN**

The following are the being maintained in that organization. Each table has a vital task to perform.

**Table Name: logintbl**

This database is used for authentication purposes. This table looks for whenever the authentication of the user is checked.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| usr | String | 10 | User Name |
| pwd | Integer | 8 | Password |

**Table Name: pattbl**

This is the master table of the patient. It stores all the attributes of the patient that require to be monitored.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| pat\_id | varchar | 50 | Patient id |
| pat\_name | varchar | 50 | Patient name |
| pat\_address | varchar | 50 | Patient address |
| pat\_contact | varchar | 50 | Patient contact |
| pat\_gender | varchar | 50 | Patient gender |
| pat\_aadhar | varchar | 50 | Patient aadhar |
| p\_date | date | 20 | Patient admit date |
| status | varchar | 50 | Discharge/admit status |
| p\_dob | varchar | 50 | Patient DOB |
| discharge\_d | varchar | 50 | Discharge date |

**Table Name: doctbl**

This is the master table of the doctor. It stores all the attributes of the doctor that require to be monitored.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| doctor\_id | varchar | 50 | Doctor id |
| doctor\_name | varchar | 50 | Doctor name |
| doc\_address | varchar | 50 | Doctor address |
| doc\_email | varchar | 50 | Doctor email |
| doc\_doj | varchar | 50 | Doctor date of joining |
| doc\_contact | varchar | 50 | Doctor id |
| doc\_gender | date | 20 | Doctor id |
| doc\_specializaton | varchar | 50 | Doctor id |
| doc\_aadhar | varchar | 50 | Doctor id |
| doc\_dob | varchar | 50 | Doctor id |

**Table Name: testrequesttbl**

This table stores the details of the test being undertaken by the patient on a regular basis.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| test\_id | varchar | 50 | Test id |
| test\_name | varchar | 50 | Test name |
| pname | varchar | 50 | Patient name |
| tcharge | varchar | 50 | Test charge |
| type | varchar | 50 | Patient type |
| admin\_id | varchar | 50 | Admission id |
| pt\_reqid | varchar | 50 | Request id |
| t\_reqid | varchar | 50 | Test req id |
| contact | varchar | 50 | Patient contact |
| net\_payable | number | 50 | Amt payable |
| test\_date | date | 20 | Test date |

SPLASH FORM

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class splash\_page extends JFrame implements ActionListener

{

int ctr = 0;

JProgressBar pbar = new JProgressBar();

JLabel lblback2 = new JLabel();

JLabel lblback1 = new JLabel();

JLabel lblback = new JLabel(new ImageIcon("zimg1.jpeg"));

JLabel l1 = new JLabel("NURSING HOME");

JLabel l2 = new JLabel("", SwingConstants.CENTER);

JLabel l4 = new JLabel("DEVELOPED BY - sourabh");

public splash\_page()

{

setVisible(true);

setSize(826,700);

setTitle("Nursing Home");

setLayout(null);

setLocationRelativeTo(null);

setBackground(new Color(120,120,120));

lblback2.setBounds(00,05,826,80);

lblback2.setBackground(new Color(0,0,105));

lblback2.setOpaque(true);

pbar.setBounds(13,500,784,30);

lblback.setBounds(-10,100,826,300);

l1.setBounds(280,-100,826,300);

l2.setBounds(-10,450,826,40);

l4.setBounds(480,600,350,40);

lblback.setOpaque(true);

lblback1.setOpaque(true);

pbar.setValue(0 );

pbar.setBorderPainted(false);

pbar.setStringPainted(true);

pbar.setBackground(new Color(200,206,211));

pbar.setForeground(new Color(20,20,160));

pbar.setFont(new Font("Arial", Font.BOLD, 20));

lblback1.setBounds(0,0,826,1000);

lblback1.setBackground(new Color(200,206,211));

l1.setFont(new Font("Algerian",Font.BOLD,36));

l1.setForeground(new Color(255,255,255));

l2.setFont(new Font("Arial",Font.BOLD,22));

l2.setForeground(new Color(20,20,160));

l4.setFont(new Font("Algerian",Font.BOLD,24));

l4.setForeground(new Color(20,20,160));

add(l1);

add(lblback2);

add(pbar);

add(l2);

add(l4);

// add(b1);

add(lblback);

add(lblback1);

start1();

}

public void actionPerformed(ActionEvent e)

{

}

public void start1()

{

int i=0;

pbar.setValue(0);

ctr = 0;

while(ctr<=100)

{

pbar.setValue(ctr);

try

{

if(ctr<30)

{

i=50;

l2.setText("Working on it");

}

else if(ctr>30 && ctr<34)

{

i=500;

l2.setText("Loading......");

}

else if(ctr>60 && ctr<64)

{

i=500;

l2.setText("Loading the system for you!!!");

}

else

{

i=50;

l2.setText("Processing......");

}

Thread.sleep(i);

}

catch(InterruptedException e)

{

e.printStackTrace();

}

ctr++;

if(ctr==100)

{

this.dispose();

login obj1 = new login();

}

}

}

public static void main(String[] args)

{

splash\_page obj = new splash\_page();

}

}

login FORM

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.sql.\*;

public class login extends JFrame implements ActionListener,MouseListener,MouseMotionListener

{

    JLabel  lblback = new JLabel("");

    JLabel  lblimg = new JLabel(new ImageIcon("zimg3.jpg"));

    JLabel  lblhead = new JLabel("");

    JLabel  lbltxt = new JLabel(" USER LOGIN");

    JLabel l2 = new JLabel(" user\_id");

    JLabel l3 = new JLabel("password");

    JLabel lblback1a =new JLabel("");

    JLabel lblback1 =new JLabel("");

    JTextField tx1 = new JTextField();

    JPasswordField tx2 = new JPasswordField();

    JButton bok = new JButton("login");

    JButton bcancel = new JButton("CANCEL");

    public login()

    {

        setVisible(true);

        setSize(900,500);

        setTitle("Login Page");

        setLayout(null);

        setLocationRelativeTo(null);

        setResizable(false);

        lblimg.setBounds(00,00,500,500 );

lblhead.setBounds(502,05,400,70);

        lblhead.setBackground(new Color(00,00,105));

        lblhead.setOpaque(true);

        lbltxt.setBounds(650,20,250,40);

        lbltxt.setForeground(new Color(255,255,255));

        lbltxt.setFont(new Font("Arial", Font.BOLD,22));

        lblback.setBounds(00,00,900,500);

        lblback.setBackground(new Color(200,206,211));

        //back1

        lblback1.setBounds(550,110,300,250);

        lblback1.setBackground(new Color(0,0,105));

        lblback1.setOpaque(true);

        lblback1a.setBounds(552,112,296,246);

        lblback1a.setBackground(new Color(200,206,211));

        lblback1a.setOpaque(true);

        l2.setBounds(620,120,70,30);

        l3.setBounds(620,200,70,30);

        tx1.setBounds(620,150,170,30);

        tx2.setBounds(620,230,170,30);

        bok.setBounds(620,300,80,30);

        bok.setBackground(new Color(0,0,105));

        bok.setForeground(new Color(255,255,255));

        bcancel.setBounds(720,300,80,30);

        bcancel.setBackground(new Color(0,0,105));

        bcancel.setForeground(new Color(255,255,255));

        lblback.setOpaque(true);

        bok.addActionListener(this);

        bcancel.addActionListener(this);

        bok.addMouseListener(this);

        bcancel.addMouseListener(this);

        add(lblimg);

        add(lbltxt);

        add(lblhead);

        add(l2);

        add(l3);

        add(tx1);

        add(tx2);

        add(bok);

        add(bcancel);

        add(lblback1a);

        add(lblback1);

        add(lblback);

repaint();

    }

    public void actionPerformed(ActionEvent e)

    {

        if(e.getSource() == bok)

        {

            int flg=0;

          try

            {

                Connection con=DriverManager.getConnection("jdbc:odbc:nursing")

                String str="select \* from logintbl where usr='"+tx1.getText()+"' and pwd='"+tx2.getText()+"'";

                Statement st=con.createStatement();

                ResultSet res=st.executeQuery(str);

                while(res.next())

                {

                    flg=1;

                    third obj2 = new third(tx1.getText(),tx2.getText());

                    this.dispose();

                }

                if(flg==0)

                {

JOptionPane.showMessageDialog(null,"Invalid login");

                    tx1.setText("");

                    tx2.setText("");

                    tx1.requestFocus()

                    con.close();

                }

            catch(Exception ee)

            {

                System.out.println(ee);

            }

        }

        if(e.getSource() == bcancel)

        {

           this.dispose();

        }

    }

    public void mouseClicked(MouseEvent m)

    {}

    public void mouseDragged(MouseEvent m)

    {}

    public void mouseEntered(MouseEvent m)

    {

      if(m.getSource()==bok)

      {

        bok.setForeground(new Color(0, 0, 105));

        bok.setBackground(new Color(255,255,255));

      }

      if(m.getSource()==bcancel)

      {

        bcancel.setForeground(new Color(0, 0, 105));

        bcancel.setBackground(new Color(255,255,255));

      }

    }

    public void mouseExited(MouseEvent m)

    {

        if(m.getSource()==bok)

        {

            bok.setForeground(new Color(255,255,255));

            bok.setBackground(new Color(0, 0, 105));

        }

        if(m.getSource()==bcancel)

        {

            bcancel.setForeground(new Color(255,255,255));

            bcancel.setBackground(new Color(0, 0, 105));

        }

    }

    public void mouseMoved(MouseEvent m)

    {}

    public void mousePressed(MouseEvent m)

    {}

    public void mouseReleased(MouseEvent m)

    {

 public static void main(String[] args) {

     login obj1=new login();

 }

menu FORM

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.sql.\*;

public class third extends JFrame implements ActionListener,MouseListener,MouseMotionListener

{

SimpleDateFormat sdf = new SimpleDateFormat("dd/MMM/yyyy");

Date dt=new Date();

JLabel lblbackground = new JLabel("")

JLabel lblhead = new JLabel("");

JLabel lbltxt = new JLabel("NURSING HOME MANAGMENT SYSTEM")

JLabel lblback1 = new JLabel("");

JLabel lbl1[] = new JLabel[10];

JLabel lbl2[] = new JLabel[10];

JMenuBar mb = new JMenuBar();

JMenu menmaster = new JMenu("MMASTER");

JMenu mentrans = new JMenu("TRANSACTION");

JMenu menreports = new JMenu("REPORTS");

JMenu menexit = new JMenu("EXIT");

JMenuItem mastpatient = new JMenuItem("PATIENT");

JMenuItem mastdoctor = new JMenuItem("DOCTOR");

JMenuItem mastward = new JMenuItem("WARD");

JMenuItem masttest = new JMenuItem("TESTS");

JMenuItem transadmit = new JMenuItem("ADMIT");

JMenuItem transdischarge = new JMenuItem("DISCHARGE");

JMenuItem transpayment = new JMenuItem("PAYMENT");

JMenuItem transtestrequest = new JMenuItem("TEST REQUEST");

JMenuItem transrefund = new JMenuItem("REFUND");

JMenuItem transbedchange = new JMenuItem("BED CHANGE");

JMenuItem reporttest = new JMenuItem("TEST REPORT");

JMenuItem reportpay = new JMenuItem("PAYMENT REPORT");

JMenuItem exiting = new JMenuItem("EXITING");

JLabel label1 = new JLabel("Patients Currently Admitted",JLabel.CENTER);

JLabel label11 = new JLabel("",JLabel.CENTER);

JLabel label2 = new JLabel("Payment Recived Today",JLabel.CENTER);

JLabel label22 = new JLabel("",JLabel.CENTER);

JLabel label3 = new JLabel("Tests Conducted Today",JLabel.CENTER);

JLabel label33 = new JLabel("",JLabel.CENTER);

JLabel label4 = new JLabel("No.Of Doctors",JLabel.CENTER);

JLabel label44 = new JLabel("",JLabel.CENTER);

JLabel label5 = new JLabel("Tests Available",JLabel.CENTER);

JLabel label55 = new JLabel("",JLabel.CENTER);

JLabel label6 = new JLabel("Patients Discharged Today",JLabel.CENTER);

JLabel label66 = new JLabel("",JLabel.CENTER);

JLabel label7 = new JLabel("Beds available in CCU",JLabel.CENTER);

JLabel label77 = new JLabel("",JLabel.CENTER);

JLabel label8 = new JLabel("Beds available in ICU",JLabel.CENTER);

JLabel label88 = new JLabel("",JLabel.CENTER);

JLabel label9 = new JLabel(" beds available in GENM",JLabel.CENTER);

JLabel label99 = new JLabel("",JLabel.CENTER);

JLabel label10 = new JLabel(" beds available in GENF",JLabel.CENTER);

JLabel label110 = new JLabel("",JLabel.CENTER);

JLabel lbldate = new JLabel();

JLabel lbllogout = new JLabel("LOGOUT");

JButton blogout = new JButton(new ImageIcon("zimg6.png"));

JLabel l1 = new JLabel();

String t1,t2;

int xy;

Dimension di = Toolkit.getDefaultToolkit().getScreenSize();

String n1 ;

String n2;

public third(String string, String string2)

{

setVisible(true);

setSize((int)di.getWidth(),(int)di.getHeight());

setTitle("Menu");

setLayout(null);

setResizable(false);

l1.setBounds(50,50,500,100);

//l1.setFont(new Font("Arial" , Font.BOLD,40));

//box

label1.setBounds(270,110,280,40);

label1.setOpaque(true);

label1.setBackground(Color.decode("#00A36C"));

label11.setBounds(270,150,280,110);

label11.setOpaque(true);

label11.setBackground(Color.decode("#3090C7"));

label11.setFont(new Font("Arial",Font.BOLD,48))

label2.setBounds(610,110,280,40);

label2.setOpaque(true);

label2.setBackground(Color.decode("#00A36C"));

label22.setBounds(610,150,280,110);

label22.setOpaque(true);

label22.setBackground(Color.decode("#3090C7"));

label22.setFont(new Font("Arial",Font.BOLD,48));

label3.setBounds(940,110,280,40);

label3.setOpaque(true);

label3.setBackground(Color.decode("#00A36C"));

label33.setBounds(940,150,280,110);

label33.setOpaque(true);

label33.setBackground(Color.decode("#3090C7"));

label33.setFont(new Font("Arial",Font.BOLD,48));

label4.setBounds(270,310,280,40);

label4.setOpaque(true);

label4.setBackground(Color.decode("#00A36C"));

label44.setBounds(270,350,280,110);

label44.setOpaque(true);

label44.setBackground(Color.decode("#3090C7"));

label44.setFont(new Font("Arial",Font.BOLD,48));

label5.setBounds(610,310,280,40);

label5.setOpaque(true);

label5.setBackground(Color.decode("#00A36C"));

label55.setBounds(610,350,280,110);

label55.setOpaque(true);

label55.setBackground(Color.decode("#3090C7"));

label55.setFont(new Font("Arial",Font.BOLD,48));

label6.setBounds(940,310,280,40);

label6.setOpaque(true);

label6.setBackground(Color.decode("#00A36C"));

label66.setBounds(940,350,280,110);

label66.setOpaque(true);

label66.setBackground(Color.decode("#3090C7"));

label66.setFont(new Font("Arial",Font.BOLD,48));

label7.setBounds(240,510,230,40);

label7.setOpaque(true);

label7.setBackground(Color.decode("#00A36C"));

label77.setBounds(240,550,230,110);

label77.setOpaque(true);

label77.setBackground(Color.decode("#3090C7"));

label77.setFont(new Font("Arial",Font.BOLD,48));

label8.setBounds(530,510,230,40);

label8.setOpaque(true);

label8.setBackground(Color.decode("#00A36C"));

label88.setBounds(530,550,230,110);

label88.setOpaque(true);

label88.setBackground(Color.decode("#3090C7"));

label88.setFont(new Font("Arial",Font.BOLD,48));

label9.setBounds(820,510,230,40);

label9.setOpaque(true);

label9.setBackground(Color.decode("#00A36C"));

label99.setBounds(820,550,230,110);

label99.setOpaque(true);

label99.setBackground(Color.decode("#3090C7"));

label99.setFont(new Font("Arial",Font.BOLD,48));

label10.setBounds(1110,510,230,40);

label10.setOpaque(true);

label10.setBackground(Color.decode("#00A36C"));

label110.setBounds(1110,550,230,110);

label110.setOpaque(true);

label110.setBackground(Color.decode("#3090C7"));

label110.setFont(new Font("Arial",Font.BOLD,48))

//back1

lblback1.setBounds(00,15,200,700);

lblback1.setBackground(new Color(07, 115, 180));

lblback1.setOpaque(true)

//heading

lblhead.setBounds(00,00,1360,80);

lblhead.setBackground(new Color(0,0,105));

lblhead.setOpaque(true)

lbltxt.setBounds(370,05,650,70);

lbltxt.setForeground(new Color(255,255,255));

lbltxt.setFont(new Font("Algerian",Font.BOLD,32)

//background

lblbackground.setBounds(0,0,(int)di.getWidth(),(int)di.getHeight());

lblbackground.setBackground(new Color(200,206,211));

lblbackground.setOpaque(true)

lbldate.setBounds(1220,70,130,50);

lbldate.setBackground(new Color(200,206,211));

lbldate.setOpaque(true);

lbldate.setFont(new Font("Algerian",Font.BOLD,16))

lbllogout.setBounds(20,650,110,50);

lbllogout.setBackground(new Color(07, 115, 180));

lbllogout.setOpaque(true);

lbllogout.setFont(new Font("Arial",Font.BOLD,16))

blogout.setBounds(130,650,50,52);

blogout.setBackground(new Color(07, 115, 180));

//lbl

for(int j=0 ; j<10 ; j++)

{

lbl1[j] = new JLabel();

}

int a =0;

for(int i=0;i<10;i++)

{

a+=60;

lbl1[i].setBounds(30,60+a,110,30);

lbl1[0].setText("Doctor");

add(lbl1[0]);

lbl1[1].setText("Patient");

add(lbl1[1]);

lbl1[2].setText("Test Request");

add(lbl1[2]);

lbl1[3].setText("Payment");

add(lbl1[3]);

lbl1[4].setText("Admit");

add(lbl1[4]);

lbl1[5].setText("Bed Change");

add(lbl1[5]);

lbl1[6].setText("Discharge");

add(lbl1[6]);

lbl1[7].setText("Test Report");

add(lbl1[7]);

lbl1[8].setText("Pay Report");

add(lbl1[8]);

}

//MENUBAR

add(menmaster);

add(mentrans);

mb.add(menmaster);

mb.add(mentrans);

mb.add(menreports);

mb.add(menexit);

menmaster.add(mastpatient);

menmaster.add(mastdoctor);

menmaster.add(mastward);

menmaster.add(masttest);

mentrans.add(transadmit);

mentrans.add(transdischarge);

mentrans.add(transpayment);

mentrans.add(transtestrequest);

mentrans.add(transrefund);

mentrans.add(transbedchange);

menreports.add(reporttest);

menreports.add(reportpay);

menexit.add(exiting);

// add(mb);

add(l1);

menmaster.addActionListener(this);

add(label4);

add(label44);

add(label5);

add(label55);

add(label6);

add(label66);

add(label7);

add(label77);

add(label8);

add(label88);

add(label9);

add(label99);

add(label10);

add(label110);

add(lbldate);

add(lbllogout);

add(blogout);

//back1

add(lblback1);

//backgrou-nd

add(lblbackground);

int ctr=0;

try

{

Connection con = DriverManager.getConnection("jdbc:odbc:nursing");

String str = "select pat\_id from pattbl where status = 'admitted'";

Statement st = con.createStatement();

ResultSet res = st.executeQuery(str);

while(res.next())

{

ctr++;

}

label11.setText(String.valueOf(ctr));

}

catch(Exception ee)

{

System.out.print(ee);

}

int ctrd=0;

try

{

Connection con = DriverManager.getConnection("jdbc:odbc:nursing");

String str = "select doctor\_id from doctbl '";

Statement st = con.createStatement();

ResultSet res = st.executeQuery(str);

while(res.next())

{

ctrd++;

}

label44.setText(String.valueOf(ctrd));

}

catch(Exception ee)

{

System.out.print(ee);

}

int ctrt=0;

try

{

Connection con = DriverManager.getConnection("jdbc:odbc:nursing");

String str = "select t\_reqid from testrequesttbl where test\_date ='"+sdf.format(dt)+"'";

Statement st = con.createStatement();

ResultSet res = st.executeQuery(str);

while(res.next())

{

ctrt++;

}

system.out.print(ctrt);

label33 .setText(String.valueOf(ctrt));

}

catch(Exception ee)

{

System.out.print(ee);

}

int ctrta=0;

{

Connection con = DriverManager.getConnection("jdbc:odbc:nursing");

String str = "select testr\_id from testdetailstbl '";

Statement st = con.createStatement();

ResultSet res = st.executeQuery(str);

while(res.next())

{

ctrta++;

}

label55.setText(String.valueOf(ctrta));

}

catch(Exception ee)

{

System.out.print(ee);

}

int ctrdt=0;

try

{

Connection con = DriverManager.getConnection("jdbc:odbc:nursing");

String str = "select pat\_id from pattbl where status = 'discharged'"+ " and discharge\_date = '"+sdf.format(dt)+"'";

Statement st = con.createStatement();

ResultSet res = st.executeQuery(str);

while(res.next())

{

ctrdt++;

}

System.out.print(ctrdt);

label66.setText(String.valueOf(ctrdt));

}

catch(Exception ee)

{

System.out.print(ee);

}

int ctrp=0;

try

{

Connection con = DriverManager.getConnection("jdbc:odbc:nursing");

String str ="select pay\_id from paymenttbl where payment\_date ='"+sdf.format(dt)+"'";

Statement st = con.createStatement();

ResultSet res = st.executeQuery(str);

while(res.next())

{

ctrp++;

}

label22.setText(String.valueOf(ctrp));

}

catch(Exception ee)

{

System.out.print(ee);

}

int ctrbc=0;

try

{

Connection con = DriverManager.getConnection("jdbc:odbc:nursing");

String str ="select \* from wardtbl where ward ='CCU'";

Statement st = con.createStatement();

ResultSet res = st.executeQuery(str);

while(res.next())

{

for(int i=3;i<18;i++)

{

String cstat=res.getString(i);

if(cstat.equalsIgnoreCase("A"))

{

ctrbc++;

}

}

}

label77.setText(String.valueOf(ctrbc));

}

catch(Exception ee)

{

System.out.print(ee);

}

int ctrbi=0;

try

{

Connection con = DriverManager.getConnection("jdbc:odbc:nursing");

String str ="select \* from wardtbl where ward ='ICU'";

Statement st = con.createStatement();

ResultSet res = st.executeQuery(str);

while(res.next())

{

for(int i=3;i<18;i++)

{

String istat=res.getString(i);

if(istat.equalsIgnoreCase("A"))

{

ctrbi++;

}

}

}

label88.setText(String.valueOf(ctrbi));

System.out.print(ee);

{

wardsel obj2 = new wardsel(1);

this.dispose();

}

if(m.getSource() == lbl1[5])

{

bedchange bdobj = new bedchange("");

this.dispose();

}

if(m.getSource() == lbl1[6])

{

doctor objd = new doctor();

this.dispose();

}

if(m.getSource() == lbl1[7])

{

test\_report trobj = new test\_report();

this.dispose();

}

if(m.getSource() == lbl1[8])

{

payment\_report bdobj = new payment\_report();

this.dispose();

}

}

public void mouseEntered(MouseEvent m)

{}

public void mouseExited(MouseEvent m)

{}

public void mouseMoved(MouseEvent m)

{}

public void mousePressed(MouseEvent m)

{}

public void mouseReleased(MouseEvent m)

{}

public static void main(String[] args)

{

third obj = new third ("s","e");

}

}

DOCTOR FORM

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.sql.\*;

public class doctor extends JFrame implements ActionListener,MouseListener,MouseMotionListener,KeyListener

{

SimpleDateFormat sdf = new SimpleDateFormat("dd/MMM/yyyy");

Date dt=new Date();

JLabel lblback = new JLabel(" ")

JLabel lblhead = new JLabel(" ");

JLabel lbldoctor = new JLabel("DOCTOR");

JLabel lblback1 = new JLabel("");

JLabel lblback1a = new JLabel("");

JLabel lbltxt1 = new JLabel("Doctor Details");

JLabel lblback2 = new JLabel("");

JLabel lblback2a = new JLabel("");

JLabel lbldoctorid = new JLabel("DOCTOR\_ID ");

JLabel lbldoctorname = new JLabel("DOCTOR\_NAME ");

JLabel lbladress = new JLabel("ADDRESS ");

JLabel lblemail = new JLabel("Email id ");

JLabel lblcontact = new JLabel("CONTACT");

JLabel lbldob = new JLabel("DOB(dd/mm/yyy)");

JLabel lbldoj = new JLabel("DOJ(dd/mm/yyy)");

JLabel lblgender = new JLabel("Gender ");

JLabel lblspecialized = new JLabel("Specialization ");

JLabel lblaadhar = new JLabel("UID NUMBER ");

JLabel lblm = new JLabel("male ");

JLabel lblf = new JLabel("female ");

JTextField txdoctorid = new JTextField();

JTextField txdoctorname = new JTextField();

JTextField txadress = new JTextField();

JTextField txemail = new JTextField();

JTextField txcontact = new JTextField();

JTextField txdob = new JTextField();

JTextField txdoj = new JTextField();

JTextField txgender = new JTextField();

JTextField txspecialized = new JTextField();

JTextField txaadhar = new JTextField();

JRadioButton rbmale = new JRadioButton();

JRadioButton rbfemale = new JRadioButton();

ButtonGroup bg = new ButtonGroup();

JButton bnew = new JButton("New");

JButton bsave = new JButton("Save");

JButton bsearch = new JButton("Search");

JButton bedit = new JButton("Edit");

JButton bexit = new JButton("Exit");

String s1,s2;

Dimension di = Toolkit.getDefaultToolkit().getScreenSize();

public doctor()

{

setVisible(true);

setSize(900,600);

setTitle("Doctor");

setLayout(null);

setResizable(false);

setLocationRelativeTo(null);

//head

lblhead.setBounds(00,05,900,60);

lblhead.setBackground(new Color (0,0,105));

lblhead.setOpaque(true);

lbldoctor.setBounds(370,5,150,60);

lbldoctor.setFont(new Font("Arial",Font.BOLD,18));

lbldoctor.setForeground(new Color(255,255,255));

//back1

lblback1.setBounds(100,100,650,350);

lblback1.setBackground(new Color(0,0,105));

lblback1.setOpaque(true);

lblback1a.setBounds(102,102,646,346);

lblback1a.setBackground(new Color(200,206,211));

lblback1a.setOpaque(true);

lbltxt1.setBounds(110,80,125,40);

lbltxt1.setFont(new Font("Arial",Font.BOLD,18));

lbltxt1.setBackground(new Color(200,206,211));

lbltxt1.setOpaque(true);

lbldoctorid.setBounds(140,150,110,30);

lbladress.setBounds(140,210,110,30);

lblcontact.setBounds(140,270,110,30);

lbldoj.setBounds(140,330,110,30);

lblspecialized.setBounds(140,390,110,30);

lbldoctorname.setBounds(450,150,110,30);

lblgender.setBounds(450,210,110,30);

lblemail.setBounds(450,270,110,30);

lbldob.setBounds(450,330,110,30);

lblaadhar.setBounds(450,390,110,30);

txdoctorid.setBounds(240,150,110,30);

txadress.setBounds(240,210,110,30);

txcontact.setBounds(240,270,110,30);

txdoj.setBounds(240,330,110,30);

txspecialized.setBounds(240,390,110,30);

txdoctorname.setBounds(570,150,110,30);

txemail.setBounds(570,270,110,30);

txdob.setBounds(570,330,110,30);

txaadhar.setBounds(570,390,110,30);

rbmale.setBounds(520,215,20,20);

rbmale.setBackground(new Color(200,206,211));

rbmale.setOpaque(true);

lblm.setBounds(550,209,100,30);

rbfemale.setBounds(610,215,20,20);

rbfemale.setBackground(new Color(200,206,211));

] lblback2a.setBackground(new Color(200,206,211));

lblback2a.setOpaque(true);

bnew.setBounds(200,490,70,30);

bnew.setBackground(new Color(0,0,105));

bnew.setForeground(new Color(255,255,255));

bsearch.setBounds(300,490,80,30);

bsearch.setBackground(new Color(0,0,105));

bsearch.setForeground(new Color(255,255,255));

bsave.setBounds(400,490,70,30);

bsave.setBackground(new Color(0,0,105));

bsave.setForeground(new Color(255,255,255));

bedit.setBounds(500,490,70,30);

bedit.setBackground(new Color(0,0,105));

bedit.setForeground(new Color(255,255,255));

bexit.setBounds(590,490,70,30);

bexit.setBackground(new Color(0,0,105));

bexit.setForeground(new Color(255,255,255));

//back

lblback.setBounds(00,00,900,630);

lblback.setBackground(new Color(200,206,211));

lblback.setOpaque(true);

bnew.addActionListener(this);

txdoctorid.setEditable(false);

txdoj.setEditable(false)

txdoj.setText(sdf.format(dt));

//lblhead

add(lbldoctor);

add(lblhead);

add(lbldoctorname);

add(lblgender);

add(lblemail);

add(lbldob);

add(lblaadhar);

add(txdoctorid);

add(txadress);

add(txcontact);

add(txdoj);

add(txspecialized);

add(txdoctorname);

add(txemail);

add(txdob);

add(txaadhar);

bg.add(rbmale);

bg.add(rbfemale);

repaint();

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource() == bnew)

{

try

System.out.println("111111111");

Connection con=DriverManager.getConnection("jdbc:odbc:nursing");

String str = "select doctor\_id from keytbl";

Statement st=con.createStatement();

ResultSet res=st.executeQuery(st

System.out.println("22222222222222");

while(res.next())

{

System.out.println("33333333333333");

txdoctorid.setText("Doc/"+res.getString(1));

}

System.out.println("4444444");

[(false);

txdob.setEditable(false);

txspecialized.setEditable(false);

txdoj.setEditable(false);

txaadhar.setEditable(false);

txcontact.setEditable(false);

txemail.setEditable(false);

txadress.setEditable(false);

rbmale.setEnabled(false);

rbfemale.setEnabled(false);

String id = "";

try

{

id = txdoctorid.getText();

JOptionPane.showMessageDialog(null , "Please provide a Doctor UID");

Connection con=DriverManager.getConnection("jdbc:odbc:nursing");

String str = "INSERT INTO doctbl (doctor\_id,doctor\_name,doc\_adress,doc\_email,doc\_doj,doc\_contact,doc\_gender,doc\_specialised,doc\_aadhar,doc\_dob)values(?,?,?,?,?,?,?,?,?,?)";

PreparedStatement ps = con.prepareStatement(str);

ps.setString(1, txdoctorid.getText());

ps.setString(2, txdoctorname.getText());

ps.setString(3, txadress.getText());

ps.setString(4, txemail.getText());

ps.setString(5, txdoj.getText());

ps.setString(6, txcontact.getText());

if(rbmale.isSelected())

{

gen = "Male";

}

else

{

gen = "Female";

}

ps.setString(7, gen);

ps.setString(9, txaadhar.getText());

ps.setString(8, txspecialized.getText());

ps.setString(10, txdob.getText());

ps.executeUpdate();

String str1 = "update keytbl set doctor\_id = doctor\_id + 1";

Statement st1 = con.createStatement();

st1.executeUpdate(str1);

JOptionPane.showMessageDialog(null, "Record Saved Successfully");

}

txdoctorname .setText("");

txadress.setText("");

txemail.setText("");

txcontact.setText("");

txdob.setText("");

txspecialized.setText("");

txaadhar.setText("");

bg.clearSelection();

}

if(e.getSource() == bedit)

{

}

}

public void focusGained(FocusEvent fe)

{}

public void mouseClicked(MouseEvent m)

{}

public void mouseDragged(MouseEvent m)

{}

public void mouseEntered(MouseEvent m)

{

if(m.getSource()==bnew)

{

bnew.setForeground(new Color(0, 0, 105));

bnew.setBackground(new Color(255,255,255));

}

if(m.getSource()==bsearch)

{

bsearch.setForeground(new Color(0, 0, 105));

bsearch.setBackground(new Color(255,255,255));

}

if(m.getSource()==bedit)

{

bedit.setForeground(new Color(0, 0, 105));

bedit.setBackground(new Color(255,255,255));

}

if(m.getSource()==bsave)

{

bsave.setForeground(new Color(0, 0, 105));

bsave.setBackground(new Color(255,255,255));

}

if(m.getSource()==bexit)

{

bexit.setForeground(new Color(0, 0, 105));

bexit.setBackground(new Color(255,255,255));

}

}

public void mouseExited(MouseEvent m)

{

if(m.getSource()==bnew)

{

bnew.setForeground(new Color(255,255,255));

bnew.setBackground(new Color(0, 0, 105));

}

if(m.getSource()==bsearch)

{

bsearch.setForeground(new Color(255,255,255));

bsearch.setBackground(new Color(0, 0, 105));

}

if(m.getSource()==bedit)

{

bedit.setForeground(new Color(255,255,255));

bedit.setBackground(new Color(0, 0, 105));

if(m.getSource()==bsave)

{

bsave.setForeground(new Color(255,255,255));

bsave.setBackground(new Color(0, 0, 105));

}

if(m.getSource()==bexit)

{

bexit.setForeground(new Color(255,255,255));

bexit.setBackground(new Color(0, 0, 105));

}

}

public void mouseMoved(MouseEvent m)

public void mousePressed(MouseEvent m)

public void mouseReleased(MouseEvent m)

{}

public void keyPressed(KeyEvent ke)

{}

public void keyReleased(KeyEvent ke)

{

int i;

if(ke.getSource() == txdoctorname)

{

String dname = txdoctorname.getText();

for(i=0 ; i<dname.length();i++)

{

char ch = dname.charAt(i);

if(ch>='a'& ch<='z'||ch>='A' & ch<='Z'|| ch == ' ')

}

}

}

}

public void keyTyped(KeyEvent ke)

{}

public static void main(String[] args)

{

doctor obj3 = new doctor();

}}

PATIENT FORM

mport java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.sql.\*;

public class patient extends JFrame implements ActionListener,MouseListener,MouseMotionListener,KeyListener

{

SimpleDateFormat sdf = new SimpleDateFormat("dd/MMM/yyyy");

Date dt=new Date();

JLabel lblback = new JLabel(" ")

JLabel lblhead = new JLabel(" ");

JLabel lblpatient = new JLabel("Patient");

JLabel lblback1 = new JLabel("");

JLabel lblback1a = new JLabel("");

JLabel lbltxt1 = new JLabel("Patient Details")

JLabel lblback2 = new JLabel("");

JLabel lblback2a = new JLabel("");

JLabel lblpatientid = new JLabel("Patient id");

JLabel lblpatientname = new JLabel("Patient\_NAME :-");

JLabel lbladress = new JLabel("ADRESS :-");

JLabel lblcontact = new JLabel("CONTACT:- ");

JLabel lbldate = new JLabel("date ");

JLabel lblgender = new JLabel("Gender :- ");

JLabel lblaadhar = new JLabel("UID NUMBER :- ");

JLabel lbldob = new JLabel("DOB ")

JLabel lblm = new JLabel("male ");

JLabel lblf = new JLabel("female ");

JTextField txpatientid = new JTextField();

JTextField txpatientname = new JTextField();

JTextField txadress = new JTextField();

JTextField txcontact = new JTextField();

ButtonGroup bg = new ButtonGroup();

JRadioButton rbmale = new JRadioButton();

JRadioButton rbfemale = new JRadioButton();

Dimension di = Toolkit.getDefaultToolkit().getScreenSize();

JButton bnew = new JButton("New");

JButton bsave = new JButton("Save");

JButton bsearch = new JButton("Search");

JButton bedit = new JButton("Edit");

JButton bexit = new JButton("Exit");

String str1,str2;

public patient()

{

setVisible(true);

setSize(900,600);

setTitle("Patient");

setLayout(null);

setLocationRelativeTo(null);

//head

lblhead.setBounds(00,05,900,60);

lblhead.setBackground(new Color (0,0,105));

lblhead.setOpaque(true);

lblpatient.setBounds(370,5,150,60);

lblpatient.setFont(new Font("Arial",Font.BOLD,18));

lblpatient.setForeground(new Color(255,255,255));

//back1

lblback1.setBounds(100,100,650,350);

lblback1.setBackground(new Color(0,0,105));

lblback1.setOpaque(true);

lblback1a.setBounds(102,102,646,346);

lblback1a.setBackground(new Color(200,206,211));

lblback1a.setOpaque(true);

bexit.setBackground(new Color(0,0,105));

bexit.setForeground(new Color(255,255,255));

//back

lblback.setBounds(00,00,900,630);

lblback.setBackground(new Color(200,206,211));

lblback.setOpaque(true);

txpatientid.setEditable(false);

txdate.setEditable(false);

txpatientname.addKeyListener(this);

txcontact.addKeyListener(this);

txaadhar.addKeyListener(this);

//back1

add(lbltxt1);

add(lblm);

add(lblf);

add(lblback1a);

add(lblback1);

//back2

add(bnew);

add(bsearch);

add(bsave);

add(bedit);

add(bexit);

add(lblback2a);

add(lblback2);

//back

add(lblback);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource() == bnew)

{

try

{

System.out.println("111111111");

Connection con=DriverManager.getConnection("jdbc:odbc:nursing");

String str = "select pat\_id from keytbl";

Statement st=con.createStatement();

ResultSet res=st.executeQuery(str);

System.out.println("22222222222222");

while(res.next())

{

System.out.println("33333333333333");

txpatientid.setText("pat-"+res.getString(1));

}

String tpatid=JOptionPane.showInputDialog(null,"Enter patient id:");

String str = "select \* from pattbl where pat\_id ='"+tpatid+"'";

Statement st=con.createStatement();

ResultSet res=st.executeQuery(str);

while(res.next())

{

txpatientid.setText(res.getString(1));

txpatientname.setText(res.getString(2));

txadress.setText(res.getString(3));

txcontact.setText(res.getString(4));

txaadhar.setText(res.getString(6));

txdate.setText(res.getString(7));

txdob.setText(res.getString(9));

if(res.getString(5).equalsIgnoreCase("Male"))

{

rbmale.setSelected(true);

}

else

{

rbfemale.setSelected(true);

}

txcontact.setText(res.getString(5));

}

}

catch(Exception ee)

{

System.out.print(ee);

ps.setString(9,txdob.getText());

System.out.println("hhvhhfh");

ps.setString(10,"na");

ps.executeUpdate();

String str1 = "update keytbl set pat\_id = pat\_id + 1";

Statement st1 = con.createStatement();

st1.executeUpdate(str1);

JOptionPane.showMessageDialog(null, "Record Saved Successfully");

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

//bnewclearform

if(e.getSource()==bnew)

{

txpatientname.setText("");

txadress.setText("");

txcontact.setText("");

txgender.setText("");

txaadhar.setText("");

bg.clearSelection();

}

if(e.getSource()==bexit)

third obj2 = new third(str1,str2);

this.dispose();

}

}

public void focusGained(FocusEvent fe)

{}

public void mouseClicked(MouseEvent m)

public void mouseDragged(MouseEvent m)

{}

public void mouseEntered(MouseEvent m)

{

if(m.getSource()==bnew)

{

bnew.setForeground(new Color(0, 0, 105));

bnew.setBackground(new Color(255,255,255));

}

if(m.getSource()==bsearch)

{

bsearch.setForeground(new Color(0, 0, 105));

bsearch.setBackground(new Color(255,255,255));

}

if(m.getSource()==bedit)

{

bedit.setForeground(new Color(0, 0, 105));

bedit.setBackground(new Color(255,255,255));

}

if(m.getSource()==bsave)

{

bsave.setForeground(new Color(0, 0, 105));

bsave.setBackground(new Color(255,255,255));

}

if(m.getSource()==bexit)

{

bexit.setForeground(new Color(0, 0, 105));

{}

}

if(ke.getSource() == txaadhar)

{

String puid = txaadhar.getText();

for(i=0 ; i<puid.length();i++)

{

char ch = puid.charAt(i);

if(ch>='0'& ch<='9')

{}

else

{

puid = puid.substring(0,puid.length()-1);

txaadhar.setText(puid);

}

}

}

}

public void keyTyped(KeyEvent ke)

{}

public static void main(String[] args)

{

patient obj4 = new patient();

}

}

**WARD FORM**

import java.awt.\*;

import java.awt.event.\*;

import java.sql.ResultSet;

import java.sql.\*;

import javax.naming.spi.DirStateFactory.Result;

import javax.swing.\*;

import javax.swing.plaf.DimensionUIResource;

import java.text.SimpleDateFormat;

import java.util.Date;

public class ward extends JFrame implements ActionListener,MouseListener,MouseMotionListener

{

JLabel lblback = new JLabel(" ");

JLabel lblhead = new JLabel(" ");

JLabel lblward = new JLabel("Ward");

JLabel lblback1 = new JLabel("");

JLabel lblback1a = new JLabel("");

JLabel lblback2 = new JLabel("");

JLabel lblback2a = new JLabel("");

JLabel lblward\_box = new JLabel("WARD TYPE");

JLabel lblward\_bed = new JLabel("BED");

JLabel lblward\_charge = new JLabel("BED CHARGE");

JTextField txtward\_bed = new JTextField();

JTextField txtward\_charge = new JTextField()

JComboBox wardbox = new JComboBox();

JButton bsave = new JButton("SAVE");

JButton bexit = new JButton("EXIT");

String st1 , st2

public ward()

{

setVisible(true);

setBackground(new Color(211,126,239));

setSize(600,500);

setLocationRelativeTo(null);

setLayout(null);

wardbox.addItem("CCU");

wardbox.addItem("GEN MALE");

wardbox.addItem("GEN FEMALE");

wardbox.addItem("ICU");

//head

lblhead.setBounds(00,05,900,60);

lblhead.setBackground(new Color (0,0,105));

lblhead.setOpaque(true);

(new Color(200,206,211));

lblback1a.setOpaque(true);

lbltxt1.setBounds(70,80,115,40);

lbltxt1.setFont(new Font("Arial",Font.BOLD,18));

lbltxt1.setBackground(new Color(200,206,211));

lbltxt1.setOpaque(true);

lblward\_box.setBounds(180,140,110,30);

lblward\_bed.setBounds(180,240,110,30);

lblward\_charge.setBounds(180,290,110,30);

wardbox.setBounds(280,140,110,30);

txtward\_bed.setBounds(280,240,110,30);

txtward\_charge.setBounds(280,290,110,30);

//back2

lblback2.setBounds(180,370,200,60);

lblback2.setBackground(new Color(0,0,105));

lblback2.setOpaque(true);

lblback2a.setBounds(182,372,196,56);

lblback2a.setBackground(new Color(200,206,211));

lblback2a.setOpaque(true);

bsave.setBounds(200,385,70,30);

bsave.setBackground(new Color(0,0,105));

bsave.setForeground(new Color(255,255,255));

bexit.setBounds(300,384,70,30);

bexit.setBackground(new Color(0,0,105));

bexit.setForeground(new Color(255,255,255));

lblback.setBounds(00,00,600,500);

lblback.setBackground(new Color(200,206,211));

lblback.setOpaque(true);

bexit.addActionListener(this);

bsave.addActionListener(this);

bexit.addMouseListener(this);

bsave.addMouseListener(this);

wardbox.addActionListener(this)

//head

add(lblward);

add(lblhead);

//back1

add(lbltxt1);

add(lblward\_box);

add(lblward\_bed);

add(lblward\_charge);

add(wardbox);

}

if(e.getSource() == bsave)

{

int i;

try

{

Connection con=DriverManager.getConnection("jdbc:odbc:nursing");

Statement st=con.createStatement();

for(i=1;i<=Integer.parseInt(txtward\_bed.getText());i++)

{

st.executeUpdate("update wardtbl set x"+i+"='A' where ward='"+wardbox.getSelectedItem()+"'");

}

for(;i<=15;i++)

{

st.executeUpdate("update wardtbl set x"+i+"='X' where ward='"+wardbox.getSelectedItem()+"'");

}

st.executeUpdate("update wardtbl set Charges='"+ txtward\_charge.getText()+"' where ward='"+wardbox.getSelectedItem()+"'");

JOptionPane.showMessageDialog(null,"Beds Added");

String a = wardbox.getSelectedItem().toString();

int b = Integer.parseInt(txtward\_bed.getText());

}

catch(Exception ee)

{

System.out.print(ee);

}

}

}

public void mouseClicked(MouseEvent m)

{}

public void mouseDragged(MouseEvent m)

{}

public void mouseEntered(MouseEvent m)

{

if(m.getSource()==bsave)

{

bsave.setForeground(new Color(0, 0, 105));

bsave.setBackground(new Color(255,255,255));

}

if(m.getSource()==bexit)

{

bexit.setForeground(new Color(0, 0, 105));

bexit.setBackground(new Color(255,255,255));

}

}

public void mouseExited(MouseEvent m)

{

if(m.getSource()==bsave)

{

bsave.setForeground(new Color(255,255,255));

bsave.setBackground(new Color(0, 0, 105));

}

if(m.getSource()==bexit)

{

bexit.setForeground(new Color(255,255,255));

bexit.setBackground(new Color(0, 0, 105));

}

}

public void mouseMoved(MouseEvent m)

{}

public void mousePressed(MouseEvent m)

{}

public void mouseReleased(MouseEvent m)

{}

public static void main(String[] args)

{

ward objw = new ward();

}

}

**WARDSEL FORM**

import java.awt.\*;

import java.awt.event.\*;

import java.sql.ResultSet;

import java.sql.\*;

import javax.naming.spi.DirStateFactory.Result;

import javax.swing.\*;

import java.text.SimpleDateFormat;

import java.util.Date;

public class wardsel extends JFrame implements ActionListener, ItemListener

{

JButton b[] = new JButton[20];

JButton a = new JButton();

JComboBox wardbox = new JComboBox();

JButton bexit = new JButton("exit");

int xx;

String st1,st2;

JLabel lbl[] = new JLabel[20];

Dimension di = Toolkit.getDefaultToolkit().getScreenSize();

public wardsel( int xy)

{

xx=xy;

SimpleDateFormat sdf = new SimpleDateFormat("dd/MMM/yyyy");

Date dt=new Date();

setVisible(true);

setSize(900,700);

setTitle("BED SELECTION");

setLayout(null);

setSize((int)di.getWidth(),(int)di.getHeight());

wardbox.addItem("ICU");

wardbox.addItem("GENM");

wardbox.addItem("GENF");

wardbox.addItem("CCU");

wardbox.setBounds(500,500,100,60);

bexit.setBounds(300,500,80,40);

add(wardbox);

add(bexit);

wardbox.addActionListener(this);

bexit.addActionListener(this);

repaint();

int x=100;

int y = 100;

for(int j=0 ; j<15 ; j++)

{

b[j] = new JButton();

}

for(int j=0 ; j<15 ; j++)

{

if(x%500==0)

{

x = 100;

y=y+100;

}

b[j].setBounds(x,y,70,70);

public void itemStateChanged(ItemEvent ie)

{

int flag = 0;

int i;

if(ie.getStateChange()==ItemEvent.SELECTED)

{

flag =1;

try

{

int ctr=0;

Connection con=DriverManager.getConnection("jdbc:odbc:nursing");

String str="select \* from wardtbl where ward='"+wardbox.getSelectedItem().toString()+"'";

Statement st=con.createStatement();

ResultSet res=st.executeQuery(str);

while(res.next())

{

for(i=3;i<18;i++)

{

String tstat=res.getString(i);

if(tstat.equalsIgnoreCase("x"))

{

ctr++;

b[i-3].setVisible(false);

b[i-3].setBackground(new Color(255,68,68));

b[i-3].setEnabled(false); }

else

{

if(tstat.equalsIgnoreCase("NA"))

{

b[i-3].setVisible(true);

b[i-3].setBackground(new Color(255,68,68));

b[i-3].setEnabled(false);

}

if(tstat.equalsIgnoreCase("A"))

{

b[i-3].setVisible(true);

b[i-3].setBackground(new Color(125,255,125));

b[i-3].setEnabled(true);

b[i-3].setText(wardbox.getSelectedItem().toString()+" "+(i-2));

}

}

}

for(i=0;i<14-ctr;i++)

{

b[i].setVisible(true);

}

for(i=3;i<18;i++)

{

//System.out.print(i);

b[i].setBackground(new Color(255,68,68));

patient\_details pdobj = new patient\_details(txt);

this.dispose();

}

else

{

String txt = b[i].getText().toString();

b[i].setBackground(new Color(255,68,68));

bedchange bdobj = new bedchange(txt);

this.dispose();

}

}

}

if(e.getSource()==bexit)

{

third obj2 = new third(st1,st2);

this.dispose();

}

public static void main(String[] args) {

wardsel obj2 = new wardsel(10);

}

}

**TEST FORM**

import java.awt.\*;

import java.awt.event.\*;

import java.sql.\*;

import javax.swing.\*;

import org.w3c.dom.events.EventException;

import java.text.SimpleDateFormat;

import java.util.Date;

public class test extends JFrame implements ActionListener,MouseListener,MouseMotionListener

{

JLabel lblback = new JLabel(" ");

JLabel lblhead = new JLabel(" ");

JLabel lbltest = new JLabel("TEST");

JLabel lblback1 = new JLabel("");

JLabel lblback1a = new JLabel("");

JLabel lbltxt1 = new JLabel("Test Details");

JLabel lblback2 = new JLabel("");

JLabel lblback2a = new JLabel("");

JLabel lbltestid = new JLabel("Test id :-");

JLabel lbltestname = new JLabel("Test NAME :-");

JLabel lbltestcharge = new JLabel("testcharge :-");

JLabel lbltestdetails = new JLabel("test Details :-");

JTextField txtestid = new JTextField();

JTextField txtestname = new JTextField();

JTextField txtestcharge = new JTextField();

JTextField txtestdetails = new JTextField();

JButton bnew = new JButton("New");

JButton bnext = new JButton("Next");

JButton bexit = new JButton("Exit");

int ctr;

String st1,st2;

public test()

{

setVisible(true);

setSize(600,500);

setTitle("Test");

setLayout(null);

setLocationRelativeTo(null);

);

lblback1.setBackground(new Color(0,0,105));

lblback1.setOpaque(true);

lblback1a.setBounds(62,102,446,246);

lblback1a.setBackground(new Color(200,206,211));

lblback1a.setOpaque(true);

lbltestid.setBounds(180,140,110,30);

lbltestname.setBounds(180,190,110,30);

lbltestcharge.setBounds(180,240,110,30);

lbltestdetails.setBounds(180,290,110,30);

txtestid.setBounds(280,140,110,30);

txtestname.setBounds(280,190,110,30);

txtestcharge.setBounds(280,240,110,30);

txtestdetails.setBounds(280,290,110,30);

//back2

lblback2.setBounds(110,370,350,60);

lblback2.setBackground(new Color(0,0,105));

lblback2.setOpaque(true);

lblback2a.setBounds(112,372,346,56);

lblback2a.setBackground(new Color(200,206,211));

lblback2a.setOpaque(true);

bnew.setBounds(150,385,70,30);

bnew.setBackground(new Color(0,0,105));

bnew.setForeground(new Color(255,255,255));

bnext.setBounds(250,385,70,30);

bnext.setBackground(new Color(0,0,105));

bnext.setForeground(new Color(255,255,255));

bexit.setBounds(350,384,70,30);

bexit.setBackground(new Color(0,0,105));

bexit.setForeground(new Color(255,255,255));

//back

lblback.setBounds(00,00,600,500);

lblback.setBackground(new Color(200,206,211));

lblback.setOpaque(true);

txtestid.setEditable(false);

bexit.addActionListener(this);

bnext.addActionListener(this);

bnew.addActionListener(this);

add(lbltestid);

add(lbltestname);

add(lbltestcharge);

add(lbltestdetails);

add(txtestid);

add(txtestname);

add(txtestcharge);

add(txtestdetails);

add(lblback1a);

add(lblback1);

//back2

add(bnew);

add(bnext);

add(bexit)

add(lblback2a);

add(lblback2);

//back

add(lblback);

repaint();

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource() == bnew)

{

try

{

System.out.println("111111111");

Connection con=DriverManager.getConnection("jdbc:odbc:nursing");

String str = "select tst\_id from keytbl";

Statement st=con.createStatement();

ResultSet res=st.executeQuery(str);

System.out.println("22222222222222");

while(res.next())

{

System.out.println("33333333333333");

txtestid.setText("tst/"+res.getString(1));

}

System.out.println("4444444");

}

catch(Exception ee)

{

System.out.print(ee);

}

public void mouseExited(MouseEvent m)

{

if(m.getSource()==bnew)

{

bnew.setForeground(new Color(255,255,255));

bnew.setBackground(new Color(0, 0, 105));

}

if(m.getSource()==bnext)

{

bnext.setForeground(new Color(255,255,255));

bnext.setBackground(new Color(0, 0, 105));

}

if(m.getSource()==bexit)

{

bexit.setForeground(new Color(255,255,255));

bexit.setBackground(new Color(0, 0, 105));

{}

public static void main(String[] args)

{

test obj5 = new test();

}

}

**TEST SUBMIT FORM**

import java.awt.\*;

import java.awt.event.\*;

import java.sql.\*;

import javax.swing.\*;

import java.text.SimpleDateFormat;

import java.util.Date;

public class test\_parameters extends JFrame implements ActionListener,KeyListener,MouseListener,MouseMotionListener

{

JLabel lblback = new JLabel(" ");

JLabel label1 = new JLabel("");

JLabel label1a = new JLabel("");

JLabel label2 = new JLabel("");

JLabel label2a = new JLabel("");

JLabel lbldiv = new JLabel("");

JLabel lblparaname = new JLabel("PARAMETERS NAME",JLabel.CENTER);

JLabel lblparavalue = new JLabel("DEFAULT VALUE",JLabel.CENTER);

JLabel lblhead = new JLabel(" ");

JLabel lbltestdetail = new JLabel("TEST PARAMETERS ");

JLabel lbltestid = new JLabel("Test id :-");

JLabel lbltestname = new JLabel("Test NAME :-");

JLabel lbltestcharge = new JLabel("testcharge :-");

JLabel lblparameter = new JLabel("Parameter Count:-");

JTextField txtestid = new JTextField();

JTextField txtestname = new JTextField();

JTextField txtestcharge = new JTextField();

JTextField txparameter = new JTextField();

JButton bsave = new JButton("Save");

JButton bexit = new JButton("Exit");

JLabel lblpara[] = new JLabel[20];

JTextField txtparaname[] = new JTextField[7];

JTextField txtparavalue[] = new JTextField[7];

int testpara = 0;

String st1,st2;

public test\_parameters(String a,String b,String c,Integer d)

{

setVisible(true);

setSize(800,730);

setTitle("Test");

setLayout(null);

setLocationRelativeTo(null);

String testid = a;

String testname = b;

String testcharge = c;

testpara = d;

String testparastr = String.valueOf(testpara);

//back

lblback.setBounds(00,00,800,900);

lblback.setBackground(new Color(200,206,211));

lblback.setOpaque(true);

//head

lblhead.setBounds(00,05,900,60);

lblhead.setBackground(new Color (0,0,105));

lblhead.setOpaque(true);

lbltestdetail.setBounds(290,5,800,60);

lbltestdetail.setFont(new Font("Arial",Font.BOLD,18));

lbltestdetail.setForeground(new Color(255,255,255));

5, txtparaname[0].getText());

ps.setString(6, txtparaname[1].getText());

ps.setString(7, txtparaname[2].getText());

ps.setString(8, txtparaname[3].getText());

ps.setString(9, txtparaname[4].getText());

ps.setString(10, txtparaname[5].getText());

ps.setString(11, txtparaname[6].getText());

ps.executeUpdate();

JOptionPane.showMessageDialog(null, "Record Saved Successfully");

}

catch(Exception ee)

{

System.out.println(ee);

}

try

{

Connection con1=DriverManager.getConnection("jdbc:odbc:nursing");

String str1 = "INSERT INTO testvaluetbl (test\_id,test\_name,nos,value1,value2,value3,value4,value5,value6,value7)values(?,?,?,?,?,?,?,?,?,?)";

PreparedStatement ps1 = con1.prepareStatement(str1);

ps1.setString(1, txtestid.getText());

ps1.setString(2, txtestname.getText());

ps1.setString(3, txparameter.getText());

ps1.setString(4, txtparavalue[0].getText());

ps1.setString(5, txtparavalue[1].getText());

ps1.setString(6, txtparavalue[2].getText());

ps1.setString(7, txtparavalue[3].getText());

ps1.setString(8, txtparavalue[4].getText());

ps1.setString(9, txtparavalue[5].getText());

ps1.setString(10, txtparavalue[6].getText());

ps1.executeUpdate();

JOptionPane.showMessageDialog(null, "Record Saved Successfully");

}

catch(Exception ee)

{

System.out.println(ee);

}

}

else

{

JOptionPane.showMessageDialog(null, "Please fill all fields!!!");

}

}

if(ae.getSource()==bexit)

{

third obj2 = new third(st1,st2);

this.dispose();

}

}

public void focusGained(FocusEvent fe)

{}

public void focusLost(FocusEvent fe)

{}

public void mouseClicked(MouseEvent m)

{}

public void mouseDragged(MouseEvent m)

{}

bexit.setForeground(new Color(0, 0, 105));

bexit.setBackground(new Color(255,255,255));

}

}

public void mouseExited(MouseEvent m)

{

if(m.getSource()==bsave)

{

bsave.setForeground(new Color(255,255,255));

bsave.setBackground(new Color(0, 0, 105));

}

if(m.getSource()==bexit)

{

bexit.setForeground(new Color(255,255,255));

bexit.setBackground(new Color(0, 0, 105));

}

{

pat\_contact = pat\_contact.substring(0, pat\_contact.length()-1);

txtparaname[j].setText(pat\_contact);

}

}

}

}

for(int j=0 ; j<7 ; j++)

{

if(ke.getSource() == txtparavalue[j])

{

String pat\_contact = txtparavalue[j].getText();

for(int i=0 ; i<pat\_contact.length() ; i++)

{

else

{

pat\_contact = pat\_contact.substring(0, pat\_contact.length()-1);

txtparavalue[j].setText(pat\_contact);

}

}

}

}

}

public static void main(String[] args)

{

test\_parameters obj4 = new test\_parameters("aa","bb","cc",7);

}

}

**PATIENT DETAILS FORM**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.sql.\*;

public class patient\_details extends JFrame implements ActionListener,MouseListener,MouseMotionListener

{

SimpleDateFormat sdf = new SimpleDateFormat("dd/MMM/yyyy");

Date dt=new Date();

SimpleDateFormat sdf1 = new SimpleDateFormat("dd/MMM/yyyy");

Date dt1 = new Date();

JLabel lblback = new JLabel(" ");

JLabel lblhead = new JLabel(" ");

JLabel lbladmit = new JLabel("Admit");

JLabel lblback1 = new JLabel("");

JLabel lblback1a = new JLabel("");

JLabel lbltxt1 = new JLabel("Patient Details");

JLabel lblback2 = new JLabel("");

JLabel lblback2a = new JLabel("");

JLabel lbladmission = new JLabel("Admission.id");

JLabel lblpatientid = new JLabel("Patient id");

JLabel lblpatientname = new JLabel("Patient\_NAME :-");

JLabel lbladress = new JLabel("ADRESS :-");

JLabel lblcontact = new JLabel("CONTACT:- ");

JLabel lbldate = new JLabel("DATE");

JLabel lblgender = new JLabel("Gender :- ");

JLabel lblaadhar = new JLabel("UID NUMBER :- ");

JLabel lblbed = new JLabel("BED allocation :- ")

JLabel lblm = new JLabel("male ");

JLabel lblf = new JLabel("female ")

JTextField txadmission = new JTextField();

JTextField txpatientid = new JTextField();

JTextField txpatientname = new JTextField();

JTextField txadress = new JTextField();

JTextField txcontact = new JTextField();

JTextField txdate = new JTextField();

JTextField txgender = new JTextField();

JTextField txaadhar = new JTextField();

JTextField txbed = new JTextField()

ButtonGroup bg = new ButtonGroup();

JRadioButton rbmale = new JRadioButton();

JRadioButton rbfemale = new JRadioButton();

JButton bnew = new JButton("New");

JButton bsave = new JButton("Save");

JButton bsearch = new JButton("Search");

JButton bedit = new JButton("Edit");

JButton bexit = new JButton("Exit");

Dimension di = Toolkit.getDefaultToolkit().getScreenSize();

String str1,str2;

public patient\_details(String txt)

{

setVisible(true);

setSize(900,600);

setTitle("Admit");

setLayout(null);

setLocationRelativeTo(null);

bnew.addActionListener(this);

bsave.addActionListener(this);

bsearch.addActionListener(this);

bedit.addActionListener(this);

bexit.addActionListener(this);

bnew.addMouseListener(this);

bsave.addMouseListener(this);

bsearch.addMouseListener(this);

bedit.addMouseListener(this);

bexit.addMouseListener(this);

txdate.setText(sdf.format(dt));

//lblhead

add(lbladmit);

add(lblhead);

//back1

add(lbltxt1);

add(lblpatientid);

add(lbladress);

add(lblcontact);

add(lbldate);

add(lbladmission);

add(txadmission);

add(lblpatientname);

add(lblgender);

add(lblbed);

add(lblaadhar);

add(txpatientid);

add(txadress);

add(txcontact);

add(txdate);

add(txpatientname);

add(txaadhar);

add(txbed);

bg.add(rbmale);

bg.add(rbfemale);

add(rbmale);

add(rbfemale);

add(lblm);

add(lblf);

add(lblback1a);

add(lblback1);

//back2

add(bnew);

add(bsearch);

add(bsave);

add(bedit);

add(bexit);

public void actionPerformed(ActionEvent e)

{

if(e.getSource() == bnew)

{

String dat=txdate.getText();

char tdat = dat.charAt(dat.length()-1);

String dt[]=dat.split("/");

String yr = dt[2];

System.out.println(ee);

}

}

{

bexit.setForeground(new Color(0, 0, 105));

bexit.setBackground(new Color(255,255,255));

}

}

public void mouseExited(MouseEvent m)

{

if(m.getSource()==bnew)

{

bnew.setForeground(new Color(255,255,255));

bnew.setBackground(new Color(0, 0, 105));

}

if(m.getSource()==bsearch)

{

bsearch.setForeground(new Color(255,255,255));

bsearch.setBackground(new Color(0, 0, 105));

}

if(m.getSource()==bedit)

{

bedit.setForeground(new Color(255,255,255));

bedit.setBackground(new Color(0, 0, 105));

}

if(m.getSource()==bsave)

{

bsave.setForeground(new Color(255,255,255));

bsave.setBackground(new Color(0, 0, 105));

}

if(m.getSource()==bexit)

{

bexit.setForeground(new Color(255,255,255));

bexit.setBackground(new Color(0, 0, 105));

}

}

public static void main(String[] args)

{

patient\_details pdobj = new patient\_details("eeee");

}

}

**PATIENT BED DETAILS FORM**

import java.awt.\*;

import java.awt.event.\*;

import java.lang.Character.Subset;

import java.sql.ResultSet;

import java.sql.\*;

import javax.naming.NameNotFoundException;

import javax.naming.spi.DirStateFactory.Result;

import javax.sound.sampled.FloatControl.Type;

import javax.swing.\*;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.math.\*;

import javax.swing.table.DefaultTableModel;

import java.util.\*;

public class patient\_bed\_details extends JFrame implements ActionListener,MouseListener,MouseMotionListener,FocusListener

{

JLabel lblback = new JLabel(" ");

JLabel lblhead = new JLabel(" ");

JLabel lblbeddetail = new JLabel("PATIENT BED RECORDS");

JLabel lblback1 = new JLabel("");

JLabel lblback1a = new JLabel("");

JLabel lbltxt1 = new JLabel("Bed Records");

JLabel lblback2 = new JLabel("");

JLabel lblback2a = new JLabel("");

JLabel lblpat\_id = new JLabel("PATIENT ID - ");

JTextField txtpat\_id = new JTextField();

//back1

lblback1.setBounds(60,100,650,380);

lblback1.setBackground(new Color(0,0,105));

lblback1.setOpaque(true);

lblback1a.setBounds(62,102,646,376);

lblback1a.setBackground(new Color(200,206,211));

lblback1a.setOpaque(true)

lbltxt1.setBounds(70,80,115,40);

lbltxt1.setFont(new Font("Arial",Font.BOLD,18));

lbltxt1.setBackground(new Color(200,206,211));

lbltxt1.setOpaque(true);

lblpat\_id.setBounds(280,110,120,30);

lblpat\_id.setFont(new Font("Arial",Font.BOLD,14));

txtpat\_id.setBounds(380,110,80,30);

txtpat\_id.setFont(new Font("Arial",Font.BOLD,14));

//table

scb=new JScrollPane(jtblbed);

scb.setBounds(120,160,525 ,290);

jtblbed.setModel(bmodel);

add(scb);

jtblbed.getTableHeader().setFont(new Font("SansSerif",Font.BOLD,14));

jtblbed.setFont(new Font("Arial",Font.BOLD,18));

jtblbed.setForeground(new Color(0,0,108));

jtblbed.setRowHeight(25);

jtblbed.setAutoResizeMode(JTable.AUTO\_RESIZE\_OFF);

jtblbed.getColumnModel().getColumn(0).setPreferredWidth(165);

public void focusGained(FocusEvent fe)

{}

public void focusLost(FocusEvent fe)

{

if(fe.getSource() == txtpat\_id)

{

int count = 0;

try

{

bmodel.setRowCount(0);

Connection con2=DriverManager.getConnection("jdbc:odbc:nursing");

String str2="select \* from patientbedtbl where patient\_id ='"+txtpat\_id.getText()+"'";

Statement st2=con2.createStatement();

ResultSet res2=st2.executeQuery(str2);

while(res2.next())

{

count++;

Vector v = new Vector();

v.add(res2.getString(1));

v.add(res2.getString(5));

v.add(res2.getString(7));

v.add(res2.getString(6));

bmodel.addRow(v);

}

if(count == 0)

{

JOptionPane.showMessageDialog(null,"Patient Not Found!!!");

}

}

catch(Exception ee)

{

System.out.println(ee);

}

}

}

public void mouseClicked(MouseEvent m)

{}

public void mouseDragged(MouseEvent m)

{}

public void mouseEntered(MouseEvent m)

{

if(m.getSource()==bexit)

{

bexit.setForeground(new Color(0, 0, 105));

bexit.setBackground(new Color(255,255,255));

}

}

public void mouseExited(MouseEvent m)

{

if(m.getSource()==bexit)

{

bexit.setForeground(new Color(255,255,255));

bexit.setBackground(new Color(0, 0, 105));

}

}

public void mousePressed(MouseEvent m)

{}

public void mouseReleased(MouseEvent m)

{}

public static void main(String[] args)

{

patient\_bed\_details objj = new patient\_bed\_details("p/2");

}

}

**PATIENT TEST RECORD FORM**

import java.awt.\*;

import java.awt.event.\*;

import java.lang.Character.Subset;

import java.sql.ResultSet;

import java.sql.\*;

import javax.naming.NameNotFoundException;

import javax.naming.spi.DirStateFactory.Result;

import javax.sound.sampled.FloatControl.Type;

import javax.swing.\*;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.math.\*;

import javax.swing.table.DefaultTableModel;

import java.util.\*;

public class patient\_test\_record extends JFrame implements ActionListener,MouseListener,MouseMotionListener,FocusListener

{

JLabel lblback = new JLabel(" ");

JLabel lblhead = new JLabel(" ");

JLabel lbltestdetail = new JLabel("PATIENT TEST RECORDS");

JLabel lblback1 = new JLabel("");

JLabel lblback1a = new JLabel("");

JLabel lbltxt1 = new JLabel("Test Records");

JLabel lblback2 = new JLabel("");

JLabel lblback2a = new JLabel("");

JLabel lblpat\_id = new JLabel("PATIENT ID - ");

JTextField txtpat\_id = new JTextField();

JButton bexit = new JButton("Exit");

String[] col= {"Test ReqId","Test Name","Test Id","Test Charge","Test Date"};

String test\_reqid,ttid,ttname,ttcharge,test\_date;

Object[][] data = {{test\_reqid,ttid,ttname,ttcharge,test\_date}};

DefaultTableModel model = new DefaultTableModel(col,0);

JTable jtbltest = new JTable(data,col);

JScrollPane sct;

String pat\_id = "";

String st1 , st2;

public patient\_test\_record(String patient\_id)

{

setVisible(true);

setSize(1000,600);

setTitle("TEST RECORDS");

setLayout(null);

setLocationRelativeTo(null);

//head

lblhead.setBounds(00,05,1000,60);

lblhead.setBackground(new Color (0,0,105));

lblhead.setOpaque(true);

lbltestdetail.setBounds(350,5,250,60);

lbltestdetail.setFont(new Font("Arial",Font.BOLD,18));

lbltestdetail.setForeground(new Color(255,255,255));

//back1

lblback1.setBounds(50,100,850,380);

lblback1.setBackground(new Color(0,0,105));

lblback1.setOpaque(true);

lblback1a.setBounds(52,102,846,376);

lblback1a.setBackground(new Color(200,206,211));

lblback1a.setOpaque(true);

lbltxt1.setBounds(70,80,123,40);

lbltxt1.setFont(new Font("Arial",Font.BOLD,18));

lbltxt1.setBackground(new Color(200,206,211));

lbltxt1.setOpaque(true);