DELHI PUBLIC SCHOOL

JANJGIR-CHAMPA

SESSION-2018-2019

INFORMATICS PRACTICES INVESTIGATORY PROJECT

On

HOSPITAL MANAGEMENT

{ E – GOVERNANCE }

AISSCE-2018-19 (Main) EXAMINATION

[As a part of the Informatics Practices Course (065)]

SUBMITTED BY	GUIDED BY
PAPPU BISWAS	STEVE SAMSON
Roll No – 6657724	PGT- INFORMATICS PRACTICES
	Delhi Public School
	Janjgir – Champa

DELHI PUBLIC SCHOOL

JANJGIR-CHAMPA

SESSION-2018-2019

***** " *CERTIFICATE*" *****

****

This is to certify that Pappu Biswas   Student of XII-Science has successfully completed the I.P. Investigatory project on "**HOSPITAL MANAGEMENT *{ E – GOVERNANCE }*** " under my supervision in the Partial fulfillment of I.P. Practical Examination conducted by AISSCE New Delhi.

____

Principal signature

STEVE SAMSON
(Internal Examiner)

_ _ _ _ _ _ _ _ _ _ _ _
(External Examiner)

***** ACKNOWLEDGEMENT *****

In the accomplishment of this project successfully, many people have best owned upon me their blessings and heart pledged support, this time I am utilizing to thank all the people who have been concerned with this project.

Primarily I would thank god for being able to complete this project with success. Then I would like to thank to our Principal Mrs. Kalpana Singh and I.P. teacher Mr. Steve Samson whose valuable guidance has been the ones that helped me patch this project make it full proof success. Their suggestions and instructions have served as the major contributor towards the contribution of project.

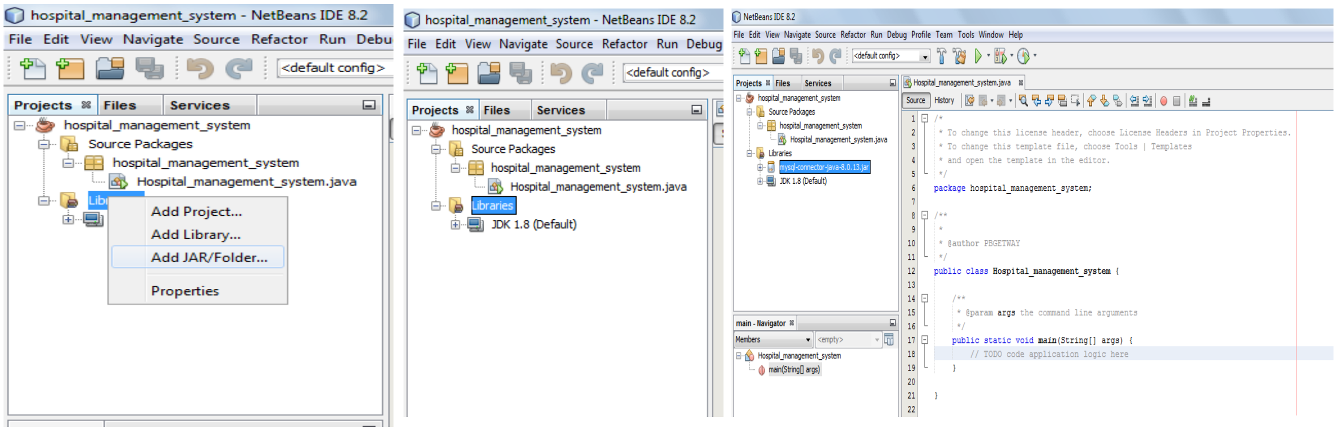
I would also like to thank our parents and friends who have helped me with their valuable suggestions and guidance have been very helpful in various phases of completion of the project . Last but not the least I would like to thanks our classmates who have helped me a lot.

Pappu Biswas

Std.- XII-Science

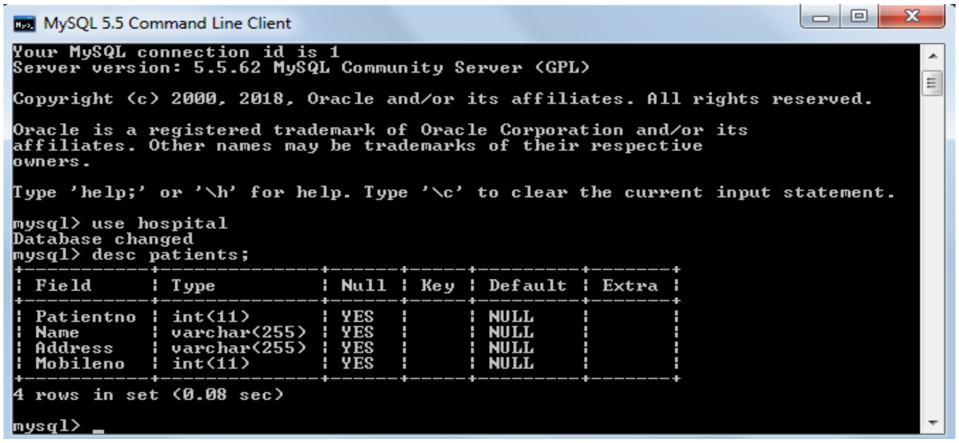
CONTENTS

LIST OF FIGURES


hospital_management_system - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Projects Files Services
hospital_management_system
Source Packages
hospital_management_system
Hospital_management_system.java
Libraries
Add Project...
Add Library...
Add JAR/Folder...
Properties
mysql-connector-java-8.0.13.jar
JDK 1.8 (Default)
Hospital_management_system.java
Source History
main - Navigator
Members
<empty>
main(String[] args)

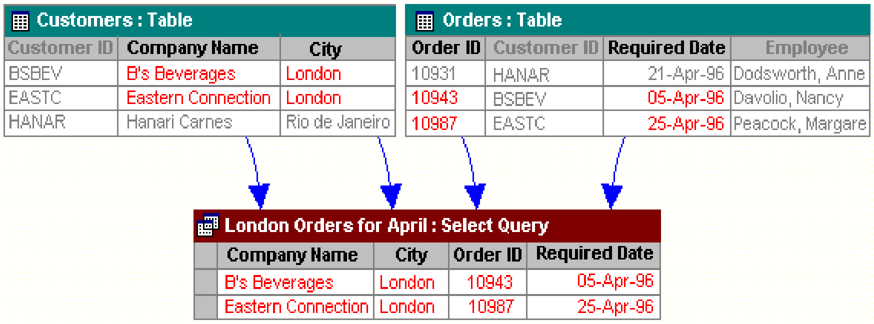
```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package hospital_management_system;

/**
 *
 * @author PBGETWAY
 */
public class Hospital_management_system {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```

Fig 1 :- JDBC driver connections procedure



```
Your MySQL connection id is 1
Server version: 5.5.62 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use hospital
Database changed
mysql> desc patients;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| Patientno  | int(11)      | YES  |     | NULL    |       |
| Name       | varchar(255) | YES  |     | NULL    |       |
| Address    | varchar(255) | YES  |     | NULL    |       |
| Mobileno   | int(11)      | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
4 rows in set (0.08 sec)

mysql> _
```

Fig 2 : - MySQL Command Client



Customers : Table

Customer ID	Company Name	City
BSBEV	B's Beverages	London
EASTC	Eastern Connection	London
HANAR	Hanari Carnes	Rio de Janeiro

Orders : Table

Order ID	Customer ID	Required Date	Employee
10931	HANAR	21-Apr-96	Dodsworth, Anne
10943	BSBEV	05-Apr-96	Davolio, Nancy
10987	EASTC	25-Apr-96	Peacock, Margare

London Orders for April : Select Query

Company Name	City	Order ID	Required Date
B's Beverages	London	10943	05-Apr-96
Eastern Connection	London	10987	25-Apr-96

This query retrieves the company name, city, order ID,
and required date information for customers in London
whose orders were required in April.

Fig 3 : - Table designs sample showing reference .

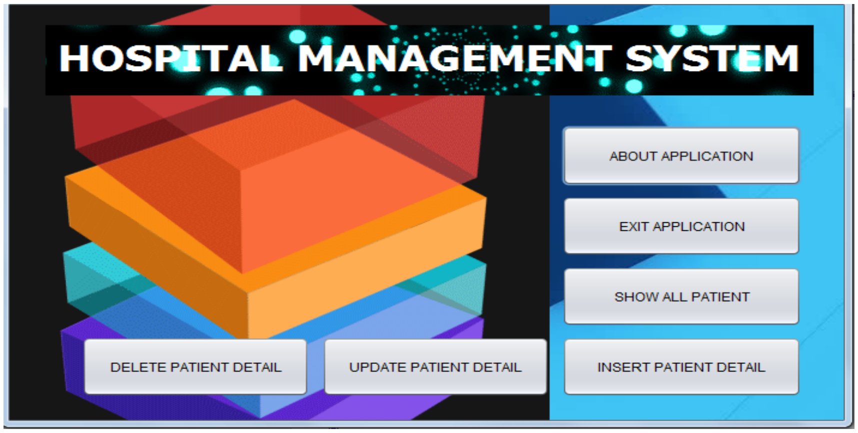
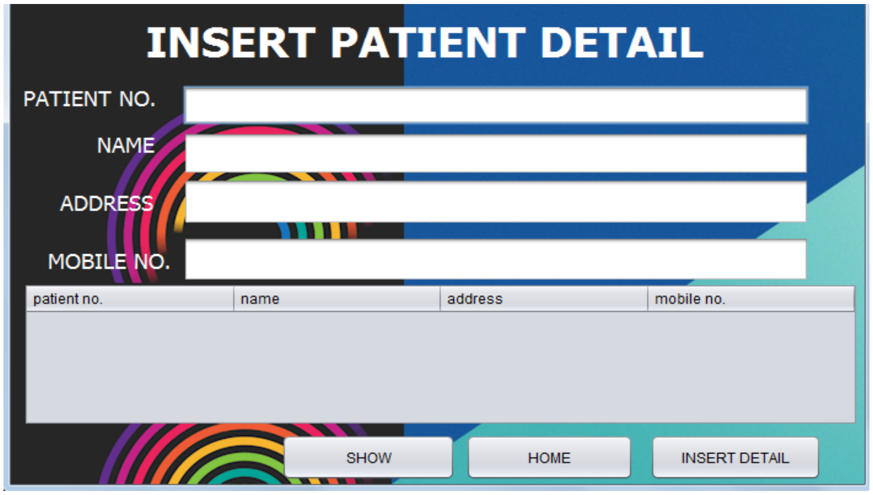
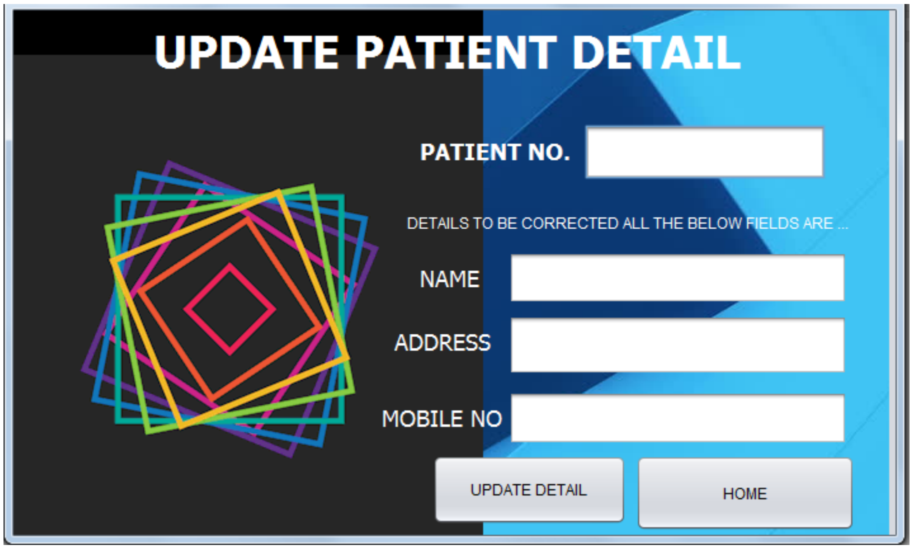


Fig 4 : - Home main page of the software .



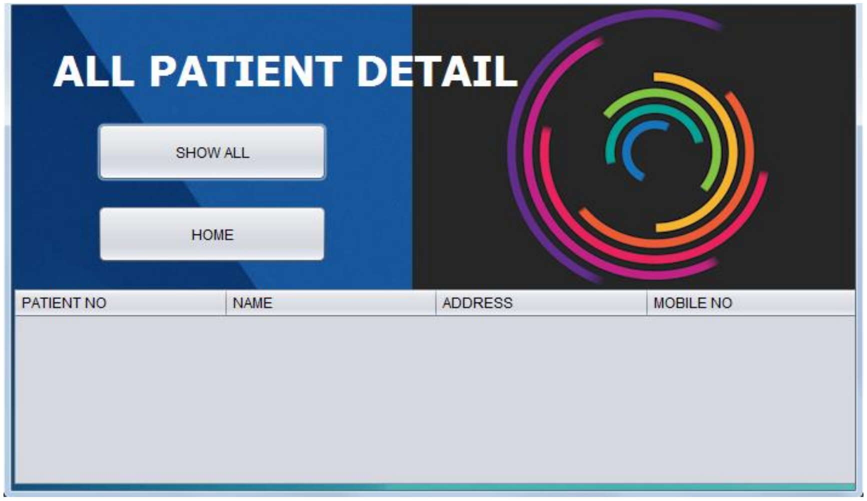
patient no.	name	address	mobile no.

Fig 5 : - Insert patient detail page of the software .



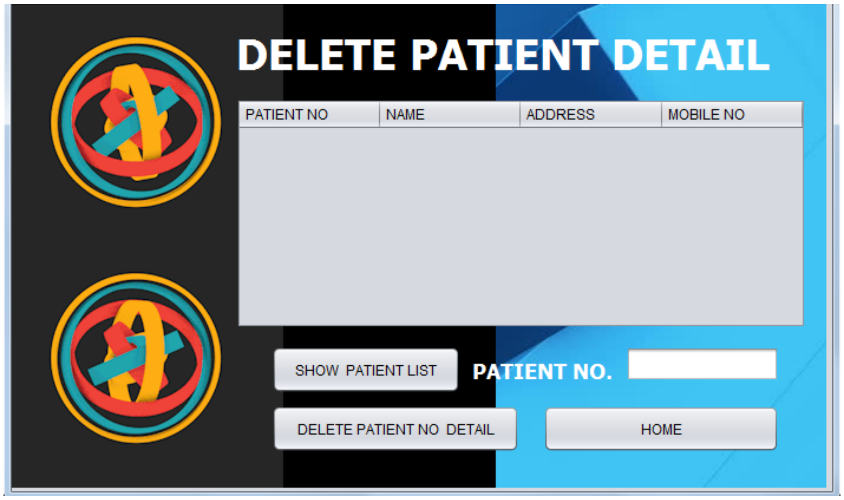
DETAILS TO BE CORRECTED ALL THE BELOW FIELDS ARE ...

Fig 6 : - update patient detail page of the software .



PATIENT NO	NAME	ADDRESS	MOBILE NO

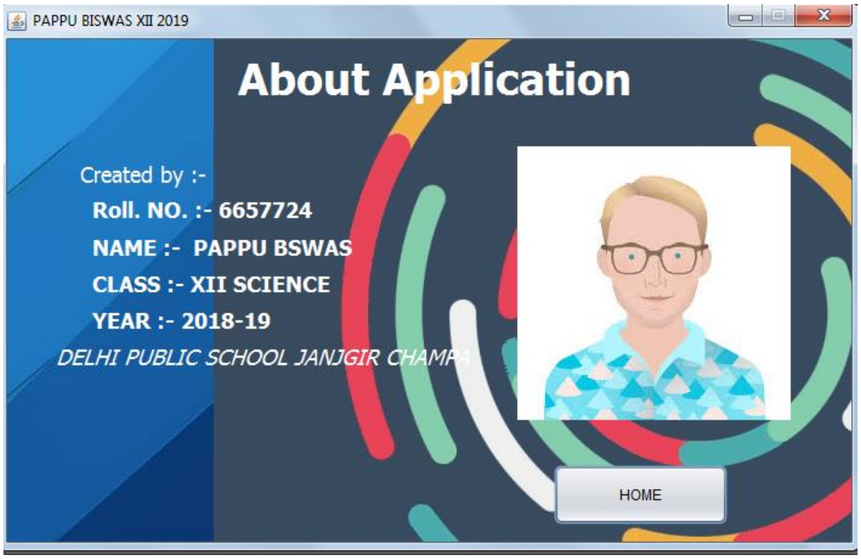
Fig 7 : - all patient detail page of the software .



PATIENT NO	NAME	ADDRESS	MOBILE NO

PATIENT NO.

Fig 8 : - delete patient detail page of the software .



Created by :-

Roll. NO. :- 6657724

NAME :- PAPPU BSWAS

CLASS :- XII SCIENCE

YEAR :- 2018-19

DELHI PUBLIC SCHOOL JANJGIR CHAMPA

Fig 9 : - About Application page of the software .

INTRODUCTION

Computer operators who store and handle the database which stores all necessary information concerning the hospital. Which is much needed is a proper interface which may enable us to handle such systems efficiently.

This project entitled "HOSPITAL MANAGEMENT SYSTEM" has been developed with an aim to help one such a hospital to easily manage and handle the data concerning it.

There are mainly four types of actions performed by this project:-

- Retrive :-There are dedicated tables made for serving the purpose of viewing the details of paitents existing in a hospital database at a particular time.
- Insert :- Data such as names, address ,mobile no and paitent no. can be inserted easily.
- Update :- We can simply update anything whether it is address, name or mobile no. .
- Delete :- We can simply delete the data regarding our paitents.

A Management Information System (MIS) mainly consists of a computerized database , a collection of inter-related tables for a particular subject or purpose, capable to produce different reports relevant to the user. An application program is tied with the database for easy access and interface to the database. Using Application program or front-end, we can store, retrieve and manage all information in proper way.

This software, being simple in design and working, does not require much of training to users, and can be used as a powerful tool for automating a Hospital Management System.

During coding and design of the software Project, Java NetBeans IDE, a powerful front-end tool is used for getting Graphical User Interface (GUI) based integrated platform and coding simplicity. As a back-end a powerful, open source RDBMS, MySQL is used .

Objective and Scope

The objective of the software project is to develop a computerized MIS to automate the functions of a hospital . This software project is also aimed to enhance the current record keeping system , which will help managers to retrieve the up-to-date information at right time in right shape.

The proposed software system is expected to do the following functionality-

- ✓ **To provide a user friendly, Graphical User Interface (GUI) based integrated and centralized environment for MIS activities.**
- ✓ **The proposed system should maintain all the records and transactions, and should generate the required reports and information when required.**
- ✓ **To provide graphical and user-friendly interface to interact with a centralized database based on client-server architecture.**
- ✓ **To identify the critical operation procedure and possibilities of simplification using modern IT tools and practices.**

In its current scope, the software enables user to retrieve and update the information from centralized database designed with MySQL . This software does not require much training time of the users due to limited functionality and simplicity. During the development of Hospital Management System project, Java NetBeans IDE, a powerful, open source event-driven form-based development environment is used for modular design and future expandability of the system. Despite of the best effort of the developer, the following limitations and functional boundaries are visible, which limits the scope of this application software.

1. **This software can store records and produce reports in pre-designed format in soft copy. There is no facility yet to produce customized reports. Only specified reports are covered.**
2. **There is no provision to calculate fine or penalty etc.**

So far as future scope of the project is concerned, firstly it is open to any modular expansion i.e. other modules or functions can be designed and embedded to handle the user need in future. Any part of the software and reports can be modified independently without much effort.

Theoretical Background

3.1 What is Database?

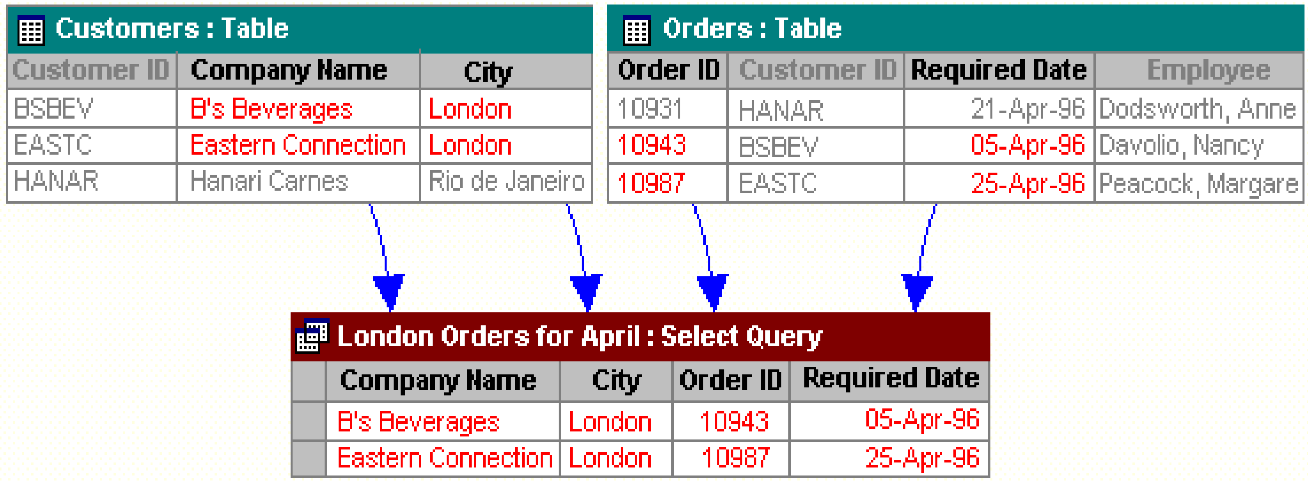
Introduction and Concepts:

A database is a collection of information related to a particular subject or purpose, such as tracking customer orders or maintaining a music collection. Using any RDBMS application software like MS SQL Server, MySQL, Oracle, Sybase etc. , you can manage all your information from a single database file. Within the file, divide your data into separate storage containers called tables. You may and retrieve the data using queries.

A table is a collection of data about a specific topic, such as products or suppliers. Using a separate table for each topic means you can store that data only once, which makes your database more efficient and reduces data-entry errors. Table organises data into columns (called fields) and rows (called records).

A Primary key is one or more fields whose value or values uniquely identify each record in a table. In a relationship, a primary key is used to refer to specific record in one table from another table. A primary key is called foreign key when it is referred to from another table.

To find and retrieve just the data that meets conditions you specify, including data from multiple tables, create a query. A query can also update or delete multiple records at the same time, and perform built-in or custom calculations on your data.



Customers : Table

Customer ID	Company Name	City
BSBEV	B's Beverages	London
EASTC	Eastern Connection	London
HANAR	Hanari Carnes	Rio de Janeiro

Orders : Table

Order ID	Customer ID	Required Date	Employee
10931	HANAR	21-Apr-96	Dodsworth, Anne
10943	BSBEV	05-Apr-96	Davolio, Nancy
10987	EASTC	25-Apr-96	Peacock, Margare

London Orders for April : Select Query

Company Name	City	Order ID	Required Date
B's Beverages	London	10943	05-Apr-96
Eastern Connection	London	10987	25-Apr-96

This query retrieves the company name, city, order ID,
and required date information for customers in London
whose orders were required in April.

Role of RDBMS Application Program:

A computer database works as a electronic filing system, which has a large number of ways of cross-referencing, and this allows the user many different ways in which to re-organize and retrieve data. A database can handle business inventory, accounting and filing and use the information in its files to prepare summaries, estimates and other reports. The management of data in a database system is done by means of a general-purpose software package called a Database Management System (DBMS). Some commercially available DBMS are MS SQL Server, MS ACCESS, INGRES, ORACLE, and Sybase. A database management system, therefore, is a combination of hardware and software that can be used to set up and monitor a database, and can manage the updating and retrieval of database that has been stored in it. Most of the database management systems have the following capabilities:

- Creating of a table, addition, deletion, modification of records.
- Retrieving data collectively or selectively.
- The data stored can be sorted or indexed at the user's discretion and direction.
- Various reports can be produced from the system. These may be either standardized report or that may be specifically generated according to specific user definition.
- Mathematical functions can be performed and the data stored in the database can be manipulated with these functions to perform the desired calculations.
- To maintain data integrity and database use.

The DBMS interprets and processes users' requests to retrieve information from a database. In most cases, a query request will have to penetrate several layers of software in the DBMS and operating system before the physical database can be accessed. The DBMS responds to a query by invoking the appropriate subprograms, each of which performs its special function to interpret the query, or to locate the desired data in the database and present it in the desired order.

3.2 What is My SQL ?

The management of data in a database system is done by means of a general-purpose software package called a Database Management System (DBMS). Some commercially available RDBMS are MS SQL Server, MS ACCESS, INGRES, ORACLE, and Sybase.

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. MySQL is named after co-founder Monty Widenius's daughter, My. The name of the MySQL Dolphin (our logo) is Sakila .

- **MySQL is a database management system.**

  A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL is based on SQL.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of MySQL stands for Structured Query Language. SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. MySQL software is Open Source.

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License).

- **The MySQL Database Server is very fast, reliable, and easy to use.**

Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

- **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends , several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

The Main Features of MySQL

- **Written in C and C++.**
- **Works on many different platforms.**
- **Uses multi-layered server design with independent modules . Designed to make it relatively easy to add other storage engines. This is useful if you want to provide an SQL interface for an in-house database.**
- **Uses a very fast thread-based memory allocation system.**
- **Executes very fast joins using an optimized nested-loop join.**
- **Implements SQL functions using a highly optimized class library that should be as fast as possible. Usually there is no memory allocation at all after query initialization.**
- **Provides the server as a separate program for use in a client/server networked environment, and as a library that can be embedded (linked) into standalone applications. Such applications can be used in isolation or in environments where no network is available.**
- **Password security by encryption of all password traffic when you connect to a server.**
- **Support for large databases. We use MySQL Server with databases that contain 50 million records. We also know of users who use MySQL Server with 200,000 tables and about 5,000,000,000 rows.**

- **The Connector/ODBC (MyODBC) interface provides MySQL support for client programs that use ODBC (Open Database Connectivity) connections.**
- **The Connector/J interface provides MySQL support for Java client programs that use JDBC connections. Clients can be run on Windows or Unix. Connector/J source is available.**

What is NetBeans IDE ?

NetBeans started as a student project (originally called Xelfi) in the Czech Republic in 1996. The goal was to write a Delphi-like Java IDE in Java. Xelfi was the first Java IDE (Integrated Development Environment) written in Java, with its first pre-releases in 1997. Xelfi was a fun project to work on, especially since Java IDE space was uncharted territory at that time. The project attracted enough interest that these students, once they graduated, decided that they could market it as a commercial product. Soliciting resources from friends and relatives for a web space, they formed a company around it.

In the spring of 1999, NetBeans DeveloperX2 was released, supporting Swing. The performance improvements that came in JDK 1.3, released in the fall of 1999, made NetBeans a viable choice for development tools. By the summer of 1999, the team was hard at work re-architecting DeveloperX2 into the more modular NetBeans that forms the basis of the software today.

Something else was a foot in the summer of 1999: Sun Microsystems wanted better Java development tools, and had become interested in NetBeans. It was a dream come true for the NetBeans team: NetBeans would become the flagship tool set of the maker of Java itself! By the Fall, with the next generation of NetBeans Developer in beta, a deal was struck. Sun Microsystems had also acquired another tools company, During the a question , the young developers who had been involved in open-source projects for most of their programming careers, mentioned the idea of open-sourcing NetBeans. Fast forward to less than six months later, the decision was made that NetBeans would be open sourced. While Sun had contributed considerable amounts of code to open source projects over the years, this was Sun's first *sponsored* open source project, one in which Sun would be paying for the site and handling the infrastructure.

Features of NetBeans

A free, open-source Integrated Development Environment for software developers. You get all the tools you need to create professional desktop, enterprise, web, and mobile applications with the Java platform, as well as C/C++, PHP, JavaScript, Groovy, and Ruby. NetBeans IDE 6.9 introduces the JavaFX Composer, support for JavaFX SDK 1.3, support for the PHP Zend framework and Ruby on Rails 3.0, and more.

Problem defination and Analysis

The hardest part of building a software system is deciding precisely what to build. No other part of the conceptual work is so difficult as establishing the detailed technical requirement. Defining and applying good, complete requirements are hard to work, and success in this endeavor has eluded many of us. Yet, we continue to make progress.

Problem definition describes the *What* of a system, not *How* . The quality of a software product is only as good as the process that creates it. Problem definition is one of the most crucial steps in this creation process. Without defining a problem, developers do not know what to build, customers do not know what to expect, and there is no way to validate that the built system satisfies the requirement.

Problem definition and Analysis is the activity that encompasses learning about the problem to be solved, understanding the needs of customer and users, trying to find out who the user really is, and understanding all the constraints on the solution. It includes all activities related to the following:

- ✓ Identification and documentation of customer's or user's needs.
- ✓ Creation of a document that describes the external behavior and the association constraints that will satisfies those needs.
- ✓ Analysis and validation of the requirements documents to ensure consistency, completeness, and feasibility
- ✓ Evolution of needs.

After the analysis of the functioning of a Public Library system, the proposed System is expected to do the following: -

- ✓ To provide a user friendly, Graphical User Interface (GUI) based integrated and centralized environment for computerized hospital management system.
- ✓ The proposed system should maintain all the records and transactions, and should generate the required reports and information when required.
- ✓ To provide efficient and secured Information storage, flow and retrieval system, ensuring the integrity and validity of records.
- ✓ To provide graphical and user-friendly interface to interact with a centralized database based on client-server architecture.
- ✓ To identify the critical operation procedure and possibilities of simplification using modern IT tools and practices.

System design and Development

Database Design:

An important aspect of system design is the design of data storage structure. To begin with a logical model of data structure is developed first. A database is a container object which contains tables, queries, reports and data validation policies enforcement rules or constraints etc. A logical data often represented as a records are kept in different tables after reducing anomalies and redundancies. The goodness of data base design lies in the table structure and its relationship.

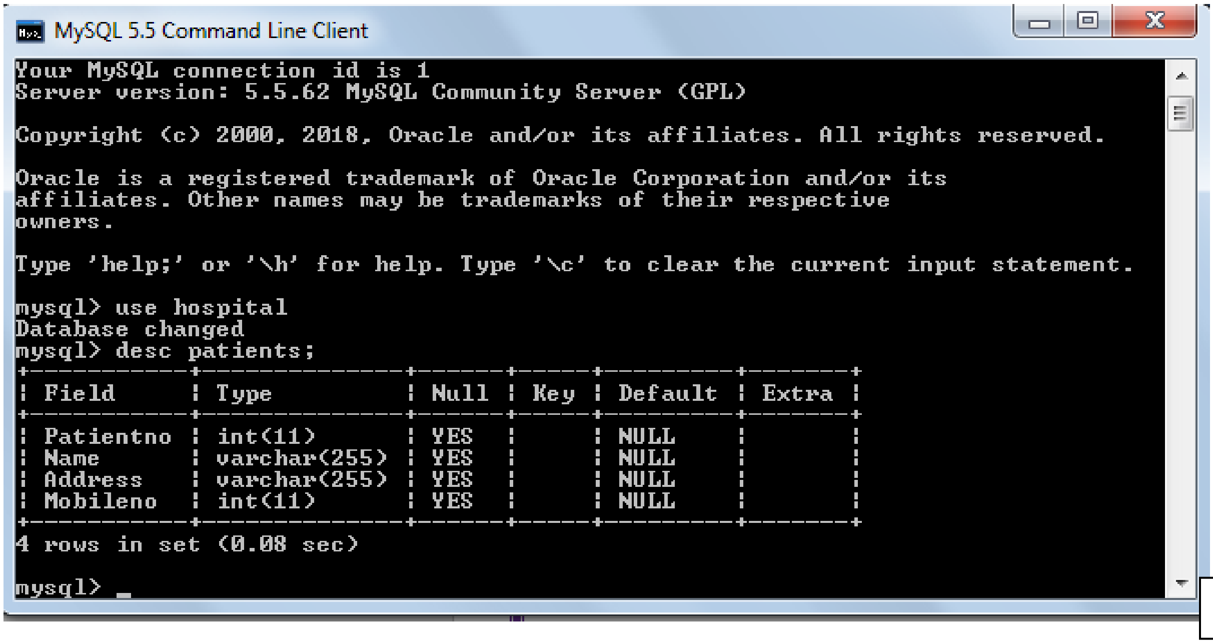
This software project maintains a database named **hospital** which contains the following table.

Table Design:

The database of hospital management System contains 1 tables. The table is normalized to minimize the redundancies of data and enforcing the validation rules of the organization. Most of the tables are designed to store master records. The tables and their structure are given below.

Table: patients

Column Name	*Type*	*Size*
Patientno	int	11
Name	char	255
Address	char	255
Mobileno	int	11



```
Your MySQL connection id is 1
Server version: 5.5.62 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use hospital
Database changed
mysql> desc patients;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| Patientno | int(11)      | YES  |     | NULL    |       |
| Name      | varchar(255) | YES  |     | NULL    |       |
| Address   | varchar(255) | YES  |     | NULL    |       |
| Mobileno  | int(11)      | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
4 rows in set (0.08 sec)

mysql> _
```

Hardware & Software

Requirements

Hardware Requirements

Processor: Intel i3-2350M or Greater

Main Memory: 6 GB

Hard Disk: 20 GB or more

Network Card: Ethernet or Any other

Software Requirements

Platform: Windows 7/8/8.1/10

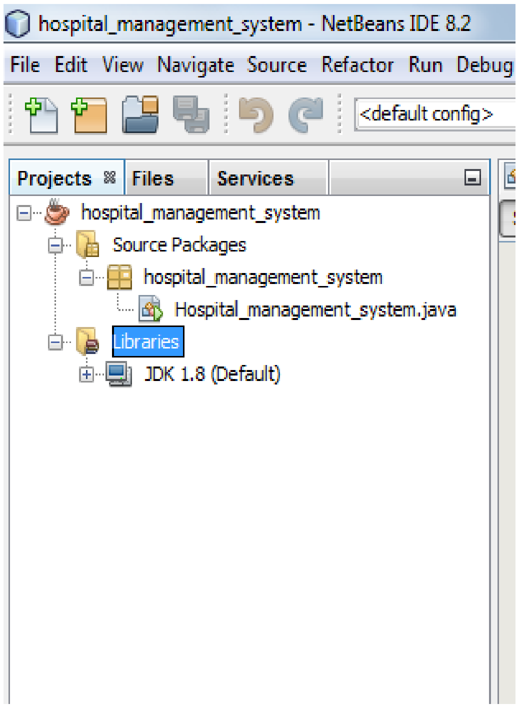
Client Side Validation: java/Netbeans 8.2

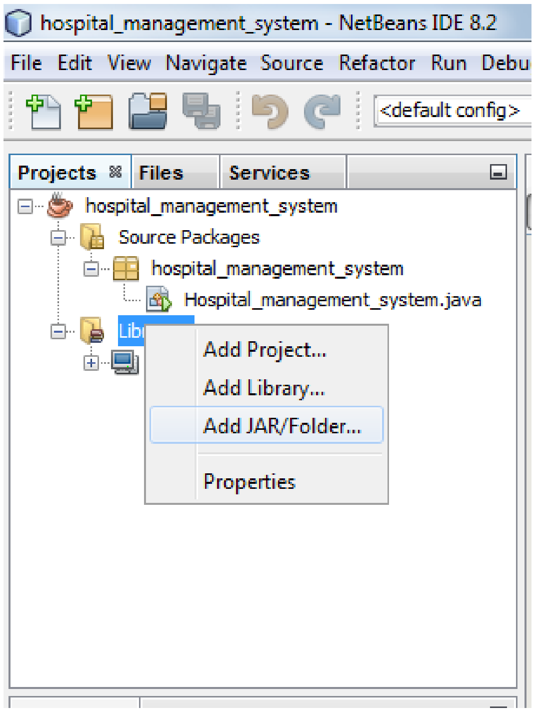
Server Side Validation: MySQL Server

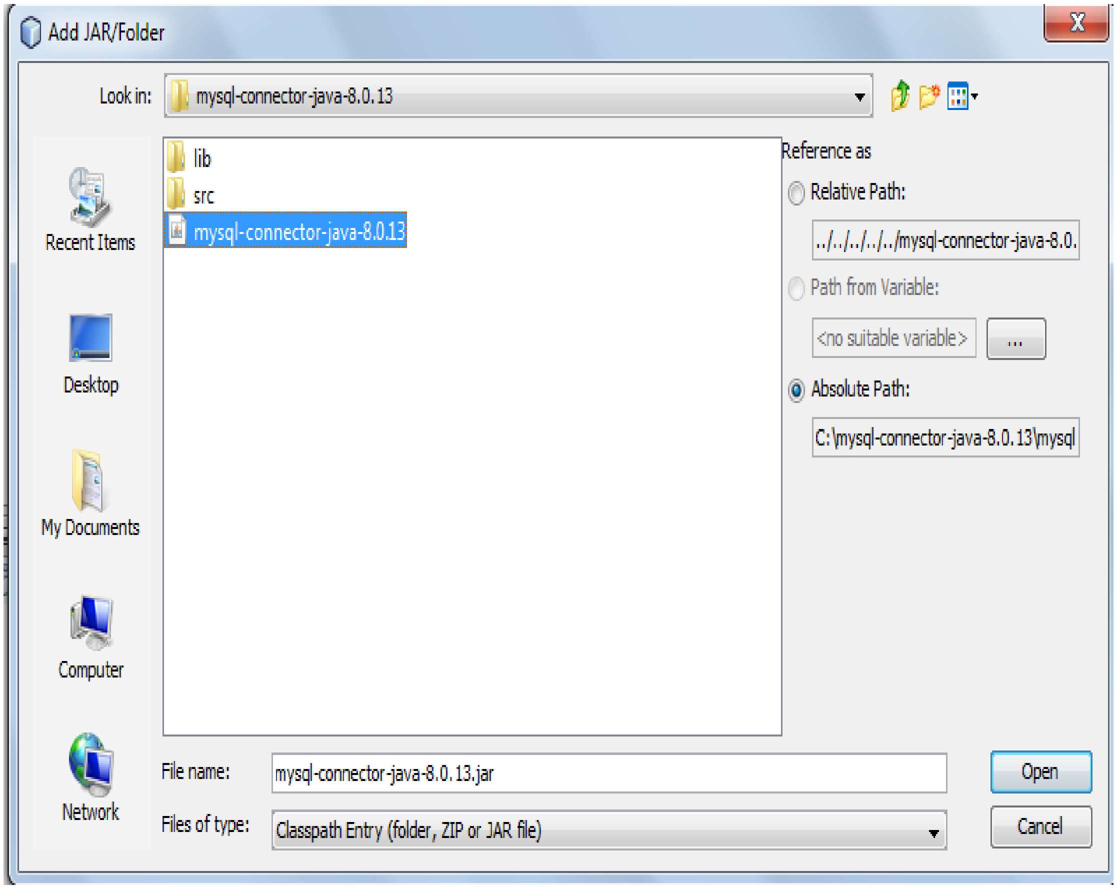
Database Connectivity: JDBC/ODBC

RDBMS: MySQL 5.5.62

SQL JAVA CONNECTOR


hospital_management_system - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug
<default config>
Projects Files Services
hospital_management_system
Source Packages
hospital_management_system
Hospital_management_system.java
Libraries
JDK 1.8 (Default)


hospital_management_system - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debu
<default config>
Projects Files Services
hospital_management_system
Source Packages
hospital_management_system
Hospital_management_system.java
Libr
Add Project...
Add Library...
Add JAR/Folder...
Properties



Look in: mysql-connector-java-8.0.13

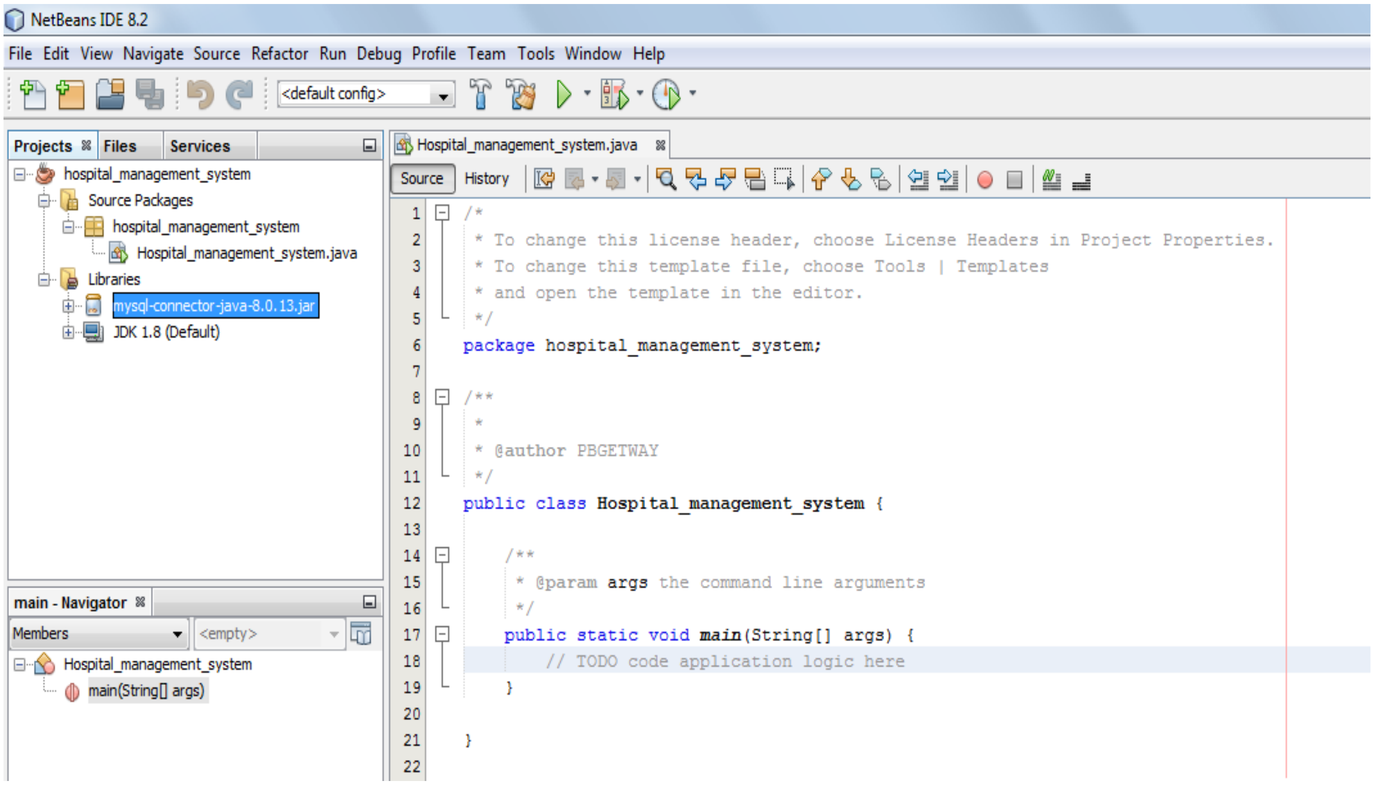
○ Relative Path: ../../../../../mysql-connector-java-8.0.

○ Path from Variable:

● Absolute Path: C:\mysql-connector-java-8.0.13\mysql

File name: mysql-connector-java-8.0.13.jar

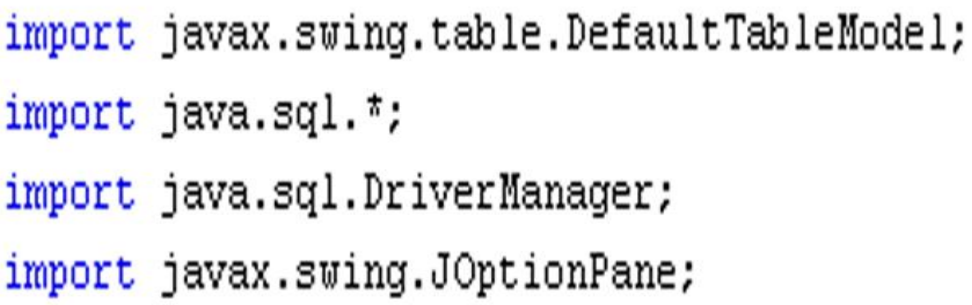
Files of type: Classpath Entry (folder, ZIP or JAR file)


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Projects
Files
Services
hospital_management_system
Source Packages
hospital_management_system
Hospital_management_system.java
Libraries
mysql-connector-java-8.0.13.jar
JDK 1.8 (Default)
main - Navigator
Members
<empty>
Hospital_management_system
main(String[] args)
Hospital_management_system.java
Source
History

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package hospital_management_system;

/**
 *
 * @author PBGETWAY
 */
public class Hospital_management_system {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```

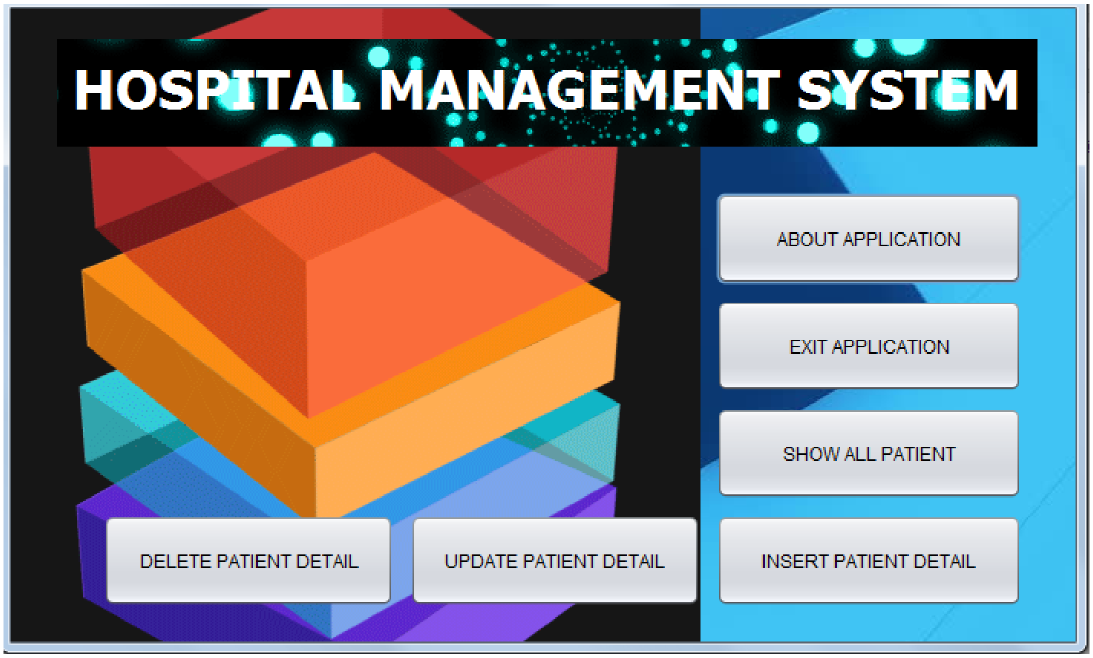
IMPORT STATEMENT

Import Statement :-

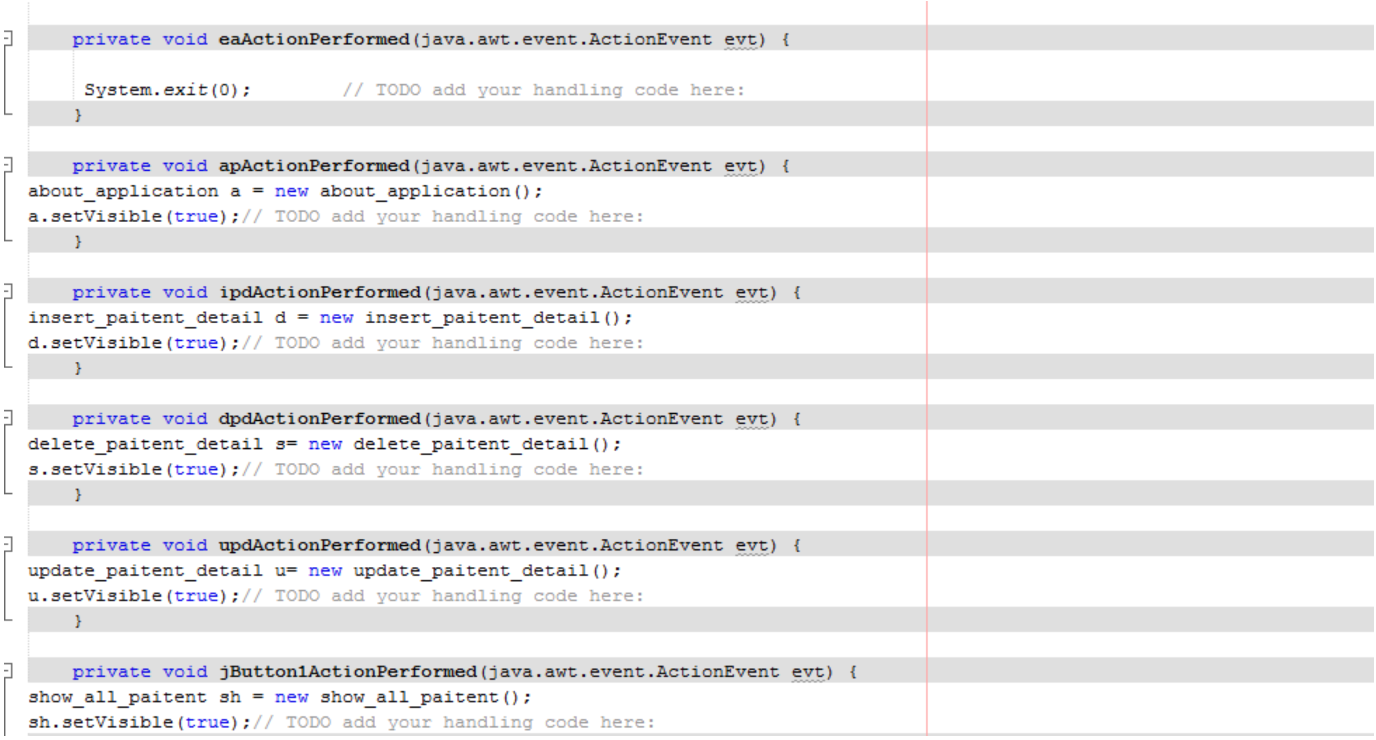


```
import javax.swing.table.DefaultTableModel;
import java.sql.*;
import java.sql.DriverManager;
import javax.swing.JOptionPane;
```

FORM : HOME:-

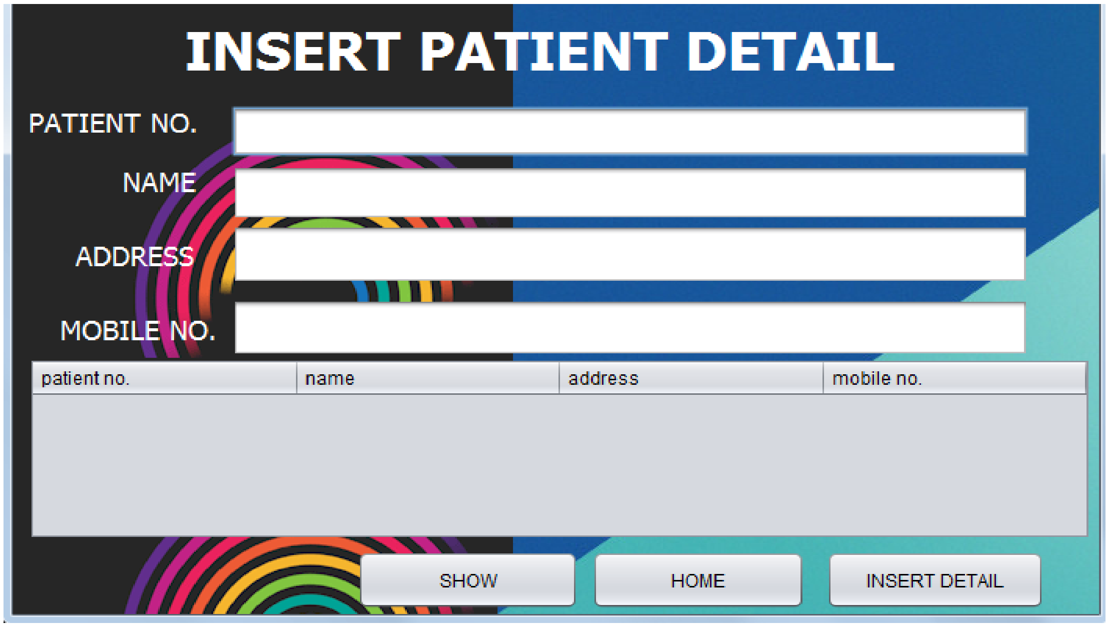


Code:-

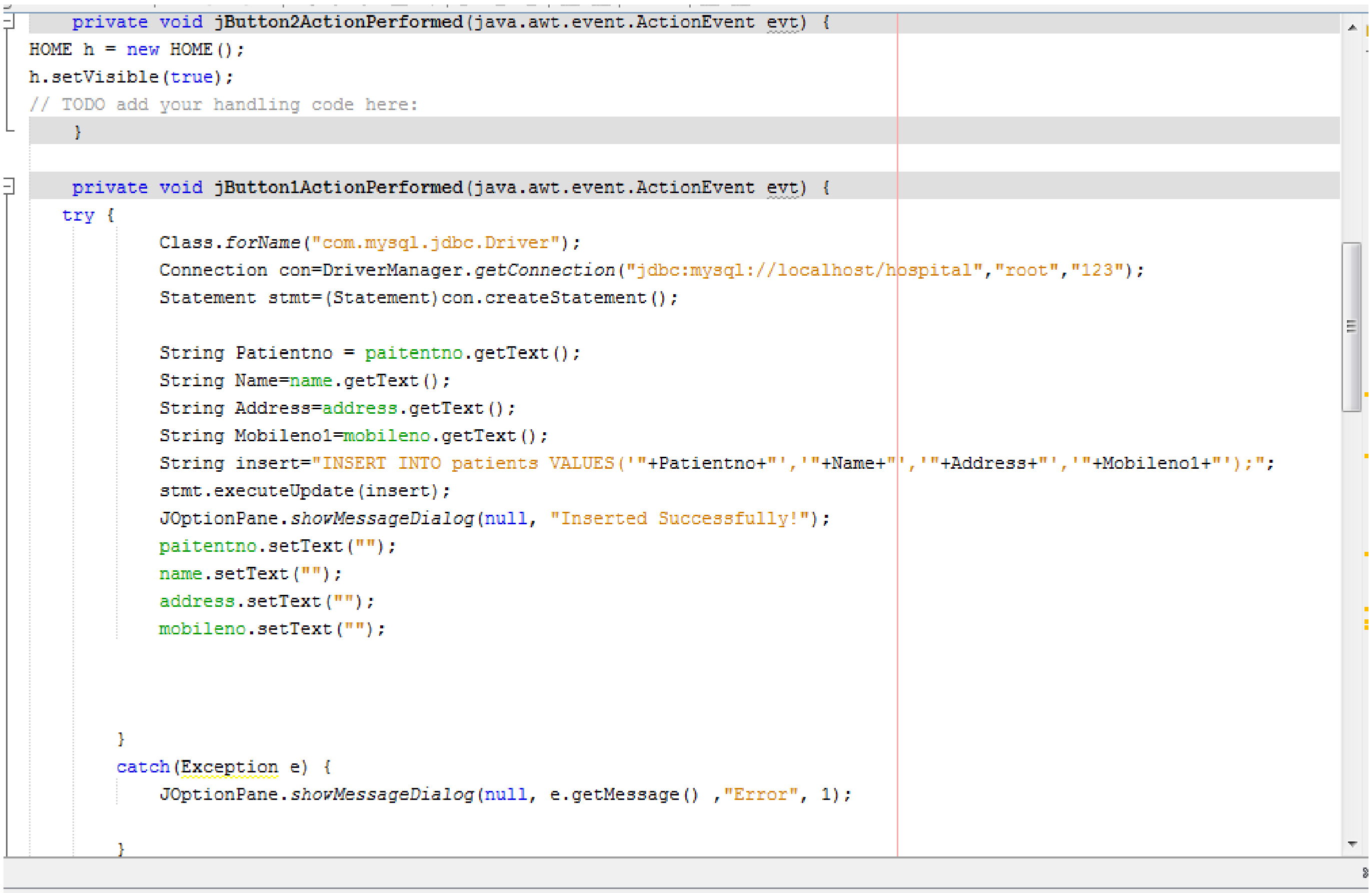


```
    private void eaActionPerformed(java.awt.event.ActionEvent evt) {

     System.exit(0);         // TODO add your handling code here:
    }

    private void apActionPerformed(java.awt.event.ActionEvent evt) {
about_application a = new about_application();
a.setVisible(true);// TODO add your handling code here:
    }

    private void ipdActionPerformed(java.awt.event.ActionEvent evt) {
insert_paitent_detail d = new insert_paitent_detail();
d.setVisible(true);// TODO add your handling code here:
    }

    private void dpdActionPerformed(java.awt.event.ActionEvent evt) {
delete_paitent_detail s= new delete_paitent_detail();
s.setVisible(true);// TODO add your handling code here:
    }

    private void updActionPerformed(java.awt.event.ActionEvent evt) {
update_paitent_detail u= new update_paitent_detail();
u.setVisible(true);// TODO add your handling code here:
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
show_all_paitent sh = new show_all_paitent();
sh.setVisible(true);// TODO add your handling code here:
```

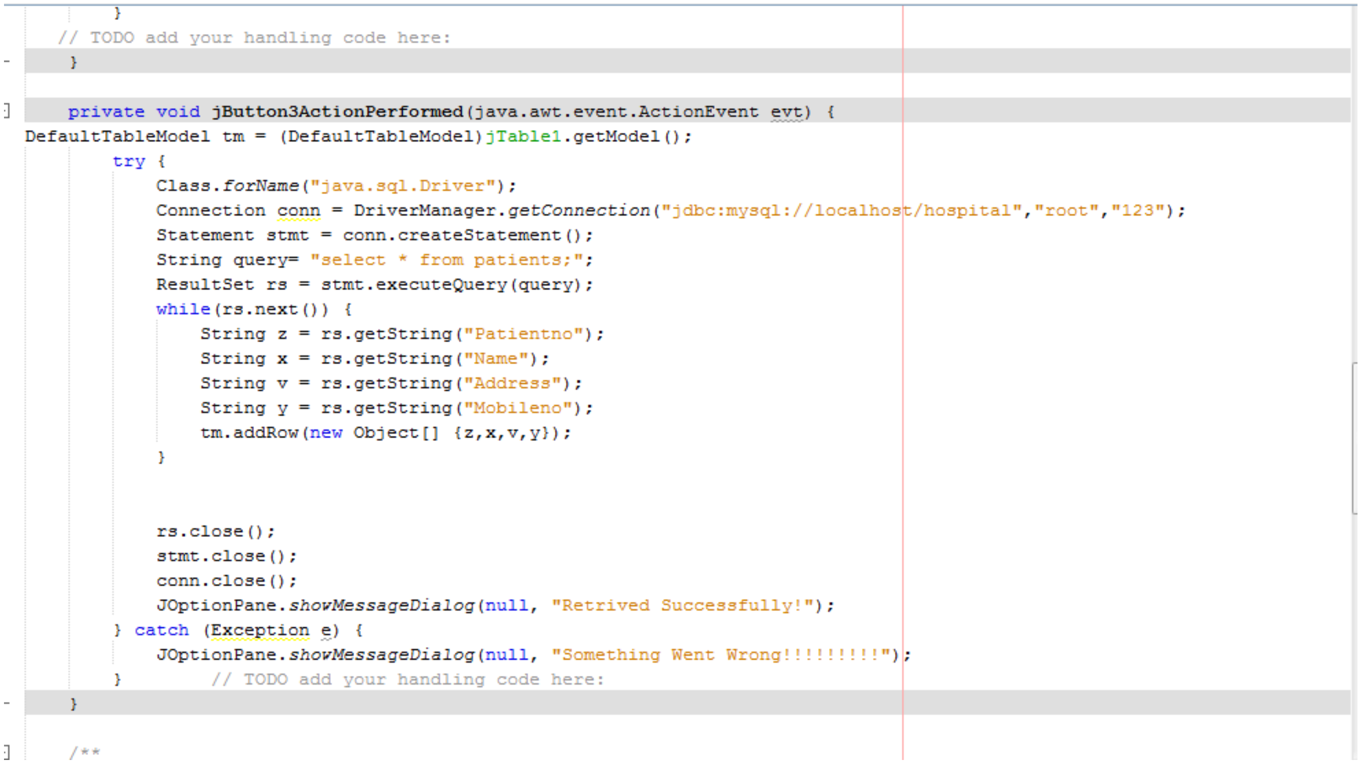
FORM : INSERT PATIENT DETAIL:-



CODE:-

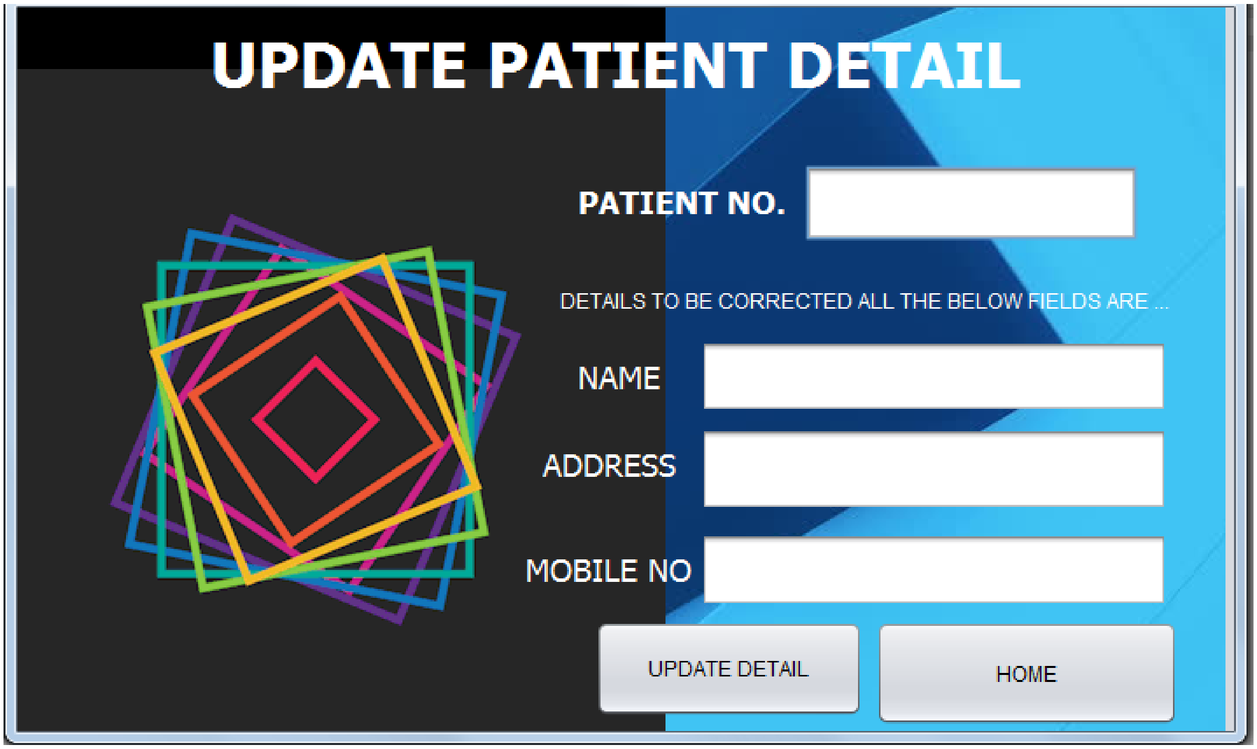


```
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
HOME h = new HOME();
h.setVisible(true);
// TODO add your handling code here:
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
   try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con=DriverManager.getConnection("jdbc:mysql://localhost/hospital","root","123");
            Statement stmt=(Statement)con.createStatement();

            String Patientno = paitentno.getText();
            String Name=name.getText();
            String Address=address.getText();
            String Mobileno1=mobileno.getText();
            String insert="INSERT INTO patients VALUES('"+Patientno+"','"+Name+"','"+Address+"','"+Mobileno1+"');";
            stmt.executeUpdate(insert);
            JOptionPane.showMessageDialog(null, "Inserted Successfully!");
            paitentno.setText("");
            name.setText("");
            address.setText("");
            mobileno.setText("");



        }
        catch(Exception e) {
            JOptionPane.showMessageDialog(null, e.getMessage() ,"Error", 1);

        }
```



```
        }
    // TODO add your handling code here:
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
DefaultTableModel tm = (DefaultTableModel)jTable1.getModel();
        try {
            Class.forName("java.sql.Driver");
            Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/hospital","root","123");
            Statement stmt = conn.createStatement();
            String query= "select * from patients;";
            ResultSet rs = stmt.executeQuery(query);
            while(rs.next()) {
                String z = rs.getString("Patientno");
                String x = rs.getString("Name");
                String v = rs.getString("Address");
                String y = rs.getString("Mobileno");
                tm.addRow(new Object[] {z,x,v,y});
            }


            rs.close();
            stmt.close();
            conn.close();
            JOptionPane.showMessageDialog(null, "Retrived Successfully!");
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Something Went Wrong!!!!!!!!!");
        }       // TODO add your handling code here:
    }

    /**
```

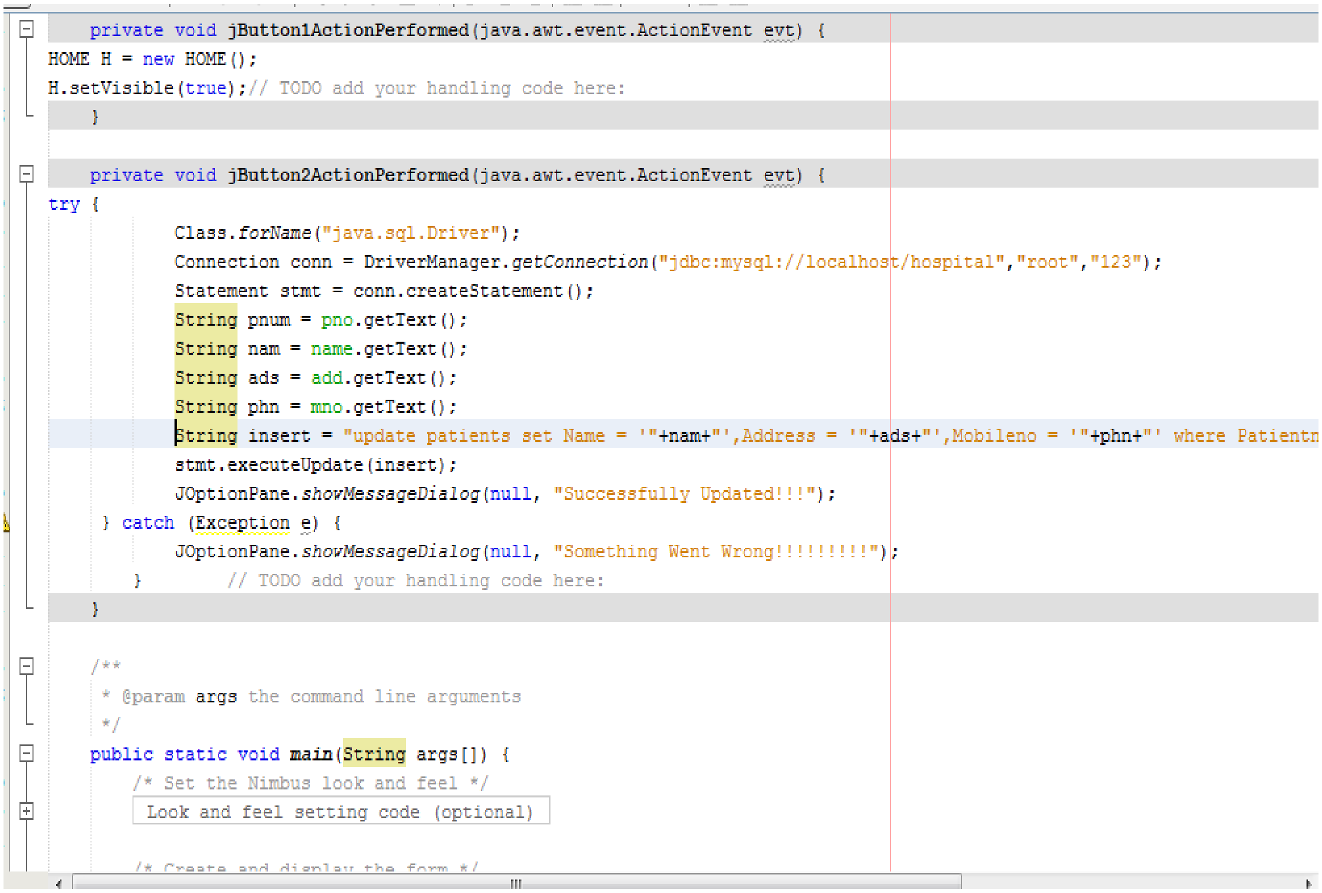
FORM: UPDATE PATIENT DETAIL:-



UPDATE PATIENT DETAIL

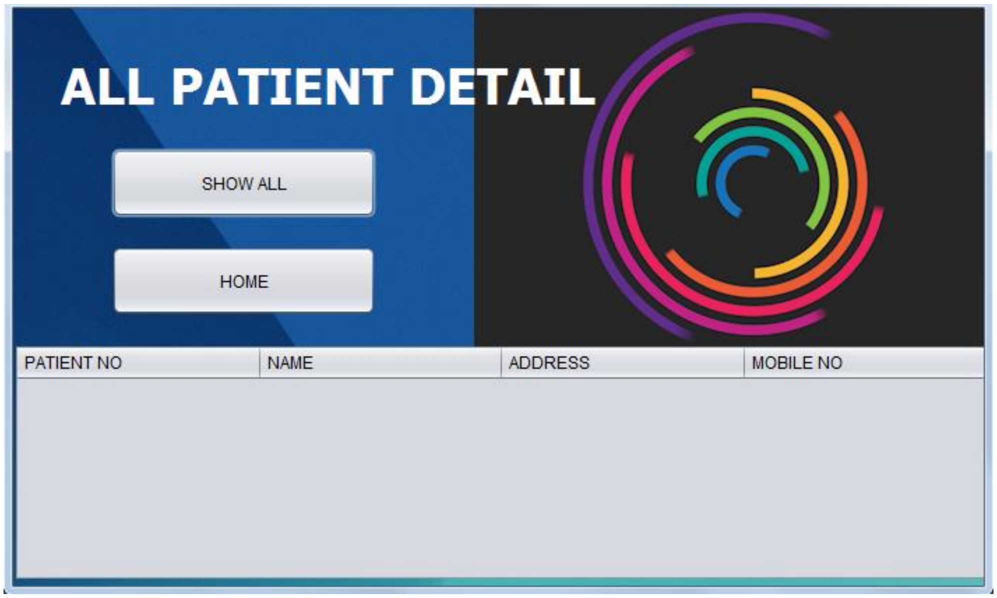
DETAILS TO BE CORRECTED ALL THE BELOW FIELDS ARE ...

CODE:-

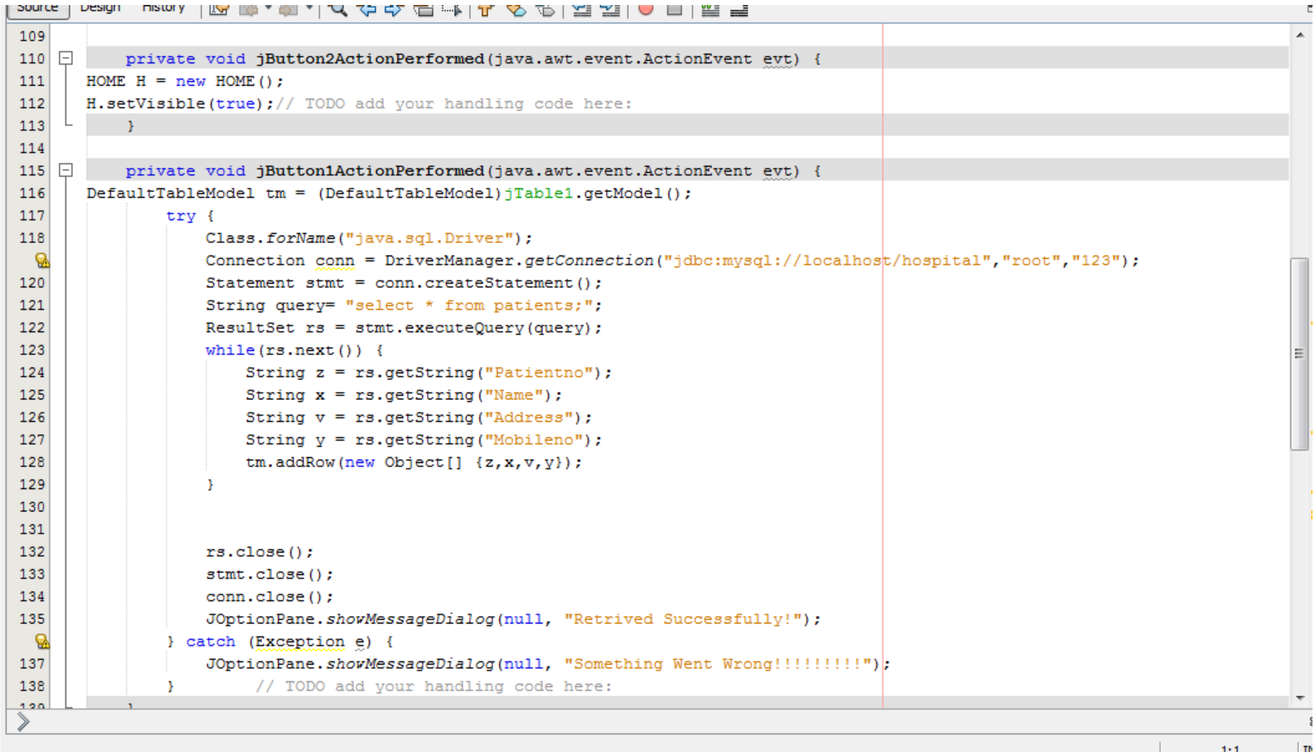


```
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
HOME H = new HOME();
H.setVisible(true);// TODO add your handling code here:
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
try {
            Class.forName("java.sql.Driver");
            Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/hospital","root","123");
            Statement stmt = conn.createStatement();
            String pnum = pno.getText();
            String nam = name.getText();
            String ads = add.getText();
            String phn = mno.getText();
            String insert = "update patients set Name = '"+nam+"',Address = '"+ads+"',Mobileno = '"+phn+"' where Patientn
            stmt.executeUpdate(insert);
            JOptionPane.showMessageDialog(null, "Successfully Updated!!!");
     } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Something Went Wrong!!!!!!!!!");
        }       // TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)

        /* Create and display the form */
```

FORM : SHOW ALL PATIENT DETAIL :-

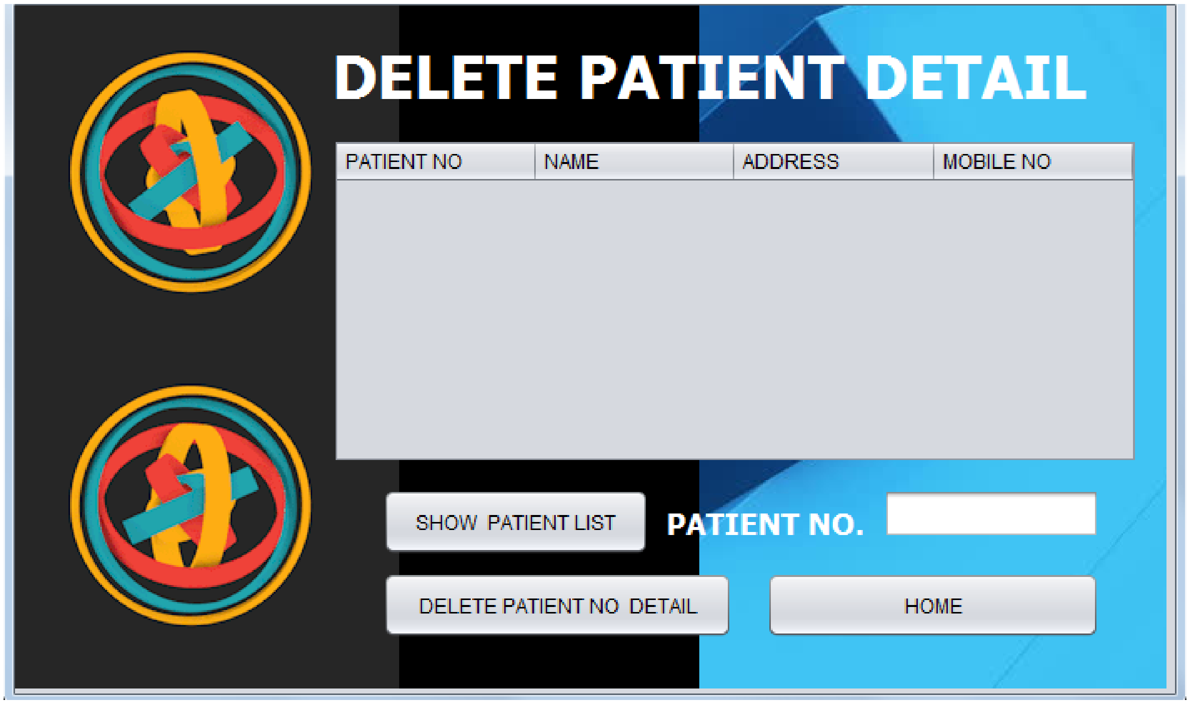


CODE:-



```java
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
HOME H = new HOME();
H.setVisible(true);// TODO add your handling code here:
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
DefaultTableModel tm = (DefaultTableModel)jTable1.getModel();
        try {
            Class.forName("java.sql.Driver");
            Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/hospital","root","123");
            Statement stmt = conn.createStatement();
            String query= "select * from patients;";
            ResultSet rs = stmt.executeQuery(query);
            while(rs.next()) {
                String z = rs.getString("Patientno");
                String x = rs.getString("Name");
                String v = rs.getString("Address");
                String y = rs.getString("Mobileno");
                tm.addRow(new Object[] {z,x,v,y});
            }


            rs.close();
            stmt.close();
            conn.close();
            JOptionPane.showMessageDialog(null, "Retrived Successfully!");
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Something Went Wrong!!!!!!!!!");
        }         // TODO add your handling code here:
    }
```

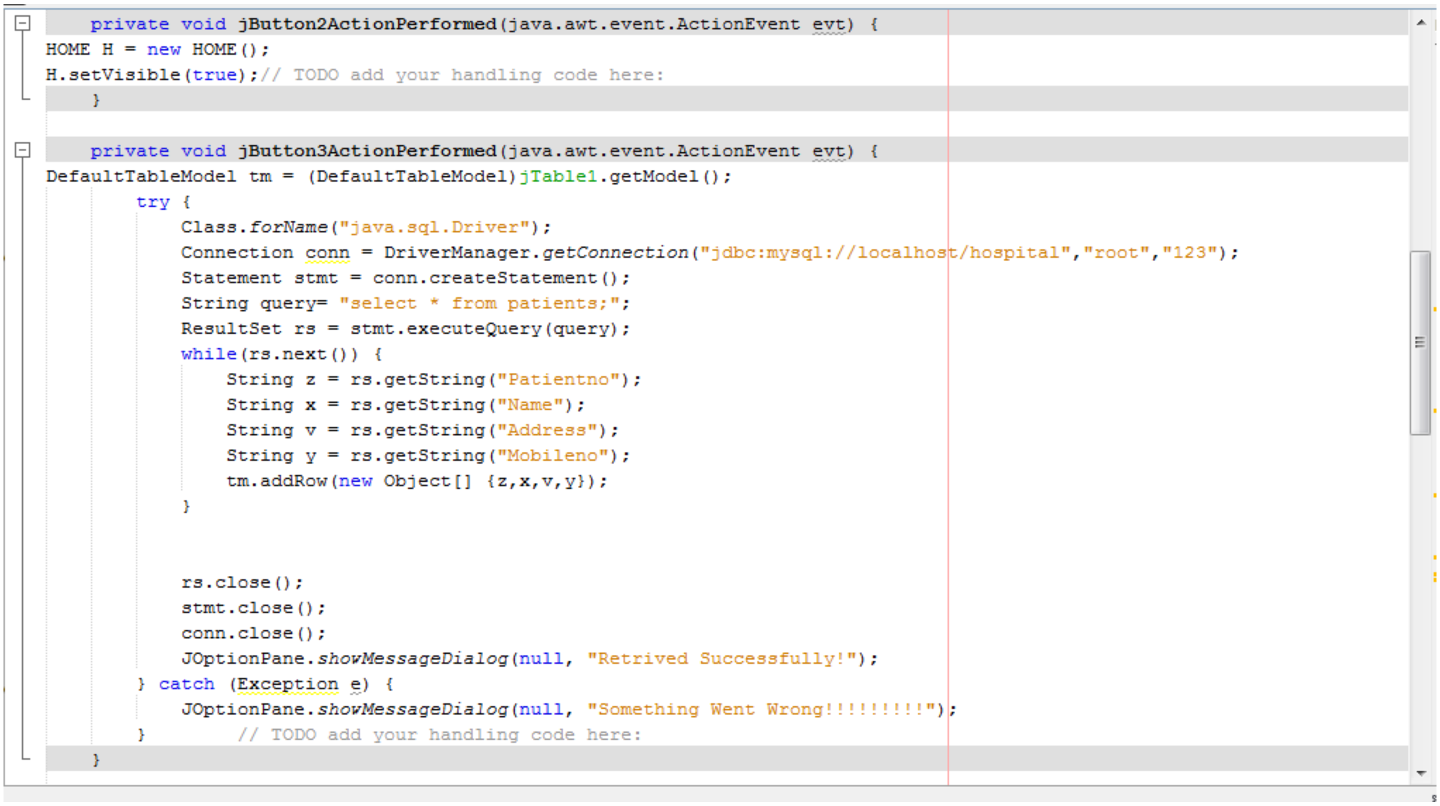
FORM : DELETE PATIENT DETAIL:-



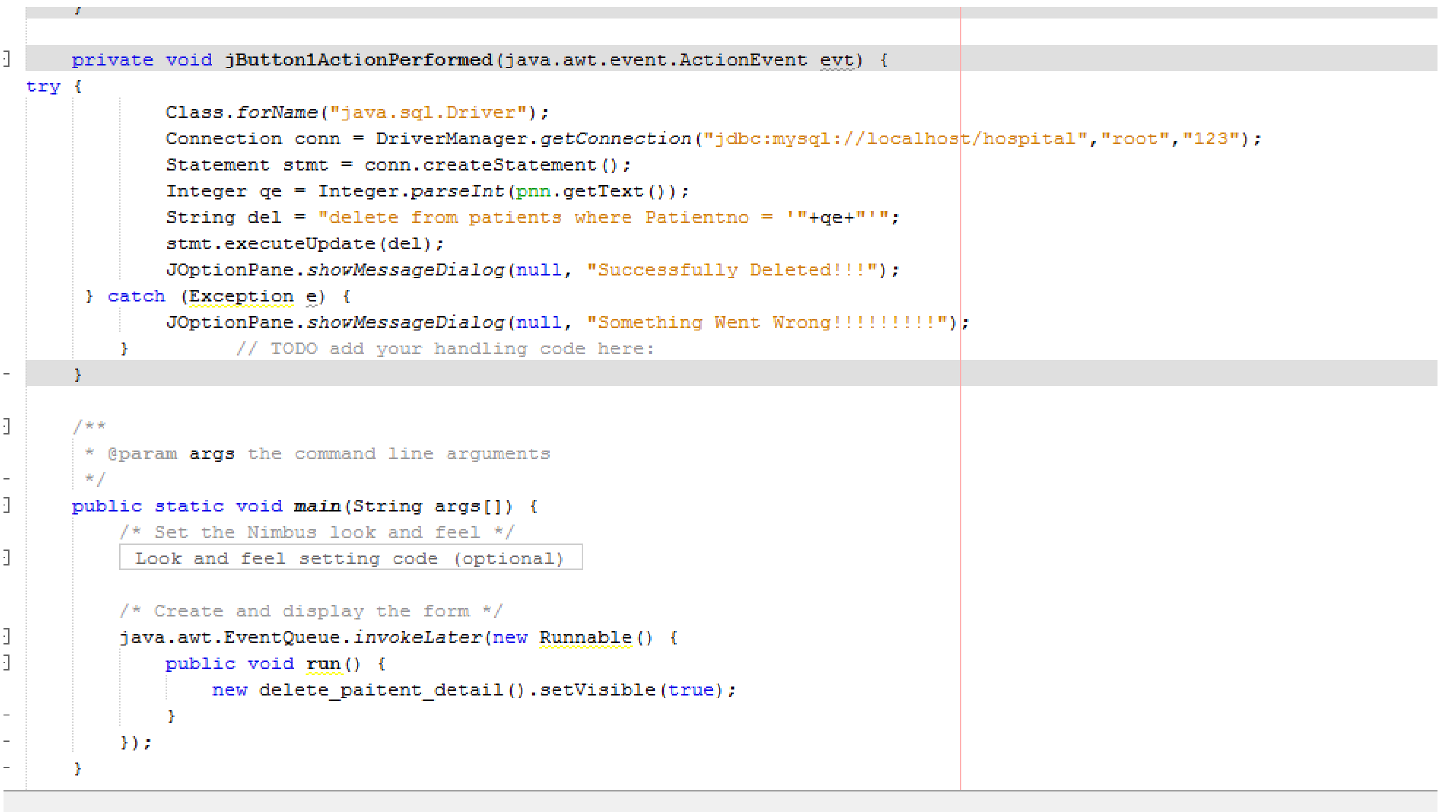
DELETE PATIENT DETAIL

PATIENT NO	NAME	ADDRESS	MOBILE NO

CODES:-

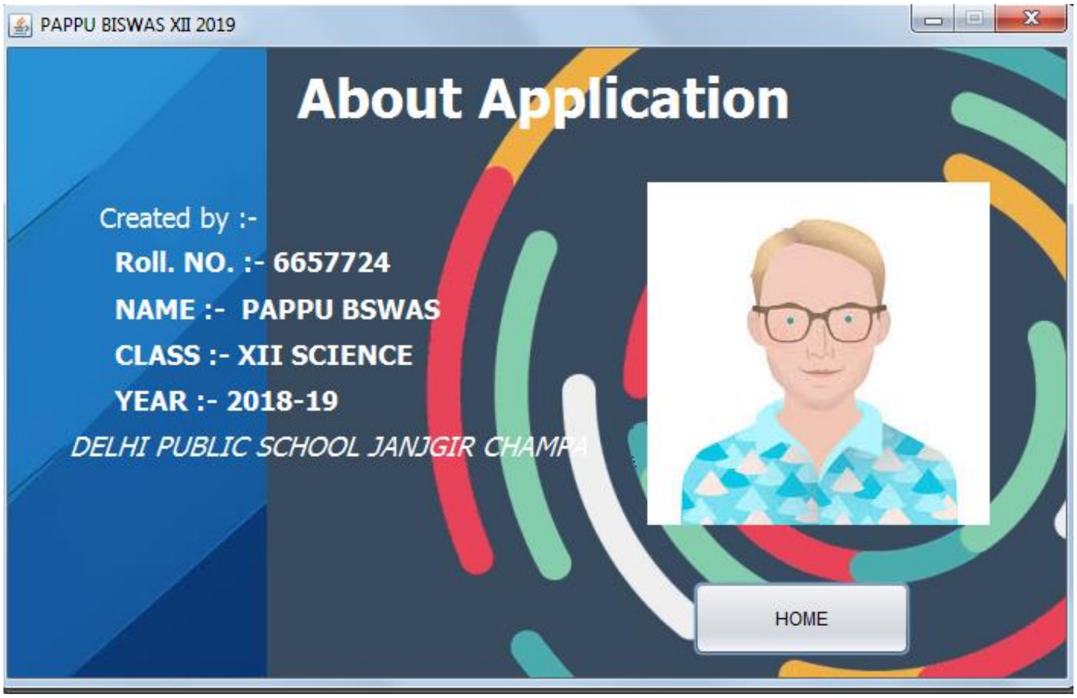


```
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
HOME H = new HOME();
H.setVisible(true);// TODO add your handling code here:
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
DefaultTableModel tm = (DefaultTableModel)jTable1.getModel();
        try {
            Class.forName("java.sql.Driver");
            Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/hospital","root","123");
            Statement stmt = conn.createStatement();
            String query= "select * from patients;";
            ResultSet rs = stmt.executeQuery(query);
            while(rs.next()) {
                String z = rs.getString("Patientno");
                String x = rs.getString("Name");
                String v = rs.getString("Address");
                String y = rs.getString("Mobileno");
                tm.addRow(new Object[] {z,x,v,y});
            }


            rs.close();
            stmt.close();
            conn.close();
            JOptionPane.showMessageDialog(null, "Retrived Successfully!");
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Something Went Wrong!!!!!!!!!");
        }         // TODO add your handling code here:
    }
```



```
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
try {
            Class.forName("java.sql.Driver");
            Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/hospital","root","123");
            Statement stmt = conn.createStatement();
            Integer qe = Integer.parseInt(pnn.getText());
            String del = "delete from patients where Patientno = '"+qe+"'";
            stmt.executeUpdate(del);
            JOptionPane.showMessageDialog(null, "Successfully Deleted!!!");
    } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Something Went Wrong!!!!!!!!!");
        }         // TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new delete_paitent_detail().setVisible(true);
            }
        });
    }
```

FORM : ABOUT:-



PAPPU BISWAS XII 2019

About Application

Created by :-

Roll. NO. :- 6657724

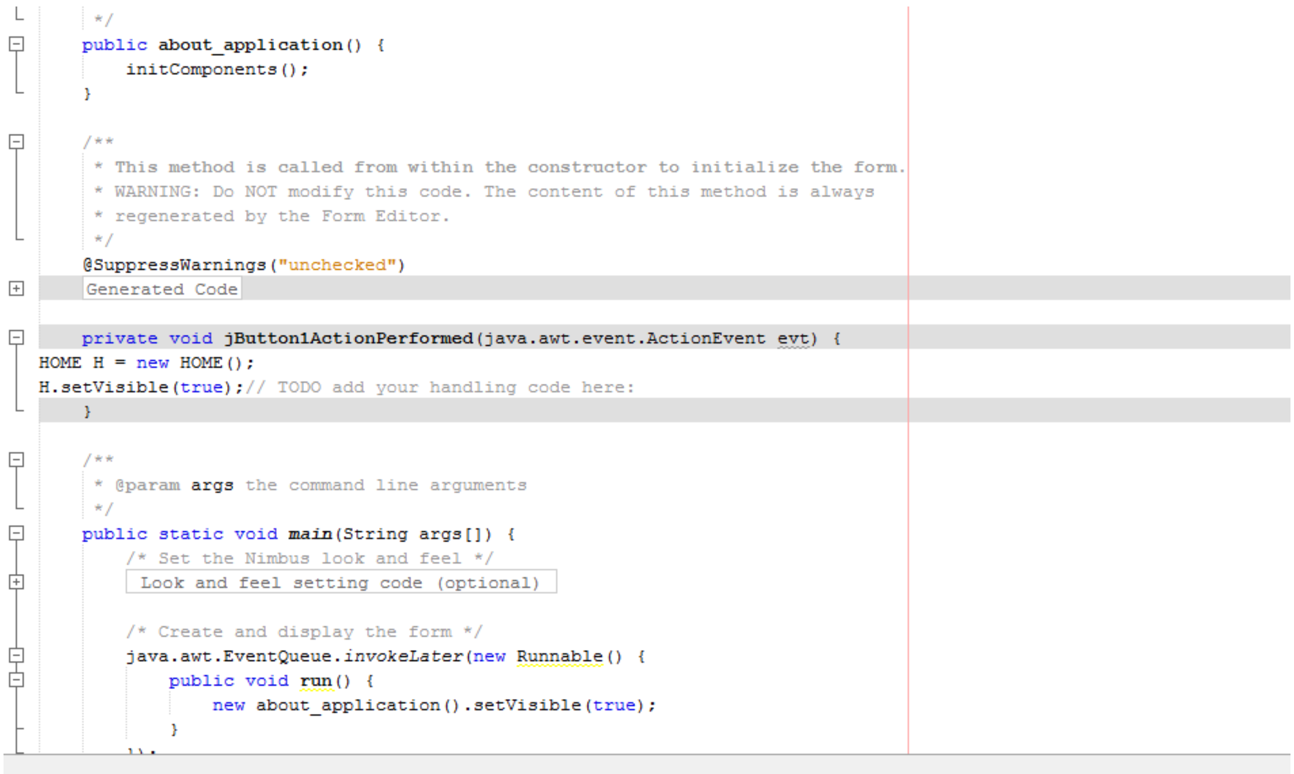
NAME :- PAPPU BSWAS

CLASS :- XII SCIENCE

YEAR :- 2018-19

DELHI PUBLIC SCHOOL JANJGIR CHAMPA

CODES:-



```
    */
    public about_application() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
HOME H = new HOME();
H.setVisible(true);// TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new about_application().setVisible(true);
            }
        });
```

ADVANTAGES

This project has got a number of advantages. Following are some of the advantages:

Using this project, one can easily Insert, Update, Retrieve and delete the patient details.

- ✓ The interface between the user and the project has been designed to let the user proceed with Insert, Update, Retrieve and delete of data easily.
- ✓ Managing data has been made easy.
- ✓ Efforts have been made to keep navigation as easy as possible.
- ✓ For the sake of simplicity and convenience, if in future it is decided to launch this project as an open source, available-to-all project, one will be able to amend the setup without much trouble to modify the project according to one's need.
- ✓ The user gets an extra edge due the inter-relatedness of the tables through the project.

CONCLUSION

This was an effort to develop a simple Hospital Management System which may be useful to a Patient management to insert, delete, update and retrieve information about any patients.

The proposed software system is expected to do the following functionality-

- ✓ **To provide a user friendly, Graphical User Interface (GUI) based integrated and centralized environment for MIS activities.**
- ✓ **The proposed system should maintain all the records and transactions, and should generate the required reports and information when required.**
- ✓ **To provide graphical and user-friendly interface to interact with a centralized database based on client-server architecture.**
- ✓ **To identify the critical operation procedure and possibilities of simplification using modern IT tools and practices.**

Installation manual

Database Installation

The software project is distributed with a backup copy of a Database named **hospital** with required tables. Some dummy records are present in the tables for testing purposes, which can be deleted before inserting real data. The project contains **patients.sql** file which installs a database and tables in the computer system.

Note: The PC must have MySQL server with user (***root***) and password (123) . If root password is any other password, it can be changed by running MySQL Server Instance Configure Wizard.

Start ▸ Program ▸ MySQL ▸ MySQL Server ▸ MySQL Server Instance Config Wizard

Provide current password of root and new password as '123', this will change the root password.

To install a MySQL database from a dump file ***(patients.sql)*** , simply follow the following steps.

Step 1: Copy the patients.sql file in **C:\Program files\MySQL\MySQL server 5.1\Bin** folder.

Step 2: Open MySQL and type the following command to create the database named hospital.

```
mysql> create database hospital;
```

Step 3: Open Command Window (Start ▸ Run ▸ cmd)

Step 4: Go to the following folder using CD command of DOS.

C:\Program files\Mysql\MySql server 5.1\Bin>

Step 5: type the following command on above prompt -

```
C:\\bin> mysql -u root –p 123 patients< patients.sql
```

This will create a hospital database with required tables.

references

In order to work on this project titled ***Hospital Management System,*** the following books and literature are referred by me during the various phases of development of the project.

(1)www.google.com

(2) http://www.mysql.org/

(3) http://www.netbeans.org/

(4)images from google/images

(4) On-line Help of NetBeans

(6) Informatics Practices for class XII

-by Sumita Arora

(8) Various Websites of Discussion Forum and software development activities.

Other than the above-mentioned books, the suggestions and supervision of my teacher and my class experience also helped me to develop this software project.