

ABSTRACT

This software helps in managing Student and Teacher details, Batches, and Marks. Provides easy reports like Outstanding Marks etc. The software guarantees saving of money and time for owners apart from organizing the institute information. You can streamline the coaching institute business. Your data is safe and secure in this software.

Flow chart of this software:-

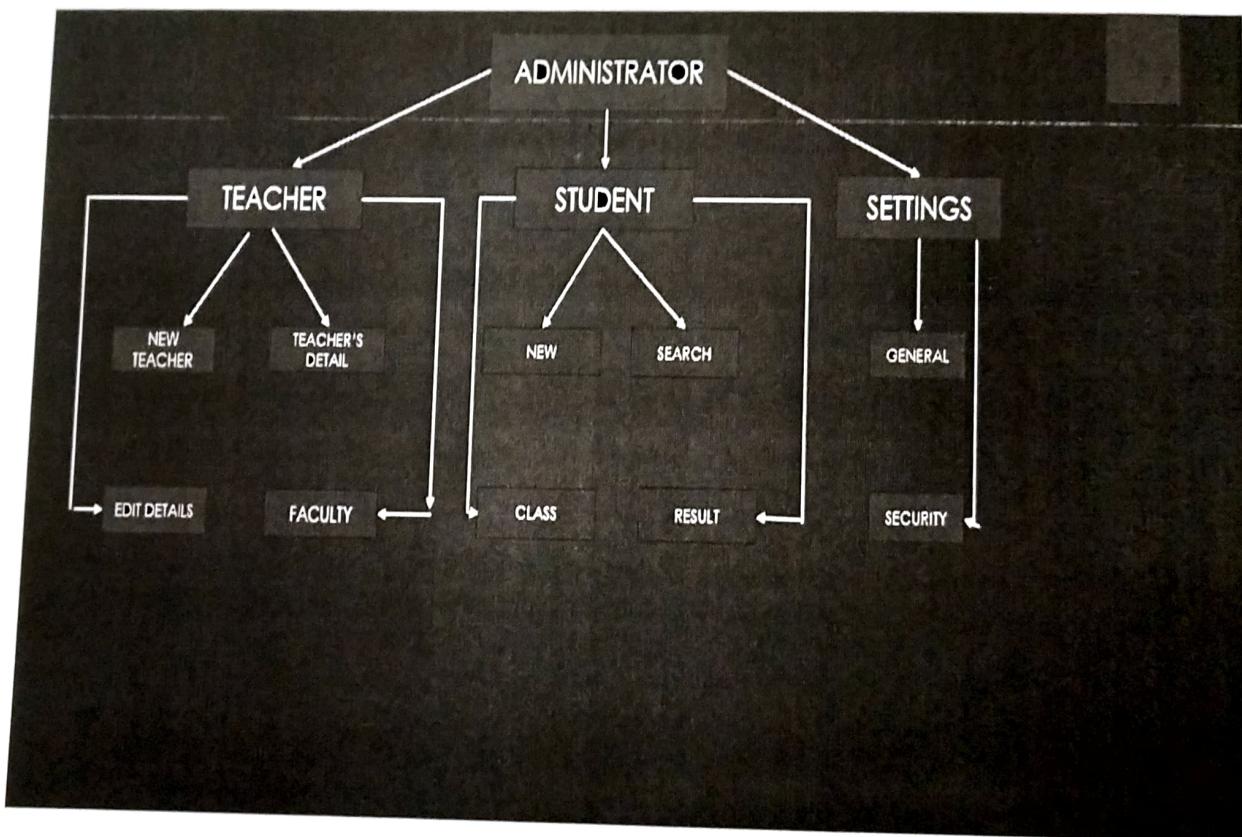


TABLE OF CONTENTS

Chapter	Coaching management system		Page No.
	Certificate		1,2
	Abstract		3
Chapter 1:	Introduction		
	1.1	objectives	5
	1.2	Advantages	6
Chapter 2:	Packages and snaps of project		
	2.1	Package snap shot	7
	2.2	Software snap shots and coding	8 to 31
Chapter 3:	Software and hardware Description		
	3.1	Java and MY SQL	32 to 34
	3.2	Hardware	35
Chapter 4:	Operation and Application		
	4.1	Operations	36
	4.2	Applications	37,38
Chapter 5:	Conclusion and Future Scope		
	5.1	Conclusion	39
	5.2	Future scopes	39
	Reference		40

CHAPTER 1

Introduction:

This software helps in managing Student and Teacher details, Batches, and Marks. Provides easy reports like Outstanding Marks etc. The software guarantees saving of money and time for owners apart from organizing the institute information. You can streamline the coaching institute business. Your data is safe and secure in this software.

The education management system is a business optimization of school or educational institute, coaching institutes, online mock test and its processes. Coaching institute management system centralizes institutes by providing features to its administrator.

FEATURES

1. Connected database.
2. Administrator login.
3. Forgot pass word.
4. Safe logout on closing.
5. Binary searching.
6. Two step authentication on changing password through forgot password.
7. Clear Instructions are given inside software for making it easy to use.

Objectives of Coaching Management:

1. Consolidation of spiritual strength of the society.
2. Maintaining the historic continuity of the society.
3. Securing the past achievements of the society.
4. Guaranteeing the future of the society.

A institute should be From the Point of View of Society

Broadly speaking the organization should be managed for the following objectives

Managed:

- To train his faculties.
- To widen his outlook.
- To cultivate his mind.
- To form and strength his character.
- To develop and cultivate his aesthetic faculty.
- To build up his body and give him health and community and the state.

ADVANTAGES

- Better performance by the students.
- Simplifying and streamlining all the tasks.
- Better communication.
- Easy access to all and Complete tracking of the students.
- Complete tracking of the students

CHAPTER 2

PACKAGES:

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects school login.java

Files

Source Packages school

- School.java
- forgotpassword.java
- login.java
- main.java
- pass1.java
- pass2.java
- pass3.java

Test Packages <default package>

Libraries

- Absolute Layout - AbsoluteLayout.jar
 - [mysql-connector-java-5.0.4-bin.jar]
 - META-INF
 - com.mysql.jdbc
 - com.mysql.jdbc.configs
 - com.mysql.jdbc.exceptions
 - com.mysql.jdbc.integration.c3p0
 - com.mysql.jdbc.integration.jboss
 - com.mysql.jdbc.jdbc2.optional
 - com.mysql.jdbc.log
 - com.mysql.jdbc.profiler
 - com.mysql.jdbc.util
 - org.gjt.mm.mysql
- JDK 1.8 (Default)

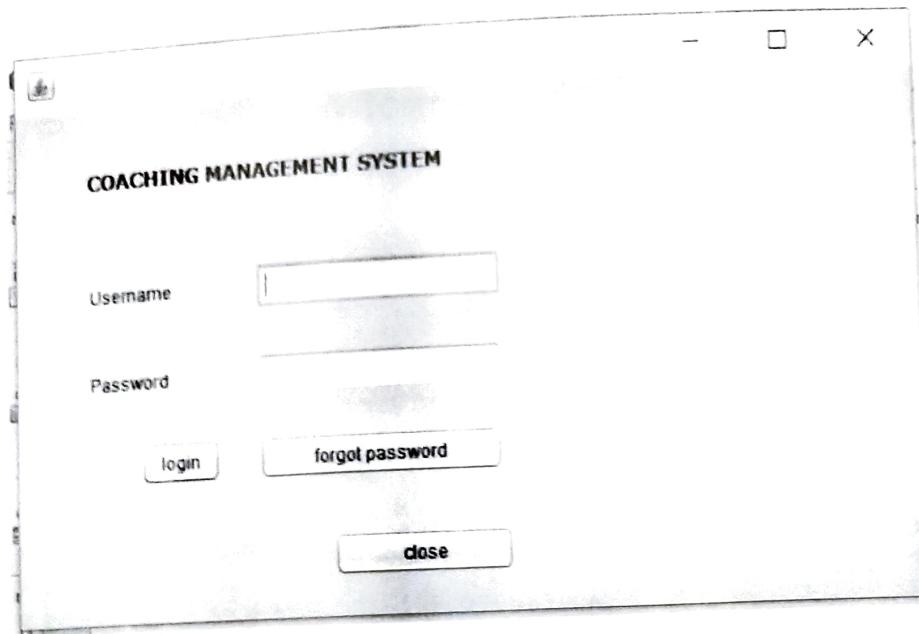
Test Libraries

Java Editor

```
username");  
SQLException e)  
  
ActionPerfomed(java.awt.event.ActionEvent evt) {  
    // code here  
  
ActionPerfomed(java.awt.event.ActionEvent evt) {  
    // code here  
  
ng args()) {  
    nd tool /  
ode (optional)  
e form /
```

CODES

- LOGIN PAGE



Close:-

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
    this.setVisible(false);
}
```

Forgot password:-

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
    new forgotpassword().setVisible(true);
}
```

Login:-

```
try
```

```
{
```

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

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

Statement stmt= con.createStatement();

ResultSet rs = stmt.executeQuery("select * from login");

String user=jTextField1.getText();

String pass=jPasswordField1.getText();

while(rs.next())

{

String user1=rs.getString(1);

if(user.equals(user1))

{

String pass1=rs.getString(2);

jLabel4.setText(" ");

if( pass.equals(pass1))

{

new main().setVisible(true);

jLabel6.setText("");

}

else

{

jLabel6.setText("Incorect Password");

}

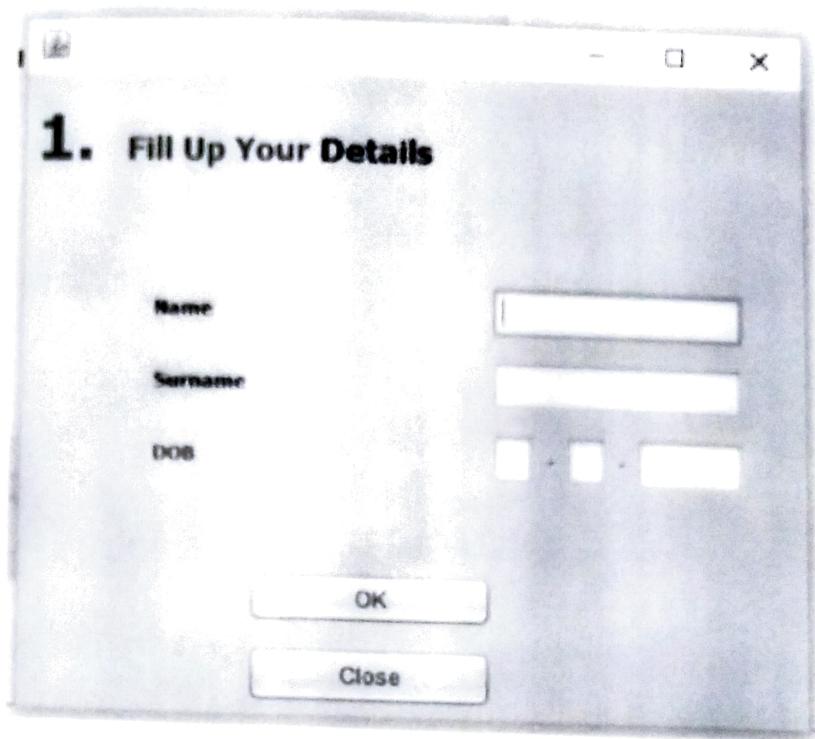
}

else

{
```

```
        jLabel4.setText("Incorrect Username");  
    }  
}  
}  
  
catch(ClassNotFoundException | SQLException e)  
{  
    System.out.println(e);  
}
```

- 1-Step Security



Ok:-

```
try
{
    Class.forName("com.mysql.jdbc.Driver");
    Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/school","root","golu");
    Statement stmt= con.createStatement();
    ResultSet rs = stmt.executeQuery("select * from pass1");
    String name=jTextField1.getText();
    String surname=jTextField2.getText();
    try
    {
        int date = Integer.parseInt(jTextField3.getText());
```

```
jTextField2.setText("");
jTextField3.setText("");
jTextField4.setText("");
jTextField5.setText("");
}

}

else
{
    jLabel7.setText("incorrect");

    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
    jTextField5.setText("");
}

}

else
{
    jLabel7.setText("incorrect");

    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
    jTextField5.setText("");
}
```

```
        else
        {
            jLabel7.setText("incorrect");
            jTextField1.setText("");
            jTextField2.setText("");
            jTextField3.setText("");
            jTextField4.setText("");
            jTextField5.setText("");
        }
    }

    else
    {
        jLabel7.setText("incorrect");
        jTextField1.setText("");
        jTextField2.setText("");
        jTextField3.setText("");
        jTextField4.setText("");
        jTextField5.setText("");
    }

}

catch(NumberFormatException e)
{
    jLabel7.setText("enter numbers in dob");
}

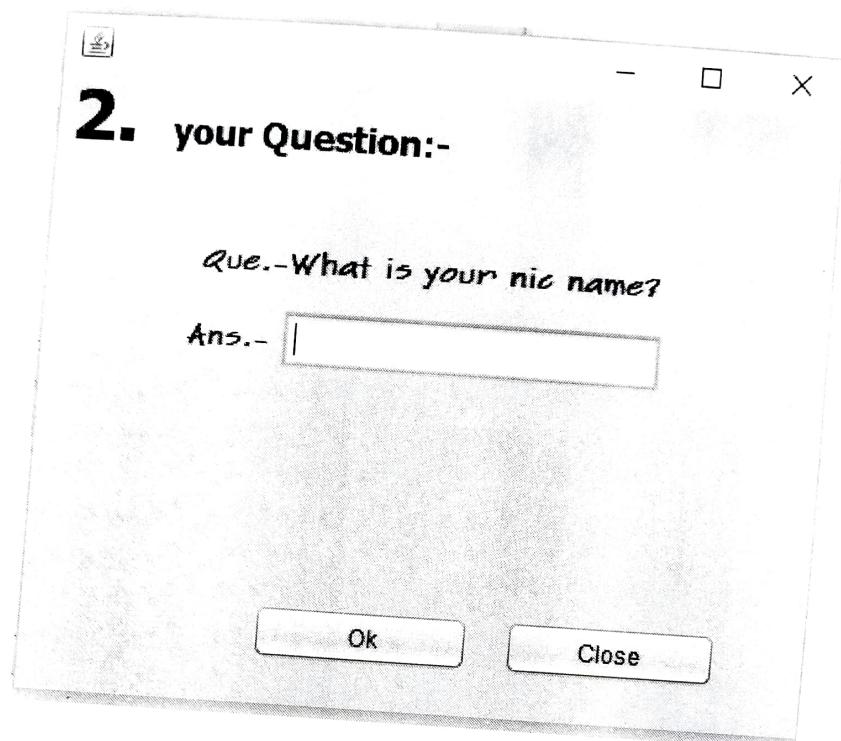
}

catch(ClassNotFoundException | SQLException e)
```

```
{
```

```
System.out.println(e);
```

- **2. STEP SECURITY**



Ok:-

try

```
{
```

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

```
Connection
```

```
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/school","root","golu");
```

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

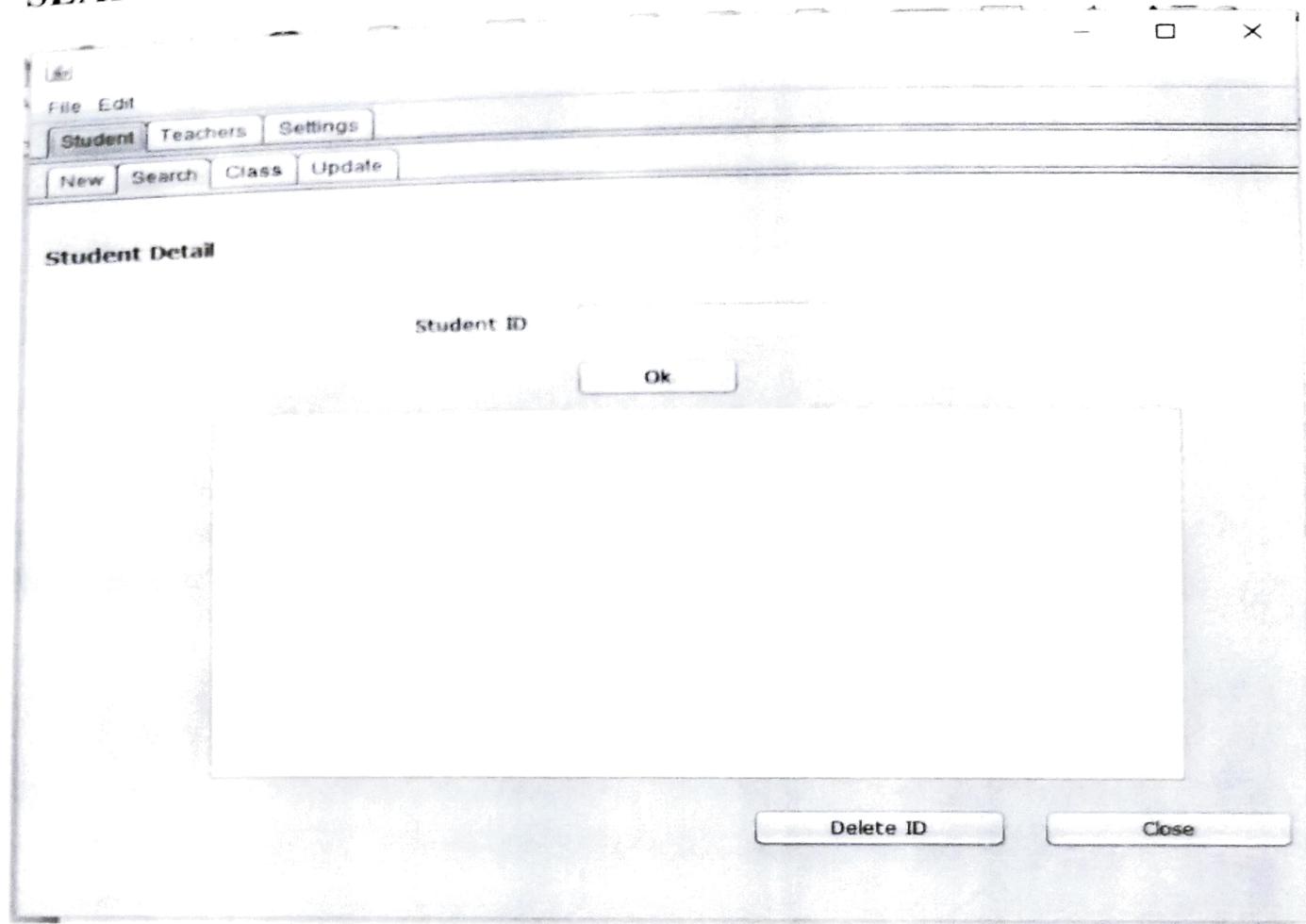
```
ResultSet rs = stmt.executeQuery("select * from login");
```

```
String answer=jTextField1.getText();
```

```
rs.next();
```

```
String answer1=rs.getString(3);
if(answer.equals(answer1))
{
    new pass3().setVisible(true);
}
else
{
    jLabel5.setText("Incorrect");
}
catch(ClassNotFoundException | SQLException e)
{
    System.out.println(e);
}
```

• SEARCH STUDENT



Ok:-

```
private void jButton17ActionPerformed(java.awt.event.ActionEvent evt) {  
try  
{  
Class.forName("com.mysql.jdbc.Driver");  
Connection  
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/school","root","golu");  
Statement stmt= con.createStatement();  
ResultSet rs = stmt.executeQuery("select * from student");  
int student = Integer.parseInt(jTextField27.getText());  
int student1;
```

```

int a=0;

while(rs.next())
{
student1=rs.getInt(1);

if(student1==student)
{

jTextArea8.setText("");

jTextArea8.append("Name:- "+rs.getString(2)+"\n"+ "Sex:- "+rs.getString(3)+"\n"+ "DOB:- "
+rs.getString(4)+"\n"+ "father's name:- "+rs.getString(5)+"\n"+ "father's occupation:- "
+rs.getString(6)+"\n"+ "Mother's name:- "+rs.getString(7)+"\n"+ "Mother's occupation:- "
+rs.getString(8)+"\n"+ "Mobile no.: - "+rs.getString(9)+"\n"+ "phone:- "
+rs.getString(10)+"\n"+ "address:- "+rs.getString(11)+"\n"+ "class: "+rs.getString(12));

a=a+1;
}

if(a==0)

{
jTextArea8.setText("\n\n\n\n      no record found");

}
catch(ClassNotFoundException | SQLException e)

{
System.out.println(e);
}
}

```

Delete:-

```

private void jButton18ActionPerformed(java.awt.event.ActionEvent evt) {

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

```

```
Connection  
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/school","root","golu");  
PreparedStatement stmt= con.prepareStatement("delete from student where id=?");  
int ac = Integer.parseInt(jTextField27.getText());  
stmt.setInt(1, ac);  
int i=stmt.executeUpdate();  
jTextArea8.setText("");  
System.out.println(i+ " record deleted");  
Statement stm= con.createStatement();  
ResultSet rs = stm.executeQuery("select * from teacher");  
int id = Integer.parseInt(jTextField27.getText());  
int id1=0;  
int a=0;  
while(rs.next())  
{  
    id1=rs.getInt(1);  
    if(id1==id)  
    {  
        a=a+0;  
    }  
    else  
    {  
        a=a+1;  
    }  
}  
if(a==0)  
{  
    jLabel54.setText("record is not deleted");
```

```
    }

else

{

    jLabel54.setText("record successfully deleted");

}

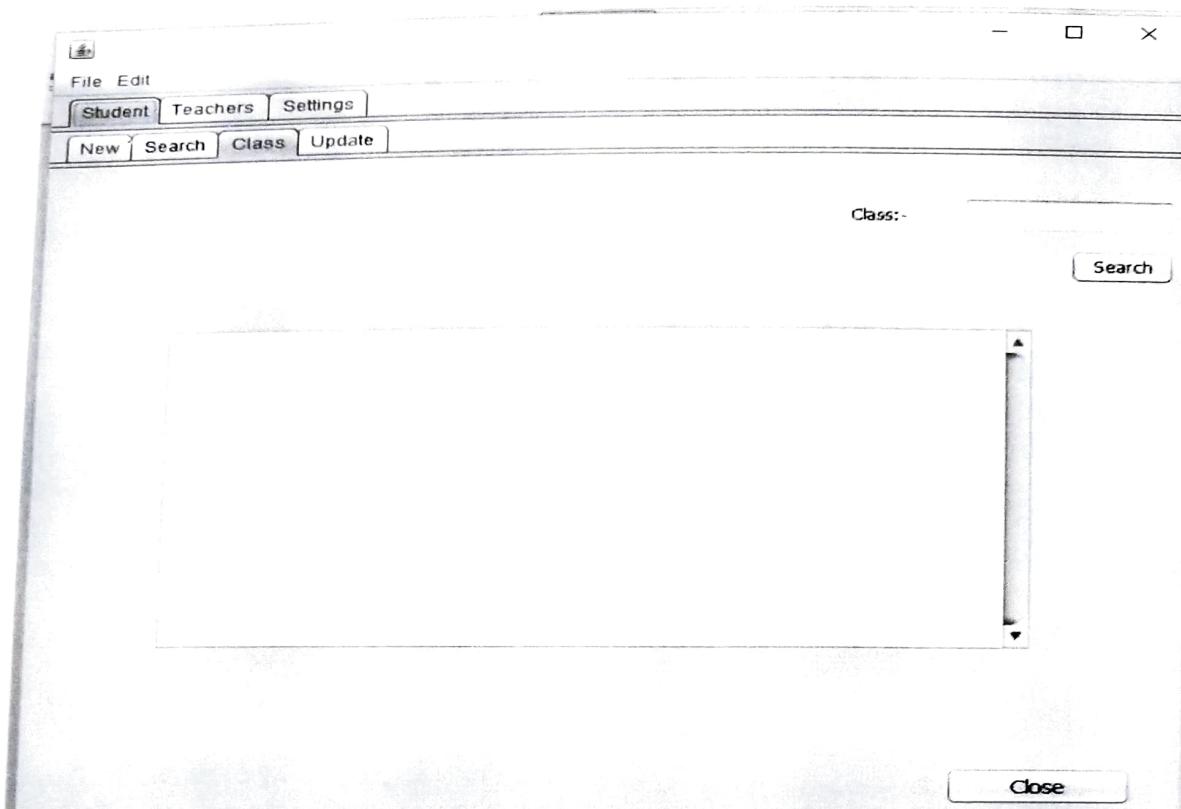
catch (ClassNotFoundException | SQLException e)

{

    System.out.println(e);

}
```

- CLASS STUDENT

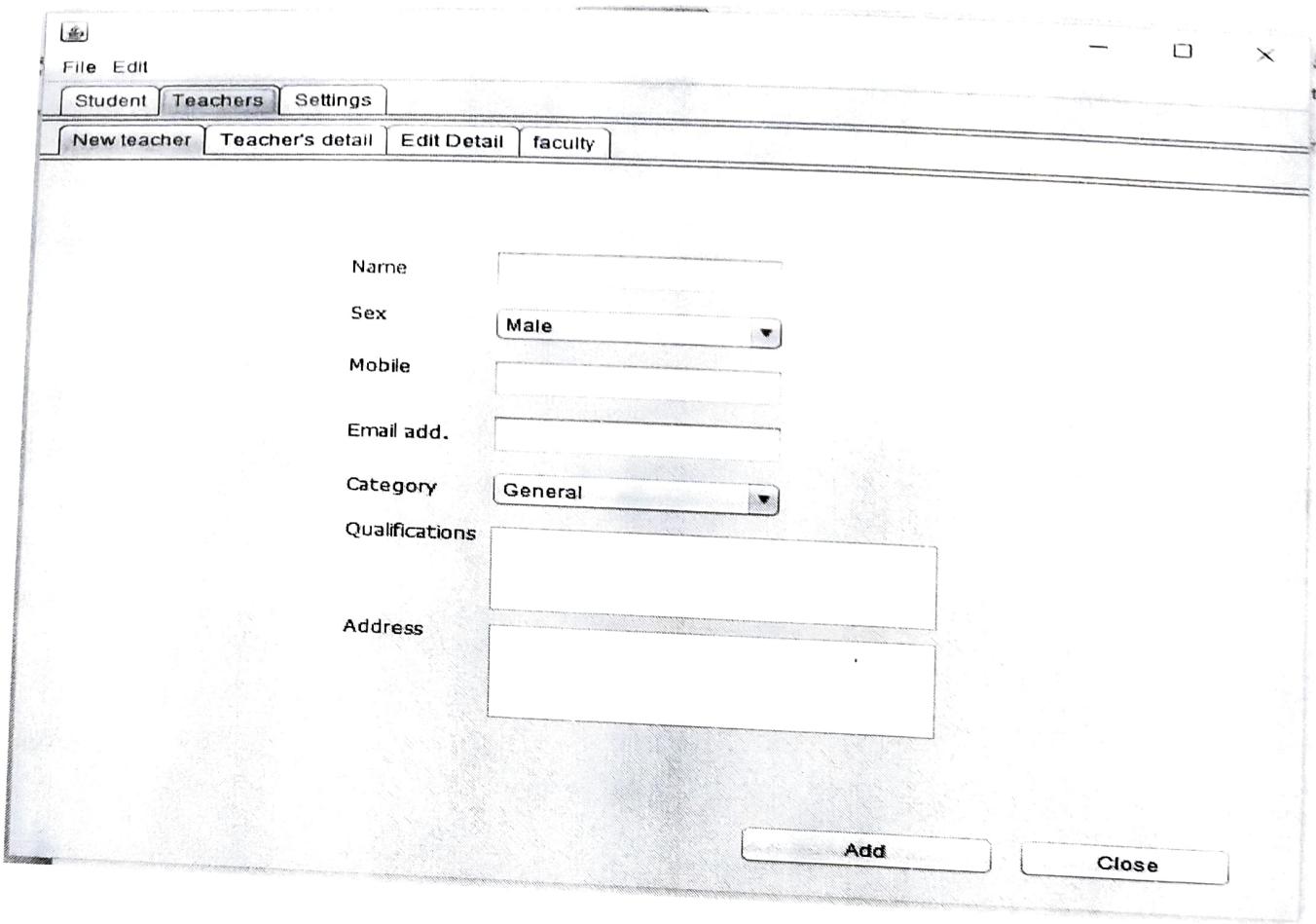


Search:-

```
private void jButton20ActionPerformed(java.awt.event.ActionEvent evt) {  
    try  
    {  
        Class.forName("com.mysql.jdbc.Driver");  
        Connection  
        con=DriverManager.getConnection("jdbc:mysql://localhost:3306/school","root","golu");  
        Statement stmt= con.createStatement();  
        ResultSet rs = stmt.executeQuery("select * from student");  
        int clas = Integer.parseInt(jTextField28.getText());  
        int a=0;  
        int clas1;
```

```
while(rs.next()) {  
    clas1=rs.getInt(12);  
    if(clas1==clas)  
        {  
        jTextArea9.append("Name:- "+rs.getString(2)+"\t\t "+"Id:- "+rs.getString(1)+"\n");  
        a=a+1;  
    }  
    if(a==0)  
    {  
        jTextArea9.setText("\n\n\n\n          no record found");  
    }  
}  
catch(ClassNotFoundException | SQLException e)  
{  
    System.out.println(e);  
}  
}
```

- NEW TEACHER TABLE



Add:-

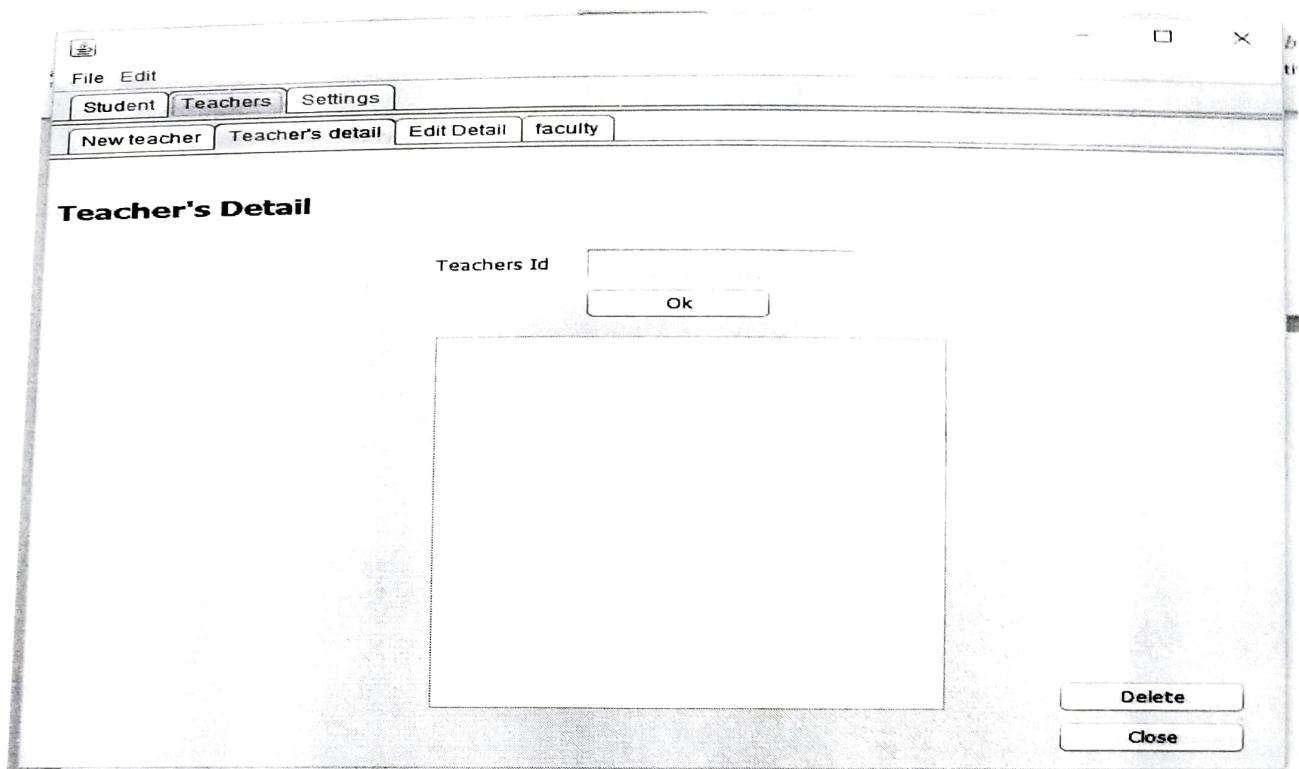
```

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/school","root","golu");
        PreparedStatement stmt = con.prepareStatement("insert into teacher
values(?,?,?,?,?,?,?,?,?)");
        ResultSet rs = stmt.executeQuery("select * from teacher");
    }
}
```

```
int id,flag=0,i;
for(i=0;i<=100;i++)
{
    Random r=new Random();
    int x=r.nextInt(100);
    while(rs.next())
    {
        id=rs.getInt(1);
        if(id==x)
        {
            flag=flag+1;
        }
    }
    if(flag==0)
    {
        String name=jTextField2.getText();
        String sex = jComboBox2.getSelectedItem().toString();
        String mob=jTextField20.getText();
        String email= jTextField21.getText();
        String category = jComboBox3.getSelectedItem().toString();
        String qualification= jTextArea3.getText();
        String address= jTextArea4.getText();
        stmt.setInt(1, x);
        stmt.setString(2, name);
        stmt.setString(3, sex);
        stmt.setString(4, mob );
        stmt.setString(5, email);
    }
}
```

```
        stmt.setString(6, category);
        stmt.setString(7, qualification);
        stmt.setString(8, address);
        jLabel38.setText(" teacher's ID="+x);
        System.out.println("1 record inserted");
        stmt.executeUpdate();
        stmt.close();
        con.close();
        break;
    }
}
catch (ClassNotFoundException | NumberFormatException e)
{
    System.out.println(e);
}
catch (SQLException ex)
{
    Logger.getLogger(main.class.getName()).log(Level.SEVERE, null, ex);
}
}
```

- TEACHER's DETAIL

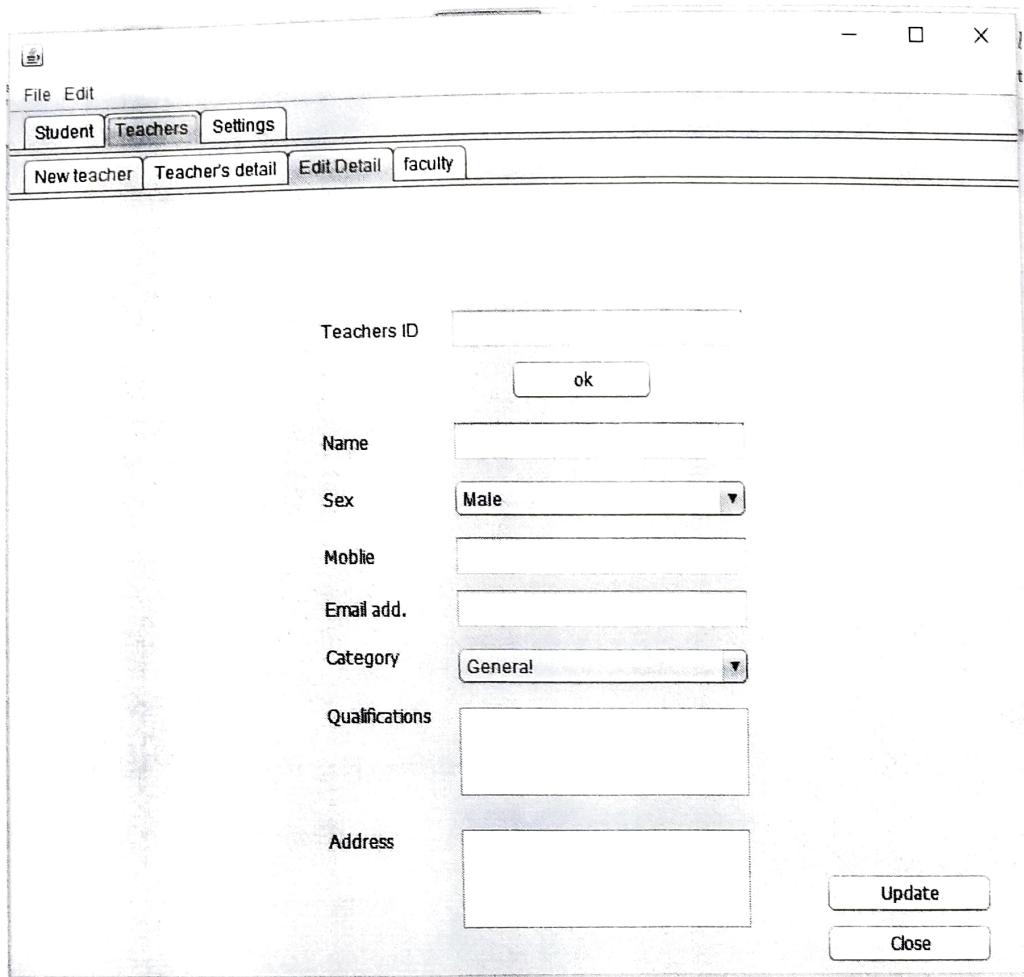


Ok :-

```
private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {  
    try  
    {  
        Class.forName("com.mysql.jdbc.Driver");  
        Connection  
        con=DriverManager.getConnection("jdbc:mysql://localhost:3306/school","root","golu");  
        Statement stmt= con.createStatement();  
        ResultSet rs = stmt.executeQuery("select * from teacher");  
        int teacher = Integer.parseInt(jTextField22.getText());  
        int teacher1;
```

```
int a=0;
while(rs.next())
{
teacher1=rs.getInt(1);
if(teacher1==teacher)
{
jTextArea5.setText("");
jTextArea5.append("Name:- "+rs.getString(2)+" \n"+ "Sex:- "+rs.getString(3)+"\n"+ "Mobile
no.: - "+rs.getString(4)+"\n"+ "Email ID:- "+rs.getString(5)+"\n"+ "Category:-
"+rs.getString(6)+"Qualifications: "+rs.getString(7)+" \n"+ "Address:- "+rs.getString(8));
a=a+1;
}}
if(a==0)
{
jTextArea5.setText("no record found");
}}
catch(ClassNotFoundException | SQLException e)
{
System.out.println(e);
}}
```

• EDIT DETAIL(TEACHER)



Ok:-

```
private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {  
    try  
    {  
        Class.forName("com.mysql.jdbc.Driver");  
        Connection  
        con=DriverManager.getConnection("jdbc:mysql://localhost:3306/school","root","golu");  
        Statement stmt= con.createStatement();  
        ResultSet rs = stmt.executeQuery("select * from teacher");  
        int id = Integer.parseInt(jTextField23.getText());
```

```
int id1;

while(rs.next())
{
    id1=rs.getInt(1);

    if(id1==id)
    {
        String name=rs.getString(2);

        String mob=rs.getString(4);

        String email=rs.getString(5);

        String qualification=rs.getString(7);

        String address=rs.getString(8);

        jTextField24.setText(name);

        jTextField25.setText(String.valueOf(mob));

        jTextField26.setText(email);

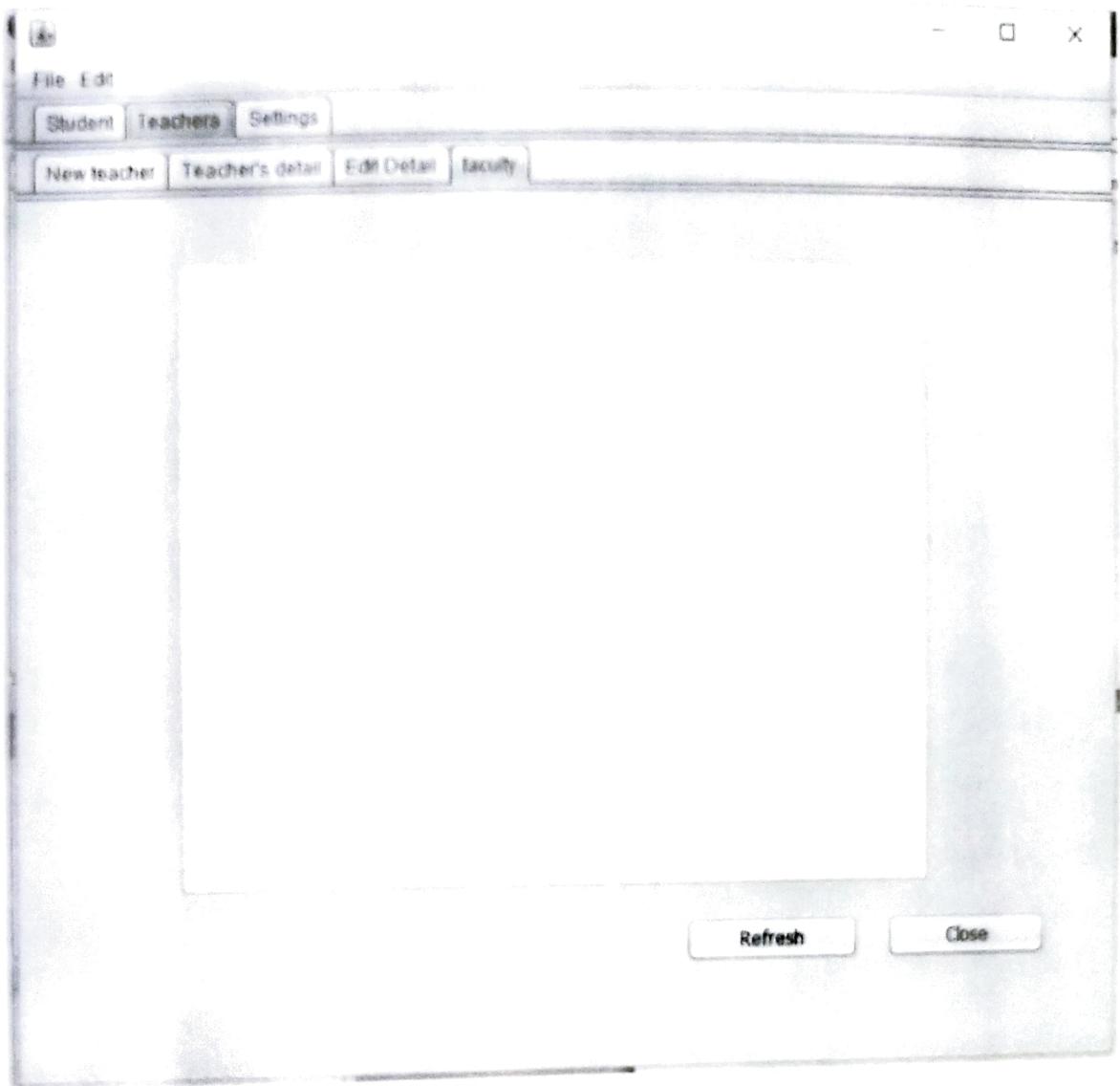
        jTextArea6.setText(qualification);

        jTextArea7.setText(address);

    }
}

catch(ClassNotFoundException | SQLException e)
{
    System.out.println(e);
}
```

• FACULTY TEACHER



Refresh:-

```
private void jButton11ActionPerformed(java.awt.event.ActionEvent evt) {  
    try  
    {  
        Class.forName("com.mysql.jdbc.Driver");  
        Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/school","root","golu");  
        Statement stmt= con.createStatement();  
        ResultSet rs = stmt.executeQuery("select * from teacher");  
    }  
}
```

```
while(rs.next())
{
jTextArea10.append("Name:- "+rs.getString(2)+"\t\t"+ "Id:- "+rs.getString(1)+"\n");
}
catch(ClassNotFoundException | SQLException e)
{
System.out.println(e);
}
```

CHAPTER 3

SOFTWARE USED:

MY SQL

MySQL is an open source relational database management system (RDBMS).^[6] Its name is a combination of "My", the name of co founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation).^[8] In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Google(though not for searches), Facebook, Twitter, Flickr,^[1] and YouTube.

Release history

Release	General availability	Latest minor version	Latest release	End of support ^[45]
5.1	November 14, 2008; 10 years ago	5.1.73	2013-12-03	December 2013
5.5	December 3, 2010; 8 years ago	5.5.62	2018-10-22	December 2018
5.6	February 5, 2013; 6 years ago ^[46]	5.6.43	2019-01-21	February 2021
5.7	October 21, 2015; 3 years ago	5.7.25	2019-01-21	October 2023
8.0	April 19, 2018; 11 months ago	8.0.15	2019-02-01	April 2026

Features

MySQL is offered under two different editions: the open source MySQL Community Server and the proprietary Enterprise Server.^[1] MySQL Enterprise Server is differentiated by a series

of proprietary extensions which install as server plugins, but otherwise shares the version numbering system and is built from the same code base.

Major features as available in MySQL 5.6:

- A broad subset of ANSI SQL 99, as well as extensions
- Cross-platform support
- Stored procedures, using a procedural language that closely adheres to SQL/PSM
- Triggers
- Cursors
- Updatable views
- Online Data Definition Language (DDL) when using the InnoDB Storage Engine.
- Information schema
- Performance Schema that collects and aggregates statistics about server execution and query performance for monitoring purposes.^[76]
- A set of SQL Mode options to control runtime behavior, including a strict mode to better adhere to SQL standards.
- X/Open XA distributed transaction processing (DTP) support; two phase commit as part of this, using the default InnoDB storage engine
- Transactions with savepoints when using the default InnoDB Storage Engine. The NDB Cluster Storage Engine also supports transactions.
- ACID compliance when using InnoDB and NDB Cluster Storage Engines
- SSL support
- Query caching
- Sub-SELECTs (i.e. nested SELECTs)
- Built-in replication support
 - Asynchronous replication: master-slave from one master to many slaves or many masters to one slave
 - Semi synchronous replication: Master to slave replication where the master waits on replication
 - Synchronous replication: Multi-master replication is provided in MySQL Cluster^[83].
 - Virtual Synchronous: Self managed groups of MySQL servers with multi master support can be done using: Galera Cluster or the built in Group Replication plugin
- Full-text indexing and searching
- Embedded database library
- Unicode support
- Partitioned tables with pruning of partitions in optimizer
- Shared-nothing clustering through MySQL Cluster
- Multiple storage engines, allowing one to choose the one that is most effective for each table in the application.
- Native storage engines InnoDB, MyISAM, Merge, Memory (heap), Federated, Archive, CSV, Blackhole, NDB Cluster.
- Commit grouping, gathering multiple transactions from multiple connections together to increase the number of commits per second.

Java NetBeans allows applications to be developed from a set of modular software components called *modules*. NetBeans runs on Windows, macOS, Linux and Solaris. In addition to Java development, it has extensions for other languages like PHP, C, C++, HTML5, and JavaScript. Applications based on NetBeans, including the NetBeans IDE, can be extended by third party developers.

NetBeans IDE releases

NetBeans IDE 6.0 introduced support for developing IDE modules and rich client applications based on the NetBeans platform, a Java Swing GUI builder (formerly known as "Project Matisse"), improved CVS support, WebLogic 9 and JBoss 4 support, and many editor enhancements. NetBeans 6 is available in official repositories of major Linux distributions.

NetBeans IDE 6.5, released in November 2008, extended the existing Java EE features (including Java Persistence support, EJB 3 and JAX-WS). Additionally, the NetBeans Enterprise Pack supports the development of Java EE 5 enterprise applications, including SOA visual design tools, XML schema tools, web services orchestration (for BPEL), and UML modeling. The NetBeans IDE Bundle for C/C++ supports C/C++ and FORTRAN development.

NetBeans IDE 6.8 is the first IDE to provide complete support of Java EE 6 and the GlassFish Enterprise Server v3. Developers hosting their open-source projects on kenai.com additionally benefit from instant messaging and issue tracking integration and navigation right in the IDE, support for web application development with PHP 5.3 and the Symfony framework, and improved code completion, layouts, hints and navigation in JavaFX projects.

NetBeans IDE 6.9, released in June 2010, added support for OSGi, Spring Framework 3.0, Java EE dependency injection (JSR-299), Zend Framework for PHP, and easier code navigation (such as "Is Overridden/Implemented" annotations), formatting, hints, and refactoring across several languages.

NetBeans IDE 7.0 was released in April 2011. On August 1, 2011, the NetBeans Team released NetBeans IDE 7.0.1, which has full support for the official release of the Java SE 7 platform.

NetBeans IDE 7.3 was released in February 2013 which added support for HTML5 and web technologies.^[13]

NetBeans IDE 7.4 was released on 15 October 2013.

NetBeans IDE 8.0 was released on 18 March 2014.

NetBeans IDE 8.1 was released on 4 November 2015.

NetBeans IDE 8.2 was released on 3 October 2016.

Netbeans 9.0, which adds support for Java 9 and 10, was released on 29 July 2018, by the Apache Incubator project.

HARDWARE

Display and Configuration

The Asus Vivobook X541U-DM846D flaunts a 15.6-inch HD LED backlit display which has a screen resolution of 1,366 x 768 pixels. Under the hood, there is a 2GHz Intel Core i3 processor from the sixth generation which acts as the brain of the system. It is armed with a 4GB of DDR4 RAM and an integrated Intel HD Graphics card. The laptop runs on DOS operating system sailing upon 64-bit architecture.

Camera and Battery

A 720p HD webcam is present above the display screen. It captures good quality videos. The user can enjoy video calling and shoot videos when needed. The laptop gets its energy from a 3 cell Li-ion battery which keeps the show running for long hours.

Storage and Connectivity

In terms of storage, the user can create a library of files in a 1TB SATA hard disk which performs the task at a speed of 5,400RPM. On the connectivity ground, the Asus Vivobook X541U-DM846D features a lot of options such as Wireless LAN 802.11 b/g/n, Bluetooth v4.0, one USB 3.0 slot, one USB 2.0 slot, an SD card reader, a headphone jack, a microphone jack and a VGA port.

Input Devices and Body Looks

Like other laptops, the Asus Vivobook X541U-DM846D also features common input devices say a touchpad with multi-touch gesture support, a chiclet keyboard and a DVD writer. Design wise, the laptop comes in a silver coloured body measuring 381.4 x 251.5 x 27.6mm and weighing 2Kg.

LANGUAGES

JAVA

Java is a general-purpose computer-programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to "bytecode" that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The language derives much of its original features from SmallTalk, with a syntax similar to C and C++, but it has fewer low-level facilities than either of them. As of 2018, Java was according to Github one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by a Canadian James Gosling at Sun Microsystems (which has since been acquired by Oracle) and released in 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GNU General Public License. Meanwhile, others have developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Classpath (standard libraries), and IcedTea-Web (browser plugin for applets).

The latest version is Java SE 12, released in March 2019. Since Java 9 is no longer supported, Oracle advises its users to "immediately transition" to Java 12. Oracle released the last public update for the legacy Java 8 LTS, which is free for commercial use, in January 2019. Java 8 will be supported with public updates for personal use up to at least December 2020. Oracle and others "highly recommend that you uninstall older versions of Java"^[21] because of serious risks due to unresolved security issues

MYSQL

MySQL is an open source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

CHAPTER 4

OPERATIONS AND APPLICATIONS: OPERATIONS

In this software we can perform following operations

Insertion of data

There are two basic syntaxes of the INSERT INTO statement which are shown below.

```
INSERT INTO TABLE_NAME (column1, column2, column3,...columnN)
VALUES (value1, value2, value3,...valueN);
```

Here, column1, column2, column3,...columnN are the names of the columns in the table into which you want to insert the data.

You may not need to specify the column(s) name in the SQL query if you are adding values for all the columns of the table. But make sure the order of the values is in the same order as the columns in the table.

The **SQL INSERT INTO** syntax will be as follows –

```
INSERT INTO TABLE_NAME VALUES (value1,value2,value3,...value
```

Deletion of data

Syntax

The basic syntax of the DELETE query with the WHERE clause is as follows –

```
DELETE FROM table_name
WHERE [condition];
```

You can combine N number of conditions using AND or OR operators.

Updation of data

Syntax

The basic syntax of the UPDATE query with a WHERE clause is as follows –

```
UPDATE table_name
SET column1 = value1, column2 = value2...., columnN = valueN
WHERE [condition];
You can combine N number of conditions using the AND or the OR operators.
```

APPLICATION

- We can use this software in schools.
- We can use this software in coaching institute.
- We can use this software in tuitions.
- We can use this software in various places for storing database of any institution and in organization.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE:

CONCLUSION

- We can save our details on this software.
- An unauthorized person cannot access it.
- We can add unlimited data and details and data will be secured.
- Future updates can be done.
- Easy to access.

FUTURE SCOPE

- If we enter class no. 8 then it will show the number of students and their details of class 8.
- We can add more subjects .
- We can add passport size photos and signature for details.
- We can also print results directly.
- We can make the interface more attractive.

REFERENCES

<https://www.google.com/search?q=WIKIPEDIA+MY+SQL&oq=WIKIPEDIA+MY+SQL&aqs=chrome..69i57j0l5.8083j0j8&sourceid=chrome&ie=UTF-8>

https://www.google.com/search?ei=zJa6XLb_Kqffz7sPxLKKQA&q=wikipedia+java&oq=WIKIPEDIA+JAVA&gs_l=psy-ab.1.0.0l10.54042.59065..60949...2.0..0.450.2237.0j7j1j1j1.....0....1..gws-wiz.....0i71.ox1WaFkgu00

<https://en.wikipedia.org/wiki/NetBeans>