"COACHING MANAGEMENT SYSTEM "

Major Project Report

Submitted in partial fulfillment of the requirements for the

Award of degree of Bachelor of Computer Applications

2016-2020

Submitted by
Vibhor Jain (0161BCA067)

Guided by
Mrs. Gargi Mukherjee

BHARATI VIDYAPEETH
(DEEMED TO BE UNIVERSITY)
INSTITUTE OF MANAGEMENT & RESEARCH, NEW DELHI
A-4, Paschim Vihar, New Rohtak Road, New Delhi-110063

TITLE OF THE PROJECT

"Coaching Management System"

INTRODUCTION

I have decide to start the project of "**Coaching Management system**" come over in my mind when I saw my cousin brother institute where I see all the jobs is done manually. He maintain all the student data in register. The record of student like batch time ,fees due, batch days ,class, subject are handled manually. If he want to see any due date of fees then all the records will be seen one by one. This is laborious and dull process. The entire work of institute is depending upon only one man. Like search ,update , delete view and add the student information are difficult. Current approach of job is based on manual system in which all the details is first received and then entered in the register. When any new student take admission all the record is stored into the register. The most difficult part is to store the record of fees manually. One thing I will also notes that how he managed the marks of monthly test he taken. He managed the monthly test by writing the name , date , marks , subject, class into the register. This project was made so that coaching work comfortable and responsive. The project will hold the information of the coaching. It come up with provisions remain the record the overall tasks of student in coaching center.

OBJECTIVES

This project is build on java and RDBMS technology. The main objectives of our work are:

- It will reduce the load work .

- It will save paper work and save time .
- We can find any record easily.
- To change and update record is simple task.
- This system can provide the flexible report.
- It will also reduce Data redundancy**.**
- Centralized database system.
- Data can be share.

PROJECT CATEGORY

The project "**Coaching Management System**" is creates under **RDBMS[Relational Database Management System].** The application is progress with the help of Net Beans java and mysql.

MODULES AND THEIR DESCRIPTION

- **User Management:-**

This model deals with the management of data stored by maintaining the following details :

- Storing details of data stored by various independently.
- Maintaining the details about data .

- **USER REGISTRATER:-**

This model deals with the management of user personal details and user account information.

- It first verifies and then stores the details of various users permanently for future references.
- Assignment of a unique login ID i.e. user name and password.
- This user authentication is made on the basis of the data stored in the user details database.

- **ADMINISTRATOR:-**

Administrator performs the administrative activities for the application:-

- Admin can view the personal details of any registered user.
- Administrator can delete or modify any category.

REPORT GENERATION

- Student Details
- Fees Details
- Class Details
- Subject Details
- Time Details
- Batch Day Details

DATA MODELING

Registration table

FIELD	TYPE
Name	char(20)
Address	char(20)
Emilid	varchar(20)
Phoneno	Int(11)

Classes	Int(11)
Stream	Char(20)
Subject	Char(20)

Login table

FIELD	TYPE
user name	Char(20)
Password	Char(20)

TOOLS AND PLATFORM

Hardware Requirement:

Processor: intel CORE i5

RAM: 8GB

HD: 1 TB

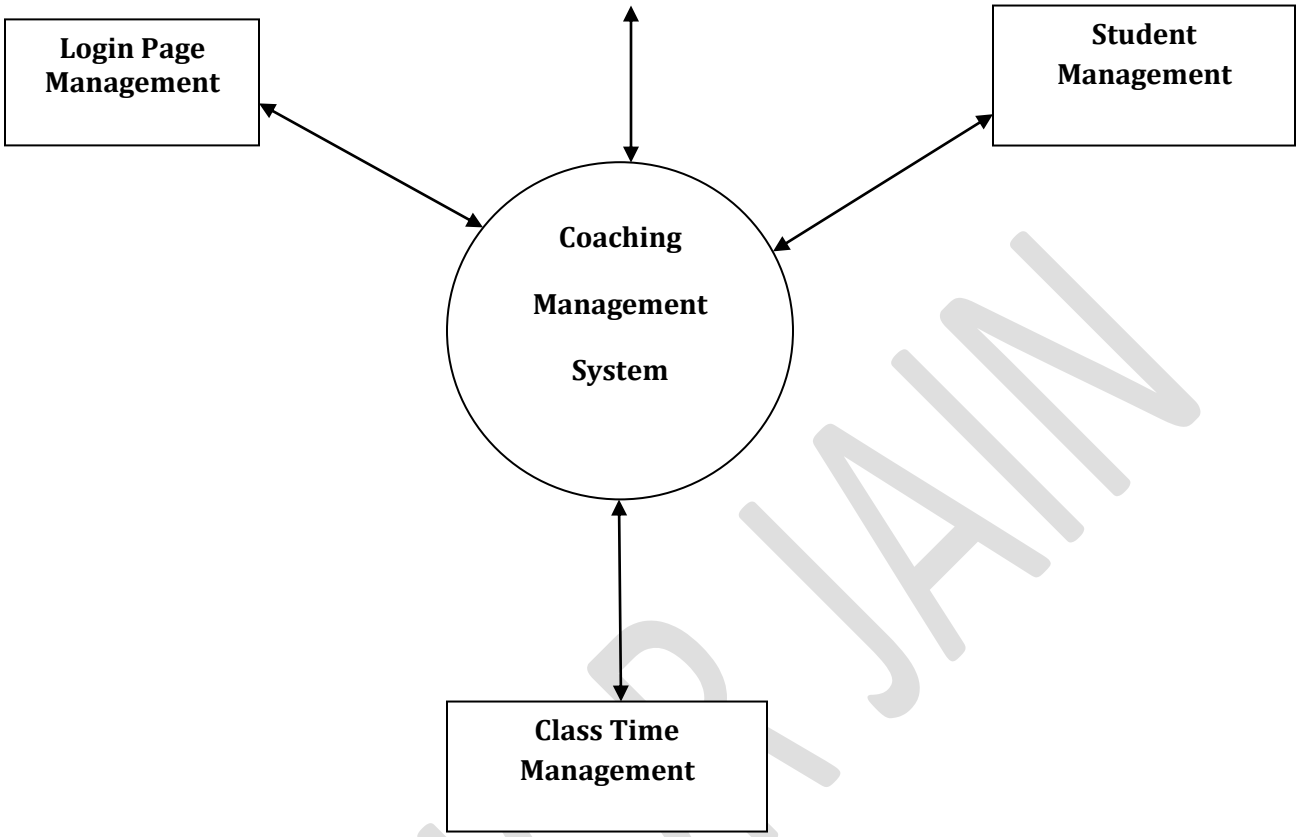
Software Requirement:

Operating System : Windows 10

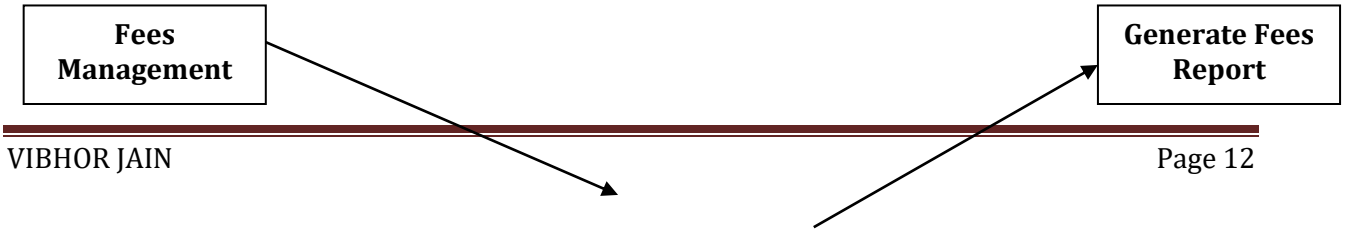
Database : Mysql

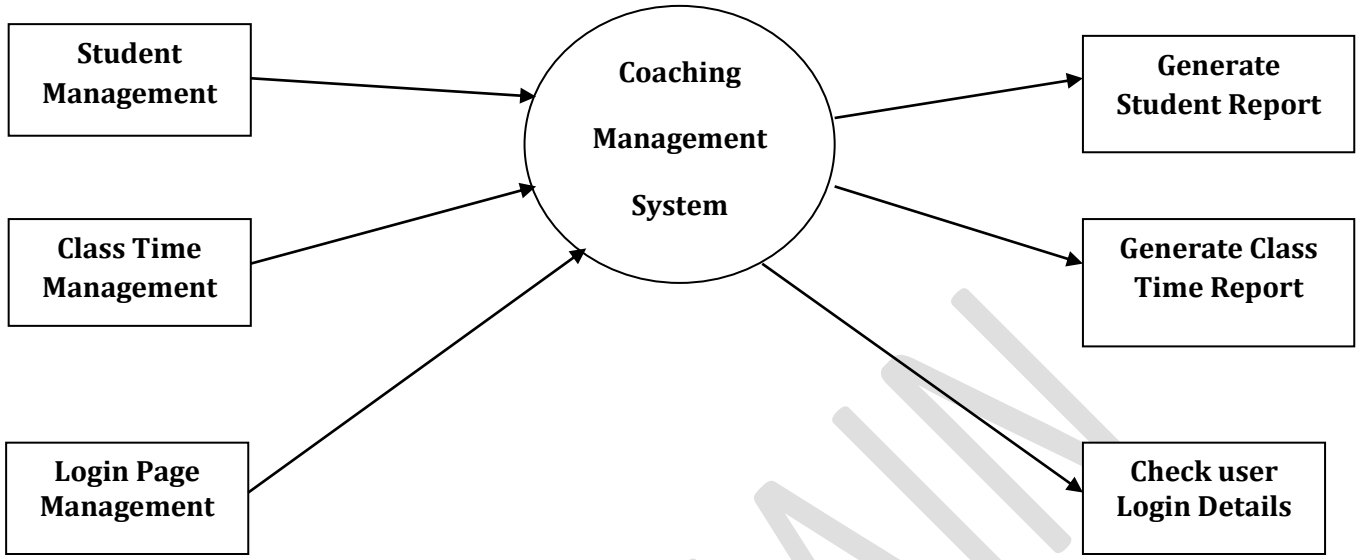
Java (NetBeans IDE)

DATAFLOW DIAGRAM

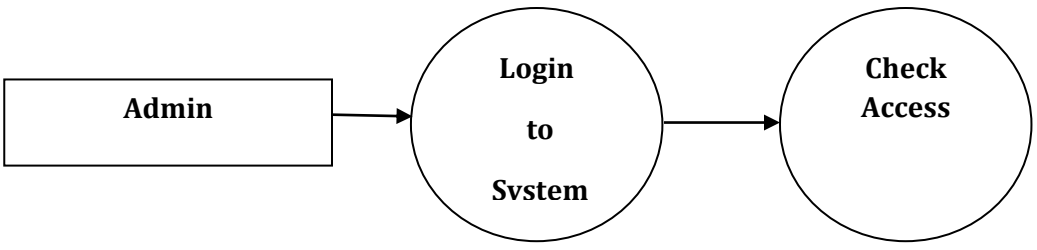


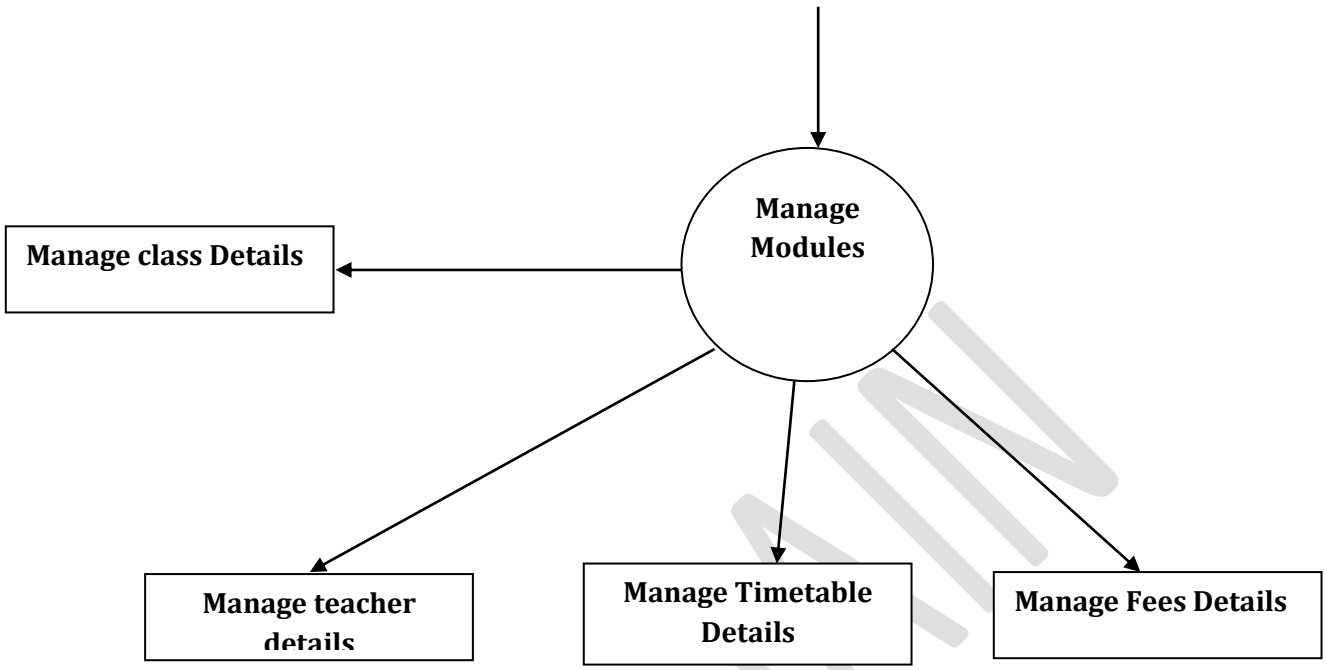
Zero Level DFD -Coaching Management System





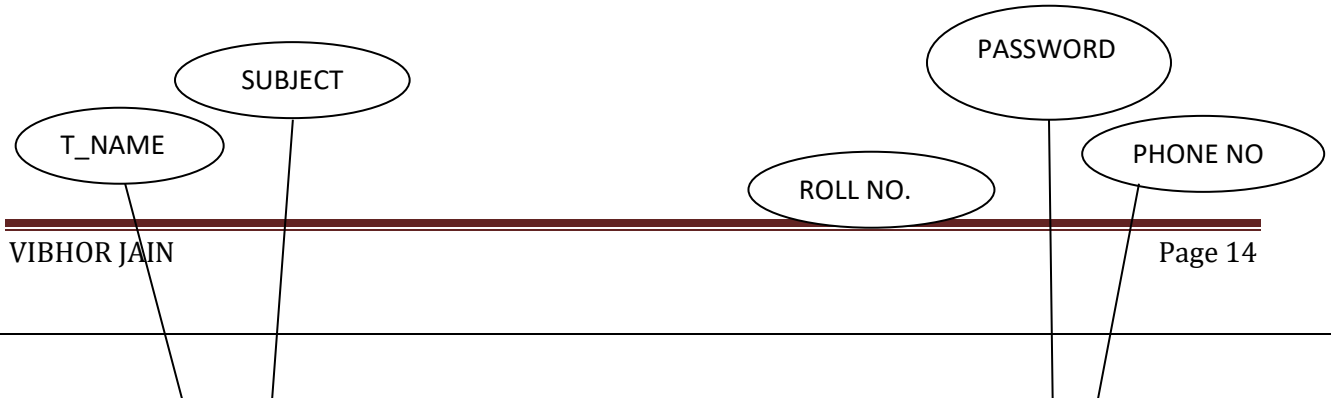
First Level DFD -Coaching Management System

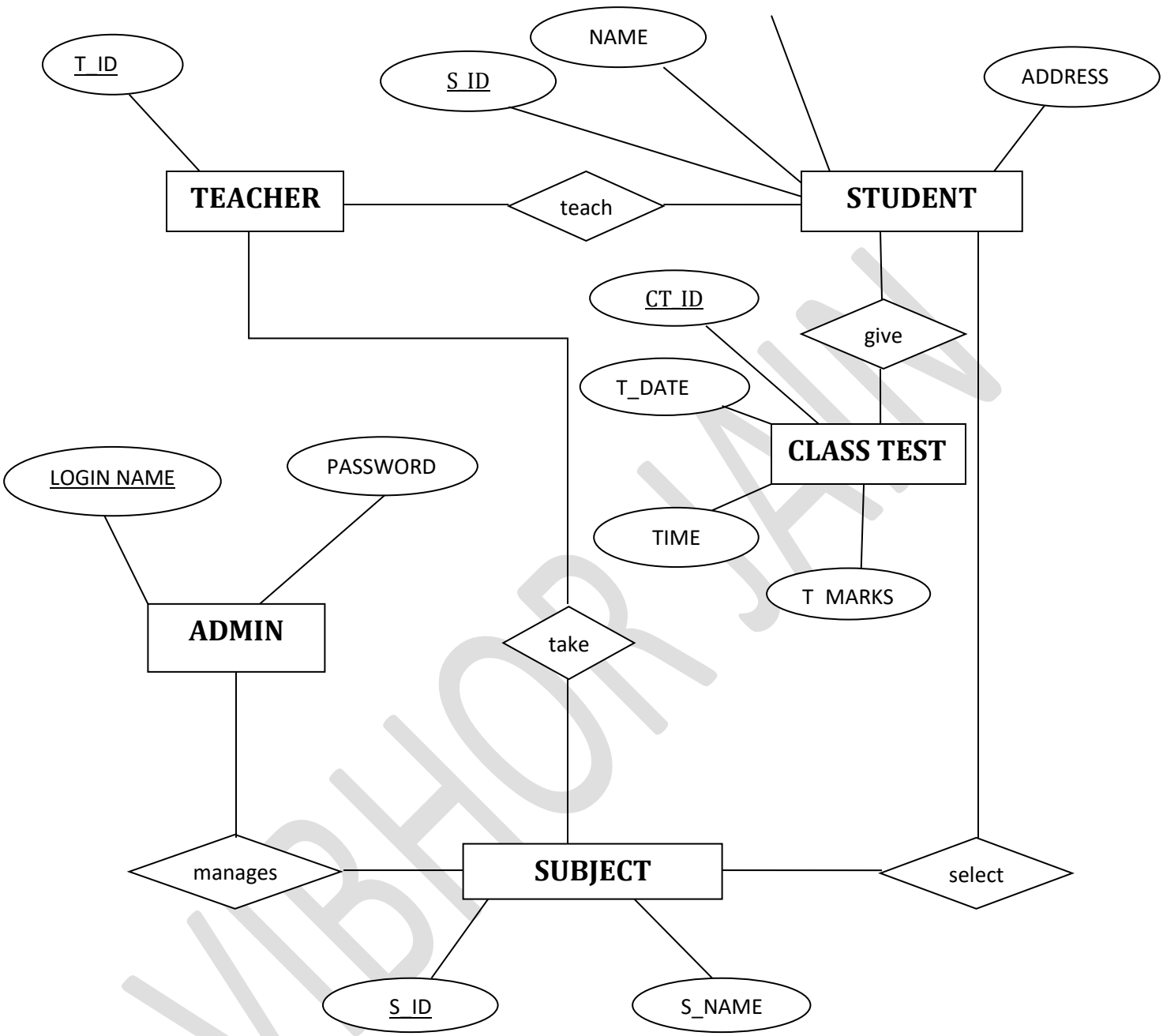




Second Level DFD -Coaching Management System

E-R Diagram



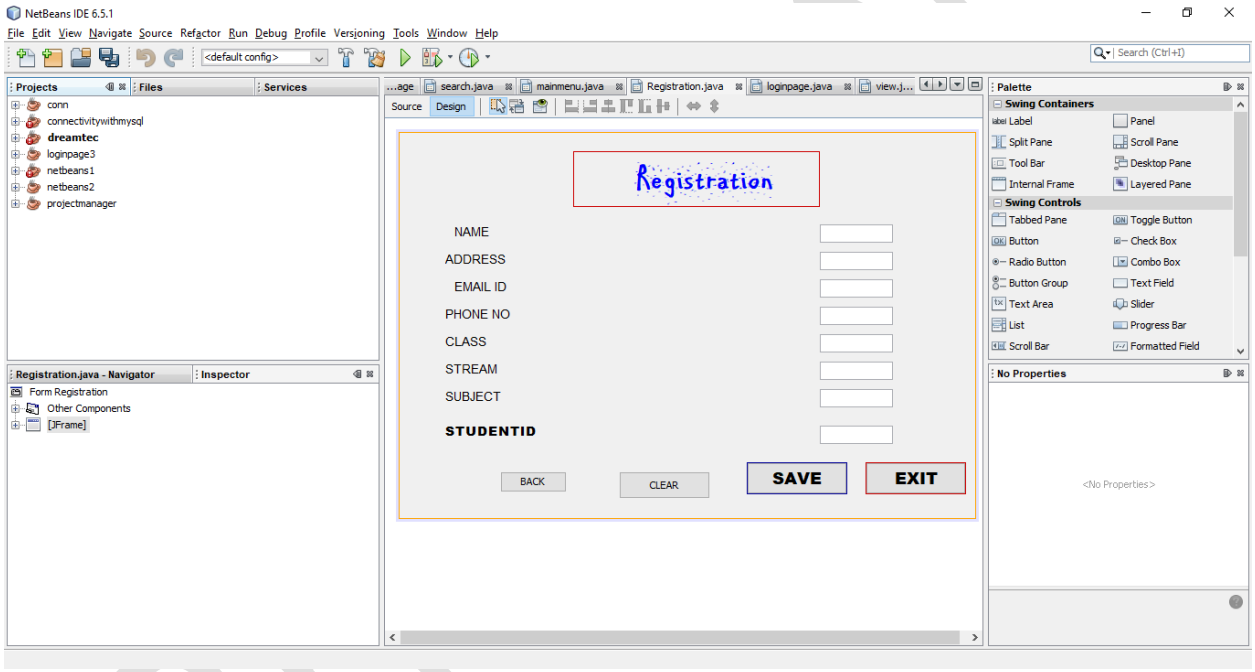


Are you doing this project for any Industry/Client?

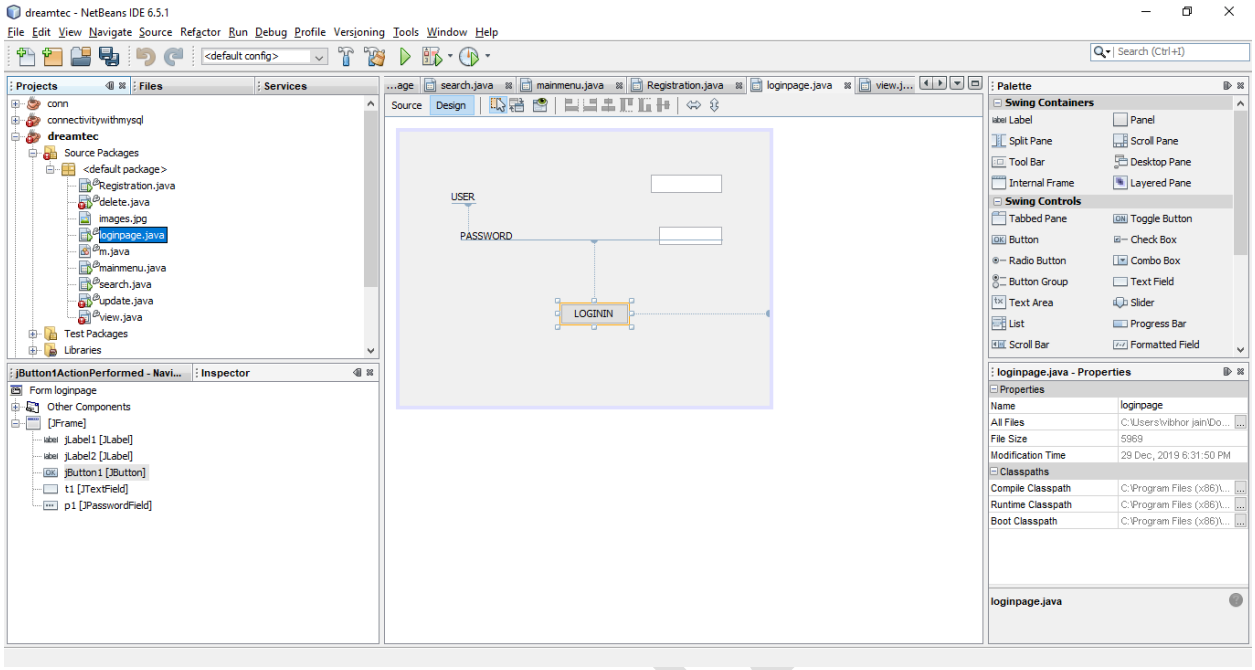
No, I am not doing this project for any industry.

Interface Design (Screen shots of forms)

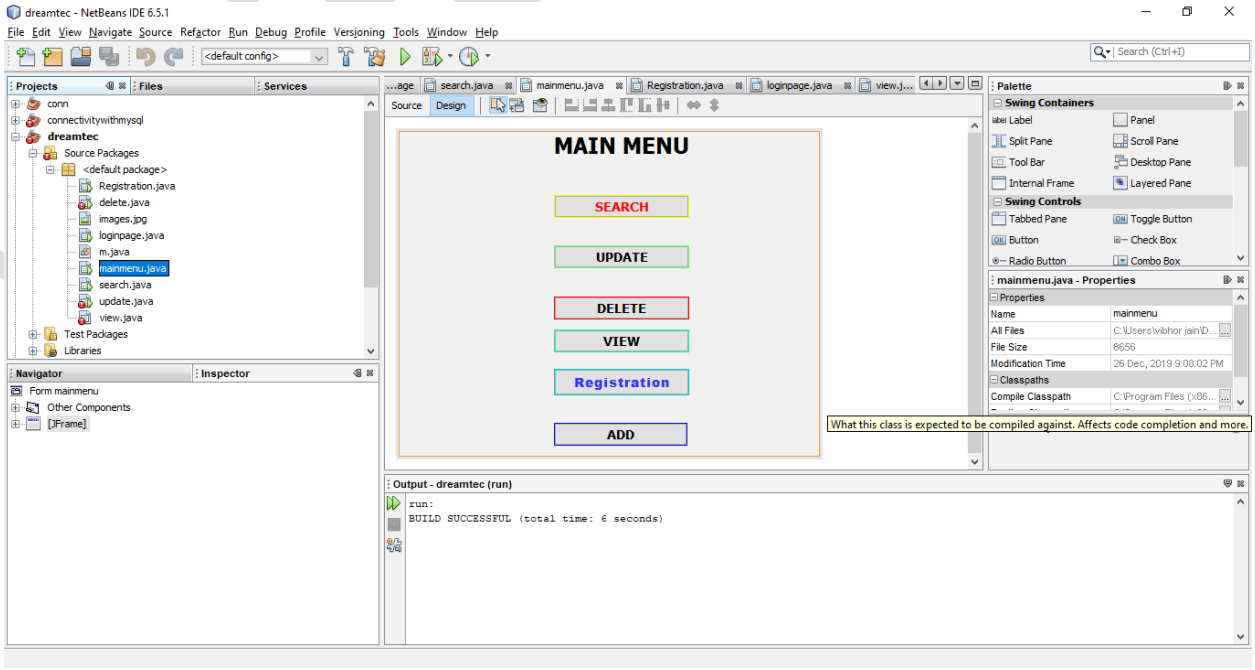
1 REGISTRATION FORM



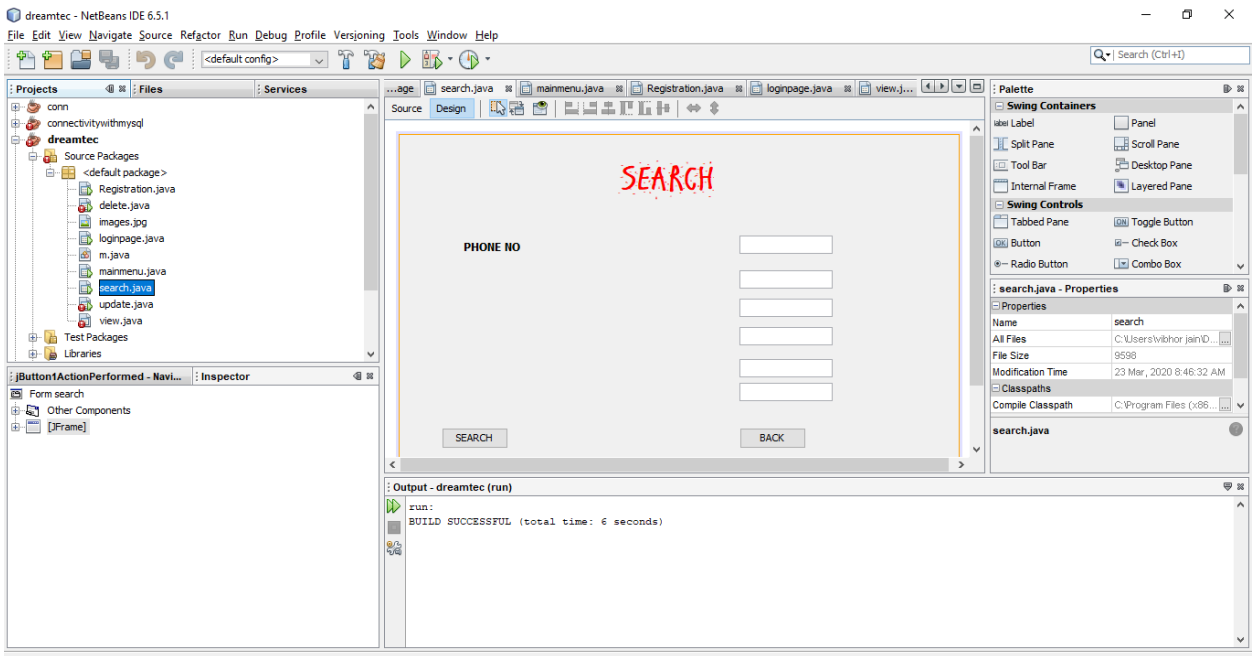
2 login page


dreamtec - NetBeans IDE 6.5.1
File Edit View Navigate Source Refactor Run Debug Profile Versioning Tools Window Help
Projects
Files
Services
conn
connectivitywithmysql
dreamtec
Source Packages
<default package>
Registration.java
delete.java
images.jpg
loginpage.java
m.java
mainmenu.java
search.java
update.java
view.java
Test Packages
Libraries
Source
Design
USER
PASSWORD
LOGININ
Palette
Swing Containers
Swing Controls
loginpage.java - Properties
Properties
Name
loginpage
Classpaths
Inspector
Form loginpage
Other Components
[JFrame]
jLabel1 [JLabel]
jLabel2 [JLabel]
jButton1 [JButton]
t1 [JTextField]
p1 [JPasswordField]

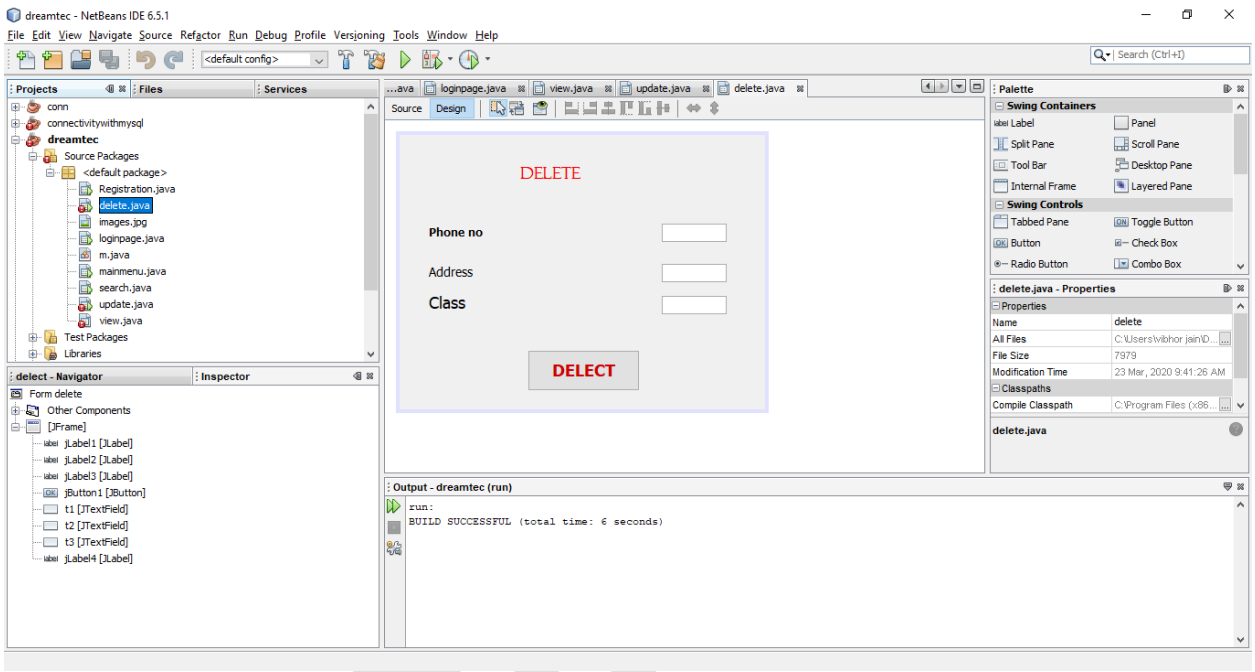
3 main menu


dreamtec - NetBeans IDE 6.5.1
File Edit View Navigate Source Refactor Run Debug Profile Versioning Tools Window Help
MAIN MENU
SEARCH
UPDATE
DELETE
VIEW
Registration
ADD
mainmenu.java - Properties
What this class is expected to be compiled against. Affects code completion and more.
Output - dreamtec (run)
run:
BUILD SUCCESSFUL (total time: 6 seconds)

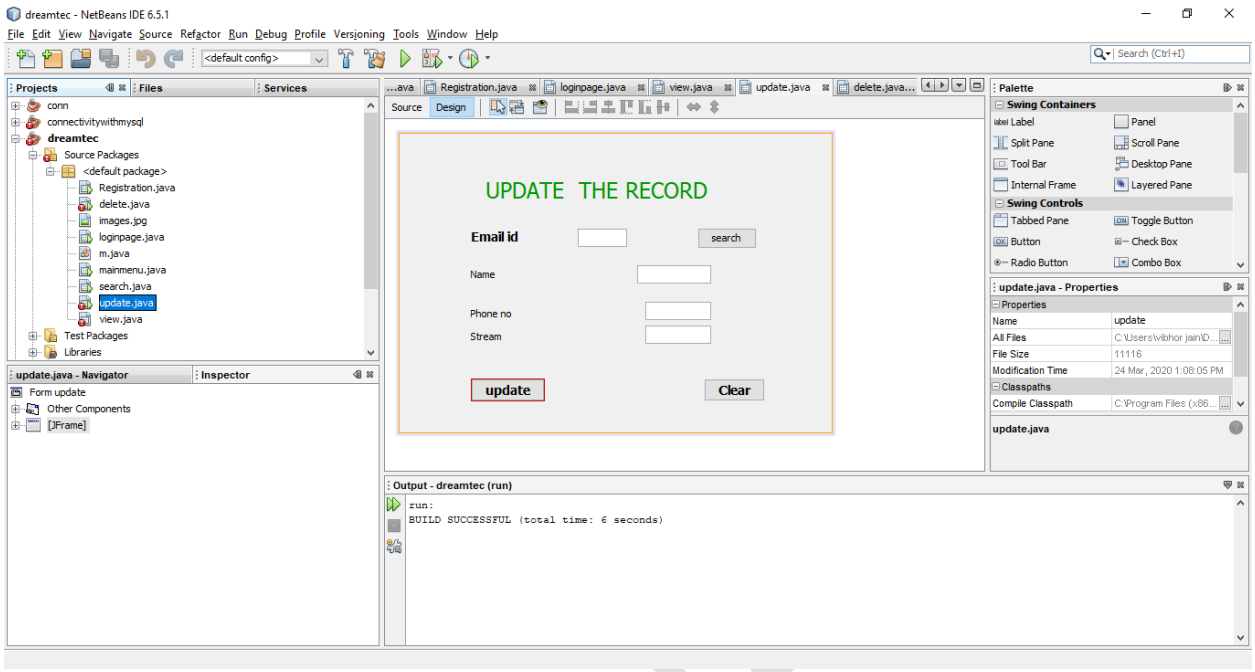
4 search



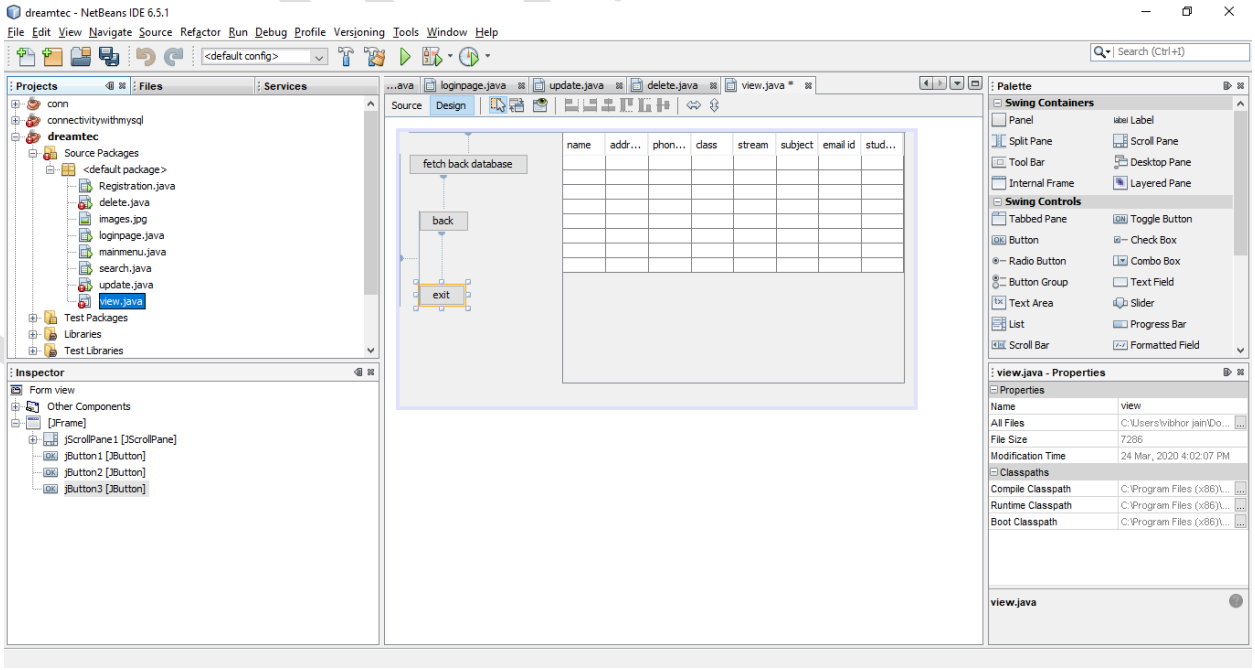
5 delete



6 update

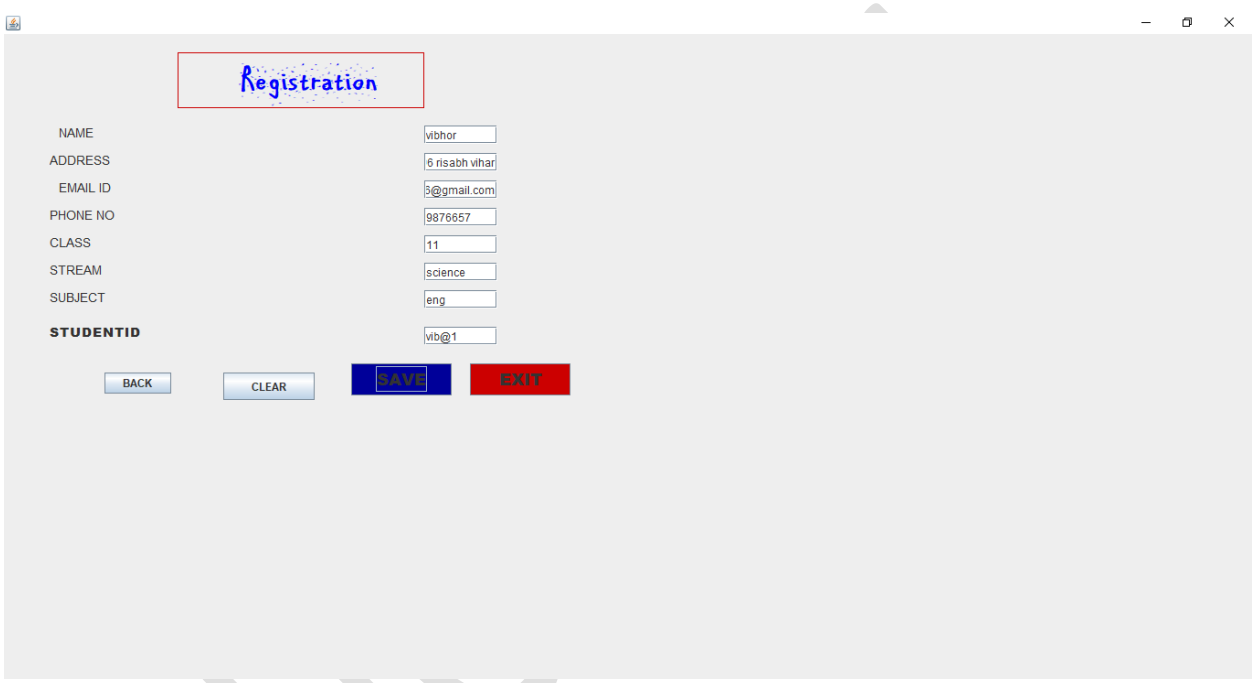

dreamtec - NetBeans IDE 6.5.1
File Edit View Navigate Source Refactor Run Debug Profile Versioning Tools Window Help
Projects
Files
Services
conn
connectivitywithmysql
dreamtec
Source Packages
<default package>
Registration.java
delete.java
images.jpg
loginpage.java
m.java
mainmenu.java
search.java
update.java
view.java
Test Packages
Libraries
update.java - Navigator
Inspector
Form update
Other Components
[JFrame]
Source
Design
UPDATE THE RECORD
Email id
search
Name
Phone no
Stream
update
Clear
Palette
Swing Containers
Swing Controls
update.java - Properties
Properties
Name
update
Output - dreamtec (run)
run:
BUILD SUCCESSFUL (total time: 6 seconds)

7 view

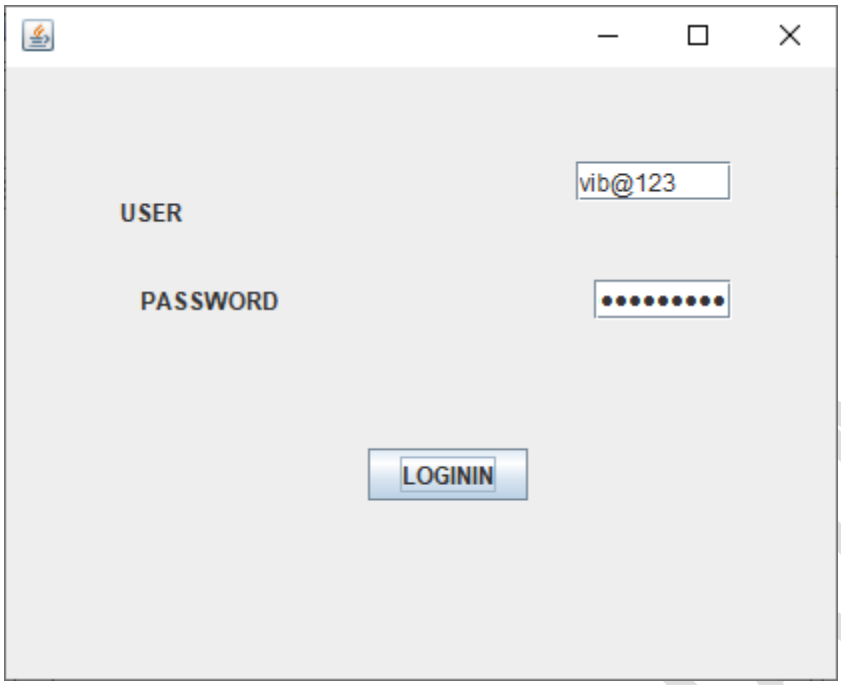

dreamtec - NetBeans IDE 6.5.1
File Edit View Navigate Source Refactor Run Debug Profile Versioning Tools Window Help
Projects
Files
Services
conn
connectivitywithmysql
dreamtec
Source Packages
<default package>
Registration.java
delete.java
images.jpg
loginpage.java
mainmenu.java
search.java
update.java
view.java
Test Packages
Libraries
Test Libraries
Inspector
Form view
Other Components
[JFrame]
jScrollPane1 [JScrollPane]
jButton1 [JButton]
jButton2 [JButton]
jButton3 [JButton]
Source
Design
fetch back database
back
exit
name
addr...
phon...
class
stream
subject
email id
stud...
Palette
Swing Containers
Swing Controls
view.java - Properties
Properties
Name
view

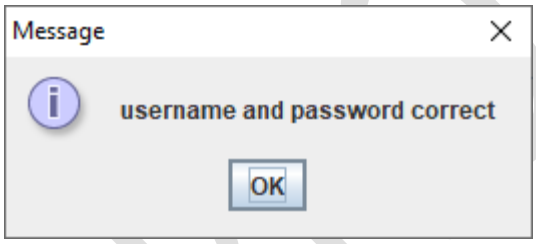
Output Design (Screen Shots of Report)

1 REGISTRATION FORM

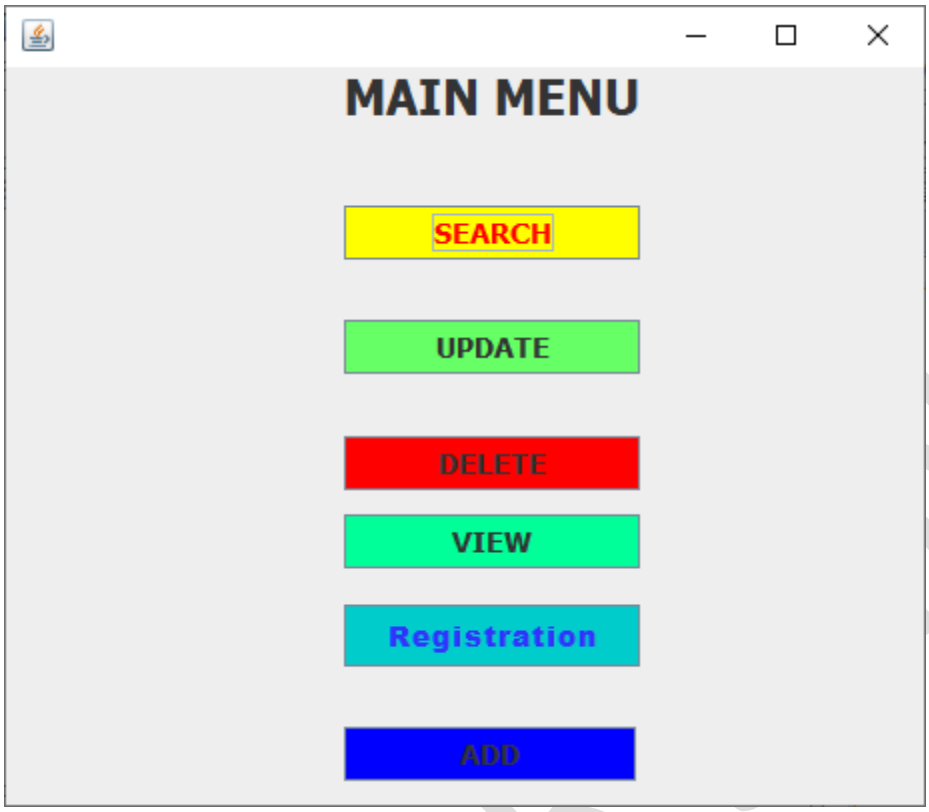


2 login page

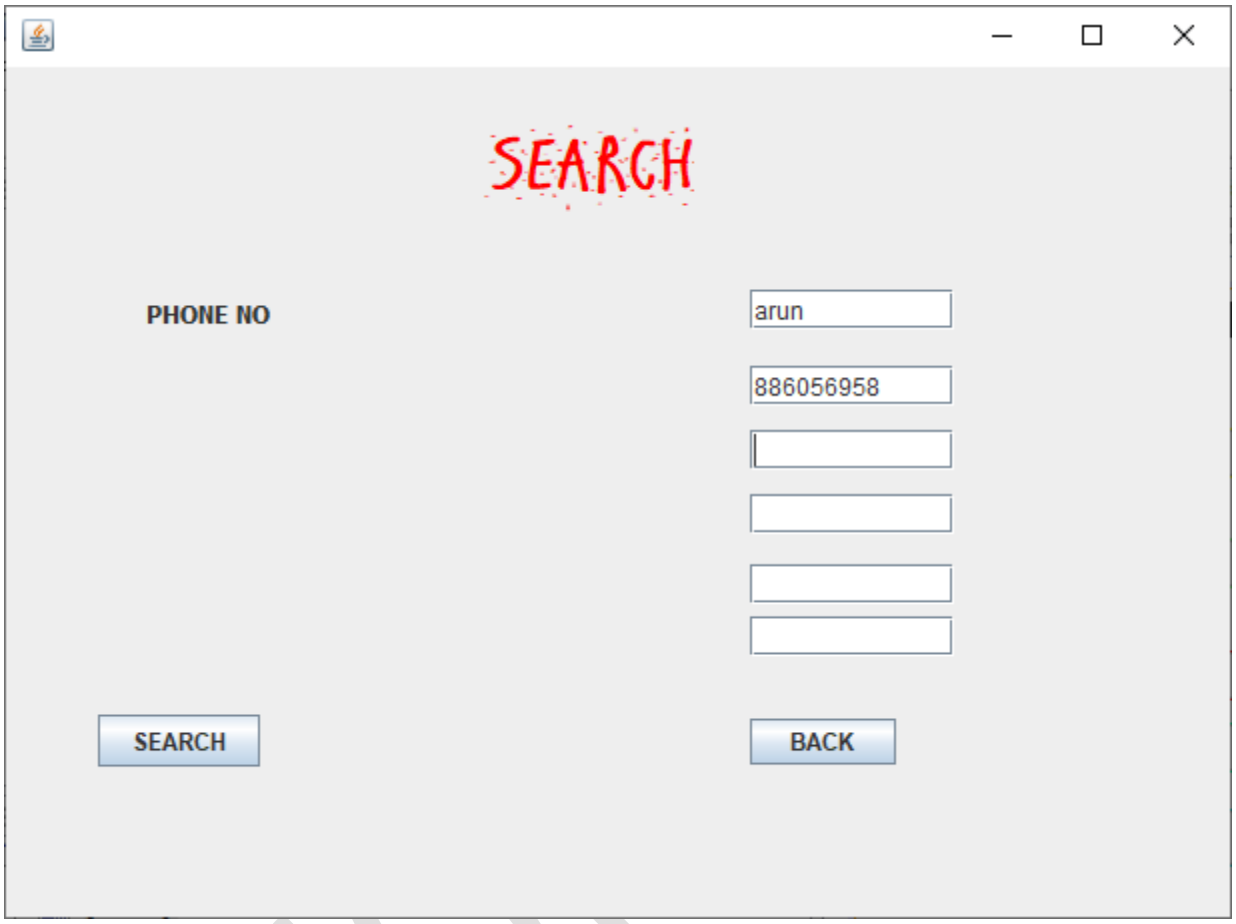




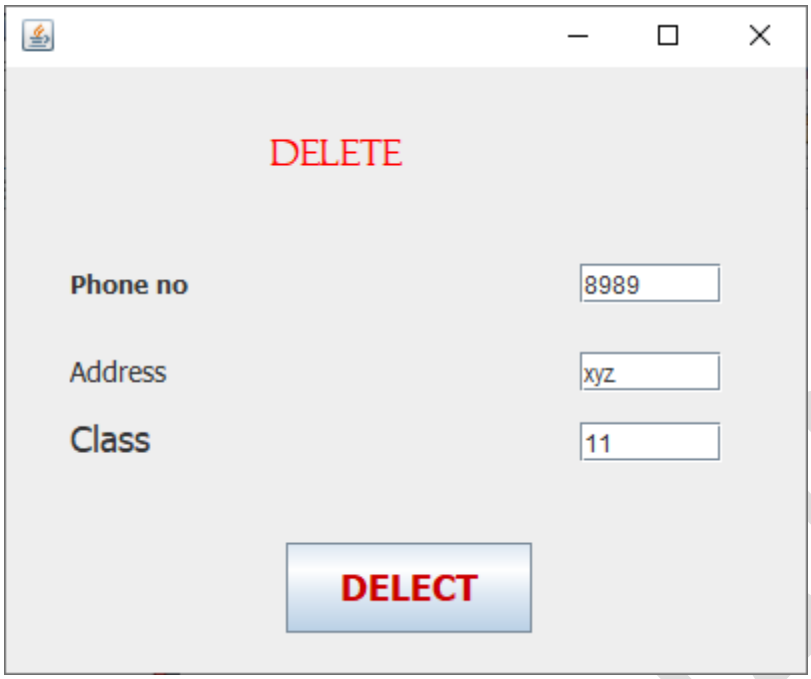
3 main menu

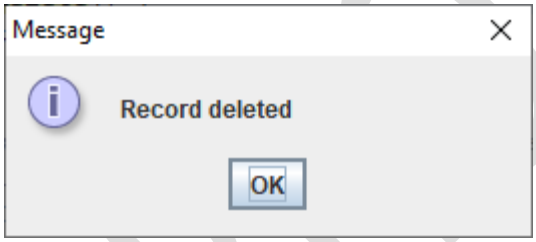


4 search

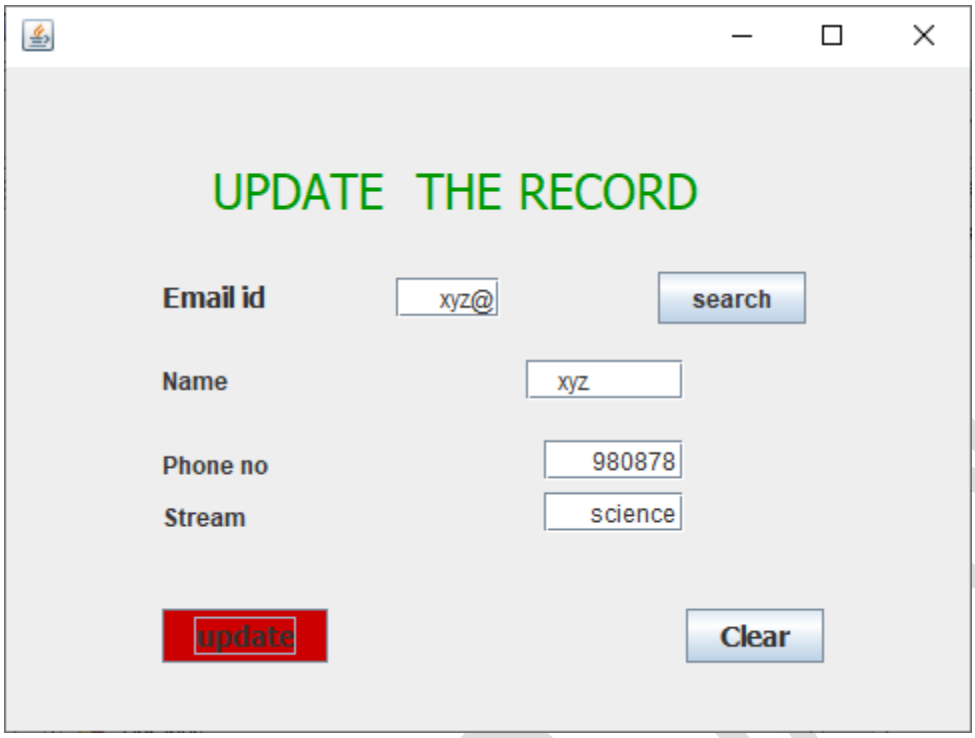


5 delete

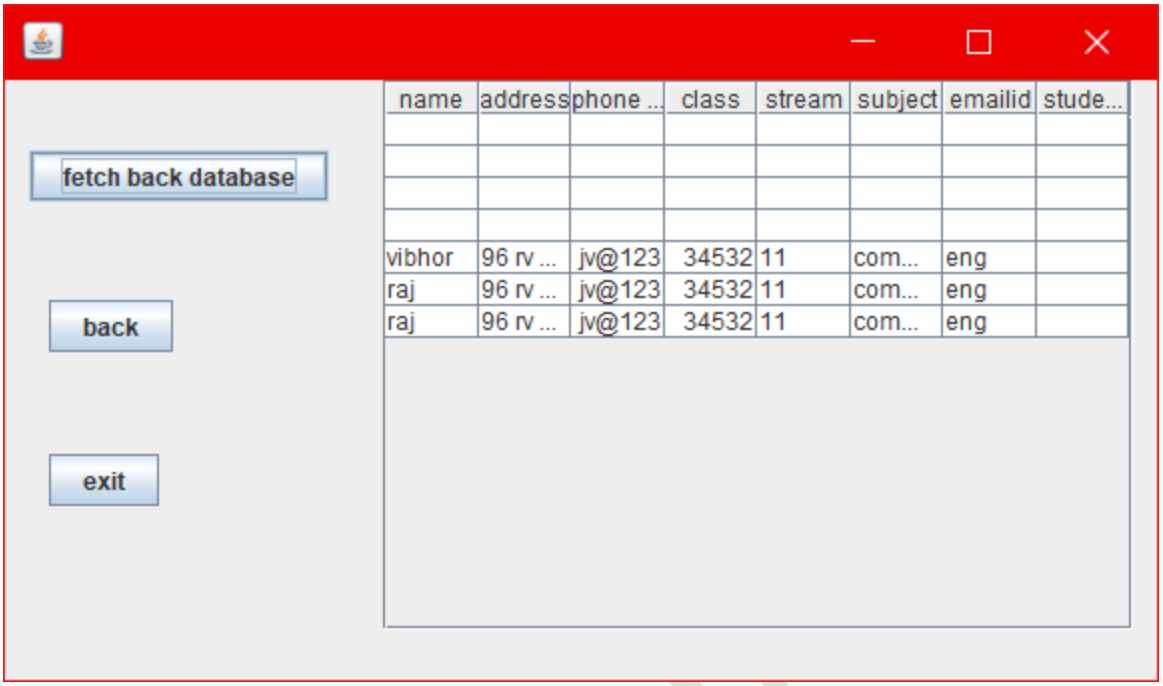




6 update



7 view



Testing Methodology (Types)

sotware Testing Methodology is defined as strategies and testing types used to certify that the Application Under Test meets client expectations. Test Methodologies include functional and non-functional testing to validate the AUT. Examples of Testing Methodologies are Unit Testing, Integration Testing, System Testing, Performance Testing etc. Each testing methodology has a defined test objective, test strategy, and deliverables.

Note: Since Software Testing is an integral part of any Development Methodology, many companies use the term Development Methodologies & Testing Methodologies colloquially. Hence Testing Methodologies could also refer to Waterfall, Agile and other QA models as against the above definition of Testing Methodologies. Discussion on various testing types does not add value to the readers. Hence, we will discuss the different development models.

Testing Methodology applied (Why)

Software **Testing Methodology** is defined as strategies and **testing** types used to certify that the Application Under **Test** meets client expectations. **Test Methodologies** include functional and non-functional **testing** to validate the AUT. ... Each **testing methodology** has a defined **test** objective, **test** strategy, and deliverables.

Functional vs. Non-functional Testing

The goal of utilizing numerous testing methodologies in your development process is to make sure your software can successfully operate in multiple environments and across different platforms. These can typically be broken down between functional and non-functional testing. Functional testing involves testing the application against the business requirements. It incorporates all test types designed to guarantee each part of a piece of software behaves as expected by using uses cases provided by the design team or business analyst. These testing methods are usually conducted in order and include:

- Unit testing
- Integration testing
- System testing
- Acceptance testing

Non-functional testing methods incorporate all test types focused on the operational aspects of a piece of software. These include:

- Performance testing
- Security testing
- Usability testing
- Compatibility testing

The key to releasing high quality software that can be easily adopted by your end users is to build a robust testing framework that implements both functional and non-functional software testing methodologies.

Unit Testing

Unit testing is the first level of testing and is often performed by the developers themselves. It is the process of ensuring individual components of a piece of software at the code level are functional and work as they were designed to. Developers in a test-driven environment will typically write and run the tests prior to the software or feature being passed over to the test team. Unit testing can be conducted manually, but automating the process will speed up delivery cycles and expand test coverage. Unit testing will also make debugging easier because finding issues earlier means they take less time to fix than if they were discovered later in the testing process. TestLeft is a tool that allows advanced testers and developers to shift left with the fastest test automation tool embedded in any IDE.

Start Shifting Left and Automate now with **TestLeft**

Integration Testing

After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities. These are then tested as group through integration testing to ensure whole segments of an application behave as expected (i.e, the interactions between units are seamless). These tests are often framed by user scenarios, such as logging into an application or opening files. Integrated tests can be conducted by either developers or independent testers and are usually comprised of a combination of automated functional and manual tests.

System Testing

System testing is a black box testing method used to evaluate the completed and integrated system, as a whole, to ensure it meets specified requirements. The functionality of the software is tested from end-to-end and is typically conducted by a separate testing team than the development team before the product is pushed into production.

Acceptance Testing

Acceptance testing is the last phase of functional testing and is used to assess whether or not the final piece of software is ready for delivery. It involves ensuring that the product is in compliance with all of the original business criteria and that it meets the end user's needs. This requires the product be tested both internally and externally, meaning you'll need to get it into the hands of your end users for beta testing along with those of your QA team. Beta testing is key to getting real feedback from potential customers and can address any final usability concerns.

Performance Testing

Performance testing is a non-functional testing technique used to determine how an application will behave under various conditions. The goal is to test its responsiveness and stability in real user situations. Performance testing can be broken down into four types:

- **Load testing** is the process of putting increasing amounts of simulated demand on your software, application, or website to verify whether or not it can handle what it's designed to handle.
- **Stress testing** takes this a step further and is used to gauge how your software will respond at or beyond its peak load. The goal of stress testing is to overload the application on purpose until it breaks by applying both realistic and unrealistic load scenarios. With stress testing, you'll be able to find the failure point of your piece of software.
- **Endurance testing,** also known as soak testing, is used to analyze the behavior of an application under a specific amount of simulated load over longer amounts of time. The goal is to understand how your system will behave under sustained use, making it a longer process than load or stress testing (which are designed to end after a few hours). A critical piece of endurance testing is that it helps uncover memory leaks.
- **Spike testing** is a type of load test used to determine how your software will respond to substantially larger bursts of concurrent user or system activity over varying amounts of time. Ideally, this will help you understand what will happen when the load is suddenly and drastically increased.

Security Testing

With the rise of cloud-based testing platforms and cyber attacks, there is a growing concern and need for the security of data being used and stored in software. Security testing is a non-functional software testing technique used to determine if the information and data in a system is protected. The goal is to purposefully find loopholes and security risks in the system that could result in unauthorized access to or the loss of information by probing the application for weaknesses. There are multiple types of this testing method, each of which aimed at verifying six basic principles of security:

1. Integrity
2. Confidentiality
3. Authentication
4. Authorization
5. Availability
6. Non-repudiation

Usability Testing

Usability testing is a testing method that measures an application's ease-of-use from the end-user perspective and is often performed during the system or acceptance testing stages. The goal is to determine whether or not the visible design and aesthetics of an application meet the intended workflow for various processes, such as logging into an application. Usability testing is a great way for teams to review separate functions, or the system as a whole, is intuitive to use.

Compatibility Testing

Compatibility testing is used to gauge how an application or piece of software will work in different environments. It is used to check that your product is compatible with multiple operating systems, platforms, browsers, or resolution configurations. The goal is to ensure that your software's functionality is consistently supported across any environment you expect your end users to be using.

Testing With TestComplete

TestComplete is our robust automated GUI testing tool that excels in compatibility and integration testing. It helps QA teams create and run tests across desktop, mobile, and web applications – enabling testing professionals to speed up delivery cycles and improve software quality. Testcomplete comes with built-in support for various test environments, integrations to performance testing tools, as well as support for developer friendly SCMs, allowing you to seamlessness integrate it into your development process. Using TestComplete will enable you to build a robust testing framework that utilizes the broad spectrum of available software testing methodologies.

Test Cases

A **TEST CASE** is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly.

The process of developing test cases can also help find problems in the requirements or design of an application.

Test Data

Test data is data which has been specifically identified for use in tests, typically of a computer program.

Some data may be used in a confirmatory way, typically to verify that a given set of input to a given function produces some expected result. Other data may be used in order to challenge the ability of the program to respond to unusual, extreme, exceptional, or unexpected input.

Test data may be produced in a focused or systematic way (as is typically the case in domain testing), or by using other, less-focused approaches (as is typically the case in high-volume randomized automated tests). Test data may be produced by the tester, or by a program or function that aids the tester. Test data may be recorded for re-use, or used once and then forgotten.

Gap Analysis (Planned Vs Achieved)

The need for new products or additions to existing lines may emerge from portfolio analysis, in particular from the use of the Boston Consulting Group Growth-share matrix—or the need may emerge from the regular process of following trends in the requirements of consumers. At some point, a gap emerges between what existing products offer and what the consumer demands. The organization must fill that gap to survive and grow.

Gap analysis can identify gaps in the market. Thus, comparing forecast profits to desired profits reveals the *planning gap*. This represents a goal for new activities in general, and new products in particular. The planning gap can be divided into three main elements: usage gap, existing gap, and product gap

Rework/ Retest

Retesting is testing of a particular **bug** after it has been fixed. Usually tester raises the bug when they find it while testing the product or its component. This bug is assigned to a developer and he fixes it. Post fixing the bug is assigned to the tester for its **verification**. This testing is known as retesting.

Conclusion

This part incorporates the Conclusion came to in the wake of making the present rendition of the product to meet the framework targets. The correlation is done between the framework that was fabricated and unique prerequisites that were structured toward the start of the venture. It additionally depicts the Future Work that is proposed to be practiced with later forms of the product. As expressed in the presentation the base objective toward the start of this venture was to show the center functionalists in an easy to understand GUI interface. This segment delineate which issues happened during the task. It will portray where the arranging was practical and will likewise give suggestion for comparable or additionally extends. The primary

target of the application is to help software engineering understudies comprehend the nuts and bolts of Java, mysql. By perusing through the application and taking a gander at the code for each graphical translation, understudies ought to have the option to effectively get the implementation. The accompanying outcomes have been accomplished after the finishing the framework andrelate back to the framework's objective.1. Ought to permit software engineering understudies to peruse through the code and application: This is accomplished when clients, i.e., software engineering Ought to permit clients to peruse through various item classifications: This is achieved when the client first runs the application and is coordinated to a landing page that has categories available for all the diverse thing types that can be bought with this web based shopping-cart application. The client can peruse and tap on any classification to see the things recorded for that specific category.3. Ought to permit clients to spare things to the t and furthermore to see point by point data about a specific thing: The clients can add any number of things to the shopping basket from any of 67t he recorded classes by just tapping the coaching symbol at the right-hand corner of every item. students

can see a nitty gritty portrayal of the thing and cost by tapping on the more symbol next o the truck icon.4. Ought to permit clients to look at the things: This is accomplished when clients click the checkout button in the shopping basket. The checkout button vanishes when there are no things in the coaching management system. This suggests students can possibly tap the checkout button when there are items in the coaching management system

Limitation of System

The core functionality was reached in the following parts. It is possible to insert a new record of customer, change records and delete records. Further development of the project is possible. It will be interesting to implement more functions in the project.

FUTURE SCOPE OF THE PROJECT

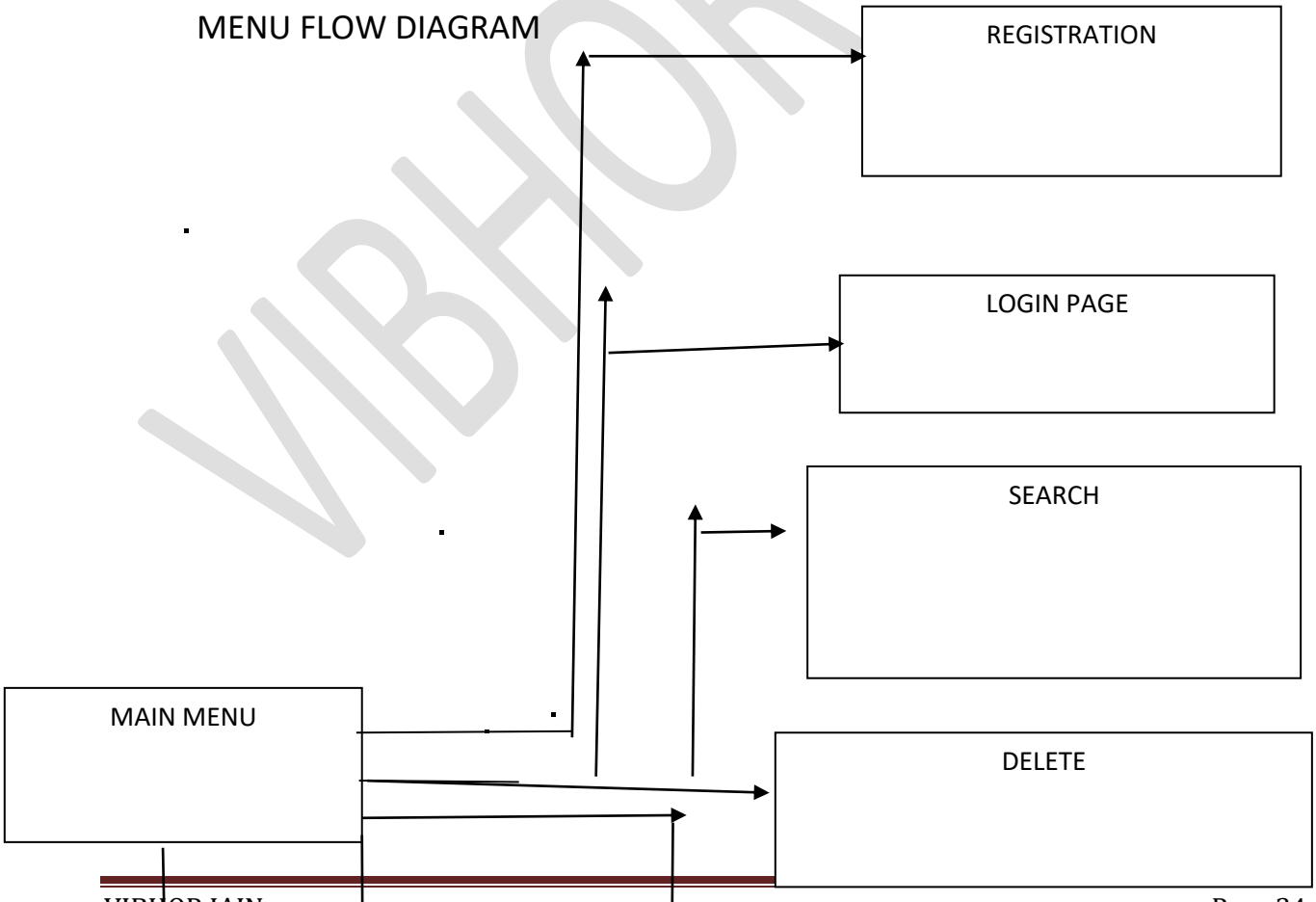
This project can be easily apply under different situations. We can attach new futures as we can add new attribute as and when we demand. Long-lasting is possible as and when need in the project. There is flexibility all the modules

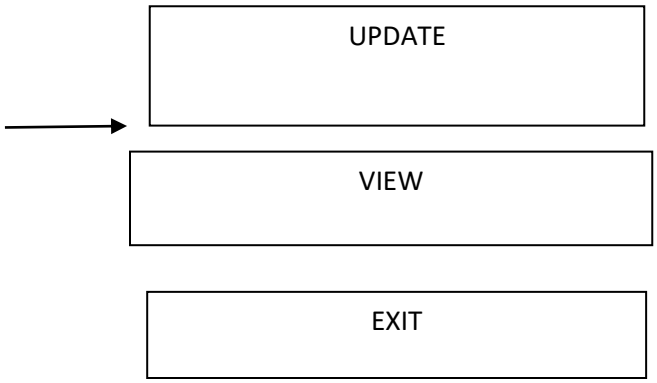
References/ Bibliography

https://www.wikipedia.org/

tutorialspoint

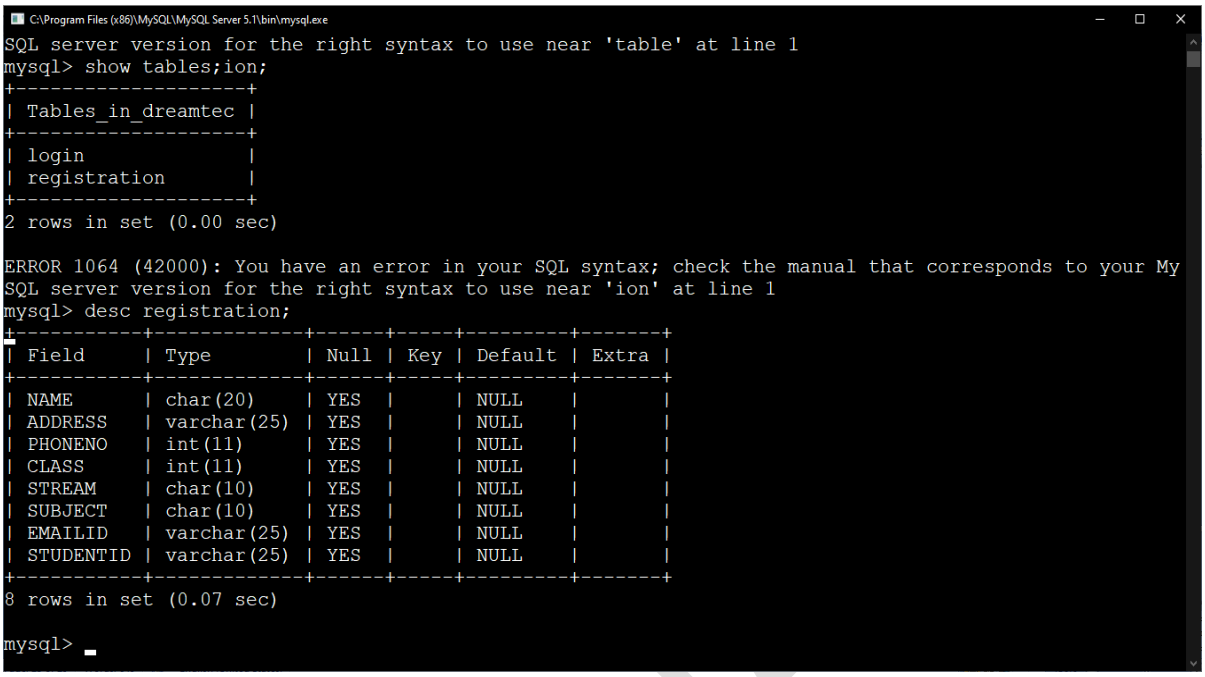
MENU FLOW DIAGRAM



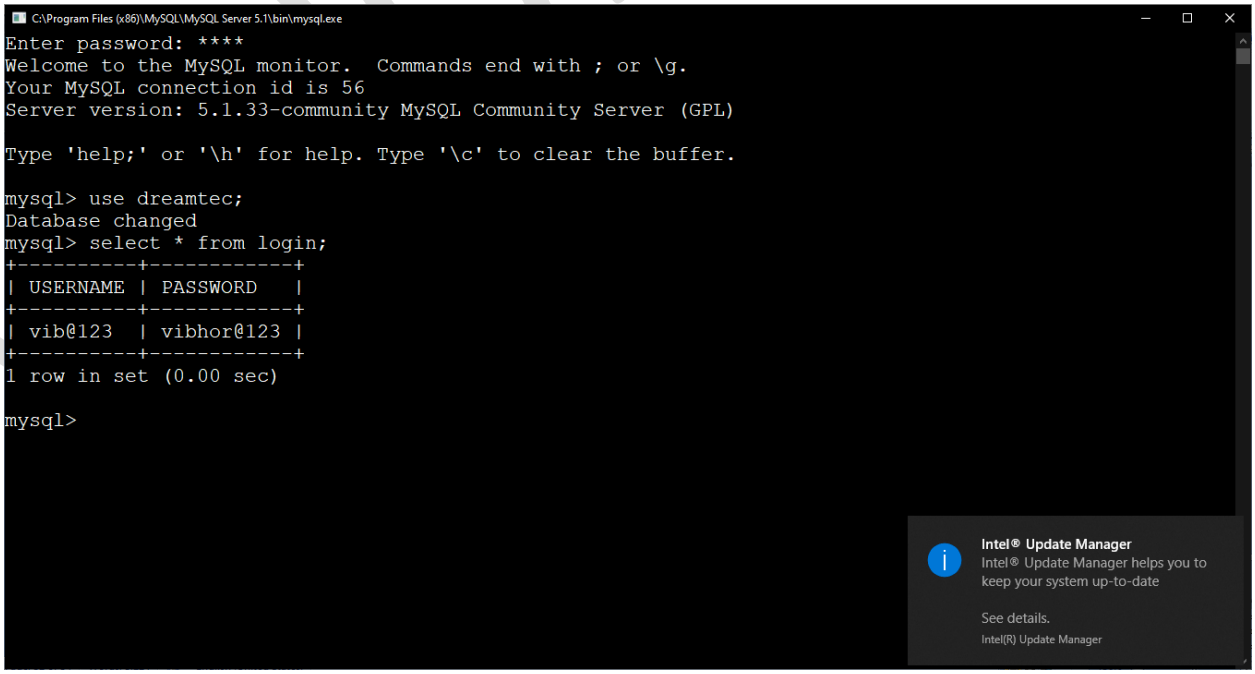


SHOW TABLES

```
C:\Program Files (x86)\MySQL\MySQL Server 5.1\bin\mysql.exe
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 54
Server version: 5.1.33-community MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show tables;
ERROR 1046 (3D000): No database selected
mysql> use dreamtec;
Database changed
mysql> show tables;
+--------------------+
| Tables_in_dreamtec |
+--------------------+
| login              |
| registration       |
+--------------------+
2 rows in set (0.00 sec)

mysql> show tables;
```

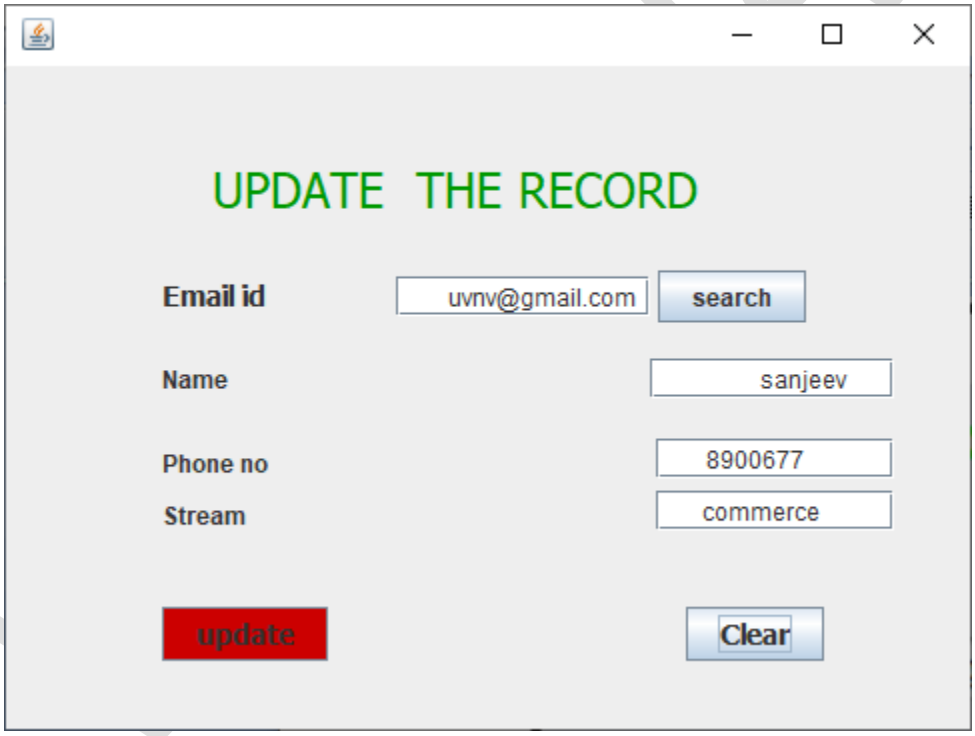

C:\Program Files (x86)\MySQL\MySQL Server 5.1\bin\mysql.exe
SQL server version for the right syntax to use near 'table' at line 1
mysql> show tables;ion;
+--------------------+
| Tables_in_dreamtec |
+--------------------+
| login |
| registration |
+--------------------+
2 rows in set (0.00 sec)
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your My
SQL server version for the right syntax to use near 'ion' at line 1
mysql> desc registration;
+-----------+-------------+------+-----+---------+-------+
| Field | Type | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
NAME	char(20)	YES		NULL	
ADDRESS	varchar(25)	YES		NULL	
PHONENO	int(11)	YES		NULL	
CLASS	int(11)	YES		NULL	
STREAM	char(10)	YES		NULL	
SUBJECT	char(10)	YES		NULL	
EMAILID	varchar(25)	YES		NULL	
STUDENTID	varchar(25)	YES		NULL	
+-----------+-------------+------+-----+---------+-------+
8 rows in set (0.07 sec)
mysql>

LOGIN TABLE


C:\Program Files (x86)\MySQL\MySQL Server 5.1\bin\mysql.exe
Enter password: ****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 56
Server version: 5.1.33-community MySQL Community Server (GPL)
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql> use dreamtec;
Database changed
mysql> select * from login;
+----------+------------+
| USERNAME | PASSWORD |
+----------+------------+
| vib@123 | vibhor@123 |
+----------+------------+
1 row in set (0.00 sec)
mysql>
Intel® Update Manager
Intel® Update Manager helps you to keep your system up-to-date
See details.
Intel(R) Update Manager

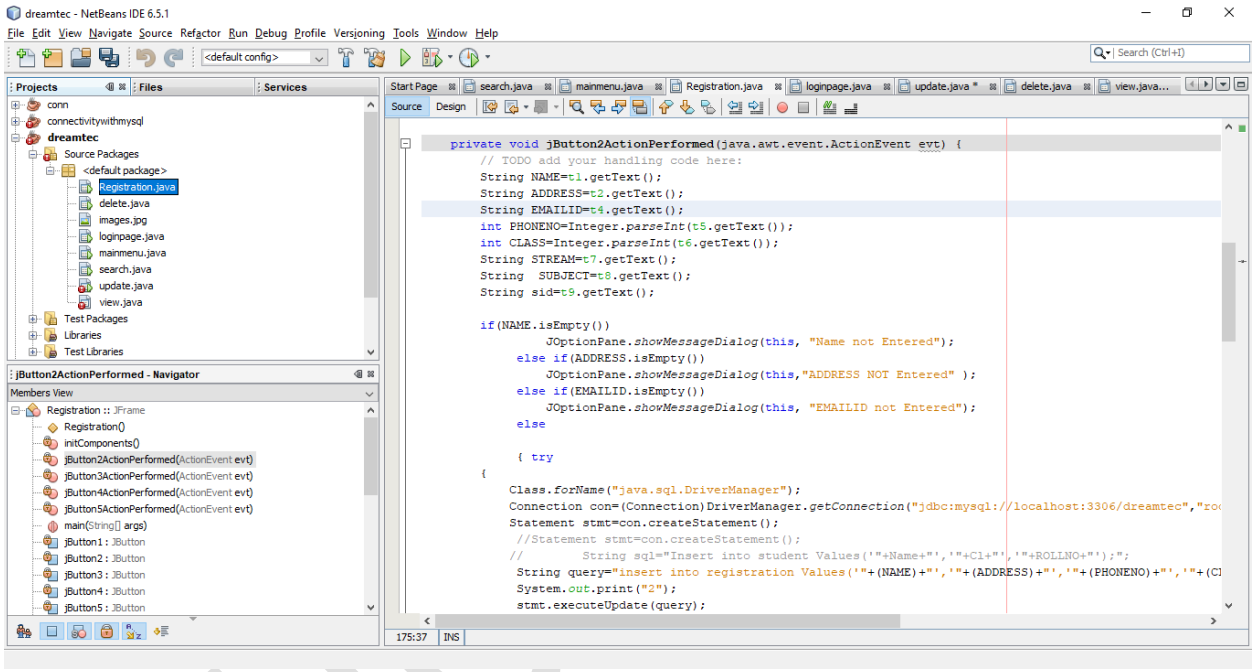
Sample Output

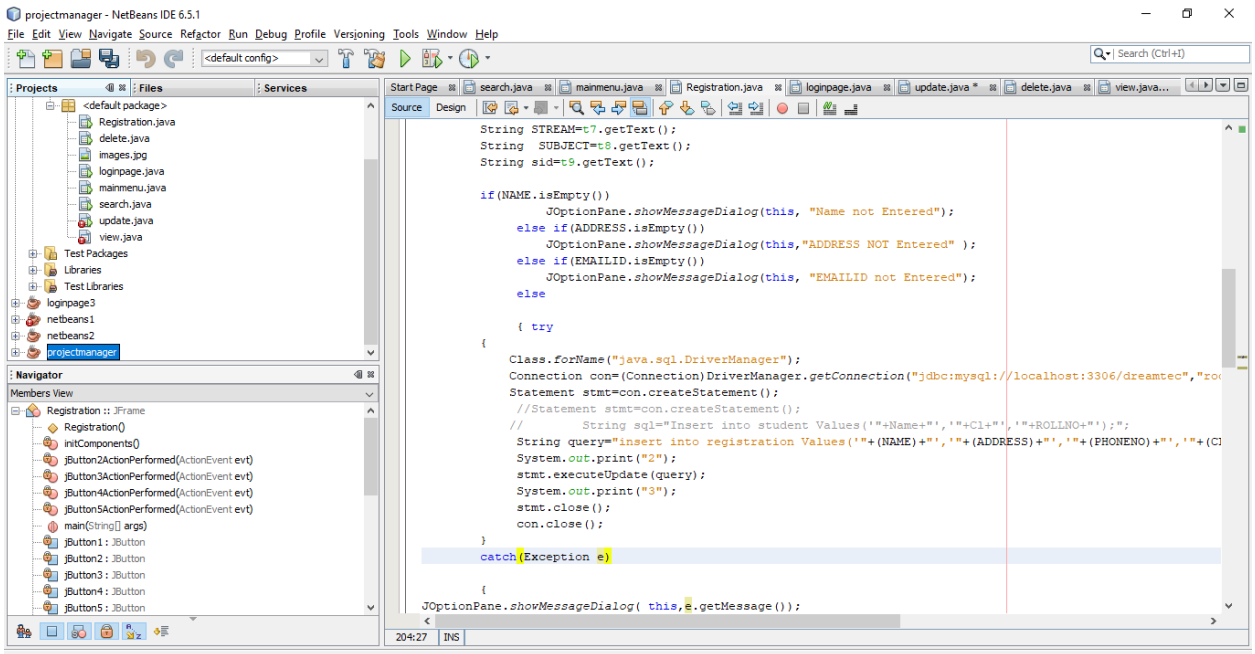
UPDATE

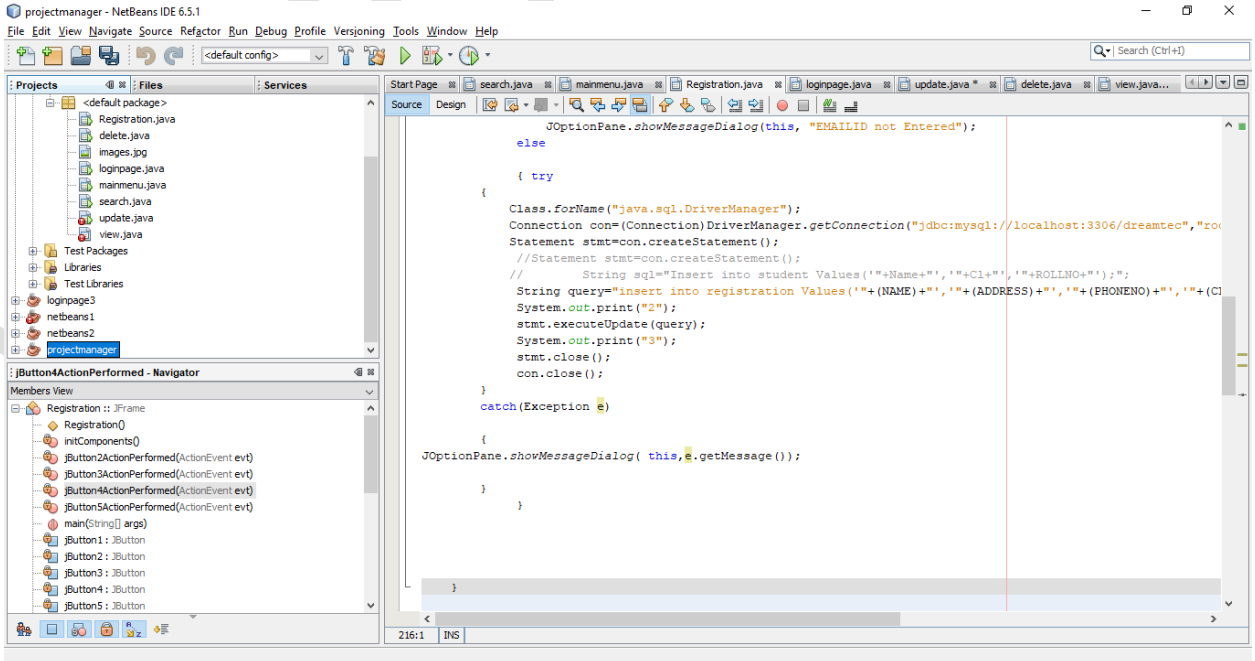


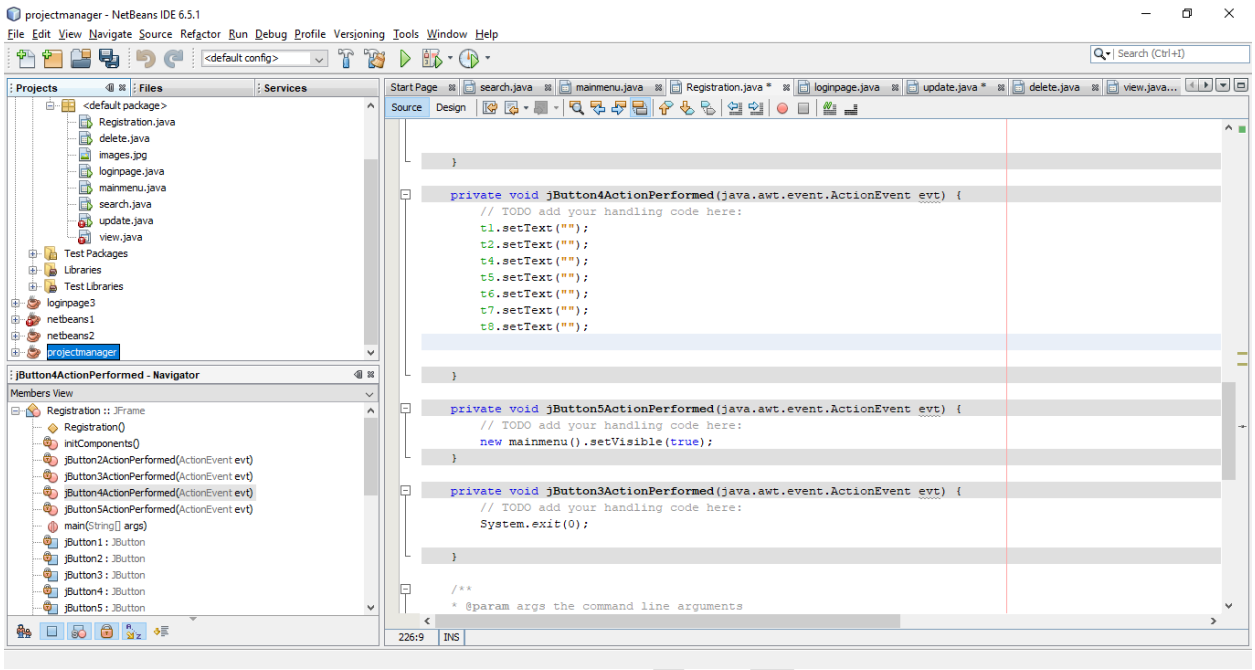
Program Code

REGISTRATION

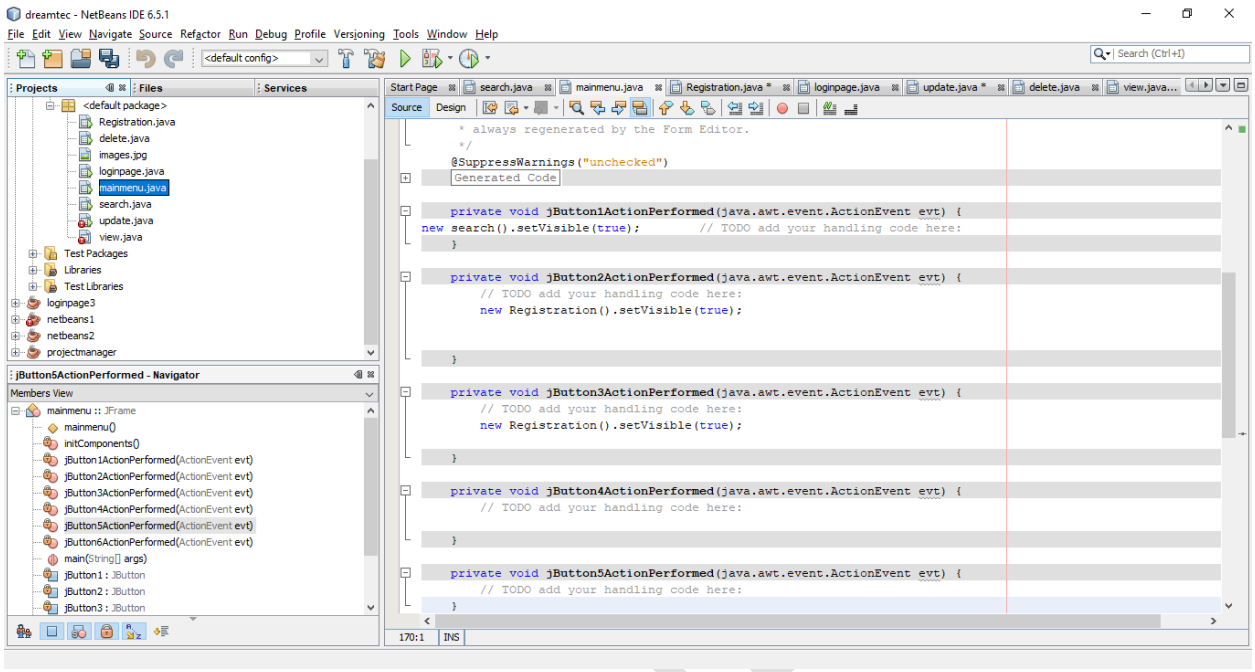



projectmanager - NetBeans IDE 6.5.1
File Edit View Navigate Source Refactor Run Debug Profile Versioning Tools Window Help
Projects
Files
Services
Navigator
Start Page
search.java
mainmenu.java
Registration.java
loginpage.java
update.java *
delete.java
view.java...
Source
Design
String STREAM=t7.getText();
String SUBJECT=t8.getText();
String sid=t9.getText();
if(NAME.isEmpty())
JOptionPane.showMessageDialog(this, "Name not Entered");
else if(ADDRESS.isEmpty())
JOptionPane.showMessageDialog(this,"ADDRESS NOT Entered");
else if(EMAILID.isEmpty())
JOptionPane.showMessageDialog(this, "EMAILID not Entered");
else
{ try
{
Class.forName("java.sql.DriverManager");
Connection con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/dreamtec","ro
Statement stmt=con.createStatement();
//Statement stmt=con.createStatement();
// String sql="Insert into student Values('"+Name+"','"+Cl+"','"+ROLLNO+"');";
String query="insert into registration Values('"+(NAME)+"','"+(ADDRESS)+"','"+(PHONENO)+"','"+(Cl
System.out.print("2");
stmt.executeUpdate(query);
System.out.print("3");
stmt.close();
con.close();
}
catch(Exception e)
{
JOptionPane.showMessageDialog(this,e.getMessage());
204:27
INS

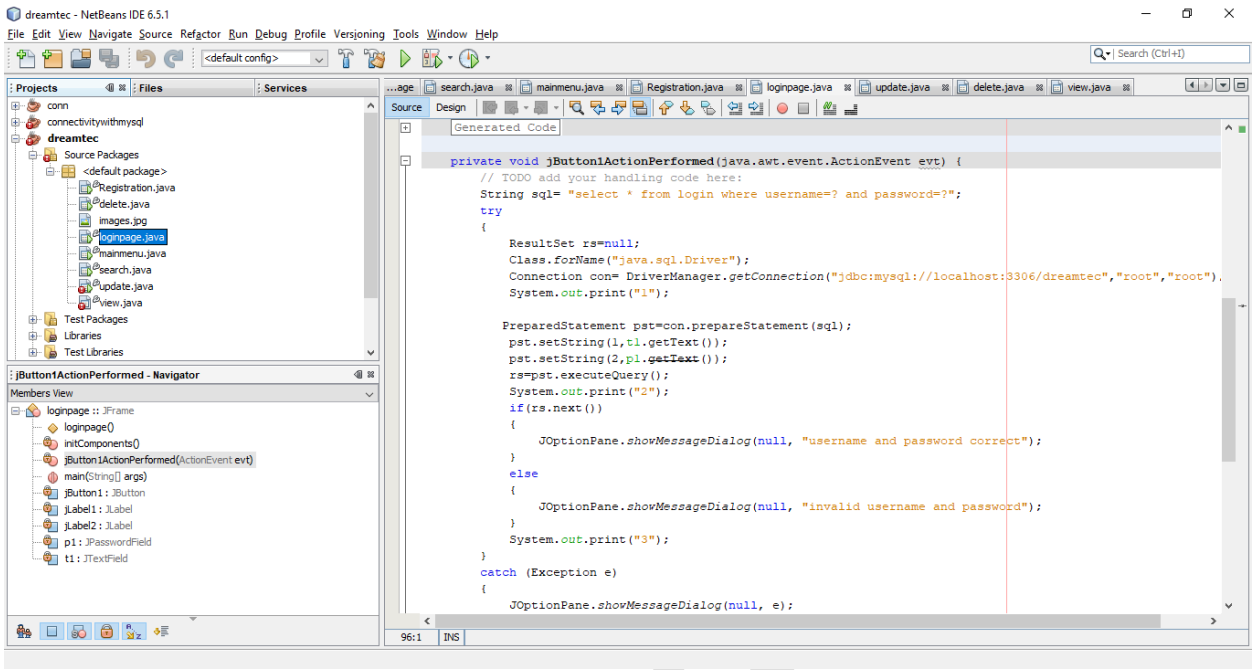

projectmanager - NetBeans IDE 6.5.1
File Edit View Navigate Source Refactor Run Debug Profile Versioning Tools Window Help
Projects
Files
Services
jButton4ActionPerformed - Navigator
Start Page
search.java
mainmenu.java
Registration.java
loginpage.java
update.java *
delete.java
view.java...
Source
Design
JOptionPane.showMessageDialog(this, "EMAILID not Entered");
else
{ try
{
Class.forName("java.sql.DriverManager");
Connection con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/dreamtec","ro
Statement stmt=con.createStatement();
//Statement stmt=con.createStatement();
// String sql="Insert into student Values('"+Name+"','"+Cl+"','"+ROLLNO+"');";
String query="insert into registration Values('"+(NAME)+"','"+(ADDRESS)+"','"+(PHONENO)+"','"+(Cl
System.out.print("2");
stmt.executeUpdate(query);
System.out.print("3");
stmt.close();
con.close();
}
catch(Exception e)
{
JOptionPane.showMessageDialog(this,e.getMessage());
}
}
}
216:1
INS

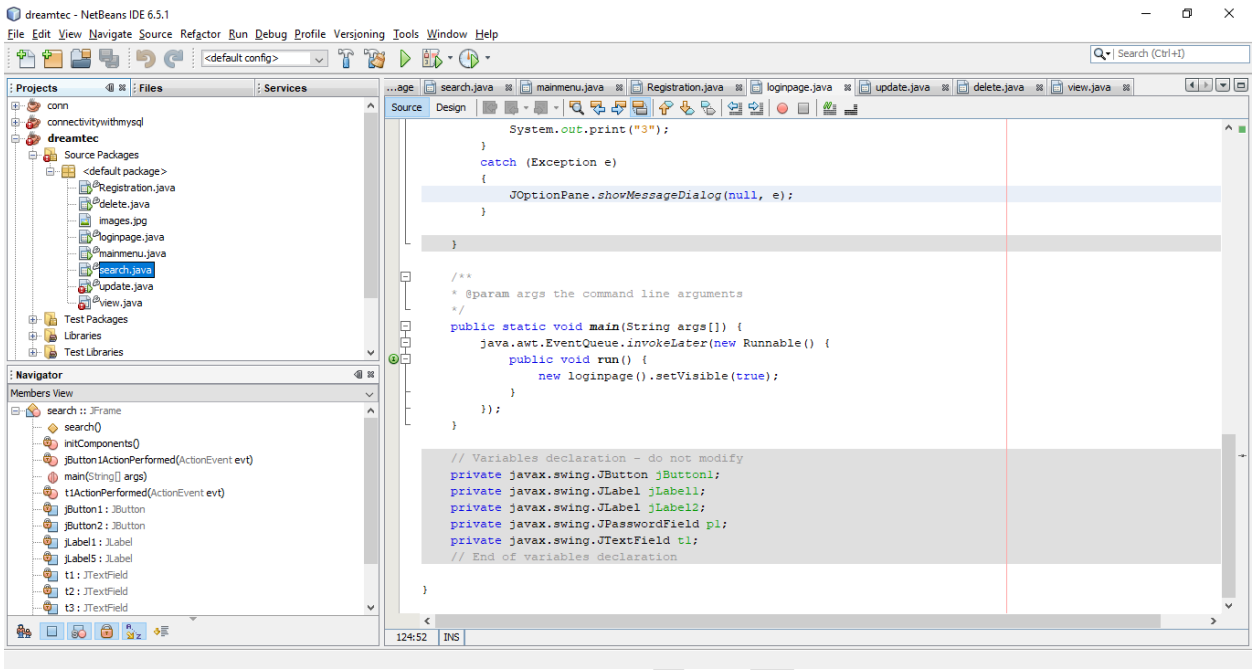


2 MAIN MENU

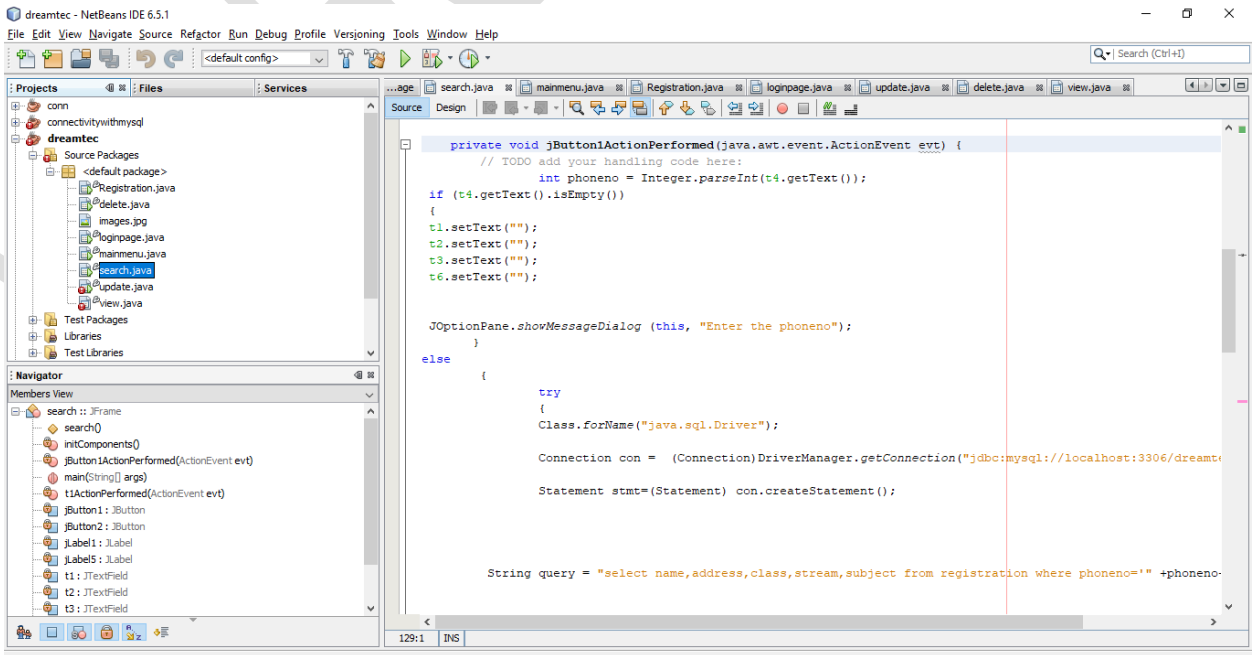


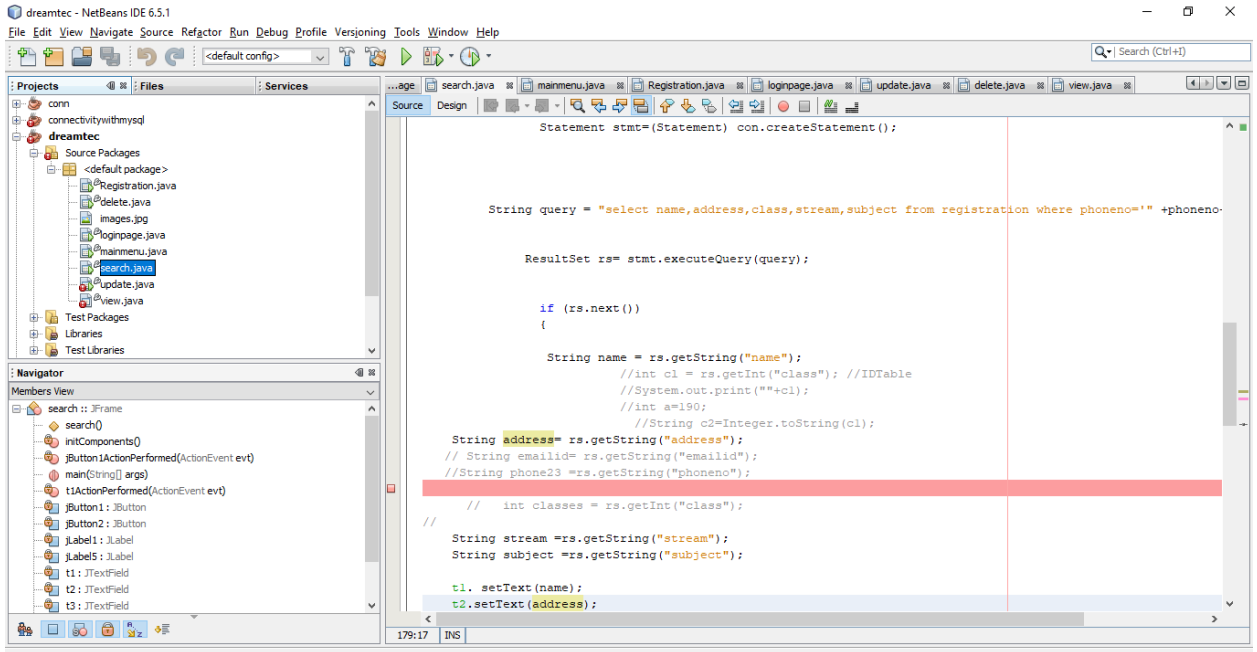
3 LOGIN PAGE

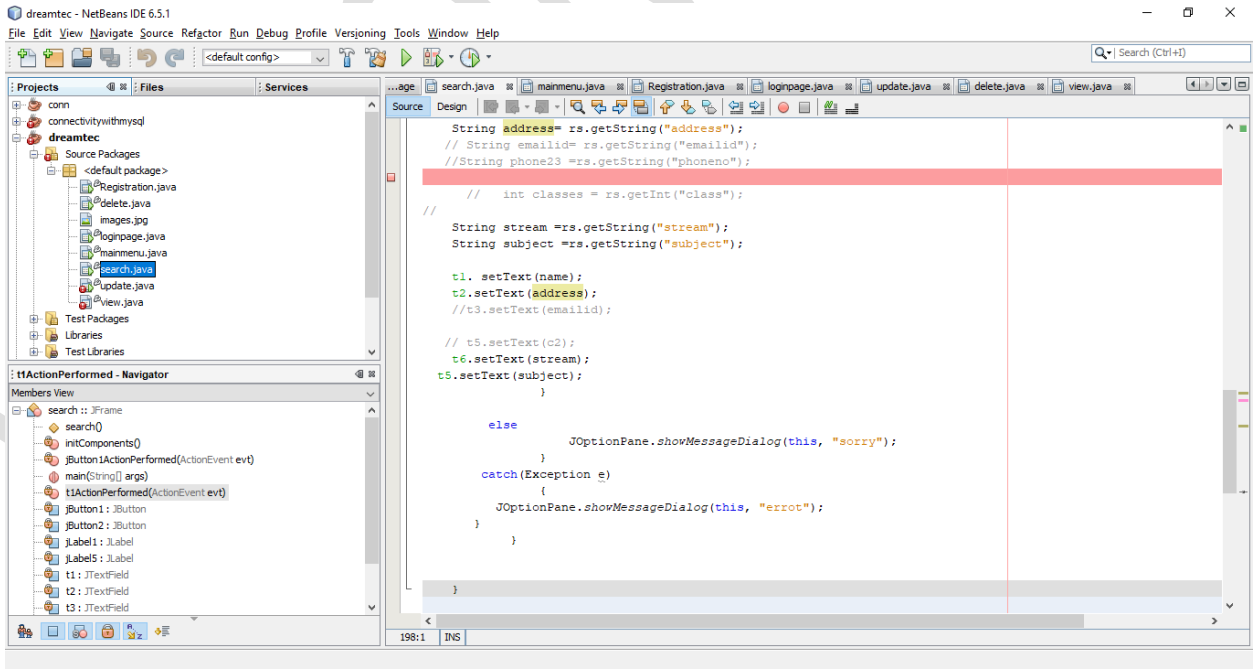

dreamtec - NetBeans IDE 6.5.1
File Edit View Navigate Source Refactor Run Debug Profile Versioning Tools Window Help
Projects
Files
Services
conn
connectivitywithmysql
dreamtec
Source Packages
<default package>
Registration.java
delete.java
images.jpg
loginpage.java
mainmenu.java
search.java
update.java
view.java
Test Packages
Libraries
Test Libraries
jButton1ActionPerformed - Navigator
Members View
loginpage :: JFrame
loginpage()
initComponents()
jButton1ActionPerformed(ActionEvent evt)
main(String[] args)
jButton1 : JButton
jLabel1 : JLabel
jLabel2 : JLabel
p1 : JPasswordField
t1 : JTextField
search.java
mainmenu.java
Registration.java
loginpage.java
update.java
delete.java
view.java
Source
Design
Generated Code
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
String sql= "select * from login where username=? and password=?";
try
{
ResultSet rs=null;
Class.forName("java.sql.Driver");
Connection con= DriverManager.getConnection("jdbc:mysql://localhost:3306/dreamtec","root","root");
System.out.print("1");
PreparedStatement pst=con.prepareStatement(sql);
pst.setString(1,t1.getText());
pst.setString(2,p1.getText());
rs=pst.executeQuery();
System.out.print("2");
if(rs.next())
{
JOptionPane.showMessageDialog(null, "username and password correct");
}
else
{
JOptionPane.showMessageDialog(null, "invalid username and password");
}
System.out.print("3");
}
catch (Exception e)
{
JOptionPane.showMessageDialog(null, e);
96:1
INS


dreamtec - NetBeans IDE 6.5.1
File Edit View Navigate Source Refactor Run Debug Profile Versioning Tools Window Help
Projects
Files
Services
conn
connectivitywithmysql
dreamtec
Source Packages
<default package>
Registration.java
delete.java
images.jpg
loginpage.java
mainmenu.java
search.java
update.java
view.java
Test Packages
Libraries
Test Libraries
Navigator
Members View
search :: JFrame
search()
initComponents()
jButton1ActionPerformed(ActionEvent evt)
main(String[] args)
t1ActionPerformed(ActionEvent evt)
jButton1 : JButton
jButton2 : JButton
jLabel1 : JLabel
jLabel5 : JLabel
t1 : JTextField
t2 : JTextField
t3 : JTextField
Source
Design
System.out.print("3");
}
catch (Exception e)
{
JOptionPane.showMessageDialog(null, e);
}
}
/**
* @param args the command line arguments
*/
public static void main(String args[]) {
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
new loginpage().setVisible(true);
}
});
}
// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JPasswordField p1;
private javax.swing.JTextField t1;
// End of variables declaration
}
124:52
INS

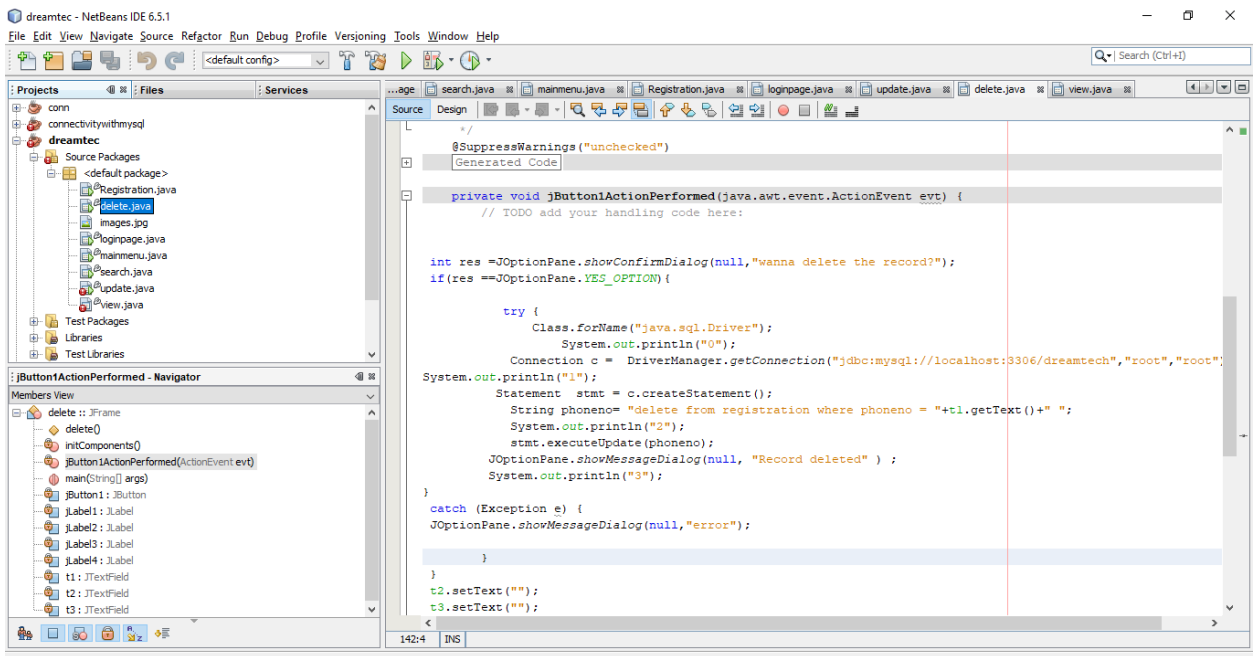
4 SEARCH

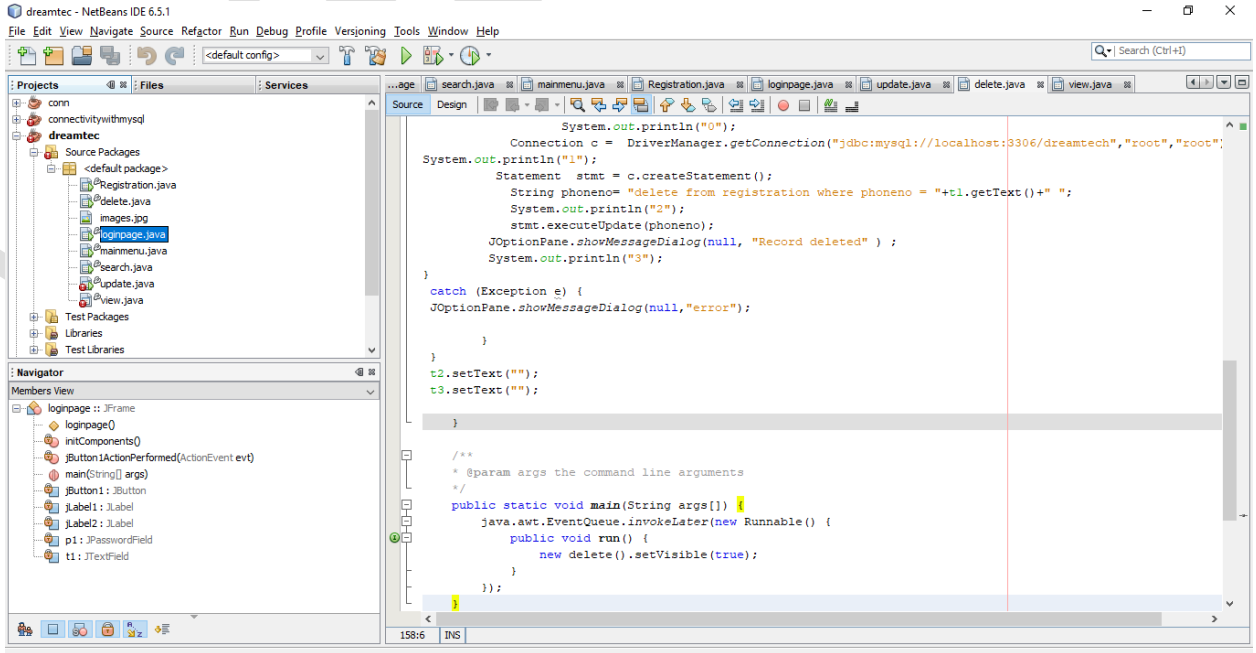

dreamtec - NetBeans IDE 6.5.1
File Edit View Navigate Source Refactor Run Debug Profile Versioning Tools Window Help
Projects
Files
Services
conn
connectivitywithmysql
dreamtec
Source Packages
<default package>
Registration.java
delete.java
images.jpg
loginpage.java
mainmenu.java
search.java
update.java
view.java
Test Packages
Libraries
Test Libraries
Navigator
Members View
search :: JFrame
search()
initComponents()
jButton1ActionPerformed(ActionEvent evt)
main(String[] args)
t1ActionPerformed(ActionEvent evt)
jButton1 : JButton
jButton2 : JButton
jLabel1 : JLabel
jLabel5 : JLabel
t1 : JTextField
t2 : JTextField
t3 : JTextField
Source
Design
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
int phoneno = Integer.parseInt(t4.getText());
if (t4.getText().isEmpty())
{
t1.setText("");
t2.setText("");
t3.setText("");
t6.setText("");
JOptionPane.showMessageDialog (this, "Enter the phoneno");
}
else
{
try
{
Class.forName("java.sql.Driver");
Connection con = (Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/dreamt
Statement stmt=(Statement) con.createStatement();
String query = "select name,address,class,stream,subject from registration where phoneno='" +phoneno
129:1
INS


dreamtec - NetBeans IDE 6.5.1
File Edit View Navigate Source Refactor Run Debug Profile Versioning Tools Window Help
Projects
Files
Services
conn
connectivitywithmysql
dreamtec
Source Packages
<default package>
Registration.java
delete.java
images.jpg
loginpage.java
mainmenu.java
search.java
update.java
view.java
Test Packages
Libraries
Test Libraries
Navigator
Members View
search :: JFrame
search()
initComponents()
jButton1ActionPerformed(ActionEvent evt)
main(String[] args)
t1ActionPerformed(ActionEvent evt)
jButton1 : JButton
jButton2 : JButton
jLabel1 : JLabel
jLabel5 : JLabel
t1 : JTextField
t2 : JTextField
t3 : JTextField
search.java
mainmenu.java
Registration.java
loginpage.java
update.java
delete.java
view.java
Source
Design
Statement stmt=(Statement) con.createStatement();
String query = "select name,address,class,stream,subject from registration where phoneno='" +phoneno
ResultSet rs= stmt.executeQuery(query);
if (rs.next())
{
String name = rs.getString("name");
//int cl = rs.getInt("class"); //IDTable
//System.out.print(""+cl);
//int a=190;
//String c2=Integer.toString(cl);
String address= rs.getString("address");
// String emailid= rs.getString("emailid");
//String phone23 =rs.getString("phoneno");
// int classes = rs.getInt("class");
//
String stream =rs.getString("stream");
String subject =rs.getString("subject");
t1. setText(name);
t2.setText(address);
179:17
INS

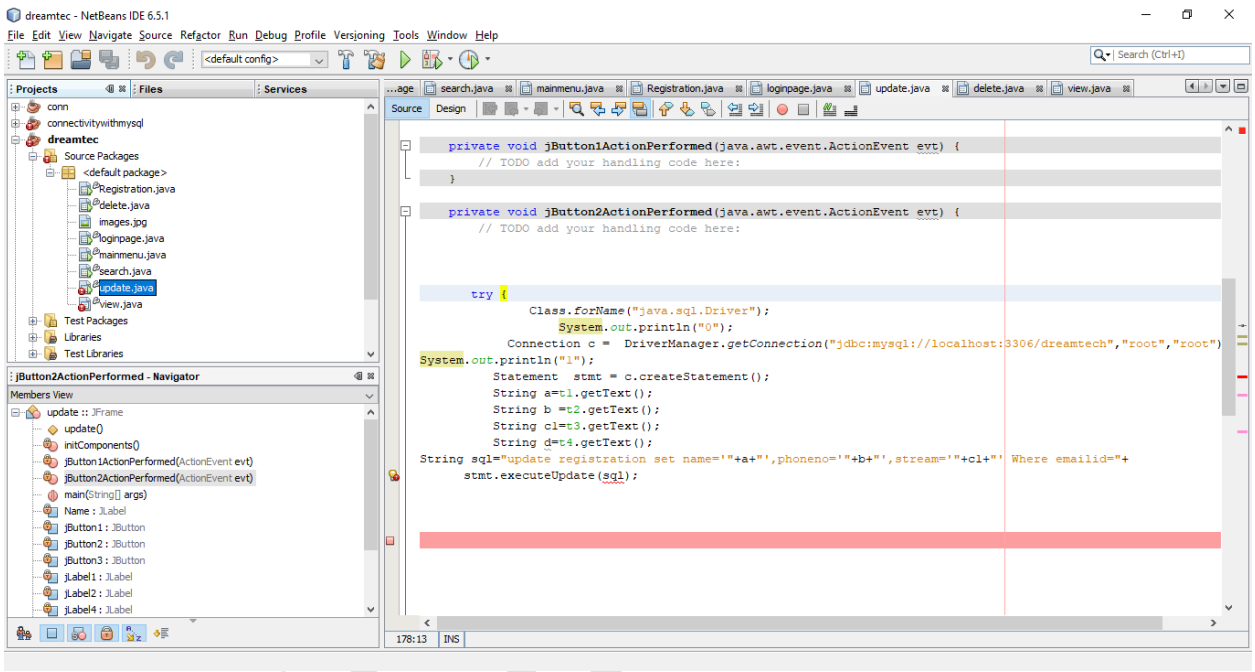

dreamtec - NetBeans IDE 6.5.1
File Edit View Navigate Source Refactor Run Debug Profile Versioning Tools Window Help
Projects
Files
Services
conn
connectivitywithmysql
dreamtec
Source Packages
<default package>
Registration.java
delete.java
images.jpg
loginpage.java
mainmenu.java
search.java
update.java
view.java
Test Packages
Libraries
Test Libraries
t1ActionPerformed - Navigator
Members View
search :: JFrame
search()
initComponents()
jButton1ActionPerformed(ActionEvent evt)
main(String[] args)
t1ActionPerformed(ActionEvent evt)
jButton1 : JButton
jButton2 : JButton
jLabel1 : JLabel
jLabel5 : JLabel
t1 : JTextField
t2 : JTextField
t3 : JTextField
search.java
mainmenu.java
Registration.java
loginpage.java
update.java
delete.java
view.java
Source
Design
String address= rs.getString("address");
// String emailid= rs.getString("emailid");
//String phone23 =rs.getString("phoneno");
// int classes = rs.getInt("class");
//
String stream =rs.getString("stream");
String subject =rs.getString("subject");
t1. setText(name);
t2.setText(address);
//t3.setText(emailid);
// t5.setText(c2);
t6.setText(stream);
t5.setText(subject);
}
else
JOptionPane.showMessageDialog(this, "sorry");
}
catch(Exception e)
{
JOptionPane.showMessageDialog(this, "errot");
}
}
198:1
INS

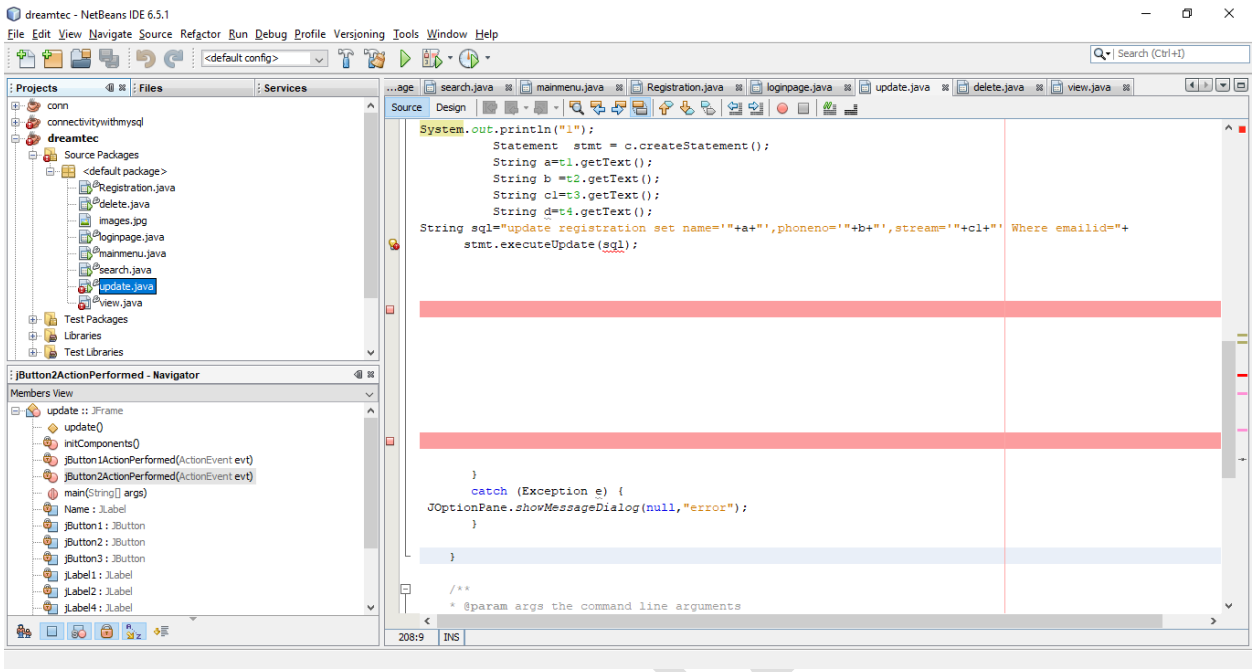
5 DELETE


dreamtec - NetBeans IDE 6.5.1
File Edit View Navigate Source Refactor Run Debug Profile Versioning Tools Window Help
Projects
Files
Services
conn
connectivitywithmysql
dreamtec
Source Packages
<default package>
Registration.java
delete.java
images.jpg
loginpage.java
mainmenu.java
search.java
update.java
view.java
Test Packages
Libraries
Test Libraries
jButton1ActionPerformed - Navigator
Members View
delete :: JFrame
delete()
initComponents()
jButton1ActionPerformed(ActionEvent evt)
main(String[] args)
jButton1 : JButton
jLabel1 : JLabel
jLabel2 : JLabel
jLabel3 : JLabel
jLabel4 : JLabel
t1 : JTextField
t2 : JTextField
t3 : JTextField
Source
Design
@SuppressWarnings("unchecked")
Generated Code
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
int res =JOptionPane.showConfirmDialog(null,"wanna delete the record?");
if(res ==JOptionPane.YES_OPTION){
try {
Class.forName("java.sql.Driver");
System.out.println("0");
Connection c = DriverManager.getConnection("jdbc:mysql://localhost:3306/dreamtech","root","root"
System.out.println("1");
Statement stmt = c.createStatement();
String phoneno= "delete from registration where phoneno = "+t1.getText()+" ";
System.out.println("2");
stmt.executeUpdate(phoneno);
JOptionPane.showMessageDialog(null, "Record deleted") ;
System.out.println("3");
}
catch (Exception e) {
JOptionPane.showMessageDialog(null,"error");
}
}
t2.setText("");
t3.setText("");
142:4
INS

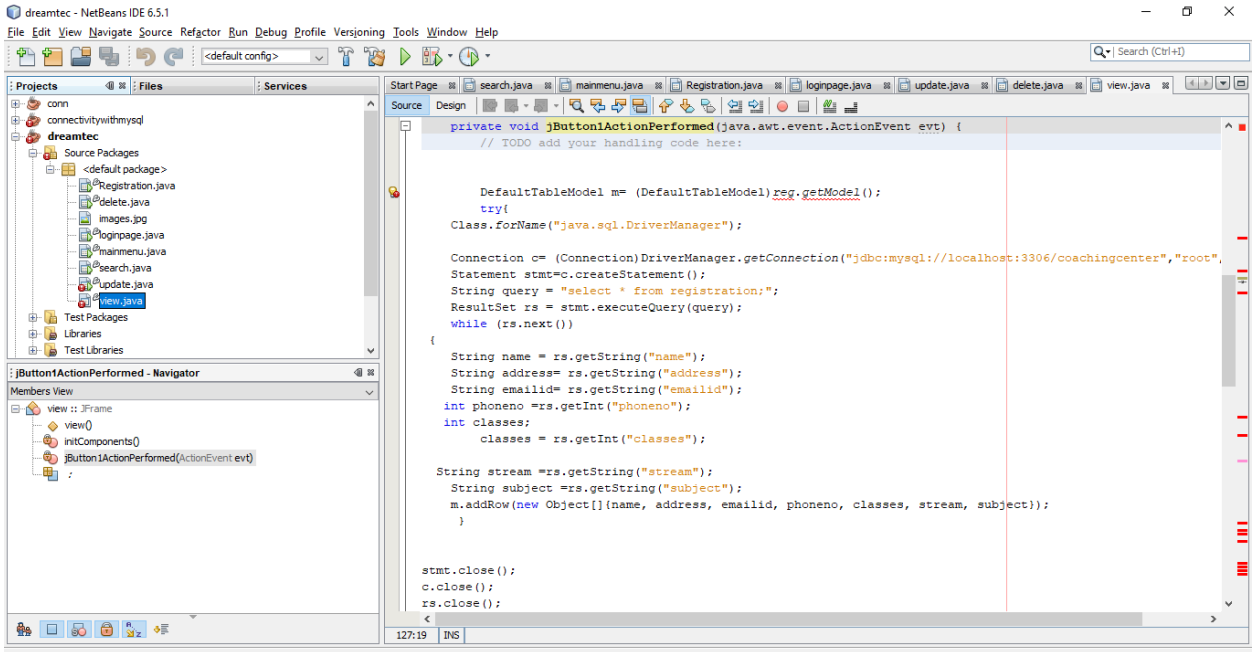

dreamtec - NetBeans IDE 6.5.1
File Edit View Navigate Source Refactor Run Debug Profile Versioning Tools Window Help
Projects
Files
Services
conn
connectivitywithmysql
dreamtec
Source Packages
<default package>
Registration.java
delete.java
images.jpg
loginpage.java
mainmenu.java
search.java
update.java
view.java
Test Packages
Libraries
Test Libraries
Navigator
Members View
loginpage :: JFrame
loginpage()
initComponents()
jButton1ActionPerformed(ActionEvent evt)
main(String[] args)
jButton1 : JButton
jLabel1 : JLabel
jLabel2 : JLabel
p1 : JPasswordField
t1 : JTextField
Source
Design
System.out.println("0");
Connection c = DriverManager.getConnection("jdbc:mysql://localhost:3306/dreamtech","root","root"
System.out.println("1");
Statement stmt = c.createStatement();
String phoneno= "delete from registration where phoneno = "+t1.getText()+" ";
System.out.println("2");
stmt.executeUpdate(phoneno);
JOptionPane.showMessageDialog(null, "Record deleted") ;
System.out.println("3");
}
catch (Exception e) {
JOptionPane.showMessageDialog(null,"error");
}
}
t2.setText("");
t3.setText("");
}
/**
* @param args the command line arguments
*/
public static void main(String args[]) {
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
new delete().setVisible(true);
}
});
}
158:6
INS

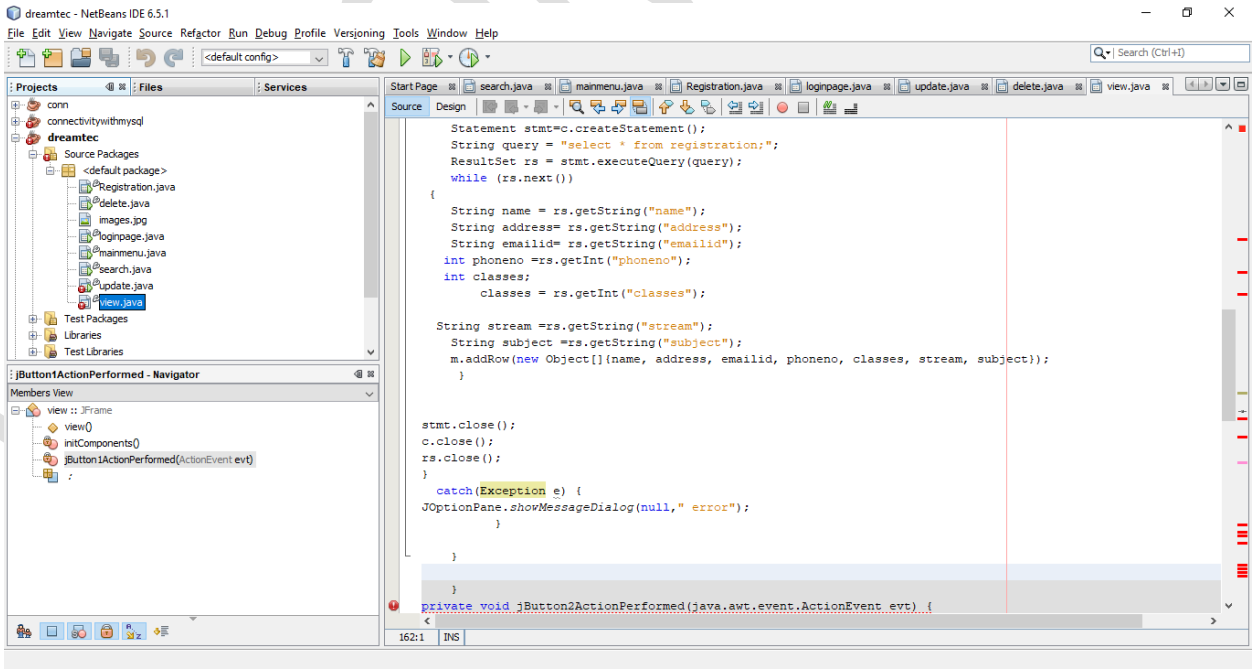
6 UPDATE



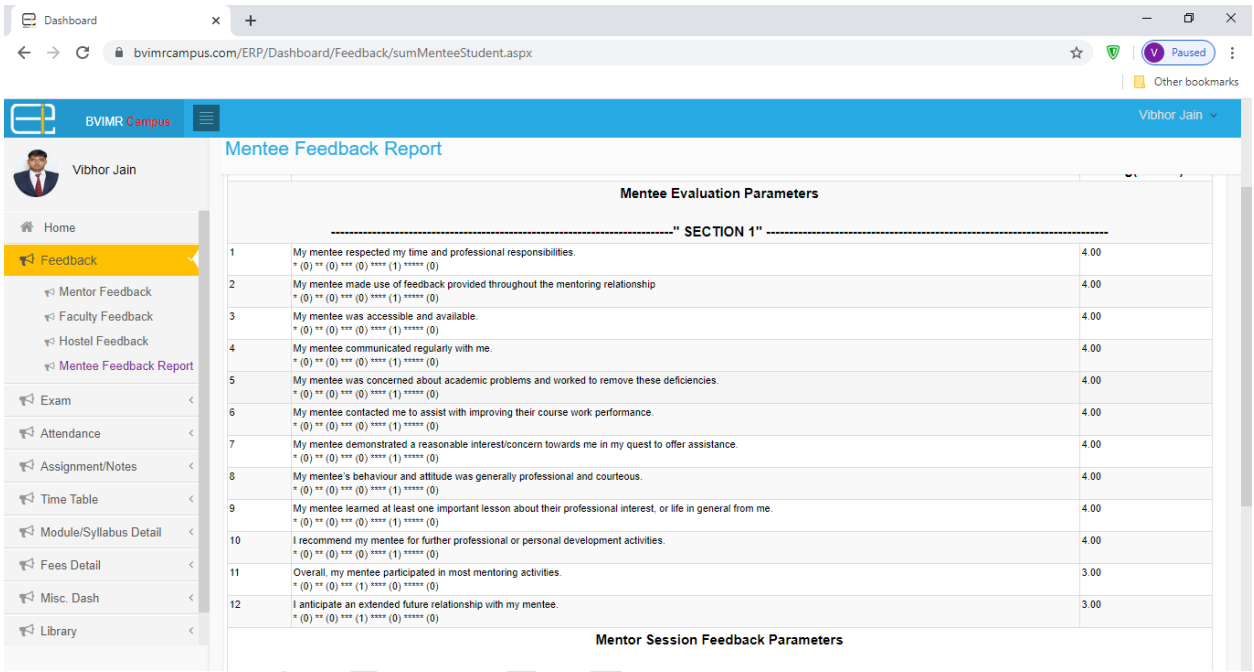


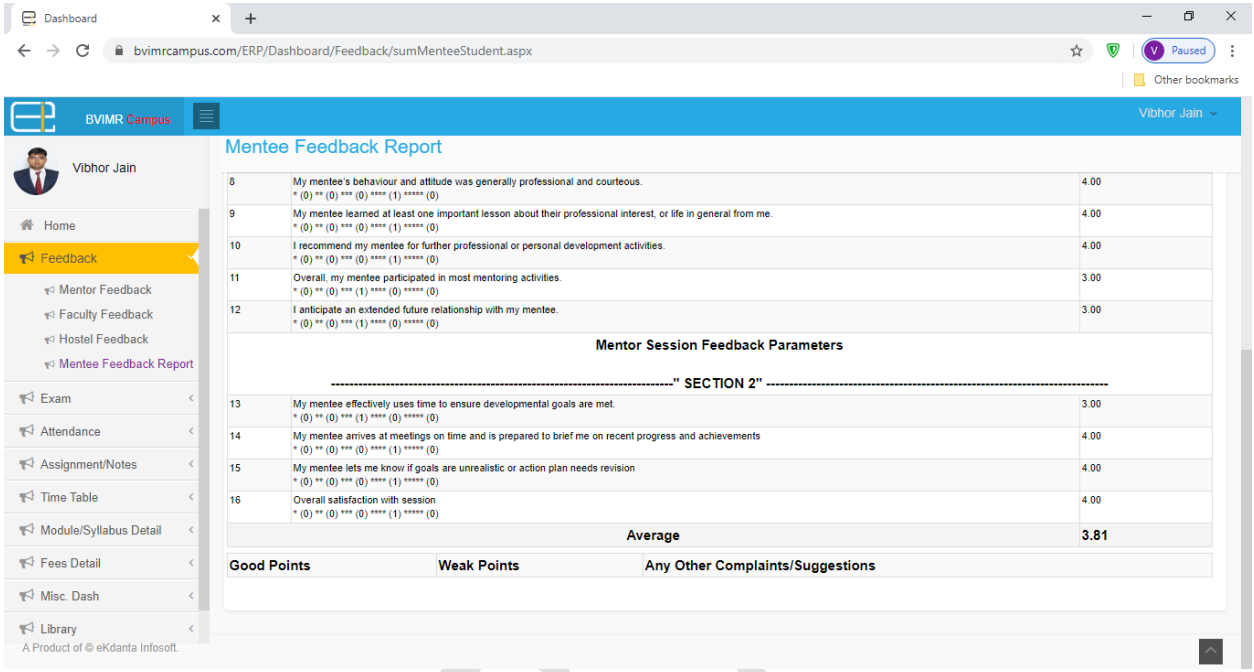
7 VIEW


dreamtec - NetBeans IDE 6.5.1
File Edit View Navigate Source Refactor Run Debug Profile Versioning Tools Window Help
Projects
Files
Services
conn
connectivitywithmysql
dreamtec
Source Packages
<default package>
Registration.java
delete.java
images.jpg
loginpage.java
mainmenu.java
search.java
update.java
view.java
Test Packages
Libraries
Test Libraries
jButton1ActionPerformed - Navigator
Members View
view :: JFrame
view()
initComponents()
jButton1ActionPerformed(ActionEvent evt)
Start Page
search.java
mainmenu.java
Registration.java
loginpage.java
update.java
delete.java
view.java
Source
Design
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
DefaultTableModel m= (DefaultTableModel)reg.getModel();
try{
Class.forName("java.sql.DriverManager");
Connection c= (Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/coachingcenter","root",
Statement stmt=c.createStatement();
String query = "select * from registration;";
ResultSet rs = stmt.executeQuery(query);
while (rs.next())
{
String name = rs.getString("name");
String address= rs.getString("address");
String emailid= rs.getString("emailid");
int phoneno =rs.getInt("phoneno");
int classes;
classes = rs.getInt("classes");
String stream =rs.getString("stream");
String subject =rs.getString("subject");
m.addRow(new Object[]{name, address, emailid, phoneno, classes, stream, subject});
}
stmt.close();
c.close();
rs.close();
127:19
INS


dreamtec - NetBeans IDE 6.5.1
File Edit View Navigate Source Refactor Run Debug Profile Versioning Tools Window Help
Projects
Files
Services
conn
connectivitywithmysql
dreamtec
Source Packages
<default package>
Registration.java
delete.java
images.jpg
loginpage.java
mainmenu.java
search.java
update.java
view.java
Test Packages
Libraries
Test Libraries
jButton1ActionPerformed - Navigator
Members View
view :: JFrame
view()
initComponents()
jButton1ActionPerformed(ActionEvent evt)
Start Page
search.java
mainmenu.java
Registration.java
loginpage.java
update.java
delete.java
view.java
Source
Design
Statement stmt=c.createStatement();
String query = "select * from registration;";
ResultSet rs = stmt.executeQuery(query);
while (rs.next())
{
String name = rs.getString("name");
String address= rs.getString("address");
String emailid= rs.getString("emailid");
int phoneno =rs.getInt("phoneno");
int classes;
classes = rs.getInt("classes");
String stream =rs.getString("stream");
String subject =rs.getString("subject");
m.addRow(new Object[]{name, address, emailid, phoneno, classes, stream, subject});
}
stmt.close();
c.close();
rs.close();
}
catch(Exception e) {
JOptionPane.showMessageDialog(null," error");
}
}
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
162:1
INS

Mentor Feedback Report





PLAGARISIM REPORT