**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**

(AN AUTONOMOUS INSTITUTION)

(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM, APPROVED BY AICTE & GOVT.OF KARNATAKA)

****

JAVA COURSE-PROJECT REPORT

ON

**DATABASE CREATOR USING JAVA**

*Submitted by:*

1. RAHUL SARAN (INT15EC126)
2. SUMANTH G V (1NT15EC159)
3. TUSHAR C DESAI (1NT15EC168)

In partial fulfillment of the requirements for the completion of *IV Semester Object-Oriented Programming Course-Project work* during the academic year 2016-2017.

Department of Computer Science and Engineering

NitteMeenakshi Institute of Technology,

Yelahanka, Bangalore– 560064

Academic Year 2017-18

**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**

(AN AUTONOMOUS INSTITUTION)

(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM, APPROVED BY AICTE & GOVT.OF KARNATAKA)

****

**CERTIFICATE**

**This is to certify that the Project Report**

**DATABASE CREATOR USING JAVA**

Is an authentic work carried out by

1. RAHUL SARAN (1NT15EC126)
2. SUMANTH GV (1NT15EC159)
3. TUSHAR.C.DESAI(1NT15EC168)

In partial fulfillment of the requirements for the completion of *VI Semester, Introduction to Java, Open Elective Course-Project work* during the academic year 2017-2018.

Name & Signature of the Guide Name & Signature of HOD

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of any task would be but incomplete without the meaning of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

I would express our gratitude to our **Dr. Thippeswamy .M.N** for his valuable support without which the project would not have been completed.

I would express our gratitude to our lecturer **Mrs. VIJAYA SHETTY** for constant support to complete the project.

I would like to mention a special thanks to staff of CSE department, NMIT for their support.

**ABSTRACT**

A **database application** is a computer program whose primary purpose is entering and retrieving information from a computerized [database](https://en.wikipedia.org/wiki/Database). Early examples of database applications were accounting systems and airline reservations systems, such as [SABRE](https://en.wikipedia.org/wiki/Sabre_(computer_system)), developed starting in 1957.

A characteristic of modern database applications is that they facilitate simultaneous updates and queries from multiple users. Systems in the 1970s might have accomplished this by having each user in front of a [3270](https://en.wikipedia.org/wiki/3270) terminal to a [mainframe computer](https://en.wikipedia.org/wiki/Mainframe_computer). By the mid-1980s it was becoming more common to give each user a [personal computer](https://en.wikipedia.org/wiki/Personal_computer) and have a program running on that PC that is connected to a database [server](https://en.wikipedia.org/wiki/Server_(computing)). Information would be pulled from the database, transmitted over a network, and then arranged, graphed, or otherwise formatted by the program running on the PC. Starting in the mid-1990s it became more common to build database applications with a Web interface. Rather than develop custom software to run on a user's PC, the user would use the same Web browser program for every application. A database application with a Web interface had the advantage that it could be used on devices of different sizes, with different hardware, and with different operating systems. Examples of early database applications with Web interfaces include [amazon.com](https://en.wikipedia.org/wiki/Amazon.com), which used the [Oracle relational database management system](https://en.wikipedia.org/wiki/Oracle_database), the photo.net online community, whose implementation on top of [Oracle](https://en.wikipedia.org/wiki/Oracle_database) was described in the book Database-Backed Web Sites (Ziff-Davis Press; May 1997), and [eBay](https://en.wikipedia.org/wiki/EBay), also running Oracle.[[1]](https://en.wikipedia.org/wiki/Database_application#cite_note-ebaypresentation-1)

[Electronic medical records](https://en.wikipedia.org/wiki/Electronic_medical_records) are referred to on emrexperts.com,[[2]](https://en.wikipedia.org/wiki/Database_application" \l "cite_note-emrexperts.com-2) in December 2010, as "a software database application". A 2005 O'Reilly book uses the term in its title: Database Applications and the Web.

Some of the most complex database applications remain accounting systems, such as [SAP](https://en.wikipedia.org/wiki/SAP_ERP), which may contain thousands of tables in only a single module.[[3]](https://en.wikipedia.org/wiki/Database_application#cite_note-oreillysapbook-3) Many of today's most widely used computer systems are database applications, for example, Facebook, which was built on top of [MySQL](https://en.wikipedia.org/wiki/MySQL).[[4]](https://en.wikipedia.org/wiki/Database_application#cite_note-4)

The etymology of the phrase "database application" comes from the practice of dividing computer software into systems programs, such as the operating system, compilers, the file system, and tools such as the database management system, and application programs, such as a payroll check processor. On a standard PC running Microsoft Windows, for example, the Windows operating system contains all of the systems programs while games, word processors, spreadsheet programs, photo editing programs, etc. would be application programs. As "application" is short for "application program", "database application" is short for "database application program".

**HIGHLIGHTS OF THE PROJECT**

The existing application allows to create a database using java.

**LANGUAGE USED :**  Java

**COMPILER :**  Eclipse Luna JEE

**OPERATING SYSTEM :** Windows 8

SQL is a standard language for storing, manipulating and retrieving data in databases.

Our SQL tutorial will teach you how to use SQL in: MySQL, SQL Server, MS Access, Oracle, Sybase, Informix, Postgres, and other database systems.

* SQL stands for Structured Query Language
* SQL lets you access and manipulate databases
* SQL is an ANSI (American National Standards Institute) standard

If you are adding values for all the columns of the table, you do not need to specify the column names in the SQL query. However, make sure the order of the values is in the same order as the columns in the table. The INSERT INTO syntax would be as follows:

It is possible to write the INSERT INTO statement in two ways.

The first way specifies both the column names and the values to be inserted:

INSERT INTO table\_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);

INSERT INTO table\_name  
VALUES (value1, value2, value3, ...);

The DELETE statement is used to delete existing records in a table.

**Note:** Be careful when deleting records in a table! Notice the WHERE clause in the DELETE statement. The WHERE clause specifies which record(s) that should be deleted. If you omit the WHERE clause, all records in the table will be deleted!

It is possible to delete all rows in a table without deleting the table. This means that the table structure, attributes, and indexes will be intact:

DELETE FROM table\_name;

Or

DELETE \* FROM table\_name;

**CORELATING THEORY CONCEPTS**

Class

A class is a user defined blueprint or prototype from which objects are created.  It represents the set of properties or methods that are common to all objects of one type. In general, class declarations can include these components, in order:

1. **Modifiers** : A class can be public or has default access (Refer [this](https://www.geeksforgeeks.org/access-specifiers-for-classes-or-interfaces-in-java/) for details).
2. **Class name:** The name should begin with a initial letter (capitalized by convention).
3. **Superclass(if any):** The name of the class’s parent (superclass), if any, preceded by the keyword extends. A class can only extend (subclass) one parent.
4. **Interfaces(if any):** A comma-separated list of interfaces implemented by the class, if any, preceded by the keyword implements. A class can implement more than one interface.
5. **Body:** The class body surrounded by braces, { }.

Constructors are used for initializing new objects. Fields are variables that provides the state of the class and its objects, and methods are used to implement the behavior of the class and its objects.

Object is a basic unit of Object Oriented Programming and represents the real life entities.  A typical Java program creates many objects, which as you know, interact by invoking methods. An object consists of :

1. **State**: It is represented by attributes of an object. It also reflects the properties of an object.
2. **Behavior**: It is represented by methods of an object. It also reflects the response of an object with other objects.
3. **Identity**: It gives a unique name to an object and enables one object to interact with other objects.

When an object of a class is created, the class is said to be **instantiated**. All the instances share the attributes and the behavior of the class. But the values of those attributes, i.e. the state are unique for each object. A single class may have any number of instances.

Functions or Methods

A Java method is a collection of statements that are grouped together to perform an operation. When you call the System.out.println() method, for example, the system actually executes several statements in order to display a message on the console.

Now you will learn how to create your own methods with or without return values, invoke a method with or without parameters, and apply method abstraction in the program design.

For using a method, it should be called. There are two ways in which a method is called i.e., method returns a value or returning nothing (no return value).

The process of method calling is simple. When a program invokes a method, the program control gets transferred to the called method. This called method then returns control to the caller in two conditions, when −

* the return statement is executed.

it reaches the method ending closing brace.

* Intuitive web interface
* Support for most MySQL features:
  + browse and drop databases, tables, views, fields and indexes
  + create, copy, drop, rename and alter databases, tables, fields and indexes
  + maintenance server, databases and tables, with proposals on server configuration
  + execute, edit and bookmark any SQL-statement, even batch-queries
  + manage MySQL user accounts and privileges
  + manage stored procedures and triggers
* Import data from CSV and SQL
* Export data to various formats: CSV, SQL, XML, PDF, ISO/IEC 26300 - OpenDocument Text and Spreadsheet, Word, LATEX and others
* Administering multiple servers
* Creating graphics of your database layout in various formats
* Creating complex queries using Query-by-example (QBE)
* Searching globally in a database or a subset of it
* Transforming stored data into any format using a set of predefined functions, like displaying BLOB-data as image or download-link

**PROGRAMMING CODE**

import java.awt.EventQueue;

public class DataEntry {

private JFrame frame;

private JTextField txtDataentryPage;

private JButton btnDeleteData;

String datas[][]=new String[20][20];

int i,j;

String col1[]={"firstname","lastmname","emailid","usn"};

// private JPanel contentPane;

//private JFrame frame;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

DataEntry window = new DataEntry();

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the application.

\*/

public DataEntry() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

private void initialize() {

frame = new JFrame();

frame.setBounds(100, 100, 450, 600);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.getContentPane().setLayout(null);

txtDataentryPage = new JTextField();

txtDataentryPage.setFont(new Font("Tw Cen MT", Font.BOLD, 17));

txtDataentryPage.setText("DataEntry Page");

txtDataentryPage.setBounds(105, 11, 136, 25);

frame.getContentPane().add(txtDataentryPage);

txtDataentryPage.setColumns(10);

JButton btnInsertData = new JButton("Insert Data");

btnInsertData.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) //here is change

{

InsertData iobj=new InsertData();

iobj.setVisible(true);

}

});

btnInsertData.setBounds(32, 72, 102, 23);

frame.getContentPane().add(btnInsertData);

JButton btnDisplay = new JButton("Display Data ");

btnDisplay.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e)

{

//ViewFrame vobj=new ViewFrame();

//vobj.setVisible(true);

try

{

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

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

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery("select \* from data1");

while(rs.next())

{

datas[i][j]=rs.getString(1);

System.out.println(datas[i][j]);

j++;

datas[i][j]=rs.getString(2);

System.out.println(datas[i][j]);

j++;

datas[i][j]=rs.getString(3);

System.out.println(datas[i][j]);

j++;

datas[i][j]=rs.getString(4);

System.out.println(datas[i][j]);

i++;

j--;

}

JTable jt2=new JTable(datas,col1);

JScrollPane js=new JScrollPane(jt2);

js.setBounds(83, 300, 250, 100);

frame.getContentPane().add(js);

con.close();

}

catch(Exception e1)

{

System.out.println(e1.getMessage());

}

}

});

btnDisplay.setBounds(188, 72, 95, 23);

frame.getContentPane().add(btnDisplay);

btnDeleteData = new JButton("Delete Data");

btnDeleteData.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e)

{

DeleteFrame1 dobj=new DeleteFrame1();

dobj.setVisible(true);

}

});

btnDeleteData.setBounds(105, 162, 102, 23);

frame.getContentPane().add(btnDeleteData);

}

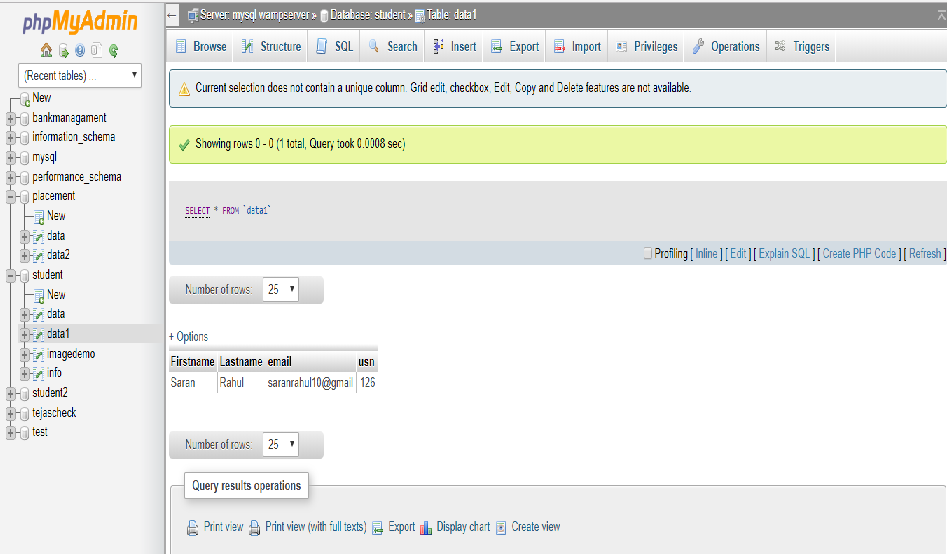
}

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Sl.No** | **Description** | **Page No.** |
|  | Acknowledgement | 3 |
|  | Abstract | 4 |
|  | Related Concepts | 6 |
|  | Program code | 8 |
|  | Snapshots | 14 |
|  | Conclusion | 16 |
| 7. | Plagiarism Report | 14 |
|  |  |  |

**SNAPSHOTS**

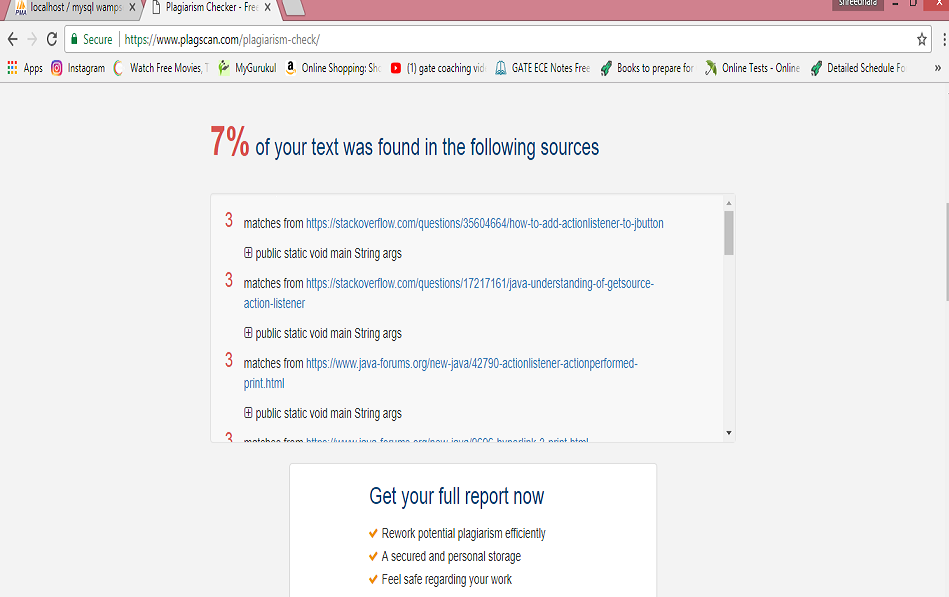
****

****

**CONCLUSION**

The agenda of this project is to create databases using java and store the data in the cloud and search for the databases in cloud whenever required. The data can be inserted or deleted or displayed in the databases using java.

**PLAGIARISM REPORT**

****

**BIBLIOGRAPHY**

* A complete reference to Java J2EE programming by Herbert Schildt.
* Wikipedia
* www.google,com