***INTRODUCTION TO PROJECT***

Our project is based on an offline version of justdial or we may say an offline replica of justdial using java and SQL. It is a desktop application.

It tells some categories on thing near by which have been saved in the database it just retrieves the data and show to the user and if the user wants to change the details it can be changed. It helps to locate the nearby shops, garages easily.

To enter the you need to login without it you can’t enter the app. This app also let us add new entries from user so as we can get more of data to show other user. It also increases the efficiency of the app.

It also lets the user delete the data which lets the app to get rid of incorrect data. The modification of data is also possible it lets correction in data which improve the app performance.

The most important option which is provided that is search which let the user to search the near by location based on the criteria provided in the app to get the nearby location present in the database.

***TECHNOLOGY USED***

The technology used are: -

1. JAVA
2. SQL

***JAVA***

It is a programming language widely used across world among programmer for developing app, websites and many other things.

This language was being in programming world by James Gosling in 1991. It was brought to overcome the features of C/C++.

The language can be used to develop both offline and online applications. Unlike C++ it’s a complete or pure object-oriented language.

This language is different from other as it does not just compile the code and runs it but first it compiles the code then convert it into a byte code and then interprets it and executes. The benefit of creating the byte code makes the language architecturally neutral and portable.

To execute we use Java Virtual Machine (JVM). JVM is basically used to interpret and execute the byte code.

Java provided support to the coder by providing an endless inbuilt library which helps to create application designs.

***SQL***

It is language which allows user to store data in organized format. It follows all rules RDBMS i.e. Relational Database Management System.

In this language we use queries to execute. There are many software providing this language but we have used MySQL.

***HARDWARE/SOFTWARE REQUIREMENTS***

* Intel CORE i3
* 2 GB RAM
* Windows 7/8/10, Ubuntu 14.06/16.06/19.06
* HDD 5 GB FREE
* JDK 1.6 and above
* Eclipse or NetBeans
* MySQL or Oracle

***DESCRIPTION OF SOFTWARE LIBRARY USED***

Main Library used are: -

1. Swing
2. SQL jar External
3. Awt

Sub Library used are: -

1. Font
2. JLabel
3. JFrame
4. Color
5. Font
6. Image
7. ImageIcon
8. JPanel
9. JButton
10. ActionListener
11. BorderFactory
12. JTextField
13. FlowLayout
14. JOptionPane
15. Connection
16. DriverManager
17. DriverManager
18. ResultSet
19. SQLException
20. Statement

***java.awt.Font: -***

This class is used for adding font, font size, font type (like bold, italics etc.). it can set it anywhere in code one initialized.

For ex: - f1 = **new** Font("Serif",Font.***BOLD***,25);

***java.awt.Color: -***

This class help to color anything in the code like giving color to a background or giving color to text or a button or to border of panel or background color of a panel.

For ex: - label.setForeground(Color.***white***);

***java.awt.Image: -***

This class help to give changes to a given image like changing size, resizing it etc.

For ex: - Image temp\_img = i.getScaledInstance(1920, 1200, Image.***SCALE\_SMOOTH***);

***java.awt.event.ActionListener: -***

This class help to activate button i.e. to provide operation to the that’s to be performed after clicking it.

For ex: - but3.addActionListener(**this**);

***java.swing.FlowLayout: -***

This class is for setting the frame items in arrangement without providing items there location in a linear format.

For ex: - FlowLayout layout =new FlowLayout();

login.setLayout(layout);

***java.swing.ImageIcon: -***

This class helps to add image the frame on which everything is added. Basically it helps to get the image on the screen i.e. to get the image from the system to the displaying area.

For ex: - img = new ImageIcon("src\\images.jpg ");

***java.swing.JButton: -***

This class helps to create button in the application and can change the properties of button like size, placement, color etc.

For ex: - okay = new JButton("Login ");

***java.swing.JLabel: -***

This class helps to create a label and also can change the properties of label like size, intext, placement etc.

For ex: - lblu = new JLabel("Enter user name here");

***java.swing.JPanel: -***

This class helps to create a panel in the application and let’s change the properties like size, border, placement etc.

For ex: - p1 = new JPanel();

p1.setBounds(500, 100, 400, 180);

***java.swing.JFrame: -***

This class helps to create and manipulate a frame.

For ex: - login = new JFrame("Login window ");

***java.swing.JOptionPane: -***

This class helps to print a message required to known that either a task is performed mostly after pressing the button.

For ex: - JOptionPane.showMessageDialog(login,"Sorry not registered");

***java.swing.JTextField: -***

This class helps to create and manipulate a text box.

For ex: - t3 = new JTextField(30);

***java.sql.Connection: -***

This class let’s us establish a connection between java program and database.

For ex: - con = DriverManager.getConnection("jdbc:mysql://localhost/aroundme", "root", "root");

***java.sql.DriverManager: -***

This class a driver which lets connection class to establish connection between java program and database.

For ex: - con = DriverManager.getConnection("jdbc:mysql://localhost/aroundme", "root", "root");

***java.sql.PreparedStatement: -***

This class helps to execute SQL statement in which there are missing entries and need to get it from user.

For ex: - pstmt = con.prepareStatement(sql);

***java.sql.ResultSet: -***

This class creates a virtual pointer in the SQL table so it can be accessed by the user for insertion, updating, deleting and searching.

For ex: - rs = stmt.executeQuery(msg);

***java.sql.Statement: -***

This class helps to execute SQL statement in which there is execution or updating.

For ex: - stmt = con.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,ResultSet.CONCUR\_UPDATABLE);

***java.sql.SQLException: -***

This class is for exception got during programs.

For ex: - catch (SQLException e)***SCREENSHOTS***

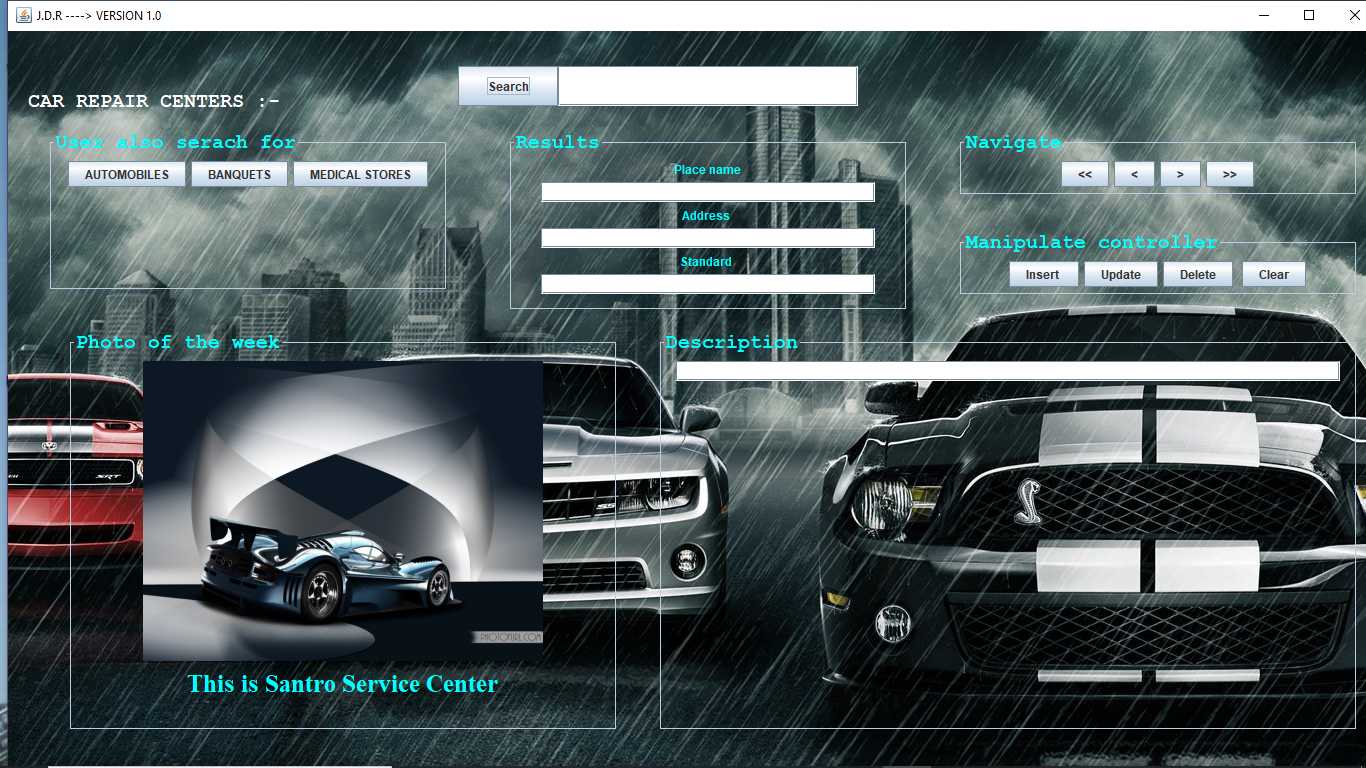
***LOGIN SCREEN: -***



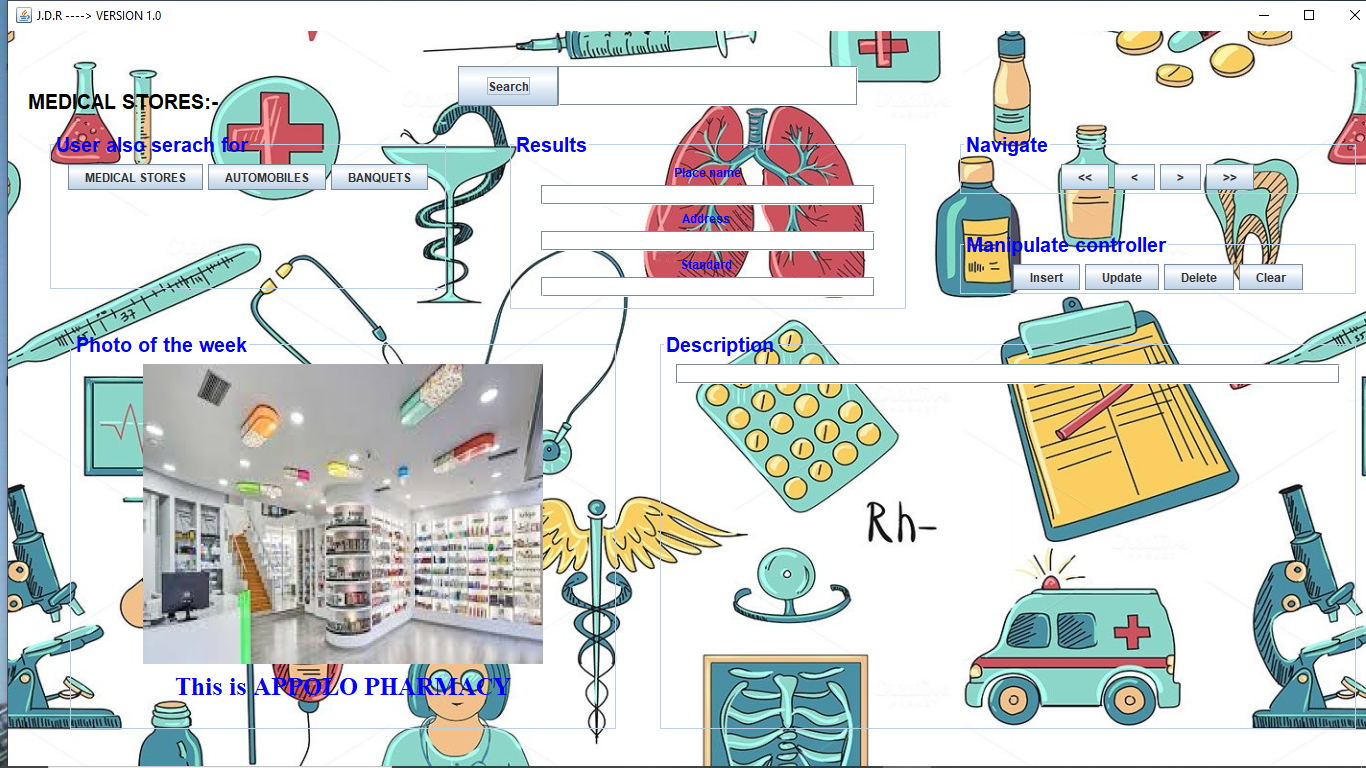
***BANQUET SCREEN: -***



***AUTOMOBILE SCREEN: -***



***MEDICALSTORE SCREEN: -***



***DATABASE TABLES***

**create** **table** automobiles

(

Name **varchar**(20),

Address **varchar**(30),

standard **varchar**(20),

description **varchar**(100),

state **varchar**(20)

);

**create** **table** banquets

(

Name **varchar**(20),

Address **varchar**(30),

standard **varchar**(20),

description **varchar**(100),

state **varchar**(20)

);

**create** **table** medicalstores

(

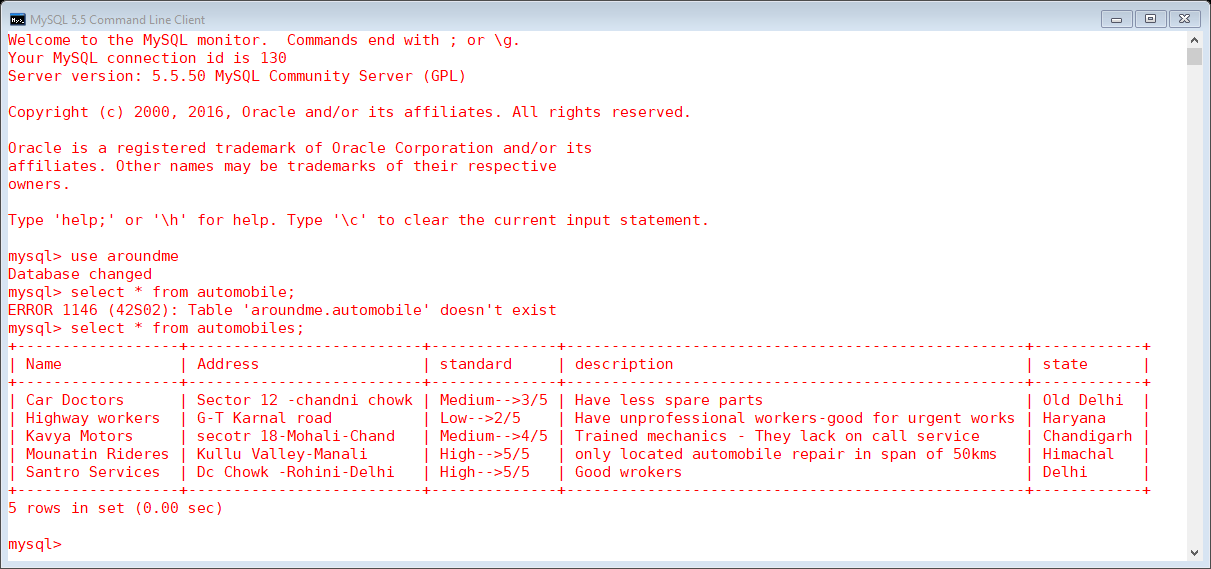
Name **varchar**(20),

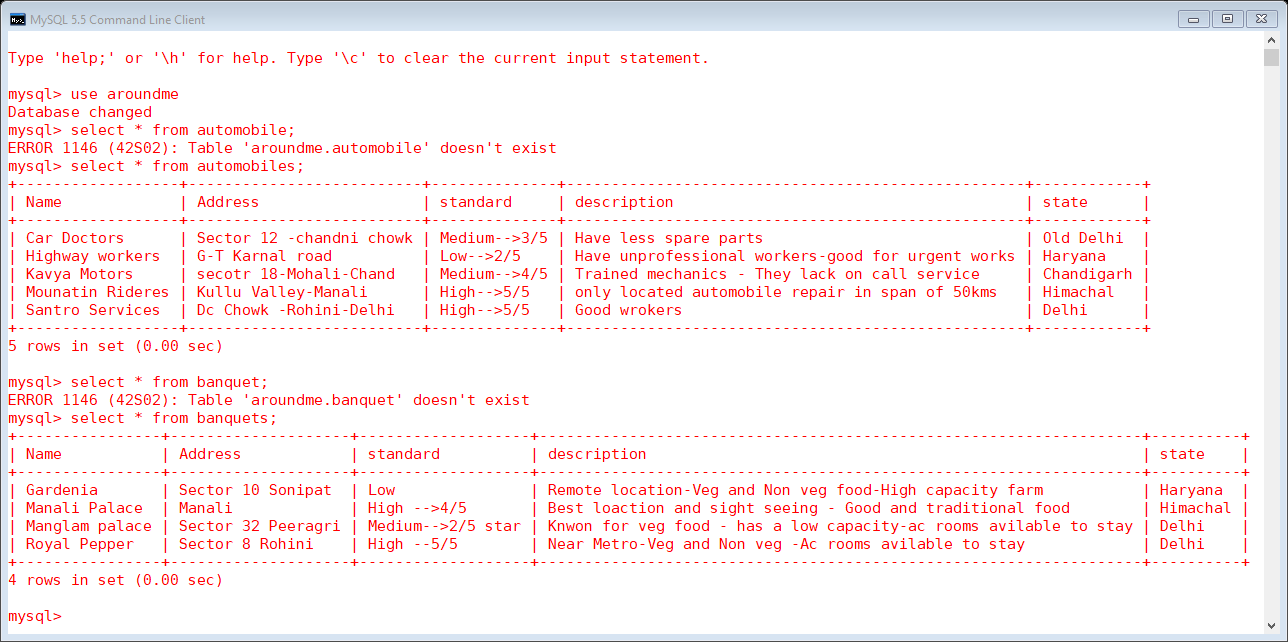
Address **varchar**(30),

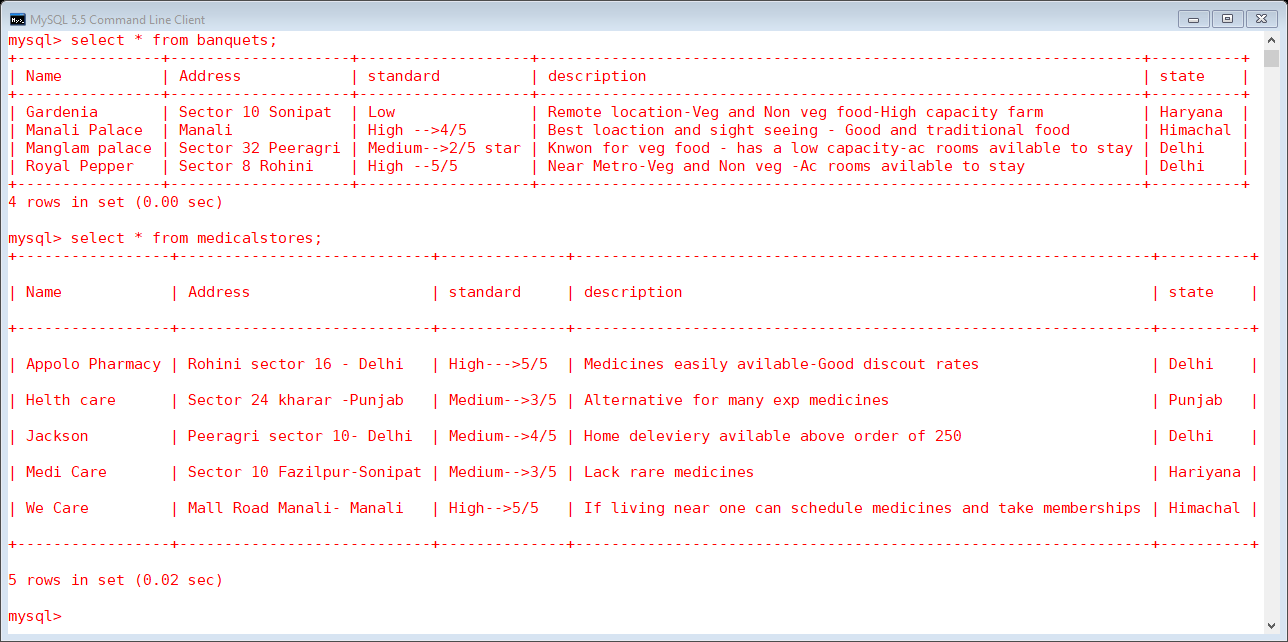
standard **varchar**(20),

description **varchar**(100),

state **varchar**(20));







***ADVANTAGES***

None

***DISADVANTAGES***

None

***REFERENCES***

Greekforgeek.com

Google.com

Sandeep Tiwari (Issac It labs)

Youtube.com