# ABSTRACT

---

This application is designed for the use of CAB BOOKING System and its functioning Main Objective of this project to facilitate the transactions through use of electronic medium. We use java NetBeans and MySQL to program our project. This project mainly deals with creating a java application regarding cab booking. For this application we will store some car model names, their registration number, rent rate on the basis of per day, and the amount to be deposited accordingly. Cab booking system provides reliable online cab(car) booking facility to people in various locations in India, free of cost. Bumpacars acts like a bridge between the cab operators and the users who book a cab. This is the online cab booking service provided to customers. This brings together the registered cab owners and the customers.

# AIM

---

The purpose of this document is to specify the cab booking functionality. This kind of development is now present is most of the developing cities. It would be user friendly and makes people on the easier side for transportation in large cities. This would change the traditional system and helpful in time management of the workload in the modern-day livelihood. Here the customers can book a cab /taxi/car by viewing all the cab details and pricing details available, according to selected city and area. It is the reliable service provided to both customers and drivers. This provides service with well-conditioned new vehicles, with experienced drivers for a happy journey of the customers.

# Algorithm

---

1. START

2. Create a database on MySQL and use the database[use <database_name>;]

3. Create tables for storing and retrieving the login data, for retrieving the driver details and for retrieving the distances between the locations.

4. Create the JFrames for a user WELCOME message and provide the mouseclicked event handler.

5. Create JFrame for Login and sign up details to be retrieved from the user.

6. Initialise the MySQL server and connect the JFrames(Login and Sign Up) with the database and tables for storing and retrieving the details. On Login the details are fetched from the logininfo table created in the bumpacars database. For Sign up, the values fetched from all the text fields are stored into a logininfo table into respective columns(also called attributes).

7. Create a JFrame to input pick-up location, drop location and to select the car model.

8. Create a JFrame to redirect the user to a confirmation page that shows driver name, fare, car number and a picture of the route map.

9. Provide the connection for the gmail stmp to connect with the internet and the email address of the user retrieved from the loginfo table(from the MySQL).

10. Create a confirmation JFrame to appear on sending the mail to the user.

11. END

# JAVA MAIL API

The JavaMail API provides a platform-independent and protocol-independent framework to build mail and messaging applications. The JavaMail API provides a set of abstract classes defining objects that comprise a mail system. It is an optional package (standard extension) for reading, composing, and sending electronic messages.

Following are some of the protocols supported in JavaMail API:
- SMTP: Acronym for Simple Mail Transfer Protocol. It provides a mechanism to deliver email.

- **POP**: Acronym for **Post Office Protocol**. POP is the mechanism most people on the Internet use to get their mail. It defines support for a single mailbox for each user. RFC 1939 defines this protocol.

- **IMAP**: Acronym for **Internet Message Access Protocol**. It is an advanced protocol for receiving messages. It provides support for multiple mailbox for each user, in addition to, mailbox can be shared by multiple users. It is defined in RFC 2060.

- **MIME** : Acronym for **Multipurpose Inter net Mail Extensions.** .It is not a mail transfer protocol. Instead, it defines the content of what is transferred: the format of the messages, attachments, and so on. NNTP and Others:There are many protocols that are provided by third-party providers. Some of them are Network News Transfer Protocol (NNTP), Secure Multipurpose Internet Mail Extensions (S/MIME) etc.

# JAVA

___

Java is a general purpose object oriented programming language . It is the first programming language that is not tied to any particular hardware or operating system. The language is based on the concept of an  object. Java is highly derived from C++. Most striking feature of the language is that it is platform neutral language.

There were five primary goals in the creation of the Java language:
• It should use the object-oriented programming methodology.
• It should allow the same program to be executed on multiple operating systems.
• It should contain built-in support for using computer networks.
• It should be designed to execute code from remote sources  securely.
• It should be easy to use and borrow the good parts of older object-oriented languages like C++.

# MySQL

SQL is not a procedural language. It is not used to define complex processes; we can use SQL to use commands that define and manipulate data. SQL is different from other languages.

• SQL is very readable.

• In SQL we always issue commands.

SQL statements fall into two groups:-

● <u>Data Definition Language</u> (DDL) – DDL statements are used to describe the tables and the data they contain. The subset of SQL statements used for modeling the structure (rather than the contents) of a database or cube. The DDL gives you the ability to create, modify, and remove databases and database objects.

● <u>Data Manipulation Language</u> (DML) – DML statements that are used to operate on data in the database. These are statements that allow you to create or alter objects (such as tables, indexes, views, and so on) in the database. The subset of SQL statements used to retrieve and manipulate data.

# NetBeans

NetBeans IDE is a free, open source, integrated development environment (IDE) that enables you to develop desktop, mobile and web applications. The IDE supports application development in various languages, including Java, HTML5, PHP and C++. The IDE provides integrated support for the complete development cycle, from project creation through debugging, profiling and deployment. The IDE runs on Windows, Linux, Mac OS X, and other UNIX-based systems.
The IDE provides comprehensive support for JDK 8 technologies and the most recent Java enhancements. It is the first IDE that provides support for JDK 8, Java EE 7, and JavaFX 2.

# PROJECT EXPLORER
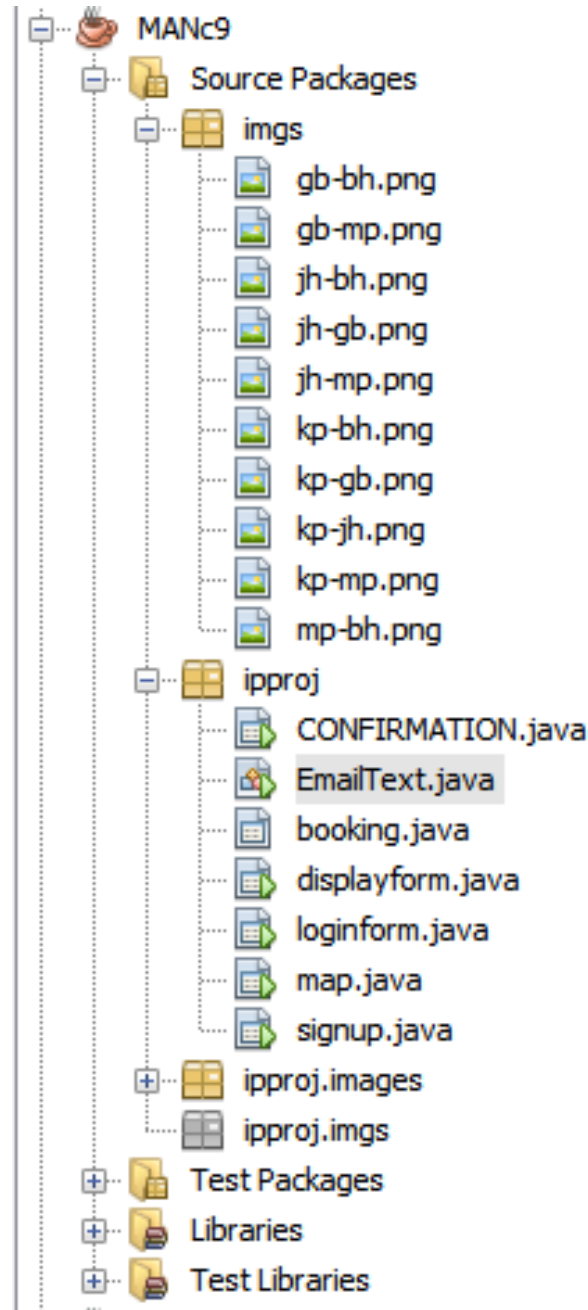
# JAVA FRAMES



DISPLAY FORM

## CODE (On Clicking the frame)-

```
private void jLabel1MouseClicked(java.awt.event.MouseEvent evt) {
new signupform().setVisible(true);
```

```java
  this.dispose();
 }
 public static void main(String args[]) {
java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new displayform().setVisible(true);
}
     });
   }
```

SIGNUP FORM

# Code for SignUp button

```
 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

String pwd1=new String(jPasswordField1.getPassword());
   String pwd2=new String(jPasswordField2.getPassword());
   System.out.println(pwd1);System.out.println(pwd2);
   String pass= new String(jPasswordField1.getPassword()).trim();
   String rpass= new String(jPasswordField2.getPassword()).trim();
   String email=new String(jTextField4.getText().trim());
if(jTextField3.getText().length()==0  ||  jTextField2.getText().length()==0  ||
jTextField1.getText().length()==0      ||rpass.length()==0||pass.length()==0||
email.length()==0)
```

```java
    {
        JOptionPane.showMessageDialog(null,"Fill the required fields");
    }
    else if(jTextField2.getText().length()!=10){
        JOptionPane.showMessageDialog(null,"Please enter a valid mobile
number");
    }

    else if(!(pwd1.equals(pwd2))){
            JOptionPane.showMessageDialog(null,"Passwords must match");
        }


    else{

        try{
            Connection con=DriverManager.getConnection("jdbc:mysql://
localhost:3306/bumpacars","root","");
            Statement stmt=con.createStatement();
            String str="insert into logininfo values('"+jTextField3.getText()
+"','"+jTextField1.getText()+"','"+pwd1+"','"+jTextField2.getText()+"','"+ email
+"');";
            System.out.println(str);
            int rows=stmt.executeUpdate(str);

            stmt.close();
            con.close();

        }catch(Exception e){
            System.out.println(e);
            JOptionPane.showMessageDialog(null,"error");
        }
        jTextField1.setText("");
```
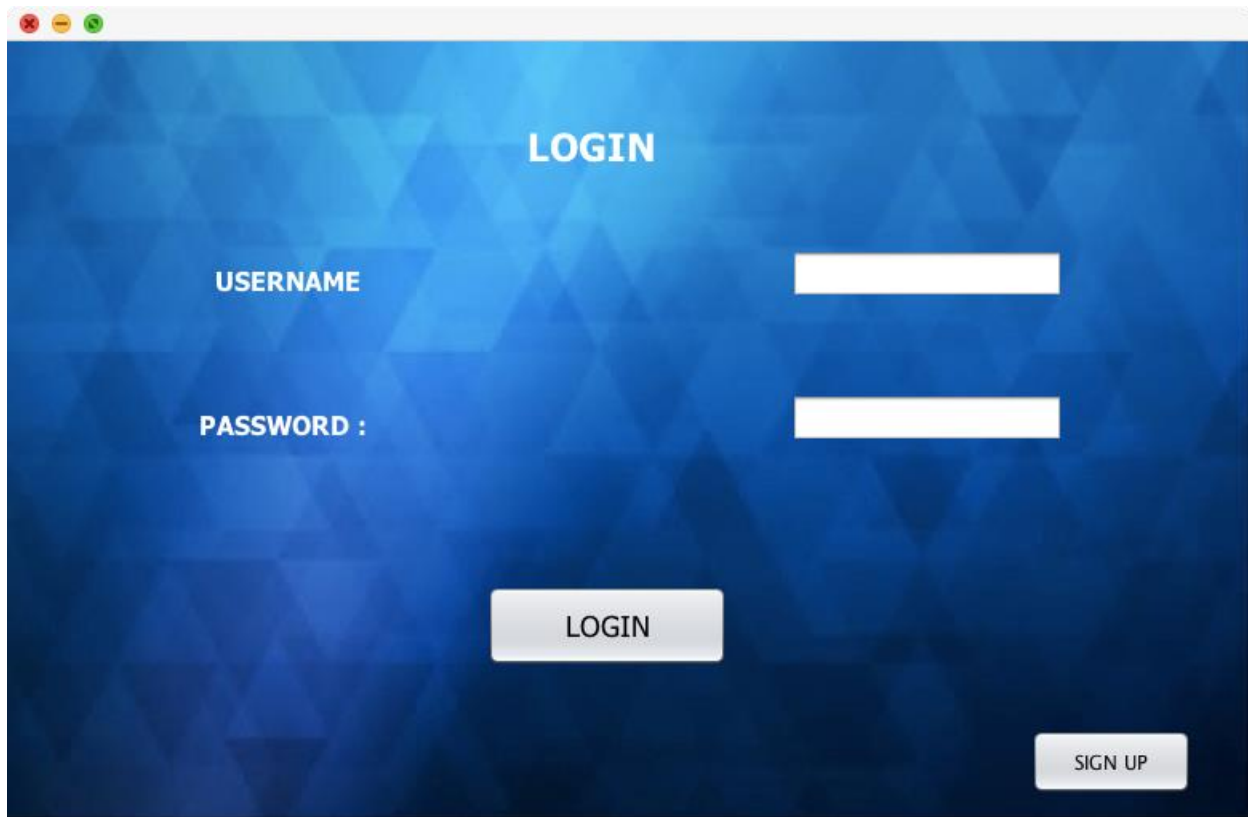
```
jTextField2.setText("");
jTextField3.setText("");
jTextField4.setText("");
jPasswordField1.setText("");
jPasswordField2.setText("");

}
```

## CODE FOR LOGIN BUTTON-

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
new loginform().setVisible(true);
this.dispose();
```

Login Form

# Code For Login Button-

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    String uname = jTextField1.getText();
    String pass= new String(jPasswordField1.getPassword());

    try{
        Class.forName("java.sql.Driver");
        Connection con=DriverManager.getConnection("jdbc:mysql://
localhost:3306/bumpacars","root","");
        Statement stmt=con.createStatement();
        String query="select userid,password,email from LOGININFO;";
        System.out.println(query);
        ResultSet rs=stmt.executeQuery(query);
```

```java
String DBuname;

    String DBpass,DBemail="";
boolean LoginF=false;
while(rs.next()) {
   DBuname=rs.getString(1);
   DBpass=rs.getString(2);
   DBemail=rs.getString(3);
   System.out.println( " " + DBuname + DBpass+DBemail );

      if(uname.equals(DBuname) && pass.equals(DBpass))
      {
                        LoginF=true;
         break;

      }



      }
if(LoginF == true){
     this.setVisible(false);
 new booking(DBemail).setVisible(true);
}
else
     {
       JOptionPane.showMessageDialog(this,"Invalid login or
password",
       "Error",JOptionPane.ERROR_MESSAGE);
     }
 }
catch(Exception e){
   JOptionPane.showMessageDialog(null,"Wrong UID or Password");
     }
```

# Code For SignUp button-

```java
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    new signup().setVisible(true);
    this.dispose();
 }
```

Booking Form

## CODE FOR NEXT BUTTON(retrieving details from user)-

```
public class booking extends javax.swing.JFrame {
    double carprice;
     String start;
    String end;
    String email1;
public booking(String emailID) {
    initComponents();
    this.email1=emailID;
    }
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    start=cb1.getSelectedItem().toString().trim(); //TODO add your handling
code here:
    end=cb2.getSelectedItem().toString().trim();
```

```java
        if (start.equals(end)){
            JOptionPane.showMessageDialog(null,"Please select different pickup
and dropoff location");


        }
        else if(mercedes.isSelected()== false && bmw.isSelected()== false &&
audi.isSelected()== false)
        { JOptionPane.showMessageDialog(null,"Please select a car");
            }
        else{
            System.out.println(email1);
        map mp=new map(String.valueOf(carprice),start,end,email1);
        mp.setVisible(true);
      //mp.main(new String[]{String.valueOf(carprice),start,end});

        this.dispose();
        }

 private void mercedesItemStateChanged(java.awt.event.ItemEvent evt) {
        if(mercedes.isSelected())
      {
         carprice=1.25*40;
      }
    }

    private void bmwItemStateChanged(java.awt.event.ItemEvent evt) {
      if(bmw.isSelected())
      {
         carprice=1.5*40;
      }
    }

    private void audiItemStateChanged(java.awt.event.ItemEvent evt) {
```
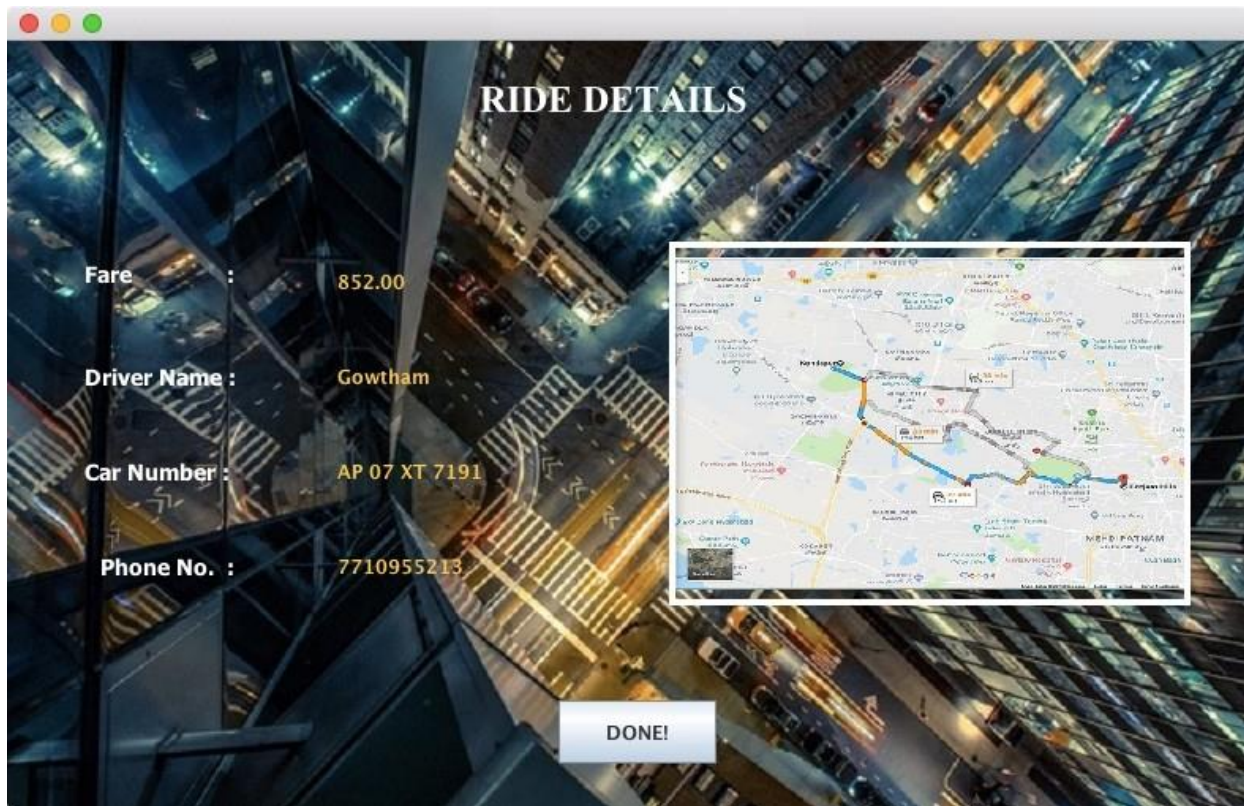
```
  if(audi.isSelected())
{
   carprice=40;
}
}
```

## CODE FOR BACK BUTTON-

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

Map Form

## CODE FOR RETRIEVING DETAILS FROM PREVIOUS FORM-

```
public map(String carp,String pick, String drop,String email) {
     initComponents();
  carprice=carp;
       start=pick;
      end=drop;
      this.email=email;
      try{

       Connection con=DriverManager.getConnection("jdbc:mysql://
localhost:3306/bumpacars","root","");
            String query = ("select distance from distance where pickup
='"+ start + "'&& dropoff='"+end+"';");
```

```java
System.out.println("q "+query);
  Statement  stmt=con.createStatement();
  ResultSet rs=stmt.executeQuery(query);
  rs.first();
  double dis = rs.getDouble("distance");
System.out.println("from map"+dis);
bill=Double.parseDouble(carprice) *  dis;
  System.out.println("bill"+bill);
  jLabel7.setText(""+ bill);
  if(start=="Banjara Hills")
  { jLabel8.setText("Balaraju");
    jLabel6.setText("AP 45 H 8324");
    jLabel11.setText("9090356116");


        }
  else if(start=="Jubilee Hills")
  { jLabel8.setText("Naveen");
  jLabel6.setText("TS 69 BK 007");
  jLabel11.setText("9056226186");


        }
  else if(start=="Kondapur")
  { jLabel8.setText("Gowtham");
    jLabel6.setText("AP 07 XT 7195");
    jLabel11.setText("7701955213");
  }
  else if(start=="Madhapur")
  { jLabel8.setText("Salim");
    jLabel6.setText("AP 15 BN 4056");
    jLabel11.setText("9110337877");
        }
  else
  { jLabel8.setText("Sushanth");
```

```java
            jLabel6.setText("AP 09 JK 1274");
            jLabel11.setText("8410984613");
            }
        System.out.println(""+jLabel7.getText());


        if(start=="Banjara Hills" && end=="Jubilee Hills" || start=="Jubilee
Hills" && end=="Banjara Hills" )
        {
            jLabel3.setIcon(new ImageIcon("src\\imgs\\jh-bh.png"));
        }
        else if(start=="Banjara Hills" && end=="Madhapur" ||
start=="Madhapur" && end=="Banjara Hills" )
        {
            jLabel3.setIcon(new ImageIcon("src\\imgs\\mp-bh.png"));
        }
         else if(start=="Banjara Hills" && end=="Gachibowli" ||
start=="Gachibowli" && end=="Banjara Hills" )
        {
            jLabel3.setIcon(new ImageIcon("src\\imgs\\gb-bh.png"));
        }
         else if(start=="Banjara Hills" && end=="Kondapur" ||
start=="Kondapur" && end=="Banjara Hills" )
        {
            jLabel3.setIcon(new ImageIcon("src\\imgs\\kp-bh.png"));
        }
         else if(start=="Gachibowli" && end=="Madhapur" ||
start=="Madhapur" && end=="Gachibowli" )
        {
            jLabel3.setIcon(new ImageIcon("src\\imgs\\gb-mp.png"));
        }
         else if(start=="Gachibowli" && end=="Kondapur" ||
start=="Kondapur" && end=="Gachibowli" )
        {
```

```java
            jLabel3.setIcon(new ImageIcon("src\\imgs\\kp-gb.png"));
        }
        else if(start=="Gachibowli" && end=="Jubilee Hills" || start=="Jubilee
Hills" && end=="Gachibowli" )
        {
            jLabel3.setIcon(new ImageIcon("src\\imgs\\jh-gb.png"));
        }
        else if(start=="Kondapur" && end=="Madhapur" ||
start=="Madhapur" && end=="Kondapur" )
        {
            jLabel3.setIcon(new ImageIcon("src\\imgs\\kp-mp.png"));
        }
        else if(start=="Kondapur" && end=="Jubilee Hills" || start=="Jubilee
Hills" && end=="Kondapur" )
        {
            jLabel3.setIcon(new ImageIcon("src\\imgs\\kp-jh.png"));
        }
        else
        {
            jLabel3.setIcon(new ImageIcon("src\\imgs\\jh-mp.png"));
        }
        System.out.println(email+"");



    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,e+"from map");

    }
  }
```

# CODE FOR NEXT BUTTON-

```java
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        EmailText ET=new EmailText();

        ET.main(new String[]
{String.valueOf(carprice),start,end,String.valueOf(bill),String.valueOf(email)});


        new CONFIRMATION().setVisible(true);
    this.dispose();
    }catch(Exception e){

    }
    }
```
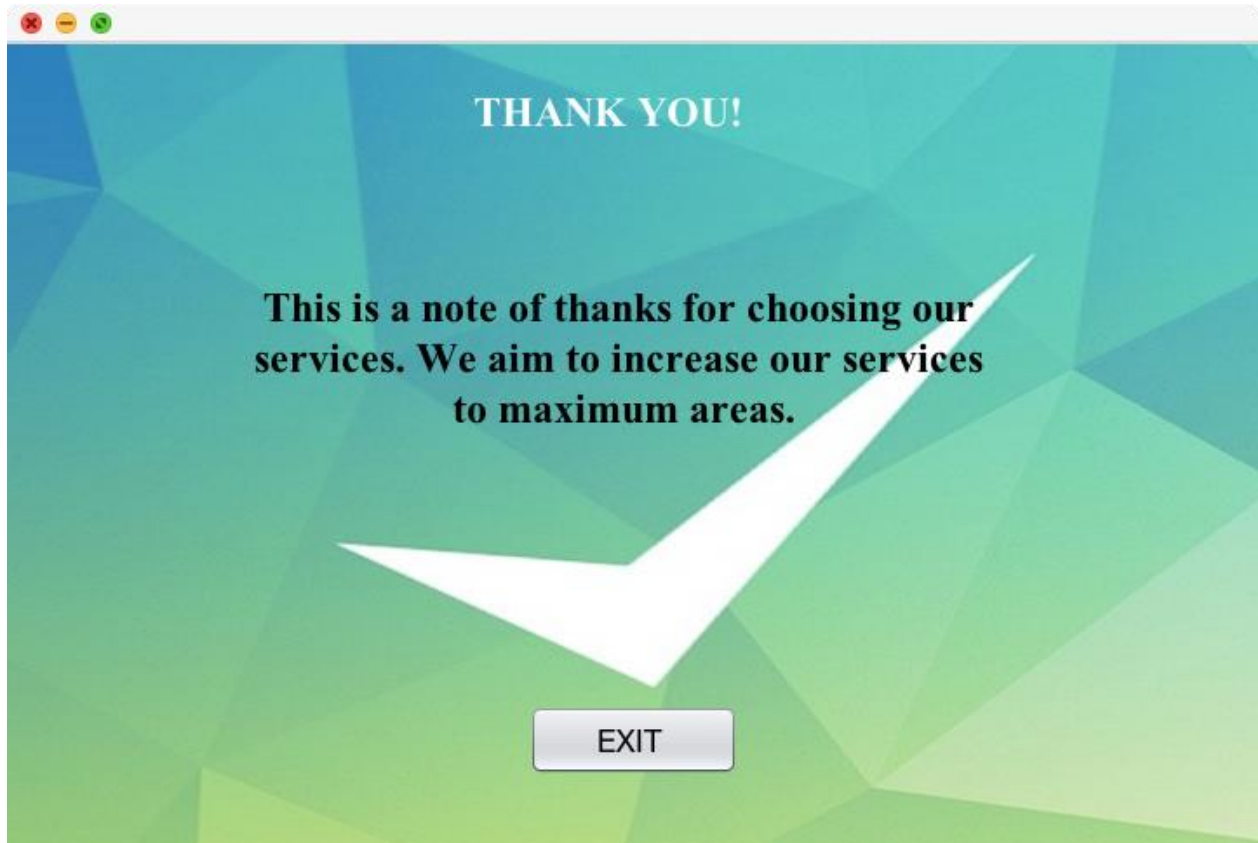
CONFIRMATION PAGE

# CODE FOR EXIT BUTTON-

System.exit(0);

# EMAIL TEXT

The code used below is used for sending mails to the user on successful booking.

```java
import static ipproj.map.carprice;
import java.util.Properties;

import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.AddressException;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;
public class EmailText {




    Properties emailProperties;
    Session mailSession;
    MimeMessage emailMessage;
   static String carPrice,start,end,bill,email;


    public static void main(String args[]) throws AddressException,
               MessagingException {
       carPrice=args[0];
        start=args[1];
        end=args[2];
```

```java
            bill=args[3];
            email=args[4];
            System.out.println("email"+email);
                EmailText javaEmail = new EmailText();


                javaEmail.setMailServerProperties();
                javaEmail.createEmailMessage();
                javaEmail.sendEmail();
        }

        public void setMailServerProperties() {

                String emailPort = "587";//gmail's smtp port

                emailProperties = System.getProperties();
                emailProperties.put("mail.smtp.port", emailPort);
                emailProperties.put("mail.smtp.auth", "true");
                emailProperties.put("mail.smtp.starttls.enable",  "true");


        }

        public void createEmailMessage() throws AddressException,
                    MessagingException {
                String[] toEmails = {email};
                String emailSubject = "This is an auto-generated bill";
                String emailBody = "Your cab ride from "+ start +" to "+ end +"
has been confirmed."+"\n"+"Your total bill is: "+bill+" rupees"
                    +"\n"+"\n"+"Thank you for choosing our service."+"\n"+"-
Team Bumpacars";

                mailSession = Session.getDefaultInstance(emailProperties, null);
                emailMessage = new MimeMessage(mailSession);
```

```java
            for (int i = 0; i < toEmails.length; i++) {
                    emailMessage.addRecipient(Message.RecipientType.TO,
new InternetAddress(toEmails[i]));
            }

            emailMessage.setSubject(emailSubject);
            emailMessage.setContent(emailBody, "text/html");//for a html
email
            //emailMessage.setText(emailBody);// for a text email

    }

    public void sendEmail() throws AddressException, MessagingException
{

            String emailHost = "smtp.gmail.com";
            String fromUser = "bumpacarscabs";//just the id alone without
@gmail.com
            String fromUserEmailPassword = "bumpacars123";


            Transport transport = mailSession.getTransport("smtp");

            transport.connect(emailHost, fromUser, fromUserEmailPassword);
            transport.sendMessage(emailMessage,
emailMessage.getAllRecipients());
            transport.close();
            System.out.println("Email sent successfully.");
    }

}
```

## This is an auto-generated bill  Inbox ×

🖨 ⤴

**bumpacarscabs@gmail.com**

to me ▾

Fri, Nov 27, 12:35 PM (2 days ago)  ☆  ↩  ⋮

Your cab ride from Madhapur to Gachibowli has been confirmed. Your total bill is: 648.0 rupees Thank you for choosing our service. -Team Bumpacars

↩ Reply    ➡ Forward

MAIL SENT TO USERS AFTER BOOKING

# MySQL TABLES -

**Tables created for the project-**

```
mysql> show tables;
+---------------------+
| Tables_in_bumpacars |
+---------------------+
| distance            |
| driver              |
| logininfo           |
+---------------------+
3 rows in set (0.00 sec)
```

**For creating table logininfo for storing login details of users-**

Create table logininfo(full name varchar(23), userid varchar(21), password varchar(21), phone int, email varchar(30));

```
mysql> select * from logininfo;
+----------+-------------+----------+------------+----------------------+
| fullname | userid      | password | phone      | email                |
+----------+-------------+----------+------------+----------------------+
| Gowtham  | gg123       | venkat   | 9848022338 | gowtham.vabc@gmail.com |
| Mohan    | mohanreddy  | #mb@#    | 9275177867 | mohanr123@gmail.com  |
| BIndu    | bindu999    | %6hi9    | 7998446532 | bindu999@yohoo.com   |
| kailash  | kailash3455 | HP       | 9876664554 | kail12@yohoo.com     |
| vishnu   | vct45       | piop     | 9455513456 | vishnucharan@yohoo.com |
+----------+-------------+----------+------------+----------------------+
5 rows in set (0.00 sec)
```

**For creating table driver for storing the driver name, car number, phone number -**

Create table driver( name varchar(20), car varchar(20), phone int );

```
mysql> select * from driver;
+-----------+--------------+-------------+
| name      | car          | phon        |
+-----------+--------------+-------------+
| Balaraju  | AP 45 H 1324 | 9090356116  |
| Naveen    | TS 69 BK 007 | 9056226186  |
| Lingoud   | AP 15 BN 4056| 9110337877  |
| Gowtham   | AP 07 XT 7191| 7710955213  |
| Sushanth  | AP 09 JK 1274| 8410984613  |
+-----------+--------------+-------------+
5 rows in set (0.05 sec)
```

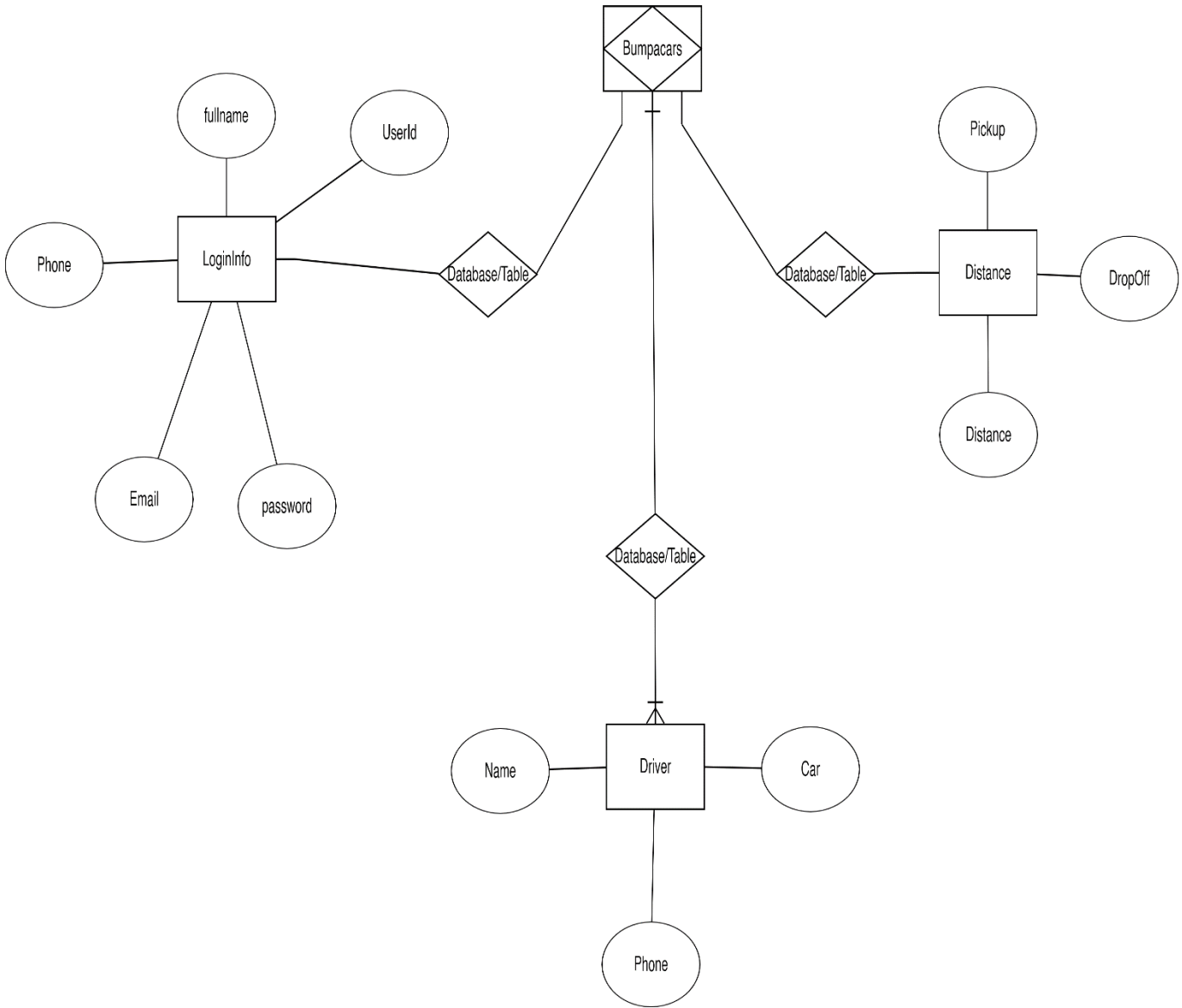**Table for creating table distance for storing the pickup and destination places with distance(in km)-**

Create table distance( pickup varchar(20), drop-off varchar(21) , distance float(2,1));

```
mysql> select * from distance;
+----------------+----------------+----------+
| pickup         | dropoff        | distance |
+----------------+----------------+----------+
| Banjara Hills  | Jubilee Hills  |      5.4 |
| Banjara Hills  | Madhapur       |      7.7 |
| Banjara Hills  | Gachibowli     |     13.9 |
| Jubilee Hills  | Gachibowli     |     11.2 |
| Jubilee Hills  | Madhapur       |      2.8 |
| Banjara Hills  | Kondapur       |     14.2 |
| Jubilee Hills  | Banjara Hills  |      5.4 |
| Jubilee Hills  | Kondapur       |      7.6 |
| Gachibowli     | Kondapur       |      6.8 |
| Gachibowli     | Jubilee Hills  |     11.2 |
| Gachibowli     | Banjara Hills  |     13.9 |
| Gachibowli     | Madhapur       |     10.8 |
| Kondapur       | Madhapur       |      6.4 |
| Kondapur       | Jubilee Hills  |      7.6 |
| Kondapur       | Banjara Hills  |     14.2 |
| Kondapur       | Gachibowli     |      6.8 |
| Madhapur       | Gachibowli     |     10.8 |
| Madhapur       | Banjara Hills  |      7.7 |
| Madhapur       | Kondapur       |      6.4 |
| Madhapur       | Jubilee Hills  |      2.8 |
+----------------+----------------+----------+
20 rows in set (0.00 sec)
```

# ER DIAGRAM

# REFERENCES

- https://stackoverflow.com

- https://stackexchange.com

- https://images.google.com

- https://en.wikipedia.org