ONLINE CAB SYSTEM (RENTALS/OUTSTATION)

Name : **Achalendra Velingkar**

Regd. No. : 21MIS0422

Date : 28- 11 - 2022

Course : SWE1007 – Programming in JAVA Course

Faculty :Prof. Nirmala M

Abstract

When it comes to cab rental services, Taxi Service is the most trusted and reliable name in the travel business. The most advanced travel agents offering cab rental and car hire in India, making full use of information technology to improve the level of our efficiency. However, this is only one aspect of services. And this project continually strives to offer the best of services - both in terms of man and machine, to our clients Moreover, this project has a fleet of cars ranging from luxury to budget cabs. While, it offers online cab hire service for corporate houses. And this project claims to offer the best of rates, which are tailor-made depending upon the facilities, availed and offer both intercity and intra-city cab facilities. All cabs have proper permits and documentation so that the clients couldn't be hassled for the lack of documents. However, this project has strategic backup system for any eventuality. Cab drivers are educated, polite, and reliable and are trained to handle acute breakdowns. The cab service includes all categories of cars from luxury to budget. Further, this project's utmost priority is quality. To achieve this, vehicles are well maintained and tested for delivering optimum and uninterrupted performance. Team of professionals in the travel business enables this system to design trips that suits to all budgets and preferences of the travelers. In addition, workforce including drivers and administrative staff are well trained to discharge their duties with a lot of efficiency

INTRODUCTION:

Introduction Online Cab Booking System specializing in Hiring cabs to customers. It is an online system through which customers can view available cabs; register the cabs, view profile and book cabs.Cab booking service is a major transport service provided by the various transport operators in a particular city. Mostly peoples use cab service for their daily transportations need. The company must be a registered and fulfils all the requirements and security standards set by the transport department. Online Cab Booking

System is a web based platform that allows your customers to book their taxi's and executive taxis all online from the comfort of their own home or office. The platform should offer an administration interface where the taxi company can manage the content, and access all bookings and customer information. More and more Taxi companies are looking for integrated taxi booking systems as it makes life much easier for (1) The traveler - this is highly important and in today's internet age people should be able to book taxis online without having to pick up the phone and (2) the taxi company as all their bookings are now managed via an automated

system which means they have an electronic record of future and historic bookings

AIM AND OBJECTIVES:

¬ To keep the information of Customer.

¬ To keep the information of number of bookings in current month.

¬ To keep the detail of taxis route.

¬ To keep the information of cancellation and modification of booking in current month.

¬ To maintain the record of every employee of every route.

Hardware and Software Requirements :

Hardware Requirements : CPU : Intel Pentium 4 Processor, Dual Core (MIN) – 1.2 GHz RAM : 512 MB (MIN) HDD : 80 GB (MIN) 3.2)

Software Requirements : Front End : Netbeans IDE 8.0.2 Back End : MySQL – 5.7.6 with Wampserver – 3.2.6, JDBC Connectivity. Operating System : Windows XP & Above.

3.3 Diagrams

Level 1 Admin:

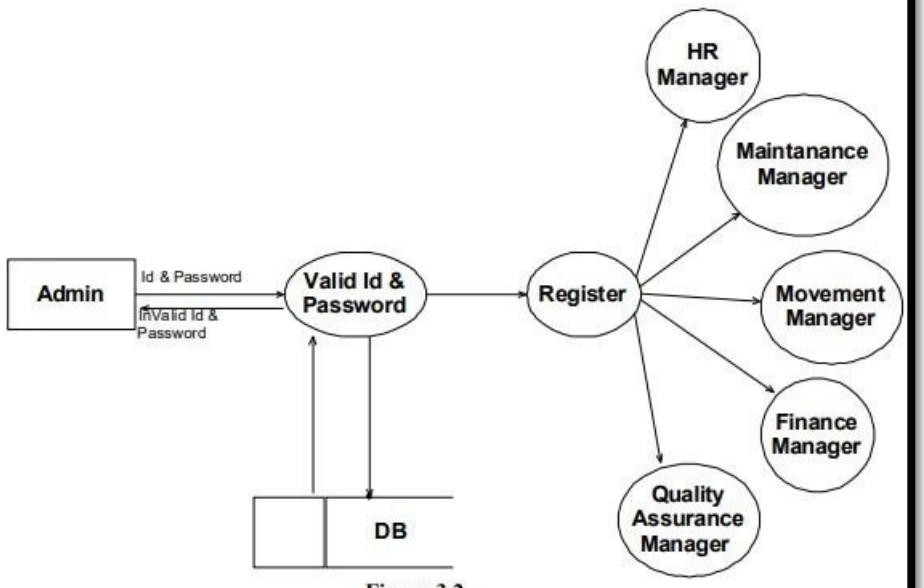


Figure 3.2

3.3.1 Use Case:

Admin :

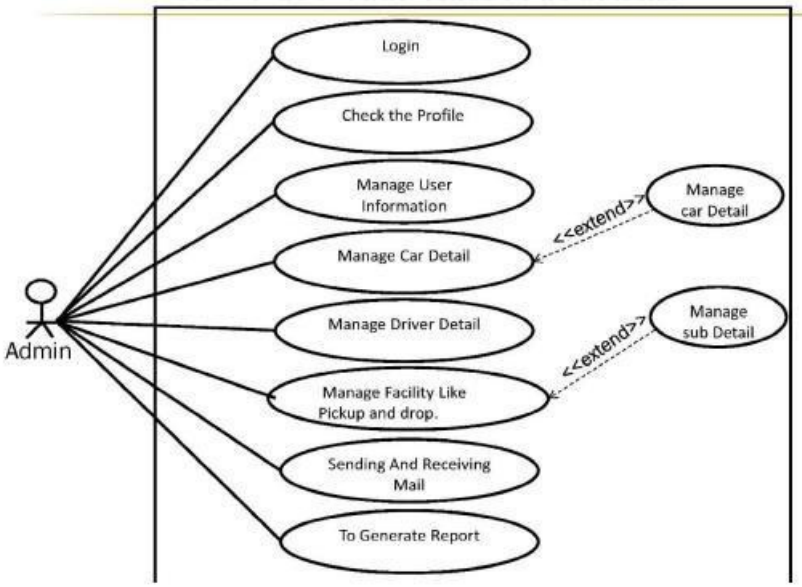


Figure 3.3

User :

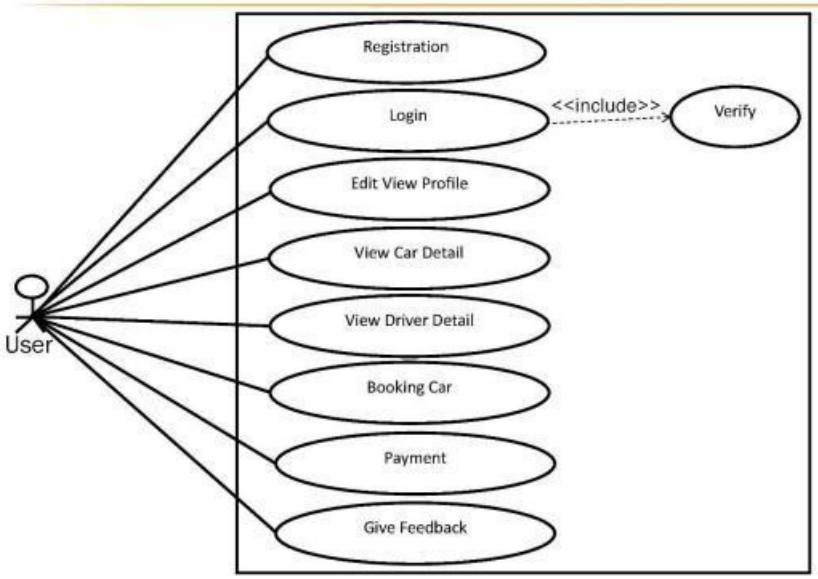
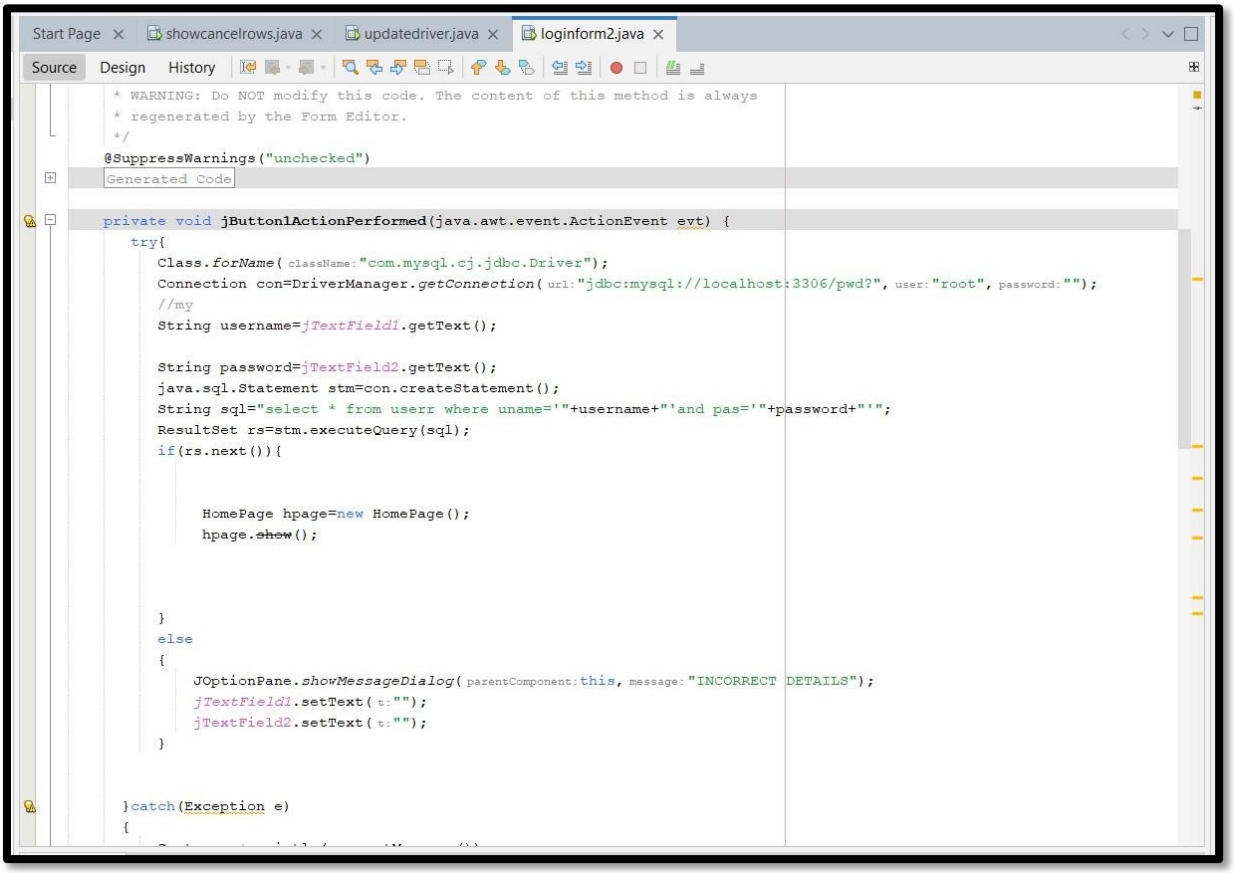


Figure 3.4

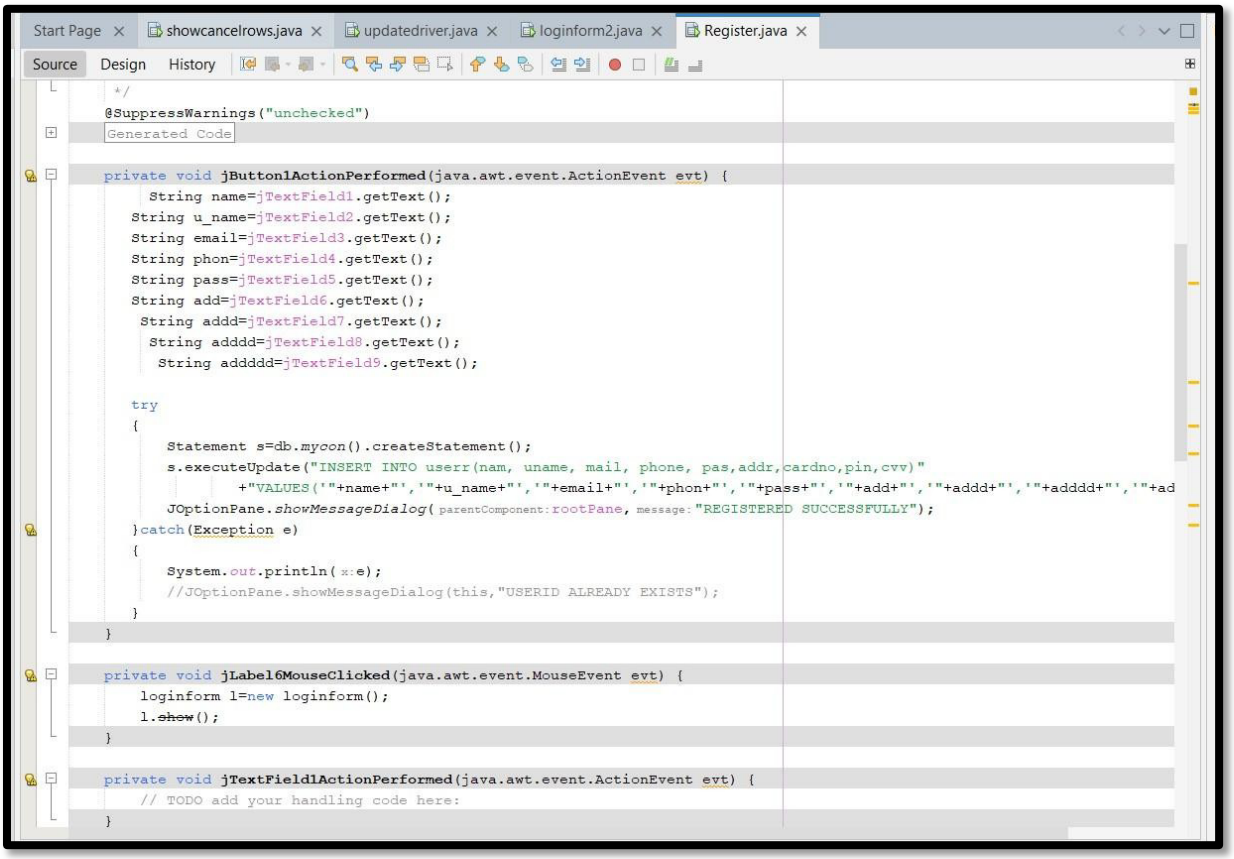
EXPERIMENTAL ANALYSIS:

CODING:-

Login page:

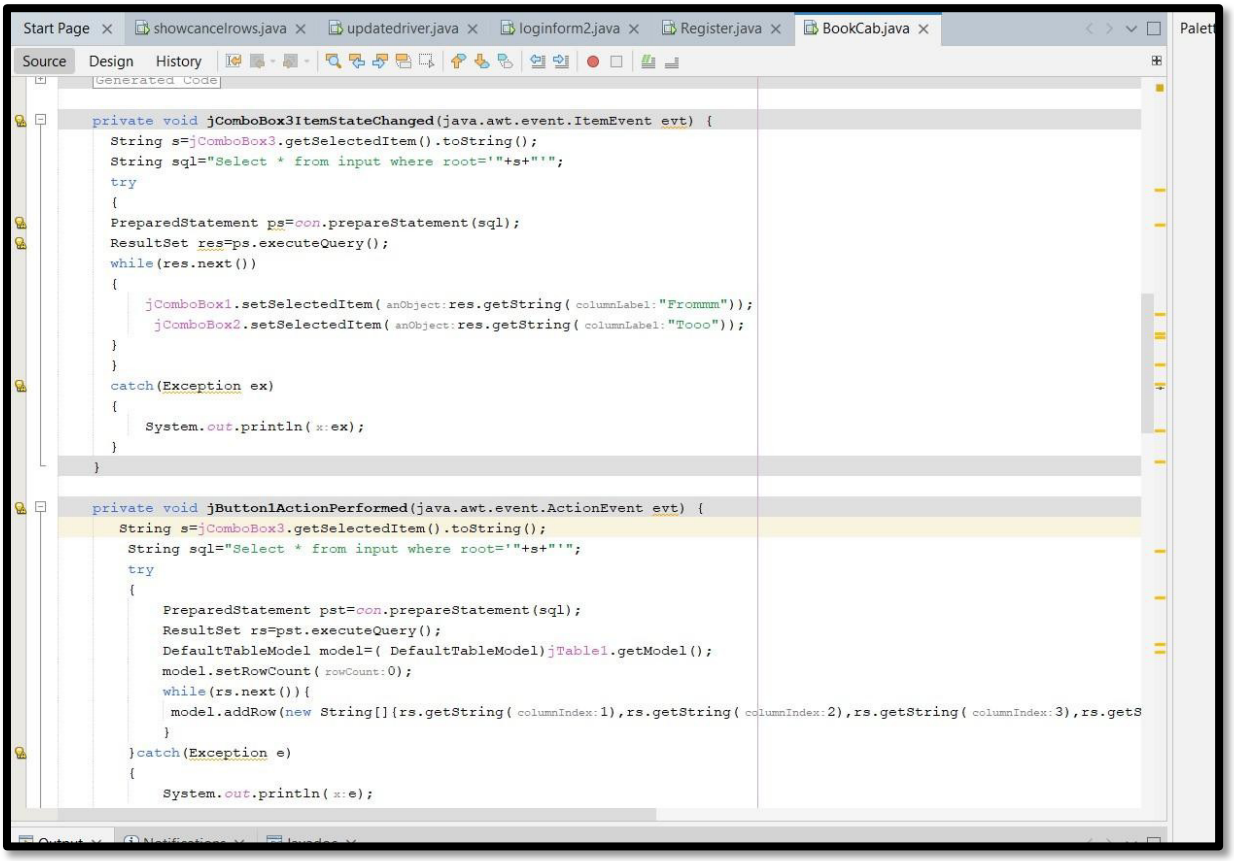


REGISTER PAGE:



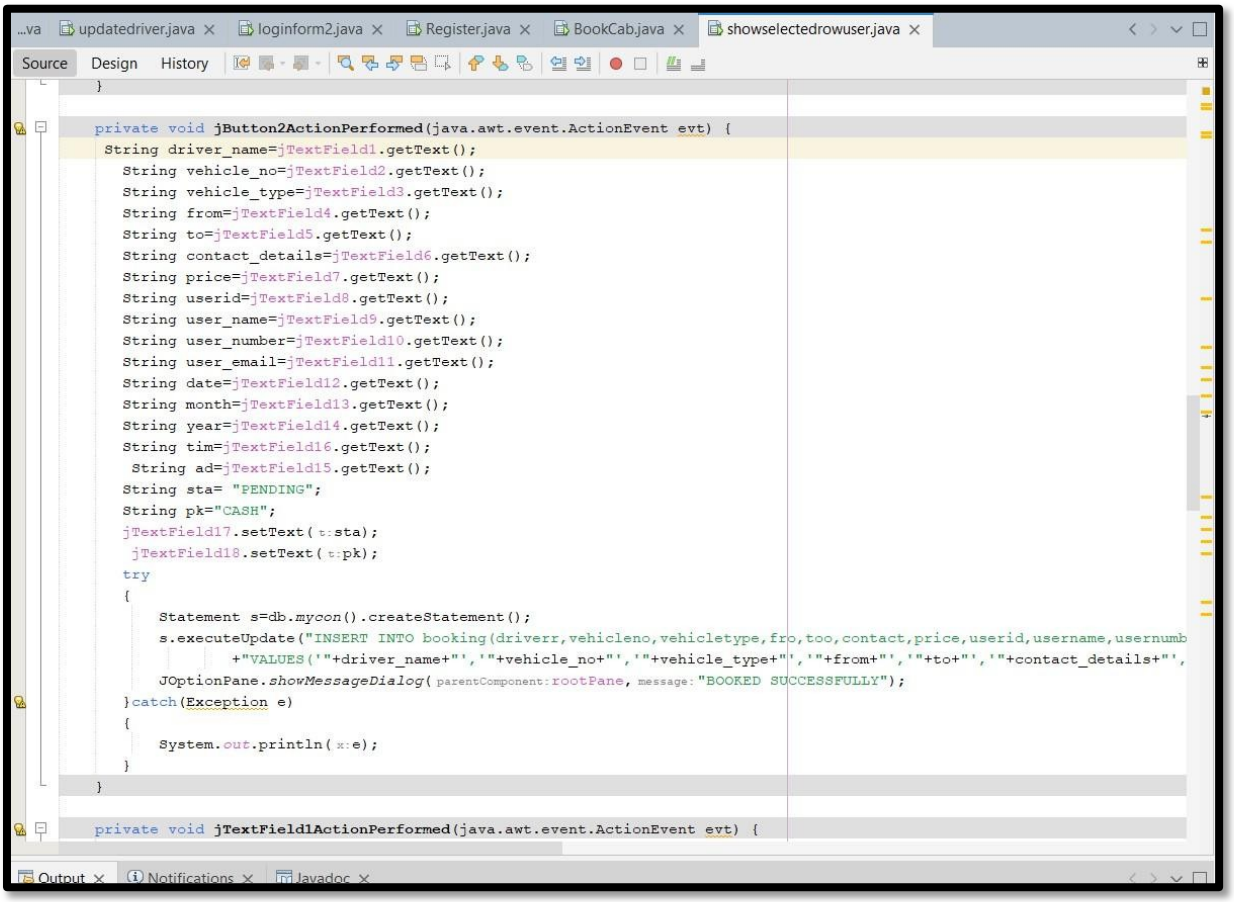
```
Start Page | showcancelrows.java | updatedriver.java | loginform2.java | Register.java
Source | Design | History
 */
@SuppressWarnings("unchecked")
Generated Code

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
     String name=jTextField1.getText();
    String u_name=jTextField2.getText();
    String email=jTextField3.getText();
    String phon=jTextField4.getText();
    String pass=jTextField5.getText();
    String add=jTextField6.getText();
     String addd=jTextField7.getText();
      String adddd=jTextField8.getText();
       String addddd=jTextField9.getText();

    try
    {
        Statement s=db.mycon().createStatement();
        s.executeUpdate("INSERT INTO userr(nam, uname, mail, phone, pas,addr,cardno,pin,cvv)"
                +"VALUES('"+name+"','"+u_name+"','"+email+"','"+phon+"','"+pass+"','"+add+"','"+addd+"','"+adddd+"','"+ad
        JOptionPane.showMessageDialog( parentComponent: rootPane, message: "REGISTERED SUCCESSFULLY");
    }catch(Exception e)
    {
        System.out.println( x: e);
        //JOptionPane.showMessageDialog(this,"USERID ALREADY EXISTS");
    }
}

private void jLabel6MouseClicked(java.awt.event.MouseEvent evt) {
    loginform l=new loginform();
    l.show();
}

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}
```

SEARCH CABS:-



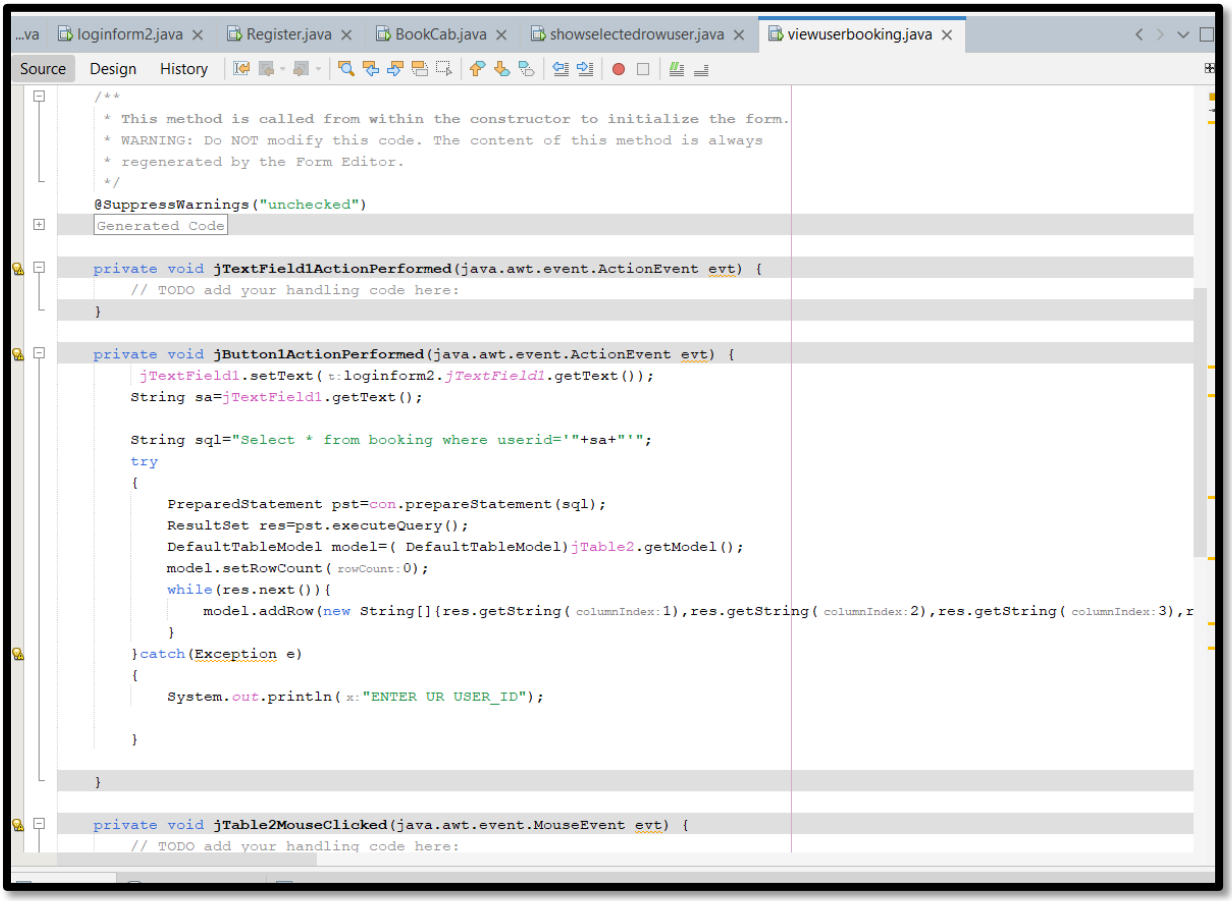
```
Start Page | showcancelrows.java | updatedriver.java | loginform2.java | Register.java | BookCab.java
Source | Design | History
Generated Code

private void jComboBox3ItemStateChanged(java.awt.event.ItemEvent evt) {
  String s=jComboBox3.getSelectedItem().toString();
  String sql="Select * from input where root='"+s+"'";
  try
  {
  PreparedStatement ps=con.prepareStatement(sql);
  ResultSet res=ps.executeQuery();
  while(res.next())
  {
      jComboBox1.setSelectedItem( anObject: res.getString( columnLabel: "Fromm"));
       jComboBox2.setSelectedItem( anObject: res.getString( columnLabel: "Tooo"));
  }
  }
  catch(Exception ex)
  {
      System.out.println( x: ex);
  }
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
   String s=jComboBox3.getSelectedItem().toString();
    String sql="Select * from input where root='"+s+"'";
    try
    {
        PreparedStatement pst=con.prepareStatement(sql);
        ResultSet rs=pst.executeQuery();
        DefaultTableModel model=( DefaultTableModel)jTable1.getModel();
        model.setRowCount( rowCount: 0);
        while(rs.next()){
         model.addRow(new String[]{rs.getString( columnIndex: 1),rs.getString( columnIndex: 2),rs.getString( columnIndex: 3),rs.getS
        }
    }catch(Exception e)
    {
        System.out.println( x: e);
```

BOOK CAB:

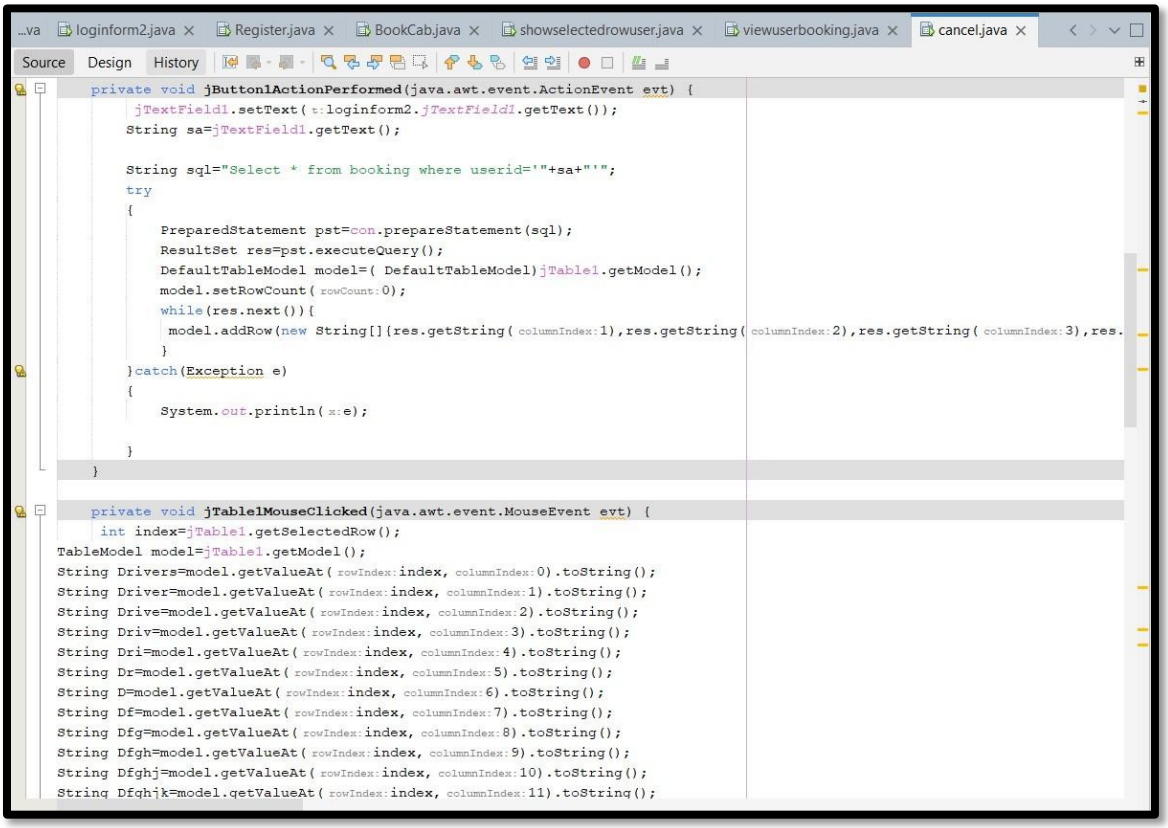


```
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
     String driver_name=jTextField1.getText();
       String vehicle_no=jTextField2.getText();
       String vehicle_type=jTextField3.getText();
       String from=jTextField4.getText();
       String to=jTextField5.getText();
       String contact_details=jTextField6.getText();
       String price=jTextField7.getText();
       String userid=jTextField8.getText();
       String user_name=jTextField9.getText();
       String user_number=jTextField10.getText();
       String user_email=jTextField11.getText();
       String date=jTextField12.getText();
       String month=jTextField13.getText();
       String year=jTextField14.getText();
       String tim=jTextField16.getText();
        String ad=jTextField15.getText();
       String sta= "PENDING";
       String pk="CASH";
       jTextField17.setText( t:sta);
        jTextField18.setText( t:pk);
       try
       {
           Statement s=db.myoon().createStatement();
           s.executeUpdate("INSERT INTO booking(driverr,vehicleno,vehicletype,fro,too,contact,price,userid,username,usernumb
                   +"VALUES('"+driver_name+"','"+vehicle_no+"','"+vehicle_type+"','"+from+"','"+to+"','"+contact_details+"',
           JOptionPane.showMessageDialog( parentComponent: rootPane, message: "BOOKED SUCCESSFULLY");
       }catch(Exception e)
       {
           System.out.println( x:e);
       }
    }

    private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
```

VIEW BOOKING



CANCEL CAB :-

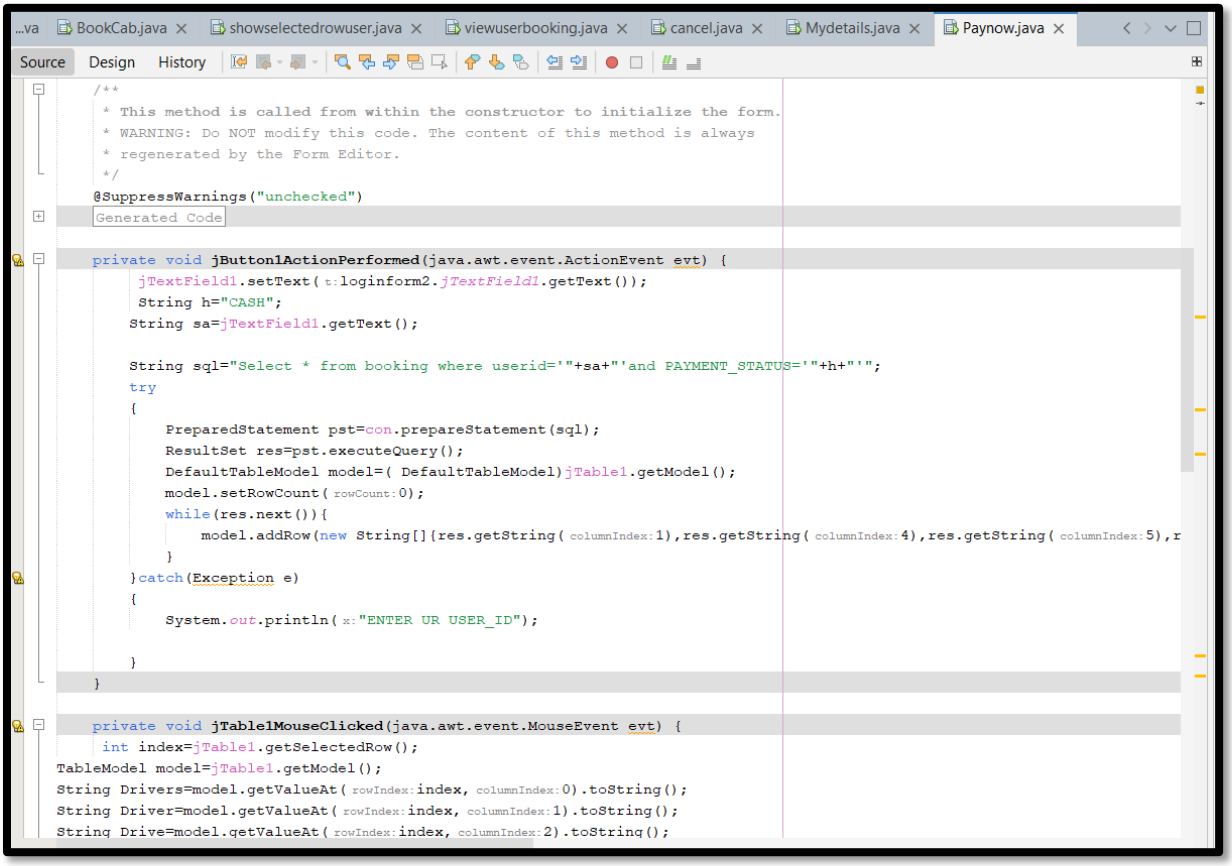


```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    jTextField1.setText( t: loginform2.jTextField1.getText());
    String sa=jTextField1.getText();

    String sql="Select * from booking where userid='"+sa+"'";
    try
    {
        PreparedStatement pst=con.prepareStatement(sql);
        ResultSet res=pst.executeQuery();
        DefaultTableModel model=( DefaultTableModel)jTable1.getModel();
        model.setRowCount( rowCount: 0);
        while(res.next()){
         model.addRow(new String[]{res.getString( columnIndex: 1),res.getString( columnIndex: 2),res.getString( columnIndex: 3),res.
        }
    }catch(Exception e)
    {
        System.out.println( x: e);

    }
}

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
    int index=jTable1.getSelectedRow();
TableModel model=jTable1.getModel();
String Drivers=model.getValueAt( rowIndex: index, columnIndex: 0).toString();
String Driver=model.getValueAt( rowIndex: index, columnIndex: 1).toString();
String Drive=model.getValueAt( rowIndex: index, columnIndex: 2).toString();
String Driv=model.getValueAt( rowIndex: index, columnIndex: 3).toString();
String Dri=model.getValueAt( rowIndex: index, columnIndex: 4).toString();
String Dr=model.getValueAt( rowIndex: index, columnIndex: 5).toString();
String D=model.getValueAt( rowIndex: index, columnIndex: 6).toString();
String Df=model.getValueAt( rowIndex: index, columnIndex: 7).toString();
String Dfg=model.getValueAt( rowIndex: index, columnIndex: 8).toString();
String Dfgh=model.getValueAt( rowIndex: index, columnIndex: 9).toString();
String Dfghj=model.getValueAt( rowIndex: index, columnIndex: 10).toString();
String Dfghjk=model.getValueAt( rowIndex: index, columnIndex: 11).toString();
```

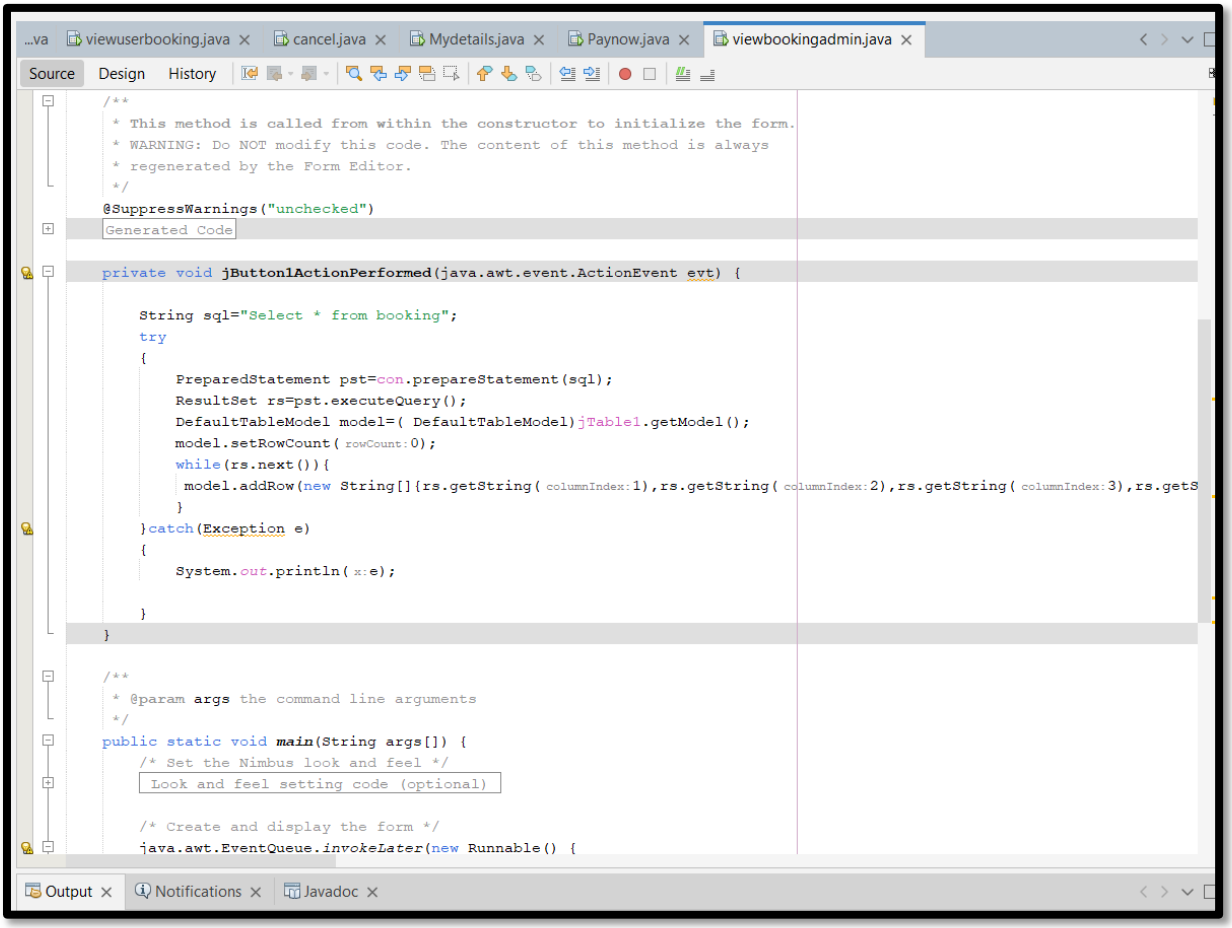
MY DETAILS:-

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

    jTextField2.setText( t: loginform2.jTextField1.getText());
    String s=jTextField2.getText();
    String sql="Select * from userr where uname='"+s+"'";
    try
    {
        PreparedStatement pst=con.prepareStatement(sql);
        ResultSet rs=pst.executeQuery();

        if(rs.next()){
        jTextField1.setText( t: rs.getString( columnIndex: 1));
       // jTextField2.setText(rs.getString(2));
        jTextField3.setText( t: rs.getString( columnIndex: 3));
        jTextField4.setText( t: rs.getString( columnIndex: 4));
        jTextField5.setText( t: rs.getString( columnIndex: 6));
        }
        else
        {
            JOptionPane.showMessageDialog( parentComponent: this, message: "ENTER UR USER_ID TO VIEW YOUR DETAILS");
        }

    }catch(Exception e)
    {
        JOptionPane.showMessageDialog( parentComponent: this, message: "USER NOT FOUNDsdghfyg");

    }
}
```

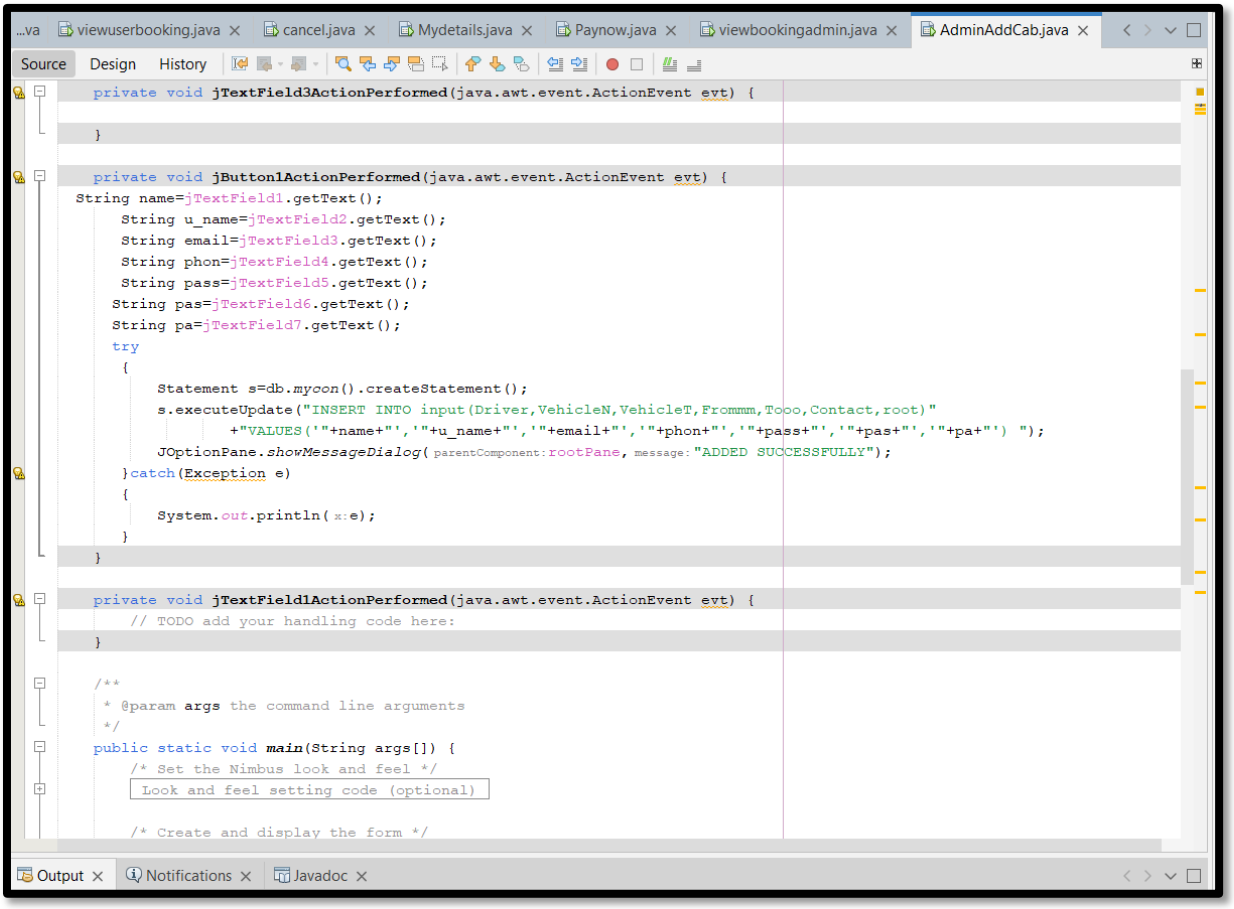
PAYMENT PAGE:-



VIEW ADMIN BOOKING PAGE:

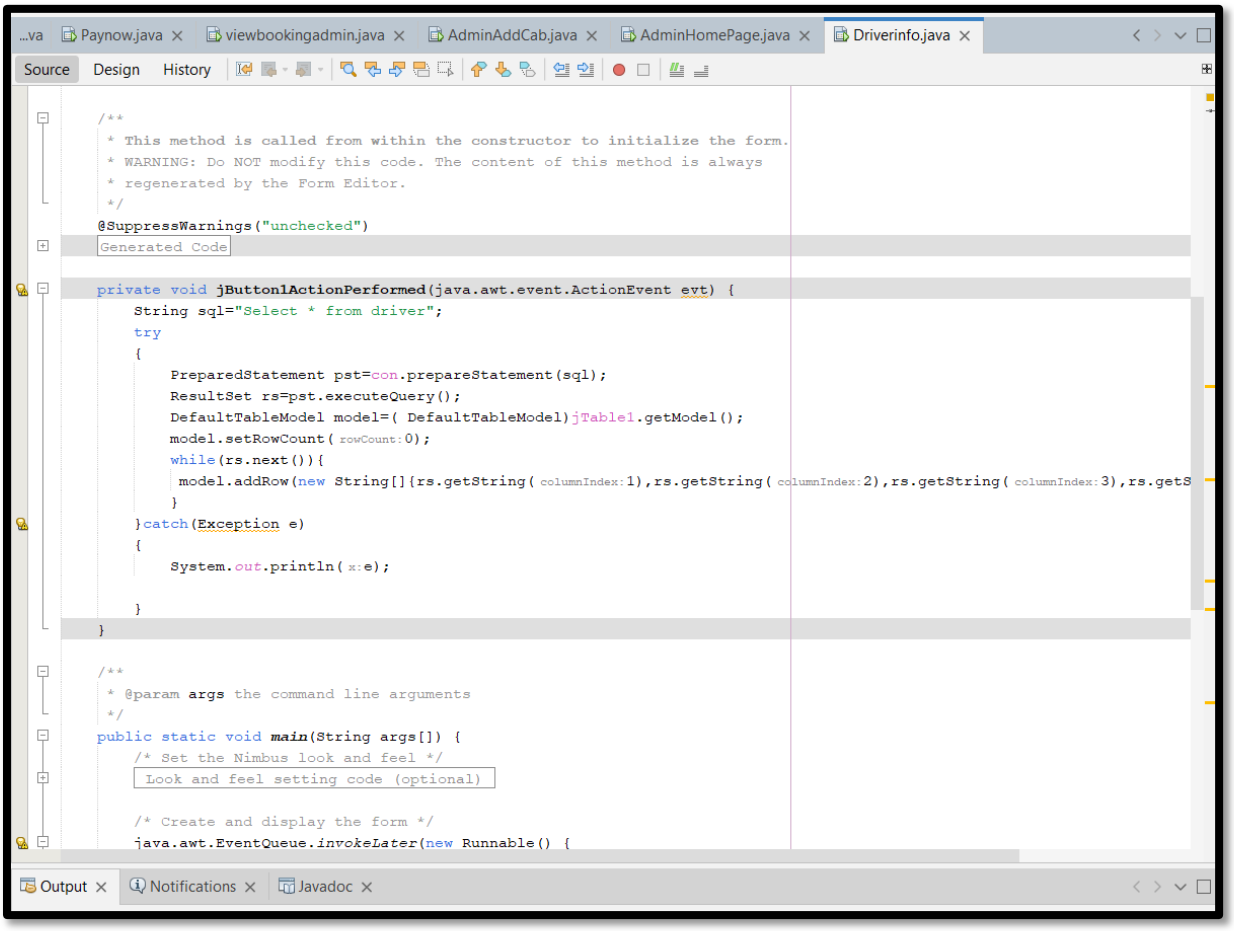


ADMIN ADD CAB PAGE:



```
    private void jTextField3ActionPerformed(java.awt.event.ActionEvent evt) {

    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
String name=jTextField1.getText();
     String u_name=jTextField2.getText();
     String email=jTextField3.getText();
     String phon=jTextField4.getText();
     String pass=jTextField5.getText();
    String pas=jTextField6.getText();
    String pa=jTextField7.getText();
    try
     {
         Statement s=db.mycon().createStatement();
         s.executeUpdate("INSERT INTO input(Driver,VehicleN,VehicleT,Frommm,Tooo,Contact,root)"
                 +"VALUES('"+name+"','"+u_name+"','"+email+"','"+phon+"','"+pass+"','"+pas+"','"+pa+"') ");
         JOptionPane.showMessageDialog( parentComponent: rootPane, message: "ADDED SUCCESSFULLY");
     }catch(Exception e)
     {
         System.out.println( x: e);
     }
    }

    private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)

        /* Create and display the form */
```

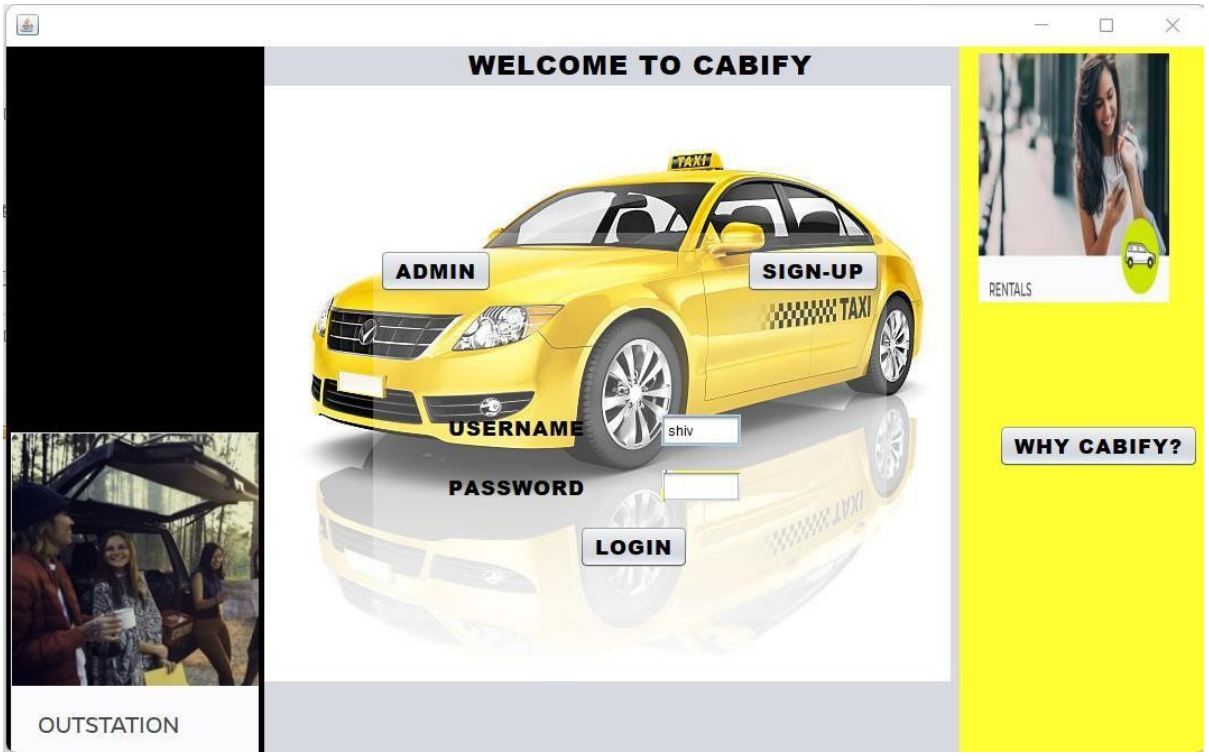
ADMIN DRIVER INFO PAGE:



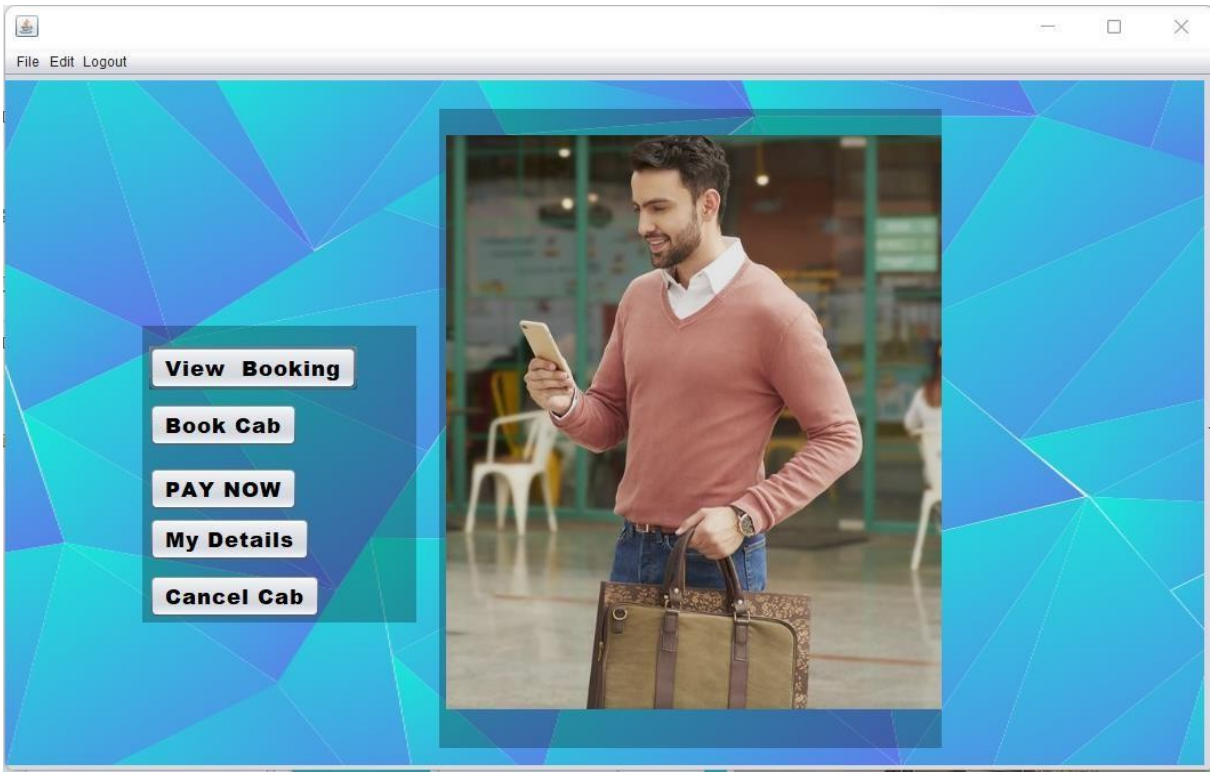
DRIVER VIEW BOOKING:

```
...va  AdminAddCab.java  AdminHomePage.java  Driverinfo.java  DriverHome.java  Driverviewbook.java
Source  Design  History
    Generated Code

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        jTextField1.setText( t: Driverlogin.jTextField1.getText());
        String sa=jTextField1.getText();
         String sql="Select * from booking where driverr='"+sa+"'";
        try
        {
            PreparedStatement pst=con.prepareStatement(sql);
            ResultSet res=pst.executeQuery();
            DefaultTableModel model=( DefaultTableModel)jTable1.getModel();
            model.setRowCount( rowCount: 0);
            while(res.next()){
                model.addRow(new String[]{res.getString( columnIndex: 1),res.getString( columnIndex: 2),res.getString( columnIndex: 3),r
            }
        }catch(Exception e)
        {
            System.out.println( x: "ENTER UR USER_ID");

        }
    }

    private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
        int index=jTable1.getSelectedRow();
TableModel model=jTable1.getModel();
String Drivers=model.getValueAt( rowIndex: index, columnIndex: 0).toString();
String Driver=model.getValueAt( rowIndex: index, columnIndex: 1).toString();
String Drive=model.getValueAt( rowIndex: index, columnIndex: 2).toString();
String Driv=model.getValueAt( rowIndex: index, columnIndex: 3).toString();
String Dri=model.getValueAt( rowIndex: index, columnIndex: 4).toString();
String Dr=model.getValueAt( rowIndex: index, columnIndex: 5).toString();
String D=model.getValueAt( rowIndex: index, columnIndex: 6).toString();
String Df=model.getValueAt( rowIndex: index, columnIndex: 7).toString();
String Dfg=model.getValueAt( rowIndex: index, columnIndex: 8).toString();
String Dfgh=model.getValueAt( rowIndex: index, columnIndex: 9).toString();
String Dfghj=model.getValueAt( rowIndex: index, columnIndex: 10).toString();
```

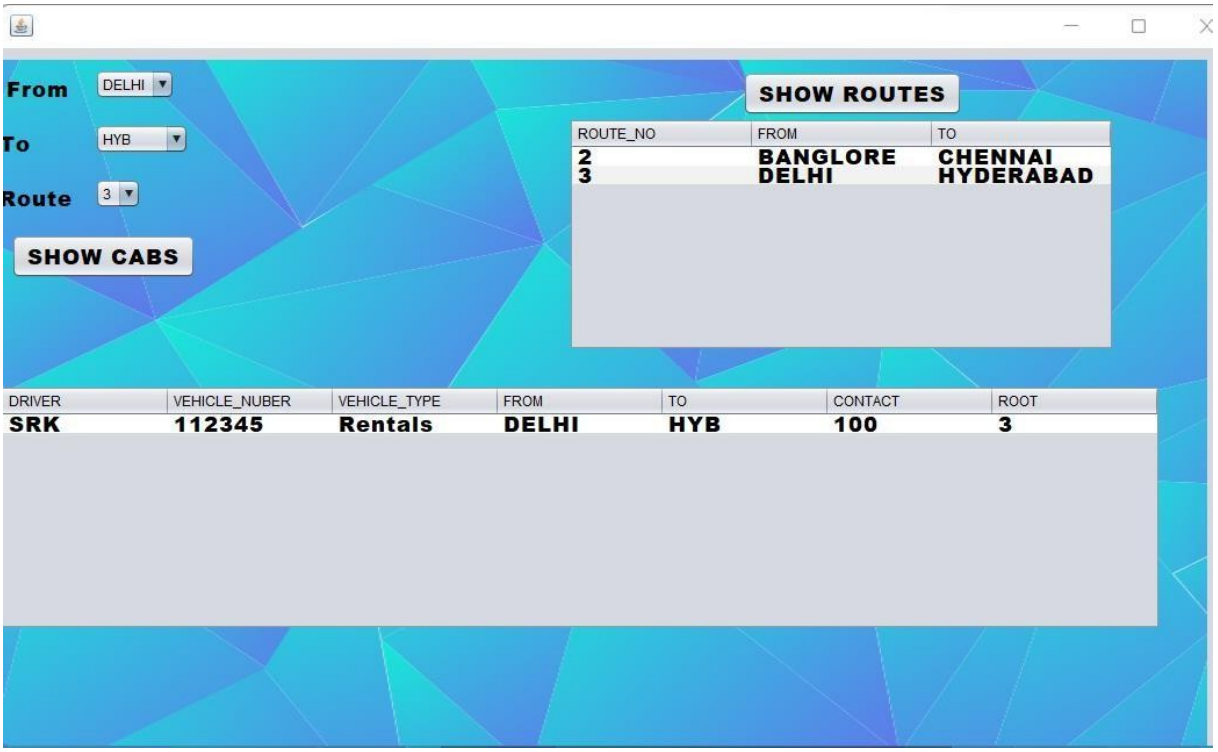
LOGIN PAGE SCREENSHOT:



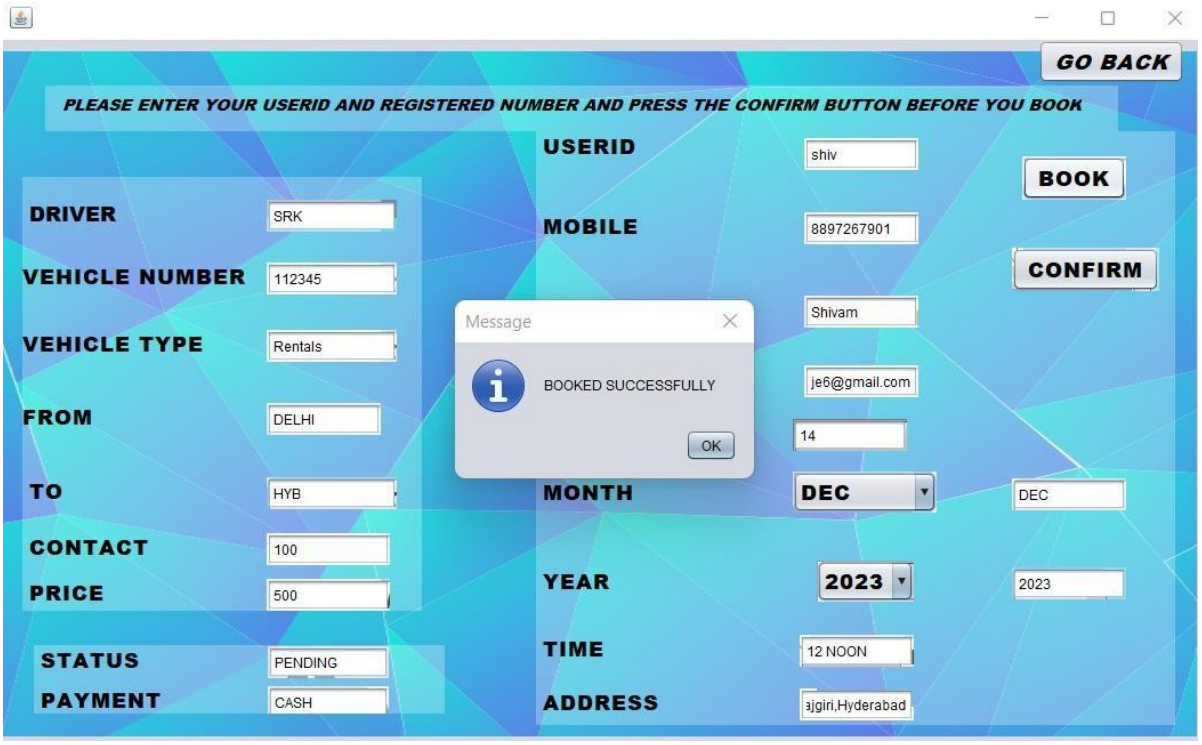
HOME PAGE SCREEN SHOT:



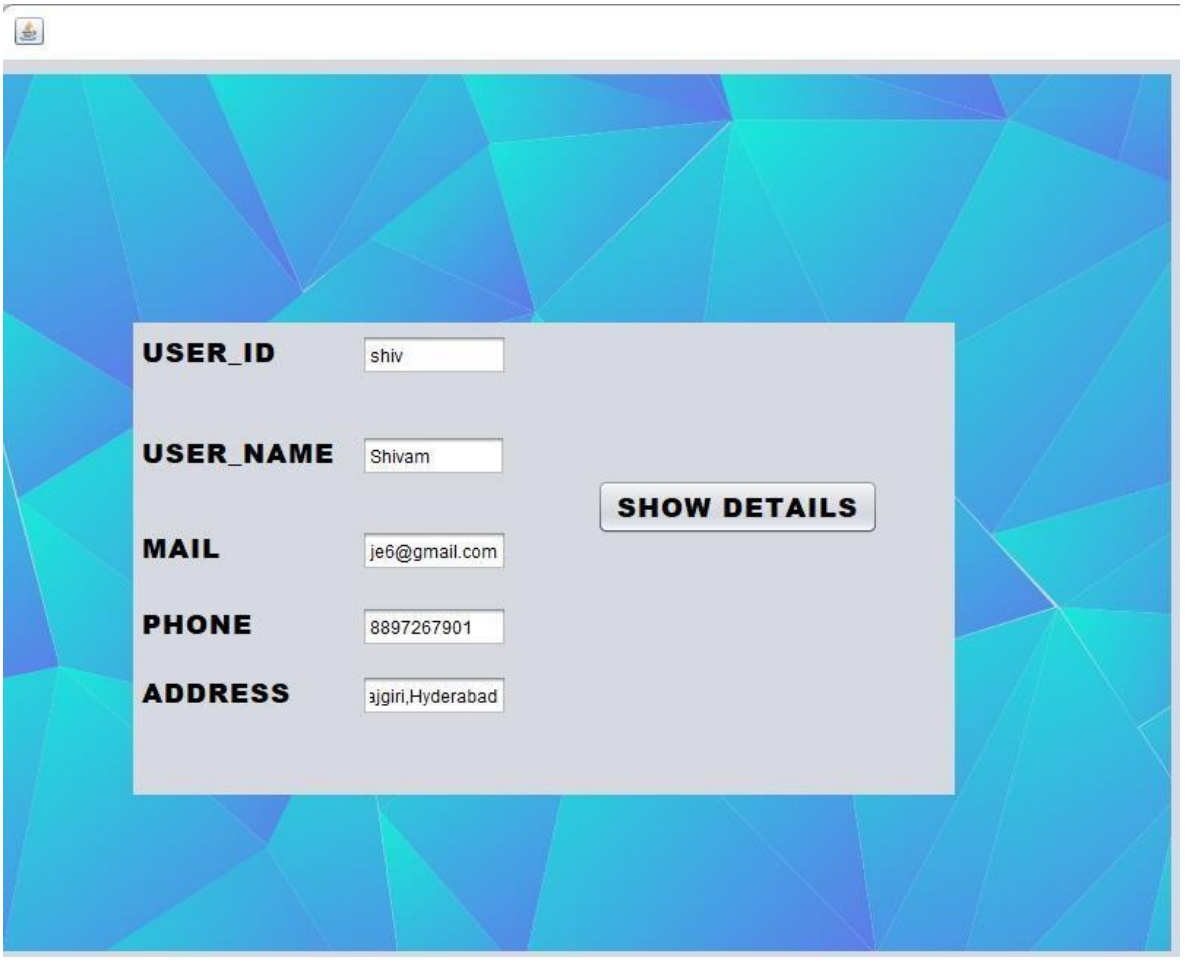
FIND CAB SCREEN SHOT:



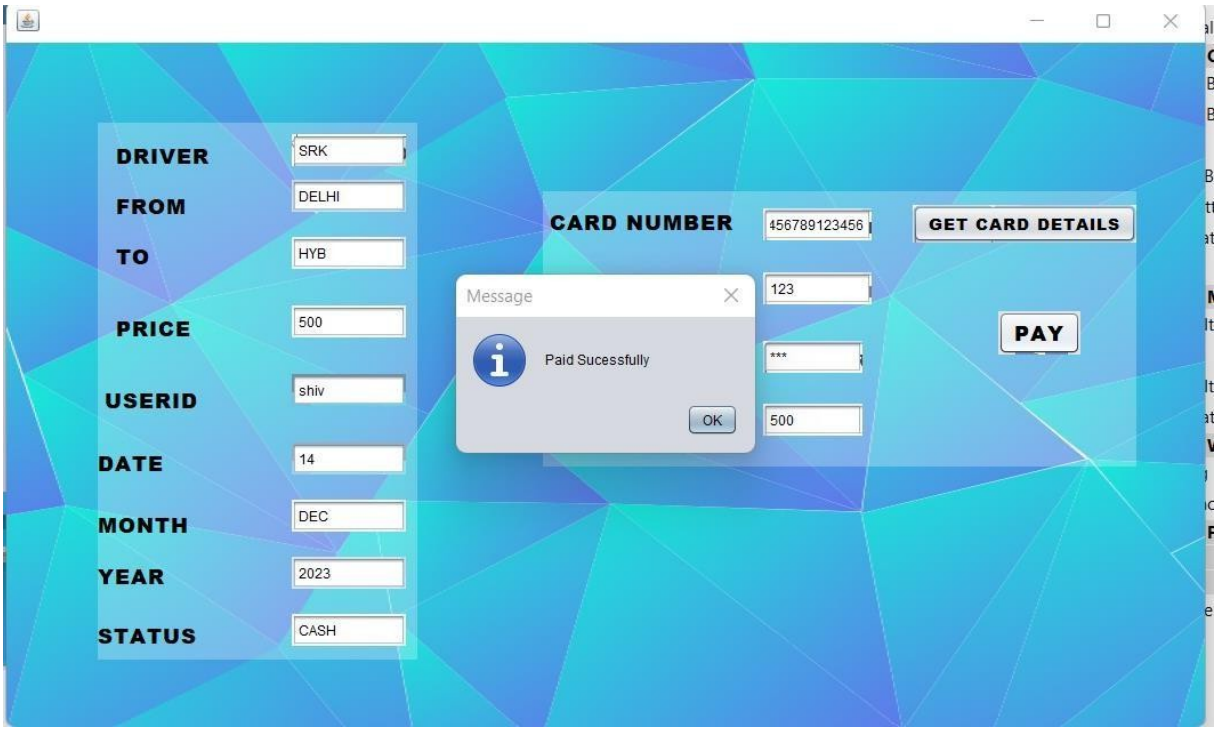
BOOK CAB SCREEN SHOT:



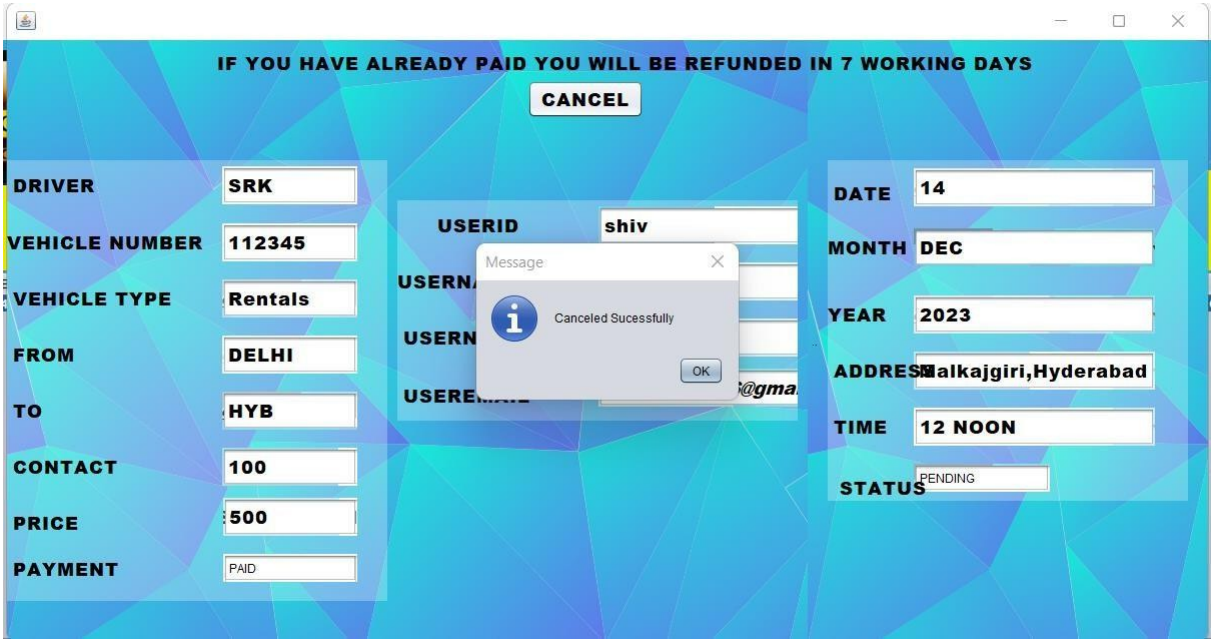
MY DETAILS SCREEN SHOT:



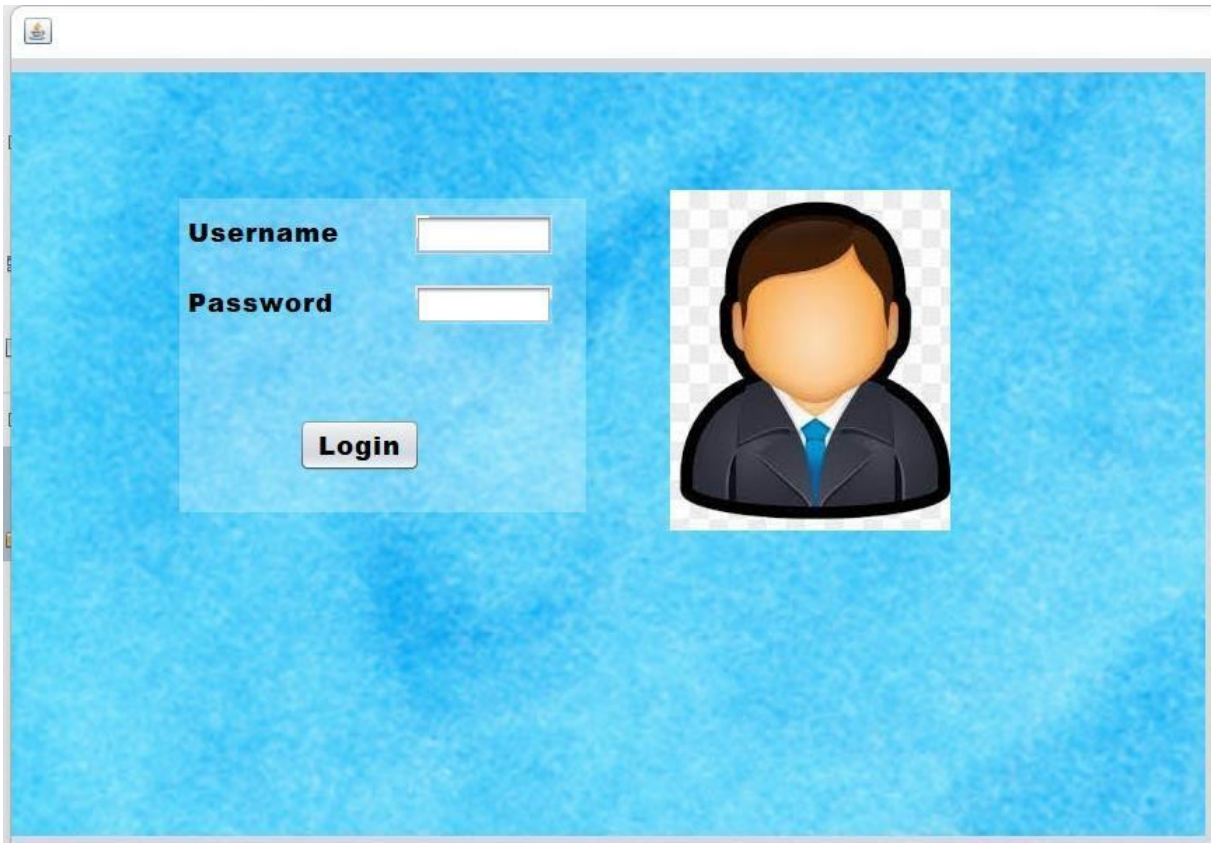
PAYMENT SCREEN SHOT:



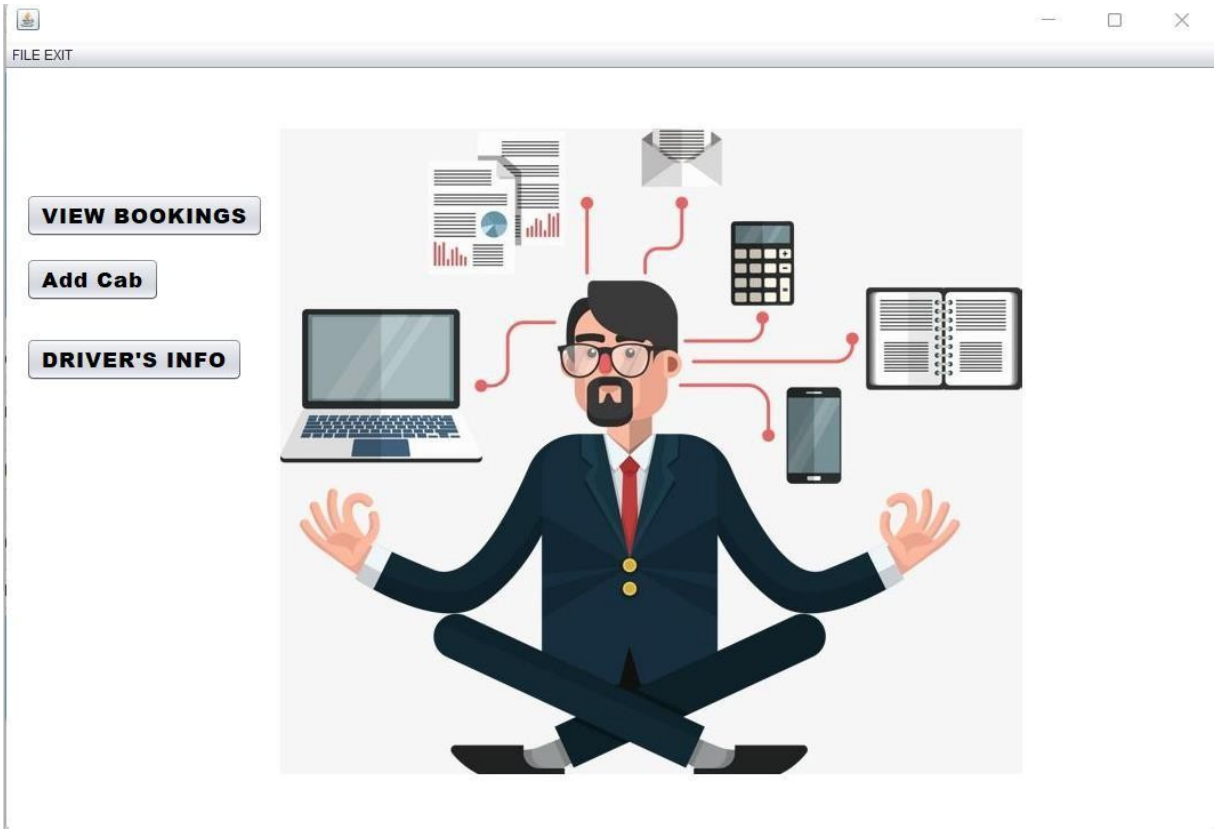
CANCEL CAB SCREEN SHOT:



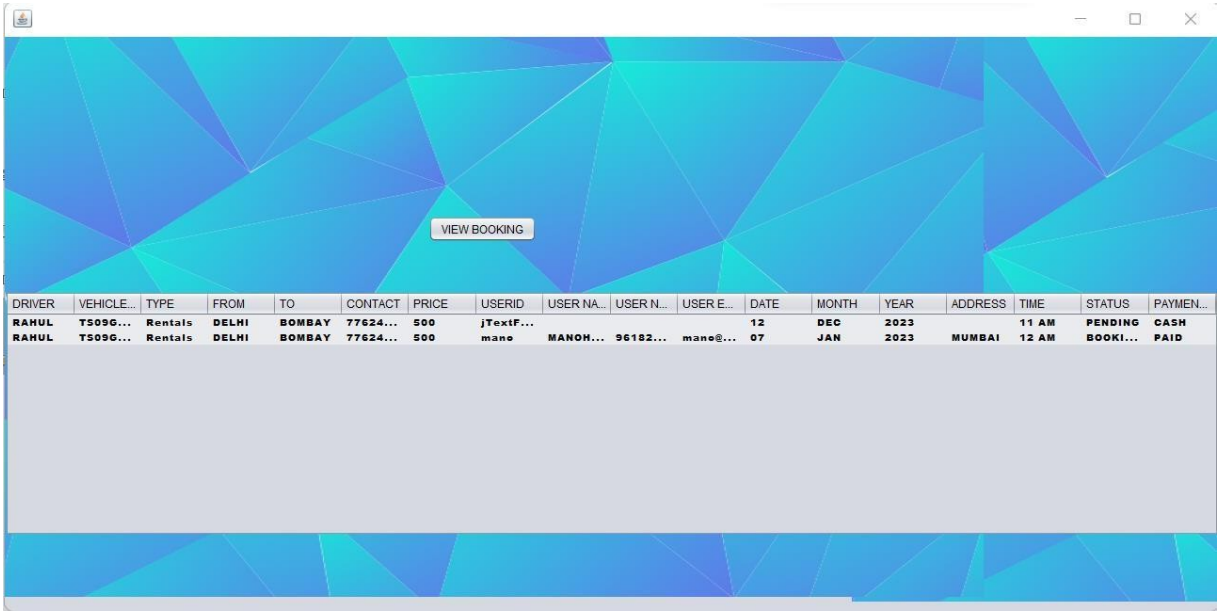
ADMIN LOGIN PAGE:



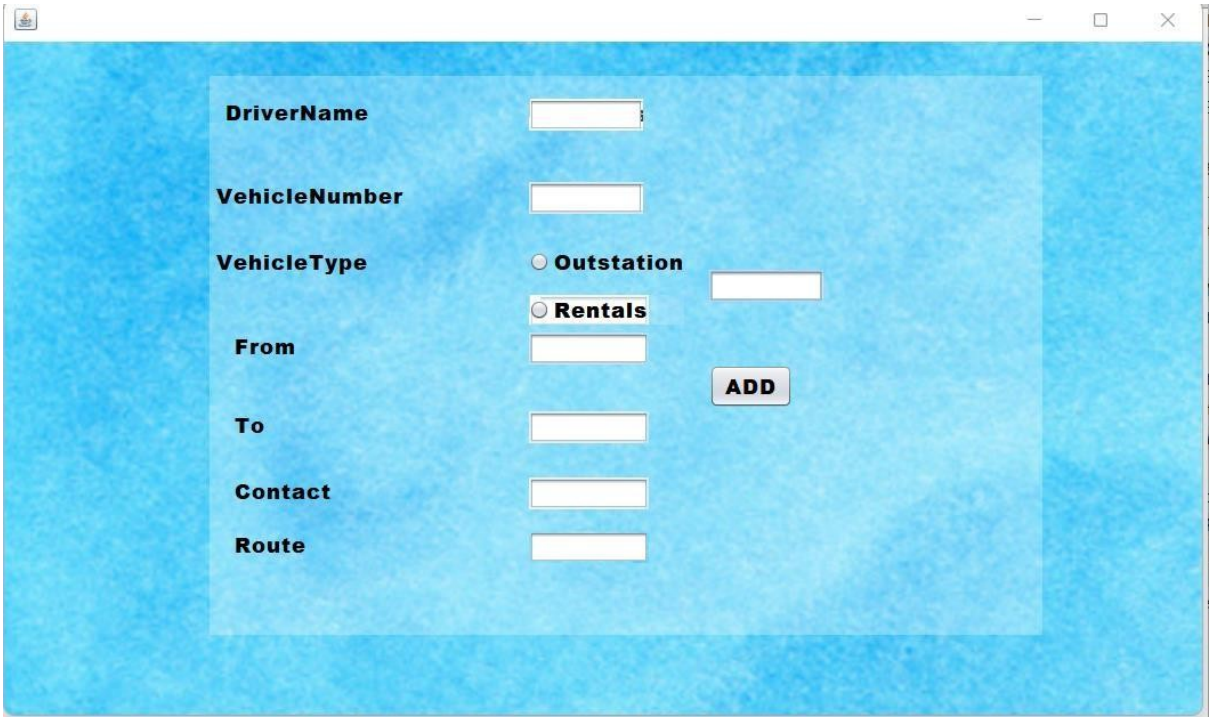
ADMIN HOME PAGE:



ADMIN VIEW BOOKING SCREEN SHOT:

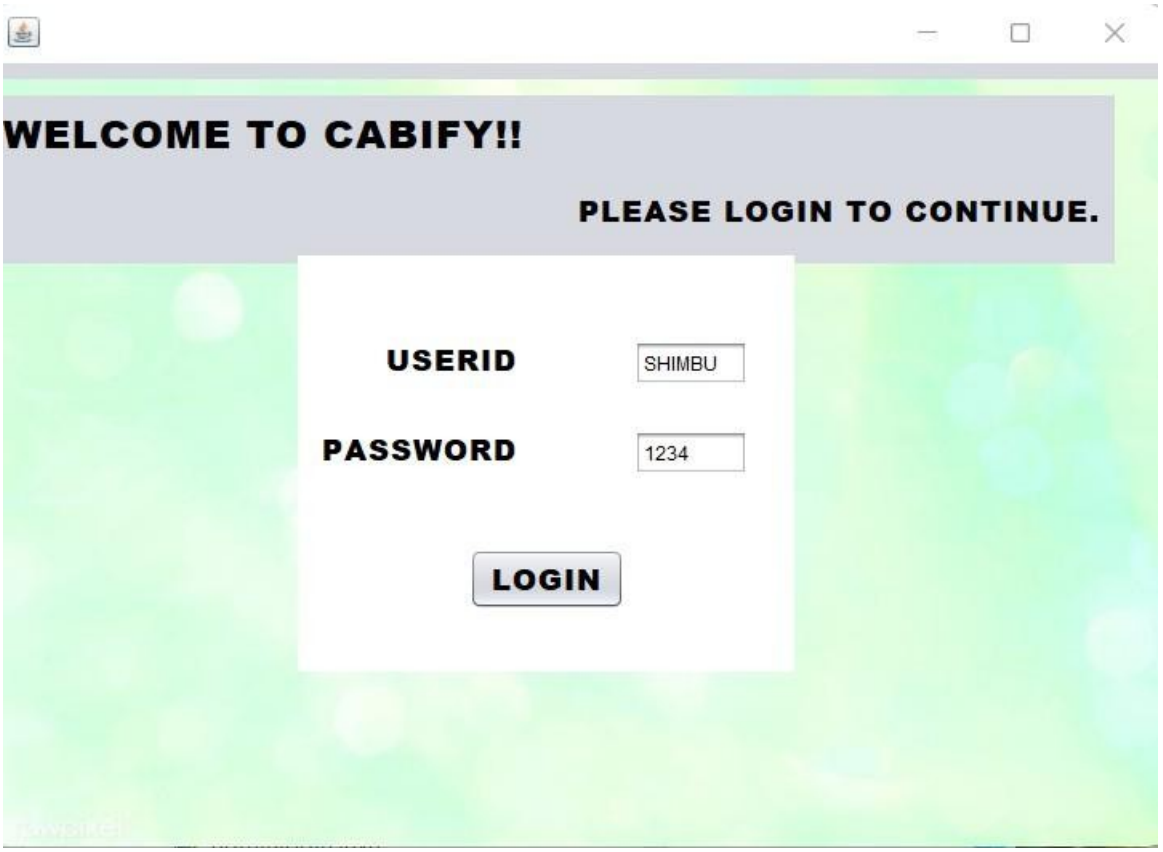


ADMIN ADD CAB SCREEN SHOT:



ADMIN VIEW DRIVER SCREEN SHOT:

DRIVER LOGIN:



DRIVER VIEW BOOKING:

RESULT AND DISCUSSION:-

Today the need of simplicity has driven application software programming to a new level. This project is a transaction related information storing project which can be used by the various users for online cab booking through internet.This project “Online Cab Booking System” has a wide scope as it is generalised software and can be easily used in any Cab booking process system with little or no change. The changes in software can be easily accommodated. The addition and deletion of the modules in software can be easily adjusted. This project has a lot of scope for further enhancement too. This project can save money and efforts in managing the records and just a mouse click can make the task easy and faster.

Future scope of the project Every Edition of a book comes with new topics and modifications if any errors are present. In the similar way, in near future, our application will overcome the flaws if occurred, and attains new features offered to employees for the Flexible and easy Transportation. Following are the Enhancements to the application.

¬ Providing Good User Interface.

¬ Providing access permissions to the employees

¬ Try to Implement the GPS system in the Cabs.

CONCLUSIONS

Information Technology plays a vital role not only in a particular field, it provides various kinds of solutions and services to the various problems prevailing in many fields. Cabs exploits information technology at the maximum extent. It uses the information technology in an efficient way for providing better passenger services. The online booking system helps to solve the everyday problems of the world biggest Indian Limitations: Cool Cab Services is a Web application and it is restricted to only limited type of users. In this application, Different types of managers have been

given access rights and they are restricted up to their functionalities, so that the data is maintained securely and redundant data is prevented. As the Data is stored electronically, it is necessary to have a Computer and Network connection to access the Application. Here The Details of Employees and Drivers, cabs are maintained but accounts to these people are not created. using this application manger do assign or update the batch, shift of cabs to drivers and employees. But employees are unable to view their details.