# JAVA PROJECT

## *Smart City Guide*

EARN ∘ SERVE ∘ EXCE
RAJAGIRI

Submitted By : Merin Jai

# **TOPIC** : SMART CITY GUIDE

## AIM

The aim of the Smart City Guide project is to develop a user-friendly software application that serves as a comprehensive guide for residents and visitors of a city. The project encompasses both administrative functionalities for managing city information and customer functionalities for accessing relevant information about various aspects of the city. The system facilitates secure user authentication and authorization mechanisms, allowing administrators and customers to access the system with appropriate privileges.

Users of the system :

1. ADMIN
2. CUSTOMERS

### ADMIN FUNCTIONALITIES

- **Addition of Places**: Administrators can add new places to the system, along with their descriptions.
- **Management of Entities**: Administrators can manage various entities within the city, such as hotels, restaurants, and tourist spots.
- **Adding Specific Spots**: Administrators can add specific spots within the city, providing details such as name, address, rating, and operating hours.

### KEY FEATURES OF THE PROJECT

- ➢ Centralized Information Management

  The project aims to provide a centralized platform for managing information related to different aspects of the city, such as places of interest, entities (hotels, restaurants, tourist spots, etc.), and specific spots within those areas.

- User Authentication and Authorization

  The system facilitates secure user authentication and authorization mechanisms, allowing administrators and customers to access the system with appropriate privileges.

- User-Friendly Interface

  The project prioritizes the development of a user-friendly graphical user interface (GUI) using Java Swing, ensuring ease of navigation and interaction for both administrators and customers.

- Database Integration

  The project integrates with a MySQL database to store and retrieve information efficiently. It utilizes JDBC (Java Database Connectivity) for database connectivity and SQL queries for data manipulation.

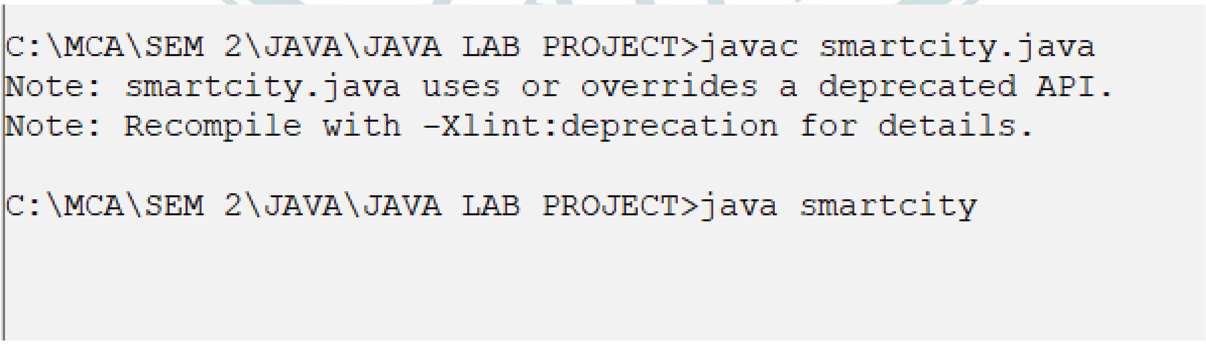
- Dynamic Updates

  The system allows for dynamic updates to the database, enabling real-time addition and modification of city information by administrators.

- Scalability and Extensibility

  The project is designed to be scalable and extensible, allowing for the inclusion of additional features and functionalities in the future as per the evolving needs of the city and its users.

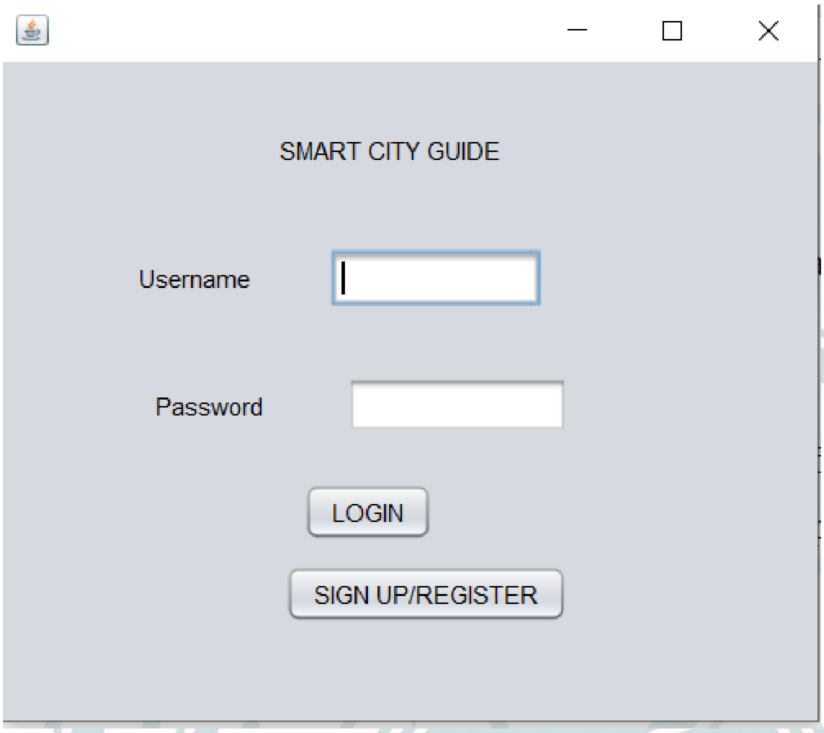
## WORKING OF PROJECT



```
C:\MCA\SEM 2\JAVA\JAVA LAB PROJECT>javac smartcity.java
Note: smartcity.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\MCA\SEM 2\JAVA\JAVA LAB PROJECT>java smartcity
```

## LOGIN PAGE

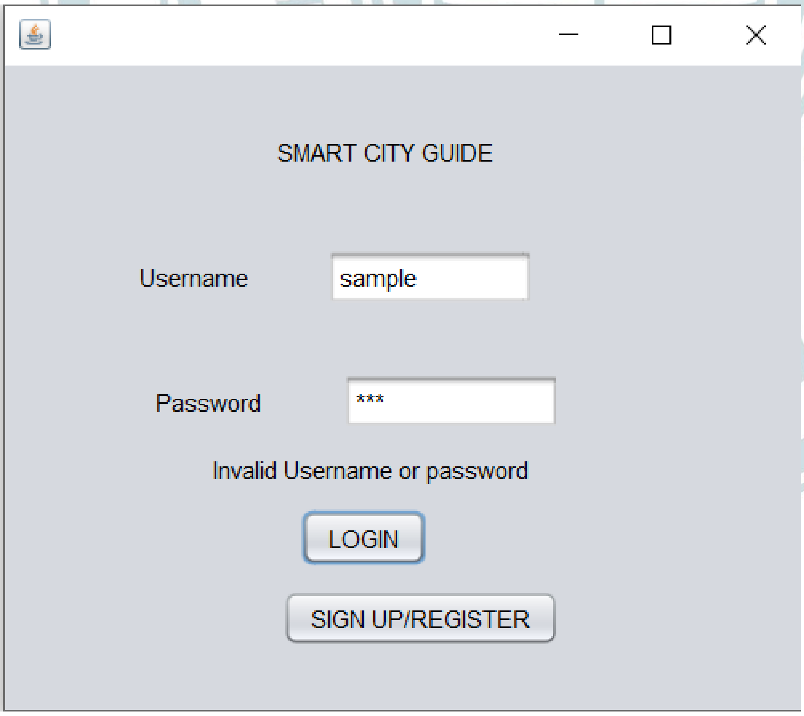


SMART CITY GUIDE

Username

Password

## LOGIN PAGE WITH VALIDATIONS



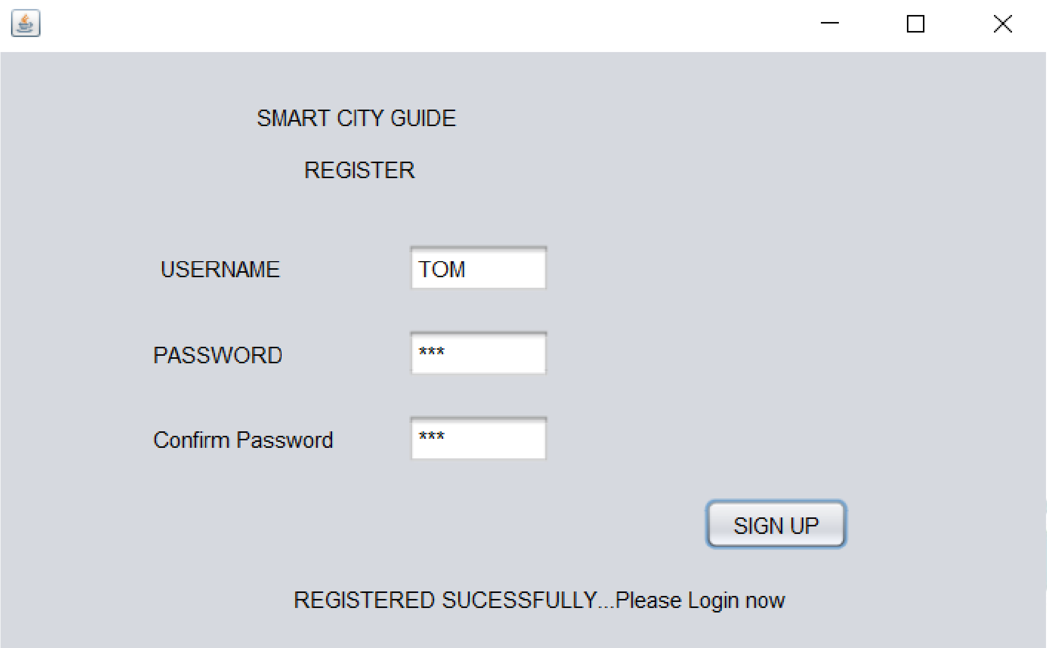
SMART CITY GUIDE

Username sample

Password ***

Invalid Username or password

## REGISTER/SIGN UP PAGE



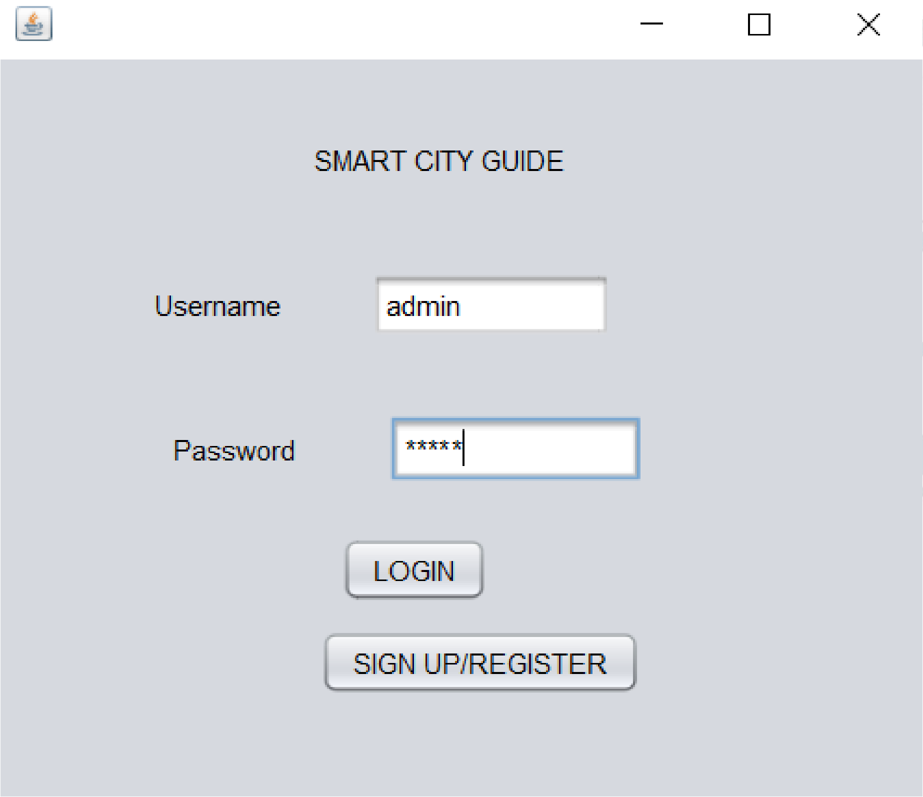
USERNAME TOM

PASSWORD ***

Confirm Password ***

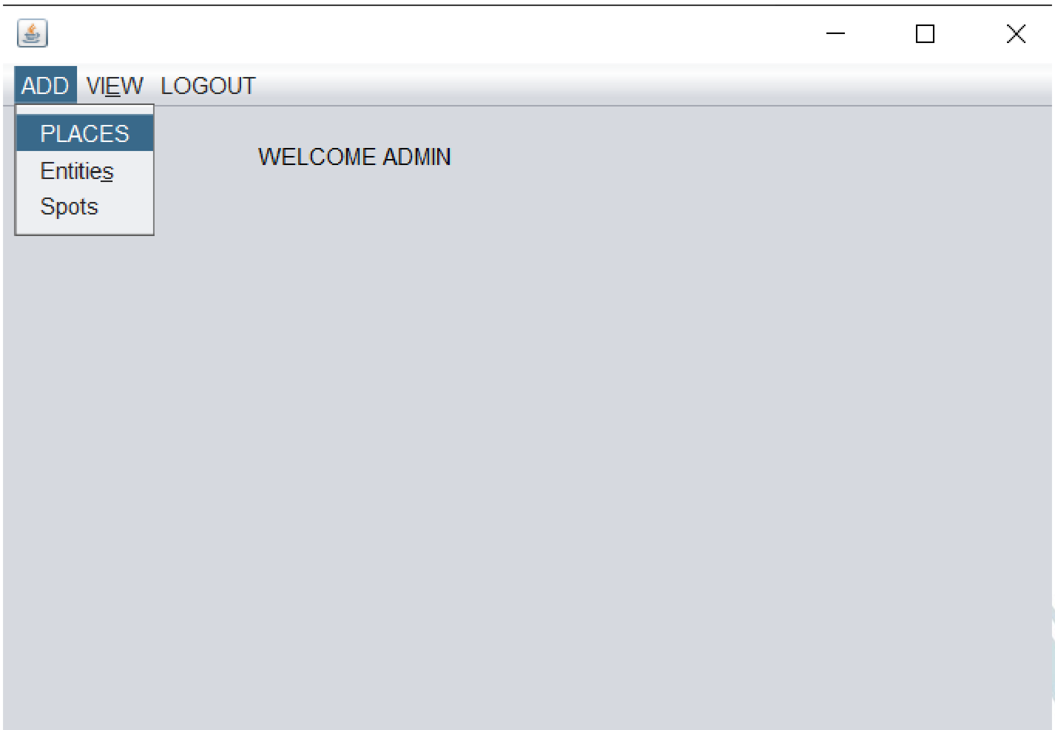
REGISTERED SUCESSFULLY...Please Login now

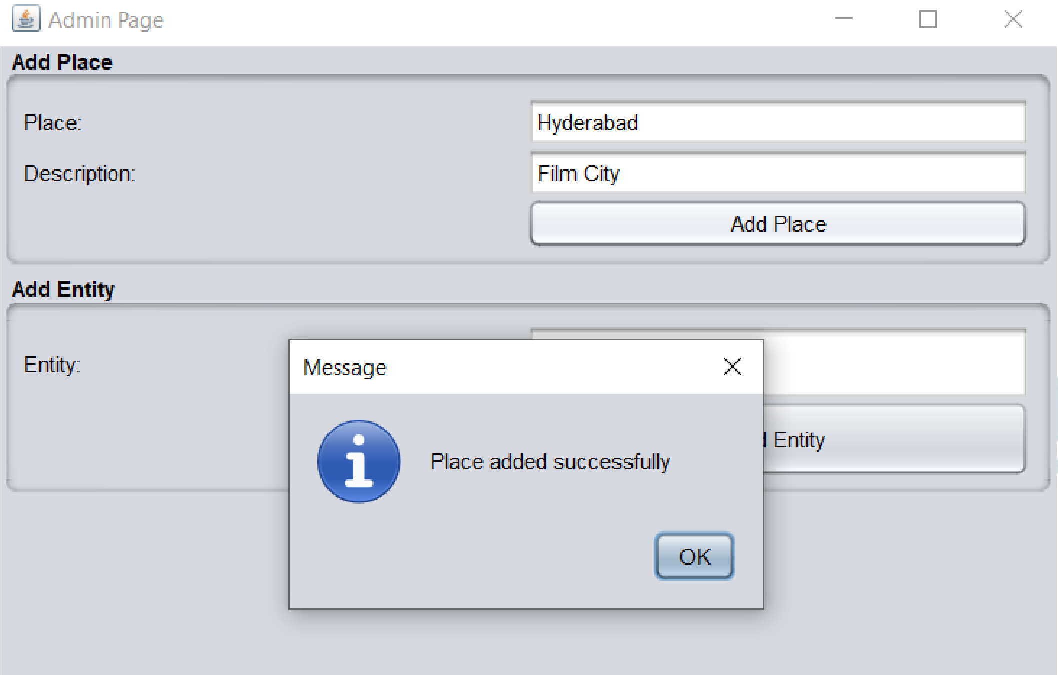
## ADMIN PANEL



Username admin

Password *****





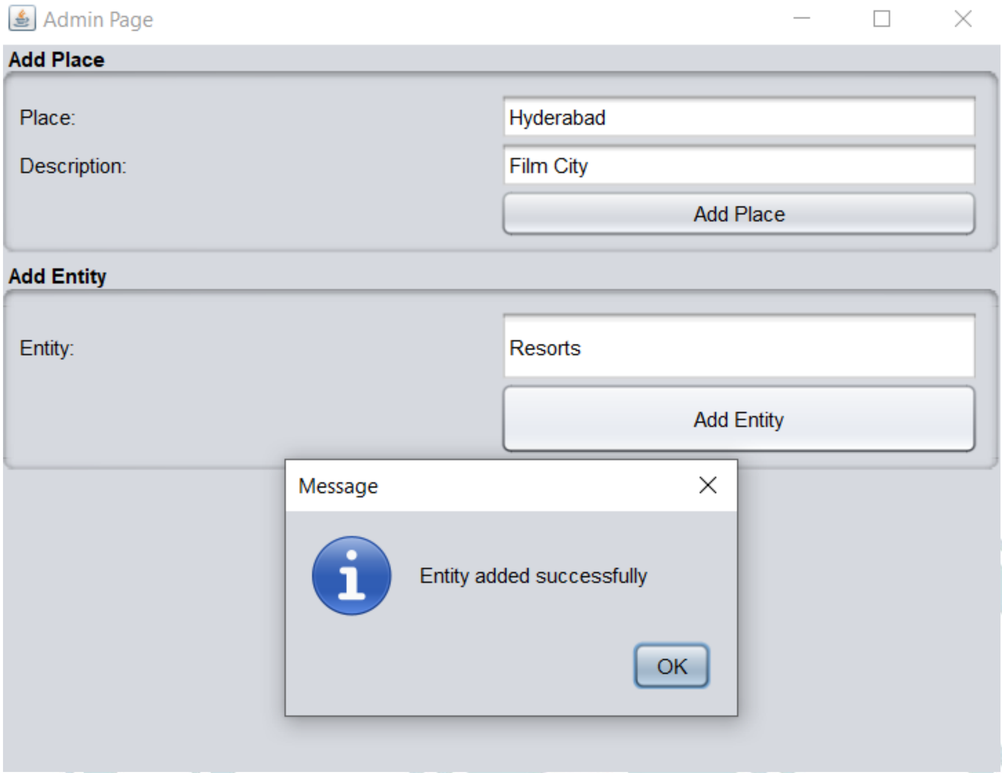
Place: Hyderabad

Description: Film City

Entity:

Message

Place added successfully



Admin Page

Add Place

Place: Hyderabad

Description: Film City

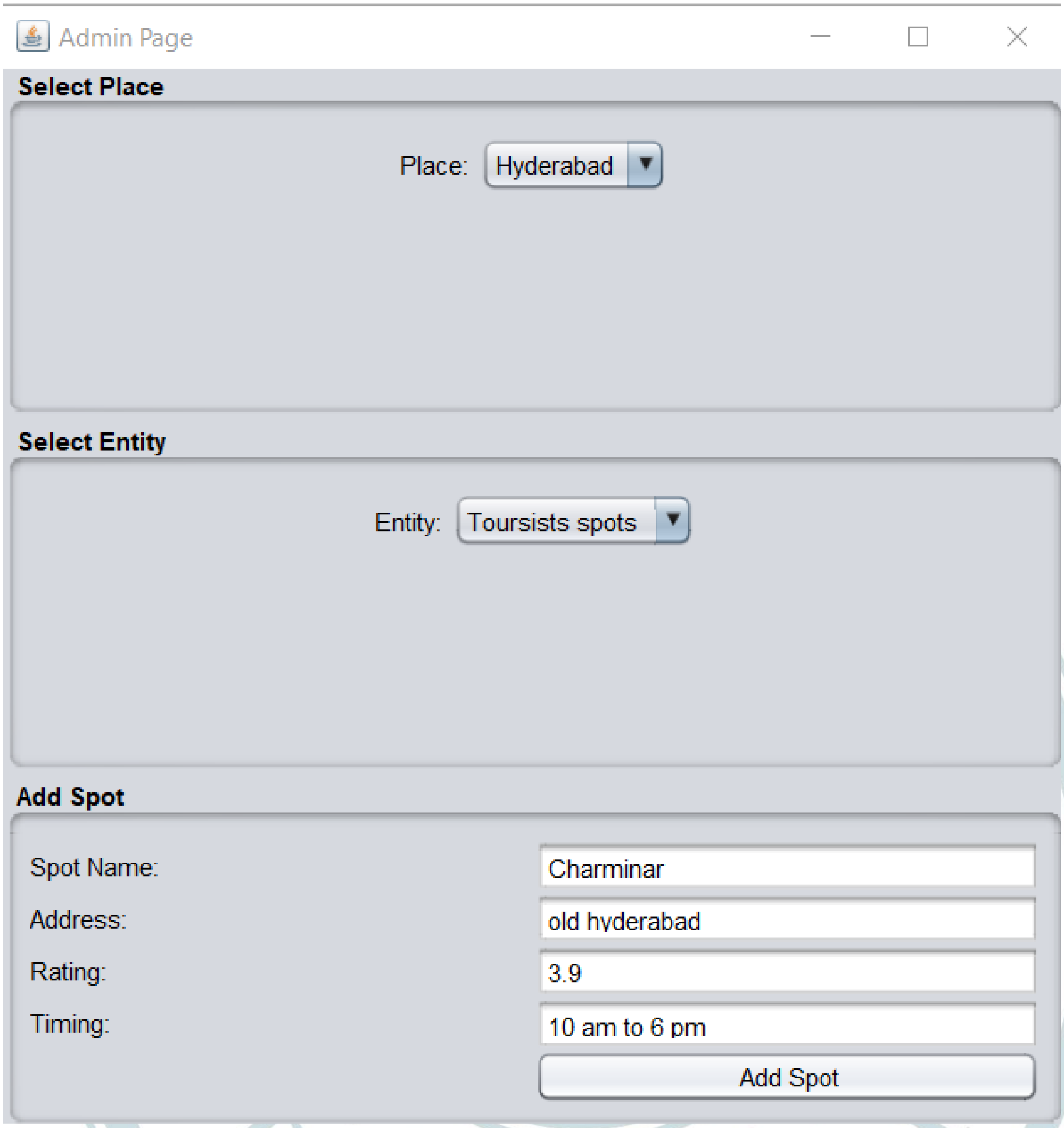
Add Entity

Entity: Resorts

Message

Entity added successfully

LEARN ◦ SERVE ◦ EXCEL
RAJAGIRI



Admin Page

Select Place

Place: Hyderabad

Select Entity

Entity: Toursists spots

Add Spot

Spot Name: Charminar

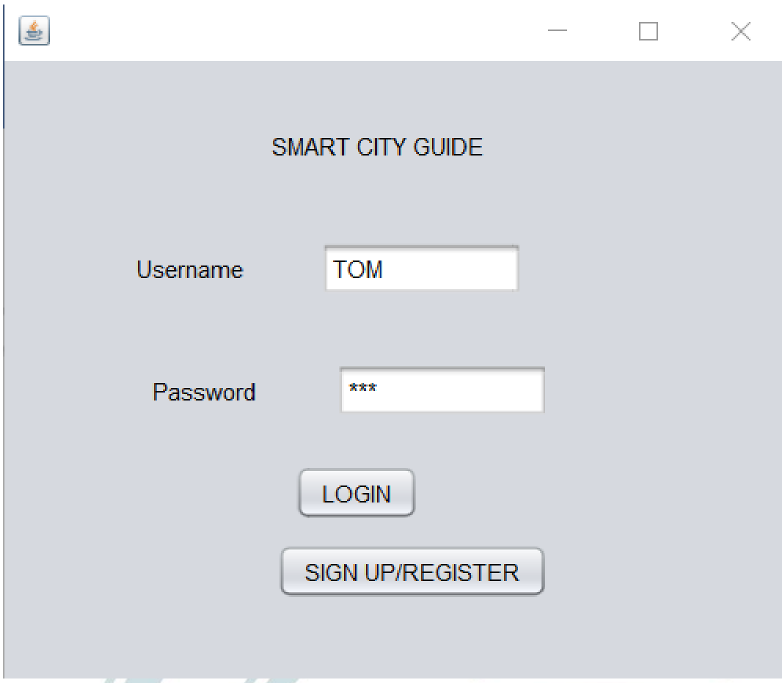
Address: old hyderabad

Rating: 3.9

Timing: 10 am to 6 pm

SERVE RAJAGIRI

## CUSTOMER LOGIN

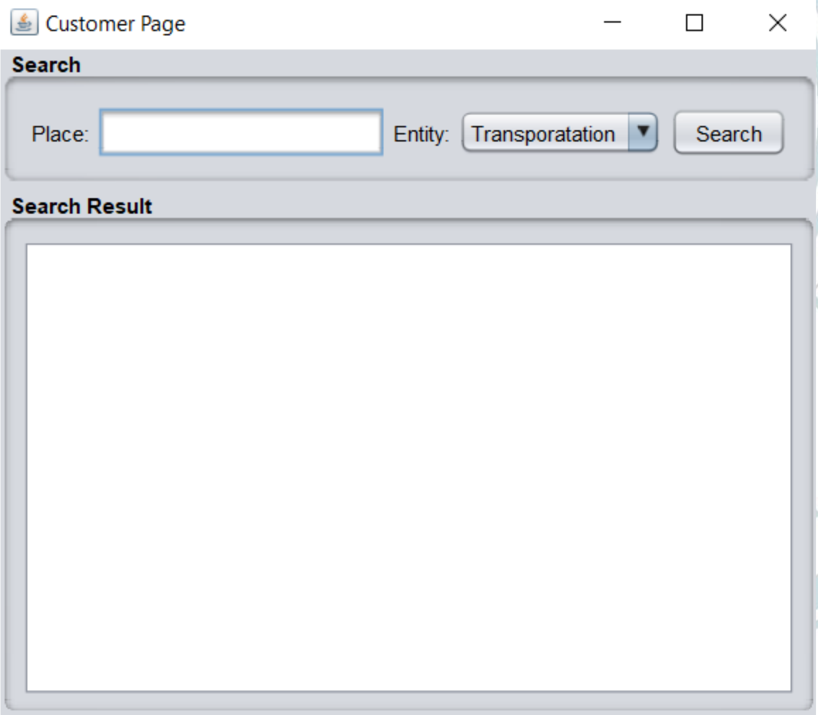


SMART CITY GUIDE

Username: TOM

Password: ***

# CUSTOMER PAGE



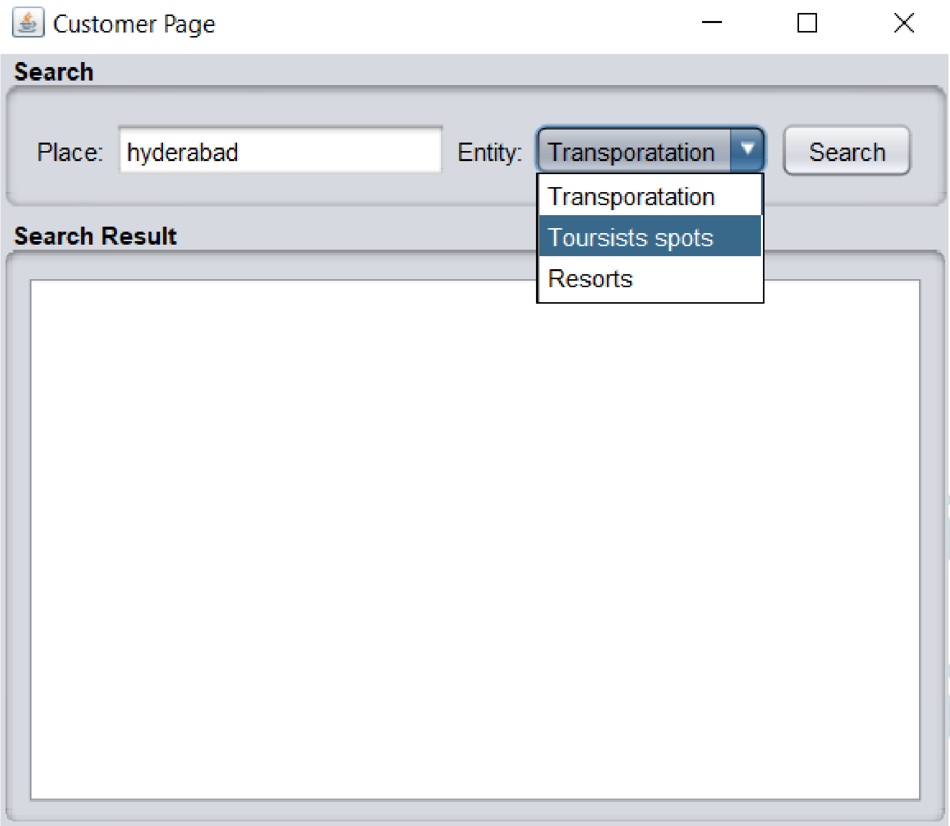
Customer Page

**Search**

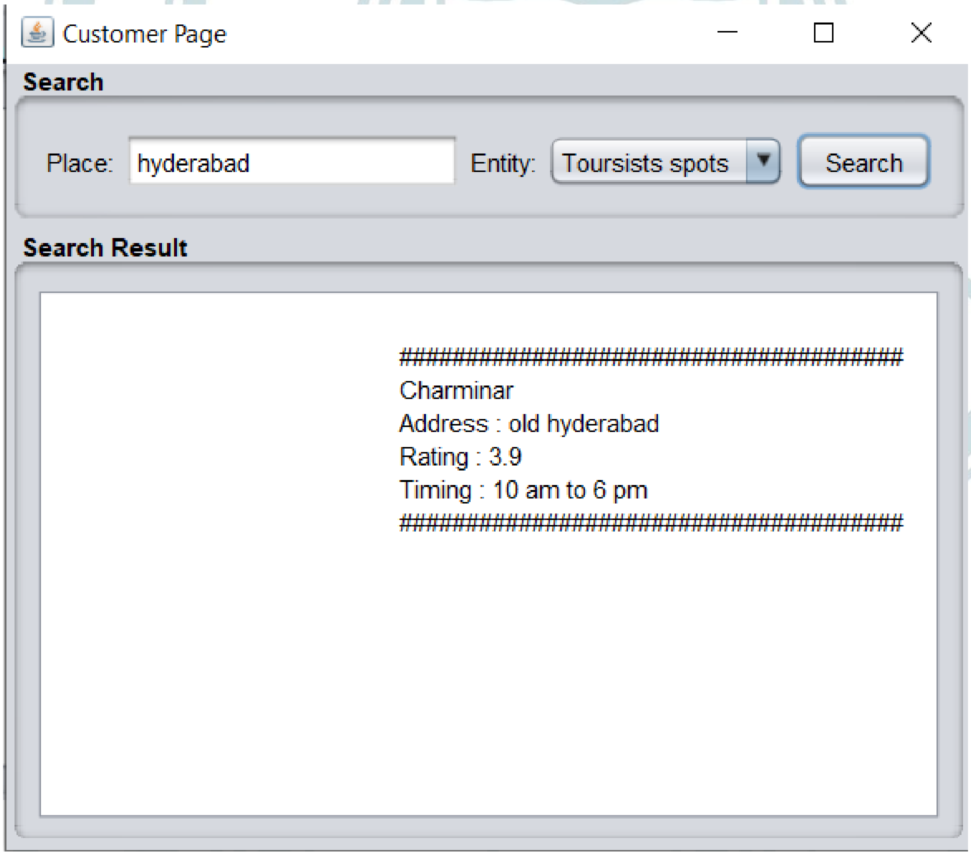
Place:

Entity: Transporatation

**Search Result**



Place: hyderabad Entity: Transporatation



Place: hyderabad Entity: Toursists spots

```
#########################################
Charminar
Address : old hyderabad
Rating : 3.9
Timing : 10 am to 6 pm
#########################################
```

## SOURCE CODE

### Smartcity.java

```
import java.sql.*;
import javax.swing.SwingUtilities;

public class smartcity extends javax.swing.JFrame {

    public smartcity() {
        initComponents();
    }
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jtuname = new javax.swing.JTextField();
        jtpass = new javax.swing.JPasswordField();
        blogin = new javax.swing.JButton();
        jLabel4 = new javax.swing.JLabel();
        bsignup = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jLabel1.setText("SMART CITY GUIDE");

        jLabel2.setText("Username");

        jLabel3.setText("Password");

        jtuname.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jtunameActionPerformed(evt);
            }
        });

        blogin.setText("LOGIN");
        blogin.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                bloginActionPerformed(evt);
            }
        });

        bsignup.setText("SIGN UP/REGISTER");
        bsignup.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                bsignupActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
layout.createSequentialGroup()
                        .addGap(75, 75, 75)
                        .addComponent(jLabel3,  javax.swing.GroupLayout.PREFERRED_SIZE,  66,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(28, 28, 28)
                        .addComponent(jtpass,     javax.swing.GroupLayout.DEFAULT_SIZE,     108,
Short.MAX_VALUE))
                    .addGroup(layout.createSequentialGroup()
                        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                                .addGroup(layout.createSequentialGroup()
                                    .addComponent(jLabel2,
javax.swing.GroupLayout.PREFERRED_SIZE, 66,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                    .addGap(28, 28, 28)
                                    .addComponent(jtuname,
javax.swing.GroupLayout.PREFERRED_SIZE,                                              103,
javax.swing.GroupLayout.PREFERRED_SIZE))
                                .addGroup(layout.createSequentialGroup()
                                    .addComponent(blogin)
                                    .addGap(53, 53, 53)))
                            .addGroup(layout.createSequentialGroup()
                                .addGap(72, 72, 72)
                                .addComponent(bsignup)))))
                .addGap(123, 123, 123))
            .addGroup(layout.createSequentialGroup()
                .addGap(136, 136, 136)
                .addComponent(jLabel1,     javax.swing.GroupLayout.PREFERRED_SIZE,     113,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE)
                .addComponent(jLabel4,     javax.swing.GroupLayout.PREFERRED_SIZE,     196,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(100, 100, 100))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(35, 35, 35)
                .addComponent(jLabel1)
                .addGap(41, 41, 41)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel2)
                    .addComponent(jtuname,          javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(34, 34, 34)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel3)
                    .addComponent(jtpass,               javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jLabel4)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(blogin)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(bsignup)
                .addContainerGap(48, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

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

    private void bloginActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try{
            String uname=jtuname.getText();
            String pass=jtpass.getText();
            connection c=new connection();
            System.out.println(uname+"  "+pass);
            ResultSet rs;
            String str="Select * from login where username= '";
            str=str+uname+"' and password ='"+pass+"'";
            System.out.println(uname+"  "+pass);
        // System.out.println(str);
        rs=c.st.executeQuery(str);
        if(rs.next())
        {
          String id=rs.getString("id");
          switch (rs.getString("usertype"))
          {
             case "admin" : System.out.println("admin");
                          try {
                      for          (javax.swing.UIManager.LookAndFeelInfo          info          :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                          if ("Nimbus".equals(info.getName())) {
                             javax.swing.UIManager.setLookAndFeel(info.getClassName());
                             break;
                               }
                           }
                          } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(AdminMain.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
                          } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(AdminMain.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
                          } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(AdminMain.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
                          } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(AdminMain.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
                          }

                          java.awt.EventQueue.invokeLater(new Runnable() {
                             public void run() {
                                new AdminMain().setVisible(true);
                             }
                          });
                         break;

             case "customer" : SwingUtilities.invokeLater(new Runnable() {
                          public void run() {
                          new CustPageGUI().setVisible(true);
                             }
                          });
          }
        }
        else
        {
```

LEARN ○ SERVE ○ EXCEL
RAJAGIRI

```
                jLabel4.setText("Invalid Username or password");
            }
        }
        catch(Exception e)
        {
            e.printStackTrace();
            jLabel4.setText("Database Error: " + e.getMessage());
        }

        }

    private void bsignupActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            for              (javax.swing.UIManager.LookAndFeelInfo              info              :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(signup.class.getName()).log(java.util.logging.Level.SEV
ERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(signup.class.getName()).log(java.util.logging.Level.SEV
ERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(signup.class.getName()).log(java.util.logging.Level.SEV
ERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(signup.class.getName()).log(java.util.logging.Level.SEV
ERE, null, ex);
        }
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new signup().setVisible(true);
            }
        });
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        try {
```

```
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
          if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
          }
        }
      } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(smartcity.class.getName()).log(java.util.logging.Level.SE
VERE, null, ex);
      } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(smartcity.class.getName()).log(java.util.logging.Level.SE
VERE, null, ex);
      } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(smartcity.class.getName()).log(java.util.logging.Level.SE
VERE, null, ex);
      } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(smartcity.class.getName()).log(java.util.logging.Level.SE
VERE, null, ex);
      }
      //</editor-fold>

      /* Create and display the form */
      java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
          new smartcity().setVisible(true);
        }
      });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton blogin;
    private javax.swing.JButton bsignup;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JPasswordField jtpass;
    private javax.swing.JTextField jtuname;
    // End of variables declaration
}
```

## Signup.java

```
public class signup extends javax.swing.JFrame {
```

```
public signup() {
   initComponents();
}

private void initComponents() {

   jLabel1 = new javax.swing.JLabel();
   jLabel2 = new javax.swing.JLabel();
   jLabel3 = new javax.swing.JLabel();
   jLabel4 = new javax.swing.JLabel();
   jLabel5 = new javax.swing.JLabel();
   jname = new javax.swing.JTextField();
   jpass = new javax.swing.JPasswordField();
   jcpass = new javax.swing.JPasswordField();
   jLabel6 = new javax.swing.JLabel();
   jtfshow = new javax.swing.JLabel();
   bsignup = new javax.swing.JButton();

   setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

   jLabel1.setText("SMART CITY GUIDE");

   jLabel2.setText("REGISTER");

   jLabel3.setText("USERNAME");

   jLabel4.setText("PASSWORD");

   jLabel5.setText("Confirm Password");

   jpass.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
         jpassActionPerformed(evt);
      }
   });

   bsignup.setText("SIGN UP");
   bsignup.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
         bsignupActionPerformed(evt);
      }
   });

   javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
   getContentPane().setLayout(layout);
   layout.setHorizontalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
```

LEARN SERVE EXCEL RAJAGIRI

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(layout.createSequentialGroup()
            .addGap(140, 140, 140)
            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 130,
javax.swing.GroupLayout.PREFERRED_SIZE))
          .addGroup(layout.createSequentialGroup()
            .addGap(166, 166, 166)
            .addComponent(jLabel2,  javax.swing.GroupLayout.PREFERRED_SIZE,  66,
javax.swing.GroupLayout.PREFERRED_SIZE))
          .addGroup(layout.createSequentialGroup()
            .addGap(84, 84, 84)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                  .addComponent(jLabel3)
                  .addGap(49, 49, 49))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                  .addComponent(jLabel4)
                  .addGap(48, 48, 48)))
              .addGroup(layout.createSequentialGroup()
                .addComponent(jLabel5)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)))
            .addGap(19, 19, 19)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
              .addComponent(jpass)
              .addComponent(jcpass)
              .addComponent(jname, javax.swing.GroupLayout.PREFERRED_SIZE, 78,
javax.swing.GroupLayout.PREFERRED_SIZE)))
          .addGroup(layout.createSequentialGroup()
            .addGap(155, 155, 155)
            .addComponent(jLabel6,  javax.swing.GroupLayout.PREFERRED_SIZE,  37,
javax.swing.GroupLayout.PREFERRED_SIZE))
          .addGroup(layout.createSequentialGroup()
            .addGap(160, 160, 160)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
              .addComponent(bsignup,javax.swing.GroupLayout.PREFERRED_SIZE,
78, javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jtfshow,     javax.swing.GroupLayout.PREFERRED_SIZE,
300, javax.swing.GroupLayout.PREFERRED_SIZE))))
        .addGap(108, 108, 108))
    );
```

```
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(27, 27, 27)
                .addComponent(jLabel1)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jLabel2)
                .addGap(32, 32, 32)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel3)
                    .addComponent(jname,                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel4)
                    .addComponent(jpass,                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel5)
                    .addComponent(jcpass,               javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)
                .addComponent(bsignup)
                .addGap(18, 18, 18)
                .addComponent(jtfshow)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel6)
                .addContainerGap(30, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

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

    private void bsignupActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        //get text from jname,jpass,jcpass
        try{
            String name=jname.getText();
            String pass=jpass.getText();
            String cpass=jcpass.getText();
            System.out.println(name+""+pass+""+cpass);
            connection c=new connection();
            if(name!=""&&pass!=""&&cpass!=""){
            if(pass.equals(cpass))
            {
              String str="insert into login(username,password,usertype) values('";
              str=str+name+"','"+pass+"','customer')";
              c.st.executeUpdate(str);
              System.out.println("\t"+name+"   REGISTERED   SUCESSFULLY...Please   Login
now");
              jtfshow.setText("REGISTERED SUCESSFULLY...Please Login now");
           //    this.setVisible(false);
            }
            else
            {
               jtfshow.setText("Password Mismatch!! TRY AGAIN");
               System.out.println("Password Mismatch!! TRY AGAIN");
            }
         }
         else
         {
            jtfshow.setText("Please enter all fields");
         }

         }
         catch(Exception e)
         {
            System.out.println(e);
         }

      }

      /**
       * @param args the command line arguments
       */
      // public static void main(String args[]) {
      //    /* Set the Nimbus look and feel */
      //    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
      //     /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
feel.
      //                                             *        For        details        see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
      //     */
      //    try {
      //                        for    (javax.swing.UIManager.LookAndFeelInfo    info    :
javax.swing.UIManager.getInstalledLookAndFeels()) {
      //            if ("Nimbus".equals(info.getName())) {
      //                javax.swing.UIManager.setLookAndFeel(info.getClassName());
      //                break;
```

LEARN ○ SERVE ○ EXCEL
RAJAGIRI

```
   //           }
   //         }
   //     } catch (ClassNotFoundException ex) {
   //
java.util.logging.Logger.getLogger(signup.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
   //     } catch (InstantiationException ex) {
   //
java.util.logging.Logger.getLogger(signup.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
   //     } catch (IllegalAccessException ex) {
   //
java.util.logging.Logger.getLogger(signup.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
   //     } catch (javax.swing.UnsupportedLookAndFeelException ex) {
   //
java.util.logging.Logger.getLogger(signup.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
   //     }
   //     //</editor-fold>

   //     /* Create and display the form */
   //     java.awt.EventQueue.invokeLater(new Runnable() {
   //         public void run() {
   //             new signup().setVisible(true);
   //         }
   //     });
   // }

   // Variables declaration - do not modify
   private javax.swing.JButton bsignup;
   private javax.swing.JLabel jLabel1;
   private javax.swing.JLabel jLabel2;
   private javax.swing.JLabel jLabel3;
   private javax.swing.JLabel jLabel4;
   private javax.swing.JLabel jLabel5;
   private javax.swing.JLabel jLabel6;
   private javax.swing.JPasswordField jcpass;
   private javax.swing.JTextField jname;
   private javax.swing.JPasswordField jpass;
   private javax.swing.JLabel jtfshow;
   // End of variables declaration
}
```

**Adminmain.java**

```
import javax.swing.SwingUtilities;

 public class AdminMain extends javax.swing.JFrame {
```

```
/**
 * Creates new form NewApplication
 */
public AdminMain() {
    initComponents();
}


private void initComponents() {

    jLabel2 = new javax.swing.JLabel();
    menuBar = new javax.swing.JMenuBar();
    addmenu = new javax.swing.JMenu();
    placesMenuItem = new javax.swing.JMenuItem();
    entityMenuItem = new javax.swing.JMenuItem();
    spotsMenuItem = new javax.swing.JMenuItem();
    editMenu = new javax.swing.JMenu();
    cutMenuItem = new javax.swing.JMenuItem();
    copyMenuItem = new javax.swing.JMenuItem();
    pasteMenuItem = new javax.swing.JMenuItem();
    helpMenu = new javax.swing.JMenu();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jLabel2.setText("WELCOME ADMIN");

    addmenu.setMnemonic('f');
    addmenu.setText("ADD");
    addmenu.addMenuListener(new javax.swing.event.MenuListener() {
        public void menuCanceled(javax.swing.event.MenuEvent evt) {
        }
        public void menuDeselected(javax.swing.event.MenuEvent evt) {
        }
        public void menuSelected(javax.swing.event.MenuEvent evt) {
            addmenuMenuSelected(evt);
        }
    });

    placesMenuItem.setMnemonic('o');
    placesMenuItem.setText("PLACES");
    placesMenuItem.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            placesMenuItemActionPerformed(evt);
        }
    });
    addmenu.add(placesMenuItem);

    entityMenuItem.setMnemonic('s');
    entityMenuItem.setText("Entities");
```

```
        entityMenuItem.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                entityMenuItemActionPerformed(evt);
            }
        });
        addmenu.add(entityMenuItem);

        spotsMenuItem.setMnemonic('a');
        spotsMenuItem.setText("Spots");
        spotsMenuItem.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                spotsMenuItemActionPerformed(evt);
            }
        });
        addmenu.add(spotsMenuItem);

        menuBar.add(addmenu);

        editMenu.setMnemonic('e');
        editMenu.setText("VIEW");

        cutMenuItem.setMnemonic('t');
        cutMenuItem.setText("PLACES");
        editMenu.add(cutMenuItem);

        copyMenuItem.setMnemonic('y');
        copyMenuItem.setText("ENTITIES");
        editMenu.add(copyMenuItem);

        pasteMenuItem.setMnemonic('p');
        pasteMenuItem.setText("SPOTS");
        editMenu.add(pasteMenuItem);

        menuBar.add(editMenu);

        helpMenu.setMnemonic('h');
        helpMenu.setText("LOGOUT");
        menuBar.add(helpMenu);

        setJMenuBar(menuBar);

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(132, 132, 132)
                .addComponent(jLabel2,    javax.swing.GroupLayout.PREFERRED_SIZE,    113,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(304, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel2,     javax.swing.GroupLayout.PREFERRED_SIZE,     38,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(281, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void addmenuMenuSelected(javax.swing.event.MenuEvent evt) {
        // TODO add your handling code here:


    }

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

        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new placesentity().setVisible(true);
            }
        });


    }

    private void entityMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new placesentity().setVisible(true);
            }
        });

    }

    private void spotsMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new AdminPageGUI().setVisible(true);
            }
        });
    }
```

LEARN ∘ SERVE ∘ EXCEL
RAJAGIRI

```
    /**
     * @param args the command line arguments
     */

    // Variables declaration - do not modify
    private javax.swing.JMenu addmenu;
    private javax.swing.JMenuItem copyMenuItem;
    private javax.swing.JMenuItem cutMenuItem;
    private javax.swing.JMenu editMenu;
    private javax.swing.JMenuItem entityMenuItem;
    private javax.swing.JMenu helpMenu;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JMenuBar menuBar;
    private javax.swing.JMenuItem pasteMenuItem;
    private javax.swing.JMenuItem placesMenuItem;
    private javax.swing.JMenuItem spotsMenuItem;
    // End of variables declaration

}
```

**AdminpageGUI.java**

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class AdminPageGUI extends JFrame {
    private Connection conn;
    private Statement st;
    private ResultSet rs;
    private JTextField spotNameField, addressField, ratingField, timingField;
    private JComboBox<String> placeComboBox, entityComboBox;
    private JButton addSpotBtn;

    public AdminPageGUI() {
        setTitle("Admin Page");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new GridLayout(5, 1));

        connectToDatabase();

        createPlacePanel();
        createEntityPanel();
        createSpotPanel();

        pack();
        setLocationRelativeTo(null);
    }
    private void connectToDatabase() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            conn    =    DriverManager.getConnection("jdbc:mysql://localhost:3306/smartcity_db",
"root", "");
            st = conn.createStatement();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private void createPlacePanel() {
        JPanel placePanel = new JPanel();
        placePanel.setBorder(BorderFactory.createTitledBorder("Select Place"));
        placePanel.setLayout(new FlowLayout());

        JLabel placeLabel = new JLabel("Place:");
        placeComboBox = new JComboBox<>();
        loadPlaces();

        placePanel.add(placeLabel);
        placePanel.add(placeComboBox);

        add(placePanel);
    }

    private void loadPlaces() {
        try {
            String query = "SELECT area FROM places";
            rs = st.executeQuery(query);
            while (rs.next()) {
                placeComboBox.addItem(rs.getString("area"));
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }

    private void createEntityPanel() {
        JPanel entityPanel = new JPanel();
        entityPanel.setBorder(BorderFactory.createTitledBorder("Select Entity"));
        entityPanel.setLayout(new FlowLayout());

        JLabel entityLabel = new JLabel("Entity:");
        entityComboBox = new JComboBox<>();
        loadEntities();

        entityPanel.add(entityLabel);
        entityPanel.add(entityComboBox);
```

RAJAGIRI LEARN ◦ SERVE ◦ EXCEL

```
        add(entityPanel);
    }

    private void loadEntities() {
        try {
            String query = "SELECT entity_name FROM entity";
            rs = st.executeQuery(query);
            while (rs.next()) {
                entityComboBox.addItem(rs.getString("entity_name"));
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }

    private void createSpotPanel() {
        JPanel spotPanel = new JPanel();
        spotPanel.setBorder(BorderFactory.createTitledBorder("Add Spot"));
        spotPanel.setLayout(new GridLayout(5, 2));

        JLabel spotNameLabel = new JLabel("Spot Name:");
        spotNameField = new JTextField();
        JLabel addressLabel = new JLabel("Address:");
        addressField = new JTextField();
        JLabel ratingLabel = new JLabel("Rating:");
        ratingField = new JTextField();
        JLabel timingLabel = new JLabel("Timing:");
        timingField = new JTextField();

        addSpotBtn = new JButton("Add Spot");

        spotPanel.add(spotNameLabel);
        spotPanel.add(spotNameField);
        spotPanel.add(addressLabel);
        spotPanel.add(addressField);
        spotPanel.add(ratingLabel);
        spotPanel.add(ratingField);
        spotPanel.add(timingLabel);
        spotPanel.add(timingField);
        spotPanel.add(new JLabel()); // Empty label for layout
        spotPanel.add(addSpotBtn);

        addSpotBtn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                addSpot();
            }
        });

        add(spotPanel);
    }
    private void addSpot() {
        try {
            String spotName = spotNameField.getText();
            String address = addressField.getText();
            String rating = ratingField.getText();
            String timing = timingField.getText();

            String selectedPlace = (String) placeComboBox.getSelectedItem();
            String selectedEntity = (String) entityComboBox.getSelectedItem();

            int placeId = getPlaceId(selectedPlace);
            int entityId = getEntityId(selectedEntity);

            String query = "INSERT INTO spot(placeid, entityid, name, address, rating, timing)
VALUES (?, ?, ?, ?, ?, ?)";
            PreparedStatement pst = conn.prepareStatement(query);
            pst.setInt(1, placeId);
            pst.setInt(2, entityId);
            pst.setString(3, spotName);
            pst.setString(4, address);
            pst.setString(5, rating);
            pst.setString(6, timing);
            pst.executeUpdate();

            JOptionPane.showMessageDialog(this, "Spot added successfully");
        } catch (Exception ex) {
            ex.printStackTrace();
            JOptionPane.showMessageDialog(this, "Error adding spot: " + ex.getMessage(),
"Error", JOptionPane.ERROR_MESSAGE);
        }
    }

    private int getPlaceId(String placeName) {
        int placeId = 0;
        try {
            String query = "SELECT id FROM places WHERE area = ?";
            PreparedStatement pst = conn.prepareStatement(query);
            pst.setString(1, placeName);
            ResultSet rs = pst.executeQuery();
            if (rs.next()) {
                placeId = rs.getInt("id");
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
        return placeId;
    }

    private int getEntityId(String entityName) {
        int entityId = 0;
        try {
            String query = "SELECT id FROM entity WHERE entity_name = ?";
            PreparedStatement pst = conn.prepareStatement(query);
            pst.setString(1, entityName);
            ResultSet rs = pst.executeQuery();
            if (rs.next()) {
                entityId = rs.getInt("id");
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
        return entityId;
    }

    // public static void main(String[] args) {
    //     SwingUtilities.invokeLater(new Runnable() {
    //         public void run() {
    //             new AdminPageGUI().setVisible(true);
    //         }
    //     });
    // }
}
```

**Placesentities.java**

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class placesentity extends JFrame {
    private Connection conn;
    private Statement st;
    private ResultSet rs;
    private JTextField placeField, placeDescField, entityField, addressField;
    private JButton addPlaceBtn, addEntityBtn;

    public placesentity() {
        setTitle("Admin Page");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new GridLayout(4, 1));

        connectToDatabase();

        createPlacePanel();
        createEntityPanel();

        pack();
        setLocationRelativeTo(null);
    }

    private void connectToDatabase() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/smartcity_db",
"root", "");
            st = conn.createStatement();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private void createPlacePanel() {
        JPanel placePanel = new JPanel();
        placePanel.setBorder(BorderFactory.createTitledBorder("Add Place"));
        placePanel.setLayout(new GridLayout(3, 2));

        JLabel placeLabel = new JLabel("Place:");
        placeField = new JTextField();
        JLabel placeDescLabel = new JLabel("Description:");
        placeDescField = new JTextField();

        addPlaceBtn = new JButton("Add Place");

        placePanel.add(placeLabel);
        placePanel.add(placeField);
        placePanel.add(placeDescLabel);
        placePanel.add(placeDescField);
        placePanel.add(new JLabel()); // Empty label for layout
        placePanel.add(addPlaceBtn);

        addPlaceBtn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                addPlace();
            }
        });

        add(placePanel);
    }

    private void addPlace() {
        try {
            String place = placeField.getText();
            String placeDesc = placeDescField.getText();
            String query = "INSERT INTO places(area, area_desc) VALUES (?, ?)";
            PreparedStatement pst = conn.prepareStatement(query);
            pst.setString(1, place);
            pst.setString(2, placeDesc);
            pst.executeUpdate();
            JOptionPane.showMessageDialog(this, "Place added successfully");
        } catch (Exception ex) {
            ex.printStackTrace();
            JOptionPane.showMessageDialog(this, "Error adding place: " + ex.getMessage(),
"Error", JOptionPane.ERROR_MESSAGE);
        }
    }

    private void createEntityPanel() {
        JPanel entityPanel = new JPanel();
        entityPanel.setBorder(BorderFactory.createTitledBorder("Add Entity"));
        entityPanel.setLayout(new GridLayout(2, 2));

        JLabel entityLabel = new JLabel("Entity:");
        entityField = new JTextField();
        addEntityBtn = new JButton("Add Entity");

        entityPanel.add(entityLabel);
        entityPanel.add(entityField);
        entityPanel.add(new JLabel()); // Empty label for layout
        entityPanel.add(addEntityBtn);

        addEntityBtn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                addEntity();
            }
        });

        add(entityPanel);
    }

    private void addEntity() {
        try {
            String entity = entityField.getText();
            String query = "INSERT INTO entity(entity_name) VALUES (?)";
            PreparedStatement pst = conn.prepareStatement(query);
            pst.setString(1, entity);
            pst.executeUpdate();
            JOptionPane.showMessageDialog(this, "Entity added successfully");
        } catch (Exception ex) {
            ex.printStackTrace();
            JOptionPane.showMessageDialog(this, "Error adding entity: " + ex.getMessage(),
"Error", JOptionPane.ERROR_MESSAGE);
        }
    }


    // public static void main(String[] args) {
    //     SwingUtilities.invokeLater(new Runnable() {
    //         public void run() {
```

LEARN ◦ SERVE ◦ EXCEL
RAJAGIRI

```
    //            new placesentity().setVisible(true);
    //        }
    //    });
    // }
}
```

**Custpage.java**

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class CustPageGUI extends JFrame {
    private JTextField placeField;
    private JComboBox<String> entityComboBox;
    private JButton searchBtn;
    private JTextArea resultArea;

    private Connection conn;
    private Statement st;
    private ResultSet rs;

    public CustPageGUI() {
        setTitle("Customer Page");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        connectToDatabase();

        createSearchPanel();
        createResultPanel();

        pack();
        setLocationRelativeTo(null);
    }

    private void connectToDatabase() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/smartcity_db",
"root", "");
            st = conn.createStatement();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private void createSearchPanel() {
        JPanel searchPanel = new JPanel();
        searchPanel.setBorder(BorderFactory.createTitledBorder("Search"));
        searchPanel.setLayout(new FlowLayout());

        JLabel placeLabel = new JLabel("Place:");
        placeField = new JTextField(15);

        JLabel entityLabel = new JLabel("Entity:");
        entityComboBox = new JComboBox<>();
        loadEntities();

        searchBtn = new JButton("Search");
        searchBtn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
              search();
            }
        });

        searchPanel.add(placeLabel);
        searchPanel.add(placeField);
        searchPanel.add(entityLabel);
        searchPanel.add(entityComboBox);
        searchPanel.add(searchBtn);

        add(searchPanel, BorderLayout.NORTH);
    }

    private void loadEntities() {
        try {
            String query = "SELECT entity_name FROM entity";
            rs = st.executeQuery(query);
            while (rs.next()) {
              entityComboBox.addItem(rs.getString("entity_name"));
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }

    private void createResultPanel() {
        JPanel resultPanel = new JPanel();
        resultPanel.setBorder(BorderFactory.createTitledBorder("Search Result"));
        resultPanel.setLayout(new BorderLayout());

        resultArea = new JTextArea(15, 40);
        resultArea.setEditable(false);
        JScrollPane scrollPane = new JScrollPane(resultArea);

        resultPanel.add(scrollPane, BorderLayout.CENTER);

        add(resultPanel, BorderLayout.CENTER);
```

RAJAGIRI LEARN • SERVE • EXCEL

```
    }

    private void search() {
        try {
            String place = placeField.getText();
            String selectedEntity = (String) entityComboBox.getSelectedItem();

            String query = "SELECT id FROM places WHERE area = ?";
            PreparedStatement pst = conn.prepareStatement(query);
            pst.setString(1, place);
            ResultSet rs = pst.executeQuery();
            if (rs.next()) {
                int placeId = rs.getInt("id");

                query = "SELECT id FROM entity WHERE entity_name = ?";
                pst = conn.prepareStatement(query);
                pst.setString(1, selectedEntity);
                rs = pst.executeQuery();
                if (rs.next()) {
                    int entityId = rs.getInt("id");

                    query = "SELECT * FROM spot WHERE placeid = ? AND entityid = ?";
                    pst = conn.prepareStatement(query);
                    pst.setInt(1, placeId);
                    pst.setInt(2, entityId);
                    rs = pst.executeQuery();

                    if (rs.next()) {
                        resultArea.setText("");
                        do {
                            resultArea.append("\n\t\t########################################");
                            resultArea.append("\n\t\t" + rs.getString("name"));
                            resultArea.append("\n\t\tAddress : " + rs.getString("address"));
                            resultArea.append("\n\t\tRating : " + rs.getString("rating"));
                            resultArea.append("\n\t\tTiming : " + rs.getString("timing"));
                        } while (rs.next());
                        resultArea.append("\n\t\t########################################");
                    } else {
                        resultArea.setText("\n\n\tNo information available...");
                    }
                }
            } else {
                resultArea.setText("\nLocation not found in the system...!!");
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
            JOptionPane.showMessageDialog(this, "Error searching: " + ex.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
        }
    }
    // public static void main(String[] args) {
    //     SwingUtilities.invokeLater(new Runnable() {
    //         public void run() {
    //             new CustPageGUI().setVisible(true);
    //         }
    //     });
    // }
}
```