

Zero Hunger

16-04-2024

PROBLEM STATEMENT:

Design a Zero Hunger Portal using Java programming language with three interfaces: User, Hotel, and NGO. Users can register, search for nearby entities, request food donations, or volunteer. Hotels can register, update food inventory, and accept donation requests. NGOs can register, create events, and request donations. The portal ensures secure communication, incorporates a reliable database, and has robust error handling mechanisms to prevent data loss or system failure.

CODE:

```
package management;

import javax.swing.*;
import java.awt.*;

public class Splash extends JFrame implements Runnable {

    Thread t;

    Splash() {

        ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("icons/MAIN1.png"));

        Image i2 = i1.getImage().getScaledInstance(1000, 700,
Image.SCALE_DEFAULT);

        ImageIcon i3 = new ImageIcon(i2);

        javax.swing.JLabel image = new javax.swing.JLabel(i3);

        add(image);

        t = new Thread(this);

        t.start();

        setVisible(true);
    }

    public void run() {
        try {
            for (int i = 0; i < 100; i++) {
                int r = (int) (Math.random() * 255);
                int g = (int) (Math.random() * 255);
                int b = (int) (Math.random() * 255);
                setForeground(new Color(r, g, b));
                System.out.println("Color: " + r + ", " + g + ", " + b);
                Thread.sleep(100);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
int x = 1;

for (int i = 2; i <= 600; i += 4, x += 1) {
    setLocation(600 - ((i + x) / 2), 350 - (i / 2));
    setSize(i + 3 * x, i + x / 2);

    try {
        Thread.sleep(10);
    } catch (Exception e) {
    }
}

public void run() {
    try {
        Thread.sleep(7000);
        setVisible(false);

        // Next Frame
        new Login();
    } catch (Exception e) {
    }
}

public static void main(String[] args) {
    new Splash();
}
```

```
    }

}

package management;

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

public class Project extends JFrame implements ActionListener {

    Project() {
        setSize(1540, 850);

        ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("icons/third.jpg"));

        Image i2 = i1.getImage().getScaledInstance(1500, 750,
Image.SCALE_DEFAULT);

        ImageIcon i3 = new ImageIcon(i2);
        javax.swing.JLabel image = new javax.swing.JLabel(i3);
        add(image);

        JMenuBar mb = new JMenuBar();

        // New Information
        JMenu newInformation = new JMenu("New Information");
        newInformation.setForeground(Color.BLUE);
        mb.add(newInformation);
```

```
JMenuItem facultyInfo = new JMenuItem("New Hotel Information");
facultyInfo.setBackground(Color.WHITE);
facultyInfo.addActionListener(this);
newInformation.add(facultyInfo);

JMenuItem studentInfo = new JMenuItem("New Need Information");
studentInfo.setBackground(Color.WHITE);
studentInfo.addActionListener(this);
newInformation.add(studentInfo);

// Details
JMenu details = new JMenu("View Details");
details.setForeground(Color.RED);
mb.add(details);

JMenuItem facultydetails = new JMenuItem("View Hotel Details");
facultydetails.setBackground(Color.WHITE);
facultydetails.addActionListener(this);
details.add(facultydetails);

JMenuItem studentdetails = new JMenuItem("View NeedDetails");
studentdetails.setBackground(Color.WHITE);
studentdetails.addActionListener(this);
details.add(studentdetails);
```

```
// UpdateInfo

JMenu updateInfo = new JMenu("Update Details");
updateInfo.setForeground(Color.RED);
mb.add(updateInfo);

JMenuItem updatefacultyinfo = new JMenuItem("Update Hotel
Details");
updatefacultyinfo.setBackground(Color.WHITE);
updatefacultyinfo.addActionListener(this);
updateInfo.add(updatefacultyinfo);

JMenuItem updatestudentinfo = new JMenuItem("Update Need
Details");
updatestudentinfo.setBackground(Color.WHITE);
updatestudentinfo.addActionListener(this);
updateInfo.add(updatestudentinfo);

// fee

JMenu fee = new JMenu("Food Details");
fee.setForeground(Color.BLUE);
mb.add(fee);

JMenuItem feeform = new JMenuItem("Food Details");
feeform.setBackground(Color.WHITE);
feeform.addActionListener(this);
fee.add(feeform);
```

```
// Utility

JMenu utility = new JMenu("Utility");
utility.setForeground(Color.RED);
mb.add(utility);

JMenuItem notepad = new JMenuItem("Notepad");
notepad.setBackground(Color.WHITE);
notepad.addActionListener(this);
utility.add(notepad);

JMenuItem calc = new JMenuItem("Calculator");
calc.setBackground(Color.WHITE);
calc.addActionListener(this);
utility.add(calc);

// about

JMenu about = new JMenu("About");
about.setForeground(Color.BLUE);
mb.add(about);

JMenuItem ab = new JMenuItem("About");
ab.setBackground(Color.WHITE);
ab.addActionListener(this);
about.add(ab);

// exit
```

```
JMenu exit = new JMenu("Exit");

exit.setForeground(Color.RED);

mb.add(exit);

JMenuItem ex = new JMenuItem("Exit");

ex.setBackground(Color.WHITE);

ex.addActionListener(this);

exit.add(ex);

setJMenuBar(mb);

setVisible(true);

}

public void actionPerformed(ActionEvent ae) {

String msg = ae.getActionCommand();

if (msg.equals("Exit")) {

setVisible(false);

} else if (msg.equals("Calculator")) {

try {

Runtime.getRuntime().exec("calc.exe");

} catch (Exception e) {

}

} else if (msg.equals("Notepad")) {

try {
```

```
Runtime.getRuntime().exec("notepad.exe");

} catch (Exception e) {

}

} else if (msg.equals("New Hotel Information")) {

new AddHotel();

} else if (msg.equals("New Need Information")) {

new AddNeeds();

} else if (msg.equals("View Hotel Details")) {

new HotelDetails();

} else if (msg.equals("View Need Details")) {

new NeedDetails();

}

else if (msg.equals("Update Hotel Details")) {

new UpdateHotel();

} else if (msg.equals("Update Need Details")) {

new UpdateNeedDetails();

/*else if (msg.equals("Enter Marks")) {

new EnterMarks();*/



} else if (msg.equals("About")) {

new About();

} else if (msg.equals("Food Details")) {

new FoodFeeForm();

}
```

```
}

public static void main(String[] args) {
    new Project();
}

}

package management;

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

public class Login extends JFrame {

    Login() {
        setTitle("Login");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Set a custom font
        Font customFont = new Font("Arial", Font.BOLD, 18);
        UIManager.put("Button.font", customFont);
        UIManager.put("Label.font", customFont);

        // Create content pane with GridBagLayout
    }
}
```

```
JPanel contentPane = new JPanel(new GridBagLayout()) {  
    // Override paintComponent to draw background gradient  
    @Override  
    protected void paintComponent(Graphics g) {  
        super.paintComponent(g);  
        // Draw gradient background  
        Graphics2D g2d = (Graphics2D) g;  
        int w = getWidth();  
        int h = getHeight();  
        Color color1 = new Color(255, 255, 255);  
        Color color2 = new Color(220, 220, 220);  
        GradientPaint gp = new GradientPaint(0, 0, color1, 0, h,  
color2);  
        g2d.setPaint(gp);  
        g2d.fillRect(0, 0, w, h);  
    }  
};  
  
GridBagConstraints gbc = new GridBagConstraints();  
gbc.gridx = 0;  
gbc.gridy = 0;  
gbc.insets = new Insets(10, 10, 10, 10); // Padding  
  
// Add buttons with some styling  
gbc.gridy++;  
JButton hotelButton = new JButton("Hotel Interface");  
styleButton(hotelButton);  
JButton ngoButton = new JButton("NGO Interface");
```

```
styleButton(ngoButton);

JButton userButton = new JButton("User Interface");

styleButton(userButton);

contentPane.add(hotelButton, gbc);
gbc.gridx++;
contentPane.add(ngoButton, gbc);
gbc.gridx++;
contentPane.add(userButton, gbc);

// Button action listeners
hotelButton.addActionListener(e -> openInterface("Hotel"));
ngoButton.addActionListener(e -> openInterface("NGO"));
userButton.addActionListener(e -> openInterface("User"));

setContentPane(contentPane);
pack(); // Adjust frame size
setSize(600, 400); // Set a custom size (you can adjust this as
needed)

setLocationRelativeTo(null); // Center the frame
setVisible(true);

}

private void openInterface(String interfaceType) {
if (interfaceType.equals("Hotel")) {
// Open EnterFood directly for the hotel interface
setVisible(false); // Hide the login window
new EnterFood().setVisible(true); // Open EnterFood
}
}
```

```
    } else {

        // For other interfaces, show a message (you can modify this as
needed)

        JOptionPane.showMessageDialog(this, interfaceType + " Interface
clicked");

        // Add code to open LoginBoard Interface

        Loginboard loginBoard = new Loginboard();
        loginBoard.setVisible(true);

    }

}

// Method to style buttons

private void styleButton(JButton button) {

    button.setBackground(new Color(240, 240, 240));

    button.setForeground(Color.DARK_GRAY);

    button.setFocusPainted(false);

    button.setBorder(BorderFactory.createCompoundBorder(
        BorderFactory.createLineBorder(new Color(180, 180, 180),
1),
        BorderFactory.createEmptyBorder(10, 20, 10, 20)
    ));

}

public static void main(String[] args) {

    SwingUtilities.invokeLater(Login::new); // Run GUI code in Event
Dispatch Thread

}

}
```

```
package management;

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

public class Loginboard extends JFrame implements ActionListener{

    JButton login, cancel;
    JTextField tfusername, tfpassword;

    Loginboard () {

        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        javax.swing.JLabel lblusername = new
        javax.swing.JLabel("Username");
        lblusername.setBounds(40, 20, 100, 20);
        add(lblusername);

        tfusername = new JTextField();
        tfusername.setBounds(150, 20, 150, 20);
        add(tfusername);
```

```
javax.swing.JLabel lblpassword = new
javax.swing.JLabel("Password");

lblpassword.setBounds(40, 70, 100, 20);
add(lblpassword);

tfpassword = new JPasswordField();
tfpassword.setBounds(150, 70, 150, 20);
add(tfpassword);

login = new JButton("Login");
login.setBounds(40, 140, 120, 30);
login.setBackground(Color.BLACK);
login.setForeground(Color.WHITE);
login.addActionListener(this);
login.setFont(new Font("Tahoma", Font.BOLD, 15));
add(login);

cancel = new JButton("Cancel");
cancel.setBounds(180, 140, 120, 30);
cancel.setBackground(Color.BLACK);
cancel.setForeground(Color.WHITE);
cancel.addActionListener(this);
cancel.setFont(new Font("Tahoma", Font.BOLD, 15));
add(cancel);

ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("icons/second.jpg"));
```

```
        Image i2 = i1.getImage().getScaledInstance(200, 200,
Image.SCALE_DEFAULT);

        ImageIcon i3 = new ImageIcon(i2);

        javax.swing.JLabel image = new javax.swing.JLabel(i3);

        image.setBounds(350, 0, 200, 200);

        add(image);

        setSize(600, 300);

        setLocation(500, 250);

        setVisible(true);

    }

}

public void actionPerformed(ActionEvent ae) {

    if (ae.getSource() == login) {

        String username = tfusername.getText();

        String password = tfpassword.getText();

        String query = "select * from login where
username='"+username+"' and password='"+password+"'";

        try {

            Conn c = new Conn();

            ResultSet rs = c.s.executeQuery(query);

            if (rs.next()) {

                setVisible(false);

            }

        } catch (Exception e) {

            JOptionPane.showMessageDialog(null, e);

        }

    }

}
```

```
        new Project();

    }

    else {
        JOptionPane.showMessageDialog(null, "Invalid username
or password");
        setVisible(false);
    }

} catch (Exception e) {
    e.printStackTrace();
}

} else if (ae.getSource() == cancel) {
    setVisible(false);
}

}

public static void main(String[] args) {
    new Loginboard();
}

}

package management;
```

```
import javax.swing.*;  
import java.awt.*;  
import java.util.*;  
import java.awt.event.*;  
  
public class AddHotel extends JFrame implements ActionListener{  
  
    JTextField tfname, tffname, tfaddress, tfphone, tfemail, tfx, tfxii,  
    tfaadhar , tfgstin;  
    javax.swing.JLabel labelrollno;  
  
    JComboBox cbcourse, cbbranch;  
    JButton submit, cancel;  
  
    Random ran = new Random();  
    long first4 = Math.abs((ran.nextLong() % 9000L) + 1000L);  
  
    AddHotel() {  
  
        setSize(900, 700);  
        setLocation(350, 50);  
  
        setLayout(null);  
    }  
}
```

```
javax.swing.JLabel heading = new javax.swing.JLabel("New Hotel  
Details");  
  
heading.setBounds(310, 30, 500, 50);  
  
heading.setFont(new Font("serif", Font.BOLD, 30));  
  
add(heading);  
  
  
javax.swing.JLabel lblname = new javax.swing.JLabel("Name");  
  
lblname.setBounds(50, 150, 100, 30);  
  
lblname.setFont(new Font("serif", Font.BOLD, 20));  
  
add(lblname);  
  
  
tfname = new JTextField();  
  
tfname.setBounds(200, 150, 150, 30);  
  
add(tfname);  
  
  
  
  
javax.swing.JLabel lblrollno = new javax.swing.JLabel("Hotel Id");  
  
lblrollno.setBounds(50, 200, 200, 30);  
  
lblrollno.setFont(new Font("serif", Font.BOLD, 20));  
  
add(lblrollno);  
  
  
labelrollno = new javax.swing.JLabel("1533"+first4);  
  
labelrollno.setBounds(200, 200, 200, 30);  
  
labelrollno.setFont(new Font("serif", Font.BOLD, 20));  
  
add(labelrollno);
```

```
javax.swing.JLabel lbladdress = new javax.swing.JLabel("Address");  
lbladdress.setBounds(50, 250, 200, 30);  
lbladdress.setFont(new Font("serif", Font.BOLD, 20));  
add(lbladdress);  
  
tfaddress = new JTextField();  
tfaddress.setBounds(200, 250, 150, 30);  
add(tfaddress);  
  
javax.swing.JLabel lblphone = new javax.swing.JLabel("Phone");  
lblphone.setBounds(400, 250, 200, 30);  
lblphone.setFont(new Font("serif", Font.BOLD, 20));  
add(lblphone);  
  
tfphone = new JTextField();  
tfphone.setBounds(600, 250, 150, 30);  
add(tfphone);  
  
javax.swing.JLabel lblgstin = new javax.swing.JLabel("Gst no. :");  
lblgstin.setBounds(400, 150, 200, 30);  
lblgstin.setFont(new Font("serif", Font.BOLD, 20));  
add(lblgstin);  
  
tfgstin = new JTextField();
```

```
tfgstin.setBounds(600, 150, 150, 30);

add(tfgstin);

javax.swing.JLabel lblemail = new javax.swing.JLabel("Email Id");
lblemail.setBounds(50, 300, 200, 30);
lblemail.setFont(new Font("serif", Font.BOLD, 20));
add(lblemail);

tfemail = new JTextField();
tfemail.setBounds(200, 300, 150, 30);
add(tfemail);

javax.swing.JLabel lblaadhar = new javax.swing.JLabel("Aadhar
Number");
lblaadhar.setBounds(400, 350, 200, 30);
lblaadhar.setFont(new Font("serif", Font.BOLD, 20));
add(lblaadhar);

tfaadhar = new JTextField();
tfaadhar.setBounds(600, 350, 150, 30);
add(tfaadhar);

javax.swing.JLabel lblcourse = new javax.swing.JLabel("Area");
lblcourse.setBounds(50, 400, 200, 30);
lblcourse.setFont(new Font("serif", Font.BOLD, 20));
add(lblcourse);
```

```
String course[] = {"Pune", "Pcmc", "alandi", "vishrantwadi",
"Lavasa", "Magarpatta", "Nagar", "Bhosari", "chakan", "Karvenagar"};
```

```
cbcourse = new JComboBox(course);
```

```
cbcourse.setBounds(200, 400, 150, 30);
```

```
cbcourse.setBackground(Color.WHITE);
```

```
add(cbcourse);
```



```
submit = new JButton("Submit");
```

```
submit.setBounds(250, 550, 120, 30);
```

```
submit.setBackground(Color.BLACK);
```

```
submit.setForeground(Color.WHITE);
```

```
submit.addActionListener(this);
```

```
submit.setFont(new Font("Tahoma", Font.BOLD, 15));
```

```
add(submit);
```



```
cancel = new JButton("Cancel");
```

```
cancel.setBounds(450, 550, 120, 30);
```

```
cancel.setBackground(Color.BLACK);
```

```
cancel.setForeground(Color.WHITE);
```

```
cancel.addActionListener(this);
```

```
cancel.setFont(new Font("Tahoma", Font.BOLD, 15));
```

```
add(cancel);
```



```
setVisible(true);
```

```
}
```

```
public void actionPerformed(ActionEvent ae) {  
    if (ae.getSource() == submit) {  
        String name = tfname.getText();  
        // String fname = tffname.getText();  
        String Hotelid = labelrollno.getText();  
        // String dob = ((JTextField)  
dcdob.getDateEditor().getUiComponent()).getText();  
        String address = tfaddress.getText();  
        String phone = tfphone.getText();  
        String GstIn = tfgstin.getText();  
        String emailid = tfemail.getText();  
  
        String adharno = tfaadhar.getText();  
        String area = (String) cbcourse.getSelectedItem();  
  
        try {  
  
            String query = "INSERT INTO Hotel (name, Hotelid, address, phone, emailid,  
adharno, area, GstIn) VALUES ('"+name+"', '"+Hotelid+"', '"+address+"',  
 '"+phone+"', '"+emailid+"', '"+adharno+"', '"+area+"', '"+GstIn+"')";  
  
            Conn con = new Conn();  
            con.s.executeUpdate(query);  
  
            JOptionPane.showMessageDialog(null, "Hotels Details  
Inserted Successfully");  
            setVisible(false);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    } else {  
        setVisible(false);  
    }  
  
}  
  
public static void main(String[] args) {  
    new AddHotel();  
}  
}  
//AddNeeds.java class
```

```
package management;

import javax.swing.*;
import java.awt.*;
import java.util.*;
import java.awt.event.*;

public class AddNeeds extends JFrame implements ActionListener{

    JTextField tfname, tffname, tfaddress, tfphone, tfemail, tfx, tfxii,
    tfaadhar ,tffood , tfgstin,tflandmarks,tfroad,    tffoodname;

    javax.swing.JLabel labelempId ,  labelneedId;

    JComboBox cbcourse, cbbranch;

    JButton submit, cancel;
```

```
Random ran = new Random();

long first4 = Math.abs((ran.nextLong() % 9000L) + 1000L);

AddNeeds() {
    setLayout(null);

    setSize(900, 700);

    setLocation(350, 50);

    heading = new javax.swing.JLabel("Peoples
Details");

    heading.setBounds(310, 30, 500, 50);

    heading.setFont(new Font("serif", Font.BOLD, 30));

    add(heading);
}
```

```
javax.swing.JLabel lblname = new javax.swing.JLabel("Name");

lblname.setBounds(50, 150, 100, 30);

lblname.setFont(new Font("serif", Font.BOLD, 20));

add(lblname);

tfname = new JTextField();

tfname.setBounds(200, 150, 150, 30);

add(tfname);

javax.swing.JLabel lblrollno = new javax.swing.JLabel("need Id");

lblrollno.setBounds(50, 200, 200, 30);

lblrollno.setFont(new Font("serif", Font.BOLD, 20));

add(lblrollno);
```

```
labelneedId = new javax.swing.JLabel("1533"+first4);

labelneedId.setBounds(200, 200, 200, 30);

labelneedId.setFont(new Font("serif", Font.BOLD, 20));

add(labelneedId);

javax.swing.JLabel lbladdress = new javax.swing.JLabel("Address");

lbladdress.setBounds(50, 250, 200, 30);

lbladdress.setFont(new Font("serif", Font.BOLD, 20));

add(lbladdress);

tfaddress = new JTextField();

tfaddress.setBounds(200, 250, 150, 30);

add(tfaddress);
```

```
javax.swing.JLabel lblphone = new javax.swing.JLabel("Total adult  
");  
  
lblphone.setBounds(400, 250, 200, 30);  
  
lblphone.setFont(new Font("serif", Font.BOLD, 20));  
  
add(lblphone);  
  
  
  
  
tfphone = new JTextField();  
  
tfphone.setBounds(600, 250, 150, 30);  
  
add(tfphone);  
  
  
  
  
javax.swing.JLabel lbgstin = new javax.swing.JLabel("Total  
Family Members : ");  
  
lbgstin.setBounds(400, 150, 200, 30);
```

```
lblgstin.setFont(new Font("serif", Font.BOLD, 20));  
  
add(lblgstin);  
  
tfgstin = new JTextField();  
  
tfgstin.setBounds(600, 150, 150, 30);  
  
add(tfgstin);  
  
javax.swing.JLabel lblemail = new javax.swing.JLabel("Total kids  
:");  
  
lblemail.setBounds(50, 300, 200, 30);  
  
lblemail.setFont(new Font("serif", Font.BOLD, 20));  
  
add(lblemail);  
  
tfemail = new JTextField();  
  
tfemail.setBounds(200, 300, 150, 30);
```

```
add(tfemail);

javax.swing.JLabel lblcourse = new javax.swing.JLabel("Area");

lblcourse.setBounds(50, 400, 200, 30);

lblcourse.setFont(new Font("serif", Font.BOLD, 20));

add(lblcourse);

String course[] = {"Pune", "Pcmc", "alandi", "vishrantwadi",
"Lavasa", "Magarpatta", "Nagar", "Bhosari", "chakan", "Karvenagar"};

cbcourse = new JComboBox(course);

cbcourse.setBounds(200, 400, 150, 30);

cbcourse.setBackground(Color.WHITE);

add(cbcourse);
```

```
submit = new JButton("Submit");

submit.setBounds(250, 550, 120, 30);

submit.setBackground(Color.BLACK);

submit.setForeground(Color.WHITE);

submit.addActionListener(this);

submit.setFont(new Font("Tahoma", Font.BOLD, 15));

add(submit);

cancel = new JButton("Cancel");

cancel.setBounds(450, 550, 120, 30);

cancel.setBackground(Color.BLACK);

cancel.setForeground(Color.WHITE);

cancel.addActionListener(this);

cancel.setFont(new Font("Tahoma", Font.BOLD, 15));

add(cancel);
```

```
        setVisible(true);

    }

public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == submit) {
        String name = tfname.getText();
        String needid = labelneedId.getText();
        ;
        String address = tfaddress.getText();
        String noOfFamilyMembers= tfphone.getText();
        String adult = tfgstin.getText();
        String kids = tfemail.getText();
    }
}
```

```
String area = (String) cbcourse.getSelectedItem();  
  
try {  
  
    String query = "INSERT INTO people (name, needid, address, FamilyMembers,  
adult,kids, area) VALUES ('"+name+"', '"+needid+"', '"+address+"',  
'"+noOfFamilyMembers+"', '"+adult+"', '"+kids+"', '"+area+"')";  
  
    Conn con = new Conn();  
  
    con.s.executeUpdate(query);  
  
    JOptionPane.showMessageDialog(null, "Need Details  
Inserted Successfully");  
  
    setVisible(false);
```

```
        } catch (Exception e) {  
  
            e.printStackTrace();  
  
        }  
  
    } else {  
  
        setVisible(false);  
  
    }  
  
}  
  
}  
  
public static void main(String[] args) {  
  
    new AddNeeds();  
  
}  
  
}  
  
//HotelDetails.java class  
  
package management;
```

```
import javax.swing.*;  
import java.awt.*;  
import java.sql.*;  
  
import java.awt.event.*;  
import javax.swing.table.DefaultTableModel;  
  
public class HotelDetails extends JFrame implements ActionListener {  
  
    Choice cId;  
    JTable table;  
    JButton search, print, update, add, cancel;  
  
    HotelDetails() {  
  
        getContentPane().setBackground(Color.WHITE);  
        setLayout(null);  
  
        javax.swing.JLabel heading = new javax.swing.JLabel("Search by  
Hotel Id");  
        heading.setBounds(20, 20, 150, 20);  
        add(heading);  
  
        cId = new Choice();  
        cId.setBounds(180, 20, 150, 20);  
        add(cId);  
  
        try {
```

```
Conn c = new Conn();

ResultSet rs = c.s.executeQuery("select * from hotel");

while(rs.next()) {

    cId.add(rs.getString("Hotelid"));

}

} catch (Exception e) {

    e.printStackTrace();

}

table = new JTable();

try {

    Conn c = new Conn();

    ResultSet rs = c.s.executeQuery("select * from hotel");

}

} catch (Exception e) {

    e.printStackTrace();

}

JScrollPane jsp = new JScrollPane(table);

jsp.setBounds(0, 100, 900, 600);

add(jsp);

search = new JButton("Search");

search.setBounds(20, 70, 80, 20);

search.addActionListener(this);

add(search);
```

```
print = new JButton("Print");
print.setBounds(120, 70, 80, 20);
print.addActionListener(this);
add(print);

add = new JButton("Add");
add.setBounds(220, 70, 80, 20);
add.addActionListener(this);
add(add);

update = new JButton("Update");
update.setBounds(320, 70, 80, 20);
update.addActionListener(this);
add(update);

cancel = new JButton("Cancel");
cancel.setBounds(420, 70, 80, 20);
cancel.addActionListener(this);
add(cancel);

setSize(900, 700);
 setLocation(300, 100);
setVisible(true);
}
```

```
public void actionPerformed(ActionEvent ae) {  
  
    if (ae.getSource() == search) {  
  
        String selectedId = cId.getSelectedItem();  
  
        if (selectedId != null && !selectedId.isEmpty()) {  
  
            String query = "select * from hotel where Hotelid = '" +  
selectedId + "'";  
  
            try {  
  
                Conn c = new Conn();  
  
                ResultSet rs = c.s.executeQuery(query);  
  
                // Create a table model to hold the data  
  
                DefaultTableModel model = new DefaultTableModel();  
  
                ResultSetMetaData metaData = rs.getMetaData();  
  
                int columnCount = metaData.getColumnCount();  
  
                // Add column names to the table model  
  
                for (int columnIndex = 1; columnIndex <= columnCount;  
columnIndex++) {  
  
                    model.addColumn(metaData.getColumnName(columnIndex));  
  
                }  
  
                // Add rows to the table model  
  
                while (rs.next()) {  
  
                    Object[] row = new Object[columnCount];  
  
                    for (int columnIndex = 1; columnIndex <= columnCount;  
columnIndex++) {  
  
                        row[columnIndex - 1] = rs.getObject(columnIndex);  
  
                    }  
  
                    model.addRow(row);  
  
                }  
  
                // Set the table model to the JTable  
            }  
        }  
    }  
}
```

```
        table.setModel(model);

    } catch (Exception e) {
        e.printStackTrace();
    }

} else {
    JOptionPane.showMessageDialog(this, "Please select a hotel
ID.");
}

} else if (ae.getSource() == print) {
    try {
        table.print();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

} else if (ae.getSource() == add) {
    setVisible(false);
    new AddHotel();
}

} else if (ae.getSource() == update) {
    setVisible(false);
    new UpdateHotel();
}

} else {
    setVisible(false);
}

}

public static void main(String[] args) {
    new HotelDetails();
}
```

```
    }  
}  
  
//NeedDetails.java class
```

```
package management;

import javax.swing.*;
import java.awt.*;
import java.sql.*;
import java.awt.event.*;
import javax.swing.table.DefaultTableModel;

public class NeedDetails extends JFrame implements ActionListener {

    Choice cId;
    JTable table;
    JButton search, print, update, add, cancel;

    NeedDetails() {
```

```
getContentPane().setBackground(Color.WHITE);

setLayout(null);

javax.swing.JLabel heading = new javax.swing.JLabel("Search by
need Id");

heading.setBounds(20, 20, 150, 20);

add(heading);

cId = new Choice();

cId.setBounds(180, 20, 150, 20);

add( cId );

try {

Conn c = new Conn();

ResultSet rs = c.s.executeQuery("select * from people");
}
```

```
while(rs.next()) {  
  
    cId.add(rs.getString("needid"));  
  
}  
  
} catch (Exception e) {  
  
    e.printStackTrace();  
  
}  
  
table = new JTable();  
  
try {  
  
    Conn c = new Conn();  
  
    ResultSet rs = c.s.executeQuery("select * from people");  
  
} catch (Exception e) {  
  
    e.printStackTrace();  
}
```

```
}

JScrollPane jsp = new JScrollPane(table);

jsp.setBounds(0, 100, 900, 600);

add(jsp);

search = new JButton("Search");

search.setBounds(20, 70, 80, 20);

search.addActionListener(this);

add(search);

print = new JButton("Print");

print.setBounds(120, 70, 80, 20);

print.addActionListener(this);

add(print);
```

```
add = new JButton("Add");

add.setBounds(220, 70, 80, 20);

add.addActionListener(this);

add(add);

update = new JButton("Update");

update.setBounds(320, 70, 80, 20);

update.addActionListener(this);

add(update);

cancel = new JButton("Cancel");

cancel.setBounds(420, 70, 80, 20);

cancel.addActionListener(this);

add(cancel);
```

```
        setSize(900, 700);

        setLocation(300, 100);

        setVisible(true);

    }

public void actionPerformed(ActionEvent ae) {

    if (ae.getSource() == search) {

        String selectedId = cId.getSelectedItem();

        if (selectedId != null && !selectedId.isEmpty()) {

            String query = "select * from people where needid = '" +
selectedId + "'";

            try {

                Conn c = new Conn();

                ResultSet rs = c.s.executeQuery(query);
```

```
// Create a table model to hold the data

DefaultTableModel model = new DefaultTableModel();

ResultSetMetaData metaData = rs.getMetaData();

int columnCount = metaData.getColumnCount();

// Add column names to the table model

for (int columnIndex = 1; columnIndex <= columnCount;
columnIndex++) {

    model.addColumn(metaData.getColumnName(columnIndex));

}

// Add rows to the table model

while (rs.next()) {

    Object[] row = new Object[columnCount];

    for (int columnIndex = 1; columnIndex <= columnCount;
columnIndex++) {

        row[columnIndex - 1] = rs.getObject(columnIndex);

    }

}
```

```
        model.addRow(row);

    }

    // Set the table model to the JTable

    table.setModel(model);

} catch (Exception e) {

    e.printStackTrace();

}

} else {

    JOptionPane.showMessageDialog(this, "Please select a need ID.");

}

} else if (ae.getSource() == print) {

    try {

        table.print();

    } catch (Exception e) {

        e.printStackTrace();

    }

}
```

```
    }

} else if (ae.getSource() == add) {

    setVisible(false);

    new AddNeeds();

} else if (ae.getSource() == update) {

    setVisible(false);

    new UpdateNeedDetails();

} else {

    setVisible(false);

}

}

public static void main(String[] args) {

    new NeedDetails();
}
```

```
}

}

//UpdateHotel.java class

package management;

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

public class UpdateHotel extends JFrame implements ActionListener{

    JTextField tfcourse, tfaddress, tfphone, tfemail, tfbranch;
    javax.swing.JLabel labelEmpId;
    JButton submit, cancel;
    Choice cEmpId;

    UpdateHotel () {

        setSize(900, 650);
        setLocation(350, 50);

        setLayout(null);

        javax.swing.JLabel heading = new javax.swing.JLabel("Update Hotel
Details");
    }
}
```

```
heading.setBounds(50, 10, 500, 50);

heading.setFont(new Font("Tahoma", Font.ITALIC, 35));

add(heading);

javax.swing.JLabel lblrollnumber = new javax.swing.JLabel("Select
Hotel Id");

lblrollnumber.setBounds(50, 100, 200, 20);

lblrollnumber.setFont(new Font("serif", Font.PLAIN, 20));

add(lblrollnumber);

cEmpId = new Choice();

cEmpId.setBounds(250, 100, 200, 20);

add(cEmpId);

try {

    Conn c = new Conn();

    ResultSet rs = c.s.executeQuery("select * from hotel");

    while(rs.next()) {

        cEmpId.add(rs.getString("Hotelid"));

    }

} catch (Exception e) {

    e.printStackTrace();

}

javax.swing.JLabel lblname = new javax.swing.JLabel("Name");

lblname.setBounds(50, 150, 100, 30);

lblname.setFont(new Font("serif", Font.BOLD, 20));
```

```
add(lblname);

javax.swing.JLabel labelname = new javax.swing.JLabel();
labelname.setBounds(200, 150, 150, 30);
labelname.setFont(new Font("Tahoma", Font.PLAIN, 18));
add(labelname);

javax.swing.JLabel lblrollno = new javax.swing.JLabel("Hotel
Id");
lblrollno.setBounds(50, 200, 200, 30);
lblrollno.setFont(new Font("serif", Font.BOLD, 20));
add(lblrollno);

labelEmpId = new javax.swing.JLabel();
labelEmpId.setBounds(200, 200, 200, 30);
labelEmpId.setFont(new Font("Tahoma", Font.PLAIN, 18));
add(labelEmpId);

javax.swing.JLabel lbladdress = new javax.swing.JLabel("Address");
lbladdress.setBounds(50, 250, 200, 30);
lbladdress.setFont(new Font("serif", Font.BOLD, 20));
add(lbladdress);
```

```
tfaddress = new JTextField();
tfaddress.setBounds(200, 250, 150, 30);
add(tfaddress);

javax.swing.JLabel lblphone = new javax.swing.JLabel("Phone");
lblphone.setBounds(400, 250, 200, 30);
lblphone.setFont(new Font("serif", Font.BOLD, 20));
add(lblphone);

tfphone = new JTextField();
tfphone.setBounds(600, 250, 150, 30);
add(tfphone);

javax.swing.JLabel lblemail = new javax.swing.JLabel("Email Id");
lblemail.setBounds(50, 300, 200, 30);
lblemail.setFont(new Font("serif", Font.BOLD, 20));
add(lblemail);

tfemail = new JTextField();
tfemail.setBounds(200, 300, 150, 30);
add(tfemail);

javax.swing.JLabel lblgstin = new javax.swing.JLabel("Gst no. :");
lblgstin.setBounds(400, 150, 200, 30);
lblgstin.setFont(new Font("serif", Font.BOLD, 20));
```

```
add(lblgstin);

JTextField tfgstin = new JTextField();
tfgstin.setBounds(600, 150, 150, 30);
add(tfgstin);

try {
    Conn c = new Conn();
    String query = "select * from hotel where
Hotelid='"+cEmpId.getSelectedItem()+"'";
    ResultSet rs = c.s.executeQuery(query);
    while(rs.next()) {
        labelname.setText(rs.getString("name"));
        tfaddress.setText(rs.getString("address"));
        tfphone.setText(rs.getString("phone"));
        tfemail.setText(rs.getString("emailid"));

        labelEmpId.setText(rs.getString("Hotelid"));
        tfgstin.setText(rs.getString("Gstin"));
    }
} catch (Exception e) {
    e.printStackTrace();
}

cEmpId.addItemListener(new ItemListener() {
```

```
public void itemStateChanged(ItemEvent ie) {  
    try {  
        Conn c = new Conn();  
        String query = "select * from hotel where  
Hotelid='"+cEmpId.getSelectedItem()+"'";  
        ResultSet rs = c.s.executeQuery(query);  
        while(rs.next()) {  
            labelname.setText(rs.getString("name"));  
  
            tfaddress.setText(rs.getString("address"));  
            tfphone.setText(rs.getString("phone"));  
            tfemail.setText(rs.getString("email"));  
  
            labelEmpId.setText(rs.getString("HotelId"));  
            tfgstin.setText(rs.getString("Gstin"));  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}  
  
submit = new JButton("Update");  
submit.setBounds(250, 500, 120, 30);  
submit.setBackground(Color.BLACK);  
submit.setForeground(Color.WHITE);  
submit.addActionListener(this);  
submit.setFont(new Font("Tahoma", Font.BOLD, 15));
```

```
        add(submit);

        cancel = new JButton("Cancel");
        cancel.setBounds(450, 500, 120, 30);
        cancel.setBackground(Color.BLACK);
        cancel.setForeground(Color.WHITE);
        cancel.addActionListener(this);
        cancel.setFont(new Font("Tahoma", Font.BOLD, 15));
        add(cancel);

        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == submit) {
            String hotelId = labelEmpId.getText();
            String address = tfaddress.getText();
            String phone = tfphone.getText();
            String emailid = tfemail.getText();

            try {
                String query = "update hotel set address='"+address+"',
phone='"+phone+"', emailid='"+emailid+"' where Hotelid='"+hotelId+"'";
                Conn con = new Conn();
                con.s.executeUpdate(query);
            }
        }
    }
}
```



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

public class UpdateNeedDetails extends JFrame implements ActionListener{

    JTextField tfcourse, tfaddress, tfphone, tfemail, tfbranch;
    javax.swing.JLabel labelEmpId;
    JButton submit, cancel;
    Choice cEmpId;
    private Object tfgstin;

    UpdateNeedDetails() {
        setSize(900, 650);
        setLocation(350, 50);

        setLayout(null);

        javax.swing.JLabel heading = new javax.swing.JLabel("Update Need
Details");
        heading.setBounds(50, 10, 500, 50);
        heading.setFont(new Font("Tahoma", Font.ITALIC, 35));
        add(heading);

        javax.swing.JLabel lblrollnumber = new javax.swing.JLabel("Select
Need Id");
```

```
lblrollnumber.setBounds(50, 100, 200, 20);

lblrollnumber.setFont(new Font("serif", Font.PLAIN, 20));

add(lblrollnumber);

cEmpId = new Choice();

cEmpId.setBounds(250, 100, 200, 20);

add(cEmpId);

try {

    Conn c = new Conn();

    ResultSet rs = c.s.executeQuery("select * from people");

    while(rs.next()) {

        cEmpId.add(rs.getString("needid"));

    }

} catch (Exception e) {

    e.printStackTrace();

}

javax.swing.JLabel lblname = new javax.swing.JLabel("Name");

lblname.setBounds(50, 150, 100, 30);

lblname.setFont(new Font("serif", Font.BOLD, 20));

add(lblname);

javax.swing.JLabel labelname = new javax.swing.JLabel();

labelname.setBounds(200, 150, 150, 30);

labelname.setFont(new Font("Tahoma", Font.PLAIN, 18));
```

```
add(labelname);

javax.swing.JLabel lblrollno = new javax.swing.JLabel("Need Id");
lblrollno.setBounds(50, 200, 200, 30);
lblrollno.setFont(new Font("serif", Font.BOLD, 20));
add(lblrollno);

labelEmpId = new javax.swing.JLabel();
labelEmpId.setBounds(200, 200, 200, 30);
labelEmpId.setFont(new Font("Tahoma", Font.PLAIN, 18));
add(labelEmpId);

javax.swing.JLabel lbladdress = new javax.swing.JLabel("Address");
lbladdress.setBounds(50, 250, 200, 30);
lbladdress.setFont(new Font("serif", Font.BOLD, 20));
add(lbladdress);

tfaddress = new JTextField();
tfaddress.setBounds(200, 250, 150, 30);
add(tfaddress);

javax.swing.JLabel lblphone = new javax.swing.JLabel("Total
members");
```

```
lblphone.setBounds(400, 250, 200, 30);

lblphone.setFont(new Font("serif", Font.BOLD, 20));

add(lblphone);

tfphone = new JTextField();

tfphone.setBounds(600, 250, 150, 30);

add(tfphone);

javax.swing.JLabel lblemail = new javax.swing.JLabel("Total

adults");

lblemail.setBounds(50, 300, 200, 30);

lblemail.setFont(new Font("serif", Font.BOLD, 20));

add(lblemail);

tfemail = new JTextField();

tfemail.setBounds(200, 300, 150, 30);

add(tfemail);

javax.swing.JLabel lblgstin = new javax.swing.JLabel("Total

kid");

lblgstin.setBounds(400, 150, 200, 30);

lblgstin.setFont(new Font("serif", Font.BOLD, 20));

add(lblgstin);

JTextField tfgstin = new JTextField();

tfgstin.setBounds(600, 150, 150, 30);

add(tfgstin);
```

```
try {  
  
    Conn c = new Conn();  
  
    String query = "select * from people where  
needid='"+cEmpId.getSelectedItem()+"'";  
  
    ResultSet rs = c.s.executeQuery(query);  
  
    while(rs.next()) {  
  
        labelname.setText(rs.getString("name"));  
  
        tfaddress.setText(rs.getString("address"));  
        tfphone.setText(rs.getString("FamilyMembers"));  
        tfemail.setText(rs.getString("Adult"));  
  
        labelEmpId.setText(rs.getString("needid"));  
        tfgstin.setText(rs.getString("kids"));  
  
    }  
}  
}  
  
} catch (Exception e) {  
    e.printStackTrace();  
}  
  
cEmpId.addItemListener(new ItemListener() {  
  
    public void itemStateChanged(ItemEvent ie) {  
        try {  
            Conn c = new Conn();  
            String query = "select * from people where  
needid='"+cEmpId.getSelectedItem()+"'";  
        }  
    }  
}
```

```
ResultSet rs = c.s.executeQuery(query);

while(rs.next()) {
    labelname.setText(rs.getString("name"));

    tfaddress.setText(rs.getString("address"));
    tfphone.setText(rs.getString("FamilyMembers"));
    tfemail.setText(rs.getString("Adult"));

    labelEmpId.setText(rs.getString("needId"));
    tfgstin.setText(rs.getString("Kids"));
}

} catch (Exception e) {
    e.printStackTrace();
}

}

}) ;

submit = new JButton("Update");
submit.setBounds(250, 500, 120, 30);
submit.setBackground(Color.BLACK);
submit.setForeground(Color.WHITE);
submit.addActionListener(this);
submit.setFont(new Font("Tahoma", Font.BOLD, 15));
add(submit);

cancel = new JButton("Cancel");
cancel.setBounds(450, 500, 120, 30);
```

```
cancel.setBackground(Color.BLACK);
cancel.setForeground(Color.WHITE);
cancel.addActionListener(this);
cancel.setFont(new Font("Tahoma", Font.BOLD, 15));
add(cancel);

setVisible(true);

}

public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == submit) {
        String needid = labelEmpId.getText();
        String address = tfaddress.getText();
        String FamilyMembers = tfphone.getText();
        String adult = tfemail.getText();
        // String kid = tfgstin.getText();

        try {
            String query = "update hotel set address='"+address+"',
phone='"+FamilyMembers+"', emailid='"+adult+"' where
Hotelid='"+needid+"'";

            Conn con = new Conn();
            con.s.executeUpdate(query);

            JOptionPane.showMessageDialog(null, "Need Details Updated
Successfully");
            setVisible(false);
        }
    }
}
```

```
        } catch (Exception e) {
            e.printStackTrace();
        }
    } else {
        setVisible(false);
    }
}

public static void main(String[] args) {
    new UpdateNeedDetails();
}
}

//FoodfeeForm.java class
package management;

import javax.swing.*;
import java.awt.*;
import java.sql.*;
import javax.swing.table.DefaultTableModel;

public class FoodFeeForm extends JFrame {

    JTable table;
```

```
JScrollPane jsp;

FoodFeeForm() {
    setSize(1000, 700);
    setLocation(250, 50);
    setLayout(new BorderLayout());
    getContentPane().setBackground(Color.WHITE);

    javax.swing.JLabel heading = new javax.swing.JLabel("Food
Details", SwingConstants.CENTER);
    heading.setFont(new Font("Tahoma", Font.BOLD, 30));
    add(heading, BorderLayout.NORTH);

    table = new JTable();
    jsp = new JScrollPane(table);
    add(jsp, BorderLayout.CENTER);

    loadTableData();

    setVisible(true);
}

private void loadTableData() {
    try {
        Conn c = new Conn();
        ResultSet rs = c.s.executeQuery("select * from items");
        // Get ResultSet metadata
    }
}
```

```
ResultSetMetaData metaData = rs.getMetaData();

int columnCount = metaData.getColumnCount();

// Create table model with column names

String[] columnNames = new String[columnCount];

for (int i = 1; i <= columnCount; i++) {

    columnNames[i - 1] = metaData.getColumnName(i);

}

DefaultTableModel model = new DefaultTableModel(columnNames,
0);

// Populate table model with data

while (rs.next()) {

    Object[] rowData = new Object[columnCount];

    for (int i = 1; i <= columnCount; i++) {

        rowData[i - 1] = rs.getObject(i);

    }

    model.addRow(rowData);

}

table.setModel(model);

// Close ResultSet and Statement

rs.close();

c.s.close();

} catch (Exception e) {

    e.printStackTrace();

}
```

```
}

public static void main(String[] args) {
    new FoodFeeForm();
}

}

//About.java class

package management;

import javax.swing.*;
import java.awt.*;
import javax.swing.JLabel;

public class About extends JFrame {

    About() {
        setSize(700, 500);
        setLocation(400, 150);

        // Creating a JPanel to hold the background image
        JPanel backgroundPanel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {
                super.paintComponent(g);
                ImageIcon backgroundImage = new
ImageIcon(ClassLoader.getSystemResource("icons/about.jpg"));

```

```
        g.drawImage(backgroundImage.getImage(), 0, 0, getWidth(),
getHeight(), this);

    }

};

setContentPane(backgroundPanel);

// Creating other components to be added on top of the background

JLabel heading = new JLabel("<html>Zero<br/>Hunger
System</html>");

heading.setBounds(70, 20, 300, 130);

heading.setFont(new Font("Tahoma", Font.BOLD, 30));

backgroundPanel.add(heading);

JLabel name = new JLabel("Developed By: ");

name.setBounds(70, 220, 550, 40);

name.setFont(new Font("Tahoma", Font.BOLD, 30));

backgroundPanel.add(name);

javax.swing.JLabel rollno1 = new javax.swing.JLabel("Roll number:
UCE2023401");

rollno1.setBounds(70, 270, 550, 40);

rollno1.setFont(new Font("Tahoma", Font.PLAIN, 30));

add(rollno1);

javax.swing.JLabel rollno2 = new javax.swing.JLabel("Roll number:
UCE2023406");

rollno2.setBounds(70, 300, 550, 40);

rollno2.setFont(new Font("Tahoma", Font.PLAIN, 30));

add(rollno2);
```

```
javax.swing.JLabel rollno3 = new javax.swing.JLabel("Roll number:  
UCE2023424");  
  
rollno3.setBounds(70, 330, 550, 40);  
  
rollno3.setFont(new Font("Tahoma", Font.PLAIN, 30));  
  
add(rollno3);  
  
  
javax.swing.JLabel rollno4 = new javax.swing.JLabel("Roll number:  
UCE2023422");  
  
rollno4.setBounds(70, 360, 550, 40);  
  
rollno4.setFont(new Font("Tahoma", Font.PLAIN, 30));  
  
add(rollno4);  
  
// Add more components as needed  
  
  
setLayout(null);  
  
setVisible(true);  
  
}  
  
  
public static void main(String[] args) {  
  
    new About();  
  
}  
  
}  
  
  
//EnterFood.java class  
  
  
package management;
```

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

public class EnterFood extends JFrame implements ActionListener {

    Choice crollno;
    JComboBox cbsemester;
    JTextField tfsub1,
    tfsub2,tfsub3,tfsub4,tfsub5,tfmarks1,tfmarks2,tfmarks3,tfmarks4,tfmarks5;
    JButton cancel, submit;

    EnterFood() {
        setSize(1000, 500);
        setLocation(300, 150);
        setLayout(null);

        getContentPane().setBackground(Color.WHITE);

        ImageIcon i1 = new
        ImageIcon(ClassLoader.getSystemResource("icons/exam.jpg"));

        Image i2 = i1.getImage().getScaledInstance(400, 300,
        Image.SCALE_DEFAULT);

        ImageIcon i3 = new ImageIcon(i2);
        javax.swing.JLabel image = new javax.swing.JLabel(i3);
        image.setBounds(500, 40, 400, 300);
```

```
add(image);

javax.swing.JLabel heading = new javax.swing.JLabel("Enter Food
Details");
heading.setBounds(50, 0, 500, 50);
heading.setFont(new Font("Tahoma", Font.BOLD, 20));
add(heading);

javax.swing.JLabel lblrollnumber = new javax.swing.JLabel("Select
Hotel id Number");
lblrollnumber.setBounds(50, 70, 150, 20);
add(lblrollnumber);

crollno = new Choice();
crollno.setBounds(200, 70, 150, 20);
add(crollno);

try {
    Conn c = new Conn();
    ResultSet rs = c.s.executeQuery("select * from hotel");
    while(rs.next()) {
        crollno.add(rs.getString("Hotelid"));
    }
} catch (Exception e) {
    e.printStackTrace();
}
```

```
javax.swing.JLabel lblsemester = new javax.swing.JLabel("Select  
food course type");  
  
lblsemester.setBounds(50, 110, 150, 20);  
  
add(lblsemester);  
  
  
String semester[] = {"Breakfast", "Drinks", "Main course",  
"lunch"};  
  
cbsemester = new JComboBox(semester);  
  
cbsemester.setBounds(200, 110, 150, 20);  
  
cbsemester.setBackground(Color.WHITE);  
  
add(cbsemester);  
  
  
javax.swing.JLabel lblentersubject = new javax.swing.JLabel("Enter  
food item");  
  
lblentersubject.setBounds(100, 150, 200, 40);  
  
add(lblentersubject);  
  
  
javax.swing.JLabel lblentermarks = new javax.swing.JLabel("Enter  
quantity");  
  
lblentermarks.setBounds(320, 150, 200, 40);  
  
add(lblentermarks);  
  
  
tfsub1 = new JTextField();  
  
tfsub1.setBounds(50, 200, 200, 20);  
  
add(tfsub1);  
  
  
tfsub2 = new JTextField();  
  
tfsub2.setBounds(50, 230, 200, 20);
```

```
add(tfsub2);

tfsub3 = new JTextField();
tfsub3.setBounds(50, 260, 200, 20);
add(tfsub3);

tfsub4 = new JTextField();
tfsub4.setBounds(50, 290, 200, 20);
add(tfsub4);

tfsub5 = new JTextField();
tfsub5.setBounds(50, 320, 200, 20);
add(tfsub5);

tfmarks1 = new JTextField();
tfmarks1.setBounds(250, 200, 200, 20);
add(tfmarks1);

tfmarks2 = new JTextField();
tfmarks2.setBounds(250, 230, 200, 20);
add(tfmarks2);

tfmarks3 = new JTextField();
tfmarks3.setBounds(250, 260, 200, 20);
add(tfmarks3);

tfmarks4 = new JTextField();
```

```
tfmarks4.setBounds(250, 290, 200, 20);

add(tfmarks4);

tfmarks5 = new JTextField();

tfmarks5.setBounds(250, 320, 200, 20);

add(tfmarks5);

submit = new JButton("Submit");

submit.setBounds(70, 360, 150, 25);

submit.setBackground(Color.BLACK);

submit.setForeground(Color.WHITE);

submit.addActionListener(this);

submit.setFont(new Font("Tahoma", Font.BOLD, 15));

add(submit);

cancel = new JButton("Back");

cancel.setBounds(280, 360, 150, 25);

cancel.setBackground(Color.BLACK);

cancel.setForeground(Color.WHITE);

cancel.addActionListener(this);

cancel.setFont(new Font("Tahoma", Font.BOLD, 15));

add(cancel);

setVisible(true);

}

public void actionPerformed(ActionEvent ae) {
```

```
if (ae.getSource() == submit) {  
    try {  
        Conn c = new Conn();  
  
        String query1 = "insert into items  
values('"+crollno.getSelectedItem()+"',  
'"+cbsemester.getSelectedItem()+"', '"+tfsub1.getText()+"',  
'"+tfsub2.getText()+"', '"+tfsub3.getText()+"', '"+tfsub4.getText()+"')";  
  
        String query2 = "insert into quantity  
values('"+crollno.getSelectedItem()+"',  
'"+cbsemester.getSelectedItem()+"', '"+tfmarks1.getText()+"',  
'"+tfmarks2.getText()+"', '"+tfmarks3.getText()+"',  
'"+tfmarks4.getText()+"')";  
  
        c.s.executeUpdate(query1);  
        c.s.executeUpdate(query2);  
  
        JOptionPane.showMessageDialog(null, "details Inserted  
Successfully");  
        setVisible(false);  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    } else {  
        setVisible(false);  
    }  
}  
  
public static void main(String[] args) {  
    new EnterFood();  
}
```

```
}

//steps to connect sqy database to java class

// Register the driver class

//Creating Connection String

//Creating Statemet

//Executing MySql Queries

//Closing the Connection

package management;

import com.sun.jdi.connect.spi.Connection;
import java.beans.Statement;
import java.sql.PreparedStatement;

public class Conn {

    java.sql.Connection c ;
```

```
java.sql.Statement s;

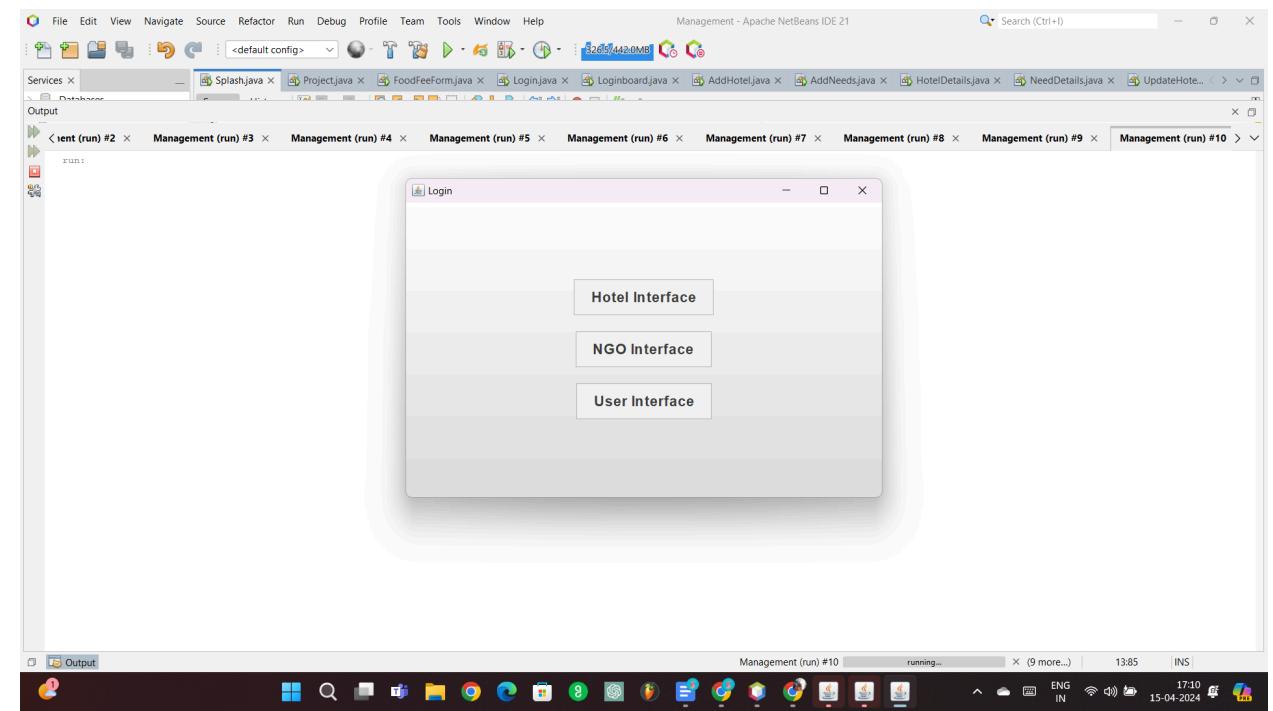
Conn () {
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        c =
java.sql.DriverManager.getConnection("jdbc:mysql:///management", "root",
"vedant@123"); //hotel

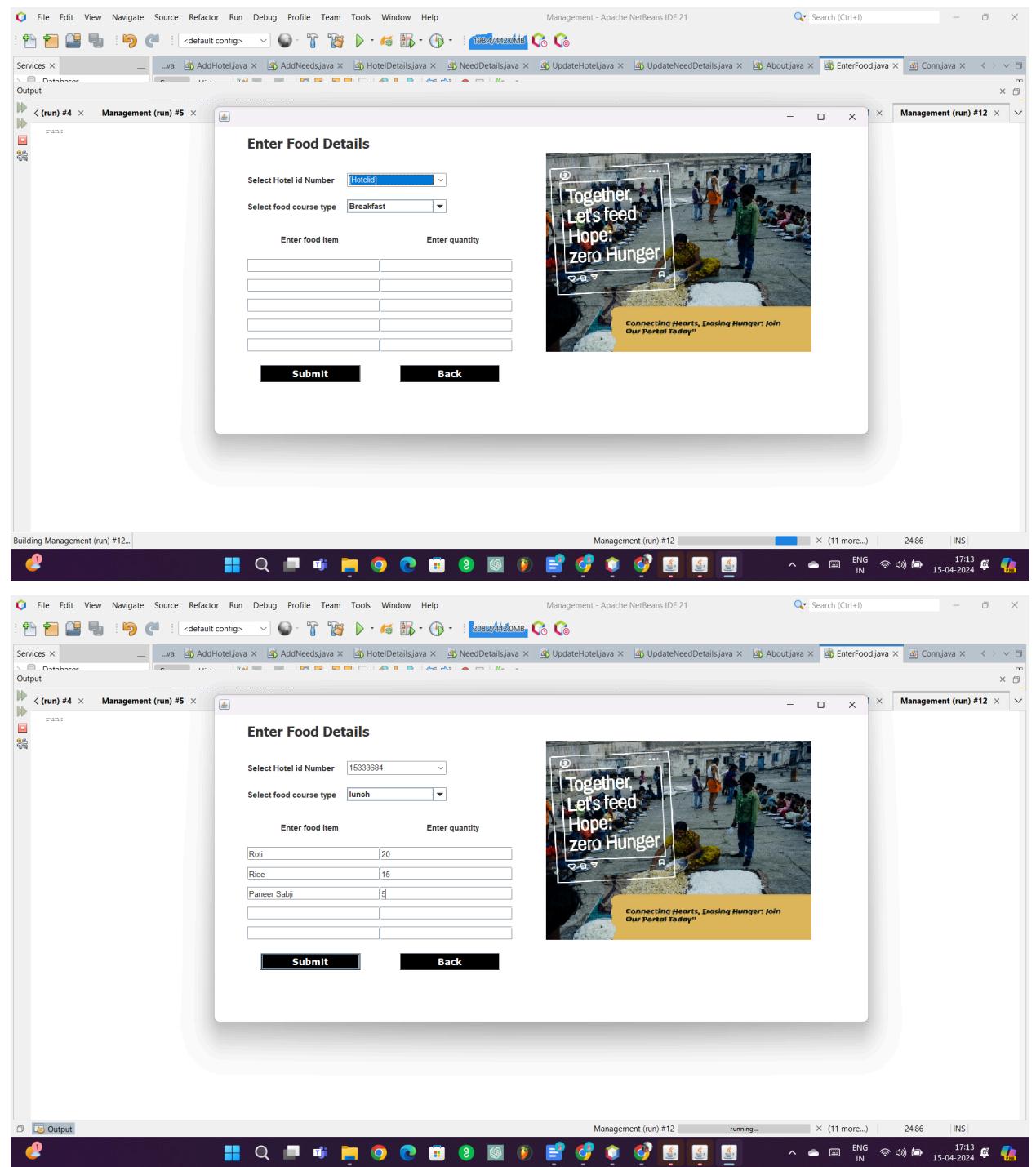
        s = c.createStatement();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

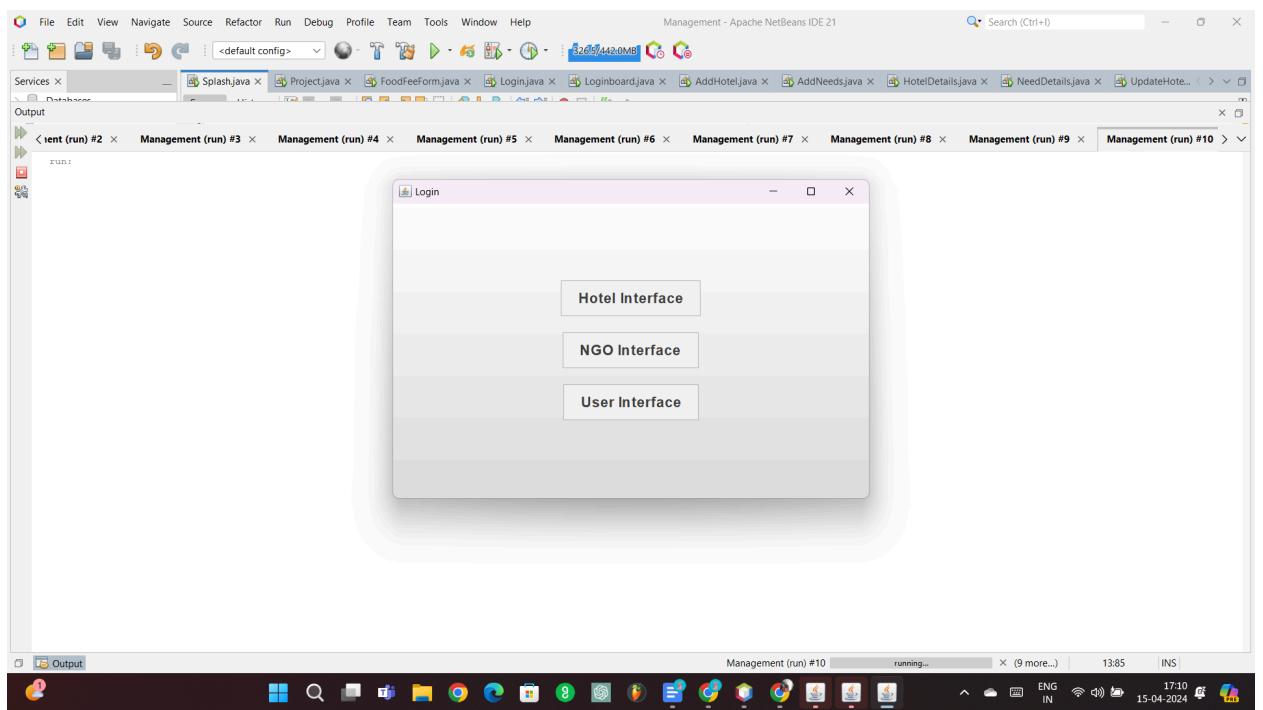
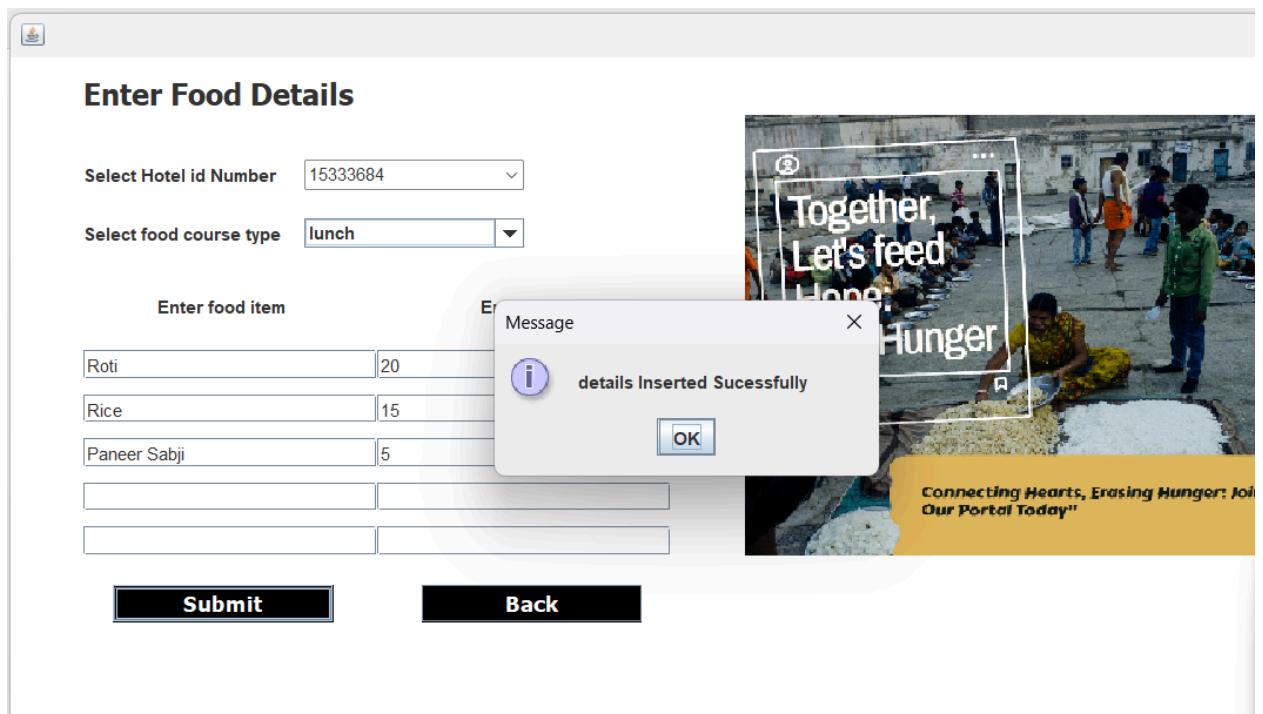
PreparedStatement prepareStatement(String query) {
    throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
}
}
```

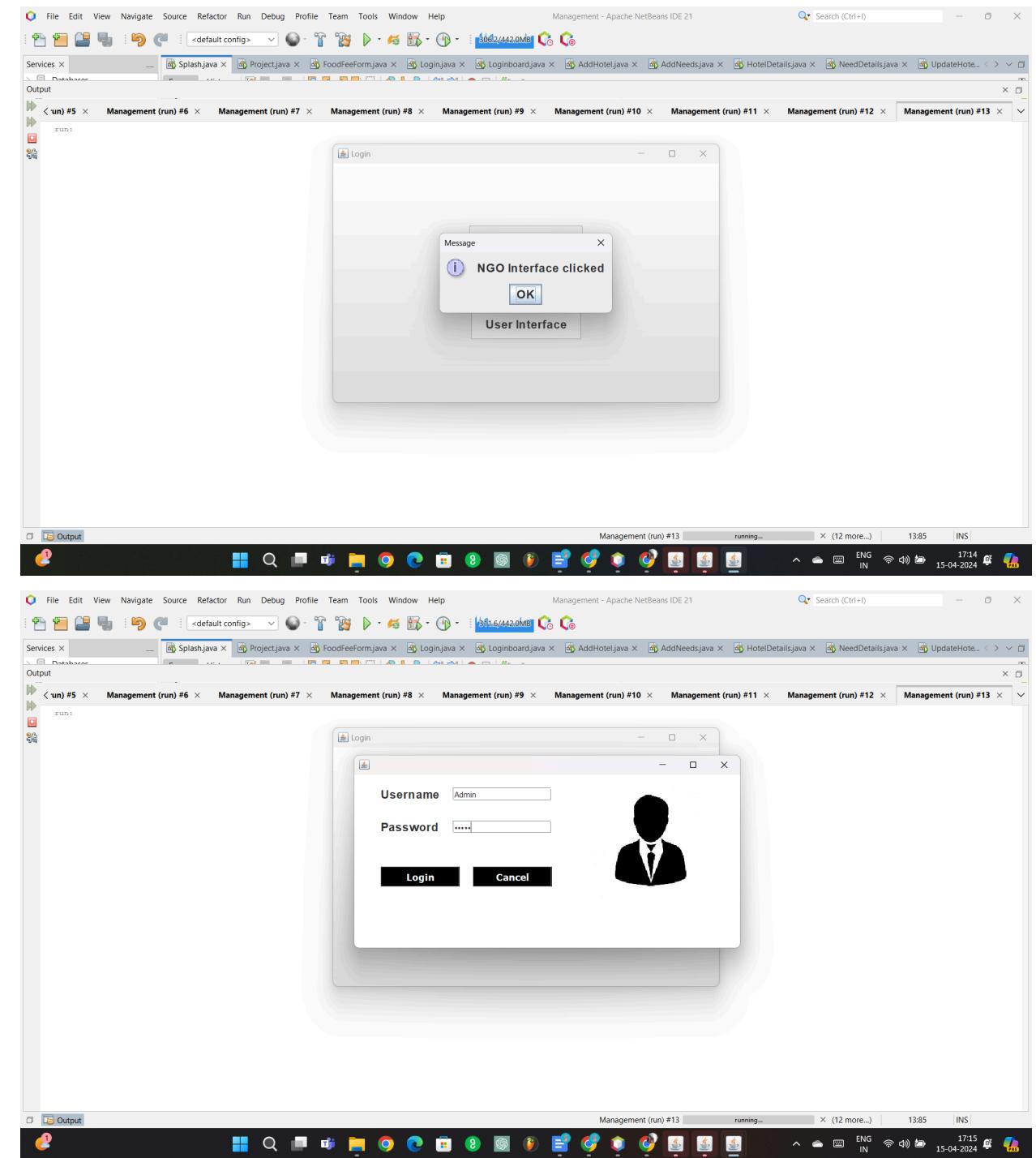
OUTPUT:

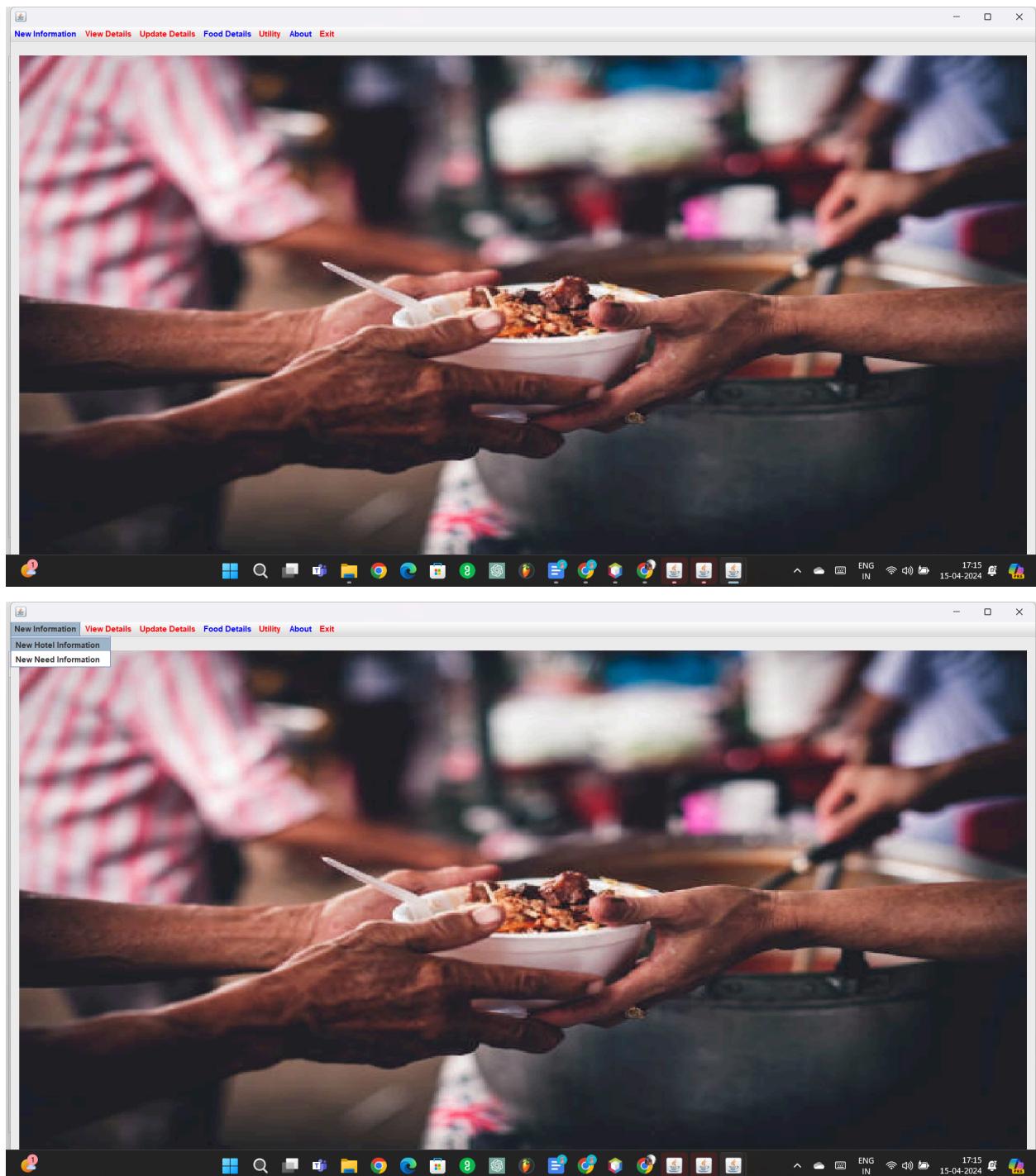


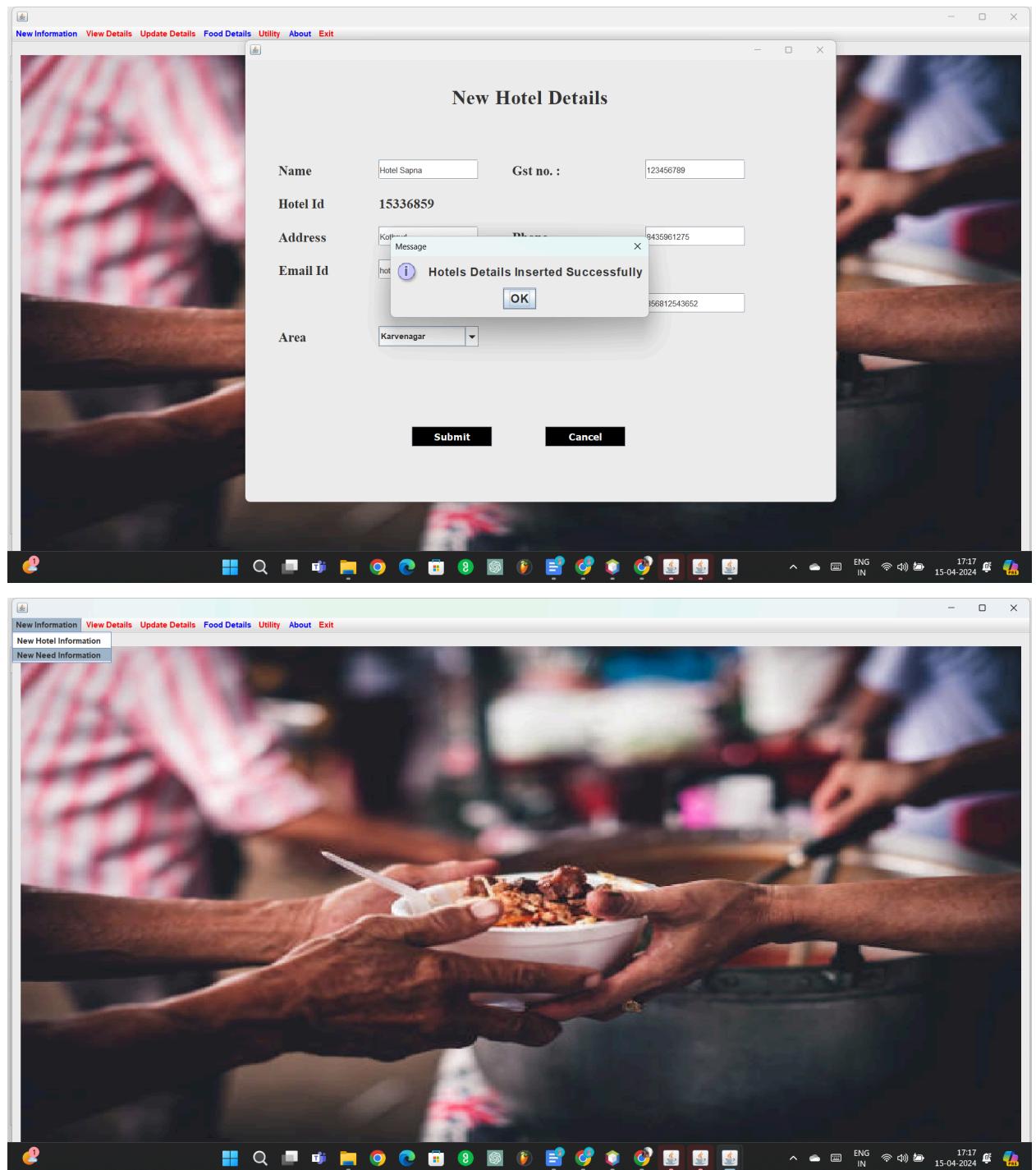


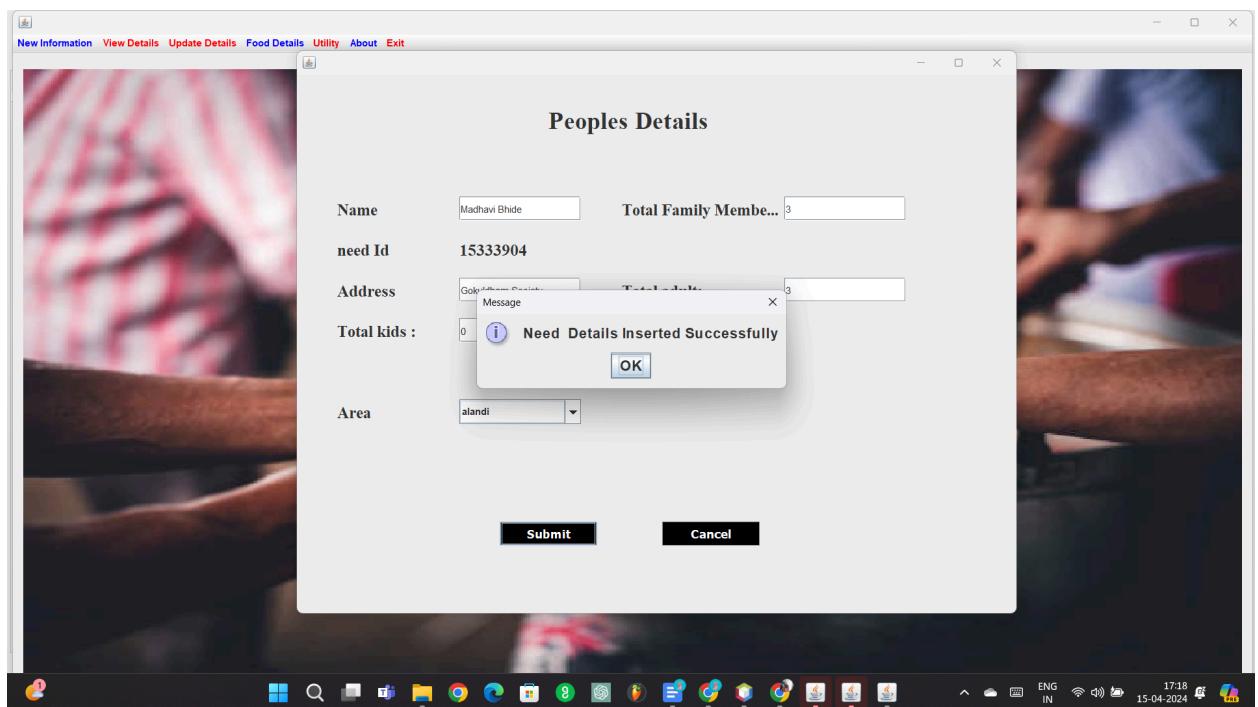
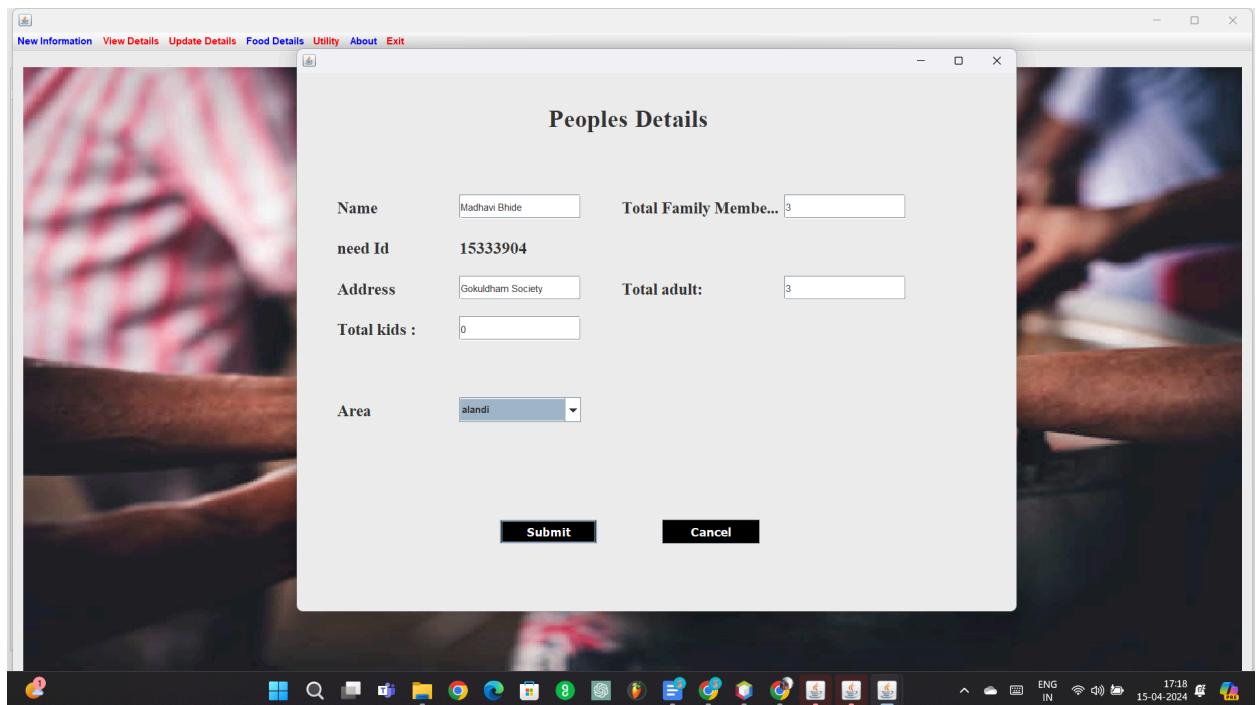


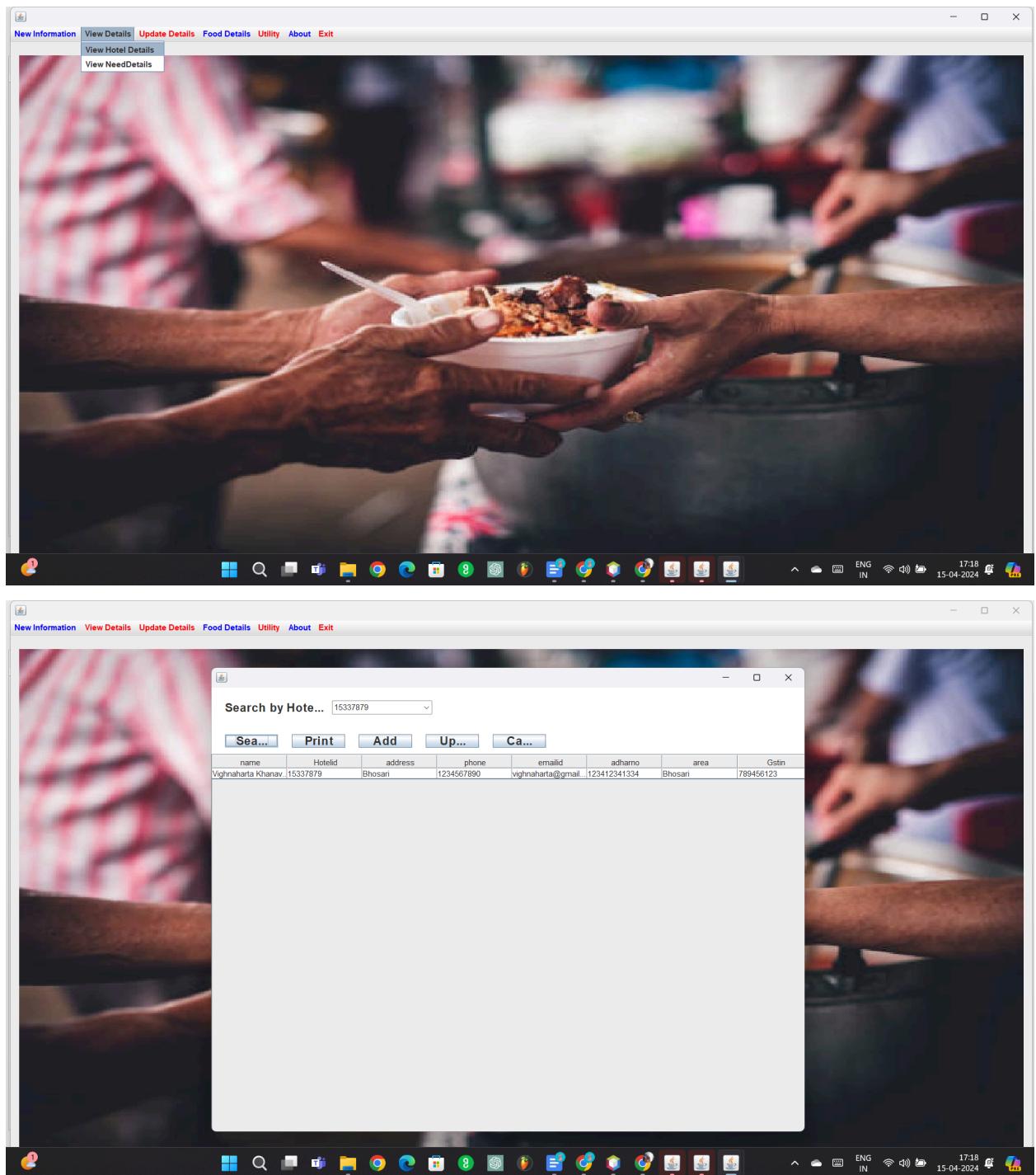








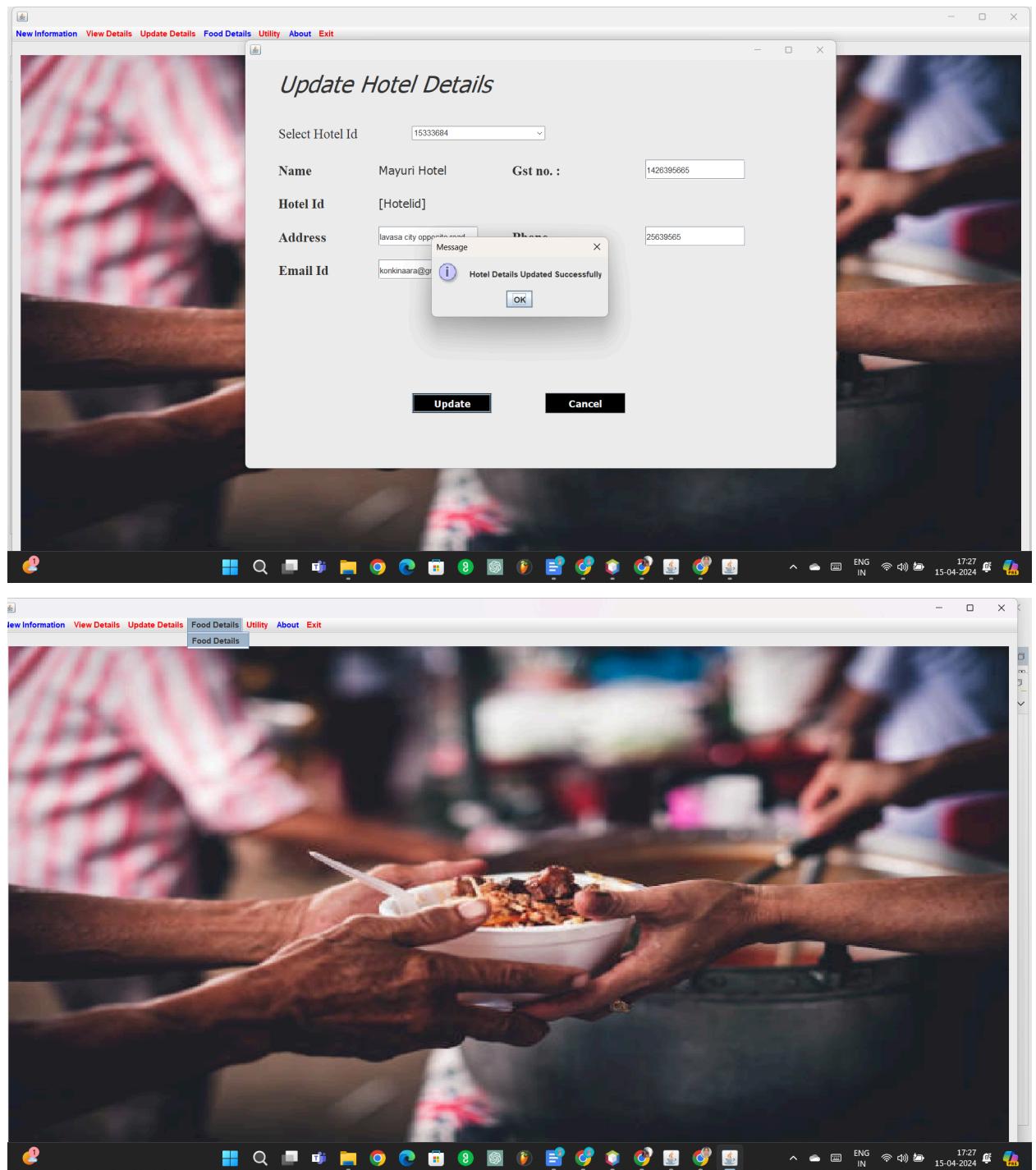




The image shows a Windows desktop environment with two windows open. The top window is a food delivery application with a background image of hands holding a bowl of food. The menu bar includes 'New Information', 'View Details', 'Update Details', 'Food Details', 'Utility', 'About', and 'Exit'. The 'Update Details' option is highlighted. The bottom window is a modal dialog titled 'Update Hotel Details'. It contains the following form fields:

Field	Value
Select Hotel Id	15333684
Name	Mayuri Hotel
Gst no. :	1426395665
Hotel Id	[HotelId]
Address	lavasa city opposite road
Phone	25639565
Email Id	konkinaara@gmail.com

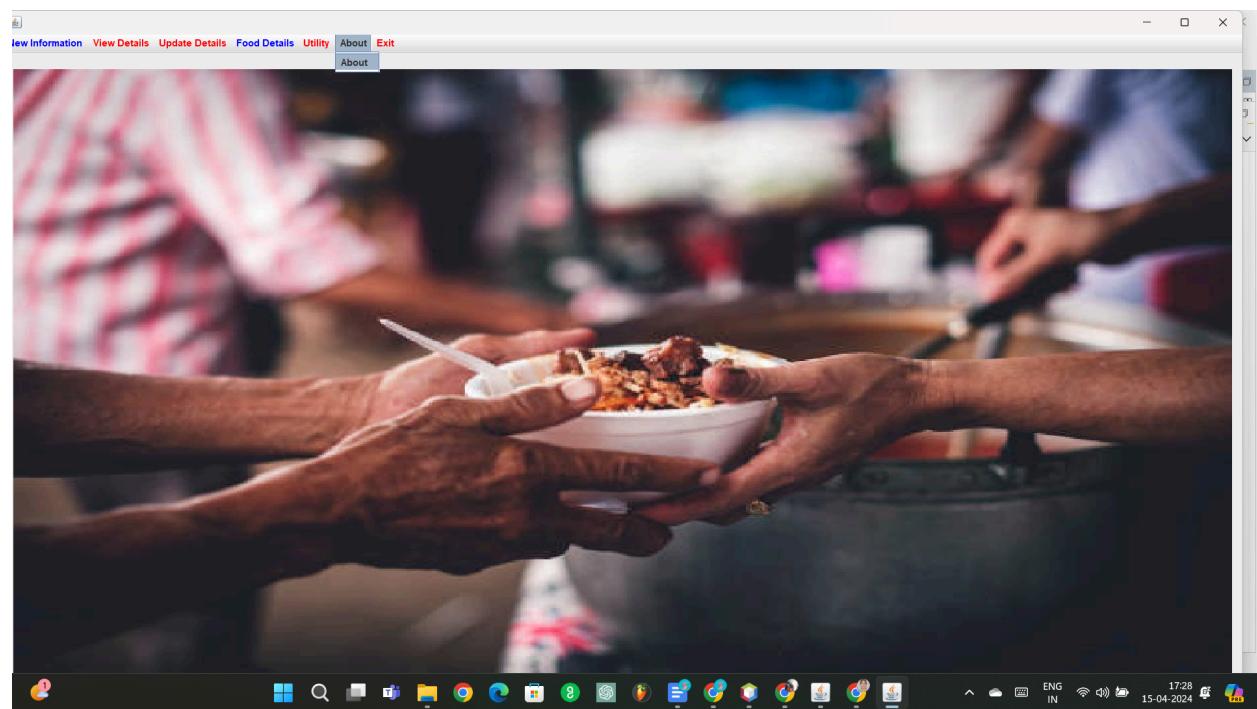
At the bottom of the dialog are 'Update' and 'Cancel' buttons.

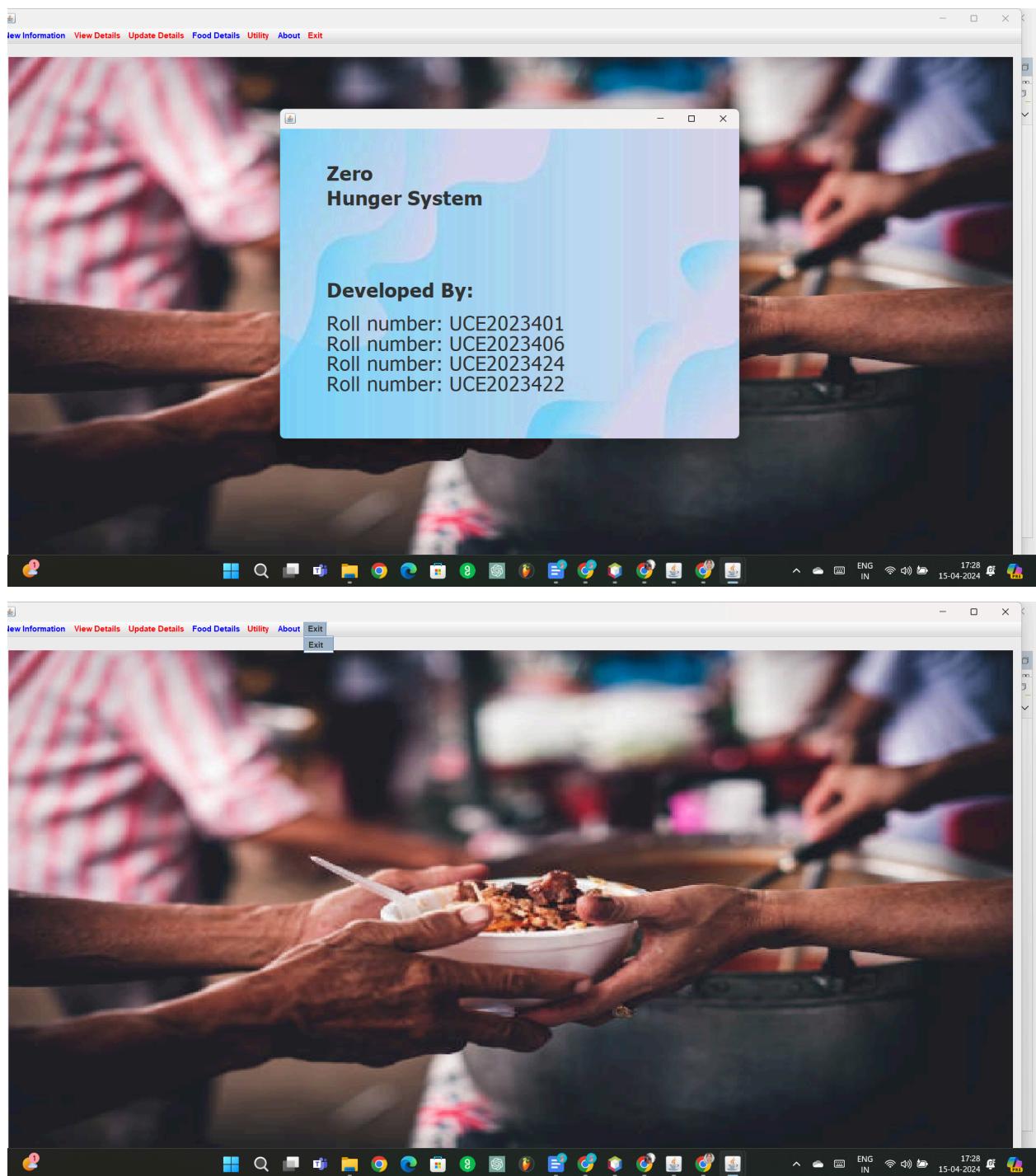


The image shows a Windows desktop environment with two windows open. The top window is titled "Food Details" and contains a table with columns: Hotelid, course, items, item2, item3, and item4. The table data is as follows:

Hotelid	course	items	item2	item3	item4
15334593	Breakfast				
15334593	lunch	roti	sabji	rice	
15336802	Main course	Panier	Veg Maratha	Special Thali	SHAHIPANEER
15331915	Main course	paneer	matar		
15334593	Drinks	soft drinks	cold coffee	milk	
15333684	lunch	Roti	Rice	Paneer Sabji	

The bottom window is partially visible, showing a blurred image of a person's hands holding a bowl of food, likely from a restaurant or cafeteria setting.





DATABASE :

The screenshot shows the MySQL Workbench interface. The main window title is "MySQL Workbench" and the connection is "Local instance MySQL80". The left sidebar has sections for MANAGEMENT (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), INSTANCE (Startup / Shutdown, Server Logs, Options File), and PERFORMANCE (Dashboard, Performance Reports, Performance Schema Setup). The central area has tabs for Navigator, mysilearn*, SQL File 6*, and SqlManagement*. The SQL Management tab contains the following SQL code:

```
1 • show databases;
2 • use management;
3 • show tables from management;
4
5 • select * from hotel;
6 • select * from items;
7 • select * from login;
8 • select * from people;
9 • select * from quantity;
10 • DROP TABLE qnt;
11
12 • DELETE FROM quantity
13 WHERE Hotelid = '15331915';
14
15 • SET SQL_SAFE_UPDATES = 0;
16
17
```

The Output pane shows the results of the executed queries:

Action	Time	Message	Duration / Fetch
show databases	1 17:39:07	12 row(s) returned	0.000 sec / 0.000 sec
use management	2 17:39:11	0 row(s) affected	0.000 sec

The bottom status bar shows the system tray icons and the date/time: 15-04-2024.

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: myslearn SQL File 6 SqlManagement x

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

No object selected

Object Info Session

Output:

#	Time	Action	Message	Duration / Fetch
1	17:39:07	show databases	12 row(s) returned	0.000 sec / 0.000 sec
2	17:39:11	use management	0 row(s) affected	0.000 sec

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: myslearn SQL File 6 SqlManagement x

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

No object selected

Object Info Session

Output:

#	Time	Action	Message	Duration / Fetch
1	17:39:07	show databases	12 row(s) returned	0.000 sec / 0.000 sec
2	17:39:11	use management	0 row(s) affected	0.000 sec

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: myslearn SQL File 6 SqlManagement x

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

No object selected

Object Info Session

Output:

#	Time	Action	Message	Duration / Fetch
1	17:39:07	show databases	12 row(s) returned	0.000 sec / 0.000 sec
2	17:39:11	use management	0 row(s) affected	0.000 sec
3	17:39:27	select * from hotel LIMIT 0, 1000	11 row(s) returned	0.000 sec / 0.000 sec

MySQL Workbench Local instance MySQL80

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Log
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

No object selected

Information

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	17:39:07	show databases	12 row(s) returned	0.000 sec / 0.000 sec
2	17:39:11	use management	0 row(s) affected	0.000 sec
3	17:39:27	select * from hotel LIMIT 0, 1000	11 row(s) returned	0.000 sec / 0.000 sec
4	17:39:44	select * from items LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

Result Grid

HotelId	course	items	item2	item3	item4
15334593	Breakfast	roti	sabji	rice	
15336802	Main course	Paneer	Veg Maratha	Special Thai	SHAHI PANEER
15331915	Main course	paneer	matar		
15334593	Drinks	soft drinks	cold coffee	milk	
1533684	lunch	Roti	Rice	Paneer Sabji	

SQL Additions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

username	password	role
admin	12345	NULL
Aradhana	2024	NULL

Form Editor

Field Types

Query Stats

Object Info Session

40°C Mostly sunny

Search

17:39 15-04-2024 ENG IN

MySQL Workbench Local instance MySQL80

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Log
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

No object selected

Information

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	17:39:07	show databases	12 row(s) returned	0.000 sec / 0.000 sec
2	17:39:11	use management	0 row(s) affected	0.000 sec
3	17:39:27	select * from hotel LIMIT 0, 1000	11 row(s) returned	0.000 sec / 0.000 sec
4	17:39:44	select * from items LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
5	17:40:05	select * from login LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Result Grid

username	password	role
admin	12345	NULL
Aradhana	2024	NULL

SQL Additions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

username	password	role
admin	12345	NULL
Aradhana	2024	NULL

Form Editor

Field Types

Query Stats

Object Info Session

40°C Mostly sunny

Search

17:40 15-04-2024 ENG IN

MySQL Workbench

Local instance MySQL80

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Log
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration

Schemas

No object selected

Information

people 5 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
2	17:39:11	use management	0 row(s) affected	0.000 sec
3	17:39:27	select * from hotel LIMIT 0, 1000	11 row(s) returned	0.000 sec / 0.000 sec
4	17:39:44	select * from items LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
5	17:40:05	select * from login LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
6	17:40:23	select * from people LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

40°C Mostly sunny

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: mysqllearn SQL File 6* SqlManagement

Result Grid | Filter Rows: Export: Wrap Cell Content: Limit to 1000 rows

```

1 • show databases;
2 • use management;
3 • show tables from management;
4
5 • select * from hotel;
```

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: mysqllearn SQL File 6* SqlManagement

Result Grid | Filter Rows: Export: Wrap Cell Content: Limit to 1000 rows

```

1 • show databases;
2 • use management;
3 • show tables from management;
4
5 • select * from hotel;
```

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

THANK YOU