

# **Hotel Booking Form**

**Aryak Lahane**

19302E0033

—

SY.Bsc.It - 3

—

Core Java Mini-Project

---

## DESIGNING

The designing of the project is done using the awt & swing packages of Java.

For the designing purposes various swing and awt classes like JLabel ,TextField , JComboBox ,JRadioButton ,Button.

These were use to get the various functionality inputs from the users like name, contact, address , room type , number of adults etc..

The screenshot displays a Java Swing window titled "Hotel App" with a yellow background. The form contains the following fields and controls:

- First Name** and **Last Name**: Text input fields.
- Address Line 1** and **Address Line 2**: Text input fields.
- City**, **State**, and **Zip Code**: Text input fields.
- Phone number** and **Email**: Text input fields.
- Check-in Month**: A dropdown menu with "January" selected.
- Check-in Date**: A date picker showing "1".
- Check-in Time**: A time picker showing "00 hr".
- Check-out Month**: A dropdown menu with "January" selected.
- Check-out Date**: A date picker showing "1".
- Check-out Time**: A time picker showing "00 hr".
- Room Preference**: Radio buttons for **# Standard**, **Delux**, and **Suite**.
- number of Adults**: A spinner box with "1" selected.
- number of Children**: A spinner box with "0" selected.
- Number of Rooms**: A spinner box with "1" selected.
- Book Now**: A black button with white text.

## THE LAYOUT

### GridBagLayout

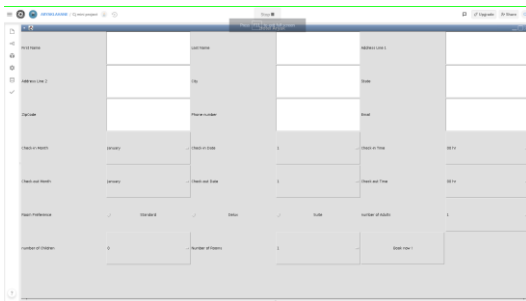
The GridBagLayout was chosen for the layouting purposes cause it gives the versatility while maintaining the structure.

I did various trial and error with other layouts as well such as the GridLayout , the FlowLayout , the Box layout .

- The GridLayout had the structural integrity but it didn't provide the required fluidity .
- The FlowLayout had given the fluidity but it didn't have a spine and it shifted components with the changing in size of the window.
- The BoxLayout was eliminated because it didn't provide any basic functionality and it felt plain.

Hence, the GridBagLayout was Zeroed down on , it did needed some extra bits of code but it provided with the necessary result we wanted

### GridLayout Try :



### FlowLayout Try :



# GridBagLayout Output:

The screenshot shows a Java Swing window titled "Hotel Aryak" with a yellow background. The form is organized using GridBagLayout, with labels on the left and input fields on the right. The labels are in a yellow font, and the input fields are white with black text. The form includes fields for First Name, Last Name, Address Line 1, Address Line 2, City, State, ZipCode, Phone number, Email, Check-in Month, Check-in Date, Check-in Time, Check-out Month, Check-out Date, Check-out Time, Room Preference (Standard, Deluxe, Suite), number of Adults, number of Children, and Number of Rooms. A "Book now!" button is located at the bottom of the form.

First Name		Last Name	
Address Line 1		Address Line 2	
City	State	ZipCode	
Phone number		Email	
Check-in Month	January	Check-in Date	1
Check-in Time		Check-in Time	00 hr
Check-out Month	January	Check-out Date	1
Check-out Time		Check-out Time	00 hr
Room Preference	<input checked="" type="radio"/> Standard <input type="radio"/> Deluxe <input type="radio"/> Suite		
number of Adults	1	number of Children	0
Number of Rooms	1		

Book now!

## RESTRICTIONS & VALIDATIONS

---

### Restrictions :

There are restrictions added to the application with the use of keylistener interface. Here the keylistener is used with the help of adapter class .Java adapter classes provide the default implementation of listener interfaces. If you inherit the adapter class, you will not be forced to provide the implementation of all the methods of listener interfaces. So it saves code.

The Java KeyListener is notified whenever you change the state of key. It is notified against KeyEvent. The KeyListener interface is found in java.awt.event package. It has three methods.

- keyPressed()
- keyReleased()
- keyTyped()

Here , the KeyTyped method is used to apply restrictions in the form for the following purposes:

- Restrict user from entering numbers in the first name & last name TextField.
  - Restrict user from entering numbers in the city & state TextField.
  - Restrict user from entering alphabets in the zipcode and Phone number TextField.
- 

### Validations :

The Validations used in the forms are made from the ActionListener interface. The ActionListener interface consists of the actionPerformed() method. The actionPerformed() method is invoked automatically whenever you click on the registered component. The Validations in actionPerformed() is done with the help of if ladders and switch cases.

Validations in the code are used for :

- Validating that none of the TextFields are left empty.
- Validating that zipcode Textfield consists only of 6 numbers.
- Validating that Phone number TextField consists of only 10 numbers.
- Validating that the selected Check-out date and time is after Check-in .

## EXTRAS

---

- The WindowListner in the program to stop the execution when we click on the x button on the top right of the window to close it.
- The code in such a way to fetch the details and selections given by the user and get them in the console to virtualize the server side i.e. The hotel getting notified of the new booking.

```
----NEW BOOKING---  
Name       : Aryak Lahane  
Address    : haji ismail gani bldg ,Byculla ,  
            Maharastra ,Mumbai :400027  
Phone no   : 9123456780  
Email      : aryakprakashlahane@gmail.com  
Check in   : 1 January 00 hr  
Check out  : 5 January 10 hr  
Adults     : 3  
Children   : 4  
Rooms      : 3  
Room Type  : Standard  
□
```

## CODE

---

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

class Main extends Frame implements ActionListener {
    // JLabel declaration
    JLabel title1, res1, fname1, lname1, add1l, add2l;
    JLabel cname1, sname1, zip1, phone1, email1;
    JLabel cinmonth1, cindatel, cintimel, coutmonth1, coutdatel, couttimel;
    JLabel rpref1, adult1, children1, nroom1, result1;

    // Textfield declaration
    TextField fname1, lname1, add1t, add2t;
    TextField cname1, sname1, zip1, phone1, email1;

    // Combobox declaration

    JComboBox cinmonthc, coutmonthc;
    JComboBox cindatec, coutdatec;
    JComboBox cintimec, couttimec;
    JComboBox adultc, childrenc, nroomc;

    // radiobutton
    JRadioButton standard;
    JRadioButton deluxe;
    JRadioButton suite;

    // button declaration
    Button book;

    //GridBagLayout
    GridBagLayout gb = new GridBagLayout();

    Main() {
```



```

    super("Hotel Aryak");
setBackground(Color.orange);
setSize(400, 400);
setVisible(true);
setLayout(gb);
GridBagConstraints c = new GridBagConstraints();
c.gridx = 0;
c.gridy = 0;
c.fill = GridBagConstraints.BOTH;
c.weightx = 1;
c.weighty = 1;
c.anchor = GridBagConstraints.PAGE_END;
c.insets = new Insets(2,2,2,2);

titlel = new JLabel("HOTEL ARYAK");
fnamel = new JLabel("First Name ");
lnamel = new JLabel("Last Name");
add1l = new JLabel("Address Line 1");
add2l = new JLabel("Address Line 2");
cnamel = new JLabel("City");
snamel = new JLabel("State");
zip1 = new JLabel("ZipCode");
phonel = new JLabel("Phone number");
email1 = new JLabel("Email");
cinmonth1 = new JLabel("Check-in Month");
cindatel = new JLabel("Check-in Date");
cintimel = new JLabel("Check-in Time");
coutmonth1 = new JLabel("Check-out Month");
coutdate1 = new JLabel("Check-out Date");
couttimel = new JLabel("Check-out Time");
rpref1 = new JLabel("Room Preference");
adult1 = new JLabel("number of Adults");
children1 = new JLabel("number of Children");
nroom1 = new JLabel("Number of Rooms");
result1 = new JLabel();
result1.setForeground(Color.red);

// Textfield defination
fnamet = new TextField(20);
lnamet = new TextField(20);
add1t = new TextField(30);
add2t = new TextField(30);
cnamet = new TextField(20);
snamet = new TextField(20);
zipt = new TextField(10);
phonet = new TextField(20);
emailt = new TextField(20);

```



```

// Combobox defination
String month[] = { "January", "February", "March", "April", "May", "June", "July",
"August", "September", "October",
"November", "December" };

String date[] = { "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12",
"13", "14", "15", "16", "17", "18",
"19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31" };

String hour[] = { "00 hr", "01 hr", "02 hr", "03 hr", "04 hr", "05 hr", "06 hr",
"07 hr", "08 hr", "09 hr", "10 hr",
"11 hr", "12 hr", "13 hr", "14 hr", "15 hr", "16 hr", "17 hr", "18 hr", "19
hr", "20 hr", "21 hr", "22 hr",
"23" };

String nadult[] = { "1", "2", "3", "4", "5" };
String nchildren[] = { "0", "1", "2", "3", "4", "5" };

cinmonthc = new JComboBox(month);
coutmonthc = new JComboBox(month);
cindatec = new JComboBox(date);
coutdatec = new JComboBox(date);
cintimec = new JComboBox(hour);
couttimec = new JComboBox(hour);
adultc = new JComboBox(nadult);
childrenc = new JComboBox(nchildren);
nroomc = new JComboBox(nadult);

// JRadioButton Group
standard = new JRadioButton("Standard", true);
delux = new JRadioButton("Delux", false);
suite = new JRadioButton("Suite", false);
ButtonGroup rprefr = new ButtonGroup();
rprefr.add(standard);
rprefr.add(delux);
rprefr.add(suite);

// Button
book = new Button("Book now !");
book.setBackground(Color.black);
book.setForeground(Color.white);

// Adding components to Frame
c.gridx = 0;
c.gridy = 0;

```

```
add(fname1,c);
c.gridx = 1;
c.gridy = 0;
c.gridwidth = 2;
add(fnamet,c);
c.gridx = 3;
c.gridy = 0;
c.gridwidth = 1;
add(lname1,c);
c.gridx = 4;
c.gridy = 0;
c.gridwidth = 2;
add(lnamet,c);
c.gridx = 0;
c.gridy = 1;
c.gridwidth = 1;
add(add1l,c);
c.gridx = 1;
c.gridy = 1;
c.gridwidth = 2;
add(add1t,c);
c.gridx = 3;
c.gridy = 1;
c.gridwidth = 1;
add(add2l,c);
c.gridx = 4;
c.gridy = 1;
c.gridwidth = 2;
add(add2t,c);
c.gridx = 0;
c.gridy = 2;
c.gridwidth = 1;
add(cname1,c);
c.gridx = 1;
c.gridy = 2;
add(cnamet,c);
c.gridx = 2;
c.gridy = 2;
add(sname1,c);
c.gridx = 3;
c.gridy = 2;
add(snamet,c);
c.gridx = 4;
c.gridy = 2;
add(zipl,c);
c.gridx = 5;
c.gridy = 2;
```

```
add(zipt,c);
c.gridx = 0;
c.gridy = 3;
add(phone1,c);
c.gridx = 1;
c.gridy = 3;
c.gridwidth = 2;
add(phonet,c);
c.gridx = 3;
c.gridy = 3;
c.gridwidth = 1;
add(emaill,c);
c.gridx = 4;
c.gridy = 3;
c.gridwidth = 2;
add(emailt,c);
c.gridx = 0;
c.gridy = 4;
c.gridwidth = 1;
add(cinmonth1,c);
c.gridx = 1;
c.gridy = 4;
add(cinmonthc,c);
c.gridx = 2;
c.gridy = 4;
add(cindatel,c);
c.gridx = 3;
c.gridy = 4;
add(cindatec,c);
c.gridx = 4;
c.gridy = 4;
add(cintimel,c);
c.gridx = 5;
c.gridy = 4;
add(cintimec,c);
c.gridx = 0;
c.gridy = 5;
add(coutmonth1,c);
c.gridx = 1;
c.gridy = 5;
add(coutmonthc,c);
c.gridx = 2;
c.gridy = 5;
add(coutdatel,c);
c.gridx = 3;
c.gridy = 5;
add(coutdatec,c);
```

```
c.gridx = 4;
c.gridy = 5;
add(couttimel,c);
c.gridx = 5;
c.gridy = 5;
add(couttimec,c);
c.gridx = 0;
c.gridy = 6;
c.gridwidth = 3;
add(rprefl,c);
c.gridx = 3;
c.gridy = 6;
c.gridwidth = 1;
add(standard,c);
c.gridx = 4;
c.gridy = 6;
add(delux,c);
c.gridx = 5;
c.gridy = 6;
add(suite,c);
c.gridx = 0;
c.gridy = 7;
add(adultl,c);
c.gridx = 1;
c.gridy = 7;
add(adultc,c);
c.gridx = 2;
c.gridy = 7;
add(childrenl,c);
c.gridx = 3;
c.gridy = 7;
add(childrenc,c);
c.gridx = 4;
c.gridy = 7;
add(nrooml,c);
c.gridx = 5;
c.gridy = 7;
add(nroomc,c);
c.gridx = 0;
c.gridy = 8;
c.gridwidth = 6;
add(book,c);
c.gridx = 0;
c.gridy = 9;
add(resultl,c);
```

```
// keylistners
```

```

fnamet.addKeyListener(new KeyAdapter() {
    public void keyTyped(KeyEvent e) {
        char c = e.getKeyChar();
        if (!((c >= 'A') && (c <= 'Z') || (c >= 'a') && (c <= 'z') || (c ==
KeyEvent.VK_BACK_SPACE)
            || (c == KeyEvent.VK_DELETE))) {
            e.consume();
        }
    }
});

```

```

lnamet.addKeyListener(new KeyAdapter() {
    public void keyTyped(KeyEvent e) {
        char c = e.getKeyChar();
        if (!((c >= 'A') && (c <= 'Z') || (c >= 'a') && (c <= 'z') || (c ==
KeyEvent.VK_BACK_SPACE)
            || (c == KeyEvent.VK_DELETE))) {
            e.consume();
        }
    }
});

```

```

cnamet.addKeyListener(new KeyAdapter() {
    public void keyTyped(KeyEvent e) {
        char c = e.getKeyChar();
        if (!((c >= 'A') && (c <= 'Z') || (c >= 'a') && (c <= 'z') || (c ==
KeyEvent.VK_BACK_SPACE)
            || (c == KeyEvent.VK_DELETE))) {
            e.consume();
        }
    }
});

```

```

snamet.addKeyListener(new KeyAdapter() {
    public void keyTyped(KeyEvent e) {
        char c = e.getKeyChar();
        if (!((c >= 'A') && (c <= 'Z') || (c >= 'a') && (c <= 'z') || (c ==
KeyEvent.VK_BACK_SPACE)
            || (c == KeyEvent.VK_DELETE))) {
            e.consume();
        }
    }
});

```

```

zipt.addKeyListener(new KeyAdapter() {
    public void keyTyped(KeyEvent e) {
        char c = e.getKeyChar();

```

```

        if (!(((c >= '0') && (c <= '9')) || (c == KeyEvent.VK_BACK_SPACE) || (c ==
KeyEvent.VK_DELETE))) {
            e.consume();
        }
    }
});

phonet.addKeyListener(new KeyAdapter() {
    public void keyTyped(KeyEvent e) {
        char c = e.getKeyChar();

        if (!(((c >= '0') && (c <= '9')) || (c == KeyEvent.VK_BACK_SPACE) || (c ==
KeyEvent.VK_DELETE))) {
            e.consume();
        }
    }
});

// Window Listner
addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e) {
        dispose();
    }
});

book.addActionListener(this);
}

// ActionListener
public void actionPerformed(ActionEvent e) {
    String fname = fnamet.getText();
    String lname = lname1t.getText();
    String add1 = add1t.getText();
    String add2 = add2t.getText();
    String cname = cname1t.getText();
    String sname = sname1t.getText();
    String zip = zipt.getText();
    String phone = phonet.getText();
    String email = emailt.getText();
    String cinmonth = (String) cinmonthc.getSelectedItem();
    String cindate = (String) cindatec.getSelectedItem();
    String cintime = (String) cintimec.getSelectedItem();
    String coutmonth = (String) coutmonthc.getSelectedItem();
    String coutdate = (String) coutdatec.getSelectedItem();
    String couttime = (String) couttimec.getSelectedItem();
    String adult = (String) adultc.getSelectedItem();
    String children = (String) childrenc.getSelectedItem();

```

```

String nroom = (String) nroomc.getSelectedItem();
String result = "Enter Correct Details :";
int cin = 0;
int cout = 0;

// switch cases for checkin and checkout validity
switch (cinmonth) {
case "January":
    cin = 1000;
    break;
case "February":
    cin = 2000;
    break;
case "March":
    cin = 3000;
    break;
case "April":
    cin = 4000;
    break;
case "May":
    cin = 5000;
    break;
case "June":
    cin = 6000;
    break;
case "July":
    cin = 7000;
    break;
case "August":
    cin = 8000;
    break;
case "September":
    cin = 9000;
    break;
case "October":
    cin = 10000;
    break;
case "November":
    cin = 11000;
    break;
case "December":
    cin = 12000;
    break;
}

switch (cindate) {
case "1":

```



```
    cin = cin + 10;
    break;
case "2":
    cin = cin + 20;
    break;
case "3":
    cin = cin + 30;
    break;
case "4":
    cin = cin + 40;
    break;
case "5":
    cin = cin + 50;
    break;
case "6":
    cin = cin + 60;
    break;
case "7":
    cin = cin + 70;
    break;
case "8":
    cin = cin + 80;
    break;
case "9":
    cin = cin + 90;
    break;
case "10":
    cin = cin + 100;
    break;
case "11":
    cin = cin + 110;
    break;
case "12":
    cin = cin + 120;
    break;
case "13":
    cin = cin + 130;
    break;
case "14":
    cin = cin + 140;
    break;
case "15":
    cin = cin + 150;
    break;
case "16":
    cin = cin + 160;
    break;
```

```
case "17":
    cin = cin + 170;
    break;
case "18":
    cin = cin + 180;
    break;
case "19":
    cin = cin + 190;
    break;
case "20":
    cin = cin + 200;
    break;
case "21":
    cin = cin + 210;
    break;
case "22":
    cin = cin + 220;
    break;
case "23":
    cin = cin + 230;
    break;
case "24":
    cin = cin + 240;
    break;
case "25":
    cin = cin + 250;
    break;
case "26":
    cin = cin + 260;
    break;
case "27":
    cin = cin + 270;
    break;
case "28":
    cin = cin + 280;
    break;
case "29":
    cin = cin + 290;
    break;
case "30":
    cin = cin + 300;
    break;
case "31":
    cin = cin + 310;
    break;
}
```

```
switch (cintime) {  
case "00 hr":  
    cin = cin + 1;  
    break;  
case "01 hr":  
    cin = cin + 2;  
    break;  
case "02 hr":  
    cin = cin + 3;  
    break;  
case "03 hr":  
    cin = cin + 4;  
    break;  
case "04 hr":  
    cin = cin + 5;  
    break;  
case "05 hr":  
    cin = cin + 6;  
    break;  
case "06 hr":  
    cin = cin + 7;  
    break;  
case "07 hr":  
    cin = cin + 8;  
    break;  
case "08 hr":  
    cin = cin + 9;  
    break;  
case "09 hr":  
    cin = cin + 10;  
    break;  
case "10 hr":  
    cin = cin + 11;  
    break;  
case "11 hr":  
    cin = cin + 12;  
    break;  
case "12 hr":  
    cin = cin + 13;  
    break;  
case "13 hr":  
    cin = cin + 14;  
    break;  
case "14 hr":  
    cin = cin + 15;  
    break;  
case "15 hr":
```

```

        cin = cin + 16;
        break;
    case "16 hr":
        cin = cin + 17;
        break;
    case "17 hr":
        cin = cin + 18;
        break;
    case "18 hr":
        cin = cin + 19;
        break;
    case "19 hr":
        cin = cin + 20;
        break;
    case "20 hr":
        cin = cin + 21;
        break;
    case "21 hr":
        cin = cin + 22;
        break;
    case "22 hr":
        cin = cin + 23;
        break;
    case "23 hr":
        cin = cin + 24;
        break;
}

```

```

switch (coutmonth) {
    case "January":
        cout = 1000;
        break;
    case "February":
        cout = 2000;
        break;
    case "March":
        cout = 3000;
        break;
    case "April":
        cout = 4000;
        break;
    case "May":
        cout = 5000;
        break;
    case "June":
        cout = 6000;
        break;
}

```

```

case "July":
    cout = 7000;
    break;
case "August":
    cout = 8000;
    break;
case "September":
    cout = 9000;
    break;
case "October":
    cout = 10000;
    break;
case "November":
    cout = 11000;
    break;
case "December":
    cout = 12000;
    break;
}

switch (coutdate) {
case "1":
    cout = cout + 10;
    break;
case "2":
    cout = cout + 20;
    break;
case "3":
    cout = cout + 30;
    break;
case "4":
    cout = cout + 40;
    break;
case "5":
    cout = cout + 50;
    break;
case "6":
    cout = cout + 60;
    break;
case "7":
    cout = cout + 70;
    break;
case "8":
    cout = cout + 80;
    break;
case "9":
    cout = cout + 90;

```

```
    break;
case "10":
    cout = cout + 100;
    break;
case "11":
    cout = cout + 110;
    break;
case "12":
    cout = cout + 120;
    break;
case "13":
    cout = cout + 130;
    break;
case "14":
    cout = cout + 140;
    break;
case "15":
    cout = cout + 150;
    break;
case "16":
    cout = cout + 160;
    break;
case "17":
    cout = cout + 170;
    break;
case "18":
    cout = cout + 180;
    break;
case "19":
    cout = cout + 190;
    break;
case "20":
    cout = cout + 200;
    break;
case "21":
    cout = cout + 210;
    break;
case "22":
    cout = cout + 220;
    break;
case "23":
    cout = cout + 230;
    break;
case "24":
    cout = cout + 240;
    break;
case "25":
```

```

        cout = cout + 250;
        break;
    case "26":
        cout = cout + 260;
        break;
    case "27":
        cout = cout + 270;
        break;
    case "28":
        cout = cout + 280;
        break;
    case "29":
        cout = cout + 290;
        break;
    case "30":
        cout = cout + 300;
        break;
    case "31":
        cout = cout + 310;
        break;
}

```

```

switch (couttime) {
    case "00 hr":
        cout = cout + 1;
        break;
    case "01 hr":
        cout = cout + 2;
        break;
    case "02 hr":
        cout = cout + 3;
        break;
    case "03 hr":
        cout = cout + 4;
        break;
    case "04 hr":
        cout = cout + 5;
        break;
    case "05 hr":
        cout = cout + 6;
        break;
    case "06 hr":
        cout = cout + 7;
        break;
    case "07 hr":
        cout = cout + 8;
        break;
}

```



```
case "08 hr":
    cout = cout + 9;
    break;
case "09 hr":
    cout = cout + 10;
    break;
case "10 hr":
    cout = cout + 11;
    break;
case "11 hr":
    cout = cout + 12;
    break;
case "12 hr":
    cout = cout + 13;
    break;
case "13 hr":
    cout = cout + 14;
    break;
case "14 hr":
    cout = cout + 15;
    break;
case "15 hr":
    cout = cout + 16;
    break;
case "16 hr":
    cout = cout + 17;
    break;
case "17 hr":
    cout = cout + 18;
    break;
case "18 hr":
    cout = cout + 19;
    break;
case "19 hr":
    cout = cout + 20;
    break;
case "20 hr":
    cout = cout + 21;
    break;
case "21 hr":
    cout = cout + 22;
    break;
case "22 hr":
    cout = cout + 23;
    break;
case "23 hr":
    cout = cout + 24;
```

```

        break;
    }

    if ((fname.length() == 0) || (lname.length() == 0) || (add1.length() == 0) ||
        (add2.length() == 0)
        || (cname.length() == 0) || (sname.length() == 0) || (zip.length() != 6) ||
        (phone.length() != 10)
        || (email.length() == 0) || (cin >= cout))
    {
        if (fname.length() == 0) {
            result = result + (" first name ,");
        }
        if (lname.length() == 0) {
            result = result + (" last name ,");
        }
        if (add1.length() == 0) {
            result = result + (" Address line 1 ,");
        }
        if (add2.length() == 0) {
            result = result + (" Address line 2 ,");
        }
        if (cname.length() == 0) {
            result = result + (" City ,");
        }
        if (sname.length() == 0) {
            result = result + (" State ,");
        }
        if (zip.length() != 6) {
            result = result + (" a valid Zip code ,");
        }
        if (phone.length() != 10) {
            result = result + (" a valid phone number ,");
        }
        if (email.length() == 0) {
            result = result + (" email ,");
        }
        if (cin >= cout) {
            result = result + ("Checkout must be after the Checkin");
        }
        result1.setText(result);
    }

    if (!((fname.length() == 0) || (lname.length() == 0) || (add1.length() == 0) ||
        (add2.length() == 0)

```

```

        || (cname.length() == 0) || (sname.length() == 0) || (zip.length() != 6) ||
(phone.length() != 10)
        || (email.length() == 0) || (cin >= cout))) {
    result1.setText("");
    System.out.println("----NEW BOOKING---");
    System.out.println("Name      : " + fname + " " + lname);
    System.out.println("Address   : " + add1 + " , " + add2 + " ,");
    System.out.println("          " + sname + " , " + cname + " : " + zip);
    System.out.println("Phone no : " + phone);
    System.out.println("Email    : " + email);
    System.out.println("Check in : " + cindate + " " + cinmonth + " " + cintime);
    System.out.println("Check out : " + coutdate + " " + coutmonth + " " + couttime);
    System.out.println("Adults   : " + adult);
    System.out.println("Children : " + children);
    System.out.println("Rooms    : " + nroom);
    if (standard.isSelected() == true) {
        System.out.println("Room Type : Standard");
    }
    if (delux.isSelected() == true) {
        System.out.println("Room Type : Delux");
    }
    if (suite.isSelected() == true) {
        System.out.println("Room Type : Suite");
    }

    JOptionPane.showMessageDialog(this, "Booking Complete!");

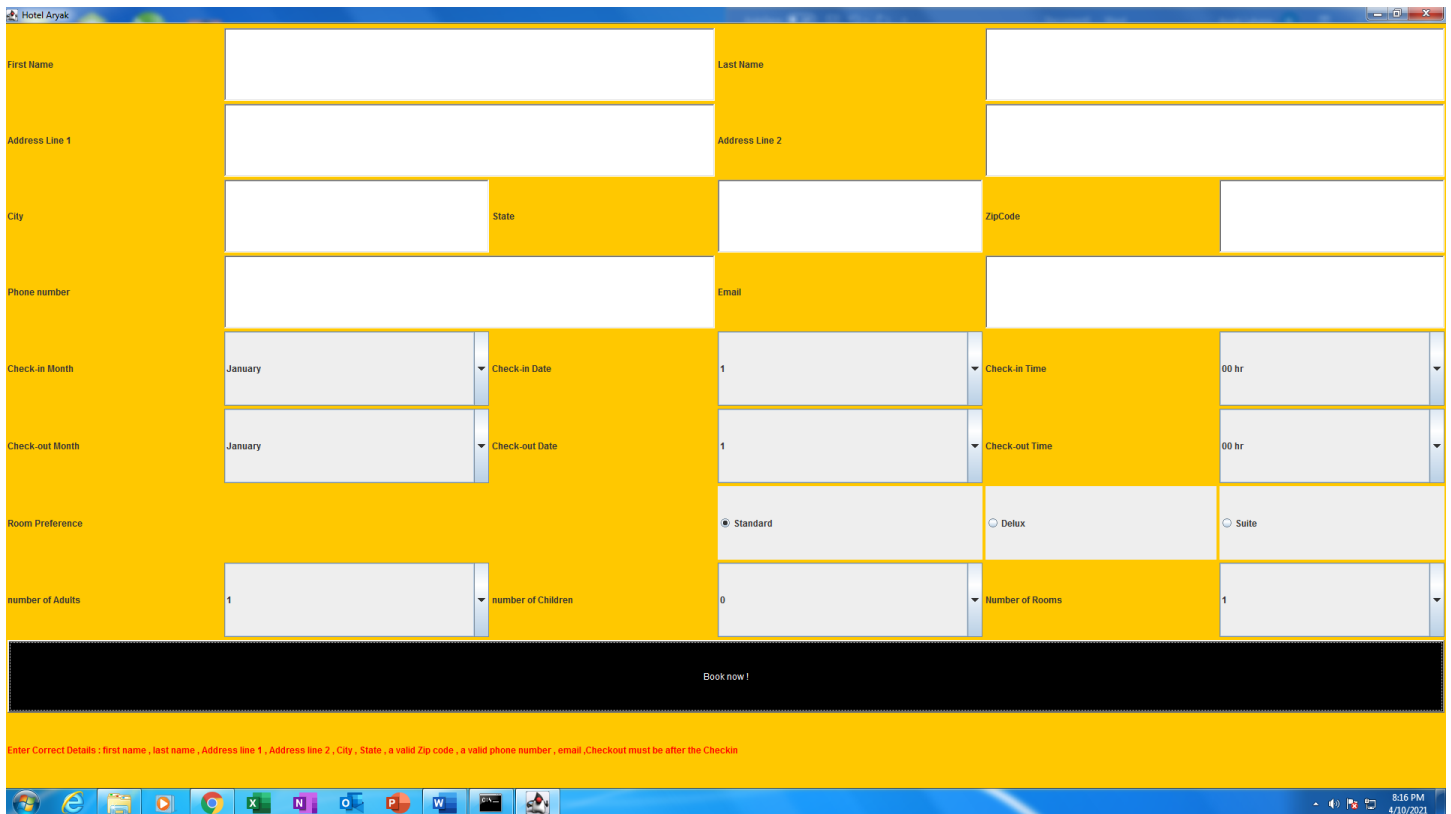
    }

}

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

```

## SCREENSHOTS



The screenshot shows a web browser window titled "Hotel Aryak" displaying a booking form. The form is organized into several rows of input fields. The first row contains "First Name" and "Last Name". The second row contains "Address Line 1" and "Address Line 2". The third row contains "City", "State", "ZipCode", and an empty field. The fourth row contains "Phone number" and "Email". The fifth row contains "Check-in Month" (dropdown), "Check-in Date" (calendar), "Check-in Time" (dropdown), and "00 hr" (dropdown). The sixth row contains "Check-out Month" (dropdown), "Check-out Date" (calendar), "Check-out Time" (dropdown), and "00 hr" (dropdown). The seventh row contains "Room Preference" with three radio buttons: "Standard" (selected), "Delux", and "Suite". The eighth row contains "number of Adults" (dropdown), "number of Children" (dropdown), "Number of Rooms" (dropdown), and "1" (dropdown). Below the form is a black "Book now!" button. At the bottom, a red error message reads: "Enter Correct Details : first name , last name , Address line 1 , Address line 2 , City , State , a valid Zip code , a valid phone number , email ,Checkout must be after the Checkin". The Windows taskbar at the bottom shows various application icons and the system clock indicating 8:38 PM on 4/10/2021.

First Name		Last Name	
Address Line 1		Address Line 2	
City	State	ZipCode	
Phone number		Email	
Check-in Month	January	Check-in Date	1
Check-out Month	January	Check-out Date	1
Room Preference	<input checked="" type="radio"/> Standard <input type="radio"/> Delux <input type="radio"/> Suite		
number of Adults	1	number of Children	0
		Number of Rooms	1

Book now!

Enter Correct Details : first name , last name , Address line 1 , Address line 2 , City , State , a valid Zip code , a valid phone number , email ,Checkout must be after the Checkin

Hotel Aryak

First Name	abc		Last Name	abc	
Address Line 1	abc		Address Line 2	abc	
City	aaa	State	aaa	ZipCode	12345
Phone number	12345		Email	aaa	
Check-in Month	January	Check-in Date	1	Check-in Time	00 hr
Check-out Month	January	Check-out Date	1	Check-out Time	00 hr
Room Preference	<input checked="" type="radio"/> Standard <input type="radio"/> Delux <input type="radio"/> Suite				
number of Adults	1	number of Children	0	Number of Rooms	1

Book now!

Enter Correct Details : a valid Zip code , a valid phone number ,Checkout must be after the Checkin

Windows Taskbar: 8:17 PM 4/10/2021

Hotel Aryak

First Name	abc		Last Name	abc	
Address Line 1	abc		Address Line 2	abc	
City	aaa	State	aaa	ZipCode	123456
Phone number	1234567890		Email	aaa	
Check-in Month	May	Check-in Date	4	Check-in Time	00 hr
Check-out Month	January	Check-out Date	1	Check-out Time	00 hr
Room Preference	<input checked="" type="radio"/> Standard <input type="radio"/> Delux <input type="radio"/> Suite				
number of Adults	1	number of Children	0	Number of Rooms	1

Book now!

Enter Correct Details :Checkout must be after the Checkin

Windows Taskbar: 8:18 PM 4/10/2021

