**Assignment 1**

* Course Name -

Visual Programming

* Course Code -

CSCI 22042

* Student Name -

D. Bulitha Kawushika de Zoysa

* Student Index Number -

**CS-2019-025**

using System;

**Question 1** - Code

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Find\_Prime

{

internal class Program

{

static void Main(string[] args)

{

int num\_start,num\_end,num\_of\_divisor=0,max\_divisor=0,max\_divisor\_number=0;

//input numbers

Console.WriteLine(" Enter the first number of Range ");

num\_start=int.Parse(Console.ReadLine());

Console.WriteLine(" Enter the last number of Range ");

num\_end=int.Parse(Console.ReadLine());

Console.WriteLine(" Prime Numbers ");

//identify 1 or 2 are in this range

if (num\_start == 1)

{

num\_start = 2;

}

if (num\_start==2)

{

num\_start = 3;

Console.WriteLine("2 , ");

}

//check prime or not and number of divisors

for(int i=num\_start;i<=num\_end;i++)

{

bool prime\_value = true;

for (int j= 2; j < i; j++)

{

if(i%j == 0)

{

num\_of\_divisor += 1;

prime\_value = false;

}

}

if( prime\_value )

{

Console.WriteLine(i+" , ");

}

//update the number which has maximum number of divisors yet

if (max\_divisor<num\_of\_divisor)

{

max\_divisor\_number = i;

max\_divisor = num\_of\_divisor;

}

}

Console.WriteLine("Number which has highest number of divisors is " + max\_divisor\_number);

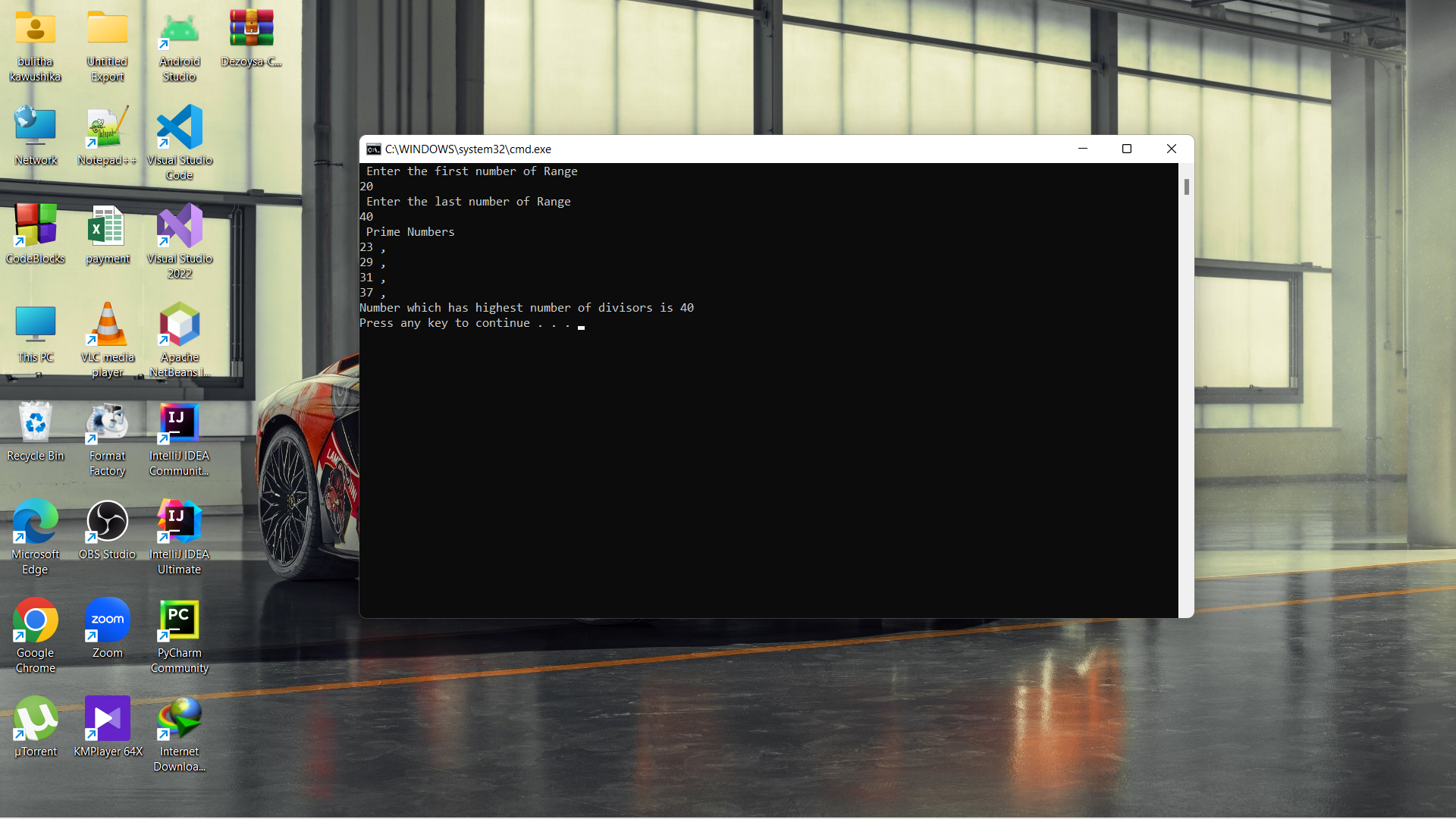
}

}

}

Working of Program

Example 1 - Enter 20 – 40 Range As input



Example 2 - Enter 1 – 25 Range As input



**Question 2 -** Code

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Hotel\_Booking\_System

{

public partial class Form1 : Form

{

int available\_single = 10,available\_double=7,available\_family=4,available\_suite=2;

double cost = 0;

public Form1()

{

InitializeComponent();

}

//total cost calculate method

public double calculate\_cost (int room\_rate,int number\_of\_Days,int number\_of\_rooms,string room\_type)

{

cost = cost+(room\_rate \* number\_of\_Days \* number\_of\_rooms);

if (room\_type == "Double" & number\_of\_Days > 5)

{

cost = cost \* 0.95;

}

if (room\_type == "Suite" & number\_of\_Days > 5)

{

cost = cost \* 0.88;

}

return cost;

}

//clearn text after one selection

public void clean\_form\_room\_data()

{

Number\_of\_Days\_text.Clear();

Number\_of\_room\_text.Clear();

Room\_type\_combo.SelectedIndex = -1;

Another\_booking\_combo.SelectedIndex = -1;

}

//exit button

private void Exit\_btn\_Click(object sender, EventArgs e)

{

this.Close();

}

//calculate the price button

private void button1\_Click(object sender, EventArgs e)

{

string room\_type,Another\_room= "No" ;

int room\_rate = 0, number\_of\_rooms = 0, number\_of\_Days = 0;

room\_type = Room\_type\_combo.Text;

//Cheeck personal information

if (Name\_text.Text == "" | Address\_text.Text == "")

{

MessageBox.Show("Enter your personal details correctly (Name and Address) ", "Message", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

//exception handeling number of days,rooms are not fill and invalid input

try

{

number\_of\_rooms = int.Parse(Number\_of\_room\_text.Text);

number\_of\_Days = int.Parse(Number\_of\_Days\_text.Text);

}

catch

{

MessageBox.Show("Form is incomplete or invalid input", "Message", MessageBoxButtons.OK, MessageBoxIcon.Error);

};

}

if (number\_of\_Days > 0)

{

//select type of room

switch (room\_type)

{

case "Single":

room\_rate = 5000;

//check availability of room

if (number\_of\_rooms <= available\_single)

{

available\_single -= number\_of\_rooms;

cost = calculate\_cost(room\_rate, number\_of\_Days, number\_of\_rooms, room\_type);

}

else

{

//rooms are not available message

MessageBox.Show("Single Rooms are not Available at this moment", "Message", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

break;

case "Double":

room\_rate = 7500;

if (number\_of\_rooms <= available\_double)

{

available\_double -= number\_of\_rooms;

cost = calculate\_cost(room\_rate, number\_of\_Days, number\_of\_rooms, room\_type);

}

else

{

MessageBox.Show("Double Rooms are not Available at this moment", "Message", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

break;

case "Family":

room\_rate = 8000;

if (number\_of\_rooms <= available\_family)

{

cost = calculate\_cost(room\_rate, number\_of\_Days, number\_of\_rooms, room\_type);

available\_family -= number\_of\_rooms;

}

else

{

MessageBox.Show("Family Rooms are not Available at this moment", "Message", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

break;

case "Suite":

room\_rate = 12500;

if (number\_of\_rooms <= available\_suite)

{

cost = calculate\_cost(room\_rate, number\_of\_Days, number\_of\_rooms, room\_type);

available\_suite -= number\_of\_rooms;

}

else

{

MessageBox.Show("Suite Rooms are not Available at this moment", "Message", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

break;

//room type is not selected message

default:

MessageBox.Show(" Select Room Type", "Incomplete Form", MessageBoxButtons.OK, MessageBoxIcon.Error);

break;

}

}

Cost\_text.Text = Convert.ToString(Math.Round(cost, 2));

//Another booking for same user ( by default anoter room is set as "No" )

Another\_room = Another\_booking\_combo.Text;

if (Another\_room == "Yes")

{

clean\_form\_room\_data();

}

}

//clear button

private void Clear\_btn\_Click(object sender, EventArgs e)

{

cost = 0;

Cost\_text.Clear();

Name\_text.Clear();

Address\_text.Clear();

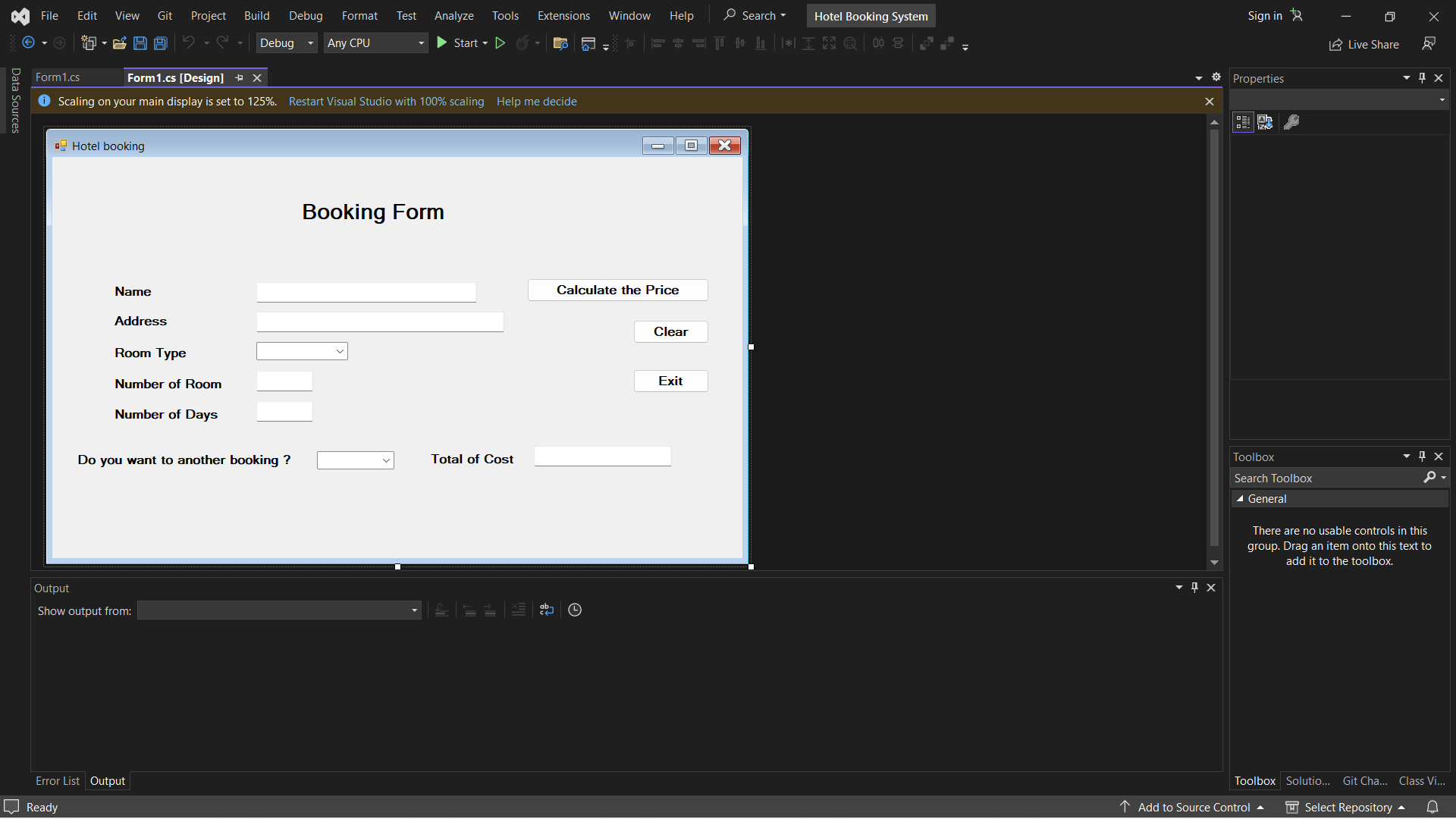
clean\_form\_room\_data();

}

}

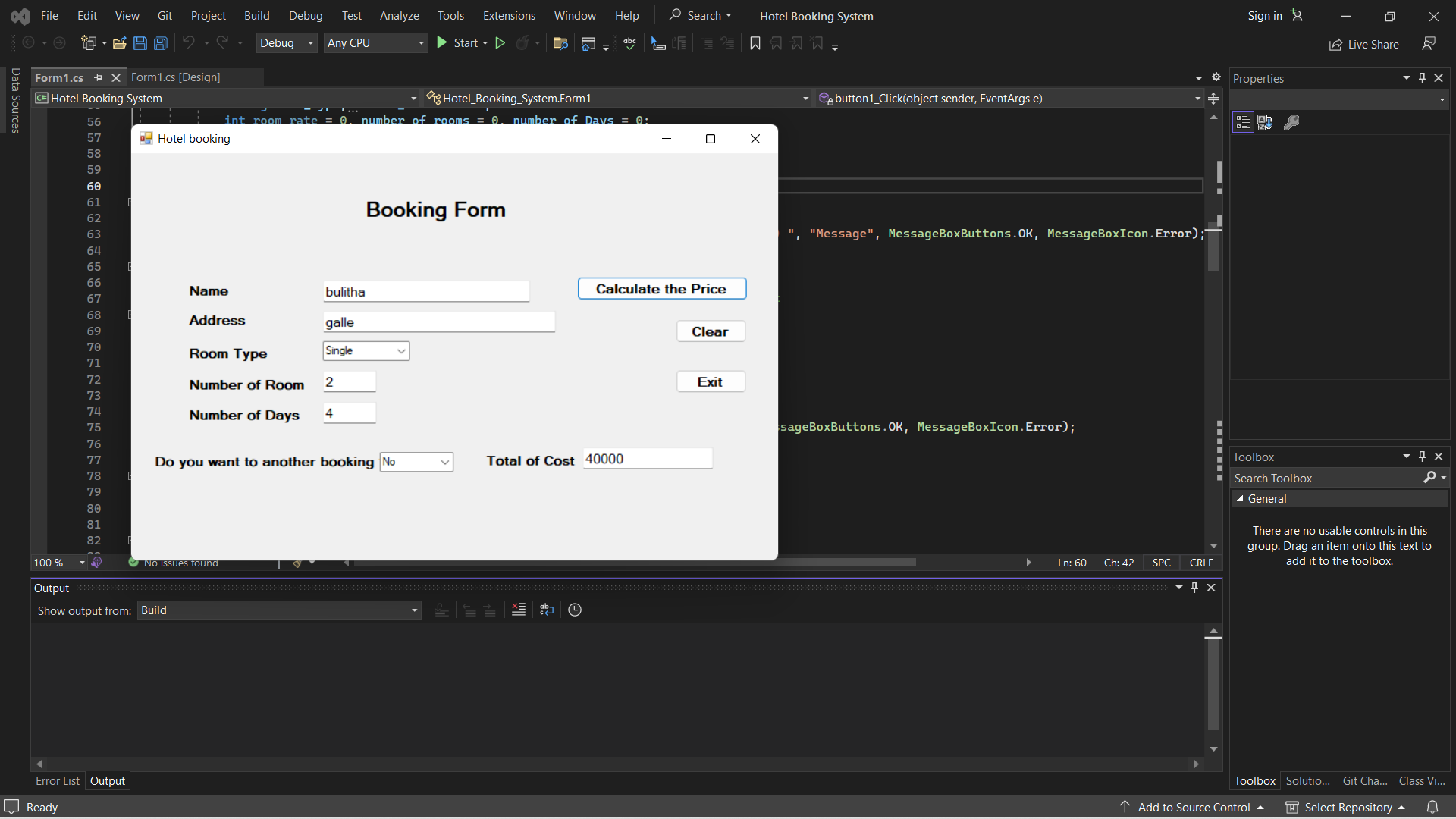
}

Design window Form –



Working of Programing Features

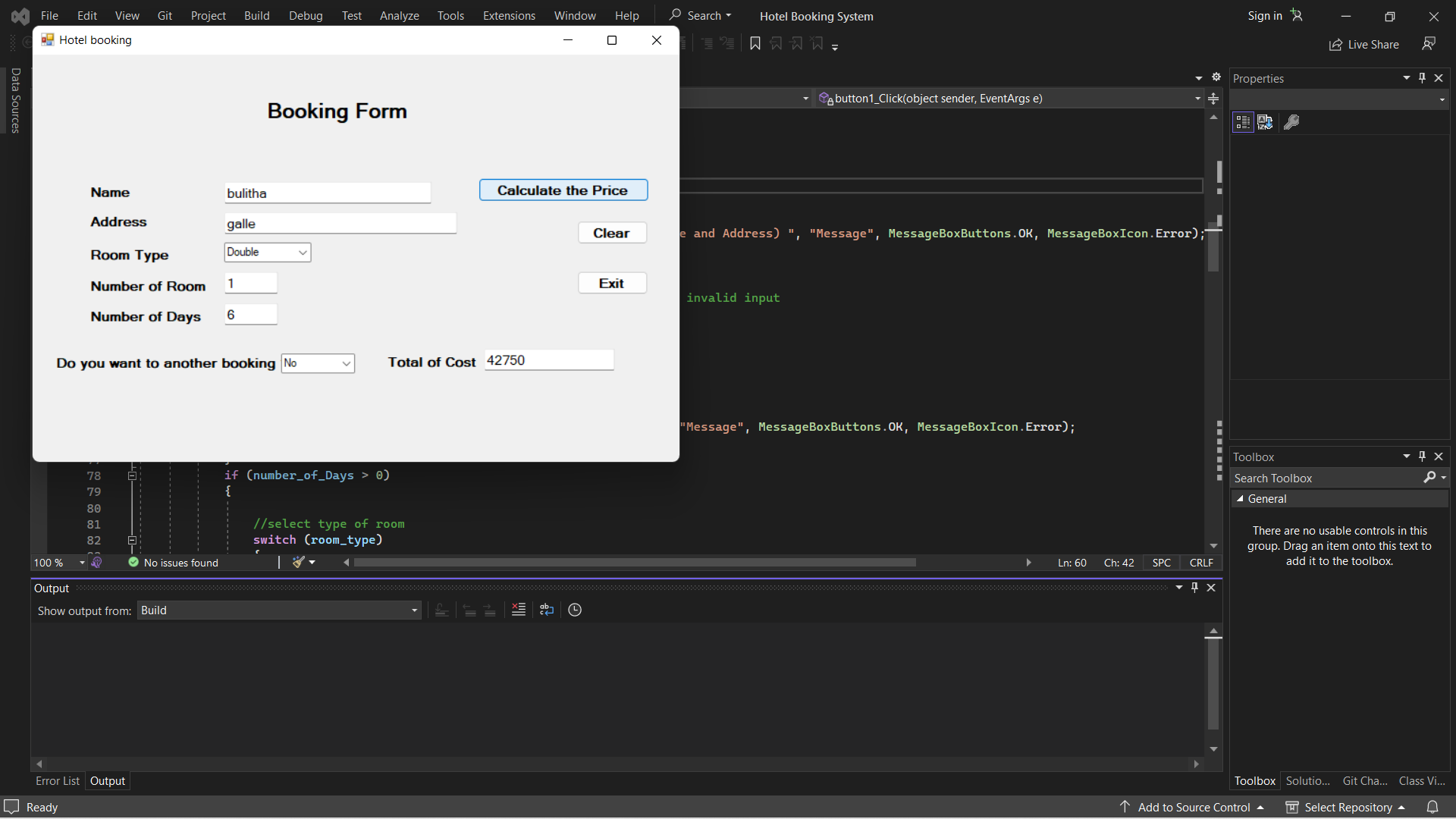
* Calculate total cost of customers and offer discount who stay more than 5 days in double room or suite room. (5% and 12% respectively)



Example 1 -

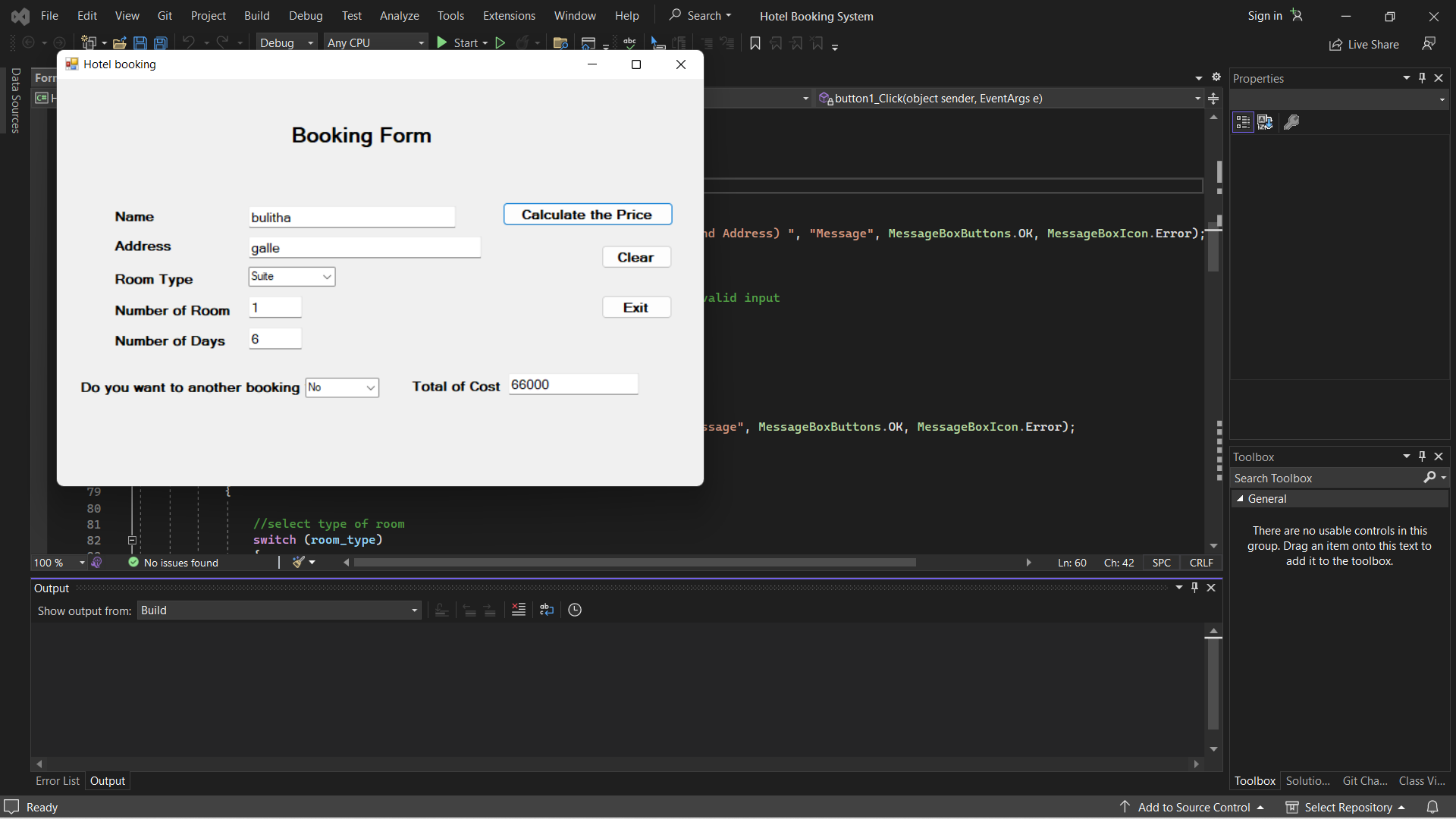
2 Single Rooms booking for 4 Days.

(Customers who stay more than 5 days in Suites or Double rooms are entitled for discounts)



Example 2 -

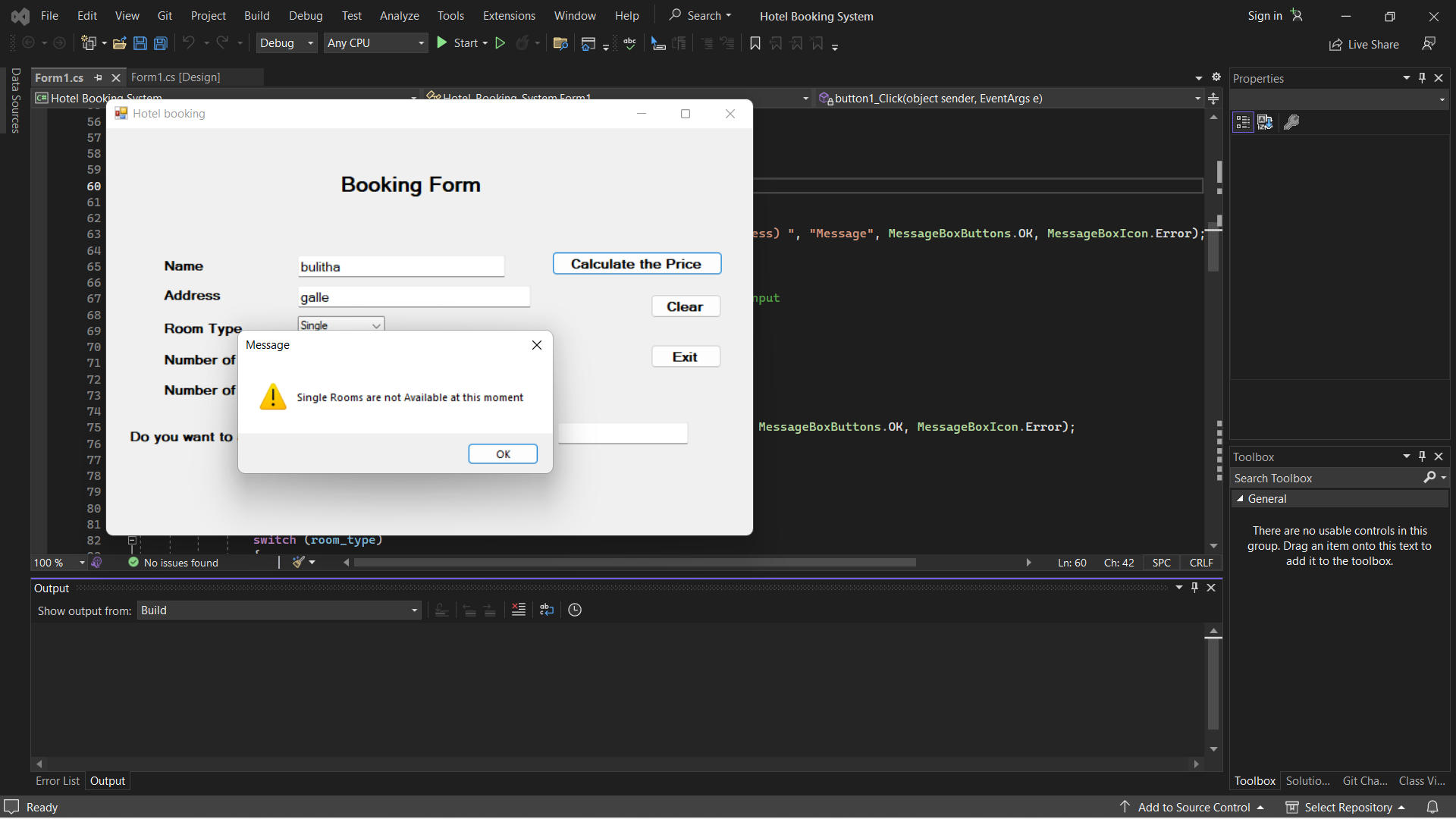
1 Double Room booking for 6 Days. ( 5% Discount is given )



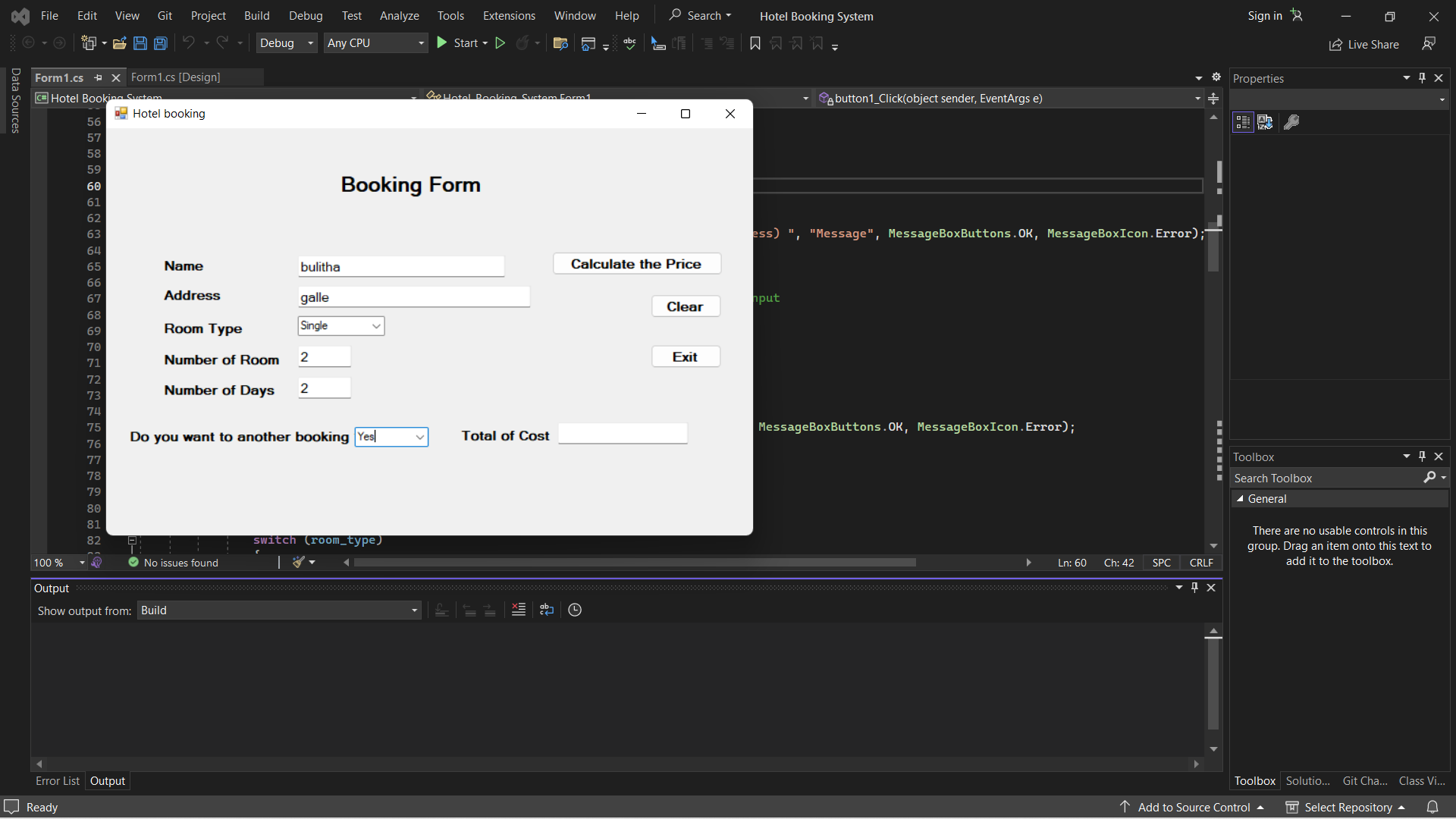
Example 3 -

1 Suite Room booking for 6 Days. ( 12% Discount is given )

* Manage number of rooms and if all rooms are booked in some room category message box will appear.



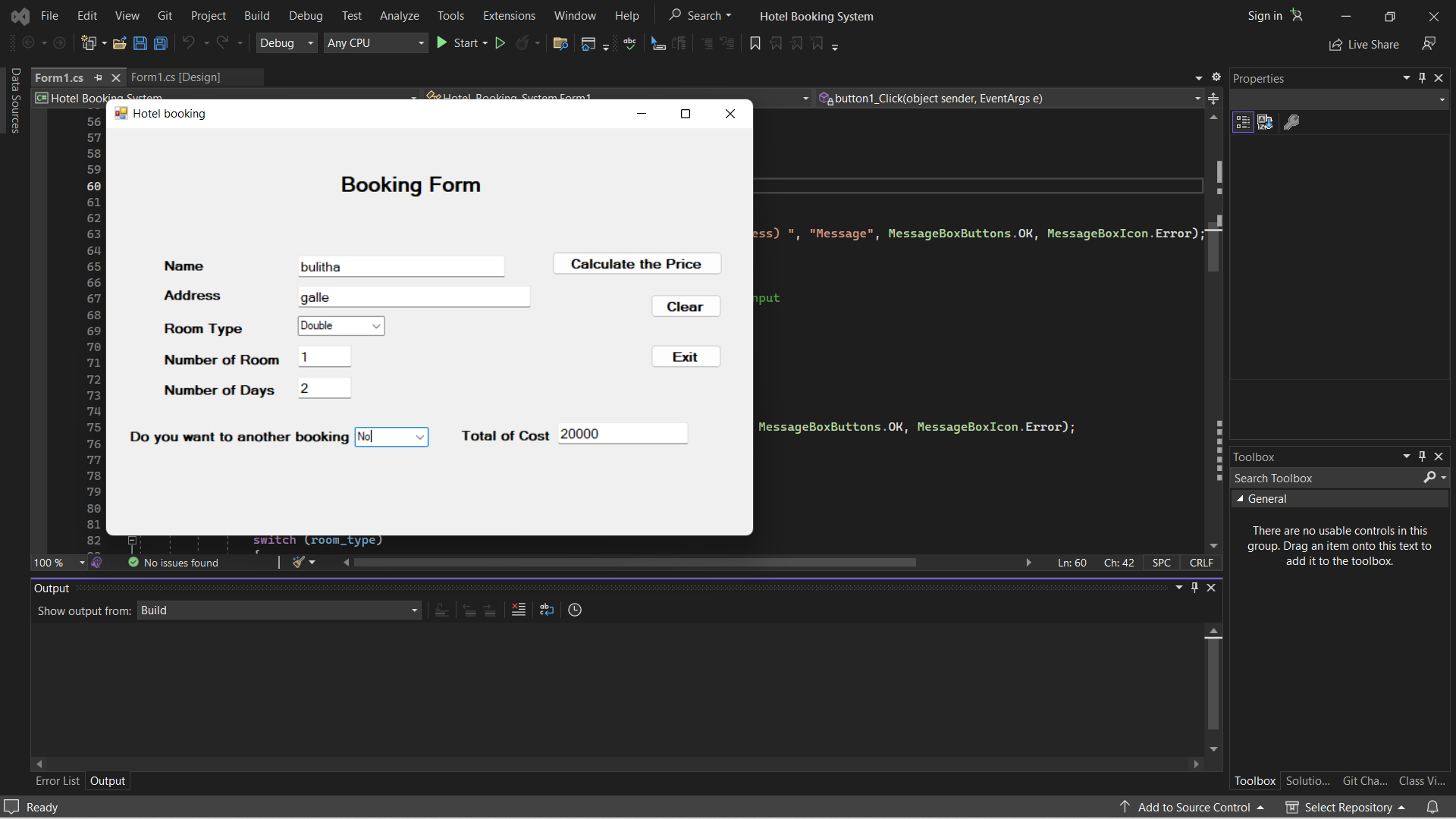
* Program can run for multiple booking for one person as long as he/she indicates it by selecting “Yes” for another booking.



Example 4 -

(Step 1) -

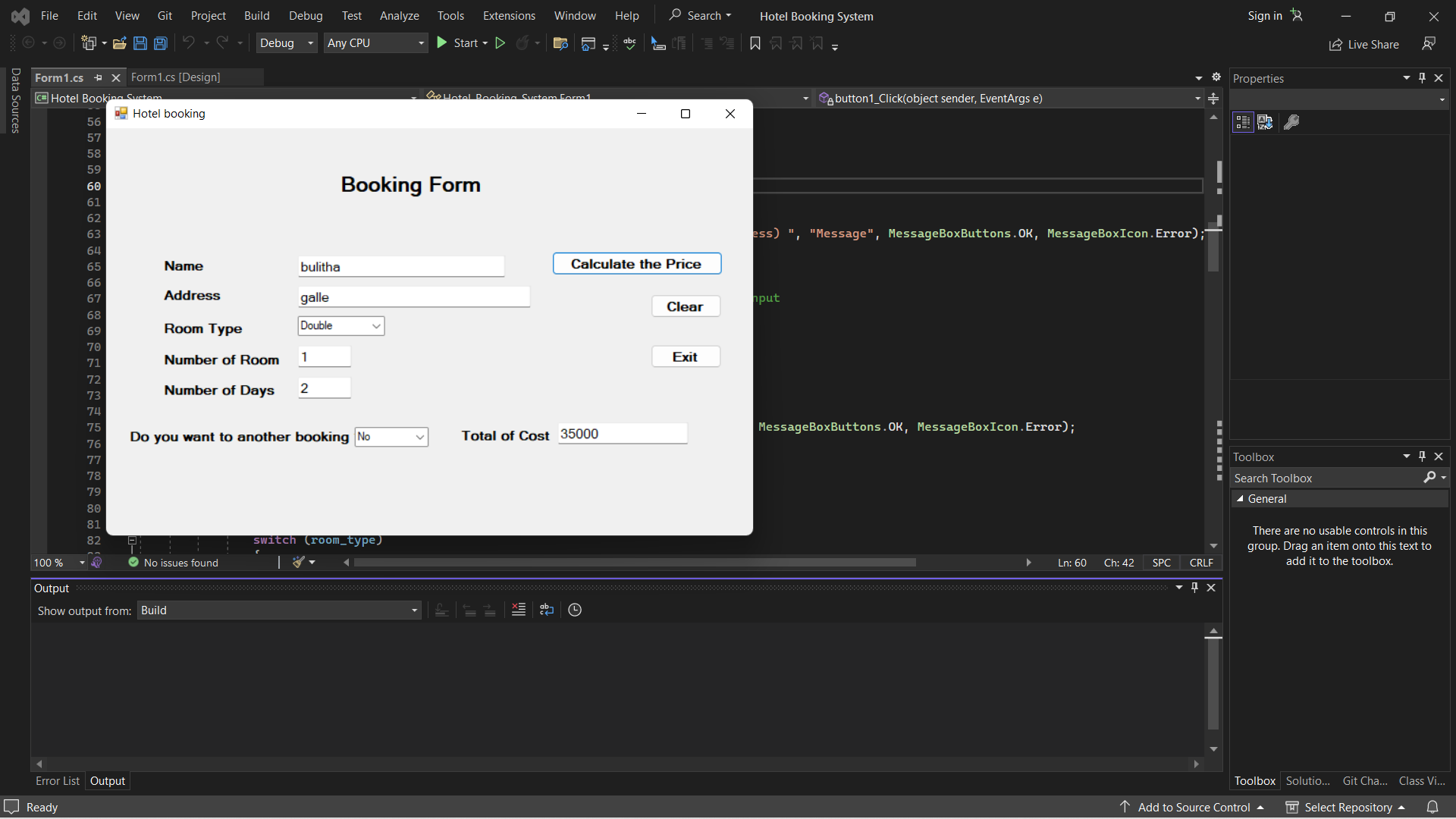
Fill the form for 1st booking and Press “Calculate the Price” button



(Step 2) -

Fill the form for 2nd booking.

Already Program display the Total of Cost for Previous booking ( i.e. 1st booking )



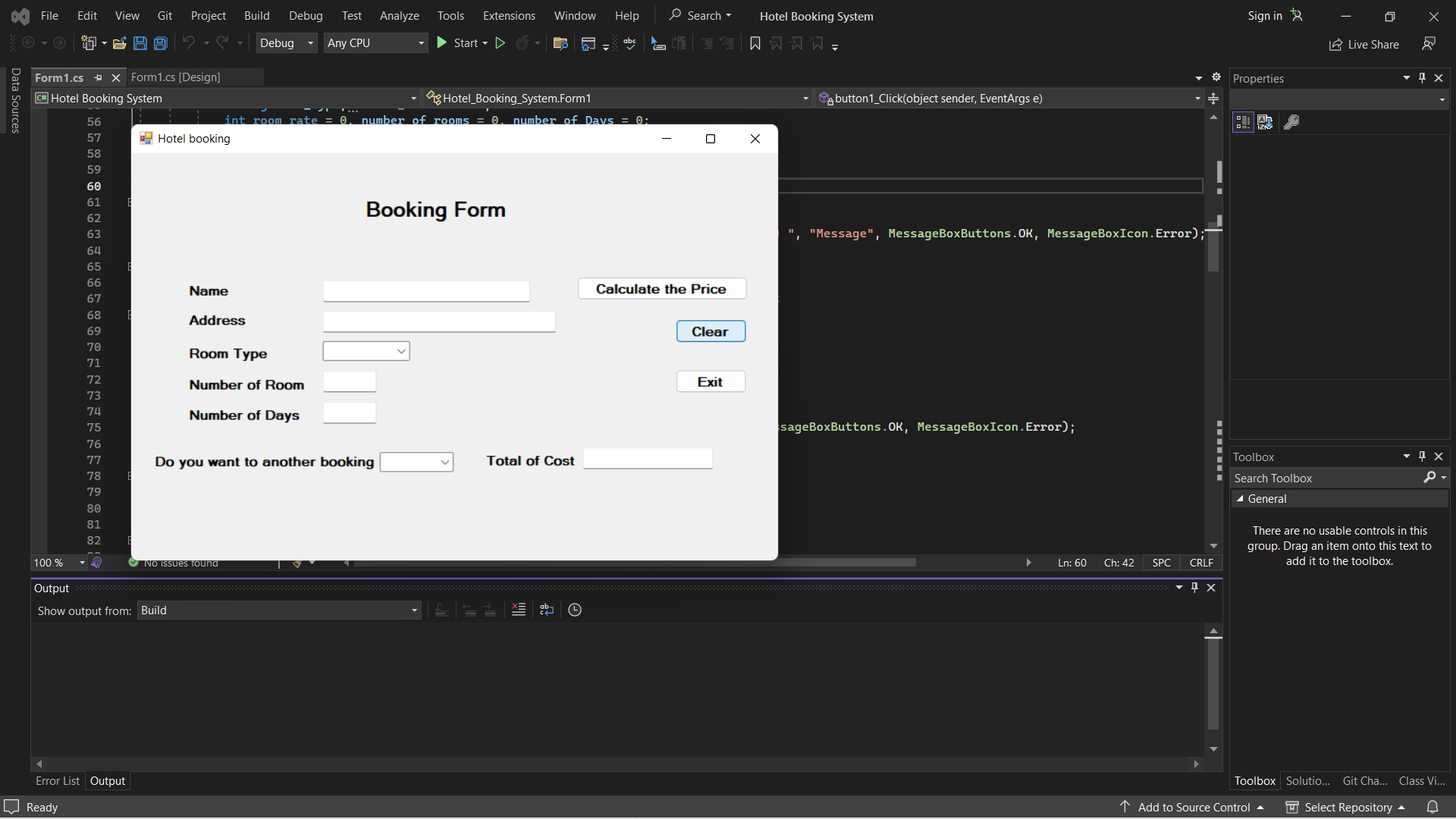
(Step 3) -

Then Click “Calculate the Price” button.

After that Program display the Total of Cost for all multiple bookings.

If you do not need any other booking Select “No” in the “Do you want to another booking” combo box.

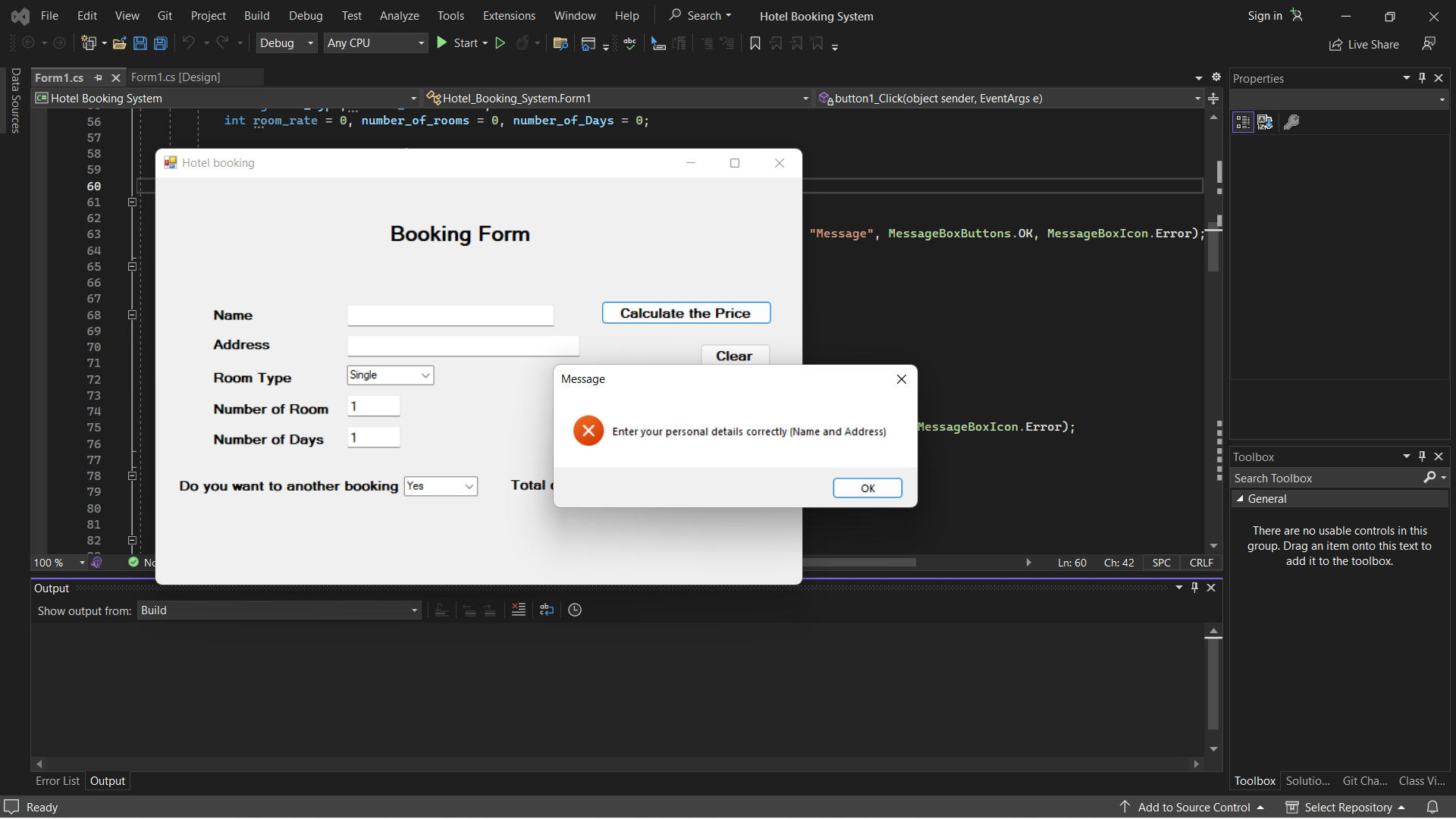
* Clear button for erase all contents on textboxes.



* Exit button for exit program.

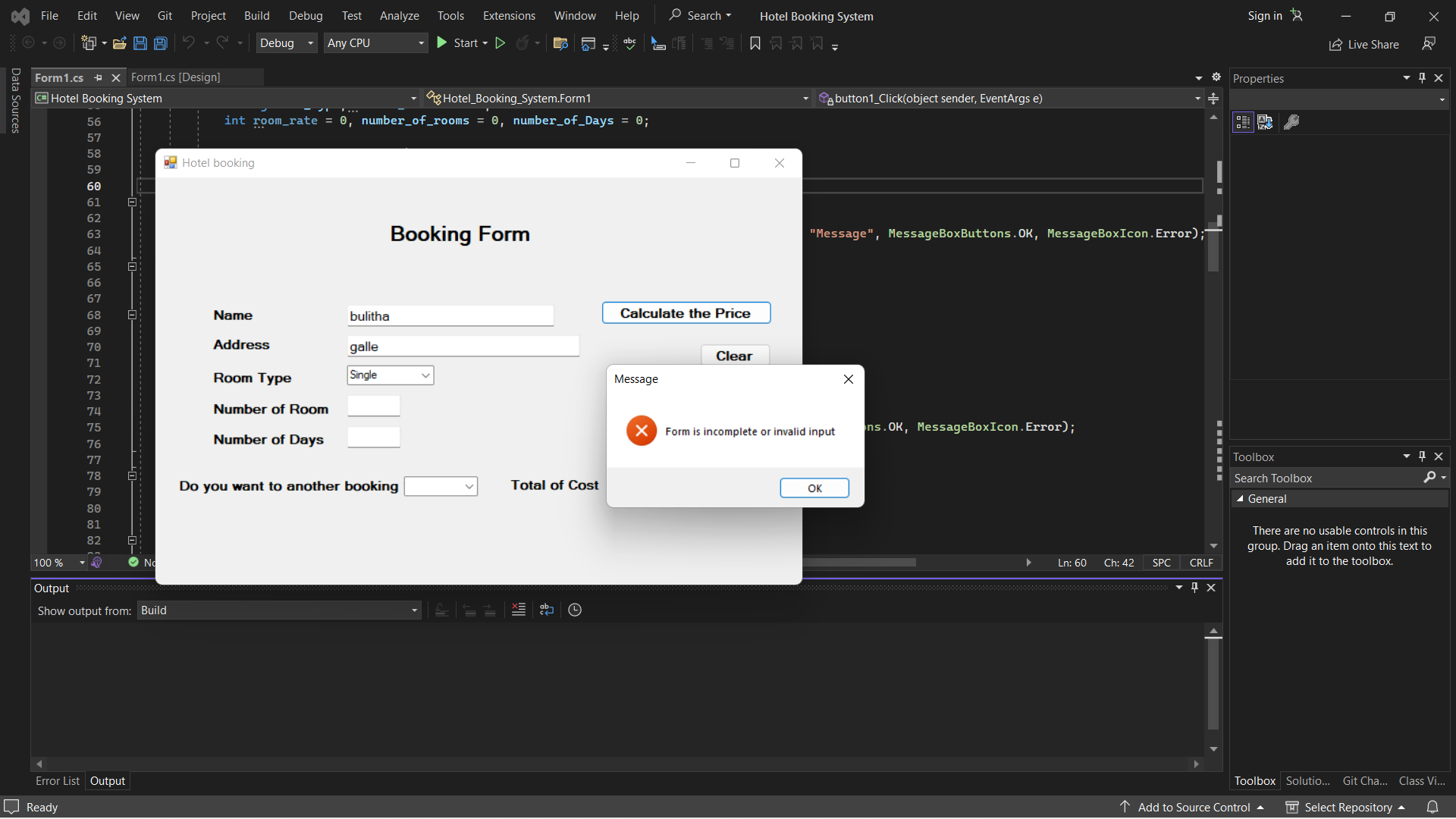
Additional Features

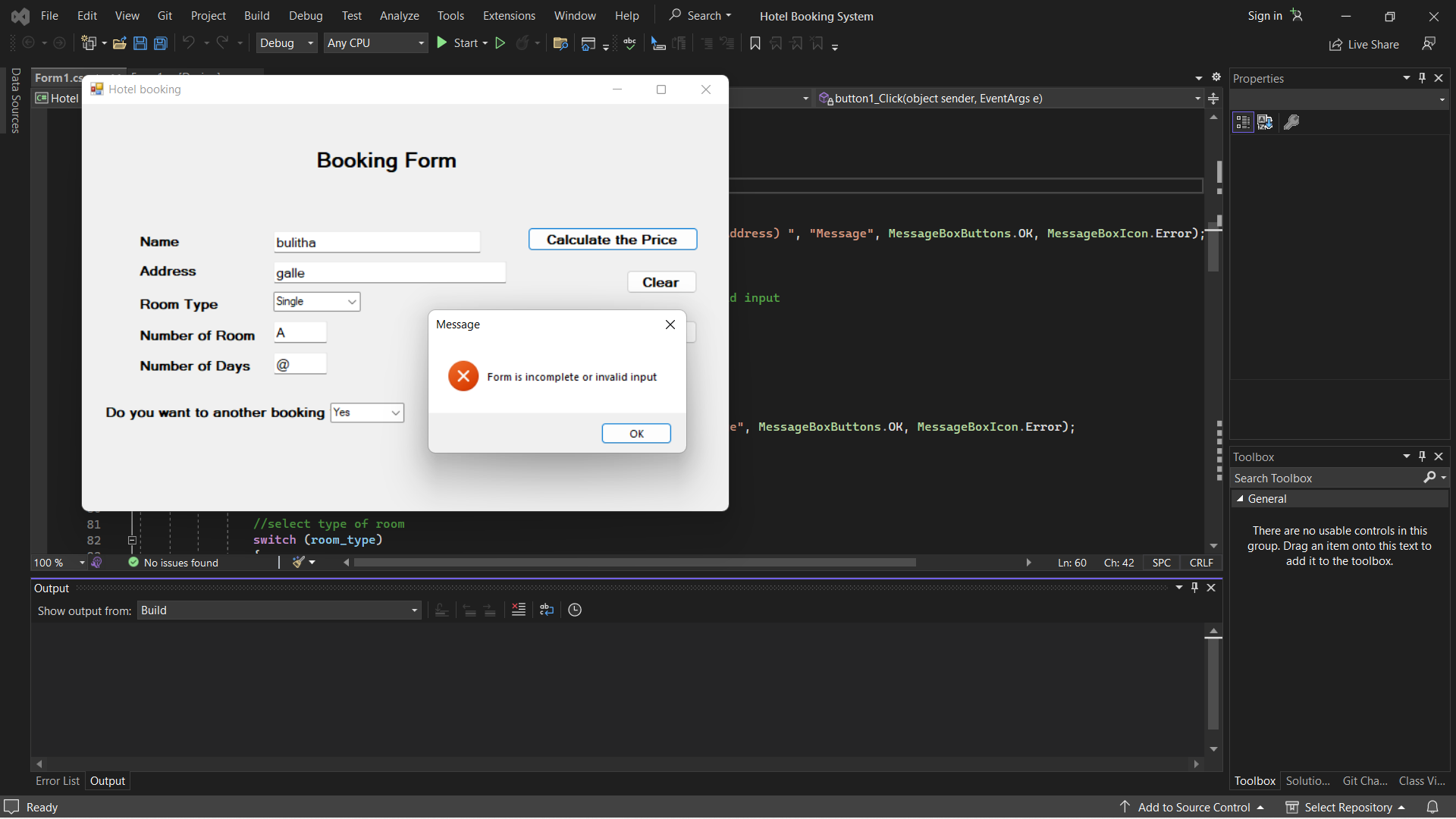
* If some user not fill name an address (customer information) message box will appear.



* If some user not fill number of rooms or number of days or if user input invalid data (such as letters , symbols, emojis ) message box will appear.

( exceptions are being also handled )





* If some user not select item in Another booking combo box, by default choice is get as “No”
* If some user not select item in Room type combo box, message box will appear.

