Drink Dispenser

TEAM 3G

CS 3101

Rapid Application Development and Visual Programming Technologies

Group Members

M.F.Zaidh S14597 (Leader)

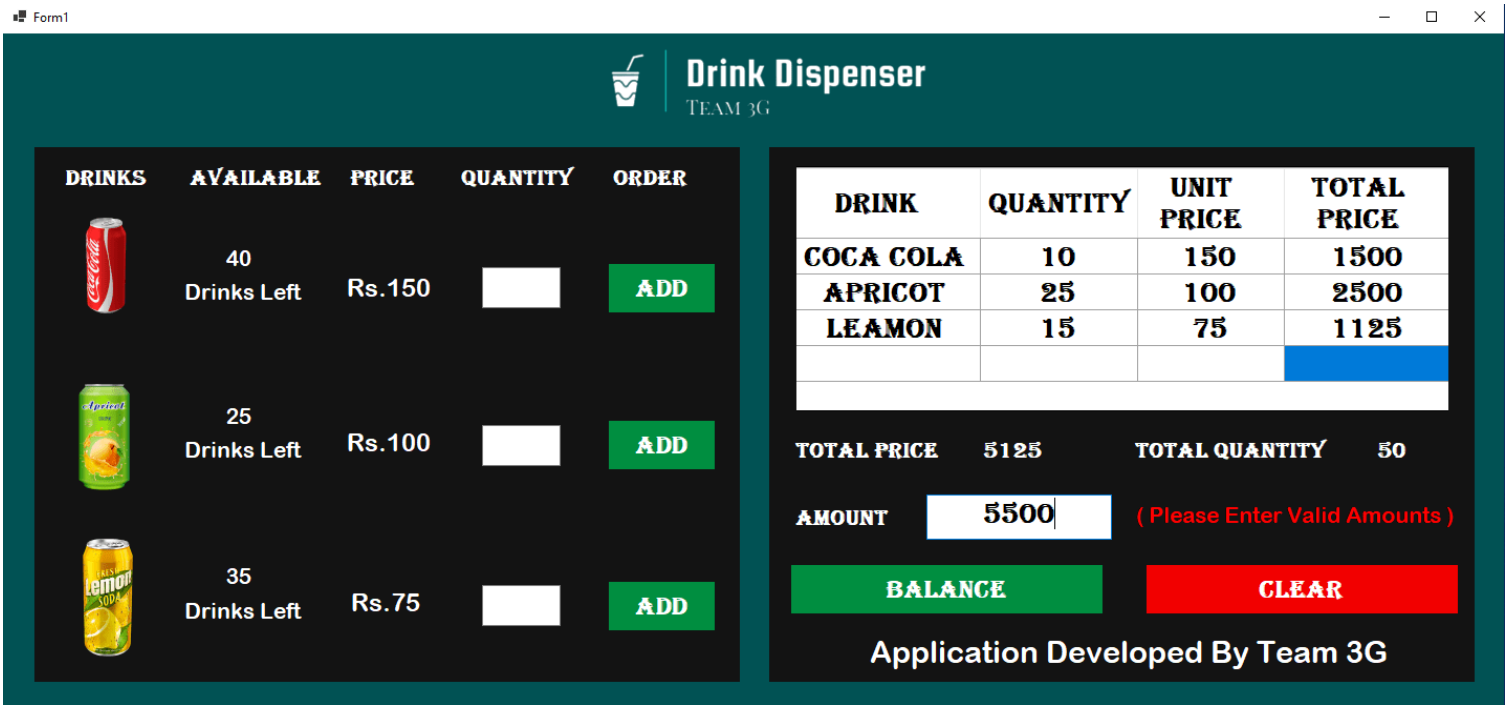
R.Harithas S14057

MH.Ali S14599

S.Panujan S14095

S.Piramilan S14570

Introduction



This is our Drink Dispenser GUI application in this application customer can choose the drinks from drink menu and enter the avaliable quantity drinks in Textbox and simply click the Add button, Selected drink item, quantity, unit price and total price will updated to the order table view while customer ordering drinks item total quantity and total price will automatically updated in the total quantity and total price label.

Then the customer can enter the paid amount to the Amount TextBox and click the Balance button. Our application will show the balance and total bill amount in MessageBox.

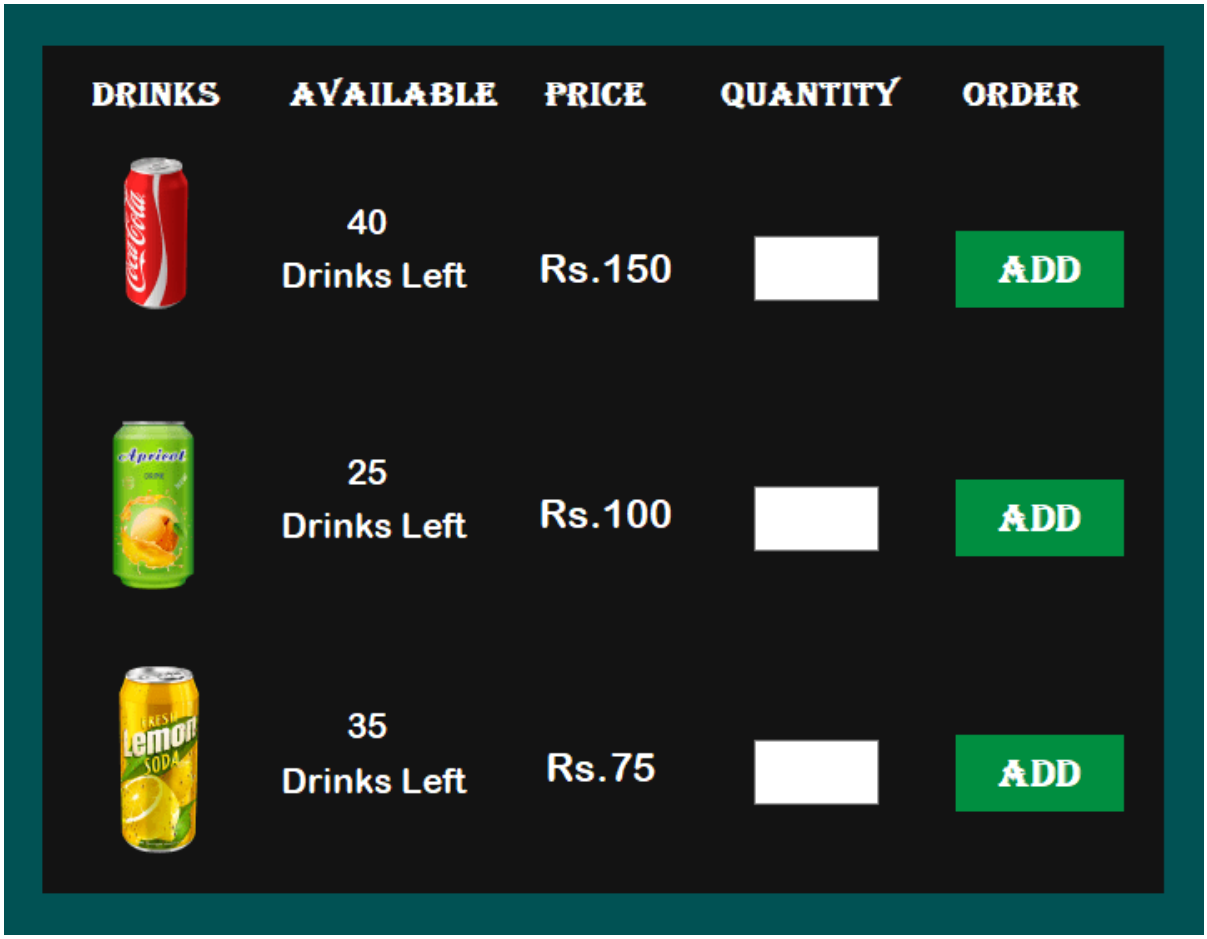
If available stock drinks decrease to 10 or less,at the time the Add button will disable and that drink item will be updated in 90 seconds by the Employee then the customer can order it but he can't Add more than 50 of the same drink item at one time.

If Customer don't want to order the drink items he can Click the Clear button and all list items will clear and the available items will rearrange automatically.

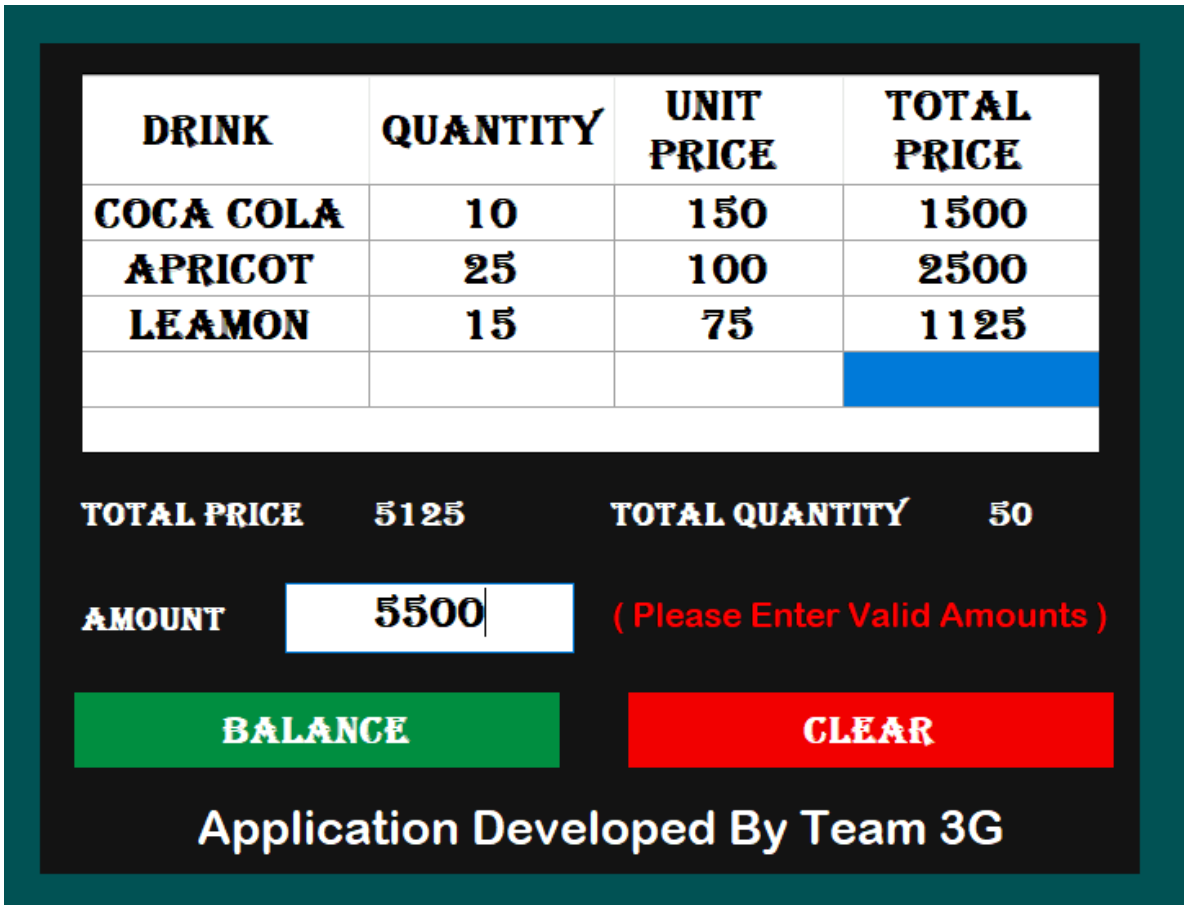
For creating GUI, we have used below mentioned elements.

- Frame
- Panel
- DataGridView
- Label
- TextBox
- Button
- Timer
- PictureBox

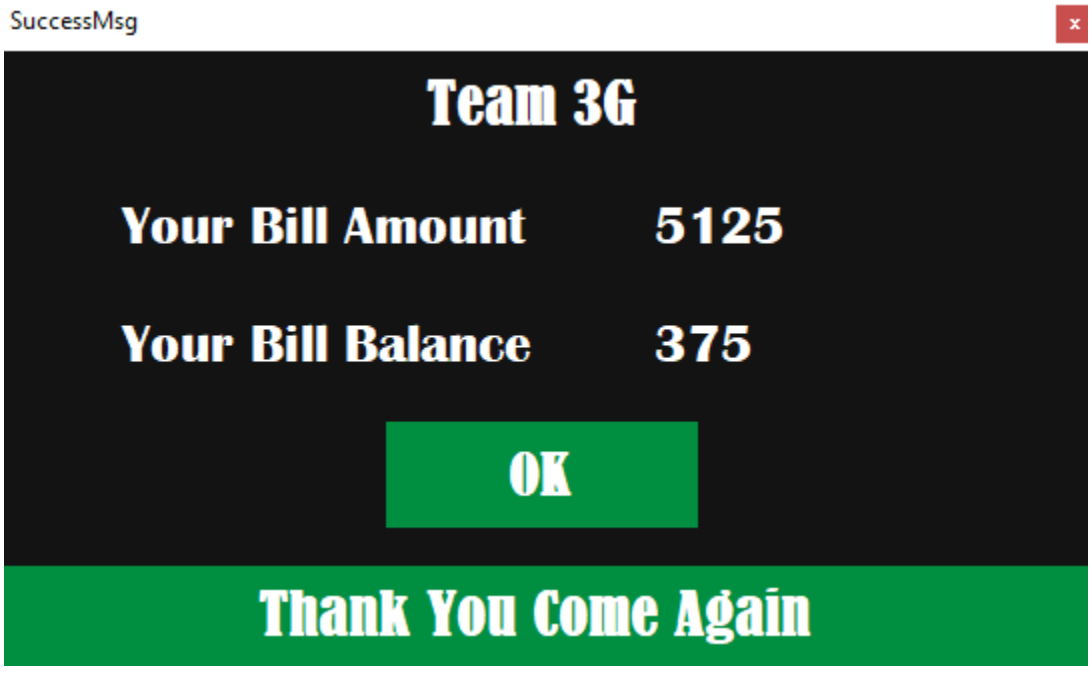
Drinks Menu



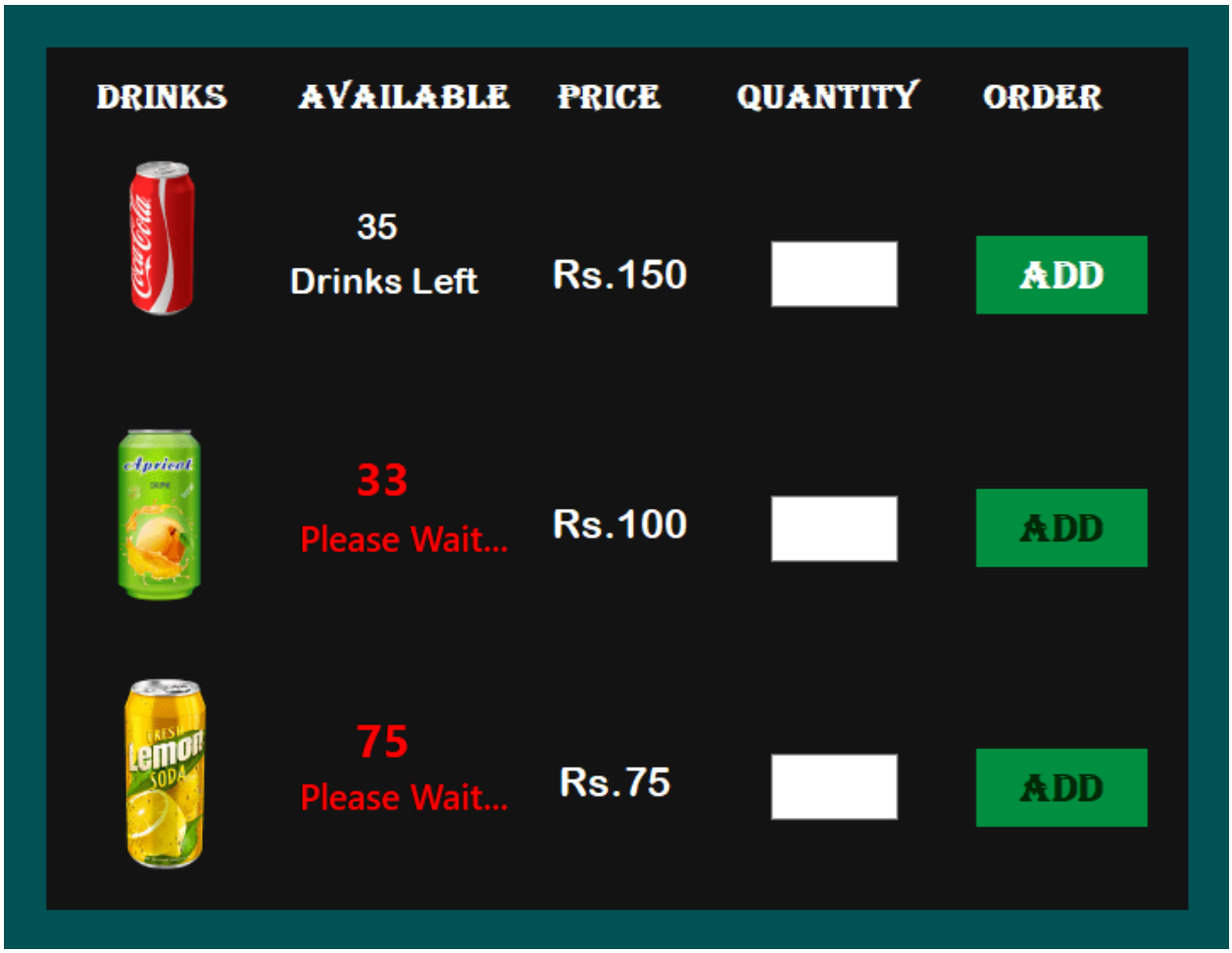
Order Menu



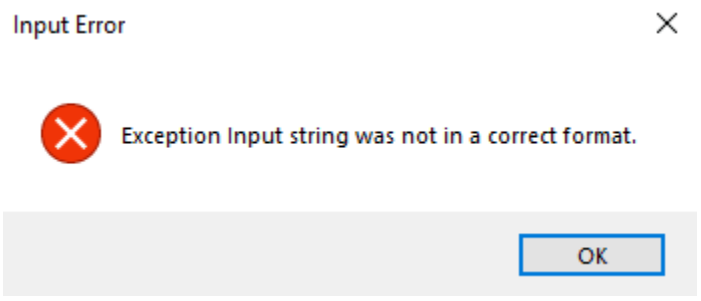
Success MessageBox

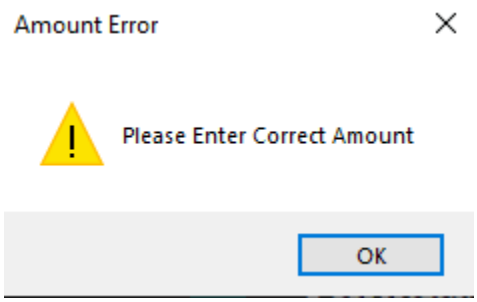


Full Fill The Out of Stocks



Some Alert Error Message Boxes





Some Main Functions Which Are Used In This Application

1) Add Button OnClick Function

This Function Execute When Add Button Click First Get The Quantity Value From TextBox Then Check The Value With Some Conditions And Execute Codes.

And Also Check The Available Stock Values and Full Fill If Stock Drinks Less Than 10.

```
private void BtnDrink1_Click(object sender, EventArgs e)
        {
            try
            {
                txt1 = int.Parse(textBox1.Text);
                if (txt1 <= 0 || txt1 > 50)
                {
                    MessageBox.Show("Please Enter Between Minmum & Maximum Available Quantity Drinks",
                    "Quantity Error", MessageBoxButtons.RetryCancel, MessageBoxIcon.Warning);
                    textBox1.Text = "";
                }
                else if (drinks1 >= txt1)
                {
                    drinks1 -= txt1;
                    quantity1 += txt1;
                    total1 += quantity1 * 150;
                    AddDataOnTable("Coca Cola", quantity1, 150, total1);
                    textBox1.Text = "";
                    lbl1.Text = drinks1.ToString();
                    CalculateTotal();
                }
                else if (drinks1 < txt1)
                    AddDrinksTimer1();
            }
            catch (Exception ex)
            {
                MessageBox.Show("Exception " + ex.Message, "Input Error",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
                textBox1.Text = "";
            }

            int drinks1More = int.Parse(lbl1.Text);
            if (drinks1More >= 0 && drinks1More <= 10)
            {
                AddDrinksTimer1();
            }
        }
```

2) Add Drinks When Out Of Stock Function

This Function Executes When Our Application Available Drinks Less Than 10. The Timer Will Start For 90 Seconds That Time Our Employee Will Full Fill The Drinks.

That Time The Label Will So The Time Add The Waiting Message To The Customer.At That Time Customer Can't Add That Drink To Order Menu.After Finished Full Fill He Can Add Drinks

```
1  private void AddDrinksTimer1()
2  {
3      addDrinksTime1 = 90;
4      AddDrinks1.Start();
5  }
6
7  private void AddDrinks1_Tick(object sender, EventArgs e)
8  {
9      BtnDrink1.Enabled = false;
10     lblWait1.Text = "Please Wait...";
11     lblAddDrink1.Text = addDrinksTime1--.ToString();
12     if (addDrinksTime1 == 0)
13     {
14         AddDrinks1.Stop();
15         lblAddDrink1.Text = "";
16         lblWait1.Text = "";
17         BtnDrink1.Enabled = true;
18         lbl1.Text = "50";
19         drinks1 = 50;
20     }
21 }
22
```

3) Balance Button OnClick Function

This Function Executes When Customer Complete His Order He Need To Enter Amount To Complete This Transaction That Time He Need To Add Amount That He Have,

Then Click Balance Some Conditions Will Execute To Check Your Entered Amount Finally Got Success Message With Bill Amount And Available Balance

```
private void BtnBalance_Click(object sender, EventArgs e)
        {
            try
            {
                int amount = int.Parse(textBoxAmount.Text);
                int balance = amount - total_p;
                if (balance > 0)
                {
                    Form SuccessMsg = new SuccessMsg(total_p.ToString(),balance.ToString());
                    SuccessMsg.ShowDialog();
                    ClearValues();
                    ValuesZero();
                }
                else
                    MessageBox.Show("Please Enter Correct Amount", "Amount Error",
                    MessageBoxButtons.OK, MessageBoxIcon.Warning);
            }
            catch (Exception ex)
            {
                MessageBox.Show("Exception " + ex.Message, "Input Error",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
            textBoxAmount.Text = "";
        }
```

4) Calculate and AddValues Functions

This Functions Executes When The System Values Going To Change, Thai Functions Will Execute Inside Main Functions These Are Some Nested Functions To Reuse Many Places.

```
private void CalculateTotal()
{
    total_q = quantity1 + quantity2 + quantity3;
    total_p = total1 + total2 + total3;
    lbl_TotalQuantity.Text = (total_q).ToString();
    lbl_TotalPrice.Text = (total_p).ToString();
}
private void ClearValues()
{
    dataGridView1.Rows.Clear();
    lbl_TotalQuantity.Text = "0";
    lbl_TotalPrice.Text = "0";
}

private void ValuesZero()
{
    quantity1 = 0;
    quantity2 = 0;
    quantity3 = 0;
    total1 = 0;
    total2 = 0;
    total3 = 0;
}

private void ValuesRearange()
{
    drinks1 += quantity1;
    drinks2 += quantity2;
    drinks3 += quantity3;
    lbl1.Text = drinks1.ToString();
    lbl2.Text = drinks2.ToString();
    lbl3.Text = drinks3.ToString();
}

```

5) Add Order Items On Grid Table Functions

This Functions Executes When Customers Click The Add Button To Add The Items Into The Table One By One.

Before Purchasing If The Customer Adds One Drink Item First Then He Wants To Add The Same Drink Again He Doesn't Need To Make A New Row For That Drink, That Drink Item Will Add On The Same Drink Item Before He Add.

```
private void AddDataOnTable(String drink_name, int quantity, int unit_price, int total_price)
{
    CheckRowValue(drink_name);
    DataGridViewRow row = (DataGridViewRow)dataGridView1.Rows[0].Clone();
    row.Cells[0].Value = drink_name;
    row.Cells[1].Value = quantity;
    row.Cells[2].Value = unit_price;
    row.Cells[3].Value = total_price;
    dataGridView1.Rows.Add(row);
}

private void CheckRowValue(String drink_name)
{
    foreach (DataGridViewRow row in dataGridView1.Rows)
    {
        bool v = row.Cells[0].Value == drink_name;
        if (v)
        {
            dataGridView1.Rows.Remove(row);
        }
    }
}
```