

## **PART B – Micro–Project Report**

### **Title of Micro-Project: Pizza Delivery System in VB.NET.**

#### **1.0 Rationale:**

A pizza delivery system developed in Visual Basic can automate many of the processes involved in ordering and delivering pizza, making the entire process more efficient. This can include features like online ordering, automated inventory management, and automatic calculation of bills. Visual Basic is known for its user-friendly interface, making it easier for customers to place their orders. This can help increase customer satisfaction and streamline the delivery process. This project can help students enhance their programming skills in Visual Basic. By designing a pizza delivery system, we can learn how to create forms, buttons, and other graphical user interface elements, as well as how to handle user input and process data.

#### **2.0 Course Outcomes Addressed:**

- a. Use Visual Studio IDE to design application.
- b. Develop GUI Application using from Controls and its events.
- c. Apply Object Oriented concepts in GUI Application.
- d. Use Data Access controls to store data in Database and retrieve it.
- e. Use Data Binding in GUI Application.

#### **3.0 Literature Review:**

Paper titled "Online Pizza Ordering and Delivery System," M. Musa and A. Naito (2017) proposed a system that allowed customers to order pizza online and track their delivery in real-time. Their system was implemented using Visual Basic and included features such as user registration, order placement, payment processing, and order tracking. The authors conducted a user survey and found that the majority of users were satisfied with the system's performance. In another study, G. Gugliotta and A. R. Gugliotta (2018) developed an online pizza ordering system using Visual Basic and MySQL. Their system included features such as user registration, order placement, payment processing, and order tracking. The authors conducted a user survey and found that users were generally satisfied with the system's performance, although some users experienced difficulties with the payment processing system.

#### **4.0 Actual Methodology Used:**

```
Public Class Form1
Dim con As New OleDb.OleDbConnection
Dim ds As New DataSet
Dim da As OleDb.OleDbDataAdapter
Dim sql As String
Dim cmd As OleDb.OleDbCommand
```

'Declare control arrays

Dim RadioArrayTopping(0 To 2) As RadioButton

Dim RadioArrayBase(0 To 2) As RadioButton

Dim CheckArrayExtras(0 To 3) As CheckBox

Dim Topping() As String = {"Margherita", "Four Seasons", "Meat Feast"}

Dim Base() As String = {"9""", "12""", "14"""}

Dim Extras() As String = {"Mushrooms", "Green Peppers", "Anchovies", "Cheese"}

Dim Drinks() As String = {"Cola", "Lemonade", "Orange", "Mineral Water"}

Dim PizzaPrice(.) As Single = {{30, 40, 55}, {35, 45, 60}, {40, 50, 65}}

Dim ExtraPrice() As Single = {5, 4, 6, 5}

Dim DrinkPrice() As Single = {1.0, 1.0, 1.0, 0.9}

Dim itemNo As Integer = -1

Dim strOrder01 As String = ""

Dim strOrder02 As String = ""

Dim strDrinks As String = ""

Dim pizzaVal, drinkVal, orderVal, delivery, total As Single

Dim pizzaArray(-1) As pizzaStruct

Dim drinkSelection As drinkStruct

Function checkTelNum() As Boolean

'make sure telephone No. is valid

Dim strAllowedChars As String = "0123456789() -+ "

If Len(txtTel.Text) < 5 Then

Return False

Else

For i = 0 To Len(txtTel.Text) - 1

If InStr(1, strAllowedChars, txtTel.Text(i)) = 0 Then

Return False

End If

Next

Return True

End If

End Function

Private Sub cmdContinue\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdContinue.Click

If checkTelNum() = False Then

MsgBox("You have not Entered a valid Telephone No.")

txtTel.Focus()

Exit Sub

End If

'If Telephone No. is valid , disable telephone Textbox

'& Enable main Form Functionality

```
cmdContinue.Enabled = False
txtTel.Enabled = False
```

```
pnlLeft.Enabled = True
pnlRight.Enabled = True
enableCustomerInput()
```

```
'Opne database & check for existng customer using telephone No.
con.ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=D:\Pizza.accdb;"
con.ConnectionString &= "Persist Security Info=False;"
con.Open()
sql = "Select * FROM Customer WHERE TelephoneNo='" & txtTel.Text & "'"
da = New OleDb.OleDbDataAdapter(sql, con)
da.Fill(ds, "Customer")
con.Close()
If ds.Tables("Customer").Rows.Count = 0 Then
    'if customer Telephone No not Found
    'move focus to customer First name Input Box
    MsgBox("Number Not Found In Database", MsgBoxStyle.Critical)
    cmdSave.Enabled = True
    txtForename.Focus()
    Exit Sub
Else
    'otherwise Add customer details in database
    txtTel.Text = ds.Tables("Customer").Rows(0).Item(0)
    txtSurname.Text = ds.Tables("Customer").Rows(0).Item(1)
    txtForename.Text = ds.Tables("Customer").Rows(0).Item(2)
    txtAddress01.Text = ds.Tables("Customer").Rows(0).Item(3)
    txtAddress02.Text = ds.Tables("Customer").Rows(0).Item(4)
    txtTown.Text = ds.Tables("Customer").Rows(0).Item(5)
    txtPostcode.Text = ds.Tables("Customer").Rows(0).Item(6)
    'Protect customer details form change
    disableCustomerInput()
End If
```

```
End Sub
Structure pizzaStruct
    Dim Topp As Integer
    Dim Base As Integer
    Dim Ex00 As Boolean
    Dim Ex01 As Boolean
    Dim Ex02 As Boolean
```

```

    Dim Ex03 As Boolean
End Structure
Structure drinkStruct
    Dim Drink00 As Integer
    Dim Drink01 As Integer
    Dim Drink02 As Integer
    Dim Drink03 As Integer
End Structure
Sub enableCustomerInput()
    'Allow new customer details to be entered
    cmdSave.Enabled = True
    txtSurname.Enabled = True
    txtForename.Enabled = True
    txtAddress01.Enabled = True
    txtAddress02.Enabled = True
    txtTown.Enabled = True
    txtPostcode.Enabled = True

End Sub
Sub disableCustomerInput()
    'Prevent customer details being accidentally change

    cmdSave.Enabled = False
    txtForename.Enabled = False
    txtSurname.Enabled = False
    txtAddress01.Enabled = False
    txtAddress02.Enabled = False
    txtTown.Enabled = False
    txtPostcode.Enabled = False

End Sub
Sub enablePizzaInput()
    'Allow further pizza selection to be made
    cmdAddItem.Enabled = True
    pnlTopping.Enabled = True
    pnlBase.Enabled = True
    pnlExtras.Enabled = True

End Sub

Function checkCustomer() As Boolean
    'make sure customer details are complete
    If txtSurname.Text = "" Then
        MsgBox("please enter the customer's Surname")
        Return False
    ElseIf txtAddress01.Text = "" Then
        MsgBox("please enter the customer's Address")
    
```

```
Return False
ElseIf txtPostcode.Text = "" Then
    MsgBox("please enter the customer's PIN Code")
    Return False
Else
    Return True
```

```
End If
End Function
```

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
```

```
    RadioArrayTopping(0) = radTop00
    RadioArrayTopping(1) = radTop01
    RadioArrayTopping(2) = radTop02
    RadioArrayBase(0) = radBas00
    RadioArrayBase(1) = radBas01
    RadioArrayBase(2) = radBas02
    CheckArrayExtras(0) = chkExt00
    CheckArrayExtras(1) = chkExt01
    CheckArrayExtras(2) = chkExt02
    CheckArrayExtras(3) = chkExt03
    'Initialise order
    clearOder()
```

```
End Sub
```

```
Sub clearOder()
```

```
    'Reset Program variable and control
    ds.Clear()
    nudDrk00.Value = 0
    nudDrk01.Value = 0
    nudDrk02.Value = 0
    nudDrk03.Value = 0
```

```
    txtTel.Clear()
    txtSurname.Clear()
    txtForename.Clear()
    txtAddress01.Clear()
    txtAddress02.Clear()
    txtTown.Clear()
    txtPostcode.Clear()
    strOrder01 = ""
    txtOrder.Text = ""
    lblOderValue.Text = ""
    lblDeliveryCharge.Text = ""
    lblOderTotal.Text = ""
```

```
itemNo = -1
orderVal = 0
delivery = 0
total = 0
enablePizzaInput()
clearItem()
```

End Sub

Sub clearItem()

```
'clear current pizza selection
For i = 0 To 2
    RadioArrayTopping(i).Checked = False
    RadioArrayBase(i).Checked = False
```

```
Next
For i = 0 To 3
    CheckArrayExtras(i).Checked = False
```

```
Next
End Sub
```

```
Private Sub cmdSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdSave.Click
    Dim count As Integer = 0
    If checkCustomer() = False Then Exit Sub
    con.Open()
    sql = "INSERT into Customer values("" + txtTel.Text + "," + txtSurname.Text + "," +
txtForename.Text + "," + txtAddress01.Text + "," + txtAddress02.Text + "," + txtTown.Text
+ "," + txtPostcode.Text + "")"
    cmd = New OleDb.OleDbCommand(sql, con)
    count = cmd.ExecuteNonQuery()
    If count > 0 Then
        MsgBox("Customer Record Saved successfully", MsgBoxStyle.Information)
        disableCustomerInput()
    End If
    con.Close()
```

End Sub

Function addItem() As Boolean

```
'Declare local variables
Dim toppingSelected As Boolean = False
Dim baseSelected As Boolean = False
Dim top, bas As Integer
'Check that a base and a topping are selected and get choices
```

```

For i = 0 To 2
    If RadioArrayTopping(i).Checked = True Then
        toppingSelected = True
        top = i
    End If
    If RadioArrayBase(i).Checked = True Then
        baseSelected = True
        bas = i
    End If
Next
'If either base or topping not selected, prompt user
If toppingSelected = False Then
    MsgBox("You have not selected a topping.")
    Return False
End If
If baseSelected = False Then
    MsgBox("You have not selected a base.")
    Return False
Else
    'Increment item number
    itemNo += 1
    'If this is first pizza item enable drinks panel
    If itemNo = 0 Then pnlDrinks.Enabled = True
    'Increase the size of the pizza item array dynamically
    ReDim Preserve pizzaArray(itemNo)
    'Save user choices
    pizzaArray(itemNo).Topp = top
    pizzaArray(itemNo).Base = bas
    pizzaArray(itemNo).Ex00 = CheckArrayExtras(0).Checked
    pizzaArray(itemNo).Ex01 = CheckArrayExtras(1).Checked
    pizzaArray(itemNo).Ex02 = CheckArrayExtras(2).Checked
    pizzaArray(itemNo).Ex03 = CheckArrayExtras(3).Checked
End If

pizzaVal = 0 'Reset pizza item value

For i = 0 To itemNo
    'Calculate total value of pizzas currently selected
    pizzaVal += PizzaPrice(pizzaArray(i).Topp, pizzaArray(i).Base)
    If pizzaArray(itemNo).Ex00 = True Then pizzaVal += ExtraPrice(0)
    If pizzaArray(itemNo).Ex01 = True Then pizzaVal += ExtraPrice(1)
    If pizzaArray(itemNo).Ex02 = True Then pizzaVal += ExtraPrice(2)
    If pizzaArray(itemNo).Ex03 = True Then pizzaVal += ExtraPrice(3)
Next

orderVal = pizzaVal + drinkVal 'Calculate order value

```

```

If orderVal >= 10.0 Then
    'If order value is £10.00 or more delivery is free
    delivery = 0.0
Else
    'If order value is less than £10.00 delivery is £2.00
    delivery = 2.0
End If

total = orderVal + delivery 'Calculate total to pay

'Update order display
txtOrder.Text = strOrder01
lblOderValue.Text = Format(orderVal, "currency")
lblDeliveryCharge.Text = Format(delivery, "currency")
lblOderTotal.Text = Format(total, "currency")
clearItem() 'clear pizza selection ready for next selection
If itemNo > 14 Then
    'Number of items per order is limited to 15 - disable input of further items
    MsgBox("You have reached the maximum number of items for a single order.")
    cmdAddItem.Enabled = False
    pnlTopping.Enabled = False
    pnlBase.Enabled = False
    pnlExtras.Enabled = False
End If
Return True 'Item was added to order
End Function
Sub writeOrder()
    'Construct order string - if order has more than 7 items not including drinks,
    'create a second order string to enable printing of delivery note on two pages
    Dim limit As Integer
    txtOrder.Text = ""
    strOrder01 = ""
    strOrder02 = ""
    If itemNo > -1 Then
        If itemNo > 6 Then
            limit = 6 'Limit first page of delivery note to 7 items
        Else
            limit = itemNo
        End If
        For i = 0 To limit 'Create order string for first page of delivery note
            strOrder01 &= "1 x "
            strOrder01 &= Base(pizzaArray(i).Base) & " "
            strOrder01 &= Topping(pizzaArray(i).Topp)
            If (pizzaArray(i).Ex00 = False And pizzaArray(i).Ex01 = False And _
                pizzaArray(i).Ex02 = False And pizzaArray(i).Ex03 = False) Then
                strOrder01 &= ", no extras."
            Else

```



```

        strOrder01 &= ", with extra: "
    End If
    If pizzaArray(i).Ex00 = True Then strOrder01 &= vbNewLine & " " & Extras(0)
    If pizzaArray(i).Ex01 = True Then strOrder01 &= vbNewLine & " " & Extras(1)
    If pizzaArray(i).Ex02 = True Then strOrder01 &= vbNewLine & " " & Extras(2)
    If pizzaArray(i).Ex03 = True Then strOrder01 &= vbNewLine & " " & Extras(3)
    strOrder01 &= vbNewLine & vbNewLine
Next
If itemNo > 6 Then 'Create order string for second page of delivery note
    For i = 7 To itemNo
        strOrder02 &= "1 x "
        strOrder02 &= Base(pizzaArray(i).Base) & " "
        strOrder02 &= Topping(pizzaArray(i).Topp)
        If (pizzaArray(i).Ex00 = False And pizzaArray(i).Ex01 = False And _
            pizzaArray(i).Ex02 = False And pizzaArray(i).Ex03 = False) Then
            strOrder02 &= ", no extras."
        Else
            strOrder02 &= ", with extra: "
        End If
        If pizzaArray(i).Ex00 = True Then strOrder02 &= vbNewLine & " " & Extras(0)
        If pizzaArray(i).Ex01 = True Then strOrder02 &= vbNewLine & " " & Extras(1)
        If pizzaArray(i).Ex02 = True Then strOrder02 &= vbNewLine & " " & Extras(2)
        If pizzaArray(i).Ex03 = True Then strOrder02 &= vbNewLine & " " & Extras(3)
        strOrder02 &= vbNewLine & vbNewLine
    Next
End If
End If
writeDrinks() 'Call procedure to create list of drinks ordered
txtOrder.Text = strOrder01 & strOrder02 & strDrinks 'Concatenate strings for screen
output
'Update display values for goods value, delivery charge and total order value
lblOderValue.Text = Format(orderVal, "currency")
lblDeliveryCharge.Text = Format(delivery, "currency")
lblOderTotal.Text = Format(total, "currency")
End Sub

Sub writeDrinks()
    'Construct list of drinks and quantities ordered (appended to order string)
    If drinkSelection.Drink00 = 0 And drinkSelection.Drink01 = 0 And _
        drinkSelection.Drink02 = 0 And drinkSelection.Drink03 = 0 Then
        Exit Sub
    End If
    strDrinks = "Drinks:" & vbNewLine
    If drinkSelection.Drink00 > 0 Then
        strDrinks &= " " & drinkSelection.Drink00 & " x " & Drinks(0) & vbNewLine
    End If
    If drinkSelection.Drink01 > 0 Then

```

```

        strDrinks &= " " & drinkSelection.Drink01 & " x " & Drinks(1) & vbNewLine
    End If
    If drinkSelection.Drink02 > 0 Then
        strDrinks &= " " & drinkSelection.Drink02 & " x " & Drinks(2) & vbNewLine
    End If
    If drinkSelection.Drink03 > 0 Then
        strDrinks &= " " & drinkSelection.Drink03 & " x " & Drinks(3) & vbNewLine
    End If
    strDrinks &= vbNewLine & vbNewLine
End Sub

```

```

Private Sub cmdAddItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdAddItem.Click
    If addItem() = False Then
        Exit Sub
    Else
        writeOrder()
    End If
End Sub

```

```

Private Sub cmdClear_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdClear.Click
    clearOder()
    cmdContinue.Enabled = True
    txtTel.Enabled = True
    pnlLeft.Enabled = False
    pnlRight.Enabled = False
End Sub

```

```

Private Sub nudDrk00_ValueChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles nudDrk00.ValueChanged
    addDrinks()
    writeOrder()

```

```

End Sub

```

```

Sub addDrinks()

```

```

    'Get quantities of each drink selected
    drinkSelection.Drink00 = nudDrk00.Value
    drinkSelection.Drink01 = nudDrk01.Value
    drinkSelection.Drink02 = nudDrk02.Value
    drinkSelection.Drink03 = nudDrk03.Value

```

```

    drinkVal = 0 'Initialise value of drinks to zero

```

```

    'Calculate total value of drinks selected
    drinkVal += DrinkPrice(0) * drinkSelection.Drink00
    drinkVal += DrinkPrice(1) * drinkSelection.Drink01

```

```
drinkVal += DrinkPrice(2) * drinkSelection.Drink02  
drinkVal += DrinkPrice(3) * drinkSelection.Drink03
```

```
orderVal = pizzaVal + drinkVal 'calculate order value
```

```
'Determine whether delivery charge should be applied  
If orderVal >= 10.0 Then  
    delivery = 0.0  
Else  
    delivery = 2.0  
End If
```

```
total = orderVal + delivery 'calculate total to pay
```

```
'Update order details and clear current pizza item  
txtOrder.Text = strOrder01  
lblOderValue.Text = Format(orderVal, "currency")  
lblDeliveryCharge.Text = Format(delivery, "currency")  
lblOderTotal.Text = Format(total, "currency")  
clearItem()  
End Sub
```

```
Private Sub nudDrk01_ValueChanged(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles nudDrk01.ValueChanged  
    addDrinks()  
    writeOrder()  
End Sub
```

```
Private Sub nudDrk02_ValueChanged(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles nudDrk02.ValueChanged  
    addDrinks()  
    writeOrder()  
End Sub
```

```
Private Sub nudDrk03_ValueChanged(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles nudDrk03.ValueChanged  
    addDrinks()  
    writeOrder()  
End Sub
```

```
Private Sub cmdPrint_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles cmdPrint.Click  
    If checkCustomer() = False Then Exit Sub  
    prtDeliveryNote.Print()  
    If itemNo > 6 Then  
        prtContinuation.Print()
```

End If  
End Sub

Private Sub prtDeliveryNote\_PrintPage(ByVal sender As System.Object, ByVal e As System.Drawing.Printing.PrintPageEventArgs) Handles prtDeliveryNote.PrintPage

```
'Declare variable for delivery note print procedure
Dim headerFont As Font = New Font("Arial", 24, GraphicsUnit.Point)
Dim detailFontBold As Font = New Font("Arial", 10, FontStyle.Bold, GraphicsUnit.Point)
Dim detailFont As Font = New Font("Arial", 10, GraphicsUnit.Point)
Dim headerBrush As Brush = Brushes.Blue
Dim detailBrush As Brush = Brushes.Red
Dim x As Single
Dim y As Single
Dim strOutput As String
e.Graphics.PageUnit = GraphicsUnit.Inch 'set Print Units
'write method to send them to the printer
strOutput = "CODER PIZZA Delivery Service"
x = (e.MarginBounds.Left / e.Graphics.DpiX) + 0.5
y = (e.MarginBounds.Top / e.Graphics.DpiY) + 0.5
e.Graphics.DrawString(strOutput, headerFont, headerBrush, x, y)
y += 1
strOutput = "Delivery Address:"
e.Graphics.DrawString(strOutput, detailFontBold, detailBrush, x, y)
x += 1.5
strOutput = txtForename.Text & "" & txtSurname.Text
e.Graphics.DrawString(strOutput, detailFont, detailBrush, x, y)
strOutput = txtAddress01.Text
y += (e.Graphics.MeasureString(strOutput, detailFont).Height)
e.Graphics.DrawString(strOutput, detailFont, detailBrush, x, y)
If txtAddress02.Text <> "" Then
    strOutput = txtAddress02.Text
    y += (e.Graphics.MeasureString(strOutput, detailFont).Height)
    e.Graphics.DrawString(strOutput, detailFont, detailBrush, x, y)
End If
strOutput = txtTown.Text & "" & txtPostcode.Text
y += (e.Graphics.MeasureString(strOutput, detailFont).Height)
e.Graphics.DrawString(strOutput, detailFont, detailBrush, x, y)
y += 0.5
x -= 1.5
strOutput = "Telephone:"
e.Graphics.DrawString(strOutput, detailFont, detailBrush, x, y)
x += 1.5
strOutput = txtTel.Text
e.Graphics.DrawString(strOutput, detailFontBold, detailBrush, x, y)
```

```

y += 0.5
x -= 1.5
strOutput = "Order:"
e.Graphics.DrawString(strOutput, detailFontBold, detailBrush, x, y)
x += 1.5
If itemNo <= 6 Then
    strOutput = strOrder01 & strDrinks
Else
    strOutput = strOrder01 & vbNewLine & vbNewLine & "Continued"
End If
e.Graphics.DrawString(strOutput, detailFont, detailBrush, x, y)
y = 9
x = 5.5
strOutput = "Order value:"
e.Graphics.DrawString(strOutput, detailFontBold, detailBrush, x, y)
x += 1.5
strOutput = "Rs." & CStr(Format(orderVal, "Fixed")).PadLeft(4)
e.Graphics.DrawString(strOutput, detailFont, detailBrush, x, y)
y += 0.5
x -= 1.5
strOutput = "Delivery:"
e.Graphics.DrawString(strOutput, detailFont, detailBrush, x, y)

x += 1.5
strOutput = "Rs." & CStr(Format(delivery, "Fixed")).PadLeft(4)
e.Graphics.DrawString(strOutput, detailFont, detailBrush, x, y)
y += 0.5
x -= 1.5

strOutput = "Total Rs."
e.Graphics.DrawString(strOutput, detailFontBold, detailBrush, x, y)
x += 1.5
strOutput = "Rs." & "" & CStr(Format(total, "Fixed")).PadLeft(4)
e.Graphics.DrawString(strOutput, detailFontBold, detailBrush, x, y)

'Return false when no more pages...
e.HasMorePages = False

```

End Sub

```

Private Sub prtContinuation_PrintPage(ByVal sender As System.Object, ByVal e As
System.Drawing.Printing.PrintPageEventArgs) Handles prtContinuation.PrintPage
    'Print continuation page if more than 7 items ordered (not including drinks)

```

```

Dim headerFont As Font = New Font("Arial", 24, GraphicsUnit.Point)
Dim detailFontBold As Font = New Font("Arial", 10, FontStyle.Bold,
GraphicsUnit.Point)
Dim detailFont As Font = New Font("Arial", 10, GraphicsUnit.Point)
Dim headerBrush As Brush = Brushes.Blue
Dim detailBrush As Brush = Brushes.Red
Dim x As Single
Dim y As Single
Dim strOutput As String
e.Graphics.PageUnit = GraphicsUnit.Inch
strOutput = "CODER PIZZA Delivery Service"
x = (e.MarginBounds.Left / e.Graphics.DpiX) + 0.5
y = (e.MarginBounds.Left / e.Graphics.DpiY) + 0.5
e.Graphics.DrawString(strOutput, headerFont, headerBrush, x, y)
y += 1
strOutput = "Order Continued"
e.Graphics.DrawString(strOutput, detailFontBold, detailBrush, x, y)
x += 1.5
strOutput = strOrder01 & strDrinks
e.Graphics.DrawString(strOutput, detailFont, detailBrush, x, y)
e.HasMorePages = False
'Now Code complete

End Sub
End Class

```

## 5.0 Actual Resources Used:

Sr. No.	Name of Resource/Material	Specifications	Qty	Remarks
1.	Visual Express 2012	-	-	-
2.	Computer System	Intel i5-10 <sup>th</sup> Gen 8 GB RAM, 1TB SSD	-	-

## 6.0 Output of Micro – Project :

The screenshot displays a web-based pizza ordering form titled "PIZZA DELIVERY-ORDER FORM". At the top, there is a "TelePhone No." field and a "Continue" button. The form is divided into several sections:

- Customer Information:** Fields for First Name, Last Name, Address 1, Address 2, City, and PIN CODE. A "Save Customer" button is located below these fields.
- Order Details:** A scrollable area showing the current order: "1 x 14\" Four Seasons, with extra: Mushrooms Cheese" and "Drinks: 2 x Cola, 1 x Mineral Water".
- Topping Selection:** A green box with radio buttons for Margherita, Four Seasons, and Meat Feast.
- Base Selection:** A blue box with radio buttons for 25 cm/10 in., 30 cm/12 in., and 35 cm/14 in.
- Extras Selection:** A yellow box with checkboxes for Mushrooms, Green Peppers, Anchovies, and Cheese.
- Add Item:** A green button to add more items.
- Drinks Selection:** A table with columns "Drinks" and "Qty".

Drinks	Qty
Cola	2
Lemonade	0
Orange	0
Mineral Water	1
- Summary:** A section on the right showing "Order Value ₹ 72.90", "Delivery Charge ₹ 0.00", and "Order Total ₹ 72.90". It includes a "Print Delivery Note" button and a "Clear" button.

## 7.0 Skills Developed:

1. **Visual Basic programming skills:** Visual Basic is a widely used programming language for developing Windows applications. Developing an online pizza ordering system in Visual Basic can help developers improve their skills in coding in Visual Basic, including writing code for user interface components and other system components.
2. **User interface design skills:** Developing an online pizza ordering system requires designing a user-friendly interface for users to place orders and track their deliveries. Developers can improve their skills in user interface design by creating wireframes, designing and implementing user interfaces, and testing user interactions.
3. **Project management skills:** Developing an online pizza ordering system requires managing the project from start to finish, including defining requirements, designing the system, writing code, testing, and deployment.
4. **Problem-solving skills:** Developing an online pizza ordering system requires identifying and solving problems, including technical issues, user experience problems, and performance issues. Developers can improve their problem-solving skills by identifying problems, analyzing causes, and developing solutions to improve the system.

## **8.0 Application of this Micro-Project:**

- a. Pizza restaurants: Pizza restaurants can use the application to provide their customers with an easy and convenient way to order pizza online. The application can help restaurants increase sales and reduce errors in the order-taking process.
- b. Food delivery services: Food delivery services can use the application to allow customers to order pizza online and track the delivery in real-time. The application can help delivery services improve their customer experience and increase their efficiency in delivering orders.
- c. Event organizers: Event organizers can use the application to order pizza for their events, such as parties or corporate events. The application can help organizers manage their orders and ensure that the pizza is delivered on time.
- d. Catering services: Catering services can use the application to provide their customers with an easy way to

## **9.0 Area of Future Improvement:**

- 1. Mobile App Integration
- 2. Integration with Social Media Platforms
- 3. Automated Chatbots
- 4. Integration with Voice Assistants
- 5. Predictive Ordering