

Default.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="IADLAB07.Default"%>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
<title>Lab 7 - Customer Segmentation</title>
```

```
<style>
```

```
    body {  
        font-family: Arial, sans-serif;  
        margin: 50px;  
        background-color: #f5f5f5;  
    }
```

```
    .container {  
        max-width: 500px;  
        margin: auto;  
        background: white;  
        padding: 30px;  
        border: 1px solid #ddd;  
        border-radius: 10px;  
        text-align: center;  
    }
```

```
    h1 {  
        color: #4CAF50;  
        border-bottom: 2px solid #4CAF50;  
        padding-bottom: 10px;  
    }
```

```
    .link {  
        display: block;  
        padding: 12px 20px;  
        margin: 15px 0;  
        background: #4CAF50;  
        color: white;  
        text-decoration: none;  
        border-radius: 5px;  
        font-size: 16px;  
    }
```

```
    .link:hover {  
        background: #45a049;  
    }
```

```
</style>
```

```

</head>

<body>

    <form id="form1" runat="server">

        <div class="container">

            <h1>Lab 7 - Customer Segmentation</h1>

            <a href="CustomerSegmentation.aspx" class="link">
                Customer Segmentation Analysis
            </a>

            <a href="TargetedMarketing.aspx" class="link">
                Targeted Marketing Automation
            </a>

        </div>

    </form>

</body>

</html>

```

Default.aspx.cs:

```

using System;

namespace IADLAB07
{
    public partial class Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}

```

CustomerSegmentation.aspx:

```

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="CustomerSegmentation.aspx.cs"
Inherits="IADLAB07.CustomerSegmentation" %>

```

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title>Customer Segmentation Analysis</title>

<style>

body {
    font-family: Arial, sans-serif;
    margin: 20px;
    background-color: #f5f5f5;
}

.container {
    max-width: 1300px;
    margin: auto;
    background: white;
    padding: 20px;
    border: 1px solid #ddd;
    border-radius: 8px;
}

h2 {
    color: #333;
    margin-top: 0;
    padding-bottom: 10px;
    border-bottom: 2px solid #4CAF50;
    text-align: center;
}

.stats {
    background: #e8f5e9;
    padding: 15px;
    margin: 20px 0;
    border: 1px solid #c8e6c9;
    border-radius: 5px;
    display: flex;
    justify-content: space-around;
    flex-wrap: wrap;
}

.stat-card {
    text-align: center;
    padding: 10px 20px;
    background: white;
    border-radius: 5px;
    min-width: 150px;
}

.stat-number {
    font-size: 24px;
```

```
    font-weight: bold;
    color: #4CAF50;
}

.stat-label {
    font-size: 12px;
    color: #666;
    margin-top: 5px;
}

.segment {
    margin-bottom: 30px;
    border: 1px solid #ddd;
    background: white;
    border-radius: 5px;
    overflow: hidden;
}

.segment-title {
    background: #4CAF50;
    color: white;
    padding: 12px 15px;
    font-weight: bold;
    font-size: 16px;
}

.segment-desc {
    font-size: 12px;
    font-weight: normal;
    margin-left: 10px;
    opacity: 0.9;
}

table {
    width: 100%;
    border-collapse: collapse;
}

th {
    background: #f2f2f2;
    padding: 10px;
    text-align: left;
    border: 1px solid #ddd;
    font-weight: bold;
}

td {
    padding: 8px 10px;
    border: 1px solid #ddd;
}

tr:hover {
    background-color: #f9f9f9;
}
```

```

.btn {
    padding: 8px 20px;
    margin: 5px;
    cursor: pointer;
    border: none;
    background: #4CAF50;
    color: white;
    border-radius: 4px;
}

.btn:hover {
    background: #45a049;
}

.btn-back {
    background: #666;
}

.footer {
    text-align: center;
    margin-top: 20px;
    padding-top: 20px;
    border-top: 1px solid #ddd;
    color: #666;
    font-size: 12px;
}

.refresh-btn {
    text-align: right;
    margin-bottom: 15px;
}

.button-group {
    text-align: center;
    margin-top: 20px;
}

</style>

</head>

<body>

<form id="form1" runat="server">

    <div class="container">

        <h2>Customer Segmentation Analysis</h2>

        <div class="refresh-btn">
            <asp:Button ID="btnRefresh"
                runat="server"
                Text="Refresh Data"

```

```

        CssClass="btn"
        OnClick="btnRefresh_Click" />
</div>

<div class="stats">

    <div class="stat-card">
        <div class="stat-number">
            <asp:Label ID="lblTotalCustomers" runat="server" Text="0"></asp:Label>
        </div>
        <div class="stat-label">Total Customers</div>
    </div>

    <div class="stat-card">
        <div class="stat-number">
            <asp:Label ID="lblActiveCustomers" runat="server" Text="0"></asp:Label>
        </div>
        <div class="stat-label">Active Customers</div>
    </div>

    <div class="stat-card">
        <div class="stat-number">
            <asp:Label ID="lblInactiveCustomers" runat="server" Text="0"></asp:Label>
        </div>
        <div class="stat-label">Inactive Customers</div>
    </div>

    <div class="stat-card">
        <div class="stat-number">
            $<asp:Label ID="lblTotalRevenue" runat="server" Text="0"></asp:Label>
        </div>
        <div class="stat-label">Total Revenue</div>
    </div>

</div>

<div class="segment">
    <div class="segment-title">
        Premium Customers
        <span class="segment-desc">(Spent more than $2000)</span>
    </div>

    <asp:GridView ID="gvPremium"
        runat="server"
        Width="100%"
        AutoGenerateColumns="true"
        EmptyDataText="No premium customers found">
    </asp:GridView>
</div>

<div class="segment">
    <div class="segment-title">
        Frequent Customers
        <span class="segment-desc">(More than 3 orders)</span>
    </div>

```

```
</div>

<asp:GridView ID="gvFrequent"
    runat="server"
    Width="100%"
    AutoGenerateColumns="true"
    EmptyDataText="No frequent customers found">
</asp:GridView>
</div>

<div class="segment">
    <div class="segment-title">
        Bulk Buyers
        <span class="segment-desc">(Average quantity more than 3)</span>
    </div>

    <asp:GridView ID="gvBulk"
        runat="server"
        Width="100%"
        AutoGenerateColumns="true"
        EmptyDataText="No bulk buyers found">
    </asp:GridView>
</div>

<div class="segment">
    <div class="segment-title">
        Occasional Customers
        <span class="segment-desc">(1 or fewer orders)</span>
    </div>

    <asp:GridView ID="gvOccasional"
        runat="server"
        Width="100%"
        AutoGenerateColumns="true"
        EmptyDataText="No occasional customers found">
    </asp:GridView>
</div>

<div class="segment">
    <div class="segment-title">
        Low Value Customers
        <span class="segment-desc">(Spent less than $500)</span>
    </div>

    <asp:GridView ID="gvLow"
        runat="server"
        Width="100%"
        AutoGenerateColumns="true"
        EmptyDataText="No low value customers found">
    </asp:GridView>
</div>

<div class="segment">
    <div class="segment-title">
```

```
        Inactive Customers
        <span class="segment-desc">(No orders in last 6 months)</span>
    </div>

    <asp:GridView ID="gvInactive"
        runat="server"
        Width="100%"
        AutoGenerateColumns="true"
        EmptyDataText="No inactive customers found">
    </asp:GridView>
</div>

<div class="segment">
    <div class="segment-title">
        Loyal Customers
        <span class="segment-desc">(Frequent + Premium)</span>
    </div>

    <asp:GridView ID="gvLoyal"
        runat="server"
        Width="100%"
        AutoGenerateColumns="true"
        EmptyDataText="No loyal customers found">
    </asp:GridView>
</div>

<div class="footer">

    <p>
        Generated on:
        <asp:Label ID="lblGeneratedDate" runat="server"></asp:Label>
    </p>

</div>

<div class="button-group">

    <asp:Button ID="btnBack"
        runat="server"
        Text="Back To Dashboard"
        CssClass="btn btn-back"
        OnClick="btnBack_Click" />

</div>

</div>

</form>

</body>

</html>
```

CustomerSegmentation.aspx.cs:

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

namespace IADLAB07
{
    public partial class CustomerSegmentation : System.Web.UI.Page
    {
        string connStr =
            ConfigurationManager.ConnectionStrings["PVFCCConnectionString"].ConnectionString;

        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                LoadAllSegments();
                lblGeneratedDate.Text = DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss");
            }
        }

        protected void btnRefresh_Click(object sender, EventArgs e)
        {
            LoadAllSegments();
            lblGeneratedDate.Text = DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss");
        }

        protected void btnBack_Click(object sender, EventArgs e)
        {
            Response.Redirect("Default.aspx");
        }

        private void LoadAllSegments()
        {
            try
            {
                using (SqlConnection con = new SqlConnection(connStr))
                {
                    con.Open();

                    // PREMIUM CUSTOMERS
                    string sqlPremium = @"
SELECT
    C.Customer_Id AS [ID],
    C.Customer_Name AS [Customer Name],
    C.Customer_City AS [City],
    ISNULL(SUM(P.Standard_Price * OL.Ordered_Quantity),0) AS [Total Spent],
    COUNT(DISTINCT O.Order_Id) AS [Orders]
FROM CUSTOMER_t C
LEFT JOIN Users U ON C.Customer_Id = U.Customer_Id
LEFT JOIN Orders_t O ON U.UserID = O.UserID
LEFT JOIN Order_line_t OL ON O.Order_Id = OL.Order_Id";
```

```

LEFT JOIN PRODUCT_t P ON OL.Product_Id = P.Product_Id
GROUP BY C.Customer_Id, C.Customer_Name, C.Customer_City
HAVING ISNULL(SUM(P.Standard_Price * OL.Ordered_Quantity),0) > 2000
ORDER BY [Total Spent] DESC";

```

```

LoadData(con, sqlPremium, gvPremium);

```

// FREQUENT CUSTOMERS

```

string sqlFrequent = @"

```

```

SELECT
    C.Customer_Id AS [ID],
    C.Customer_Name AS [Customer Name],
    C.Customer_City AS [City],
    COUNT(DISTINCT O.Order_Id) AS [Orders],
    ISNULL(SUM(P.Standard_Price * OL.Ordered_Quantity),0) AS [Total Spent]
FROM CUSTOMER_t C
LEFT JOIN Users U ON C.Customer_Id = U.Customer_Id
LEFT JOIN Orders_t O ON U.UserID = O.UserID
LEFT JOIN Order_line_t OL ON O.Order_Id = OL.Order_Id
LEFT JOIN PRODUCT_t P ON OL.Product_Id = P.Product_Id
GROUP BY C.Customer_Id, C.Customer_Name, C.Customer_City
HAVING COUNT(DISTINCT O.Order_Id) > 3
ORDER BY [Orders] DESC";

```

```

LoadData(con, sqlFrequent, gvFrequent);

```

// BULK BUYERS

```

string sqlBulk = @"

```

```

SELECT
    C.Customer_Id AS [ID],
    C.Customer_Name AS [Customer Name],
    C.Customer_City AS [City],
    CAST(AVG(CAST(OL.Ordered_Quantity AS FLOAT)) AS DECIMAL(10,2)) AS [Avg Quantity],
    SUM(OL.Ordered_Quantity) AS [Total Items]
FROM CUSTOMER_t C
INNER JOIN Users U ON C.Customer_Id = U.Customer_Id
INNER JOIN Orders_t O ON U.UserID = O.UserID
INNER JOIN Order_line_t OL ON O.Order_Id = OL.Order_Id
GROUP BY C.Customer_Id, C.Customer_Name, C.Customer_City
HAVING AVG(CAST(OL.Ordered_Quantity AS FLOAT)) > 3
ORDER BY [Avg Quantity] DESC";

```

```

LoadData(con, sqlBulk, gvBulk);

```

// OCCASIONAL CUSTOMERS

```

string sqlOccasional = @"

```

```

SELECT
    C.Customer_Id AS [ID],
    C.Customer_Name AS [Customer Name],
    C.Customer_City AS [City],
    ISNULL(COUNT(DISTINCT O.Order_Id),0) AS [Orders],
    ISNULL(SUM(P.Standard_Price * OL.Ordered_Quantity),0) AS [Total Spent]
FROM CUSTOMER_t C
LEFT JOIN Users U ON C.Customer_Id = U.Customer_Id

```

```

LEFT JOIN Orders_t O ON U.UserID = O.UserID
LEFT JOIN Order_line_t OL ON O.Order_Id = OL.Order_Id
LEFT JOIN PRODUCT_t P ON OL.Product_Id = P.Product_Id
GROUP BY C.Customer_Id, C.Customer_Name, C.Customer_City
HAVING ISNULL(COUNT(DISTINCT O.Order_Id),0) <= 1
ORDER BY [Orders] DESC";

```

```
LoadData(con, sqlOccasional, gvOccasional);
```

// LOW VALUE CUSTOMERS

```
string sqlLow = @"
```

```

SELECT
    C.Customer_Id AS [ID],
    C.Customer_Name AS [Customer Name],
    C.Customer_City AS [City],
    ISNULL(SUM(P.Standard_Price * OL.Ordered_Quantity),0) AS [Total Spent],
    ISNULL(COUNT(DISTINCT O.Order_Id),0) AS [Orders]
FROM CUSTOMER_t C
LEFT JOIN Users U ON C.Customer_Id = U.Customer_Id
LEFT JOIN Orders_t O ON U.UserID = O.UserID
LEFT JOIN Order_line_t OL ON O.Order_Id = OL.Order_Id
LEFT JOIN PRODUCT_t P ON OL.Product_Id = P.Product_Id
GROUP BY C.Customer_Id, C.Customer_Name, C.Customer_City
HAVING ISNULL(SUM(P.Standard_Price * OL.Ordered_Quantity),0) < 500
ORDER BY [Total Spent] ASC";

```

```
LoadData(con, sqlLow, gvLow);
```

// INACTIVE CUSTOMERS

```
string sqlInactive = @"
```

```

SELECT
    C.Customer_Id AS [ID],
    C.Customer_Name AS [Customer Name],
    C.Customer_City AS [City],
    CASE
        WHEN MAX(O.Order_Date) IS NULL THEN 'Never Ordered'
        ELSE CONVERT(VARCHAR, MAX(O.Order_Date), 106)
    END AS [Last Order]
FROM CUSTOMER_t C
LEFT JOIN Users U ON C.Customer_Id = U.Customer_Id
LEFT JOIN Orders_t O ON U.UserID = O.UserID
GROUP BY C.Customer_Id, C.Customer_Name, C.Customer_City
HAVING MAX(O.Order_Date) IS NULL
OR MAX(O.Order_Date) < DATEADD(MONTH,-6,GETDATE());

```

```
LoadData(con, sqlInactive, gvInactive);
```

// LOYAL CUSTOMERS

```
string sqlLoyal = @"
```

```

SELECT
    C.Customer_Id AS [ID],
    C.Customer_Name AS [Customer Name],
    C.Customer_City AS [City],
    COUNT(DISTINCT O.Order_Id) AS [Orders],

```

```

        ISNULL(SUM(P.Standard_Price * OL.Ordered_Quantity),0) AS [Total Spent]
FROM CUSTOMER_t C
LEFT JOIN Users U ON C.Customer_Id = U.Customer_Id
LEFT JOIN Orders_t O ON U.UserID = O.UserID
LEFT JOIN Order_line_t OL ON O.Order_Id = OL.Order_Id
LEFT JOIN PRODUCT_t P ON OL.Product_Id = P.Product_Id
GROUP BY C.Customer_Id, C.Customer_Name, C.Customer_City
HAVING COUNT(DISTINCT O.Order_Id) > 3
AND ISNULL(SUM(P.Standard_Price * OL.Ordered_Quantity),0) > 2000";

LoadData(con, sqlLoyal, gvLoyal);

    UpdateStatistics(con);
}
}
catch (Exception ex)
{
    Response.Write("<div style='color:red;'>Error: " + ex.Message + "</div>");
}
}

private void UpdateStatistics(SqlConnection con)
{
    SqlCommand cmdTotal =
        new SqlCommand("SELECT COUNT(*) FROM CUSTOMER_t", con);

    int totalCustomers =
        Convert.ToInt32(cmdTotal.ExecuteScalar());

    lblTotalCustomers.Text = totalCustomers.ToString();

    SqlCommand cmdActive = new SqlCommand(@"
        SELECT COUNT(DISTINCT C.Customer_Id)
        FROM CUSTOMER_t C
        INNER JOIN Users U ON C.Customer_Id = U.Customer_Id
        INNER JOIN Orders_t O ON U.UserID = O.UserID", con);

    int activeCustomers =
        Convert.ToInt32(cmdActive.ExecuteScalar());

    lblActiveCustomers.Text = activeCustomers.ToString();

    lblInactiveCustomers.Text =
        (totalCustomers - activeCustomers).ToString();

    SqlCommand cmdRevenue = new SqlCommand(@"
        SELECT ISNULL(SUM(P.Standard_Price * OL.Ordered_Quantity),0)
        FROM Order_line_t OL
        INNER JOIN PRODUCT_t P
        ON OL.Product_Id = P.Product_Id", con);

    decimal revenue =
        Convert.ToDecimal(cmdRevenue.ExecuteScalar());

```

```

        lblTotalRevenue.Text = revenue.ToString("N2");
    }

    private void LoadData(SqlConnection con,
        string query,
        System.Web.UI.WebControls.GridView gv)
    {
        SqlDataAdapter da = new SqlDataAdapter(query, con);

        DataTable dt = new DataTable();

        da.Fill(dt);

        gv.DataSource = dt;
        gv.DataBind();
    }
}

```

TargetedMarketing.aspx:

```

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="TargetedMarketing.aspx.cs"
Inherits="IADLAB07.TargetedMarketing" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

    <title>Targeted Marketing</title>

    <style>

        body {
            font-family: Arial;
            margin: 20px;
            background: #f5f5f5;
        }

        .container {
            max-width: 1000px;
            margin: auto;
            background: white;
            padding: 20px;
            border: 1px solid #ddd;
        }

        h2 {

```

```
    color: #333;
    border-bottom: 2px solid #4CAF50;
    padding-bottom: 10px;
}

.segment {
    margin-bottom: 20px;
    padding: 15px;
    border: 1px solid #ddd;
    background: #fafafa;
}

.segment h3 {
    margin: 0 0 10px 0;
    color: #4CAF50;
}

.btn {
    padding: 8px 15px;
    margin: 5px;
    cursor: pointer;
    border: none;
    background: #4CAF50;
    color: white;
}

.btn-back {
    background: #666;
}

table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 10px;
}

th {
    background: #f2f2f2;
    padding: 8px;
    text-align: left;
    border: 1px solid #ddd;
}

td {
    padding: 8px;
    border: 1px solid #ddd;
}

.button-group {
    text-align: center;
    margin-top: 20px;
}

</style>
```

```
</head>

<body>

    <form id="form1" runat="server">

        <div class="container">

            <h2>Targeted Marketing</h2>

            <!-- ROYAL CUSTOMERS -->

            <div class="segment">

                <h3>Royal Customers (Premium + Frequent)</h3>

                <p>
                    Offer: Complimentary Gift + 25% Discount
                </p>

                <asp:Button ID="btnRoyal"
                    runat="server"
                    Text="Send To Royal Customers"
                    CssClass="btn"
                    OnClick="btnRoyal_Click" />

                <br /><br />

                <asp:Label ID="lblRoyal" runat="server"></asp:Label>

                <asp:GridView ID="gvRoyal"
                    runat="server"
                    Width="100%"
                    AutoGenerateColumns="true"
                    EmptyDataText="No royal customers">
                </asp:GridView>

            </div>

            <!-- PREMIUM CUSTOMERS -->

            <div class="segment">

                <h3>Premium Customers</h3>

                <p>
                    Offer: 15% Discount + VIP Access
                </p>

                <asp:Button ID="btnPremium"
                    runat="server"
                    Text="Send To Premium Customers"
                    CssClass="btn"
```

```
OnClick="btnPremium_Click" />

<br /><br />

<asp:Label ID="lblPremium" runat="server"></asp:Label>

<asp:GridView ID="gvPremium"
    runat="server"
    Width="100%"
    AutoGenerateColumns="true"
    EmptyDataText="No premium customers">
</asp:GridView>

</div>

<!-- FREQUENT CUSTOMERS -->

<div class="segment">

    <h3>Frequent Customers</h3>

    <p>
        Offer: Loyalty Points
    </p>

    <asp:Button ID="btnFrequent"
        runat="server"
        Text="Send To Frequent Customers"
        CssClass="btn"
        OnClick="btnFrequent_Click" />

    <br /><br />

    <asp:Label ID="lblFrequent" runat="server"></asp:Label>

    <asp:GridView ID="gvFrequent"
        runat="server"
        Width="100%"
        AutoGenerateColumns="true"
        EmptyDataText="No frequent customers">
    </asp:GridView>

</div>

<!-- INACTIVE CUSTOMERS -->

<div class="segment">

    <h3>Inactive Customers</h3>

    <p>
        Offer: Welcome Back Discount
    </p>
```

```

<asp:Button ID="btnInactive"
    runat="server"
    Text="Send To Inactive Customers"
    CssClass="btn"
    OnClick="btnInactive_Click" />

<br /><br />

<asp:Label ID="lblInactive" runat="server"></asp:Label>

<asp:GridView ID="gvInactive"
    runat="server"
    Width="100%"
    AutoGenerateColumns="true"
    EmptyDataText="No inactive customers">
</asp:GridView>

</div>

<!-- LOW VALUE CUSTOMERS -->

<div class="segment">

    <h3>Low Value Customers</h3>

    <p>
        Offer: Free Shipping
    </p>

    <asp:Button ID="btnLow"
        runat="server"
        Text="Send To Low Value Customers"
        CssClass="btn"
        OnClick="btnLow_Click" />

    <br /><br />

    <asp:Label ID="lblLow" runat="server"></asp:Label>

    <asp:GridView ID="gvLow"
        runat="server"
        Width="100%"
        AutoGenerateColumns="true"
        EmptyDataText="No low value customers">
    </asp:GridView>

</div>

<!-- BULK BUYERS -->

<div class="segment">

    <h3>Bulk Buyers</h3>

```

```

<p>
    Offer: Volume Discount
</p>

<asp:Button ID="btnBulk"
    runat="server"
    Text="Send To Bulk Buyers"
    CssClass="btn"
    OnClick="btnBulk_Click" />

<br /><br />

<asp:Label ID="lblBulk" runat="server"></asp:Label>

<asp:GridView ID="gvBulk"
    runat="server"
    Width="100%"
    AutoGenerateColumns="true"
    EmptyDataText="No bulk buyers">
</asp:GridView>

</div>

<div class="button-group">

    <asp:Button ID="btnBack"
        runat="server"
        Text="Back To Dashboard"
        CssClass="btn btn-back"
        OnClick="btnBack_Click" />

</div>

</div>

</form>

</body>

</html>

```

TargetedMarketing.aspx.cs:

```

using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

```

```

namespace IADLAB07
{
    public partial class TargetedMarketing : System.Web.UI.Page
    {
        string connStr =
            ConfigurationManager.ConnectionStrings["PVFCCConnectionString"].ConnectionString;

        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                LoadCustomers();
            }
        }

        private void LoadCustomers()
        {
            LoadRoyal();
            LoadPremium();
            LoadFrequent();
            LoadInactive();
            LoadLowValue();
            LoadBulk();
        }

        private void LoadRoyal()
        {
            string sql = @"
            SELECT
                C.Customer_Name,
                C.Customer_City,
                COUNT(O.Order_Id) AS Orders,
                SUM(P.Standard_Price * OL.Ordered_Quantity) AS TotalSpent
            FROM CUSTOMER_t C
            JOIN Users U ON C.Customer_Id = U.Customer_Id
            JOIN Orders_t O ON U.UserID = O.UserID
            JOIN Order_line_t OL ON O.Order_Id = OL.Order_Id
            JOIN PRODUCT_t P ON OL.Product_Id = P.Product_Id
            GROUP BY C.Customer_Name, C.Customer_City
            HAVING COUNT(O.Order_Id) > 3
            AND SUM(P.Standard_Price * OL.Ordered_Quantity) > 2000";

            LoadData(sql, gvRoyal);
        }

        private void LoadPremium()
        {
            string sql = @"
            SELECT
                C.Customer_Name,
                C.Customer_City,
                SUM(P.Standard_Price * OL.Ordered_Quantity) AS TotalSpent
            FROM CUSTOMER_t C
            JOIN Users U ON C.Customer_Id = U.Customer_Id

```

```

        JOIN Orders_t O ON U.UserID = O.UserID
        JOIN Order_line_t OL ON O.Order_Id = OL.Order_Id
        JOIN PRODUCT_t P ON OL.Product_Id = P.Product_Id
        GROUP BY C.Customer_Name, C.Customer_City
        HAVING SUM(P.Standard_Price * OL.Ordered_Quantity) > 2000";

        LoadData(sql, gvPremium);
    }

    private void LoadFrequent()
    {
        string sql = @"
        SELECT
            C.Customer_Name,
            C.Customer_City,
            COUNT(O.Order_Id) AS Orders
        FROM CUSTOMER_t C
        JOIN Users U ON C.Customer_Id = U.Customer_Id
        JOIN Orders_t O ON U.UserID = O.UserID
        GROUP BY C.Customer_Name, C.Customer_City
        HAVING COUNT(O.Order_Id) > 3";

        LoadData(sql, gvFrequent);
    }

    private void LoadInactive()
    {
        string sql = @"
        SELECT
            C.Customer_Name,
            C.Customer_City,
            MAX(O.Order_Date) AS LastOrder
        FROM CUSTOMER_t C
        JOIN Users U ON C.Customer_Id = U.Customer_Id
        LEFT JOIN Orders_t O ON U.UserID = O.UserID
        GROUP BY C.Customer_Name, C.Customer_City
        HAVING MAX(O.Order_Date) < DATEADD(MONTH,-6,GETDATE())
        OR MAX(O.Order_Date) IS NULL";

        LoadData(sql, gvInactive);
    }

    private void LoadLowValue()
    {
        string sql = @"
        SELECT
            C.Customer_Name,
            C.Customer_City,
            ISNULL(SUM(P.Standard_Price * OL.Ordered_Quantity),0) AS TotalSpent
        FROM CUSTOMER_t C
        JOIN Users U ON C.Customer_Id = U.Customer_Id
        LEFT JOIN Orders_t O ON U.UserID = O.UserID
        LEFT JOIN Order_line_t OL ON O.Order_Id = OL.Order_Id
        LEFT JOIN PRODUCT_t P ON OL.Product_Id = P.Product_Id

```

```
GROUP BY C.Customer_Name, C.Customer_City  
HAVING ISNULL(SUM(P.Standard_Price * OL.Ordered_Quantity),0) < 500";
```

```
LoadData(sql, gvLow);  
}
```

```
private void LoadBulk()  
{  
    string sql = @"  
    SELECT  
        C.Customer_Name,  
        C.Customer_City,  
        AVG(CAST(OL.Ordered_Quantity AS FLOAT)) AS AvgQuantity  
    FROM CUSTOMER_t C  
    JOIN Users U ON C.Customer_Id = U.Customer_Id  
    JOIN Orders_t O ON U.UserID = O.UserID  
    JOIN Order_line_t OL ON O.Order_Id = OL.Order_Id  
    GROUP BY C.Customer_Name, C.Customer_City  
    HAVING AVG(CAST(OL.Ordered_Quantity AS FLOAT)) > 3";
```

```
LoadData(sql, gvBulk);  
}
```

```
protected void btnRoyal_Click(object sender, EventArgs e)  
{  
    int count = gvRoyal.Rows.Count;  
  
    if (count > 0)  
    {  
        lblRoyal.Text =  
            "Offer sent to " + count +  
            " royal customers!";  
    }  
    else  
    {  
        lblRoyal.Text =  
            "No royal customers found.";  
    }  
}
```

```
protected void btnPremium_Click(object sender, EventArgs e)  
{  
    int count = gvPremium.Rows.Count;  
  
    if (count > 0)  
    {  
        lblPremium.Text =  
            "Offer sent to " + count +  
            " premium customers!";  
    }  
    else  
    {  
        lblPremium.Text =  
            "No premium customers found.";
```

```

    }
}

protected void btnFrequent_Click(object sender, EventArgs e)
{
    int count = gvFrequent.Rows.Count;

    if (count > 0)
    {
        lblFrequent.Text =
            "Offer sent to " + count +
            " frequent customers!";
    }
    else
    {
        lblFrequent.Text =
            "No frequent customers found.";
    }
}

protected void btnInactive_Click(object sender, EventArgs e)
{
    int count = gvInactive.Rows.Count;

    if (count > 0)
    {
        lblInactive.Text =
            "Offer sent to " + count +
            " inactive customers!";
    }
    else
    {
        lblInactive.Text =
            "No inactive customers found.";
    }
}

protected void btnLow_Click(object sender, EventArgs e)
{
    int count = gvLow.Rows.Count;

    if (count > 0)
    {
        lblLow.Text =
            "Offer sent to " + count +
            " low value customers!";
    }
    else
    {
        lblLow.Text =
            "No low value customers found.";
    }
}

```

```

protected void btnBulk_Click(object sender, EventArgs e)
{
    int count = gvBulk.Rows.Count;

    if (count > 0)
    {
        lblBulk.Text =
            "Offer sent to " + count +
            " bulk buyers!";
    }
    else
    {
        lblBulk.Text =
            "No bulk buyers found.";
    }
}

protected void btnBack_Click(object sender, EventArgs e)
{
    Response.Redirect("Default.aspx");
}

private void LoadData(string query,
    System.Web.UI.WebControls.GridView gv)
{
    using (SqlConnection con =
        new SqlConnection(connStr))
    {
        SqlDataAdapter da =
            new SqlDataAdapter(query, con);

        DataTable dt = new DataTable();

        da.Fill(dt);

        gv.DataSource = dt;
        gv.DataBind();
    }
}
}
}

```