```
 1  using System;
 2  using System.Collections.Generic;
 3  using System.Linq;
 4  using System.Web;
 5  using System.Web.UI;
 6  using System.Web.UI.WebControls;
 7  using System.Text;
 8  using System.Data.OleDb;
 9
10  namespace GIP_Smart.Pages
11  {
12      public partial class Lights : System.Web.UI.Page
13      {
14          Pages.Classes.Netwerkcommunicatie netwerkcommunicatie = new    ⇄
                Pages.Classes.Netwerkcommunicatie();
15
16          string query;
17          string connstring = connStrings.connString;
18
19
20          private string val1 = "90deg";
21
22          public string Val1
23          {
24              get { return val1; }
25              set { val1 = value; }
26          }
27
28          private string val2 = "90deg";
29
30          public string Val2
31          {
32              get { return val2; }
33              set { val2 = value; }
34          }
35
36          private string colorCode = "#ffffff";
37
38          public string ColorCode
39          {
40              get { return colorCode; }
41              set { colorCode = value; }
42          }
43
44          protected void Page_Load(object sender, EventArgs e)
45          {
46
47
48              netwerkcommunicatie.Network();
49              netwerkcommunicatie.mqttClient.MqttMsgPublishReceived +=   ⇄
                    netwerkcommunicatie.client_receivedMessage;
50
51              ProgressText.InnerText = "0°C";
```

```
52
53              query = "SELECT Temperatuur FROM Verlichting WHERE id =    ⮑
                    (SELECT max(id) FROM Verlichting);";
54              DisplayData(connstring, query);
55
56          }
57
58          private void CalculateActiveUsersAngle(int TotalUser)
59          {
60              if (TotalUser == 0)
61              {
62                  Val2 = "90deg";
63                  Val1 = "90deg";
64                  ColorCode = "#ffffff";
65              }
66              else if (TotalUser < 50 && TotalUser > 0)
67              {
68                  double percentageOfWholeAngle = 360 * (Convert.ToDouble ⮑
                        (TotalUser) / 100);
69                  Val2 = (90 + percentageOfWholeAngle).ToString() +     ⮑
                        "deg";
70                  Val1 = "90deg";
71                  ColorCode = "#ffffff";
72              }
73              else if (TotalUser > 50 && TotalUser < 100)
74              {
75                  double percentage = 360 * (Convert.ToDouble           ⮑
                        (TotalUser) / 100);
76                  Val1 = (percentage - 270).ToString() + "deg";
77                  Val2 = "270deg";
78                  ColorCode = "#18bc9c";
79              }
80              else if (TotalUser == 50)
81              {
82                  Val1 = "-90deg";
83                  Val2 = "270deg";
84                  ColorCode = "#18bc9c";
85              }
86              else if (TotalUser >= 100)
87              {
88                  Val1 = "90deg";
89                  Val2 = "270deg";
90                  ColorCode = "#18bc9c";
91              }
92
93              ProgressText.InnerText = TotalUser + "°C";
94          }
95
96          protected void btn_On_Click(object sender, EventArgs e)
97          {
98              if (netwerkcommunicatie.mqttClient != null &&            ⮑
                    netwerkcommunicatie.mqttClient.IsConnected)
99              {
```

```csharp
100                 netwerkcommunicatie.mqttClient.Publish("topic1",
                        Encoding.UTF8.GetBytes("on"));
101             }
102         }
103
104         protected void btn_Off_Click(object sender, EventArgs e)
105         {
106             if (netwerkcommunicatie.mqttClient != null &&
                  netwerkcommunicatie.mqttClient.IsConnected)
107             {
108                 netwerkcommunicatie.mqttClient.Publish("topic1",
                        Encoding.UTF8.GetBytes("off"));
109             }
110         }
111
112         protected void DisplayData(string connstring, string query)
113         {
114             OleDbConnection connection = new OleDbConnection();
115             connection.ConnectionString = connstring;
116
117
118             try
119             {
120                 connection.Open();
121
122                 OleDbCommand command = new OleDbCommand();
123                 command.Connection = connection;
124                 command.CommandText = query;
125
126                 OleDbDataReader reader = command.ExecuteReader();
127
128                 while (reader.Read())
129                 {
130                     CalculateActiveUsersAngle(Convert.ToInt32(reader
                        [0]));
131                 }
132             }
133
134             catch (OleDbException error)
135             {
136                 Console.WriteLine(error.Message);
137
138             }
139
140             finally
141             {
142                 connection.Close();
143             }
144         }
145
146     }
147 }
```