BIE 33103

DOTNET PROGRAMMING

LAB REPORT 6

DR. NUREZAYANA BINTI ZAINAL

Name	Matric
Muhamad Irfan Fitri Bin Suhaimi	AI180293

Part 1

lab_6.aspx

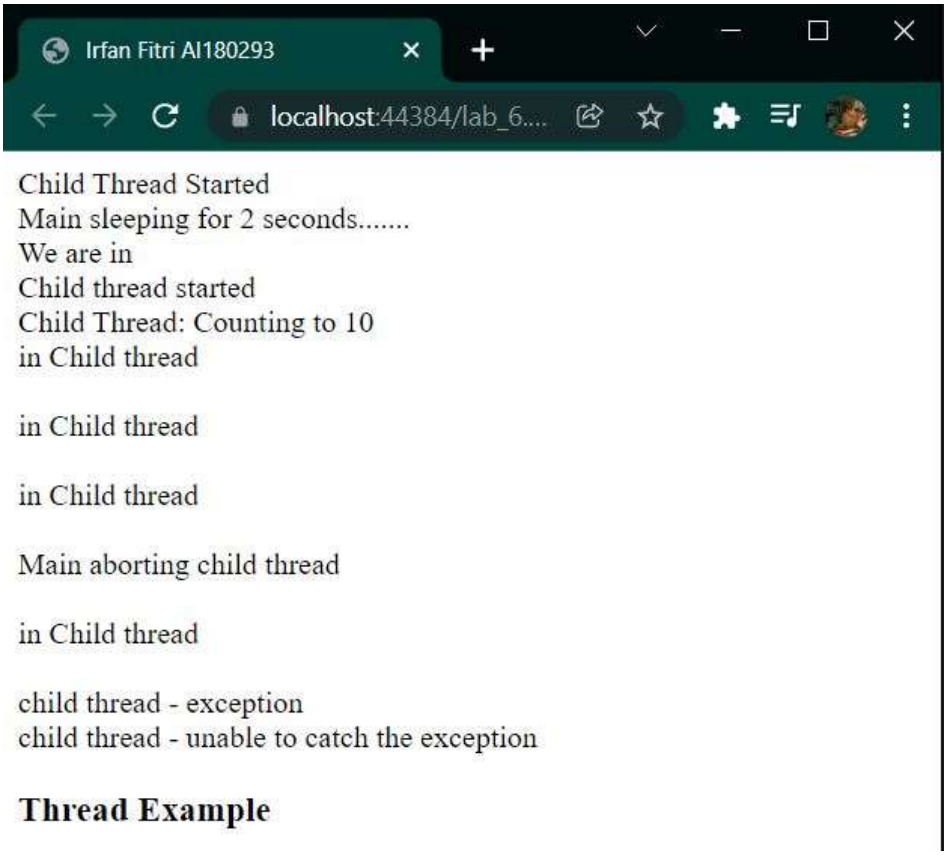
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="lab_6.aspx.cs"
Inherits="Lab_6.Lab_6" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

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

<head runat="server">
    <title>
        Irfan Fitri AI180293
    </title>
</head>

<body>
    <form id="form1" runat="server">
        <div>
            <h3>Thread Example</h3>
       </div>

        <!-- <asp:Label ID="lblmessage" runat="server" Text="Label">
              </asp:Label>-->
    </form>
</body>
</html>
```

lab_6.aspx.cs

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;

using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;
using System.Threading;
```

```
namespace Lab_6
{
    public partial class Lab_6 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            ThreadStart childthreat = new ThreadStart(childthreadcall);
            Response.Write("Child Thread Started <br/>");
            Thread child = new Thread(childthreat);

            child.Start();

            Response.Write("Main sleeping for 2 seconds.......<br/>");
            Thread.Sleep(2000);
            Response.Write("<br/>Main aborting child thread<br/>");

            child.Abort();
        }

        public void childthreadcall()
        {

                Response.Write("We are in");
                try
                {
                    Response.Write("<br />Child thread started <br/>");
                    Response.Write("Child Thread: Counting to 10");

                    for (int i = 0; i < 10; i++)
                    {
                        Thread.Sleep(500);
                        Response.Write("<br/> in Child thread </br>");
                    }

                    Response.Write("<br/> child thread finished");
                }
                catch (ThreadAbortException e)
                {
                    Response.Write("<br /> child thread - exception");
                }
                finally
                {
                    Response.Write("<br /> child thread - unable to catch the
exception");
                }


        }
    }
}
```

Output



On Page_Load, ThreadStart(childthreadcall) is created given “childthreat” polymorphism name. The system write “Child Thread Started” and create Thread(childthreat) given “child” polymorphism name. child call Start() and write “Main sleeping for 2 seconds…….”. After that the system goes to childthreadcall() and display “we are in”, The process of counting child thread to 10 begins by initialize i = 0. The main thread sleeps for 2000 ms, during which the child thread executes. The child thread runs till it is aborted by the main thread. It raises the ThreadAbortException and is terminated. Control returns to the main thread.

Part 2:

lab_6.aspx

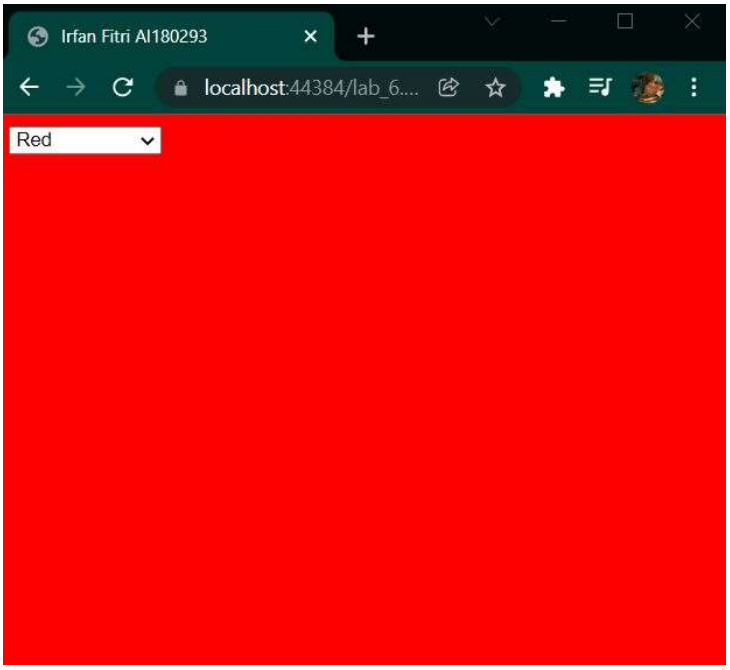
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="lab_6.aspx.cs"
Inherits="Lab_6.Lab_6" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

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

<head runat="server">
```

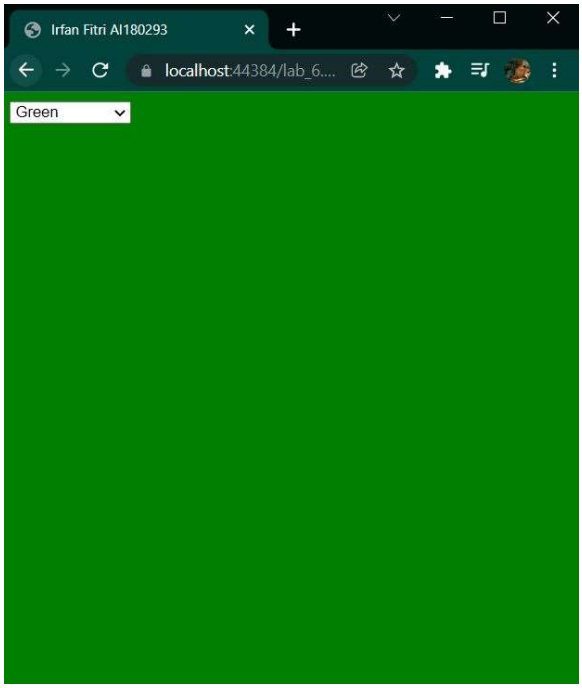
```
    <title>
        Irfan Fitri AI180293
    </title>
</head>

<body runat="server" id="BodyTag">
    <form id="form1" runat="server">
        <asp:DropDownList runat="server" id="ColorSelector" autopostback="true"
            onselectedindexchanged="ColorSelector_IndexChanged">
            <asp:ListItem value="White" selected="True">Select
color...</asp:ListItem>
            <asp:ListItem value="Red">Red</asp:ListItem>
            <asp:ListItem Value="Green">Green</asp:ListItem>
            <asp:ListItem Value="Blue">Blue</asp:ListItem>
        </asp:DropDownList>
    </form>
</body>
</html>
```

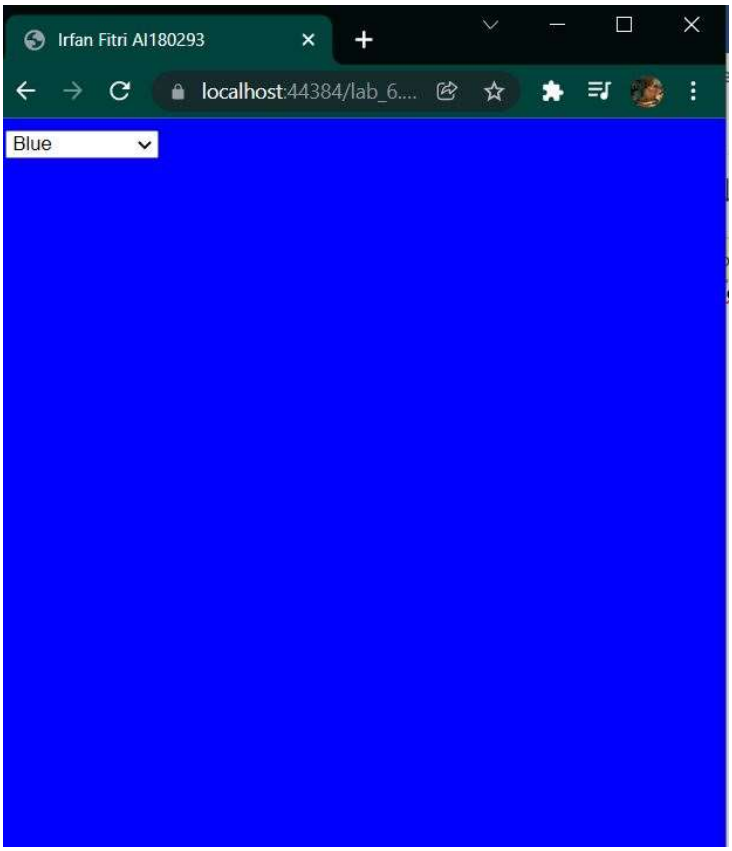
lab_6.aspx.cs

```
using System;
using System.Data;
using System.Web;

namespace Lab_6
{
    public partial class Lab_6 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if(Session["BackgroundColor"] != null)
            {
                ColorSelector.SelectedValue =
Session["BackgroundColor"].ToString();
                BodyTag.Style["background-color"] = ColorSelector.SelectedValue;
            }
        }

        protected void ColorSelector_IndexChanged(object sender, EventArgs e)
        {
            BodyTag.Style["background-color"] = ColorSelector.SelectedValue;
            Session["BackgroundColor"] = ColorSelector.SelectedValue;
        }

    }
}
```

Output


Irfan Fitri AI180293
localhost:44384/lab_6....
Red


Irfan Fitri AI180293
localhost:44384/lab_6....
Green


Irfan Fitri AI180293
localhost:44384/lab_6....
Blue

The code example use asp list item containing three colour name “Red”, “Green”, “Blue”.

On Page_Load function, conditional statement use Session[“BackgroundColor”] to determine the null value. If the BackgroundColor is selected between three colours, the Session will assign value of BackgroundColor.ToString() based on an given id name “ColorSelector”