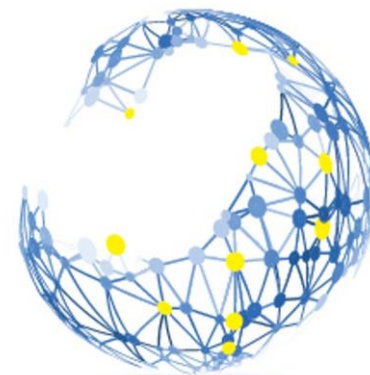


השתלמות מורים

תכנות בסביבת האינטרנט

גלעד מרקמן

קריית החינוך פארק המדע,
נס ציונה

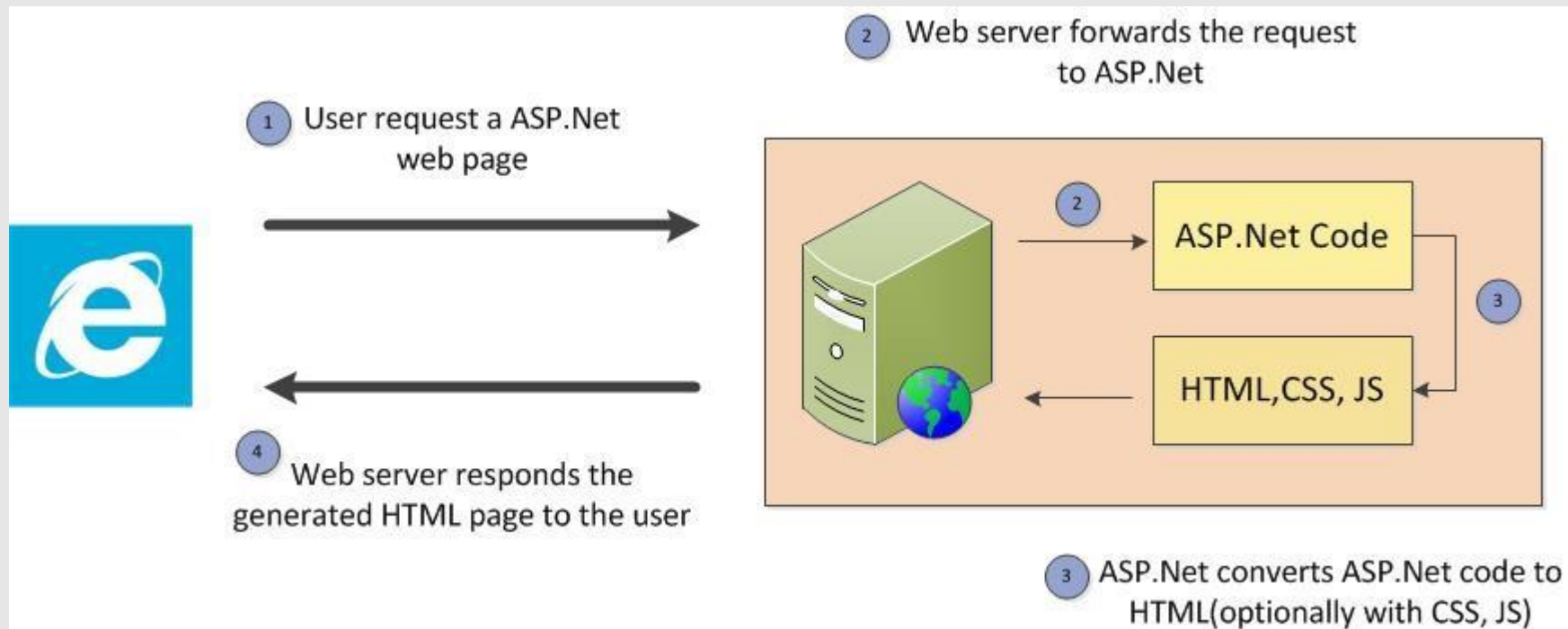


קריית החינוך
השש שנתית
פארק המדע
עתיד של חדשנות



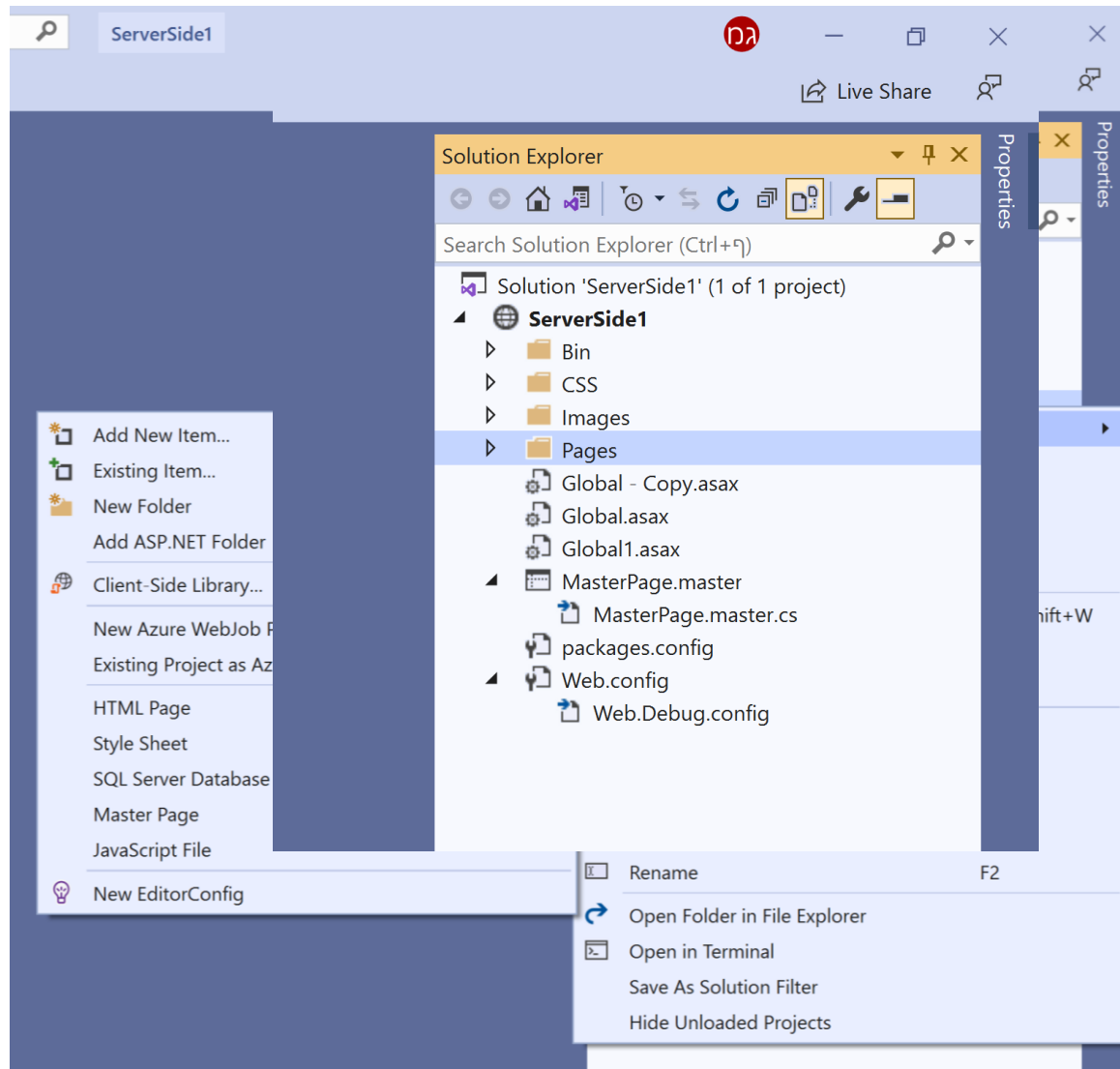
- יצירת דפי aspx מבוססי Master Page.
- האויבקטים Response & Request.
- כתיבת קוד בדף נפרד והאירוע Page_Load.
- תכנות חסר מצב והאובייקטים Session & Application.
- דף גלובלי ואירועים ברמת היישום.
- דוגמה – אבטחת דפים.

- במודל שרת לקוח הדפדפן של הלקוח שולח בקשה (request) לשרת לקבלת דף אינטרנט. השרת מקבל את הבקשה, יוצר את הדף, ומחזיר אותו ללקוח.



- במודל שרת לקוח אנו מחלקים את הקוד "הרץ" בתהליך זה לשני סוגים:
 - קוד צד לקוח.
 - קוד צד שרת.
- קוד צד לקוח – הם תוכניות שרצות על המחשב של הלקוח. הכוונה בדרך כלל לתסריטי גאווה סקריפט אשר נכללים בדף HTML ומופעלים על ידי הדפדפן של הלקוח.
- קוד צד שרת – הוא קוד שרץ על השרת "מאחורי הקלעים". קוד צד שרת בטכנולוגיית ASP.Net יכול להכתב בשפת C# או Visual Basic.
- קוד צד השרת מטפל בדרך כלל בבסיסי הנתונים באמצעות פניה אל שרת ה DB.

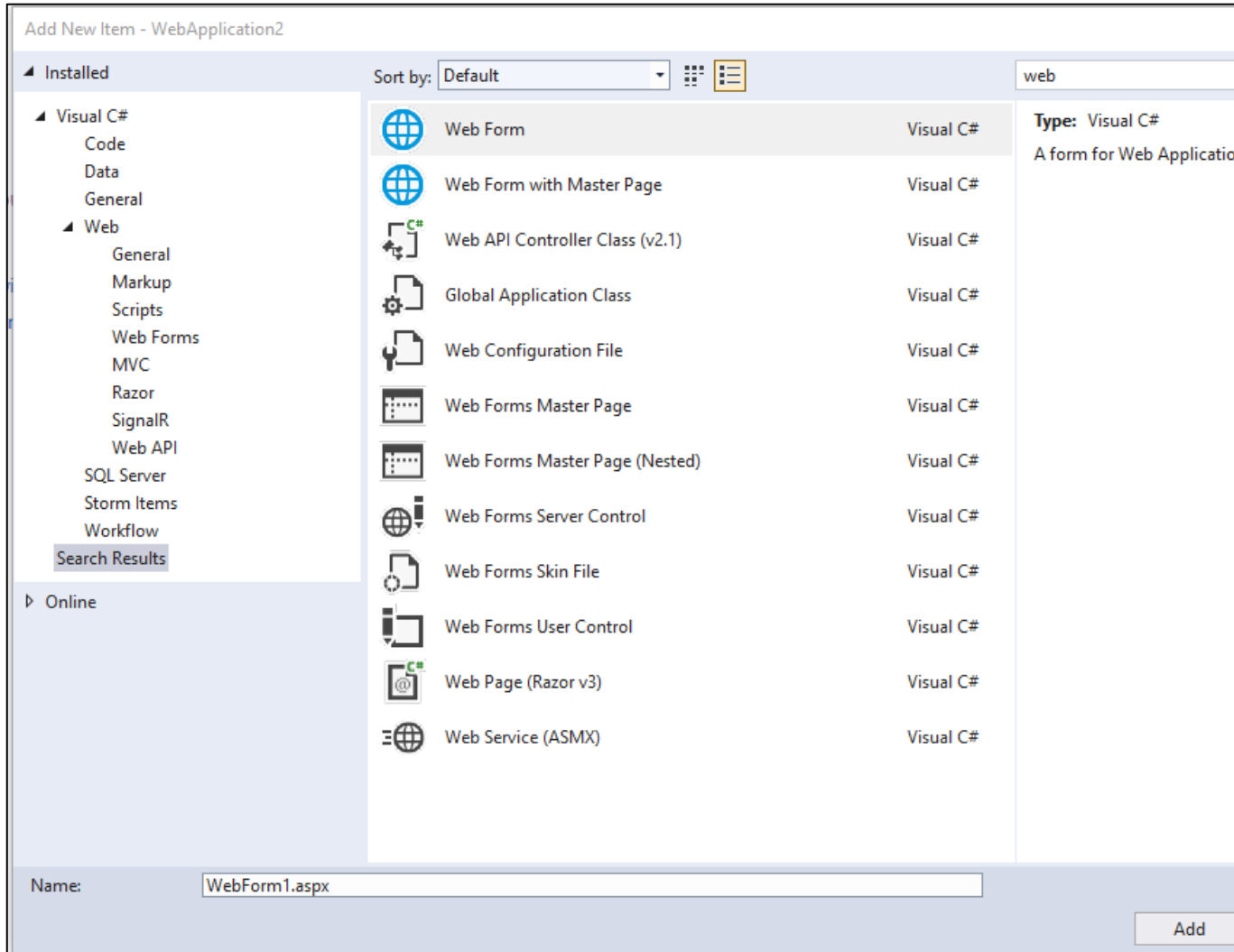
יצירת דפי aspx יצירת דף Master Page



- ליצירת דף aspx יש להצביע על הספרייה הרלוונטית ב solution Explorer וללחוץ על הכפתור הימני בעכבר.

- בתפריט הבחירה יש לבחור: Add

- ולאחר מכן: Add New Item



• בבחירת סוג הפריט יש לבחור:

• Web Form Visual C#

• אם הפריט לא נמצא ניתן לחפש בתיבת החיפוש.

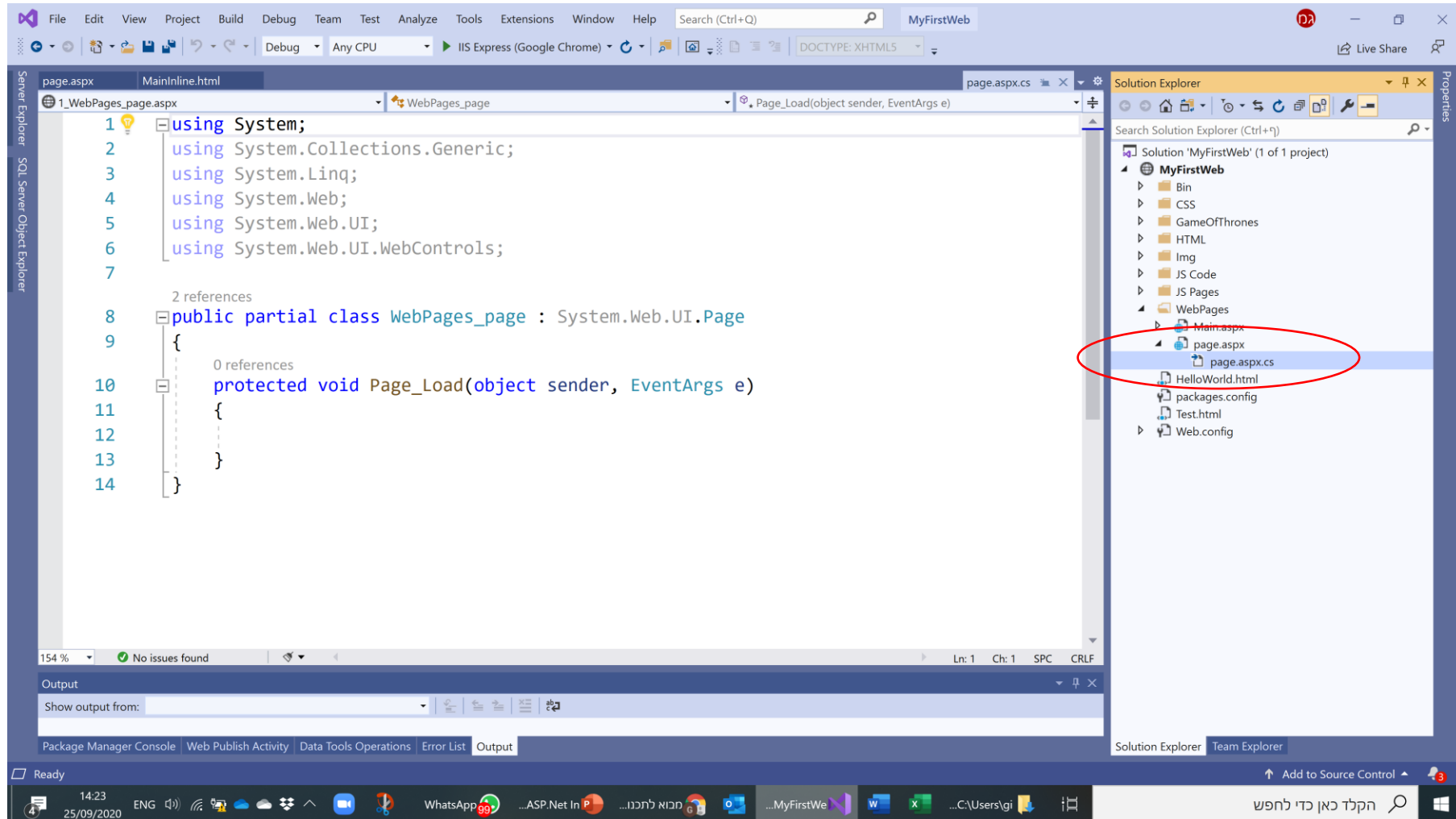
```
1 <%@ Page Language="C#" AutoEventWireup="true" CodeFile="page.aspx.cs" Inherits="Main" %>
2
3 <!DOCTYPE html>
4
5 <html xmlns="http://www.w3.org/1999/xhtml">
6 <head runat="server">
7 <title></title>
8 </head>
9 <body>
10 <form id="form1" runat="server">
11 <div>
12 </div>
13 </form>
14 </body>
15 </html>
16
```

• דף aspx בנוי למעשה משני דפים:

- א. דף HTML
- ב. דף cs לקוד בצד השרת.

- חשוב: כל התקשורת עם צד השרת מתבצע רק בתוך תגית `<form runat="server">`

דף הקוד בצד השרת



The screenshot shows the Visual Studio IDE with the following components:

- Code Editor:** Displays the code for `page.aspx.cs`. The code includes several `using` statements and a partial class definition:


```

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
8 public partial class WebPages_page : System.Web.UI.Page
9 {
10     protected void Page_Load(object sender, EventArgs e)
11     {
12     }
13 }
14

```
- Solution Explorer:** Shows the project structure for `MyFirstWeb`. The `page.aspx.cs` file is highlighted with a red circle. Other files visible include `Main.aspx`, `page.aspx`, `HelloWorld.html`, `packages.config`, `Test.html`, and `Web.config`.
- Output Window:** Shows "No issues found" at 154% zoom.
- Taskbar:** Shows the system tray with the date 25/09/2020 and various application icons.

- Master Page הוא למעשה תבנית (Template) לדפי .aspx.
- מטרתו לאפשר בצורה נוחה יצירת דפים באתר במבנה אחיד, למשל:
 - יצירת כותרת אחידה בכל הדפים – לוגו.
 - יצירת חלק תחתון footer אחיד בכל הדפים.
- שימוש במסטר פייג' חוסך בכתיבת קוד, הן בצד המשתמש (HTML) והן בצד השרת.
- ניתן להגדיר מספר דפי מסטר פייג'.
- ניתן להגדיר דפי מסטר פייג' מכוננים (nested). דף מסטר פייג' בתוכו דף מסטר פייג' נוסף ובתוכו דף .aspx.

Add New Item - WebApplication2

Sort by: Default

master

Web Form with Master Page	Visual C#	Type: Visual C# A Master Page for Web Applic
Web Forms Master Page	Visual C#	
Web Forms Master Page (Nested)	Visual C#	

Name: Site1.Master

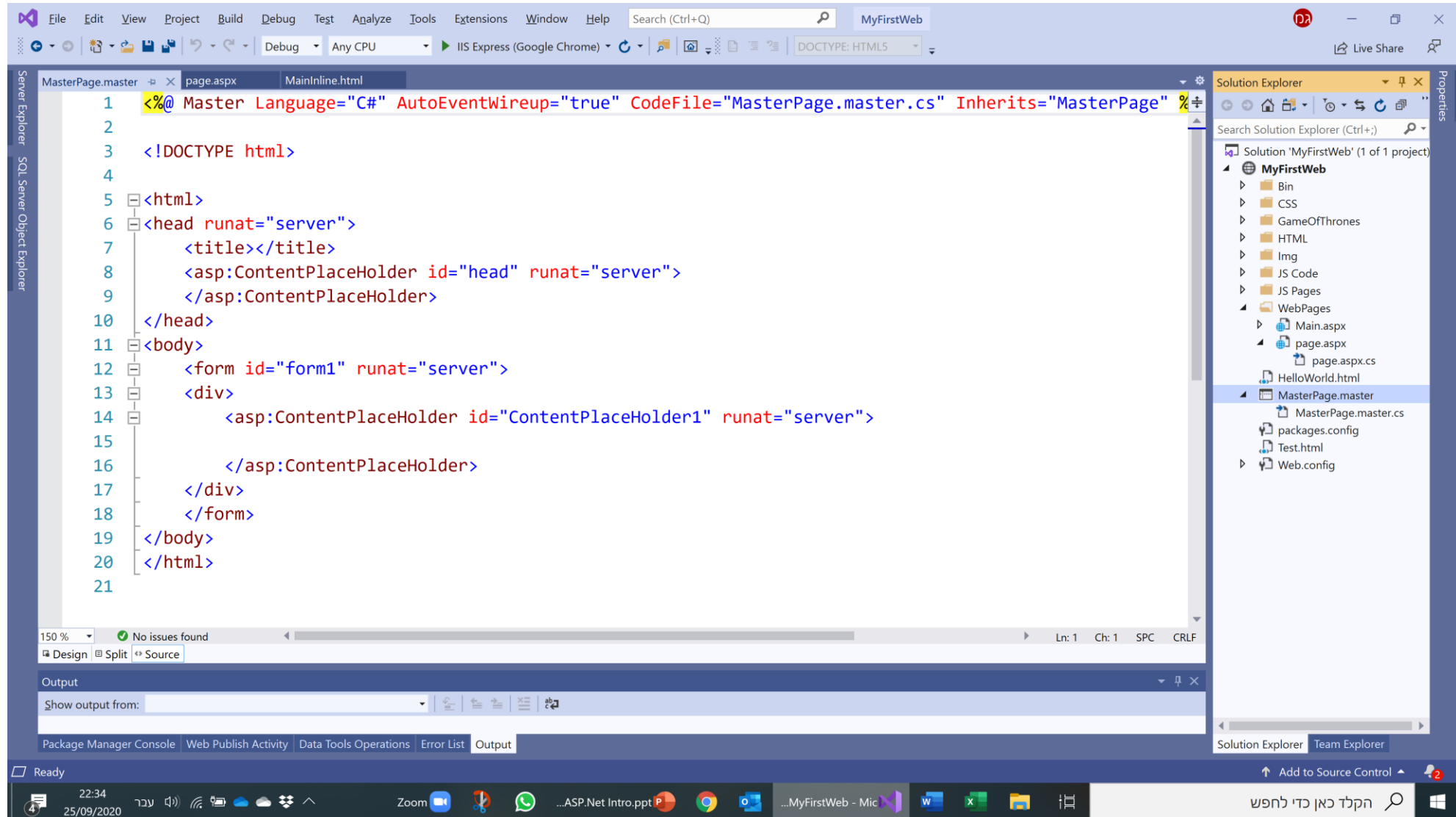
Add

- מצביעים על שם הפרויקט ולוחצים על הלחצן הימני בעכבר.

- בוחרים פריט חדש: Master Page Visual C#

- משנים את שם הדף – יש להשאיר את הסיומת .master

דף Master Page



The screenshot displays the Visual Studio IDE with the following components:

- Code Editor:** Shows the source code for `MasterPage.master`. The code is as follows:


```

1  <%@ Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs" Inherits="MasterPage" %>
2
3  <!DOCTYPE html>
4
5  <html>
6  <head runat="server">
7      <title></title>
8      <asp:ContentPlaceHolder id="head" runat="server">
9      </asp:ContentPlaceHolder>
10 </head>
11 <body>
12 <form id="form1" runat="server">
13 <div>
14 <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
15
16 </asp:ContentPlaceHolder>
17 </div>
18 </form>
19 </body>
20 </html>
21

```
- Solution Explorer:** Shows the project structure for 'MyFirstWeb'. The 'WebPages' folder is expanded, showing 'Main.aspx', 'page.aspx', 'HelloWorld.html', and 'MasterPage.master'. The 'MasterPage.master' folder is also expanded, showing 'MasterPage.master.cs'.
- Output Window:** Shows 'No issues found'.
- Taskbar:** Shows the system tray with the time 22:34 on 25/09/2020 and various application icons.

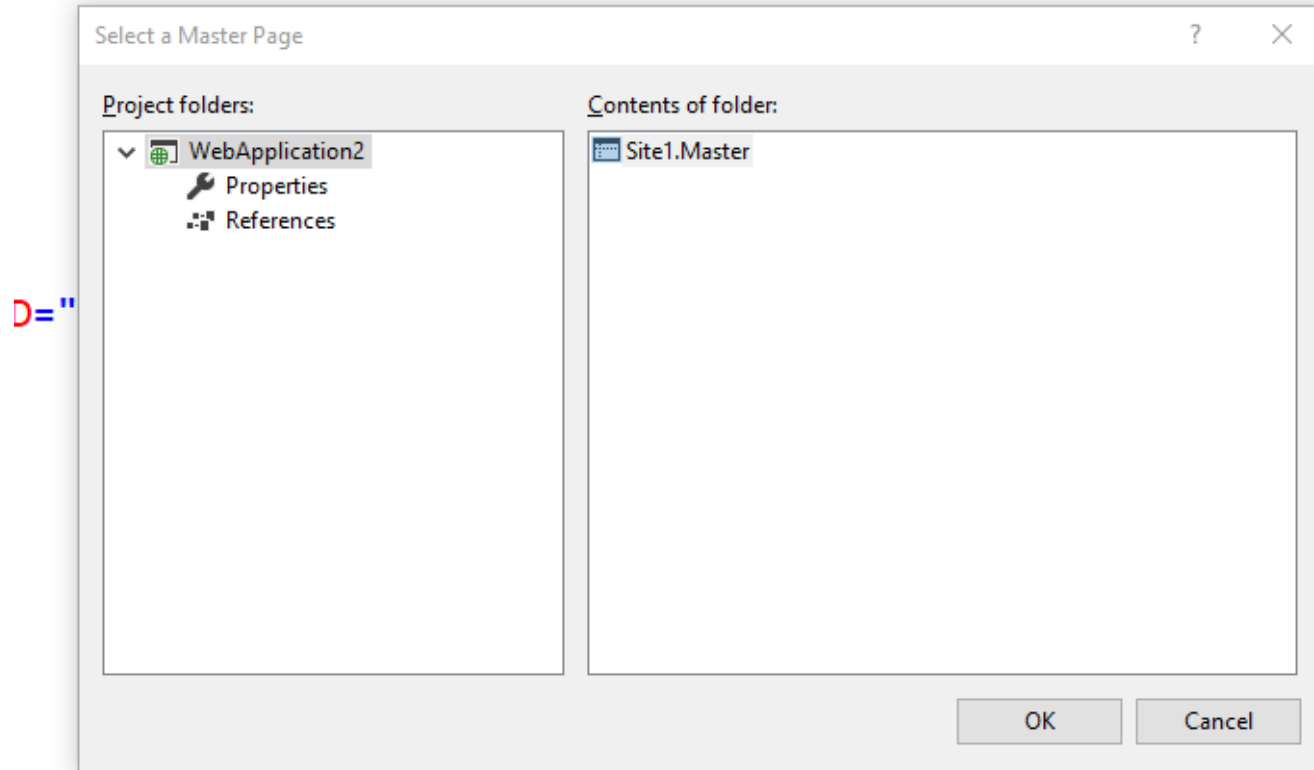
```
1 <%@ Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs" Inherits="MasterPage" ClientIDMode="Static" %>
2
3 <!DOCTYPE html>
4
5 <html>
6 <head runat="server">
7 <title></title>
8 <asp:ContentPlaceHolder id="head" runat="server">
9 </asp:ContentPlaceHolder>
10 </head>
11 <body>
12 <form id="form1" runat="server">
13 <div>
14 <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
15 </asp:ContentPlaceHolder>
16 </div>
17 </form>
18 </body>
19 </html>
20
21
```

• מומלץ להוסיף בכותרת את הפקודה:

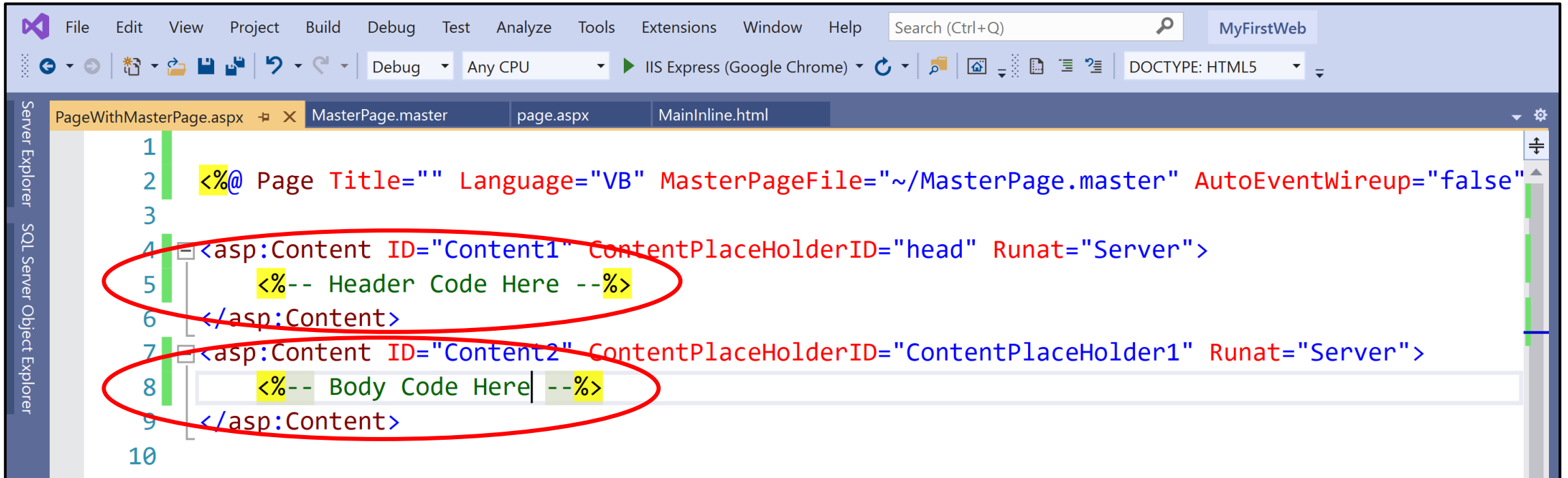
```
ClientIDMode="Static"
```

- אין לכתוב קוד בתוך ה-ContentPlaceHolder. מקום זה נשמר לקוד של הדף.aspx.
- שימו לב – ברירת המחדל היא שכל הקוד בדף יהיה בתוך טופס <form>.

```
runat="server">
```



- קליק ימני בתיקייה בה רוצים ליצור את הדף.
- יצירת דף המבוסס על תבנית Master Page.
- מתן שם לדף – סיומת .aspx
- בסיום .add
- בחירת התבנית הרלוונטית.
- OK



```
1
2 <%@ Page Title="" Language="VB" MasterPageFile="~/MasterPage.master" AutoEventWireup="false"
3
4 <asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
5     <!-- Header Code Here -->
6 </asp:Content>
7 <asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
8     <!-- Body Code Here -->
9 </asp:Content>
10
```

- בדף המבוסס על Master Page יש לכתוב את קוד HTML אך ורק בתוך תגית התוכן <Content>.

- הקוד הנכתב בתוך תגית התוכן משתלב בקוד של הדף הראשי.

דף המבוסס על Master Page

```
MasterPage.master  page.aspx  MainInline.html
1  <%@ Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs" Inherits="MasterPage" %>
2
3  <!DOCTYPE html>
4
5  <html>
6  <head runat="server">
7      <title></title>
8      <asp:ContentPlaceHolder id="head" runat="server">
9      </asp:ContentPlaceHolder>
10 </head>
11 <body>
12     <form id="form1" runat="server">
13     <div>
14         <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
15
16         </asp:ContentPlaceHolder>
17     </div>
18 </form>
19 </body>
20 </html>
```

```
1
2  <%@ Page Title="" Language="VB" MasterPageFile="~/MasterPage.master" AutoEventWireup="false"
3
4  <asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
5      <!-- Header Code Here -->
6  </asp:Content>
7  <asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
8      <!-- Body Code Here -->
9  </asp:Content>
10
```

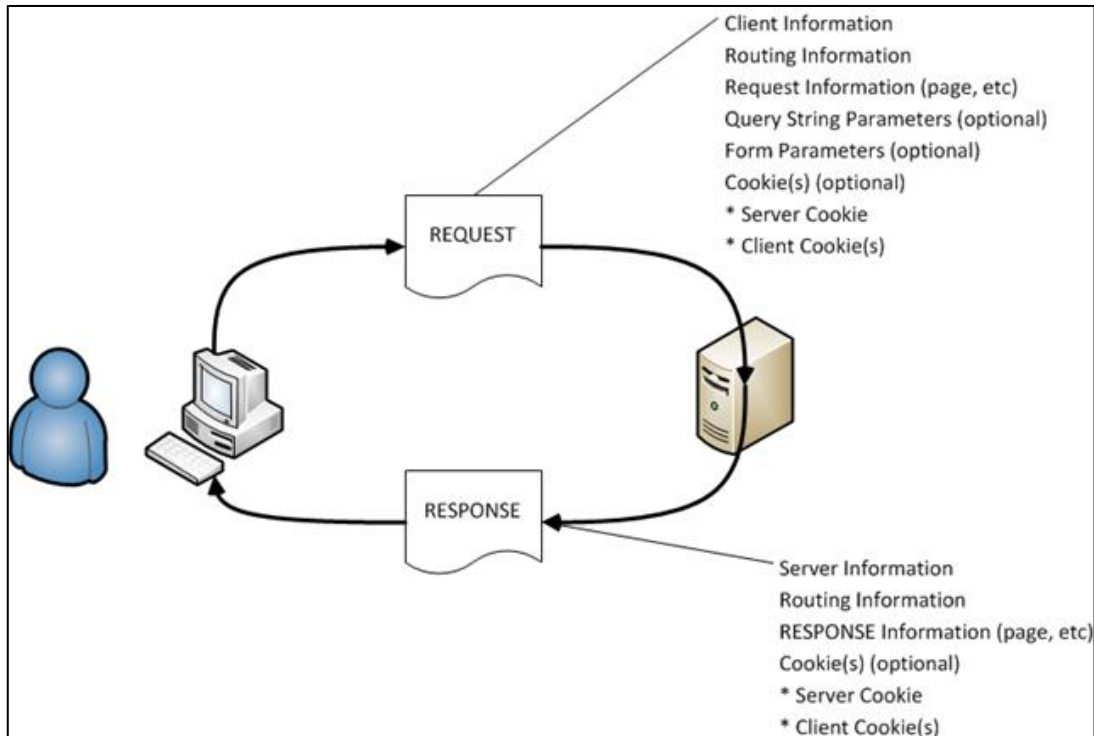

- לא ניתן ליצור שני טפסים Form הרצים בצד השרת. כמו כן, אין לבצע כינון של טפסים.
- על כן, מומלץ להשאיר את הטופס ב Master Page. אז הטופס כולל גם את התגיות במסטר פייג' וגם את התגיות בתוכן, ואין להוסיף תגית Form בדפים עצמם.
- כיוון שהטופס יהיה במסטר פייג' האירוע להפעלת בדיקת הטפסים חייב להיות רק בדף של התוכן. כך, ניתן להפעיל אירועים שונים בכל טופס. על כן, מומלץ להפעיל את בדיקת האימות של הטופס בעת לחיצה על לחצן submit ולא באמצעות האירוע onsubmit.

תכנות צד שרת

אובייקטים

Response & Request

- ניתן לכתוב קוד צד שרת בשני אופנים:
- א. כתיבת הקוד בתוך הדף לצד תגיות HTML באמצעות התגית `<% %>`. בתוך התגית ניתן לכתוב קוד `.c#`.
- ב. כתיבת קוד בדף נפרד – Code Behind.
- הדרך השניה היא המומלצת והנוחה יותר.



• **Request** – אובייקט הכולל את הנתונים בטופס שנשלחים בעת בקשה מהדפדפן של המשתמש לשרת האינטרנט.

• **Response** – אובייקט הכולל את התשובה של השרת שנשלחת למשתמש. אנו נשתמש במתודות של האובייקט לכתיבה לדף או מעבר לדף אחר.

כתיבת קוד בצד השרת – בדף HTML Response.Write

- אחת הדרכים להדפסי לדף המוחזר למשתמש תעשה באמצעות Response.Write

```
<div style="text-align:center">  
  <br />  
  <%Response.Write("Hello Gilad <br />"); %>  
  <%= "Wellcome back <br />" %>  
</div>
```

- דרך מקוצרת לכתוב Response.Write נעשית באמצעות '='. אין להוסיף ; בסוף.
- הכתיבה לדף נעשית במיקום בו מופיעה התגית <% ...%>.

• Request.Form – כולל את השדות והנתונים ב Form לאחר שבוצע submit.

```
<form id="form1" runat="server">
  <div>
    <input type="text" name="userName" id="userName" />
    <input type="password" name="password" id="password" /> <br />
    <input type="submit" name="submit" id="submit" />
  </div>
  <div style="text-align:center; width: 500px; overflow-wrap: break-word;">
    <br />
    <%= "Request Form: <br/>" %>
    <%= Request.Form %>
  </div>
</form>
```

Lesson 1 - Server Side

<input type="text" value="gilad"/>	<input type="password"/>
<input type="submit" value="Submit"/>	

Request Form:

__VIEWSTATE=Y%2bbma5nclxUhA4BUyuVXPcGTRPOa5wd1LwCO7V
ENp5Lpt%2fpbnizO7BLtJ6%2fh%2fPsx2zo7p5oJjC5KizMdyVQi3Ggp%2
fsBSBWIPMKXo8ZkLbg%3d&__VIEWSTATEGENERATOR=CBF602F1
&userName=gilad&password=123234&submit=Submit

קבלת תוכן של שדה בטופס

- ניתן לקבל את הערך של שדה ספציפי בטופס (Form) באמצעות שימוש בשם השדה (name).

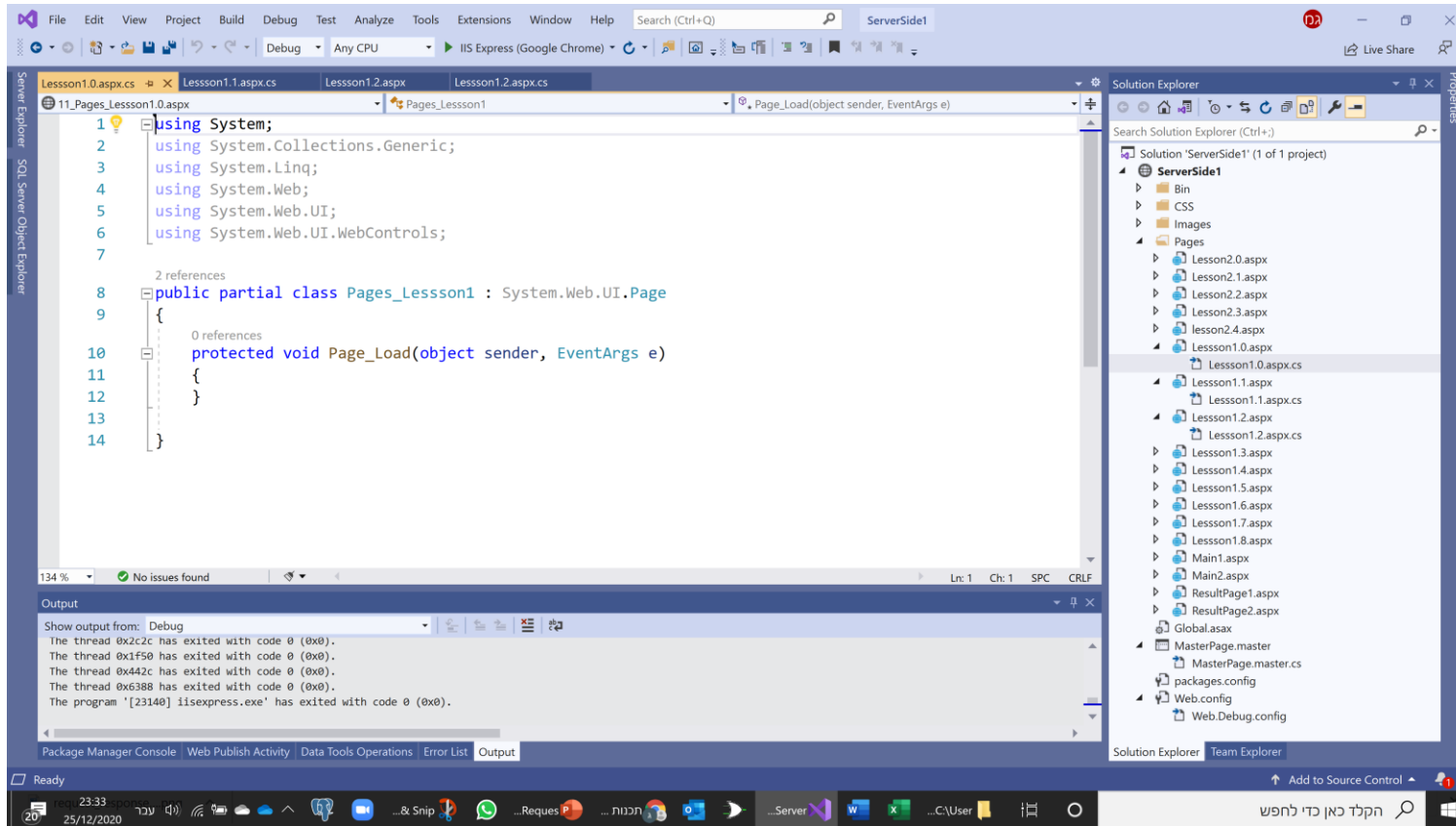
```
<form id="form1" runat="server">  
  <div>  
    <input type="text" name="userName" id="userName" />  
    <input type="password" name="password" id="password" /> <br />  
    <input type="submit" name="submit" id="submit" />  
  </div>  
  <div style="text-align:center">  
    <%= "<div> <br/>Wellcome: " + Request.Form["userName"] + "</div>" %>  
  </div>  
</form>
```

Lesson 1 - Server Side

Wellcome: gilad

כתיבת Code Behind

אירוע Page_Load



```

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
8 public partial class Pages_Lesson1 : System.Web.UI.Page
9 {
10     protected void Page_Load(object sender, EventArgs e)
11     {
12     }
13 }
14

```

The screenshot shows the Visual Studio IDE with the code behind of an ASP.NET page. The code includes using statements for System, System.Collections.Generic, System.Linq, System.Web, System.Web.UI, and System.Web.UI.WebControls. It defines a partial class Pages_Lesson1 that inherits from System.Web.UI.Page and contains a Page_Load method. The Solution Explorer on the right shows the project structure with various ASP.NET files and folders.

- כתיבת קוד צד שרת יכולה להעשות בקובץ נפרד.
- ב ASP.Net לכל דף אסף קיים דף נוסף שבו ניתן לכתוב את קוד צד השרת.
- הסיומת .cs מלמדת כי הקוד בצד השרת ייכתב ב CSharp.

0 references

```
protected void Page_Load(object sender, EventArgs e)
{
    ...
}
```

1 reference

```
public string insertMessage ()
{
    if (Request.Form["userName"]== "Gilad" && Request.Form["password"] == "1968")
    {
        return "Welcome " + Request.Form["userName"];
    }
    else
    {
        return "Welcome Visitor";
    }
}
```

```
<div style="text-align:center">
    <br />
    <%=insertMessage() %>
</div>
```

זימון הפעולה

Gilad

Submit

Welcome Gilad

פניה לדף aspx באמצעות קוד בצד השרת

- ניתן לפנות לתגיות בדף aspx באמצעות קוד בצד השרת.
- הפניה נעשות באמצעות הגדרת התגית כתגית "הרצה" בצד השרת.
- יש להוסיף למאפייני התגית בדף aspx את המאפיין: `runat="server"`
- לדוגמה: `<div style="text-align:center" name="div" id="div" runat="server">`
- בצד השרת הוגדר משתנה (אובייקט) הנקרא div (לפי ה id) ולו מאפיינים ששינוי שלהם ישנה את הדף (בדומה לאובייקטים ב HTML).
- למשל `div.InnerHtml = "Unknown UserName Or Wrong Password";` יוסיף לתגית div את הטקסט.
- הוספת `runat="server"` יכולה להעשות לכל תגית HTML. במקרה כזה לא ניתן עוד להשתמש ב `Request.Form`.

פניה לתגית aspx באמצעות runat="server"

```
<div style="text-align:center" name="div" id="div" runat="server">  
  
</div>
```

```
0 references  
protected void Page_Load(object sender, EventArgs e)  
{  
    if (Request.Form["userName"] == "Gilad" && Request.Form["password"] == "1968")  
    {  
        div.InnerHtml = "Welcome " + Request.Form["userName"];  
    }  
    else  
    {  
        div.InnerHtml = "Unknown UserName Or Wrong Password";  
    }  
}
```

- כדי להפעיל פעולה שנכתבה בצד השרת עלינו לקרוא ("לזמן") את הפעולה.
- א. ניתן לזמן פעולה באמצעות התגית `<% %>` בדף HTML :

```
<%=insertMessage() %>
```

- ב. ניתן להגדיר ב code behind כי הפעולה תופעל כאשר יתרחש אירוע בדף (בדומה לאירועים של Javascript).
- אחד האירועים החשובים הוא לאחר טעינת הדף Page_Load. אם מגדירים את הפעולה בצד השרת בשם page_load היא תזומן בכל טעינת דף.

```
protected void Page_Load(object sender, EventArgs e)
{
    // code to execute when page load
}
```

- הפעולה הבאה תזומן בכל טעינה של הדף.

```
protected void Page_Load(object sender, EventArgs e)
{
    if (Request.Form["userName"] == "Gilad" && Request.Form["password"] == "1968")
    {
        div.InnerHtml = "Welcome " + Request.Form["userName"];
    }
    else
    {
        div.InnerHtml = "Unknown UserName Or Wrong Password";
    }
}
```

- לעיתים אנו רוצים לדעת האם טעינת הדף התרחשה בעת הטעינה הראשונה של הדף או לאחר שהמשתמש ביצע submit לטופס בדף.
- המאפיין Page.IsPostBack או בקיצור IsPostBack יחזיר false בעת הטעינה הראשונה ו true אם הדף נטען לאחר שנשלח request לשרת באמצעות שליחת הטופס.

• דוגמה:

```
if (IsPostBack)
{
    if (Request.Form["userName"] == "Gilad" && Request.Form["password"] == "1968")
    {
        div.InnerHtml = "Welcome " + Request.Form["userName"];
    }
    else
    {
        div.InnerHtml = "Unknown UserName Or Wrong Password";
    }
}
```

- במקום לעשות שימוש במאפיין IsPostBack ניתן להשתמש באובייקט request ולברר אם המשתמש לחץ על כפתור .submit.
- אם המשתמש לא לחץ על כפתור submit (כלומר טעינה בפעם הראשונה) אז `Request.Form["submit"] = null`. על כן, דרך נוספת לכתוב את הפעולה הקודמת:

```
if (Request.Form["submit"] != null)
{
    if (Request.Form["userName"] == "Gilad" && Request.Form["password"] == "1968")
    {
        div.InnerHtml = "Welcome " + Request.Form["userName"];
    }
    else
    {
        div.InnerHtml = "Unknown UserName Or Wrong Password";
    }
}
```


- הגדרת תגית כרצה בשרת (runat="server") מאפשרת לנו לפנות לכל המאפיינית של התגית, לרבות מאפייני עיצוב CSS.
- דוגמה לשינוי צבע גופן ורקע, ולביטול הגדרת צבע רקע (Remove):

```
if (Request.Form["userName"] == "Gilad" && Request.Form["password"] == "1968")  
{  
    div.InnerHtml = "Welcome " + Request.Form["userName"];  
    h1.Style.Add("color", "green");  
    h1.Style.Add("background-color", "lightBlue");  
}  
else  
{  
    div.InnerHtml = "Unknown UserName Or Wrong Password";  
    h1.Style.Remove("background-color");  
    h1.Style.Add("color", "red");  
}
```

- נלמד להסתיר תגיות של דף aspx באמצעות קוד צד שרת בשתי דרכים:

```
Registration.Visible = false;  
Registration.Visible = true;
```

- באמצעות המאפיין Visible של תגית:

- לא לשכוח להגדיר id לתגית וכן `.runat="server"`.

```
Registration.Style.Add("display", "block");  
Registration.Style.Add("display", "none");
```

- באמצעות עיצוב CSS של המאפיין `display`:

- באפשרות הראשונה המקום של התגית נשמר אך התגית נסתרת למשתמש ואילו באפשרות השניה המקום של התגית כלל לא נשמר בדף.

תכנות חסר מצב

stateless Programming

והאובייקטים

Session & Application

• נושאי הפרק:

- תכנות "חסר מצב" – Stateless Programming
- אובייקט Session, אובייקט Application.
- Global.asax – ניהול אירועים ברמת היישום.

- השרת אינו שומר נתונים על ההתקשרות עם הלקוח.
- לאחר שהלקוח שלח Request וקיבל Response הקשר עם המשתמש (הדפדפן) "מנותק" והשרת אינו שומר נתונים על הדף שנשלח.
- שיטה זו מאפשר לשרת לטפל במקביל במספר רב של משתמשים.

- כתוב קוד בצד השרת שבכל לחיצה המונה בדף עולה באחד.
- לפתוח דף aspx חדש. להגדיר תגיות HTML בהתאם לתמונה למטה.
- לכתוב את הקוד ב `page_Load` כך שבכל לחיצה על הכפתור המונה גדל באחד.

Lesson 2

Counter 0

Add

- לחיצה על הכפתור מעלה מ-0 ל-1, אבל לא ל-2.

Lesson 2

Counter 0

Add

```
<body>
  <h1 style="text-align:center">Lesson 2</h1>
  <form id="form1" runat="server">
    <div>
      Counter
      <input type="text" name="counter" id="counter" runat="server"/>
      <br />
      <input type="submit" name="Add" id="Add" value="Add" />
    </div>
  </form>
</body>
```

2 references

```
public partial class Pages_Lesson2_0 : System.Web.UI.Page
{
    0 references
    protected void Page_Load(object sender, EventArgs e)
    {
        int val=0;
        if (!IsPostBack)
        {
            val = 0;
            counter.Value = val.ToString();
        }
        else
        {
            val++;
            counter.Value = val.ToString();
        }
    }
}
```

Lesson 2

Counter 5

Add

0 references

```
protected void btnClick(object sender, EventArgs e)
{
    if (counter.Value == "")
    {
        counter.Value = "1";
    }
    else
    {
        int val = int.Parse(counter.Value);
        val++;
        counter.Value = val.ToString();
    }
}
```

```
<body>
  <h1 style="text-align:center">Lesson 2</h1>
  <form id="form1" runat="server">
    <div>
      Counter
      <input type="text" name="counter" id="counter" runat="server"/>
      <br />
      <input type="button" name="Add" id="Add" value="Add" runat="server" onclick="btnClick"/>
    </div>
  </form>
</body>
```


Lesson 2

Counter 5

Add

```
<body>
  <h1 style="text-align:center">Lesson 2</h1>
  <form id="form1" runat="server">
    <div>
      Counter
      <input type="text" name="counter" id="counter" runat="server"/>
      <br />
      <input type="button" name="Add" id="Add" value="Add" runat="server" onserverclick="btnClick"/>
    </div>
  </form>
</body>
```

```
0 references
protected void btnClick(object sender, EventArgs e)
{
    if (counter.Value == "")
    {
        counter.Value = "1";
    }
    else
    {
        int val = int.Parse(counter.Value);
        val++;
        counter.Value = val.ToString();
    }
}
```

```
2 references
public partial class Pages_Lesson2_0 : System.Web.UI.Page
{
    0 references
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["counter"]==null)
            Session["counter"] = 0;
        counter.Value = Session["counter"].ToString();
    }

    0 references
    protected void btnClick(object sender, EventArgs e)
    {
        Session["counter"] = (int)Session["counter"] + 1;
        counter.Value = Session["counter"].ToString();
    }
}
```

- אובייקט Session מאפשר לשמור נתונים המשותפים ל Session ("שיחה") עם משתמש.
- המשתנה בתוך Session הוא אובייקט, ולכן יש לבצע המרה Casting כדי לעשות בו שימוש.
- האובייקט נשמר גם אם הדף נסגר.
- האובייקט משותף לכל הדפים באותו .session

- באובייקט Session ניתן להגדיר מספר משתנים (מאפיינים).
- הפניה למשתנה נעשית על ידי המילה Session ובסוגריים שם המשתנה:
Session ["name"]
- אין צורך להגדיר משתנה Session. ברגע שמציבים ערך בתוך משתנה, המשתנה מוגדר אוטומטית.
- האובייקט נמחק בסיום ה-Session. ה-Session מסתיים באחד משניים:
 - העדר פעילות למשך זמן שנקבע במאפיין Session.Timeout. ברירת המחדל 20 דק'.
 - יציאה באמצעות קוד Session.Abandon().
- ה-Session אינו מסתיים כאשר המשתמש סוגר את הדפדפן.

- האובייקט Application משותף לכל הפרויקט שלנו ולכל ה Sessions.
- הפניה למשתנים בתוך ה- Application נעשית באמצעות שם המשתנה: Application["Name"].
- המשתנה הוא אובייקט ושימוש בו מחייב המרה Casting.
- האובייקט נוצר בהפעלת האתר ונמחק כאשר האתר מפסיק "לרוץ".

- כתוב מונה שכולל שני שדות:
- האחד מתעדכן באמצעות Application
- אילו השני באמצעות Session.
- בכל לחיצה האובייקטים גדלים ב- 1.
- מה ההבדלים ביניהם ?

Session Counter

Application Counter

```
2 references
public partial class Pages_Lesson2_0 : System.Web.UI.Page
{
    0 references
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["counter"]==null)
            Session["counter"] = 0;
        if (Application["counter"]==null)
            Application["counter"] = 0;

        ApplicationCounter.Value = Application["counter"].ToString();
        SessionCounter.Value = Session["counter"].ToString();
    }
    0 references
    protected void btnClick(object sender, EventArgs e)
    {
        Application["counter"] = (int)Application["counter"] + 1;
        ApplicationCounter.Value = Application["counter"].ToString();
        Session["counter"] = (int)Session["counter"] + 1;
        SessionCounter.Value = Session["counter"].ToString();
    }
}
```

Session Counter

4

Application Counter

26

Add

דף גלובלי Global.asax

אירועים ברמת היישום

```
Global.aspx  X
1  <%@ Application Language="C#" %>
2
3  <script runat="server">
4
5  void Application_Start(object sender, EventArgs e)
6  {
7      // Code that runs on application startup
8  }
9
10
11 void Application_End(object sender, EventArgs e)
12 {
13     // Code that runs on application shutdown
14 }
15
16
17 void Application_Error(object sender, EventArgs e)
18 {
19     // Code that runs when an unhandled error occurs
20 }
21
22
23 void Session_Start(object sender, EventArgs e)
24 {
25     // Code that runs when a new session is started
26     Session["Login"] = false;
27 }
28
29 void Session_End(object sender, EventArgs e)
30 {
31     // Code that runs when a session ends.
32     // Note: The Session_End event is raised only when the sessionstate mode
33     // is set to InProc in the Web.config file. If session mode is set to StateServer
34     // or SQLServer, the event is not raised.
35 }
36
37
```

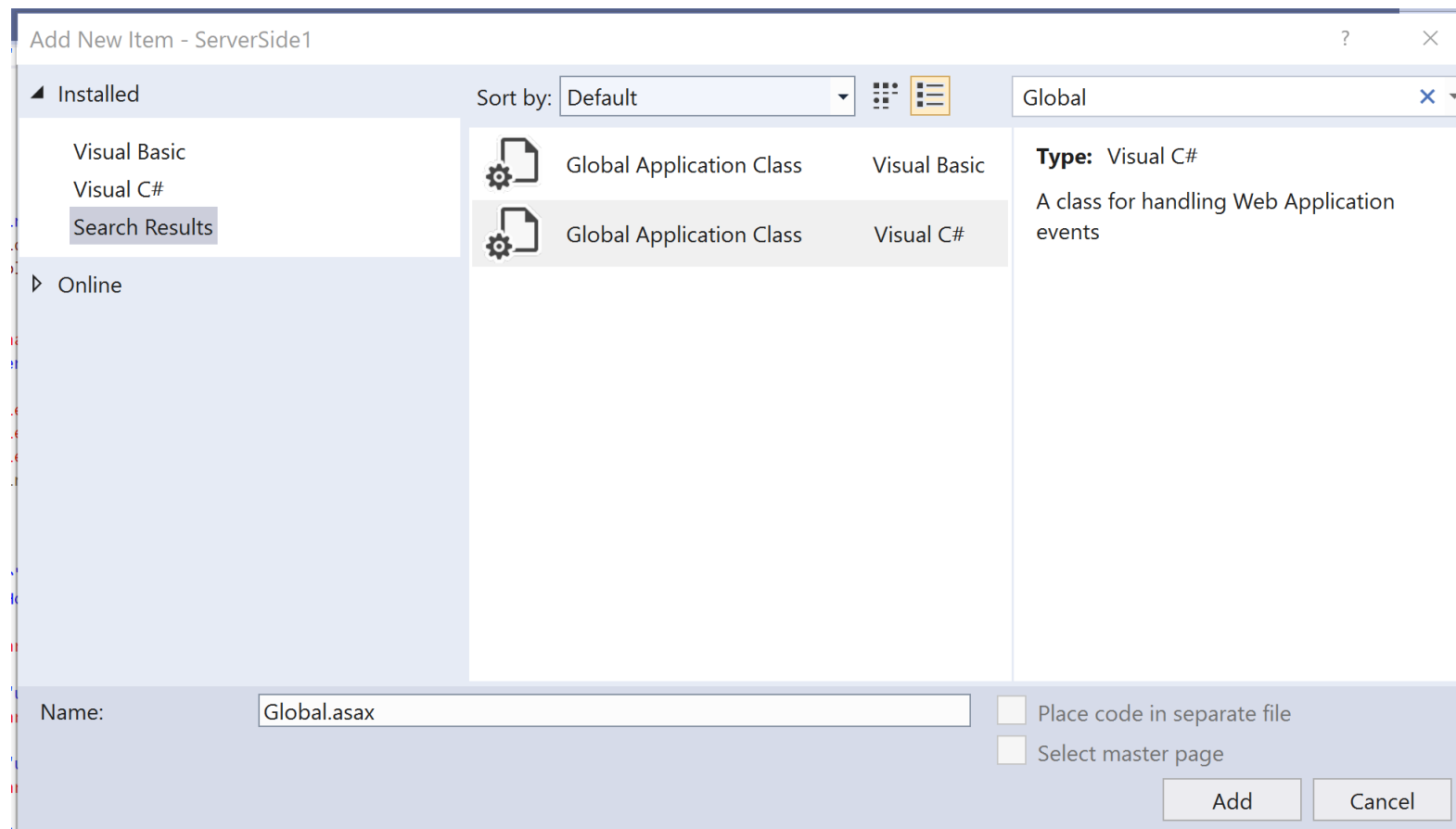
• דף הממוקם בספריית השורש ובו קיימים "אירועים" ברמת היישום:

- Application_Start
- Application_End
- Application_Error
- Session_Start
- Session_End

• הדף מאפשר לבצע פעולות, כגון איתחול משתנים, בעת קרות אחד האירועים הללו.

יצירת דף Global.aspx

- ניתן ליצור רק דף גלובלי אחד בפרויקט, באמצעות Add New Item.

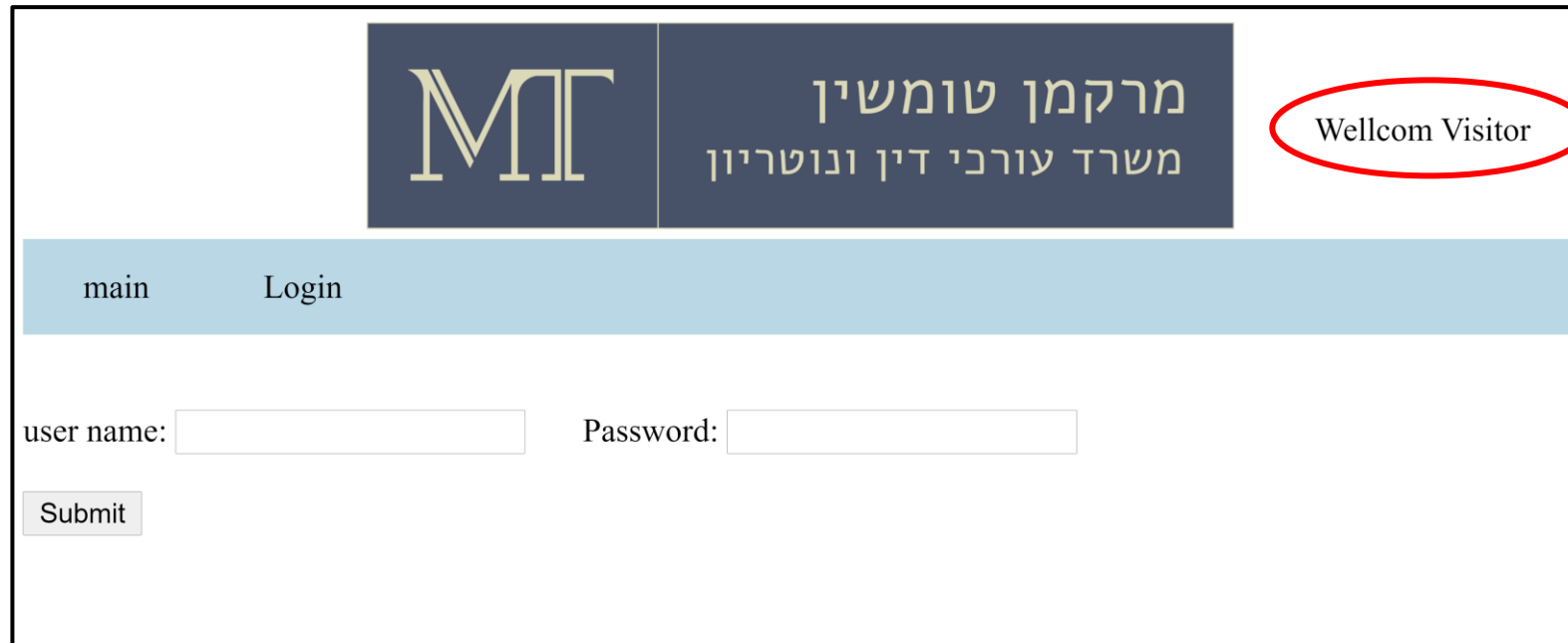


דוגמה לתכנות צד שרת

אבטחת דפים

- בפרק זה ניישם את מה שלמדנו, ונתרגל:
- כתיבת קוד צד שרת בדפים מבוססי Master Page
- שימוש בפעולה `Response.Redirect()`
- העברת נתונים בין דפים באותו יישום – שימוש באובייקט `Session`
- אבטחת דפים באמצעות קוד צד שרת.

- הוספת שם המשתמש שהתחבר בראש הדף בכל האתר.
- אם המשתמש לא עשה Login ייכתב: Welcome Visitor.



MIT מרקמן טומשין
משרד עורכי דין ונוטריון

Wellcom Visitor

main Login

user name: Password:

Submit

- כיוון שאנחנו מעוניינים ששם המשתמש שהתחבר יופיע בכל הדפים באתר בו גולש המשתמש נגדיר שני משתנים ב Session:
 - האחד משתנה מסוג String ובו נשמור את שם המשתמש
 - השני משתנה מסוג Boolean שיהיה True אם המשתמש ביצע Login.
- נאתחל את המשתנים בדף Global בתחילת Session כך ששם משתמש יהיה "Visitor" ו-במשתנה ההתחברות יהיה False.

```
void Session_Start(object sender, EventArgs e)
{
    // Code that runs when a new session is started
    Session["userName"] = "Visitor";
    Session["Login"] = false;
}
```

- אנו מעוניינים ששם המשתמש יופיע בכל הדפים ולכן נציג את שם המשתמש בתוך ה Master Page.
- שם המשתמש שמור לנו במשתנה [Session["userName"]], ולכן נדפיס את תוכן המשתנה במקום המתאים ב Master Page, באמצעות הפקודה: `<%= "Wellcom " + Session["userName"] %>`

```
<table align="center" class="tblHeader">
  <tr>
    <td style="width:20%"></td>
    <td style="width:60%"></td>
    <td style="width:20%"><%= "Wellcom " + Session["userName"] %></td>
  </tr>
</table>
```



מרקמן טומשין
משרד עורכי דין ונוטריון

Wellcom Gilad

[main](#)

[Gallery](#)

[About](#)

[Login](#)

[XPages](#)

user name:

Password:

Submit

```
protected void Page_Load(object sender, EventArgs e)
{
    if (IsPostBack)
    {
        if (Request.Form["userName"]=="Gilad" && Request.Form["password"] == "1968")
        {
            Session["userName"] = Request.Form["userName"];
            Session["Login"] = true;
            message.InnerHtml = "";
            Response.Redirect("/Pages/Main2.aspx");
        }
        else
        {
            Session["userName"] = "Visitor";
            Session["Login"] = false;
            message.InnerHtml = "Unknown user name or password";
        }
    }
}
```

- כתיבת הקוד בדף Login.
- הקוד מופעל רק אם המשתמש מילא פרטים, כלומר רק ב Post Back.
- אם שם משתמש וסיסמה נכונים משנים את משתני ה Session בהתאם.
- אם בוצעה כניסה המשתמש מפונה לדף הראשי.

מרקמן טומשין

משרד עורכי דין ונוטריון

Wellcom Visitor

main Login

user name: Password:

Unknown user name or password

מרקמן טומשין

משרד עורכי דין ונוטריון

Wellcom Gilad

main Gallery About Login XPages

user name: Password:

```
<li runat="server" id="bar1">  
  <a href="Gallery.aspx">Gallery</a>  
</li>
```

```
0 references  
protected void Page_Load(object sender, EventArgs e)  
{  
    if (!(bool)Session["Login"])  
    {  
        bar1.Style.Add("display", "none");  
        bar2.Style.Add("display", "none");  
        bar3.Style.Add("display", "none");  
    }  
    else  
    {  
        bar1.Style.Add("display", "block");  
        bar2.Style.Add("display", "block");  
        bar3.Style.Add("display", "block");  
    }  
}  
0 references
```

- הקוד נכתב בדף ה Master Page.
- יש להוסיף לתפריט אותו אנחנו מבקשים להסתיר `runat="server"`, על מנת שהשרת "יכיר" אותו.
- הפקודה מוסיפה עיצוב CSS לתגית.

- הסתרת תפריטים אינה מספיקה כדי למנוע כניסה לדפים ללא הרשאה. עלינו להוסיף הגנה נוספת באמצעות קוד בדף.
- נוסיף באירוע Page_Load בדיקה אם המשתמש מורשה להכנס לדף. אם לא מורשה – יועבר לדף הראשי או לכל דף אחר שנבחר.

```
0 references
protected void Page_Load(object sender, EventArgs e)
{
    if (!(bool)Session["Login"])
    {
        Response.Redirect("http://localhost:63171/Pages/Main2.aspx");
    }
}
```

- עליכם להמיר את כל הדפים בפרויקט לדפי aspX המבוססים על Master Page.
- עליכם לתכנת בצד השרת את הדברים הבאים בהתאם לאפיון להלן:
 - הרשאת דפים – משתמש לא רשום יוכל לצפות רק בדף הראשי, דף כניסה ודף הרשמה. יתר הדפים יהיו פתוחים רק למשתמש רשום.
 - פרטי המשתמשים בשלב זה יירשמו בקובץ Global במשתנה מסוג application. אפשרות נוספת לכלול את הפרטים במחלקה סטטית בפרויקט ולרשום שם שמות משתמש וססמאות.
- בכותרת של כל הדפים - יש לציין את שם המשתמש המחובר לאתר או לציין כי מדובר באורח.
- רשות - להוסיף כפתור ל- Logout.