

## עבוא לתכנות בסביבת האינטרנט – עשיית קזינו

ההוראה של "תכנות בסביבת האינטרנט" (פרוייקט היחידה ה-3), מכילה הרבה נושאים, השייכים לסביבות שונות. כדי לא ללמד הכל ביחד, ולגרום לתלמידים בלבול רב המרפה את ידיהם כבר בשלב מוקדם זה, מורים רבים מלמדים את עולם ה-Asp.net, תוך כדי יצירת "אתר" קטן המרכז את הנושאים הבסיסיים, וזאת לפני שנכנסים לעולם ה-DataBase. משימת קזינו זו מכסה את התרגול הנלווה להוראת עולם ה-Asp.net ב-4 שלבים תוך שימת דגש על הנקודות החשובות והעיקריות ללמידה ביחידה זו.

במשימה זו, המחולקת ל-4 מערכי שיעור (4 שלבים) הנבנים אחד על-גבי השני, נסביר / נמחיש / נתרגל את:

- שינוי דף ה-Html (aspx) תוך שימוש במשתנים של צד שרת (פלט לתוך דף Html, צד לקוח).
- קליטת נתונים מהלקוח תוך שימוש בשדות קלט שונים (טקסט, רדיו, checkbox, רשימה נפתחת Select)
- עבודה מול הלקוח בשיטות Request השונות (Get & Post)
- שימוש + המחשה למשמעות ולצורך בשימוש ב-Session.
- תרגול שיטות שונות למעבר בין דפים (בצד שרת ו/או צד לקוח)
- עבודה עם MasterPage.

**משימה זו מיועדת לתלמידים שלמדו Html (ללא שדות הקלט).  
זו משימת "מבוא" לפני בניית האתר האמיתי, המשתף גם עבודה עם DataBase.**

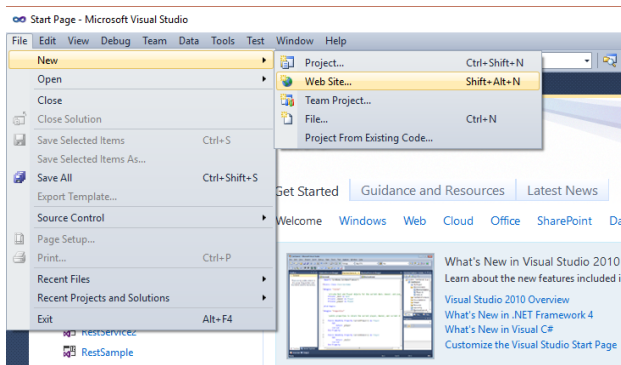
כל אחד מ-4 מערכי השיעור הנ"ל נבנה עבור שיעור של 3 שעות. כאשר בחלק הראשון המורה מראה / מדגים לפני כל הכיתה, עם הסברים מילוליים, ובשני התלמיד מתרגל את החומר שנלמד, בעצמו.

כל שיעור נבנה, כנדבך נוסף, על-גבי התוצרים (אתר) של השיעור הקודם.

בסיום משימה זו ("משימת הקזינו") התלמיד בנה "אתר" המכיל:

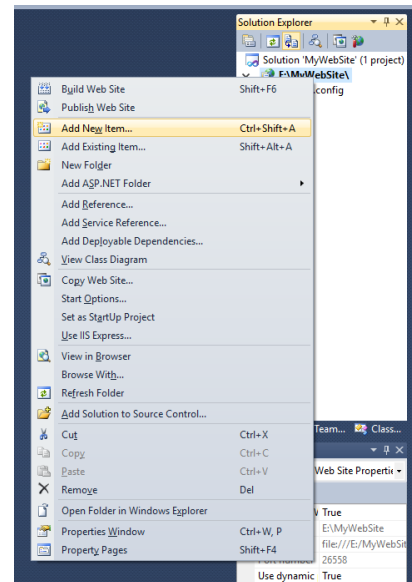
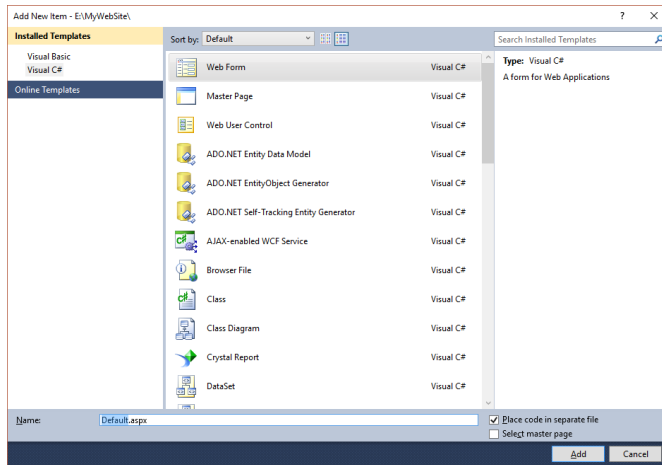
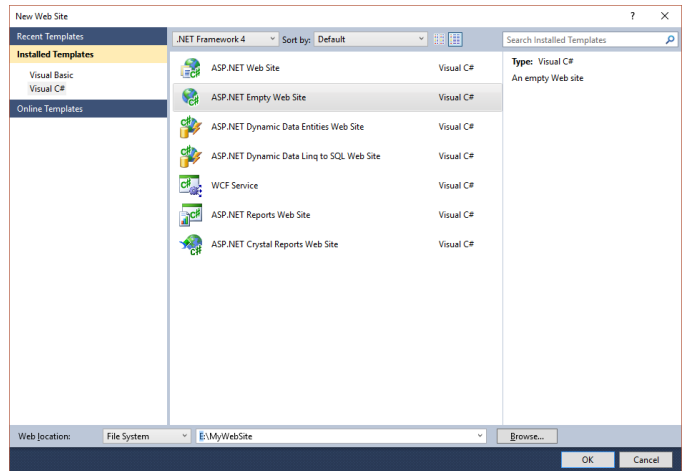
- קלט מ-4 סוגי שדות שונים.
- שימוש ב-session + איפוסו.
- 3 מעברים שונים בין דפים.
- שימוש ב-MasterPage.
- 2 סוגי קזינו אחד העובד בשיטת Post והשני בשיטת Get.

# יצירת אתר "קרינו" בסביבת VisualStudio



נפתח את ה-VisualStudio ונבחר file->new->website

ושם נבחר ASP.NET Empty Web Site.  
 כפתור Browse מאפשר לנו למקם את האתר כרצוננו (נגיד ב-Desktop) וכן לבחור לו שם כרצוננו, למשל MyWebSite.

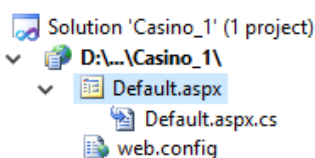


בצד ימין, בחלון Solution Explorer, נעמוד על שם האתר, ימני עם העכבר ונבחר : Add New Item.  
 בחלון שנקבל נבחר Web Form וניתן לו שם כרצוננו ( גם Default.aspx יתאים לנו, כרגע).

קיבלנו דף המורכב משני חלקים :

1. Default.aspx - מכיל קוד שרץ בצד לקוח : Html , JavaScript , .css

2. Default.aspx.cs - מכיל קוד C# שרץ בצד שרת (CodeBehind) ובו את הפעולה הראשית של הדף Page\_Load.

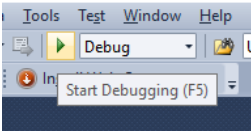


**חשוב:**

ליד שם הדף שנפתח יופיע הסימן >.

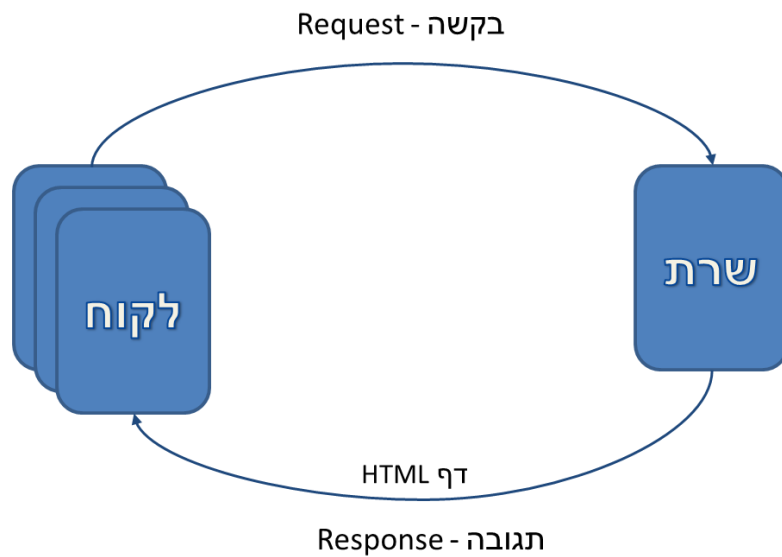
לחיצה עליו תציג את שני הקבצים מהם הדף מורכב, שני חלקי הדף.

כאשר כתבנו Html ו- JavaScript עבדנו בעצם בצד לקוח בלבד. בסביבת VisualStudio לא נעבוד עם דפי Html אלא עם דפי .Aspx. דפים אלו מכילים Html + JavaScript ו-css, השייכים לצד לקוח וכן קוד ב- C#, השייך לצד השרת (CodeBehind). נוכל לכתוב Html + JavaScript ו-css בדף, כפי שכבר למדנו. כדי להריץ את האתר, דרך שרת וירטואלי של ה-VisualStudio נלחץ F5 או על המשולש הירוק הממוקם ב-VisualStudio למעלה באמצע, ממש מתחת לשורת התפריטים.



נגלה שנפתח לנו דפדפן אם הקוד Html + JavaScript שכתבנו בצד לקוח. בשורת הכותרת של הדפדפן נקבל את הבקשה של הדף Default.aspx מהשרת.

בכל בקשה לדף, השרת "מייצר" דף html ושולח לדפדפן. והוא מציג אותו לנו, ללקוח.



ניתן להציג בדף ה-HTML משתנים שהוגדרו (כ-protected) בצד השרת (cs) ע"י הפקודה: `Response.Write(variable);` אך כך ערך המשתנה יוצג בתחילת הדף, מחוץ לתחום ה-HTML.

לכן נגדיר פקודה זו בדף ה-HTML בתוך תחום המגדיר פקודות מהשרת (cs): `<% Response.Write(variable); %>` או בקיצור: `<% =variable %>`

## המשימה :

נייצר "מכונת מזל" המגרילה, בכל פעם, 3 מספרים (בין 1 ל-9) ומחשבת את הנקודות בהם "זכינו" בעקבות "הגרלה" זו. חישוב הנקודות:

הניקוד שיצבור השחקן	היחס בין שלושת המספרים
-1	שלושת המספרים שונים זה מזה
+1	שניים מבין המספרים שווים זה לזה
+10	שלושת המספרים זהים זה לזה
+50	שלושת המספרים זהים ל-7

בכל פעם נציג את המספרים שהוגרלו, את מספר הנקודות בהם "זכינו" וכן את סה"כ הנקודות שצברנו עד כה. כדי לזמן את הדף שוב ושוב נשתמש באופציית הריענון (Refresh) של הדפדפן.

```
Random rnd = new Random();
protected void Page_Load(object sender, EventArgs e)
{
    y = rnd.Next(1, 11);
    x = rnd.Next(1, 11);
    z = rnd.Next(1, 11);
}
```

הדף של "מכונת המזל" ייבנה ויוצג בצד לקוח, ע"י html. הגרלת המספרים תעשה, בפעולה Page\_Load, בצד שרת, ע"י עצם מסוג Random וזימונו 3 פעמים, עבור 3 מספרים שונים.

כמו-כן, בצד שרת, נחשב את הנקודות בהם "זכינו" בעקבות "הגרלה" זו.

כדי להציג את המספרים ש"עלו בהגרלה" וכן את הנקודות בהם "זכינו", ערכים הנמצאים במשתנים הנמצאים בצד שרת (CodeBehind) במיקום מסויים בדף ה-html נגדיר תחום צהוב <%....%>, המאפשר כתיבת קוד של צד שרת באמצע דף html, קוד לקוח.

בתחום (צהוב) זה נכתוב את הפקודה <% Response.Write(variable); %> או <% =variable %>. ב-2 המקרים ייכתב ערך המשתנה variable במקום התחום הצהוב הנ"ל תוך שימוש בפקודה Response.Write(variable); השימוש ב- =Variable הינו כתיבה מקוצרת וקלה לפקודה Response.Write(variable);

## אופס,

משום מה מספר הנקודות בו "זכינו" לא צומח. אלה כאילו הנקודות מחושבות בכל סיבוב מחדש, הן "נשכחות" מסיבוב אחד למשנהו. צורת העבודה באינטרנט היא "סרת מצב" (Stateless). כלומר בכל פעם שמתקבלת בקשה השרת מטפל בה, מייצר דף Html כתשובה ו"שוכח" את כל המשתנים.



כדי שבכל אופן נוכל "לזכור" משהו, למשל את מספר הנקודות בהן זכינו, נשתמש ב-Session, שזה משתנה הקיים בצד השרת ובסביבת עבודה של שרת-לקוח !!

```
Session["totalPoints"] = total; : שמירת הנקודות ב-Session
```

הפעולה Page\_Load נקראה ראשונה בכל בקשת דף מהשרת, לכן בתחילת הפעולה Page\_Load נרצה ל"שחזר" את הערך הזה, בחזרה לתוך המשנה total.

אך... א. המשתנה ב-Session הוא כללי (מטיפוס Object) ולכן יש להמיר אותו ל-int לפני שנוציב את הערך בחזרה במשתנה total.

ב. יש לוודא שאכן יש ערך במשתנה (ששונה מ-null) וזאת לפני ההמרה של הערך ל-int. (קריטי עבור הפעם הראשונה, כי אז אנו מנסים לשחזר את הערך מה-Session וזאת עוד לפני ששמנו בו ערך, אפילו לא פעם אחת).

```
if (Session["totalPoints"] != null)
    total = (int)Session["totalPoints"];
else total = 0;
```



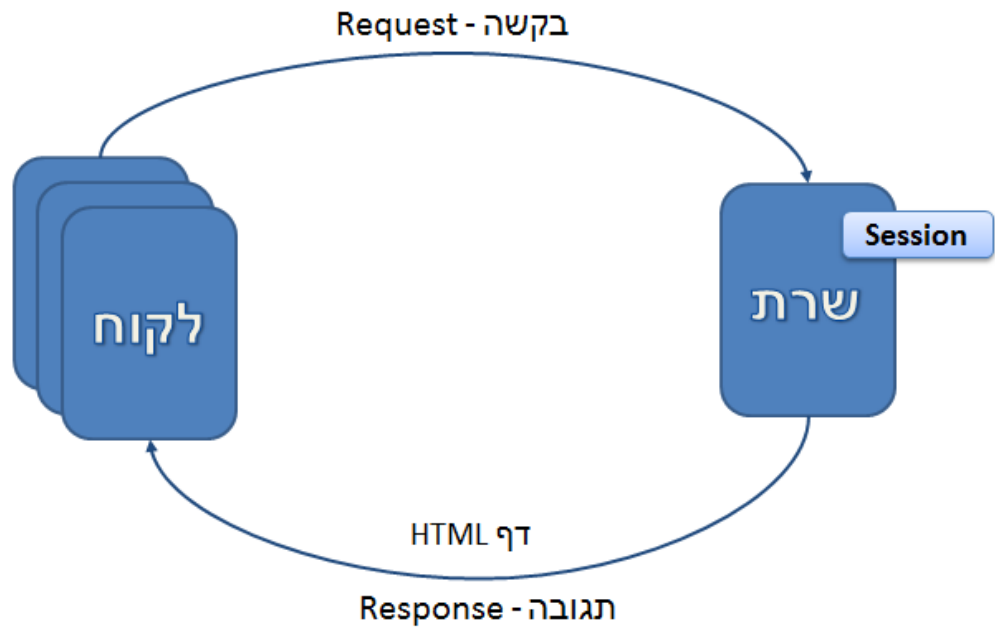
עכשיו הכל עובד נכון ותקין !

## חשוב:

יש לוודא שהאתר שלנו גם מעוצב ויפה.  
למשל.... שהמספרים שהוגרלו יוצגו ברשימה (ol או ul).  
שתהייה הודעה יפה וברורה בכמה נקודות זכינו, בסבב זה ו..בסה"כ.

```
</body>
  <h1 align="center"><u>קזינו</u></h1>
  המספרים שעלו בגורל:
  <ol>
  <li><%=y %></li>
  <li><%=x%></li>
  <li><%=z %></li>
  </ol>
  נקודות <span dir="ltr"><%=point %></span> זכית ב
  <br />
  נקודות <span dir="ltr"><%=total %></span> סה"כ לזכותך
</body>
```

## סיכום



# שלב ב': קבלת פרמטרים

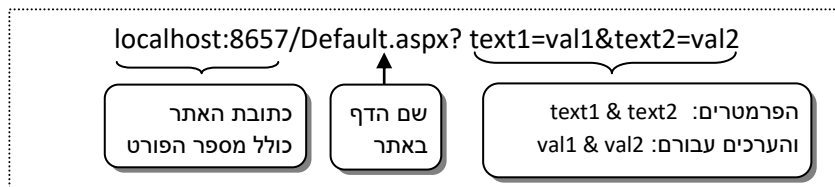
נוסיף ל"אתר" שלנו כותרת יפה (H1) "שלום <משתמש>". לשם כך עלינו לקבל את שם המשתמש וזאת ע"י יצירת טופס בשפת ה-HTML (תגית Form). בתחום נגדיר שדה קלט (Input) מסוג Text. ו..כן כפתור השולח את הנתונים לשרת – שדה Input מסוג Submit. כדי לוודא ששדה הקלט יגיע לשרת נגדיר לו מאפיין Name. (את התגיות ניתן לגרור לתוך ה-HTML מתוך "ארגז הכלים" בקטיגוריה HTML).



כדי לקבל את הנתונים (מהטופס Form) מתוך הבקשה, נשתמש באובייקט (עצם) של מחלקת **Request**. נוכל לראות את הפרמטרים שהועברו כחלק מהבקשה (מהכתובת בדפדפן), בזמן הרצה.

localhost:8657/Default.aspx?name=Dorit

ה-URL של הדף ייראה, לדוגמא:



במחלקת Request יש מאפיין Count המחזיר לנו את מספר הפרמטרים שהגיעו בבקשה (מהטופס). תמיד, לפני ש"ניקח" מהבקשה את הערך שהמשתמש הקליד בטופס נוודא שאכן יש פרמטרים בבקשה.

```
string header="";  
if (Request.QueryString.Count>0)  
    header="<h1>שלום" + Request.QueryString["name"]+"</h1>";
```

מכיון שבפעם הראשונה המשתמש טרם הקליד את שמו, אין משמעות להציג את ההודעה בכותרת. ונרצה שלא להציג את הכותרת וזאת עד שנקבל את השם מה-Request. כלומר נייצר משתנה שיכיל "" (בהתחלה) או את תגית הכותרת ("<h1>משתמש</h1>"), כולל שם המשתמש, אחרי שקיבלנו את שם המשתמש. משתנה זה יכיל בעצם את כותרת הדף.

בפעם השנייה, אין טעם לבקש מהמשתמש להקליד שוב את שמו, כלומר, יש להסתיר את הטופס Form, או, למען הדיוק, נייצר תחום div שיכיל את כל החלקים של הטופס שנרצה להסתיר מהמשתמש (כלומר כל שדות הקלט למעט כפתור השליחה submit המאפשר לנו, בעצם, refresh לדף).

לשם הצגה / הסתרה נשתמש ב-css, במאפיין .style

נכתוב "visibility:visible" או "visibility:hidden".

אם נייצר משתנה vis מסוג string שיכיל את הערך "hidden" או "visible" נוכל לכתוב: `style="visibility :<%=Vis%>";` ואז התחום div יוסתר או יוצג ע"פ ערך המשתנה vis, ע"פ הערך שקיבל בצד שרת.



## שימו: ♥

למען הסדר הטוב תמיד נבדיל בין ה-"מה" ל-"איך".  
כלומר נפריד בין יצירת ערכים נכונים, ע"פ הלוגיקה שנרצה באתר, במשתנים ← בצד שרת.  
לבין הצגתם במיקומים המתאימים להם, בדף ה-html (Aspx) ← צד לקוח.

את הערכים נוכל למקם בכל מקום בו נחפוץ:

- אפשר להציג נתונים, כמו הנקודות שזכינו.
- אפשר להציג "חתיכת HTML" (כותרת הדף), כי HTML הוא בסיס "כ" טקסט.
- אפשר להציג חלק של CSS (למשל המשתנה vis), כי גם הוא בסיס "כ" טקסט.

ניתן להשתמש במשתנים של צד שרת כדי לבנות את דף ה-html כרצונו.  
ברגע והשרת סיים את עבודתו, ובנה את דף ה-html, ע"פ בקשתנו תוך שימוש במשתנים של צד שרת, הוא אותו לדפדפן.  
הדפדפן מקבל דף HTML בלי שיש לו מושג כלל אם חלק כלשהו נכתב ע"י משתנה או בצורה סטטית.

חץ לשמאל+Alt	אחורה
חץ לימין+Alt	קדימה
Ctrl+R	טען מחדש
Ctrl+S	שמור בשם...
Ctrl+P	הדפס...
	העבר...
	תרגם ל-English
Ctrl+U	צפה במקור הדף
Ctrl+Shift+I	בדוק

ניתן לבקש מהדפדפן להציג "קוד מקור" (ימני בעכבר, ו... "צפה במקור הדף") ואז נראה את ה-html כפי שנשלח לדפדפן.  
ניתן לראות שאין כלל הבדל בין קוד HTML שמקורו בדף ASPX לבין זה שמקורו במשתנה.



## המשימה :

נשדרג את ה"קזינו" שלנו כך שיקלוט מהמשתמש:

- שם פרטי + שם משפחה - Input מסוג Text
- עיר מגורים - Select
- סוג הקזינו (1/2/3) - 3 תגיות Input מסוג Radio לכולם אותו Name אך לכל אחד מהם יהיה Value שונה (1/2/3).
- האם להציג את המספרים שהוגרלו ? - Input מסוג CheckBox.

## סוגי הקזינו :

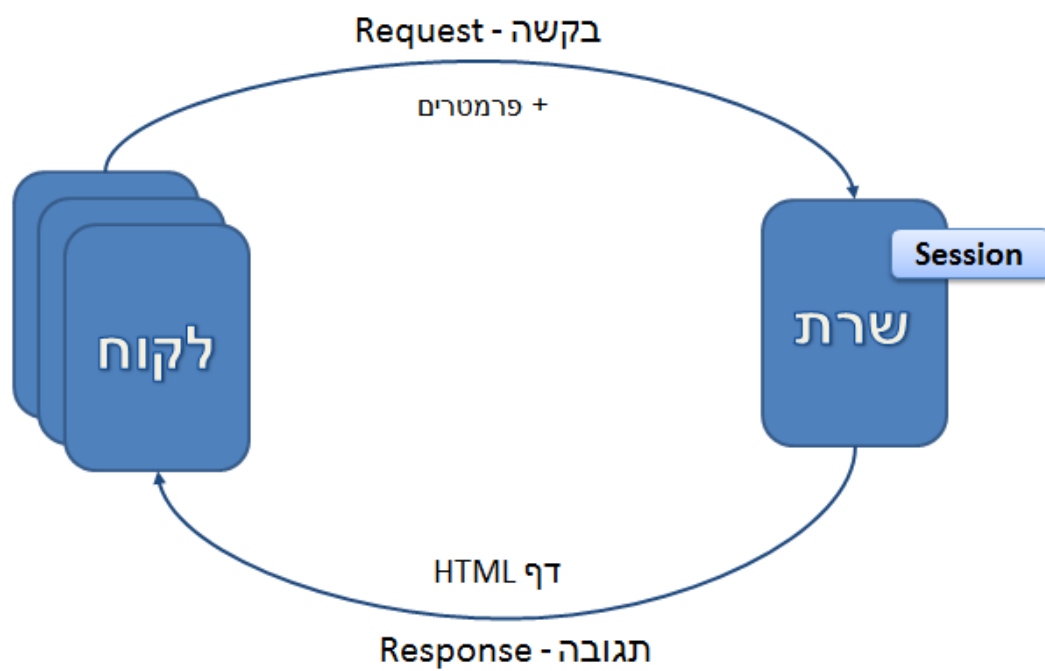
1. כפי שהגדרנו בשלב הקודם.
2. כמו סוג 1 וגם – אם שלושת המספרים זוגיים אז זכינו ב-2+ נקודות.
3. 7 בום – כלומר אם סכום 3 המספרים שהוגרלו מתחלק ב-7 או מכיל 7 אז זכינו ב-40+ נקודות.

## שימו: ♥

נבחין בין 3 מקרים:

- א. הרצה ראשונית, ללא פרמטרים כלל.
- ב. קבלת פרמטרים מהבקשה (מהטופס) ושמירה ב-Session.
- ג. שחזור הערכים מה-Session והצגתם. – ללא הצגת הטופס בדף ה-HTML







# שלב ג': מטברים בין דפים באתר

נוכחנו שקליטת נתונים והטיפול בהם, באותו הדף, מוסיף הרבה מורכבות מיותרת לאתר. אם נוסיף לתגית הטופס (Form) את המאפיין Action אז נוכל לקבוע לאיזה דף תגיע הבקשה הכוללת את הנתונים מהטופס.

## המשימה :

נשדרג את ה"קזינו" שלנו כך שיתפרס על 3 דפים שונים:

- דף המכיל טופס הקולט מהמשתמש את כל הנתונים (כמו שכתבנו קודם)
- דף המכיל את ה"קזינו" עצמו, ב-3 סוגים – ממש כמו שכתבנו קודם
- דף המכיל הודעה "כל הכבוד" ומאפשר מעבר לדף הטופס, ל"סיבוב" נוסף.

א. נפצל את דף ה"קזינו" כפי שהוגדר קודם ל-2 דפים. אחד - דף שיכיל את הטופס עצמו, נוסיף בתגית ה-Form את מאפיין Action כך שהבקשה + הנתונים יגיעו ישירות לדף ה"קזינו".

ושני - דף ה"קזינו" עצמו המקבל את הנתונים מהטופס, שומר אותם ב-Session (בפעם הראשונה), מחשב את הנקודות בהן זכינו ע"פ המספרים שהוגרלו וסוג ה"קזינו" שבחרנו ומציג את התוצאות למשתמש.

ב. במידה ו"זכינו" ב-100± נקודות נעבור לדף החדש "כל הכבוד".

מעבר זה יתבצע בצד שרת ע"י הפקודה: `Response.Redirect("win.aspx");`

ג. בדף "כל הכבוד" נוסיף להודעה את שם המשתמש והעיר ממנו בא.

נתונים אלה נשמרו, קודם, ב-Session ולכן ניתן להציגם ישירות בדף ה-`<%=Session["name"]%>` HTML.

Session הוא משתנה בצד שרת ה"שייך" למשתמש מסוים אך לא לדף מסוים. לכן הוא נגיש בכל אחד מהדפים. 💡

ד. נוסיף תגית קישור (`<a href=...></a>`) שתעביר אותנו, בפשטות ובקלות, מדף "כל הכבוד" לדף הטופס כדי למלא נתונים ל"סיבוב" נוסף.

נוודא שהכל מעוצב יפה, נריץ ו...נוודא שגם עובד יפה.

אופס,

משום מה הכל עובד עבור המשתמש הראשון.

אך...כשמכניסים את פרטי המשתמש השני ישר עוברים לדף "כל הכבוד" !!

שכחנו לאפס את הנקודות שנצברו !!!



באיזה דף נכון לאפס את הנקודות?

א. דף הקזינו – בטח שלא, שם ממש ממש צריך את השמירה של הנקודות ב-Session.

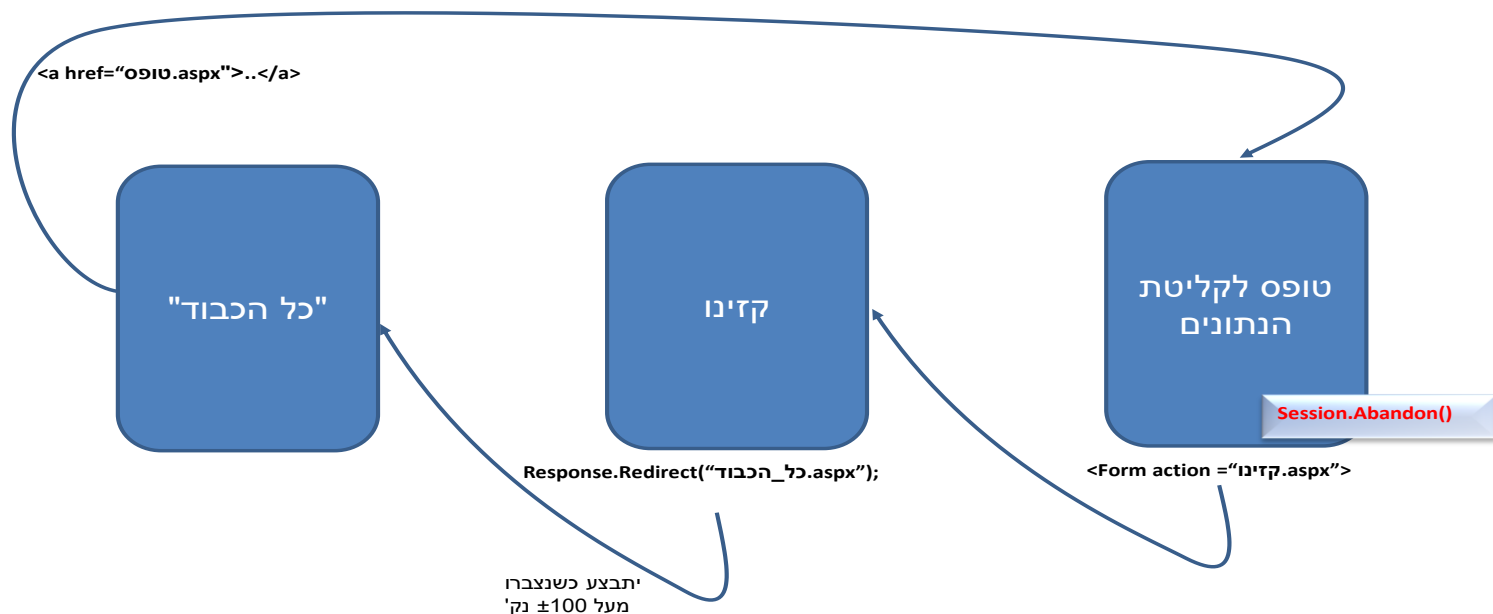
ב. דף "כל הכבוד" – אנחנו צריכים את נתונים ה-Session להצגה. והמעבר לדף הטופס מתבצע בלקוח, אחרי שהדף כבר נבנה בשרת. כלומר-מאוחר מדי.

ג. מה שנותר- דף הטופס.

איפוס כל ה-Session יתבצע ע"י נטישת ה-session, הפעולה: `Session.Abandon()`.

## סיכום

- נוכל לראות שיש לנו 3 סוגים שונים עבור מעברי דף:
1. בטופס ע"י המאפיין action
  2. בקוד, בצד שרת, ע"י הפקודה Response.Redirect
  3. בתגית html רגילה.



# שלב ד': טכודה כ-Post ואחידות

ראינו כי הפרמטרים מהטופס עוברים לשרת בבקשה Request, כחלק מהכתובת. אך לא תמיד נרצה שהפרמטרים יעברו כך, בצורה כל-כך גלויה.

בתגית הטופס (Form) קיים מאפיין Method הקובע את שיטת העברת הפרמטרים לשרת. קיימות שתי שיטות להעברת פרמטרים בבקשה Request: **Get** - הפרמטרים של הבקשה הם חלק מכתובת הדף, חלק מה-URL, ולכן יופיעו בחלון הכתובת בדפדפן. זו ברירת המחדל, בה השתמשנו עד כה. **Post** - הפרמטרים של הבקשה מועברים לשרת "מאחורי הקלעים" ולכן לא יופיעו בשורת הכתובת בדפדפן.

## חרונות לשיטת Get:

- העברה גלויה (ב-Get) לא "בריאה" אם רוצים להעביר נתונים רגישים כמו סיסמאות.
- העברת פרמטרים בשורת הכתובת מגבילה את אורך הבקשה ל-1024 תווים.
- העברה בשורת הכתובת מגבילה את הנתונים לתווים "דפיסים", לא נתונים בינאריים.

## חרונות לשיטת Post:

- לא תמיד ניתן להעביר נתונים ב-Post. למשל כשהשרת "מייצר" בקשה ע"י Request.Redirect או כששתמשים ב-html בתגית קישור (<a>).

ההתייחסות לפרמטרים שהועברו בשיטת Get היו ע"י **Request.QueryString**  
בשיטת Post כל התייחסות לפרמטרים תהייה ע"י **Request.Form**

## **המשימה :**

נייצר שני אתרי "קזינו" אחד שעובד בשיטת Get (מה שכבר עשינו) והשני שעובד בשיטת Post. כל השאר יכול להיות זהה.

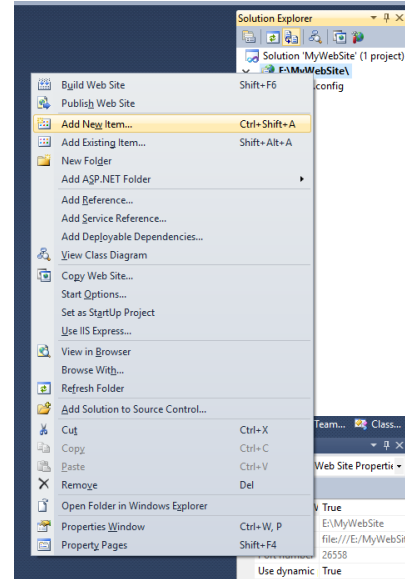
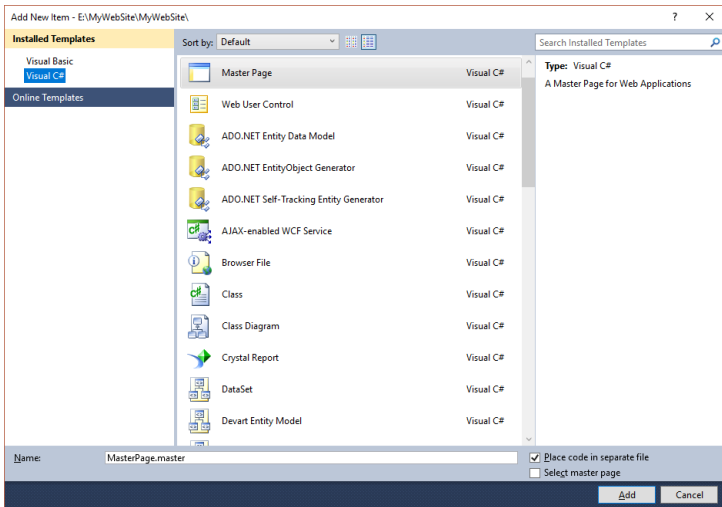
כרגע יש באתר שלנו 3 דפים להצגה בכל אחת מהשיטות (כלומר 6 דפים):

1. דף הטופס.
2. דף הקזינו.
3. דף "כל הכבוד".

כדי לשמור על אחידות בכל הדפים באתר נהוג להשתמש בתבנית הקובעת את המידע שיופיע בראש הדף ובתחתיתו. בכותרת הדף - יופיע בד"כ שם האתר, לוגו, תפריט ראשי, וכל דבר שיופיע בכל דפי האתר. בתחתית הדף - פרטי האתר, כתובת, טלפון או מידע אחר. כדי ליצור את המראה האחד נשתמש ב-MasterPage. כל דפי האתר שנייצר יקושרו לדף MasterPage.

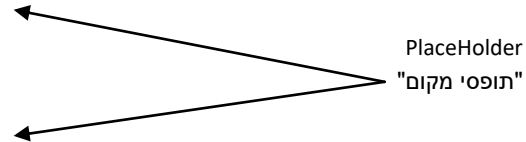
ליצירת MasterPage:

בחלונית Solution Explorer נלחץ ימני בעבר על שם האתר, AddNewItem -> MasterPage, בחלונית שנקבל אין מה לשנות אז- ישר לחיצה על Add. חשוב לשים לב ששפת הדף הינה C#.



גם הדף MasterPage, כמו כל דפי ה-Aspx, מורכב מ-2: חלק Html (צד לקוח) וחלק C# (צד שרת). בצד ה-html נמחק את תגית ה-Form (כי... היא לא משותפת לכל האתר) כך שהקוד שנוטר הוא:

```
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs" Inherits="MasterPage" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transit
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
<asp:ContentPlaceHolder id="head" runat="server">
</asp:ContentPlaceHolder>
</head>
<body>
<asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
</asp:ContentPlaceHolder>
</body>
</html>
```



נעצב את דף ה-MasterPage כרצוננו, תוך שימוש בטבלה. נמקם את הדברים הקבועים באתר: כותרת, תפריט לדפים השונים באתר, תמונת רקע ועוד כרצוננו. וכן נמקם את התוכן המתחלף מדף לדף (מיקומו ייתפס ע"י ה- Placeholder).

למה הדבר דומה ?

לתמונה בלונה פארק בה אנו מכניסים את ראשו ל"חורים" הקיימים וכך נוצרת תמונה אחידה, אך שונה בפרצופים הנמצאים בתוכה.



תמונת ה"לונה פארק" עם הכיתובים והרקע זה ה-MasterPage. הראשים בתמונה הם החלקים המתחלפים מדף לדף. שימו לב, יש 2 Placeholder. אחד הממוקם ב-head והשני ב-Body.

**חשוב:**

מכיון שהטופס (Form) שלנו לא ייהיה זהה בכל הדפים באתר, נוציא אותו מדף ה-Master-Page

דף ה-MasterPage יראה, למשל :

```

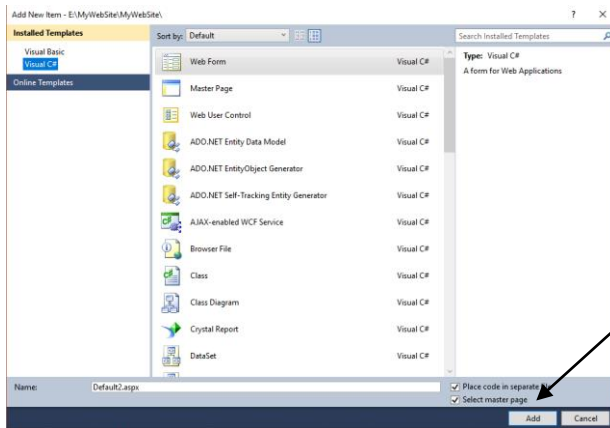
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs" Inherits="MasterPage" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transit
<html xmlns="http://www.w3.org/1999/xhtml" dir="rtl">
<head runat="server">
<title></title>
<asp:ContentPlaceHolder id="head" runat="server">
</asp:ContentPlaceHolder>
</head>
<body>
<table width="100%">
<tr colspan="2">
<td colspan="2">כותרת האתר</td>
</tr>
<tr>
<td width="30%">תפריט</td>
<td>
<!-- תוכן מחולף -->
<asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
</asp:ContentPlaceHolder>
</td>
</tr>
</table>
</body>
</html>
    
```



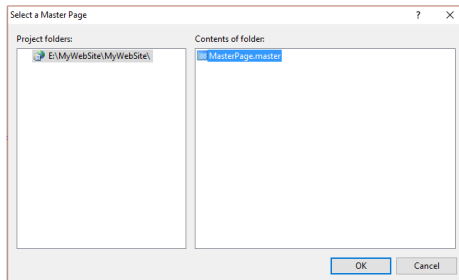
או:

```

<%@ Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs" Inherits="MasterPage" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitio
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
<asp:ContentPlaceHolder id="head" runat="server">
</asp:ContentPlaceHolder>
</head>
<body>
<table width="100%">
<tr>
<td colspan="2">כותרת האתר</td>
</tr>
<tr>
<td>תפריט</td>
<td>
<!-- תוכן מחולף -->
<asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
</asp:ContentPlaceHolder>
</td>
</tr>
</table>
</body>
</html>
    
```



יצירת דף Content המחובר ל-MasterPage:  
 ע"י AddNewItem -> WebForm  
 יש לבחור SelectMasterPage.  
 וניתן לו שם כרצוננו (למשל default.aspx)  
 (בתור דף ראשון).



בחלון שנקבל נבחר את ה-MasterPage שנרצה (במידה ויש מספר MasterPage) ונאשר.

קיבלנו דף aspx המורכב משני קבצים. חלק html (צד לקוח) וחלק C# (צד שרת).

חלק ה-html ייראה עכשיו כך:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true" CodeFile="Default.aspx.cs"
<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
</asp:Content>
```

יש לנו שתי תגיות תוכן (Content) . אחת שתוכנה ימוקם ב-PlaceHolder ש-Head והשניה שתוכנה ימוקם ב-PlaceHolder ש-Body.

- כדי להעביר את הדפים שכבר ייצרנו לדפי-Content המחוברים ל-MasterPage
- ✘ נעתיק את תוכן ה-Head מהדף המקורי לתוך ה-Content הראשון (של ה-Head).
  - ✔ נעתיק את תוכן ה-Body מהדף המקורי לתוך ה-Content השני (של ה-Body).
  - ➔ וכן נעתיק את תוכן צד שרת (דף ה-C#) מהדף המקורי לדף החדש שייצרנו.



עתה קיבלנו דף דומה אך, מחובר ל-MasterPage, כלומר משולב בתבנית של האתר.

נסיים את בניית האתר שלנו כך שיכיל שני קישורים, אחד ל"קזינו" שעובד בשיטת Get והשני ל"קזינו" שעובד בשיטת Post. כל "קזינו" מכיל, כמו קודם, 3 דפים: טופס, הקזינו עצמו ודף "כל הכבוד". מן הסתם, כל "קזינו" יתחיל מהטופס – לכן הקישורים ב-MasterPage יובילו לדף הטופס המתאים.

