תכנות באינטרנט

ASP.Net Core

MVC

גלעד מרקמן

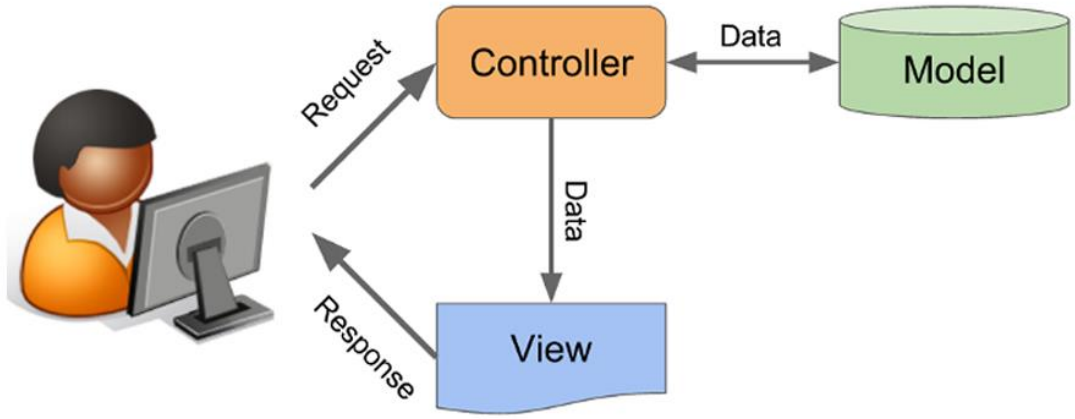
קריית החינוך פארק המדע,
נס ציונה

Microsoft
ASP.NET Core

ASP.Net Core
MVC

https://www.youtube.com/playlist?list=PL6n9fhu94yhVkdrusLaQsfERmL_Jh4XmU

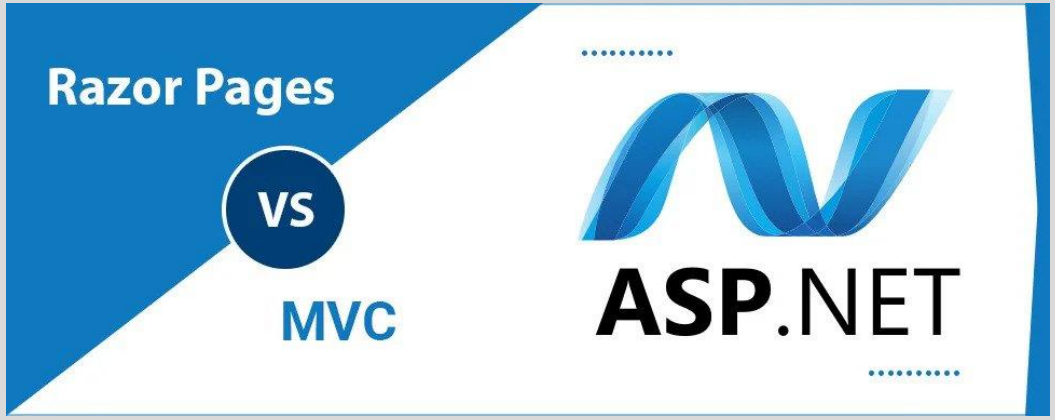
MVC – Module View Controller



Understanding asp.net MVC (Model View Controller) architecture

- המודל מחלק את הפיתוח לשלוש שכבות.
- תבנית זו עוזרת לכתוב קוד ברור, נוח לתחזוקה ובדרך כלל קצר יותר.

Razor Pages vs MVC

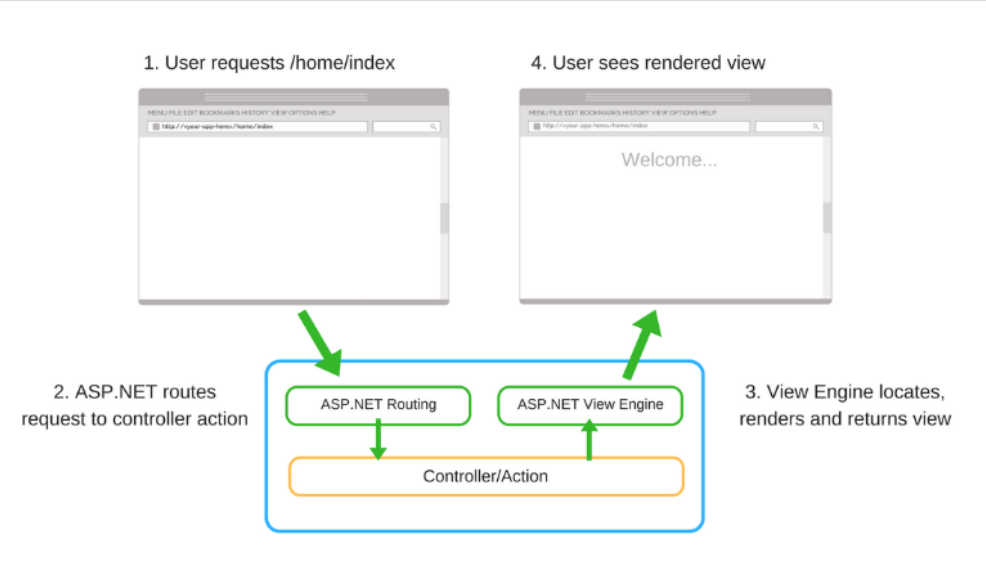


- Razor Pages דומים ל web forms הם כוללים דף HTML ואת הקוד מאחורי הדף בקובץ נפרד.

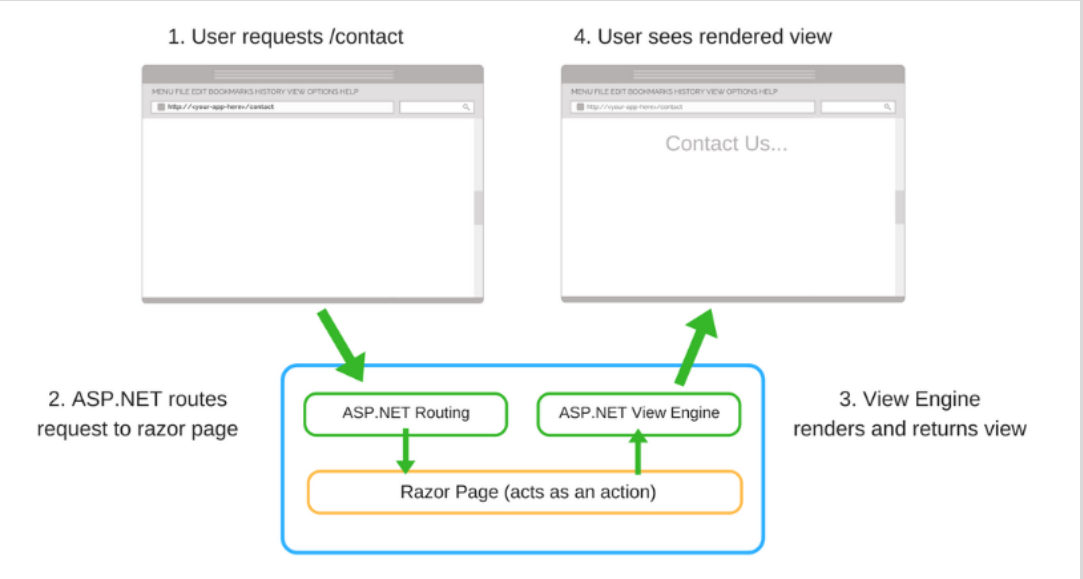
- בפרויקט MVC אנחנו משתמשים ב Razor view שכן הקוד נמצא ב Controller בקובץ נפרד לחלוטין.

NVC vs. Razor Pages

MVC

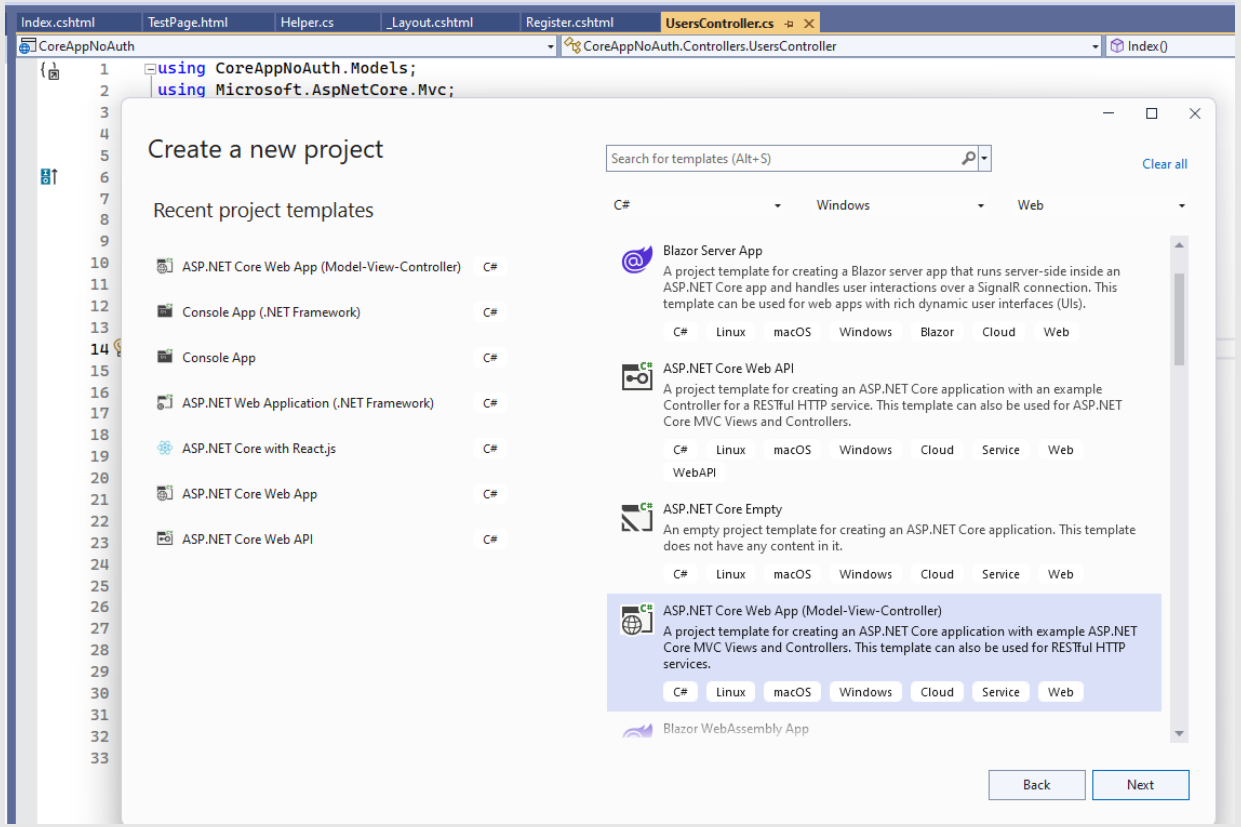


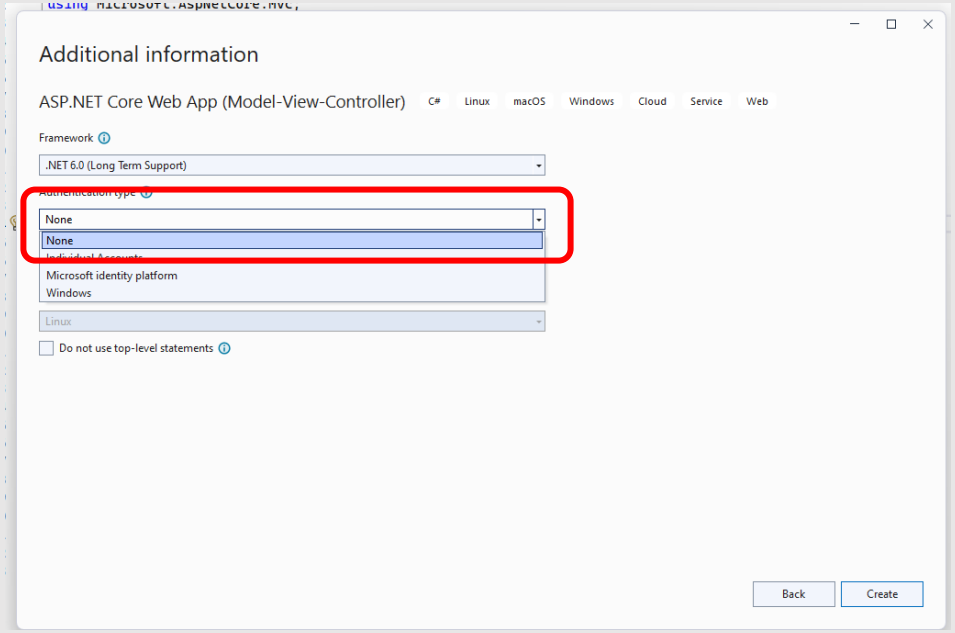
Razor Page



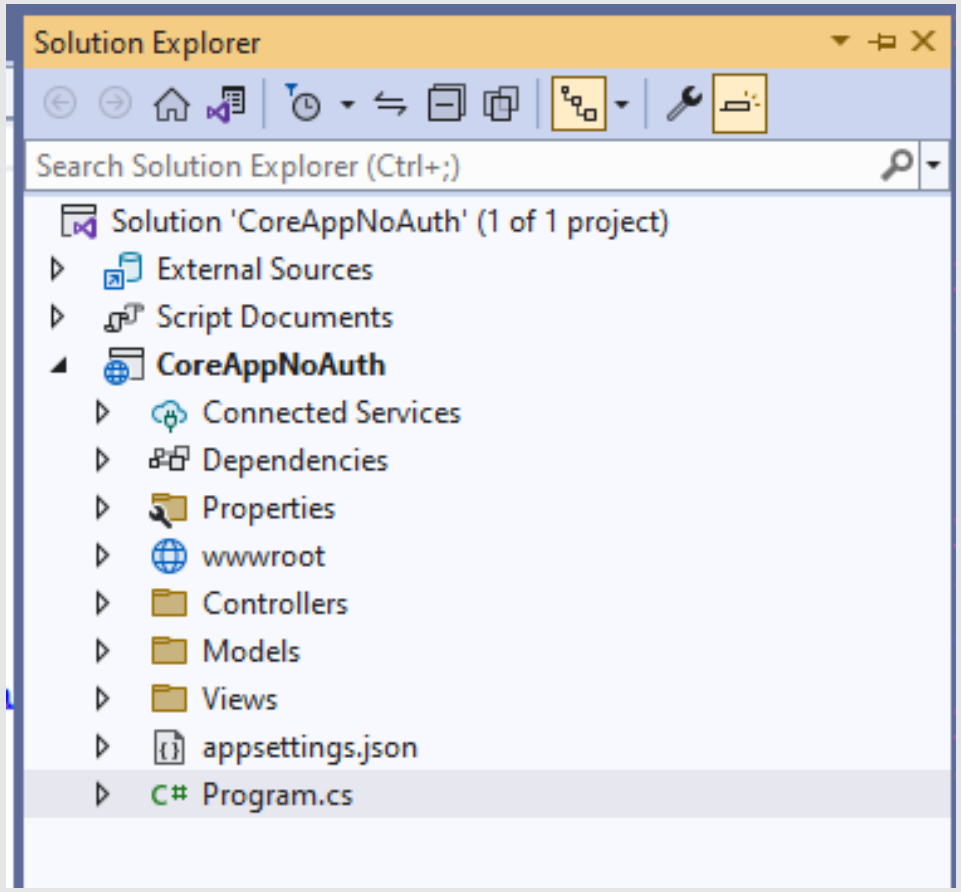
יצירת פרויקט חדש

- ללא הרשאות גישה.





מבנה הפרויקט



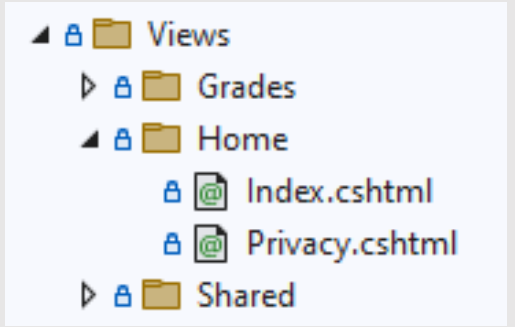
- תוכנית ראשית Program.cs.
- appsettings.json
- wwwroot – קבצים סטטיים.
- Controllers
- Models
- Views
- Data – בהמשך.

routing

- הדגמה של Routing בפרויקט הקיים באמצעות URL.
- הגדרת דף ברירת מחדל
- Controller, Action.

```
app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");
```

```
public IActionResult Index()
{
    //return Redirect("~/pages/testpage.html");
    return View();
}

public IActionResult Privacy()
{
    //return View("Privacy");
    return View();
}
```



דפי HTML בפרויקט MVC

- ניתן ליצור ולהציג דפי HTML בדיוק כמו בפרויקט מסוג Razor pages.
- הדפים (וכל הקבצים הסטטיים) ייווצרו תחת הספריה wwwroot.
- כדי להפעיל אוטומטית את דפי HTML ולא הדף הרגיל נוסיף ב HomeController את הפקודה הבאה:

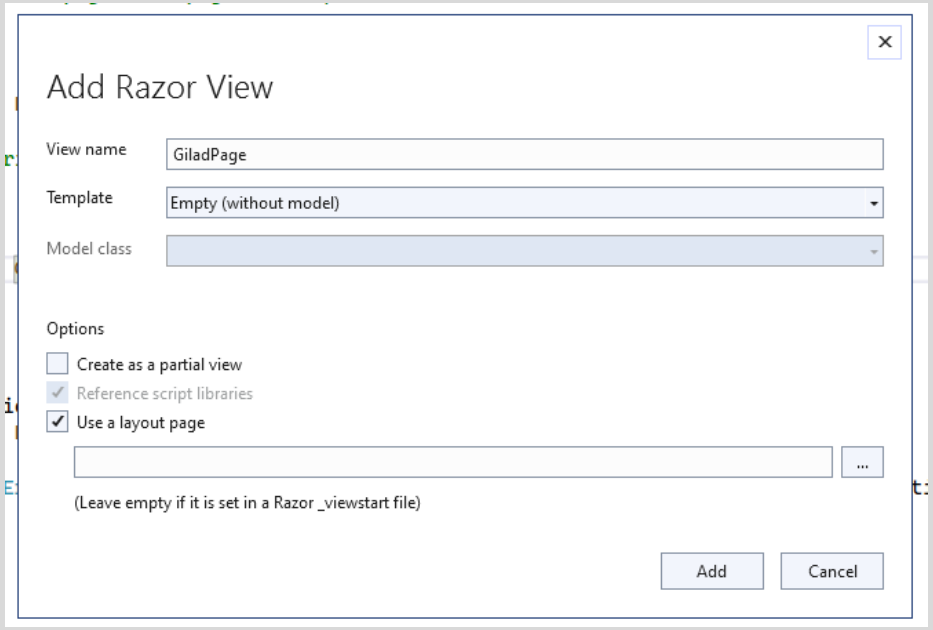
```
public IActionResult Index()
{
    return Redirect("~/pages/testpage.html");
    //return View();
}
```

יצירת VIEW חדש

```
public IActionResult GiladPage()
{
    return View();
}
```

- כתיבת פעולה חדשה ב Controller:

- קליק ימני על שם הפעולה ולחיצה על AddView, בחירה ב Razor View.
- ניתן לבחור:
  - שם הקובץ.
  - תבנית Template.
  - layout page.
- נוצר קובץ cshtml בספריה המתאימה.



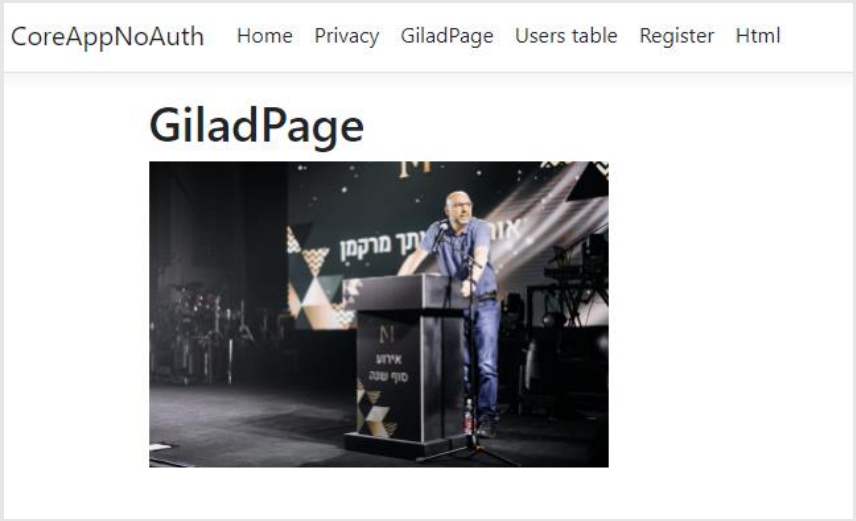
GiladPage.cshtml | HomeController.cs | Index.cshtml | _Layout.cshtml

```
1  @{
2      ViewData["Title"] = "GiladPage";
3  }
4  <h1>GiladPage</h1>
5  <img src="~/Img/Gilad.jpg" style="width:50%" />
6
```

הוספת קישור לדף בתפריט הראשי

- נעבור לתבנית _Layout.cshtml ונוסיף עוד תפריט מתאים:

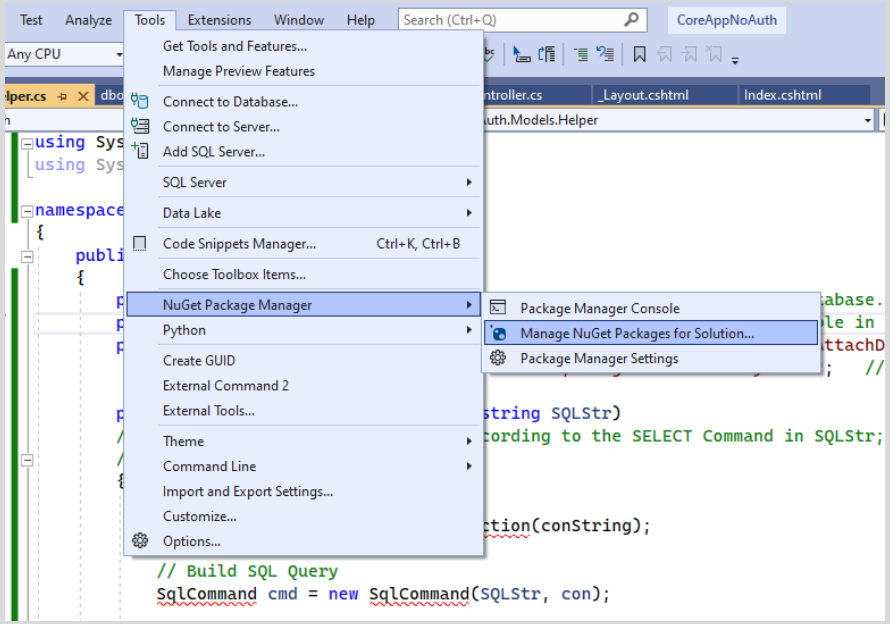
```
<li class="nav-item">
    <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>
</li>
<li class="nav-item">
    <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="GiladPage">GiladPage</a>
</li>
```

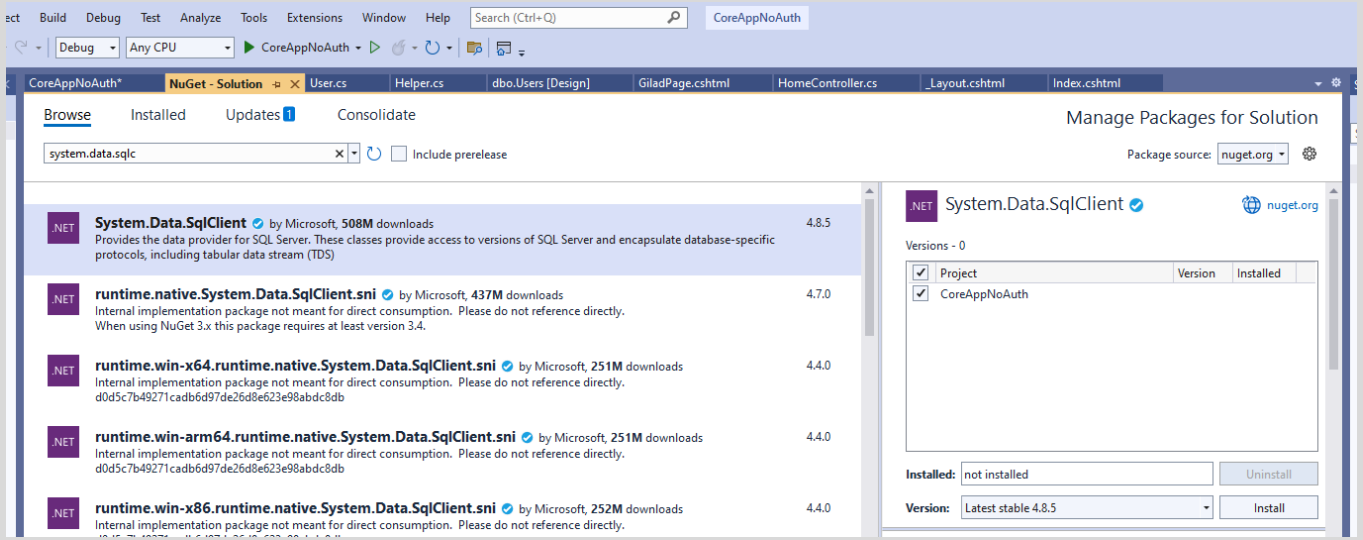


בסיס נתונים
Ado.Net

בסיס נתונים ADO.Net

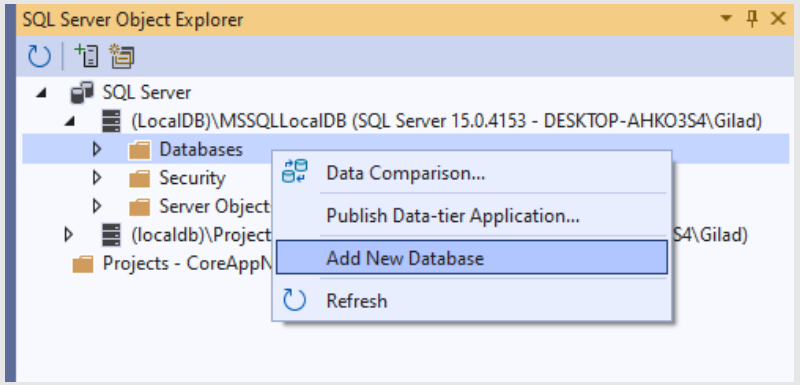
- על מנת שנוכל להשתמש ב ADO.Net אנו צריכים להתקין חבילה מתאימה.
- נשתמש ב NuGet Package Manager. ונתקין את החבילה System.Data.Sql Client.

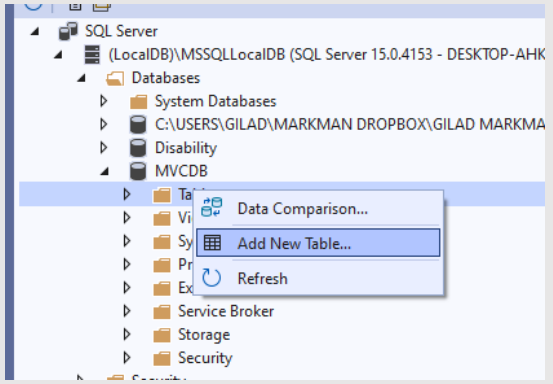




הוספת בסיס נתונים וטבלת Users

- נוסיף בסיס נתונים באמצעות SQL Server Object Explorer.
- קליק ימני על על databases ויצירת בסיס נתונים חדש.
- ולאחר מכן, קליק ימני על טבלאות ויצירת טבלה חדשה.
- ניצור טבלת משתמשים Users בדיוק כמו שלמדנו.





בניית Users Model ו- Helper

- בספריה מודל נוסיף שתי מחלקות:
  - מחלקה של User התואמת את הטבלה שלנו.
  - מחלקה של Helper בדומה למחלקה שהצגנו בקורס. נעדכן את הקוד בהתאם לטבלה שבנינו. נעדכן את מחרוזת ההתקשרות.

```
namespace CoreAppNoAuth.Models
{
    public class User
    {
        public int id { get; set; }
        public string Firstname { get; set; }
        public string Lastname { get; set; }
        public string Username { get; set; }
        public string Pass{ get; set; }
    }
}
```

Connection string

- במאפיינים של בסיס הנתונים נמצא את מחרוזת ההתחברות:

Data Source=(LocalDB)\MSSQLLocalDB;Initial Catalog=MVCDB;Integrated Security=True;Connect Timeout=30;~~Encrypt=False;Trust Server Certificate=False;Application Intent=ReadWrite;Multi Subnet Failover=False~~

- יש למחוק את המאפיינים המסומנים ולהעתיק לקוד במקום המתאים.

בניית ה Action ב Controller

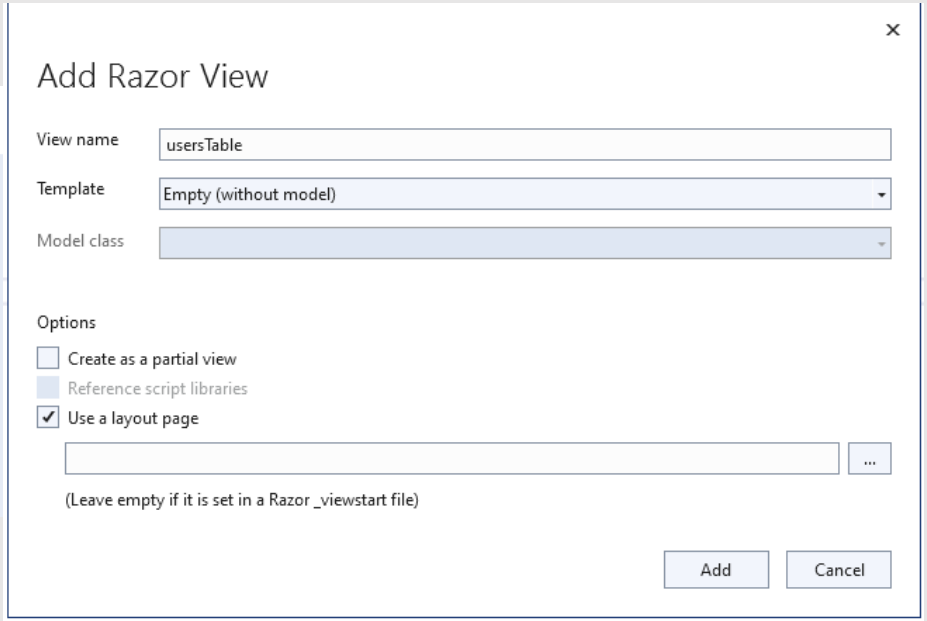
- נוסיף ל Controller שלנו פעולה שנקראת UsersTable.
- הפעולה תשלוף את טבלת המשתנים ותעביר אותה ל view

```
public IActionResult UsersTable()
{
    string SQL = "SELECT * FROM Users";
    DataSet ds = new DataSet();
    ds = Helper.RetrieveTable(SQL);
    return View(ds.Tables[0]);
}
```

- קליק ימני על שם הפעולה וניצור view מתאים.

הוספת view

- קליק ימני על ספריה Home.
- בחירה ב add ובחירה ב ...view



View usersTable

```
@model System.Data.DataTable;
@using System.Data;
@{
    ViewData["Title"] = "UsersTable";
}

<h1>UsersTable</h1>
<table class='table'>
    <thead>
        <tr>
            @foreach (DataColumn column in Model.Columns)
            {
                <th>@column.ColumnName</th>
            }
        </tr>
    </thead>
    <tbody>
        @foreach (DataRow row in Model.Rows)
        {
            <tr>
                @foreach (DataColumn column in Model.Columns)
                {
                    <td>@row[column]</td>
                }
            </tr>
        }
    </tbody>
</table>
```

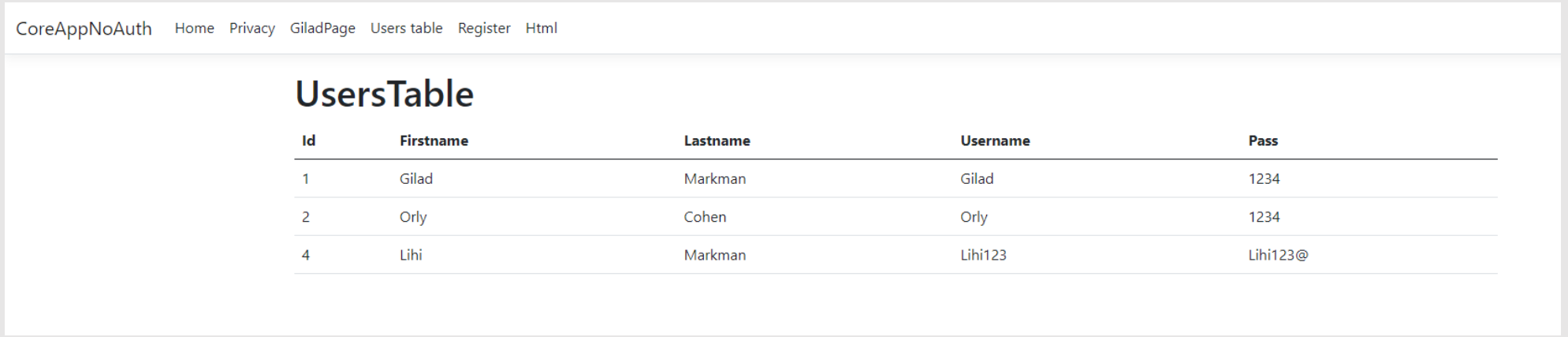
- בראש הדף נוסיף את המשתנה model שיקבל את הנתונים שהועברו מהקונטרולר.
- שימו לב כי כל שורה שמתחילה ב @ היא למעשה שורת C# הרצה בצד השרת.
- נוסיף את הקוד שמדפיס את הטבלה.

עדכון התפריט הראשי

- נוסיף קישור לדף בתפריט הראשי הנמצא ב Shared/_Layout.cshtml

```
<li class="nav-item">
    <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="UsersTable">Users table</a>
</li>
```

- נריץ את האפליקציה !!!!



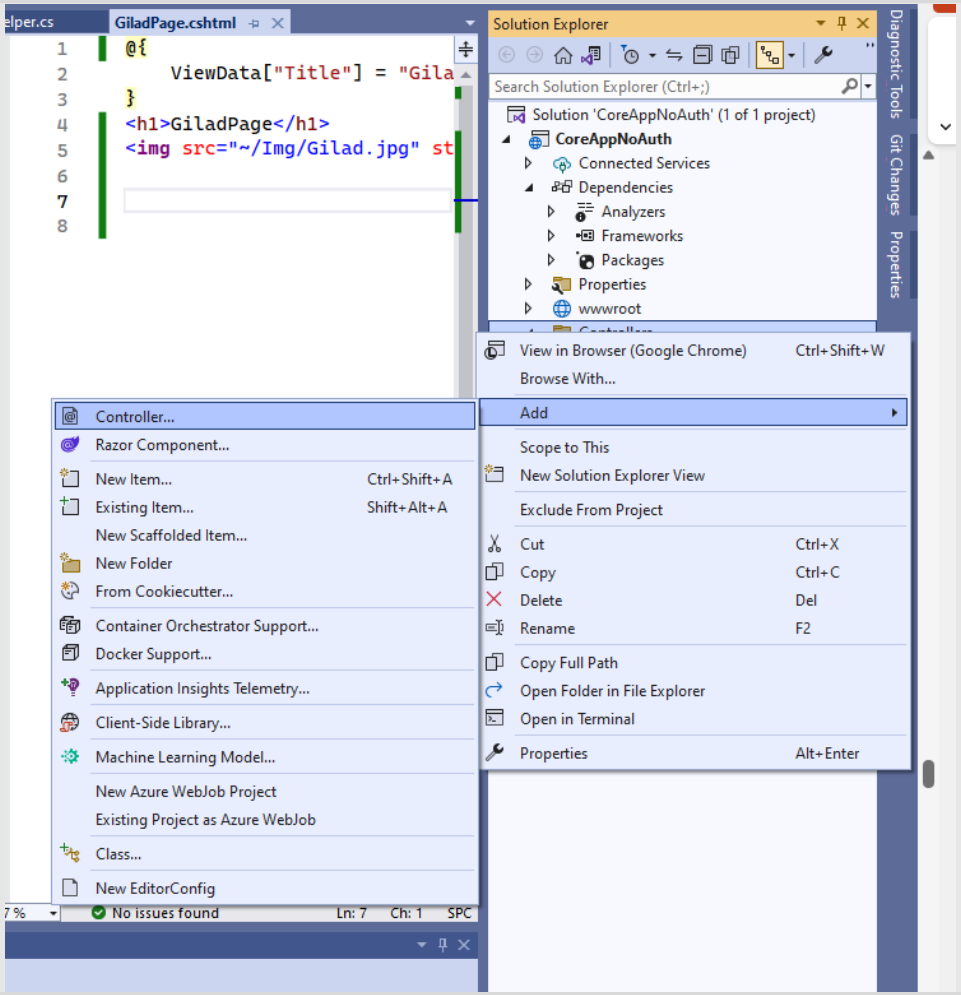
Register

הוספת משתמש

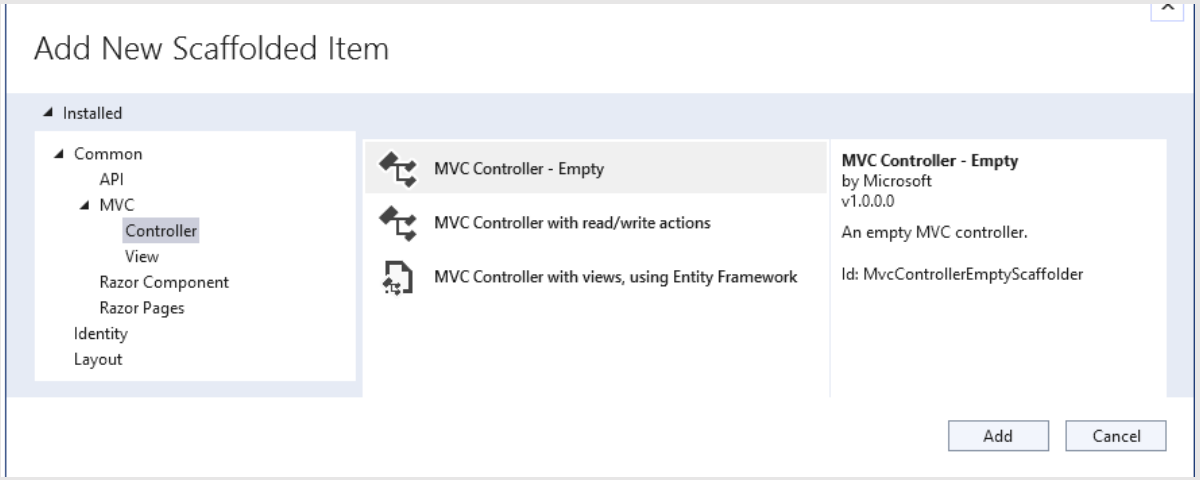
- הוספת משתמש חדש תעשה בשלבים הבאים:
  - יצירת קונטרולר חדש בשם UsersController.
  - יצירת הפעולה המפנה לדף ההרשמה.
  - יצירת הפעולה המבצעת את העדכון מול בסיס הנתונים.
  - העברה אל דף ההצלחה.

- נלמד כיצד להעביר נתונים מהקונטרולר ל view ולהפך.

הוספת קונטרולר



- נוסיף קונטרולר בשם Users.
- קליק ימני על ספריה Controllers.
- נבחר Empty וניתן לו שם



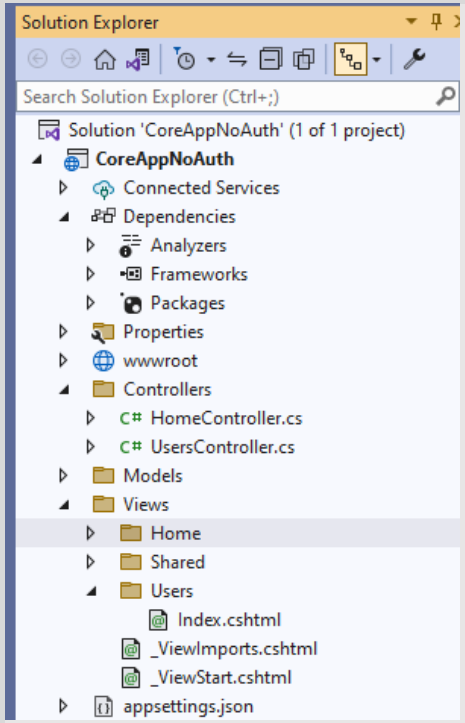
נעדכן את הקונטרולר

```
public class UsersController : Controller
{
    public IActionResult Index()
    {
        User user = new User();
        user.id = 6;
        user.Username = "test";
        user.Pass = "testPass";
        user.Firstname = "first";
        user.Lastname = "last";
        return View(user);
    }

    [HttpGet]
    public IActionResult Register()
    {
        return View();
    }

    [HttpPost]
    public IActionResult Insert(User user)
    {
        int n = Helper.Insert(user);
        return Redirect("../Home");
    }
}
```

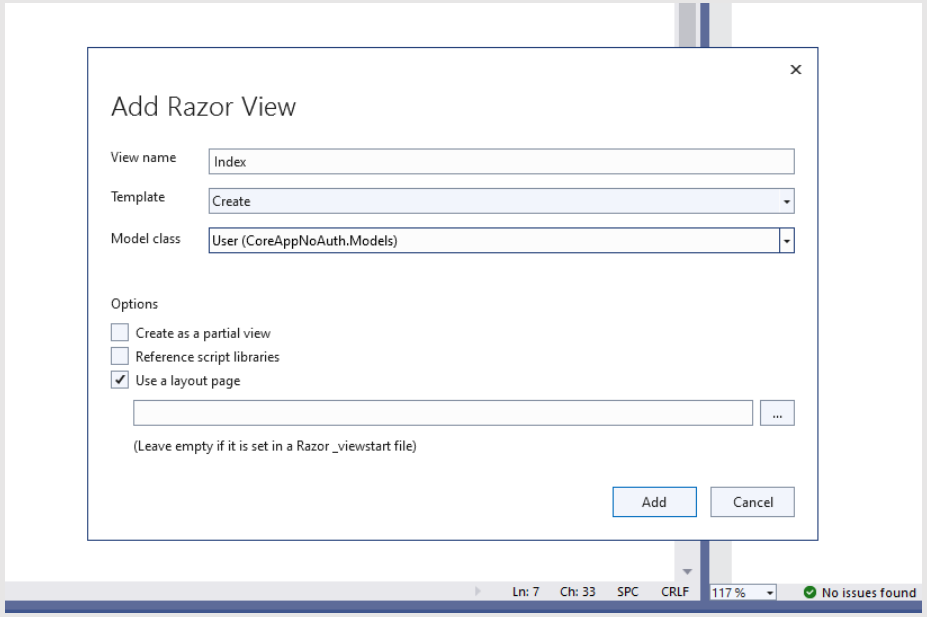
- נוסיף את הפעולות הבאות:
  - פעולה index רק לתרגול. מציגה משתמש בדף.
  - פעולה register מציגה דף הרשמה.
  - פעולה Insert מבצעת את עדכון בסיס הנתונים ולאחר מכן מחזירה לדף הרלוונטי.

- חשוב – פעולה Insert מקבלת אובייקט מסוג User. כלומר מוחזר מידע מהדף.

דוגמה – הוספת דף index

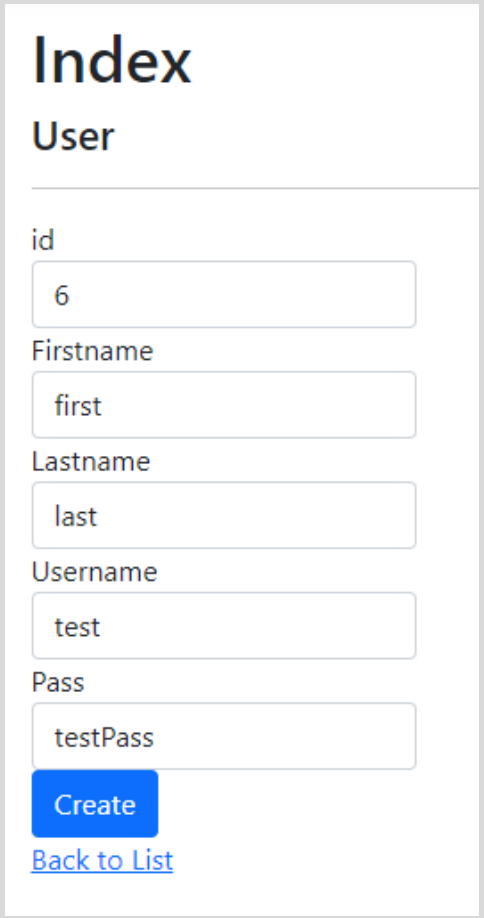


- קליק ימני, הוסף view, בחירת razor view.
- תבנית נבחר Create.
- מודל נבחר User.



- נקבל דף מוכן להוספת משתמשים.

יצירת view המבוסס על model




```
@model CoreAppNoAuth.Models.User

@{
    ViewData["Title"] = "Index";
}

<h1>Index</h1>

<h4>User</h4>
<hr />
<div class="row">
    <div class="col-md-4">
        <form asp-action="Index">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <div class="form-group">
                <label asp-for="id" class="control-label"></label>
                <input asp-for="id" class="form-control" />
                <span asp-validation-for="id" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Firstname" class="control-label"></label>
                <input asp-for="Firstname" class="form-control" />
                <span asp-validation-for="Firstname" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Lastname" class="control-label"></label>
                <input asp-for="Lastname" class="form-control" />
                <span asp-validation-for="Lastname" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Username" class="control-label"></label>
                <input asp-for="Username" class="form-control" />
                <span asp-validation-for="Username" class="text-danger"></span>
            </div>
```

- פניה לדף באמצעות URL
- הנתונים מוצגים בדף.

```
public IActionResult Index()
{
    User user = new User();
    user.id = 6;
    user.Username = "test";
    user.Pass = "testPass";
    user.Firstname = "first";
    user.Lastname = "last";
    return View(user);
}
```

יצירת Register view

```
@{
    ViewData["Title"] = "Register";
}

<h1>Register</h1>

<h4>User</h4>
<hr />
<div class="row">
    <div class="col-md-4">
        <form asp-action="Insert">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <div class="form-group">
                <label for="id" class="control-label">id</label>
                <input name="id" class="form-control" />
            </div>
            <div class="form-group">
                <label for="Firstname" class="control-label">Firstname</label>
                <input name="Firstname" class="form-control" />
            </div>
            <div class="form-group">
                <label for="Lastname" class="control-label">Lastname</label>
                <input name="Lastname" class="form-control" />
            </div>
            <div class="form-group">
                <label for="Username" class="control-label">Username</label>
                <input name="Username" class="form-control" />
            </div>
            <div class="form-group">
                <label for="Pass" class="control-label">password</label>
                <input name="Pass" class="form-control" />
            </div>
            <div class="form-group">
                <input type="submit" value="Save" class="btn btn-primary" />
            </div>
        </form>
    </div>
</div>
```

- נוסיף את הדף באותה צורה (עם תבנית Create).
- נשנה את הדף ונמחק דברים מיותרים. הדף לא צריך לקבל נתונים.
- נשים לב שיש לשנות את התגית של ה form כך שהטופס ישלח לפעולה Insert ולא ל Register.

קבלת נתונים מהמשתמש

- הוספת user לפונקציה מאפשרת להעביר את הנתונים מן הטופס לצד השרת.
- יש להקפיד כי הנתונים יהיו במבנה שיתאים לאובייקט user. אחרת נקבל שגיאה.
- כלומר לכל שדה בטופס חייב להיות מאפיין באובייקט בקונטרולר.

```
[HttpPost]
public IActionResult Insert(User user)
{
    int n = Helper.Insert(user);
    return Redirect("../Home");
}
```

הסבר הדף שיצרנו

```
[HttpGet]
public IActionResult Register()
{
    return View();
}

[HttpPost]
public IActionResult Insert(User user)
{
    int n = Helper.Insert(user);
    return Redirect("../Home");
}
```

```
<form asp-action="Insert">
    <div asp-validation-summar
    <div class="form-group">
        <label for="id" class=
        <input name="id" class
    </div>
```

- יצרנו בקונטרולר פעולה Register המציגה דף ריק.
- הגדרנו את ה action של הדף ל פעולה Insert של הקונטרולר (לא Register).
- הפעולה Insert מקבלת אובייקט User שאליו מוכנים הנתונים מהדף.
- הפעולה מעדכן בבסיס הנתונים את ה User ומעבירה לדף אחר.

תודה רבה

גלעד מרקמן