

תכנות באינטרנט

ASP.Net Core

MVC

גלעד מרקמן

קריית החינוך פארק המדע,

נס ציונה

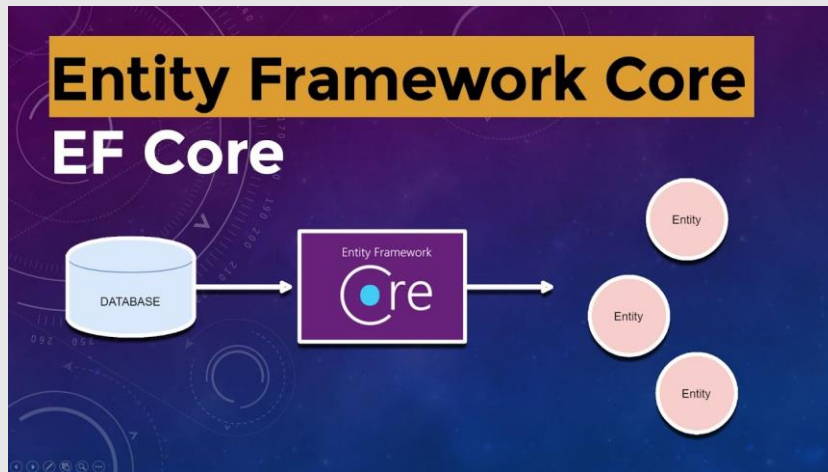


ASP.NET Core



# ASP.Net Core MVC – Entity Framework

- ORM-Object Relation Mapper של מיקרוסופט.
- Code First – הטכנולוגיה בונה את בסיס הנתונים בהתאם למחלקות (המודלים) שנבנים בקוד.
- Data First – הטכנולוגיה בונה את המחלקות (המודלים) בהתאם לבסיס נתונים קיים.



# אימות משתמשים

<https://www.youtube.com/watch?v=HSP6EZ5i9RY&t=57s>

<https://www.youtube.com/watch?v=I-ZzFLruiuo>

<https://www.youtube.com/watch?v=MxZ6PYARGJ4&t=10s>

<https://www.youtube.com/watch?v=fCS5LLOs3Qg>

- נפתח פרויקט חדש של ASP.Net Core MVC ונסמן את שיטת האימות כ Individual Accounts.

Additional information

ASP.NET Core Web App (Model-View-Controller) C# Linux macOS Windows

Framework ⓘ

.NET 6.0 (Long Term Support)

Authentication type ⓘ

Individual Accounts

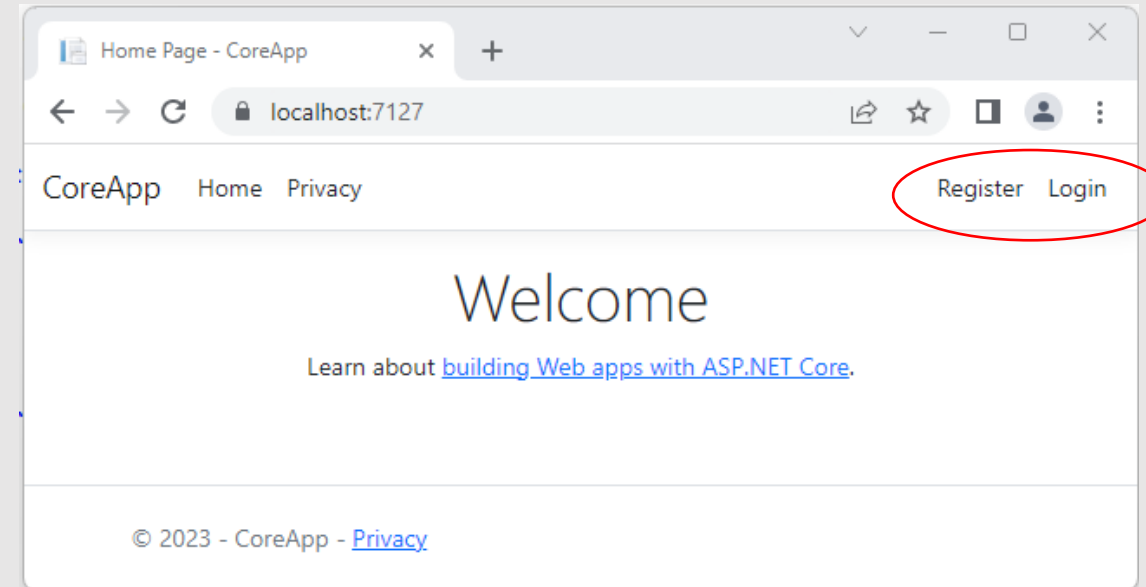
Configure for HTTPS ⓘ

Enable Docker ⓘ

Docker OS ⓘ

Linux

Do not use top-level statements ⓘ



- על מנת לבנות את המודל ובסיס הנתונים נפתח את  
Tools/ NuGet Package Manager/Package Manager Console •
- נכתוב את הפקודות הבאות:
- add-migration "AuthInit"
- Update-database
- ויצרנו מערכת אימות והרשמה לפרויקט שלנו, כולל בסיס נתונים.

## Log in

Use a local account to log in.

 Remember me?[Forgot your password?](#)[Register as a new user](#)[Resend email confirmation](#)

## Register

Create a new account.

aspnet-EFTutorial-644107c7-496a-49eb-87ef-d539f5d9a869

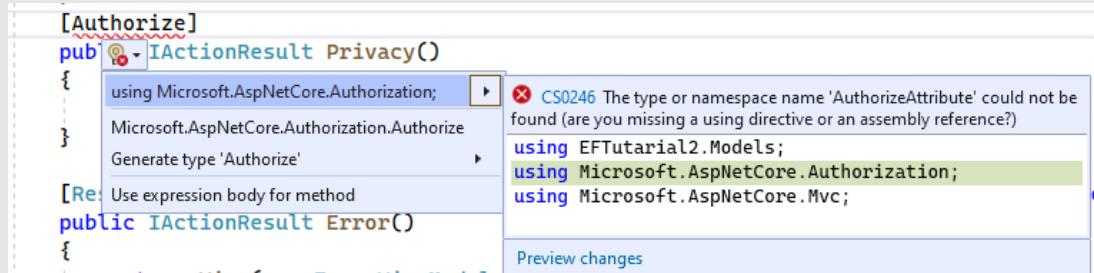
Tables

- ▶ System Tables
- ▶ External Tables
- ▶ dbo.\_EFMigrationsHistory
- ▶ dbo.AspNetRoleClaims
- ▶ dbo.AspNetRoles
- ▶ dbo.AspNetUserClaims
- ▶ dbo.AspNetUserLogins
- ▶ dbo.AspNetUserRoles
- ▶ dbo.AspNetUsers
- ▶ dbo.AspNetUserTokens

- במידה ונרצה להגביל כניסה לדף רק למי שעבר הרשמה נוסיף [Authorize] ב controller.

```
[Authorize]
public IActionResult Privacy()
{
    return View();
}
```

- יש להוסיף את המחלקות של אימות באמצעות using.



```
using Microsoft.AspNetCore.Authorization;
```

- כניסה לא מורשית לדף תעביר את המשתמש לדף הכניסה.



# בניית בסיס נתונים Entity Framework code first

<https://www.youtube.com/watch?v=SryQxUeChMc&t=7s>

- בניית המודלים (המחלקות) Student, Grade:
- יחס של אחד לרבים בין הטבלאות

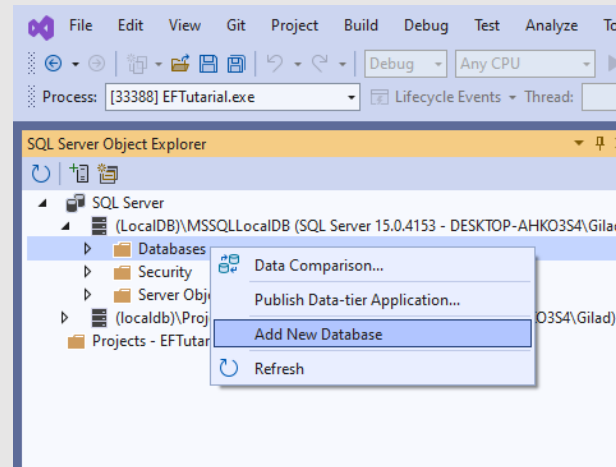
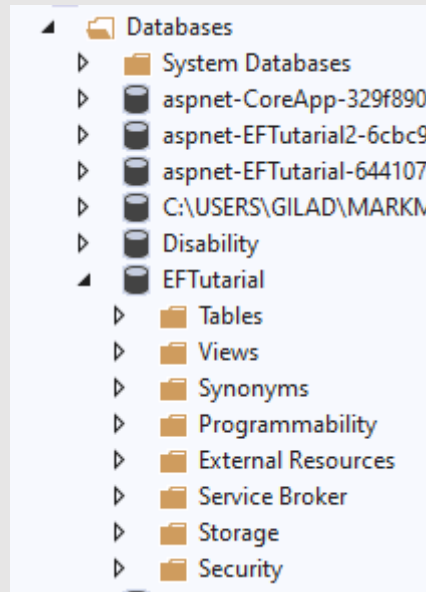
```
public class Student
{
    [Key]
    public int Id { get; set; }
    [Required]
    public string? Firstname { get; set; }
    [Required]
    public string? Lastname { get; set; }
    [Required]
    public string Class { get; set; } = null!;
    public int ClassNum { get; set; }
    public DateTime Birtday { get; set; }
    public ICollection<Grade>? Grades { get; set; }

    public string Fullname ()
    {
        return Lastname + " " + Firstname;
    }
}
```

```
public class Grade
{
    [Key]
    public int Id { get; set; }
    [Required]
    public string? Subject { get; set; }
    [Required]
    public int numGrade { get; set; }
    [Required]
    public string? Semester { get; set; }
    [Required]
    public int StudentId { get; set; }
    [Required]
    public Student? Student { get; set; }
}
```

• נוסף בסיס נתונים באמצעות SQL Server Object Explorer.

• קליק ימני על Databases:



• אין צורך לבנות טבלאות.

• ניתן להשתמש גם בבסיס נתונים שנבנה לצורך האימות. אז לא יהיה צורך ליצור בסיס נתונים חדש.

# בניית ה Context



- בספריית Data נבנה מחלקה שתגדיר את בסיס הנתונים ותשמש לנו לצורך הפניה לבסיס הנתונים.
- המחלקה יורשת מ DbContext.
- שני המאפיינים מגדירים טבלאות על בסיס המודלים שהגדרנו.
- הוספנו בנאי אשר מגדיר את סוג בסיס הנתונים באמצעות Dependency Injection.

```
public class StudentsDbContext : DbContext
{
    public DbSet<Student> Students { get; set; } = null!;
    public DbSet<Grade> Grades { get; set; } = null!;

    public StudentsDbContext() {}

    public StudentsDbContext(DbContextOptions<StudentsDbContext> options) : base(options) {}
}
```

## • במחלקה Program נוסיף את ההגדרות הבאות:

```
// Add services to the container.  
var connectionString = builder.Configuration.GetConnectionString("DefaultConnection");  
var connString = @"Data Source=(LocalDB)\MSSQLLocalDB;Initial Catalog=EFTutorial;Integrated Security=True;Connect Timeout=30;";  
builder.Services.AddDbContext<ApplicationDbContext>(options =>  
    options.UseSqlServer(connectionString));  
builder.Services.AddDbContext<StudentsDbContext>(options =>  
    options.UseSqlServer(connString));  
  
builder.Services.AddDatabaseDeveloperPageExceptionFilter();
```

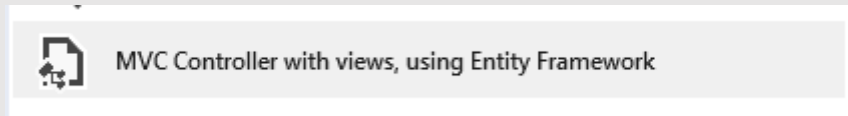
- את ה Connection string העתקנו מהמאפיינים של בסיס הנתונים, לאחר שמחקנו את הפרטים המיותרים. ניתן להשתמש בבסיס הנתונים הקיים (ובמחרוזת ההתקשרות שלו)
- אנו מגדירים את בסיס הנתונים כ sqlServer ו"מזריקים" את ההגדרה ל context.

# בניית בסיס הנתונים - Migration



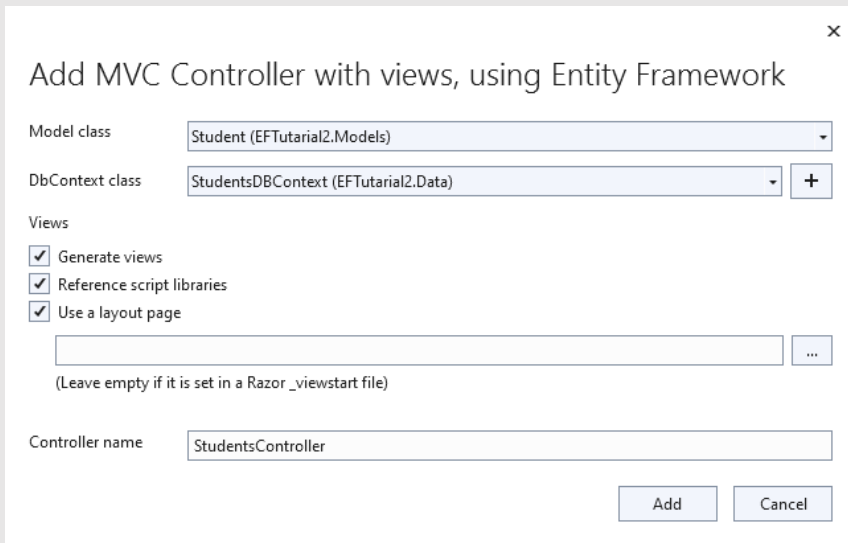
- נבנה את בסיס הנתונים באמצעות EF Migration.
- נפתח את ה Package Manager Console ונכתוב:
  - add-migration "StudentM1" -Context "StudentsDBContext"
  - Update-database –Context "StudentsDBContext"
- הטבלאות בבסיס הנתונים נוצרו.

- נבנה קונטרולר עבור המחלקה Student באמצעות קליק ימני על הספרייה. Add/Controller.



- יש לבחור את האפשרות עם EF.

- לבחור את המודל וה DbContext.



Add MVC Controller with views, using Entity Framework

Model class: Student (EFTutorial2.Models)

DbContext class: StudentsDbContext (EFTutorial2.Data)

Views

- Generate views
- Reference script libraries
- Use a layout page

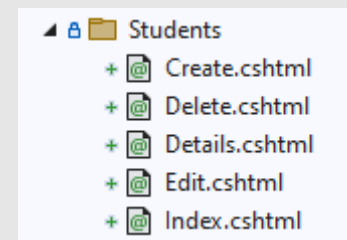
Controller name: StudentsController

Buttons: Add, Cancel

- המערכת בונה עבורנו

- הקונטרולר

- תצוגות



- נוסף קישור ל Student/Index ב- \_layout.

```
<li class="nav-item">  
  <a class="nav-link text-dark" asp-area="" asp-controller="Students" asp-action="Index">Students</a>  
</li>
```

- קיבלנו מערכת CRUD שלמה עם מסכים:

## Index

[Create New](#)

Firstname	Lastname	Class	ClassNum	Birthday	
Gilad	Markman	י	1	25/01/1968 12:00:00	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Orly	Markman	יא	2	06/08/1969 12:10:00	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
הילה	לוי	יב	2	12/12/2010 12:00:00	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>



# Grades Controller



- נבנה בדרך דומה GradesController.
- נבחר את המודל Grades
- נבחר את StudentDbContext (שייך לשני המודלים).
- נוסיף קישור גם ל Grades

```
<li class="nav-item">  
  <a class="nav-link text-dark" asp-area="" asp-controller="Grades" asp-action="Index">Grades</a>  
</li>
```

- נשים לב שלכל ציון המערכת מציגה את הסטונדנט המקושר אליו. אולם, היא מציגה את הכיתה ולא השם.
- נשנה את השורה ונשתמש בפעולה FullName() שיצרנו.

```
<td>  
    @*@Html.DisplayFor(modelItem => item.Student.Class)*@  
    @item.Student.FullName()  
</td>
```

שורה בהערה / מחיקה

- שינוי דומה נבצע ב Details.cshtml.

```
<dd class = "col-sm-10">  
    @*@Html.DisplayFor(model => model.Student.Class)*@  
    @Model.Student.FullName()  
</dd>
```

- מסך יצירה ועדכון מקבל רשימה של סטודנטים ומציג אותה בתיבת בחירה. כדי לעדכן את השדה שיוצג יש לשנות את ה controller.

```
public IActionResult Create()  
{  
    ViewData["StudentId"] = new SelectList(_context.Students, "Id", "Lastname");  
    return View();  
}
```

- בצורה דומה יש לשנות את הפעולה Edit.

## Details

### Grade

<b>Subject</b>	English
<b>numGrade</b>	0
<b>Semester</b>	2020A
<b>Student</b>	Markman Gilad

[Edit](#) | [Back to List](#)

## Edit

### Grade

Subject

numGrade

Semester

StudentId

רשימת הסטודנטים

# Razor Pages

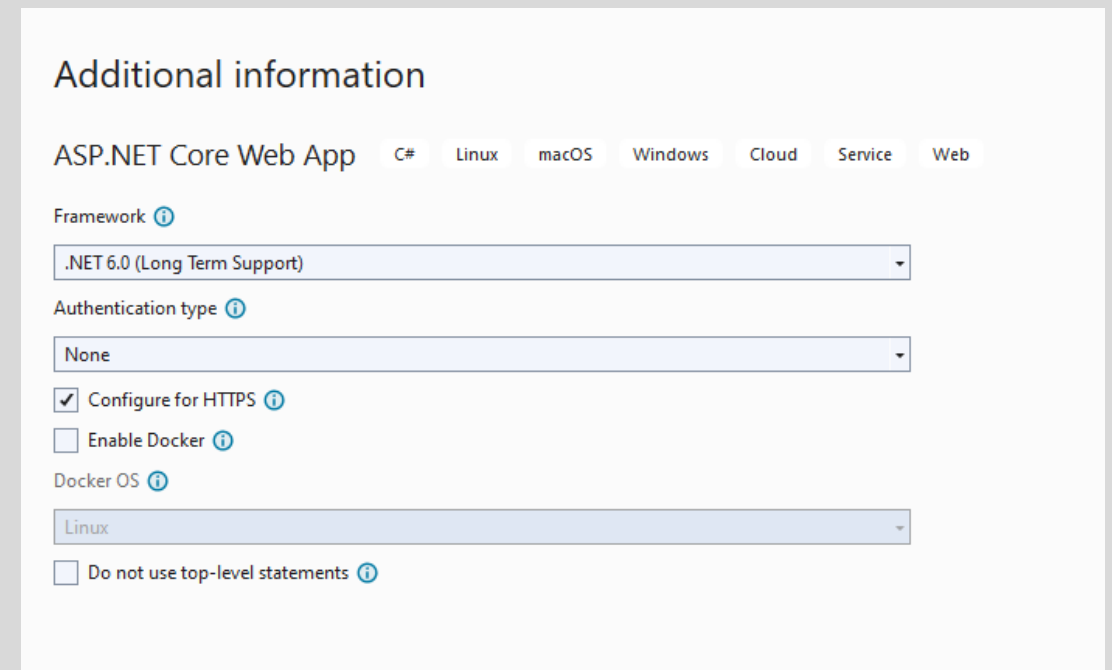
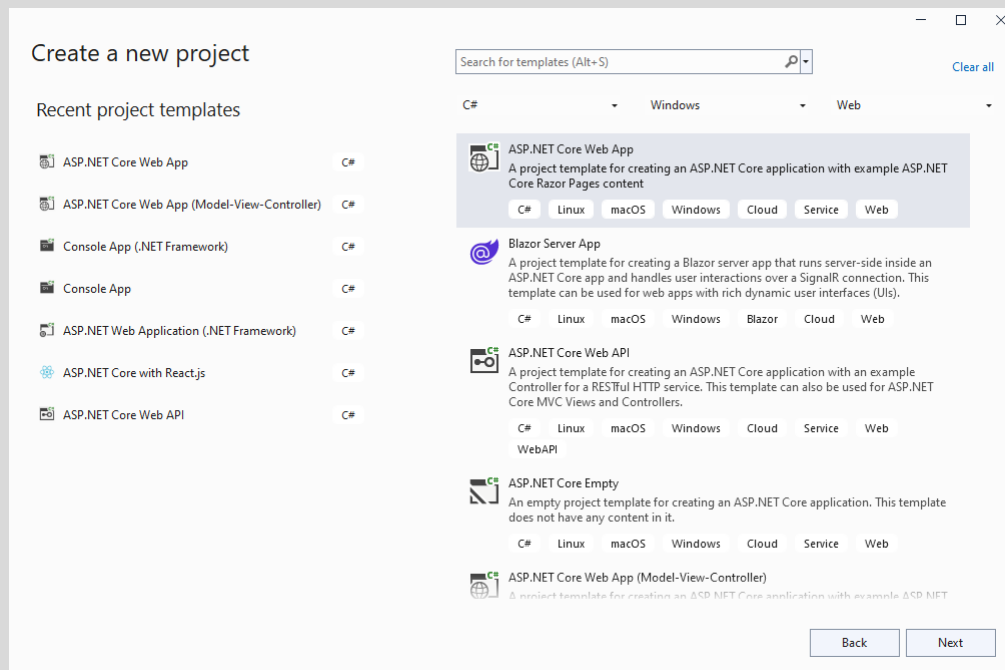
<https://www.youtube.com/watch?v=6GRFDkeCv3k&t=42s>

# Razor Paged vs MVC

- Razor Pages דומים ל web forms כך וכוללים את הקוד מאחורי הדף.
- שימוש ב Razor Pages אינו מצריך לעשות שימוש ב controller כיוון שהקוד מאחור עושה את העבודה הזו.
- בפרויקט MVC אנחנו משתמשים ב Razor view ואילו כאן אנחנו נשתמש ב Razor Page, שכולל את ה view ובנוסף את הקוד מאחורי הדף.

# יצירת פרויקט חדש

- ביצירת פרויקט חדש נבחר ASP.Net Core Web App.
- נקבע שם לפרויקט ומקום לשמור אותו.
- לאחר מכן נבחר פרטים נוספים (כגון: האם להוסיף Authentication)









קריית החינוך  
השש שנתית  
פארק המדע  
עתי של חוגשת



קריית החינוך  
השש שנתית  
פארק המדע  
עתי של חוגשת





**תודה רבה**



קריית החינוך  
השש שנתית  
פארק המדע  
עתי של חוגשת



קריית החינוך  
השש שנתית  
פארק המדע  
עתי של חושבת



קריית החינוך  
השש שנתית  
פארק המדע  
עתי של חוגשת



קריית החינוך  
השש שנתית  
פארק המדע  
עתי של חוגשת