ASP.NET CORE PROJECT (ΘΕΩΡΙΑ)

* ΕΓΚΑΤΑΣΤΑΣΗ ΠΕΡΙΒΑΛΛΟΝΤΟΣ

A) ΕΓΚΑΤΑΣΤΑΣΗ .NET SDK

ΓΙΑ ΝΑ ΚΑΝΩ ΕΛΕΓΧΟ ΑΝ ΟΝΤΩΣ ΤΟ ΕΓΚΑΤΕΣΤΗΣΑ ΣΤΟ CMD ΓΡΑΦΩ : dotnet -- version

Β) ΕΓΚΑΤΑΣΤΑΣΗ NODE.JS

ΓΙΑ ΝΑ ΚΑΝΩ ΕΛΕΓΧΟ ΑΝ ΟΝΤΩΣ ΤΟ ΕΓΚΑΤΕΣΤΗΣΑ ΣΤΟ CMD ΓΡΑΦΩ : node -- version

(ΑΠΟ ΤΗ ΣΤΙΓΜΗ ΠΟΥ ΤΟ ΚΑΝΩ INSTALL ΘΑ ΕΧΩ ΚΑΙ ΤΟ NPM

ΓΙΑ ΝΑ ΚΑΝΩ ΕΛΕΓΧΟ ΑΝ ΟΝΤΩΣ ΤΟ ΕΓΚΑΤΕΣΤΗΣΑ ΣΤΟ CMD ΓΡΑΦΩ : npm -- version)

Γ) ΕΓΚΑΤΑΣΤΑΣΗ EDITOR (VISUAL STUDIO CODE)

Δ) ΕΓΚΑΤΑΣΤΑΣΗ POSTMAN

E) ΕΓΚΑΤΑΣΤΑΣΗ DB BROWSER (SQL LITE)

* DOTNET ΕΝΤΟΛΕΣ (CMD)
* Dotnet -h (information για τα διαφορα command)

2) dotnet -new -h

3) dotnet new webapi (Για να φτιαξω ενα νεο web api)

4) Θελω σε ενα φακελο DatingApp να φτιαξω ενα φακελο DatingApp.API που θα εχει μεσα το project DatingApp.API

a) mkdir DatingApp.API

b) cd DatingApp.API

c) dotnet new -o DatingApp.API -n DatingApp.API

d) Για να τρεξουμε ενα web api dotnet run

3)Extensions που πρεπει να κανω Install

Πρωτα απο ολα παταω το extension tab (τελευταιο εικονιδιο)

1)c#

2)Nuget Package Manager

4)Entity Framework (Code First)

1) Πρωτα δημιουργουμε το model , φτιαχνουμε ενα φακελο models και μεσα μια c# class με το καθε μοντελο πχ:

namespace DatingApp*.*API*.*Models

{

*public* class Value

{

*public* int Id { get; set; }

*public* string Name { get; set; }

}

}

2)Δημιουργουμε ενα φακελο data και εν συνδεχια μια c# class DataContext η οποια θα κενει Inherit απο dbcontext

using Microsoft.EntityFrameworkCore;

namespace DatingApp*.*API*.*Data

{

*public* class DataContext : DbContext

{

}

}

3)Και δημιουργουμε σαν properties τα tables

*public* class DataContext : DbContext

{

*public* *DataContext* (DbContextOptions<DataContext> options) : base(options)

{

}

*public* DbSet<Value> Values {get;set;}

}

4)Στο startup class

Services.AddDbContext<DataContext>(x=>x.UseSqlite());

5)Εντολη για migration

Εστω οτι θελω να δημιουργησω migration με ονομα InitialCreate:

Dotnet ef migrations add InitialCreate

6)Δημιουργια Βασης Δεδομενων Μετα Το Migration

Dotnet ef update database

7) Install Angular cli ekdosh 6.0.8

Npm install -g @angular/cli@6.0.8

8) Δημιουργια Custom Login system

A)Πρωτα δημιουργουμε το model User:

*public* class User

{

*public* int Id { get; set; }

*public* string Username { get; set; }

*public* byte[] PasswordHash { get; set;}

*public* byte[] PasswordSalt { get; set; }

}

Β)Δημιoυργω το migration

Dotnet ef migrations add AddedUserEntity

Γ)Δημιουργω τη βαση δεδομενων

Dotnet ef update database

Δ)Δημιουργω το repository interface

using System.Threading.Tasks;

using DatingApp.API.Models;

namespace DatingApp*.*API*.*Data

{

*public* interface IAuthRepository

{

Task<User> *Register*(User user,string password);

Task<User> *Login*(string Username,string password);

Task<bool> *UserExists*(string username);

}

}

koita kwdika......