**FormulaOne Project – ASP.NET MVC .NET Core**

*Készítette: Kozák Péter*

# **Feladat**

**.NET webalkalmazás készítése**

Készíts egy vagy [ASP.NET](http://asp.net/) Core webalkalmazást Visual Studio és C# nyelv használatával. A webalkalmazás célja, hogy Forma 1-es csapatokat lehessen listázni, létrehozni, módosítani és törölni. A létrehozás, módosítás és törlés csak bejelentkezés után lehetséges. A Forma-1-es csapat mint entitás az alábbi tulajdonságokkal rendelkezik: Név, Alapítás éve, Megnyert világbajnokságok száma, Befizette-e a nevezési díjat.

* Tervezz meg és hozz létre egy adatbázist a leírt entitás alapján Entity Framework Core Code First megoldás alkalmazásával, melyhez providerként az SQLite InMemory mode-ját konfiguráld be!
* Rétegezd megfelelően az alkalmazást, ahogyan valós körülmények között is tennéd! Ügyelj a kód újrafelhasználhatóságára, kódszervezésre, formázásra!
* Implementáld az adatelérési réteget LINQ műveletek segítségével.
* Tervezd meg és hozd létre az alábbi felületeket: Listázás, Létrehozás, Módosítás. Ne feledkezz meg az adatok validálásáról és a törlés funkcióról sem. A felületek létrehozásakor használhatod a Bootstrap keretrendszert.
* Konfiguráld be, hogy a létrehozás, módosítás és törlés műveletek csak bejelentkezett felhasználók számára legyenek elérhetők. Használd az [ASP.NET](http://asp.net/) Identity Core megoldását.
* A bejelentkezett felhasználó neve legyen: admin, jelszava pedig: f1test2018

# **Az alkalmazás rétegezése (Elkészített API-k):**

* **FormulaOneProject**

Tartalmazza a futtatható állományokat.

Tartalmazza:

* + - Konfigurációs beállítások (***appsettings.json*** (Lásd:”***ConnectionString***”, „***DevLoginConfig***”)
    - A feladathoz szükséges konfigurációs beállításokat a ***Startup.cs*** ***ConfigureServices*** és ***Configure*** metódusban (***SqliteConnection, AddIdentity, AddDbContext, IdentityOptions, LoginConfigModel, UseAuthentication, DataBaseMigration***).
    - Az alkalmazáshoz szükséges Új ***Controllerek***.
    - Az alkalmazáshoz szükséges Új ***ViewModel***-ek.
    - Az alkalmazáshoz szükséges Új ***View***-ok.
* **Business.Engine**

Tartalmazza azokat az ***Osztályokat*** és ***Interfészeket***, amelyek megvalósítják az ***Üzleti Logikát***.

* **Business.Entities**

Tartalmazza ***az adatbázishoz szükséges Modelleket***.

* **Core.Deploy.ApplicationDeploy**

Tartalmazza azokat az ***Osztályokat***, amelyek megvalósítják, az ***adatbázis Migration***-jét, amikor az alkalmazás elindul.

* **Core.Handlers.BackEndExceptionHandler**

Tartalmazza az egységes ***BackEnd Exception***-ok kezelését (*Részletekért lásd az osztály dokumentációját*)

* **Core.Common**

Tartalmazza az alkalmazáshoz használt ***DTO*** ***objektumokat***.

Tartalmazza az alkalmazáshoz használt ***PropertyMapper***-t (*Részletekért lásd az osztály dokumentációját*)

* **Data.DataAccessLayer**

Tartalmazza az adatbázishoz definiált ***Context***-et.

Tartalmazza az adatbázis műveletekhez használt ***Repository*** ***Osztályokat*** és ***Interfészeket***.

Tartalmazza az ***EF*** ***CodeFirst*** által létrehozott *Migration*-t.

# **Az adatáramlás folyamata**

A feldolgozandó adathalmaz a ***View***-ról a ***Controller***-hez kerül, mint ***ViewModel*** ***objektum*** vagy ***objektum*** ***halmaz***. Ez az ***objektum*** vagy ***objektum*** ***halmaz***, a ***Controller***-en belül ***átmappelésre*** kerül egy ***DTO*** objektummá vagy objektum halmazzá.

A ***Controller-en*** belül a végrehajtási folyamat átadódik a ***Business*** ***Enginnek***, ahol az üzleti logika definiálva van. A ***Business*** ***Engine*** a ***Controller***-től a ***DTO*** objektumot kapja meg, így azzal fog dolgozni.

Az üzleti logika definiálása során, ha van adatbázis művelet végrehajtása, akkor az ***Engine***-n belül a ***DTO*** objektum ***átmappelődik*** egy ***Adatbázis*** ***Entitássá***, majd ezt az objektumot a megfelelő ***Repository*** megfelelő ***Metódusának*** adja át, amely a ***megfelelő*** ***adatbázis*** ***műveletet*** ***végrehajtja*** *(Tehát:* ***ViewModel*** *->* ***DTO*** *->* ***DB******Entitás****)*.

Az adatáramlás, például mikor egy adat lekérdezést hajtunk végre az adatbázisból, akkor is ugyan ilyen módszerrel valósul meg, csak fordított sorrendben (*Tehát:* ***DB******Entitás*** *->* ***DTO*** *->* ***ViewModel****)*.

# **Az alkalmazás használata röviden:**

Az alkalmazás elindulásakor a ***Team/Index*** oldalt látjuk. Ezen az oldalon megjelenik a kért adathalmaz egy listában, azonban a felhasználó nem tud ***Létrehozni***, ***Szerkeszteni***, ***Törölni***. Ahhoz, hogy ezt meg tudja valósítani szükséges bejelentkeznie.

Az alkalmazás indulásakor legelőször az ***Account/Register Method*** fut le, így az adatbázisba beszúrásra kerül, a feladatban meghatározott ***felhasználó***, azonban lehetőséget biztosítottam arra, hogy a fejlécen, ha a „***Regisztráció***” gombra kattintanánk akkor a felhasználó, ha még nincs benne az adatbázisban, akkor beszúrásra kerül.

A fejlécen található bejelentkezés gombra kattintva jelenik meg számunkra a ***Login*** felület. Az Input mezők kitöltésével (***a feladatban meghatározott adatokkal***), és a ***bejelentkezés*** gombra kattintva, megtörténik a ***bejelentkezés***. Bejelentkezés sikeressége után, átnavigálunk a ***Team/Index*** oldalra.

Bejelentkezés után, az oldal címkéjén most már megtalálható a ***bejelentkezett felhasználó neve***, illetve a további műveletek is elérhetőek, azaz tudunk új ***Team***-et ***létrehozni***, meglévőt ***szerkeszteni***, illetve ***törölni***.

Mikor az új ***Team*** ***létrehozására*** kattintunk, akkor átnavigálunk a ***Team/AddOrEdit View***-ra. Az input mezők kitöltésével és a ***Mentés*** gombra kattintva rögzíthetjük az új ***Team***. Ha mégsem szeretnénk elmenteni, akkor „***Vissza a listához***” gombra kattintva szakíthatjuk meg ezt. Figyelem! Mivel egy ***Team*** ***név*** az egyedi kell, hogy legyen, hiszen két ugyan olyan ***Team*** nem létezik, így csak egyedi ***Team*** ***nevű*** csapatot lehet létrehozni. Ez az ***INPUT*** mező ***TOOLTIP***-jében megjelenítésre is kerül. Továbbá az évszámok listája az ***1900***-as évektől kezdődik, és bezárólag az ***aktuális évvel*** történik.

Ha egy meglévő ***Team***-et szeretnénk szerkeszteni, akkor a korábbi eljárást kell alkalmazni. Annyi a különbség, hogy ebben az esetben a beviteli mezők kitöltésre kerülnek.

***Törlés*** esetén egy felugró ablakot kapunk, amely megerősítést kér a ***Team*** törléséhez. ***Igen*** esetén ***Törlődik*** csak a felhasználó az adatbázisból.