

# Autószerelő munkanyilvántartó

---

## Alapkövetelmények

- Git repo
  - Rendszeres commit minden csapattagtól (egyéni értékelés)
- Egy solution használata
  - Legalább .NET 6.0 használata
  - Legalább három külön projekt megléte a feladat követelményei alapján
    - ASP.NET Core Web API
    - Blazor WebAssembly App (2x)
- Kódolási konvenciók alkalmazása:
  - StyleCop.Analyzers nuget package használata az összes projektben  
<https://www.nuget.org/packages/StyleCop.Analyzers/>
- A solution buildelhető

**Bármely alapkövetelmény hiánya esetén a feladat értékelhetetlennek minősül! (Elégtelen érdemjegy)**

## Feladatléírás

A feladat egy autószerelő műhelyben üzemelő webes alkalmazás elkészítése.

Az alkalmazás három részből áll. Egy az irodai-ügyintéző irodájában működő felhasználói felület, ahol az új megrendeléseket, ügyfeladatokat lehet felvenni, illetve nyomonkövethető a már felvett javítások állapota. Egy másik webes felületen ami az autószerelő műhelyben működik, legyen megtekinthető minden felvett rendelés, amiknek a szerelő láthatja a részleteit és átállíthatja a munka adott állapotát. Az alkalmazás backend-jét egy SQL adatbázis alkotja, ebben tárolja az adatokat amit egy Web API applikáció endpoint-jain keresztül módosíthatunk és érhetünk el. Az API-n keresztül kell történjen minden adattal kapcsolatos művelet és lekérdezés.

- Munkaóra esztimáció:
  - Átlagos munkaóra kategóriákra lebontva
    - Karosszéria = 3 óra
    - Motor = 8 óra
    - Futómű = 6 óra
    - Fékberendezés = 4 óra
  - Az autó életkorából származó munkaóra súlyozás
    - 0-5 év: 0.5
    - 5-10 év: 1
    - 10-20 év: 1.5
    - 20- év: 2
  - Hiba súlyosságából származó munkaóra súlyozás
    - 1-2: 0.2
    - 3-4: 0.4
    - 5-7: 0.6
    - 8-9: 0.8
    - 10: 1

- Képlet: kategória \* kor \* hiba súlyossága

## Szerver - ASP.NET Core WEB API alkalmazás

Az adatbázisban tárolt adatok az API-n keresztül érhetők el. A Blazor kliensek az API végpontjait hívva érik el illetve módosítják az adatokat. A munkaidő esztimáció számítását külön endpoint-on érhető el.

- Adatbázis Entity Framework Core használatával (MSSQL LocalDB)
- CRUD műveletek implementálása a feladatnak megfelelően
- Endpointokon beérkező adatok validálása backend oldalon
- Munkaóra esztimáció implementálása
- UNIT tesztek írása a munkaóra esztimáció minden esetére

## Munka felvevő kliens - Blazor WebAssembly alkalmazás

Az irodai-ügyintéző ezen a felületen látja a korábban felvitt adatokat, az érkező ügyfelek megrendeléseit itt tudja felvenni. A felület csak és kizárólag megjelenítő funkciót lát el, logikai megoldásokat, számításokat nem végez.

- A irodai-ügyintéző látja a felvett munkákat
  - Az adatok táblázatos formában megjelenítve
  - Indításkor betölti a korábbi adatokat
  - Adatok frissítésére alkalmas gomb
  - A sorok a különböző kategóriák szerint rendezhetők növekvő vagy csökkenő sorrend (dátum, abc)
  - Keresés lehetőség az összes attribútum alapján. Validáció is szükséges az input mezőkre.
  - Munkaóra esztimáció megjelenítése (API számolja)
  - Egy kiválasztott munka adatait
    - Meg tudja nézni
    - Módosítani tudja
- Munkafelvételhez szükséges mezők:
  - Ügyfél neve
    - Validáció
      - Nem lehet: üres, whitespace, különleges karakterek szűrése pl !?\_~;#
  - Autó típusa és rendszáma
    - Validáció
      - Nem lehet: üres, whitespace, különleges karakterek szűrése pl !?\_~;#
      - Rendszám formátuma: XXX-000
  - Gyártási év
    - Validáció
      - Kötelező mező, csak számjegyek
  - Munka kategória
    - Karosszéria, motor, futómű, fékberendezés
    - Validáció
      - Kötelező mező
      - A felsorolt értékek közül az egyik

- Az autó hibájának rövid leírása
    - Validáció
      - Kötelező mező
  - A hiba súlyossága (1-10)
    - Validáció
      - Kötelező mező
      - 1 és 10 közötti érték
- UNIT tesztek írása a bevitt adatok validációjára

## Autószerelő kliens - Blazor WebAssembly alkalmazás

A dolgozó a műhelyben ezen a felületen kapja meg az irodában felvett megrendeléseket. A felületen adatmódosításokat nem tud elvégezni, az adott munka állapotának frissítésének a kivételével. Az állapot állítását ezt követően az irodai-ügyintéző nyomon tudja követni a saját felületén.

- Az adatok táblázatos formában megjelenítve
- Indításkor betölti a korábbi adatokat
- Adatok frissítésére alkalmas gomb
- A sorok a különböző kategóriák szerint rendezhetőek növekvő vagy csökkenő sorrendben (dátum, abc)
- Keresés lehetőség az összes attribútum alapján
- Munkaóra esztimáció megjelenítése (API számolja)
- Ki tud választani egy munkát
  - Aminek az állapotát változtathatja
    - **Felvett munka** -> **Elvégzés alatt** -> **Befejezett**