

Vizsgafeladat azonosító száma: 2505-A

.....
vizsgabizottsági tag

.....
vizsgázó neve

.....
vizsgabizottsági tag

.....
eredmény

.....
vizsgafelügyelő

PROJEKTFELADAT

Szakma azonosítószáma és megnevezése:

5 0613 12 03 – Szoftverfejlesztő és -tesztelő

A vizsgatevékenység megnevezése: **Szoftverfejlesztő és -tesztelő projektfeladat**

Asztali- és webes szoftverfejlesztés, adatbázis-kezelés vizsgarész

Időtartam: 240 perc

A vizsgatevékenység aránya a szakmai vizsgán belül: 90 %

Pótlapok száma	
----------------	--

Jóváhagyta:

Kovács István
vizsgaközpont vezető

2025

A feladatsor az Informatika és távközlés ágazat 5 0613 12 03 – Szoftverfejlesztő és -tesztelő szakma részére a szakképzésről szól törvény végrehajtásáról szóló 12/2020 (II.7.) Korm.

rendeletben foglaltak alapján, a 2023.11.21-én kiadott Képzési és Kimeneti Követelményeknek megfelelően készült.

Fontos tudnivalók

A vizsgázó a feladatlagra írja fel a nevét!

Ha a vizsgafeladat kidolgozásához a vizsgaközpont által biztosított pótlapot is használ, a nevét valamennyi pótlapra írja fel, és sorszámozza is azokat!

Sikeres megoldást és jó munkát kívánunk!

Használható segédeszköz:

—

Értékelési skála külön erre a feladatrészre nem releváns.

A megszerezhető maximális pontszám: 65 pont.

A vizsgarész megfelelt, ha a maximális pontszám 40%-át (26 pont) elérte.

Figyelmesen olvassa el a következőket!

- Munkáját folyamatosan mentse!
- Ügyeljen arra, hogy a kiírások pontosan kövessék a megadott mintát!
- Az azonosítókat a nyelvi konvencióknak megfelelően nevezze el, és törekedjen a Clean Code elvek betartására!
- A rendelkezésre bocsátott forrásállományok helyes szerkezetét nem kell ellenőriznie; feltételezheti, hogy az adatok a leírásnak megfelelnek. Minden kapott állomány UTF-8 kódolású.
- Megoldását úgy készítse el, hogy az azonos szerkezetű, de tetszőleges bemeneti adatok esetén is helyes eredményt adjon!
- Amennyiben a feladat nem ad egyértelmű útmutatást egy döntési helyzetre, saját belátása szerint járhat el.
- A forráskód védett részeit nem módosíthatja! Ezek jelölése:
`// ■ ■ ■ -> védett rész // <- ■ ■ ■`
- Sok időt spórolhat, ha a PDF formátumú feladatlapból másol ki szövegrészleteket.

I. Konzolos asztali alkalmazásfejlesztés C# nyelven 15 pont

Forrásállomány ismertetése

A forrásállomány minden sora Magyarország egy településének adatait, köztük a 2024-es és 2025-ös évekre vonatkozó népességi adatokat tartalmazza. A CSV állomány UTF-8 kódolású és [Source](#) mappában található!

lakossag_2024_2025.csv egy részlete:

```
id;vármegye;település;típus;férfi_2024;nő_2024;férfi_2025;nő_2025
17376;FEJ;Aba;város;2346;2276;2344;2284
12441;SZO;Abádszalók;város;2036;2108;1984;2086
12548;BAR;Abaliget;község;339;311;331;296
```

Magyarázat:

- **id** – Elsődleges kulcs
- **vármegye** – A település vármegyekódja
- **település** – A település neve
- **típus** – A település típusa lehet:
fővárosi terület; község; nagyközség; vármegye székhely; vármegyei jogú város; város
- **férfi_2024, nő_2024** –
A településen 2024. január 1-én bejelentett lakcímmel rendelkező férfiak és nők száma
- **férfi_2025, nő_2025** –
A településen 2025. január 1-én bejelentett lakcímmel rendelkező férfiak és nők száma

Osztályok ismertetése

A **Models** mappában található fájlok:

- **Settlement.cs** – Egy osztály, amely a településhez kapcsolódó adatokat tartalmazza.
- **AbstractSettlementManager.cs** – Egy absztrakt osztály, amely biztosítja a települési adatok beolvasását egy **List<Settlement>** listába, valamint több, erre épülő metódus fejléce is itt kerül meghatározásra.

1. feladat

3 pont

- Nyissa meg a kapott **ConNepesseseg** projektet, majd folytassa a **Settlement** osztály implementálását! Az osztály feladata egy település adatainak tárolása. Hozzon létre megfelelő adattagokat, de figyeljen arra, hogy a település adatait ne lehessen az osztályon kívül módosítani!
- Implementálja az előkészített konstruktort
- implementálja **TotalPopulation** metódust!
- Végezetül írja felül az örökölt **ToString()** metódust úgy, hogy a következő formátumban adjon vissza értéket!

Jánoshalma város lakossága 2024-ben **8480** fő, 2025-ben **8424** fő

```
public class Settlement
{
    // ■ ■ ■ ->
    /// <summary> A konstruktor egy település adatait tölti be egy CSV sorból.
    public Settlement(String sourceLine) // <- ■ ■ ■ ...

    // ■ ■ ■ ->
    /// <summary> A település teljes lakosságát adja vissza a megadott évben. Az év, ...
    public int TotalPopulation(int year) // <- ■ ■ ■ ...
}
```

2. feladat

4 pont

- Az **AbstractSettlementManager** absztrakt osztály **LoadFromCSV** metódusát alakítsa át úgy, hogy nem létező fájl esetén **FileNotFoundException** kivételt dobjon!
- Hozzon létre egy új osztályt **SettlementManager** néven a **Models** mappában! Az osztály implementálja a **AbstractSettlementManager** absztrakt osztályt!
- Implementálja a **SaveToCSV** metódust!
- Valósítsa meg a **RegionalStatistics** metódust! Az elkészítést meggyorsítja, ha az absztrakt osztálynál elhelyezett kódrészt kimásolja a megjegyzésből. A sikeres implementációt a 6. feladat mintája alapján ellenőrizheti. A metódus használatára is talál kódrészletet megjegyzésben.

```
/// <summary> A megadott települések adatait elmenti egy CSV fájlba. A CSV szerk ...
public abstract void SaveToCSV(string path, List<Settlement> settlementsList);
```

```
/// <summary> Visszaadja a megadott vármegye településeinek statisztikáit.
public abstract List<String> RegionalStatistics(string varmegye);
```

A következő feladatokat a Program.cs fájlban valósítsa meg! Ha nehézsége adódott egy osztály implementálásával, az alábbi feladatokat attól függetlenül is elkészítheti.

3. feladat

1 pont

Írassa képernyőre a Hajdú-Bihar vármegyei nagyközségek adatait! A vármegye kódja: **HAI**

Tipp: A kiírásnál a **Settlement** osztály **ToString()** metódusát használja!

3. feladat

Hajdú-Bihar vármegyében lévő nagyközségek listája:

```
Bagamér nagyközség lakossága 2024-ben 2513 fő, 2025-ben 2501 fő
Csökmő nagyközség lakossága 2024-ben 1869 fő, 2025-ben 1855 fő
Egyek nagyközség lakossága 2024-ben 5286 fő, 2025-ben 5247 fő
Földes nagyközség lakossága 2024-ben 3869 fő, 2025-ben 3869 fő
Hosszúpályi nagyközség lakossága 2024-ben 5864 fő, 2025-ben 5879 fő
Nagyrábé nagyközség lakossága 2024-ben 2016 fő, 2025-ben 2010 fő
Nyírábrány nagyközség lakossága 2024-ben 3948 fő, 2025-ben 3915 fő
Pocsaj nagyközség lakossága 2024-ben 2861 fő, 2025-ben 2856 fő
Sárrétudvari nagyközség lakossága 2024-ben 2857 fő, 2025-ben 2843 fő
Zsáka nagyközség lakossága 2024-ben 1551 fő, 2025-ben 1535 fő
```

4. feladat

1 pont

Listázza ki a 10 legnagyobb mértékű növekedést mutató település nevét és növekedésének %-os mértékét!

Tipp: Használja a **SettlementManager** osztály **GetTop10SettlementsByHighestGrowthRate** metódusát!

4. feladat

A 10 legnagyobb növekedési mértékkel rendelkező település:

```
Keresztéte : 100,0%
Kisvásárhely : 56,5%
Teresztenye : 37,5%
Lendvajakabfa : 30,6%
Szijártóháza : 23,1%
Várad : 21,2%
Tiszatardos : 18,8%
Gosztola : 18,8%
Szegi : 18,0%
Kára : 17,7%
```

5. feladat

2 pont

Hány település lakossága éri el a 10 000 főt 2025-ben vármegyénként? A választ írassa ki a képernyőre a vármegye kódja szerint növekvő sorrendben, az alábbi formátumban!

Megjegyzés: A főváros kerületeit ne vegye figyelembe a számolás során!

5. feladat

A 2025-ben 10 000 fő feletti lakosságszámú települések száma vármegyék szerint rendezve:

```
BAC : 9 db
BAR : 3 db
BEK : 6 db
BOR : 8 db
CSO : 5 db
FEJ : 6 db
GYO : 3 db
HAJ : 9 db
HEV : 4 db
KOM : 6 db
NOG : 3 db
PES : 45 db
SOM : 4 db
SZA : 6 db
SZO : 8 db
TOL : 5 db
VAS : 4 db
VES : 7 db
ZAL : 3 db
```

6. feladat

4 pont

- Kérje be a billentyűzetről egy vármegye kódját! Ha a megadott kód nem létezik, ismételje meg a bekérést, amíg érvényes kódot nem kap!
- Érvényes kód megadása után jelenítse meg a képernyőn a vármegye statisztikáit a korábban elkészített **RegionalStatistics** metódus felhasználásával!
- Írja ki a vármegye településeinek adatait CSV formátumban, a feladatsorban korábban megadott struktúrát követve! A fájl neve a vármegyekódja, kiterjesztése pedig ".CSV" legyen!

Tipp: Használja a **SettlementManager** osztály **SaveToCSV** metódusát!

6. feladat

```
Kérem a vármegye kódját! :HAJ
Vármegye: HAJ
Települések száma: 82
Összes lakos 2024-ben: 532227 fő
Összes lakos 2025-ben: 529761 fő
Növekedés: -2466 fő
Növekedés %-ban: -0,46%
```

HAJ.CSV fájl első 4 sora:

```
id;Vármegye;Település;Típus;Férfi_2024;Nő_2024;Férfi_2025;Nő_2025
27641;HAJ;Álmosd;község;893;927;903;941
03319;HAJ;Ártánd;község;386;322;394;323
20011;HAJ;Bagamér;nagyközség;1251;1262;1239;1262
```

II. Grafikus asztali alkalmazásfejlesztés C# nyelven

10 pont

Nyissa meg a **WPFNepesség** projektet, majd folytassa az alábbiak szerint!

A program futás közbeni képernyőképét megtalálja a 10. oldalon és a források mappában is.

1. feladat

1 pont

A 3 oszlopos 4 soros **GRID** rácsban helyezze át azokat a blokkokat, amelyek rossz helyen vannak! Alakítsa át az XAML leírófájlt úgy, hogy az alábbi mintának megfeleljen az elrendezés!

The screenshot shows a WPF application window titled "Magyarország településeinek név 1* sége". The interface is organized into a 3x4 grid. The first row contains a search bar labeled "Szűrés a vármegye kódjára:" and a dropdown menu. The second row contains three labels: "Települések száma: ■ db", "Lakosok száma 2024-ben: ■ fő", and "Lakosok száma 2025-ben: ■ fő". The third row contains a table with columns: "Település", "Népesség 2024-ben", "Népesség 2025-ben", "Változás", "Változás %-ban", and "Városnak számít?". The fourth row contains a panel titled "Kiválasztott település adatai" with labels for "Neve:", "Típusa:", "Népesség 2024-ben:", and "Népesség 2025-ben:". The fifth row contains four buttons: "Mentés (CSV) fájlba", "Betöltés (CSV) fájlból", "Szűrő alaphelyzetbe állítása", and "Település törlése".

2. feladat

1 pont

- A [Kiválasztott település adatai] résznél hozzon létre keretet! A keret vastagsága **3px**, a sarkok kerekítésének mértéke **20** és végül a keret színe **Brown**.
- Bővítse a már meglévő **Button** vezérlőket egy [Betöltés (CSV) fájlból] feliratú gombbal, ami kinézetében megegyezik a többivel! A nyomógombra történő kattintásra a már meglévő **btnCSVtoltes_Click** metódust hívja meg!

3. feladat

2 pont

A települések adatait egy **DataGrid** táblázat jeleníti meg. Az adatok forrása a korábban elkészített **Telepules** nevű osztály. Tekintse át ennek a tulajdonságait és működését!

- a) Bővítse a **Telepules** osztályt egy új, **VarosTipusuTelepules** nevű logikai (bool) típusú property-vel! Ennek az értéke legyen **true**, ha a település városi rangú település (pl. város, vármegyei jogú város, vármegye székhely). Fővárosi kerületet ne tekintse önálló városnak!

Ezt követően: A **DataGrid** definíciójában keresse meg a megjegyzésben elrejtett **DataGridCheckBoxColumn** típusú oszlopot és emelje ki a megjegyzésből!

Egészítse ki a táblázat oszlopait a következők szerint!

- b) Adjon a táblázathoz egy új oszlopot [**Változás**] névvel! Az oszlop megjelenését és formátumát állítsa be a minta alapján! A hozzá tartozó property már létezik a **Telepules** osztályban – keresse meg és használja fel!
- c) A [**Változás %-ban**] oszlopban a számok két tizedesjegyet mutassanak, mint ahogyan a mintán!

4. feladat

1 pont

A további feladatok megoldásához a **MainWindow.xaml.cs** és a **MainWindow.xaml** fájlokra lesz szüksége!

A **VezerlokElokeszites** metódusban töltsse fel a **cbTelepulesek** legördülő listát a **telepulesek** listában szereplő vármegyekódokkal! A kód csak egyszer szerepeljen és ábécé sorrendben! Az első sor az [**Összes megye**] szöveg legyen. Az első sor legyen kiválasztva alapértelmezetten!

Szűrés a vármegye kódjára:

Összes megye

Összes megye

Települések száma: 3177 db

en: 9 764 626 fő

Lakosok száma 2025-be

Település	Népesség 2024	Változás	Változás %-ban	Városnak számít?
Aba	4,6	6 fő	0.13 %	<input checked="" type="checkbox"/>
Abádszalók	4,1	-74 fő	-1.79 %	<input checked="" type="checkbox"/>
Abaliget	6	-23 fő	-3.54 %	<input type="checkbox"/>
Abasár	2,5	-8 fő	-0.32 %	<input type="checkbox"/>
Abaújalpár		8 fő	11.94 %	<input type="checkbox"/>
Abaújkér	5	7 fő	1.18 %	<input type="checkbox"/>
Abaújlak		-6 fő	-7.41 %	<input type="checkbox"/>
Abaújszántó	2,8	-22 fő	-0.78 %	<input checked="" type="checkbox"/>
Abaújszolnok	2	11 fő	4.58 %	<input type="checkbox"/>
Abaújvár	2	-10 fő	-3.76 %	<input type="checkbox"/>
Abda	3,3	-68 fő	-2.02 %	<input type="checkbox"/>

5. feladat

2 pont

A táblázat tartalmát szűrje le a legördülő listából kiválasztott vármegye kódja alapján! Gondoskodjon arról is, hogy ha az [Összes megye] opció van kiválasztva, akkor a táblázatban az összes település adata megjelenjen!

A szűrés végrehajtásával egyidejűleg frissítse a kapcsolódó statisztikai adatokat a **SzurtAdatokSzerintFrissites** metódus meghívásával! A metódusban a ■ jellel jelölt helyeken határozza meg a szükséges értékeket!

Tipp: Minden szűréskor érdemes a **szurt_telepulesek** listát először kiüríteni, majd újra feltölteni azokkal az elemekkel a **telepulesek** listából, amelyek megfelelnek a kiválasztott feltételnek.

Szűrés a vármegye kódjára:

Települések száma: 75 db Lakosok száma 2024-ben: 332 351 fő Lakosok száma 2025-ben: 329 082 fő

Település	Népesség 2024-ben	Népesség 2025-ben	Változás	Változás %-ban	Városnak számít?
Almáskamarás	787 fő	763 fő	-24 fő	-3.05 %	<input type="checkbox"/>
Battonya	5,793 fő	5,748 fő	-45 fő	-0.78 %	<input checked="" type="checkbox"/>

6. feladat

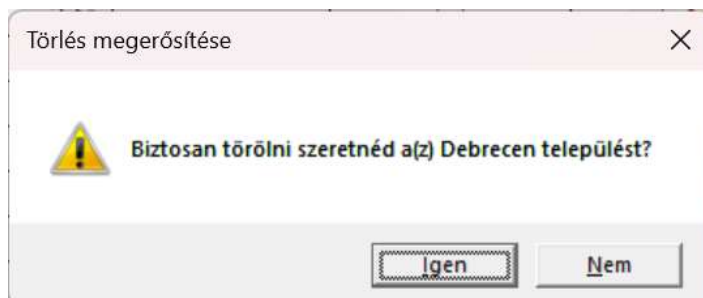
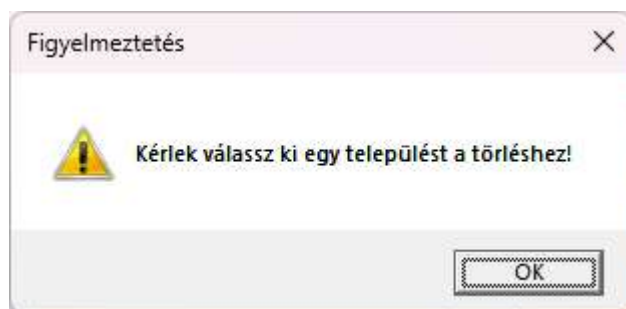
1 pont

- Fejezze be a **DataGrid** táblázathoz rendelt már elkezdett **dgTelepulesek_SelectionChanged** metódust! A táblázatban kiválasztott település további adatait jeleníti meg a metódus.
- Implementálja a **btnSzuroAlaphelyzetbe_Click** metódust, ami a szűrt alaphelyzetbe állítja. Figyeljen arra, hogy a statisztika is változik ilyenkor!

7. feladat

2 pont

- A [Település törlése] gombra kattintva távolítsa el a kijelölt települést mind a **telepulesek**, mind a **szurt_telepulesek** listából! A törlést követően frissítse a statisztikai adatokat is!
- Ha nincs kijelölve egyetlen település sem, akkor jelezze ezt a felhasználó felé a mintán látható módon!
- Törlés előtt kérdezzen rá, hogy a felhasználó valóban törölni szeretné-e az adott települést. A válasz alapján hajtsa végre, vagy szakítsa meg a műveletet!
- A [Mentés (CSV) fájlba] gombra kattintva nem jelenik meg a fájl mentési ablak. Derítse fel a hiba okát, és javítsa ki a kódot úgy, hogy a mentési ablak megfelelően megjelenjen és a válasznak megfelelő legyen a folytatás!



Magyarország településeinek népessége

Szűrés a vármegye kódjára:

H-AJ

Települések száma: 82 db

Lakosok száma 2024-ben: 532 227 fő

Lakosok száma 2025-ben: 529 761 fő

Kiválasztott település adatai

Neve: **Debrecen**

Típusa: **vármegye székhely**

Népesség 2024-ben: **Nők: 102268 fő Férfiak: 91711 fő**

Népesség 2025-ben: **Nők: 101277 fő Férfiak: 90932 fő**

Mentés (CSV) fájlba

Betöltés (CSV) fájlból

Szűrő alaphelyzetbe állítása

Település törlése

Település	Népesség 2024-ben	Népesség 2025-ben	Változás	Változás %-ban	Városnak számít?
Bedő	410 fő	410 fő	0 fő	0.00 %	<input type="checkbox"/>
Berekböszörmény	2,001 fő	2,016 fő	15 fő	0.75 %	<input type="checkbox"/>
Berettyóújfalú	14,378 fő	14,342 fő	-36 fő	-0.25 %	<input checked="" type="checkbox"/>
Bihardancsháza	158 fő	160 fő	2 fő	1.27 %	<input type="checkbox"/>
Biharkeresztes	4,846 fő	4,900 fő	54 fő	1.11 %	<input checked="" type="checkbox"/>
Biharnagybajom	2,597 fő	2,556 fő	-41 fő	-1.58 %	<input type="checkbox"/>
Bihartorda	981 fő	982 fő	1 fő	0.10 %	<input type="checkbox"/>
Bocskalkert	4,232 fő	4,293 fő	61 fő	1.44 %	<input type="checkbox"/>
Bojt	602 fő	594 fő	-8 fő	-1.33 %	<input type="checkbox"/>
Csökmő	1,869 fő	1,855 fő	-14 fő	-0.75 %	<input type="checkbox"/>
Darvas	509 fő	504 fő	-5 fő	-0.98 %	<input type="checkbox"/>
Debrecen	193,979 fő	192,209 fő	-1,770 fő	-0.91 %	<input checked="" type="checkbox"/>
Derecske	8,654 fő	8,576 fő	-78 fő	-0.90 %	<input checked="" type="checkbox"/>
Ebes	4,940 fő	4,926 fő	-14 fő	-0.28 %	<input type="checkbox"/>
Egyek	5,286 fő	5,247 fő	-39 fő	-0.74 %	<input type="checkbox"/>
Eztár	1,350 fő	1,343 fő	-7 fő	-0.52 %	<input type="checkbox"/>
Folyás	302 fő	303 fő	1 fő	0.33 %	<input type="checkbox"/>
Földes	3,869 fő	3,869 fő	0 fő	0.00 %	<input type="checkbox"/>
Furta	1,188 fő	1,215 fő	27 fő	2.27 %	<input type="checkbox"/>
Fülöp	1,745 fő	1,749 fő	4 fő	0.23 %	<input type="checkbox"/>
Gáborján	906 fő	911 fő	5 fő	0.55 %	<input type="checkbox"/>
Görbeháza	2,412 fő	2,408 fő	-4 fő	-0.17 %	<input type="checkbox"/>

10

III. Reszponzív viselkedésű weboldal készítés és formázás

10 pont

1. feladat

2 pont

A React-ben inline styling-el oldja meg, hogy a megjelenő kép keretének színe

- `var(--female-pink)` legyen, ha a kép neve tartalmazza a woman szót, és
- `var(--male-blue)` ha nem tartalmazza!

A további formázásokat az exam.css fájlban végezze el!

2. feladat

2 pont

Írjon szelektort, amely az `aboutCards` osztálykijelölővel ellátott elemre 100px-es felső margót rak és a benne lévő szöveget középre igazítja!

3. feladat

2 pont

Oldja meg, hogy az `aboutCards` tartalma rezponzívan viselkedjen!

- 768px ablakméret alatt a benne lévő elemek egy oszlopban,
- 768px fölött három oszlopban jelenjenek meg!

4. feladat

1 pont

Az `aboutCards`-ban lévő elemek között legyen 20px-es rés!

5. feladat

1 pont

Oldja meg, hogy a `navbarLinks` osztálykijelölővel ellátott számozatlan listában a pontok ne jelenjenek meg!

6. feladat

2 pont

A `navbarLinks` osztályon belüli link elemek ne legyenek aláhúzottak, és a színük legyen a -- font változóban feltüntetett szín (hivatkozzon a -- font változóra)!

IV. Backend programozás

(adatbázis lekérdezést is végző,
néhány végpontot tartalmazó REST API kiszolgáló létrehozása)

15 pont

A TúraRandi oldal backend szerverének forráskódja a [SajatNev_TuraRandi](#) könyvtárban található. Nevezze át a könyvtárat úgy, hogy a „SajatNev” helyére a saját neve kerüljön!

A szerver a <https://localhost:7161> vagy a <http://localhost:5202> címen érhető el.

1. feladat 3 pont

Nyissa meg a projektet és a feladatban leírt módosításokat ebben a projektben végezze el!

A [Models](#) könyvtárban lévő [Testimonial.cs](#) állományban hozzon létre [Testimonial](#) néven osztályt az alábbi tulajdonságokkal! (Ez reprezentálja az adatbázisban lévő tábla rekordjait.)

```
Testimonial(  
    Id: egész szám (ez lesz a tábla elsődleges kulcsa),  
    Name: szöveg,  
    Age: egész szám,  
    Location: szöveg (lehet NULL érték),  
    TestimonialText: szöveg,  
    ImageUrl: szöveg (lehet NULL érték)  
)
```

2. feladat 1 pont

A [TuraRandiDbContext](#) osztályban vegye fel a [Testimonials](#) táblát az adatbázisba!

3. feladat 2 pont

Konfigurálja az SQLite adatbázis kapcsolatot a [Program.cs](#) állományban a [Data\TuraRandi.db](#) adatbázishoz!

4. feladat 2 pont

A [/admin](#) oldalon elérhető az adatbázisban található adatok listája.

- Hozzon létre szervert oldalon renderelt (Razor Pages) oldalt Edit néven, amely biztosítja a tábla rekordjainak szerkeszthetőségét!
- Állítsa be az admin oldalon, hogy minden rekordnál elérhető legyen az adatok szerkeszthetősége!

5. feladat

1 pont

A `Controllers\AppController.cs` állományban találhatók a szerver által biztosított API végpontok, amelyek közül néhány hiányzik vagy hiányos. A végpontok a `/app` címen érhetők el.

Példányosítsa az adatbázis kapcsolat osztályt az `AppController` osztályban!

6. feladat

6 pont

Hozzon létre „API” végpontokat az `AppControllers` osztályban az alábbiak szerint:

- a) Hozzon létre a `/app` címen GET kéréshez végpontot, amely a bejegyzések listáját adja vissza!
- b) Hozzon létre DELETE kéréshez végpontot, amely az url-ben megadott azonosítójú rekordot törli az adatbázisból! Törlés előtt ellenőrizze, hogy létezik-e rekord az adatbázisban és jelezze azt vissza, ha nem!

A végpontok teszteléshez használhatja a `/swagger` oldalt.

V. Frontend programozás (HTML / CSS / JavaScript / REST API kliens) 15 pont

1. feladat 1 pont

Hozza létre a **Testimonials** komponenst az **src/components/Testimonials/** mappában található **Testimonials.tsx** fájlban, és exportálja alapértelmezettként!

2. feladat 1 pont

A komponens a **testimonialsBase.txt** fájlban leírt szerkezetet renderelje ki!

3. feladat 1 pont

Az **App** komponens az **About** szekció után, de a **Footer** előtt renderelje ki az imént létrehozott **Testimonials** komponenst!

4. feladat 2 pont

Hozza létre a **TestimonialType** típust, amely alkalmas a vélemények tárolására. Példa vélemény:

```
{
  "name": "Tóth János",
  "age": 20,
  "location": "Szeged",
  "testimonial": "A túrák remek alkalmat adnak az ismerkedésre!",
  "image": "/img/man2.jpg"
}
```

5. feladat 1 pont

Hozzon létre egy állapotot, amely **TestimonialType** típusú elemek listában való tárolására alkalmas! A value érték **testimonials** legyen, az update function pedig **setTestimonials**! Alapértelmezett értéke üres lista legyen!

6. feladat 2 pont

Olvassa be a backend feladatban használt **/app** végpontot. A megérkező adatokat tárolja az imént létrehozott **testimonials** állapotban!

Az api hívás csak az oldal betöltődésekor fusson le!

Megjegyzés: Ha még nem készült el a backendes feladat, vagy hibásan működik, használja a testimonials.json fájlt

7. feladat**1 pont**

Hozzon létre egy állapotot, amely az aktuális elem indexét fogja tárolni. A **value** érték **currentIndex**, az update function pedig **setCurrentIndex** legyen! Alapértelmezett értéke legyen 0.

8. feladat**1 pont**

Oldja meg, hogy a **section**-ban lévő **testimonialCard** osztálykijelölővel ellátott elem csak akkor renderelődjön ki, ha a **testimonials** listának van legalább egy eleme!

9. feladat**1 pont**

Oldja meg, hogy a szögletes zárójelek között jelzett értékek valóban megjelenjenek (**image, name, location, testimonial**)!

Használja a **testimonials** lista **currentIndex** indexű elemét!

10. feladat**2 pont**

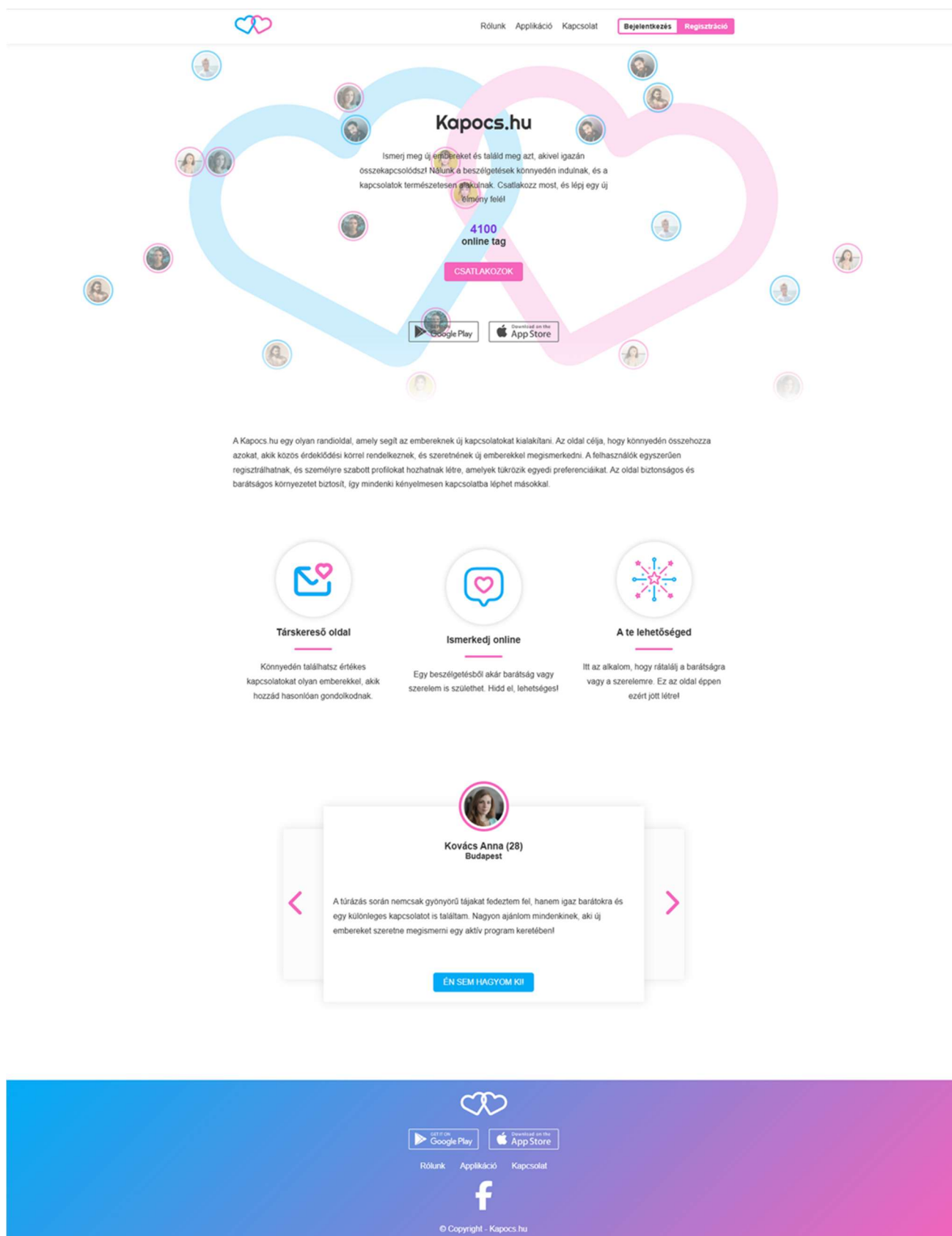
Hozzon létre egy **increaseIndex** függvényt, amely úgy növeli a **currentIndex** értékét, hogy az mindig a **testimonials** lista keretein belül maradjon (az utolsó elem indexe után az első elem indexére váltson)!

11. feladat**1 pont**

Hozza létre az előző logikához hasonlóan a **decreaseIndex** függvényt, amely csökkenti az értéket, de a 0 indexű elem után a lista végére ugrik!

12. feladat**1 pont**

A gombokhoz rendelje hozzá az index változtató függvényeket (a bal gombhoz a **decreaseIndex**-et, a jobbhoz pedig az **increaseIndex**-et)!





[Bejelentkezés](#)
[Regisztráció](#)

Kapocs.hu

Ismerj meg új embereket és találj meg azt, akivel igazán összekapcsolódsz! Nálunk a beszélgetések könnyedén indulnak, és a kapcsolatok természetesen alakulnak. Csatlakozz most, és lépj egy új élmény felé!



4643
online tag





CSATLAKOZOK




A Kapocs.hu egy olyan randioldal, amely segít az embereknek új kapcsolatokat kialakítani. Az oldal célja, hogy könnyedén összehozza azokat, akik közös érdeklődési körrel rendelkeznek, és szeretnék új emberekkel megismerkedni. A felhasználók egyszerűen regisztrálhatnak, és személyre szabott profilet hozhatnak létre, amelyek tükrözik egyedi preferenciáikat. Az oldal biztonságos és barátságos környezetet biztosít, így mindenki kényelmesen kapcsolatba léphet másokkal.



Társskereső oldal

Könnyedén találhatsz értékes kapcsolatokat olyan emberekkel, akik hozzád hasonlóan gondolkodnak.



Ismerkedj online

Egy beszélgetésből akár barátság vagy szerelem is születhet. Hidd el, lehetséges!



A te lehetőség

Itt az alkalom, hogy rátalálj a barátságra vagy a szerelemre. Ez az oldal éppen ezért jött létre!



Kovács Anna (28) Budapest

A túrázás során nemcsak gyönyörű tájakat fedeztem fel, hanem igaz barátokra és egy különleges kapcsolatot is találtam. Nagyon ajánlom mindenkinek, aki új embereket szeretne megismerni egy aktív program keretében!

ÉN SEM HAGYOM KI!






[Rólunk](#)
[Applikáció](#)
[Kapcsolat](#)



© Copyright - Kapocs.hu