**Szent Benedek Technikum És Gimnázium Szegedi Tagintézménye**

Informatika és Távközlés

Szoftverfejlesztő és -tesztelő

5 0613 12 03

Asztali- és webes szoftverfejlesztés, adatbázis-kezelés vizsga

…………………………………………………………………

Tanuló neve

………..

Osztálya

Szeged, 2025. május 19.

1. **Asztali szoftverfejlesztés (25 pont)**

**Pilóták**

A következő feladatban a Forma 1 pilótáinak adataiból készített szöveges állományból kell információkat kinyernetek. Ehhez mind konzolos, mind űrlapos alkalmazást is használnotok kell.

A megoldás során vegyétek figyelembe a következőket:

* *A képernyőre írást igénylő részfeladatok eredményének megjelenítése előtt írátok a*

*képernyőre a feladat sorszámát (például:* 3. feladat:)!

* *Az egyes feladatokban a kiírásokat a minta szerint készítsétek el!*
* *Az ékezetmentes azonosítók és kiírások is elfogadottak.*
* *A program megírásakor az állományban Iévő adatok helyes szerkezetét nem kell eIIenőriznetek, feltételezhetitek, hogy a rendelkezésre álló adatok a leírtoknak megfelelnek.*
* *Megoldásotokat úgy készítsék el, hogy az azonos szerkezetű, de tetszőIeges bemeneti adatok mellett is helyes eredményt adjon!*

**A** pilotak.csv **UTF-8 kódolású forrásállomány soraiban a következő sorrendben találjátok meg az adatokat:**

* + a pilóta neve, például: Lewis Hamilton
  + a pilóta születési dátuma, például: 1985.01.07
  + a pilóta nemzetisége, például: brit

**Az állomány első sora a mezőneveket tartalmazza, az adatokat pontosvesszővel választottuk el.**

1. Készítsetek konzolos alkalmazást a következő feladatok megoldásához, melynek projektjét ***VersenyzokKonzol***néven mentsétek el! **(1 pont)**
2. Hozzatok létre egy osztályt ***Versenyzo*** néven. Ebben az adattagokhoz a fenti három adatot használjátok. **(2 pont)**

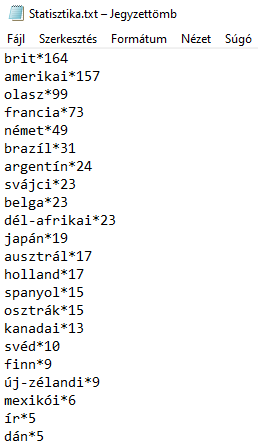
Olvassátok be az adatokat a ***pilotak.csv*** állományból és tároljátok el azokat egy olyan statikus adatszerkezetben az osztályban, amellyel később dolgozni tudtok! **(3 pont)**

**Minden további feladatot a konzolos részen ebben az osztályban létrehozott metódusokkal oldjatok meg!**

1. Írassátok ki a konzolra, hogy hány versenyző (adatsor) szerepel az állományban! **(1 pont)**
2. Keressétek meg a legfiatalabb pilótát. Írassátok ki a nevét és a születési dátumát az adott minta szerint. **(2 pont)**
3. Írassátok ki a konzolra az 1990 után született spanyol pilótákat. **(2 pont)**
4. Készítsetek statisztikát arról, hogy melyik nemzet hány pilótát adott a Forma-1-nek. A kapott eredményt írassátok ki egy ***Statisztika.txt*** nevű állományba! **Ezt a statisztikát használni fogjátok a grafikus résznél is!** A konzolra viszont csak az első 10 nemzetet írassátok ki pilótaszám szerint csökkenő sorrendben. **(4 pont)**

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás



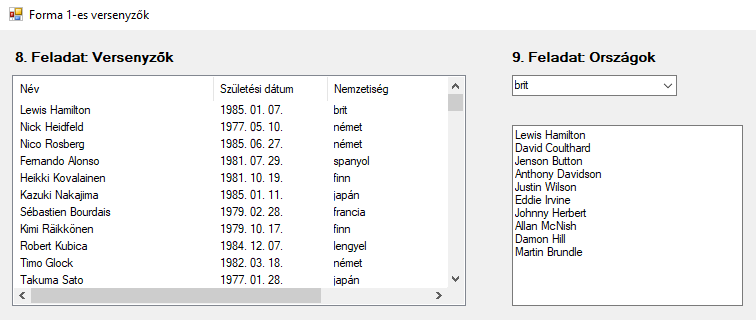
1. Készítsetek egy grafikus alkalmazást a következő feladatok megoldásához, melynek projektjét ***VersenyzokGrafikus***néven mentsétek el! Az űrlap neve a következő legyen: "**Forma 1-es versenyzők**"! **(2 pont)**
2. Egy **listView** elembe írassátok ki a versenyzők adatait a lenti minta alapján. A megoldáshoz használhatjátok a konzolos megoldás során létrehozott adatszerkezeteket! **(2 pont)**
3. Olvassátok be a **Statisztika.txt** állomány tartalmát, majd a nemzetek neveit egy legördülő listában mentsétek el! Kiválasztva egy tetszőleges nemzetet a listából egy **textBox** elemben soroljátok fel a versenyzőiket. Ha egy nemzet 10-nél több sofőrt adott, akkor csak az első 10-et. **A kiválasztás előtt a textBox legyen láthatatlan! (3 pont)**
4. Helyezzetek el az űrlapon, két dátum kezelésére alkalmas vezérlőelemet. Olvassátok be az értékeiket, majd a minta szerint írassátok ki az adott értékek között versenyzőket. Ha nincs ilyen versenyző, akkor a minta szerinti szöveget jelenítsétek meg! **(3 pont)**

A képen szöveg, képernyőkép, képernyő, szám látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, képernyő, szám látható

Automatikusan generált leírás



A képen szöveg, képernyőkép, képernyő, szoftver látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, képernyő, szám látható

Automatikusan generált leírás

1. **Webes szoftverfejlesztés és adatbázis-kezelés (40 pont)**

**Webshop**

Ebben a feladatrészben a MERN stack segítségével egy webes alkalmazást fogtok létrehozni egy különböző okostelefonokat árusító webshopról. Nem kell az egész feladatot az alapoktól felépíteni, hanem csak a hiányzó részeket kell pótolnotok. Az adatbázishoz a gépetekre telepített **XAMPP** szoftvercsomagot kell használnotok. A szerver oldal futtatásához a [http://localhost:5000](http://localhost:5000%20es) címet használjátok. A webalkalmazáshoz használható képek mellékelve lettek a feladat mappájában található **images** mappába.

1. A **backend** mappában telepíteni kell a függőségeket és szerkeszteni a **package.json** állományt. **(1 pont)**
2. Hozzatok létre egy adatbázist **telefonok** néven a phpmyadmin felületen. A megoldáshoz használt kódot másoljátok be a mellékelt **megoldas.sql** állományba. **(1 pont)**
3. Az adatbázis a következő adattáblákat tartalmazza:

**samsung** tábla

**samsungID** – egész szám, a samsung azonosítója, elsődleges kulcs, kötelező

**samsung\_nev** – szöveges, kötelező

**samsung\_max\_memoria** – szöveges, kötelező

**samsung\_mennyisegi\_ar** – egész szám, kötelező

**iphone** tábla

**iphoneID** – egész szám, az iphone azonosítója, elsődleges kulcs, kötelező

**iphone\_nev** – szöveges, kötelező

**iphone\_max\_memoria** – szöveges, kötelező

**iphone\_mennyisegi\_ar** – egész szám, kötelező

**xiaomi** tábla

**xiaomiID** – egész szám, a xiaomi azonosítója, elsődleges kulcs, kötelező

**xiaomi\_nev** – szöveges, kötelező

**xiaomi\_max\_memoria** – szöveges, kötelező

**xiaomi\_mennyisegi\_ar** – egész szám, kötelező

**vasarlasok** tábla

**vasarlasID** – egész szám, a vásárlás azonosítója, elsődleges kulcs, kötelező

**samsung\_id** – egész szám

**samsung\_mennyiseg** – egész, szám, kötelező

**iphone\_id** – egész szám

**iphone\_mennyiseg** – egész szám, kötelező

**xiaomi\_id** – egész szám

**xiaomi\_mennyiseg** – egész szám, kötelező

Az első három táblát nektek kell létrehozni. A megoldáshoz használt kódot másoljátok be a mellékelt **megoldas.sql** állományba. **(2 pont)**

1. Idegen kulcsok beállítása:

**samsungID** ↔ **samsung\_id**

**iphoneID** ↔ **iphone\_id**

**xiaomiID** ↔ **xiaomi\_id**

A megoldáshoz létrehozott kódokat másoljátok be a mellékelt **megoldas.sql** állományba. **(1 pont)**

1. A **.env** állományban állítsátok be a PORT számot és az adatbázis csatlakozáshoz szükséges adatokat. **(1 pont)**
2. Állítsátok be a **PORT**-ot és a **middleware**-eket a **server.js** állományban. **(1 pont)**

* public-mappára hivatkozás beállítása
* body-parser beállítása
* cors beállítása

1. Mappastruktúra alapján **route**-ok beállítása a **server.js** állományban. **(1 pont)**
2. Töltsétek fel a **samsung**, az **iphone** és a **xiaomi** táblákat a **feltoltRoutes.js** állomány segítségével a **Thunder Client** kiterjesztést használva. Az ehhez használt SQL utasításokat másoljátok be a **megoldas.sql** állományba. **(2 pont)**
3. Az **iPhone 16** ára megváltozott az **iPhone** táblában, 449900 forintról 469900 forintra drágult. A **megoldas.sql**-ben található SQL lekérdezés segítségével módosítsátok a **iphoneModositRoutes.js** állományt. A **Thunder Client** kiterjesztést használva végezzétek el a módosítást! **(2 pont)**
4. A **Galaxy A55** samsung okostelefon elfogyott, ezért törölni kell a **samsung** táblából. A **megoldas.sql**-ben található SQL lekérdezés segítségével módosítsátok a **samsungTorolRoutes.js** állományt. A **Thunder Client** kiterjesztést használva végezzétek el a törlést!

A **XAMPP** felületén adjátok a táblához a helyére lépő új árut a következő adatokkal. A hozzá tartozó kép a többi mellett található. **(3 pont)**

* **samsung\_nev**: Galaxy A56
* **samsung\_max\_memoria**: 256
* **samsung\_mennyisegi\_ar**: 219989

1. A **frontend** részen hiányzik egy szükséges függőség. Telepítsétek! **(1 pont)**
2. A **frontend** mappában szerkesszétek az **App.jsx** állományt. **(3 pont)**
3. Hozzátok létre a hiányzó **Home.jsx** állományt és szerkesszétek. Ehhez a mellékelt képek lesznek a segítségetekre. **(3 pont)**
4. Mivel a **xiaomi** okos telefonok csak később kerültek bele a kínálatba, ezért Nektek kell kiegészíteni a szükséges állományokkal és kódrészletekkel a kódot! Ehhez is a mellékelt képek nyújtanak segítséget. **(6 pont)**
5. Az **App.css** is egy kicsit hiányos. Egészítsétek ki, hogy jó legyen! **(1 pont)**
6. Az **images** mappa rossz helyen van. Helyezzétek el a megfelelő helyre, hogy az alkalmazás helyesen tudjon hozzá férni. **(1 pont)**
7. A reszponzivitáshoz több állomány szerkesztése szükséges. Ehhez bármilyen, általatok ismert módszer használható. Lehetnek média lekérdezések, vagy **Bootstrap5**, esetleg a **React** saját modulja, talán egy a szerkesztőbe épített kiterjesztés (**extension**). Először a **Home** oldalt tegyétek reszponzívvá. Eközben a szükséges stílus beállításokat is végezzétek el! A mellékelt képek lesznek a segítségetekre. **(2 pont)**
8. Ezután a **Samsung**, **IPhone** és **Xiaomi** oldalak következnek. **(2 pont)**
9. A **Cart** oldal a következő. **(2 pont)**
10. Végül a **Navbar**. Ehhez használnotok kell a **Google Icon**-okat is, mert a mobil nézet kialakításához használnotok kell a hamburger menüt is! **(4 pont)**