**„B”**

**Gyakorlati vizsgatevékenység B) feladatsor**

Szakképesítésmegnevezése:

5-0613-12-03 Szoftverfejlesztő-tesztelő

Vizsgatevékenység:

B) Asztali- és webes szoftverfejlesztés, adatbázis-kezelés vizsgarész

Vizsgarészhez rendelt vizsgafeladat megnevezése:

1. **a) Konzolos asztali alkalmazásfejlesztés Java, vagy C# nyelven – 15 pont**

**b)** **Grafikus asztali alkalmazásfejlesztés Java, vagy C# nyelven – 10 pont**

1. **Reszponzív viselkedésű weboldal készítés és formázás – 10 pont**
2. **Backend programozás (adatbázis lekérdezést is végző, néhány végpontot tartalmazó REST API kiszolgáló létrehozása) – 15 pont**
3. **Frontend programozás (HTML/CSS/JavaScript/ REST API kliens) -15 pont**

A vizsgafeladat időtartama: 240 perc

|  |  |  |
| --- | --- | --- |
| A gyakorlati vizsgafeladat elfogadását javaslom: | dátum | a vizsgaszervező intézmény képviselője |
| A gyakorlati vizsgafeladattal egyetértek: | dátum | a vizsgabizottság tagja |
| A gyakorlati vizsgafeladattal egyetértek: | dátum | a vizsgabizottság tagja |
| A gyakorlati vizsgafeladatot jóváhagyom: | dátum | a vizsgabizottság elnöke |

*A gyakorlati feladatlapokra vonatkozó javaslatot – a vizsga megkezdése előtt – a vizsgaelnök a vizsgabizottsági tagok véleménye alapján hagyja jóvá.*

**1. Feladat**

**Konzol és grafikus elemeket tartalmazó asztali alkalmazás fejlesztése. A feladat egy projekten belül, illetve két különálló projektként is elfogadható:**

1. **Konzolos asztali alkalmazásfejlesztés Java, vagy C# nyelven:**

Egy online casino kezelő cégnél a nyeremenyek.txt forrásfájl alapján kell elszámolni a feladott játék és kiszámolni a kifizetéseket online rulett játékkal kapcsolatban.

A forrásfájlban az első elem a rulett játék sorszáma (1-től kezdődik és egyesével növekszik), utána következik a felhasználónév, a fogadott tét és a szorzó, amivel a nyeremény összegét lehet kiszámolni, végül az utolsó adat, hogy nyertes, vagy vesztes játszma. A nyeremény kiszámolásához a fogadott tétet és a szorzót kell szorozni egymással, ha nyertes játszmáról van szó.

A forrásfájl felhasználásával oldja meg a következő programozási feladatokat:

1. Olvassa be a forrásfájlt és objektumorientált megoldással tárolja el egy megfelelő adatstruktúrában (pl: tömb, vagy lista).

2. Kérjen be egy számot és az alapján írja ki a bekért számmal megegyező sorszámú játszma adatait.

3. Adja meg, melyik felhasználó tette meg a legtöbb játszmát és ez mennyibe került összesen!

4. Adja meg hány nyertes játszma van összesen!

5. Írja ki annak a játszmának az adatait, amely után a legkisebb nyereményt lehetett felvenni.

6. Keresse ki az összes ’a’ betűvel kezdődő felhasználónevet, írja ki hány darabot talált és az első ötöt írja is ki!

7. Írja az összes nyertes játszma sorszámát, a felhasználó nevét és az adott játszmához tartozó nyeremény összegét.

**1.feladat a.) értékelő lap**

|  |  |  |
| --- | --- | --- |
| **Szakmai ismeretek / feladatprofil alkalmazása a szakmai és vizsgakövetelmény szerint** | **Pontszámok** | |
| **Max.** | **Elért** |
| **1. Szöveges fájl beolvasása (0.5), beolvasott adatok feldolgozása(0.5), adatstruktúrában való tárolása (1), objektumorientált megoldással(1)** | **3** |  |
| **2. Inputbekérés(0.5), kereséshez szükséges algoritmus megírása(1), eredmény kiíratása(0.5)** | **2** |  |
| **3. Megoldáshoz szükséges algoritmus megírása(1), eredmény kiíratása(1)** | **2** |  |
| **4. Megoldáshoz szükséges algoritmus megírása(1), eredmény kiíratása(1)** | **2** |  |
| **5. Megoldáshoz szükséges algoritmus megírása(0.5), eredmény kiíratása(0.5)** | **2** |  |
| **6. Megoldáshoz szükséges algoritmus megírása(0.5), eredmény kiíratása(0.5)** | **2** |  |
| **7. Megoldáshoz szükséges algoritmus megírása, megfelelő eredmény kiíratása(2)** | **2** |  |
| **Összesen** | **15** |  |

1. **Grafikus asztali alkalmazásfejlesztés Java, vagy C# nyelven:**

Egy online casino kezelő cégnél a nyeremenyek.txt forrásfájl alapján kell elszámolni a feladott játék és kiszámolni a kifizetéseket online rulett játékkal kapcsolatban.

A forrásfájlban az első elem a rulett játék sorszáma (1-től kezdődik és egyesével növekszik), utána következik a felhasználónév, a fogadott tét és a szorzó, amivel a nyeremény összegét lehet kiszámolni, végül az utolsó adat, hogy nyertes, vagy vesztes játszma. A nyeremény kiszámolásához a fogadott tétet és a szorzót kell szorozni egymással, ha nyertes játszmáról van szó.

A forrásfájl felhasználásával oldja meg a következő programozási feladatokat:

1. Olvassa be a forrásfájlt és objektumorientált megoldással tárolja el egy megfelelő adatstruktúrában (pl: tömb, vagy lista).

2. Kérjen be egy számot és az alapján írja ki a bekért számmal megegyező sorszámú játszma adatait.

3. Adja meg, melyik felhasználó tette meg a legtöbb játszmát és ez mennyibe került összesen!

4. Adja meg hány vesztes játszma van összesen!

5. Írja ki annak a játszmának az adatait, amely után a legnagyobb nyereményt lehetett felvenni.

6. Számolja ki és írja is ki, hogy anti12 felhasználó mennyi pénzt keresett a játszmákkal, ha a nyereményekből levonjuk az általa tett összes nyertes és vesztes játszmák tétjeit.

7. Írja ki egy táblázatba az összes nyertes játszma sorszámát, a felhasználó nevét és az adott játszmához tartozó nyeremény összegét.

8. Az asztali alkalmazást a képnek megfelelő design szerint hozza létre (piros háttér, a képnek megfelelő elemek használatával!

Table

Description automatically generated

**1.feladat b.) értékelő lap**

|  |  |  |
| --- | --- | --- |
| **Szakmai ismeretek / feladatprofil alkalmazása a szakmai és vizsgakövetelmény szerint** | **Pontszámok** | |
| **Max.** | **Elért** |
| **1. Szöveges fájl beolvasása (0.5), beolvasott adatok feldolgozása(0.5), adatstruktúrában való tárolása (0.5), objektumorientált megoldással(0.5)** | **2** |  |
| **2. Inputbekérés(0.5), kereséshez szükséges algoritmus megírása(1), eredmény kiíratása(0.5)** | **2** |  |
| **3. Megoldáshoz szükséges algoritmus megírása(0.5), eredmény kiíratása(0.5)** | **1** |  |
| **4. Megoldáshoz szükséges algoritmus megírása(0.5), eredmény kiíratása(0.5)** | **1** |  |
| **5. Megoldáshoz szükséges algoritmus megírása(0.5), eredmény kiíratása(0.5)** | **1** |  |
| **6. Megoldáshoz szükséges algoritmus megírása(0.5), eredmény kiíratása(0.5)** | **1** |  |
| **7. Megoldáshoz szükséges algoritmus megírása, megfelelő eredmény kiíratása(0.5), datagridview/ egyéb táblázat formátum használata(0.5)** | **1** |  |
| **8. A design megegyezik a feladatban mellékelt képen vázolt javaslattal.** | 1 |  |
| **Összesen** | **10** |  |

**2. Feladat**

**Reszponzív viselkedésű weboldal készítés és formázás**

Készítse el a következő reszponzív weboldalt, ami egy szakszervezeti tagnyilvántartási rendszer felületéért felelős. A megoldásokhoz használjon bootstrap frameworkot, vagy css-t, vagy egyéb megoldás is elfogadható az oldal formázására!

1. Hozzon létre 2 db html fájlt, az egyik az TAGFELVÉTEL menüpont alatt legyen elérhető, a másik a TAGNYILVÁNTARTÁS menüpont alatt. Adja hozzá mindkét html fájlhoz a kiinduló kódot!
2. A weboldal oldalainak adjon világos piros háttérszínt, a navigációs elemnek és a formokat tartalmazó elemeknek piros háttérszínt
3. A navigációs elemhez adjon hozzá egy logót brandként, ami a logo.png képet tartalmazza, a menüpontoknál használjon bootstrap btn osztályt úgy hogy a gombok legyenek körvonalasak, úgy hogy az aktuális oldalnál legyen active a nyomógomb. Hasonló eredményt produkáló más megoldás is elfogadott.
4. A menüpontokra kattintva váltson a webldal az oldalak között!
5. A brandnél hozzáadott képnél adjon meg 80 px magasságot!
6. A TAGFELVÉTEL oldalon hozzon létre egy formot a képen látható módon. A formon belül található szöveget placeholder-ként adja hozzá a kódhoz! Adjon minden input mezőnek egy id attributumot is.
7. A RÖGZÍTÉS gombra kattintva váltson át a weboldal a tagnyilvántartást tartalmazó oldalra!
8. A TAGNYILVÁNTARTÁS menüpont alatt hozza létre a képen lázható módon az oldalt. Ehhez használjon táblázatot és a táblázat cellái az id oszlop celláinak kivételével tartalmazzon inputmezőket, melyeknék value-ként adja meg a forrásfájlok közötti excel táblában mellékelt adatokat!
9. A TAGNYILVÁNTARTÁS táblázathoz adjon hozzá két button-t is a képen látható módon!

Az oldal legyen reszponzív, csökkentve a böngésző szélességét a menőpontok legyenek elrejtve egy „hamburger button” mögé, melyre rákattintva hozza elő a menüpontokat a képen látható módon!

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

A „hamburger” gombra kattintva így jelenjenek meg az elrejtett menüpontok

Graphical user interface, application

Description automatically generated

**2. feladat értékelő lap**

|  |  |  |  |
| --- | --- | --- | --- |
| **Szakmai ismeretek / feladatprofil alkalmazása a szakmai és vizsgakövetelmény szerint** | **Pontszámok** | | |
| **Max.** | **Elért** | |
| **1. Html fájlok létrehozása, kiinduló forráskód hozzáadása** | **0,5** |  | |
| **2. Háttérszín megadása, elemek színének beállítása** | **1** |  | |
| **3. Navigációs elem hozzáadása a menüpontokkal (0,5) és ezek formázása (0,5), kép hozzáadása a brandhez(0,5)** | **1,5** |  | |
| **4. A menüpontokhoz hivatkozás hozzáadása** | **1** |  | |
| **5. A kép méretének beállítása** | **0,5** |  | |
| **6. Form létrehozása(0,5), formázása(0,5), placeholderek hozzáadása(0,5)** | **1,5** |  | |
| **7. Nyomógombhoz hivatkozás hozzáadása(0,5)** | **0,5** |  | |
| **8. Táblázat létrehozása (0,5), inputmezők hozzáadása(0,5), value attributumok hozzáadása(0,5)** | **1,5** |  | |
| **9.Nyomógombok hozzáadása a táblázat soraihoz(0,5)** | **0,5** |  | |
| **10. Formázás megegyezik a forrásfájlokban található screenshotokkal(0,5), a feladatleírás szerinti reszponzív működés(1)** | **1,5** | |  |
| **Összesen** | **10** | |  |

**3. Feladat**

**Backend programozás (adatbázis lekérdezést is végző, néhány végpontot tartalmazó REST API kiszolgáló létrehozás**

A feladat során egy REST API webapplikáció szerver oldalát kell létrehozni. A webapplikáció egy szakszervezeti tagnyilvántartó rendszer feladatait látja el, a tagok adatait lehet létrehozni, kiíratni, törölni és módosítani.

1. Az adatok tárolásához használjon SQLite adatbázist, ami tartalmaz egy tagok táblát, a táblán belül legyen képes tárolni egy automatikusan generált ID-t, a vezetéknevet, keresztnevet, a munkahelyet és az e-mail címét. Az adatbázist a webapplikációval azonos programnyelvben megírt kódsor futtatásával hozza létre.
2. A webapplikáció legyen képes adatok létrehozására, kiolvasására, módosítására és törlésére.
3. A következő végpontokon lehessen elérni a különféle funkciókat:
   1. Create: /new
   2. Read: /read
   3. Update: /upd
   4. Delete: /del
4. A get requestre a response során az adatok legyenek json formátumban átadva.
5. A webapplikáció futtatható legyen és a megadott 4 útvonalon leállás és hiba nélkül működjön!

**3. feladat értékelő lap**

|  |  |  |  |
| --- | --- | --- | --- |
| **Szakmai ismeretek / feladatprofil alkalmazása a szakmai és vizsgakövetelmény szerint** | **Pontszámok** | | |
| **Max.** | **Elért** | |
| **1. Adatbázis létrehozására kódsor megírása a választott programnyelven(1), tábla létrehozása(1), megfelelő attributumok létrehozása a táblán belül(1) db fájl létrehozása(1), API-n belül connection létrehozása az adatbázissal(1)** | **5** |  | |
| **2. Create funkció megírása(1), Read funkció megírása(1), Update funkció megírása(1), Delete funkció megírása(1)** | **4** |  | |
| **3. Útvonalak megfelelő elnevezése (0.5 pont/route)** | **2** |  | |
| **4. Response válasz json formátumban érkezik a get requestre** | **2** |  | |
| **5. Web api helyes futása minden útvonalnál** | **2** |  | |
| **Összesen** | **15** | |  |

**4. Feladat**

**Frontend programozás (HTML/CSS/JavaScript/ REST API kliens)**

Hozzon létre egy REST API klienst, ami a 3. feladatban létrehozott REST API kiszolgálóval kommunikál!

1. Formázza meg a weboldalt a 2. feladat leírása szerint és egészítse ki javascript fájllal!

2. Az TAGFELVÉTEL menüpont alatti oldalt kösse össze a backend feladat /new végpontjával, tegye alkalmassá post request küldésére a szerver oldal felé! A megoldás paraméterátadással, vagy json formátumú adathalmaz átadásával is történhet.

3. A TAGNYILVÁNTARTÁS menüpont alatti táblázatot úgy töltse fel a szerver oldaltól kapott tagsági adatokkal, hogy ugyanott törölni és módosítani is lehessen ezeket az adatokat, ebből a célból a szerver oldal felé induljon post request is.

4. A frontend oldal mind a 4 végponttal helyesen kommunikáljon, ne okozzon leállást, vagy hibát!

5. A megoldást lehetőség szerint Angular/VueJs/React frameworkben oldja meg! Más egyéb megoldás is elfogadható részpontszámért!

**4. feladat értékelő lap**

|  |  |  |  |
| --- | --- | --- | --- |
| **Szakmai ismeretek / feladatprofil alkalmazása a szakmai és vizsgakövetelmény szerint** | **Pontszámok** | | |
| **Max.** | **Elért** | |
| **1. Weboldal és html fájlok formázása(1), új diák oldal elemeinek létrehozása(1), diáknyilvántartás oldal elemeinek létrehozása(1), Javascript fájl létrehozása(1)** | **4** |  | |
| **2. form és a hozzátartozó javascript funkció post requestre megírása(2), adatok helyes átadása(1)** | **3** |  | |
| **3. A táblázat feltöltése adatokkal get request javascript funkcióval(2), módosítás post request javascript funkció megírása(2), törlés post request javascript funkció megírása(1)** | **5** |  | |
| **4. Backenddel való kommunikáció minden útvonalon helyesen működik (0.5 pont/útvonal)** | **2** |  | |
| **5. Angular/VueJs/ReactJS használata a megoldásban** | **1** |  | |
| **Összesen** | **15** | |  |