

Szoftverfejlesztő és tesztelő szakmai vizsga

Kedves Vizsgázó!

Kérem olvassa el figyelmesen az alábbi tájékoztatót, mielőtt elkezdené a vizsgát.

- A vizsga időtartama 140 perc.
- A vizsga során semmilyen segédeszköz (könyv, jegyzet, elektronikus eszköz, stb) használata nem megengedett, kivéve az iskolai gép fejlesztéshez, illetve a csatolt szöveges állományban megjelölt hivatalos dokumentációk.
- A vizsgafeladatok megoldása során bármilyen csalási kísérlet a vizsgáról való azonnali kizárást vonja maga után.
- A vizsgafeladat megoldásához bármilyen JavaScript frontend keretrendszer (pl. React), bármilyen nyelvben íródott backend keretrendszer (pl. Node.JS-hez ExpressJS), illetve bármilyen relációs adatbáziskezelő (pl. MySQL/MariaDB) technológiákat lehet használni.
- A feladatok megoldását egy lokális könyvtárban kell elkészíteni, amely neve “*Sajat_Nev*” névre hallgat, és *frontend* illetve *backend* alkönyvtárakat tartalmaz. Ha a beadott állomány eltér ettől, pontlevonás jár.
- A *backend* könyvtárban készítsen egy *database* könyvtárat, amely tartalmazza az adatbázis elkészítéséhez szükséges **összes** utasítást. Ha ez az állomány hiányzik, az adatbázis nem értékelhető.
- A beadott állományban a *node_modules* könyvtár (vagy ehhez hasonló, függőségek kezelésére szolgáló könyvtár) **nem** szerepelhet, és pontlevonással jár annak beadása.
- A vizsga végén a megoldott vizsgafeladatot tartalmazó könyvtár tartalmát egy *ZIP* formátumú állományba tömörítse, és töltsse fel a megadott helyre.
- A vizsga sikeres, ha legalább 40%-ot, azaz 14 pontot elért.
- A vizsga során felmerülő kérdéseivel forduljon a vizsgabiztoshoz.

Feladatleírás

Feladata egy játékokkal foglalkozó felhasználók által kezelt webáruház alkalmazás egyszerűsített változatának elkészítése szerver-kliens architektúrára alapulva, amely az alábbi funkciókkal rendelkezik.

A főoldalon jelenjen meg egy navigációs sáv, amely 3 fült tartalmaz:

- **Kezdőlap:** Jelenjenek meg kártyákban a játékok és a hozzájuk tartozó adatok. Ha a felhasználó már be van jelentkezve, jelenjen meg az játék ára.

- **Bejelentkezés:** Felhasználónév vagy e-mail cím, illetve jelszó megadásával tudjon bejelentkezni a már regisztrált felhasználó. Bejelentkezés után ennek a menüpontnak a helyén kijelentkezés gomb jelenjen meg, illetve egy saját játékaim fül, ahol az adott felhasználó által közzétett játékok jelennek meg.
- **Regisztráció:** Felhasználónév, e-mail cím, jelszó segítségével tudjon regisztrálni a vendég felhasználó. Ügyeljen arra, hogy egy e-mail címhez, illetve egy felhasználónévhez egy felhasználó tartozzon.

Adatbázis

Hozza létre a következő táblákat a relációs adatbáziskezelőben:

users (felhasználók):

- id (azonosító, egész szám, elsődleges kulcs, automatikusan növekvő)
- username (felhasználónév, szöveg, egyedi)
- email (e-mail cím, szöveg, egyedi)
- password (titkosított jelszó, szöveg)

games (játékok):

- id (azonosító, egész szám, elsődleges kulcs, automatikusan növekvő)
- owner (tulaj, egész szám, idegen kulcs → users tábla id mezője)
- developer (fejlesztő cég, szöveg)
- name (név, szöveg, GTA V)
- price (ár, egész szám)

Az adatbázisba hozzon létre három felhasználót (köztük egynek legyen admin a felhasználóneve, admin@admin.hu az e-mail címe és admin a jelszava), és a játékok táblába töltsön fel három játékot, tetszőleges adatokkal, egyik felhasználóhoz.

Az adatbázis és táblák létrehozásához, illetve az adatok feltöltéséhez használt utasításokat illessze be a *backend/database/input.sql* állományba. Az is elfogadható, ha SQL dump-ot készít (phpMyAdmin-on keresztül *Exportálás* menüpont) és ennek tartalmát adja be. Az állomány tartalmazza az **összes** SQL utasítást, amennyiben valami hiányzik, az úgy tekintendő, mintha az adott feladatrésze nincsen megoldás.

Backend

Készítsen egy szerveret, amely a következő tulajdonságoknak felel meg:

- REST API-t implementál,
- JSON formátumban fogadja a kéréseket és küldi a válaszokat,
- kapcsolódik az adatbázis-szerverhez, és utasításokat futtat rajta (pl. lekérdezés),
- megfelelő HTTP státuszkódokkal jelez a kliensnek,
- **kezeli a hibákat minden végponton,**
- **ellenőrzi a kérésben található adatokat minden végponton,**
- JSON Web Token alapú hitelesítéssel alkalmaz védett végpontokat,

ezen szerveren pedig implementálja az alábbi végpontokat:

POST /register

Regisztrál egy új felhasználót, és elmenti azt az adatbázisba. A jelszavakat titkosítva tárolja az adatbázisban!

POST /login

Egy már regisztrált felhasználóhoz készít JSON Web Tokent, és azt visszaadja a kliensnek.

GET /games

Visszaadja az adatbázisban eltárolt játékokat.

POST /games

Védett végpont, felhasználói azonosítással. Egy új játékot elment az adatbázisba, és annak tulaját beállítja a token által megadott felhasználóra.

DELETE /games/{id}, ahol *id* lehet 0, 1, 2, stb...

Védett végpont, felhasználói azonosítással. A token által megadott felhasználó által közzétett játékokat lehet törölni, csak akkor, ha az *id* által megadott játék tulaja a token által megadott felhasználó.

GET /owned_games

Védett végpont, felhasználói azonosítással. A token által megadott felhasználó által közzétett játékokat adja vissza.

Frontend

Készítsen egy klienst az előbb elkészített szerverhez, amely egy reszponzív weboldal legyen. A kliens a következő követelményeknek feleljen meg:

- jelezze a felhasználónak a hibákat,
- **minden** felhasználói bemenetnél elfogadható mennyiségű ellenőrzést végezzen (pl. kötelező mezők kitöltése, stb),

- próbálja meg a lehető legjobb módon használni a frontend keretrendszer által szolgáltatott eszközöket,
- megjelenésre nincsen megkötés, próbálja meg magából a legtöbbet kihozni,
- felhasználói élményre sincsen megkötés, szintén próbálja meg a lehető leginkább gördülékennyé tenni azt,

ezen a weboldalon pedig implementálja az alábbiakat:

Navigációs sáv

Három menüponttal rendelkezik: *Főoldal*, *Bejelentkezés*, *Regisztráció*. Amint bejelentkezett a felhasználó, ehelyett a következők jelenjenek meg: *Főoldal*, *Saját Játékok*, *Kijelentkezés*.

Regisztráció

Egy űrlap segítségével regisztrálhat a felhasználó, a megfelelő végponton.

Bejelentkezés

Egy űrlap segítségével bejelentkezhet a felhasználó, a megfelelő végponton. Bejelentkezés után a szerver által elkészített tokent mentse el, és későbbiekben ennek segítségével azonosítsa a felhasználót.

Kezdőlap

Kártyás formátumban jelenjenek meg az játékok fejlesztővel és névvel.

Saját Játékok

Kártyás formátumban jelenjenek meg a felhasználó által közzétett játékok. Minden kártyának az alján legyen egy *Eltávolítás* gomb, amely az adott játékot eltávolítja az adatbázisból. Az oldal tetején legyen egy *Új játék közzétetele* gomb, amelyen egy űrlap segítségével egy új játékot tölthet fel a felhasználó.

Kijelentkezés

A már bejelentkezett felhasználó jelentkezzen ki. Törölje a tokent.

Javítási-értékelési útmutató

Név	Elérhető pontszám	Elért pontszám
Backend, adatbázis	15	
users tábla létrehozása	1	
games tábla létrehozása	1	
Adatbázisba felhasználók feltöltése	1	
Adatbázisba játékok feltöltése felhasználóhoz	1	
Adatbázishoz való kapcsolódás szerveren	1	
Regisztráció, jelszó titkosításával	2	
Bejelentkezés, token elkészítésével	2	
Játékok lekérdezésére szolgáló végpont	1	
Játékok feltöltésére szolgáló végpont	2	
Saját játékok törlésére szolgáló végpont	2	
Saját játékok lekérdezésére szolgáló végpont	1	
Frontend	15	
Ki- és bejelentkezett navigációs sáv elkészítése	2	
Regisztrációs űrlap elkészítése, ellenőrzésekkel	2	
Hibák megjelenítése felhasználónak regisztrációnál	2	
Bejelentkezés, token elmentése, hibák megjelenítése	2	
Kezdőlapon játékok megjelenítése	1	
Hibák kezelése játékok lekérdezésénél	1	
Új játék közzététele	1	
Figyelmeztető üzenet kedvenceknél	1	
Saját játék megjelenítése	1	
Saját játék eltávolítása	1	
Kijelentkezés	1	
Weboldal megjelenése	10	
Navigációs sáv	2	
Űrlapok	2	
Kártyák	2	
Hibaüzenetek	2	
Gombok	2	
Összesen	40	