# Szoftverfejlesztő Technikus Vizsgafeladat

## Tájékoztató

Kedves Vizsgázó!

Kérlek, olvasd el figyelmesen az alábbi tájékoztatót, mielőtt elkezdenéd a vizsgát.

* A vizsga időtartama 180 perc.
* A vizsga során semmilyen segédeszköz (könyv, jegyzet, elektronikus eszköz, stb.) használata nem megengedett, kivéve a saját gépedet a fejlesztéshez.
* A vizsgafeladatok megoldása során törekedj az önálló munkavégzésre. Bármilyen csalási kísérlet a vizsgáról való azonnali kizárást vonja maga után.
* A vizsgafeladatok megoldásához React frontend, Node.js backend és MariaDB adatbázis technológiákat kell használni.
* A feladatok megoldását egy lokális mappában kell elkészíteni, amely a frontend és backend almappákat tartalmazza.
* A backend mappában lennie kell egy database mappának, azon belül pedig egy input.sql fájlnak, amely tartalmazza az adatbázis és a táblák létrehozásához, illetve az adatok feltöltéséhez szükséges SQL parancsokat.
* A vizsga végén a teljes, megoldott vizsgafeladatot tartalmazó mappát kell **ZIP formátumba tömöríteni**, és feltölteni a megadott helyre. A ZIP fájl neve a következő formátumú legyen: vezeteknev\_keresztnev\_vizsgamunka.zip (pl. minta\_bela\_vizsgamunka.zip).
* A vizsga sikeres, ha legalább 40 pontot elértél.
* A vizsga során felmerülő kérdéseiddel fordulj bizalommal a vizsgabiztoshoz.

Sok sikert kívánunk!

## Feladatleírás

Készíts egy **blog** alkalmazást, amely a következő funkciókkal rendelkezik:

### **Adatbázis (15 pont)**

* **Tervezd meg és hozd létre** a MariaDB adatbázist és a szükséges táblákat a blog alkalmazáshoz. A táblákban legyenek a következők:
  + **Felhasználók tábla:**
    - id (azonosító, auto-increment, primary key)
    - username (felhasználónév, egyedi)
    - display\_name (megjelenítendő név)
    - password (titkosított jelszó)
  + **Posztok tábla:**
    - id (azonosító, auto-increment, primary key)
    - user\_id (felhasználó azonosító, foreign key - users tábla)
    - title (poszt címe)
    - content (poszt tartalma)
  + **Like-ok tábla:**
    - id (azonosító, auto-increment, primary key)
    - user\_id (felhasználó azonosító, foreign key - users tábla)
    - post\_id (poszt azonosító, foreign key - posts tábla)
* Biztosítsd, hogy egy felhasználó csak egyszer like-olhasson egy posztot (unique kulcs a user\_id és post\_id oszlopokon), de a like-ot vissza is vonhassa.
* A backend/database mappában hozz létre egy input.sql fájlt, amely tartalmazza az adatbázis és a táblák létrehozásához, illetve az adatok feltöltéséhez szükséges **összes SQL parancsot**.
* Az adatbázisban hozz létre legalább 3 felhasználót, és minden felhasználóhoz legalább 2 posztot. Az ezekhez szükséges SQL parancsokat is illeszd be az input.sql fájlba.

### **Backend (Node.js) (35 pont)**

* **CRUD műveletek (15 pont)**
  + Készíts REST API végpontokat a posztok létrehozásához, olvasásához, frissítéséhez és törléséhez.
  + A posztok olvasásánál legyen lehetőség a posztok szerző szerinti szűrésére is.
* **Hitelesítés (10 pont)**
  + Valósítsd meg a felhasználók regisztrációját és bejelentkezését **JSON Web Token (JWT)** alapú hitelesítéssel.
  + A jelszavakat titkosítva tárold az adatbázisban.
  + A bejelentkezett felhasználók csak a saját profiljukat módosíthassák.
  + A posztokat csak bejelentkezett felhasználók hozhassák létre, módosíthassák és törölhessék. A nem bejelentkezett felhasználók megtekinthetik a posztokat.
  + Minden poszt csak a saját felhasználója által legyen módosítható és törölhető.
* **Like-olás (5 pont)**
  + Valósítsd meg a posztok like-olását és a like-ok visszavonását.
  + Egy felhasználó csak egyszer like-olhasson egy posztot, de a like-ot bármikor visszavonhassa.
* **Keresés (5 pont)**
  + Valósítsd meg a posztok keresését a posztok címében és tartalmában **paraméterek** segítségével.

### **Frontend (React, Bootstrap) (50 pont)**

* **Alapok (5 pont)**
  + Hozz létre egy React alkalmazást.
  + Használj Bootstrap-et a reszponzív design kialakításához.
* **Bejelentkezés, regisztráció (15 pont)**
  + Készíts egy bejelentkezési és egy regisztrációs űrlapot.
  + Az űrlapokon végezz validációt (pl. kötelező mezők kitöltése, jelszó erőssége, e-mail formátum).
  + Sikertelen bejelentkezés vagy regisztráció esetén jeleníts meg hibaüzenetet egy felugró ablakban.
* **Posztok megjelenítése (15 pont)**
  + Jelenítsd meg a posztokat egy áttekinthető formában.
  + Minden posztnál jelenítsd meg a poszt címét, tartalmát, szerzőjét (**display\_name** és **username**), és a like-ok számát.
  + Legyen lehetőség a posztok like-olására és a like-ok visszavonására.
* **Profil módosítás (5 pont)**
  + Készíts egy űrlapot a felhasználói profil módosításához.
  + Legyen lehetőség a felhasználónév és egy megjelenítendő név (display name) megadására.
* **Posztok kezelése (5 pont)**
  + Készíts űrlapokat a posztok létrehozásához, módosításához és törléséhez.
  + Biztosítsd, hogy a posztokat csak a saját felhasználójuk módosíthassa és törölhesse.
* **Keresés (5 pont)**
  + Készíts egy keresőmezőt, amellyel a posztok között lehet keresni a cím és a tartalom alapján, paraméterek segítségével.

**Összesen: 100 pont**