**Webes szoftverfejlesztés gyakorló feladat – Túra**

A feladatod egy túrákkal foglalkozó cég reszponzív (Bootstrap alapú) weboldalának elkészítése lesz. A meghirdetett túrákra jelentkezni email és résztvevő létszám megadásával kell. Mivel a célban mindenkinek emlékkitűzőt osztanak, így a túrán megadott maximális jelentkezőszámot meghaladni nem lehet! A szerver oldali REST API kiszolgálót Laravel projektként, a frontend funkciókat Vue modulokként kell létrehoznod!

**Adatbázis-kezelés feladatrész (3 pont)**

1. Hozz létre a lokális SQL szerveren tura\_db néven adatbázist! Állítsd be az UTF8 kódolást alapértelmezettnek az adatbázis létrehozásánál! Az adatbázis alapértelmezett rendezési sorrendje a magyar szabályok szerinti legyen!
2. Az iménti adatbázisban készítsd el a következő táblákat a megadott mezőkkel. Egy túrára természetesen több jelentkezés is érkezik, de egy jelentkezés csak egy túrához kapcsolódik.

* turak
* id: automatikusan növekvő egész érték, a tábla kulcsa
* nev: szöveg
* tav: egész szám
* elerheto\_hely: egész szám
* jelentkezesek
* id: automatikusan növekvő egész érték, a tábla kulcsa
* tura\_id: egész szám, idegen kulcs a turak.id-ra
* email: szöveg
* letszam: egész szám

**Reszponzív viselkedésű weboldal feladatrész (10 pont)**

Az oldal megjelenését az alábbi leírás és a mellékelt mintaképek szerint készítsd! Ahol szükséges olvasd hozzá a Frontend és Backend feladatrészben leírtakat!

1. Helyezd el az oldalon a Bootstrap keretrendszer legfrissebb verziójának CDN linkjeit és állítsd be a nyelvét magyarra, karakterkódolását UTF8-ra!
2. A teljes oldal kapjon reszponzív margót, melynek mérete a képernyőszélesség csökkenésével szintén csökken, hogy aztán mobileszközön nézve már ne is látszódjon!
3. Az oldal tetején helyezz el közepes ablakszélességnél összecsukódó Navbar komponenst! Az brand-be a mellékelt logo.png kép kerüljön 150 képpontos szélességben! A navigációs sáv menüpontjai a következők legyenek:

* Túrák
* Jelentkezések
* Kapcsolat

1. A három menüpontra állítsd be a következő tulajdonságokat:

* Az alapértelmezettél nagyobb betűméret
* 5 képpontos „orange” alsó szegély
* 10 képpontos bal- és jobb margó

1. Az oldal és a navbar háttérszínének egyaránt az RGB(255, 238, 222) kódú színt állítsd!
2. A „Túrák” és a „Jelentkezések” menüpontokhoz a Frontend feladatrész alapján egy-egy Vue komponenst kell majd rendelned! Mindkét komponensre teljesüljön a mintán látható kétoszlopos elrendezés! Az oszlopok nagy (lg) ablakszélesség alatt ugorjanak egymás alá!
3. Hozd létre a mintán látható feliratokat, címkéket és beviteli komponenseket mindkét oldalon! A mintán nem látható, de a Jelentkezesek komponensen a túra kiválasztása lenyíló listából történik.
4. Mindkét oldalon legyen kötelező az összes mező kitöltése a rögzítéshez! Ezt tetszőleges módon megoldhatod.

**Frontend feladatrész (15 pont)**

1. Hozz létre Vue projektet tura néven és telepítsd a függőségeket! A projekten belül kötelezően készítsd el a Turak.vue és Jelentkezesek.vue komponenseket és importáld őket az App.vue-ban!
2. A Turak komponensen hozz létre tetszőleges néven Vue objektumot és kapcsold adattagjaihoz az oldalon lévő beviteli komponenseket!
3. A „Hozzáadás” gomb lenyomásakor a backend feladatrészben leírtak szerint POST típusú kéréssel küld el az imént létrehozott objektum adatait rögzítésre!
4. Hozz létre Vue változóként egy listát és gondoskodj róla, hogy tartalma az oldal betöltődésétől kezdve egyezzen meg a szerveren lévő túraadatokkal! A szerverről való lekéréshez GET request-et használj a lentebb leírtak szerint!
5. Az oldal jobb oldali oszlopában tetszőleges módon listázd a rögzített túrák nevét! A másik két információ egy „Részletek” feliratú gomb hatására váljon láthatóvá, vagy tűnjön el!
6. A Jelentkezesek nevű komponensen szinte mindenben hasonlóan járj el, mint a Turak esetén. Legyen itt is objektum és lista, valamint használd a POST és GET kéréseket a backend részben leírtak szerint! A komponens tetszőleges módon szerezze meg a rögíztett túrák adatait, és a „Túra kiválasztása” lenyíló listát az alapján töltse fel!
7. Amint a mintán láthatod, itt „Törlés” feliratú gombot kell készítened, mely a lentebb leírtak szerint (DELETE request) az azonosítója alapján törli a jelentkezést az adatbázisból és így az oldalról is!
8. Az oldalon tetszőleges módon (felugró ablakban, külön komponensben) jelenítsd meg a felhasználónak a requestek után, a szervertől kapott üzeneteket!

**Backend feladatrész (12 pont)**

1. Hozz létre Eloquent Modelleket a tura\_db tábláihoz a tanult Laravel konvenció alapján, határozd meg a Model attribútumait, és gondoskodj az adatbáziskapcsolat létrehozásáról!
2. Minden *túrával* kapcsolatos kérést a TuraController, minden *jelentkezéssel* kapcsolatos kérést a JelentkezesController kezeljen!
3. Hozz létre két API végpontot, melyek az alábbi címeken legyenek elérhetők: http://localhost/api/turak

http://localhost/api/jelentkezesek

1. A turak végpontot úgy alakítsd ki, hogy képes legyen GET és POST típusú kérések kiszolgálására a létrehozott Controller metódusainak segítségével!

POST kérés esetén:

A rögzítendő adatok JSON formátumban érkeznek a következő minta szerint:

{"nev":"Téli Mátra", "tav":"65", "elerheto\_hely":"300"}

A választ asszociatív tömbként küld vissza a minta alapján:

["uzenet" => "Túra rögzítve!"]

GET kérés esetén:

A teljes adattábla tartalmát JSON formátumú tömbként küldjük vissza, melynek elemei az adatbázis mezőneveivel megegyező adattagokkal rendelkező objektumok.

1. A jelentkezesek végpontot úgy alakítsd ki, hogy képes legyen GET, POST és DELETE típusú kérések kiszolgálására a létrehozott Controller metódusainak segítségével!

POST kérés esetén:

A rögzítendő adatok JSON formátumban érkeznek a következő minta szerint:

{"tura\_id":"1", "email":" winch.eszter@gmail.com", "letszam":"2"}

Gondoskodj arról, hogy rögzítés előtt megvizsgáld:

* a túra létezik-e,
* e-mail-ként történik a jelentkező „azonosítása”,
* van-e a jelentkező által igényelt létszám számára még elérhető hely!

Készíts/módosíts magyar nyelven íródó validációs hibaüzenetet!

A választ asszociatív tömbként küld vissza a minta alapján:

["uzenet" => "Sikeres jelentkezés!"]

GET kérés esetén:

A teljes adattábla tartalmát JSON formátumú tömbként küldjük vissza, melynek elemei az adatbázis mezőneveivel megegyező adattagokkal rendelkező objektumok.

DELETE kérés esetén:

A végpontnak paraméterezhetőnek kell lenni a törölni kívánt rekord azonosítójával:

http://localhost/api/jelentkezesek?id=1

A választ asszociatív tömbként küld vissza a minta alapján:

["uzenet" => "Jelentkezés törölve"]

**Minták:**

A képen szöveg, képernyőkép, Betűtípus, szám látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, képernyőkép, Betűtípus, tervezés látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.