

Gyakorlat vizsgán bemutatott egyéni feladat

Egy személyesen választott témakörre kell egy kis alkalmazást tervezni (csak néhány funkcionalitást, nem teljes honlapot). A feladatban szerver oldali funkcionalitásokat kell megvalósítani. Ajánlott egy olyan adatbázist felhasználni, amit az *Adatbázisok* tantárgynál vagy egyéb projektben készítettek, hogy ez ne kerüljön külön munkába.

A feladat alapján történik a szóbeli vizsga.

A következő megkötéseket kell teljesíteni:

A honlap nézeteinek megjelenítéséhez saját HTML/CSS megoldást ajánlott tervezni, a teljes oldal keret (*layout*) HTML része legyen *flexbox* vagy *grid* megjelenítés alapú.

Alternatív lehet kliens oldali könyvtárakat is használni (ebben az esetben ezek működését alaposan kell ismerni és kérdésekre válaszolni).

Adatbázis: 4-6 tábla ami leírja a feladatot + egy **felhasználókat nyilvántartó tábla**. Legalább egy tábla tartalmazzon idegen kulcsot is. Adatbázisként bármilyen relációs adatbázist (Oracle, MySQL, PostgreSQL, SQLite, stb.) lehet használni, akár SQL lekérdezésekkel, akár ORM (Object Relational Mapping) modult használva (Sequelize). Alternatív lehet NoSQL adatbázis is használni.

A feladatból használt adatbázis elérést ismerni kell és kérdésekre válaszolni.

Lehet az egyetemi Oracle adatbázist használni a Moodle oldalon megadott SQL csatolóval (*Oracle példák* fájlból).

Az **Express keretrendszert** kell használni szerver oldalon, tetszőlegesen kiválasztott kiegészítő modulokkal.

A feladatról és a megvalósított funkcionalitásokról **egy rövid leírást** kell készíteni (PDF dokumentum, 2-3 oldal), ami tartalmazza:

- Röviden a téma megfogalmazását (kb. 5-10 sor).
- Kik a felhasználók.
- A felhasználók funkcionalitásait egy-egy listában (mit tudnak elvégezni a honlapon).
- Mi az ami egyéni munka és lényeges megvalósítás.

A feladatnak ajánlott tartalmaznia funkcionalitásokat az alábbi használati esetekre

Felhasználó és szesszió kezelés

A honlap használjon olyan funkcionalitásokat, amelyet csak bejelentkezett felhasználó végezhet. Mindenképp kell tartalmazzon szesszió kezelést, és csak a bejelentkezett felhasználó által elért webcímeteket.

Az alkalmazás többet ér, ha meg tud különböztetni szerepköröket. Ehhez lehet a bemutatott alap Express szessziókezelést, vagy a kis alkalmazást (18_webapp) használni, alternatív lehet más Node modult használni, pl. JSON Web Token, Passport vagy más megoldást.

Megjelenítés

Ajánlott táblázatokat vagy listákat megjeleníteni, amelyeket a felhasználók megtekinthetnek, pl. termék vagy szolgáltatás lista.

A termékeket, szolgáltatásokat lehessen részletesen is megjeleníteni, a listákban amik böngészhetőek ne legyen benne minden adat (kattintás a lista elemére, megjelenik a részletes leírás).

Keresés

Legalább egy kereső funkcionalitást ajánlott megvalósítani, kiválasztott opciók (pl. termék típus *select* lista) vagy beírt szöveg alapján szöveges mezőkben. A módszer tetszőlegesen választható (SQL LIKE, *full text* vagy mező alapú keresés).

Űrlapon keresztüli adat beküldés a webszerverre

Legalább egy tábla adatait a webes felületről szükséges kezelni:

- új adatok bevitele,
- meglevő adatok módosítása,
- adatok törlése.

Az űrlapokat be lehet küldeni standard HTTP POST kéréssel (csak HTML, kliens oldali JavaScript nélkül) vagy lehet a Fetch API-t használni. Az űrlapok beküldéskor használt MIME típusait kell ismerni, kérdésekre bemutatni.

Állomány feltöltés

Egy, a témakörben használatos fájl típusra (pl. kép, szerződés vagy nyilatkozatot tartalmazó pdf, stb.) meg kell valósítani fájl feltöltést és annak felhasználását az alkalmazásban. A fájlok fel és letöltéséhez ki lehet indulni a 9. labor 3. feladatából feladatából, illetve a Fetch előadásból.

Kliens oldali hálózati kérés alkalmazása

Legalább egy funkcionalitást kliens oldali hálózati HTTP kéréssel kell megoldani, a Fetch API használatával.

Ilyen funkcionalitás lehet:

- egy űrlap bevitele,
- a HTML oldal egy részének frissítése (9. labor 1. feladat),
- egy keresés eredményének megjelenítése (9. labor 2. feladat),
- fájl feltöltés (9. labor 3. feladat).

Ezen kívül tervezhető bármilyen saját funkcionalitás.

A feladatot be kell mutatni, kérdésekre kell válaszolni, és esetenként egy, ott helyben rögtönzött kis feladattal ki kell tudni egészíteni az alkalmazást.

A használt modulokat, felhasználói interfészeket, adatbázis elérést biztosító csatolókat, stb. kell ismerni, és kérdésekre válaszolni.

A feladat bemutatása során nyilatkozni kell arról, mi a bemutatott alkalmazásban a saját munka. A feladatra a vizsgajegyben 2 pontot lehet kapni.

Mit jelent az „**ajánlott**” szó a feladat leírásban?

- Jó, ha van ilyen funkcionalitás.
- Nem gond, ha nincs.
- Legyen legalább 5 funkcionalitás a felsoroltakból.
- Minél több van, annál többet ér a feladat.