

FRONTEND PROGRAMOZÁS VIZSGAFELADAT

Központi Gyakorlati Vizsgafeladat: Motoros Garázs és Szerviznapló

A következő feladatban egy motoros garázs és járműkarbantartó alkalmazás (Motoros Garázs) kliensoldali felületét kell elkészítenie a megadott specifikáció alapján. Az alkalmazás segítségével a felhasználók áttekinthetik a garázsban lévő járműveket, új szervizbejegyzéseket rögzíthetnek, valamint megtekinthetik az egyes járművek egyedi szerviztörténetét.

A backend szerver elérése:

Az API szerver a `http://localhost:3000` címen érhető el. A fejlesztéshez és teszteléshez indítsa el a háttérszolgáltatást a vizsgakönyvtárban kiadott parancsok segítségével. A beépített API dokumentációt a `http://localhost:3000/docs` címen tanulmányozhatja.

1. Adatbázis és API végpontok specifikációja

A backend két fő entitást kezel: a járműveket (*vehicles*) és a hozzájuk kapcsolódó szerviznaplókat (*service_logs*). Egy járműhöz több szervizbejegyzés is tartozhat (1:N kapcsolat).

Elérhető REST API végpontok:

Végpont	Metódus	Leírás / Válaszstruktúra
GET /api/vehicles	GET	A garázsban tárolt összes jármű listája. Mezők: <i>id, brand, model, year, pictureUrl</i>
GET /api/vehicles/{id}/logs	GET	A megadott <i>id</i> -jú járműhöz tartozó szervizbejegyzések listája szűrt formában. Mezők: <i>id, vehicleId, date, component, description</i>
POST /api/logs	POST	Új szerviznapló bejegyzés rögzítése. Kötelező törzs: <i>{ vehicleId, date, component, description }</i>

2. Fejlesztési és Megvalósítási Feladatok

Készítsen egy három oldalból álló egyoldalas webalkalmazást (SPA) a választott keretrendszerben (Angular, React vagy Vue). Az oldalak közötti navigációt és adatátvitelt dinamikus **query paraméterek** segítségével kell megoldani (ne használjon komplex útvonal-paramétereket a részletes nézetekhez). Legördülő választólisták (select) használata a felületen nem megengedett.

Feladat 2.1: Kezdőlap – Járművek kártyás megjelenítése

Alakítsa ki az alkalmazás főoldalát, amely a teljes járműparkot mutatja be.

1. Töltse le a szerverről a járművek adatait a `/api/vehicles` végponton keresztül.
2. Az adatokat **kártyás elrendezésben** jelenítse meg, több sorban és oszlopban (reszponzív módon).
Táblázatos megjelenítés nem fogadható el!
3. Minden kártyán kötelezően szerepelnie kell:
 - A jármű fotójának (`pictureUrl`).
 - A jármű márkájának és típusának (pl. *Yamaha FZ6* vagy *Babetta 210*).
 - A gyártási évnek (pl. *2007*).
4. Minden egyes kártya alján helyezzen el **két akciógombot** a következő funkciókkal:
 - **„Szerelés rögzítése” gomb:** Navigáljon az űrlap oldalra, átadva a jármű egyedi azonosítóját query paraméterként (pl. `?vehicleId=1`).
 - **„Szerviztörténet” gomb:** Navigáljon a listázó oldalra, szintén query paraméterben átadva a jármű azonosítóját.

TERVEZETT FELÜLETI MINTA (KEZDŐLAP):

[YAMAHA FZ6 FOTÓ]	[BABETTA 210 FOTÓ]
<p>Yamaha FZ6 Évjárat: 2007</p> <p>Szerelés rögzítése Szerviztörténet</p>	<p>Babetta 210 Évjárat: 1990</p> <p>Szerelés rögzítése Szerviztörténet</p>

Feladat 2.2: Űrlap oldal – Új szervizbejegyzés rögzítése

Készítsen egy adatfelvételi felületet, ahol a kiválasztott járműhöz lehet új karbantartási eseményt rendelni.

1. Az oldal olvassa ki a böngésző címsorából a `vehicleId` query paramétert. Ez határozza meg, hogy melyik motorhoz rögzítünk adatot.
2. Hozzon létre egy HTML űrlapot az új rekord felvételéhez. Az űrlap az alábbi 3 beviteli mezőt tartalmazza:
 - **Dátum:** `input [type="date"]` mező a javítás idejének megadására.
 - **Alkatrész / Egység:** Egyszerű szöveges mező (pl. *Karburátor, Kuplung, Láncszett*).
 - **Leírás:** Szöveges terület (`textarea`) a elvégzett munka részletezésére.
3. A „Mentés” gombra kattintva az űrlap adatait küldje el a backend **POST** `/api/logs` végpontjára a megfelelő `vehicleId`-val együtt.

4. Hibakezelés és Visszajelzés:

- Sikeres rögzítés esetén az alkalmazás navigáljon vissza a Kezdőlapra.
- Ha a backend hibát jelez vissza, vagy a hálózati kérés sikertelen, a hibaüzenetet **felugró (toast) üzenetben, beúszó panelen vagy egy erre dedikált figyelmeztető div-ben** jelenítse meg a felületen. Globális hibaoldalra történő átirányítás nem megengedett!

Feladat 2.3: Szerviztörténet oldal – Listázás és szűrés

Készítsen egy letisztult, egyszerű felületet a karbantartási előzmények visszakövetésére.

1. Az oldal a bejövő query paraméter alapján kérje le a specifikus jármű szerviznaplóit a `/api/vehicles/{id}/logs` dinamikus végponton keresztül.
2. Az adatforrásból beérkező elemeket egy egyszerű **számozott vagy pontozott listában** (`` vagy ``) kell feldolgoznia és megjelenítenie tömb-leképezéssel (`.map()` használatával).
3. A lista minden elemében jól láthatóan formázva jelenítse meg a javított **alkatrész nevét**, a **dátumot** és a hozzá fűzött részletes **leírást**.
4. Helyezzen el az oldal alján egy „Vissza a garázsba” feliratú navigációs gombot, amely visszaviszi a felhasználót a kártyás főoldalra.

3. Értékelési kritériumok (Pontozási útmutató)

A feladat értékelése a sikeresen megvalósított funkciók és a kódminőség alapján történik az alábbi eloszlás szerint:

Értékelt modul / Funkcionalitás	Maximális pontszám
Kezdőlap kártyás dizájn: Adatok sikeres lekérése és reszponzív, több oszlopos kártyás elrendezésben történő dinamikus megjelenítése fotóval és adatokkal.	15 pont
Query paraméteres navigáció: A kártyákon elhelyezett akciógombok megfelelő query paraméterekkel (<code>?vehicleId=...</code>) indítják el a belső oldalváltásokat.	10 pont
Űrlap megvalósítás és POST kérés: A 3 kötelező beviteli mező összeállítása, az adatok validálása és sikeres beküldése a backend felé megfelelő adatstruktúrában.	15 pont
Toast / Beúszó hibakezelés: Szerverhiba vagy sikertelen hálózati kommunikáció esetén a hibák dinamikus kezelése és felugró/beúszó értesítésben való ábrázolása.	5 pont
Szerviztörténet listázása: Dinamikus végpontról történő adatlekérés query paraméter alapján, és a rekordok elemeinek tiszta leképezése számozott/pontozott listává.	15 pont
Kódminőség és architektúra: Keretrendszer-specifikus konvenciók betartása, tiszta állapotkezelés (state management) és áttekinthető komponens-struktúra.	10 pont
Összesen elérhető pontszám:	60 pont