

Formula 1 - Garage Market (fullstack)

Szomor Levente SZFT5

Dátum: 2026-02-27

Feladat: készítsen egy F1 témájú „garage marketplace” alkalmazást, ahol csapatok (teams) és garázs-hirdetések (garage_listings) kezelhetők.

A rendszer célja: a csapatok eladhatnak / meghirdethetnek alkatrészeket (aero, engine, tyre, gearbox), a felhasználó pedig tud listázni, szűrni, létrehozni és módosítani hirdetéseket.

Források (a csomagban)

- TeamSeeder_F1.php
- GarageListingSeeder_F1.php
- f1_fullstack/openapi.yaml (a fájl neve kötelezően: openapi.yaml)

1. Projektindítás és alapok

Hozzon létre Laravel backend projektet (API), és készítsen külön frontend alkalmazást (tanult technológia: Vue/React – ahogy eddig is).

A backend alap URL: <http://backend.vm1.test/>.

A frontend az API-t ezen a címen éri el (proxy vagy .env használata megengedett).

2. Adatmodell és migrációk

2.1 teams tábla

Mezők:

- id (PK)
- name VARCHAR(60) – kötelező
- base_city VARCHAR(40) – kötelező
- rating TINYINT – 1..5, kötelező
- created_at, updated_at

2.2 garage_listings tábla

Mezők:

- id (PK)
- team_id (FK → teams.id, ON DELETE CASCADE)
- category ENUM: aero, engine, tyre, gearbox
- title VARCHAR(80) – kötelező
- tag VARCHAR(20) – kötelező (pl. ECU, FRONTWING, TYRE-M)

- price INT - ≥ 1
- stock INT - ≥ 0
- is_active BOOLEAN - default true
- listed_from DATE - kötelező
- listed_to DATE - NULL (ha van, akkor \geq listed_from)
- created_at, updated_at

3. Eloquent modellek és kapcsolatok

Készítsen modelleket:

- Team - hasMany(GarageListing)
- GarageListing - belongsTo(Team)

A hirdetés listázásánál a Team adatok is jelenjenek meg (eager loading).

4. Seeding

Használja a kapott seedereket:

- TeamSeeder_F1.php
- GarageListingSeeder_F1.php

A DatabaseSeeder-ben hívja meg őket a megfelelő sorrendben (előbb Team, utána Listing).

5. Validáció és hibakezelés

Minden create/update végpont validáljon, hiba esetén 422 JSON választ adjon (Laravel default validation formátum elfogadott).

Kötelező validációk:

- Team: name (min 2, max 60), base_city (min 2, max 40), rating (integer 1..5).
- GarageListing: team_id (exists: teams,id), category (enum), title (min 3, max 80), tag (min 2, max 20).
- price (integer, min 1), stock (integer, min 0).
- listed_from (date), listed_to (nullable date, ha van: after_or_equal: listed_from).
- is_active (nullable boolean; ha nincs megadva, default true).

6. REST API végpontok

Az API-t a mellékelt openapi.yaml szerint készítse el. Minimum elvárt végpontok:

- GET /api/teams – csapatok listája
- POST /api/teams – új csapat
- GET /api/teams/{id} – csapat részletei
- PUT /api/teams/{id} – csapat módosítása
- DELETE /api/teams/{id} – csapat törlése (204)
- GET /api/garage-listings – hirdetések listája (szűrhető)
- POST /api/garage-listings – új hirdetés
- GET /api/garage-listings/{id} – hirdetés részletei
- PUT /api/garage-listings/{id} – hirdetés módosítása
- DELETE /api/garage-listings/{id} – hirdetés törlése (204)

6.1 Szűrés (GET /api/garage-listings)

A listázás támogassa az alábbi query paramétereket (tetszőleges kombinációban):

- category: aero/engine/tyre/gearbox
- tag: részleges keresés (LIKE, case-insensitive elfogadott)
- active: true/false – is_active alapján

6.2 Rendezés és lapozás (bonus, de ajánlott)

Ha megvalósítja:

- sort: price|listed_from|stock, direction: asc|desc
- page/per_page: lapozás (Laravel paginator is jó).

7. Swagger / OpenAPI megjelenítés

A backendben tegye elérhetővé a dokumentációt a projektben használt megoldással (swagger-ui / I5-swagger / redoc).

A specifikáció fájl neve kötelezően: openapi.yaml.

8. Frontend követelmények

A frontend egy egyszerű admin jellegű felület: listázás, szűrés, létrehozás, szerkesztés, törlés.

Használjon:

- routert (külön oldalak)
- központi állapotkezelést (store / context) vagy kompozíciós state-et
- form validációt (minimum: kötelező mezők, szám mezők)

8.1 Oldalak

- / - Hirdetések lista: táblázat vagy kártyák + szűrők (category dropdown, tag search, active switch).
- /listings/create - Új hirdetés űrlap (team választó, category, title, tag, price, stock, dates, active).
- /listings/:id - Szerkesztés: betöltés API-ból, mentés (PUT), törlés (DELETE) megerősítéssel.
- /teams - Csapat lista: név, bázisváros, rating; létrehozás gomb (modal vagy külön oldal).
- /teams/:id - Csapat részletek: csapat adatok + hirdetések listája kategória szerint csoportosítva.

8.2 UI/UX elvárások

- A szűrők változásakor frissüljön a lista (debounce a tag keresésnél opcionális).
- Törlés előtt confirm (modal/confirm()).
- Mentés után navigáljon vissza a listára és jelezzen sikert (toast/alert).
- Hiba esetén (422) jelenítse meg a validációs üzeneteket mezőnként.

9. Leadandó

- Backend Laravel projekt (migrációk, modellek, controllerek, seedelés, swagger)
- Frontend projekt
- Források: TeamSeeder_F1.php, GarageListingSeeder_F1.php, openapi.yaml