

Galéria alkalmazás (React)

Ebben a feladatban egy egyszerű képgaléria alkalmazást kell készítened React segítségével, ahol a felhasználók képeket tölthetnek fel, és értékelhetik is azokat csillagokkal. Az alkalmazás egy előre elkészített backenddel kommunikál REST API-n keresztül.

Hasznos linkek:

React dokumentáció: <https://react.dev/>

MUI (Material UI): <https://mui.com/>

Axios: <https://axios-http.com/docs/intro>

A frontend projekt már elő van készítve a proxy használatára (Vite `vite.config.js` fájlban), így az API hívásokat a `/api/...` útvonalon keresztül lehet elvégezni. Ha ez valamilyen okból nem működne, a backend CORS modullal is fel van készítve, ezért nyugodtan használhatod a teljes URL-t is, pl. `http://localhost:3333/api/images`.

A projektkönyvtárban elérhető lesz:

- néhány példa kép URL (images.txt), amelyet a fejlesztés során teszteléshez használhatsz,
- valamint egy vagy több képernyőkép, amely megmutatja, hogyan nézzen ki az elkészült alkalmazás végleges formája.

A backend minden POST kérés (kép feltöltés, értékelés) után a teljes képlistát visszaadja. Ez azért van így, hogy a változások könnyebben, egyértelműen újrenderelhetők legyenek a frontend oldalon.

Backend indítása:

1. Navigálj a `backend` mappába.
2. Futtasd az `npm install` parancsot.
3. Indítsd el a szerveret: `npm start`

A backend ezután a `http://localhost:3333` címen lesz elérhető.

Swagger API dokumentáció: <http://localhost:3333/api-docs>

Feladatok (összesen 15 pont)

1. Reszponzív kinézet (2p)

Az alkalmazás mobilon és asztali gépen is legyen jól használható. Használj rugalmas elrendezést (pl. Flexbox, Grid), hogy a galéria igazodjon a kijelző méretéhez.

2. Kód és projektstruktúra (2p)

Legyen a projekted tagolt és átlátható. A komponenseket, a logikát (pl. API hívások, állapotkezelés) érdemes külön fájlokba szervezni. Ha tudod, használj contextet a globális állapothoz. A kinézet és a működés maradjon külön, ne keveredjenek össze.

3. Modulok telepítése (1p)

Használd az npm-et a szükséges csomagok (pl. axios, MUI, react-icons) telepítésére. Ezeket integráld megfelelően a projektbe.

4. Képek lekérdezése (2p)

A /api/images végpontról kérd le a képeket a backendtől egy GET kérés segítségével. Az adatokat tárold state-ben, és gondoskodj arról, hogy az oldal betöltésekor automatikusan megjelenjenek.

5. Képek megjelenítése (3p)

A lekérdezett képeket jelenítsd meg galéria elrendezésben. Minden kép alatt jelenjen meg a jelenlegi értékelés csillag ikonokkal, jól olvasható formában.

6. Új kép beküldése (2p)

A felhasználó egy mezőbe URL-t írva küldhet be új képet. Az URL formátumát nem kell ellenőrizni, de az üres beküldést kezeld hibaüzenettel. A sikeres feltöltés után jelenjen meg az új kép is.

7. Értékelés küldése (2p)

Minden kép alatt legyen lehetőség csillagokra kattintva értékelni. A kiválasztott értéket küldd el a backendnek POST /api/rate formájában.

8. Animáció használata (1p)

Legalább egy látványos animáció vagy átmenet jelenjen meg az alkalmazásban (pl. gomb hover effektek, kártyák animációja, galéria megjelenés). Az animáció történhet CSS-sel vagy animációs könyvtárral is.

Leadás

Határidő: legkésőbb 17:15-ig.

A projektet tömörített formában (ZIP) kell leadni. A ``node_modules`` mappákat mind a frontend, mind a backend mappából törölni kell a tömörítés előtt.

A feltöltött fájlt kicsomagolás után újra próbáld ki, és indítsd el a projektet. Ezzel biztosíthatod, hogy valóban működő, értékelhető állapotban adod le a feladatod.