

# Egyetemi tárgyak követelményeit számontartó alkalmazás

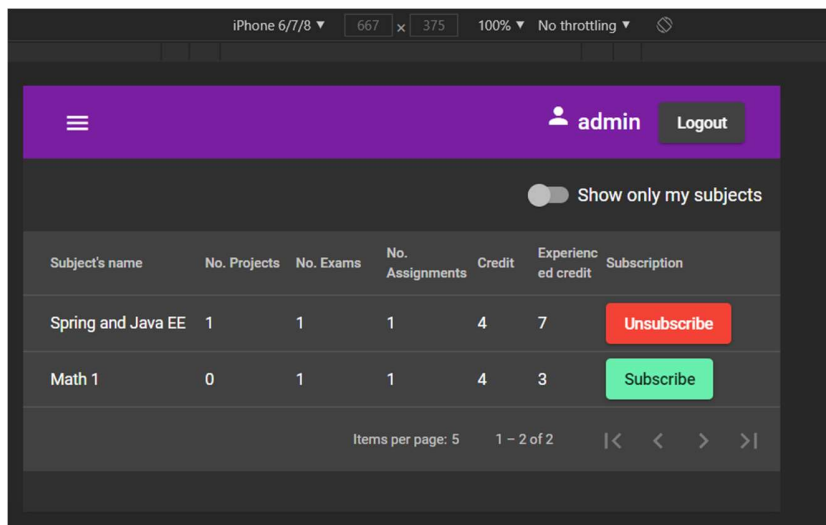
Kliensoldali technológiák házi feladat – Készítette: Kovács Bence Zoltán (S30A6K)

## Összefoglaló

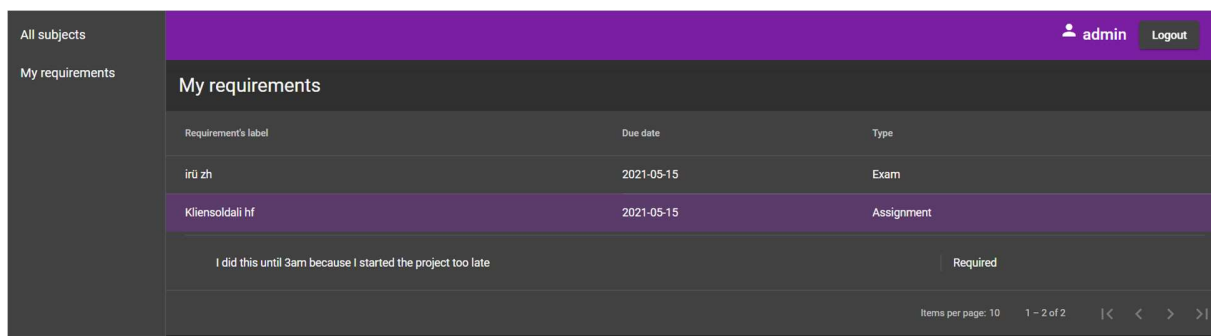
Ez az alkalmazás lehetővé teszi, hogy a felhasználó könnyedén számontarthassa a tárgyai egyes követelményeit.

A fejlesztői verzióban az alkalmazás a localhost:4200 oldalon érhető el indítás után. A kezdő oldalon láthatóak a tárgyak amik megtalálhatóak az egyetemen. Ha regisztrál a felhasználó az oldalra akkor lehetősége nyílik “feliratkozni” az egyes tárgyakra. A feliratkozott tárgyak követelményeit egy külön nézetben tudja egy oldalon átlátni, így könnyen

számontarthatóvá téve hogy melyik tárgyból mi van még hátra. De ha a felhasználó nem szeretne regisztrálni az oldalra, akkor is elérhető számára a tárgyak adatai, gyakorlatilag a szűrési funkció veszik el számára. Az adatokat egy általam írt backend biztosítja – ezt az API-nak szóló résznél részletezem majd. A frontend megvalósítására Angular-t használtam, illetve igyekeztem a Material UI témáját tartani.



A képen a kezdőoldal látható mobilnézetben



A képen az admin felhasználó követelményei látható desktop nézetben

Az autentikációhoz jwt technológiát használok, amit a local storageben tárolok el. Az oldalon való navigációhoz a material ui egy schematic-ját használtam fel ezzel is elősegítve a szép design-t. Igyekeztem megfelelő mennyiségben kommentelni, de kevés helyen láttam értelmét. Szerintem a kód jól olvasható, ahol bonyolultabb a kód ott használok kommentet.

## Komponensek

- Login: Az alkalmazás bejelentkezési pontja
- Register: Az alkalmazás regisztrálási pontja
- Assignment: Egy beadandó adatainak megjelenítésére szolgáló komponens
- Exam: Egy vizsga/zh adatainak megjelenítésére szolgáló komponens
- Project: Egy projekt feladat megjelenítésére szolgáló komponens. Egy projektnek több beadandója lehet, így a projektek megjelenítésénél újrahasznosítottam a beadandó komponens
- Subject: Egy tárgy megjelenítésére szolgáló komponens. Mivel egy tárgynak mindhárom fajta követelménye lehet, illetve ezeken túl is vannak hozzátartozó információk, így az előbb említett három komponens mind újrahasznosítottam ebben a komponensben
- SubjectsTable: A tárgyak megjelenítésére szolgáló komponens. Egy sorára kattintva megjeleníti az adott tárgy minden információját
- MyRequirementsTable: A követelmények megjelenítésére szolgáló komponens

## Szolgáltatások

- SubjectService: A SubjectsTablet látja el adatokkal httpn keresztül
- RequirementService: A MyRequirementsTablet látja el adatokkal httpn keresztül. Gyakorlatilag a SubjectService egyik funkcióját bővíti ki ez a szolgáltatás
- AuthenticationService: Az autentikáláshoz szükséges műveleteket biztosítja, például bejelentkezés, jelenlegi felhasználó adatai, regisztrálás

- **AuthGuardService:** Az Angular routerjének működéséhez szükséges szolgáltatás. Mivel ahhoz hogy tárolni tudjuk azt az adatot backenden hogy melyik felhasználó mire van feliratkozva, így nem lenne értelme ha egy nem bejelentkezett felhasználó be tudna lépni a "Saját követelményeim nézetbe".
- **Jwt-interceptor, error-interceptor:** Itt nem vagyok benne biztos hogy ez servicenek számít-e én úgy döntöttem hogy nem. Ezek feladata az egyes http kérések "lehallgatása" -> ha be vagyunk jelentkezve akkor hozzáadjuk a jwt-t az authorization headerbe, illetve ha olyan http kérést indítunk amire nincs jogosultságunk akkor átirányítunk a /loginra.

## Felhasznált api

Az adatokat egy általam írt Java Spring backend biztosítja. A következő entitásokat hoztam létre a probléma megoldására:

- **Subject:**
  - /subjects: Visszaadja az összes tárgyat
  - /subjects/subscribed: Visszaadja a feliratkozott tárgyakat ha van jwt
  - /subscribe/{id}: Feliratkozik az adott idjű tárgyra ha van jwt
  - /unsubscribe/{id}: Leiratkozik az adott idjű tárgyról ha van jwt
- **User**
  - /login: felhasználónév és jelszó postolásával megkapjuk a jwt-t
  - /register: felhasználónév és jelszó postolásával létrehozza a kért felhasználót és rögtön meg is kapjuk a jwt-t
- Exam, Assignment, Project

Egy tantárgyhoz tartozhatnak vizsgák, beadandók és projektek, egy projekthez tartozhatnak beadandók illetve felhasználókhöz tartozhatnak feliratkozott tárgyak. A bejelentkezéshez és a regisztráláshoz a felhasználónevet és a jelszavat kell megadni, majd minden autorizálandó kérésnél tartalmaznia kell a headernek a jwt-t. A megvalósítás követi a REST api alapelveit.

## Források

Az implementálás során gyakran segítségül vettem az Angular Material UI hivatalos weboldalát: [material.angular.io](https://material.angular.io), és a <https://bezkoder.com/angular-spring-boot-jwt-auth/> oldalt az autentikációhoz.

