

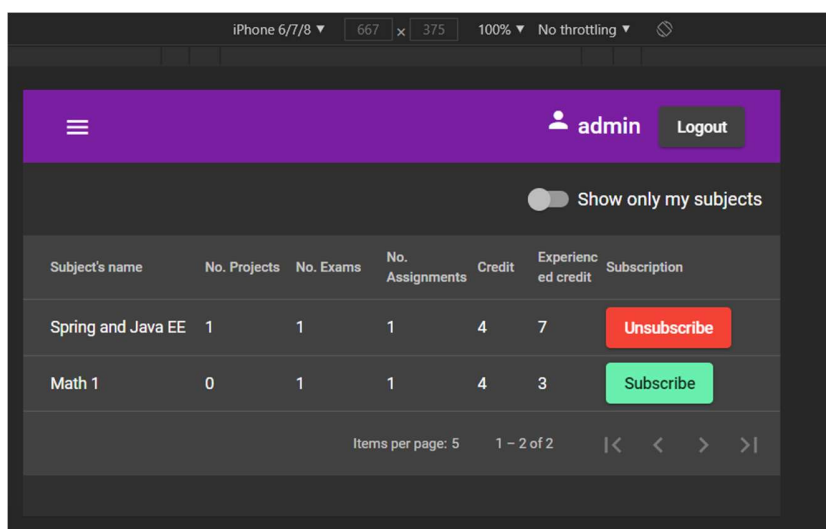
Egyetemi tárgyak követelményeit számontartó alkalmazás

Java alapú webes technológiák házi feladat – Készítette: Kovács Bence Zoltán (S3OA6K)

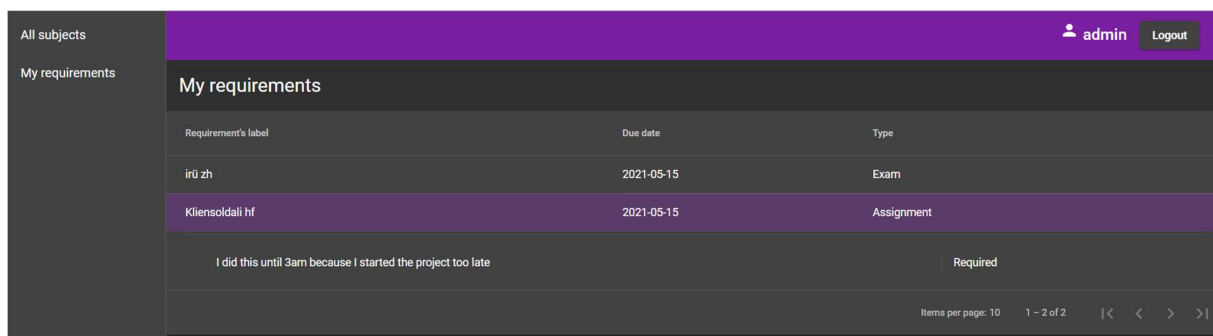
Összefoglaló

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

A kezdő oldalon láthatóak a tárgyak amik megtalálhatóak az egyetemen. Ha regisztrál a felhasználó az oldalra akkor lehetősége nyílik "feliratkozni" az egyes tárgyra. 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.

Api: /api

- Subject /subjects
 - GET
 - Visszaadja az összes tárgyat
 - /id: Visszaadja az adott idjű tárgyat
 - /subscribed: Visszaadja a feliratkozott tárgyakat ha van jwt
 - POST
 - Létrehoz egy tárgyat
 - /subscribe/{id}: Feliratkozik az adott idjű tárgyra ha van jwt
 - /unsubscribe/{id}: Leiratkozik az adott idjű tárgyról ha van jwt
 - PATCH
 - /id: Frissíti az adott idjű tárgyat
 - DELETE
 - /id: Törli az adott idjű tárgyat
- User
 - POST
 - /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
 - GET
 - exams/id: Visszaadja az adott idjű vizsgát/zht
 - POST
 - /subjects/{subjectID}/exams: Létrehoz egy vizsgát/zht az adott tárgyhöz
 - PATCH
 - /exams/id: Frissíti az adott idjű vizsgát/zht
 - DELETE
 - /id: Törli az adott idjű vizsgát/zht
- Assignment
 - GET
 - assignments/id: Visszaadja az adott idjű beadandót
 - POST
 - /subjects/{subjectID}/assignments: Létrehoz egy beadandót az adott tárgyhöz
 - PATCH
 - /assignments/id: Frissíti az adott idjű beadandót
 - DELETE
 - /id: Törli az adott idjű beadandót
- Project
 - GET
 - projects/id: Visszaadja az adott idjű projektet
 - POST
 - /subjects/{subjectID}/projects: Létrehoz egy projektet az adott tárgyhöz
 - /subjects/{subjectID}/projects/{projectId}/assignments: Létrehoz egy project assignmentet az adott tárgyhöz az adott projekttel.
 - PATCH
 - /projects/id: Frissíti az adott idjű projektet
 - DELETE
 - /id: Törli az adott idjű projektet

Entitások

- Assignment

```
private Boolean optional;  
private Date deadline;  
@Size(max = 255)  
private String description;  
@Size(max = 31)  
private String label;
```

-
- Tartozik egy tárgyhoz (ManyToOne)

- Exam

```
private Date dueDate;  
@Size(max = 255)  
private String experiences;  
@Size(max = 255)  
@Column(length = 63)  
private String scoring;  
private Boolean resit;  
@Size(max = 31)  
private String label;
```

-
- Tartozik egy tárgyhoz (ManyToOne)

- ProjectAssignment

```
private Boolean optional;  
private Date deadline;  
@Size(max = 255)  
private String description;  
@Size(max = 31)  
private String label;
```

-
- Tartozik egy Projecthez, és egy tárgyhoz (ManyToOne)

- Project

```
@Size(max = 255)  
private String experiences;
```

-
- Tartozhatnak hozzá project assignmentek (OneToMany)
- Tartozik egy tárgyhoz (ManyToOne)

- Subject

```
@Size(min = 4, max = 100)
private String name;
@Size(max = 511)
private String description;
@Size(max = 63)
private String scoring;
@Min(1) @Max(30)
private Short officialCredit;
@Min(1) @Max(30)
private Short experiencedCredit;
```

-
- Tartozhatnak hozzá vizsgák, projektek és beadandók (OneToMany)
- Feliratkozhatnak rá felhasználók (ManyToMany)

- User

```
@Size(min=4, max = 20)
private String username;
@Size(min = 4, max = 50)
private String password;
```

-
- Tartoznak hozzá szerepkörök (ADMIN, USER) (ManyToMany)
- Tartozhatnak hozzá feliratkozott tárgyak (ManyToMany)

Minden autorizálandó kérsnél tartalmaznia kell a headernek a jwt-t. A megvalósítás követi a REST api alapelveit. A readonly api endpointokhoz nem szükséges autorizálni, viszont a feliratkozott tárgyakat nem lehet lekérni autentikáció nélkül, hiszen a userhez kötjük a feliratkozásokat.

Források

Az implementálás során gyakran segítségül vettem az Angular Material UI hivatalos weboldalát: material.angular.io, és a <https://bezkoder.com/angular-spring-boot-jwt-auth/> oldalt az autentikációhoz. Továbbá felhasználtam az angular+rest gyakorlat anyagát is például a jelenlegi felhasználó meghatározására.