

Programarea aplicatilor WEB

Music Shop

Giurge Diana

November 2023

Contents

1	Scop	3
2	Obiective	3
3	Arhitectura si Use Cases	4
3.1	Arhitectura	4
3.2	Use Cases	5
3.3	Baza de date	6
3.4	Integrare security - JWT (JSON Web Token)	7
4	Implementare frontend/backend	8
5	Testare	11
6	Prezentare site web	13
7	Concluzie	13

1 Scop

Scopul acestui proiect este crearea unui music shop online care comercializează CD-uri, DVD-uri și obiecte muzicale, printre care și obiecte de colecție. Scopul primar este să dezvolt o platformă web care să ofere utilizatorilor posibilitatea de a căuta, vizualiza și achiziționa produse muzicale.

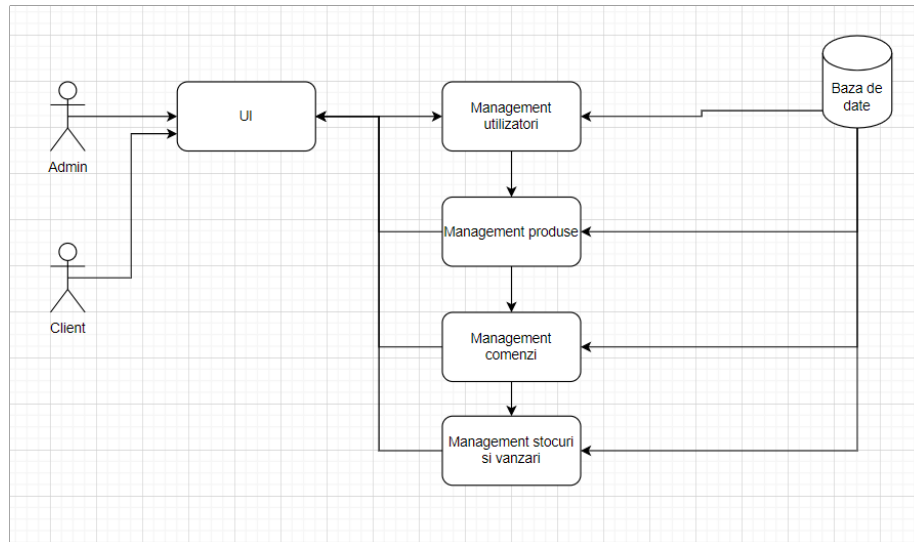
2 Obiective

Obiectivele aplicației sunt:

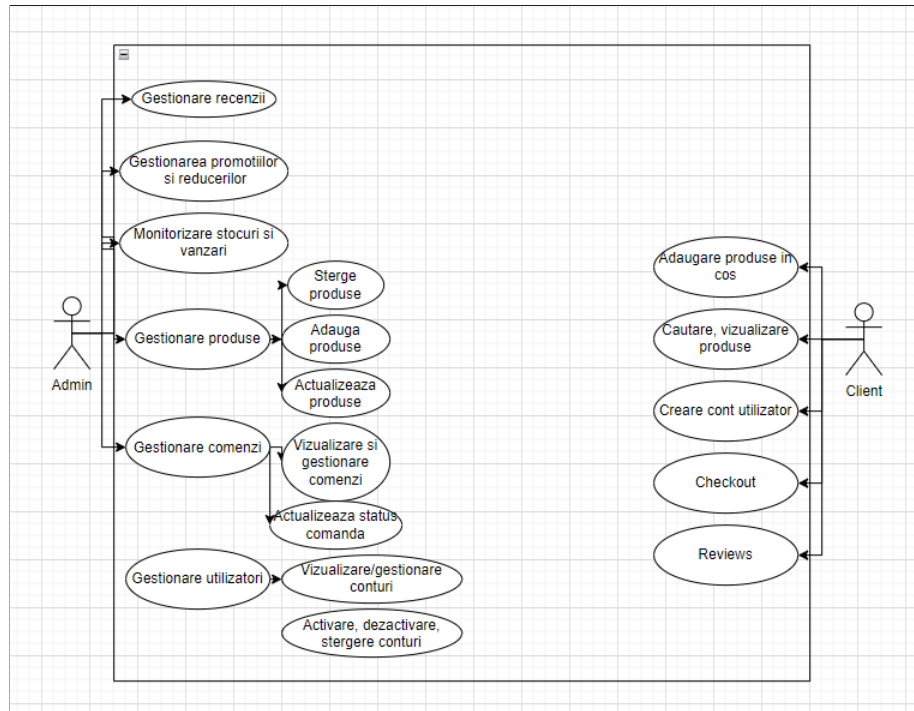
1. Crearea și implementarea unui site web funcțional și atractiv care să ofere o experiență de cumpărare plăcută și simplă pentru utilizatori.
2. Crearea unui site web, mai exact a unui online music cu SpringBoot pentru BackEnd, Angular pentru FrontEnd și MySQL ca baza de date.
3. Introducerea unei game variate și actualizate de CD-uri, DVD-uri, vinil-uri și alte obiecte muzicale, incluzând atât titluri populare, cât și ediții limitate sau produse rare, de colecție.
4. Asigurarea unei experiențe online fluide și plăcute pentru utilizatori prin intermediul unei navigări ușoare, a unor informații detaliate despre produse și a unor opțiuni de plată și de livrare convenabile.
5. Adaugarea unui cos de cumparaturi simplu cu un flux de plata simplificat și securizat și opțiuni de plata multiple
6. Implementarea unui sistem de autentificare securizat care să permită utilizatorilor să-și creeze un cont personal și să-și gestioneze datele personale în siguranță.

3 Arhitectura si Use Cases

3.1 Arhitectura



3.2 Use Cases



Use-Case-uri pentru Administratori:

1. Gestionarea produselor: adăugare, ștergere, actualizare;
2. Gestionare comenzi: vizualizare și gestionare comenzi, actualizare status comandă;
3. Gestionare utilizatori: vizualizare/gestionare conturi, activare/dezactivare/ștergere conturi;
4. Monitorizare stocuri și vânzări;
5. Gestionarea promoțiilor și reducerilor;
6. Gestionare reviews

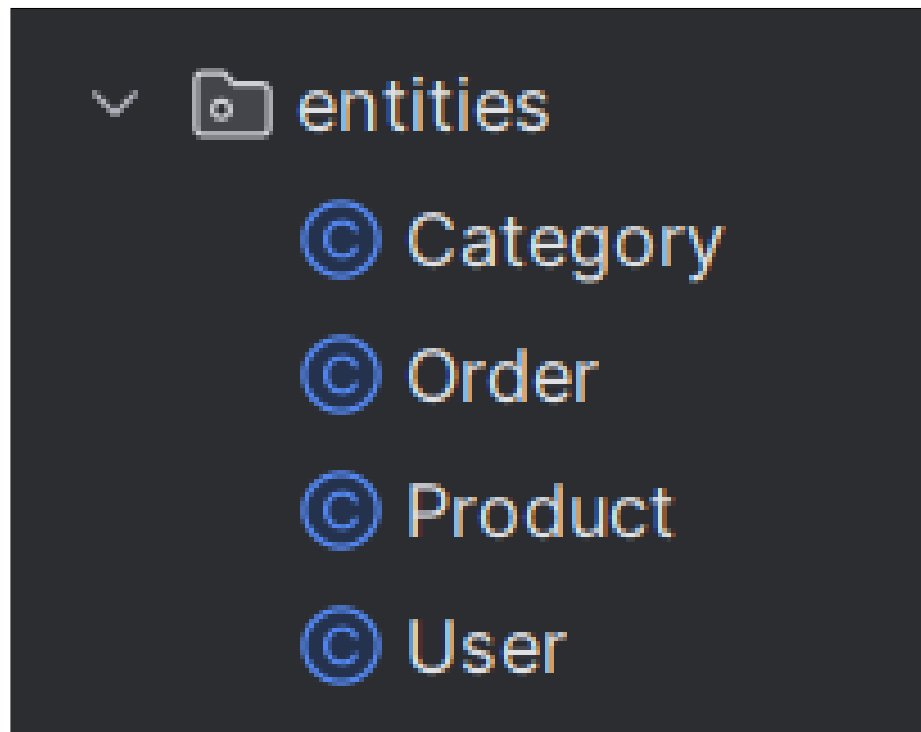
Use-Case-uri pentru clienți (utilizatori obișnuiți):

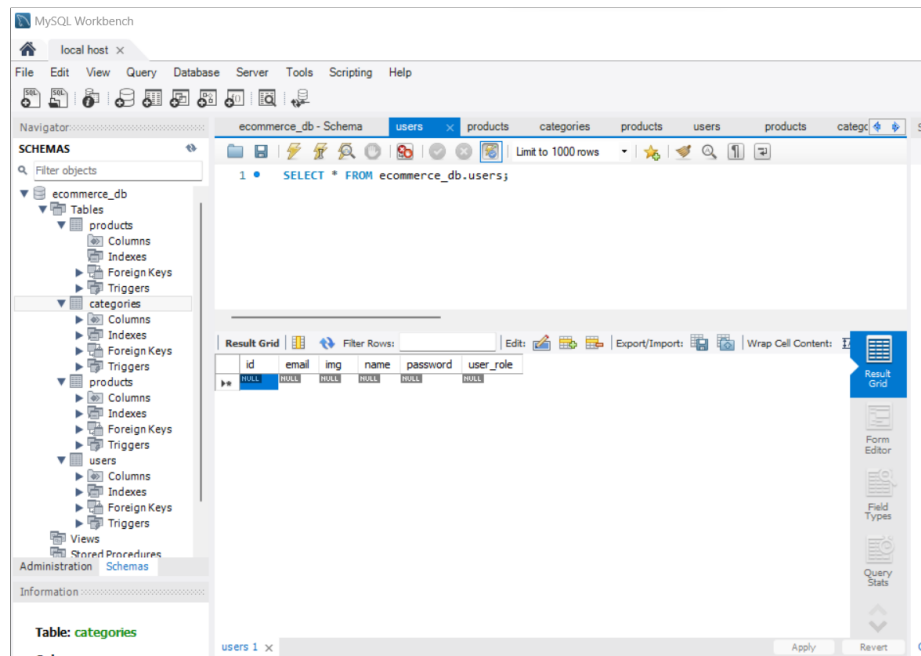
1. Creare cont utilizator;
2. Căutare și vizualizare produse;
3. Adăugare produse în coș;
4. Checkout;
5. Partea de reviews

3.3 Baza de date

Ca bază de date am ales să folosesc MySQL. Aceasta conține următoarele tabele:

1. users: id, email, img, name, password, user role;
2. products: id, name, description, price, image, category;
3. orders: id, order, name, adress, nr tel, products, price, order status;
4. categories: id, name, description





3.4 Integrare security - JWT (JSON Web Token)

Pentru security am ales să folosesc JWT. Sistemul de autentificare și autorizare din cadrul magazinului online de muzică se bazează pe standardul JWT. L-am folosit pentru a asigura securitatea, autenticitatea și autorizarea în cadrul aplicației mele.

Procesul de autentificare începe odată cu furnizarea de către utilizator a credențialelor valide: email și parola. Serverul autentifică utilizatorul și, în caz de succes, emite un token JWT, care va fi utilizat pentru a confirma identitatea acestuia în conexările anterioare pe site. La primirea unui token JWT, serverul validează și decodifică informațiile. Aceasta confirmă autenticitatea tokenului și furnizează informații despre utilizator, inclusiv rolurile și permisiunile acestuia.

În următoarea imagine este prezentată clasa `JwtRequestFilter`, care extinde clasa `OncePerRequestFilter`, utilizată pentru a intercepta și verifica cererile HTTP pentru a asigura că sunt autorizate, că acestea conțin un token valid:

```

@Autowired
private JwtUtil jwtUtil;

no usages

@Override
protected void doFilterInternal(HttpServletRequest request, HttpServletResponse response, FilterChain filterChain) throws ServletException, IOException {
    String authHeader = request.getHeader("Authorization");
    String token = null;
    String username = null;

    if (authHeader != null && authHeader.startsWith("Bearer ")){
        token = authHeader.substring(7);
        username = jwtUtil.extractUsername(token);
    }

    if (username != null && SecurityContextHolder.getContext().getAuthentication() == null){
        UserDetails userDetails = userService.loadUserByUsername(username);

        if (jwtUtil.validateToken(token, userDetails)){
            UsernamePasswordAuthenticationToken authToken = new UsernamePasswordAuthenticationToken(userDetails, null, userDetails.getAuthorities());
            authToken.setDetails(new WebAuthenticationDetailsSource().buildDetails(request));
            SecurityContextHolder.getContext().setAuthentication(authToken);
        }
    }
    filterChain.doFilter(request, response);
}
}

```


4 Implementare frontend/backend














Pentru frontend am ales să folosesc Angular, dar cu acesta am avut probleme, am avut o eroare pe care nu am reușit să o rezolv, pentru niciun proiect pe care l-am început. Am avut următoarea eroare legată de bara de navigare:

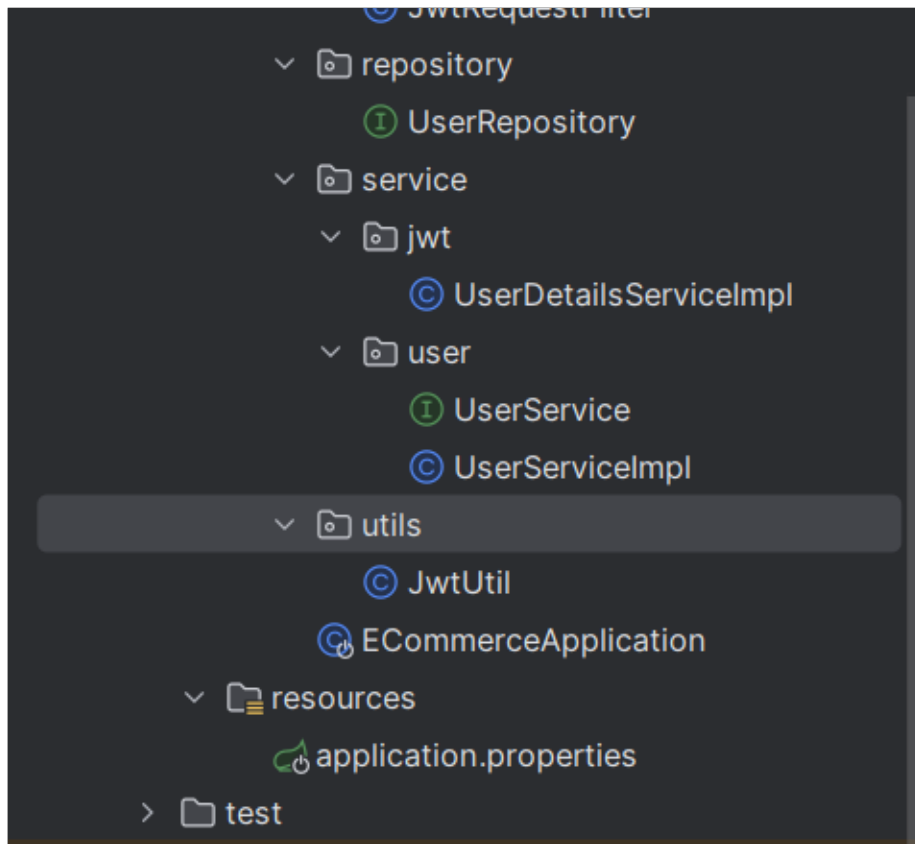
Eroare Angular - 'NG8001: 'app-navbar' is not a known element'

Orice am încercat, toate posibilitățile, chiar și să reinstalez componentele Angular, nu a mers nimic.

Pentru backend am folosit Spring Boot, versiunea 3.0.6. În proiectul meu am avut următoarele pachete și clase, după cum se poate vedea în această imagine:

▼  **e-commerce** C:\Facultate\e-commerce

- >  .idea
- >  .mvn
- ▼  src
 - ▼  main
 - ▼  java
 - ▼  com.example.ecommerce
 - ▼  configurataion
 - © CorsConfiguration
 - © WebSecurityConfiguration
 - ▼  controller
 - © AuthenticationController
 - © SignupController
 - ▼  dto
 - © AuthenticationRequest
 - © AuthenticationResponse
 - © SignupDTO
 - © UserDTO
 - ▼  entities
 - © Category
 - © Order
 - © Product
 - © User
 - ▼  enums
 - © UserRole
 - ▼  filters
 - © ⁹JwtRequestFilter
 - ▼  repository



Proiectul a fost conectat la o bază de date așa cum se vede în următoarea imagine:

```
#data source properties
spring.datasource.url=jdbc:mysql://localhost:3306/ecommerce_db?useUnicode=true&characterEncoding=UTF-8&useSSL=false
spring.datasource.username=root
spring.datasource.password=DianaLovevXkookie7#
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
```

Am avut următoarele dependențe utilizate:

```
19 <dependencies>
20 <dependency>
21 <groupId>org.springframework.boot</groupId>
22 <artifactId>spring-boot-starter-security</artifactId>
23 </dependency>
24 <dependency>
25 <groupId>org.springframework.boot</groupId>
26 <artifactId>spring-boot-starter</artifactId>
27 </dependency>
28 <dependency>
29 <groupId>org.springframework.boot</groupId>
30 <artifactId>spring-boot-starter-test</artifactId>
31 </dependency>
32 <dependency>
33 <groupId>com.mysql</groupId>
34 <artifactId>mysql-connector-j</artifactId>
35 <scope>runtime</scope>
36 </dependency>
37 <dependency>
38 <groupId>org.projectlombok</groupId>
39 <artifactId>lombok</artifactId>
40 <optional>true</optional>
41 </dependency>
42 <dependency>
43 <groupId>org.springframework.boot</groupId>
44 <artifactId>spring-boot-starter-test</artifactId>
45 <scope>test</scope>
46 </dependency>
47 <dependency>
48 <groupId>io.jsonwebtoken</groupId>
49 <artifactId>jjwt-api</artifactId>
50 <version>0.11.5</version>
51 </dependency>
52 <dependency>
53 <groupId>io.jsonwebtoken</groupId>
54 <artifactId>jjwt-impl</artifactId>
55 <version>0.11.5</version>
56 </dependency>
57 <dependency>
58 <groupId>io.jsonwebtoken</groupId>
59 <artifactId>jjwt-jackson</artifactId>
60 <version>0.11.5</version>
61 </dependency>
62 <dependency>
63 <groupId>org.springframework.security</groupId>
64 <artifactId>spring-security-test</artifactId>
65 <scope>test</scope>
66 </dependency>
67 <dependency>
68 <groupId>io.jsonwebtoken</groupId>
69 <artifactId>jjwt</artifactId>
70 <version>0.9.1</version>
71 </dependency>
72 <dependency>
73 <groupId>com.vaadin.external.google</groupId>
74 <artifactId>android-json</artifactId>
75 <version>0.0.20131108.vaadin1</version>
76 </dependency>
77 </dependencies>
```

5 Testare

Am testat pe parcurs pașii pe care i-am urmat pentru a implementa site-ul dorit. Din păcate nu am făcut screenshot-uri la toți pașii și cu tot ce am ramas până în momentul când am avut eroarea sunt următoarele

Am testat in Postman să aflu dacă funcțiile pentru conectarea la baza de date și pentru JWT-ului merg:

The image displays two screenshots of the Postman application interface, showing a REST client request and its response.

Top Screenshot:

- URL:** `http://localhost:8080/sign-up`
- Method:** `POST`
- Body (raw):**

```
{  "name": "user",  "email": "user@test.com",  "password": "user"}
```
- Status:** 201 Created, Time: 980 ms, Size: 407 B
- Body (JSON):**

```
{  "id": null,  "name": null,  "email": null,  "password": null,  "userRole": null}
```

Bottom Screenshot:

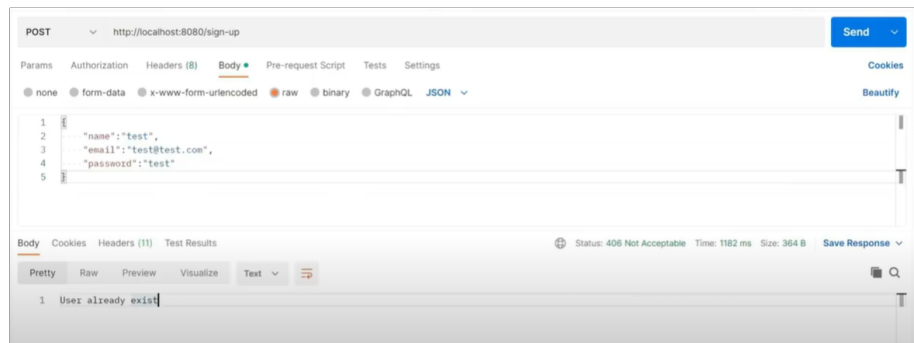
- URL:** `http://localhost:8080/sign-up`
- Method:** `POST`
- Body (raw):**

```
{  "name": "test",  "email": "test@test.com",  "password": "test"}
```
- Status:** 201 Created, Time: 857 ms, Size: 417 B
- Body (JSON):**

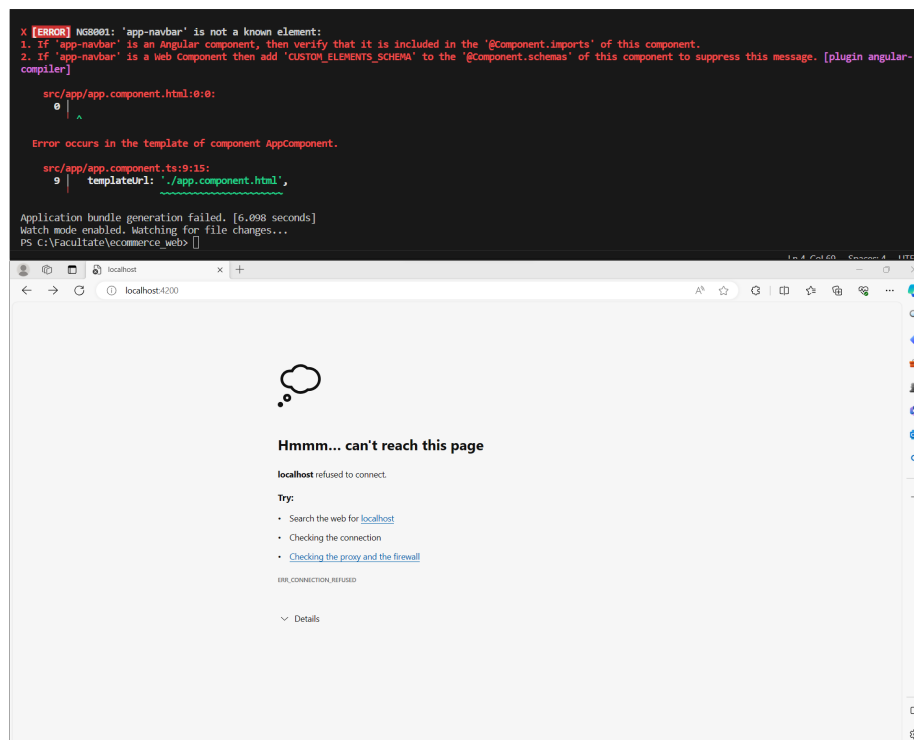
```
{  "id": 4,  "name": "test",  "email": "test@test.com",  "password": null,  "userRole": null}
```

Result Grid (Bottom Screenshot):

id	email	img	name	password	user_role
3	user@test.com	null	user	\$2a\$10\$QGxZlwC600bn3.nKoetbu0EITbSDUq...	1
4	test@test.com	null	test	null	1



6 Prezentare site web



7 Concluzie

Am încercat să creez un site web, dar nu am reușit să îmi duc proiectul până la capăt din cauza unei erori pe care nu am reușit să o rezolv. Pentru magazinul de muzică online am avut: BackEnd - SpringBoot, FrontEnd - Angular, baza de date relațională: MySQL, JPA, Dependency Injection si JWT.