

2021-2022

**Département Mathématiques et Informatique
GLSID 2 – S4**

Architecture JEE et Middlewares

Rapport TP
Spring MVC Thymeleaf

Préparé par : Fatima Zahra HASBI
Encadré par : Mr Mohamed YOUSSEFI

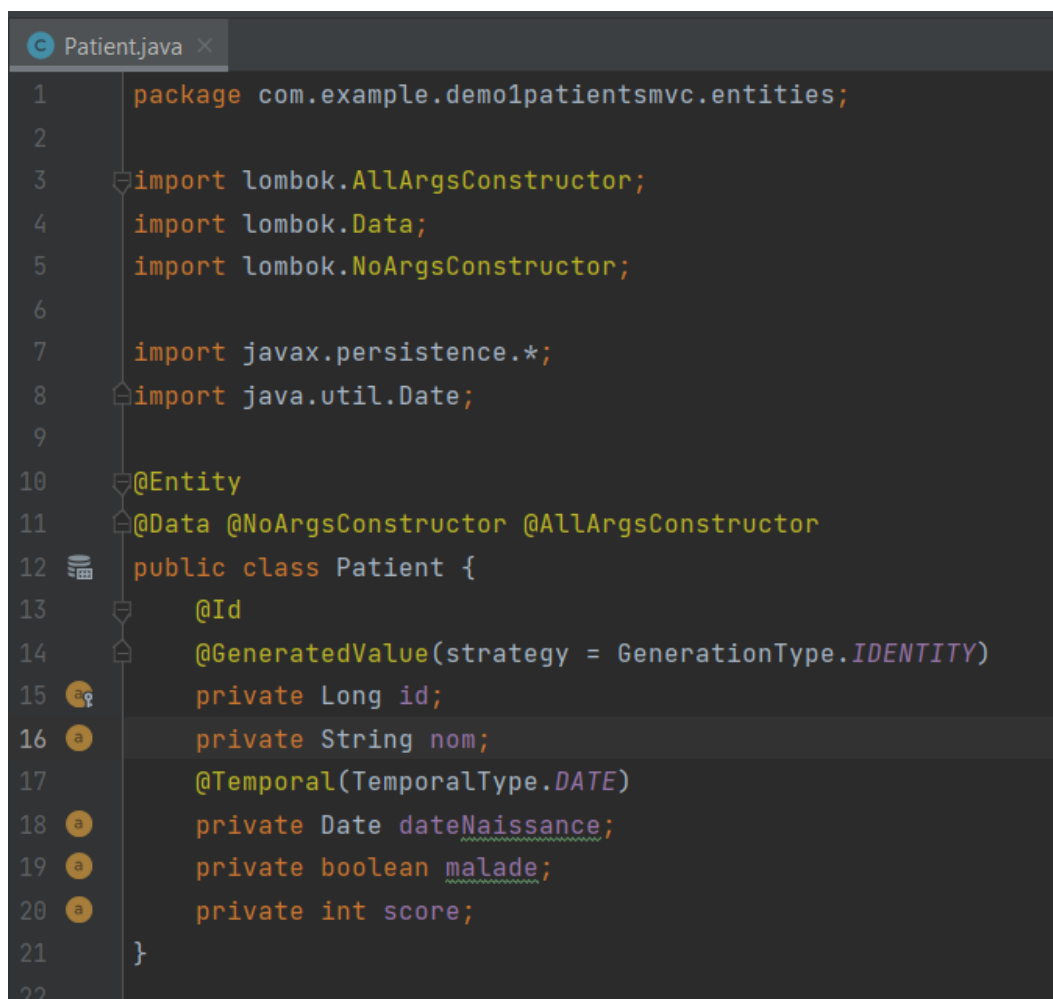
Introduction

Ce tp consiste à créer une application Web JEE basée sur Spring MVC, Thymeleaf et Spring Data JPA qui permet de gérer les patients. L'application doit permettre les fonctionnalités suivantes :

- Afficher les patients
- Faire la pagination
- Chercher les patients
- Supprimer un patient
- Configurer l'authentification avec SpringSecurity
- Ajouter les rôles ADMIN et USER

Réalisation

Dans le dossier entities on crée la classe patient



```
1 package com.example.demo1patientsmvc.entities;
2
3 import lombok.AllArgsConstructor;
4 import lombok.Data;
5 import lombok.NoArgsConstructor;
6
7 import javax.persistence.*;
8 import java.util.Date;
9
10 @Entity
11 @Data @NoArgsConstructor @AllArgsConstructor
12 public class Patient {
13     @Id
14     @GeneratedValue(strategy = GenerationType.IDENTITY)
15     private Long id;
16     private String nom;
17     @Temporal(TemporalType.DATE)
18     private Date dateNaissance;
19     private boolean malade;
20     private int score;
21 }
22
```

Dans le dossier repositories on crée l'interface PatientRepository

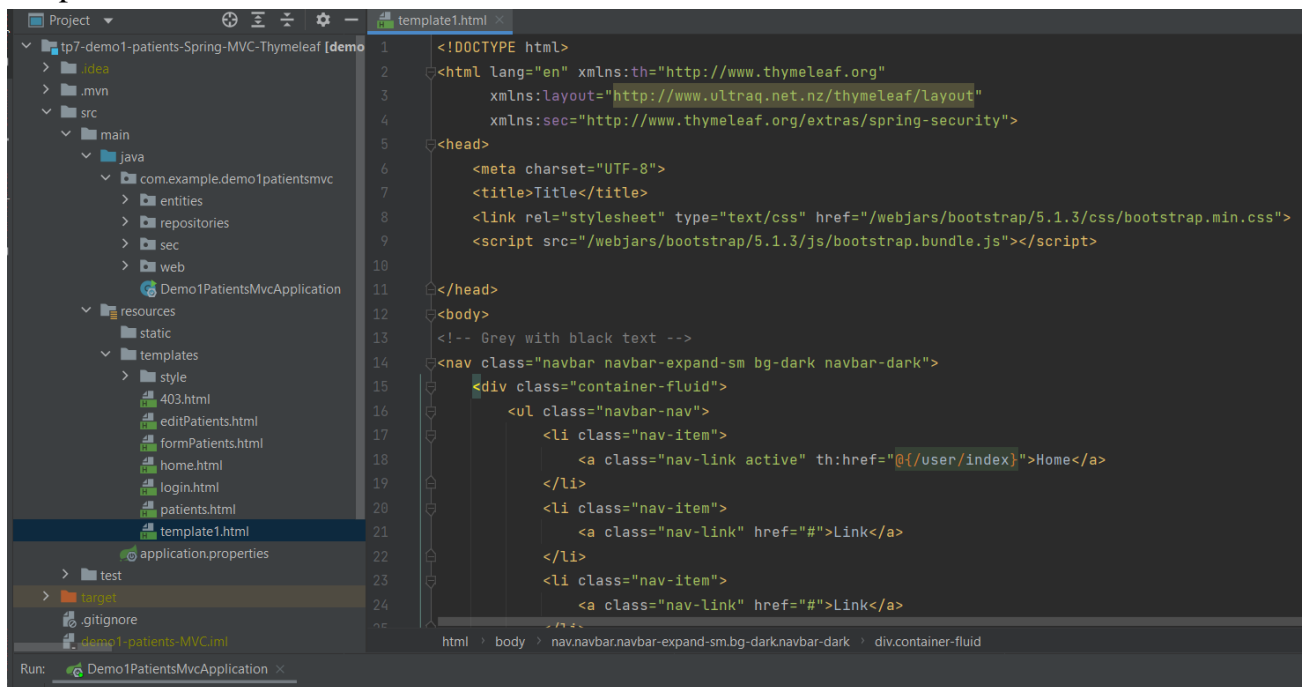
```
1 package com.example.demo1patientsmvc.repositories;
2
3 import com.example.demo1patientsmvc.entities.Patient;
4 import org.springframework.data.domain.Page;
5 import org.springframework.data.domain.Pageable;
6 import org.springframework.data.jpa.repository.JpaRepository;
7 import org.springframework.data.jpa.repository.Query;
8 import org.springframework.data.repository.query.Param;
9
10 import java.util.Date;
11 import java.util.List;
12
13 public interface PatientRepository extends JpaRepository<Patient, Long> {
14     List<Patient> findByMalade(Boolean m);
15     Page<Patient> findByMalade(boolean m, Pageable pageable);
16     List<Patient> findByMaladeAndScoreLessThan(boolean m, int score);
17     List<Patient> findByMaladeIsFalseAndScoreLessThan(int score);
18     List<Patient> findByNomContains(String kw, Pageable pageable);
19     @Query("select p from Patient p where p.dateNaissance between :x and :y or p.nom like :z")
20     public List<Patient> FindPatients(@Param("x") Date d1, @Param("y") Date d2, @Param("z") String nom);
21 }
22
```

Dans le dossier web on crée le controlleur de la page web PatientController

```
19 @Controller
20 @AllArgsConstructor
21 public class PatientController {
22     private PatientRepository patientRepository;
23     @GetMapping(path = "/user/index")
24     public String patients(Model model,
25         @RequestParam(name = "page", defaultValue = "0") int page,
26         @RequestParam(name = "size", defaultValue = "5") int size,
27         @RequestParam(name = "keyword", defaultValue = "") String keyword){
28
29         Page<Patient> pagePatients=patientRepository.findAll(PageRequest.of(page, size));
30         List<Patient> patientList=new ArrayList<Patient>();
31         model.addAttribute( attributeName: "patientList", pagePatients.getContent());
32         model.addAttribute( attributeName: "pages", new int[pagePatients.getTotalPages()]);
33         model.addAttribute( attributeName: "currentPage", page);
34         model.addAttribute( attributeName: "keyword", keyword);
35         return "patients";
36     }
37
38     @GetMapping("/admin/delete")
39     public String delete(Long id, String keyword, String page){
40         patientRepository.deleteById(id);
41         return "redirect:/user/index?page="+page+"&keyword="+keyword;
42     }
43
44     @GetMapping("/home")
45     public String home() { return "home"; }
46
47     @GetMapping("/")
48     public String home2() { return "home"; }
49
50     @GetMapping("/admin/formPatients")
51     public String formPatients() { return "formPatients"; }
52
53     @GetMapping("/admin/formPatients")
54     public String formPatients() { return "formPatients"; }
55 }

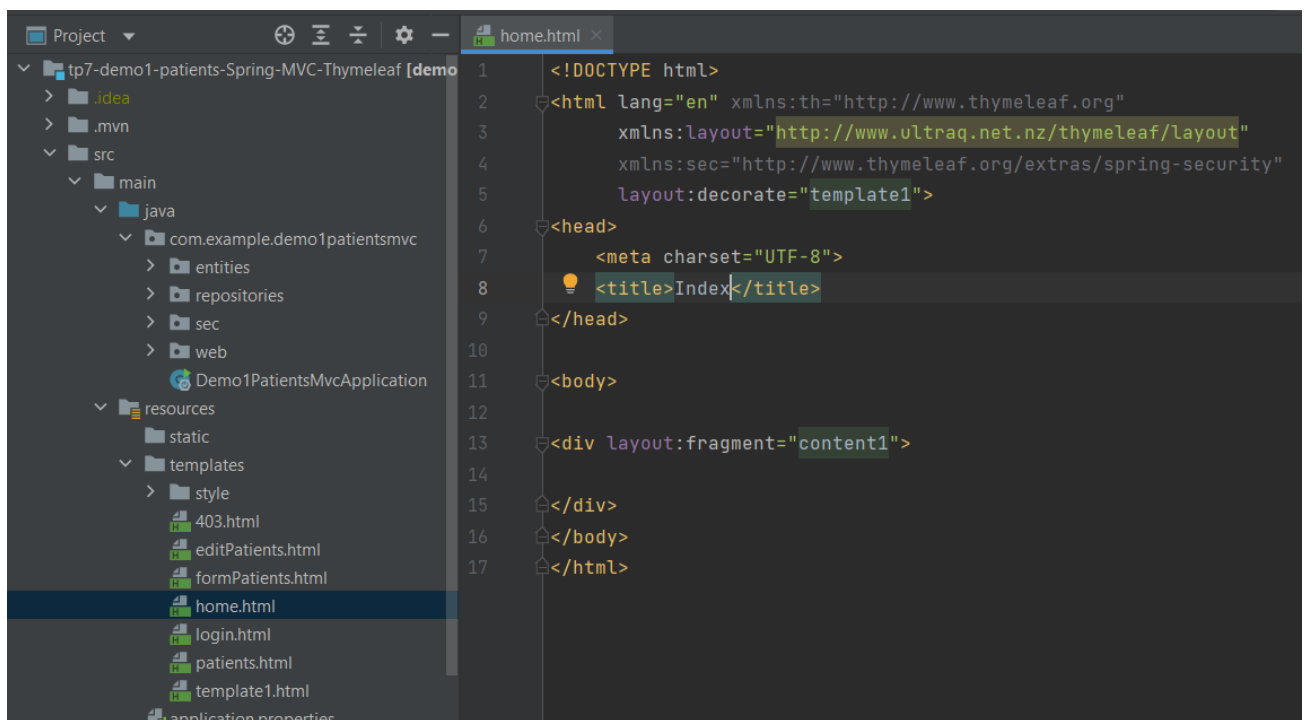
```

Sous le dossier ressource on crée un dossier template dans lequel on crée nos pages html
template1.html



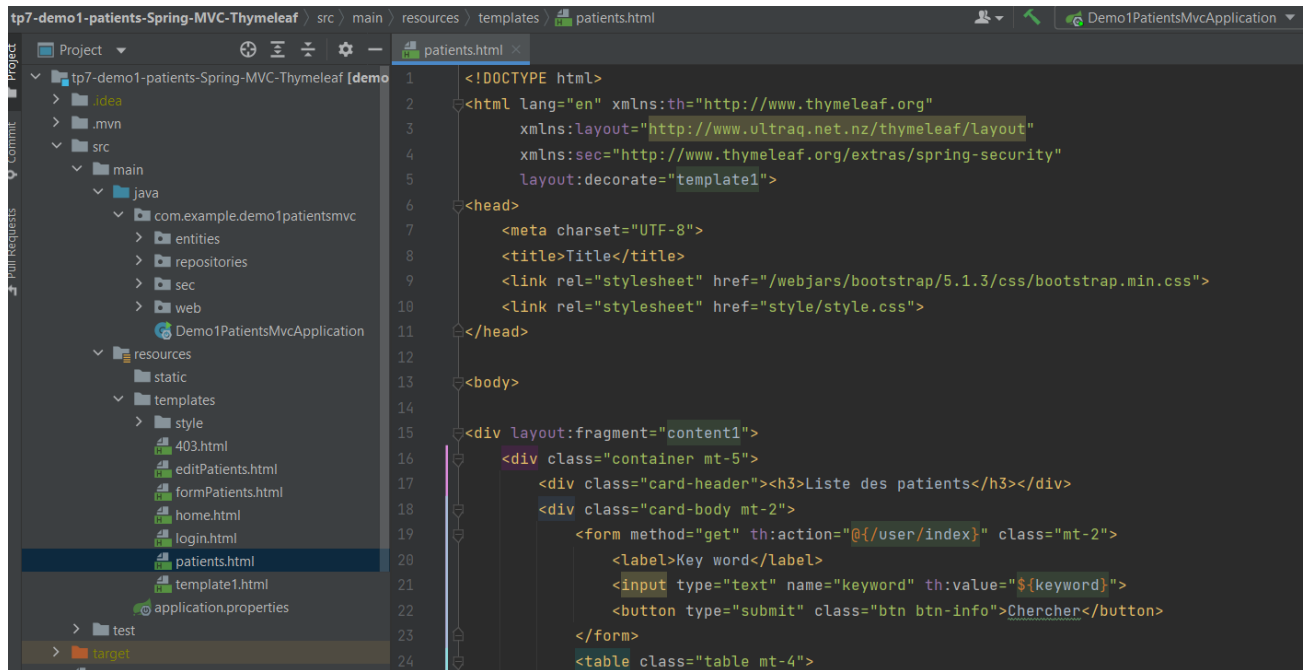
```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org"
3     xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
4     xmlns:sec="http://www.thymeleaf.org/extras/spring-security">
5 <head>
6     <meta charset="UTF-8">
7     <title>Title</title>
8     <link rel="stylesheet" type="text/css" href="/webjars/bootstrap/5.1.3/css/bootstrap.min.css">
9     <script src="/webjars/bootstrap/5.1.3/js/bootstrap.bundle.js"></script>
10
11 </head>
12 <body>
13     <!-- Grey with black text -->
14     <nav class="navbar navbar-expand-sm bg-dark navbar-dark">
15         <div class="container-fluid">
16             <ul class="navbar-nav">
17                 <li class="nav-item">
18                     <a class="nav-link active" th:href="@{/user/index}">Home</a>
19                 </li>
20                 <li class="nav-item">
21                     <a class="nav-link" href="#">Link</a>
22                 </li>
23                 <li class="nav-item">
24                     <a class="nav-link" href="#">Link</a>
25                 </li>
26             </ul>
27         </div>
28     </nav>
29 </body>
30 </html>
```

home.html



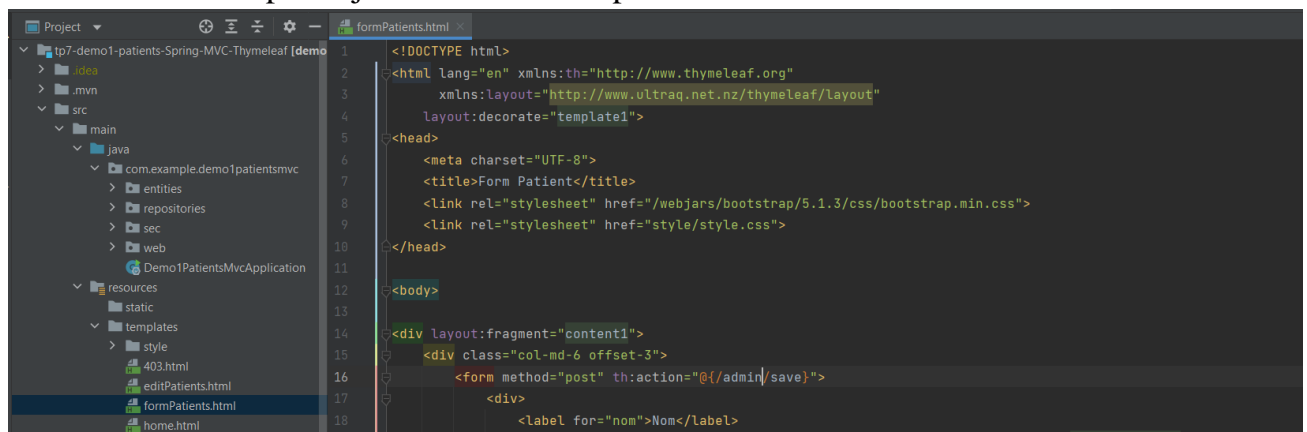
```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org"
3     xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
4     xmlns:sec="http://www.thymeleaf.org/extras/spring-security">
5     layout:decorate="template1"
6 <head>
7     <meta charset="UTF-8">
8     <title>Index</title>
9 </head>
10
11 <body>
12     <div layout:fragment="content1">
13
14     </div>
15 </body>
16 </html>
```

Patients.html



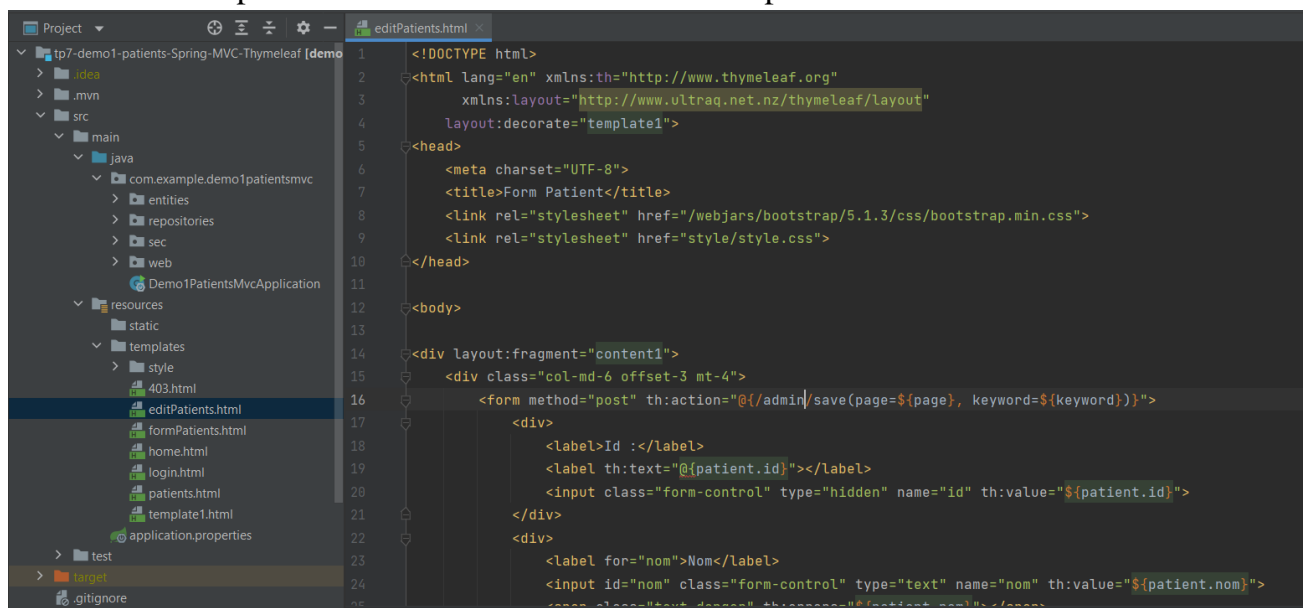
```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org"
3   xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
4   xmlns:sec="http://www.thymeleaf.org/extras/spring-security"
5   layout:decorate="template1">
6 </html>
7 <head>
8 <meta charset="UTF-8">
9 <title>Title</title>
10 <link rel="stylesheet" href="/webjars/bootstrap/5.1.3/css/bootstrap.min.css">
11 <link rel="stylesheet" href="style/style.css">
12 </head>
13 <body>
14 <div layout:fragment="content1">
15 <div class="container mt-5">
16 <div class="card-header"><h3>Liste des patients</h3></div>
17 <div class="card-body mt-2">
18 <form method="get" th:action="@{/user/index}" class="mt-2">
19 <label>Key word</label>
20 <input type="text" name="keyword" th:value="{keyword}">
21 <button type="submit" class="btn btn-info">Chercher</button>
22 </form>
23 <table class="table mt-4">
24
```

formPatients.html pour ajouter un nouveau patient



```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org"
3   xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
4   layout:decorate="template1">
5 </html>
6 <head>
7 <meta charset="UTF-8">
8 <title>Form Patient</title>
9 <link rel="stylesheet" href="/webjars/bootstrap/5.1.3/css/bootstrap.min.css">
10 <link rel="stylesheet" href="style/style.css">
11 </head>
12 <body>
13 <div layout:fragment="content1">
14 <div class="col-md-6 offset-3">
15 <form method="post" th:action="@{/admin/save}">
16 <div>
17 <label for="nom">Nom</label>
18 <input type="text" name="nom" th:value="{nom}">
19 </div>
20 </div>
21
```

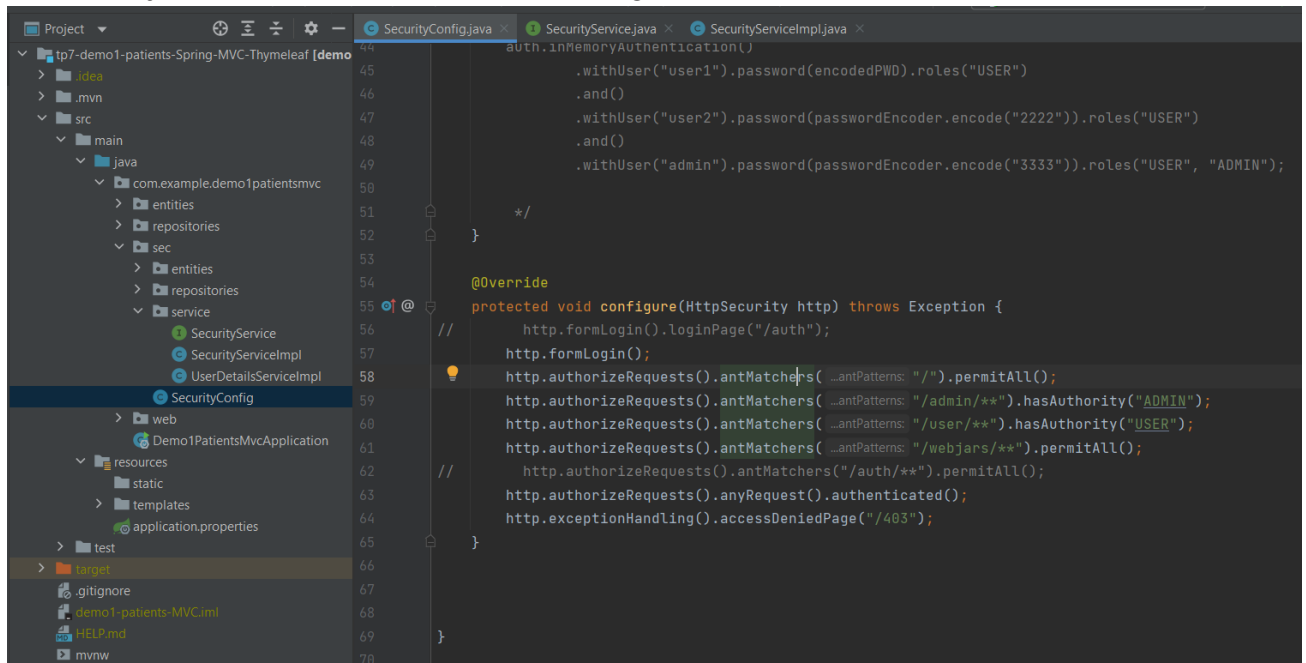
editPatient.html pour modifier les informations sur un patient



```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org"
3   xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
4   layout:decorate="template1">
5 </html>
6 <head>
7 <meta charset="UTF-8">
8 <title>Form Patient</title>
9 <link rel="stylesheet" href="/webjars/bootstrap/5.1.3/css/bootstrap.min.css">
10 <link rel="stylesheet" href="style/style.css">
11 </head>
12 <body>
13 <div layout:fragment="content1">
14 <div class="col-md-6 offset-3 mt-4">
15 <form method="post" th:action="@{/admin/save(page=${page}, keyword=${keyword})}">
16 <div>
17 <label>Id :</label>
18 <label th:text="@{patient.id}"></label>
19 <input class="form-control" type="hidden" name="id" th:value="{patient.id}">
20 </div>
21 <div>
22 <label for="nom">Nom</label>
23 <input id="nom" class="form-control" type="text" name="nom" th:value="{patient.nom}">
24
```

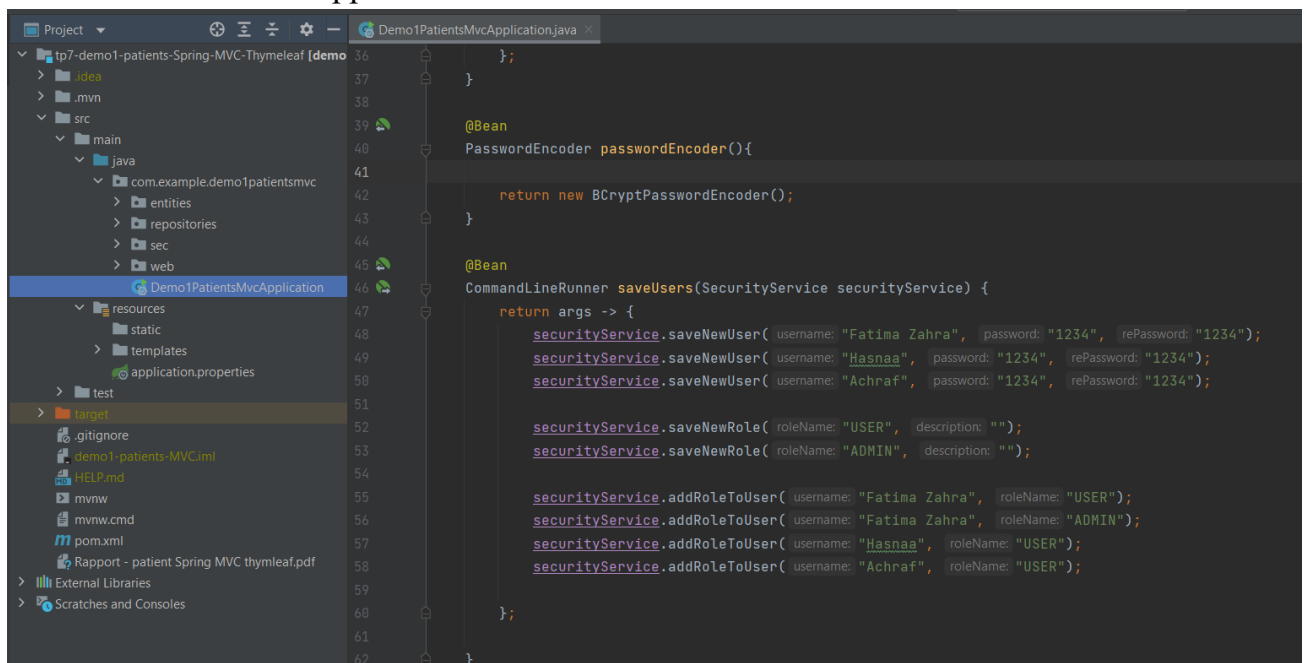
Pour la sécurité de l'application on va ajouter l'authentification et les rôles tel que chaque utilisateur avec le rôle ADMIN peut afficher, ajouter, modifier, ou supprimer des patients. Mais un utilisateur avec le rôle USER ne peut faire que l'affichage.

Donc on ajoute un dossier sec ou on va configurer cette fonctionnalité



```
44 auth.inMemoryAuthentication()
45     .withUser("user1").password(encodedPWD).roles("USER")
46     .and()
47     .withUser("user2").password(passwordEncoder.encode("2222")).roles("USER")
48     .and()
49     .withUser("admin").password(passwordEncoder.encode("3333")).roles("USER", "ADMIN");
50
51     */
52 }
53
54 @Override
55 protected void configure(HttpSecurity http) throws Exception {
56     // http.formLogin().loginPage("/auth");
57     http.formLogin();
58     http.authorizeRequests().antMatchers("/*").permitAll();
59     http.authorizeRequests().antMatchers("/admin/**").hasAuthority("ADMIN");
60     http.authorizeRequests().antMatchers("/user/**").hasAuthority("USER");
61     http.authorizeRequests().antMatchers("/webjars/**").permitAll();
62     // http.authorizeRequests().antMatchers("/auth/**").permitAll();
63     http.authorizeRequests().anyRequest().authenticated();
64     http.exceptionHandling().accessDeniedPage("/403");
65 }
66
67
68
69 }
70
```

La fonction main de l'application



```
36 };
37 }
38
39 @Bean
40 PasswordEncoder passwordEncoder(){
41
42     return new BCryptPasswordEncoder();
43 }
44
45 @Bean
46 CommandLineRunner saveUsers(SecurityService securityService) {
47     return args -> {
48         securityService.saveNewUser( username: "Fatima Zahra", password: "1234", rePassword: "1234");
49         securityService.saveNewUser( username: "Hasnaa", password: "1234", rePassword: "1234");
50         securityService.saveNewUser( username: "Achraf", password: "1234", rePassword: "1234");
51
52         securityService.saveNewRole( roleName: "USER", description: "");
53         securityService.saveNewRole( roleName: "ADMIN", description: "");
54
55         securityService.addRoleToUser( username: "Fatima Zahra", roleName: "USER");
56         securityService.addRoleToUser( username: "Fatima Zahra", roleName: "ADMIN");
57         securityService.addRoleToUser( username: "Hasnaa", roleName: "USER");
58         securityService.addRoleToUser( username: "Achraf", roleName: "USER");
59     };
60 }
61
62 }
```

Dans application.properties

```
application.properties
1-1patients-MVC] C:\Users\zertyui\OneDrive - enset-media.ac.ma\Hasbi\GLSID\S4\1- J2EE\tps\tp7-demo1-patients-Spring-MVC-Thymeleaf
2   #spring.h2.console.enabled=true
3   server.port=8082
4   spring.datasource.url=jdbc:mysql://localhost:3306/PAT_DB?createDatabaseIfNotExist=true
5   spring.datasource.username=root
6   spring.datasource.password=
7   spring.jpa.hibernate.ddl-auto = create
8   spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MariaDBDialect
9   spring.jpa.show-sql=true
10  spring.thymeleaf.cache=false
11  spring.main.allow-circular-references = true
```

Démonstration

Please sign in

Fatima Zahra

....

Sign in

FatimaZahraHASBI/Fatima

Boîte de réception - hasbi

Title

localhost:8082/user/index

Home

Link

Link

Disabled

Patients

Fatima Zahra

Liste des patients

Key word

Chercher

ID	Nom	Date	Malade	Score		
1	Fatima Zahra	2022-04-17	false	44	Delete	Edit
2	Achraf	2022-04-17	false	55	Delete	Edit
3	Hasnaa	2022-04-17	false	33	Delete	Edit
4	Noureddine	2022-04-17	false	66	Delete	Edit
5	Laila	2022-04-17	true	88	Delete	Edit

0

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

FatimaZahraHASBI/Fatima

Boîte de réception - hasbi

Form Patient

localhost:8082/admin/editPatients

Home

Link

Link

Disabled

Patients

Fatima Zahra

Id : patient.id

Nom

Fatima Zahra

Date Naissance

04/17/2022

Malade ☐

Score

44

Save

Home

Link

Link

Disabled

Patients

Fatima Zahra

Nouveau

Chercher

m

Date Naissance

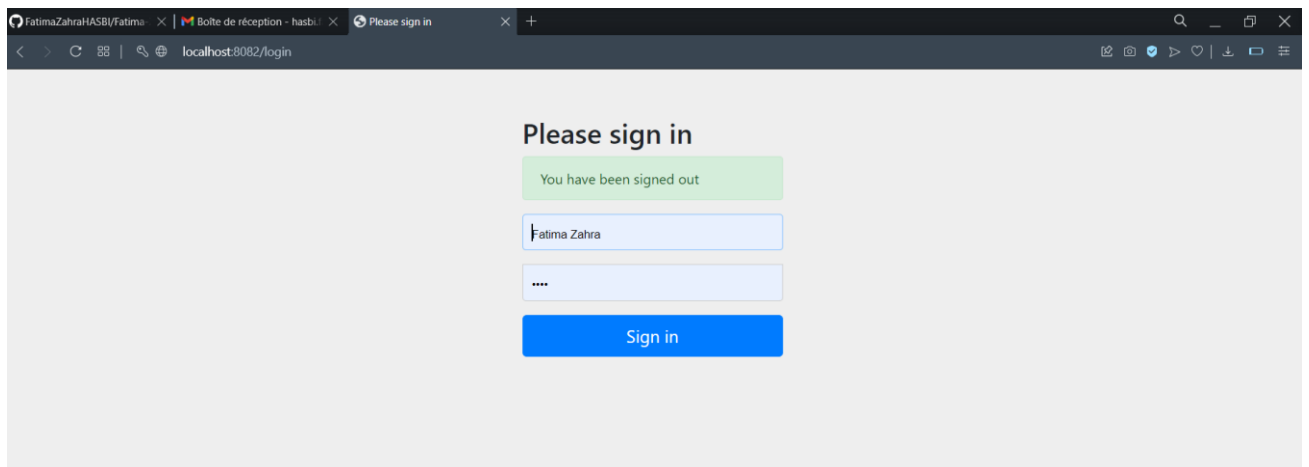
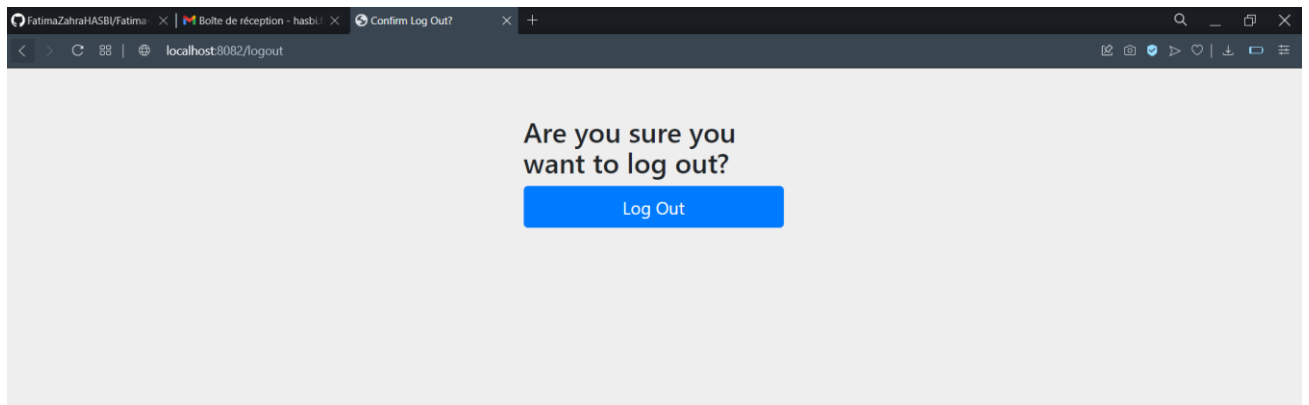
mm/dd/yyyy

Malade ☐

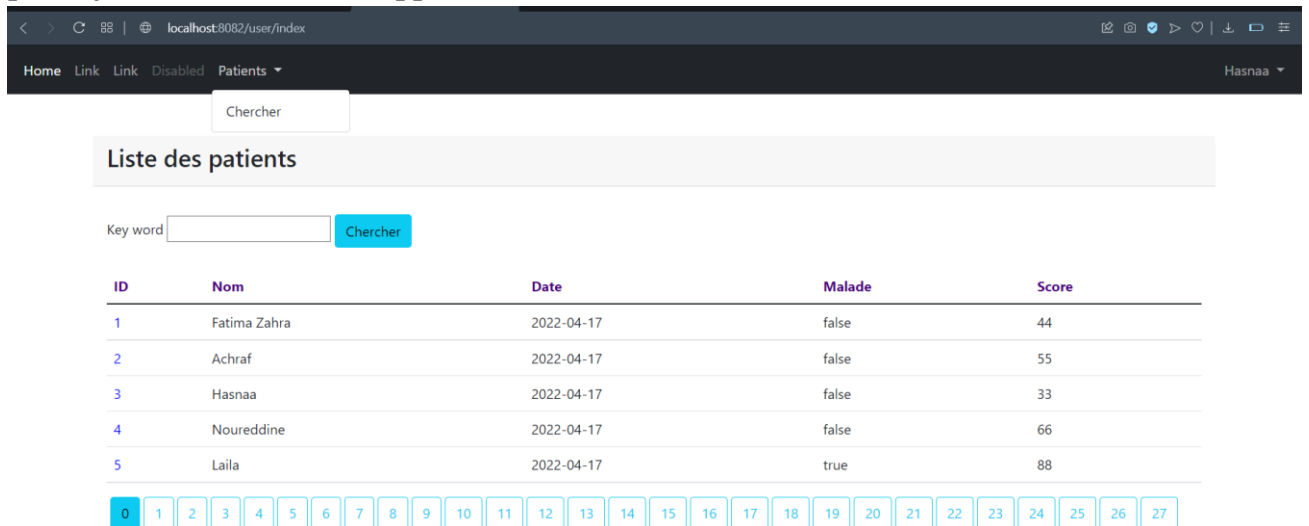
Score

0

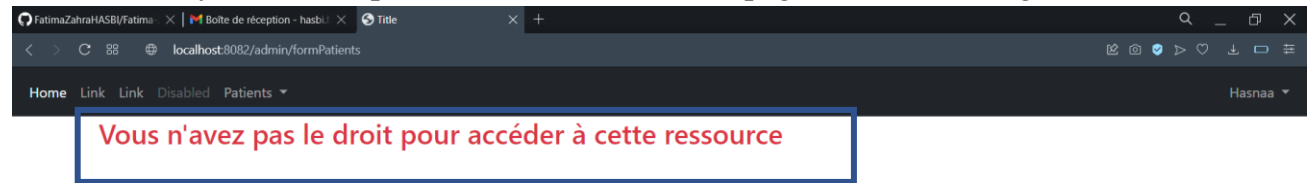
Save



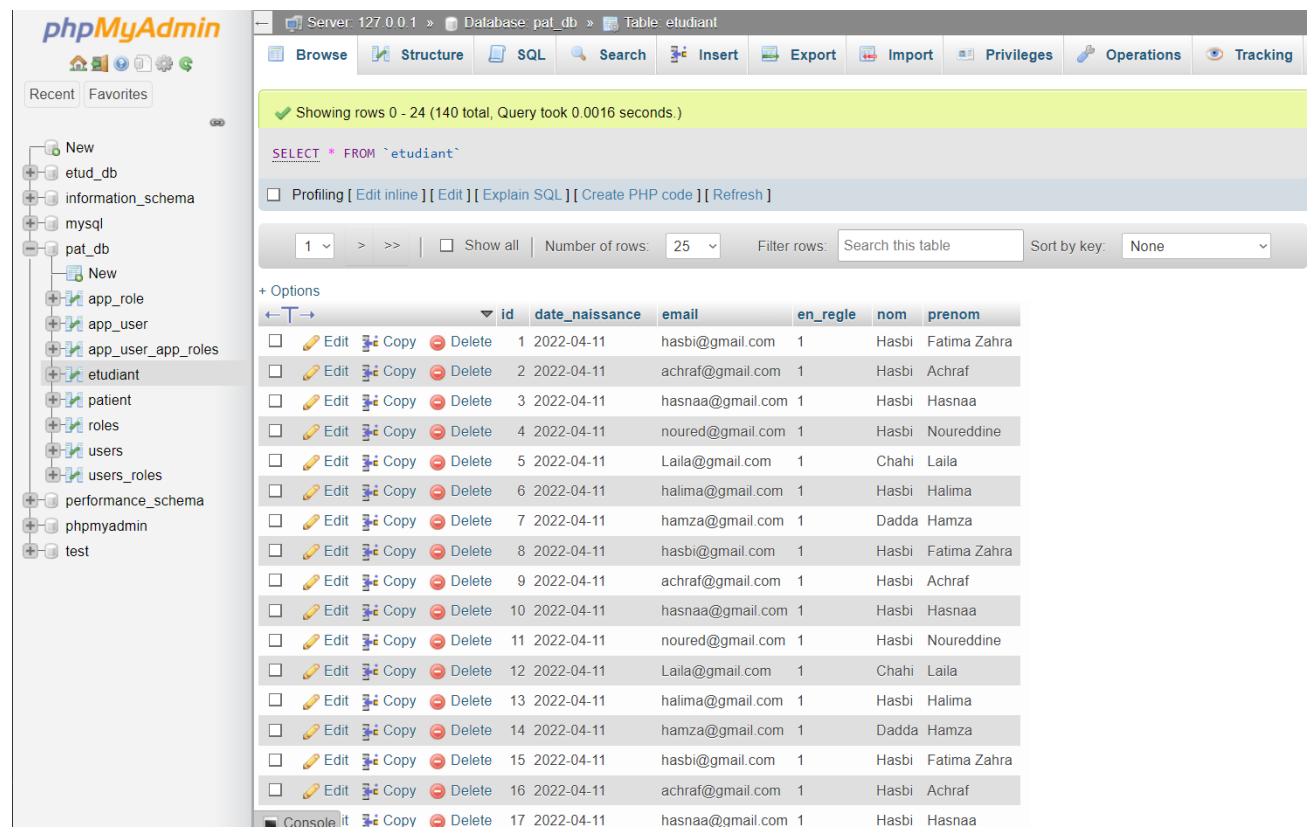
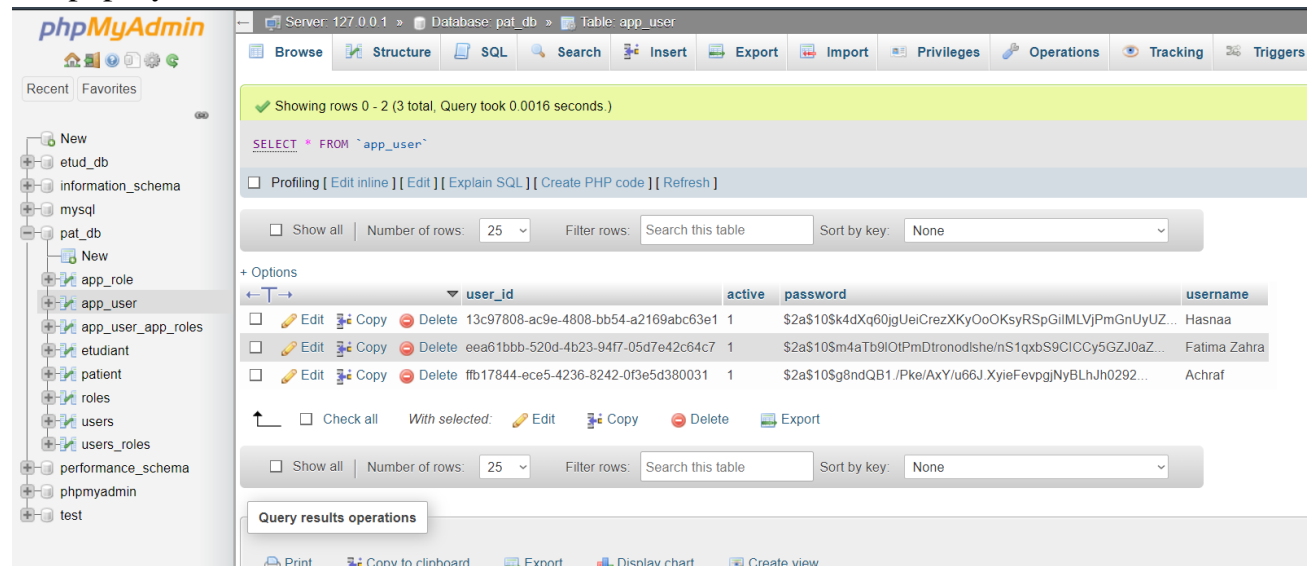
Si on fait l'authentification avec un utilisateur qui a le rôle USER on va pas avoir le droit pour ajouter, modifier ou supprimer



Si on tente d'y accéder depuis l'url on va avoir cette page avec le message suivant



Sur phpMyAdmin



Sur GitHub

The screenshot shows the GitHub interface for the repository 'FatimaZahraHASBI / Fatima-Zahra-Hasbi-JEE'. The repository is public. The navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The 'master' branch is selected. A red box highlights the repository name and the path 'Fatima-Zahra-Hasbi-JEE / tp7-demo1-patients-Spring-MVC-Thymeleaf /'. Below this, the repository's file structure is listed:

File/Folder	Repository
..	tp
.mvn/wrapper	tp
src	tp
.gitignore	tp
Rapport - patient Spring MVC thymleaf.pdf	spring mvc thymleaf
mvnw	tp
mvnw.cmd	tp
pom.xml	tp