Втора домашна задача

Информациска безбедност

Љубомир Давитков 216092

Автентикација

Системот за автентикација за домашна работа е веб-апликација изградена со помош на Java, Spring MVC, Spring Boot, Spring Data JPA, PostgreSQL, HTML, CSS и Thymeleaf.

Користени технологии:

Java 17: Основниот програмски јазик за апликацијата.

Spring Boot 3.1.4: Го поедноставува поставувањето и развојот на пролетните апликации.

Spring MVC: Управува со веб-слојот на апликацијата.

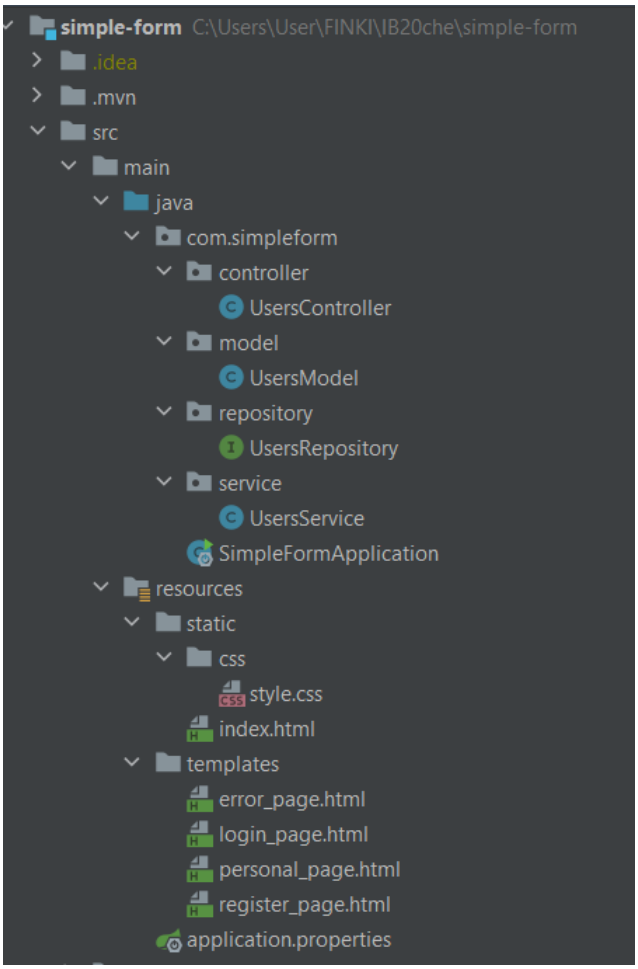
Spring Data JPA: Го поедноставува пристапот до податоци со користење на Java Persistence API (JPA).

PostgreSQL: Служи како релациона база на податоци за складирање на податоци од апликацијата.

HTML и Thymeleaf: Се користи за прикажување динамични веб-страници и шаблони.

CSS: Го подобрува визуелниот стил на апликацијата.

Структурата на апликацијата:



Контролерот на апликацијата:

```
Ljubomir
@GetMapping("/register")
public String getRegisterPage(Model model) {
    model.addAttribute( attributeName: "registerRequest", new UsersModel());
    return "register_page";
}
```

Се справува со барањата GET за крајната точка „/register“.

Го подготвува моделот со празен UsersModel и го враќа шаблонот „register_page“.

```
Ljubomir
@GetMapping("/login")
public String getLoginPage(Model model) {
    model.addAttribute( attributeName: "loginRequest", new UsersModel());
    return "login_page";
}
```

Се справува со барањата GET за крајната точка „/login“.

Го подготвува моделот со празен UsersModel и го враќа шаблонот „login_page“.

Се справува со барањата POST за крајната точка „/register“.

Регистрира корисник користејќи го дадениот UsersModel.

Се пренасочува на „/login“ доколку регистрацијата е успешна; во спротивно, се пренасочува на „error_page“.

Се справува со барањата POST за крајната точка „/login“.

Го автентицира корисникот користејќи го дадениот UsersModel.

Пренасочува кон „personal_page“ со автентицирано најавување на корисникот доколку е успешно; во спротивно, се пренасочува на „error_page“.

Service делот на апликацијата:

Регистрира нов корисник со наведеното најавување, лозинка и е-пошта.

Ги потврдува влезните параметри и проверува дали има дупликат најавувања.

Го враќа регистрираниот UsersModel доколку е успешен, или е нулатен во случај на неуспех на валидацијата или дупликат најава.

```
1 usage  Ljubomir
public UsersModel authenticate(String login, String password) {
    return usersRepository.findByLoginAndPassword(login, password).orElse( other: null);
}
```

Го автентицира корисникот со наведеното најавување и лозинка.

Го бара корисничкото складиште за соодветен корисник.

Го враќа автентицираниот UsersModel ако е успешен или нула ако автентикацијата не успее.

Repository делот на апликацијата:

```
3 usages  Ljubomir
public interface UsersRepository extends JpaRepository<UsersModel, Integer> {
    1 usage  Ljubomir
    Optional<UsersModel> findByLoginAndPassword(String login, String password);


    1 usage  Ljubomir
    Optional<UsersModel> findFirstByLogin(String login);
}
```

UsersRepository е интерфејс за складиште за пролетни податоци JPA одговорен за ракување со операциите на базата на податоци поврзани со ентитетот UsersModel.