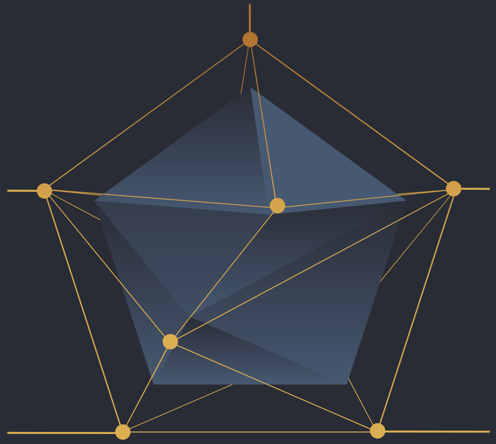
Final Project

Emanuel Draniceanu

Auction Application

Main System Functions



Home Page

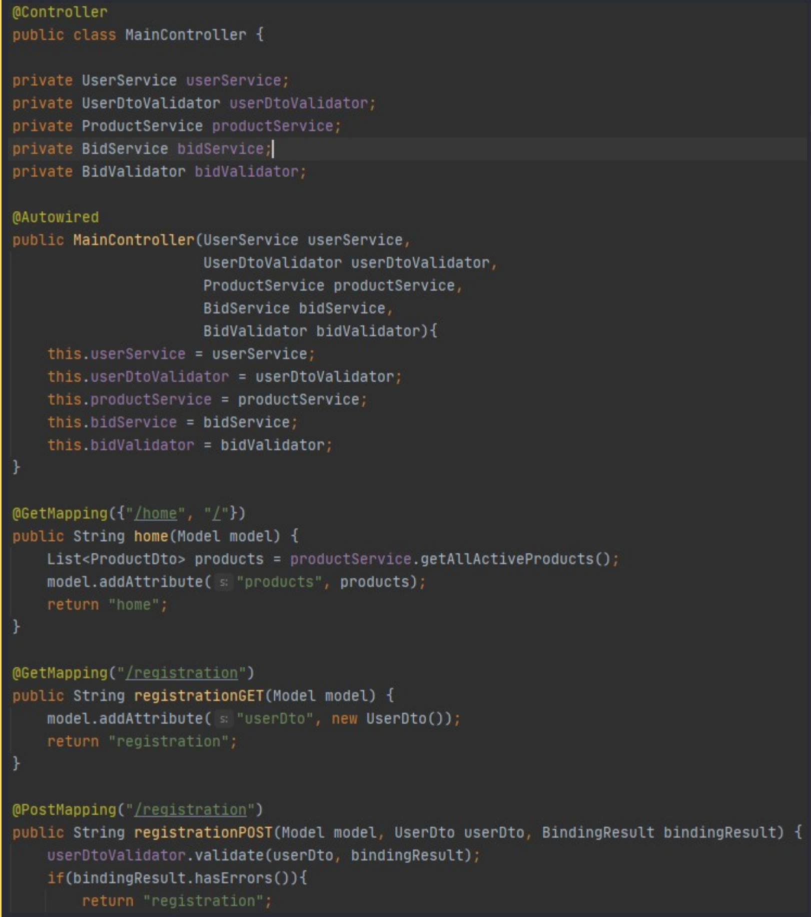
User Account

Bidding

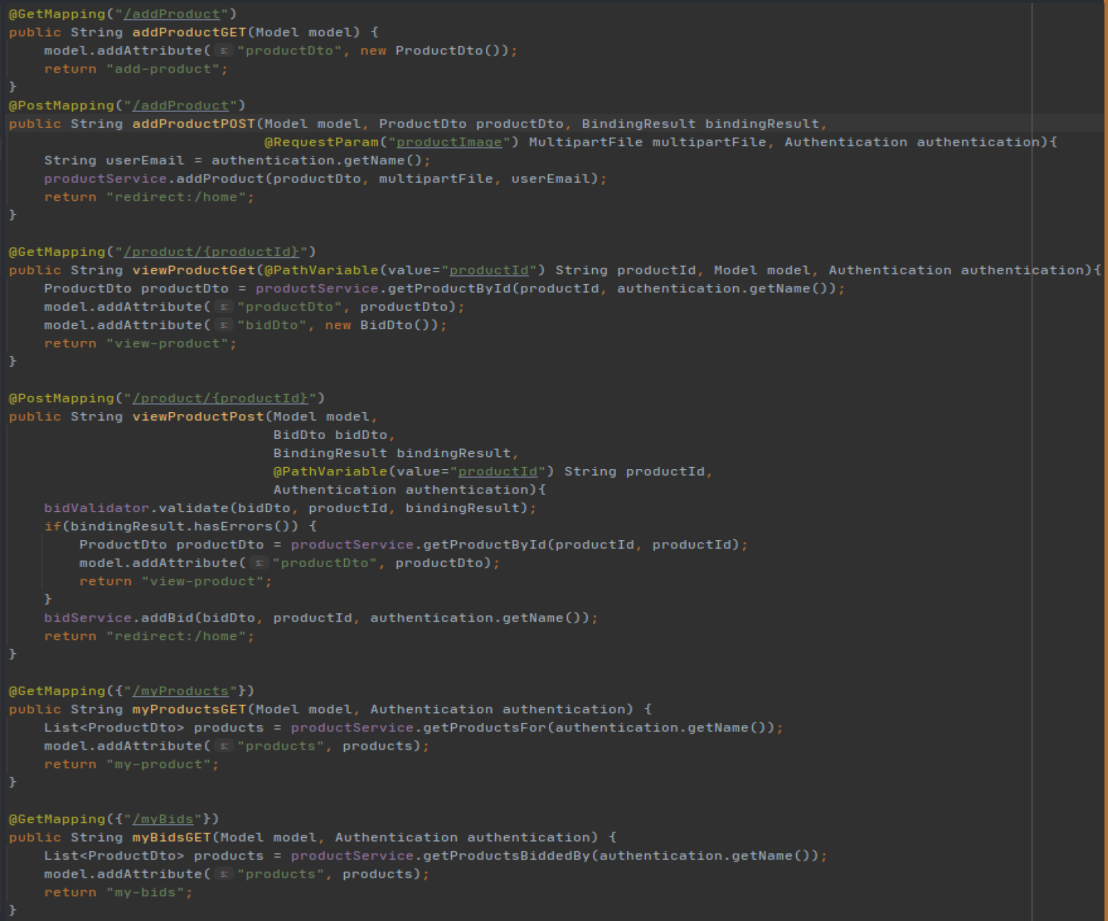
Categories

Auctions

Main Controller



```java
@Controller
public class MainController {

private UserService userService;
private UserDtoValidator userDtoValidator;
private ProductService productService;
private BidService bidService;
private BidValidator bidValidator;

@Autowired
public MainController(UserService userService,
                      UserDtoValidator userDtoValidator,
                      ProductService productService,
                      BidService bidService,
                      BidValidator bidValidator){
    this.userService = userService;
    this.userDtoValidator = userDtoValidator;
    this.productService = productService;
    this.bidService = bidService;
    this.bidValidator = bidValidator;
}

@GetMapping({"/home", "/"})
public String home(Model model) {
    List<ProductDto> products = productService.getAllActiveProducts();
    model.addAttribute( s: "products", products);
    return "home";
}

@GetMapping("/registration")
public String registrationGET(Model model) {
    model.addAttribute( s: "userDto", new UserDto());
    return "registration";
}

@PostMapping("/registration")
public String registrationPOST(Model model, UserDto userDto, BindingResult bindingResult) {
    userDtoValidator.validate(userDto, bindingResult);
    if(bindingResult.hasErrors()){
        return "registration";
```



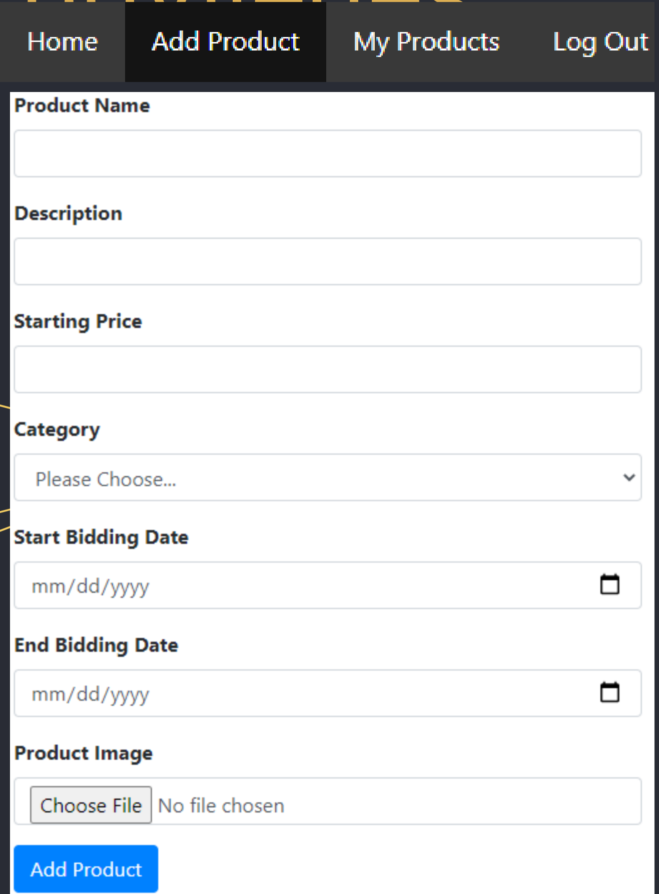
```java
@GetMapping("/addProduct")
public String addProductGET(Model model) {
    model.addAttribute( s: "productDto", new ProductDto());
    return "add-product";
}
@PostMapping("/addProduct")
public String addProductPOST(Model model, ProductDto productDto, BindingResult bindingResult,
                             @RequestParam("productImage") MultipartFile multipartFile, Authentication authentication){
    String userEmail = authentication.getName();
    productService.addProduct(productDto, multipartFile, userEmail);
    return "redirect:/home";
}

@GetMapping("/product/{productId}")
public String viewProductGet(@PathVariable(value="productId") String productId, Model model, Authentication authentication){
    ProductDto productDto = productService.getProductById(productId, authentication.getName());
    model.addAttribute( s: "productDto", productDto);
    model.addAttribute( s: "bidDto", new BidDto());
    return "view-product";
}

@PostMapping("/product/{productId}")
public String viewProductPost(Model model,
                              BidDto bidDto,
                              BindingResult bindingResult,
                              @PathVariable(value="productId") String productId,
                              Authentication authentication){
    bidValidator.validate(bidDto, productId, bindingResult);
    if(bindingResult.hasErrors()) {
        ProductDto productDto = productService.getProductById(productId, productId);
        model.addAttribute( s: "productDto", productDto);
        return "view-product";
    }
    bidService.addBid(bidDto, productId, authentication.getName());
    return "redirect:/home";
}

@GetMapping({"/myProducts"})
public String myProductsGET(Model model, Authentication authentication) {
    List<ProductDto> products = productService.getProductsFor(authentication.getName());
    model.addAttribute( s: "products", products);
    return "my-product";
}

@GetMapping({"/myBids"})
public String myBidsGET(Model model, Authentication authentication) {
    List<ProductDto> products = productService.getProductsBiddedBy(authentication.getName());
    model.addAttribute( s: "products", products);
    return "my-bids";
}
```

Functionalities

- Only admins can add products and users can only see and bid on them.
- Presentation of the list of recently added auctions.
- Presentation of the list of auctions of the logged-in user.
- Presentation of the list of auctions where the logged-in user is bidding on.
- Only logged-in users can bid on products.
- When product bid date ends, a winner will be chosen by the highest bid.

Admin privileges

User Privileges



Product Name

Description

Starting Price

Category

Please Choose...

Start Bidding Date

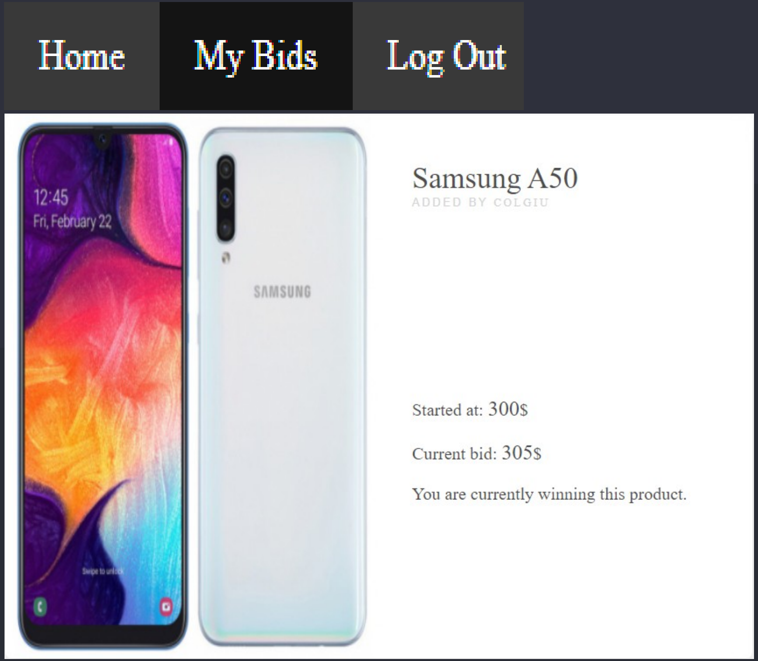
mm/dd/yyyy

End Bidding Date

mm/dd/yyyy

Product Image

No file chosen



Samsung A50

ADDED BY COLGIU

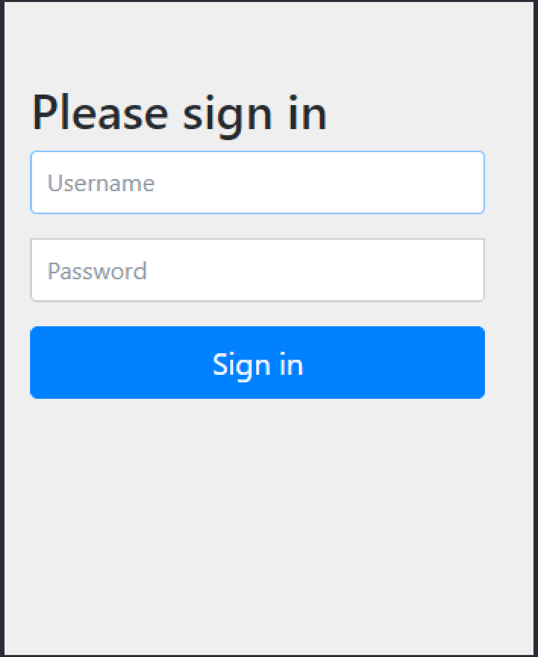
Started at: 300$

Current bid: 305$

You are currently winning this product.

User Account

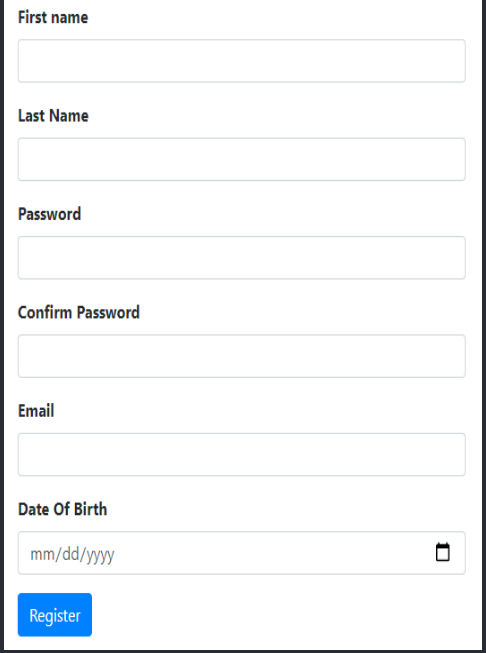
- Log-in
- Password
- Confirm Password
- Account Name
- Date of Birth



Please sign in

Username

Password



First name

Last Name

Password

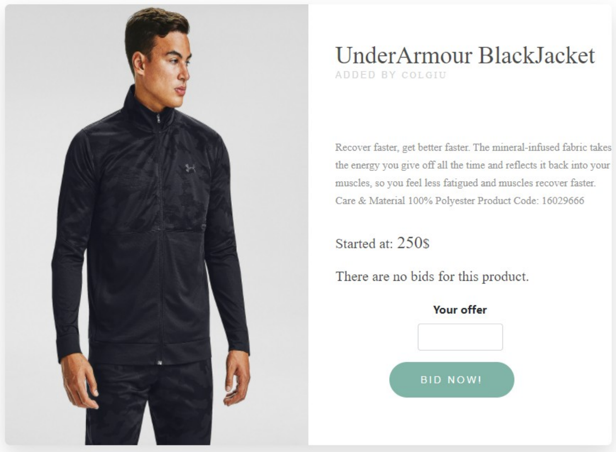
Confirm Password

Email

Date Of Birth

mm/dd/yyyy

Auction


UnderArmour BlackJacket

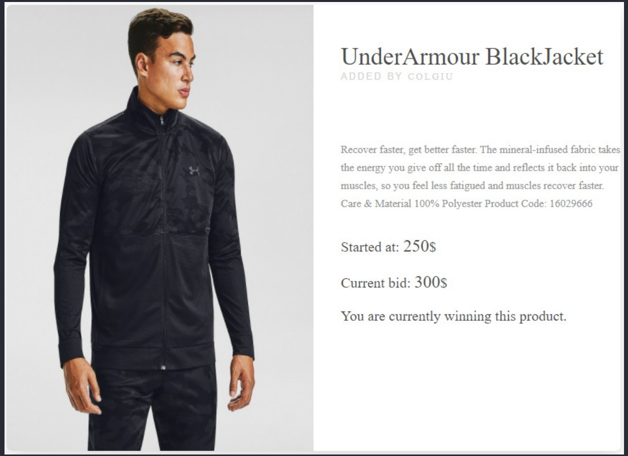
ADDED BY COLGIU

Recover faster, get better faster. The mineral-infused fabric takes the energy you give off all the time and reflects it back into your muscles, so you feel less fatigued and muscles recover faster. Care & Material 100% Polyester Product Code: 16029666

Started at: 250$

There are no bids for this product.

Title
Description
Photo (on the left)
Bidding price
Current bid (photo on the right)
The highest bidder


UnderArmour BlackJacket

ADDED BY COLGIU

Recover faster, get better faster. The mineral-infused fabric takes the energy you give off all the time and reflects it back into your muscles, so you feel less fatigued and muscles recover faster. Care & Material 100% Polyester Product Code: 16029666

Started at: 250$

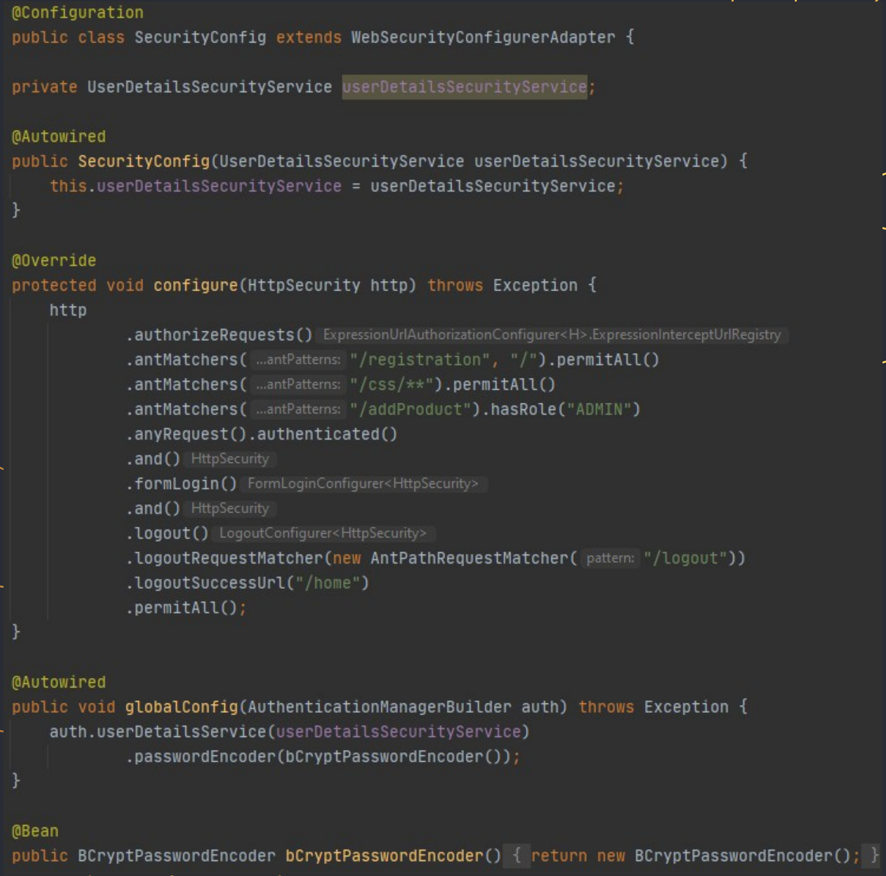
Current bid: 300$

You are currently winning this product.

Technologies

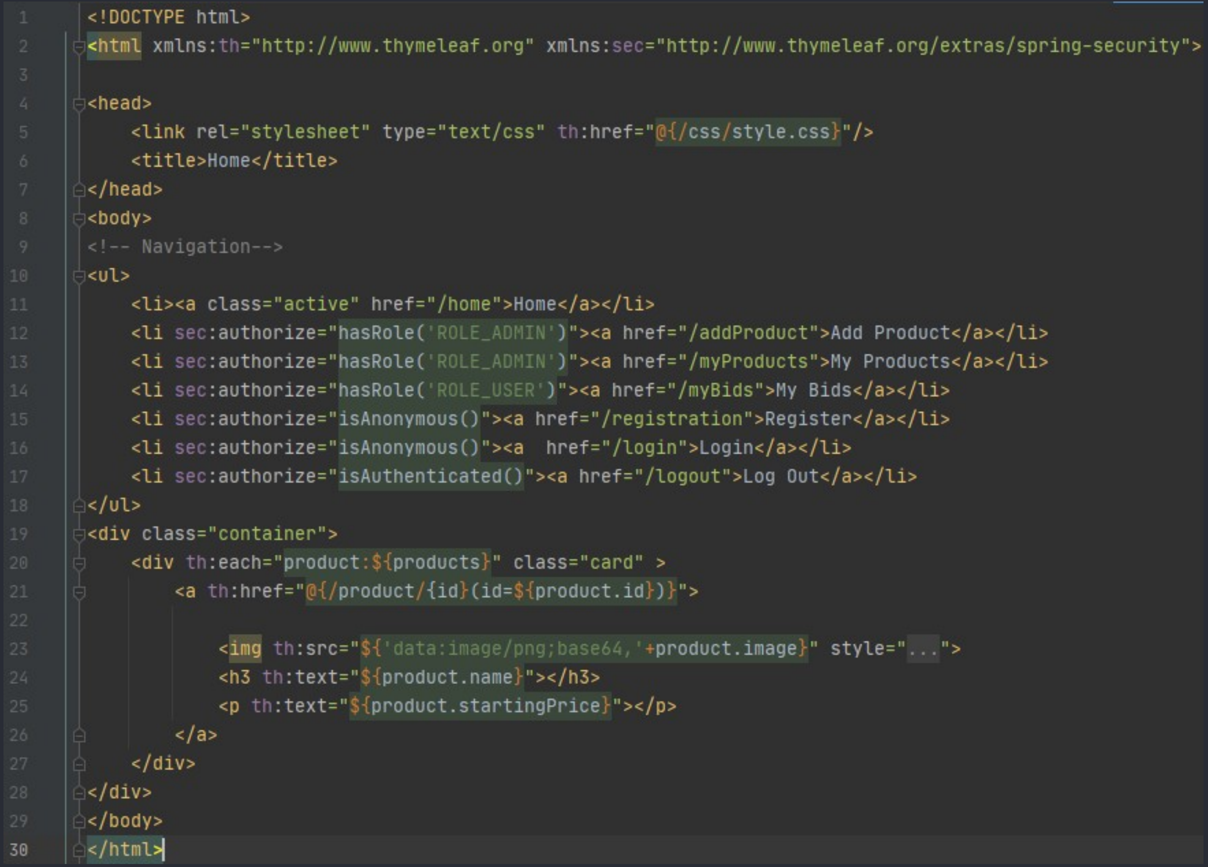
- Spring Framework
- Spring Security
- Hibernate
- Thymeleaf
- MySQL Database

Spring Security Integrated



```
@Configuration
public class SecurityConfig extends WebSecurityConfigurerAdapter {

private UserDetailsSecurityService userDetailsSecurityService;

@Autowired
public SecurityConfig(UserDetailsSecurityService userDetailsSecurityService) {
    this.userDetailsSecurityService = userDetailsSecurityService;
}

@Override
protected void configure(HttpSecurity http) throws Exception {
    http
            .authorizeRequests() ExpressionUrlAuthorizationConfigurer<H>.ExpressionInterceptUrlRegistry
            .antMatchers( ...antPatterns: "/registration", "/").permitAll()
            .antMatchers( ...antPatterns: "/css/**").permitAll()
            .antMatchers( ...antPatterns: "/addProduct").hasRole("ADMIN")
            .anyRequest().authenticated()
            .and() HttpSecurity
            .formLogin() FormLoginConfigurer<HttpSecurity>
            .and() HttpSecurity
            .logout() LogoutConfigurer<HttpSecurity>
            .logoutRequestMatcher(new AntPathRequestMatcher( pattern: "/logout"))
            .logoutSuccessUrl("/home")
            .permitAll();
}

@Autowired
public void globalConfig(AuthenticationManagerBuilder auth) throws Exception {
    auth.userDetailsService(userDetailsSecurityService)
            .passwordEncoder(bCryptPasswordEncoder());
}

@Bean
public BCryptPasswordEncoder bCryptPasswordEncoder() { return new BCryptPasswordEncoder(); }
```

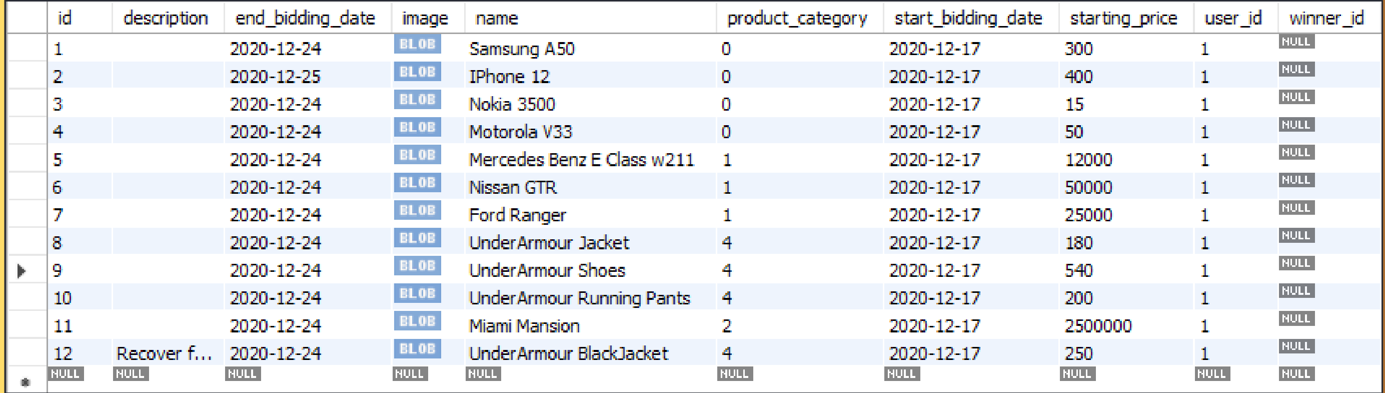
Thymeleaf Integrated



```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org" xmlns:sec="http://www.thymeleaf.org/extras/spring-security">

<head>
    <link rel="stylesheet" type="text/css" th:href="@{/css/style.css}"/>
    <title>Home</title>
</head>
<body>
<!-- Navigation-->
<ul>
    <li><a class="active" href="/home">Home</a></li>
    <li sec:authorize="hasRole('ROLE_ADMIN')"><a href="/addProduct">Add Product</a></li>
    <li sec:authorize="hasRole('ROLE_ADMIN')"><a href="/myProducts">My Products</a></li>
    <li sec:authorize="hasRole('ROLE_USER')"><a href="/myBids">My Bids</a></li>
    <li sec:authorize="isAnonymous()"><a href="/registration">Register</a></li>
    <li sec:authorize="isAnonymous()"><a  href="/login">Login</a></li>
    <li sec:authorize="isAuthenticated()"><a href="/logout">Log Out</a></li>
</ul>
<div class="container">
    <div th:each="product:${products}" class="card" >
        <a th:href="@{/product/{id}(id=${product.id})}">

            <img th:src="${'data:image/png;base64,'+product.image}" style="...">
            <h3 th:text="${product.name}"></h3>
            <p th:text="${product.startingPrice}"></p>
        </a>
    </div>
</div>
</body>
</html>
```

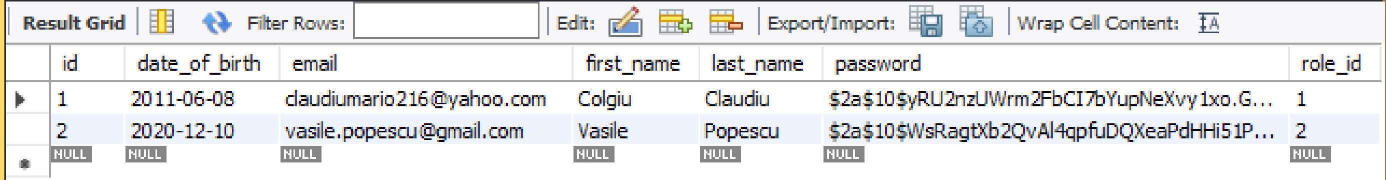
Database

Products



id	description	end_bidding_date	image	name	product_category	start_bidding_date	starting_price	user_id	winner_id
1		2020-12-24	BLOB	Samsung A50	0	2020-12-17	300	1	NULL
2		2020-12-25	BLOB	IPhone 12	0	2020-12-17	400	1	NULL
3		2020-12-24	BLOB	Nokia 3500	0	2020-12-17	15	1	NULL
4		2020-12-24	BLOB	Motorola V33	0	2020-12-17	50	1	NULL
5		2020-12-24	BLOB	Mercedes Benz E Class w211	1	2020-12-17	12000	1	NULL
6		2020-12-24	BLOB	Nissan GTR	1	2020-12-17	50000	1	NULL
7		2020-12-24	BLOB	Ford Ranger	1	2020-12-17	25000	1	NULL
8		2020-12-24	BLOB	UnderArmour Jacket	4	2020-12-17	180	1	NULL
9		2020-12-24	BLOB	UnderArmour Shoes	4	2020-12-17	540	1	NULL
10		2020-12-24	BLOB	UnderArmour Running Pants	4	2020-12-17	200	1	NULL
11		2020-12-24	BLOB	Miami Mansion	2	2020-12-17	2500000	1	NULL
12	Recover f...	2020-12-24	BLOB	UnderArmour BlackJacket	4	2020-12-17	250	1	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Users



id	date_of_birth	email	first_name	last_name	password	role_id
1	2011-06-08	claudiumario216@yahoo.com	Colgiu	Claudiu	$2a$10$yRU2nzUWrm2FbCI7bYupNeXvy1xo.G...	1
2	2020-12-10	vasile.popescu@gmail.com	Vasile	Popescu	$2a$10$WsRagtXb2QvAl4qpfuDQXeaPdHHi51P...	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Thank you for your attention!