

КАФЕДРА №

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

\_\_\_\_\_  
должность, уч. степень, звание

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
инициалы, фамилия

## ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5

Разработка приложения с использованием thymeleaf

по дисциплине: Технология разработки серверных информационных систем

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
инициалы, фамилия

Санкт-Петербург  
2023

**Текст и вариант задания:**

13. Торговля акциями на бирже.

**Описание разрабатываемого продукта:**

В программе создан сервис для покупки, продажи и просмотра портфеля акций.

**Задание на лабораторную работу.**

- 1 Реализуйте задачи из лабораторной работы 1 средствами Thymeleaf
- 2 Обратите внимание на локализацию приложения. Русский язык должен поддерживаться.

**Текст основных фрагментов кода:**

Itemobj.java

```
package com.example.ThymeleafProject;
```

```
public class ItemObj {
```

```
    private String stock_name;
```

```
    private int stockID;
```

```
    private String purchase_date;
```

```
    // Конструктор
```

```
    public ItemObj(String stock_name, int stockID, String purchase_date) {
```

```
        this.stock_name = stock_name;
```

```
        this.stockID = stockID;
```

```
        this.purchase_date = purchase_date;
```

```
    }
```

```
    // Геттеры и сеттеры
```

```
    public String getStock_name() {
```

```
        return stock_name;
```

```
}
```

```
public void setStock_name(String stock_name) {  
    this.stock_name = stock_name;  
}
```

```
public int getStockID() {  
    return stockID;  
}
```

```
public void setStockID(int stockID) {  
    this.stockID = stockID;  
}
```

```
public String getPurchase_date() {  
    return purchase_date;  
}
```

```
public void setPurchase_date(String purchase_date) {  
    this.purchase_date = purchase_date;  
}
```

```
@Override
```

```
public String toString() {  
    return "ItemObj{" +  
        "stock_name=" + stock_name + "\" +  
        ", stockID=" + stockID +  
        ", purchase_date=" + purchase_date + "\" +  
        '}'
```

```
}  
}
```

Logincontroller.java

```
package com.example.ThymeleafProject;
```

```
import jakarta.servlet.ServletException;
```

```
import jakarta.servlet.http.HttpServletRequest;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.PostMapping;
```

```
@Controller
```

```
public class LoginController {
```

```
    @GetMapping("/login")
```

```
    public String login() {
```

```
        return "login";
```

```
    }
```

```
    @PostMapping("/logout")
```

```
    public String logout(HttpServletRequest request) throws ServletException {
```

```
        request.logout();
```

```
        return "redirect:/login";
```

```
    }
```

```
}
```

Stockcontroller.java

```
package com.example.ThymeleafProject;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.Model;
```

```
import org.springframework.web.bind.annotation.*;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
@Controller
```

```
@RequestMapping("/stocks") // Изменено с /patients на /stocks
```

```
class StockController { // Изменено имя класса
```

```
    private List<ItemObj> stocks = new ArrayList<>(); // Изменено название списка
```

```
    @GetMapping
```

```
    public String getAllStocks(Model model) { // Изменено название метода
```

```
        model.addAttribute("stocks", stocks); // Изменено имя атрибута
```

```
        return "stocks"; // Изменено имя возвращаемого представления
```

```
    }
```

```
    @PostMapping
```

```
    public String addStock(@ModelAttribute ItemObj stock) { // Изменено название метода  
и параметра
```

```
        stocks.add(stock);
```

```
        return "redirect:/stocks"; // Изменен URL перенаправления
```

```
    }
```

```
    @GetMapping("/{id}")
```

```
public String getStock(@PathVariable int id, Model model) { // Изменено название метода
```

```
    model.addAttribute("stock", stocks.get(id)); // Изменено имя атрибута
```

```
    return "stock"; // Изменено имя возвращаемого представления
```

```
}
```

```
@PostMapping("/{id}")
```

```
public String deleteStock(@PathVariable int id) { // Изменено название метода
```

```
    stocks.remove(id);
```

```
    return "redirect:/stocks"; // Изменен URL перенаправления
```

```
}
```

```
}
```

Thymeleafproject.java

```
package com.example.ThymeleafProject;
```

```
import org.springframework.boot.SpringApplication;
```

```
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

```
import org.springframework.context.MessageSource;
```

```
import org.springframework.context.annotation.Bean;
```

```
import org.springframework.context.support.ReloadableResourceBundleMessageSource;
```

```
import org.springframework.web.servlet.LocaleResolver;
```

```
import org.springframework.web.servlet.config.annotation.InterceptorRegistry;
```

```
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
```

```
import org.springframework.web.servlet.i18n.LocaleChangeInterceptor;
```

```
import org.springframework.web.servlet.i18n.SessionLocaleResolver;
```

```
import java.util.Locale;
```

```
import org.apache.catalina.Context;
```

```
import org.apache.tomcat.util.descriptor.web.FilterDef;
```

```
import org.apache.tomcat.util.descriptor.web.FilterMap;
```

```
import org.springframework.boot.web.embedded.tomcat.TomcatServletWebServerFactory;
import org.springframework.boot.web.servlet.server.ServletWebServerFactory;
import org.springframework.web.filter.CharacterEncodingFilter;
```

```
@SpringBootApplication
```

```
public class ThymeleafProjectApplication implements WebMvcConfigurer {
```

```
    public static void main(String[] args) {
        SpringApplication.run(ThymeleafProjectApplication.class, args);
    }
```

```
    // Бин для MessageSource
```

```
    @Bean
```

```
    public MessageSource messageSource() {
        ReloadableResourceBundleMessageSource messageSource = new
ReloadableResourceBundleMessageSource();
        messageSource.setBasename("classpath:messages");
        messageSource.setDefaultEncoding("UTF-8");
        return messageSource;
    }
```

```
    // Бин для LocaleResolver
```

```
    @Bean
```

```
    public LocaleResolver localeResolver() {
        SessionLocaleResolver slr = new SessionLocaleResolver();
        slr.setDefaultLocale(Locale.ENGLISH); // Английский язык по умолчанию
        return slr;
    }
```

```
    // Бин для LocaleChangeInterceptor
```

@Bean

```
public LocaleChangeInterceptor localeChangeInterceptor() {  
    LocaleChangeInterceptor lci = new LocaleChangeInterceptor();  
    lci.setParamName("lang");  
    return lci;  
}
```

// Добавление интерцептора для перехвата изменений локали

@Override

```
public void addInterceptors(InterceptorRegistry registry) {  
    registry.addInterceptor(localeChangeInterceptor());  
}
```

// Настройка кодировки UTF-8 для Tomcat

@Bean

```
public ServletWebServerFactory servletContainer() {  
    TomcatServletWebServerFactory tomcat = new  
TomcatServletWebServerFactory() {  
        @Override  
        protected void postProcessContext(Context context) {  
            FilterDef filterDef = new FilterDef();  
            filterDef.setFilterName("setCharacterEncodingFilter");  
  
filterDef.setFilterClass(CharacterEncodingFilter.class.getName());  
            filterDef.addInitParameter("encoding", "UTF-8");  
            filterDef.addInitParameter("forceEncoding", "true");  
            context.addFilterDef(filterDef);  
  
            FilterMap filterMap = new FilterMap();  
            filterMap.setFilterName("setCharacterEncodingFilter");
```



```

        filterMap.addURLPattern("/");
        context.addFilterMap(filterMap);
    }
};
return tomcat;
}
}

```

Websecurityconfig.java

```
package com.example.ThymeleafProject;
```

```

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.http.HttpMethod;

import
org.springframework.security.config.annotation.authentication.builders.AuthenticationManag
erBuilder;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import
org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

import org.springframework.security.core.userdetails.User;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.provisioning.InMemoryUserDetailsManager;
import org.springframework.security.web.SecurityFilterChain;

```

@Configuration

@EnableWebSecurity

```
public class WebSecurityConfig {
```

// Определите цепочку фильтров безопасности

@Bean

```
public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {  
    http  
        .authorizeRequests(authorize -> authorize  
            .requestMatchers("/", "/home", "/login").permitAll()  
            .requestMatchers(HttpMethod.POST, "/stocks").authenticated() // Изменено  
с /patients на /stocks  
            .anyRequest().authenticated()  
        )  
        .formLogin(form -> form  
            .loginPage("/login")  
            .defaultSuccessUrl("/stocks", true) // Изменено с /patients на /stocks  
            .permitAll()  
        )  
        .logout(logout -> logout.permitAll());  
    return http.build();  
}
```

// Настройка пользователей в памяти

@Bean

```
public InMemoryUserDetailsManager userDetailsService() {  
    UserDetails user = User.withDefaultPasswordEncoder()  
        .username("user")  
        .password("password")  
        .roles("ADMIN")  
        .build();  
    return new InMemoryUserDetailsManager(user);  
}
```

```
// Настройка менеджера аутентификации
@Autowired
public void configureGlobal(AuthenticationManagerBuilder auth) throws Exception {
    auth
        .inMemoryAuthentication()
        .withUser("user")
        .password("{noop}password")
        .roles("ADMIN");
}
}
```

```
Application.properties
server.port=8088
#logging.level.org.springframework.security=DEBUG
spring.http.encoding.charset=UTF-8
spring.http.encoding.enabled=true
spring.http.encoding.force=true
```

```
messages_en.properties
login.title=Stock Management System Login
login.header=Login to Stock Management System
login.username=Username
login.password=Password
login.submit=Login
stock.title=Stock
stock.info=Stock Information
stocks.header=Stocks
```

stocks.name=Name  
stocks.id=ID  
stocks.date=Purchase Date  
stocks.details=Details  
stocks.delete=Delete  
stocks.add=Add New Stock  
stocks.stockName=Stock Name  
stocks.stockId=Stock ID  
stocks.purchaseDate=Purchase Date  
stocks.submit=Submit  
stock.name=Name  
stock.id=ID  
stock.date=Purchase date  
stock.return=Return to stock list  
stocks.title=Stock

messages\_ru.properties  
login.title=Вход в систему управления акциями  
login.header=Вход в систему управления акциями  
login.username=Имя пользователя  
login.password=Пароль  
login.submit=Войти  
stock.title=Акция  
stock.info=Информация об акции  
stocks.header=Акции  
stocks.name=Название  
stocks.id=Идентификатор  
stocks.date=Дата покупки  
stocks.details=Подробнее

stocks.delete=Удалить  
stocks.add=Добавить акцию  
stocks.stockName=Название акции  
stocks.stockId=ID акции  
stocks.purchaseDate=Дата покупки  
stocks.submit=Отправить  
stock.name=Название акции  
stock.id=ID акции  
stock.date=Дата покупки  
stock.return=Вернуться к списку акций  
stocks.title=Акции

style.css

/\* Основные стили \*/

```
body {  
    font-family: 'Arial', sans-serif;  
    margin: 0;  
    padding: 0;  
    background-color: #f4f4f4;  
    color: #333;  
}
```

```
.container {  
    width: 90%;  
    max-width: 1200px;  
    margin: 20px auto;  
    padding: 15px;  
    background: white;  
    border-radius: 10px;
```

```
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
    overflow: hidden;
}

h1, h2 {
    color: #5cb85c; /* Зеленый цвет */
    text-align: center;
    margin-bottom: 20px;
}

/* Стили таблиц */
table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 20px;
}

th, td {
    padding: 10px;
    border: 1px solid #ddd;
    text-align: left;
}

th {
    background-color: #5cb85c; /* Зеленый цвет */
    color: white;
}

tr:nth-child(even) {
```

```
background-color: #f2f2f2;
}

tr:hover {
    background-color: #e2e2e2;
}

/* Стили форм и элементов ввода */
form {
    margin-top: 20px;
    text-align: left;
    padding: 20px;
    background: #fff;
    border-radius: 5px;
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}

label {
    margin-top: 10px;
    display: block;
    font-weight: bold;
}

input[type="text"],
input[type="number"],
input[type="date"],
input[type="password"],
select {
    width: 100%;
```

```
padding: 8px;
margin-top: 5px;
border: 1px solid #ddd;
border-radius: 4px;
box-sizing: border-box;
}
```

```
/* Стили кнопок */
```

```
.delete-button, .details-button, button, input[type="submit"] {
padding: 10px 15px;
border: none;
border-radius: 4px;
cursor: pointer;
margin-top: 10px;
width: 100%;
text-align: center;
}
```

```
.details-button { /* Зеленая кнопка для Details */
background-color: #5cb85c;
color: white;
}
```

```
.details-button:hover {
background-color: #4cae4c;
}
```

```
.delete-button { /* Красная кнопка для Delete */
background-color: #d9534f;
```



```
    color: white;
}
```

```
.delete-button:hover {
    background-color: #c9302c;
}
```

```
.submit-button { /* Зеленая кнопка для Submit */
    background-color: #5cb85c;
    color: white;
}
```

```
.submit-button:hover {
    background-color: #4cae4c;
}
```

```
/* Стили для страницы входа и информации об акции */
```

```
.login-form,
.stock-info {
    max-width: 400px;
    margin: 0 auto;
    padding: 20px;
    background: white;
    border-radius: 10px;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}
```

```
/* Стили ссылок */
```

```
a {
```

```
color: #4a90e2;  
text-decoration: none;  
}
```

```
a:hover {  
    text-decoration: underline;  
}
```

```
/* Медиа-запросы для адаптивного дизайна */
```

```
@media (max-width: 768px) {  
    .container, .login-form, .stock-info {  
        width: 95%;  
        padding: 10px;  
    }  
}
```