МИНОБРНАУКИ РОССИИ

федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева»

(Самарский университет)

Институт информатики, математики и электроники

Факультет математики

Кафедра безопасности информационных систем

КУРСОВАЯ РАБОТА

**Разработка веб-интерфейса для авторизации и управления сетевым устройством**

по специальности 10.05.01 Компьютерная безопасность

(уровень специалитета)

специализация №7 «Информационно-аналитическая и техническая экспертиза компьютерных систем»

Студент 6542-100501D гр.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Е.В. Петров

Руководитель КР

к.ф.-м.н., А.В. Графкин

Нормоконтролер / /

Самара 2020

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc49718440)

[1. Постановка задачи и обоснование метода решения задачи 4](#_Toc49718441)

[1.1 Анализ предметной области 4](#_Toc49718442)

[1.1.1 Определение инструментов для создания веб-интерфейса 4](#_Toc49718443)

[2. проектирование и программная реализация метода решения задачи 5](#_Toc49718444)

[2.1 Обоснование и выбор комплекса программных средств, ЯП, ОС. 5](#_Toc49718445)

[2.2 Проектирование интерфейса пользователя 10](#_Toc49718446)

[2.2.1 Инструменты и технологии 10](#_Toc49718447)

[2.2.2 Прототип пользовательского интерфейса 11](#_Toc49718448)

[2.3 Программная реализация 13](#_Toc49718449)

[2.3.1 Серверная часть 15](#_Toc49718450)

[2.3.2 Клиентская часть 18](#_Toc49718451)

[3. Тестирование пользовательского интерфейса 24](#_Toc49718452)

[ЗАКЛЮЧЕНИЕ 26](#_Toc49718453)

[Определения. Обозначения и сокращения 27](#_Toc49718454)

[Список используемых источников 28](#_Toc49718455)

[ПРИЛОЖЕНИЕ 1 29](#_Toc49718456)

**ВВЕДЕНИЕ**

На сегодняшний день существует множество сетевых устройств, созданных для различных целей. Сложно представить современную жизнь без таких устройств, как маршрутизаторов, коммутаторов, межсетевых экранов.

В большинстве своем, многие сетевые устройства работают по своим определенным протоколам, у них как правило имеется своя файловая система, операционная система и пользовательский интерфейс.

Многие производители сетевых устройств предпочитают не тратить лишние средства на оформления пользовательского интерфейса, а зачастую реализуют его в виде команд в терминале. Это, конечно, имеет свои преимущества для тех, кто хорошо разбирается в администрировании таких устройств и как правило для обычных пользователей этот интерфейс является не очень понятным.

Поэтому всё чаще стали появляться устройства, управление которыми осуществляется через графический интерфейс, это может быть окно в Windows или Linux системах, или WEB-страница. В основном, в качестве графического интерфейса используются web-страницы, т.к они универсальны, их можно просматривать с любого компьютера, телефона, планшета, под любой операционной системой, в любом браузере.

1. **Постановка задачи и обоснование метода решения задачи**

Основной целью данной работы является создание веб интерфейса для управления контроллерами

Для достижения цели данной работы были поставлены следующие задачи:

1. Изучить особенности разработки WEB-страниц
2. Выбрать инструмент, для реализации WEB интерфейса
3. Разработать интерфейс для управления контроллером

**1.1 Анализ предметной области**

**1.1.1 Определение инструментов для создания веб-интерфейса**

* **HTML** – это способ разметки текста. Задача осуществляется при помощи тегов, помогающих разделить структуру на абзацы, разделы, списки, цитаты и параграфы. Он имеет множество параметров, благодаря которым производится оформление документа. Редактируется любой программой, добавляя теги форматирования. При исключительно его использовании реализация формата статична. Дополнительных возможностей для юзера не будет, он сможет только читать.
* **CSS** – таблицы стилей, помогающие дополнить сайт новым функционалом. Благодаря ему код можно подкорректировать, сделав его коротким и оптимизированным. CSS позволяет улучшить оформление и разгрузить страницы. Обычно определение стилей выносится разработчиками в другие файлы.
* **PHP** относится к серверным языкам, дает возможность пользователям работать с разными задачами самостоятельно, то есть комментировать, голосовать, публиковать изображения и прочее. Закрепление изменения производится после обновления..
* **Java-Script** – это вспомогательный клиентский инструмент для добавления несложных опций, не требующих перезагрузки.
* **Flash** тоже считается вспомогательным для создания анимации и интерактивности в отдельных блоках.
* **Bootstrap** — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

**2. проектирование и программная реализация метода решения задачи**

При проектировании веб-интерфейса для авторизации и управления сетевым устройством необходимо учитывать программные и аппаратные особенности сервера, где будет развернут проект.

**2.1 Обоснование и выбор комплекса программных средств, ЯП, ОС.**

В качестве ОС могут быть выбраны следующие системы:

**Windows Server** —представляет собой группу операционных систем, разработанных корпорацией Майкрософт, которая поддерживает управление на уровне предприятия, хранение данных, приложений и коммуникаций. Предыдущие версии Windows Server сосредоточились на стабильности, безопасности, сетей, а также различные улучшения в файловой системе. Другие улучшения также включали улучшения развертывания технологий, а также расширение поддержки аппаратных средств. Microsoft также создала специализированная SKUs из Windows Server, которые сосредоточены на внутреннем и рынках малого бизнеса.

Windows Server предназначен для управления аппаратным обеспечением серверов, обладающие требуемыми для этого функциональными особенностями. Также эти серверные ОС содержат специализированные службы (программные компоненты), предназначенные для организации, мониторинга и управления ИТ-инфраструктурой предприятия, начиная от управления адресным пространством протокола IP и учетными записями пользователей и заканчивая организацией отказоустойчивых сетевых сервисов.

**Linux Ubuntu** – это бесплатная серверная операционная система на базе ядра Linux. Ubuntu можно использовать в качестве платформы для Web-серверов, серверов баз данных, DNS-серверов, файловых серверов и других типов серверов. Ubuntu - популярный дистрибутив Linux, в том числе и серверный вариант, который активно используется организациями разных размеров, за счет того что главной особенностью Ubuntu, да и всех серверных операционных систем на базе Linux, является надежность, производительность и безопасность.

В качестве операционной системы была выбрана Ubuntu, за счет ее универсальности, бесплатным лицензированием и легковесностью (Ubuntu требует намного меньше ресурсов системы, в отличии от Windows Server и других систем на основе OS Windows)

Существует множество языков программирования, для создания сервера WEB-приложения, основными серверными языками программирования являются:

**PHP** — скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

В области веб-программирования, в частности серверной части, PHP — один из популярных сценарных языков (наряду с JSP, Perl и языками, используемыми в ASP.NET).

Популярность в области построения веб-сайтов определяется наличием большого набора встроенных средств и дополнительных модулей для разработки веб-приложений. Основные из них:

* автоматическое извлечение POST- и GET-параметров, а также переменных окружения веб-сервера в предопределённые массивы;
* взаимодействие с большим количеством различных систем управления базами данных через дополнительные модули (MySQL, MySQLi, SQLite, PostgreSQL, Oracle (OCI8), Oracle, Microsoft SQL Server, Sybase, ODBC, mSQL, IBM DB2, Cloudscape и Apache Derby, Informix, Ovrimos SQL, Lotus Notes, DB++, DBM, dBase, DBX, FrontBase, FilePro, Ingres II, SESAM, Firebird / InterBase, Paradox File Access, MaxDB, Интерфейс PDO, Redis);
* автоматизированная отправка HTTP-заголовков;
* работа с HTTP-авторизацией;
* работа с cookies и сессиями;
* работа с локальными и удалёнными файлами, сокетами;
* обработка файлов, загружаемых на сервер;
* работа с XForms.

**Ruby** — интерпретируемый мультипарадигмальный язык программирования: динамический, объектно-ориентированный, рефлективный, императивный, функциональный.

Этот язык активно используется в веб-разработке, в системном администрировании и в работе операционных систем (Mac OS X, Linux, BSD). Наиболее популярным Ruby-фреймворком для разработки веб-приложений является Ruby on Rails, на основе которого построены многие известные сайты и веб-приложения.

Ruby следует принципу «наименьшей неожиданности»: программа должна вести себя так, как ожидает программист.

В противовес машинно-ориентированным языкам, программы на которых пишутся так, чтобы компьютеру было максимально удобно их выполнять, Ruby — язык, ориентированный на удобство человека: он позволяет разработчику максимально быстро и просто выполнить задачу, хотя, возможно, это и потребует дополнительного времени работы компьютера. В современном мире вычислительные ресурсы стоят значительно дешевле, нежели время разработчиков, поэтому этот подход в большинстве случаев не только более комфортный для разработчиков, но и экономически более целесообразный с точки зрения бизнеса.

Ruby является мультипарадигменным языком: он поддерживает процедурный стиль, объектно-ориентированный и функциональный. Он поддерживает отражение, метапрограммирование, информацию о типах переменных на стадии выполнения. Всё это позволяет удобно работать и эффективно решать широкий класс задач.

**Python** (Питон или Пайтон) — высокоуровневый интерпретируемый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода.

Язык поддерживает несколько парадигм разработки: структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное программирование.

**Java** — это достаточно универсальный язык программирования, который часто используется для веб-разработки и для разработки под Android.

Java — это объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Дата официального выпуска — 23 мая 1995 года.

Программы на Java транслируются в байт-код, который затем выполняется виртуальной машиной Java (JVM). JVM — это программа, которая обрабатывает байтовый код и передает инструкции оборудованию как интерпретатор. Достоинством подобной реализации является независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует JVM.

Другой важной особенностью технологии Java является гибкая система безопасности благодаря тому, что исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером) вызывают немедленное прерывание.

Часто к недостаткам концепции виртуальной машины относят то, что исполнение байт-кода виртуальной машиной может снижать производительность программ и алгоритмов, реализованных на языке Java. Программы, написанные на Java, имеют репутацию более медленных и занимающих больше оперативной памяти, чем написанные на языке Си. Однако, если сравнивать Java с интерпретируемыми языками, которые наиболее часто используются в веб-разработке, то производительность Java обычно заметно выше.

**2.2 Проектирование интерфейса пользователя**

**2.2.1 Инструменты и технологии**

В качестве языка программирования был выбран Java за счет его универсальности и кросс-платформенности.

Проект был создан в IDE IntelliJ IDEA 2018

При разработке был использован фреймворк Java – Spring Framework

**Spring Framework** (или коротко Spring) — универсальный фреймворк с открытым исходным кодом для Java-платформы.

Spring стал широко распространённым в Java-сообществе главным образом как альтернатива и замена модели Enterprise JavaBeans. Spring предоставляет бо́льшую свободу Java-разработчикам в проектировании; кроме того, он предоставляет хорошо документированные и лёгкие в использовании средства решения проблем, возникающих при создании приложений корпоративного масштаба.

Между тем, особенности ядра Spring применимы в любом Java-приложении, и существует множество расширений и усовершенствований для построения веб-приложений на Java Enterprise платформе. По этим причинам Spring приобрёл большую популярность и признаётся разработчиками как стратегически важный фреймворк.

Для оформления страниц был выбран фреймворк Bootstrap

**Bootstrap** — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

**Apache Maven** — фреймворк для автоматизации сборки проектов на основе описания их структуры в файлах на языке POM, являющемся подмножеством XML. Проект Maven издаётся сообществом Apache Software Foundation, где формально является частью Jakarta Project.

**2.2.2 Прототип пользовательского интерфейса**

Интерфейс схематично можно представить следующим образом (Рис. 1):

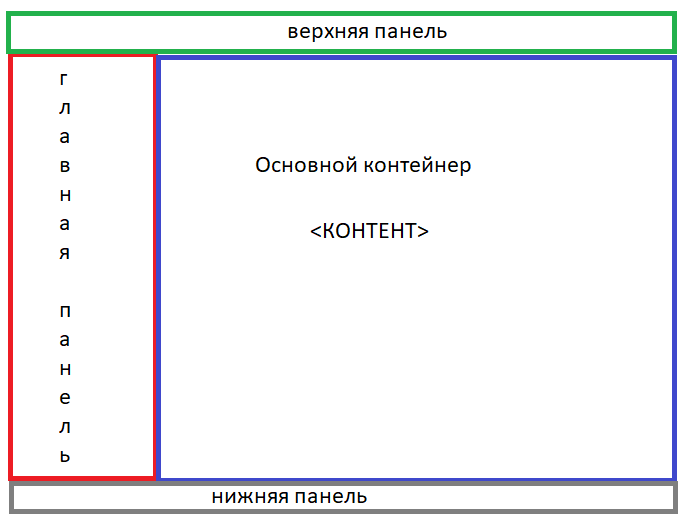


Рисунок 1 - Схема интерфейса WEB-приложения

Как видно из рисунка, интерфейс разделен на 4 части.

**Верхняя панель** – здесь будет надпись “Панель управления” и некоторые декоративные детали.

**Главная панель** – это панель, в которой будут располагаться все вкладки приложения, это навигационная панель, для перехода на соответствующие страницы сайта.

**Нижняя панель** – тут будут данные об разработчике

**Основной контейнер**– здесь будет находиться весь контент, т.е это то место, куда будут подставляться те или иные функции для управления устройством.

Определим вкладки, которые будут доступны в главной панели:

* Главная страница – это приветственная страница, на которую будет переслан пользователь, после авторизации
* Страница информации – здесь будет краткая информация по сетевому устройству
* Настройки – здесь будут страницы, которые отвечают за настройку сетевого устройства
* Файлы – отображение файлов конфигурации, загруженных на устройство
* Администрирование – здесь будет управление пользователями и группами, а так же кнопка отката до заводских настроек.
* Выход – страница logout

Определим роли пользователей в нашей системе

* **Администратор** – доступны все возможности системы, главная роль.
* **Пользователь** – доступны вкладки – Главная страница, Страница информации, настройки, Файлы и Выход (запрещена страница Администрирование)
* **Аноним** – доступны вкладки – Главная страница, Страница информации, Выход

**2.3 Программная реализация**

Для инициализации проекта был использован сайт <https://start.spring.io>. Интерфейс сайта указан на рисунке 2.

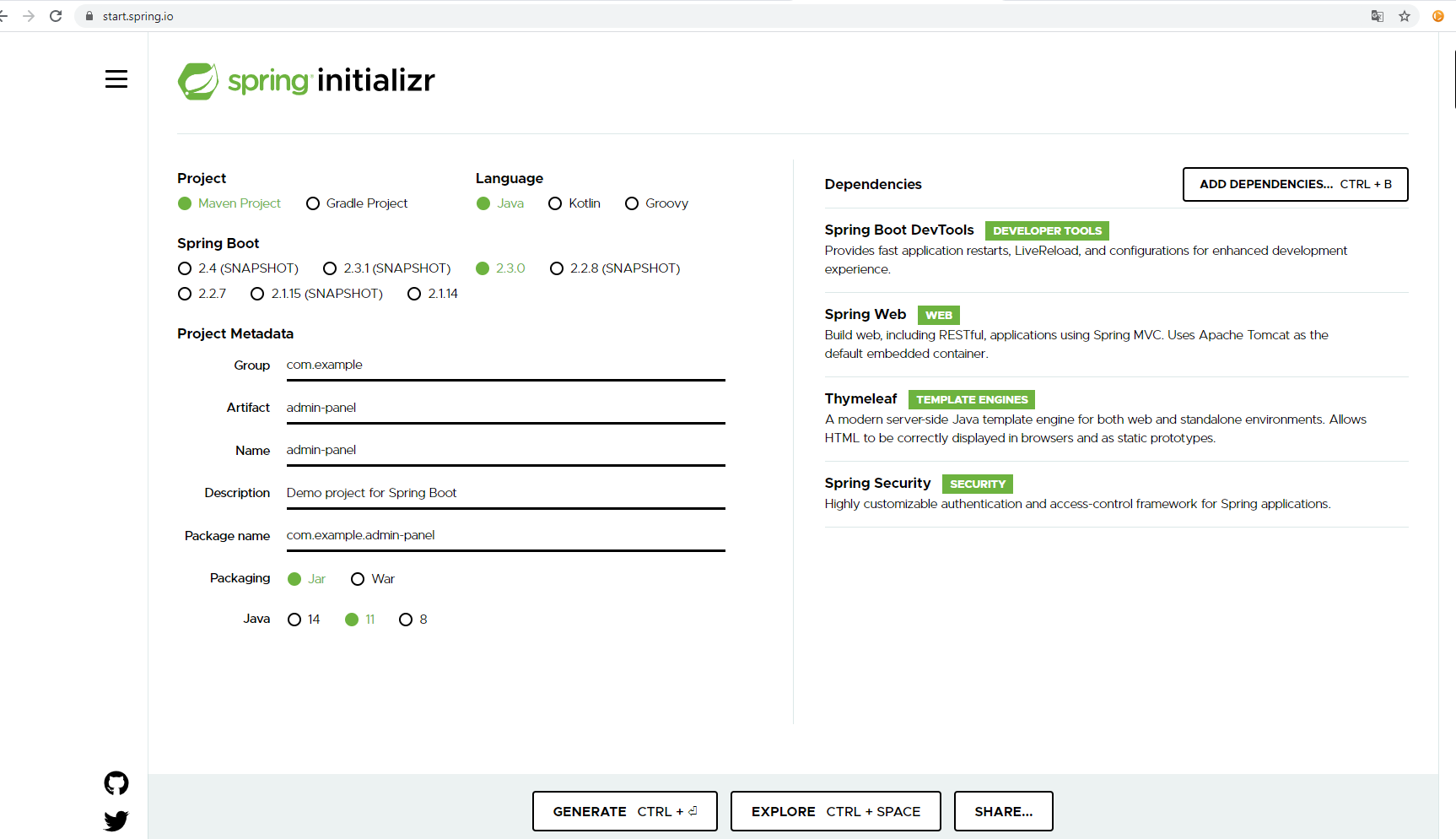


Рисунок 2 – Инициализация проекта

Данный сайт предназначен для создание пустого spring проекта, содержащего в себе лишь необходимые модули для дальнейшей разработки. Здесь выбрали версию java, версию spring-а, название проекта и подключаемые библиотеки Spring Boot, Spring Web, шаблонизатор Thymeleaf, Spring Security.

Модули Spring Boot и Spring Web содержат основные библиотеки для создания веб приложений.

Thymeleaf – язык щаблонов, позволяет встраивать в html страницу код, который будет обработан в Java-контроллере. Это позволяет “подставлять” данные из Java в html страницу.

Spring Security содержит библиотеки, связанные с безопасностью приложения.

Листинг содержимого проекта показан на рисунке 3.

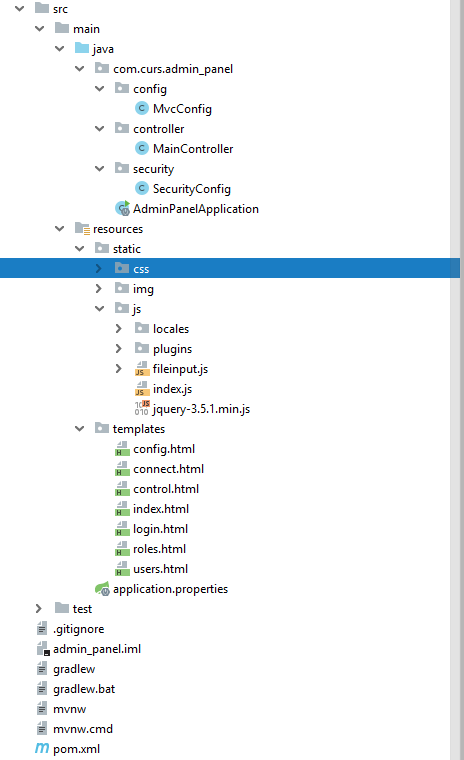


Рисунок 3 – Содержимое проекта

Для сборки проекта используется Maven. Его конфигурация находится в файле pom.xml (Рис. 4)

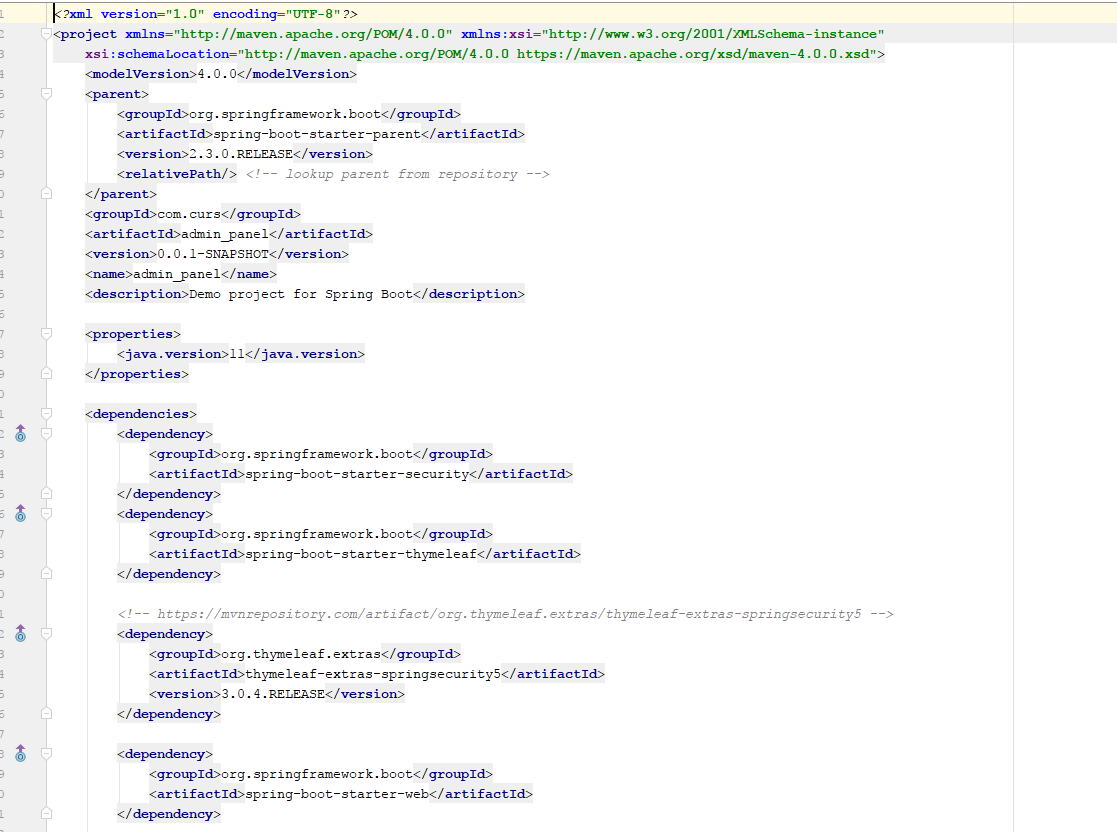


Рисунок 4 - Конфигурация Maven, файл pom.xml

В данном файле указываются параметры для сборки приложения, зависимости, версия приложения, название, описание и т.д

Проект разделен на серверную часть - бэкэнд (java классы com.curs.admin-panel/\*) и клиентскую часть – фронтэнд (resources/static/\* и resources/template)

**2.3.1 Серверная часть**

Серверная часть состоит из классов-конфигурация (MvcConfig и SecurityConfig), класса контроллера – MainController и главного Main класса, точки входа в приложение – AdminPanelApplication. Рассмотрим их подробнее:

AdminPanelApplication.java – класс, в котором находится точка входа в приложение, листинг класса показан на рис. 5

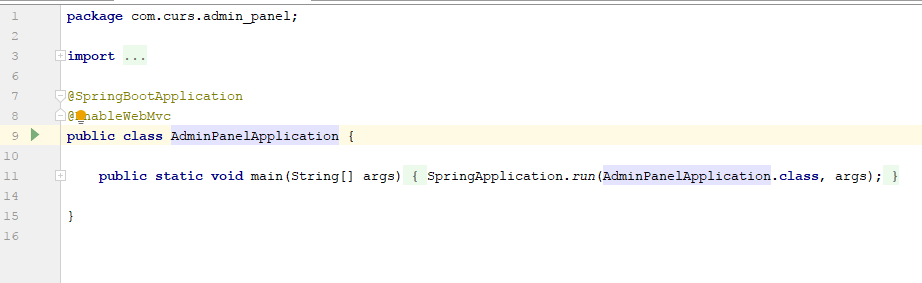


Рисунок 5 – листинг AdminPanelApplication.java

В конфигурации MVC указан путь к статическим файлам проекта /static. Листинг класса MvcConfig.java представлен на рис. 6

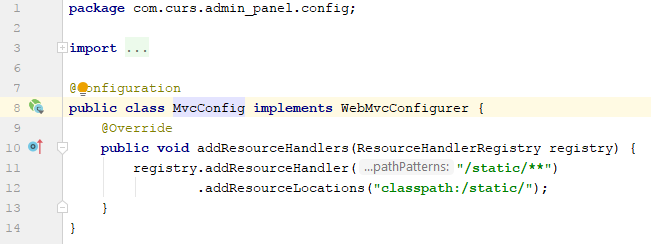


Рисунок 6 – листинг MvcConfig.java

В конфигурации безопасности SecurityConfig.java определены – страница логина /login, страницы, доступные для неавторизованных пользователей, опция – указывающая, что все остальные страницы доступны только авторизованным пользователям *anyRequest().authenticated(),* а так же статичные пользователи с определенными ролями в методе *configureGlobal*. Листинг класса представлен на рис. 7

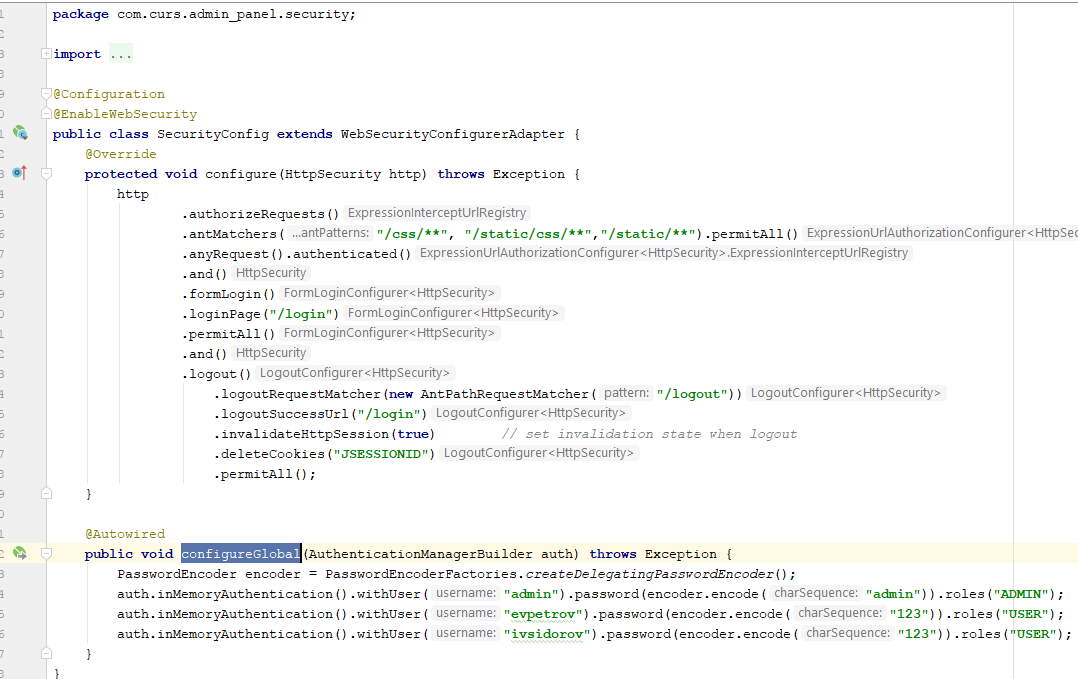


Рисунок 7 – листинг SecurityConfig.java

Основным классом, отвечающим за то, какие страницы показывать при обращении к какому-либо URL является MainController.java В нем указан маппинг страниц по определенным запросам на URL адреса. Листинг класса представлен на рис.8

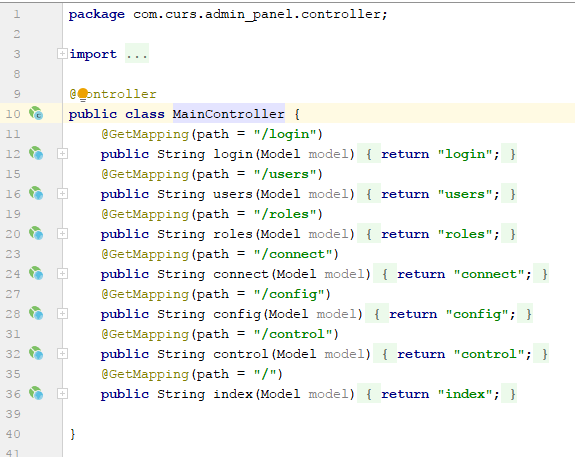


Рисунок 8 – листинг MainController.java

**2.3.2 Клиентская часть**

Файлы клиентской части расположены в папке resources. Они делятся на две части – статичные и шаблонные.

К статичным ресурсам относятся css таблицы стилей, js-скрипты, медиа-файлы (картинки, музыка и прочее)

К шаблонным относятся страницы html страницы, обрабатываемые с помощью шаблонизатора thymeleaf.

Рассмотрим статические ресурсы. Файлы проекта, относящиеся к статическим ресурсам показаны на рис. 9

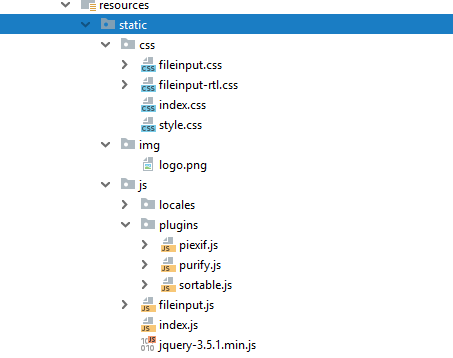


Рисунок 9 – статичные ресурсы проекта

В проект были включены следующие библиотеки для клиентской части:

Fileinput – библиотека, отвечающая за загрузку файлов на сервер и выгрузку с сервера клиентам, содержит контейнеры для загрузки файлов с клиента на сервер. Файлы библиотека – fileinput.js, fileinput.css, fileinput-rtl.css

Jquery-3.5.1 — набор функций JavaScript, фокусирующийся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API для работы с AJAX

А так же плагины для работы библиотек, расположенные в папке plugins.

logo.png – картинка с логотипом Самарского Университета

index.css – таблица стилей для страницы index.html

style.css – основная таблица стилей для всех страниц

index.js – скрипт для страницы index.html, листинг представлен на рис. 10

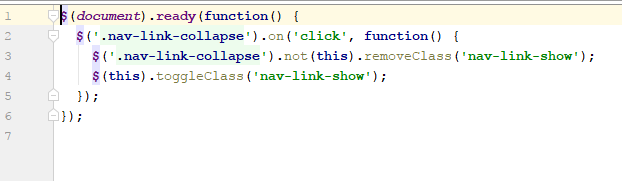


Рисунок 10 – листинг index.js

В данном скрипте описана работа навигационной панели (обработчик нажатия на пункты из панели)

Далее рассмотрим динамические ресурсы. Это html страницы, находятся в папке template. Во всех страницах подключен обработчик шаблонов thymeleaf, библиотека jquery, и таблицы стилей. Для подключения шаблонизатора используются следующие строки:

**xmlns:th="http://www.thymeleaf.org"  
xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity4"**>

Для подключения bootstrap используется следующая строчка:

<**link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous"**>

Подключение jquery:

<**script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"**></**script**>  
<**script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"**></**script**>  
<**script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js" integrity="sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI" crossorigin="anonymous"**></**script**>

Рассмотрим страницу login.html, тело данной страницы состоит из формы, которая отсылает POST-запрос на /login с параметрами username и password. Основной фрагмент данной страницы можно увидеть на рис. 11



Рисунок 11 – фрагмент кода login.html

Страница логина выглядит следующим образом (рис.12):

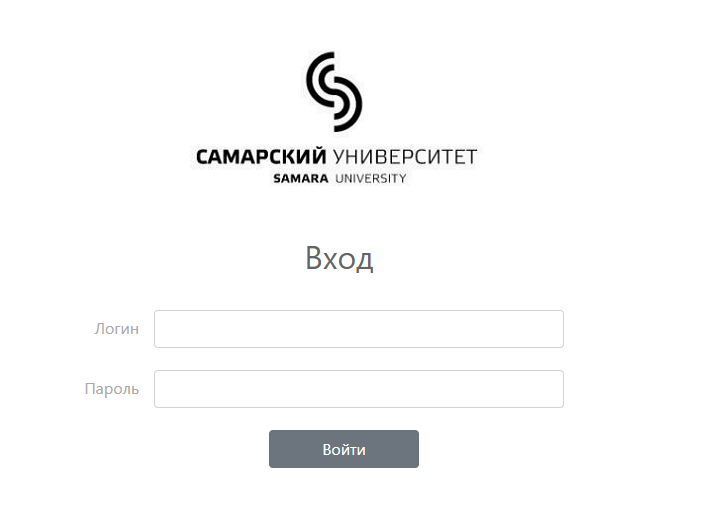


Рисунок 12 – страница логина

Далее рассмотрим основные страницы приложения.

После успешной авторизации, пользователь попадает на страницу index.html. На данной странице присутствуют элементы, одинаковые для всех последующих страниц – config.html, connect.html, users.html, roles.html, control.html, а именно – верхняя панель, главная панель и нижняя панель. Рассмотрим подробнее реализацию этих панелей.

Верхняя панель – листинг представлен на рис. 13



Рисунок 13 – листинг верхней панели приложения

В нижней панели указаны данные о разработчике (рис. 14)

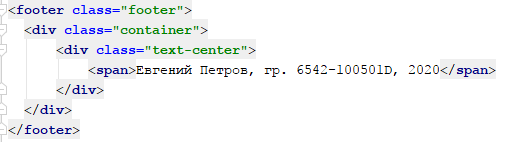


Рисунок 14 – листинг нижней панели приложения

Листинг главной панели (общей для всех страниц) указан в приложении 1.

Основной контейнер для всех страниц расположен между тегами <main></main>. Для страницы index.html основной контейнер выглядит следующим образом (рис. 15):



Рисунок 15 – основной контейнер index.html

На странице config.html указаны блоки ввода файлов конфигурации на сервер (рис. 16).



Рисунок 16 – листинг config.html

На страницах users.html и roles.html отображены таблицы соответственно пользователей и ролей и кнопки для создания пользователя и роли. Пример страниц показан на рисунке 17

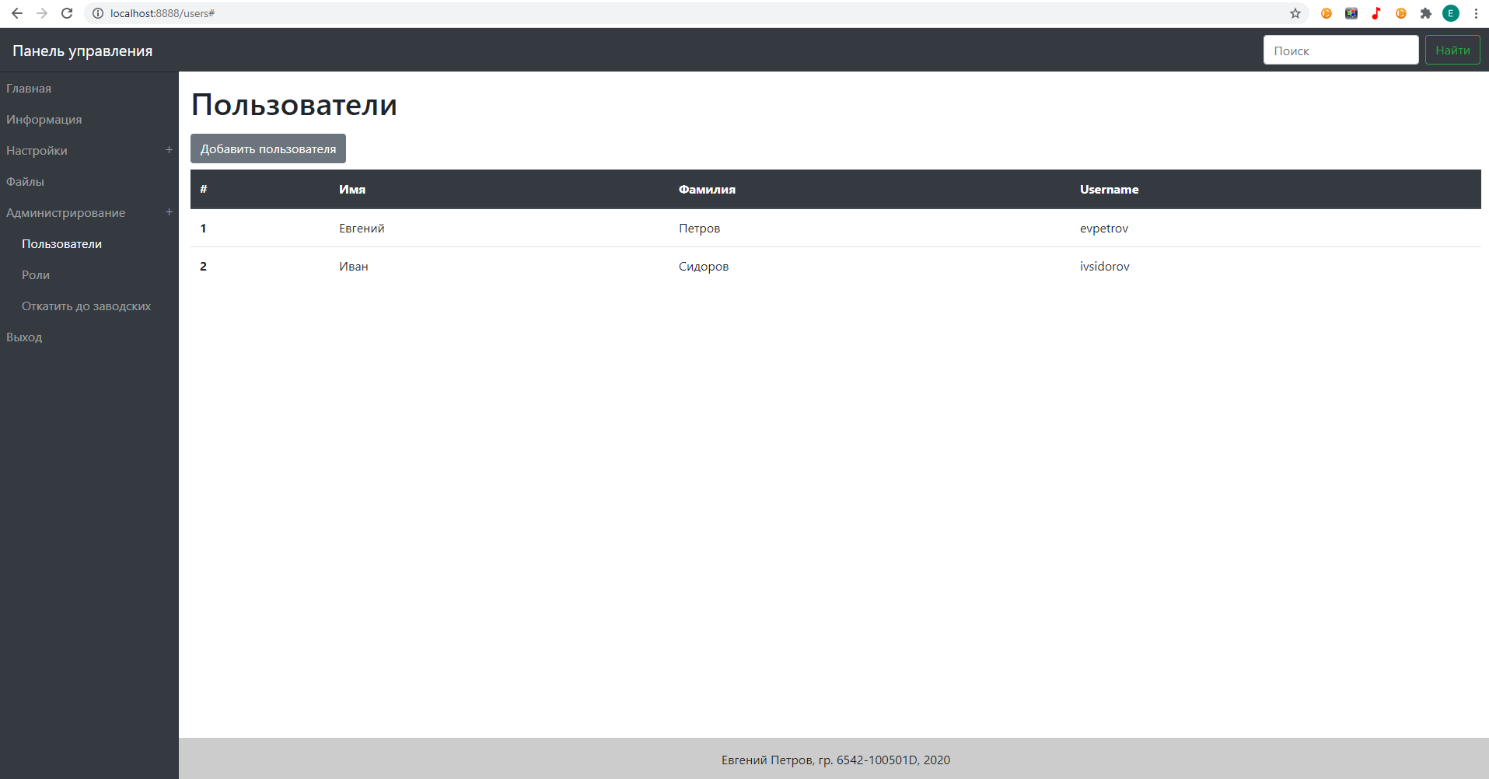


Рисунок 17 – пример интерфейса страницы пользователей

**3. Тестирование пользовательского интерфейса**

Для тестирования интерфейса, запустим приложение. Расположим его на порту 8888. При переходе на страницу требующую авторизации идет перенаправление на страницу логина. Авторизуемся под пользователем admin, как видно на рисунке 18, данный пользователь имеет доступ ко всем вкладкам.

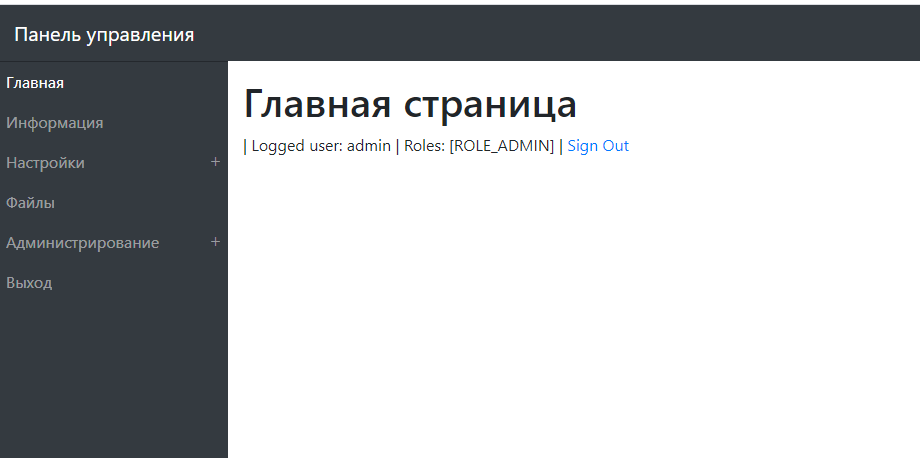


Рисунок 18 – авторизация от пользователя admin

Если зайти от пользователя с ролью USER, например пользователь “evpetrov” – то увидим, что данному пользователю не доступна вкладка администрирование (рис. 19)

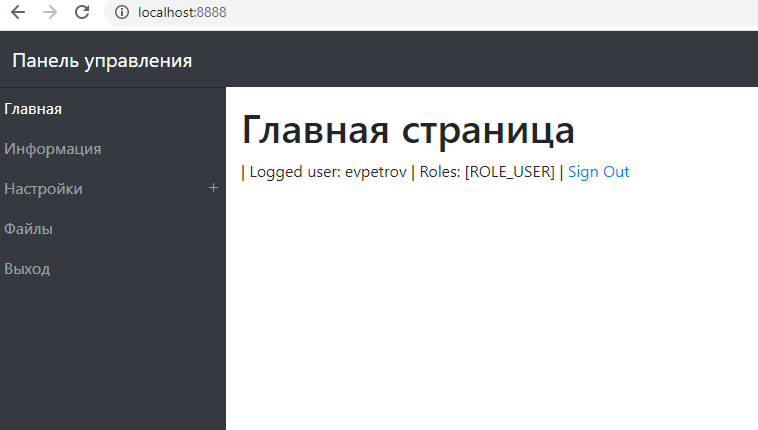


Рисунок 19 – авторизация от пользователя evpetrov

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения данной работы были изучены особенности разработки Web-страниц, реализован пользовательский интерфейс для управления контроллерами. При помощи среды программирования IntelliJ Idea 2018, с использованием фреймворка Spring Framework, на языке Java. При разработке приложения использовались такие библиотеки, как Jquery, Bootstrap, Fileinput и другие. В приложении были реализованы роли, ограничивающие правила доступа соответствующих пользователей.

**Определения. Обозначения и сокращения**

ОС – Операционная система

ЯП – Язык программирования

ИТ – Информационные Технологии

IP – Internet Protocol

JVM – Java Virtual Machine (Виртуальная машина Java)

**Список используемых источников**

1. Уоллс Крейг. Spring в действии[Текст] / Уоллс Крейг.    – ДМК Пресс: 2015. – 754 с.
2. Документация Bootstrap 4.5 [Электронный ресурс] – Режим доступа: <https://bootstrap-4.ru/docs/4.5>
3. Документация Spring Framework [Электронный ресурс] – Режим доступа: <https://docs.spring.io/spring/docs/current/spring-framework-reference/>
4. Бен Хенник. Html и CSS. Путь к совершенству[Текст] / Бен Хенник.    – Питер: 2011. – 336 с.

**ПРИЛОЖЕНИЕ 1**

<**nav class="navbar navbar-expand-lg navbar-dark fixed-top bg-dark"**>  
 <**a class="navbar-brand" href="#"**>Панель управления</**a**>  
 <**button  
 class="navbar-toggler"  
 type="button"  
 data-toggle="collapse"  
 data-target="#navbarCollapse"  
 aria-controls="navbarCollapse"  
 aria-expanded="false"  
 aria-label="Toggle navigation"** >  
 <**span class="navbar-toggler-icon"**></**span**>  
 </**button**>  
  
 <**div class="collapse navbar-collapse" id="navbarCollapse"**>  
 <**ul class="navbar-nav mr-auto sidenav" id="navAccordion"**>  
 <**li class="nav-item active"**>  
 <**a class="nav-link" href="/"**>Главная <**span class="sr-only"**>(current)</**span**></**a**>  
 </**li**>  
 <**li class="nav-item"**>  
 <**a class="nav-link" href="#"**>Информация</**a**>  
 </**li**>  
 <**li class="nav-item"**>  
 <**a  
 class="nav-link nav-link-collapse"  
 href="#"  
 id="hasSubItems"  
 data-toggle="collapse"  
 data-target="#collapseSubItems2"  
 aria-controls="collapseSubItems2"  
 aria-expanded="false"** >Настройки</**a**>  
 <**ul class="nav-second-level collapse" id="collapseSubItems2" data-parent="#navAccordion"**>  
 <**li class="nav-item"**>  
 <**a class="nav-link" href="/connect"**>  
 <**span class="nav-link-text"**>Подключение</**span**>  
 </**a**>  
 </**li**>  
 <**li class="nav-item"**>  
 <**a class="nav-link" href="/config"**>  
 <**span class="nav-link-text"**>Конфигурация</**span**>  
 </**a**>  
 </**li**>  
 <**li class="nav-item"**>  
 <**a class="nav-link" href="/control"**>  
 <**span class="nav-link-text"**>Управление</**span**>  
 </**a**>  
 </**li**>  
 </**ul**>  
 </**li**>  
 <**li class="nav-item"**>  
 <**a class="nav-link" href="#"**>Файлы</**a**>  
 </**li**>  
 <**li sec:authorize="hasRole('ADMIN')" class="nav-item"**>  
 <**a  
 class="nav-link nav-link-collapse"  
 href="#"  
 id="hasSubItems"  
 data-toggle="collapse"  
 data-target="#collapseSubItems4"  
 aria-controls="collapseSubItems4"  
 aria-expanded="false"** >Администрирование</**a**>  
 <**ul class="nav-second-level collapse" id="collapseSubItems4" data-parent="#navAccordion"**>  
  
  
 <**li class="nav-item"**>  
 <**a class="nav-link" href="/users"**>  
 <**span class="nav-link-text"**>Пользователи</**span**>  
 </**a**>  
 </**li**>  
 <**li class="nav-item"**>  
 <**a class="nav-link" href="/roles"**>  
 <**span class="nav-link-text"**>Роли</**span**>  
 </**a**>  
 </**li**>  
  
 <**li class="nav-item"**>  
 <**a class="nav-link" href="#"**>  
 <**span class="nav-link-text"**>Откатить до заводских</**span**>  
 </**a**>  
 </**li**>  
 </**ul**>  
 </**li**>  
 <**li class="nav-item "**>  
 <**a class="nav-link" href="/logout"**>Выход</**a**>  
 </**li**>  
 </**ul**>  
 <**form class="form-inline ml-auto mt-2 mt-md-0"**>  
 <**input class="form-control mr-sm-2" type="text" placeholder="Поиск" aria-label="Search"**>  
 <**button class="btn btn-outline-success my-2 my-sm-0" type="submit"**>Найти</**button**>  
 </**form**>  
 </**div**>  
</**nav**>