**Задание № 12**

**Тема: Финальное задание. Spring MVC + Spring Security**

Обращаю ваше внимание на:

1. Ваши коммиты должны быть постепенными, чтобы не одним коммитом всё, а несколькими коммитами, в описании должно быть понятно какую функциональность вы добавили.

2. Примеры приложений по Spring MVC и Spring JDBC положу в папочку spring examples. Постарайтесь разобраться в написанном вами приложении, чтобы вы не просто взяли и скопировали с источников, а понимали

3. Полезная ссылка по MVC и Security: [**https://www.baeldung.com/spring-mvc-view-resolver-tutorial**](https://www.baeldung.com/spring-mvc-view-resolver-tutorial)

Не стесняйтесь задавать вопросы!!!

Лучше пусть будет вопросов много, но вы разберётесь, чем их не будет, и вы пойдёте дальше без осознания азов.

Условие

1) Создайте в git ветку из master – task-12 и переключитесь на неё;

2) Создайте веб приложение на основе вашей предметной области, разработанной в таске SQL - без использования Spring Boot

3) При реализации вашего приложения необходимо использовать как минимум 3 вьюшки: профиль пользователя, общая страница вашей предметной области, корзина / список выбранных услуг – все остальные исключительно на ваше усмотрение (если эти три пункта тяжело выделить, напишите мне, постараемся выделить вместе с вами). При нажатии на пункт из корзины или общего списка показывать детальную информацию о товаре / услуге (всплывающее окно, например)

В качестве средства создания страниц РЕКОМЕНДУЮ использовать thymeleaf (<https://www.thymeleaf.org/>).

4) В вашем приложении должен чётко просматриваться паттерн MVC, это касается структуры проекта

5) Для хранения информации используйте embedded базу данных H2. Для работы с базой данных используйте Spring JDBC. Работа с базой данных должна быть вынесена в DAO слой (если вы не вынесли его ранее), а общение между контроллерами и DAO должно быть через сервисы (опять же, если вы этого не сделали ранее).

(<https://stackoverflow.com/questions/50759853/setting-up-in-memory-h2-database-without-spring-boot>) – если всё-таки сконфигурить не получиться используйте MySQL, но ПОСТАРАЙТЕСЬ подключить h2 будет полезно.

5) Реализуйте CRUD операции для каждой вашей сущности (те операции, которые вам необходимы для реализации вашего приложения)

6) При добавлении товара / услуги в корзину (если это какие-то услуги, то количество в день ограничено, если билеты, то в зале на спектакль, может быть только определённое количество мест) учитывайте, что товар в один момент может закончиться из чего следует, что вам нужно реализовать обработку ошибок (@ExceptionHandler – при конфигурации через аннотации или @ControllerAdvice)

7) Создайте простую форму входа для пользователя и реализуйте через Spring Security корректный доступа. Не усложняйте себе задачу, постарайтесь реализовать одного пользователя и его доступ ко всем ресурсам вашего приложения. + на доп балл реализуйте несколько ролей

8) Вид конфигурации Spring не имеет значения, можете использовать как XML-based, так и Java-based, только выбирайте что-то одно. Реализовать Dependency Injection через Property и Constructor. Реализуйте бины с разными Scope, например, Singleton и Prototype, или любой другой, но уметь объяснить выбор того или иного scope

9) Реализуйте Interceptor для логирования обработки запроса. (библиотека log4j2)  
  
10) Создайте docker-compose файл, внутри контейнера будет war файл вашего приложения, поднимите Tomcat и положите туда ваше приложения. Тем самым при запуске docker-compose, ваше приложение должно подниматься на 8083 порту и работать 😊

P.s. если вы выбрали всё-таки MySQL, то вам нужно его тоже поднять в докере и настроить коммуникацию с вашим приложением.