Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

Факультет ПИиКТ

Лабораторная работа №4

по дисциплине

«Веб-программирование»

Вариант 82183

Выполнили: студенты группы Р3210

Лукьяненко Никита Игоревич

Киселёв Сергей Владимирович

Преподаватель:

Яркеев Александр Сергеевич

г. Санкт-Петербург

2019 г.

**Задание**

Переписать приложение из лабораторной работы № 3 с использованием следующих технологий:

* Уровень back-end должен быть основан на Spring.
* Уровень front-end должен быть построен на Vue.js с использованием обычных полей ввода HTML
* Взаимодействие между уровнями back-end и front-end должно быть организовано посредством REST API.

Приложение по-прежнему должно включать в себя 2 страницы - стартовую и основную страницу приложения. Обе страницы приложения должны быть адаптированы для отображения в 3 режимах:

* "Десктопный" - для устройств, ширина экрана которых равна или превышает 1064 пикселей.
* "Планшетный" - для устройств, ширина экрана которых равна или превышает 715, но меньше 1064 пикселей.
* "Мобильный"- для устройств, ширина экрана которых меньше 715 пикселей.

**Стартовая страница должна содержать следующие элементы:**

* "Шапку", содержащую ФИО студента, номер группы и номер варианта.
* Форму для ввода логина и пароля. Информация о зарегистрированных в системе пользователях должна храниться в отдельной таблице БД (пароль должен храниться в виде хэш-суммы). Доступ неавторизованных пользователей к основной странице приложения должен быть запрещён.

**Основная страница приложения должна содержать следующие элементы:**

* Набор полей ввода для задания координат точки и радиуса области в соответствии с вариантом задания: Button {'-4','-3','-2','-1','0','1','2','3','4'} для координаты по оси X, Text (-3 ... 3) для координаты по оси Y, и Button {'-4','-3','-2','-1','0','1','2','3','4'} для задания радиуса области. Если поле ввода допускает ввод заведомо некорректных данных (таких, например, как буквы в координатах точки или отрицательный радиус), то приложение должно осуществлять их валидацию.
* Динамически обновляемую картинку, изображающую область на координатной плоскости в соответствии с номером варианта и точки, координаты которых были заданы пользователем. Клик по картинке должен инициировать сценарий, осуществляющий определение координат новой точки и отправку их на сервер для проверки её попадания в область. Цвет точек должен зависить от факта попадания / непопадания в область. Смена радиуса также должна инициировать перерисовку картинки.
* Таблицу со списком результатов предыдущих проверок.
* Ссылку, по которой аутентифицированный пользователь может закрыть свою сессию и вернуться на стартовую страницу приложения.

**Дополнительные требования к приложению:**

* Все результаты проверки должны сохраняться в базе данных под управлением СУБД Oracle.
* Для доступа к БД необходимо использовать Spring Data.

**Вывод**

В ходе выполнения этой лабораторной работы мы научились разрабатывать веб-приложения на фреймворках Vue.js и Spring и устанавливать взаимодействие между уровнями back-end и front-end посредством REST API. Мы познакомились с разработкой single-page приложений. Vue намного компактнее и понятнее, по сравнению с React и Angular. Впечатления от работы с ним намного лучше, чем после работы с другими фреймворками. Spring (Spring Boot) – не особо понравится. ИМХО на такой лабораторной не раскрывается весь его потенциал. Абсолютно непонятно, что происходит “под капотом”, из-за этого возникает желания само убиться лишь больше не работать с ним. (Речь идёт в частности о Spring Security).