

Санкт-Петербургский национальный исследовательский университет информационных технологий
механики и оптики

Факультет ПИиКТ

Дисциплина: Веб-программирование

Лабораторная работа 4

Выполнили: Григорьева С.А.
Мхитарян С.А.
Группа: Р3200

Вариант 200050

Переписать приложение из предыдущей лабораторной работы с использованием следующих технологий:

- Уровень back-end должен быть основан на Spring.
- Уровень front-end должен быть построен на React + Redux (необходимо использовать ES6 и JSX) с использованием набора компонентов PrimeReact
- 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 1139 пикселей.
- "Планшетный" - для устройств, ширина экрана которых равна или превышает 785, но меньше 1139 пикселей.
- "Мобильный" - для устройств, ширина экрана которых меньше 785 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: MultiSelect {'-5','-4','-3','-2','-1','0','1','2','3'} для координаты по оси X, Text (-5 ... 3) для координаты по оси Y, и MultiSelect {'-5','-4','-3','-2','-1','0','1','2','3'} 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 зависеть 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:

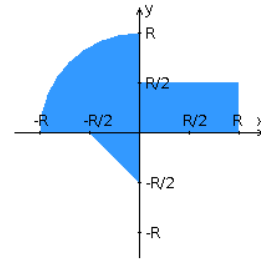
Все результаты проверки должны сохраняться в базе данных под управлением СУБД Oracle.

Для доступа к БД необходимо использовать Spring Data.

Исходники: <https://github.com/jenkeyx/pip4>, <https://github.com/jenkeyx/front4pip>

Вывод

В ходе выполнения данной лабораторной работы мы познакомились с фреймворками для фронтэнда и бекэнда. Научились использовать встроенные режимы браузера для мониторинга запросов. Довольно интересно программировать вдвоём, тк результат работы зависит не только от тебя. Твой сокомандник может внести так и классные функции так и ужасные баги.



изменение X: p:spinner {-5 ... 5}, шаг изменения - 1

изменение Y: inputText {-3 ... 5}

изменение R: selectOneMenu {'1','1.5','2','2.5','3'}