

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Факультет Программной инженерии и компьютерной техники

Лабораторная работа по
Веб-программированию №4
Вариант 20038

Работу выполнила:

Касьяненко В.М.

Группа:

P3220

Преподаватель:

Наумова Н.А.

Санкт-Петербург,

2023

Оглавление

Текст задания.....	3
Исходный код программы	4
Выводы по работе.....	4

Текст задания

Лабораторная работа #4

Вариант 20038

Внимание! У разных вариантов разный текст задания!

Переписать приложение из [предыдущей лабораторной работы](#) с использованием следующих технологий:

- Уровень back-end должен быть основан на Java EE (необходимо использовать EJB).
- 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 1101 пикселей.
- "Планшетный" - для устройств, ширина экрана которых равна или превышает 810, но меньше 1101 пикселей.
- "Мобильный" - для устройств, ширина экрана которых меньше 810 пикселей.

Стартовая страница должна содержать следующие элементы:

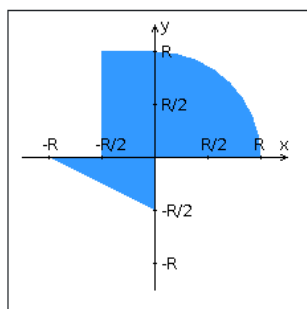
- "Шапку", содержащую ФИО студента, номер группы и номер варианта.
- Форму для ввода логина и пароля. Информация о зарегистрированных в системе пользователях должна храниться в отдельной таблице БД (пароль должен храниться в виде хэш-суммы). Доступ неавторизованных пользователей к основной странице приложения 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 $\{-3, -2, -1, 0, 1, 2, 3, 4, 5\}$ для координаты по оси X, Text $(-5 \dots 3)$ для координаты по оси Y, и MultiSelect $\{-3, -2, -1, 0, 1, 2, 3, 4, 5\}$ 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 / непадения 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 JPA.



Исходный код программы

https://github.com/CandyGoose/Web_programming_SE/tree/main/lab4

Выводы по работе

При выполнении данной лабораторной работы я разработала приложение, используя Java EE (EJB) для back-end и React + Redux (ES6, JSX, PrimeReact) для front-end, реализовав взаимодействие между уровнями через REST API, а также создала две страницы, адаптированные для "Десктопного", "Планшетного" и "Мобильного" режимов.