

# Университет ИТМО

## Лабораторная работа №4

*по дисциплине: Веб-программирование*

Вариант: 2622

Выполнили:

Кирилл Рогачёв, Р3230

Андрей Неграш, Р3230

Санкт-Петербург, 2020

### 1) Задание:

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 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 1245 пикселей.
- "Планшетный" - для устройств, ширина экрана которых равна или превышает 791, но меньше 1245 пикселей.
- "Мобильный" - для устройств, ширина экрана которых меньше 791 пикселей.

**Стартовая страница должна содержать следующие элементы:**

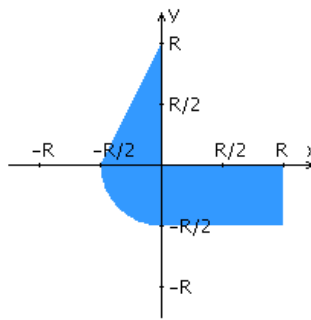
- "Шапку", содержащую ФИО студента, номер группы и номер варианта.
- Форму для ввода логина и пароля. Информация о зарегистрированных в системе пользователях должна храниться в отдельной таблице БД (пароль должен храниться в виде хэш-суммы). Доступ неавторизованных пользователей к основной странице приложения 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: Checkbox {'-3','-2','-1','0','1','2','3','4','5'} для координаты по оси X, Text (-5 ... 5) для координаты по оси Y, и Checkbox {'-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.
- Кнопку, по которой аутентифицированный пользователь может закрыть свою сессию и вернуться на стартовую страницу приложения.

**Дополнительные требования к приложению:**

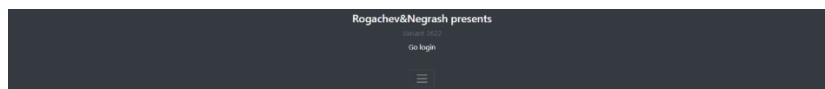
- Все результаты проверки должны сохраняться в базе данных под управлением СУБД Oracle.
- Для доступа к БД необходимо использовать JPA.



2) Ссылка на код:

<https://github.com/ANegrash/ITMO-all/tree/master/3%20Web/lab4>

3) Выполнение программы:

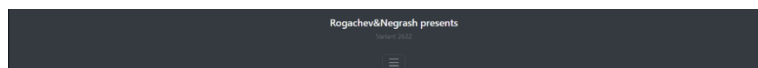


### Labwork 4: Registration

Email:  ✓

Password:  ✓

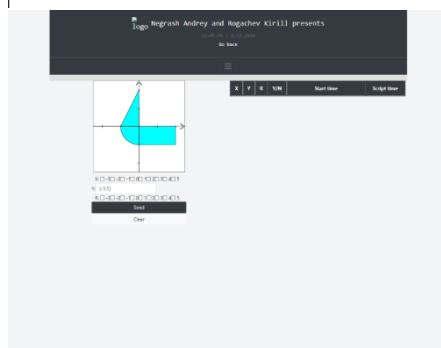
Repeat your pass:  ✓



### Labwork 4

Email:  ✓

Password:  ✓



Error 401

User didn't login

You doesn't have access to this page. Please login before go to this page

4) Вывод:

В процессе выполнения данной лабораторной работы мы познакомились с основными компонентами Vue.js, освоили работу с базой данных посредством JPA Hibernate под управлением СУБД Oracle, разработали четырёхстраничное веб-приложение согласно выданному варианту.