**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(РУТ (МИИТ))**

«Институт транспортной техники и систем управления»

Кафедра «Управление и защита информации»

**Отчет по лабораторной работе №6**

**по дисциплине**

«Web-программирование»

**Выполнил:** Ст. гр. ТКИ-541, Петросян А. А.

**Проверил:** доцент кафедры УиЗИ, к.т.н. Сафронов А. И.

**Москва**

**2023 г.**

**Цель работы:**

Разработать локальное одностраничное веб-приложение (LSPWA) под управлением фреймворка Vue.js на языке JavaScript в соответствии с указаниями вариантов индивидуального задания (покрывают два раздела).

**Постановка задачи:**

Вариант 14:

1. Изобразить в области масштабируемой векторной графики <*svg*> размерности 540 на 960 пикселей по ширине и высоте, соответственно, вертикальную координатную ось со стрелкой и засечками. Ось отстаёт от рамок <*svg*> на 50 пикселей, соответственно. Стрелка всегда фиксированных размеров, устанавливаемых автором (пользователь не управляет настройками вида этого элемента). Пользователь меняет шаг засечек посредством элементов <*input*> в режиме *range* в разумных пределах. Между засечками посередине всегда размещаются промежуточные засечки точно посередине. Пользователь также указывает начало и конец диапазона вывода числовых маркеров под засечками посредством элементов <*input*>; посредством <*input*> в режиме *checkbox* меняет направление стрелки: положительное направление либо снизу-вверх, либо сверху-вниз. В качестве элементов внутри <*svg*> использовать только <*line*> и <*text*>.
2. Организовать <*input*> со свойствами калькулятора. Вводить в <*input*> можно что угодно, но как только там, среди текста, появляется конструкции вида: «*Sin(x)*», «*Cos(x)*», «*Tg(x)*», «*Ctg(x)*», «*pi*» – выводить в этот же <*input*> на соответствующих позициях расчётные значения тригонометрических функций и константы.

**Выполнение 1:**

**Код сайта**

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Ось</title>

    <script src="https://cdn.jsdelivr.net/npm/vue@2.6.14/dist/vue.js"></script>

</head>

<body>

<div id="app">

    <div>

        <label>Шаг засечек: <input type="range" v-model="tickStep" min="10" max="100"></label><br>

        <label>Начало диапазона: <input type="number" v-model="startRange"></label><br>

        <label>Конец диапазона: <input type="number" v-model="endRange"></label><br>

        <label><input type="checkbox" v-model="invertDirection"> Инвертировать направление оси</label><br>

    </div>

    <svg width="540" height="960">

        <line

            :x1="50"

            :y1="invertDirection ? 0 : 960"

            :x2="50"

            :y2="invertDirection ? 960 : 0"

            stroke="black"

        />

        <line

            v-for="tick in ticks"

            :key="tick.position"

            :x1="50"

            :x2="70"

            :y1="tick.position"

            :y2="tick.position"

            stroke="black"

        />

        <text

            v-for="tick in ticks"

            :key="tick.value"

            x="80"

            :y="tick.position"

            font-family="Arial"

            font-size="12"

            alignment-baseline="middle"

        >

            {{ tick.value }}

        </text>

    </svg>

</div>

<script>

    new Vue({

        el: '#app',

        data: {

            tickStep: 50,

            startRange: 0,

            endRange: 10,

            invertDirection: false

        },

        computed: {

            ticks() {

                const ticks = [];

                const totalHeight = 960 - 2 \* 50;

                const numTicks = Math.floor(totalHeight / this.tickStep);

                const valueStep = (this.endRange - this.startRange) / numTicks;

                for (let i = 0; i <= numTicks; i++) {

                    const position = this.invertDirection

                        ? 50 + this.tickStep \* i

                        : 960 - (50 + this.tickStep \* i);

                    const value = this.startRange + valueStep \* i;

                    ticks.push({ value: value.toFixed(2), position });

                }

                return ticks;

            }

        }

    });

</script>

</body>

</html>

**Скриншоты работы сайта**

**Изображение выглядит как текст, снимок экрана, число

Автоматически созданное описание**

*Рис.1 - Главная страница*

Изображение выглядит как текст, снимок экрана, линия

Автоматически созданное описание

*Рис.2 – инвертировано направления и шаг засечек*

**Сеть петри**

Изображение выглядит как диаграмма, текст, круг, снимок экрана

Автоматически созданное описание

Рис.4 – Сеть петри

**Выполнение 2:**

**Код сайта**

<!DOCTYPE html>

<html>

<head>

  <title>Калькулятор</title>

</head>

<body>

  <div id="app">

    <input v-model="expression" @input="calculateExpression" placeholder="Введите выражение">

  </div>

  <script src="https://cdn.jsdelivr.net/npm/vue@2.6.14"></script>

  <script>

    new Vue({

      el: '#app',

      data: {

        expression: '',

      },

      methods: {

        calculateExpression() {

          let result = this.expression;

          // Заменяем Sin(x) на значение

          result = result.replace(/sin\(([^)]+)\)/g, (match, value) => {

            return Math.sin(parseFloat(value));

          });

          // Заменяем Cos(x) на значение

          result = result.replace(/cos\(([^)]+)\)/g, (match, value) => {

            return Math.cos(parseFloat(value));

          });

          // Заменяем Tg(x) на значение

          result = result.replace(/tg\(([^)]+)\)/g, (match, value) => {

            return Math.tan(parseFloat(value));

          });

          // Заменяем Ctg(x) на значение

          result = result.replace(/ctg\(([^)]+)\)/g, (match, value) => {

            return 1 / Math.tan(parseFloat(value));

          });

          // Заменяем pi на значение

          result = result.replace(/pi/g, Math.PI);

          this.expression = result;

        },

      },

    });

  </script>

</body>

</html>

**Скриншоты работы сайта**

**Изображение выглядит как текст, Шрифт, белый, снимок экрана

Автоматически созданное описание**

*Рис.1 - Главная страница*



*Рис.2 – Ввод значений*



*Рис.3 – Замена значения*

**Сеть петри**

Изображение выглядит как круг, диаграмма, снимок экрана, дизайн

Автоматически созданное описание

Рис.4 – Сеть петри

**Вывод:**

В ходе выполнения заданий были затронуты важные аспекты работы с фреймворком Vue.js. Была осуществлена работа с директивами.