

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет приложений»

Отчет по домашнему заданию
«Разработка прототипа SPA приложения»

Выполнил:
студент группы ИУ5-52Б
Барышников Михаил

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.

Москва, 2021 г.

Описание задания:

Необходимо доработать ваше приложение из лабораторной работы №7 или №8. Необходимо решить следующие задачи:

1. Модифицировать базу данных:

- База данных должна содержать основную таблицу (например, "акции") и вторую таблицу (например, "компания акции").
- Основная таблица ссылается через внешний ключ на вторую таблицу.

2. Доработать веб-сервис:

- Добавить в веб-сервис новый метод получения списка значений из второй таблицы.

3. Доработать фронтенд:

- Добавить на страницу с данными основной таблицы кнопки добавления, редактирования и удаления значений через API сервиса. Для добавления и редактирования использовать страницу отображения конкретного объекта.
- На всех страницах должны отображаться значения из таблиц вместо идентификаторов.
- На странице отображения конкретного объекта необходимо использовать компонент `combobox` (выпадающий список). Необходимо использовать в этом компоненте новый метод веб-сервиса для второй таблицы.

Расширенное задание, добавляющее +1 балл на экзамене:

1. Разработать страницы и методы веб-сервиса для добавления, удаления и редактирования данных второй таблицы.

Файл `server/index.js`:

```
'use strict';
```

```
const express = require('express');
const path = require('path');
const app = express();
const htmlPath = path.resolve(__dirname, '..', 'src/index.html')
const staticPath = path.resolve(__dirname, '..', 'src')
const port = 80;

app.use('/', express.static(staticPath));

app.all('*', (req, res) => {
  res.sendFile(htmlPath);
});

app.listen(port, () => {
  console.log(`Server listening http://localhost:${port}`);
});
```

Файл src/components/back-button/BackButtonComponent.js:

```
export class BackButtonComponent {
  constructor(parent) {
    this.parent = parent
  }

  getHTML() {
    return '<button id="back-button" class="btn btn-primary" type="button">Назад</button>'
  }

  addListeners(listener) {
    document.getElementById("back-button").addEventListener('click', listener)
  }

  render(listener) {
```

```

        const html = this.getHTML()
        this.parent.insertAdjacentHTML('beforeend', html)
        this.addListeners(listener)
    }
}

```

Файл src/components/button/ ButtonComponent.js:

```

export class ButtonComponent {
    constructor(parent, data) {
        this.parent = parent;
        this.data = data;
    }

    addListeners(listener, data) {
        document
            .getElementById(data+"-button")
            .addEventListener("click", listener)
    }

    getHTML(data) {
        return (
            `
                <div class="card text-white bg-dark"
style="width: 160px;">
                    <button id="${data}-button" class="btn
btn-primary" type="button">${data}</button>
                </div>
            `
        )
    }

    render(listener) {
        const html = this.getHTML(this.data)
        this.parent.insertAdjacentHTML('beforeend', html)
        this.addListeners(listener, this.data)
    }
}

```

Файл src/components/cpu-card/CpuCardComponents.js:

```
export class CpuCardComponent {
  constructor(parent) {
    this.parent = parent;
  }

  getHTML(data) {
    return (
      <div class="card text-white bg-dark"
style="width: 300px;">
        <div class="card-body">
          <h5 class="card-
title">${data.name}</h5>
          <!--<p class="card-
text">${data.frequency}</p>architecture-->
          <!--<p class="card-
text">${data.architecture}</p>-->
          <button class="btn btn-primary"
id="click-card-${data.id}" data-
id="${data.id}">Подробнее</button>
        </div>
      </div>
    )
  }

  addListeners(data, listener) {
    document
      .getElementById(`click-card-${data.id}`)
      .addEventListener("click", listener)
  }

  render(data, listener) {
    const html = this.getHTML(data)
    this.parent.insertAdjacentHTML('beforeend', html)
    this.addListeners(data, listener)
  }
}
```

Файл src/components/Cpu/CpuComponent.js:

```
export class CpuComponent {
  constructor(parent) {
    this.parent = parent
  }

  getHTML(data) {
    return (
      <div class='card text-white bg-dark' style='width:
300px'>
        <div class="card-header">
          Процессор
        </div>
        <div class='card-body'>
          <h4 class='card-title'>${data.name}</h4>
          <p class='card-text'>Рабочая частота:
${data.frequency}</p>
          <p class='card-text'>Архитектура:
${data.architecture}</p>
        </div>
      </div>
    )
  }

  getHTMLUpdate(data) {
    return (
      <div class='card text-white bg-dark' style='width:
300px'>
        <div class="card-header">
          Процессор
        </div>
        <div class='card-body'>
          <textarea id='name' class='card-
text'>${data.name}</textarea>
          <br>
          Рабочая частота:
          <br>

```

```

        <textarea id='frequency' class='card-
text'>${data.frequency}</textarea>
        <br>
        Архитектура:
        <br>
        <textarea id='architecture' class='card-
text'>${data.architecture}</textarea>
    </div>
</div>
`
    )
}

render(data, isUpdate) {
    if (isUpdate == 1)
    {
        const html = this.getHTMLUpdate(data)
        this.parent.insertAdjacentHTML('beforeend', html)
    }
    else
    {
        const html = this.getHTML(data)
        this.parent.insertAdjacentHTML('beforeend', html)
    }
}
}

```

Файл src/components/disk-card/DiskCardComponents.js:

```

export class DiskCardComponent {
    constructor(parent) {
        this.parent = parent;
    }

    getHTML(data) {
        return (
            <div class="card text-white bg-dark"
style="width: 300px;">
                <div class="card-body">
                    <h5 class="card-title">Тип накопителя:
${data.type}</h5>

```

```

        <p class='card-text'>Объем:
    ${data.volume}</p>
        <button class="btn btn-primary"
    id="click-card-${data.id}" data-
    id="${data.id}">Подробнее</button>
        </div>
    </div>
    )
}

addListeners(data, listener) {
    document
        .getElementById(`click-card-${data.id}`)
        .addEventListener("click", listener)
}

render(data, listener) {
    const html = this.getHTML(data)
    this.parent.insertAdjacentHTML('beforeend', html)
    this.addListeners(data, listener)
}
}

```

Файл src/components/disk/DiskComponent.js:

```

export class DiskComponent {
    constructor(parent) {
        this.parent = parent
    }

    getHTML(data) {
        return (
            <div class='card text-white bg-dark' style='width:
300px'>
                <div class="card-header">
                    Диск
                </div>
                <div class='card-body'>

```



```

        <h4 class='card-title'>Тип накопителя:
    </h4>
    <p class='card-text'>Скорость записи:
    </p>
    <p class='card-text'>Скорость чтения:
    </p>
    <p class='card-text'>Объем:
    </p>
    </div>
    </div>
    </div>
    )
}

getHTMLUpdate(data) {
    return (
        <div class='card text-white bg-dark' style='width:
300px'>
        <div class="card-header">
            Диск
        </div>
        <div class='card-body'>
            Тип накопителя:
            <br>
            <textarea id='type' class='card-
text'>${data.type}</textarea>
            <br>
            Скорость записи:
            <br>
            <textarea id='disk_ws' class='card-
text'>${data.ws}</textarea>
            <br>
            Скорость чтения:
            <br>
            <textarea id='disk_rs' class='card-
text'>${data.rs}</textarea>
            <br>
            Объем:
            <br>
            <textarea id='volume' class='card-
text'>${data.volume}</textarea>

```

```

        </div>
    </div>
    ,
)
}

render(data, isUpdate) {
    if (isUpdate == 1)
    {
        const html = this.getHTMLUpdate(data)
        this.parent.insertAdjacentHTML('beforeend', html)
    }
    else
    {
        const html = this.getHTML(data)
        this.parent.insertAdjacentHTML('beforeend', html)
    }
}
}
}

```

Файл src/components/hardware-card/HardwareCardComponents.js:

```

import { urls } from "../../modules/urls.js";
import { ajax } from "../../modules/ajax.js";

export class HardwareCardComponent {
    constructor(parent) {
        this.parent = parent;
    }

    getHTML(data, cpu) {
        return (
            <div class="card text-white bg-dark"
style="width: 300px;">
                <div class="card-body">
                    <h5 class="card-title">Компьютер на
процессоре ${cpu.name}</h5>
                    <button class="btn btn-primary"
id="click-card-${data.id}" data-
id="${data.id}">Подробнее</button>
                </div>

```

```

        </div>
    )
}

addListeners(data, listener) {
    document
        .getElementById(`click-card-${data.id}`)
        .addEventListener("click", listener)
}

async getCpu(data) {
    return ajax.get(urls.cpu(data.cpu))
}

async render(data, listener) {
    const cpu = await this.getCpu(data)
    const html = this.getHTML(data, cpu.data)
    this.parent.insertAdjacentHTML('beforeend', html)
    this.addListeners(data, listener)
}
}

```

Файл src/components/hardware/HardwareComponent.js:

```

import { urls } from "../../modules/urls.js";
import { ajax } from "../../modules/ajax.js";

export class HardwareComponent {
    constructor(parent) {
        this.parent = parent
    }

    getHTML(cpu, ram, disk) {
        return (
            <div class='card text-white bg-dark' style='width:
300px'>
                <div class="card-header">
                    Компьютер
                </div>

```

```

        <div class='card-body'>
            <h4 class='card-title'>Характеристики:
</h4>
            <p class='card-text'>Процессор:
${cpu.name}</p>
            <p class='card-text'>ОЗУ: ${ram.name}</p>
            <p class='card-text'>Накопитель:
${disk.type}</p>
        </div>
    </div>
    `
    )
}

getHTMLUpdate() {
    return (
        <div class='card text-white bg-dark' style='width:
300px'>
        <div class="card-header">
            Компьютер
        </div>
        <div class='card-body'>
            <h4 class='card-title'>Характеристики:
</h4>
            <br>
            Процессор:
            <br>
            <select id="cpu">
            </select>
            <br>
            ОЗУ:
            <br>
            <select id="ram">
            </select>
            <br>
            Накопитель:
            <br>
            <select id="disk">
            </select>
        </div>
    </div>

```

```

    )
}

async getCpus(data) {
    return ajax.get(urls.cpus())
}

async getRams(data) {
    return ajax.get(urls.rams())
}

async getDisks(data) {
    return ajax.get(urls.disks())
}

async getCpu(data) {
    return ajax.get(urls.cpu(data.cpu))
}

async getRam(data) {
    return ajax.get(urls.ram(data.ram))
}

async getDisk(data) {
    return ajax.get(urls.disk(data.diskstor))
}

async render(data, isUpdate = 0) {
    if (isUpdate == 1)
    {
        const html = this.getHTMLUpdate()
        this.parent.insertAdjacentHTML('beforeend', html)
        const cpus = await this.getCpus()
        const cpuSelect = document.getElementById('cpu')
        cpus.data.forEach((item) => {
            const cpuOption =
document.createElement('option')
            cpuOption.value = item.id
            cpuOption.innerHTML = item.name
            cpuSelect.appendChild(cpuOption)
        })
    }
}

```

```

        cpuSelect.value = data.cpu

        const rams = await this.getRams()
        const ramSelect = document.getElementById('ram')
        rams.data.forEach((item) => {
            const ramOption =
document.createElement('option')
            ramOption.value = item.id
            ramOption.innerHTML = item.name
            ramSelect.appendChild(ramOption)
        })
        ramSelect.value = data.ram

        const disks = await this.getDisks()
        const diskSelect = document.getElementById('disk')
        disks.data.forEach((item) => {
            const diskOption =
document.createElement('option')
            diskOption.value = item.id
            diskOption.innerHTML = item.type + ' ' +
item.volume
            diskSelect.appendChild(diskOption)
        })
        diskSelect.value = data.diskstor
    }
    else
    {
        const cpu = await this.getCpu(data)
        const ram = await this.getRam(data)
        const disk = await this.getDisk(data)
        const html = this.getHTML(cpu.data, ram.data,
disk.data)
        this.parent.insertAdjacentHTML('beforeend', html)
    }
}
}
}

```

Файл src/components/ram-card/RamCardComponents.js:

```

export class RamCardComponent {
    constructor(parent) {
        this.parent = parent;
    }
}

```

```

    }

    getHTML(data) {
        return (
            <div class="card text-white bg-dark"
style="width: 300px;">
                <div class="card-body">
                    <h5 class="card-
title">${data.name}</h5>
                    <button class="btn btn-primary"
id="click-card-${data.id}" data-
id="${data.id}">Подробнее</button>
                </div>
            </div>
        )
    }

    addListeners(data, listener) {
        document
            .getElementById(`click-card-${data.id}`)
            .addEventListener("click", listener)
    }

    render(data, listener) {
        const html = this.getHTML(data)
        this.parent.insertAdjacentHTML('beforeend', html)
        this.addListeners(data, listener)
    }
}

```

Файл src/components/ram/RamComponent.js:

```

export class RamComponent {
    constructor(parent) {
        this.parent = parent
    }

    getHTML(data) {
        return (

```

```

        <div class='card text-white bg-dark' style='width:
300px'>
            <div class="card-header">
                03У
            </div>
            <div class='card-body'>
                <h4 class='card-title'>${data.name}</h4>
                <p class='card-text'>Тип памяти:
${data.type}</p>
                <p class='card-text'>Частота работы:
${data.frequency}</p>
                <p class='card-text'>Объем:
${data.volume}</p>
            </div>
        </div>
    )
}

```

```

getHTMLUpdate(data) {
    return (
        <div class='card text-white bg-dark' style='width:
300px'>
            <div class="card-header">
                03У
            </div>
            <div class='card-body'>
                <textarea id='name' class='card-
title'>${data.name}</textarea>
                <br>
                Тип памяти:
                <br>
                <textarea id='type' class='card-
text'>${data.type}</textarea>
                <br>
                Частота работы:
                <br>
                <textarea id='frequency' class='card-
text'>${data.frequency}</textarea>
                <br>
            </div>
        </div>
    )
}

```



```

        Объем:
        <br>
        <textarea id='volume' class='card-
text'>${data.volume}</textarea>
    </div>
</div>
,
    )
}

render(data, isUpdate = 0) {
    if (isUpdate == 1)
    {
        const html = this.getHTMLUpdate(data)
        this.parent.insertAdjacentHTML('beforeend', html)
    }
    else
    {
        const html = this.getHTML(data)
        this.parent.insertAdjacentHTML('beforeend', html)
    }
}
}

```

Файл src/modules/ajax.js:

```

class Ajax {
    async get(url) {
        const response = await fetch(url);
        const responseData = await response.json();
        return {status: response.status, data: responseData};
    }
    async del(url) {
        const response = await fetch(url, {method: 'DELETE'});
        return {status: response.status};
    }
    async put(url, formData) {
        const response = await fetch(url, {method: 'PUT',
body: formData});
        return {status: response.status};
    }
    async post(url, formData) {

```

```

        const response = await fetch(url, {method: 'POST',
body: formData});
        return {status: response.status};
    }
}

```

```
export const ajax = new Ajax();
```

Файл src/modules/urls.js:

```

class Urls {
    constructor() {
        this.url = 'http://localhost:8000/api/';
    }

    cpus() {
        return `${this.url}cpu/`
    }

    cpu(id) {
        return `${this.url}cpu/${id}/`
    }

    disks() {
        return `${this.url}disk/`
    }

    disk(id) {
        return `${this.url}disk/${id}/`
    }

    rams() {
        return `${this.url}ram/`
    }

    ram(id) {
        return `${this.url}ram/${id}/`
    }

    hardwares() {
        return `${this.url}hardware/`
    }
}

```

```

        hardware(id) {
            return `${this.url}hardware/${id}/`
        }
    }
}

```

```
export const urls = new Urls();
```

Файл src/pages/main/MainPage.js:

```

import {DisksPage} from "../../pages/disk/DisksPage.js";
import { RamsPage } from "../../pages/ram/RamsPage.js";
import { CpusPage } from "../../pages/cpu/CpusPage.js";
import {HardwaresPage} from
"../../pages/hardware/HardwaresPage.js";
import { ButtonComponent } from
"../../components/button/ButtonComponent.js";

```

```

export class MainPage {
    constructor(parent) {
        this.parent = parent;
    }

```

```

    get page() {
        return document.getElementById('main-page')
    }

```

```

    getHTML() {
        return (
            `
                <div id="main-page" class="d-flex flex-
wrap"><div/>
            `
        )
    }

```

```

    addListeners(data, listener) {
        document
            .getElementById(`click-card-${data.id}`)
            .addEventListener("click", listener)
    }

```

```

    clickHardwaresPage(e) {
        const hardwaresPage = new HardwaresPage(this.parent,
document.getElementById('root'))
        hardwaresPage.render()
    }

    clickCpusPage(e) {
        const cpusPage = new CpusPage(this.parent,
document.getElementById('root'))
        cpusPage.render()
    }

    clickRamsPage(e) {
        const ramsPage = new RamsPage(this.parent,
document.getElementById('root'))
        ramsPage.render()
    }

    clickDisksPage(e) {
        const disksPage = new DisksPage(this.parent,
document.getElementById('root'))
        disksPage.render()
    }

    render() {
        this.parent.innerHTML = ''
        const html = this.getHTML()
        this.parent.insertAdjacentHTML('beforeend', html)
        const buttonHard = new ButtonComponent(this.page,
'Компьютеры')
        buttonHard.render(this.clickHardwaresPage.bind(this))
        const buttonCpu = new ButtonComponent(this.page,
'Процессоры')
        buttonCpu.render(this.clickCpusPage.bind(this))
        const buttonRam = new ButtonComponent(this.page,
'ОЗУ')
        buttonRam.render(this.clickRamsPage.bind(this))
        const buttonDisk = new ButtonComponent(this.page,
'Накопители')
        buttonDisk.render(this.clickDisksPage.bind(this))
    }

```

```
}
```

Файл src/pages/cpu/CpuPage.js:

```
import {CpuComponent} from
"../../components/Cpu/CpuComponent.js";
import {BackButtonComponent} from "../../components/back-
button/BackButtonComponent.js";
import { ButtonComponent } from
"../../components/button/ButtonComponent.js";
import {CpusPage} from "../cpu/CpusPage.js";
import { urls } from "../../modules/urls.js";
import { ajax } from "../../modules/ajax.js";
```

```
export class CpuPage {
    constructor(parent, id = 0) {
        this.parent = parent
        this.id = id
    }

    async getData() {
        return ajax.get(urls.cpu(this.id))
    }

    get page() {
        return document.getElementById('stock-page')
    }

    getHTML() {
        return (
            `
                <div id="stock-page">
                </div>
            `
        )
    }
}
```

```

    }

    clickBack() {
        const mainPage = new CpusPage(this.parent)
        mainPage.render()
    }

    async clickDel() {
        const hardwares = await ajax.get(urls.hardwares())
        for (const item of hardwares.data) {
            if (item.cpu == this.id)
            {
                await ajax.del(urls.hardware(item.id));
            }
        }
        await ajax.del(urls.cpu(this.id))
        const mainPage = new CpusPage(this.parent)
        mainPage.render()
    }

    async clickUpd() {
        const formData = new FormData();
        formData.append('name',
document.getElementById(`name`).value);
        formData.append('frequency',
document.getElementById(`frequency`).value);
        formData.append('architecture',
document.getElementById(`architecture`).value);
        await ajax.put(urls.cpu(this.id), formData)
        const mainPage = new CpusPage(this.parent)
        mainPage.render()
    }

    async clickUpdate() {
        this.parent.innerHTML = ''
        const html = this.getHTML()
        this.parent.insertAdjacentHTML('beforeend', html)
    }

```

```

        const backButton = new BackButtonComponent(this.page)
        backButton.render(this.clickBack.bind(this))
        const updateButton = new ButtonComponent(this.page,
'Sохранить')
        updateButton.render(this.clickUpd.bind(this))
        const data = await this.getData()
        const cpu = new CpuComponent(this.page)
        cpu.render(data.data, 1)
    }

```

```

    async clickAdd() {
        const formData = new FormData();
        formData.append('name',
document.getElementById(`name`).value);
        formData.append('frequency',
document.getElementById(`frequency`).value);
        formData.append('architecture',
document.getElementById(`architecture`).value);
        await ajax.post(urls.cpus(), formData)
        const mainPage = new CpusPage(this.parent)
        mainPage.render()
    }

```

```

    async render(isCreate = 0) {
        this.parent.innerHTML = ''
        const html = this.getHTML()
        this.parent.insertAdjacentHTML('beforeend', html)
        if (isCreate == 1)
        {
            const backButton = new
BackButtonComponent(this.page)
            backButton.render(this.clickBack.bind(this))
            const updateButton = new
ButtonComponent(this.page, 'Сохранить')
            updateButton.render(this.clickAdd.bind(this))
            const cpu = new CpuComponent(this.page)

```

```

        cpu.render({name: '', frequency: '', architecture:
'', 1)
    }
    else
    {
        const backButton = new
BackButtonComponent(this.page)
        backButton.render(this.clickBack.bind(this))
        const delButton = new ButtonComponent(this.page,
'Удалить')
        delButton.render(this.clickDel.bind(this))
        const updateButton = new
ButtonComponent(this.page, 'Обновить')
        updateButton.render(this.clickUpdate.bind(this))

        const data = await this.getData()
        const cpu = new CpuComponent(this.page)
        cpu.render(data.data, 0)
    }
}
}

```

Файл src/pages/cpu/CpusPage.js:

```

import {CpuCardComponent} from "../../components/cpu-
card/CpuCardComponents.js";
import { BackButtonComponent } from "../../components/back-
button/BackButtonComponent.js";
import { ButtonComponent } from
"./../components/button/ButtonComponent.js";
import { MainPage } from "../../main/MainPage.js";
import {CpuPage} from "../../cpu/CpuPage.js";
import { urls } from "../../modules/urls.js";
import { ajax } from "../../modules/ajax.js";

export class CpusPage {
    constructor(parent) {

```



```

        this.parent = parent;
    }

    async getData() {
        return ajax.get(urls.cpus())
    }

    get page() {
        return document.getElementById('main-page')
    }

    getHTML() {
        return (
            `
                <div id="main-page" class="d-flex flex-
wrap"><div/>
            `
        )
    }

    clickCard(e) {
        const cardId = e.target.dataset.id

        const cpuPage = new CpuPage(this.parent, cardId)
        cpuPage.render()
    }

    clickBack() {
        const mainPage = new MainPage(this.parent)
        mainPage.render()
    }

    async clickAdd() {
        const cpuPage = new CpuPage(this.parent)
        cpuPage.render(1)
    }

```

```

async render() {
  this.parent.innerHTML = ''
  const html = this.getHTML()
  this.parent.insertAdjacentHTML('beforeend', html)

  const data = await this.getData()
  data.data.forEach((item) => {
    const CpuCard = new CpuCardComponent(this.page)
    CpuCard.render(item, this.clickCard.bind(this))
  })
  const backButton = new BackButtonComponent(this.page)
  backButton.render(this.clickBack.bind(this))
  const updateButton = new ButtonComponent(this.page,
'Dобавить')
  updateButton.render(this.clickAdd.bind(this))
}
}

```

Файл src/pages/disk/DiskPage.js:

```

import {DiskComponent} from
"../../components/Disk/DiskComponent.js";
import {BackButtonComponent} from "../../components/back-
button/BackButtonComponent.js";
import { ButtonComponent } from
"../../components/button/ButtonComponent.js";
import {DisksPage} from "../../disk/DisksPage.js";
import { urls } from "../../modules/urls.js";
import { ajax } from "../../modules/ajax.js";

export class DiskPage {
  constructor(parent, id) {
    this.parent = parent
    this.id = id
  }
}

```

```

async getData() {
    return ajax.get(urls.disk(this.id))
}

get page() {
    return document.getElementById('stock-page')
}

getHTML() {
    return (
        `
        <div id="stock-page">
        </div>
        `
    )
}

clickBack() {
    const mainPage = new DisksPage(this.parent)
    mainPage.render()
}

async clickDel() {
    const hardwares = await ajax.get(urls.hardwares())
    for (const item of hardwares.data) {
        if (item.diskstor == this.id)
        {
            await ajax.del(urls.hardware(item.id));
        }
    }
    await ajax.del(urls.disk(this.id))
    const mainPage = new DisksPage(this.parent)
    mainPage.render()
}

async clickUpd() {

```

```

        const formData = new FormData();
        formData.append('type',
document.getElementById(`type`).value);
        formData.append('ws',
document.getElementById(`disk_ws`).value);
        formData.append('rs',
document.getElementById(`disk_rs`).value);
        formData.append('volume',
document.getElementById(`volume`).value);
        await ajax.put(urls.disk(this.id), formData)
        const mainPage = new DisksPage(this.parent)
        mainPage.render()
    }

```

```

    async clickUpdate() {
        this.parent.innerHTML = ''
        const html = this.getHTML()
        this.parent.insertAdjacentHTML('beforeend', html)
        const backButton = new BackButtonComponent(this.page)
        backButton.render(this.clickBack.bind(this))
        const updateButton = new ButtonComponent(this.page,
'Sохранить')
        updateButton.render(this.clickUpd.bind(this))
        const data = await this.getData()
        const disk = new DiskComponent(this.page)
        disk.render(data.data, 1)
    }

```

```

    async clickAdd() {
        const formData = new FormData();
        formData.append('type',
document.getElementById(`type`).value);
        formData.append('ws',
document.getElementById(`disk_ws`).value);
        formData.append('rs',
document.getElementById(`disk_rs`).value);

```

```

        formData.append('volume',
document.getElementById(`volume`).value);
        await ajax.post(urls.disks(), formData)
        const mainPage = new DisksPage(this.parent)
        mainPage.render()
    }

    async render(isCreate = 0) {
        this.parent.innerHTML = ''
        const html = this.getHTML()
        this.parent.insertAdjacentHTML('beforeend', html)
        if (isCreate == 1)
        {
            const backButton = new
BackButtonComponent(this.page)
            backButton.render(this.clickBack.bind(this))
            const updateButton = new
ButtonComponent(this.page, 'Сохранить')
            updateButton.render(this.clickAdd.bind(this))
            const disk = new DiskComponent(this.page)
            disk.render({type: '', ws: '', rs: '', volume:
'', 1)
        }
        else
        {
            const backButton = new
BackButtonComponent(this.page)
            backButton.render(this.clickBack.bind(this))
            const delButton = new ButtonComponent(this.page,
'Удалить')
            delButton.render(this.clickDel.bind(this))
            const updateButton = new
ButtonComponent(this.page, 'Обновить')
            updateButton.render(this.clickUpdate.bind(this))
            const data = await this.getData()
            const disk = new DiskComponent(this.page)
            disk.render(data.data, 0)
        }
    }
}

```

```

    }
  }
}

```

Файл src/pages/disk/DisksPage.js:

```

import {DiskCardComponent} from "../../components/disk-
card/DiskCardComponents.js";
import { BackButtonComponent } from "../../components/back-
button/BackButtonComponent.js";
import { ButtonComponent } from
"./../components/button/ButtonComponent.js";
import { MainPage } from "../main/MainPage.js";
import {DiskPage} from "../disk/DiskPage.js";
import { urls } from "../../modules/urls.js";
import { ajax } from "../../modules/ajax.js";

export class DisksPage {
  constructor(parent) {
    this.parent = parent;
  }

  async getData() {
    return ajax.get(urls.disks())
  }

  get page() {
    return document.getElementById('main-page')
  }

  getHTML() {
    return (
      <div id="main-page" class="d-flex flex-
wrap"><div/>
    )
  }
}

```

```

    }

    clickCard(e) {
        const cardId = e.target.dataset.id

        const diskPage = new DiskPage(this.parent, cardId)
        diskPage.render()
    }

    clickBack() {
        const mainPage = new MainPage(this.parent)
        mainPage.render()
    }

    async clickAdd() {
        const diskPage = new DiskPage(this.parent)
        diskPage.render(1)
    }

    async render() {
        this.parent.innerHTML = ''
        const html = this.getHTML()
        this.parent.insertAdjacentHTML('beforeend', html)

        const data = await this.getData()
        data.data.forEach((item) => {
            const DiskCard = new DiskCardComponent(this.page)
            DiskCard.render(item, this.clickCard.bind(this))
        })
        const backButton = new BackButtonComponent(this.page)
        backButton.render(this.clickBack.bind(this))
        const updateButton = new ButtonComponent(this.page,
'Dобавить')
        updateButton.render(this.clickAdd.bind(this))
    }
}

```

Файл src/pages/hardware/HardwarePage.js:

```
import {HardwareComponent} from
"../../components/hardware/HardwareComponent.js";
import {BackButtonComponent} from "../../components/back-
button/BackButtonComponent.js";
import { ButtonComponent } from
"../../components/button/ButtonComponent.js";
import {HardwarePage} from "../../hardware/HardwarePage.js";
import { urls } from "../../modules/urls.js";
import { ajax } from "../../modules/ajax.js";

export class HardwarePage {
  constructor(parent, id) {
    this.parent = parent
    this.id = id
  }

  async getData() {
    return ajax.get(urls.hardware(this.id))
  }

  get page() {
    return document.getElementById('stock-page')
  }

  getHTML() {
    return (
      `
      <div id="stock-page">
      </div>
    `
  )
}
```



```

clickBack() {
    const mainPage = new HardwaresPage(this.parent)
    mainPage.render()
}

async clickDel() {
    await ajax.del(urls.hardware(this.id))
    const mainPage = new HardwaresPage(this.parent)
    mainPage.render()
}

async clickUpd() {
    const formData = new FormData();
    formData.append('cpu',
document.getElementById('cpu').value);
    formData.append('ram',
document.getElementById(`ram`).value);
    formData.append('diskstor',
document.getElementById(`disk`).value);
    await ajax.put(urls.hardware(this.id), formData)
    const mainPage = new HardwaresPage(this.parent)
    mainPage.render()
}

async clickUpdate() {
    this.parent.innerHTML = ''
    const html = this.getHTML()
    this.parent.insertAdjacentHTML('beforeend', html)
    const backButton = new BackButtonComponent(this.page)
    backButton.render(this.clickBack.bind(this))
    const updateButton = new ButtonComponent(this.page,
'Сохранить')
    updateButton.render(this.clickUpd.bind(this))
    const data = await this.getData()
    const hardware = new HardwareComponent(this.page)
    hardware.render(data.data, 1)
}

```

```

    async clickAdd() {
        const formData = new FormData();
        formData.append('cpu',
document.getElementById('cpu').value);
        formData.append('ram',
document.getElementById(`ram`).value);
        formData.append('diskstor',
document.getElementById(`disk`).value);
        await ajax.post(urls.hardwares(), formData)
        const mainPage = new HardwaresPage(this.parent)
        mainPage.render()
    }

    async render(isCreate = 0) {
        this.parent.innerHTML = ''
        const html = this.getHTML()
        this.parent.insertAdjacentHTML('beforeend', html)
        if (isCreate == 1)
        {
            this.parent.innerHTML = ''
            const html = this.getHTML()
            this.parent.insertAdjacentHTML('beforeend', html)
            const backButton = new
BackButtonComponent(this.page)
            backButton.render(this.clickBack.bind(this))
            const updateButton = new
ButtonComponent(this.page, 'Сохранить')
            updateButton.render(this.clickAdd.bind(this))
            const hardware = new HardwareComponent(this.page)
            hardware.render({cpu: '', ram: '', diskstore: ''},
1)
        }
        else
        {
            const backButton = new
BackButtonComponent(this.page)

```

```

        backButton.render(this.clickBack.bind(this))
        const delButton = new ButtonComponent(this.page,
'Удалить')
        delButton.render(this.clickDel.bind(this))
        const updateButton = new
ButtonComponent(this.page, 'Обновить')
        updateButton.render(this.clickUpdate.bind(this))

        const data = await this.getData()
        const hardware = new HardwareComponent(this.page)
        hardware.render(data.data)
    }
}
}

```

Файл src/pages/hardware/HardwaresPage.js:

```

import { HardwareCardComponent } from
"../../components/hardware-card/HardwareCardComponents.js";
import { BackButtonComponent } from "../../components/back-
button/BackButtonComponent.js";
import { ButtonComponent } from
"../../components/button/ButtonComponent.js";
import { MainPage } from "../main/MainPage.js";
import { HardwarePage } from "../hardware/HardwarePage.js";
import { urls } from "../../modules/urls.js";
import { ajax } from "../../modules/ajax.js";

export class HardwaresPage {
    constructor(parent) {
        this.parent = parent;
    }

    async getData() {
        return ajax.get(urls.hardwares())
    }
}

```

```

get page() {
    return document.getElementById('main-page')
}

getHTML() {
    return (
        `
            <div id="main-page" class="d-flex flex-
wrap"><div/>
        `
    )
}

clickCard(e) {
    const cardId = e.target.dataset.id

    const hardwarePage = new HardwarePage(this.parent,
cardId)
    hardwarePage.render()
}

clickBack() {
    const mainPage = new MainPage(this.parent)
    mainPage.render()
}

async clickAdd() {
    const hardwarePage = new HardwarePage(this.parent)
    hardwarePage.render(1)
}

async render() {
    this.parent.innerHTML = ''
    const html = this.getHTML()
    this.parent.insertAdjacentHTML('beforeend', html)
}

```

```

        const data = await this.getData()
        data.data.forEach((item) => {
            const HardwareCard = new
HardwareCardComponent(this.page)
            HardwareCard.render(item,
this.clickCard.bind(this))
        })
        const backButton = new BackButtonComponent(this.page)
        backButton.render(this.clickBack.bind(this))
        const updateButton = new ButtonComponent(this.page,
'Добавить')
        updateButton.render(this.clickAdd.bind(this))
    }
}

```

Файл src/pages/ram/RamPage.js:

```

import {RamComponent} from
"../../components/ram/RamComponent.js";
import {BackButtonComponent} from "../../components/back-
button/BackButtonComponent.js";
import { ButtonComponent } from
"../../components/button/ButtonComponent.js";
import {RamsPage} from "../../ram/RamsPage.js";
import { urls } from "../../modules/urls.js";
import { ajax } from "../../modules/ajax.js";

export class RamPage {
    constructor(parent, id) {
        this.parent = parent
        this.id = id
    }

    async getData() {
        return ajax.get(urls.ram(this.id))
    }
}

```

```

get page() {
    return document.getElementById('stock-page')
}

getHTML() {
    return (
        `
            <div id="stock-page">
            </div>
        `
    )
}

clickBack() {
    const mainPage = new RamsPage(this.parent)
    mainPage.render()
}

async clickDel() {
    const hardwares = await ajax.get(urls.hardwares())
    for (const item of hardwares.data) {
        if (item.ram == this.id)
        {
            await ajax.del(urls.hardware(item.id));
        }
    }
    await ajax.del(urls.ram(this.id))
    const mainPage = new RamsPage(this.parent)
    mainPage.render()
}

async clickUpd() {
    const formData = new FormData();
    formData.append('name',
document.getElementById(`name`).value);

```

```

        formData.append('type',
document.getElementById(`type`).value);
        formData.append('frequency',
document.getElementById(`frequency`).value);
        formData.append('volume',
document.getElementById(`volume`).value);
        await ajax.put(urls.ram(this.id), formData)
        const mainPage = new RamsPage(this.parent)
        mainPage.render()
    }

```

```

    async clickUpdate() {
        this.parent.innerHTML = ''
        const html = this.getHTML()
        this.parent.insertAdjacentHTML('beforeend', html)
        const backButton = new BackButtonComponent(this.page)
        backButton.render(this.clickBack.bind(this))
        const updateButton = new ButtonComponent(this.page,
'Sохранить')
        updateButton.render(this.clickUpd.bind(this))
        const data = await this.getData()
        const ram = new RamComponent(this.page)
        ram.render(data.data, 1)
    }

```

```

    async clickAdd() {
        const formData = new FormData();
        formData.append('name',
document.getElementById(`name`).value);
        formData.append('type',
document.getElementById(`type`).value);
        formData.append('frequency',
document.getElementById(`frequency`).value);
        formData.append('volume',
document.getElementById(`volume`).value);
        await ajax.post(urls.rams(), formData)
        const mainPage = new RamsPage(this.parent)
    }

```

```

        mainPage.render()
    }

    async render(isCreate = 0) {
        this.parent.innerHTML = ''
        const html = this.getHTML()
        this.parent.insertAdjacentHTML('beforeend', html)
        if (isCreate == 1)
        {
            const backButton = new
BackButtonComponent(this.page)
            backButton.render(this.clickBack.bind(this))
            const updateButton = new
ButtonComponent(this.page, 'Сохранить')
            updateButton.render(this.clickAdd.bind(this))
            const data = await this.getData()
            const ram = new RamComponent(this.page)
            ram.render({name: '', type: '', frequency: '',
volume: ''}, 1)
        }
        else
        {
            const backButton = new
BackButtonComponent(this.page)
            backButton.render(this.clickBack.bind(this))
            const delButton = new ButtonComponent(this.page,
'Удалить')
            delButton.render(this.clickDel.bind(this))
            const updateButton = new
ButtonComponent(this.page, 'Обновить')
            updateButton.render(this.clickUpdate.bind(this))

            const data = await this.getData()
            const ram = new RamComponent(this.page)
            ram.render(data.data)
        }
    }

```



```
    }  
}
```

Файл src/pages/ram/RamsPage.js:

```
import {RamCardComponent} from "../../components/ram-card/RamCardComponents.js";  
import { BackButtonComponent } from "../../components/back-button/BackButtonComponent.js";  
import { ButtonComponent } from  
"../../components/button/ButtonComponent.js";  
import { MainPage } from "../../main/MainPage.js";  
import {RamPage} from "../../ram/RamPage.js";  
import { urls } from "../../modules/urls.js";  
import { ajax } from "../../modules/ajax.js";  
  
export class RamsPage {  
    constructor(parent) {  
        this.parent = parent;  
    }  
  
    async getData() {  
        return ajax.get(urls.rams())  
    }  
  
    get page() {  
        return document.getElementById('main-page')  
    }  
  
    getHTML() {  
        return (  
            `<div id="main-page" class="d-flex flex-wrap"><div/>  
            `)  
    }  
}
```

```

clickCard(e) {
    const cardId = e.target.dataset.id

    const ramPage = new RamPage(this.parent, cardId)
    ramPage.render()
}

clickBack() {
    const mainPage = new MainPage(this.parent)
    mainPage.render()
}

async clickAdd() {
    const ramPage = new RamPage(this.parent)
    ramPage.render(1)
}

async render() {
    this.parent.innerHTML = ''
    const html = this.getHTML()
    this.parent.insertAdjacentHTML('beforeend', html)

    const data = await this.getData()
    data.data.forEach((item) => {
        const RamCard = new RamCardComponent(this.page)
        RamCard.render(item, this.clickCard.bind(this))
    })
    const backButton = new BackButtonComponent(this.page)
    backButton.render(this.clickBack.bind(this))
    const updateButton = new ButtonComponent(this.page,
'Dобавить')
    updateButton.render(this.clickAdd.bind(this))
}
}

```

Файл src/scripts/snowfall.js:

```
Date.now|| (Date.now=function(){return(new
Date).getTime()}),function(){"use strict";for(var
t=["webkit","moz"],e=0;e<t.length&&!window.requestAnimationFra
me;++e){var
i=t[e];window.requestAnimationFrame=window[i+"RequestAnimation
Frame"],window.cancelAnimationFrame=window[i+"CancelAnimationF
rame"]||window[i+"CancelRequestAnimationFrame"]}if(/iP(ad|hone
|od).*OS
6/.test(window.navigator.userAgent)||!window.requestAnimationF
rame||!window.cancelAnimationFrame){var
s=0;window.requestAnimationFrame=function(t){var
e=Date.now(),i=Math.max(s+16,e);return
setTimeout(function(){t(s=i)},i-
e)},window.cancelAnimationFrame=clearTimeout}}(),function(t){t
.snowfall=function(e,i){function
s(s,n,a,o){this.x=s,this.y=n,this.size=a,this.speed=o,this.ste
p=0,this.stepSize=h(1,10)/100,i.collection&&(this.target=m[h(0
,m.length-1)]);var
r=null;i.image?(r=document.createElement("img"),r.src=i.image)
:(r=document.createElement("div"),t(r).css({background:i.flake
Color})),t(r).attr({"class":"snowfall-
flakes"}).css({width:this.size,height:this.size,position:i fla
kePosition,top:this.y,left:this.x,fontSize:0,zIndex:i.flakeInd
ex}),t(e).get(0).tagName===t(document).get(0).tagName?(t("body
").append(t(r)),e=t("body")):t(e).append(t(r)),this.element=r,
this.update=function(){if(this.y+=this.speed,this.y>l-
(this.size+6)&&this.reset(),this.element.style.top=this.y+"px"
,this.element.style.left=this.x+"px",this.step+=this.stepSize,
this.x+=y===!1?Math.cos(this.step):y+Math.cos(this.step),i.col
lection&&this.x>this.target.x&&this.x<this.target.width+this.t
arget.x&&this.y>this.target.y&&this.y<this.target.height+this.
target.y){var t=this.target.element.getContext("2d"),e=this.x-
this.target.x,s=this.y-
this.target.y,n=this.target.colData;if(void
0!==n[parseInt(e)][parseInt(s+this.speed+this.size)]||s+this.s
```

```

peed+this.size>this.target.height)if(s+this.speed+this.size>th
is.target.height){for(;s+this.speed+this.size>this.target.heig
ht&&this.speed>0;)this.speed*=.5;t.fillStyle="#fff",void
0==n[parseInt(e)][parseInt(s+this.speed+this.size)]?(n[parseIn
t(e)][parseInt(s+this.speed+this.size)]=1,t.fillRect(e,s+this.
speed+this.size,this.size,this.size):(n[parseInt(e)][parseInt
(s+this.speed)]=1,t.fillRect(e,s+this.speed,this.size,this.siz
e)),this.reset())}else
this.speed=1,this.stepSize=0,parseInt(e)+1<this.target.width&&
void 0==n[parseInt(e)+1][parseInt(s)+1]?this.x++:parseInt(e)-
1>0&&void 0==n[parseInt(e)-1][parseInt(s)+1]?this.x--
:(t.fillStyle="#fff",t.fillRect(e,s,this.size,this.size),n[par
seInt(e)][parseInt(s)]=1,this.reset())}(this.x+this.size>d-
c||this.x<c)&&this.reset()),this.reset=function(){this.y=0,thi
s.x=h(c,d-
c),this.stepSize=h(1,10)/100,this.size=h(100*i.minSize,100*i.m
axSize)/100,this.element.style.width=this.size+"px",this.eleme
nt.style.height=this.size+"px",this.speed=h(i.minSpeed,i.maxSpe
ed)}}function
n(){for(r=0;r<a.length;r+=1)a[r].update();f=requestAnimationFr
ame(function(){n()})}var
a=[],o={flakeCount:35,flakeColor:"#ffffff",flakePosition:"abso
lute",flakeIndex:999999,minSize:1,maxSize:2,minSpeed:1,maxSpee
d:5,round:!1,shadow:!1,collection:!1,collectionHeight:40,devic
eorientation:!1},i=t.extend(o,i),h=function(t,e){return
Math.round(t+Math.random()*(e-
t))};t(e).data("snowfall",this);var
r=0,l=t(e).height(),d=t(e).width(),c=0,f=0;if(i.collection!==!
1){var
p=document.createElement("canvas");if(p.getContext&&p.getConte
xt("2d"))for(var
m=[],w=t(i.collection),g=i.collectionHeight,r=0;r<w.length;r++
){var
u=w[r].getBoundingClientRect(),x=t("<canvas/>",{ "class": "snowf
all-canvas"}),z=[];if(u.top-
g>0){t("body").append(x),x.css({position:i.flakePosition,left:
u.left+"px",top:u.top-

```

```

g+"px"}).prop({width:u.width,height:g});for(var
v=0;v<u.width;v++)z[v]=[];m.push({element:x.get(0),x:u.left,y:
u.top-g,width:u.width,height:g,colData:z}})}else
i.collection=!1}for(t(e).get(0).tagName===t(document).get(0).t
agName&&(c=25),t(window).bind("resize",function(){l=t(e)[0].cl
ientHeight,d=t(e)[0].offsetWidth}),r=0;r<i.flakeCount;r+=1)a.p
ush(new s(h(c,d-
c),h(0,l),h(100*i.minSize,100*i.maxSize)/100,h(i.minSpeed,i.ma
xSpeed)));i.round&&t(".snowfall-flakes").css({"-moz-border-
radius":i.maxSize,"-webkit-border-radius":i.maxSize,"border-
radius":i.maxSize}),i.shadow&&t(".snowfall-flakes").css({"-
moz-box-shadow":"1px 1px 1px #555","-webkit-box-shadow":"1px
1px 1px #555","box-shadow":"1px 1px 1px #555"});var
y=!1;i.deviceorientation&&t(window).bind("deviceorientation",f
unction(t){y=.1*t.originalEvent.gamma}),n(),this.clear=functio
n(){t(".snowfall-canvas").remove(),t(e).children(".snowfall-
flakes").remove(),cancelAnimationFrame(f)}},t.fn.snowfall=func
tion(e){return"object"===typeof e||void
0==e?this.each(function(){new
t.snowfall(this,e)}):"string"===typeof
e?this.each(function(){var
e=t(this).data("snowfall");e&&e.clear()}):void 0}}(jQuery);

```

Файл src/index.html:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Lab7</title>
  <link rel="stylesheet"
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.1.1/dist/css/bo
otstrap.min.css">
</head>
<body>
<div id="root"></div>
<script src="main.js" type="module"></script>

```

```

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
<script src="scripts/snowfall.js"></script>
<script>
    $(document).ready(function(){
        $('.collectonme').hide();
        //Start the snow default options you can also make it
snow in certain elements, etc.
        $(document).snowfall({image : "scripts/img/snow3.png",
minSize: 10, maxSize:32});
    });
</script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/gsap/1.20.3/TweenMax.min.js"></script>
<div class="container" style="height: 400px; width: 400px;">

    <svg version="1.1" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" x="500px" y="500px"
viewBox="0 0 590 390" style="position: absolute; bottom:
0; height: 400px; width: 400px;">
    <g id="tree">
        <ellipse id="tree-base" fill="#39445B" cx="259.5" cy="319"
rx="120" ry="38"/>
        <path id="tree-par-1" fill="#42CE8E" d="M260,185c0,0,2.5-
0.8,6.7,2.6c4.3,3.4,121.1,81.2,121.1,81.2s4,3.9,1.6,9.6
C387,284,383,286,383,286s-49,41-123,41c-76,0-123-42.2-
123-42.2s-6.1-2.8-6.5-6.8s-0.4-7.3,2.6-10.1c3-2.9,120.1-
80.1,120.1-80.1
S255,185,260,185z"/>
        <path id="tree-part-1-shadow" fill="#09B56B"
d="M357.3,248.4c-16-10.7-35.8-23.9-52.8-35.3c-11.4,2.1-
26.3,3.9-44.5,3.9

```

```

c-17.3,0-32.5-2-44.5-4.4c-17.1,11.3-37.1,24.6-
53.1,35.3c13.3,7.4,50.1,25.1,97.7,25.1C309.2,273,344.3,255.9,3
57.3,248.4z"/>
<path id="tree-part-2" fill="#42CE8E" d="M260,123c0,0,2.5-
0.8,6.7,2.6c4.3,3.4,101.1,93.2,101.1,93.2s4,3.9,1.6,9.6
s-6.5,6.6-6.5,6.6S323,263,260,263c-60,0-103-28.2-103-
28.2s-6.1-2.8-6.5-6.8c-0.4-4-0.4-7.3,2.6-10.1c3-2.9,100.1-
92.1,100.1-92.1
S255,123,260,123z"/>
<path id="tree-part-2-shadow" fill="#09B56B"
d="M187,194.8c0,0,30,12.2,73,12.2c48,0,72.9-12,72.9-12s2.7-
0.7,5-3.9
c-12.5-11.6-27.1-25.1-39.8-36.8c-6.4,1.5-19.3,3.7-
38.1,3.7c-18.4,0-31.2-2.3-37.8-3.8c-12.8,11.8-27.6,25.4-
40.2,36.9
C184,193.3,187,194.8,187,194.8z"/>
<path id="tree-part-3" fill="#42CE8E" d="M260,107c0,0,2.5-
0.8,6.7,2.6c4.3,3.4,71.1,59.2,71.1,59.2s4,3.9,1.6,9.6
s-6.5,6.6-6.5,6.6S308,197,260,197c-43,0-73-12.2-73-
12.2s-6.1-2.8-6.5-6.8c-0.4-4-0.4-7.3,2.6-10.1c3-2.9,70.1-
58.1,70.1-58.1
S255,107,260,107z"/>
<path id="tree-part-3-shadow" fill="#09B56B"
d="M260,148c28,0,42.9-5,42.9-5s1.1-0.3,2.5-1.3c-17.7-14.7-
36.4-30.3-38.6-32.1
c-4.3-3.4-6.7-2.6-6.7-2.6c-5,0-6.9,2.8-6.9,2.8s-
19.8,16.3-
38.4,31.7c1.2,0.8,2.3,1.3,2.3,1.3S232,148,260,148z"/>
<path id="tree-part-4" fill="#42CE8E" d="M260,75c0,0,2.5-
0.8,6.7,2.6c4.3,3.4,41.1,41.2,41.1,41.2s4,3.9,1.6,9.6s-
6.5,6.6-6.5,6.6
s-14.9,5-42.9,5s-43-5.2-43-5.2s-6.1-2.8-6.5-6.8s-0.4-
7.3,2.6-10.1c3-2.9,40.1-40.1,40.1-40.1S255,75,260,75z"/>
</g>
<g id="gifts">
<g id="gift-yellow">

```

```

    <polygon id="gift-yellow-side-shadow" fill="#FFC84D"
points="196,345 176,339 176,319 196,323    "/>
    <polygon id="gift-yellow-side" fill="#FFDF9F"
points="216,339 196,345 196,323 216,319    "/>
    <polygon id="gift-yellow-top" fill="#FFDF9F"
points="174,315 196,311 218,315 196,319    "/>
    <polygon id="gift-yellow-top-shadow" fill="#FFC84D"
points="174,315 174,319 196,323 218,319 218,315 196,319
"/>
    <path id="gift-yellow-bow" fill="#D10F50"
d="M188.7,313.5c-0.6-0.4-1.2-0.9-1.6-1.5s-0.5-1.5-0.2-2.1c0.4-
0.8,1.3-1.2,2.2-1.3
    c1.8-0.2,3.8,0.6,4.7,2.2c0.8,1.2,0.9,2.7,1,4.1c-0.1-
1.4-0.1-2.9,0.3-4.3s1.4-2.7,2.8-3.1c0.5-0.1,1-0.2,1.5-0.2

c1.2,0,2.5,0.3,3,1.4c0.2,0.5,0.2,1.2,0,1.7C201.1,316.4,193,316
.2,188.7,313.5z"/>
  </g>
  <g id="gift-red">
    <polygon id="gift-red-side-shadow" fill="#D10F50"
points="335,345 315,339 315,319 335,323    "/>
    <polygon id="gift-red-side" fill="#DD4E85"
points="355,339 335,345 335,323 355,319    "/>
    <polygon id="gift-red-top" fill="#DD4E85"
points="313,315 335,311 357,315 335,319    "/>
    <polygon id="gift-red-top-shadow" fill="#D10F50"
points="313,315 313,319 335,323 357,319 357,315 335,319    "/>
    <path id="gift-red-bow" fill="#D10F50" d="M327.7,313.5c-
0.6-0.4-1.2-0.9-1.6-1.5c-0.4-0.6-0.5-1.5-0.2-2.1
    c0.4-0.8,1.3-1.2,2.2-1.3c1.8-
0.2,3.8,0.6,4.7,2.2c0.8,1.2,0.9,2.7,1,4.1c-0.1-1.4-0.1-
2.9,0.3-4.3c0.4-1.4,1.4-2.7,2.8-3.1
    c0.5-0.1,1-0.2,1.5-
0.2c1.2,0,2.5,0.3,3,1.4c0.2,0.5,0.2,1.2,0,1.7C340.1,316.4,332,
316.2,327.7,313.5z"/>
  </g>
</g>

```



```
<g id="Lights">
  <circle id="l1-r4-s2" class="shadow-2" opacity="0.5"
fill="#FFDF9F" cx="146" cy="256" r="15"/>
  <circle id="l1-r4-s1" class="shadow-1" opacity="0.5"
fill="#FFDF9F" cx="146" cy="256" r="10"/>
  <circle id="l1-r4" fill="#FFDF9F" cx="146" cy="256"
r="5"/>
  <circle id="l2-r4-s2" class="shadow-2" opacity="0.5"
fill="#FFDF9F" cx="196" cy="286" r="15"/>
  <circle id="l2-r4-s1" class="shadow-1" opacity="0.5"
fill="#FFDF9F" cx="196" cy="286" r="10"/>
  <circle id="l2-r4" fill="#FFDF9F" cx="196" cy="286"
r="5"/>
  <circle id="l3-r4-s2" class="shadow-2" opacity="0.5"
fill="#FFDF9F" cx="370" cy="256" r="15"/>
  <circle id="l3-r4-s1" class="shadow-1" opacity="0.5"
fill="#FFDF9F" cx="370" cy="256" r="10"/>
  <circle id="l3-r4" opacity="0.5" fill="#FFDF9F" cx="370"
cy="256" r="5"/>
  <circle id="l4-r4-s2" class="shadow-2" opacity="0.5"
fill="#FFDF9F" cx="320" cy="286" r="15"/>
  <circle id="l4-r4-s1" class="shadow-1" opacity="0.5"
fill="#FFDF9F" cx="320" cy="286" r="10"/>
  <circle id="l4-r4" fill="#FFDF9F" cx="320" cy="286"
r="5"/>
  <circle id="l5-r4-s2" class="shadow-2" opacity="0.5"
fill="#FFDF9F" cx="260" cy="296" r="15"/>
  <circle id="l5-r4-s1" class="shadow-1" opacity="0.5"
fill="#FFDF9F" cx="260" cy="296" r="10"/>
  <circle id="l5-r4" fill="#FFDF9F" cx="260" cy="296"
r="5"/>
  <circle id="l1-r3-s2" class="shadow-2" opacity="0.5"
fill="#FFDF9F" cx="196" cy="216" r="15"/>
  <circle id="l1-r3-s1" class="shadow-1" opacity="0.5"
fill="#FFDF9F" cx="196" cy="216" r="10"/>
  <circle id="l1-r3" fill="#FFDF9F" cx="196" cy="216"
r="5"/>
```

```
<circle id="l2-r3-s2" class="shadow-2" opacity="0.5"
fill="#FFDF9F" cx="320" cy="216" r="15"/>
<circle id="l2-r3-s1" class="shadow-1" opacity="0.5"
fill="#FFDF9F" cx="320" cy="216" r="10"/>
<ellipse id="l2-r3" fill="#FFDF9F" cx="320" cy="216"
rx="5.5" ry="5"/>
<circle id="l3-r3-s2" class="shadow-2" opacity="0.5"
fill="#FFDF9F" cx="260" cy="226" r="15"/>
<circle id="l3-r3-s1" class="shadow-1" opacity="0.5"
fill="#FFDF9F" cx="260" cy="226" r="10"/>
<circle id="l3-r3" fill="#FFDF9F" cx="260" cy="226"
r="5"/>
<circle id="l1-r2-s2" class="shadow-2" opacity="0.5"
fill="#FFDF9F" cx="196" cy="156" r="15"/>
<circle id="l1-r2-s1" class="shadow-1" opacity="0.5"
fill="#FFDF9F" cx="196" cy="156" r="10"/>
<circle id="l1-r2" opacity="0.5" fill="#FFDF9F" cx="196"
cy="156" r="5"/>
<circle id="l3-r2-s2" class="shadow-2" opacity="0.5"
fill="#FFDF9F" cx="320" cy="156" r="15"/>
<circle id="l3-r2-s1" class="shadow-1" opacity="0.5"
fill="#FFDF9F" cx="320" cy="156" r="10"/>
<circle id="l3-r2" fill="#FFDF9F" cx="320" cy="156"
r="5"/>
<circle id="l2-r2-s2" class="shadow-2" opacity="0.5"
fill="#FFDF9F" cx="260" cy="166" r="15"/>
<circle id="l2-r2-s1" class="shadow-1" opacity="0.5"
fill="#FFDF9F" cx="260" cy="166" r="10"/>
<circle id="l2-r2" fill="#FFDF9F" cx="260" cy="166"
r="5"/>
<circle id="l1-r1-s2" class="shadow-2" opacity="0.5"
fill="#FFDF9F" cx="260" cy="106" r="15"/>
<circle id="l1-r1-s1" class="shadow-1" opacity="0.5"
fill="#FFDF9F" cx="260" cy="106" r="10"/>
<circle id="l1-r1" fill="#FFDF9F" cx="260" cy="106"
r="5"/>
```

```

    <path id="star-s2" class="shadow-2" opacity="0.2"
fill="#FFDF9F" d="M292.5,71.5l1.4-1.4c5.9-5.9,7.7-14.8,4.5-
22.5l0,0
    c-3.2-7.7-10.6-12.7-18.9-12.8l-0.3-0.8c-3.2-7.9-10.7-13-
19.2-13c-8.5,0-16,5.1-19.2,13l-0.3,0.8c-8.3,0.1-15.7,5.1-
18.9,12.8
    c-3.2,7.8-1.4,16.6,4.5,22.5l1.4,1.4l-0.5,1.1c-3.3,8.3-
1,17.7,5.8,23.4c3.8,3.3,8.6,4.9,13.4,4.9c3.7,0,7.3-1,10.6-
3l3.1-1.9
    l3.1,1.9c3.2,1.9,6.9,2.9,10.6,2.9c4.9,0,9.7-1.8,13.4-
4.9c6.8-5.8,9.1-15.2,5.8-23.4l292.5,71.5z"/>
    <path id="star-s1" class="shadow-1" opacity="0.2"
fill="#FFDF9F" d="M283.6,68.6l1-1c4.3-4.3,5.6-10.7,3.3-
16.3l0,0c-2.3-5.6-7.7-9.2-13.7-9.3
    l-0.2-0.6c-2.3-5.7-7.8-9.4-13.9-9.4c-6.2,0-11.6,3.7-
13.9,9.4l-0.2,0.6c-6,0.1-11.4,3.7-13.7,9.3c-2.3,5.6-
1.1,12,3.3,16.3l1,1
    l-0.3,0.8c-2.4,6-
0.7,12.8,4.2,17c2.8,2.4,6.2,3.6,9.7,3.6c2.7,0,5.3-0.7,7.7-
2.1l2.3-1.4l2.3,1.4c2.3,1.4,5,2.1,7.7,2.1
    c3.6,0,7-1.3,9.7-3.6c4.9-4.2,6.6-11,4.2-
17l283.6,68.6z"/>
    <path id="star" fill="#FFDF9F" d="M278.6,55.1c-0.8-1.9-
2.6-3.1-4.6-3.1h-6.6l-2.7-6.9c-0.8-1.9-2.6-3.1-4.6-3.1s-
3.9,1.2-4.6,3.1
    l-2.7,6.9H246c-2,0-3.8,1.2-4.6,3.1c-0.8,1.9-
0.3,4,1.1,5.4l5.7,5.7l-2.8,6.9c-0.8,2-
0.2,4.3,1.4,5.7c1.6,1.4,4,1.6,5.8,0.5
    l7.4-4.5l7.4,4.5c0.8,0.5,1.7,0.7,2.6,0.7c1.2,0,2.3-
0.4,3.2-1.2c1.6-1.4,2.2-3.7,1.4-5.7l-2.8-6.9l5.7-5.7
    C279,59.1,279.4,57,278.6,55.1z"/>
</g>
<g id="cat-container">
    <g>
        <defs>

```

```

        <path id="SVGID_1_"
d="M215,240.3c5.4,0.7,11.3,2.3,14,7c3.1-4.1,8-6.5,13.1-
7.3c5.1-0.8,10.3,0,15.2,1.3
        c-0.4,3.5,0.7-44.3,0.7-44.3l-59-2c0,0-
2.1,51.2,0.5,47.4C202.6,237.7,210.4,239.6,215,240.3z"/>
    </defs>
    <clipPath id="SVGID_2_">
        <use xlink:href="#SVGID_1_" overflow="visible"/>
    </clipPath>
    <g id="cat-clippath" clip-path="url(#SVGID_2_)">
        <g id="cat">
            <path id="cat-body" fill="#39445B" d="M213.6,237.8c-
0.3,4.5,0.5,9.9,1,13.1c0.2,1.3,0.5,2.7,1.6,3.5c0.7,0.5,1.5,0.6
,2.3,0.8
            c7.6,1.1,15.4,0.7,22.8-1.3c0.5-0.1,1-0.3,1.4-
0.6c0.6-0.5,0.7-1.4,0.8-2.2c0.2-2.1,0.4-4.2,0.6-6.4c0.2-
2.1,0.4-4.2,0.2-6.2
            C234.9,242.8,222.9,242.2,213.6,237.8z"/>
            <path id="cat-body-shadow" fill="#171E2B"
d="M240.2,230.2c-1.6-1.1-3.6-1.5-5.5-1.8c-5-0.6-16.2-1.2-
19.5,3.8
            c-0.9,1.4-1.4,3.4-1.6,5.6c9.3,4.5,21.3,5,30.7,0.8c0-
0.6-0.1-1.1-0.2-1.7C243.6,234.2,242.4,231.7,240.2,230.2z"/>
        <g>
            <path id="cat-scarf" fill="#DD4E85"
d="M212.3,241.4c-0.7,3.2-1.2,6.6-2.9,9.4c-0.1,0.2-0.2,0.3-
0.2,0.5c0,0.2,0.2,0.3,0.4,0.4
            c1.5,0.9,3.2,1.4,5,1.5c0.4,0,0.9,0,1.2-0.3c0.3-
0.2,0.4-0.6,0.5-0.9c1.2-3.7,2-7.6,2.8-11.5c-2.2-0.8-4.2-2-5.9-
3.5
            C212.8,238.8,212.5,240.4,212.3,241.4z"/>
            <path id="cat-scarf-shadow" fill="#D10F50"
d="M220.1,232.5c-0.8-0.9-3.1-0.8-4-0.2c-1.4,0.8-2.3,2.9-
2.9,4.9
            c1.7,1.5,3.7,2.7,5.9,3.5c0.1-0.6,0.2-1.2,0.4-
1.9C219.8,237.3,221.2,233.6,220.1,232.5z"/>

```

```

        <path id="cat-scarf-neck" fill="#DD4E85"
d="M240.1,235.5c0.7-0.2,1.4-0.5,2-1c1.2-1,1.3-2.7,1.4-4.2c0-
0.8,0-1.7-0.6-2.3
        c-0.7-0.7-1.8-0.4-2.8-0.2c-4.1,1.1-8.3,1.5-
12.5,1.3c-2.1-0.1-4.2-0.3-6.2-0.7c-1.6-0.3-3.9-1.4-5.6-0.7
        c-1.1,0.5-1.3,1.5-1.5,2.6c-0.3,1.8-
0.8,3.6,1.1,4.4c3.6,1.5,7.7,2.2,11.6,2.3C231.5,237.3,235.9,236
.7,240.1,235.5z"/>
    </g>
    <g>
        <g id="cat-line-left-up">
            <rect x="208" y="221.3" fill="#171E2B" width="4.2"
height="2"/>
        </g>
        <g id="cat-line-left-down">
            <rect x="208" y="224.3" fill="#171E2B" width="5.9"
height="2"/>
        </g>
        <g id="cat-line-right-up">
            <rect x="243.4" y="221.3" fill="#171E2B"
width="4.2" height="2"/>
        </g>
        <g id="cat-line-right-down">
            <rect x="241.7" y="224.3" fill="#171E2B"
width="5.9" height="2"/>
        </g>
        <g id="cat-face">
            <path fill="#39445B" d="M239.5,210.9c0,0-5.7-2.9-
11.3-2.9c-5.7,0-11.4,2.9-11.4,2.9l-5.6,0c0,0,0,8.8,0,11.3
            c0,7.8,9.2,11.3,17,11.3c7.8,0,17-3.5,17-11.3c0-
3.1,0-11.3,0-11.3L239.5,210.9z"/>
        </g>
        <g id="cat-eye-left">
            <circle fill="#FFDF9F" cx="219.3" cy="222.5"
r="4.5"/>
        </g>
        <g id="cat-eye-right">

```

```

        <circle fill="#FFDF9F" cx="237" cy="222.5"
r="4.5"/>
    </g>
    <g id="cat-eye-left-pupil">
        <circle fill="#171E2B" cx="219.3" cy="222.5"
r="2.8"/>
    </g>
    <g id="cat-eye-right-pupil">
        <circle fill="#171E2B" cx="237" cy="222.5"
r="2.8"/>
    </g>
    </g>
</g>
</g>
</div>
</body>
</html>

```

Файл src/main.js:

```

import { MainPage } from "../pages/main/MainPage.js";
import 'https://code.jquery.com/jquery-3.3.1.min.js'
export default window.jQuery.noConflict(true)
import "../scripts/snowfall.js";

const root = document.getElementById('root');

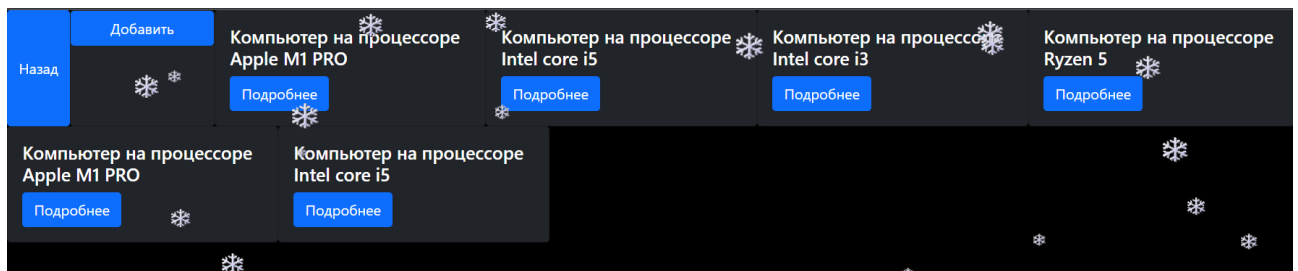
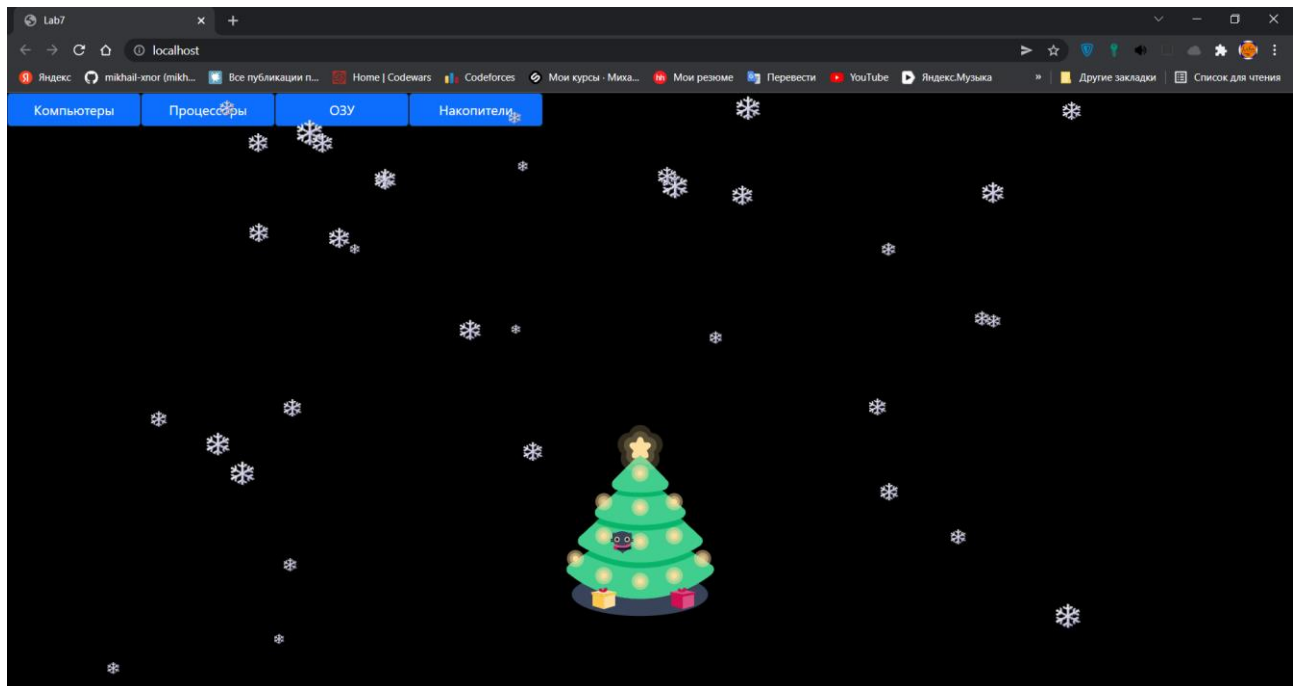
const mainPage = new MainPage(root)

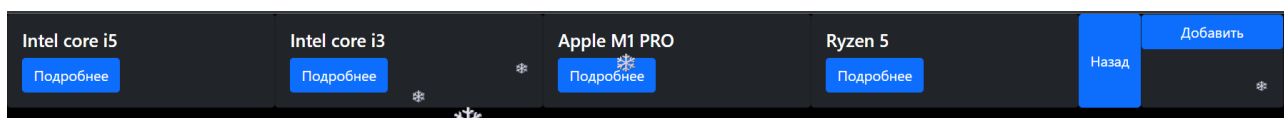
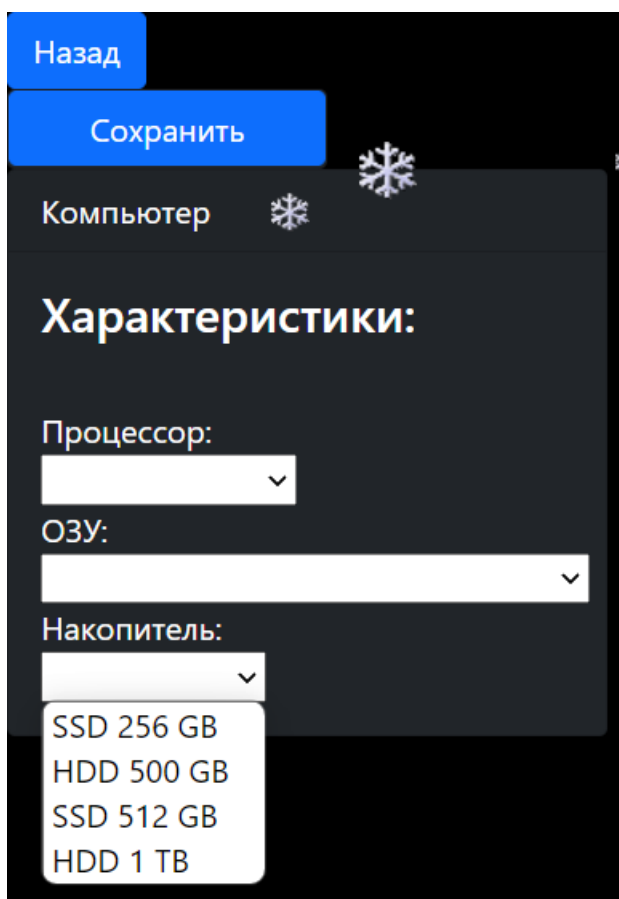
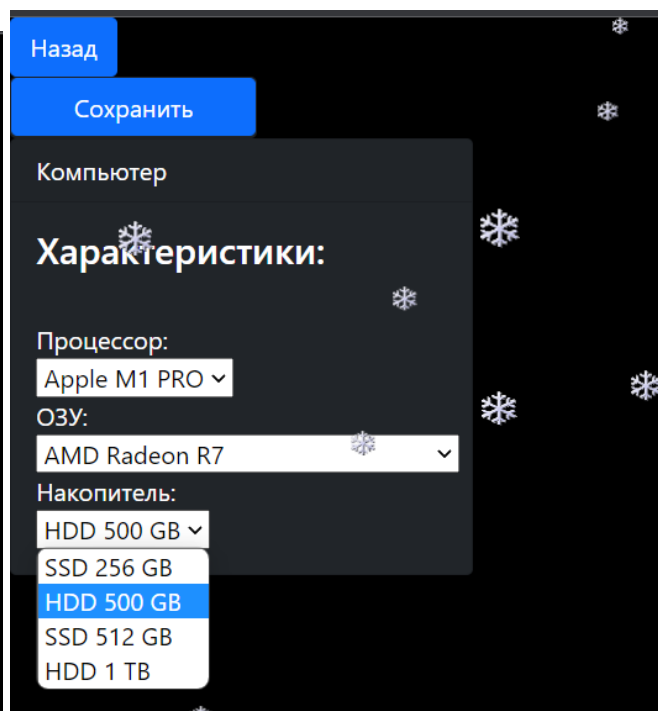
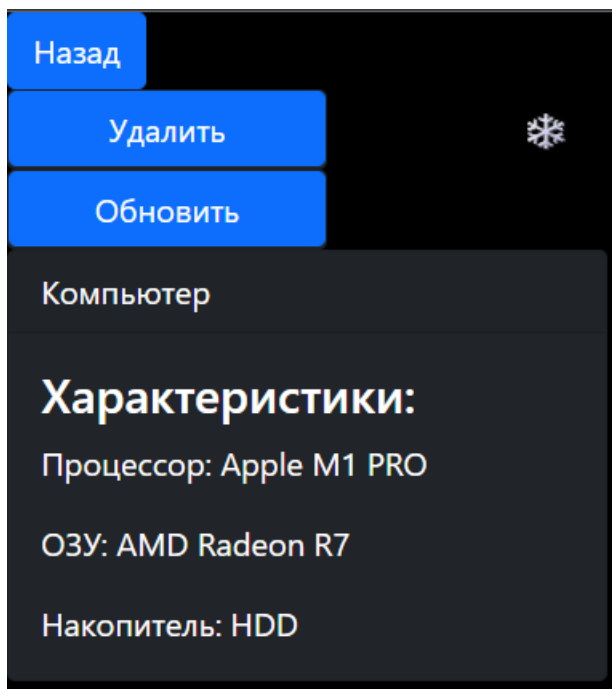
document.body.setAttribute("style", "background-color: black")

mainPage.render()

```

Пользовательский интерфейс:





Назад

Удалить

Обновить

Процессор

Intel core i5

Рабочая частота: 5Ghz

Архитектура: x86

Назад

Сохранить

Процессор

Intel core i5

Рабочая частота: 5Ghz

Архитектура: x86

Назад

Сохранить

Процессор

Рабочая частота:

Архитектура:

AMD Radeon R5

AMD Radeon R7

Crucial [CT25664BD160B]

Silicon Power [SP004GLLTU160N02]

Назад

Добавить

Подробнее

Подробнее

Подробнее

Подробнее

Назад

Удалить

Обновить

ОЗУ

AMD Radeon R5

Тип памяти: DDR3

Частота работы: 1600 MHz

Объем: 2

Назад

Сохранить

ОЗУ

AMD Radeon R5

Тип памяти:

DDR3

Частота работы:

1600 MHz

Объем:

2

Назад

Сохранить

ОЗУ

Тип памяти:

Частота работы:

Объем:

Тип накопителя: SSD Объем: 256 GB <div>Подробнее</div>	Тип накопителя: HDD Объем: 500 GB <div>Подробнее</div>	Тип накопителя: SSD Объем: 512 GB <div>Подробнее</div>	Тип накопителя: HDD Объем: 1 TB <div>Подробнее</div>	Назад	Добавить
--	--	--	--	-------	----------

Назад

Удалить

Обновить

Диск

Тип накопителя: SSD

Скорость записи: 1000 MB/s

Скорость чтения: 500 MB/s

Объем: 256 GB

Назад

Сохранить

Диск

Тип накопителя:

SSD

Скорость записи:

500 MB/s

Скорость чтения:

1000 MB/s

Объем:

256 GB

Назад

Сохранить

Диск

Тип накопителя:



Скорость записи:

Скорость чтения:



Объем: