

Министерство науки и высшего образования Российской
Федерации

Федеральное государственное бюджетное
образовательное

учреждение высшего образования

«Ульяновский государственный технический
университет»

Факультет информационных систем и технологий

Кафедра «Информационные системы»

Дисциплина «Интернет программирование»

Практическая работа № 3

Выполнил(а):

студент гр. ПИБд -21

Данилов В.В.

Проверил:

доцент кафедры

Филиппов А.А.

Ульяновск, 2023г.

Интернет магазин бытовой техники.

Цель работы:

1. По результатам лабораторной работы 2 необходимо добавить CRUD операции для страницы администратора. На данной странице администратор должен иметь возможность добавлять, редактировать, удалять и просматривать записи о некоторых объектах. Для создания и редактирования записей необходимо использовать отдельную страницу. Для просмотра сведений о выбранном объекте также необходимо использовать отдельную таблицу. В отдельных случаях для редактирования/добавления (только для этих операций) записей можно использовать диалоговые (модальные) окна. См. пример для лекции № 3.

2. В качестве сервера необходимо использовать пакет `json-server`. Для работы с сервером необходимо создать отдельный `js`-модуль с набором необходимых асинхронных функций (`async/await`). Запросы к серверу необходимо выполнять через `Fetch API`. См. пример для лекции № 3.

3. Для запуска проекта необходимо использовать задачу (скрипт) из файла `package.json`, которая будет запускать `vite` и `json-server`. См. пример для лекции № 3.

4. Разработку необходимо осуществлять в `VS Code` с установленным плагином `ESLint`. Более подробно см. в файле `readme.md` в примере для лекции № 3. Проект следует открыть в `VS Code` через меню «`File -> Open Folder`». В диалоге необходимо выбрать каталог (папку) с проектом. За основу лучше взять пример для лекции № 3. Пример загружен в `LMS` в раздел лекции № 3.

5. Файлы проекта и отчет должны быть загружены в `git`-репозиторий на сервере <http://student.git.athene.tech/>
Реализация:

lines.js

// модуль с логикой

```
import { hideUpdateModal, showUpdateModal } from "./lines-modal";
import {
  createLine, deleteLine, getAllItemTypes, getAllLines,
  updateLine, getAllProductTypes, getLine, getCatalogLines,
} from "./lines-rest-api";
import {
```

```
    // eslint-disable-next-line max-len
    cntrls, createItemsOption, createTableRow, imagePlaceholder,
    createProductOption, createCatalogItem,
  } from "./lines-ui";
```

```
async function drawItemsSelect() {
  // вызов метода REST API для получения списка типов
  товаров
  const data = await getAllItemTypes();
  // очистка содержимого select
  // удаляется все, что находится между тегами
  <select></select>
  // но не атрибуты
  cntrls.itemsType.innerHTML = "";
  // пустое значение
  cntrls.itemsType.appendChild(createItemsOption("Выберите
  значение", "", true));
  // цикл по результату ответа от сервера
  // используется лямбда-выражение
  // (item) => {} аналогично function(item) {}
  data.forEach((item) => {
```

```
    cntrls.itemsType.appendChild(createItemsOption(item.name,
    item.id));
  });
}
```

```
async function drawProductSelect() {
  // вызов метода REST API для получения списка типов
  товаров
  const data = await getAllProductTypes();
  // очистка содержимого select
  // удаляется все, что находится между тегами
  <select></select>
  // но не атрибуты
  cntrls.productType.innerHTML = "";
  // пустое значение
```

```
cntrls.productType.appendChild(createProductOption("Выберите значение", "", true));
```

```
    // цикл по результату ответа от сервера  
    // используется лямбда-выражение  
    // (item) => {} аналогично function(item) {}  
    data.forEach((item) => {
```

```
cntrls.productType.appendChild(createProductOption(item.name,  
item.id));
```

```
    });  
}
```

```
async function drawLinesTable() {  
    console.info("Try to load data");  
    if (!cntrls.table) {  
        return;  
    }
```

```
    // вызов метода REST API для получения всех записей  
    const data = await getAllLines();  
    // очистка содержимого table  
    // удаляется все, что находится между тегами
```

```
<table></table>
```

```
    // но не атрибуты
```

```
    cntrls.table.innerHTML = "";
```

```
    // цикл по результату ответа от сервера
```

```
    // используется лямбда-выражение
```

```
    // (item, index) => {} аналогично function(item, index) {}
```

```
    data.forEach((item, index) => {
```

```
        cntrls.table.appendChild(  
            createTableRow(  
                item,  
                index,  
                // функции передаются в качестве параметра  
                // это очень удобно, так как аргументы функций  
                // доступны только  
                // в данном месте кода и не передаются в  
                // сервисные модули  
                () => showUpdateModal(item),
```

```
                // в данном месте кода и не передаются в  
                // сервисные модули
```

```
                // в данном месте кода и не передаются в  
                // сервисные модули
```

```
                // в данном месте кода и не передаются в  
                // сервисные модули
```

```
                () => showUpdateModal(item),
```

```

        () => removeLine(item.id),
    ),
);
});
}

```

```

async function drawCatalog(id) {
    console.info("Try to load data");
    // вызов метода REST API для получения всех записей
    const data = await getCatalogLines(id);

    // цикл по результату ответа от сервера
    // используется лямбда-выражение
    // (item, index) => {} аналогично function(item, index) {}
    data.forEach((item) => {
        const catalogItem = createCatalogItem({
            image: item.image,
            model: item.model,
            price: item.price,
            specific: item.specific,
            id: item.id,
        });
        cntrls.catalogContainer.innerHTML += catalogItem;
    });
}

```

```

async function addLine(product, item, model, image, price, count,
description, specific) {
    console.info("Try to add item");
    // вызов метода REST API для добавления записи
    const data = await createLine(product, item, model, image,
price, count, description, specific);
    console.info("Added");
    console.info(data);
    // загрузка и заполнение table
    drawLinesTable();
}

```

```
async function editLine(id, product, item, model, image, price,
count, description, specific) {
  console.info("Try to update item");
  // вызов метода REST API для обновления записи
  // eslint-disable-next-line max-len
  const data = await updateLine(id, product, item, model,
image, price, count, description, specific);
  console.info("Updated");
  console.info(data);
  // загрузка и заполнение table
  drawLinesTable();
}
```

```
async function removeLine(id) {
  if (!confirm("Do you really want to remove this item?")) {
    console.info("Canceled");
    return;
  }
  console.info("Try to remove item");
  // вызов метода REST API для удаления записи
  const data = await deleteLine(id);
  console.info(data);
  // загрузка и заполнение table
  drawLinesTable();
}
```

```
// функция для получения содержимого файла в виде base64
строки
```

```
// https://ru.wikipedia.org/wiki/Base64
```

```
async function readFile(file) {
  const reader = new FileReader();

  // создание Promise-объекта для использования функции
  // с помощью await (асинхронно) без коллбэков (callback)
  // https://learn.javascript.ru/promise
  return new Promise((resolve, reject) => {
    // 2. "Возвращаем" содержимое когда файл прочитан
    // через вызов resolve
  });
}
```

```

    // Если не использовать Promise, то всю работу по
    // взаимодействию
    // с REST API пришлось бы делать в обработчике
    (callback) функции
    // onloadend
    reader.onloadend = () => {
        const fileContent = reader.result;
        // Здесь могла бы быть работа с REST API
        // Чтение заканчивает выполняться здесь
        resolve(fileContent);
    };
    // 3. Возвращаем ошибку
    reader.onerror = () => {
        // Или здесь в случае ошибки
        reject(new Error("oops, something went wrong with the
file reader."));
    };
    // Шаг 1. Сначала читаем файл
    // Чтение начинает выполняться здесь
    reader.readAsDataURL(file);
});
}

// функция для обновления блока с превью выбранного
// изображения
async function updateImagePreview() {
    // получение выбранного файла
    // возможен выбор нескольких файлов, поэтому необходимо
    // получить только первый
    const file = cntrls.image.files[0];
    // чтение содержимого файла в виде base64 строки
    const fileContent = await readFile(file);
    console.info("base64 ", fileContent);
    // обновление атрибута src для тега img с id image-preview
    cntrls.imagePreview.src = fileContent;
}

// Функция для обработки создания и редактирования
// элементов таблицы через модальное окно

```

```
// Если хотите делать через страницу, то удалите эту функцию
export function linesForm() {
  console.info("linesForm");

  // загрузка и заполнение select со списком товаров
  drawItemsSelect();
  drawProductSelect();
  // загрузка и заполнение table
  drawLinesTable();

  // Вызов функции обновления превью изображения при
  возникновении
  // события onchange в тэге input с id image
  cntrls.image.addEventListener("change", () =>
  updateImagePreview());

  // обработчик события нажатия на кнопку для показа
  модального окна
  cntrls.button.addEventListener("click", () =>
  showUpdateModal(null));

  // обработчик события отправки формы
  // возникает при нажатии на кнопку (button) с типом submit
  // кнопка должна находиться внутри тега form
  cntrls.form.addEventListener("submit", async (event) => {
    console.info("Form onSubmit");
    // отключение стандартного поведения формы при
    отправке
    // при отправке страница обновляется и JS перестает
    работать
    event.preventDefault();
    event.stopPropagation();
    // если форма не прошла валидацию, то ничего делать не
    нужно
    if (!cntrls.form.checkValidity()) {
      return;
    }

    let imageBase64 = "";
```



```

    // Получение выбранного пользователем изображения в
виде base64 строки
    // Если пользователь ничего не выбрал, то не нужно
сохранять в БД
    // дефолтное изображение
    if (cntrls.imagePreview.src !== imagePlaceholder) {
        // Загрузка содержимого атрибута src тэга img с id
image-preview
        // Здесь выполняется HTTP запрос с типом GET
const result = await fetch(cntrls.imagePreview.src);
        // Получение из HTTP-ответа бинарного содержимого
const blob = await result.blob();
        // Получение base64 строки для файла
        // Здесь выполняется Promise из функции readFile
        // Promise позволяет писать линейный код для работы
с асинхронными методами
        // без использования обработчиков (callback) с
помощью await
        imageBase64 = await readFile(blob);
    }

    // получение id строки для редактирования
// это значение содержится в скрытом input
const currentId = cntrls.lineId.value;
// если значение id не задано,
// то необходимо выполнить добавление записи
// иначе обновление записи
if (!currentId) {
    await addLine(
        cntrls.productType.value,
        cntrls.itemsType.value,
        cntrls.model.value,
        imageBase64,
        cntrls.price.value,
        cntrls.count.value,
        cntrls.description.value,
        cntrls.specific.value,
    );
} else {

```

```

        await editLine(
            currentId,
            cntrls.productType.value,
            cntrls.itemsType.value,
            cntrls.model.value,
            imageBase64,
            cntrls.price.value,
            cntrls.count.value,
            cntrls.description.value,
            cntrls.specific.value,
        );
    }

    // после выполнения добавления/обновления модальное
    окно скрывается
    hideUpdateModal();
});
}

export async function linesPageForm(id) {
    console.info("linesPageForm");
    drawCatalog(id);
}

export async function productPage() {
    console.info("productPage");
    // получение параметров GET-запроса из URL
    // параметры перечислены после символа ? (?
    id=1&color=black&...)
    const urlParams = new URLSearchParams(location.search);

    // получение значения конкретного параметра (id)
    // указан только при редактировании
    const currentId = urlParams.get("id");
    const item = await getLine(currentId);
    cntrls.imagePreview.src = item.image ? item.image :
    imagePlaceholder;
    cntrls.model.innerText = item.model;
    cntrls.price.innerText = item.price;

```

```
    cntrls.description.innerText = item.description;
    cntrls.specific.innerText = item.specific;
}
```

lines-ui.js

```
// модуль для работы с элементами управления

// объект для удобного получения элементов
// при обращении к атрибуту объекта вызывается
// нужная функция для поиска элемента
export const cntrls = {
  button: document.getElementById("items-add"),
  table: document.querySelector("#items-table tbody"),
  form: document.getElementById("items-form"),
  lineId: document.getElementById("items-line-id"),
  productType: document.getElementById("product"),
  itemsType: document.getElementById("item"),
  model: document.getElementById("model"),
  price: document.getElementById("price"),
  count: document.getElementById("count"),
  description: document.getElementById("description"),
  specific: document.getElementById("specific"),
  image: document.getElementById("image"),
  imagePreview: document.getElementById("image-preview"),
  catalogContainer: document.getElementById("items-catalog"),
};

// Дефолтное превью
export const imagePlaceholder =
  "https://via.placeholder.com/200";

// функция создает тег option для select
// <option value="" selected>name</option>
export function createItemsOption(name, value = "", isSelected =
false) {
  const option = document.createElement("option");
  option.value = value || "";
  option.selected = isSelected;
  option.text = name;
}
```

```

    return option;
}

export function createProductOption(name, value = "", isSelected
= false) {
    const option = document.createElement("option");
    option.value = value || "";
    option.selected = isSelected;
    option.text = name;
    return option;
}

// функция создает ссылку (a) для таблицы
// содержимое тега a заполняется необходимой иконкой (icon)
// при нажатии вызывается callback
// ссылка "оборачивается" тегом td
// <td><a href="#" onclick="callback()"><i class="fa-solid
icon"></i></a></td>
function createTableAnchor(icon, callback) {
    const i = document.createElement("i");
    i.classList.add("fa-solid", icon);

    const a = document.createElement("a");
    a.href = "#";
    a.appendChild(i);
    a.onclick = (event) => {
        // чтобы в URL не добавлялась решетка
        event.preventDefault();
        event.stopPropagation();
        callback();
    };

    const td = document.createElement("td");
    td.appendChild(a);
    return td;
}

// функция создает колонку таблицы с текстом value
// <td>value</td>

```

```

function createTableColumn(value) {
  const td = document.createElement("td");
  td.textContent = value;
  return td;
}

export function createCatalogItem(item) {
  const imageUrl = item.image ? item.image :
'https://via.placeholder.com/200';
  const catalogHtml = `
    <div class="p-2 t-2">
      <div class="navbar-nav d-flex flex-lg-row p-2 t-2"
style="background-color: white; border-radius: 12px;">
        <a class="image-container col-lg-2"></a>
        <a class="nav-link col-lg-8" href="/Product.html?id=$
{item.id}">${item.model} ${item.specific}</a>
        <div class="nav-link d-flex flex-column col-lg-2">
          <a class="nav-link" style="font-size: larger;">${
item.price}</a>
          <button class="btn btn-outline-warning"
type="submit">Купить</button>
        </div>
      </div>
    </div>
  `;
  return catalogHtml;
}

// функция создает строку таблицы
// <tr>
// <th scope="row">index + 1</th>
// <td>item.items.name</td>
// <td>parseFloat(item.price).toFixed(2)</td>
// <td>item.count</td>
// <td>parseFloat(item.sum).toFixed(2)</td>
// <td><a href="#" onclick="editCallback()"><i class="fa-solid
fa-pencil"></i></a></td>

```

```

// <td><a href="#" onclick="editPageCallback()"><i class="fa-
solid fa-pen-to-square"></i></a></td>
// <td><a href="#" onclick="deleteCallback()"><i class="fa-
solid fa-trash"></i></a></td>
// </tr>
export function createTableRow(item, index, editCallback,
deleteCallback) {
  const rowNumber = document.createElement("th");
  rowNumber.scope = "row";
  rowNumber.textContent = index + 1;

  const row = document.createElement("tr");
  row.id = `line-${item.id}`;

  row.appendChild(rowNumber);
  row.appendChild(createTableColumn(item.items.name));
  row.appendChild(createTableColumn(item.model));

  row.appendChild(createTableColumn(parseFloat(item.price).toFixed(2)));
  row.appendChild(createTableColumn(item.count));

  row.appendChild(createTableColumn(parseFloat(item.sum).toFixed(2)));
  // редактировать в модальном окне
  row.appendChild(createTableAnchor("fa-pencil", editCallback));
  // удаление
  row.appendChild(createTableAnchor("fa-trash",
deleteCallback));
  return row;
}

```

lines-rest-api.js

```
// модуль для работы с REST API сервера
```

```
// адрес сервера
```

```
const serverUrl = "http://localhost:8081";
```

```
// функция возвращает объект нужной структуры для
отправки на сервер
function createLineObject(product, item, model, image, price,
count, description, specific) {
  return {
    productId: product,
    itemsId: item,
    model,
    image,
    price: parseFloat(price).toFixed(2),
    count,
    sum: parseFloat(price * count).toFixed(2),
    description,
    specific,
  };
}
```

```
// обращение к серверу для получения всех типов товара (get)
export async function getAllItemTypes() {
  const response = await fetch(` ${serverUrl}/items` );
  if (!response.ok) {
    throw response.statusText;
  }
  return response.json();
}
```

```
export async function getAllProductTypes() {
  const response = await fetch(` ${serverUrl}/product` );
  if (!response.ok) {
    throw response.statusText;
  }
  return response.json();
}
```

```
// обращение к серверу для получения всех записей (get)
export async function getAllLines() {
  const response = await fetch(` ${serverUrl}/lines?
_expand=items` );
  if (!response.ok) {
```

```

        throw response.statusText;
    }
    return response.json();
}
export async function getCatalogLines(id) {
    const response = await fetch(` ${serverUrl}/lines?itemsId=${
{id}} `);
    if (!response.ok) {
        throw response.statusText;
    }
    return response.json();
}

// обращение к серверу для получения записи по первичному
// ключу (id) (get)
// id передается в качестве части пути URL get-запроса
export async function getLine(id) {
    const response = await fetch(` ${serverUrl}/lines/${id}?
_expand=items `);
    if (!response.ok) {
        throw response.statusText;
    }
    return response.json();
}

// обращение к серверу для создания записи (post)
// объект отправляется в теле запроса (body)
export async function createLine(product, item, model, image,
price, count, description, specific) {
    // eslint-disable-next-line max-len
    const itemObject = createLineObject(product, item, model,
image, price, count, description, specific);

    const options = {
        method: "POST",
        body: JSON.stringify(itemObject),
        headers: {
            "Accept": "application/json",
            "Content-Type": "application/json",

```



```

    },
  };

  const response = await fetch(` ${serverUrl}/lines` , options);
  if (!response.ok) {
    throw response.statusText;
  }
  return response.json();
}

// обращение к серверу для обновления записи по id (put)
// объект отправляется в теле запроса (body)
// id передается в качестве части пути URL get-запроса
// eslint-disable-next-line max-len
export async function updateLine(id, product, item, model,
image, price, count, description, specific) {
  // eslint-disable-next-line max-len
  const itemObject = createLineObject(product, item, model,
image, price, count, description, specific);

  const options = {
    method: "PUT",
    body: JSON.stringify(itemObject),
    headers: {
      "Accept": "application/json",
      "Content-Type": "application/json",
    },
  };

  const response = await fetch(` ${serverUrl}/lines/${id}` ,
options);
  if (!response.ok) {
    throw response.statusText;
  }
  return response.json();
}

// обращение к серверу для удаления записи по id (delete)
// id передается в качестве части пути URL get-запроса

```

```
export async function deleteLine(id) {
  const options = {
    method: "DELETE",
  };

  const response = await fetch(`${serverUrl}/lines/${id}`,
options);
  if (!response.ok) {
    throw response.statusText;
  }
  return response.json();
}
```

lines-modal.js

```
// Модуль для работы с модальным окном
```

```
// импорт компонента Modal из bootstrap
```

```
import { Modal } from "bootstrap";
```

```
import { cntrls, imagePlaceholder } from "./lines-ui";
```

```
// поиск модального окна на странице
```

```
const modal = document.getElementById("items-update");
```

```
// если он найден, то создается экземпляр компонента Modal
```

```
// для программного управления модальным окном
```

```
const myModal = modal ? new Modal(modal, {}) : null;
```

```
// поиск тега с заголовком модального окна для его смены
```

```
const modalTitle = document.getElementById("items-update-
title");
```

```
// обнуление значений модального окна, т. к.
```

```
// используется одно окно для всех операций
```

```
function resetValues() {
```

```
  cntrls.lineId.value = "";
```

```
  cntrls.productType.value = "";
```

```
  cntrls.itemsType.value = "";
```

```
  cntrls.model.value = "";
```

```
  cntrls.image.value = "";
```

```
  cntrls.imagePreview.src = imagePlaceholder;
```

```

    cntrls.price.value = parseFloat(0).toFixed(2);
    cntrls.count.value = 0;
    cntrls.description.value = "";
    cntrls.specific.value = "";
}
// функция для показа модального окна
// перед показом происходит заполнение формы для
редактирования
// если объект item не пуст
export function showUpdateModal(item) {
    modalTitle.innerHTML = item === null ? "Добавить" :
    "Изменить";
    console.info(item);

    if (item) {
        cntrls.lineId.value = item.id;
        cntrls.productType.value = item.productId;
        cntrls.itemsType.value = item.itemsId;
        cntrls.model.value = item.model;
        // заполнение превью
        // Если пользователь выбрал изображение, то оно
загружается
        // в тэг image с id image - preview
        // иначе устанавливается заглушка, адрес которой
указан в imagePlaceholder
        cntrls.imagePreview.src = item.image ? item.image :
imagePlaceholder;
        cntrls.price.value = item.price;
        cntrls.count.value = item.count;
        cntrls.description.value = item.description;
        cntrls.specific.value = item.specific;
    } else {
        resetValues();
    }

    myModal.show();
}

// функция для скрытия модального окна

```

```

export function hideUpdateModal() {
  resetValues();

  // удаление класса was-validated для скрытия результатов
  валидации
  cntrls.form.classList.remove("was-validated");

  myModal.hide();
}

```

index.html

```

<html lang="ru">

  <head>
    <meta charset = "utf-8">
    <title> Техник </title>
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <script type="module"
src="./node_modules/bootstrap/dist/js/bootstrap.min.js"></scrip
t>
    <link
href="./node_modules/bootstrap/dist/css/bootstrap.min.css"
rel="stylesheet"/>
    <link href="./node_modules/@fortawesome/fontawesome-
free/css/all.min.css" rel="stylesheet"/>
    <link rel = "stylesheet" href = "./style.css">
  </head>

  <body class="h-100 d-flex flex-column">

    <header>
      <nav class="navbar navbar-expand-md">
        <div class="container-fluid">
          <a class = "navbar-brand" href = "./"
style="color: #6A3502;">
            <i class="fa-solid fa-
microchip"></i>
            Техник

```

```

        </a>

        <form class="d-flex">
            <input class="form-control"
type="search" placeholder="Поиск по сайту" aria-
label="Search">
                <button class="btn btn-outline-*"
type="submit" style="color: #6A3502;"><i class="fa-solid fa-
search"></i></button>
            </form>

            <a class="nav-link"
href="./Login.html">Вход</a>
        </div>
    </nav>
</header>

    <main class="container-fluid" style="background-color:
antiquewhite;">
        <div class="row">
            <article class="col-lg-3">
                <button class="navbar-toggler"
type="button" data-bs-toggle="collapse" data-bs-
target="#navbarCatalog"
                    aria-controls="navbarNav"
aria-expanded="false" aria-label="Toggle navigation">
                    <h3>Каталог</h3>
                </button>

                <div class="navbar-collapse collapse"
id="navbarCatalog">
                    <div class="navbar-nav"
style="background-color: white; border-radius: 12px;">
                        <li class="nav-item">
                            <a class="nav-link p-2"
href="./Catalog.html?id=1"> Бытовая техника</a>
                        </li>
                        <li class="nav-item">

```

```
                <a class="nav-link p-2"
href="/Catalog.html?id=2"> Красота и здоровье</a>
            </li>
            <li class="nav-item">
                <a class="nav-link p-2"
href="/Catalog.html?id=3"> Смартфоны и фототехника</a>
            </li>
            <li class="nav-item">
                <a class="nav-link p-2"
href="/Catalog.html?id=4"> ТВ, консоли и аудио</a>
            </li>
            <li class="nav-item">
                <a class="nav-link p-2"
href="/Catalog.html?id=5"> ПК, ноутбуки, периферия</a>
            </li>
            <li class="nav-item">
                <a class="nav-link p-2"
href="/Catalog.html?id=6"> Комплектующие для ПК</a>
            </li>
            <li class="nav-item">
                <a class="nav-link p-2"
href="/Catalog.html?id=7"> Офис и мебель</a>
            </li>
            <li class="nav-item">
                <a class="nav-link p-2"
href="/Catalog.html?id=8"> Сетевое оборудование</a>
            </li>
            <li class="nav-item">
                <a class="nav-link p-2"
href="/Catalog.html?id=9"> Отдых и развлечения</a>
            </li>
            <li class="nav-item">
                <a class="nav-link p-2"
href="/Catalog.html?id=10"> Инструмент и стройка</a>
            </li>
            <li class="nav-item">
                <a class="nav-link p-2"
href="/Catalog.html?id=11"> Садовая техника</a>
            </li>
```

```

                <li class="nav-item">
                    <a class="nav-link p-2"
href="./Catalog.html?id=12"> Дом, декор и посуда</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link p-2"
href="./Catalog.html?id=13"> Автотовары</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link p-2"
href="./Catalog.html?id=14"> Аксессуары и услуги</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link p-2"
href="./Catalog.html?id=15"> Распродажа</a>
                </li>
            </div>
        </div>
    </article>
    <article class="col-lg-9">
        <div class="p-2">
            <div class="p-2"
style="background-color: white; border-radius: 12px;">
                <h2>Новости</h2>

                <section class="row">
                    <article class="col-lg-4">
                        <a class="text-
image" href="#">Скидка 10% на комплект КБТ</a>
                    </article>
                    <article class="col-lg-4">
                        <a class="text-
image" href="#">Новинка! Электрическая кофемолка Reoka RKCG-
P1</a>
                    </article>
                </section>
            </div>
        </div>
    </article>

```

```
        <a class="text-
image" href="#">Новинка! Мультистайлер Dyson Airwrap Styler
Complete Long Lite HS05</a>
```

```
    </article>
```

```
    <article class="col-lg-4">
```

```
        <a class="text-
image" href="#">Новинка! Ноутбук Xiaomi Book 14</a>
```

```
    </article>
```

```
  </section>
```

```
</div>
```

```
</div>
```

```
<div class="p-2">
```

```
  <div class="p-2"
```

```
style="background-color: white; border-radius: 12px;">
```

```
  <h2>Блоги</h2>
```

```
  <section class="row">
```

```
    <article class="col-lg-4">
```

```
        <a class="text-
image" href="#">Как выбрать осушитель воздуха</a>
```

```
    </article>
```

```
    <article class="col-lg-4">
```

```
        <a class="text-
image" href="#">Как настроить графику в Alan Wake 2</a>
```

```
    </article>
```

```
    <article class="col-lg-4">
```

```
        <a class="text-
image" href="#">Лучшие плагины для браузеров</a>
```

```
    </article>
```

```
    <article class="col-lg-4">
```

```
        <a class="text-
image" href="#">Самые ожидаемые игры - ноябрь 2023 года</a>
```

```
    </article>
```



```
        </section>
      </div>
    </div>
  </article>
</div>
</main>

<footer class="footer mt-auto d-flex flex-shrink-0
align-items-center">
  © 2023 Компания Техник. Администрация Сайта
не несет ответственности за размещаемые Пользователями
материалы (в т.ч. информацию и изображения), их
содержание и качество.
</footer>
</body>
</html>
```

Catalog.html

```
<html lang="ru">

  <head>
    <meta charset = "utf-8">
    <title> Техник </title>
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <script type="module"
src="./node_modules/bootstrap/dist/js/bootstrap.min.js"> </scrip
t>
    <link
href="./node_modules/bootstrap/dist/css/bootstrap.min.css"
rel="stylesheet"/>
    <link href="./node_modules/@fortawesome/fontawesome-
free/css/all.min.css" rel="stylesheet"/>
    <link rel = "stylesheet" href = "./style.css">
  </head>

  <body class="h-100 d-flex flex-column">

    <header>
```

```

        <nav class="navbar navbar-expand-md">
            <div class="container-fluid">
                <a class="navbar-brand" href = "./"
style="color: #6A3502;">
                    <i class="fa-solid fa-
microchip"></i>
                        Техник
                    </a>

                <form class="d-flex">
                    <input class="form-control"
type="search" placeholder="Поиск по сайту" aria-
label="Search">

                        <button class="btn btn-outline-*"
type="submit" style="color: #6A3502;"><i class="fa-solid fa-
search"></i></button>
                    </form>

                <a class="nav-link"
href="./Login.html">Вход</a>
            </div>
        </nav>
    </header>

```

```

        <main class="container-fluid" style="background-color:
antiquewhite;">
            <div class="row d-flex">
                <article class="col-lg-3">
                    <article class="nav-link" style="background-color:
white;">
                        <div class="d-flex flex-column">
                            <button class="navbar-toggler"
type="button" data-bs-toggle="collapse" data-bs-
target="#navbarAvailability"
                                aria-controls="navbarNav" aria-
expanded="false" aria-label="Toggle navigation">
                                    <h4>Наличие в магазине</h4>
                                </button>

```

```

        <div class="navbar-collapse collapse"
id="navbarAvailability">
            <div class="navbar-nav">
                <li class="nav-item">
                    <div class="form-check">
                        <input class="form-check-input"
id="checkbox1" name="checkbox1" type="checkbox" checked>
                        <label class="form-check-label"
for="checkbox1">В наличии</label>
                    </div>
                </li>
                <li class="nav-item">
                    <div class="form-check">
                        <input class="form-check-input"
id="checkbox2" name="checkbox2" type="checkbox" checked>
                        <label class="form-check-label"
for="checkbox2">Под заказ</label>
                    </div>
                </li>
            </div>
        </div>
</article>
<article class="nav-link" style="background-color:
white;">
        <div class="d-flex flex-column">
            <button class="navbar-toggler"
type="button" data-bs-toggle="collapse" data-bs-
target="#navbarScreenDg"
            aria-controls="navbarNav" aria-
expanded="false" aria-label="Toggle navigation">
                <h4>Диагональ экрана (дюйм)</h4>
            </button>
            <div class="navbar-collapse collapse"
id="navbarScreenDg">
                <div class="navbar-nav">
                    <li class="nav-item">
                        <div class="form-check">
                            <input class="form-check-input"
type="radio" name="radio-example" id="radio0" checked>

```

```

        <label class="form-check-label"
for="radio0">Все</label>
    </div>
</li>
<li>
    <div class="form-check me-md-3">
        <input class="form-check-input"
type="radio" name="radio-example" id="radio1">
        <label class="form-check-label"
for="radio1">Менее 13.9</label>
    </div>
</li>
<li>
    <div class="form-check">
        <input class="form-check-input"
type="radio" name="radio-example" id="radio2">
        <label class="form-check-label"
for="radio2">14 - 14.9</label>
    </div>
</li>
<li>
    <div class="form-check">
        <input class="form-check-input"
type="radio" name="radio-example" id="radio4">
        <label class="form-check-label"
for="radio4">16 и более</label>
    </div>
</li>
</div>
</div>
</article>
<article class="nav-link" style="background-color:
white;">
    <div class="d-flex flex-column">
        <button class="navbar-toggler"
type="button" data-bs-toggle="collapse" data-bs-
target="#navbarRAM"
        aria-controls="navbarNav" aria-
expanded="false" aria-label="Toggle navigation">

```

```

        <h4>Объем оперативной памяти
(ГБ)</h4>
    </button>
    <div class="navbar-collapse collapse"
id="navbarRAM">
        <div class="navbar-nav">
            <li class="nav-item">
                <div class="form-check">
                    <input class="form-check-input"
id="checkbox1" name="checkbox1" type="checkbox">
                    <label class="form-check-label"
for="checkbox1">4 ГБ</label>
                </div>
            </li>
            <li class="nav-item">
                <div class="form-check">
                    <input class="form-check-input"
id="checkbox2" name="checkbox2" type="checkbox">
                    <label class="form-check-label"
for="checkbox2">6 ГБ</label>
                </div>
            </li>
            <li class="nav-item">
                <div class="form-check">
                    <input class="form-check-input"
id="checkbox2" name="checkbox2" type="checkbox">
                    <label class="form-check-label"
for="checkbox2">8 ГБ</label>
                </div>
            </li>
            <li class="nav-item">
                <div class="form-check">
                    <input class="form-check-input"
id="checkbox2" name="checkbox2" type="checkbox">
                    <label class="form-check-label"
for="checkbox2">12 ГБ</label>
                </div>
            </li>
            <li class="nav-item">

```

```

        <div class="form-check">
            <input class="form-check-input"
id="checkbox2" name="checkbox2" type="checkbox">
            <label class="form-check-label"
for="checkbox2">16 ГБ</label>
        </div>
    </li>
    <li class="nav-item">
        <div class="form-check">
            <input class="form-check-input"
id="checkbox2" name="checkbox2" type="checkbox">
            <label class="form-check-label"
for="checkbox2">32 ГБ</label>
        </div>
    </li>
</div>
</div>
</article>
<article class="nav-link" style="background-color:
white;">
    <div class="d-flex flex-column">
        <button class="navbar-toggler"
type="button" data-bs-toggle="collapse" data-bs-
target="#navbarCreateCPU"
            aria-controls="navbarNav" aria-
expanded="false" aria-label="Toggle navigation">
            <h4>Производитель процессора</h4>
        </button>
        <div class="navbar-collapse collapse"
id="navbarCreateCPU">
            <div class="navbar-nav">
                <li class="nav-item">
                    <div class="form-check">
                        <input class="form-check-input"
id="checkbox1" name="checkbox1" type="checkbox">
                        <label class="form-check-label"
for="checkbox1">AMD</label>
                    </div>
                </li>

```

```
<li class="nav-item">
  <div class="form-check">
    <input class="form-check-input"
id="checkbox2" name="checkbox2" type="checkbox">
    <label class="form-check-label"
for="checkbox2">Apple</label>
  </div>
</li>
<li class="nav-item">
  <div class="form-check">
    <input class="form-check-input"
id="checkbox2" name="checkbox2" type="checkbox">
    <label class="form-check-label"
for="checkbox2">Intel</label>
  </div>
</li>
</div>
</div>
</article>
</article>

<div class="container col-lg-9">
  <div id="items-catalog">

    </div>
  </div>
</div>
</main>

<footer class="footer mt-auto d-flex flex-shrink-0
align-items-center">
  © 2023 Компания Техник. Администрация Сайта
не несет ответственности за размещаемые Пользователями
материалы (в т.ч. информацию и изображения), их
содержание и качество.
</footer>

<script type="module">
  import validation from "./js/validation";
```

```

import { linesPageForm } from "./js/lines"

document.addEventListener('DOMContentLoaded', () =>
{
    validation();
    const urlParams = new
URLSearchParams(location.search);
    linesPageForm(urlParams.get("id"));
});
</script>
</body>
</html>

```

Product.html

```

<html lang="ru">

  <head>
    <meta charset = "utf-8">
    <title> Техник </title>
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <script type="module"
src="./node_modules/bootstrap/dist/js/bootstrap.min.js"></scrip
t>
    <link
href="./node_modules/bootstrap/dist/css/bootstrap.min.css"
rel="stylesheet"/>
    <link href="./node_modules/@fortawesome/fontawesome-
free/css/all.min.css" rel="stylesheet"/>
    <link rel = "stylesheet" href = "./style.css">
  </head>

  <body class="h-100 d-flex flex-column">

    <header>
      <nav class="navbar navbar-expand-md">
        <div class="container-fluid">
          <a class = "navbar-brand" href = "./"
style="color: #6A3502;">

```



```

        <i class="fa-solid fa-
microchip"></i>
        Техник
    </a>

    <form class="d-flex">
        <input class="form-control"
type="search" placeholder="Поиск по сайту" aria-
label="Search">
        <button class="btn btn-outline-*"
type="submit" style="color: #6A3502;"><i class="fa-solid fa-
search"></i></button>
    </form>

    <a class="nav-link"
href="./Login.html">Вход</a>
    </div>
</nav>
</header>

    <main class="container-fluid" style="background-color:
antiquewhite;">
    <h4 id="model"></h4>
    <div class="d-flex row">
        <article class="d-flex flex-column p-2 col-lg-5"
style="background-color: white; border-radius: 12px;">
            <a class="image-container"></a>
        </article>
        <article class="d-flex flex-column p-2 col-lg-7">
            <div class="p-2">
                <div class="navbar-nav d-flex flex-lg-column"
style="background-color: white; border-radius: 12px;">
                    <h5 class="text-center" id="price"></h5>
                    <button class="btn btn-warning justify-content-end"
type="submit">Купить</button>
                </div>
            </div>
        </article>
    </div>

```

```
<div class="navbar-nav p-2">
  <div style="background-color: white; border-radius:
12px;">
    <h5 class="p-2">Описание</h5>
    <h5 class="nav-link p-2" style="font-size: larger;"
id="description"></h5>
  </div>
</div>
</article>
</div>
<div>
  <article class="d-flex flex-column t-2">
    <div class="navbar-nav p-2">
      <div style="background-color: white; border-radius:
12px;">
        <h5 class="p-2">Характеристика</h5>
        <a class="nav-link p-2" style="font-size: larger;"
id="specific"></a>
      </div>
    </div>
  </article>
</div>
<div>
  <article class="d-flex flex-column t-2">
    <div class="navbar-nav p-2">
      <div style="background-color: white; border-radius:
12px;">
        <h5 class="p-2">Самый популярный отзыв</h5>
        <a class="nav-link p-2" style="font-size: larger;">
<p>markoni89</p> <b>Срок использования:</b> Менее
месяца <p><b>Достоинства</b></p><p><b>отличный
процессор 12 поколения, хорош и мощностью и
энергопотреблением;</b></p>
        <p><b>Недостатки</b></p><p>сильно маркий
пластик корпуса (как и у всей линейки ноутбуков от MSI),
странно что об этом не подумали заранее и вдвойне странно
при такой удачной клавиатуре;</p>
        <p><b>Комментарий</b></p><p>На данный
момент это наилучший вариант для покупки по критерию
```

цена-качество. Железо крайне бодрое, в этой ценовой категории у других производителей нет адекватной конкуренции, даже следующий класс может уступать данному товарищу от MSI.

```
</div>
</div>
</article>
</div>
</main>
```

```
<footer class="footer mt-auto d-flex flex-shrink-0 align-items-center">
```

© 2023 Компания Техник. Администрация Сайта не несет ответственности за размещаемые Пользователями материалы (в т.ч. информацию и изображения), их содержание и качество.

```
</footer>
```

```
<script type="module">
import validation from "./js/validation";
import { productPage } from "./js/lines"

document.addEventListener('DOMContentLoaded', () => {
  validation();
  productPage();
});
</script>
</body>
</html>
```

Admin.html

```
<html lang="ru">
```

```
<head>
  <meta charset = "utf-8">
  <title> Техник </title>
  <meta name="viewport" content="width=device-width,
initial-scale=1">
```

```

    <script type="module"
src="./node_modules/bootstrap/dist/js/bootstrap.min.js"></scrip
t>
    <link
href="./node_modules/bootstrap/dist/css/bootstrap.min.css"
rel="stylesheet"/>
    <link href="./node_modules/@fortawesome/fontawesome-
free/css/all.min.css" rel="stylesheet"/>
    <link rel = "stylesheet" href = "./style.css">
</head>

<body class="h-100 d-flex flex-column">
    <header>
        <nav class="navbar navbar-expand-md">
            <div class="container-fluid">
                <a class = "navbar-brand" href = "./"
style="color: #6A3502;">
                    <i class="fa-solid fa-
microchip"></i>
                        Техник
                    </a>
                <form class="d-flex">
                    <input class="form-control"
type="search" placeholder="Поиск по сайту" aria-
label="Search">
                        <button class="btn btn-outline-*"
type="submit" style="color: #6A3502;"><i class="fa-solid fa-
search"></i></button>
                    </form>
            </div>
        </nav>
    </header>
    <main class="container-fluid p-2">
        <div class="btn-group" role="group">
            <button id="items-add" class="btn btn-
info">Добавить товар (диалог)</button>
        </div>

```

```
<div>
  <table id="items-table" class="table table-striped">
    <thead>
      <th scope="col">№</th>
      <th scope="col" class="w-25">Каталог</th>
      <th scope="col" class="w-25">Модель</th>
      <th scope="col" class="w-25">Цена</th>
      <th scope="col"
class="w-25">Количество</th>
      <th scope="col" class="w-25">Сумма</th>
      <th scope="col"></th>
      <th scope="col"></th>
    </thead>
    <tbody></tbody>
  </table>
</div>
</main>
<footer class="footer mt-auto d-flex flex-shrink-0 align-
items-center">
```

© 2023 Компания Техник. Администрация Сайта не несет ответственности за размещаемые Пользователями материалы (в т.ч. информацию и изображения), их содержание и качество.

```
</footer>
<div id="items-update" class="modal fade" tabindex="-1"
data-bs-backdrop="static" data-bs-keyboard="false">
  <div class="modal-dialog modal-dialog-scrollable">
    <form id="items-form" class="needs-validation"
novalidate>
      <div class="modal-content">
        <div class="modal-header">
          <h1 class="modal-title fs-5" id="items-
update-title"></h1>
          <button type="button" class="btn-close"
data-bs-dismiss="modal" aria-label="Close"></button>
        </div>
        <div class="modal-body">
          <div class="text-center">
```

```

        
    </div>
    <input id="items-line-id" type="number"
hidden>
        <div class="mb-2">
            <label for="item" class="form-
label">Каталог</label>
            <select id="item" class="form-select"
name="selected" required>
                </select>
            </div>
            <div class="mb-2">
                <label for="product" class="form-
label">Товары</label>
                <select id="product" class="form-select"
name="selected" required>
                    </select>
                </div>
                <div class="mb-2">
                    <label class="form-label"
for="model">Модель</label>
                    <input id="model" name="model"
class="form-control" type="text" required>
                    </div>
                    <div class="mb-2">
                        <label class="form-label"
for="image">Изображение</label>
                        <input id="image" type="file"
name="image" class="form-control" accept="image/*">
                        </div>
                        <div class="mb-2">
                            <label class="form-label"
for="price">Цена</label>
                            <input id="price" name="price"
class="form-control" type="number" value="0.00"
min="1000.00"

```

```

        step="0.50" required>
    </div>
    <div class="mb-2">
        <label class="form-label"
for="count">Количество</label>
        <input id="count" name="count"
class="form-control" type="number" value="0" min="1"
step="1"
            required>
    </div>
    <div class="mb-2">
        <label class="form-label"
for="description">Описание</label>
        <textarea id="description"
name="description" class="form-control" type="text"
rows="14"></textarea>
    </div>
    <div class="mb-2">
        <label class="form-label"
for="specific">Характеристика</label>
        <textarea id="specific" name="specific"
class="form-control" type="text" rows="3"></textarea>
    </div>
    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-
secondary" data-bs-dismiss="modal">Закрыть</button>
        <button type="submit" class="btn btn-
primary">Сохранить</button>
    </div>
</div>
</form>
</div>
</div>
<script type="module">
    import validation from "./js/validation";
    import { linesForm } from "./js/lines";

```

```

document.addEventListener('DOMContentLoaded', () =>
{
    validation();
    linesForm();
});
</script>
</body>

</html>

```

№	Каталог	Модель	Цена	Количество	Сумма
1	ПК, ноутбуки, периферия	15.6" Ноутбук MSI Modern 15 B12M-211RU черный	46999.00	239	11232761.00
2	Смартфоны и фототехника	realme C3	10000.00	560	5600000.00
3	ТВ, консоли и аудио	PlayStation 5	70000.00	3	210000.00
4	Смартфоны и фототехника	Honor 5A 5	1000.50	23	23011.50
5	Смартфоны и фототехника	Honor 8A	1000.00	2	2000.00
6	ПК, ноутбуки, периферия	17.3" Ноутбук ARDOR GAMING NEO G17-I7ND307 черный	104999.00	20	2099980.00
7	ПК, ноутбуки, периферия	17.3" Ноутбук ASUS TUF Gaming F17 FX706HF-HX014 черный	77999.00	30	2339970.00
8	ПК, ноутбуки, периферия	17.3" Ноутбук MSI GF76 Katana 17 B12VFK-271XRU черный	111999.00	10	1119990.00
9	ПК, ноутбуки, периферия	15.6" Ноутбук GIGABYTE G5 MF черный	79999.00	16	1279984.00

Наличие в магазине
 Диагональ экрана (дюйм)
 Объем оперативной памяти (ГБ)
 Производитель процессора


	15.6" Ноутбук MSI Modern 15 B12M-211RU черный Full HD (1920x1080), IPS, Intel Core i5-12500H, ядра: 4 + 8 x 2.5 ГГц, RAM 16 Гб, SSD 512 Гб, GeForce RTX 4050 для ноутбуков 6 Гб, без ОС	46999.00	Купить
	17.3" Ноутбук ARDOR GAMING NEO G17-I7ND307 черный Full HD (1920x1080), IPS, Intel Core i7-12650H, ядра: 6 + 4 x 2.3 ГГц, RAM 16 Гб, SSD 512 Гб, GeForce RTX 4060 для ноутбуков 8 Гб, без ОС	104999.00	Купить
	17.3" Ноутбук ASUS TUF Gaming F17 FX706HF-HX014 черный Full HD (1920x1080), IPS, Intel Core i5-11400H, ядра: 6 x 2.7 ГГц, RAM 16 Гб, SSD 512 Гб, GeForce RTX 2050 4 Гб, без ОС	77999.00	Купить
	17.3" Ноутбук MSI GF76 Katana 17 B12VFK-271XRU черный Full HD (1920x1080), IPS, Intel Core i7-12650H, ядра: 6 + 4 x 2.3 ГГц, RAM 16 Гб, SSD 512 Гб, GeForce RTX 4060 для ноутбуков 8 Гб, без ОС	111999.00	Купить
	15.6" Ноутбук GIGABYTE G5 MF черный Full HD (1920x1080), IPS, Intel Core i5-12500H, ядра: 4 + 8 x 2.5 ГГц, RAM 16 Гб, SSD 512 Гб, GeForce RTX 4050 для ноутбуков 6 Гб, без ОС	79999.00	Купить

Техник - Chromium

Техник

Поиск по сайту

15.6" Ноутбук MSI Modern 15 B12M-211RU черный



46999.00

купить

Описание

Тонкий корпус ультрабука MSI Modern 15 B12M-215XRU сочетает возможности для оперативного решения сложных задач и оттачивания профессиональных навыков. В качестве ведущего звена производитель выбрал процессор Intel Core i3-1215U с 6-ядерной архитектурой и графический ускоритель Intel UHD Graphics. Это аппаратное оснащение не допускает задержек при обработке файлов и загрузке программ. Ультрабук MSI Modern 15 B12M-215XRU незаменим для досуга за любимыми развлечениями. Экран диагональю 15.6 дюйма наполняет картинкой широким спектром оттенков и не допускает задержек ввода. Матовое покрытие нейтрализует блики от солнечных лучей. Гарантом качественной передачи изображений и звука при проведении совещаний станет веб-камера разрешением 1 Мп и встроенный микрофон.

Характеристика

Full HD (1920x1080), IPS, Intel Core i5-12500H, ядра: 4 + 8 x 2.5 ГГц, RAM 16 Гб, SSD 512 Гб, GeForce RTX 4050 для ноутбуков 6 Гб, без ОС

Самый популярный отзыв

markoni89

Срок использования: Менее месяца

Достоинства

отличный процессор 12 поколения, хорош и мощностью и энергопотреблением;

Недостатки

сильно маркий пластик корпуса (как и у всей линейки ноутов от MSI), странно что об этом не подумали заранее и вдвойне странно при такой удачной клавиатуре;

Комментарий

На данный момент это наилучший вариант для покупки по критерию цена-качество. Железо крайне бодрое, в этой ценовой категории у других производителей нет адекватной конкуренции, даже следующий класс может уступать данному товарищу от MSI.

© 2023 Компания Техник. Администрация Сайта не несет ответственности за размещаемые Пользователями материалы (в т.ч. информацию и изображения), их содержание и качество.

index.html - Lab3 - Visual S... Техник - Chromium ИНТЕРНЕТ.docx - LibreOff... Отчет по третьей лабора...

12:06