

Лабораторная работа № 18

Тема: Сборка и публикация проекта.

Цель работы: Подготовить «сломанный» проект к публикации, исправить ошибки оптимизации и успешно развернуть его на GitHub Pages.

Задание (для всех вариантов):

Этап 1: Аудит и исправление.

Пройдите по чек-листу и исправьте ошибки в исходном коде:

- Пути: Сделайте все пути относительными.
- Регистр: Приведите названия всех файлов и ссылок на них к нижнему регистру.
- Гигиена: Удалите отладочную информацию из консоли и «мертвый» код.
- SEO: Заполните теги <title> и <meta name="description">.
- Оптимизация: Уменьшите вес главной картинки (используйте онлайн-сервисы типа Squoosh или TinyPNG).

Этап 2: Публикация:

- Создайте публичный репозиторий на GitHub.
- Задепloyте проект на GitHub Pages.
- Проверьте работоспособность сайта по полученной ссылке.

Вариант 1. Портфолио.

HTML:

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Портфолио</title>
    <link rel="stylesheet"
    href="/C:/Users/Admin/Desktop/Portfolio/style.css">
    <link rel="icon" href="C:/Users/Admin/Documents/icons/favicon.ico"
    type="image/x-icon">
</head>
<body>
    <header>
```

```
<h1>Привет, я будущий фронтенд-разработчик!</h1>
</header>
<main>
  <section class="profile">
    

    <p>Я изучаю HTML, CSS и JavaScript. Этот сайт — моя первая
    работа.</p>
  </section>

  <section class="skills">
    <h2>Мои навыки</h2>
    <ul id="skills-list"></ul>
  </section>
</main>

<script src="/js/MainScript.js"></script>
</body>
</html>
```

CSS:

```
body {
  font-family: Arial, sans-serif;
  line-height: 1.6;
  text-align: center;
  background-color: #f4f4f4;
  padding: 20px;
}
```

```
header {
  background: #333;
  color: #fff;
  padding: 1rem;
  border-radius: 10px;
}
```

```
#main-photo {
  width: 300px;
  height: 300px;
  border-radius: 50%;
```

```
border: 5px solid #333;
margin-top: 20px;
}

.skills ul {
    list-style: none;
    padding: 0;
}

.skills li {
    background: #fff;
    margin: 5px;
    padding: 10px;
    display: inline-block;
    border-radius: 5px;
    box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}
```

JS:

```
// =====
=
// СТАРЫЙ КОД (НЕ УДАЛЯТЬ, ВДРУГ ПРИГОДИТСЯ!!!!)
// function test() {
//     var a = "привет";
//     console.log(a);
//     alert("Зачем я это написал?");
// }
//
// =====
=
/***
 * Используем событие DOMContentLoaded, чтобы гарантировать,
 * что все HTML-теги уже превратились в DOM-узлы.
 */
document.addEventListener('DOMContentLoaded', () => {
    console.log("СТАРТ ОТЛАДКИ");

    // Данные вынесены в константу (Логика)
    const skills = ["HTML", "CSS", "JavaScript", "GitHub", "LocalStorage"];
```

```

// Поиск элемента в DOM
const list = document.getElementById('skills-list');
console.log("Переменная list сейчас:", list);
// Проверка на наличие элемента (защита от ошибок)
if (list) {
    renderSkills.skills(list);
}
console.log("КОНЕЦ ОТЛАДКИ");
});

/**
 * Функция отрисовки (UI)
 * Разделяем логику данных и логику отображения
 */
function renderSkills(data, container) {
    // Очищаем контейнер перед отрисовкой (хорошая привычка)
    container.innerHTML = "";

    data.forEach(skill => {
        const li = document.createElement('li');
        li.textContent = skill; // Используем textContent вместо innerText для
        безопасности
        container.appendChild(li);
        console.log("Итерация цикла для элемента: " + skill);
    });
}

```

Вариант 2: Карточка эксклюзивного товара.

HTML:

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Новый товар</title>
    <link rel="icon" href="C:/Users/User/Downloads/icons/favicon.ico">
    <link rel="stylesheet" href="/C:/Work/Project2/css/Styles.css">
</head>
<body>
    <header class="main-header">
        <div class="logo">Fast Step</div>

```

```
<nav>Магазин спортивной обуви</nav>
</header>

<div class="product-card">
    

    <h1 id="title">Загрузка названия...</h1>
    <p class="description">Эксклюзивная модель из коллекции 2026
года.</p>

    <div class="price-tag">
        Цена: <span id="price">0</span>
    </div>

    <button id="buy-btn">Купить сейчас</button>
</div>

<script src="/JS/Logic.js"></script>
</body>
</html>
```

CSS:

```
:root {
    --accent: #fbbf24;
    --dark: #111827;
    --text: #374151;
}

body {
    font: 1rem/1.5 'Inter', sans-serif;
    background: #f3f4f6;
    color: var(--text);
    margin: 0;
    display: flex;
    flex-direction: column;
    align-items: center;
}

header {
```

```
background: var(--dark);
color: #fff;
width: 100%;
padding: 1.5rem 0;
text-align: center;
box-shadow: 0 4px 15px rgba(0,0,0,0.2);
}

header span { color: var(--accent); font-weight: 800; }

img {
  width: 50vw;
  height: 400px;
  object-fit: cover;
  display: block;
  margin: 2rem auto;
  border-radius: 12px;
  box-shadow: 0 10px 20px rgba(0,0,0,0.1);
}

.product-card {
  background: #fff;
  padding: 2rem;
  border-radius: 16px;
  text-align: center;
  max-width: 400px;
  box-shadow: 0 4px 6px -1px rgba(0,0,0,0.1);
}

.price-tag {
  font-size: 1.5rem;
  font-weight: 700;
  color: #e11d48;
  margin: 1rem 0;
}

button {
  background: var(--dark);
  color: #fff;
  border: none;
  padding: 12px 24px;
  border-radius: 8px;
```

```
width: 100%;  
cursor: pointer;  
font-weight: 600;  
transition: 0.3s;  
}  
  
button:hover { opacity: 0.9; transform: scale(1.02); }
```

JS:

```
var oldData = null;  
var version = "1.0.0.build.99";  
  
/* function deleteEverything() {  
    document.body.innerHTML = "";  
    console.log("Удалено!");  
}  
*/  
  
// Точка входа: ждем загрузки дерева тегов  
document.addEventListener('DOMContentLoaded', () => {  
    console.log("--- DEBUG START ---");  
    // Исходные данные (Логика)  
    const product = {  
        name: "Кроссовки Noname Air Max (White Edition)",  
        price: 12990,  
        currency: "〒."  
    };  
  
    // Поиск элементов  
    const titleNode = document.getElementById('title');  
    console.log("Элемент заголовка:", titleNode);  
    const priceNode = document.getElementById('price');  
    console.log("Элемент цены:", priceNode);  
    const buyBtn = document.getElementById('buy-btn');  
  
    // Если всё на месте — рисуем (UI)  
    if (titleNode && priceNode) {  
        renderProduct(product, titleNode, priceNode);  
    }  
  
    // Обработка событий  
    if (buyBtn) {
```

```
buyBtn.addEventListener('click', () => {
    alert(`Товар ${product.name} добавлен в корзину!`);
});
}
console.log("--- DEBUG END ---");
});

// Функция отрисовки: отвечает только за отображение данных в UI
function renderProduct(data, titleElement, priceElement) {
    titleElement.textContent = data.name;
    priceElement.textContent = `${data.price.toLocaleString()}
${data.currency}`;
}
```

 **Отчет должен содержать (см. образец):**

- номер и тему лабораторной работы;
- фамилию, номер группы студента и вариант задания;
- скриншоты окна VSC с исходным кодом программ;
- скриншоты с результатами выполнения программ;
- пояснения, если необходимо;
- выводы.

Отчеты в формате **pdf** отправлять на email:

colledge20education23@gmail.com