

Создание современного веб-портфолио для студентов: разработка интерфейса и логики на HTML5, CSS и JavaScript

Екимова Яна Сергеевна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В статье рассматривается методика создания современного и стильного веб-портфолио для студентов, где они смогут представлять свои научные работы. Разработан интерфейс с использованием HTML5, стилизованный с помощью CSS, и реализована функциональность для демонстрации научных работ, включая раздел о студенте, каталог работ и возможность их добавления. Материал предназначен для преподавателей и студентов, заинтересованных в применении современных технологий в обучении и стремящихся сделать процесс образовательного взаимодействия более эффективным и увлекательным.

Ключевые слова: веб-портфолио, HTML5, CSS, JavaScript, адаптивный дизайн.

Creating a modern web portfolio for students: developing an interface and logic in HTML5, CSS and JavaScript

Ekimova Yana Sergeevna

Sholom Aleichem Priamurskiy State University

Student

Abstract

The article discusses a technique for creating a modern and stylish web portfolio for students, where they can present their scientific work. An HTML5 interface has been developed, stylized using CSS, and functionality has been implemented to demonstrate scientific papers, including a section about the student, a catalog of papers, and the ability to add them. The material is intended for teachers and students interested in using modern technologies in teaching and striving to make the process of educational interaction more effective and exciting.

Keywords: web portfolio, HTML5, CSS, JavaScript, responsive design.

1 Введение

Актуальность

Современное образование и профессиональное развитие требуют эффективных инструментов для презентации достижений и научных работ. Цифровые технологии открывают новые возможности для создания структурированных и визуально привлекательных портфолио, которые не

только демонстрируют навыки студентов, но и упрощают образовательное взаимодействие. Веб-портфолио, разработанные с использованием HTML5, CSS и JavaScript, становятся ключевым элементом академического и профессионального роста, позволяя динамично представлять проекты, адаптировать контент под разные устройства и обеспечивать удобное управление материалами.

В данной статье рассматривается методика создания современного веб-портфолио для студентов: от проектирования интерфейса до реализации функциональности. Предложенное решение включает персонализированный раздел о студенте, каталог научных и учебных работ, а также механизм добавления новых материалов. Такой подход особенно актуален для преподавателей и учащихся, заинтересованных в интеграции цифровых инструментов в образовательный процесс, что способствует систематизации знаний, повышению мотивации и развитию технических навыков. Использование чистого фронтенда (HTML5, CSS3, JavaScript) гарантирует совместимость, гибкость и простоту дальнейшего расширения функционала.

1.2 Обзор исследований

В статье рассматривается организация подготовки педагогических работников к использованию технологий HTML5 для разработки электронных образовательных ресурсов в формате веб-приложений. Отмечена актуальность использования мультимедийных интерактивных электронных образовательных ресурсов в современной школе и обоснована эффективность разработки подобных электронных образовательных ресурсов именно в формате офлайновых или онлайновых веб-приложений. [1]. Показала процесс создания и использования мультимедийных интерактивных плакатов по информатике К.А. Цилинская. Представила опыт разработки мультимедийных интерактивных плакатов средствами языка HTML5 и Интернет-сервисов [2]. Королюк Я. А., Куликова Н. Ю. показали опыт разработки и использования интерактивных образовательных ресурсов на основе html5 для онлайн-обучения школьников информатике [3]. В работе рассмотрели возможности языка разметки гипертекста HTML5 для создания макета электронного учебника Д.М. Размологов, А.С. Панина, П.В. Захаров. Привели разработанный макет, а также часть таблицы стилей для дизайна макета [4]. И.С. Шегаев, А.Г. Полозов разработали новейший политологический глоссарий: основные категории, понятия, термины [5].

1.3 Цель исследования

Целью исследования является разработка современного веб-портфолио для студентов с использованием технологий HTML5, CSS3 и JavaScript.

2 Материалы и методы

Для создания приложения используется язык программирования HTML5. В качестве среды разработки используется текстовым редактор Notepad++, оптимизированным для языков программирования.

3 Результаты

Для начала создадим папку с названием "Портфолио". Открываем Notepad++ создаём файл index.html в программе Notepad++: открываем редактор, создаем новый документ (Ctrl+N или через меню "Файл" → "Новый"), сохраняем его (Ctrl+S) в папке "Портфолио", указывая полное имя "index.html" с расширением .html и выбирая тип файла "Hyper Text Markup Language file" для корректного отображения в браузере. Затем формируем основную структуру HTML-документа, которая создает полнофункциональное веб-портфолио: код начинается с обязательного объявления `<!DOCTYPE html>` и корневого элемента `<html lang="ru">`, где в секции `<head>` задаются важные метаданные (кодировка UTF-8 для поддержки кириллицы, адаптивный viewport для правильного отображения на мобильных устройствах), подключаются CSS-стили (включая шрифт Roboto через Google Fonts) и устанавливается заголовок страницы. В теле документа (`<body>`) реализованы: шапка с навигационным меню по разделам ("Обо мне", "Мои работы", "Добавить работу"), основное содержимое с тремя ключевыми секциями (раздел "Обо мне" с возможностью загрузки фото профиля и редактирования текстовой информации через форму, раздел "Мои работы" для отображения загруженных материалов, и раздел "Добавить работу" с интерактивной формой для загрузки нескольких PDF-файлов, поддерживающей динамическое добавление/удаление полей ввода), а также подвал с копирайтом. В конце документа подключается JavaScript-файл, обеспечивающий интерактивность всех элементов. Такая структура сочетает семантическую верстку для улучшения доступности и SEO, адаптивный дизайн для всех устройств и продуманную логику для удобного управления контентом, при этом все связанные файлы (HTML, CSS, JS) рекомендуется хранить в одной папке "Портфолио" для удобства организации проекта.

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Портфолио студента</title>
  <link rel="stylesheet" href="styles.css">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap"
rel="stylesheet">
</head>
<body>
```

```
<header>
  <h1>Мое портфолио</h1>
  <nav>
    <ul>
      <li><a href="#about">Обо мне</a></li>
      <li><a href="#research">Мои работы</a></li>
      <li><a href="#upload">Добавить работу</a></li>
    </ul>
  </nav>
</header>

<main>
  <section id="about">
    <div class="section-header">
      <h2>Обо мне</h2>
      <button id="editAboutBtn">Редактировать</button>
    </div>
    <div id="aboutDisplay">
      
      <div id="aboutText">
        <p>Здесь будет информация обо мне</p>
      </div>
    </div>
    <div id="aboutEdit" style="display: none;">
      <form id="aboutForm">
        <div class="form-group">
          <label for="photoUpload">Фото профиля:</label>
          <input type="file" id="photoUpload" accept="image/*">
        </div>
        <div class="form-group">
          <label for="aboutInfo">Информация обо мне:</label>
          <textarea id="aboutInfo" rows="6" placeholder="Расскажите о себе,
своих навыках и достижениях"></textarea>
        </div>
        <div class="form-actions">
          <button type="submit">Сохранить</button>
          <button type="button" id="cancelEditBtn">Отмена</button>
        </div>
      </form>
    </div>
  </section>

  <section id="research">
    <h2>Мои научные работы</h2>
    <div id="works">
      <p class="empty-message">У вас пока нет загруженных работ</p>
    </div>
  </section>

  <section id="upload">
    <h2>Добавить работы</h2>
    <form id="uploadForm">
```

```

<div id="fileInputsContainer">
  <div class="file-input-group">
    <input type="text" class="title" placeholder="Название работы"
required>
    <input type="file" class="file" accept=".pdf" required>
    <button type="button" class="remove-btn">×</button>
  </div>
</div>
<div class="form-actions">
  <button type="button" id="addMoreBtn">+ Добавить еще</button>
  <button type="submit">Загрузить все</button>
</div>
</form>
</section>
</main>

<footer>
  <p>© 2025 Мое портфолио!</p>
</footer>

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

```

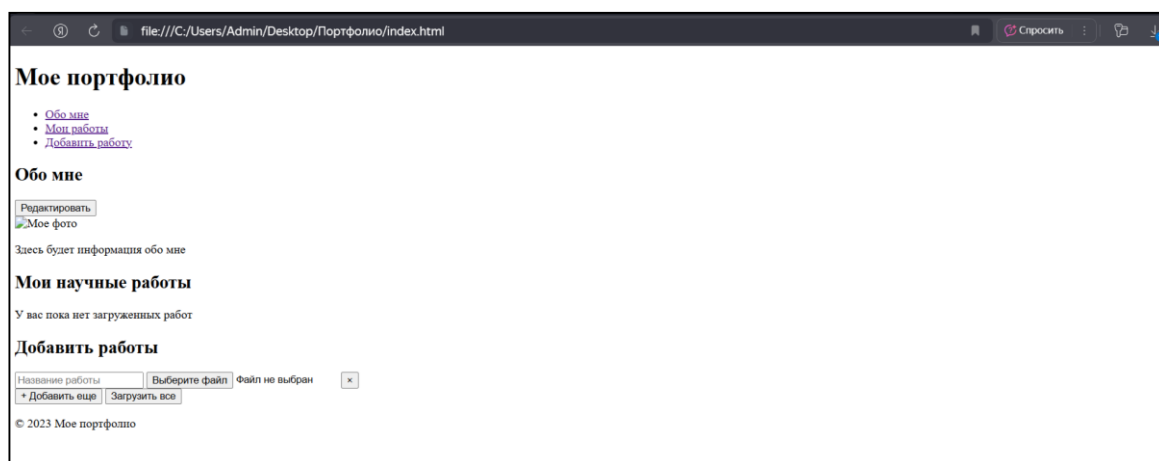


Рисунок 1. Основная структура HTML-документа

Далее создадим файл "styles.css" с CSS-стилями, который используется для оформления веб-страницы. Данный CSS-код отвечает за визуальное оформление и адаптивность веб-портфолио студента, задавая стили для всех элементов страницы. В начале сбрасываются стандартные отступы (*) и устанавливается box-sizing: border-box для корректного расчета размеров элементов. Для body задается светлый оранжевый фон (#fff3e0), шрифт Roboto, цвет текста и межстрочный интервал. Шапка (header) оформлена ярко-оранжевым цветом (#ff9800) с белым текстом и тенью, а навигационное меню (nav) стилизовано как горизонтальный список с плавными hover-эффектами. Основное содержимое (main) имеет ограниченную ширину и центрируется, а секции (section) выглядят как белые карточки с оранжевой верхней границей

и тенями. Особое внимание уделено разделу "Обо мне" - фото профиля имеет круглую форму с оранжевой рамкой, а форма редактирования включает стилизованные поля ввода с эффектами фокуса. Для раздела с работами предусмотрены: карточки работ (work-item) с оранжевым акцентом, кнопки просмотра/удаления серого цвета, а также адаптивное отображение при отсутствии работ. Форма загрузки поддерживает добавление нескольких файлов - группы полей (file-input-group) имеют оранжевую границу, а кнопки "Добавить еще" и "Удалить" интуитивно понятны. В подвале (footer) повторяется оранжевый цвет шапки. Медиазапросы (@media) обеспечивают адаптивность: на мобильных устройствах меняется расположение элементов (фото и текста в "Обо мне", навигационного меню, полей формы). Все переходы (transition) плавные, а hover-эффекты (изменение цвета, смещение) улучшают пользовательский опыт.

```
* {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
}

body {
  font-family: 'Roboto', sans-serif;
  background-color: #fff3e0;
  color: #333;
  line-height: 1.6;
}

header {
  background: #ff9800;
  color: white;
  padding: 20px;
  text-align: center;
  box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}

nav ul {
  list-style: none;
  padding: 0;
  display: flex;
  justify-content: center;
  margin-top: 15px;
}

nav ul li {
  margin: 0 15px;
}

nav a {
  color: white;
  text-decoration: none;
```

```
    font-weight: 700;
    transition: all 0.3s;
    padding: 5px 10px;
    border-radius: 4px;
  }

  nav a:hover {
    background-color: #ffb74d;
  }

  main {
    padding: 20px;
    max-width: 1200px;
    margin: 0 auto;
  }

  section {
    padding: 25px;
    margin: 20px 0;
    background: white;
    border-radius: 8px;
    box-shadow: 0 2px 15px rgba(255, 152, 0, 0.1);
    border-top: 3px solid #ff9800;
  }

  .section-header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin-bottom: 20px;
    padding-bottom: 10px;
    border-bottom: 1px solid #eee;
  }

  #aboutDisplay {
    display: flex;
    gap: 30px;
    align-items: flex-start;
  }

  #profilePhoto {
    width: 200px;
    height: 200px;
    object-fit: cover;
    border-radius: 50%;
    border: 5px solid #ff9800;
    box-shadow: 0 3px 10px rgba(255, 152, 0, 0.2);
  }

  .form-group {
    margin-bottom: 20px;
  }
```

```
label {
  display: block;
  margin-bottom: 8px;
  font-weight: bold;
  color: #555;
}

input[type="text"],
input[type="file"],
textarea {
  width: 100%;
  padding: 12px;
  border: 1px solid #ddd;
  border-radius: 6px;
  font-family: inherit;
  font-size: 16px;
  transition: border 0.3s;
}

input[type="text"]:focus,
input[type="file"]:focus,
textarea:focus {
  border-color: #ff9800;
  outline: none;
  box-shadow: 0 0 0 3px rgba(255, 152, 0, 0.2);
}

textarea {
  resize: vertical;
  min-height: 150px;
}

button, .view-btn, .delete-btn {
  background-color: #9e9e9e;
  color: white;
  border: none;
  padding: 12px 20px;
  border-radius: 6px;
  cursor: pointer;
  font-weight: bold;
  font-size: 16px;
  transition: all 0.3s;
  box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}

button:hover, .view-btn:hover, .delete-btn:hover {
  background-color: #757575;
  transform: translateY(-2px);
  box-shadow: 0 4px 8px rgba(0,0,0,0.15);
}
```

```
.form-actions {
  display: flex;
  gap: 15px;
  margin-top: 20px;
}

#addMoreBtn {
  margin-right: 15px;
}

.file-input-group {
  margin-bottom: 20px;
  display: flex;
  align-items: center;
  gap: 15px;
  padding: 15px;
  background: #f9f9f9;
  border-radius: 6px;
  border-left: 4px solid #ff9800;
}

.remove-btn {
  background: #9e9e9e;
  color: white;
  border: none;
  width: 32px;
  height: 32px;
  border-radius: 50%;
  cursor: pointer;
  font-size: 18px;
  display: flex;
  align-items: center;
  justify-content: center;
  transition: all 0.3s;
}

.remove-btn:hover {
  background: #757575;
}

.work-item {
  background: #f9f9f9;
  padding: 20px;
  margin-bottom: 15px;
  border-radius: 6px;
  border-left: 4px solid #ff9800;
  display: flex;
  justify-content: space-between;
  align-items: center;
  transition: transform 0.3s;
}
```

```
.work-item:hover {
  transform: translateX(5px);
}

.work-info h3 {
  color: #ff9800;
  margin-bottom: 5px;
}

.work-actions {
  display: flex;
  gap: 10px;
}

.view-btn, .delete-btn {
  padding: 8px 15px;
  font-size: 14px;
}

.empty-message {
  color: #9e9e9e;
  font-style: italic;
  text-align: center;
  padding: 30px;
  font-size: 18px;
}

footer {
  text-align: center;
  padding: 20px 0;
  background: #ff9800;
  color: white;
  margin-top: 40px;
  font-size: 14px;
}

@media (max-width: 768px) {
  #aboutDisplay {
    flex-direction: column;
    align-items: center;
  }

  nav ul {
    flex-direction: column;
    gap: 10px;
  }

  .file-input-group {
    flex-direction: column;
    align-items: flex-start;
  }
}
```

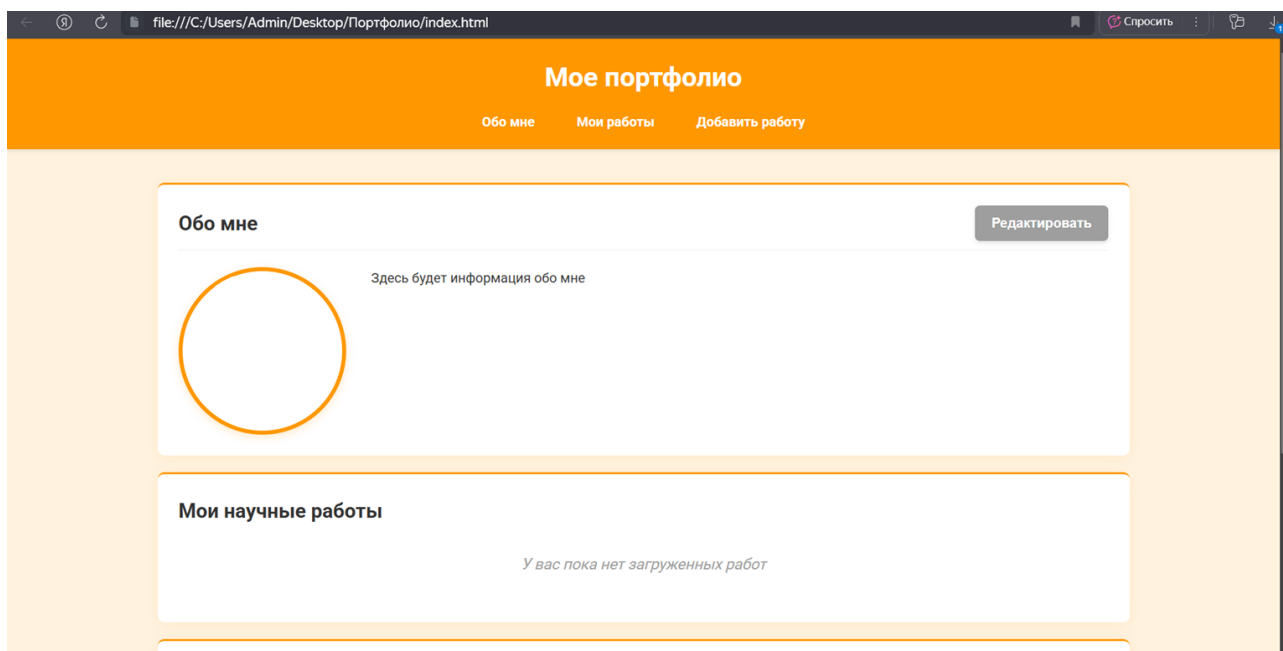


Рисунок 2. Добавление CSS-стилей

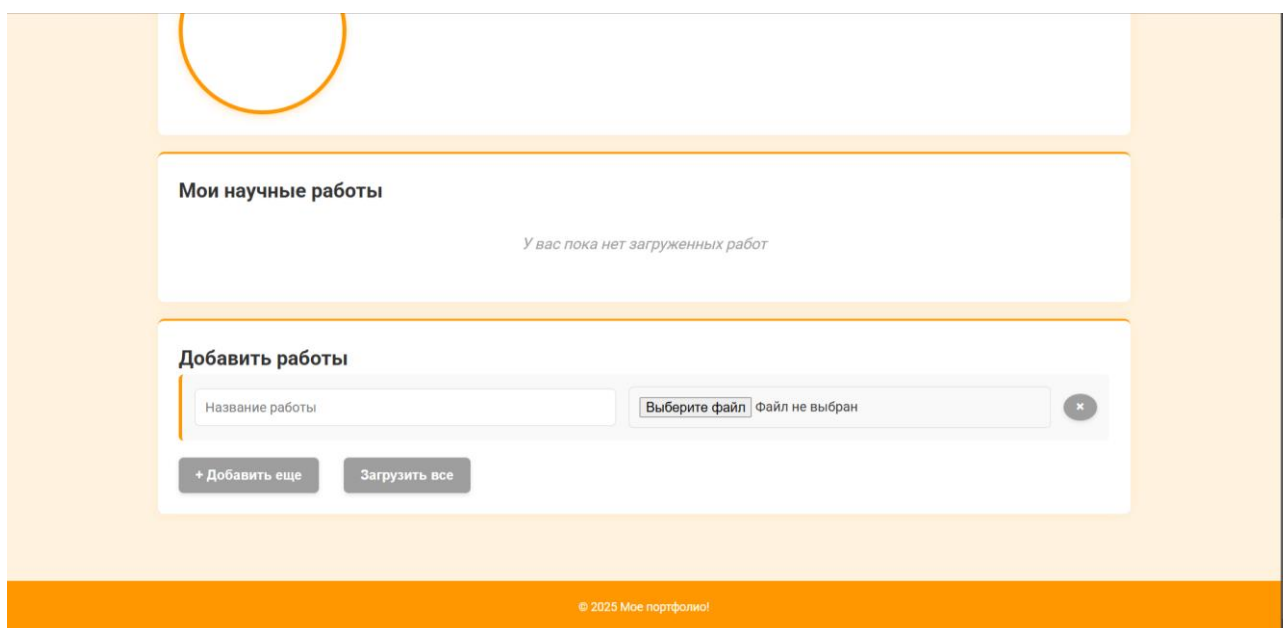


Рисунок 3. Добавление CSS-стилей

Теперь создадим файл "script.js", данный JavaScript-код реализует всю интерактивную логику веб-портфолио студента, начиная с обработки событий после полной загрузки DOM-структуры. Он обеспечивает:

- функционал раздела "Обо мне" - переключение между режимами просмотра и редактирования, сохранение текстовой информации и фотографии профиля в localStorage с последующей загрузкой при обновлении страницы;
- динамическое управление формой загрузки работ - добавление/удаление полей для нескольких файлов через кнопки "Добавить"

еще" и "×", валидацию и обработку отправки формы с преобразованием PDF-файлов в base64 и сохранением в localStorage;

- отображение списка работ с возможностью просмотра (в новом окне через embed) и удаления отдельных элементов;
- систему уведомлений пользователя через alert-сообщения.

Код использует современные Web API (FileReader для работы с файлами, localStorage для хранения данных) и реализует паттерн событийно-ориентированного программирования через addEventListener. Все данные сохраняются локально в браузере, что позволяет работать с портфолио без серверной части, а асинхронная обработка файлов обеспечивает плавность взаимодействия.

```
document.addEventListener('DOMContentLoaded', function() {  
  // Элементы раздела "Обо мне"  
  const editAboutBtn = document.getElementById('editAboutBtn');  
  const aboutDisplay = document.getElementById('aboutDisplay');  
  const aboutEdit = document.getElementById('aboutEdit');  
  const aboutForm = document.getElementById('aboutForm');  
  const cancelEditBtn = document.getElementById('cancelEditBtn');  
  const photoUpload = document.getElementById('photoUpload');  
  const aboutInfo = document.getElementById('aboutInfo');  
  const profilePhoto = document.getElementById('profilePhoto');  
  const aboutText = document.getElementById('aboutText');  
  
  // Элементы загрузки работ  
  const uploadForm = document.getElementById('uploadForm');  
  const fileInputsContainer = document.getElementById('fileInputsContainer');  
  const addMoreBtn = document.getElementById('addMoreBtn');  
  const worksSection = document.getElementById('works');  
  
  // Редактирование информации "Обо мне"  
  editAboutBtn.addEventListener('click', function() {  
    aboutDisplay.style.display = 'none';  
    aboutEdit.style.display = 'block';  
    aboutInfo.value = aboutText.innerHTML;  
  });  
  
  cancelEditBtn.addEventListener('click', function() {  
    aboutDisplay.style.display = 'flex';  
    aboutEdit.style.display = 'none';  
  });  
  
  aboutForm.addEventListener('submit', function(e) {  
    e.preventDefault();  
  
    aboutText.innerHTML = aboutInfo.value;  
  
    if (photoUpload.files[0]) {  
      const reader = new FileReader();  
      reader.onload = function(e) {
```

```
        profilePhoto.src = e.target.result;
        localStorage.setItem('profilePhoto', e.target.result);
    };
    reader.readAsDataURL(photoUpload.files[0]);
}

localStorage.setItem('aboutInfo', aboutInfo.value);
aboutDisplay.style.display = 'flex';
aboutEdit.style.display = 'none';
});

// Загрузка сохраненных данных
function loadSavedData() {
    if (localStorage.getItem('profilePhoto')) {
        profilePhoto.src = localStorage.getItem('profilePhoto');
    }

    if (localStorage.getItem('aboutInfo')) {
        aboutText.innerHTML = localStorage.getItem('aboutInfo');
    }
}

// Добавление новых полей для загрузки
addMoreBtn.addEventListener('click', function() {
    const newGroup = document.createElement('div');
    newGroup.className = 'file-input-group';
    newGroup.innerHTML = `
        <input type="text" class="title" placeholder="Название работы" required>
        <input type="file" class="file" accept=".pdf" required>
        <button type="button" class="remove-btn">×</button>
    `;
    fileInputsContainer.appendChild(newGroup);
});

// Удаление группы полей
document.addEventListener('click', function(e) {
    if (e.target.classList.contains('remove-btn')) {
        const groups = document.querySelectorAll('.file-input-group');
        if (groups.length > 1) {
            e.target.closest('.file-input-group').remove();
        } else {
            alert('Должна остаться хотя бы одна группа для загрузки');
        }
    }
});

// Обработка отправки формы
uploadForm.addEventListener('submit', function(e) {
    e.preventDefault();

    const groups = document.querySelectorAll('.file-input-group');
    let works = JSON.parse(localStorage.getItem('works')) || [];
```

```
let filesProcessed = 0;

groups.forEach(group => {
  const title = group.querySelector('.title').value;
  const file = group.querySelector('.file').files[0];

  if (title && file) {
    const reader = new FileReader();
    reader.onload = function(e) {
      const work = {
        id: Date.now() + Math.random(),
        title: title,
        fileName: file.name,
        fileData: e.target.result.split(',')[1]
      };

      works.push(work);
      filesProcessed++;

      if (filesProcessed === groups.length) {
        localStorage.setItem('works', JSON.stringify(works));
        displayWorks();
        uploadForm.reset();

        // Оставляем одну группу после отправки
        while (fileInputsContainer.children.length > 1) {
          fileInputsContainer.removeChild(fileInputsContainer.lastChild);
        }

        alert('Работы успешно добавлены!');
      }
    };
    reader.readAsDataURL(file);
  }
});

// Отображение работ
function displayWorks() {
  const works = JSON.parse(localStorage.getItem('works')) || [];

  if (works.length === 0) {
    worksSection.innerHTML = '<p class="empty-message">У вас пока нет загруженных работ</p>';
    return;
  }

  worksSection.innerHTML = "";

  works.forEach(work => {
    const workItem = document.createElement('div');
    workItem.className = 'work-item';
```

```
workItem.innerHTML = `  
  <div class="work-info">  
    <h3>${work.title}</h3>  
    <p>Файл: ${work.fileName}</p>  
  </div>  
  <div class="work-actions">  
    <button class="view-btn" data-id="${work.id}">Просмотреть</button>  
    <button class="delete-btn" data-id="${work.id}">Удалить</button>  
  </div>  
`;  
worksSection.appendChild(workItem);  
});  
}  
  
// Обработчики для кнопок работ  
worksSection.addEventListener('click', function(e) {  
  const workId = parseFloat(e.target.getAttribute('data-id'));  
  
  if (e.target.classList.contains('view-btn')) {  
    viewWork(workId);  
  } else if (e.target.classList.contains('delete-btn')) {  
    deleteWork(workId);  
  }  
});  
  
// Просмотр работы  
function viewWork(id) {  
  const works = JSON.parse(localStorage.getItem('works')) || [];  
  const work = works.find(w => w.id === id);  
  
  if (work) {  
    const pdfWindow = window.open();  
    pdfWindow.document.write(`  
      <!DOCTYPE html>  
      <html>  
      <head>  
        <title>${work.title}</title>  
      </head>  
      <body style="margin: 0; padding: 0;">  
        <embed  
          src="data:application/pdf;base64,${work.fileData}"  
          type="application/pdf"  
          width="100%"  
          height="100%"  
          style="position: fixed; top: 0; left: 0; width: 100%; height: 100%; border: none;"  
        >  
      </body>  
    </html>  
  `);  
  }  
}
```

```
// Удаление работы
function deleteWork(id) {
  if (confirm('Вы уверены, что хотите удалить эту работу?')) {
    let works = JSON.parse(localStorage.getItem('works')) || [];
    works = works.filter(w => w.id !== id);
    localStorage.setItem('works', JSON.stringify(works));
    displayWorks();
  }
}
// Инициализация
loadSavedData();
displayWorks();
});
```

Нажимаем на кнопку «Редактировать» — откроется форма, где можно загрузить новую фотографию, выбрав файл через проводник (поддерживаются изображения в форматах JPG, PNG и других), а также ввести или изменить краткую информацию о себе в специальном текстовом поле, после чего сохраняем изменения, нажав соответствующую кнопку, и ваши данные автоматически обновятся в профиле.

Мое портфолио

Обо мне Мои работы Добавить работу

Обо мне Редактировать

Фото профиля:

Выберите файл фото.jpg

Информация обо мне:

Я студент Приамурского государственного университета, учусь на 4 курсе бакалавриата. Пишу научные статьи на различные темы, исследую актуальные проблемы и предлагая новые подходы к их решению. Мое портфолио включает в себя как завершённые работы, так и проекты, находящиеся на стадии разработки, и отражает мой рост и развитие в научной области.

Сохранить Отмена

Рисунок 4. Редактирование профиля

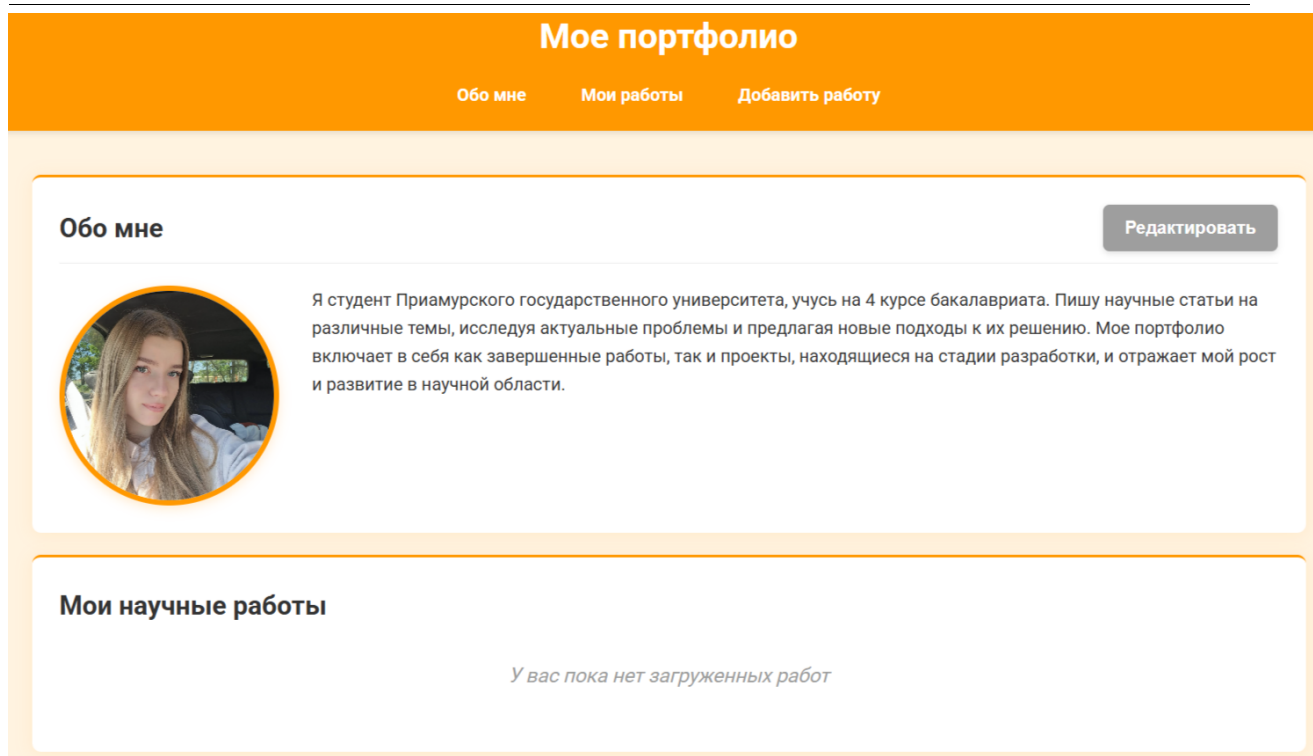


Рисунок 5. Редактирование профиля

Для добавления научной работы в портфолио вводим название работы в соответствующее поле и загружаем файл в формате PDF, нажав кнопку "Выбрать файл" "Загрузить", после чего работа будет сохранена и отображена в вашем профиле.

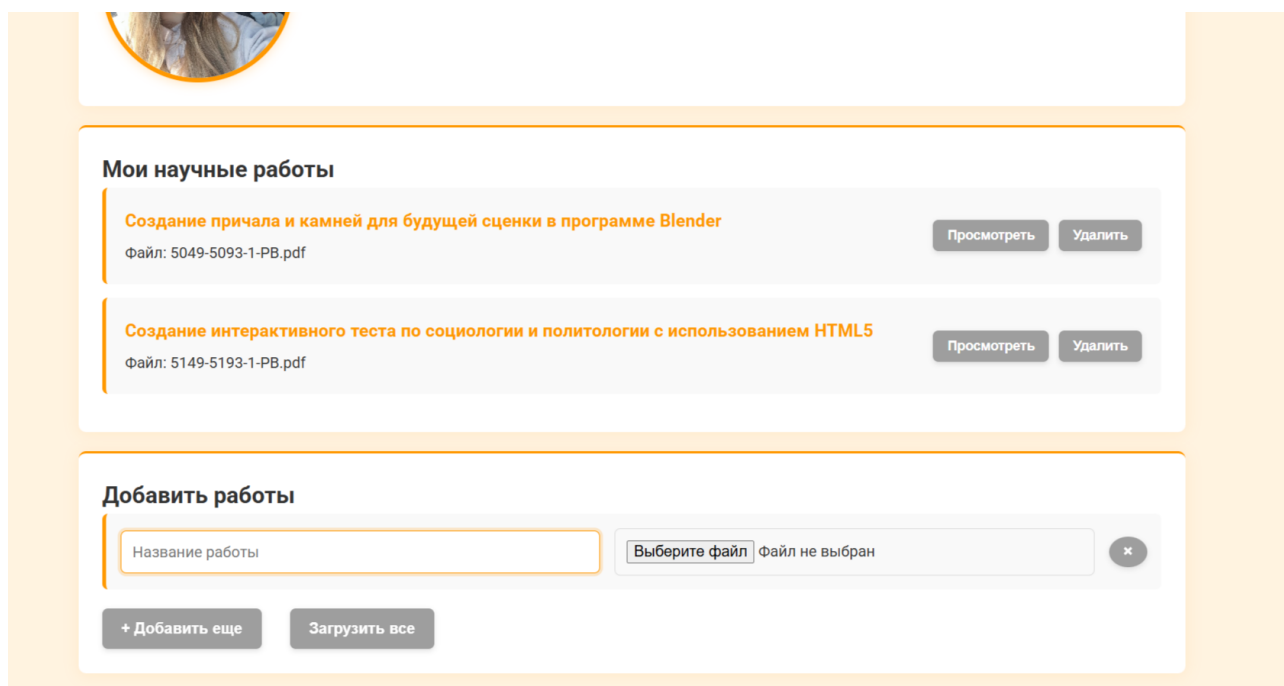


Рисунок 6. Добавление научной работы

При нажатии на кнопку «Просмотреть» в карточке научной работы откроется новая вкладка браузера, где будет отображен полный текст статьи в формате PDF с возможностью просмотра, масштабирования и печати.

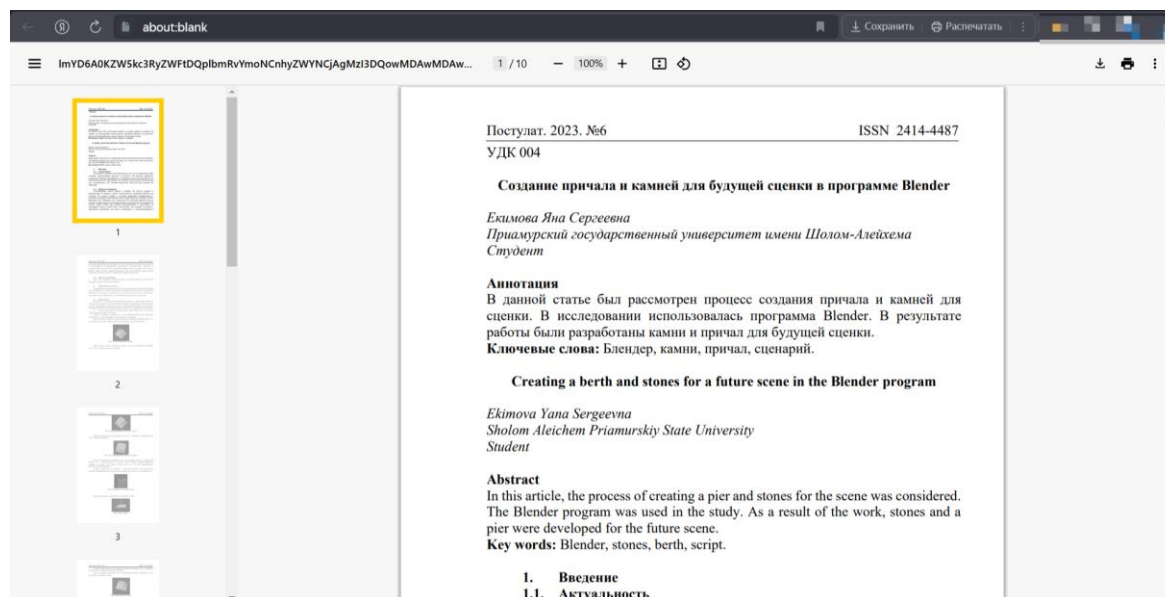


Рисунок 7. Просмотр добавленной научной работы

При нажатии на кнопку «Удалить» система отобразит диалоговое окно с подтверждением действия. Если пользователь подтвердит удаление, выбрав «Да», соответствующая научная работа будет безвозвратно удалена из портфолио и локального хранилища браузера.

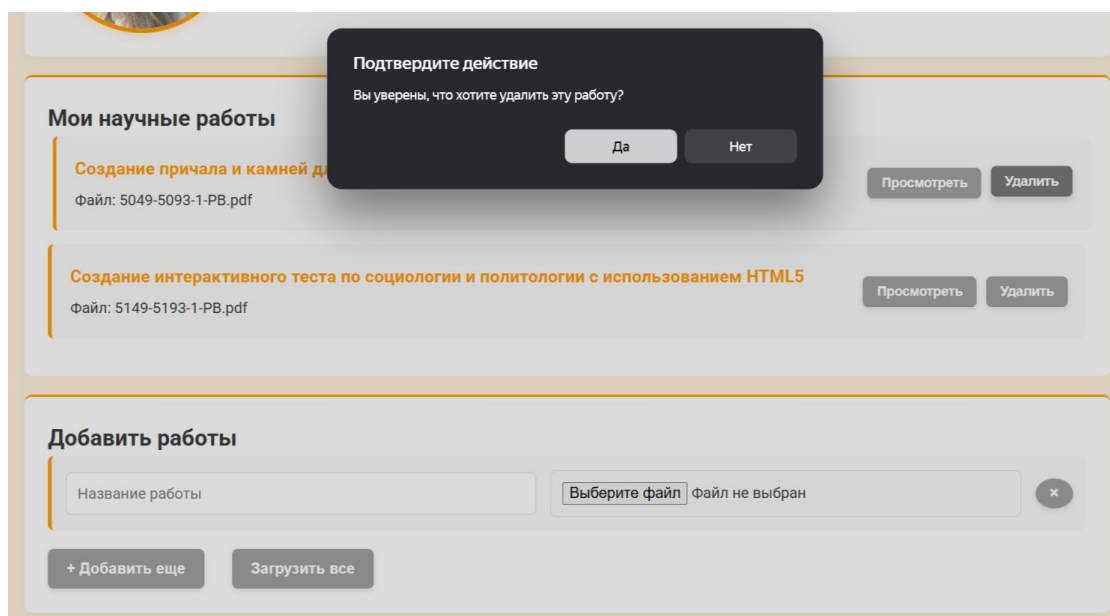


Рисунок 8. Удаление добавленной научной работы

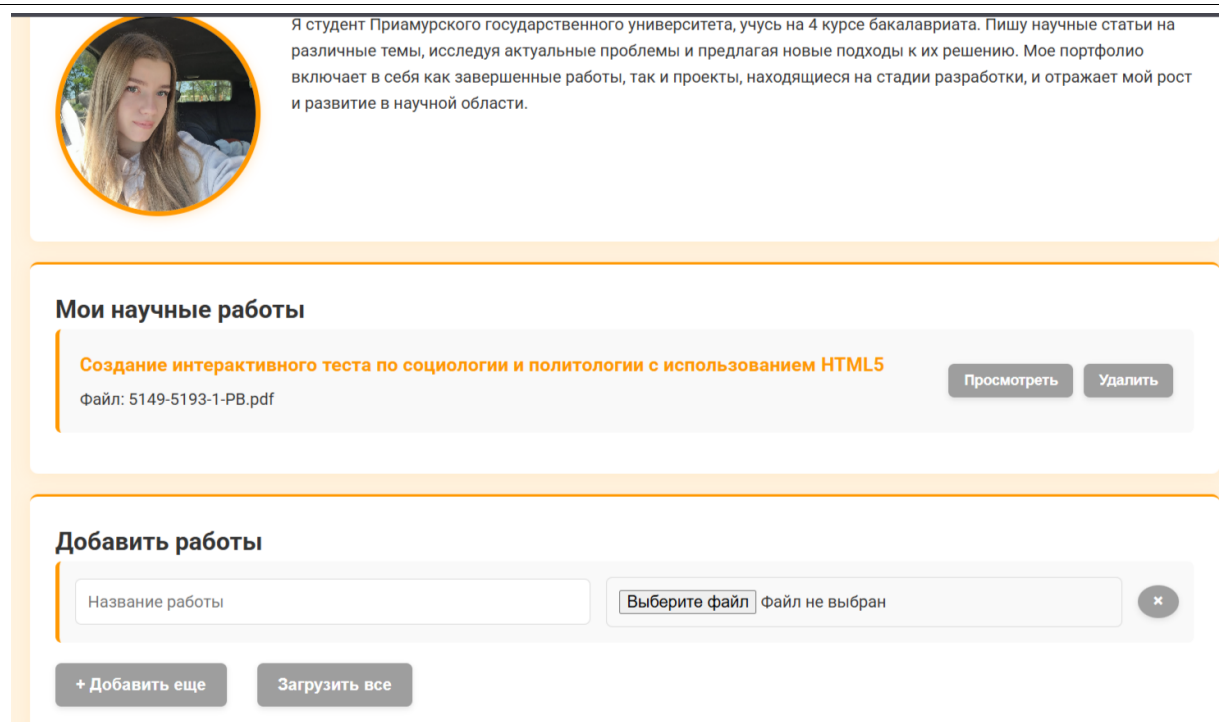


Рисунок 9. Удаление добавленной научной работы

4 Выводы

В данной работе подробно рассмотрено создание современного и стильного веб-портфолио для студентов, где они смогут представлять свои научные работы. Разработан интерфейс с использованием HTML5, стилизованный с помощью CSS, и реализована функциональность для демонстрации научных работ, включая раздел о студенте, каталог работ и возможность их добавления. Предложенное решение ориентировано на преподавателей и студентов, заинтересованных в применении современных веб-технологий для повышения эффективности образовательного процесса и превращения обучения в более увлекательный и интерактивный опыт.

Библиографический список

1. Бакулевская С. С. Подготовка педагогических работников к использованию технологий html5 для разработки электронных образовательных ресурсов //Информатика и образование. 2019. №. 5. С. 32-40.
2. Цилинская К.А. Создание и использование мультимедийных интерактивных плакатов по информатике //Будущее науки-2019. 2019. С. 340-344.
3. Королук Я. А., Куликова Н. Ю. Опыт разработки и использования интерактивных образовательных ресурсов на основе html5 для онлайн-обучения школьников информатике //XXV Всероссийская студенческая научно-практическая конференция Нижневартковского государственного университета. 2023. С. 48-55.
4. Размологов Д. М., Панина А. С., Захаров П. В. Использование html5 для разработки электронного учебника //Современные тенденции развития

- науки и технологий. 2016. №. 4-4. С. 92-94.
5. Шегаев И. С., Полозов А. Г. Новейший политологический глоссарий: основные категории, понятия, термины. Общество с ограниченной ответственностью Издательство Перо, 2015. 242 с.

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Портфолио студента</title>
  <link rel="stylesheet" href="styles.css">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap"
rel="stylesheet">
</head>
<body>
  <header>
    <h1>Мое портфолио</h1>
    <nav>
      <ul>
        <li><a href="#about">Обо мне</a></li>
        <li><a href="#research">Мои работы</a></li>
        <li><a href="#upload">Добавить работу</a></li>
      </ul>
    </nav>
  </header>

  <main>
    <section id="about">
      <div class="section-header">
        <h2>Обо мне</h2>
        <button id="editAboutBtn">Редактировать</button>
      </div>
      <div id="aboutDisplay">
        
        <div id="aboutText">
          <p>Здесь будет информация обо мне</p>
        </div>
      </div>
      <div id="aboutEdit" style="display: none;">
        <form id="aboutForm">
          <div class="form-group">
            <label for="photoUpload">Фото профиля:</label>
            <input type="file" id="photoUpload" accept="image/*">
          </div>
          <div class="form-group">
            <label for="aboutInfo">Информация обо мне:</label>
            <textarea id="aboutInfo" rows="6" placeholder="Расскажите о себе, своих навыках и
достижениях"></textarea>
          </div>
          <div class="form-actions">
            <button type="submit">Сохранить</button>
            <button type="button" id="cancelEditBtn">Отмена</button>
          </div>
        </form>
      </div>
    </section>

    <section id="research">
      <h2>Мои научные работы</h2>
      <div id="works">
        <p class="empty-message">У вас пока нет загруженных работ</p>
      </div>
    </section>
```

```
<section id="upload">
  <h2>Добавить работы</h2>
  <form id="uploadForm">
    <div id="fileInputsContainer">
      <div class="file-input-group">
        <input type="text" class="title" placeholder="Название работы" required>
        <input type="file" class="file" accept=".pdf" required>
        <button type="button" class="remove-btn">×</button>
      </div>
    </div>
    <div class="form-actions">
      <button type="button" id="addMoreBtn">+ Добавить еще</button>
      <button type="submit">Загрузить все</button>
    </div>
  </form>
</section>
</main>

<footer>
  <p>&copy; 2025 Мое портфолио!</p>
</footer>

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

```
* {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
}

body {
  font-family: 'Roboto', sans-serif;
  background-color: #fff3e0;
  color: #333;
  line-height: 1.6;
}

header {
  background: #ff9800;
  color: white;
  padding: 20px;
  text-align: center;
  box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}

nav ul {
  list-style: none;
  padding: 0;
  display: flex;
  justify-content: center;
  margin-top: 15px;
}

nav ul li {
  margin: 0 15px;
}

nav a {
  color: white;
  text-decoration: none;
  font-weight: 700;
  transition: all 0.3s;
  padding: 5px 10px;
  border-radius: 4px;
}

nav a:hover {
  background-color: #ffb74d;
}

main {
  padding: 20px;
  max-width: 1200px;
  margin: 0 auto;
}

section {
  padding: 25px;
  margin: 20px 0;
  background: white;
  border-radius: 8px;
  box-shadow: 0 2px 15px rgba(255, 152, 0, 0.1);
  border-top: 3px solid #ff9800;
}
```

```
.section-header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  margin-bottom: 20px;
  padding-bottom: 10px;
  border-bottom: 1px solid #eee;
}

#aboutDisplay {
  display: flex;
  gap: 30px;
  align-items: flex-start;
}

#profilePhoto {
  width: 200px;
  height: 200px;
  object-fit: cover;
  border-radius: 50%;
  border: 5px solid #ff9800;
  box-shadow: 0 3px 10px rgba(255, 152, 0, 0.2);
}

.form-group {
  margin-bottom: 20px;
}

label {
  display: block;
  margin-bottom: 8px;
  font-weight: bold;
  color: #555;
}

input[type="text"],
input[type="file"],
textarea {
  width: 100%;
  padding: 12px;
  border: 1px solid #ddd;
  border-radius: 6px;
  font-family: inherit;
  font-size: 16px;
  transition: border 0.3s;
}

input[type="text"]:focus,
input[type="file"]:focus,
textarea:focus {
  border-color: #ff9800;
  outline: none;
  box-shadow: 0 0 3px rgba(255, 152, 0, 0.2);
}

textarea {
  resize: vertical;
  min-height: 150px;
}

button, .view-btn, .delete-btn {
  background-color: #9e9e9e;
  color: white;
```

```
border: none;
padding: 12px 20px;
border-radius: 6px;
cursor: pointer;
font-weight: bold;
font-size: 16px;
transition: all 0.3s;
box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}

button:hover, .view-btn:hover, .delete-btn:hover {
  background-color: #757575;
  transform: translateY(-2px);
  box-shadow: 0 4px 8px rgba(0,0,0,0.15);
}

.form-actions {
  display: flex;
  gap: 15px;
  margin-top: 20px;
}

#addMoreBtn {
  margin-right: 15px;
}

.file-input-group {
  margin-bottom: 20px;
  display: flex;
  align-items: center;
  gap: 15px;
  padding: 15px;
  background: #f9f9f9;
  border-radius: 6px;
  border-left: 4px solid #ff9800;
}

.remove-btn {
  background: #9e9e9e;
  color: white;
  border: none;
  width: 32px;
  height: 32px;
  border-radius: 50%;
  cursor: pointer;
  font-size: 18px;
  display: flex;
  align-items: center;
  justify-content: center;
  transition: all 0.3s;
}

.remove-btn:hover {
  background: #757575;
}

.work-item {
  background: #f9f9f9;
  padding: 20px;
  margin-bottom: 15px;
  border-radius: 6px;
  border-left: 4px solid #ff9800;
  display: flex;
```

```
    justify-content: space-between;
    align-items: center;
    transition: transform 0.3s;
}

.work-item:hover {
    transform: translateX(5px);
}

.work-info h3 {
    color: #ff9800;
    margin-bottom: 5px;
}

.work-actions {
    display: flex;
    gap: 10px;
}

.view-btn, .delete-btn {
    padding: 8px 15px;
    font-size: 14px;
}

.empty-message {
    color: #9e9e9e;
    font-style: italic;
    text-align: center;
    padding: 30px;
    font-size: 18px;
}

footer {
    text-align: center;
    padding: 20px 0;
    background: #ff9800;
    color: white;
    margin-top: 40px;
    font-size: 14px;
}

@media (max-width: 768px) {
    #aboutDisplay {
        flex-direction: column;
        align-items: center;
    }

    nav ul {
        flex-direction: column;
        gap: 10px;
    }

    .file-input-group {
        flex-direction: column;
        align-items: flex-start;
    }
}
```

```
document.addEventListener('DOMContentLoaded', function() {
  // Элементы раздела "Обо мне"
  const editAboutBtn = document.getElementById('editAboutBtn');
  const aboutDisplay = document.getElementById('aboutDisplay');
  const aboutEdit = document.getElementById('aboutEdit');
  const aboutForm = document.getElementById('aboutForm');
  const cancelEditBtn = document.getElementById('cancelEditBtn');
  const photoUpload = document.getElementById('photoUpload');
  const aboutInfo = document.getElementById('aboutInfo');
  const profilePhoto = document.getElementById('profilePhoto');
  const aboutText = document.getElementById('aboutText');

  // Элементы загрузки работ
  const uploadForm = document.getElementById('uploadForm');
  const fileInputsContainer = document.getElementById('fileInputsContainer');
  const addMoreBtn = document.getElementById('addMoreBtn');
  const worksSection = document.getElementById('works');

  // Редактирование информации "Обо мне"
  editAboutBtn.addEventListener('click', function() {
    aboutDisplay.style.display = 'none';
    aboutEdit.style.display = 'block';
    aboutInfo.value = aboutText.innerHTML;
  });

  cancelEditBtn.addEventListener('click', function() {
    aboutDisplay.style.display = 'flex';
    aboutEdit.style.display = 'none';
  });

  aboutForm.addEventListener('submit', function(e) {
    e.preventDefault();
    aboutText.innerHTML = aboutInfo.value;

    if (photoUpload.files[0]) {
      const reader = new FileReader();
      reader.onload = function(e) {
        profilePhoto.src = e.target.result;
        localStorage.setItem('profilePhoto', e.target.result);
      };
      reader.readAsDataURL(photoUpload.files[0]);
    }

    localStorage.setItem('aboutInfo', aboutInfo.value);
    aboutDisplay.style.display = 'flex';
    aboutEdit.style.display = 'none';
  });

  // Загрузка сохраненных данных
  function loadSavedData() {
    if (localStorage.getItem('profilePhoto')) {
      profilePhoto.src = localStorage.getItem('profilePhoto');
    }

    if (localStorage.getItem('aboutInfo')) {
      aboutText.innerHTML = localStorage.getItem('aboutInfo');
    }
  }

  // Добавление новых полей для загрузки
  addMoreBtn.addEventListener('click', function() {
    const newGroup = document.createElement('div');
```

```
newGroup.className = 'file-input-group';
newGroup.innerHTML = `
  <input type="text" class="title" placeholder="Название работы" required>
  <input type="file" class="file" accept=".pdf" required>
  <button type="button" class="remove-btn">×</button>
`;
fileInputsContainer.appendChild(newGroup);
});

// Удаление группы полей (исправленный обработчик)
fileInputsContainer.addEventListener('click', function(e) {
  if (e.target.classList.contains('remove-btn')) {
    const groups = document.querySelectorAll('.file-input-group');
    if (groups.length > 1) {
      e.target.closest('.file-input-group').remove();
    } else {
      alert('Должна остаться хотя бы одна группа для загрузки');
    }
  }
});

// Обработка отправки формы (упрощенная рабочая версия)
uploadForm.addEventListener('submit', function(e) {
  e.preventDefault();

  const groups = document.querySelectorAll('.file-input-group');
  let works = JSON.parse(localStorage.getItem('works')) || [];
  let hasEmptyFields = false;

  // Проверка заполнения полей
  groups.forEach(group => {
    const title = group.querySelector('.title').value;
    const file = group.querySelector('.file').files[0];

    if (!title || !file) {
      hasEmptyFields = true;
      group.style.outline = "2px solid red";
    } else {
      group.style.outline = "";
    }
  });

  if (hasEmptyFields) {
    alert("Пожалуйста, заполните все поля!");
    return;
  }

  // Обработка файлов
  groups.forEach(group => {
    const title = group.querySelector('.title').value;
    const file = group.querySelector('.file').files[0];

    const reader = new FileReader();
    reader.onload = function(e) {
      const work = {
        id: Date.now() + Math.random(),
        title: title,
        fileName: file.name,
        fileData: e.target.result.split(',')[1]
      };
      works.push(work);
    };
  });
});
```

```
// Проверяем последний файл
if (works.length === groups.length) {
    localStorage.setItem('works', JSON.stringify(works));
    displayWorks();

    // Сбрасываем форму
    uploadForm.reset();

    // Оставляем одну группу
    while (fileInputsContainer.children.length > 1) {
        fileInputsContainer.removeChild(fileInputsContainer.lastChild);
    }

    alert('Работы успешно добавлены!');
}
};
reader.readAsDataURL(file);
});
});

// Отображение работ
function displayWorks() {
    const works = JSON.parse(localStorage.getItem('works')) || [];

    if (works.length === 0) {
        worksSection.innerHTML = '<p class="empty-message">У вас пока нет загруженных работ</p>';
        return;
    }

    worksSection.innerHTML = '';

    works.forEach(work => {
        const workItem = document.createElement('div');
        workItem.className = 'work-item';
        workItem.innerHTML = `
            <div class="work-info">
                <h3>${work.title}</h3>
                <p>Файл: ${work.fileName}</p>
            </div>
            <div class="work-actions">
                <button class="view-btn" data-id="${work.id}">Просмотреть</button>
                <button class="delete-btn" data-id="${work.id}">Удалить</button>
            </div>
        `;
        worksSection.appendChild(workItem);
    });
}

// Обработчики для кнопок работ (исправленный делегирование)
worksSection.addEventListener('click', function(e) {
    if (e.target.classList.contains('view-btn')) {
        const workId = parseFloat(e.target.getAttribute('data-id'));
        viewWork(workId);
    } else if (e.target.classList.contains('delete-btn')) {
        const workId = parseFloat(e.target.getAttribute('data-id'));
        deleteWork(workId);
    }
});

// Просмотр работы
function viewWork(id) {
    const works = JSON.parse(localStorage.getItem('works')) || [];
    const work = works.find(w => w.id === id);
```

```
if (work) {
  const pdfWindow = window.open();
  pdfWindow.document.write(`
    <!DOCTYPE html>
    <html>
    <head>
      <title>${work.title}</title>
    </head>
    <body style="margin: 0; padding: 0;">
      <embed
        src="data:application/pdf;base64,${work.fileData}"
        type="application/pdf"
        width="100%"
        height="100%"
        style="position: fixed; top: 0; left: 0; width: 100%; height: 100%; border: none;"
      >
    </body>
    </html>
  `);
}

// Удаление работы
function deleteWork(id) {
  if (confirm('Вы уверены, что хотите удалить эту работу?')) {
    let works = JSON.parse(localStorage.getItem('works')) || [];
    works = works.filter(w => w.id !== id);
    localStorage.setItem('works', JSON.stringify(works));
    displayWorks();
  }
}

// Инициализация
loadSavedData();
displayWorks();
});
```