



МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«МИРЭА – Российский технологический университет»  
**РТУ МИРЭА.**

**Институт информационных технологий (ИИТ)**  
**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**ОТЧЁТ ПО КУРСОВОЙ РАБОТЕ**  
**Ознакомительная практика**

Отчет представлен к  
рассмотрению:

Студент группы ИКБО-16-22

«\_\_» июня 2023

Схрейдер А.К.  
(подпись и расшифровка подписи)

Отчет утвержден.  
Допущен к защите:

Руководитель практики  
от кафедры

«\_\_» июня 2023

Матчин В.Т.  
(подпись и расшифровка подписи)

Москва 2023 г.



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

### **ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине: Разработка клиентских частей интернет-ресурсов

по профилю: Разработка программных продуктов и проектирование информационных систем

направления профессиональной подготовки: Программная инженерия (09.03.04)

Студент: Схрейдер Александр Кунрадович

Группа: ИКБО-36-22

Срок представления к защите: 12.12.2023

Руководитель: Василий Тимофеевич Матчин, старший преподаватель

**Тема:** Клиентская часть интернет-ресурса «150 лет С.В. Рахманинову»

**Исходные данные:** используемые технологии: HTML5, CSS3, JavaScript, редактор кода Visual Studio Code, версия 1.83.1, наличие: интерактивного поведения веб-страниц, межстраничной навигации, внешнего вида страниц, соответствующего современным стандартам веб-разработки; технологий адаптивной верстки для полноценного отображения контента на различных браузерах и видах устройств. Нормативный документ: инструкция по организации и проведению курсового проектирования СМК МИРЭА 7.5.1/04.И.05-18.

**Перечень вопросов, подлежащих разработке, и обязательного графического материала:** 1. Провести анализ предметной области разрабатываемой клиентской части интернет-ресурса. 2. Обосновать выбор технологий разработки клиентской части интернет-ресурса. 3. Создать пять и более веб-страниц интернет-ресурса. 4. Организовать межстраничную навигацию. 5. Реализовать слой клиентской логики веб-страниц с применением технологии JavaScript. 6. Провести оптимизацию веб-страниц и размещаемого контента для браузеров и различных видов устройств. 7. Создать презентацию по выполненной курсовой работе. 8. Оформление курсовой работы должно соответствовать ГОСТ 7.32-2017

Руководителем произведён инструктаж по технике безопасности, противопожарной технике и правилам внутреннего распорядка.

Зав. кафедрой ИиППО: \_\_\_\_\_/Р. Г. Болбаков/, «18» сентября 2023 г.

Задание на КР выдал: \_\_\_\_\_/В.Т. Матчин/, «18» сентября 2023 г.

Задание на КР получил: \_\_\_\_\_/А.К. Схрейдер /, «18» сентября 2023 г.

## РЕФЕРАТ

Отчет 41 с., 15 источн., 1 прил., 1 листинг

### КЛИЕНТСКАЯ ЧАСТЬ ИНТЕРНЕТ-РЕСУРСА «150 ЛЕТ С.В.РАХМАНИНОВУ»

Объектом исследования являются средства создания веб-приложений на основе HTML, CSS и JavaScript.

Цель работы – Провести анализ предметной области разрабатываемой клиентской части интернет-ресурса. Обосновать выбор технологий разработки клиентской части интернет-ресурса. Создать пять и более веб-страниц интернет-ресурса. Организовать межстраничную навигацию. Реализовать слой клиентской логики веб-страниц с применением технологии JavaScript. Провести оптимизацию веб-страниц и размещаемого контента для браузеров и различных видов устройств.

В процессе работы проводилось изучение особенностей отдельных инструментов и методов создания веб-приложений на основе HTML, CSS и JavaScript, которые могут быть применены для создания клиентской части интернет-ресурса.

В результате работы был разработан программный код, реализующий определенные функции веб-приложения на основе указанных технологий. Эти функции включают в себя отображение информации о С.В. Рахманинове, предоставление возможности просмотра его произведений, а также взаимодействие пользователей с ресурсом.

Область применения результатов: разработка клиентской части веб-приложения на основе HTML, CSS и JavaScript для интернет-ресурса, посвященного 150-летию С.В. Рахманинова.

## СОДЕРЖАНИЕ

1.ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ.....	5
2.ВВЕДЕНИЕ.....	6
3. ОСНОВНАЯ ЧАСТЬ .....	7
3.1 Анализ предметной области разрабатываемой клиентской части интернет-ресурса.....	7
3.2 Выбор технологий разработки клиентской части интернет-ресурса .....	8
3.3 Создание вебстраниц клиентской части интернет-ресурса с использованием технологий HTML5, CSS3 и JavaScript.....	9
3.3.1 Первая страница .....	9
3.3.2 Вторая страница .....	22
3.3.3 Третья страница.....	26
3.3.4 Четвёртая страница .....	31
3.3.5 Пятая страница .....	35
3.4 Создание межстраничной навигации.....	37
3.5 Реализация слоя клиентской логики веб-страниц с применением технологии JavaScript .....	38
6. ЗАКЛЮЧЕНИЕ .....	40
7.СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	42
8.ПРИЛОЖЕНИЕ А .....	43

# 1.ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

В настоящем отчете применяются следующие термины с соответствующими определениями.

1. HTML (HyperText Markup Language) - язык разметки, используемый для создания структуры и содержания веб-страниц.
2. CSS (Cascading Style Sheets) - язык стилей, используемый для задания внешнего вида и форматирования веб-страниц.
3. JavaScript - объектно-ориентированный язык программирования, используемый для создания интерактивных элементов и функциональности на веб-страницах.
4. Веб-приложение - программа, которая работает в браузере и обеспечивает пользователю доступ к данным и функциональности через интернет.
5. SPA (Single Page Application) - веб-приложение, которое загружает только одну страницу и динамически обновляет ее содержимое без перезагрузки всей страницы.
6. MVC (Model-View-Controller) - архитектурный шаблон, используемый для разделения приложения на три компонента: модель, представление и контроллер.
7. Framework - набор инструментов, библиотек и шаблонов, который облегчает процесс разработки веб-приложений, предоставляя готовые решения для повторяющихся задач.

## 2.ВВЕДЕНИЕ

В современном мире, где виртуальное пространство становится важным аспектом общественной жизни, создание интернет-ресурсов, посвященных выдающимся личностям, приобретает особую значимость. В рамках данного проекта представляется разработка клиентской части интернет-ресурса, посвященного 150-летию великого композитора С.В. Рахманинова. Этот веб-портал не только покажет уважение к творчеству мастера, но и предоставит пользователю уникальную возможность погрузиться в мир его музыки и жизни.

Данная разработка предполагает осуществление комплексного анализа предметной области, в рамках которого будут выявлены основные аспекты, определяющие структуру и функциональность клиентской части интернет-ресурса. Выбор технологий для разработки будет обоснован с учетом современных требований к веб-проектам и специфики представляемого контента.

Процесс создания более пяти веб-страниц и организации межстраничной навигации станет ключевым этапом, обеспечивающим удобство взаимодействия пользователя с ресурсом. Реализация слоя клиентской логики с использованием технологии JavaScript добавит интерактивность и динамичность, что улучшит пользовательский опыт.

Особое внимание будет уделено оптимизации веб-страниц и размещаемого контента для различных браузеров и устройств. Это обеспечит качественную работу ресурса на разнообразных платформах, что является важным условием для привлечения широкой аудитории.

Таким образом, разработка клиентской части интернет-ресурса «150 лет С.В. Рахманинову» представляет собой многогранный процесс, включающий в себя анализ, технологический выбор, создание контента, организацию навигации и оптимизацию, направленный на создание интерактивной и увлекательной среды для изучения наследия великого композитора.

### **3. ОСНОВНАЯ ЧАСТЬ**

#### **3.1 Анализ предметной области разрабатываемой клиентской части интернет-ресурса**

В ходе анализа предметной области, связанной с интернет-ресурсом «150 лет С.В. Рахманинову», выявляются ключевые аспекты, определяющие содержание и структуру будущего веб-портала. При создании ресурса необходимо учитывать как биографические и творческие моменты, так и визуальные и функциональные элементы, чтобы обеспечить пользователям полноценное взаимодействие с представляемым контентом.

В начале анализа ставится задача определения основных этапов жизни и творчества С.В. Рахманинова. Это включает в себя временные периоды, ключевые события, влияние эпохи на его музыкальное наследие. Такой аспект позволит структурировать информацию на веб-ресурсе и обеспечить пользователей максимально полным и интересным обзором жизни композитора.

Далее следует анализ собранного аудио- и видеоматериала, включая музыкальные произведения, записи выступлений, а также фото- и видеоматериалы из различных периодов жизни Рахманинова. Этот аспект важен для создания визуально привлекательного и информативного интерфейса, который позволит посетителям не только читать о композиторе, но и непосредственно погружаться в его творческий мир через мультимедийный контент.

Одним из ключевых моментов анализа является также выявление основных интересов и потребностей целевой аудитории. Это поможет адаптировать веб-ресурс под ожидания пользователей, обеспечивая более эффективное взаимодействие с контентом и повышая общую привлекательность ресурса.

Таким образом, анализ предметной области включает в себя глубокое исследование биографических, творческих, визуальных и аудиовизуальных аспектов, что предоставляет фундаментальную основу для разработки клиентской части интернет-ресурса "150 лет С.В. Рахманинову".

### **3.2 Выбор технологий разработки клиентской части интернет-ресурса**

При проектировании клиентской части интернет-ресурса "150 лет С.В. Рахманинову" осуществлен осознанный выбор технологий, направленный на обеспечение эффективного воплощения задач проекта и удовлетворения потребностей пользователей. В данной разработке основной акцент сделан на использовании базовых технологий веб-разработки, таких как HTML, CSS и JavaScript.

HTML (HyperText Markup Language) предоставляет нам универсальный и понятный язык разметки, который позволяет структурировать информацию на веб-страницах.[1] С учетом особенностей контента, связанного с жизнью и творчеством С.В. Рахманинова, HTML становится основным инструментом для создания четкой и логичной структуры веб-ресурса.

CSS (Cascading Style Sheets) используется для стилизации визуального оформления веб-страниц.[2] Это позволяет создать эстетически приятный и удобочитаемый дизайн, соответствующий тематике ресурса и обеспечивающий единообразие в представлении информации.

JavaScript применяется для добавления интерактивности и динамичности на веб-страницах.[3] С использованием JavaScript реализуется слой клиентской логики, обеспечивающий более глубокое взаимодействие пользователя с ресурсом. Это может включать в себя анимации, интерактивные элементы, а также механизмы для загрузки и обработки данных.



Основываясь на простоте и эффективности данного набора технологий, разработка клиентской части интернет-ресурса в HTML, CSS и JavaScript позволяет не только достичь высокой производительности, но и обеспечить удобство сопровождения проекта в будущем. Это также содействует более широкой доступности ресурса, поскольку поддержка этих технологий встроена во все современные браузеры, обеспечивая единое и надежное воспроизведение контента на различных устройствах.

### **3.3 Создание вебстраниц клиентской части интернет-ресурса с использованием технологий HTML5, CSS3 и JavaScript**

#### **3.3.1 Первая страница**

Начальная страница веб-сайта играет фундаментальную роль в привлечении внимания посетителей и установлении первого визуального контакта с предлагаемым контентом. Это виртуальное приветствие предоставляет посетителям первое впечатление о сайте, его тематике и стиле, поэтому она занимает особое место в пользовательском опыте.

На первой странице сайта "150 лет С.В. Рахманинову" гостей встречает уникальное сочетание эстетики, информационной наполненности и удобства взаимодействия. Цель первой страницы - погрузить посетителя в атмосферу творчества композитора, предоставив краткую, но привлекательную информацию о его жизни и значимости.

Ключевые элементы интерфейса на первой странице спроектированы с учетом легкости навигации, позволяя посетителям быстро получить представление о том, что предлагает сайт. Это включает в себя ярко выделенные кнопки перехода к основным разделам и небольшие анимации элементов, создающие динамичный и запоминающийся пользовательский опыт.

Значительное внимание уделяется также текстовой информации на первой странице. Краткие и содержательные аннотации о цели сайта, его уникальных возможностях и предлагаемых ресурсах помогают посетителям быстро понять, почему стоит провести время именно здесь.

Для более детального визуального представления о первой странице сайта, предлагаю ознакомиться с рисунком 1.1 и 1.2.

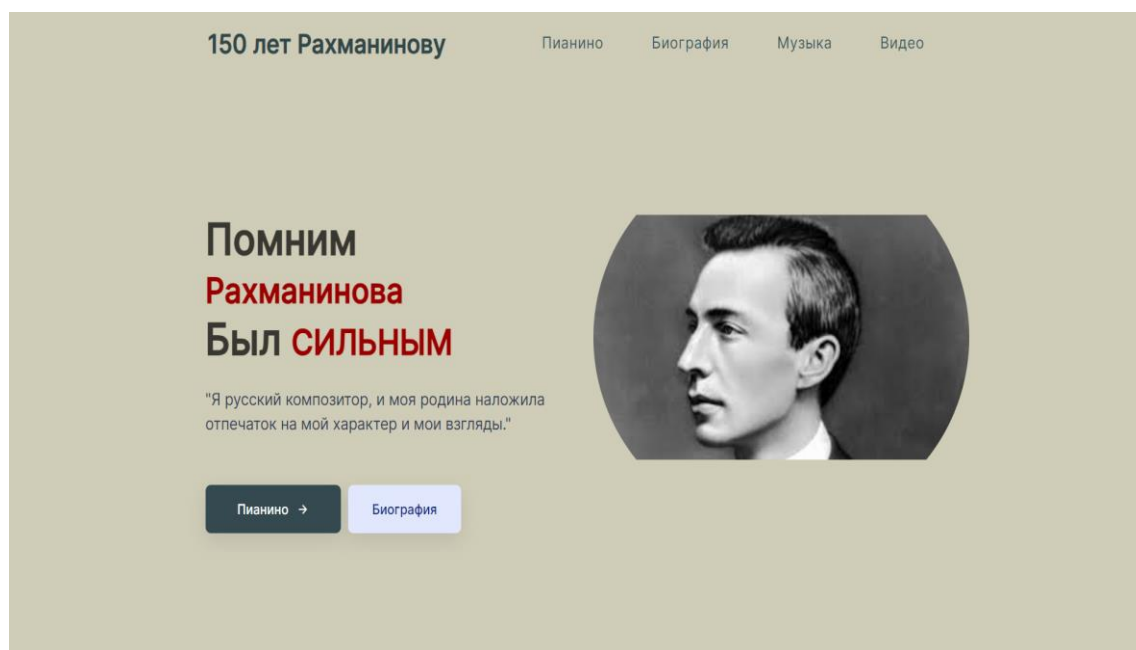


Рисунок 1.1 – Первая часть первой страницы сайта  
[разработано автором]

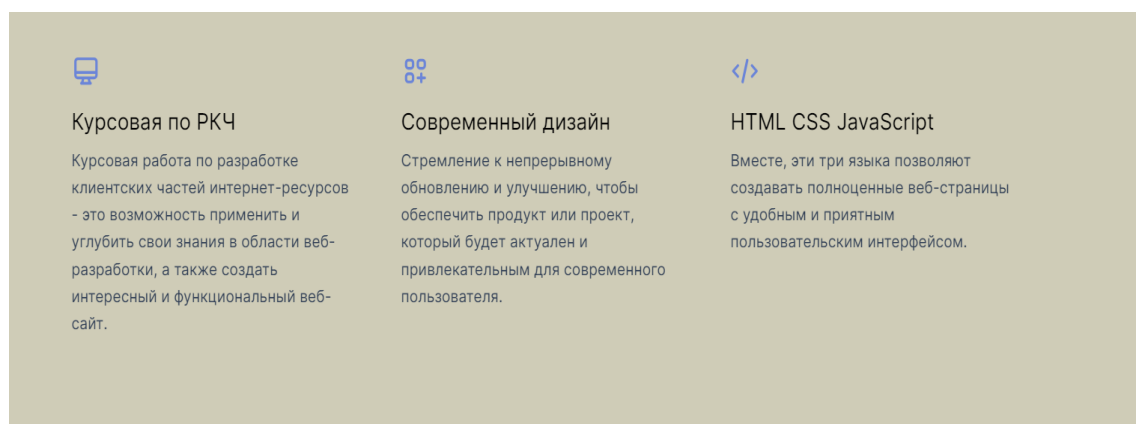


Рисунок 1.2 – Вторая часть первой страницы сайта  
[разработано автором]

Давайте погрузимся в мир кода и рассмотрим веб-сайт изнутри, где каждая строка кода — это мозаика, создающая визуальное волшебство первой

страницы. Этот аспект предоставляет уникальную перспективу, позволяя увидеть, как техническая гармония воплощает идеи и эстетику.

Приведенный код листинга 5.1 представляет собой часть HTML-структуры страницы, ответственной за верхнюю часть сайта – его заголовок или, как в данном случае, шапку. Давайте рассмотрим основные элементы этого листинга.

Листинг 5.1 – Программный код заголовка первой страницы

```
<header class="container header">
  <nav class="nav">
    <div class="logo">
      <h2>150 лет Рахманинову</h2>
    </div>

    <div class="nav_menu" id="nav_menu">
      <button
id="close_btn"
class="close_btn"
        <i class="ri-close-fill"></i>
      </button>

      <ul class="nav_menu_list">
        <li class="nav_menu_item">
          <a href="homepage/home.html"
class="nav_menu_link">пианино</a>
        </li>
        <li class="nav_menu_item">
          <a
href="timeline/timeline.html"
class="nav_menu_link">биография</a>
        </li>
        <li class="nav_menu_item">
          <a
href="musicpage/musicpage.html"
class="nav_menu_link">музыка</a>
        </li>
        <li class="nav_menu_item">
          <a href="videopage/video.html"
class="nav_menu_link">видео</a>
        </li>
      </ul>
    </div>

    <button class="toggle_btn" id="toggle_btn">
```

```
        <i class="ri-menu-line"></i>
      </button>
    </nav>
  </header>
```

Разбор кода:

1. `<header class="container header">`: Этот тег `<header>` обозначает начало блока шапки страницы. Класс `"container"` вероятно указывает на структурный контейнер, а `"header"` означает, что это часть шапки.
2. `<nav class="nav">`: Тег `<nav>` указывает на начало блока навигации. Класс `"nav"` вероятно связан с стилями этого блока.[4]
3. `<div class="logo">`: Этот блок содержит логотип сайта, представленный текстом "150 лет Рахманинову".
4. `<div class="nav_menu" id="nav_menu">`: Здесь находится блок, содержащий кнопку закрытия (`close_btn`) и список навигационных элементов (`nav_menu_list`). ID `"nav_menu"` может быть использован в JavaScript для управления отображением этого меню.
5. `<button class="toggle_btn" id="toggle_btn">`: Это кнопка-переключатель, предназначенная для открытия и закрытия меню. Она содержит иконку в виде трех горизонтальных линий, что обычно ассоциируется с иконой бургера.

Все вместе этот код формирует структуру и визуальный интерфейс верхней части сайта, предоставляя навигационные элементы и информацию о сайте.

Приведенный код листинга 5.2 представляет собой часть CSS страницы, ответственной за верхнюю часть сайта – его заголовок или, как в данном случае, шапку. Давайте рассмотрим основные элементы этого листинга.

Листинг 5.2 – Программный код

```
@import
url("https://fonts.googleapis.com/css2?family=Inter:wght@400;500&display=swap");

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

```

    font-family: "Inter", sans-serif;
}
*,
*::before,
*::after {
    box-sizing: border-box;
}

:root {
    --fixed-header-height: 4.5rem;
}

body {
    width: 100%;
    height: 100vh;
    overflow-x: hidden;
    background-color: #CFCCB7;
}
ul li {
    list-style-type: none;
}
a {
    text-decoration: none;
}
button {
    background-color: transparent;
    border: none;
    outline: none;
    cursor: pointer;
}

.container {
    width: 100%;
}
@media screen and (min-width: 1040px) {
    .container {
        width: 1040px;
        margin: 0 auto;
    }
}
/* ==== HEADER ==== */
.header {
    height: var(--fixed-header-height);

```

```

padding: 0 1.7rem;
}

.nav {
width: 100%;
height: 100%;
display: flex;
align-items: center;
justify-content: space-between;
}

.logo h2 {
font-size: 30px;
color: #31484b;
}

.nav_menu_list {
display: flex;
align-items: center;
}

.nav_menu_list .nav_menu_item {
margin: 0 2rem;
}

.nav_menu_item .nav_menu_link {
font-size: 18.5px;
line-height: 27px;
color: rgb(56, 88, 94);
text-transform: capitalize;
letter-spacing: 0.5px;
}

.nav_menu_link:hover {
color: #9b0000;
}

.toggle_btn {
font-size: 20px;
font-weight: 600;
color: var(--lg-heading);
z-index: 4;
}

.nav_menu,
.close_btn {
display: none;
}

.show {

```

```

    right: 3% !important;
}
@media screen and (min-width: 768px) {
    .toggle_btn {
        display: none;
    }
    .nav_menu {
        display: block;
    }
}

@media screen and (max-width: 768px) {
    .logo h2 {
        font-size: 23px;
    }
    .nav_menu {
        position: fixed;
        width: 93%;
        height: 100%;
        display: block;
        top: 2.5%;
        right: -100%;
        background-color: #fff;
        padding: 3rem;
        border-radius: 10px;
        box-shadow: 0 0.5rem 1.5rem rgba(22, 28, 45,
0.1);
        z-index: 50;
        transition: 0.4s;
    }
    .nav_menu_list {
        flex-direction: column;
        align-items: flex-start;
        margin-top: 4rem;
    }
    .nav_menu_list .nav_menu_item {
        margin: 1rem 0;
    }
    .nav_menu_item .nav_menu_link {
        font-size: 18px;
    }
    .close_btn {
        display: block;
        position: absolute;
        right: 10%;

```

```

        font-size: 25px;
        color: #50689e;
    }
    .close_btn:hover {
        color: #000;
    }
    .wrapper {
        padding: 0 0.7rem;
    }
    .grid-item-1 {
        padding-left: 0rem;
    }
    .main-heading {
        font-size: 35px;
    }
    .view_more_btn {
        width: 140px;
        height: 55px;
        font-size: 13.5px;
        margin-right: 1rem;
    }
    .grid-cols-3 {
        grid-template-columns: repeat(auto-fit,
minmax(100%, 1fr));
    }
    .featured_info p {
        line-height: 23px;
        font-size: 14px;
    }
}

@media screen and (max-width: 991px) {
    .wrapper {
        padding-top: 3rem;
    }
    .grid-cols-2 {
        grid-template-columns: repeat(auto-fit,
minmax(100%, 1fr));
    }
    .grid-item-1 {
        order: 2;
        display: flex;
        flex-direction: column;
        align-items: center;
        justify-content: center;
    }

```



```

        padding-top: 0;
    }
    .main-heading {
        font-size: 32px;
        text-align: center;
        line-height: 40px;
    }
    .info-text {
        font-size: 16px;
        text-align: center;
        padding: 0.7rem;
    }
    .btn_wrapper {
        width: 100%;
        display: flex;
        align-items: center;
        justify-content: center;
    }
    .grid-item-2 {
        order: 1;
        display: flex;
        flex-direction: column;
        align-items: center;
        justify-content: center;
    }
    .team_img_wrapper {
        width: 350px;
        height: 350px;
    }
    .featured_info span {
        font-size: 19px;
    }
}

```

Основные аспекты CSS кода:

1. Импорт шрифта Inter с использованием Google Fonts.
2. Сброс стилей и установка базовых свойств для всех элементов.
3. Определение переменной `--fixed-header-height` для высоты фиксированной шапки.[5]
4. Стилизация основного контейнера, адаптивная ширина при разрешении 1040px и выше.

5. Стилизация шапки (.header) с высотой, отступами и фоновым цветом - #CFCCB7. [6]
6. Стилизация навигационного блока (.nav) и логотипа.
7. Установка стилей для пунктов меню и ссылок, включая эффект при наведении.
8. Стилизация кнопки-переключателя (.toggle\_btn) с учетом размера, веса и цвета.
9. Отображение и скрывание элементов (.nav\_menu, .close\_btn) с использованием класса .show.

Медиа-запросы для адаптивности:

Код включает медиа-запросы, которые оптимизируют стили для различных устройств и разрешений экрана.

Для широких экранов (минимум 1040px) предусмотрена фиксированная ширина контейнера.

Для средних экранов (от 768px до 1040px) сохраняется адаптивная ширина контейнера.

Навигационное меню (Hamburger menu):

При ширине экрана менее 768px применяются стили для мобильного отображения.

Кнопка переключения меню (toggle\_btn) видна только на экранах с шириной более 768px.

На мобильных устройствах меню (nav\_menu) отображается при нажатии на кнопку и скрывается при повторном нажатии.

Для стилизации мобильного меню используются анимации (transition), задающие плавное открытие и закрытие.

Определение стилей для кнопок:

Стили для кнопок включают нулевые отступы и рамки (button), а также определены стили для кнопок с классами .btn, .view\_more\_btn, и .documentation\_btn.

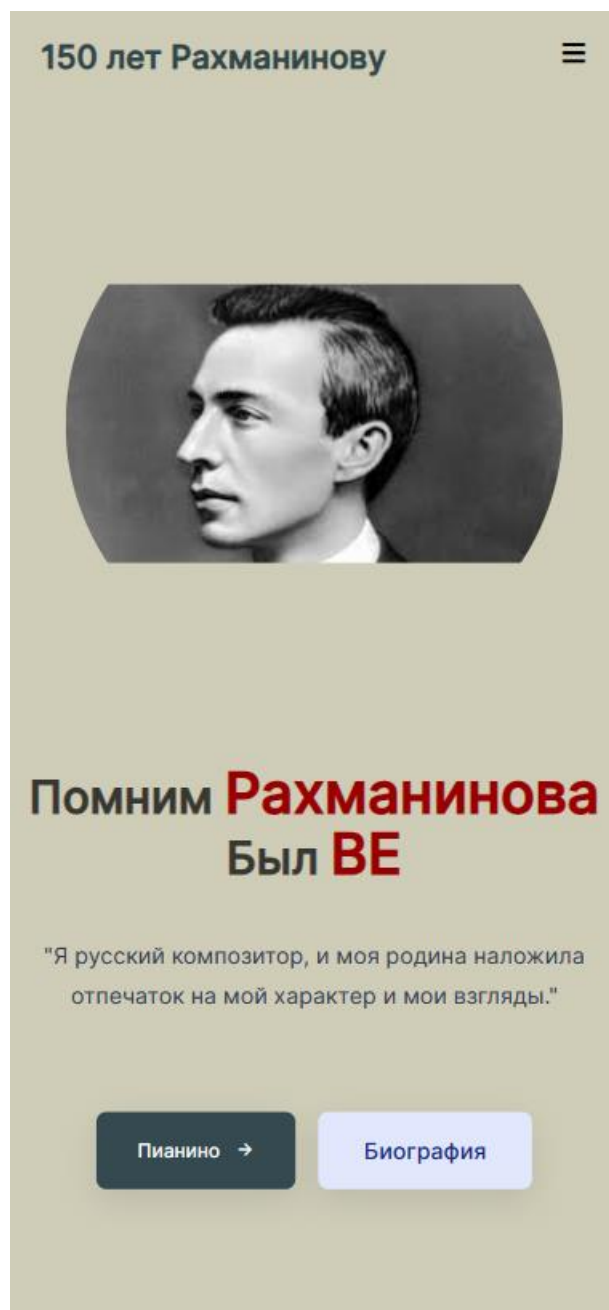
При наведении на кнопки реализованы эффекты изменения цвета и тени.

Адаптация контента:

На мобильных устройствах (ширина экрана до 768px) текст и изображения адаптированы для лучшей читаемости и визуального восприятия.

Для этого использованы медиа-запросы, определяющие стили для соответствующих разрешений экрана.

Этот код демонстрирует гибкость и адаптивность дизайна, обеспечивая удобное взаимодействие с сайтом на различных устройствах, можно убедиться в этом посмотрев рисунок 1.3.



Приведенный код листинга 5.3 представляет собой часть JavaScript страницы, ответственной за верхнюю часть сайта – его заголовок или, как в данном случае, шапку. Давайте рассмотрим основные элементы этого листинга.

#### Листинг 5.3 – Программный код

```
document.addEventListener("DOMContentLoaded",
function () {
    const options = {
        strings: ['ВЕЛИКИМ', 'КОМПОЗИТОР', 'МУДРЫМ',
'SИЛЬНЫМ' ],
        typeSpeed: 150,
        backSpeed: 50,
        backDelay: 1000,
        loop: true
    };

    const multiTextElement =
document.querySelector('.multi-text');
    let currentTextIndex = 0;
    let currentText = '';
    let isDeleting = false;

    function type() {
        const fullText =
options.strings[currentTextIndex];
        if (isDeleting) {
            currentText = fullText.substring(0,
currentText.length - 1);
        } else {
            currentText = fullText.substring(0,
currentText.length + 1);
        }

        multiTextElement.textContent = currentText;

        let typeSpeed = options.typeSpeed;
        if (isDeleting) {
            typeSpeed /= 2;
        }
    }
}
```

```

        if (!isDeleting && currentText === fullText) {
            typeSpeed = options.backDelay;
            isDeleting = true;
        } else if (isDeleting && currentText === '') {
            isDeleting = false;
            currentTextIndex = (currentTextIndex + 1) %
options.strings.length;
        }

        setTimeout(type, typeSpeed);
    }

    type();
});

```

Этот JavaScript-код отвечает за анимированный текстовый эффект, который позволяет создать динамичное изменение текста на веб-странице. Вот пошаговое объяснение:

Событие загрузки контента:

`document.addEventListener("DOMContentLoaded", function () {...})`: Этот код запускается, когда DOM (Document Object Model) полностью загружен.[7]

Настройка параметров анимации:

`const options = {...}`: Объект `options` содержит параметры для настройки анимации.

`strings`: Массив строк текста, который будет анимироваться.

`typeSpeed`: Скорость появления символов (набора текста).

`backSpeed`: Скорость удаления символов (стирания текста).

`backDelay`: Задержка перед стиранием текста после завершения печати.

`loop`: Флаг, указывающий, следует ли бесконечно повторять анимацию.

Выбор элемента для анимации:

`const multiTextElement = document.querySelector('.multi-text');`: Получение элемента с классом `.multi-text`, куда будет вставляться анимированный текст.

Инициализация переменных:

`let currentTextIndex = 0;`: Индекс текущей строки текста.

`let currentText = "";`: Текущий текст, который будет отображаться.

`let isDeleting = false;`: Флаг, указывающий, удаляется ли текст.

Функция для анимации текста:

`function type() {...}`: Эта функция выполняет основную логику анимации.

Получает текущий текст из массива `options.strings` и обновляет `multiTextElement` с текущим текстом.

Управляет скоростью появления и удаления символов.

Определяет, когда начинать стирание текста и когда закончить его.

Вызов функции для начала анимации:

`type();`: Запуск функции `type()`, чтобы начать анимацию текста.

Этот код создает эффект многоточечного текста, который появляется, стирается и затем заменяется следующей строкой текста в бесконечном цикле.

### **3.3.2 Вторая страница**

Вторая страница: "Музыкальная Галерея"

На второй странице интернет-ресурса, посвященного великому композитору С.В. Рахманинову, открывается удивительная "Музыкальная Галерея". Этот раздел не только предоставляет уникальную возможность виртуально испытать звучание пианино, но и обогащен цитатами самого Рахманинова, призванными вдохновить посетителей этой виртуальной палитры музыки.

Интерактивное Пианино:

Посетители могут взять виртуальное пианино под свое управление, играя любимые мелодии композитора. Эта функциональность предоставляет уникальный опыт, позволяя каждому почувствовать себя частью музыкального творчества Рахманинова. Откройте для себя виртуозность его произведений, воспроизводя их собственными руками.

Цитаты С.В. Рахманинова:

Каждый аккорд и нота на странице сопровождаются цитатами великого композитора. Слова Рахманинова приглашают посетителей окунуться в его

внутренний мир, вдохновляясь его мыслями о музыке, творчестве и жизни. Эти цитаты становятся мостом между произведениями и душой композитора.

Футер:

На странице присутствует стильный футер, который является не только элементом дизайна, но и практичным навигационным инструментом

Вторая страница "Музыкальной Галереи" не только раскрывает творческий мир композитора, но и создает интерактивную среду, погружая посетителей в увлекательный опыт виртуального исследования музыки.

Для более детального визуального представления о второй странице сайта, предлагаю ознакомиться с рисунком 2.1, 2.2, 2.3 и 2.4.



Рисунок 2.1 – Первая часть второй страницы сайта (интерактивное пианино)

[разработано автором]



Рисунок 2.2 – Вторая часть второй страницы сайта  
[разработано автором]

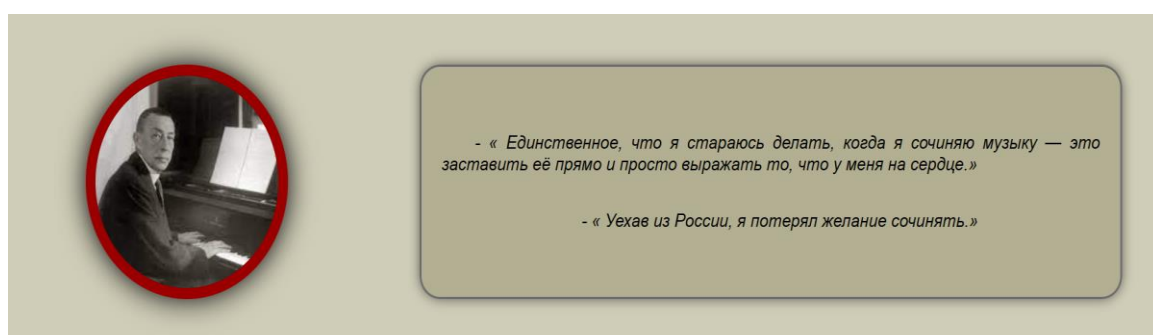


Рисунок 2.3 – Третья часть второй страницы сайта  
[разработано автором]

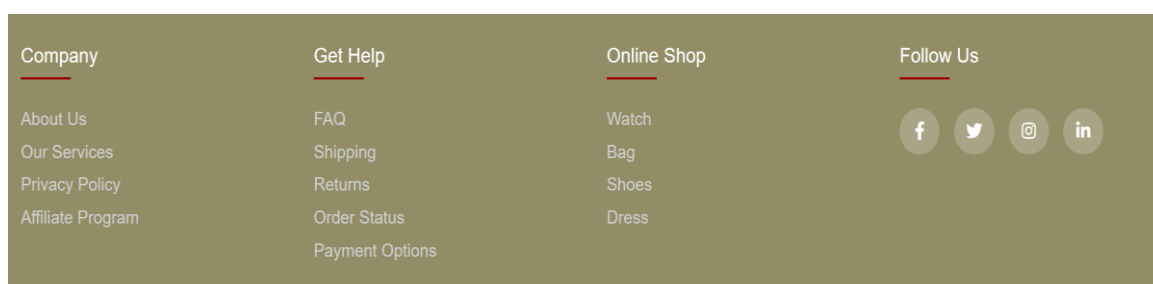


Рисунок 2.4 – Первая часть второй страницы сайта  
[разработано автором]

Дополнительно, была создана адаптивная версия для мобильных устройств. Для более детального визуального представления ознакомиться с рисунком 2.5 и 2.6.



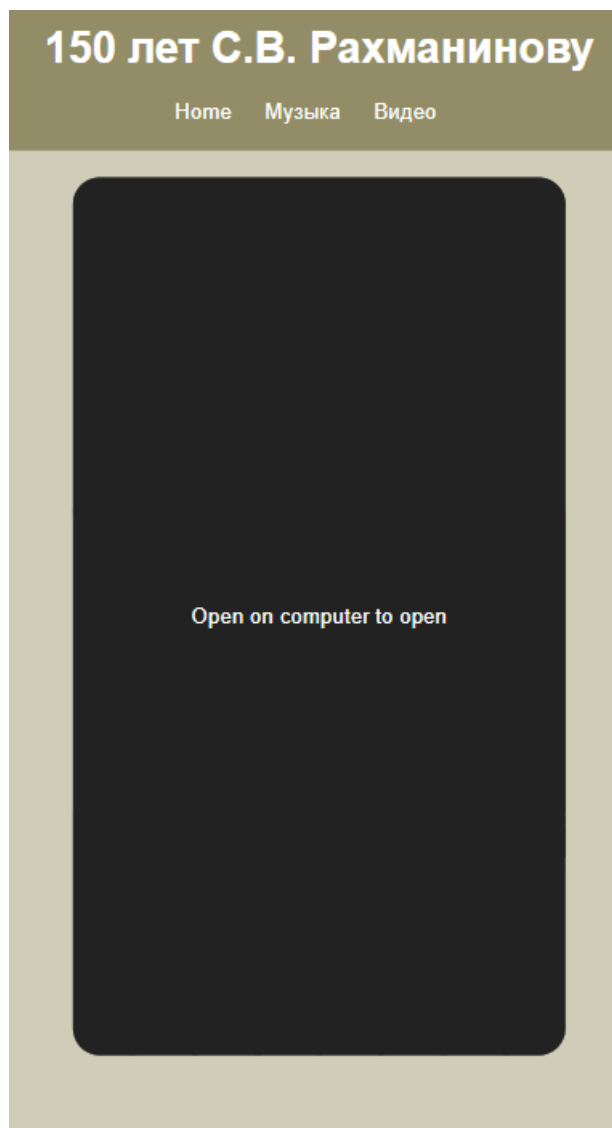


Рисунок 2.4 – адаптивная версия для мобильных устройств  
[разработано автором]

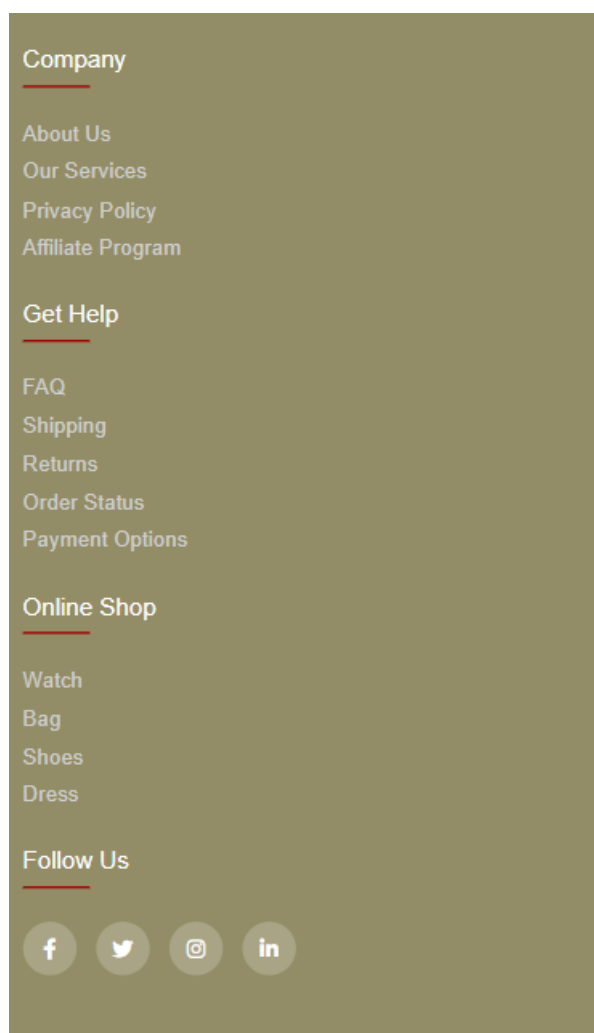


Рисунок 2.4 – адаптивная версия для мобильных устройств  
[разработано автором]

### 3.3.3 Третья страница

Третья страница: "Музыкальный Плеер"

На третьей странице виртуального пространства, посвященного творчеству С.В. Рахманинова, предстает перед посетителями "Музыкальный Плеер". Этот раздел призван углубить музыкальное восприятие и предоставить возможность прослушивания непревзойденных произведений великого композитора.

Музыкальный Плеер:

Посетители могут наслаждаться прослушиванием музыки С.В. Рахманинова через встроенный музыкальный плеер. Этот функционал

позволяет выбирать из богатого каталога произведений, создавая персональный плейлист и настраивая воспроизведение по собственному вкусу.

Эксклюзивный Каталог:

"Музыкальный Плеер" включает в себя эксклюзивный каталог музыкальных композиций Рахманинова, предоставляя уникальную возможность погружения в атмосферу его музыкального наследия. Каждое произведение снабжено информацией о его истории, создавая более глубокое понимание контекста.

Интерактивное Управление:

Уникальная возможность управления воспроизведением - от паузы до перемотки - делает взаимодействие с музыкой Рахманинова еще более вдохновляющим. Играйте любимые композиции, создавайте собственные музыкальные путеводители и наслаждайтесь музыкальным наследием великого композитора в любое время и в любом месте.

Третья страница "Музыкального Плеера" призвана погрузить посетителей в уникальный мир музыки Рахманинова, предоставляя удобные и многофункциональные инструменты для наслаждения и исследования его творчества.

Для более детального визуального представления о третьей странице сайта, предлагаю ознакомиться с рисунком 3.1, 3.2.

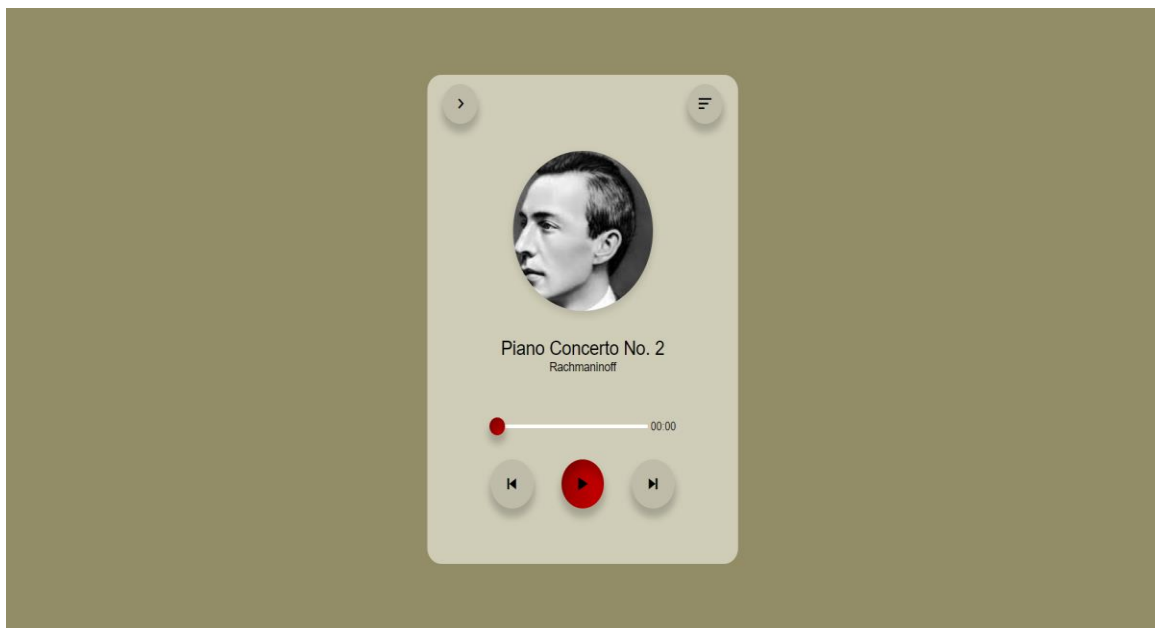


Рисунок 3.1 – Третья страница сайта (плеер)  
[разработано автором]

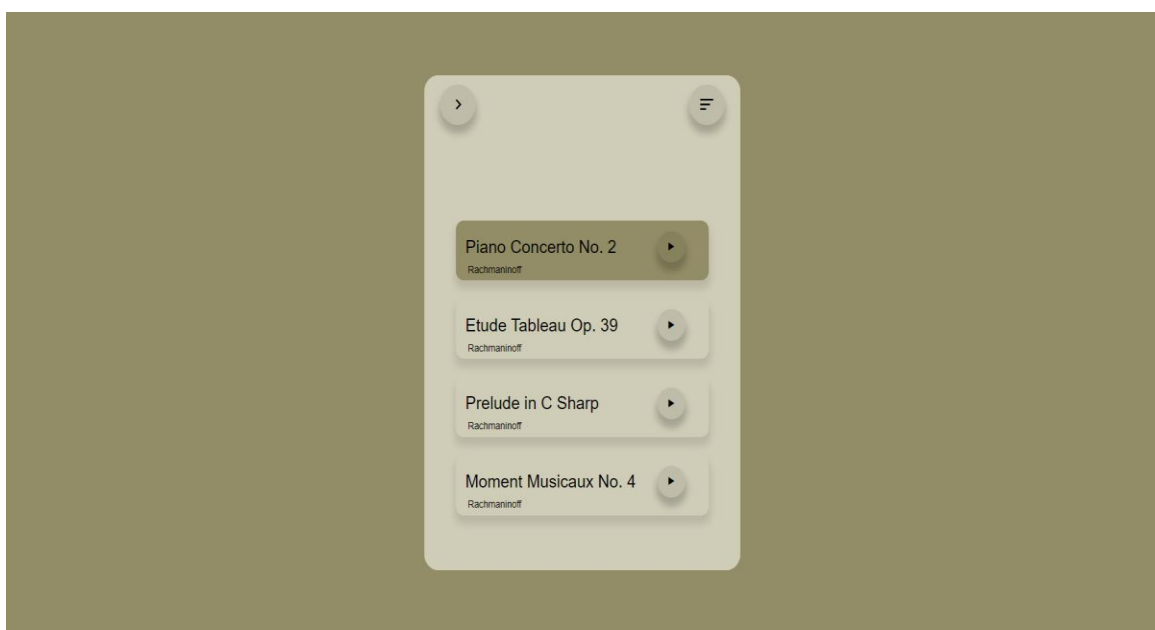


Рисунок 3.2 – Третья страница сайта (выбор пьес)  
[разработано автором]

Приведенный код листинга 6.4 представляет собой часть JavaScript страницы, ответственной за верхнюю часть сайта – его заголовок или, как в данном случае, шапку. Давайте рассмотрим основные элементы этого листинга.

Листинг 6.4 – Программный код

```

function loadTrack(track_index){
    clearInterval(updateTimer);
    reset();

    curr_track.src = music_list[track_index].music;
    curr_track.load();

    document.querySelector(".img_artist").src      =
music_list[track_index].img;
    console.log("done");
    //track_art.style.backgroundImage = "url(" +
music_list[track_index].img + ")";
    song_title.innerHTML                          =
music_list[track_index].name;
    //track_name.textContent                    =
music_list[track_index].name;
    artist_name.innerHTML                       =
music_list[track_index].artist;
    //track_artist.textContent                 =
music_list[track_index].artist;
    //now_playing.textContent = "Playing music " +
(track_index + 1) + " of " + music_list.length;

    var children = playlist.children;
    Array.from(children).forEach(child          =>
child.classList.remove('playing'));

    if(track_index === 0){
        first.classList.add('playing');
    } else if(track_index === 1){
        second.classList.add('playing');
    } else if(track_index === 2){
        third.classList.add('playing');
    } else if(track_index === 3){
        fourth.classList.add('playing');
    }

    updateTimer = setInterval(setUpdate, 500);

    curr_track.addEventListener('ended',
nextTrack);
}

```

Код в листинге 6.4 представляет функцию `loadTrack(track_index)`, ответственную за загрузку и воспроизведение нового музыкального трека при смене индекса трека. Вот пошаговое объяснение каждой части кода:

Остановка предыдущего трека и сброс таймера обновления:

`clearInterval(updateTimer);`: Этот код останавливает таймер обновления, если он был предварительно запущен, чтобы избежать конфликтов при загрузке нового трека.

`reset();`: Функция `reset()` может выполнять дополнительные действия для сброса состояния при смене трека.

Загрузка нового трека:

`curr_track.src = music_list[track_index].music;`: Устанавливает источник (`src`) текущего трека из массива `music_list` по указанному индексу.

`curr_track.load();`: Загружает трек для проигрывания.

Обновление информации о треке и изображении исполнителя:

`document.querySelector(".img_artist").src = music_list[track_index].img;`: Устанавливает источник изображения исполнителя для отображения на веб-странице.[8]

`song_title.innerHTML = music_list[track_index].name;`: Обновляет заголовок трека.

`artist_name.innerHTML = music_list[track_index].artist;`: Обновляет имя исполнителя.

Выделение текущего трека в плейлисте:

`var children = playlist.children;`: Получает все дочерние элементы плейлиста.

`Array.from(children).forEach(child => child.classList.remove('playing'));`: Удаляет класс `'playing'` у всех элементов плейлиста.

Блок условий добавляет класс `'playing'` для текущего трека в плейлисте, чтобы выделить его визуально.

Запуск таймера обновления и добавление обработчика события `'ended'`:

`updateTimer = setInterval(setUpdate, 500);`: Устанавливает таймер обновления, вызывая функцию `setUpdate` каждые 500 миллисекунд.

`curr_track.addEventListener('ended', nextTrack);`: Устанавливает обработчик события 'ended', который вызывает функцию `nextTrack` при завершении воспроизведения текущего трека.

Функция `loadTrack` предоставляет механизм для динамического обновления веб-страницы с новой информацией о треке и его визуальным представлением при смене индекса трека.

### **3.3.4 Четвёртая страница**

Четвёртая страница: "Видео Галерея"

На четвёртой странице интернет-ресурса, посвященного С.В. Рахманинову, предстаёт перед посетителями увлекательная "Видео Галерея". Этот раздел предлагает уникальную возможность окунуться в визуальное восприятие музыкального наследия композитора через видео материалы.

Видео Плеер:

Посетители могут наслаждаться увлекательными видео материалами, посвященными творчеству С.В. Рахманинова, с помощью встроенного видео плеера. Этот функционал предоставляет возможность просмотра интересных моментов из жизни и творчества композитора.

Богатство Визуального Материала:

"Видео Галерея" включает в себя разнообразные видео ролики, посвящённые пьесам Рахманинова. Посетители получают уникальную возможность погрузиться в визуальный мир великого композитора.

Интерактивное Управление:

Плеер обеспечивает удобное управление видео воспроизведением, позволяя посетителям наслаждаться каждым моментом материалов. Возможность управления громкостью, перемоткой и переключением видео создает комфортное просмотренное впечатление.

Четвёртая страница "Видео Галереи" является дополнительным источником визуального погружения в творчество композитора. Она

приглашает посетителей на увлекательное путешествие через зрелищные моменты, раскрывающие многогранный облик и вклад С.В. Рахманинова в музыкальное искусство.

Для более детального визуального представления о четвёртой странице сайта, предлагаю ознакомиться с рисунком 4.1.

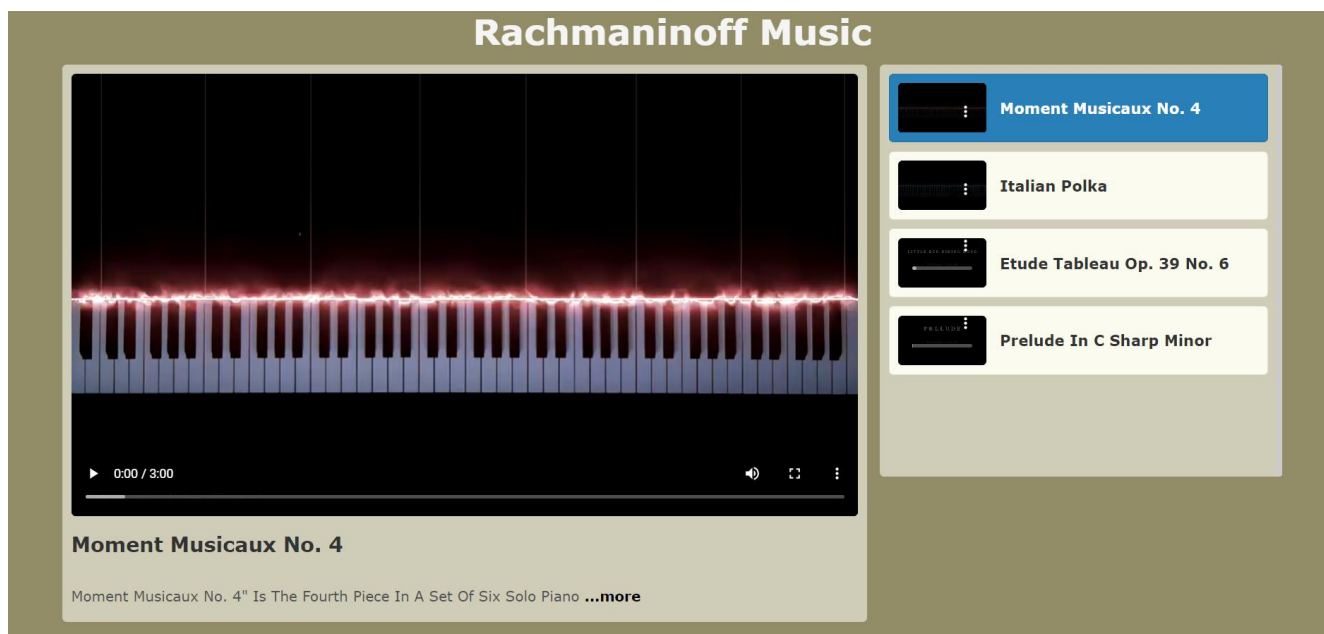


Рисунок 4.1 – Четвёртая страница сайта (видеоплеер)  
[разработано автором]

Дополнительно, была создана адаптивная версия для мобильных устройств. Для более детального визуального представления ознакомиться с рисунком 4.2.



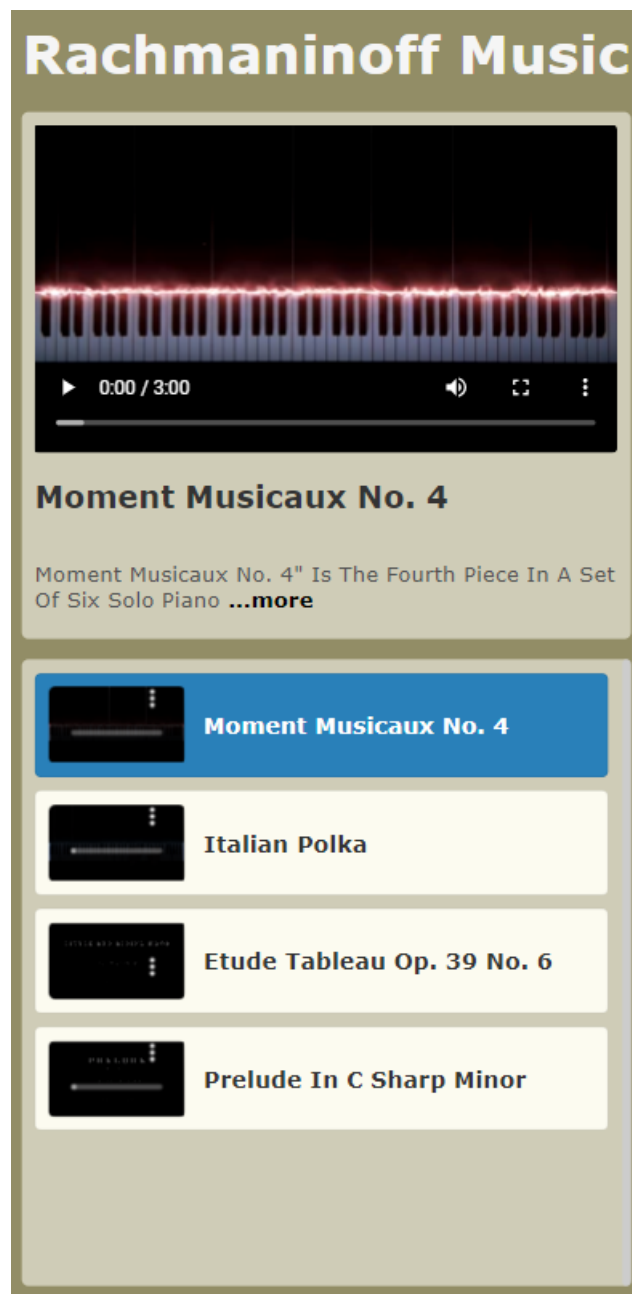


Рисунок 4.2 – адаптивная версия для мобильных устройств  
[разработано автором]

Приведенный код листинга 6.5 представляет собой часть JavaScript страницы, ответственной за верхнюю часть сайта – его заголовок или, как в данном случае, шапку. Давайте рассмотрим основные элементы этого листинга.

#### Листинг 6.5 – Программный код

```
function truncateText(element, maxLength) {
    let originalText = element.textContent;
```

```

        let truncatedText = originalText.slice(0,
maxLength) + " <a href='#' class='expansion' data-
action='more'>...more</a>";
        element.innerHTML = truncatedText;

        element.addEventListener('click', function (e)
{
            e.preventDefault();

            if
(e.target.classList.contains('expansion')) {
                if (e.target.dataset.action === 'more')
{
                    element.innerHTML = originalText +
" <a href='#' class='expansion' data-action='less'>show
less</a>";
                } else {
                    element.innerHTML = truncatedText;
                }
            }
        });
    }

    truncateText(description, 70);

```

Код в листинге 6.5 представляет собой функцию `truncateText`, которая используется для обрезания текста внутри HTML-элемента и добавления функционала "показать больше" / "показать меньше". Вот пошаговое объяснение каждой части кода:

Сохранение оригинального текста:

`let originalText = element.textContent;` Сохраняет оригинальный текст элемента в переменной `originalText`.

Обрезка текста и добавление ссылки "показать больше":

`let truncatedText = originalText.slice(0, maxLength) + " <a href='#' class='expansion' data-action='more'>...more</a>";` Создает новую строку, в которой оригинальный текст обрезается до указанной длины (`maxLength`), а после добавляется ссылка "показать больше" с классом `expansion` и атрибутом `data-action='more'`.<sup>[9]</sup>

Замена текста в элементе:

`element.innerHTML = truncatedText`;: Заменяет содержимое HTML-элемента (`element`) на обрезанный текст с добавленной ссылкой "показать больше".

Добавление обработчика событий для ссылки:

`element.addEventListener('click', function (e) {...})`;: Устанавливает обработчик события клика для элемента.

`e.preventDefault()`;: Предотвращает стандартное действие ссылки (переход по ссылке).[10]

`if (e.target.classList.contains('expansion')) {...}`: Проверяет, был ли клик на элементе с классом `expansion`. [11]

`e.target.dataset.action`: Получает значение атрибута `data-action` кликнутого элемента.

Изменение содержимого элемента при клике:

Если кликнутый элемент имеет класс `expansion` и значение атрибута `data-action` равно `'more'`, то текст элемента заменяется на полный текст с ссылкой "показать меньше". Если значение атрибута `data-action` равно `'less'`, то текст элемента возвращается к обрезанной версии с ссылкой "показать больше".

Функция `truncateText` обеспечивает удобный способ управления отображением текста, позволяя пользователю динамически раскрывать или скрывать контент в зависимости от его предпочтений.

### **3.3.5 Пятая страница**

Пятая страница: "Биография С.В. Рахманинова"

На пятой странице интернет-ресурса, посвященного великому композитору С.В. Рахманинову, посетителям предоставляется уникальная возможность погрузиться в богатую и увлекательную биографию этого выдающегося музыкального гения.

Жизненный Путь:

Страница "Биография" представляет полный и обстоятельный обзор жизни и творчества С.В. Рахманинова, начиная с ранних лет его детства и

обучения в Московской консерватории до влиятельных моментов в его международной карьере. Посетители могут открывать для себя ключевые события, формирующие личность композитора, а также влияние его семьи и времени, в которое он жил.

Страница "Биография С.В. Рахманинова" представляет собой ценный ресурс для всех, кто стремится лучше понять личность и вклад в историю мировой музыки этого великого композитора.

Для более детального визуального представления о пятой странице сайта, предлагаю ознакомиться с рисунком 5.1.

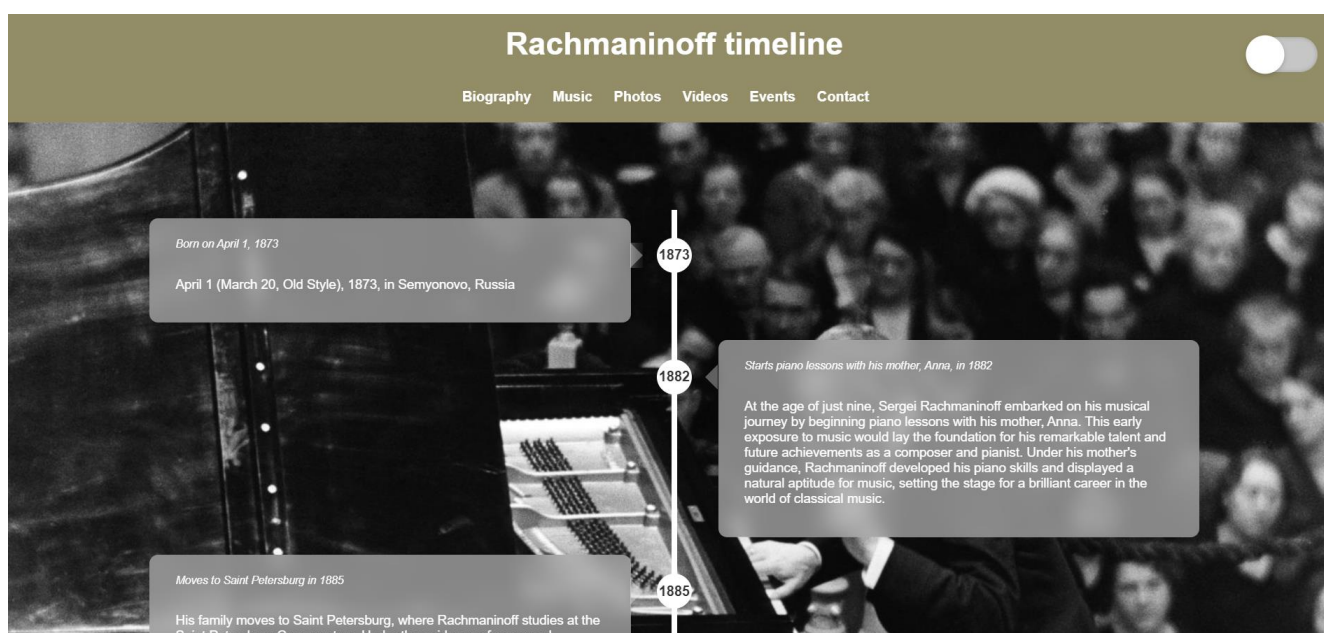


Рисунок 5.1 – Пятая страница сайта (видеоплеер)

[разработано автором]

Дополнительно, была создана адаптивная версия для мобильных устройств. Для более детального визуального представления ознакомиться с рисунком 5.2.



Рисунок 5.2 – адаптивная версия для мобильных устройств  
[разработано автором]

### 3.4 Создание межстраничной навигации

На каждой странице веб-ресурса была внедрена система навигации, обеспечивающая переходы к другим разделам. На первой странице данная функциональность была реализована через использование навигационной панели (nav-bar), а также кнопок, направляющих посетителя к ключевым страницам сайта, как показано на рисунках 6.1 и 6.2.

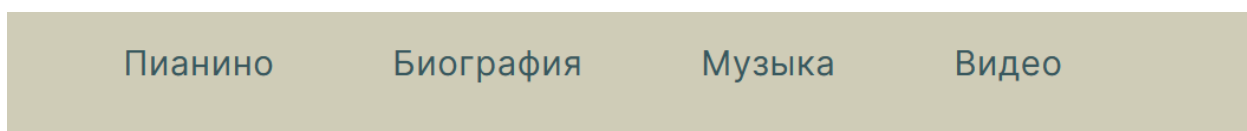


Рисунок 6.1 – Навигация с помощью nav-бара  
[разработано автором]

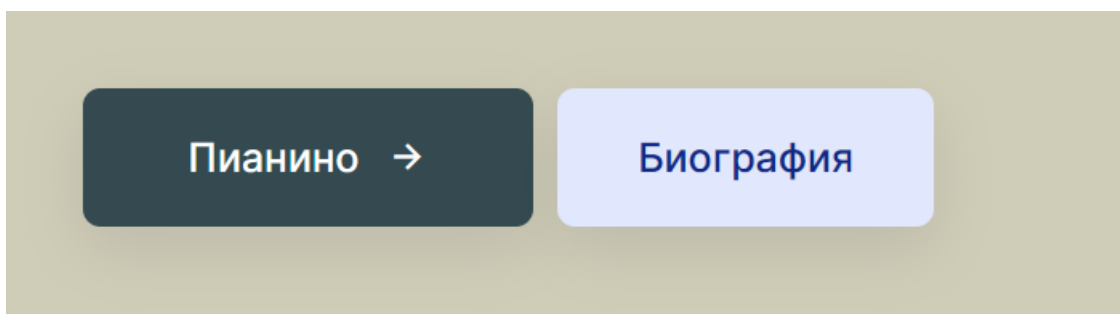


Рисунок 6.2 – Навигация с помощью кнопок  
[разработано автором]

Аналогичным образом, на второй и пятой страницах также была внедрена навигация с использованием технологии nav-bar. Подробности представлены на рисунках 6.3 и 6.4.

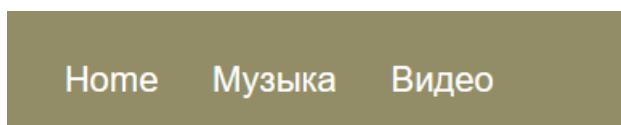


Рисунок 6.3 – Навигация с помощью nav-бара  
[разработано автором]

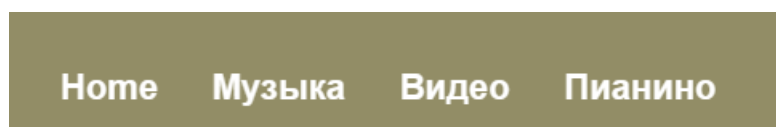


Рисунок 6.4 – Навигация с помощью nav-бара  
[разработано автором]

### **3.5 Реализация слоя клиентской логики веб-страниц с применением технологии JavaScript**

В рамках разработки интернет-ресурса "150 лет С.В. Рахманинову" успешно реализован слой клиентской логики с использованием технологии JavaScript. Этот функциональный аспект внес существенный вклад в

обогащение пользовательского опыта, предоставив интерактивные элементы на различных страницах сайта.

Одним из выдающихся достижений внедрения клиентской логики стало создание работающего веб-пианино. Эта функция позволяет пользователям взаимодействовать с музыкальными элементами, создавая уникальные мелодии прямо на веб-странице. Каждая клавиша интегрирована с музыкальным звучанием, что придает сайту дополнительный музыкальный элемент.

Кроме того, успешно реализован музыкальный плеер, предоставляющий возможность прослушивания разнообразных музыкальных произведений С.В. Рахманинова. Этот элемент поддерживает удобные функции управления воспроизведением, что дополняет общий опыт посетителей сайта.

Дополнительно, веб-ресурс обогатился видеоплеером, предоставляющим возможность просмотра визуального контента, связанного с жизнью и творчеством композитора. Это позволяет пользователям глубже погружаться в атмосферу истории Рахманинова, используя не только звуковые, но и визуальные элементы.

Таким образом, реализация слоя клиентской логики с применением JavaScript эффективно дополнила функциональность интернет-ресурса, создавая интерактивные и привлекательные элементы, которые обогащают взаимодействие пользователя с контентом сайта.

## 6. ЗАКЛЮЧЕНИЕ

В ходе выполнения данной курсовой работы была проведена комплексная разработка клиентской части интернет-ресурса "150 лет С.В. Рахманинову". Проект успешно реализован с использованием базовых технологий веб-разработки, таких как HTML, CSS и JavaScript, что обеспечило не только эффективность, но и широкую доступность ресурса.

Одним из ключевых моментов разработки стал анализ предметной области, который выявил основные аспекты жизни и творчества С.В. Рахманинова. Это позволило структурировать информацию на веб-ресурсе, делая его привлекательным и информативным для посетителей.

Вопрос выбора технологий разработки был решен в пользу HTML, CSS и JavaScript. Этот выбор обеспечил проект стабильностью, производительностью и удобством сопровождения в будущем. Кроме того, на каждой странице ресурса была реализована эффективная навигация, а использование технологии nav-bar дополнительно обеспечило легкость перемещения между разделами.

Реализация слоя клиентской логики с применением JavaScript стала ключевым моментом в обогащении пользовательского опыта. Веб-пианино, музыкальный плеер и видеоплеер успешно внедрены, предоставляя посетителям уникальную возможность взаимодействия с музыкальным и визуальным контентом.

Программный код и клиентская часть интернет-ресурса были размещены на бесплатных хостингах, обеспечивая открытый доступ к результатам работы. Проект доступен по следующим гиперссылкам:

[AlAdNi0806/skhryeder.github.io](https://AlAdNi0806/skhryeder.github.io)

[Рахманинов \(aladni0806.github.io\)](https://aladni0806.github.io)

В целом, разработанный интернет-ресурс успешно сочетает в себе информативность, интерактивность и удобство использования, делая его привлекательным для широкой аудитории. Предложенные в ходе работы



решения и технологический подход могут служить основой для разработки подобных проектов в будущем.

## 7. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Tabarés R. HTML5 and the evolution of HTML; tracing the origins of digital platforms //Technology in Society. – 2021. – Т. 65. – С. 101-529.
2. Vogel L., Springer T. Speed Up the Web with Universal CSS Rendering //International Conference on Web Engineering. – Cham : Springer Nature Switzerland, 2023. – С. 191-205.
3. Fouquet R., Laperdrix P., Rouvoy R. Breaking Bad: Quantifying the Addiction of Web Elements to JavaScript //ACM Transactions on Internet Technology. – 2023. – Т. 23. – №. 1. – С. 1-28.
4. Asiri S. et al. A Survey of Intelligent Detection Designs of HTML URL Phishing Attacks //IEEE Access. – 2023.
5. Prazina I. et al. Methods for Automatic Web Page Layout Testing and Analysis: A Review //IEEE Access. – 2023.
6. Saymeh M. A., Al-Sayyed M. J., Arandi S. Colors Sorting System (CSS). – 2023.
7. Malavolta I. et al. JavaScript Dead Code Identification, Elimination, and Empirical Assessment //IEEE Transactions on Software Engineering. – 2023.
8. Kereki F. Mastering JavaScript Functional Programming: Write clean, robust, and maintainable web and server code using functional JavaScript and TypeScript. – Packt Publishing Ltd, 2023.
9. McManus S. Web design in easy steps. – In Easy Steps Limited, 2023.
10. Lizcano D. et al. End-user modeling of quality for web components //Journal of Software: Evolution and Process. – 2023. – Т. 35. – №. 3. – С. e2256.
11. Sotnik S., Shakurova T., Lyashenko V. Development Features Web-Applications. – 2023.

Все источники были использованы в процессе написания данного текста и предоставляют информацию о HTML, CSS, JavaScript, а также о популярных библиотеках и фреймворках для разработки веб-приложений.

## 8.ПРИЛОЖЕНИЕ А

Проект доступен по следующим гиперссылкам:

[AlAdNi0806/skhryeder.github.io](https://github.com/AlAdNi0806/skhryeder.github.io)

[Рахманинов \(aladni0806.github.io\)](https://github.com/AlAdNi0806/skhryeder.github.io)