МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ   
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность 1–40 01 01 «Информационные системы и технологии»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КУРСОВОГО ПРОЕКТА**

по дисциплине «Компьютерные языки разметки»

ТемаВеб-сайт «Онлайн-кинотеатра»

**Исполнитель**

студент 1 курса 2 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.А. Шункевич

подпись, дата

**Руководитель**

Ассистент \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. В. Харланович

должность, учен. степень, ученое звание подпись, дата

Допущен(а) к защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата, подпись

Курсовой проект защищен с оценкой

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. В. Харланович

подпись дата инициалы и фамилия

**Содержание**

[Введение 4](#_Toc166710892)

[1 Постановка задачи 5](#_Toc166710893)

[1.1 Обзор аналогичных решений 5](#_Toc166710894)

[1.2 Техническое задание 6](#_Toc166710895)

[1.3 Обзор технологий и программных средств 7](#_Toc166710896)

[1.4 Вывод 7](#_Toc166710897)

[2 Проектирование страниц веб-сайта 8](#_Toc166710898)

[2.1 Выбор способа верстки 8](#_Toc166710899)

[2.2 Выбор стилевого оформления 8](#_Toc166710900)

[2.3 Выбор шрифтового оформления 8](#_Toc166710901)

[2.4 Разработка логотипа 8](#_Toc166710902)

[2.5 Разработка пользовательских элементов 9](#_Toc166710903)

[2.6 Разработка спецэффектов 9](#_Toc166710904)

[2.7 Вывод 11](#_Toc166710905)

[3 Реализация проекта 13](#_Toc166710906)

[3.1 Структура HTML-документа 13](#_Toc166710907)

[3.2 Добавление таблиц стилей CSS 15](#_Toc166710908)

[3.3 Использование стандартов SVG 17](#_Toc166710909)

[3.4 Управление элементами DOM 17](#_Toc166710910)

[3.5 Вывод 20](#_Toc166710911)

[4 Тестирование 21](#_Toc166710912)

[4.1 Адаптивный дизайн сайта 21](#_Toc166710913)

[4.2 Кроссбраузерное тестирование 22](#_Toc166710914)

[4.3 Валидность 24](#_Toc166710915)

[4.4 Руководство пользователя 24](#_Toc166710916)

[4.5 Выводы 25](#_Toc166710917)

[Заключение 26](#_Toc166710918)

[Список использованных источников 27](#_Toc166710919)

[Приложение А 28](#_Toc166710920)

[Приложение Б 33](#_Toc166710921)

[Приложение В 36](#_Toc166710922)

[Приложение Г 47](#_Toc166710923)

[Приложение Д 50](#_Toc166710924)

[Приложение Е 51](#_Toc166710925)

# Введение

Сегодня веб-сайт онлайн-кинотеатра — это не просто информационный ресурс, а полноценный сервис, который позволяет зрителям смотреть фильмы и сериалы онлайн на различных устройствах, таких как компьютеры, смартфоны, планшеты, смарт-телевизоры и игровые консоли, что делает процесс просмотра более гибким. Онлайн-кинотеатры предлагают огромный выбор фильмов и сериалов различных жанров и стран производства.

Разработка веб-сайта онлайн-кинотеатра — это не только техническая задача, но и творческий процесс, который требует внимания к деталям и учета современных тенденций веб-дизайна. Один из ключевых моментов — это создание удобного и интуитивно понятного интерфейса, который позволит зрителям быстро и легко найти нужную информацию и выполнить нужное действие.

Веб-сайт онлайн-кинотеатра должен быть адаптивным, т.е. корректно отображаться на различных устройствах и экранах, от компьютеров до мобильных телефонов. Это особенно важно, учитывая то, что все больше людей используют мобильные устройства для поиска информации и покупки билетов на киносеансы.

Разработка веб-сайта онлайн-кинотеатра — это сложная задача, требующая знания не только программирования, но и маркетинга и бизнес-анализа. Хороший веб-сайт должен не только обеспечивать удобство использования и безопасность, но и соответствовать бизнес-стратегии компании и увеличивать ее доходы.

Цель курсового проекта: разработать веб-сайт сети кинотеатров с использованием HTML5 и XML, а также с применением SCSS/CSS3 и JS.

Задачи курсового проекта:

* Проанализировать существующие языки разметки, инструменты и библиотеки для создания веб-сайта.
* Разработать макеты и прототипы веб-страниц.
* Разработать структуру веб-сайта.
* Наполнить веб-сайт информацией по теме.
* Протестировать веб-сайт.

# Постановка задачи

# 1.1 Обзор аналогичных решений

При создании сайта онлайн-кинотеатра важно использовать различные аналитические решения, которые помогают оптимизировать сайт и улучшить его удобство использования. Они позволяют учитывать потребности пользователей и достигать лучших результатов в поисковых системах. Поэтому крайне важно проанализировать сайты конкурентов. На основе анализа будет создан собственный проект.

В качестве основного аналога был выбран «Start». Заходя на главную страницу, внимание привлекают афиши новых фильмов, которые занимают значимую часть экрана (рис. 1.1)



Рисунок 1.1 – главная страница Start

Сразу на главной странице мы видим название сайта, навигационное меню, информация об условия оформления подписки.

Редакторы сайта придерживаются одного стиля во всех вкладках страницы, наполняя их максимумом информации необходимой для пользователя.

Это и стало одним из плюсов этого веб-сайта: пользователю дают всю необходимую ему информацию сразу, что делает сайт приятным и легким для пользования.

Так же на сайте присутствует раздел, где расположены контактные данные, способы оплаты , рубрика вопросы и ответы, ссылки на социальные сети (рис. 1.2).

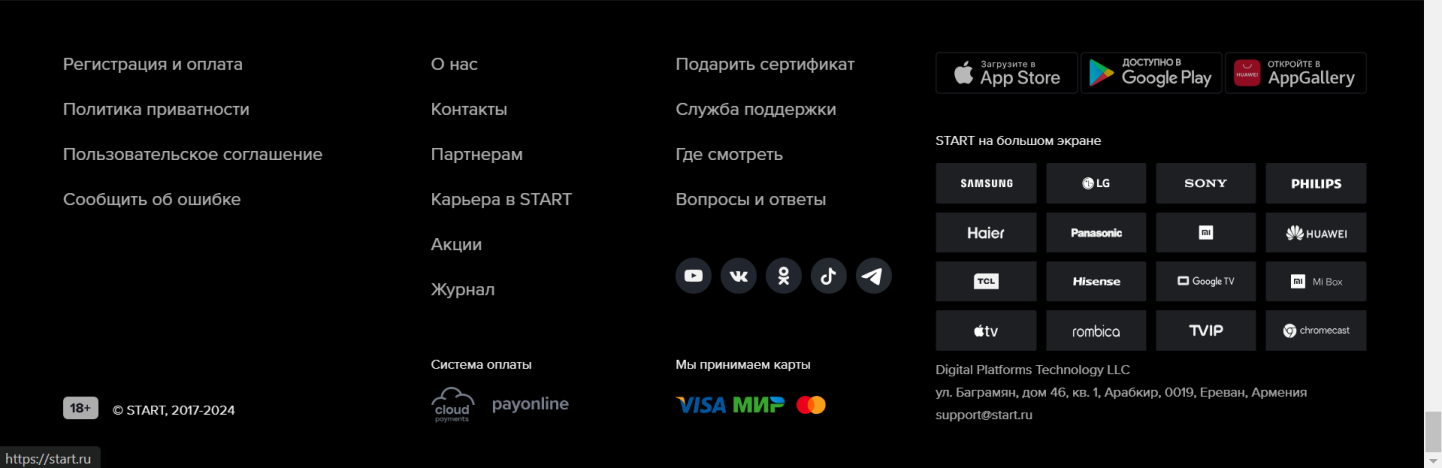


Рисунок 1.2 – footer start.ru

Следующим аналогом стал сайт Кинопоиск. На главной странице сайта представлены различные категории контента, чтобы помочь пользователям легко найти интересующие их материалы (рис. 1.3).

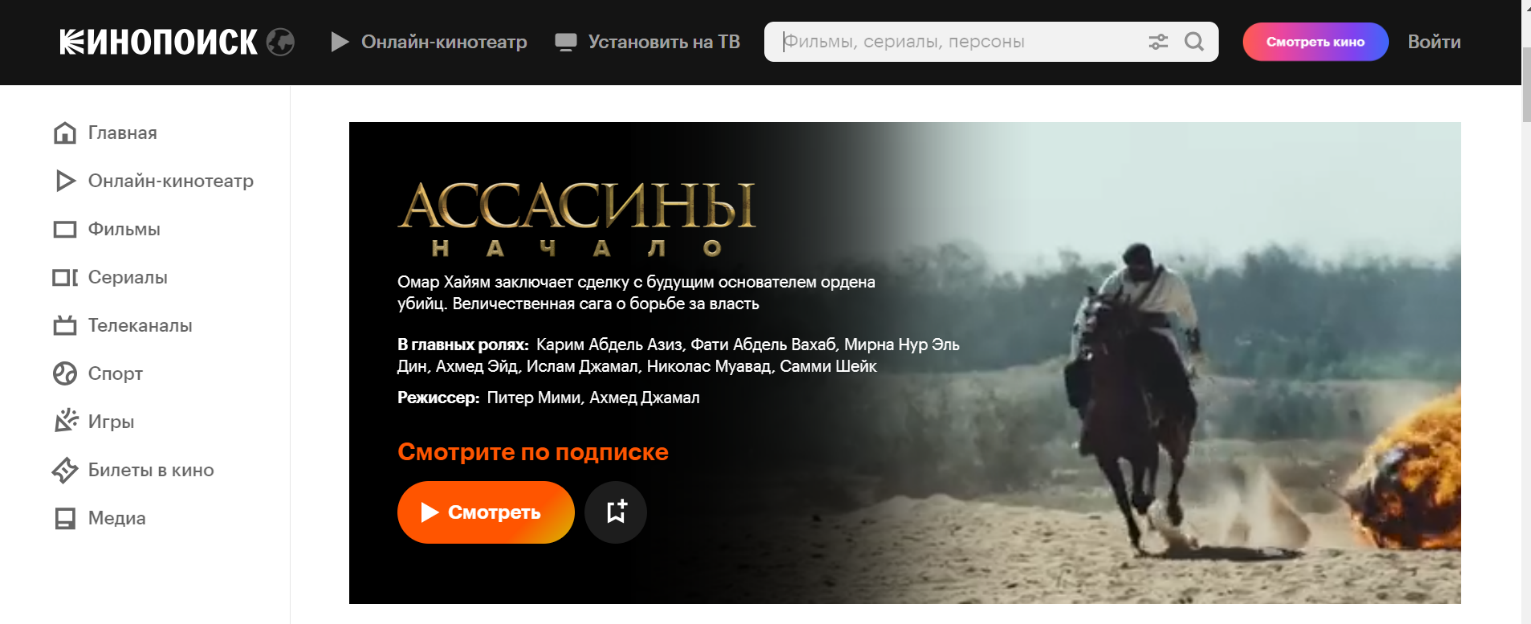


Рисунок 1.3 – главная страница Кинопоиск

Дизайн сайта является адаптивным, что позволяет пользователям просматривать его на любом устройстве. Сайт использует яркие и контрастные цвета, качественные изображения и графику, чтобы привлечь внимание пользователей. Кроме того, дизайн сайта усилен функциональностью, такой как функция поиска по сайту и оценки качества услуг.

## 1.2 Техническое задание

Поставлена цель создать многостраничный веб-сайт онлайн-кинотеатра, на котором каждый желающий может смотреть фильмы и сериалы онлайн на различных устройствах. Сайт должен быть интуитивно понятным в использовании, удобным для просмотра и ознакомления с информацией.

Меню навигации по сайту на всех страницах будет располагаться сверху. В нем будут предоставляться ссылки на основные разделы сайта (главная, каталог, подписка, вход). В самом низу расположен подвал с контактной информацией.

На странице «Главная» будет располагаться слайдер с афишами новых фильмов и сериалов. Будет представлена информация о преимуществах подписки.

На странице «Каталог» будут представлены категории контента (фильмы, сериалы, детям) и рейтинг лучших фильмов 2023 года

На страницах «Фильмы», «Сериалы», «Детям» будет представлен ассортимент фильмов и сериалов. Для одного фильма/сериала из каждой категории будет создана страница с кратким описанием, трейлером, рейтингом, информацией о режиссере, композиторе, годе производства, продолжительности и жанре. Также на этих страницах будет отображаться фильм, который будет воспроизводится только после авторизации пользователя и оформления подписки. Обложки фильмов и их названия должны храниться в XML формате.

В мобильной версии (ширина до 480px) страницы должны адаптироваться под ширину устройства и должно быть реализовано бургер-меню. Начиная с ширины в 480px должна открываться версия для персонального компьютера.

Основные требования к исполнителю: создание дизайна, проведение анализа существующих решений, создание прототипа, адаптивность и кроссбраузерность веб-сайта (корректное отображение в браузерах, таких как Google Chrome, Mozilla Firefox, Microsoft Edge, Opera.). В курсовом проекте используется гипертекстовый язык разметки HTML, CSS/SCSS, графические элементы в формате SVG; данные хранятся в XML-формате.

## 1.3 Обзор технологий и программных средств

Для создания макета сайта был использован сервис Figma (онлайн-сервис для дизайнеров, веб-разработчиков и маркетологов. Он предназначен для создания прототипов сайтов или приложений, иллюстраций и векторной графики).

Для управления элементами DOM и манипуляцией с данными будет использован JavaScript. Структура веб-сайта создана с помощью языка разметки HTML, дизайн веб-сайта оформлен с помощью CSS/SCSS. В XML формате хранятся данные.

Курсовой проект выполняется в редакторе кода Visual Studio Code.

## 1.4 Вывод

В данном разделе были рассмотрены аналогичные решения, были проанализированы их недостатки и преимущества, для создания в последующем качественного веб-сайта. Были поставлены конкретные задачи для создания интерактивного информационного ресурса. Кроме того, были рассмотрены средства реализации программного продукта, такие как языки разметки, а также редактор кода VS Code. Были проанализированы особенности языков и основные дополнительные средства редактора. Все вышеперечисленные пункты понадобятся для достижения поставленной задачи и для создания в дальнейшем качественного продукта.

# 2 Проектирование страниц веб-сайта

## 2.1 Выбор способа верстки

На сайте будет использоваться flex-box верстка, которая предоставляет простую и интуитивно понятную модель размещения элементов внутри контейнера. С помощью flexbox можно создавать адаптивные дизайны, а также улучшать доступность и удобство использования сайта для пользователей на различных устройствах и экранах. Flexbox предоставляет мощные инструменты для выравнивания и распределения элементов внутри контейнера по горизонтали и вертикали. Это позволяет нам создавать эстетически приятные и сбалансированные макеты. Также он хорошо поддерживается современными браузерами, что обеспечивает надежность и совместимость верстки.

## Выбор стилевого оформления

Выбор цветовой гаммы — важный этап разработки проекта. Цвета должны не только сочетаться между собой, но и отражать тему проекта.

Для данного веб-сайта была выбрана достаточно нейтральная цветовая палитра (рис. 2.1). Фиолетовые оттенки были выбраны для сайта из-за их ассоциаций с роскошью и креативностью, их уникальности, хорошей сочетаемости с другими цветами и возможности усиления брендирования. (рис. 2.1).

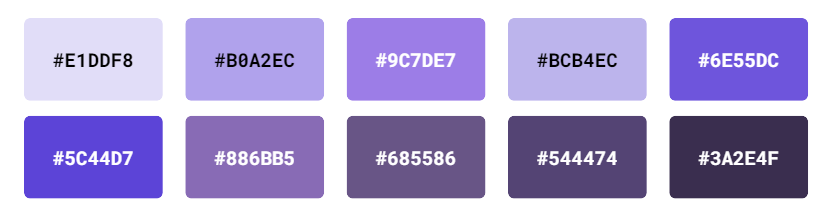


Рисунок 2.1 – Основные цвета сайта

## Выбор шрифтового оформления

В данном проекте будут использованы один основной шрифт: Century Gothic. О подключался с помощью библиотеки Google Fonts. Данный шрифт поддерживает кириллическое написание и не конфликтует по стилевым характеристикам.

## Разработка логотипа

Для разработки логотипа сайта было принято решение использовать минималистичный дизайн. Он представляет из себя ключевое слово «VOCHKI», в котором буква «О» заменена на минималистичный глаз (рис. 2.2). Это поможет создать простой и запоминающийся логотип, который будет отражать сферу деятельности. Такой логотип будет легко узнаваемым и привлекательным для целевой аудитории сайта.



Рисунок 2.2 – Логотип сайта

## Разработка пользовательских элементов

Пользовательскими элементами выступают навигационное меню (рис. 2.3), находящееся в шапке сайта, кроме того, необходимо уделять внимание удобству использования элементов на разных устройствах — они должны отображаться корректно и быть легко доступными как на компьютере, так и на мобильном устройстве, поэтому навигационное меню трансформируется в меню-бургер в зависимости от ширины экрана. Также на сайте имеется кнопка для перехода в начало страницы, чтобы пользователю было удобнее ориентироваться по сайту. На главной странице будет использоваться слайдер (рис. 2.4). Он может улучшить пользовательский опыт и повысить визуальную привлекательность страницы.



Рисунок 2.3 – Навигационное меню в шапке страницы



Рисунок 2.4 – Фрагмент слайдера

## Разработка спецэффектов

Веб-сайт будет улучшен с помощью создания различных эффектов и анимаций, которые добавят интерактивности и привлекательности для пользователей. Эти изменения позволят сделать веб-сайт более интересным и привлекательным для посетителей.

На главной странице будет присутствовать анимация подчеркивания выбранного элемента меню при наведении курсора мыши (рис. 2.5, рис. 2.6).



Рисунок 2.5 – Элемент меню до наведения курсора мыши



Рисунок 2.6 – Элемент меню после наведения курсора мыши

В мобильной версии сайта присутствует анимация иконки бургер-меню, при нажатии она изменяется (рис. 2.7, рис. 2.8).



Рисунок 2.7 – Иконка бургер-меню до нажатия



Рисунок 2.8 – Иконка бургер-меню после нажатия

В каталоге в разделе сериалы будет присутствовать анимация – при наведении на сериал появляется окно с краткой информацией о нем, а также при наведении на кнопку «Смотреть» она будет менять свой цвет с серого на фиолетовый (рис. 2.9, рис. 2.10)



Рисунок 2.9 – Сериал до наведения

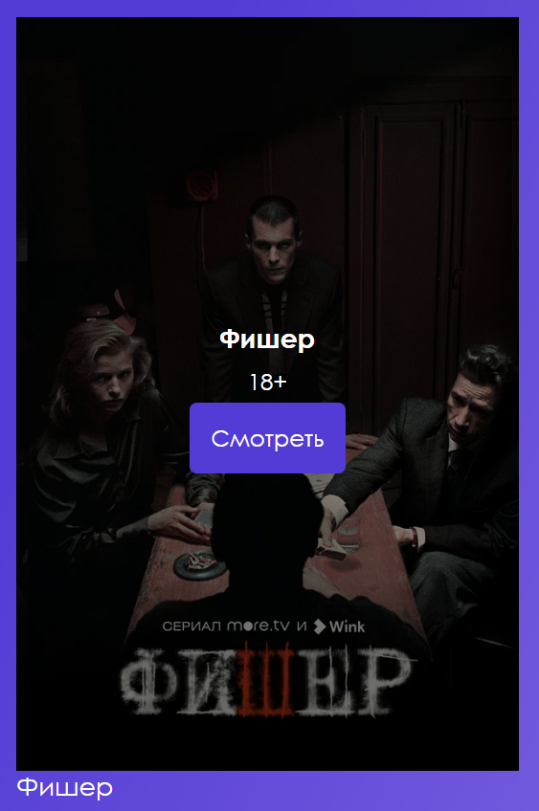


Рисунок 2.9 – Сериал до наведения

## Вывод

На данном этапе был создан макет веб-сайта для облегчения последующей верстки. Было упорядочено расположение всех элементов на страницах (навигационного меню, логотипа, основного контента каждой страницы, расположение картинок, подвала веб-сайта). Было определено стилевое оформление веб-сайта, то есть основная цветовая гамма, шрифтовое оформление, элементы пользовательского интерфейса, а также эффекты и переходы, придающие динамики и привлекательности страницам.

# 3 Реализация проекта

## 3.1 Структура HTML-документа

Структура HTML-документа выполнена по стандарту HTML-5. В листинге 3.1 представлена декларация документа и основное содержимое тега head.

 <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<link rel="stylesheet" href="./css/header.css">

<link rel="stylesheet" href="./css/style.css" />

<link rel="stylesheet" href="./css/test.css" />

<link rel="stylesheet" href="./css/scrollToTop.css">

<link rel="stylesheet" href="./css/burger.css">

<link rel="stylesheet" href="./css/footer.css">

<link

rel="stylesheet"

href="https://cdn.jsdelivr.net/npm/swiper@11/swiper-bundle.min.css"

/>

<title>Main</title>

</head>

Листинг 3.1 – Декларация и тег head

Элемент <!DOCTYPE> предназначен для указания типа текущего документа.  Атрибут lang тега html устанавливает язык документа. Метатеги, расположенные в теге head устанавливают текущую кодировку, область просмотра и её начальное масштабирование. Тег title устанавливает название страницы. В тегах link происходит подключение каскадных таблиц стилей и стилей слайдера.

При построении структуры тела документа были использованы семантические теги: header, main, footer, section и nav.

На странице располагается навигационная панель, которая зафиксирована на странице. Структура панели представлена в листинге 3.2

<header>

<div class="header">

<div class="container">

<div class="menu">

<img src="./icons/logo.png" />

<nav class="nav">

<ul>

<li class="nav\_\_item"><a style="display: inline-block; border-bottom: 3px solid #3D277D;" class="nav\_\_link" href="">Главная</a></li>

<li class="nav\_\_item"><a class="nav\_\_link" href="catalog.html">Каталог</a></li>

<li class="nav\_\_item"><a class="nav\_\_link" href="#subscribe">Подписка</a></li>

</ul>

</nav>

<div class="burger">

<span></span>

</div>

<img src="./icons/button.svg" class="button-enter" />

</div>

</div>

</div>

</header>

Листинг 3.2. – Навигационная панель

Данная навигационная панель представляет из себя маркированный список с ссылками на основные страницы веб-сайта.

В теге header также располагается логотип сайта.

Шапка веб-сайта является адаптированной под мобильные устройства. При достижении определенных размеров экрана браузера, меню скрываются, а на его месте появляется меню-бургер, содержащее ссылки как на основные разделы веб-сайта. Оно раскрывается при нажатии на иконку меню, которая “нарисована” при помощи тега span и псевдоэлементов after и before. При повторном нажатии меню закрывается.

В теге footer также содержится меню для связи и контактная информация(листинг 3.3).

<footer>

<div class="container">

<img src="./icons/line.svg">

<div class="contact">

<nav>

<ul>

<li><a href="index.html">Главная</a></li>

<li><a href="catalog.html">Каталог</a></li>

<li><a href="subscribe.html">Подписка</a></li>

<li><a href="#">Служба поддержки</a></li>

</ul>

</nav>

</div>

<div class="icons">

<nav>

<div class="telegram">

<img src="./icons/telegram.png" alt="" />

</div>

<div class="instagram">

<img src="./icons/instagram.png" alt="" />

</div>

<div class="vk">

<img src="./icons/vk.png" alt="" />

</div>

<div class="email">

<img src="./icons/email.png" alt="" />

</div>

</nav>

</div>

</div>

</footer>

Листинг 3.3 – Подвал сайта

В коде находятся несколько тегов section, каждый из которых представляет собой какую-то основную часть сайта.

# 3.2 Добавление таблиц стилей CSS

Все CSS файлы были помещены в отдельную папку css для удобства последующей работы с ними. Подключение стилей осуществлялось внешне с помощью тега link. В таблицах стилей используются селекторы по классу, по тегу, а также селекторы потомков, псевдоклассы и псевдоэлементы.

Пример использования CSS при стилизации навигационной панели представлен в листинге 3.4.

ul {

list-style: none;

display: flex;

justify-content: space-between;

}

.menu {

display: flex;

justify-content: space-between;

align-items: center;

width: 1200px;

}

.menu > img {

padding: 0 0 0 10px;

}

.button\_enter {

padding: 0 10px 0 0;

}

.nav {

width: 600px;

}

.header {

display: flex;

justify-content: center;

}

header {

padding: 10px 0 15px 0;

}

.nav\_\_link {

position: relative;

display: inline-block;

font-size: clamp(20px, 3vw, 25px);

line-height: 36px;

font-weight: 500;

color: #fff;

text-decoration: none;

}

.nav\_\_link::before {

content: "";

position: absolute;

bottom: 0;

left: 0;

height: 3px;

width: 100%;

background-color: #3D277D;

opacity: 0;

transition-property: opacity, transform;

transition-timing-function: ease-in-out;

transition-duration: 0.2s;

}

.nav\_\_item:hover .nav\_\_link::before {

transform: translateY(3px);

opacity: 1;

}

Листинг 3.4 – Использование CSS

# 3.3 Использование стандартов SVG

SVG ( Scalable Vector Graphics — масштабируемая векторная графика) — язык разметки масштабируемой векторной графики, созданный Консорциумом Всемирной паутины (W3C) и входящий в подмножество расширяемого языка разметки XML, предназначен для описания двумерной векторной и смешанной векторно/растровой графики в формате XML.

В курсовом проекте масштабируемая векторная графика (SVG) использовалась для создания кнопок и декоративных элементов. Реализация кнопки входа представлена в листинге 3.5.

<svg

width="178"

height="61"

viewBox="0 0 178 61"

fill="none"

xmlns="http://www.w3.org/2000/svg"

>

<path

fill-rule="evenodd"

clip-rule="evenodd"

d="M30.5 3H147.5C162.688 3 175 15.3122 175 30.5C175 27.0781L111.512 37.373H102.994Z"

fill="white"

/>

</svg>

Листинг 3.5 – Кнопка входа

# 3.4 Управление элементами DOM

C помощью JavaScript реализовано меню-бургер для мобильной версии (до 950px), слайдер на главной странице, кнопка для возвращения в начало страницы. В листинге 3.6 представлен код, реализующий меню-бургер. В листинге 3.7 представлена реализация кнопки для возвращения в начало страницы.

document.querySelector('.burger').addEventListener('click', function(){

this.classList.toggle('active');

document.querySelector('.nav').classList.toggle('open')

})

Листинг 3.6 – Меню-бургер

"use strict";

var sttElem = document.querySelector('.stt');

var screanHeight = window.innerHeight;

var sttScroll = function sttScroll() {

document.addEventListener('scroll', function (e) {

if (screanHeight <= window.scrollY) {

sttElem.classList.add('stt\_\_active');

} else if (e.target.scrollingElement.scrollTop <= screanHeight) {

sttElem.classList.remove('stt\_\_active');

sttElem.style.pointerEvents = 'auto';

}

});

};

var sttClick = function sttClick() {

sttElem.addEventListener('click', function () {

var docHeight = window.scrollY;

var progress = 0;

var position = docHeight;

var speed = 5;

sttElem.style.pointerEvents = 'none';

var sttAnim = function sttAnim() {

progress += 1;

position -= progress \* speed;

window.scrollTo(0, position);

if (position > 0) {

requestAnimationFrame(sttAnim);

}

};

requestAnimationFrame(sttAnim);

});

};

var sttFunc = function sttFunc() {

sttScroll();

sttClick();

};

document.addEventListener('DOMContentLoaded', sttFunc);

Листинг 3.7 – Кнопка для возвращения в начало страницы

Слайдер на главной странице был реализован с помощью библиотеки Swiper. Код представлен в листинге 3.8.

let swiper = new Swiper(".mySwiper", {

navigation: {

nextEl: ".swiper-button-next",

prevEl: ".swiper-button-prev",

},

scrollbar: {

el: ".swiper-scrollbar",

hide: true,

}

});

Листинг 3.8 – Слайдер

Также JavaScript использовался для управления видеоплеером на веб-странице. Были реализованы функции для воспроизведения видео и для его скрытия(листинг 3.9). Было использовано внутреннее подключение JavaScript.

<script>

var videoPlayer = document.getElementById("videoPlayer");

var myVideo = document.getElementById("myVideo");

function stopVideo() {

videoPlayer.style.display = "none";

}

function playVideo(file) {

myVideo.src = file;

videoPlayer.style.display = "block";

}

</script>

Листинг 3.9 – Видеоплеер

С помощью JavaScript была настроена проверка на валидность E-mail в формах для входа и оформления подписки (листинг 3.10).

function validateForm() {

var email = document.querySelector('.field[type="email"]').value;

var password = document.querySelector('.field[type="password"]').value;

if (email.trim() === "" || password.trim() === "") {

alert("Пожалуйста, заполните все поля");

return false;

}

var emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

if (!email.match(emailRegex)) {

alert("Пожалуйста, введите корректный адрес электронной почты");

return false;

}

return true;

}

Листинг 3.10 – Валидация

## 3.5 Вывод

На данном этапе была реализована структура на HTML, а также созданы таблицы стилей SCSS/CSS. Было продемонстрировано использование стандартов SVG, а также языка программирования JavaScript. Было определено стилевое оформление веб-сайта, созданы анимации, а также веб-сайт был подготовлен к следующему этапу – к тестированию.

# 4 Тестирование

## Адаптивный дизайн сайта

Для создания адаптивного дизайна использовались технологии, такие как медиа-запросы, гибкие сетки и изменяемые изображения. При этом сайт проектировался таким образом, чтобы его содержимое и функциональность оставались доступными и удобными для пользователя независимо от типа устройства, на котором он отображается. Пример адаптивности главной страницы на устройстве шириной 412px и высотой 915px представлен на рисунке 4.1.

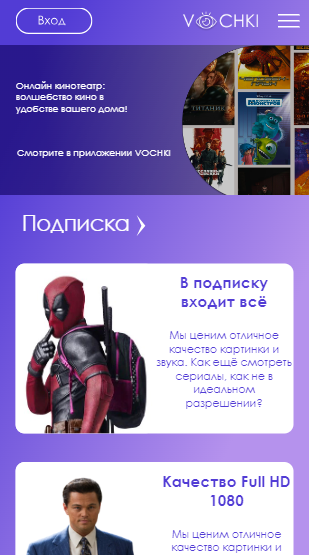


Рисунок 4.1 – Адаптивность главной страницы

Адаптивность веб-страницы «Каталог» на устройстве шириной 412px и высотой 915px представлен на рисунке 4.2.

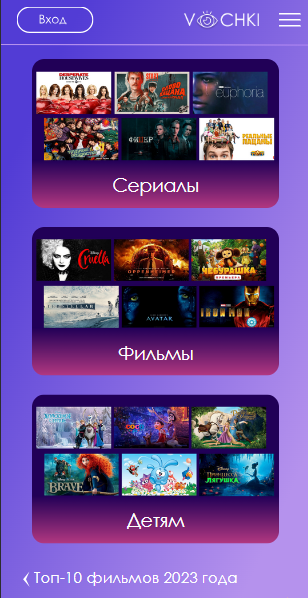


Рисунок 4.2 – Адаптивность страницы «Каталог»

## Кроссбраузерное тестирование

Кроссбраузерность — свойство веб-сайта отображаться и функционировать во всех часто используемых браузерах идентично. Под идентичностью функционирования подразумевается: отсутствие некорректной работы, ошибок в вёрстке и способность отображать материал с одинаковой степенью читабельности. Использование SVG на нашем сайте также помогает обеспечить кроссбраузерность, поскольку формат поддерживается большинством современных браузеров, включая Chrome, Firefox и Edge.

Веб-сайт был открыт при помощи различных браузеров:

* Microsoft Edge (Рисунок 4.3)
* Google Chrome (Рисунок 4.4)
* Mozilla Firefox (Рисунок 4.5)

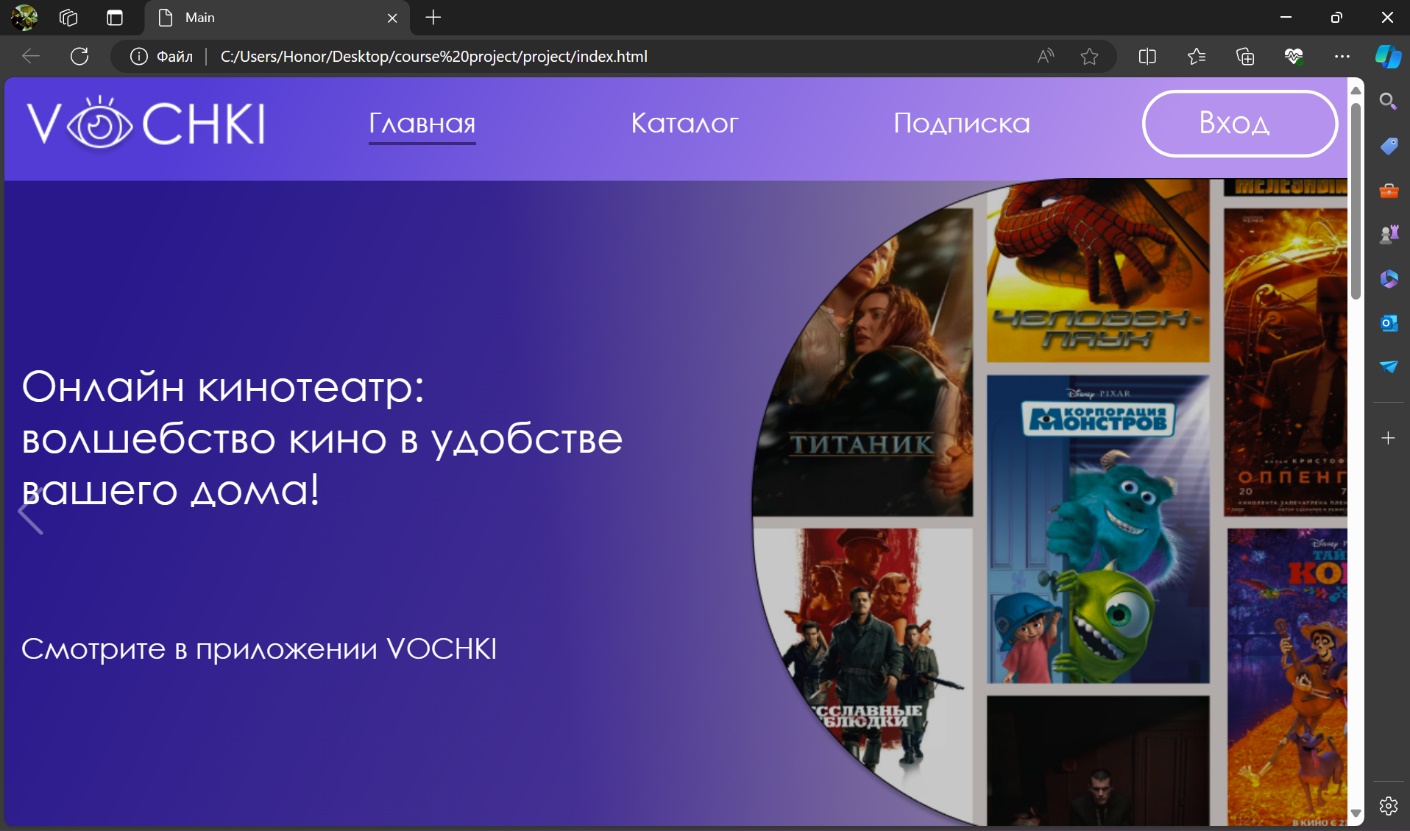


Рисунок 4.3 – Главная страница в Microsoft Edge

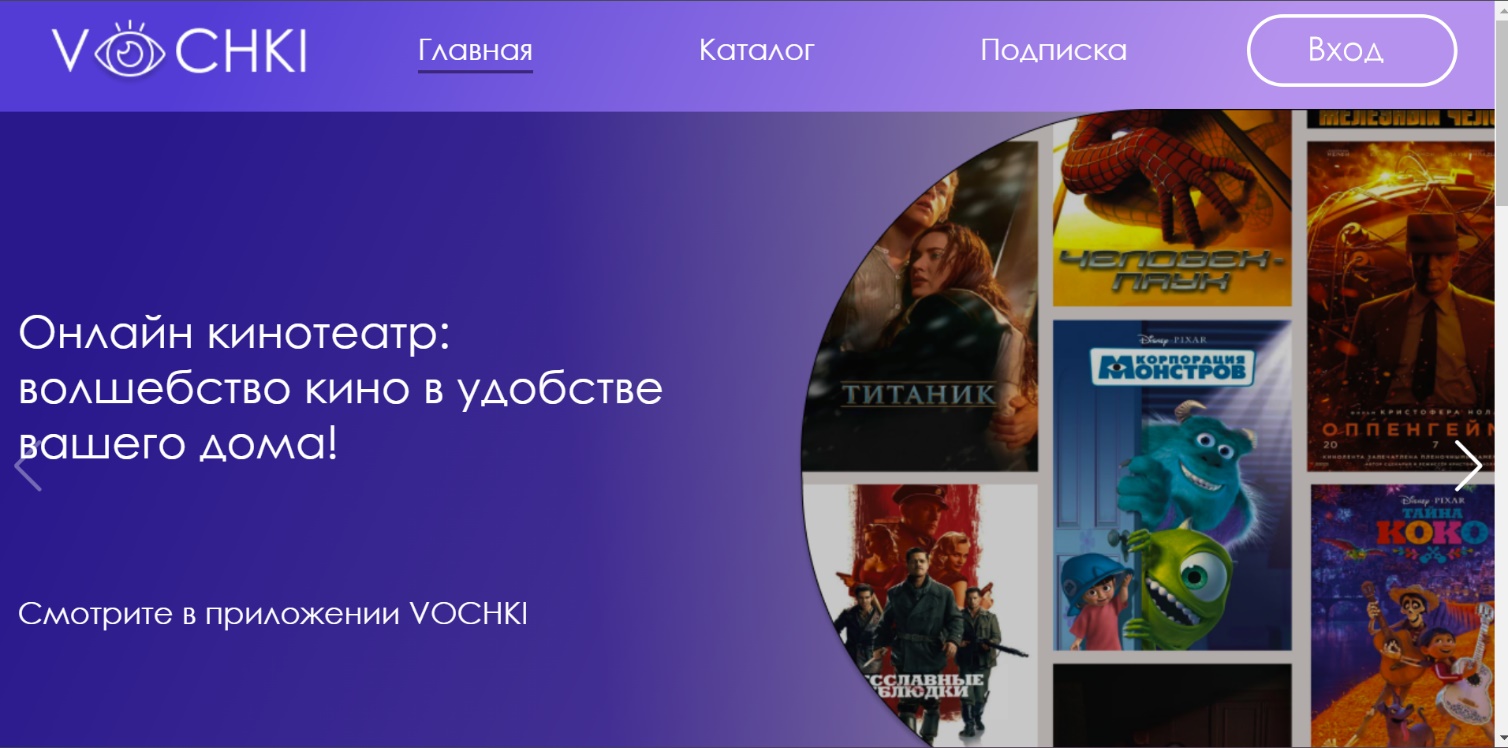


Рисунок 4.3 – Главная страница в Google Chrome

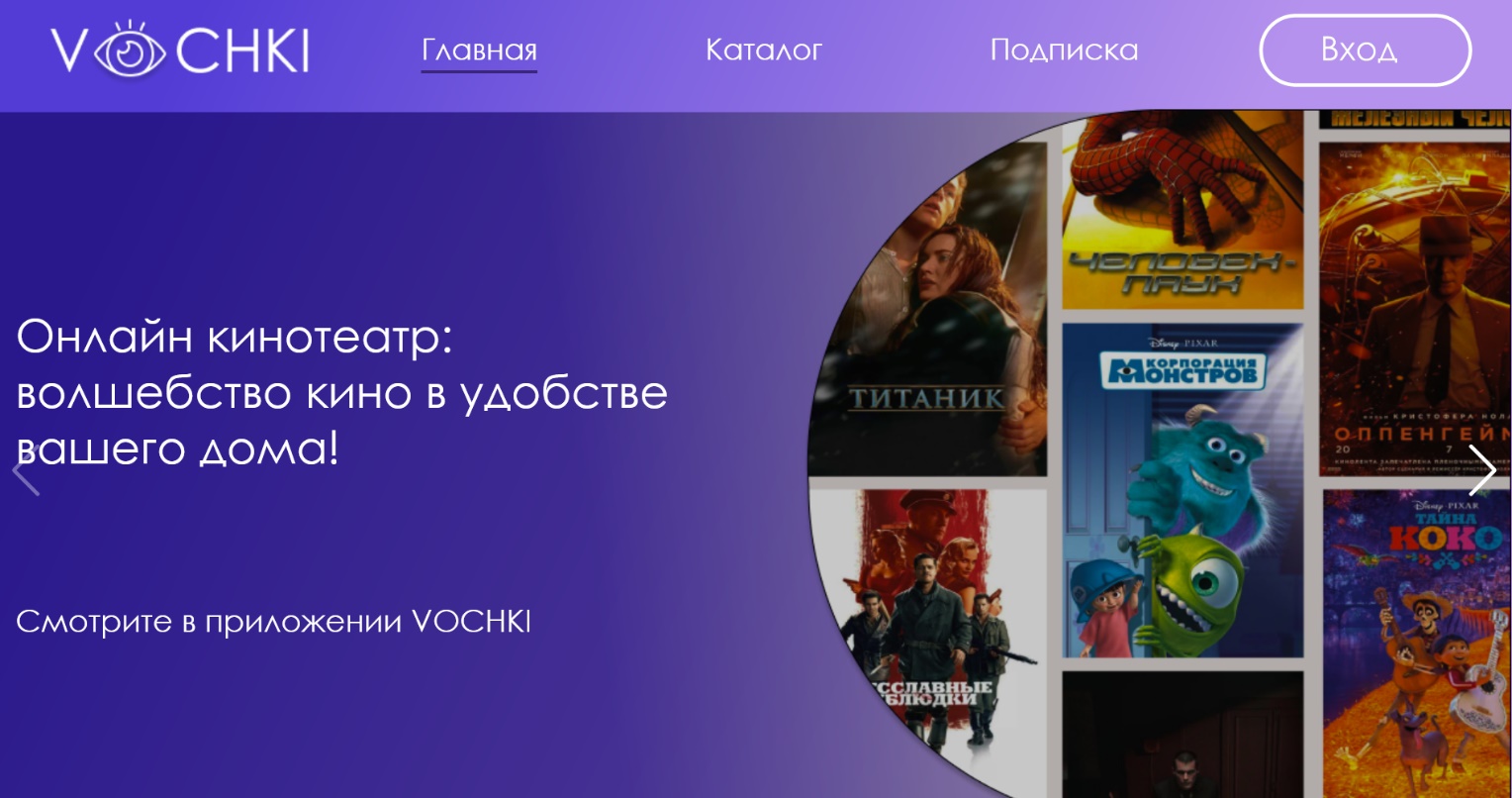


Рисунок 4.3 – Главная страница в Mozilla Firefox

## Валидность

Валидность верстки — это соответствие html-кода сайта утвержденным критериям. Стандарты валидности были разработаны The World Wide Web Consortium (W3C).

Весь HTML, CSS, XML код был протестирован с помощью онлайн-сервиса linter. Результат представлен на рисунке 4.4



Рисунок 4.4 – Проверка веб-страницы на валидность

## Руководство пользователя

Главная страница является точкой входа на веб-сайт. C помощью горизонтального меню в десктопной версии или меню-бургер на мобильной/планшетной версии осуществляется навигация по веб-сайту. При нажатии на элементы меню осуществляется переход к соответствующей веб-странице.

На странице «Каталог» содержится раскладка всех доступных категорий контента и рейтинг самых популярных фильмов 2023 года. При нажатии на них осуществляется переход к соответствующей веб-странице.

На страницах «Фильмы», «Сериалы», «Детям» можно ознакомится с контентом, который предоставляет данный веб-сайт. При наведении на обложку фильма/сериала появляется краткая информация о нем, рейтинг и возрастное ограничение. При нажатии на обложку фильма/сериала осуществляется переход к соответствующей веб-странице, на которой расположена подробная информация о контенте, трейлер, рейтинг и собственно сам фильм, который можно воспроизвести только после входа и оформления подписки.

## Выводы

В данном разделе было рассмотрено, как веб-сайт ведет в себя на разных устройствах и в разных браузерах. Проведена валидация HTML, CSS, XML кода. Так же создано руководство пользователя, где кратко описано, как пользоваться веб-сайтом.

# Заключение

В ходе курсового проекта был разработан многостраничный веб-сайт для онлайн-кинотеатра. В процессе разработки были проанализированы различные веб-сайты с подобной тематикой, выбраны способы верстки и стилевого оформления, а также разработано пользовательское руководство.

Список выполненной работы:

* Анализ аналогичных решений.
* Выбор способа верстки.
* Выбор стилевого оформления.
* Разработка логотипа.
* Разработка пользовательских элементов.
* Разработка спецэффектов.
* Добавление веб-сайту адаптивности.
* Тестирование веб-сайта.
* Создание руководства пользователя.

Результатом данного проекта стал удобный и функциональный веб-сайт, который позволяет пользователям быстро и легко выбрать фильм/сериал, а также получить дополнительную информацию фильмах/сериалах. Дизайн веб-сайта прост и понятен, а добавление адаптивности и кроссбраузерности позволяет пользователям удобно пользоваться веб-сайтом на любом устройстве и в любом браузере.

Разработанное руководство пользователя поможет посетителям быстро ориентироваться на сайте и получить все необходимые сведения. В процессе тестирования веб-сайта были выявлены и исправлены ошибки, что позволило сделать сайт еще более удобным и функциональным для пользователей.

# Список использованных источников

1. Документация по Figma [Электронный ресурс]. – Режим доступа: <https://help.figma.com> – Дата доступа 01.03.2023.
2. HTML5, CSS3 и JavaScript. Исчерпывающее руководство. [Электронный ресурс]. Режим доступа – <https://webbooks.com.ua/?p=1048> – Дата доступа 05.03.2022.
3. Новая большая книга CSS [Печатный ресурс]. Дата доступа 07.03.2023
4. Документация по Sass/SCSS [Электронный ресурс]. – Режим доступа: <https://sass-scss.ru/guide/> – Дата доступа 10.03.2023.
5. Документация по JavaScript/HTML [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org> – Дата доступа 10.03.2023.
6. Введение в кросс-браузерное тестирование [Электронный ресурс]. – Режим доступа: [https://developer.mozilla.org/ru/docs/Learn/Tools\_and\_testing/  
   Cross\_browser\_testing/Introduction](https://developer.mozilla.org/ru/docs/Learn/Tools_and_testing/Cross_browser_testing/Introduction). – Дата обращения: 01.05.2023.

# Приложение А

**Макет структуры веб-сайта**



Рисунок 1 – Главная страница

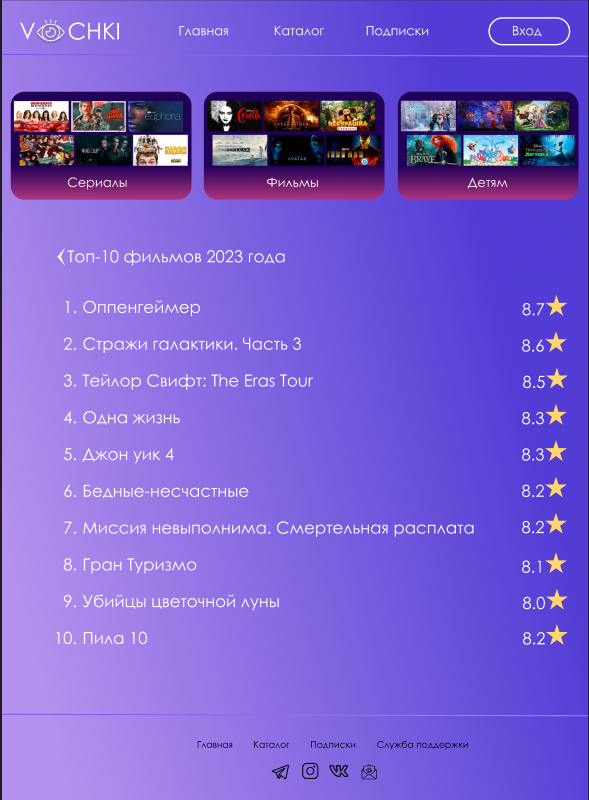


Рисунок 2 – Страница «Каталог»

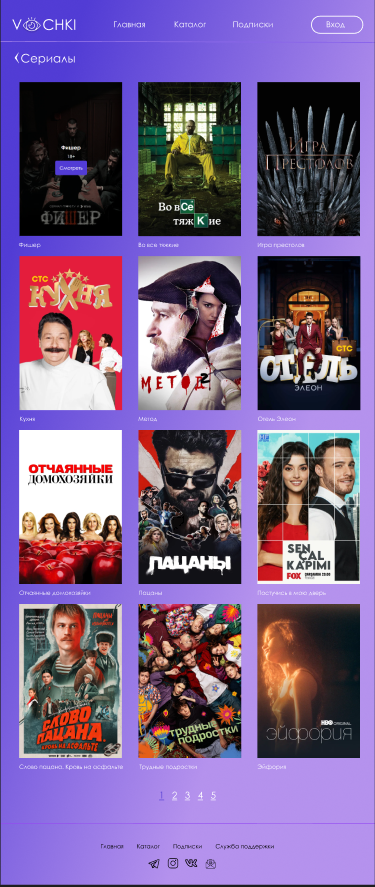


Рисунок 3 – Страница «Сериалы»

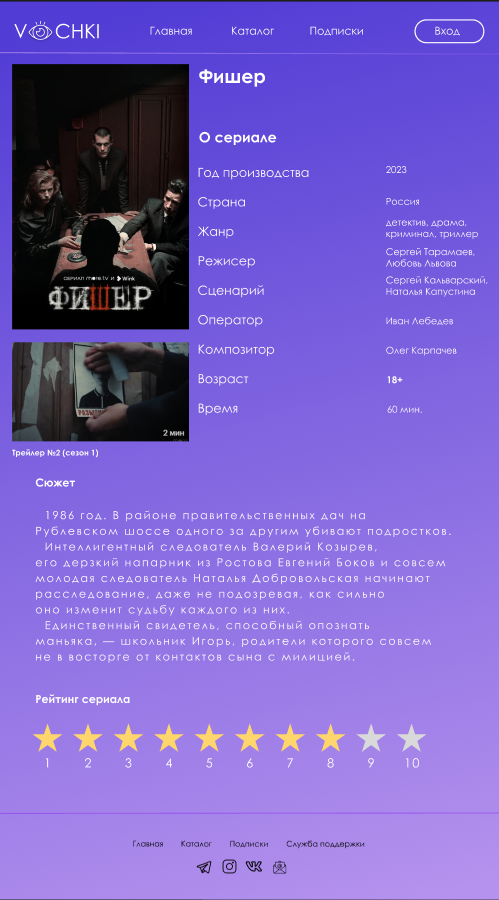


Рисунок 4 – Страница сериала «Фишер»



Рисунок 5 – Страница «Подписка»

# Приложение Б

**Листинг HTML документа**

<header>

<div class="header">

<div class="container">

<div class="menu">

<img src="./icons/logo.png" class="logo" />

<nav class="nav">

<ul>

<li class="nav\_\_item">

<a

style="

display: inline-block;

border-bottom: 3px solid #3d277d;

"

class="nav\_\_link"

href=""

>Главная</a

>

</li>

<li class="nav\_\_item">

<a class="nav\_\_link" href="catalog.html">Каталог</a>

</li>

<li class="nav\_\_item">

<a id="subscribe\_link " class="nav\_\_link" href="subscribe.html">Подписка</a>

</li>

</ul>

</nav>

<div class="button">

<a href="#" id="open\_pop\_up"

><img src="./icons/button.svg" id="" class="button\_enter"

/></a>

</div>

</div>

</div>

<div class="burger">

<span></span>

</div>

</div>

</header>

Листинг 1 – Реализация верхнего колонтитула

<footer>

<div class="container">

<img src="./icons/line.svg" />

<div class="contact">

<nav>

<ul>

<li><a href="index.html">Главная</a></li>

<li><a href="catalog.html">Каталог</a></li>

<li><a href="subscribe.html">Подписка</a></li>

<li><a href="#">Служба поддержки</a></li>

</ul>

</nav>

</div>

<div class="icons">

<nav>

<div class="telegram">

<img src="./icons/telegram.png" alt="" />

</div>

<div class="instagram">

<img src="./icons/instagram.png" alt="" />

</div>

<div class="vk">

<img src="./icons/vk.png" alt="" />

</div>

<div class="email">

<img src="./icons/email.png" alt="" />

</div>

</nav>

</div>

</div>

</footer>

Листинг 2 – Реализация нижнего колонтитула

# Приложение В

**Листинг CSS**

ul {

list-style: none;

display: flex;

justify-content: space-between;

}

.menu {

display: flex;

justify-content: space-between;

align-items: center;

width: 1200px;

}

.menu > img {

padding: 0 0 0 10px;

}

.button\_enter {

padding: 0 10px 0 0;

}

.nav {

width: 600px;

}

.header {

display: flex;

justify-content: center;

}

header {

padding: 10px 0 15px 0;

}

@media (max-width: 1200px) {

a {

font-size: 2.5vw;

}

.menu {

display: flex;

justify-content: space-between;

align-items: center;

width: 1200px;

}

.menu > img {

padding: 0 0 0 6vw;

width: 17vw;

}

.button\_enter {

padding: 0 8vw 0 0;

width: 15vw;

}

.nav {

width: 50vw;

}

.header {

display: flex;

justify-content: center;

}

header {

padding: 10px 0 15px 0;

}

}

@media (max-width: 1100px) {

.menu > img {

padding: 0 0 0 14vw;

}

.button\_enter {

padding: 0 14vw 0 0;

width: 15vw;

}

}

@media (max-width: 950px) {

.menu {

display: flex;

width: 90vw;

justify-content: space-around;

flex-direction: row-reverse;

}

.header {

display: flex;

margin: 0;

align-items: center;

}

.menu > img {

display: flex;

padding-right: 1vw;

}

.menu > img:last-child {

padding: 0 5vw 0 0;

}

}

@media (max-width: 900px) {

.menu > img {

width: 17vw;

}

}

@media (max-width: 730px) {

.menu > img {

width: 22vw;

}

a {

font-size: 3.5vw;

}

}

@media (max-width: 530px) {

a {

font-size: 4vw;

}

}

@media (max-width: 450px) {

.menu > img {

width: 25vw;

}

a {

font-size: 4.5vw;

}

.button\_enter {

width: 24vw;

}

}

.nav\_\_link {

position: relative;

display: inline-block;

font-size: clamp(20px, 3vw, 25px);

line-height: 36px;

font-weight: 500;

color: #fff;

text-decoration: none;

}

.nav\_\_link::before {

content: "";

position: absolute;

bottom: 0;

left: 0;

height: 3px;

width: 100%;

background-color: #3d277d;

opacity: 0;

transition-property: opacity, transform;

transition-timing-function: ease-in-out;

transition-duration: 0.2s;

}

.nav\_\_item:hover .nav\_\_link::before {

transform: translateY(3px);

opacity: 1;

}

Листинг 3 – Реализация стилей верхнего колонтитула

.line {

display: flex;

padding: 14px 0 0 0;

}

.contact {

display: flex;

justify-content: center;

}

.contact > nav {

width: 591px;

padding: 30px 0 0 0;

}

.contact ul:last-of-type {

list-style: none;

display: flex;

justify-content: space-between;

}

.contact a:last-of-type {

text-decoration: none;

color: black;

font-size: 18px;

}

.telegram > img {

width: 37px;

}

.instagram > img {

width: 35px;

}

.vk > img {

width: 41px;

}

.email > img {

width: 37px;

}

li {

display: flex;

justify-content: space-between;

}

.list\_\_img {

padding-right: 5vw;

width: 40px;

height: auto;

}

.icons {

display: flex;

justify-content: center;

padding: 30px 0 15px 0;

}

.icons > nav {

width: 232px;

display: flex;

justify-content: space-between;

}

@media (max-width: 730px) {

.contact ul:last-of-type {

justify-content: space-between;

}

.contact a:last-of-type {

font-size: 3vw;

}

}

@media (max-width: 520px) {

.contact ul:last-of-type {

justify-content: space-between;

}

.contact > nav {

width: 90vw;

}

.contact a:last-of-type {

font-size: 3.5vw;

}

.telegram > img {

width: 7vw;

}

.instagram > img {

width: 7vw;

}

.vk > img {

width: 7vw;

}

.email > img {

width: 7vw;

}

}

@media (max-width: 450px) {

.contact ul:last-of-type {

justify-content: space-between;

}

.contact > nav {

width: 90vw;

}

.contact a:last-of-type {

font-size: 3.5vw;

}

.telegram > img {

width: 7vw;

}

.instagram > img {

width: 7vw;

}

.vk > img {

width: 7vw;

}

.email > img {

width: 7vw;

}

.icons {

display: flex;

justify-content: center;

padding: 4vw 0 4vw 0;

}

.icons > nav {

width: 232px;

display: flex;

justify-content: space-between;

}

}

Листинг 4 – Реализация стилей нижнего колонтитула

# Приложение Г

**Листинг XML-файлов**

<?xml version="1.0" encoding="UTF-8"?>

<gallery>

<image>

<path>sitcoms\fisher.webp</path>

<caption>Фишер</caption>

<title></title>

<age>18+</age>

</image>

<image>

<path>sitcoms\во все тяжкие.webp</path>

<caption>Во все тяжкие</caption>

<age>18+</age>

</image>

<image>

<path>sitcoms\игра престолов.webp</path>

<caption>Игра престолов</caption>

<age>18+</age>

</image>

<image>

<path>sitcoms\кухня.webp</path>

<caption>Кухня</caption>

<age>16+</age>

</image>

<image>

<path>sitcoms\метод.webp</path>

<caption>Метод</caption>

<age>18+</age>

</image>

<image>

<path>sitcoms\отель элеон.webp</path>

<caption>Отель Элеон</caption>

<age>16+</age>

</image>

<image>

<path>sitcoms\отчаянные домохозяйки.webp</path>

<caption>Отчаянные домохозяйки</caption>

<age>16+</age>

</image>

<image>

<path>sitcoms\пацаны.webp</path>

<caption>Пацаны</caption>

<age>18+</age>

</image>

<image>

<path>sitcoms\постучись в мою дверь.webp</path>

<caption>Постучись в мою дверь</caption>

<age>14+</age>

</image>

<image>

<path>sitcoms\слово пацана.webp</path>

<caption>Слово пацана </caption>

<age>16+</age>

</image>

<image>

<path>sitcoms\трудные подростки.webp</path>

<caption>Трудные подростки</caption>

<age>16+</age>

</image>

<image>

<path>sitcoms\эйфория.webp</path>

<caption>Эйфория</caption>

<age>16+</age>

</image>

</gallery>

Листинг 5 – Хранение данных о сериалах

# Приложение Д

**Листинг SVG**

<svg width="1300" height="6" viewBox="0 0 1300 6" fill="none" xmlns="http://www.w3.org/2000/svg">

<path d="M1.59413e-05 1L1300 4.59115" stroke="url(#paint0\_linear\_545\_34)" stroke-width="2"/>

<defs>

<linearGradient id="paint0\_linear\_545\_34" x1="-0.0395196" y1="19" x2="1299.96" y2="22.5911" gradientUnits="userSpaceOnUse">

<stop stop-color="#B592ED"/>

<stop offset="1" stop-color="#7951EC"/>

</linearGradient>

</defs>

</svg>

Листинг 6 – Линия для оформления нижнего и верхнего колонтитулов

# Приложение Е

**Листинг JavaScript**

document.querySelector('.burger').addEventListener('click', function(){

this.classList.toggle('active');

document.querySelector('.nav').classList.toggle('open')

})

Листинг 7 – Бургер-меню

let table = (arr\_tag, arr) => {

let i = 0;

for (const elem\_arr of arr\_tag) {

arr[i++] = elem\_arr.textContent;

}

return arr;

};

let parse = (xmlString) => {

const parser = new DOMParser();

const xmlDoc = parser.parseFromString(xmlString, "text/xml");

const paths = xmlDoc.getElementsByTagName("path");

const captions = xmlDoc.getElementsByTagName("caption");

const ages = xmlDoc.getElementsByTagName("age");

let paths\_arr = [],

captions\_arr = [];

ages\_arr = [];

paths\_arr = table(paths, paths\_arr);

captions\_arr = table(captions, captions\_arr);

ages\_arr = table(ages, ages\_arr);

for (let i = 0; i < paths\_arr.length; i++) {

document.querySelector(".sitcoms").insertAdjacentHTML(

"beforeend",

`

<div class="sitcom">

<div class="hover-effect-cover">

<li>

<img class="cover" src="${paths\_arr[i]}"></img>

<div class="effect-to-top">

<h3><caption >${captions\_arr[i]}</caption></h3>

<p><age>${ages\_arr[i]}</age></p>

<a href="${

captions\_arr[i] === "Фишер" ? "fisher.html" : "#"

}">Смотреть</a>

</li>

</div>

<div class="caption"><caption >${captions\_arr[i]}</caption></div>

</div>

</div>

`

);

}

};

async function parseXML() {

let a = await fetch("xml/sitcoms.xml");

let xmlString = await a.text();

parse(xmlString);

}

parseXML();

Листинг 8 – Вывод сериалов и данных о них

const openPopUp = document.getElementById("open\_pop\_up");

const closePopUp = document.getElementById("pop\_up\_close");

const popUp = document.getElementById("pop\_up");

openPopUp.addEventListener("click", function (e) {

e.preventDefault();

popUp.classList.add("active");

});

closePopUp.addEventListener("click", () => {

popUp.classList.remove("active");

});

Листинг 9 – Открытие/закрытие модального окна