

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

а

Кафедра Информационных систем и технологий

а

Специальность 6-05-0611-01 «Информационные системы и технологии» а

Специализация Информационные системы а

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КУРСОВОГО ПРОЕКТА

по дисциплине «Компьютерные языки разметки» а

Тема Web-сайт «Фишер» а

Исполнитель

студентка 1 курса группы 3

подпись, дата

В.А.Шманай

инициалы и фамилия

Руководитель

ассистент

должность, учен. степень, ученое звание

подпись, дата

А.В.Харланович

инициалы и фамилия

Курсовой проект защищен с оценкой _____

Руководитель _____

подпись

А.В.Харланович

инициалы и фамилия

Минск 2024

Содержание

Введение

1. Постановка задачи

- 1.1 Обзор аналогичных решений
- 1.2 Техническое задание
- 1.3 Выбор средств реализации программного продукта
- 1.4 Вывод

2. Проектирование страниц веб-сайта

- 2.1 Выбор способа верстки
- 2.2 Выбор стилевого оформления
- 2.3 Выбор шрифтового оформления
- 2.4 Разработка логотипа
- 2.5 Разработка пользовательских элементов
- 2.6 Разработка спецэффектов
- 2.7 Выводы

3. Реализация структуры веб-сайта

- 3.1 Структура HTML-документа
- 3.2 Добавление таблиц стилей Sass и CSS
- 3.3 Использование стандартов XML (SVG)
- 3.4 Управление элементами DOM
- 3.5 Выводы

4. Тестирование веб-сайта

- 4.1 Кроссбраузерность веб-сайта
- 4.2 Адаптивный дизайн веб-сайта
- 4.3 Выводы

Заключение

Введение

В современном мире, где медиа оказывает огромное влияние на формирование наших взглядов и ценностей, сериалы становятся неотъемлемой частью культурного пейзажа. Они не только предлагают развлечение, но и являются зеркалом, отражающим важные аспекты современной жизни. В этом плане, сериал "Фишер" выделяется особым значением, поскольку основан на реальных событиях, а его сюжетные линии и персонажи затрагивают широкий спектр актуальных тем, начинающихся от семейных отношений между родителями и детьми до понимания всей серьезности происходящего. Анализ "Фишера" не только позволяет понять глубину и сложность исторических событий, но и помогает осмыслить их влияние на современное общество и наше восприятие социума.

Выбор данной темы для моего курсового проекта обусловлен не только моим личным интересом к сериалу "Фишер", но и его уникальной глубиной и интенсивностью сюжета. Я считаю, что этот сериал имеет потенциал расширить знания и понимание людей о страшных событиях прошлого века, оставаясь важным источником исторической информации. Мой сайт будет предлагать пользователям обширную информацию о исторических фактах, на которых основан сериал, а также проводить анализ ключевых тем и персонажей. Отметив недостаток упорядоченной информации о сериале в сети, я решила создать ресурс, который станет ценным источником для всех, кто хочет глубже погрузиться в его мир.

В процессе работы над проектом были поставлены следующие задачи:

1. Создание кросс-браузерного веб-сайта.
2. Разработка макета веб-сайта.
3. Тестирование веб-сайта.
4. Добавление анимации.

Моя цель - создать привлекательную онлайн-платформу для поклонников сериала "Фишер". Здесь они смогут окунуться в уникальную атмосферу сериала через эксклюзивные материалы, обсудить сюжетные повороты и персонажей, поделиться впечатлениями и идеями с единомышленниками. В

результате создания такого сайта, он станет не только источником развлечения, но и ценным ресурсом для всех, кто увлечен историей и обществом. Курсовой проект направлен на анализ аналогичных сайтов, изучение их преимуществ и недостатков, а также на совершенствование навыков верстки и разработки, чтобы создать максимально привлекательный и функциональный ресурс

1. Постановка задачи

Цель сайта “Фишер” – просмотр и погружение в атмосферу сериала, чтобы каждый любитель страшных историй мог прочувствовать весь тот ужас прошлого века. Сайт создан в развлекательных целях, для любителей историй, основанных на реальных событиях.

1.1. Обзор аналогичных решений

При создании сайта на тему сериала "Фишер" важно использовать различные аналитические решения, которые помогут оптимизировать сайт и улучшить его удобство использования. Эти решения позволят учитывать потребности пользователей и достигать лучших результатов в поисковых системах. Поэтому крайне важно провести анализ сайтов, посвященных аналогичным темам. В качестве аналогов были выбраны сайты, посвященные другим телесериалам, а также просмотрены различные макеты-примеры сайтов кинематографической тематики, такие как веб-сайт с примерами "Ivi.tv" и "Kinopoisk". На основе проведенного анализа будет разработан собственный проект, который учитывает особенности и интересы целевой аудитории сериала "Фишер".

1.1.1 Аналог Ivi.tv

При входе на сайт пользователь сразу же обращает внимание на большое изображение фильма, которое подталкивает его к просмотру. Для этого специально добавлена контрастная красная кнопка "Смотреть Бесплатно", что делает начало просмотра максимально интуитивным и простым. Этот элемент ярко выделяется на странице, обеспечивая превосходный пользовательский опыт и призывая к действию. Это все продемонстрировано на рисунке 1.1.

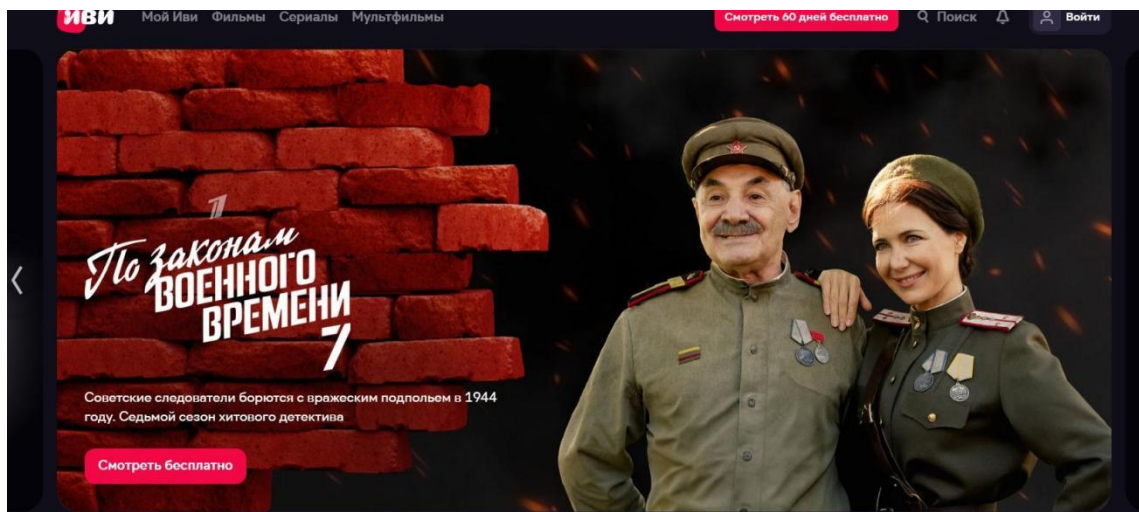


Рис. 1.1 – Главная страница сайта

Редакторы сайта соблюдают единый стиль на всех вкладках страницы, заполняя их максимальным объемом информации, важной для пользователя. Это является одним из преимуществ данного веб-сайта: пользователю предоставляется вся необходимая информация сразу, что делает сайт удобным и легким в использовании.

Если нажать кнопку “Войти”, то попадаем на страницу, изображённую на рисунке 1.2 соответственно.

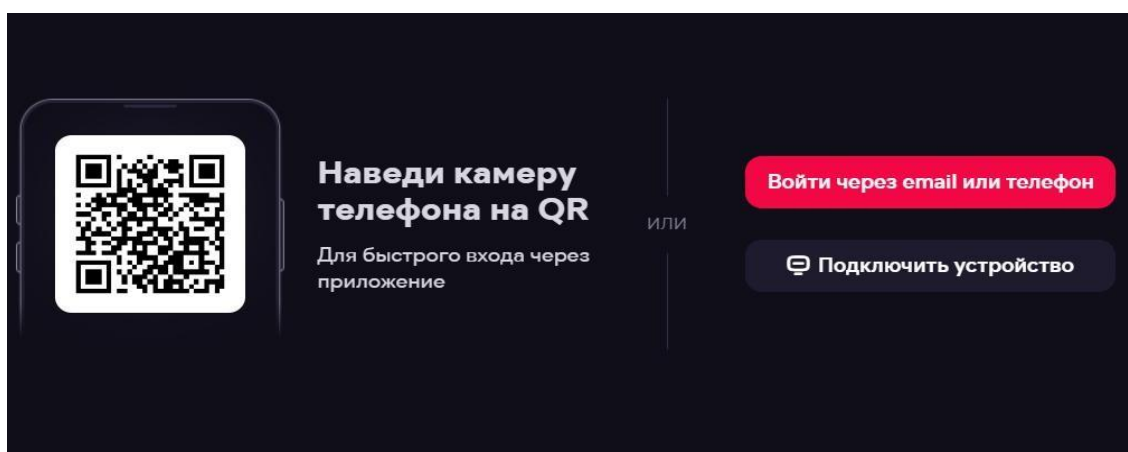


Рис. 1.2 – Вход на свой аккаунт этого сайта через “email” и регистрация через “Подключить устройство”

В нижней части сайта расположены разделы "Контакты", "Соцсети" и "Правообладатели" (рисунок 1.3).

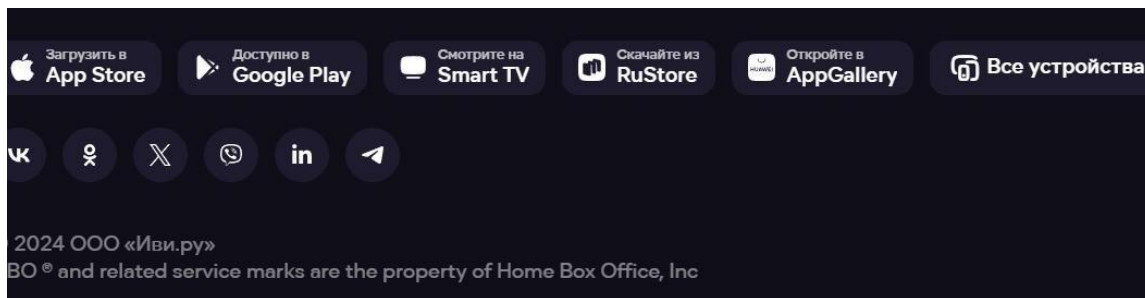


Рис.1.3 – Footer

Основная цветовая палитра у этого сайта нейтральная, представлены ниже, ео также есть и оттенок розового, что позволяет выделить важные элементы на фоне других. (рисунок 1.4).



Рисунок 1.4 – Цветовая палитра

Сайт использует HTML5 и CSS/SCSS для разметки и стилизации, а также JavaScript для создания динамического контента. Ресурс адаптивен и кросс-браузерен благодаря использованию различных стилей в CSS/SCSS и префиксов.

1.2 Техническое задание

Представлена задача создать веб-сайт, посвященный сериалу "Фишер". Этот интерактивный ресурс должен быть наделен актуальной информацией, чтобы

стать источником знаний о сюжете, персонажах и других аспектах этого увлекательного произведения.

Навигационное меню будет располагаться в верхней части каждой страницы сайта. Оно будет содержать ссылки на разделы сайта: "Главная", "Актеры", "Новости", "Смотреть", "Регистрация".



Рис.1.5 – Навигационное меню

Страница "Актеры": На странице "Актеры" будут отображаться фотографии актеров с краткой информацией о их биографии и ролях в сериале. Этот контент будет анимироваться при наведении курсора.

Главная страница: На главной странице сайта будет расположена основная информация о сериале "Фишер". Здесь будут представлены факты о том, что сериал основан на реальных событиях, и фотографии из сериала для привлечения внимания пользователей.

Страница "Новости": На этой странице будут размещены краткие статьи о том, как снимался сериал "Фишер" и интересные факты о нем. Это позволит пользователям узнать больше о создании сериала и его особенностях.

Страница просмотра: На каждой странице будет присутствовать кнопка "Смотреть", которая позволит пользователям перейти к просмотру конкретной серии сериала "Фишер".

Страница "Регистрация": Для того чтобы пользователи могли получить доступ к дополнительным функциям сайта и стать основными пользователями, предусмотрена кнопка "Регистрация", где пользователь сможет выбрать: зарегистрироваться ли или войти. После регистрации или входа пользователь сможет оставлять комментарии, ставить оценки и участвовать в обсуждениях.

В мобильной версии (ширина до 420 px) страница должна адаптироваться под ширину устройства. Также должна быть разработана

версия для планшетного устройства (ширина до 1024px). Начиная с ширины в 1024px должна открываться версия для персонального компьютера.

1.3 Выбор средств реализации программного продукта

Первым этапом является создание макета страниц веб-сайта. Для этого было использовано приложение для разработки интерфейсов и прототипирования - Figma. Это онлайн-сервис, предназначенный для дизайнеров, веб-разработчиков и маркетологов. С его помощью можно создавать прототипы сайтов или приложений, а также иллюстрации и векторную графику. В редакторе Figma можно настраивать совместную работу, вносить и обсуждать правки, как в браузере, так и через приложение на компьютере. Структура сайта создается с помощью языка разметки HTML, а его дизайн оформляется с использованием CSS. Для добавления динамичности сайту используется JavaScript.

Для разработки исходного кода был выбран бесплатный редактор Visual Studio Code. Это удобное программное обеспечение, предназначенное для написания кода, включая вёрстку сайтов. Он обладает множеством встроенных функций, таких как всплывающие подсказки, навигация по коду, автоматическое форматирование и автодополнение, что значительно ускоряет процесс разработки. Visual Studio Code также позволяет подключать различные расширения (плагины), расширяя его функциональность. Учитывая требования к проекту, в том числе адаптивность и кросс-браузерность сайта, было принято решение использовать CSS-препроцессор.

1.4. Вывод

В данном разделе были изучены аналогичные решения, проведен анализ их преимуществ и недостатков с целью разработки высококачественного веб-сайта. Были определены конкретные задачи для создания интерактивного информационного ресурса. Также были изучены инструменты, необходимые для разработки программного продукта, включая языки разметки и редактор кода Visual Studio Code. Проведен анализ особенностей языков и дополнительных средств редактора. Все вышеперечисленные шаги сыграют ключевую роль в достижении поставленной задачи и создании высококачественного продукта в будущем.

2. Проектирование страниц веб-сайта

2.1 Выбор способа верстки

Для обеспечения удобства использования сайта на различных устройствах необходимо использовать адаптивную вёрстку. Для этой цели мы планируем использовать технологии flexbox и grid, которые идеально подходят для создания различных компонентов сайта, таких как меню и футер. Эти методы позволяют легко и гибко организовать расположение элементов на странице, равномерно распределить доступное пространство и выровнять элементы, даже если они имеют разные размеры.

2.2 Выбор стилового оформления

Подбор цветовой схемы играет важную роль в разработке проекта. Цвета должны гармонизировать между собой и соответствовать тематике проекта. Для данного веб-сайта была выбрана тёмная цветовая палитра в основе (рисунок 2.1), дополненная другими цветами для выделения основного контента (рисунок 2.2).

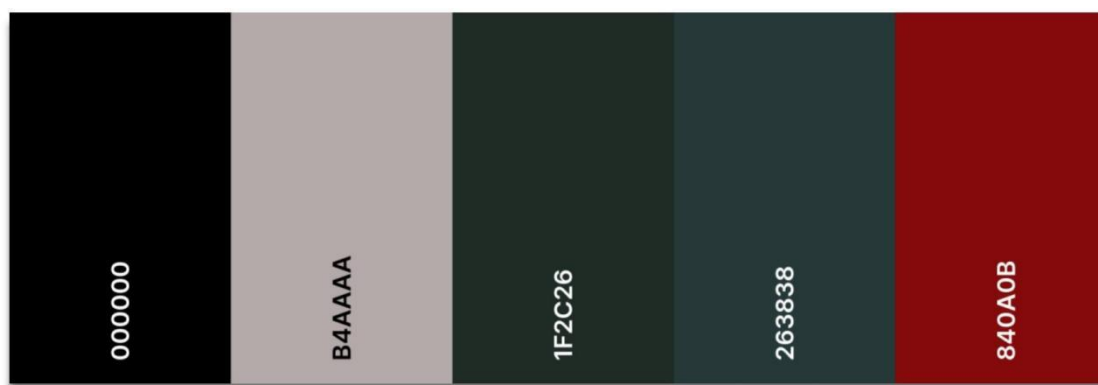


Рисунок 2.1 – Цветовая палитра

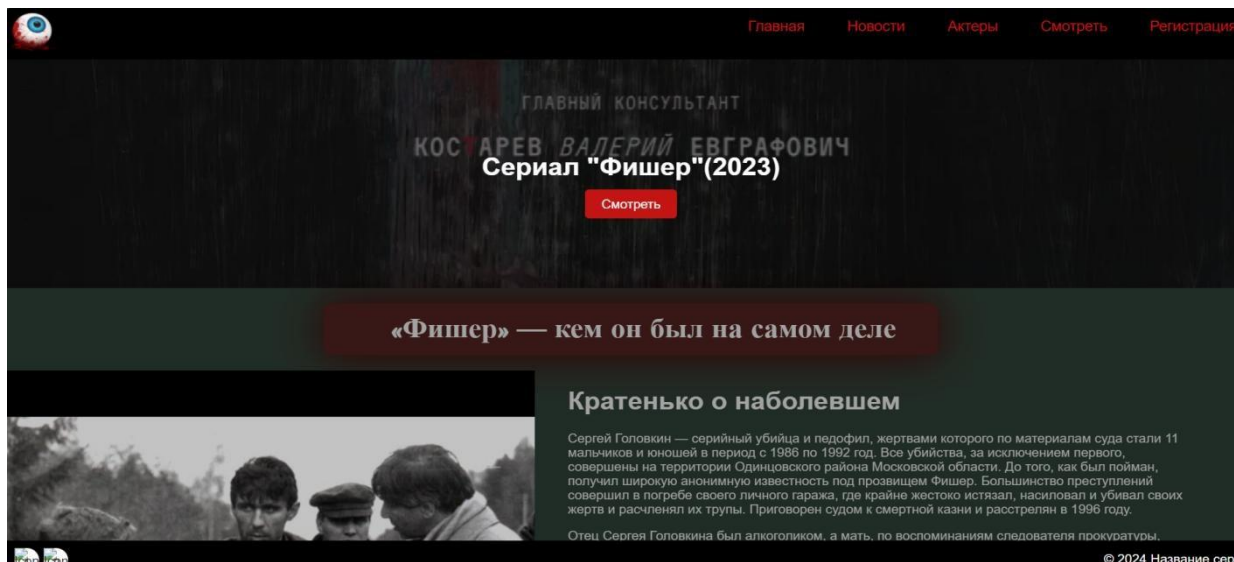


Рисунок 2.2 – Титульная страница

2.3 Выбор шрифтового оформления

В процессе создания данного макета было выбрано несколько основных шрифтов: “Brush Script MT” и “Arial или любой без засечек (sans-serif)”. Все они между собой хорошо сочетаются, а также отлично подходят под выбранный стиль сайта.

2.4 Разработка логотипа

Логотип — это символ, который выступает в качестве уникального идентификатора, помогая пользователю быстро идентифицировать его. Идеальный логотип должен быть легко узнаваемым, читаемым в любом масштабе, простым в своей концепции, запоминающимся и соответствующим стилю проекта. Для нашего веб-проекта был разработан логотип, который воплощает его суть и визуально отражает тематику и стиль.



Рисунок 2.3 – Логотип сайта

Логотип был разработан в графическом редакторе Krita с использованием цветовой палитры, соответствующей общей стилистике и тематике сайта. При выборе цветовой гаммы учитывалась не только видимость логотипа, но и его способность гармонировать с дизайном веб-страницы, чтобы не отвлекать внимание пользователей от основного контента.

2.5 Разработка пользовательских элементов

Эффективная навигация – это ключевой аспект удобства использования сайта. Она обеспечивает быстрый доступ пользователя к различным страницам и разделам сайта. Навигационное меню, представленное на изображении 2.4, способствует удобству перемещения пользователя как между страницами, так и внутри страницы.



Рисунок 2.4 – Навигационное меню

2.6 Разработка спецэффектов

На веб-сайте реализованы интересные динамические эффекты и анимации, которые придают сайту привлекательность. Например, когда пользователь наводит курсор мыши на изображения разделов сайта, они меняют свой цвет и форму, добавляя более интересный вид.(рис. 2.5)

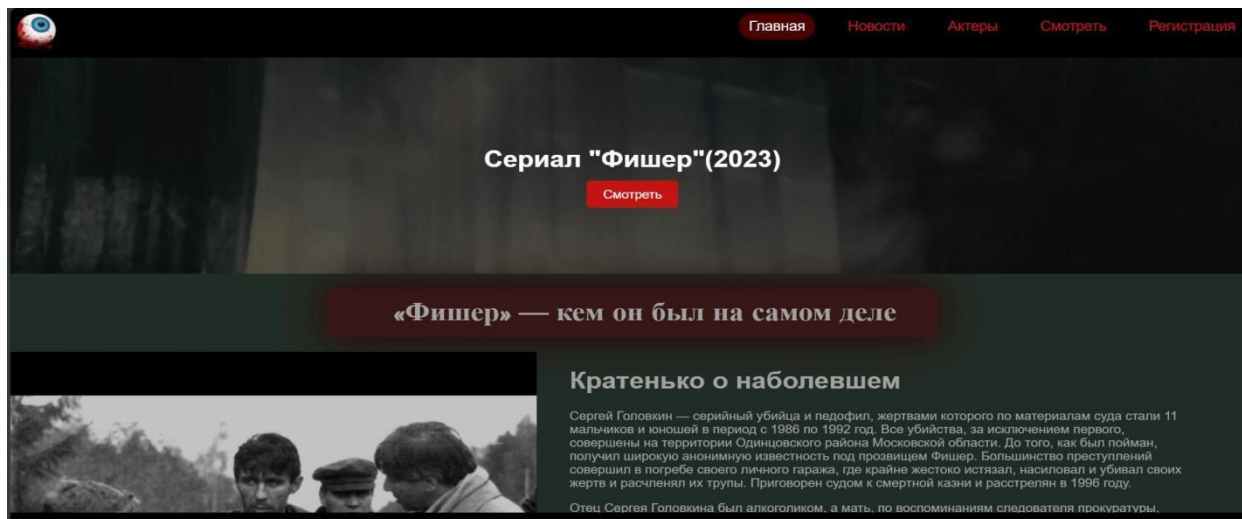


Рисунок 2.8 – Главная страница

2.7. Выводы

В этой главе была разработана структура сайта и создан макет в Figma с переходами между страницами. После анализа аналогов, разработки логотипа, пользовательских элементов и спецэффектов, а также подбора цветовой палитры и шрифтов, мы создали макет сайта. Учитывая цели проекта и необходимость создания адаптивного сайта, мы выбрали метод верстки.

3. Реализация структуры веб-сайта

3.1 Структура HTML-документа

Структура HTML-документа определяет базовый «скелет» для будущего веб-сайта. Код выполнен по стандарту HTML5.

HTML-документ состоит из двух частей: **заголовок** (head) и **тело** (body), расположенных в следующем порядке:

```
<HTML>
  <HEAD>
    ...
  </HEAD>

  <BODY> .
    ..
  </BODY>
</HTML>
```

В начале документа в теге <head>...</head> прописана служебная информация, название веб-сайта. Тег <head> предназначен для хранения других элементов, цель которых — помочь браузеру в работе с данными. Также внутри контейнера <head> находятся метатеги, которые используются для хранения информации, предназначенной для браузеров и поисковых систем. Например, механизмы поисковых систем обращаются к метатегам для получения описания сайта, ключевых слов и других данных.

Содержимое тега <head> не отображается напрямую на веб-странице, за исключением тега <title> устанавливающего заголовок окна веб-страницы.

Далее расположен тег <body>...</body>, внутри которого размещаются все теги, в которых расположен сайт.

Структура тела документа состоит из семантических тегов header, section и footer. Листинг представлен в приложении 3. Эти теги предназначены для того, чтобы компьютерные программы понимали, какой тип информации заложен в данных тегах.

В теге header содержится логотип сайта, навигационное меню с рядом главных функций.

```
<header>
  <div class="logo">
    <img src='eye256x256_114396.png' alt="Логотип"></img>
  </div>
```

```
<nav>
  <a href="index.html">Главная</a>
  <a href="news.html">Новости</a>
  <a href="ivan.html">Актеры</a>
  <a href="watch.html">Смотреть</a>
  <a href="form.html">Регистрация</a>
</nav>
</header>
```

В коде находятся несколько ссылок на html, каждый из которых представляет собой какую-то основную часть сайта.

3.2 Добавление таблиц стилей Sass и CSS

Добавление таблиц стилей CSS позволяет упростить создание стилистического оформления страницы.

Для разных компонентов веб-сайта подбирался свой набор стилей:

```
body
{ font-family: Arial, sans-serif;
background-color: #021008e0;
margin: 0;
padding: 0; }

header {
background-color: #000000;
color: #ce2e2e;
padding: 0px;
display: flex;
justify-content: space-between;
align-items: center;
}

nav {
background-color: #000000;
text-align: center;
padding: 15px 0;
position: absolute;
top: 0;
right: 0;
border-radius: 9px; }
```

```
nav a, .dropbtn
{ color: #c81414;
text-decoration: none;
margin: 0 10px;
padding: 8px 12px;
border: 2px solid #000000;
border-radius: 20px;
background-color: #000000;
transition: background-color 0.3s, color 0.3s;
font-size: 18px; /* добавляем размер шрифта */
}

nav a:hover, .dropdown:hover .dropbtn
{ background-color: #4d0000;
color: #fff; }

.main-title
{ text-align: center;
color: #a4a4a4;
font-weight: bold;
font-family: 'Brush Script MT';
font-size: 35px;
margin-top: 20px;
margin-bottom: 0;
background-color: rgba(61, 19, 19, 0.8); /* Прозрачный цвет фона */
border: 2px solid rgba(61, 19, 19, 0.351); /* Цвет рамки */
border-radius: 10px;
box-shadow: 0 0 55px 5px #300505;
text-indent: 30px;
padding: 10px 20px 10px 20px;
margin: 20px auto;
width: 700px; }

.dropdown
{ position: relative;
display: inline-block;
}

.dropdown-content
{ display: none;
position: absolute;
background-color: #000000;
color: #fff;
```



```
min-width: 160px;
box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
z-index: 1;
border-radius: 9px; }
```

```
    .dropdown-content a
{ color: #fff;
padding: 12px 16px;
text-decoration: none;
display: block;
transition: background-color 0.3s, color 0.3s; }
```

```
    .dropdown-content a:hover
{ background-color: #660000;
}
```

```
    .dropdown:hover .dropdown-content
{ display: block;
}
```

```
    section
{ padding: 5px;
}
```

```
    footer {
background-color: #000000;
color: #fff;
text-align: left;
padding: 10px;
position: fixed;
bottom: 0;
width: 100%;
display: flex;
align-items:
    center; }
```

```
    .social-icons
{ margin-right: auto;
}
```

```
    .social-icons a
{ margin-right: 1px;
display: inline-block;
```

```
width: 30px;
height: 30px;
background-color: #fff;
text-align: center;
line-height: 3px;
border-radius: 60%;
color: #333;
}
```

Это основной header и footer

```
.quarter-screen-video
{ position: relative;
width: 100%;
height: 40vh;
overflow: hidden; }
.quarter-screen-video video
{ position: absolute;
top: 0;
left: 0;
width: 100%;
height: 100%;
object-fit: cover;
z-index: -1;
} .overlay
{
position: absolute;
top: 0;
left: 0; width:
100%; height:
40vh;
background: rgba(0, 0, 0, 0.6);
color: #fff;
display: flex;
justify-content: center;
align-items: center;
} .content
{
text-align: center;
```

```
}  
/* Стили для кнопки  
*/ .watch-button {  
background-color: #c51414;  
color: #fff;  
padding: 10px 20px;  
text-decoration: none;  
border-radius: 5px;  
transition: background-color 0.3s; }  
  
.watch-button:hover  
{ background-color:  
#4d0000; }
```

Это стили для видео-заголовка и навигационных кнопок

```
/* Стили для отдельных новостей  
*/ .news-item {  
margin-bottom: 15px; /* Отступ между новостями */ }  
  
.container1  
{ display:  
flex;  
align-items: flex-start;  
margin-top: 20px;  
}  
  
.image-container1  
{ position: relative;  
margin-right: 20px; }  
  
.image-container1 img  
{ max-width: 100%;  
cursor: pointer;  
transition: transform 0.3s ease;  
float: left;  
margin-right: 20px; /* Добавьте отступ справа, чтобы текст не прижимался к  
изображению */
```

```
}

.image-container1 img:hover
{ transform: scale(1.1);
}

.text-container1
{ color:#a4a4a4;
max-width: 50%; }
.container2
    { display:
        flex;
align-items: flex-start;
margin-top: 10px;
flex-direction: row-reverse;}

.image-container2
{ position: relative;
margin-right: 20px; }

.image-container2 img
{ max-width: 100%;
cursor: pointer;
transition: transform 0.3s ease;
margin-right: 15px; /* Добавьте отступ справа, чтобы текст не прижимался к
изображению */
}

.image-container2 img:hover
{ transform: scale(1.1);
}

.text-container2
{ color: #a4a4a4;
max-width: 50%;
float: left;
} .container3
{
```

```
display: flex;
align-items: flex-start;
margin-top: 20px;
}

.image-container3
{ position: relative;
margin-right: 20px; }

.image-container3 img
{ max-width: 100%;
cursor: pointer;
transition: transform 0.3s ease;
float: left;
margin-right: 20px; /* Добавьте отступ справа, чтобы текст не прижимался к
изображению */
}

.image-container3 img:hover
{ transform: scale(1.1);
}

.text-container3
{ color: #a4a4a4;
max-width: 50%; }
    .container4
    { display:
        flex;
align-items: flex-start;
margin-top: 20px;
flex-direction: row-reverse;
clear: both;
}

.image-container4
{ position: relative;
margin-right: 20px; }
```

```
.image-container4 img
{ max-width: 100%;
cursor: pointer;
transition: transform 0.3s ease;
float: left;
margin-right: 20px; /* Добавьте отступ справа, чтобы текст не прижимался к
изображению */
}

.image-container4 img:hover
{ transform: scale(1.1);
}

.text-container4
{ color: #a4a4a4;
max-width: 50%; }

/* Анимация плавного появления модального окна
*/ .modal {
display: none;
position: fixed;
z-index: 9999;
left: 0;
top: 0;
width: 100%;
height: 100%;
overflow: auto;
background-color: rgba(0, 0, 0, 0.9);
opacity: 0; /* Начальное значение непрозрачности */
transition: opacity 0.3s; /* Анимация изменения непрозрачности */ }

/* Стили для видимости модального окна
*/ .modal.visible {
opacity: 1; /* Конечное значение непрозрачности */ }
```

```
.modal-content
{ margin: 20px
auto; display:
block; max-width:
80%; max-height:
80%; }
```

```
.modal-content img
{ width: 100%;
height: auto; }
```

```
.close {
position: absolute;
top: 15px;
right: 35px; color:
#f1f1f1; font-size:
40px; font-weight:
bold; transition:
0.3s;
}
```

```
.close:hover, .
close:focus
{ color: #bbb;
text-decoration: none;
cursor: pointer;
}
```

Это для информации на главной странице

```
/*для
карусельки*/ .carou
sel {
position: relative;
width: 100%;
max-width: 800px;
margin: 0 auto;
overflow: hidden; }
```

```
.carousel img
{ width: 100%;
display: block; }

.carousel .prev, .carousel .next
{ position: absolute;
top: 50%;
transform: translateY(-50%);
background-color: rgba(255, 255, 255, 0.5);
padding: 10px;
cursor: pointer;
transition: background-color 0.3s; }

.carousel .prev:hover, .carousel .next:hover
{ background-color: rgba(255, 255, 255,
0.8); }

.carousel .prev
{ left: 10px;
}

.carousel .next
{ right: 10px;
}

/* Модальное окно для фото на полный экран
*/ .modal {
display: none;
position: fixed;
z-index: 9999;
left: 0;
top: 0;
width: 100%;
height: 100%;
overflow: auto;
background-color: rgba(0, 0, 0,
0.9); }
```



```
.modal-content
{ display: block;
margin: auto;
max-width: 90%;
max-height:
90%; }

.close {
position: absolute;
top: 15px;
right: 35px;
color: #fff;
font-size: 40px;
font-weight: bold;
cursor: pointer;
}
/* Для размера 412x915 */
@media (max-width: 412px)
{ nav {
padding: 5px 0; }
nav a, .dropbtn
{ margin: 0 5px;
padding: 5px 8px;
font-size: 14px;
}
.main-title
{ font-size:
24px;
padding: 10px 15px;
margin: 10px auto;
width: 90%;
}
.dropdown-content
{ min-width: 120px;
}
.social-icons a
```

```
width: 25px;
height: 25px;
line-height:
2px; }
}

/* Для размера 712x1138 */
@media (max-width: 712px)
{ nav {
padding: 10px 0; }
nav a, .dropbtn
{ margin: 0 8px;
padding: 8px 10px;
font-size: 16px;
}
.main-title
{ font-size:
28px;
padding: 15px 20px;
margin: 15px auto;
width: 90%;
}
.dropdown-content
{ min-width: 140px;
}
.social-icons a
{ width: 28px;
height: 28px;
line-height:
3px; }
}

/* Для размера 375x667 */
@media (max-width: 375px)
{ nav {
padding: 8px 0; }
nav a, .dropbtn
{ margin: 0 7px;
```

```
padding: 7px 9px;
font-size: 15px;
}
.main-title
{ font-size:
26px;
padding: 12px 18px;
margin: 12px auto;
width: 90%;
}
.dropdown-content
{ min-width: 130px;
}
.social-icons a
{ width: 26px;
height: 26px;
line-height:
2px; }
}

/* Для размера 360x740 */
@media (max-width: 360px)
{ nav {
padding: 8px 0; }
nav a, .dropbtn
{ margin: 0 6px;
padding: 6px 8px;
font-size: 14px;
}
.main-title
{ font-size:
24px;
padding: 10px 15px;
margin: 10px auto;
width: 90%;
}
.dropdown-content
{ min-width: 120px;
}
.social-icons a {
```

```
width: 24px;
height: 24px;
line-height:
2px; }
}
```

Это для пролистывания фото(подобие галереи) и @media

3.4 Управление элементами DOM

JavaScript — это язык программирования, который используют разработчики для создания интерактивных веб-страниц. Функции JavaScript могут улучшить удобство взаимодействия пользователя с веб-сайтом: от обновления ленты новостей в социальных сетях и до отображения анимации и интерактивных карт.

Может подключаться как внутри самого HTML-документа, так и отдельным файлом.

В данном проекте JavaScript был использован для регистрации, открытия и закрытия фотографии на главной странице, для просмотра видео на полный экран.

```
// JavaScript код для открытия и закрытия модального окна
function openModal()
{ document.getElementById('myModal').style.display =
"block"; setTimeout(function() {
document.getElementById('myModal').classList.add('visible'); // Добавляем класс
visible для анимации
}, 100); // Ждем некоторое время перед добавлением класса для анимации }

function closeModal()
{ document.getElementById('myModal').classList.remove('visible'); //
Удаляем класс visible для анимации
setTimeout(function()
{ document.getElementById('myModal').style.display =
"none";
}, 300); // Ждем завершения анимации перед скрыванием модального окна }
```

Это для главной страницы

```
// Функция для предварительного просмотра выбранного изображения
function previewImage(event) {
var input = event.target;
var reader = new FileReader();
reader.onload = function() {
var preview = document.querySelector('.profile-preview');
preview.style.backgroundImage = 'url(' + reader.result + ')'; }
reader.readAsDataURL(input.files[0]); }

function redirectToHomepage() {
window.location.href = "index.html"; // Замените "index.html" на путь к вашей
главной странице
}

function toggleForm(formType) {
const registerTab = document.getElementById("register-tab");
const loginTab = document.getElementById("login-tab");
const registerForm = document.getElementById("register-form");
const loginForm = document.getElementById("login-form");

if (formType === "register")
{ registerTab.classList.add("active");
loginTab.classList.remove("active");
registerForm.style.display = "block";
loginForm.style.display = "none";
} else
{ registerTab.classList.remove("active")
; loginTab.classList.add("active");
registerForm.style.display = "none";
loginForm.style.display = "block";
} }

```

Это для регистрации

```
// Открыть модальное окно с видео
function showVideo(videoSrc) {

```

```
var modal = document.getElementById('videoModal');
var videoPlayer = document.getElementById('videoPlayer');
videoPlayer.src = videoSrc;
modal.style.display = 'block';
document.body.classList.add('blur-background'); // Добавляем эффект размытия
заднему фону
}

// Закрыть модальное окно при нажатии на крестик или на свободное от
видео поле
// Функция для закрытия модального окна
function closeModal() {
var modal = document.getElementById('videoModal');
modal.style.display = 'none';
var videoPlayer = document.getElementById('videoPlayer');
videoPlayer.pause(); // Пауза видео при закрытии модального окна }
```

Это для просмотра

3.5. Выводы

На текущем этапе была разработана HTML-структура, оформление с использованием таблиц стилей Sass/CSS и добавлены скрипты на языке программирования JS. Определены стили веб-сайта, созданы анимации, и он готов к следующему этапу - тестированию.

4. Тестирование веб-сайта

4.1 Кроссбраузерность веб-сайта

Кроссбраузерное тестирование включает в себя проверку общего вида проекта в различных браузерах, чтобы убедиться, что шрифты отображаются правильно, анимации работают без сбоев и т. д.

Для проверки кроссбраузерной совместимости веб-сайта были использованы современные браузеры, такие как Google Chrome, Opera, Firefox и Edge. Примеры корректной работы веб-сайта в этих браузерах представлены на рисунках ниже:

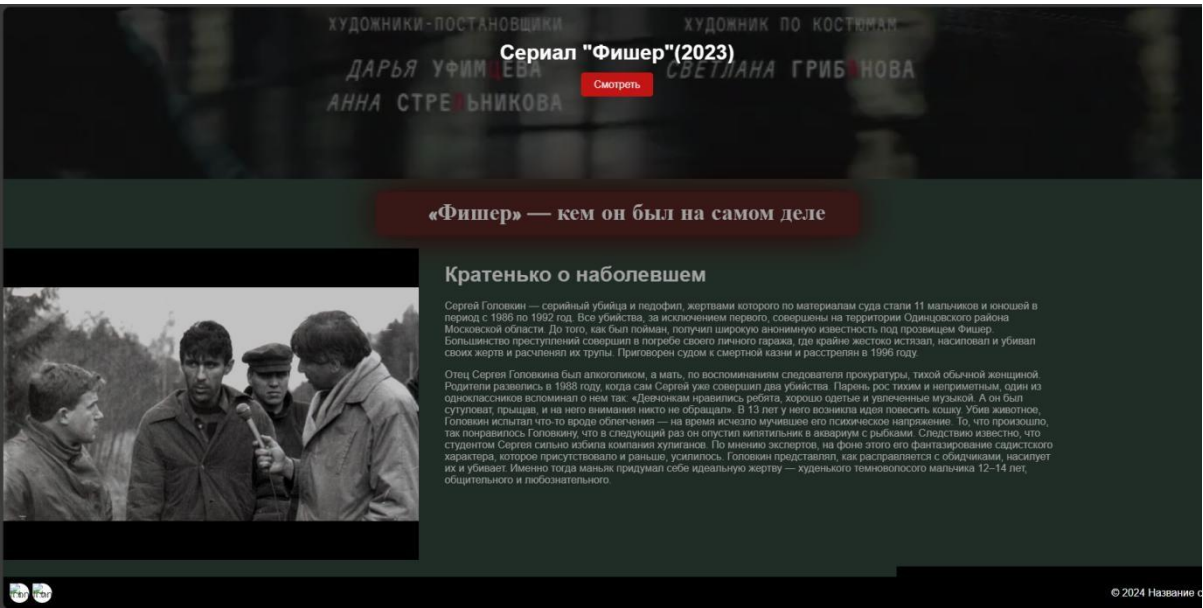


Рисунок 4.1 - Edge главная страница

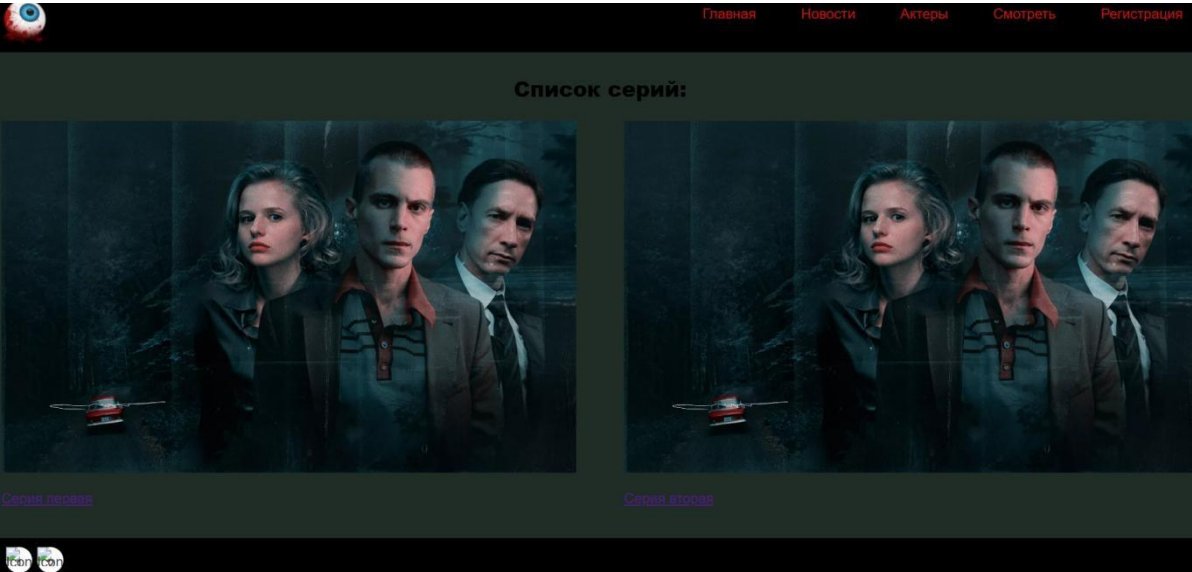


Рисунок 4.2 - Edge страница просмотра

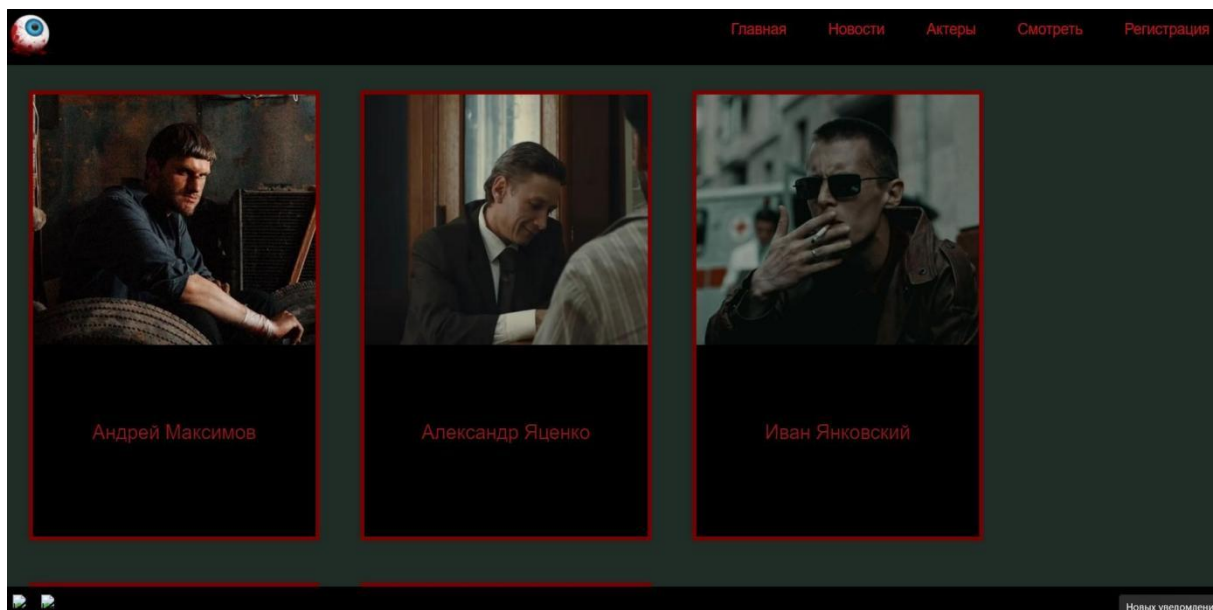


Рисунок 4.3 - Edge страница информации об Актерах

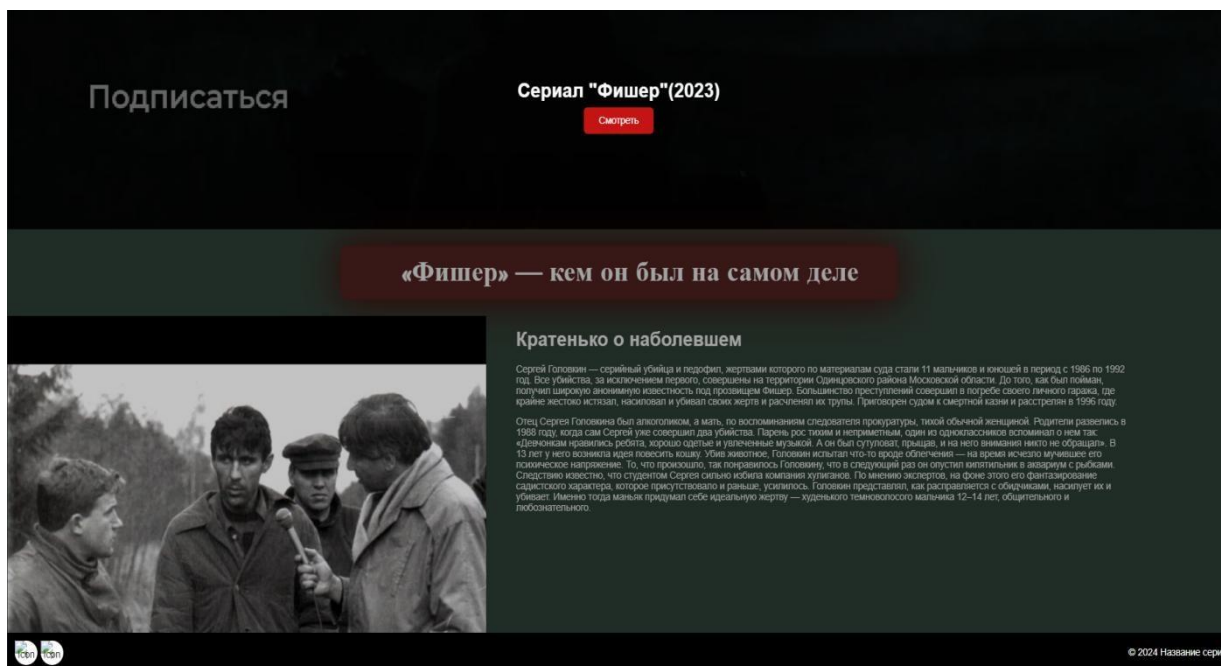


Рисунок 4.4 - Google Chrome главная страница

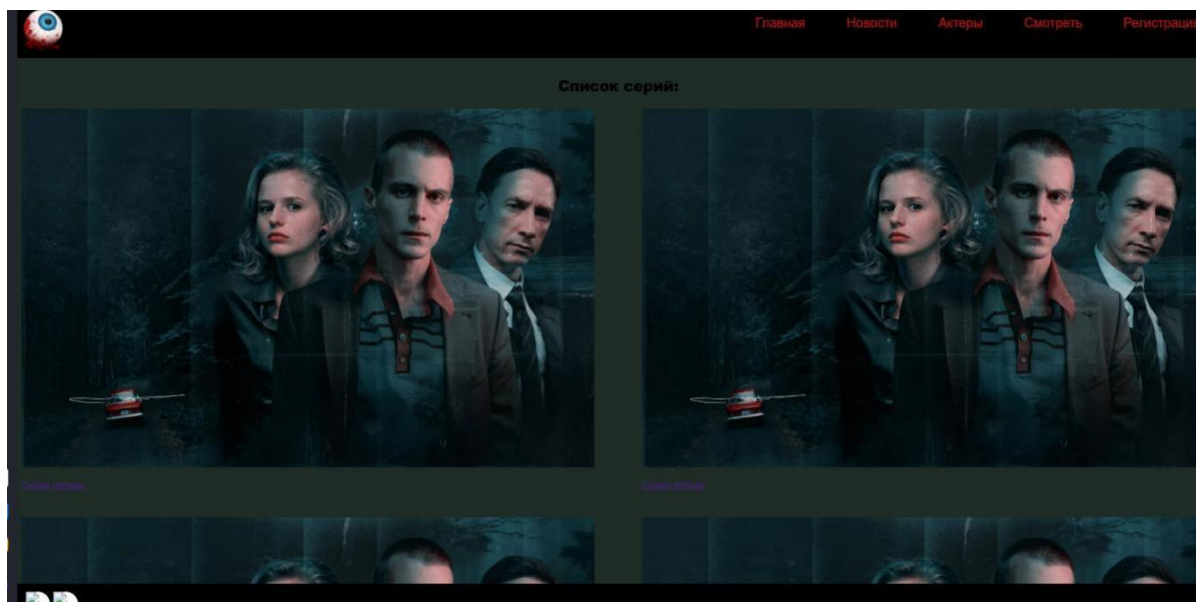


Рисунок 4.5- Google Chrome страница просмотра

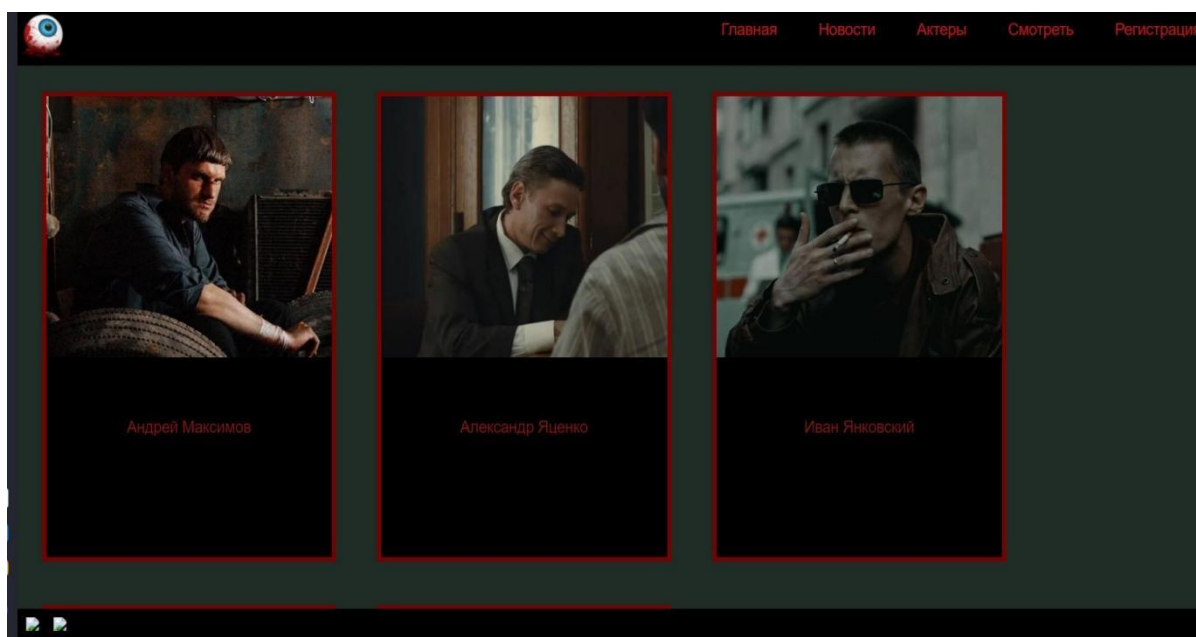


Рисунок 4.6 - Google Chrome страница информации об Актерах

4.2 Адаптивный дизайн веб-сайта

Адаптивная вёрстка сайта — это способ обеспечить оптимальное отображение интерфейсов на устройствах с разным разрешением экранов.

Адаптивный дизайн предусматривает, что пользователь будет взаимодействовать с кнопками, текстом, изображениями через разные устройства. Думает об удобстве пользователя в этом случае уже дизайнер.

Для адаптива данного проекта были выбраны медиа-запросы. Адаптив был сделан для множества устройств, но мы рассмотрим (414px – 896px).



4.4 Вывод

При проведении тестирования были обнаружены различные ошибки, которые были выявлены и в последующем исправлены. Кроме того, в разделе представлено руководство пользователя, которое содержит краткое описание основной функциональности веб-сайта, чтобы пользователи могли быстро и легко разобраться в его использовании.

Заключение

Процесс создания веб-сайта о сериале "Фишер" был сопровожден внимательным анализом, разработкой и тестированием, чтобы обеспечить качественный и полезный ресурс для пользователей. Для достижения этой цели использовались современные технологии разработки веб-приложений, такие как HTML5, CSS/SCSS и JavaScript.

Перед написанием кода было проведено исследование аналогичных веб-сайтов, а также изучены новейшие стандарты и документации по использованию технологий. Для удобства разработки кода применялась среда разработки VS Code с соответствующими плагинами.

После завершения этапа разработки веб-сайта было проведено тестирование для выявления и исправления ошибок. Важные компоненты веб-страницы и скрипты были реализованы с учетом общей архитектуры проекта.

В итоге, результатом нашей работы стал уникальный и полезный веб-ресурс, который предоставляет пользователям актуальную информацию о сериале "Фишер" и обеспечивает удобство его использования.