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

Факультет Информационных технологий

Кафедра Информационных систем и технологий

Специальность 1-40 05 01 «Информационные системы и технологии»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КУРСОВОГО ПРОЕКТА**

по дисциплине «Компьютерные языки разметки»

Тема «Web-сайт кинорецензий»

**Исполнитель**

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

подпись, дата

**Руководитель**

Преподаватель-стажёр \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Л.П.Степанова

должность, ученая степень, ученое звание подпись, дата

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

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

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

дата, подпись

Курсовой проект защищен с оценкой

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Л.П.Степанова

подпись дата инициалы и фамилия

**ЗАДАНИЕ**

**на курсовое проектирование**

по дисциплине«Компьютерные языки разметки»

Курс 1 Группа 2 Специальность:1–40 05 01 «Информационные системы и технологии»

Студент: Лях Дмитрий Анатольевич

**1. Тема:** Веб-сайт **«**Web-сайт кинорецензий»

**2. Срок выполнения курсового проекта** : с 14 февраля 2023 г. по 10 мая 2023 г.

**3. Технические требования :**

3.1 Прототип веб-сайта должен быть разработан с использованием графических редакторов Figma/Adobe XD/Sketch.

3.2 Для хранения данных должен быть использован XML-формат.

3.3 Разметка содержания сайта должна быть выполнена с применением HTML5 и XML.

3.4 Для описания внешнего вида веб-страниц использовать Sass/CSS3.

3.5 Веб-сайт должен содержать:

– семантические теги HTML5;

– графические элементы в форме SVG;

– несколько веб-страниц;

– JavaScript для управления элементами DOM.

3.6 Верстка сайта должна быть адаптивной и кроссбраузерной;

3.7 Для тестирования использовать Linter

3.8 Проект и пояснения к проекту должны быть размещены на GitHub.

**4. Содержание пояснительной записки**

1. Титульный лист;

2. Задание на курсовое проектирование;

3. Введение;

4. Постановка задачи;

5. Проектирование веб-сайта

6. Реализация структуры веб-сайта

7. Тестирование веб-сайта

8. Заключение

9. Список использованных источников

10. Приложения (полный исходный текст программы разработанного приложения с подробными комментариями)

**5. Форма представления на GitHub выполненного курсового проекта:**

– Теоретическая часть пояснительной записки курсового проекта должна быть представлена в формате MS Word.

– Оформление записки должно быть согласно правилам.

– Листинги представляются в приложении.

Содержание

Введение

1. Постановка задачи
   1. Обзор аналогичных решений
   2. Техническое задание
   3. Выбор средств реализации программного продукта
   4. Выводы
2. Макетирование страниц веб-сайта
   1. Выбор способы верстки
   2. Выбор шрифтового оформления
   3. Разработка логотипа
   4. Разработка пользовательских элементов
   5. Вывод
3. Реализация структуры Веб-сайта
   1. Структура HTML-документа
   2. Добавление таблиц стилей CSS
   3. Использование стандартов XML(SVG)
   4. Выводы
4. Тестирование веб-сайта
   1. Адаптивный дизайн веб-сайта
   2. Кроссбраузерность веб-сайта
   3. Руководство пользователя
   4. Выводы

Заключение

Список использованных источников

Приложение А

Приложение Б

Приложение В

Приложение Г

Приложение Д

Введение

В настоящее время количество новых сайтов увеличивается с неимоверной скоростью. Именно поэтому, основной задачей, ставящейся при разработке сайта – сделать его максимально привлекательным для потенциальных пользователей и придать ему индивидуальность. На заре развития интернета можно было привлечь достаточно большое количество посетителей на сайт, не прикладывая к его разработке больших усилий и без особых материальных затрат. Сейчас же, в условиях возросшей конкуренции, функциональность сайта, его стиль и запоминаемость играют очень большую роль. Ведь независимо от того создается сайт для представления какой-либо фирмы или просто для заработка на рекламе, основной его функцией является привлечение как можно большей аудитории. Поэтому для достижения успеха важен правильный подход к выбору стратегии и исполнителей.

Профессиональные студии веб – дизайна предлагают создание сайтов различного предназначения, это может быть корпоративный сайт, сайт – визитка, интернет – магазин, сайты для ведения бизнеса как онлайн, так и в реальном мире.

Наше время называют «информационным веком». Это название возникло потому, что самым важным, ценным и необходимым ресурсом является информация. Обладание информацией и умение своевременно, быстро и четко донести необходимую информацию до клиентов – ключ к успеху в реалиях современного бизнеса. В настоящий момент именно Интернет, способен оперативно и массово передавать текст, звук, изображения и даже видео-ролики, т.е. практически любую информацию. C точки зрения пользователя, Интернет – это огромный информационный ресурс, в котором можно найти все, что угодно: от прогноза погоды до личных предпочтений голливудских кинозвезд. Аудитория сети Интернет растет с каждым годом. Интернет – самое быстроразвивающееся средство передачи информации за все историю человечества. Пользователи сети интернет являются целью особого интереса для рекламодателей, потому что факт остается фактом – среди пользователей сети куча потенциальных клиентов. Актуальность сайта кинорецензий состоит в том что каждый человек хочет получить удовольствие от просмотра того или иного фильма без мысли о зря потраченном времени и средствах. А кинорецензии в свою очередь дают общую картину фильма с его оценкой, что лишь упрощает выбор потенциальным зрителям.

Цель курсовой работы: разработать веб-сайт театра с использованием HTML5 и XML, а также с применением CSS3.

Задачи:

1. Проанализировать существующие языки разметки, инструменты и библиотеки для создания веб-сайта.
2. Разработать макет и прототип сайта.
3. Разработать структуру веб-сайта.
4. Наполнить сайт информацией по теме.
5. Протестировать веб-сайт.
6. Разработать руководство пользователя.

1 Постановка задач и обзор аналогичных решений

1.1Обзор аналогичных решений

Веб-сайт – это набор взаимосвязанных веб-страниц, которые представляют собой информацию, предназначенную для просмотра в браузере. Веб-сайт представляет собой совокупность веб-страниц, которые могут быть связаны между собой посредством гипертекстовых ссылок.

Для создания быстрого и динамического веб-сайта необходимо использовать современные технологии, которые позволяют создавать веб-сайт с минимальными затратами времени и средств. Веб-сайт должен быть удобным для пользователя, а также должен быть адаптирован под мобильные устройства. Тема веб-сайта должна быть востребованной, а также должна быть актуальной. Поэтому было принято решение разработать кинорецензий, который будет представлять собой веб-сайт, на котором можно будет посмотреть краткие описания фильмов с его оценкой. Прежде чем приступить к разработке веб-сайта, необходимо проанализировать аналоги, чтобы понять, какие технологии используются для создания веб-сайтов, а также какие есть недостатки и преимущества веб-сайтов аналогов. И самое главное найти в каждом аналоге свои уникальные особенности, которые будут отличать этот сайт от других. Для этого будет рассмотрены 3 аналога.

Аналог №1—сайт Кинопоиск[1]

В качестве первого аналога возьмем кино-сервис «Кинопоиск». На рисунке 1.1 представлена главная страница сайта.

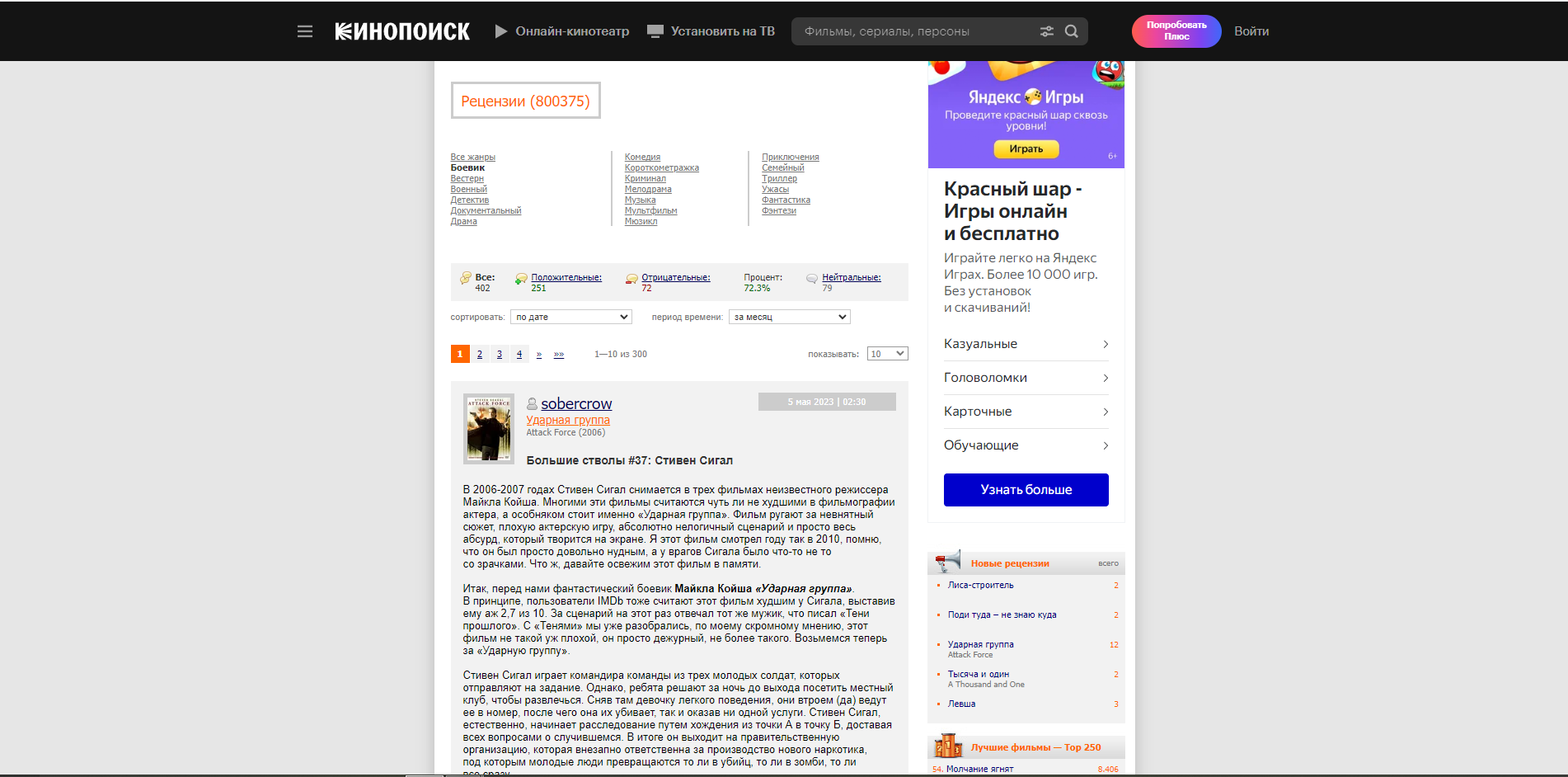


Рисунок 1.1 – Оформление кинорецензий сайта «Кинопоиск»

Сайт ориентируется на людей которые хотят узнать отзывы и рецензии к фильму или же оставить собственные отзывы. Сайт сделан в вертикальном виде, имеются переходы для поиска фильмов по жанрам, отображение недавних обзоров и сортировки по датам.

Шрифт в целом как и оформление не имеют особых стилевых решений, цвета тусклые от чего он кажется не столь привлекательным.

Данный сайт является среднего качества, но при этом сайт является достаточно информативным, подача содержимого интуитивна понятна пользователю.

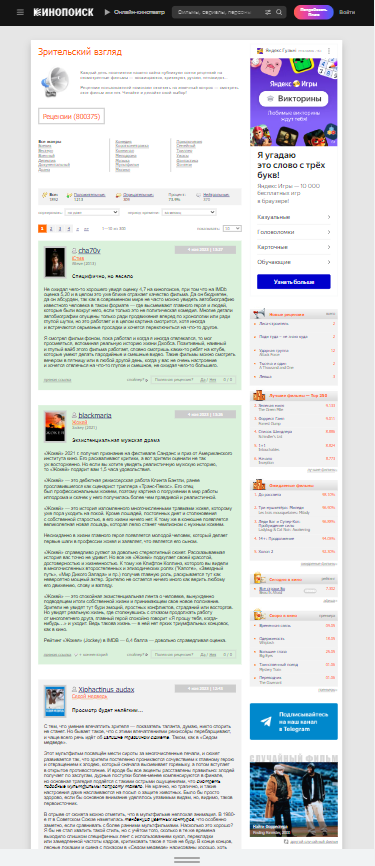
Однако при всем при этом мобильная версия этого сайта оставляет желать лучшего. По своей сути это лишь отмасштабированная версия для персональных компьютеров и ноутбуков, что очень не удобно для пользователей мобильных устройств. Вид сайта представлен на рисунке 1.2. 

Рисунок 1.2 – Мобильная версия кинорецензий сайта «Кинопоиск»

Аналог№2-веб-сайт «film.ru»[2]

Следующий аналог – сайт «film.ru». Оформление главной страницы представлено на рисунке 1.3.

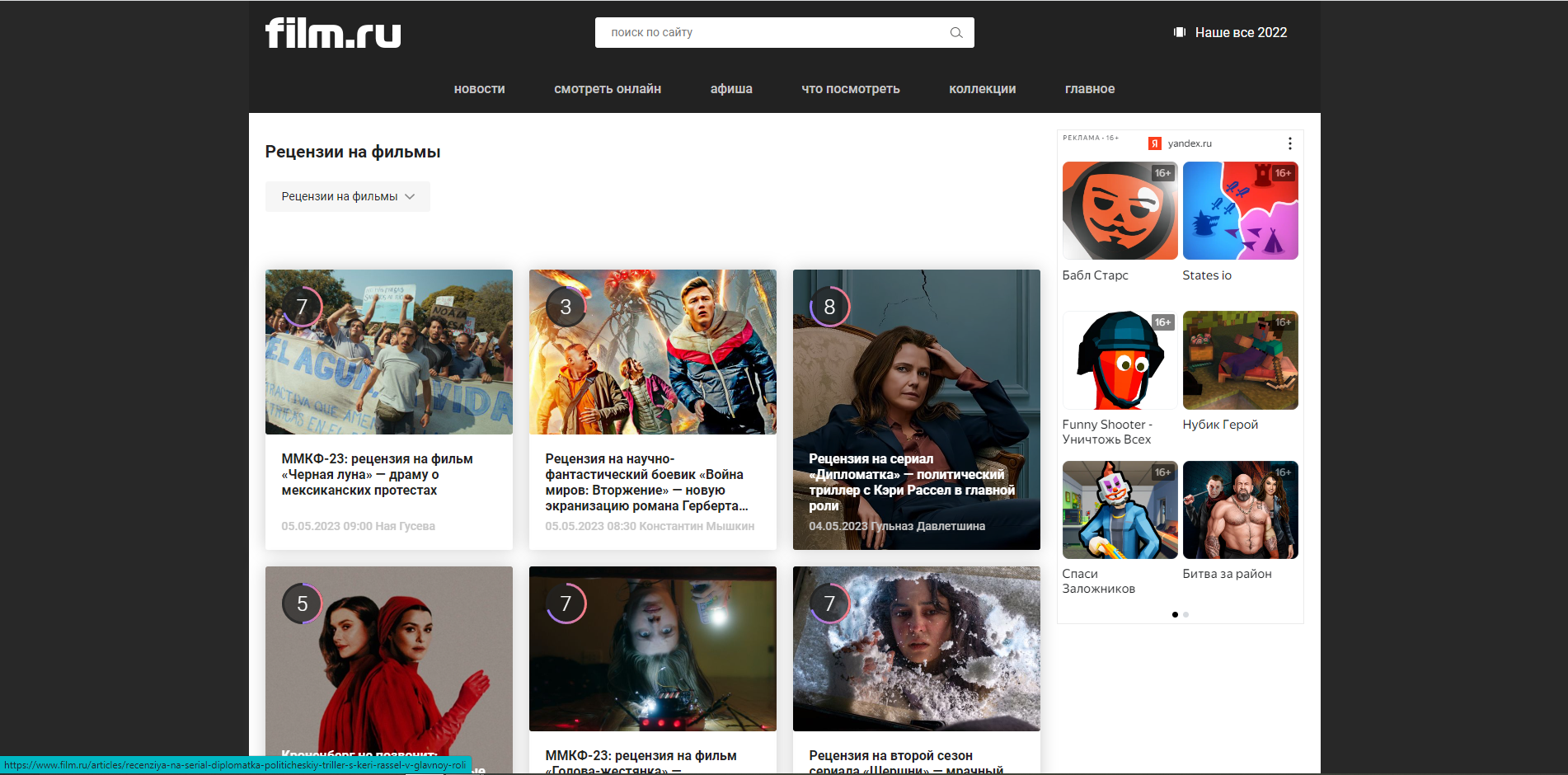


Рисунок 1.3 – Главная страница рецензий сайта «film.ru»

Это сайт уже встречает нас приятным минималистичным оформлением: Шапка, навигационное меню и главная информация располагаются друг за другом благодаря чему легко ориентироваться по сайту. На главной странице представлены последние обзоры с общей оценкой и заглавием. Текст как и цвета не вызывают дискомфорта при просмотре.

Мобильная версия ничем не уступает компьютерной версии: Выпадающее меню в углу экрана, возможность горизонтальной прокрутки навигационного меню, те же стили и оформление. Мобильная версия представлена на рисунке 1.4



Рисунок 1.4 – Мобильная версия страницы сайта «film.ru»

Аналог№3-Веб-сайт «Exler.ru»[3]

Последний аналог – сайт «Exler.ru». Оформление представлено на рисунке 1.5.



Рисунок 1.5 – Главная страница сайта «Exler.ru»

Дизайн сайта имеет приятные цвета которые не вызывают дискомфорта при просмотре. Все ссылки интуитивно понятны для пользователя. Актуальная информация показывается в центре страницы.

Мобильная версия выглядит также достаточно неплохо: Все поисковые ссылки перемещены на удобное расположение, размер текста и изображений более чем соответствующие. Рисунок мобильной версии представлен нас рисунке 1.6.

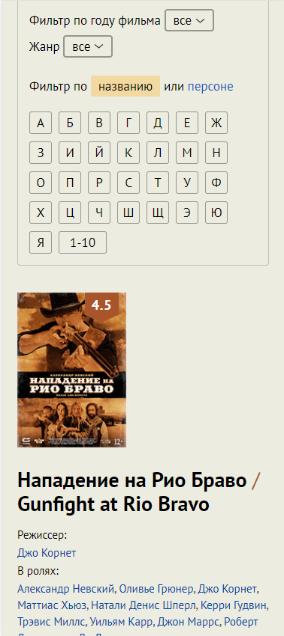


Рисунок 1.6 – Мобильная версия сайта «Exler.ru»

1.2 Техническое задание

На основе рассмотренных вариантов была поставлена задача разработать многостраничный сайт на тему «Web-сайт кинорецензий». На сайте должен быть представлен список имеющихся обзоров, разделение на жанры, поиск нужных фильмов, просмотр рецензий.

Основными задачами сайта являются:

1. предоставление навигационного меню;
2. предоставление информации о фильме;
3. предоставление рецензий;

На главной странице меню будет расположено сверху. В меню будут предоставляться ссылки на основную информацию. На боковом меню будут расположены вспомогательные ссылки для пользователя.

1.3 Выбор средств реализации программного продукта

Курсовая работа выполняется в редакторе кода Visual Studio Code.

При создании веб-страниц были использованы языки: HTML, CSS, JS, XML.

HTML (HyperText Markup Language) — стандартный язык разметки гипертекстовых страниц в Интернете. Страницы успешно интерпретируются браузерами, которые отображают их на экранах различных электронных устройств в удобном для человека виде.

CSS — это фактически язык стилей, который определяет отображение HTML-документов. CSS работает со шрифтами, с цветами символов и фона, с полями, со строками, с высотой и с шириной элементов отображения, с фоновыми изображениями, с позиционированием элементов и со многим другим.

Если HTML необходим для структурирования содержания страницы, то CSS необходим для того, чтобы форматировать это структурированное содержание.

JavaScript — это язык, который позволяет применять сложные вещи на веб-странице — каждый раз, когда на ней происходит что-то большее, чем просто её статичное отображение — без JavaScript не обошлось. Во все основные браузеры встроен интерпретатор JavaScript, именно поэтому они могут выполнять скрипты на странице. JavaScript можно использовать не только в браузере, это полноценный язык, программы на котором можно запускать и на сервере.

1.4 Вывод

В данном разделе были рассмотрены плюсы и минусы аналогов на выбранную тему сайта для предотвращения ошибок в реализации веб-сайта. Были определены основные технические задания по созданию страницы. Также выбраны средства реализации программного продукта, а именно языки HTML, CSS, JS.

2. Макетирование страниц веб-сайта

2.1 Выбор способа вёрстки

Требования к проекту включают адаптивность и кроссбраузерность, поэтому была выбрана flex-верстка, которая будет применятся для основных элементов страницы.

Flex – это новая технология, которая уже имеет достаточно широкую поддержку браузеров. Flexbox предоставляет инструменты для быстрого создания сложных, гибких макетов, и функции, которые были сложны в традиционных методах CSS.

2.2 Выбор шрифтового оформления

В данном проекте был использован шрифт Georgia для более приятного восприятия информации для пользователя.

Georgia является одним из шрифтов с засечками. Полужирное начертание у шрифта очень широкое и насыщенное.

2.3 Разработка логотипа

Разработка логотипа осуществлялась в графическом редакторе Fdobe Illustrator. Дизайн подобран исходя из главной тематики сайта. Небольшая сложность была сделана в пользу оригинальности логотипа.Логоти представлен на рисунке 2.1.

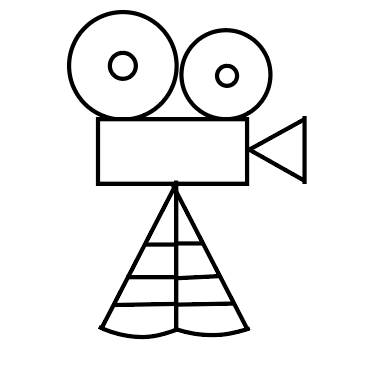


Рисунок 2.1- Логотип сайта

2.4 Разработка пользовательских элементов

В проекте буду представлены такие пользовательские элементы как навигационное меню на основную информацию, поле ввода для поиска нужных названий и дополнительное меню для выбора по нужным категориям.

Навигация – основной элемент для удобства сайта, это так называемая «карта», позволяющая перемещаться между различными страницами ресурса. Навигационное меню представлено на рисунке 2.2.



Рисунок 2.2 – Навигационное меню главной страницы

Также рассмотрим пользовательский элемент для поиска фильмов по названию. Он состоит из поля ввода и кнопки при нажатии на которую осуществляется поиск. На рисунке 2.3 показана реализация этого элемента.



Рисунок 2.3 – Поле поиска фильмов

2.5 Вывод

Выводом проделанной работы стало создание прототипов, а затем и макетов сайта для облегчения последующей верстки, определены задачи по созданию содержания на веб-сайте. Было упорядочено расположение всех элементов на страницах, разработан дизайн. Подобрали подходящее стилевое оформление сайта, то есть основная цветовая гамма, шрифтовое оформление и элементы пользовательского интерфейса.

3. Реализация структуры веб-сайта

3.1 Структура HTML-документа

Структура сайта – это логическое построение всех страниц и разделов ресурса.

Благодаря ей пользователь понимает, где располагаются основные разделы сайта, как перемещаться между страницами или вернуться на главную**.**

В начале документа присутствует тег <head></head>, в котором прописана служебная информация и название веб-страницы. Содержимое тега <head> не отображается напрямую на веб-странице, исключением является тег <title></title>, который используется для заголовка окна веб-страницы.

Далее присутствует тег <body></body>, в котором размещены теги веб-страницы, которые и задают веб-сайт.

Структуру тела составляют следующие семантические теги: header, nav, section, main, footer. Семантические теги предназначены для того, чтобы компьютерные программы (поисковые системы, сборщики информации, речевые браузеры и т. д.), понимали какой тип информации заложен в данных тегах.

Для создания шапки всех страниц был использован тег header, внутри которого размещен тег nav, который используется для навигационного меню.

Для размещения страницы и разметки главной информации были использованы теги section и main. “Подвал” страницы создан с помощью семантического тега footer.

HTML, CSS, XML и Sass код сайта был протестирован с помощью онлайн-сервиса linter. Результат представлен на рисунке 3.1.

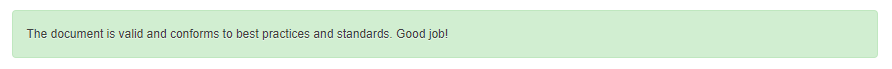


Рисунок 3.1 – Результат тестирования с помощью linter

3.2 Добавление таблиц стилей CSS

Почти все элементы на сайте должны иметь визуальное оформление. Для визуального оформления использован CSS/SCSS. Они отвечают за то, как HTML-элементы будут выведены пользователю в браузере. Существует несколько способов встраивания и подключения стилей: внешнее, внутреннее, строковое. В данном проекте используется внешнее подключение стилей с помощью тега link. Стили хранятся в отдельном файле, который может быть использован для любых веб-страниц. Данный способ один из самых удобных в реальной разработке. Так как стилей в проекте обычно много, то держать их в HTML-файле не очень удобно.

Связано это с тем, что HTML-файл становится очень большим и ориентироваться в нем становится невозможно. Чтобы этого избежать, можно создать отдельный CSS-файл.  В таблицах стилей используются селекторы по тегу, по классу, по идентификатору.

3.3 Использование стандартов XML (SVG)

SVG – это двухмерная графика и тем не менее это текстовый формат, который можно легко править в блокноте или просто рисовать в векторных редакторах. SVG одинаково хорошо выглядит без потери качества и на мобильном устройстве, и на стационарном мониторе домашнего ПК. Поэтому для создания логотипа был выбран формат SVG.

Расширяемый язык разметки позволяет определять и хранить данные совместно используемым способом. XML поддерживает обмен информацией между компьютерными системами, такими как веб-сайты, базы данных и сторонние приложения. Предопределенные правила упрощают передачу данных в виде XML-файлов по любой сети, поскольку получатель может использовать эти правила для точного и эффективного чтения данных. В проекте одна из страниц содержит XML-формат для хранения информации, используемой на странице.

3.4 Выводы

На этапе была реализована структура сайта на выбранную тему с использованием HTML, созданы и применены таблицы стилей Sass/CSS. Использованы стандарты XML для хранения данных, а также SVG для создания масштабируемого логотипа. Продемонстрировано использование языка JavaScript для динамики сайта. Определено стилевое оформление сайта, добавлены анимации, сайт полностью готов к следующему этапу – тестированию.

4. Тестирование веб-сайта

4.1. Адаптивный дизайн веб-сайта

Адаптивность — одно из ключевых требований, предъявляемых к современным сайтам. Ресурс должен одинаково хорошо демонстрироваться на экране компьютера, планшете, смартфоне. Для этого создается адаптивный дизайн и используется адаптивная верстка. В итоге процесс веб-разработки усложняется, требует дополнительных ресурсов. Но результат стоит вложений: сайт отлично позиционируется на всех основных типах устройств, что гарантирует полноценный охват аудитории.

Для адаптивности использована flex-верстка для элементов страницы, а также ширина блоков задается автоматически в пикселях, что позволяет им подстраиваться под ширину экрана пользователя.

Рисунок 4.1 – Вид шапки сайта компьютерной версии

Flex-верстка используется для шапки сайта (рисунок 4.2).Использованы медиа-запросы для изменения внешнего вида шапки для размещения блоков в необходимом порядке.



Рисунок 4.2 – Вид шапки сайта мобильной версии

Flex-верстка также используется для рецензий, представлена на рисунке 4.3. Элементы адаптируются под ширину экрана пользователя.

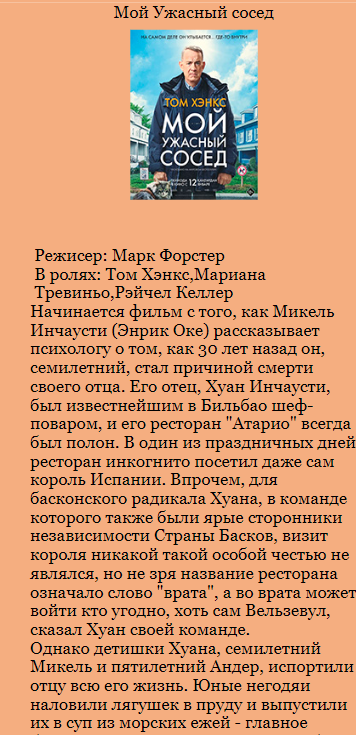


Рисунок 4.3 – Мобильная версия рецензий

В итоге сайт отлично позиционируется на всех основных типах устройств, что гарантирует полноценный охват аудитории.

4.2. Кроссбраузерность веб-сайта

Кроссбраузерность – свойственная веб-сайту характеристика, подразумевающая его корректное отображение во всех популярных браузерах. Проще говоря – это совместимость вашего сайта с приложениями-клиентами. Тестирование в браузере Google Chrome представлено на рисунке 4.4, в браузере Microsoft Edge на рисунке 4.5.

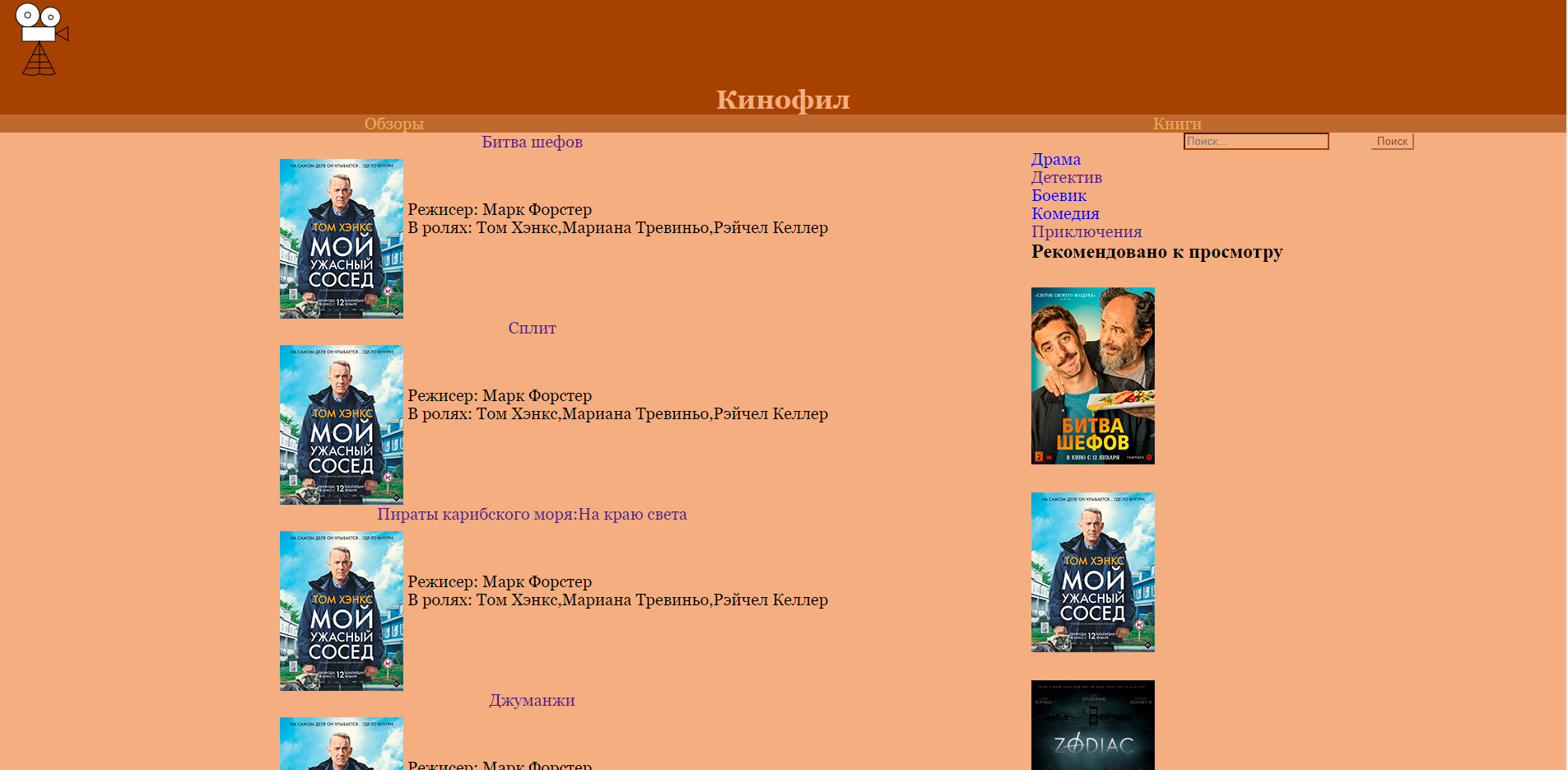


Рисунок 4.4 – Вид главной страницы в браузере Google Chrome

Сайт не обязательно должен быть попиксельно одинаковым во всех браузерах. Изучив статистику аудитории, следует верстать сайт и для самых популярных версий некоторых браузеров.

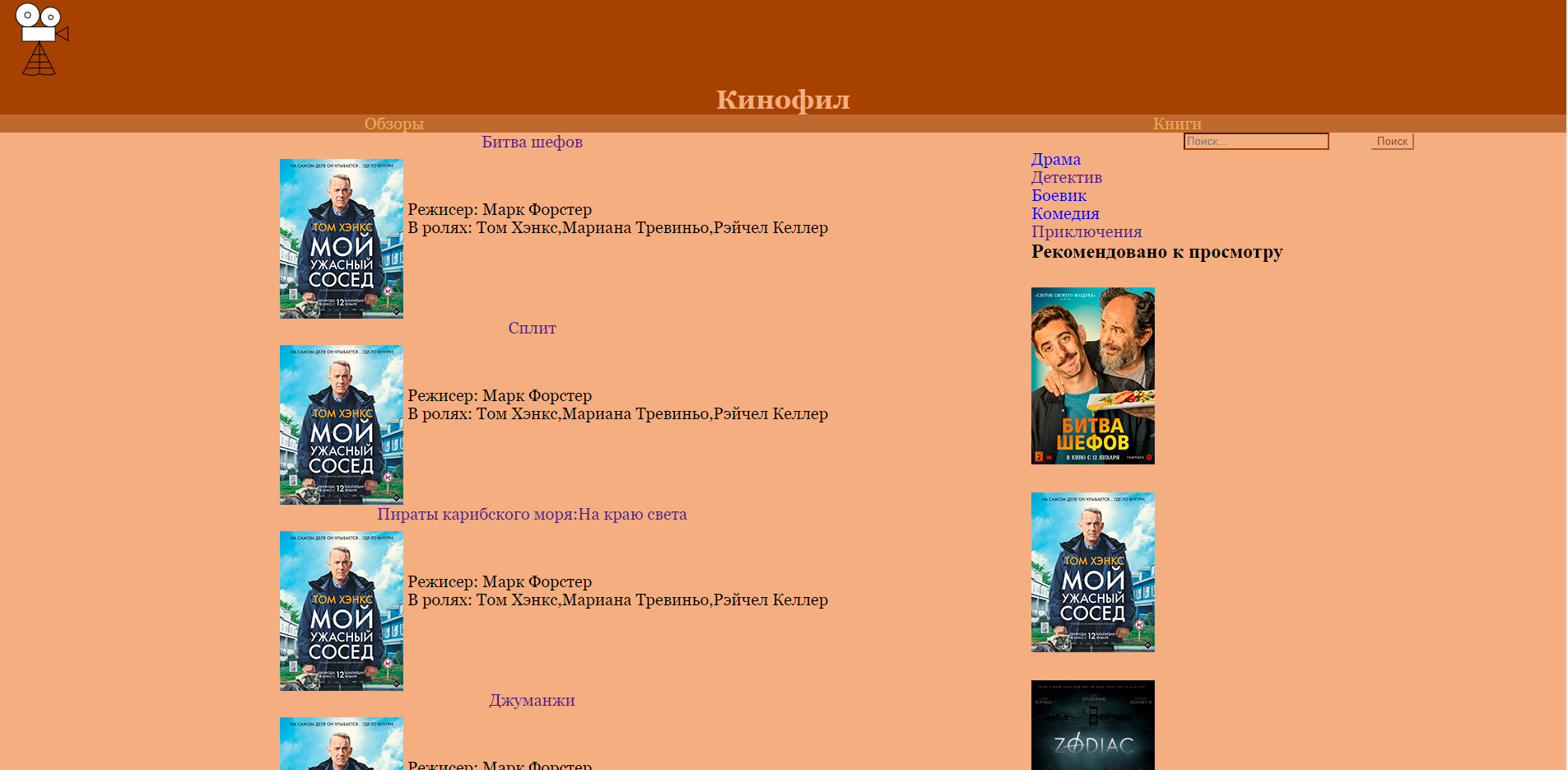


Рисунок 4.5 – Вид главной страницы в браузере Microsoft Edge

Сайт не обязательно верстать под все существующие браузеры и делать его отображение строго одинаковым во всех. Достаточно убедиться в нескольких моментах:

1. информация читабельна и выводится правильно, элементы не перекрывают друг друга и не съезжают;
2. отображается вся необходимая информация;
3. сохранена структура сайта;
4. отсутствуют недостатки вёрстки.

Сделать сайт кроссбраузерным — значит добиться схожей (не обязательно до мелочей) визуализации контента в популярных браузерах. Достичь такого результата можно несколькими путями.

У каждого браузера есть свои встроенные, экспериментальные или нестандартные свойства и для того, чтобы они корректно работали было принято решения о внедрении вендорных префиксов.

4.3. Руководство пользователя

Интерфейс сайта интуитивно понятен пользователю. Ненужные элементы отсутствуют, что делает сайт простым и удобным.

На главной странице присутствует шапка страницы, используя её пользователя может попасть на любую из основных вкладок.



Рисунок 4.6 – Шапка сайта

На каждой странице присутствует навигационное меню для облегченного поиска по жанрам и рекомендации к просмотрам, которые являются обложками содержащими ссылки.

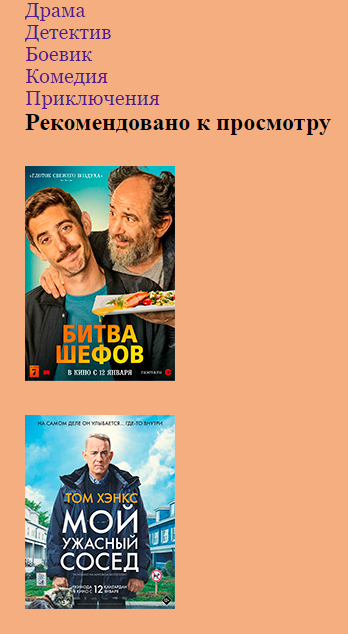


Рисунок 4.7 – Боковое меню сайта

При переходе на рецензию фильма пользователя переносит на страницу с рецензией, где также указана информация о фильме.



Рисунок 4.8 – Страница с рецензией

4.4. Выводы

В данном разделе была рассмотрена адаптивность сайта, после проверки на нескольких устройствах были найдены ошибки, которые постепенно исправлялись различными способами. Рассмотрена кроссбраузерность, в результате которой выяснилось, что сайт некорректно отображается в некоторых браузерах, данная проблема была исправлена. Разработано руководство пользователя, в котором описана функциональность сайта и принцип работы его блоков.

# Заключение

Целью курсового проекта является реализация полноценного многостраничного веб-сайта кинорецензий. Сайт предоставляет доступ к информации для людей разного возраста и разных интересов. Сделан удобный и практичный пользовательский интерфейс, сайт интуитивно понятен пользователю, адаптивен и является кроссбраузерным.

Сайт состоит из 14 страниц: главная страница фильмов, рецензия фильма, главная страница книг, рецензия книги, по 5 страниц с жанрами для книг и фильмов.

Для реализации макета страницы проведен анализ других сайтов кинорецензий, выбрана самая нужная для пользователя информация, проанализирован интерфейс, элементы взаимодействия с пользователем.

Для написания кода веб-страницы использована среда разработки Visual Studio Code, в которой установлены плагины и расширения для удобной и облегченной разработки. Были разработаны пользовательские элементы, спецэффекты и логотип, для идентификации веб-сайта, среди миллионов других интернет ресурсов.

Перед написанием кода, были изучены новые стандарты языков разметки и программирования. Были прочтены документации по использованию технологий, которые будут использоваться на этапе написания сайта.

При проектировании были разработаны прототипы, а далее созданы макеты дизайна страниц веб-сайта, при опоре на который был реализован сам веб-сайт. Для разработки прототипов и макетов станиц было использована программа Figma, ранее не изученная в ходе учебного процесса.

Реализация важных элементов страницы представлены в приложениях. Которые дают общую информацию о проекте.

Для выполнения поставленной цели курсового проекта были реализованы следующие задачи:

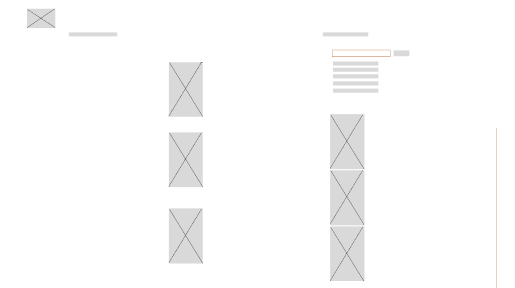
1. анализ аналогов;
2. анализ существующих языков разметки, инструментов библиотеки;
3. разработка макета и прототипа;
4. выбор способа верстки;
5. выбор стилевого оформления;
6. разработка логотипа;
7. разработка основной структуры сайта;
8. разработка пользовательских элементов;
9. разработка спецэффектов;
10. добавление веб-сайту адаптивности;
11. кроссбраузерность веб-сайта;
12. тестирование веб-сайта;
13. создание руководства пользователя;

В результате проделанной работы был подготовлен веб-сайт театра. Сайт работает стабильно, имеет полный функционал, и в дальнейшем он может подлежать эксплуатации.

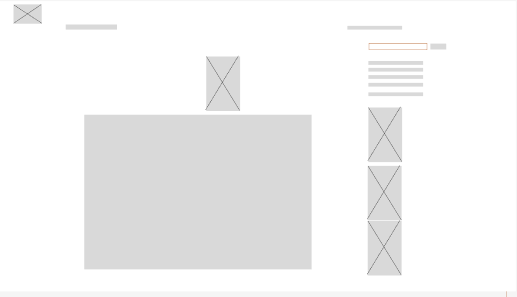
Список использованных источников

1. Веб-сайт «Кинопоиск» [Электронный ресурс]. — Режим доступа: <https://www.kinopoisk.ru/reviews/>
2. Веб-сайт «Film.ru» [Электронный ресурс]. — Режим доступа: <https://www.film.ru/topic/review>
3. Веб-сайт «Exler.ru» [Электронный ресурс]. — Режим доступа: <https://exler.ru/films/>
4. Документация по Figma [Электронный ресурс]. — Режим доступа: <https://help.figma.com>
5. Документация по Sass/SCSS [Электронный ресурс]. — Режим доступа: <https://sass-scss.ru/guide/>
6. Документация по JavaScript/HTML [Электронный ресурс]. — Режим доступа: <https://developer.mozilla.org>
7. Введение в кроссбраузерное тестирование [Электронный ресурс]. — Режим доступа: <https://developer.mozilla.org/ru/docs/Learn/Tools_and_testing/Cross-browsertesting/Introduction>.

Приложение А

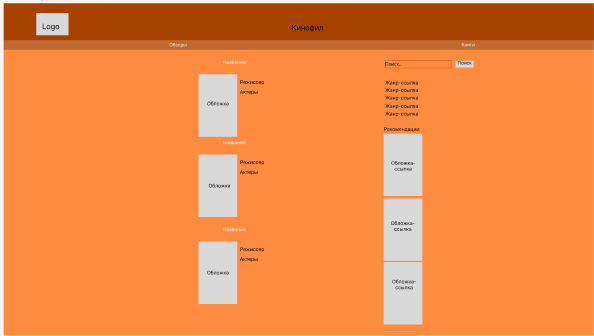


Прототип главной страницы

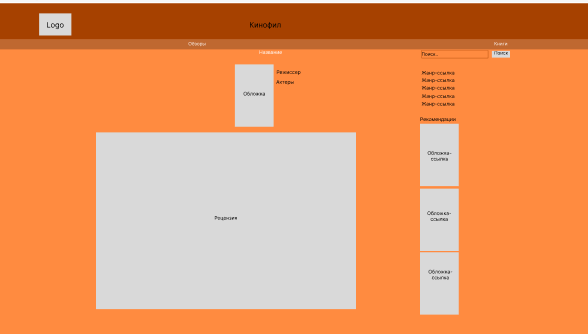


Прототип страницы рецензий

Приложение Б



Макет главной страницы



Макет страницы рецензии



Мобильной версии

Приложение В

nav{

background-color: #BF6830;

display: flex;

flex-direction: row;

justify-content: space-around;

}

main{

height: auto;

display: flex;

flex-direction: row-reverse;

}

aside{

height: auto;

width: auto;

background-color:#f5ae80;

}

aside a{

padding-right: 500px;

text-align: left;

}

aside input{

align-self: left;

}

aside button{

align-self: left;

}

article{

height: 1500px;

width: 1700px;

background-color: #f5ae80;

}

footer{

background-color: #A64100;

text-align: left;

height: 50px;

}

Листинг CSS главной страницы

Приложение Г

let input = document.getElementById("Search");

let filter = input.value.toUpperCase();

let ul = document.getElementById("list");

let li = ul.getElementsByTagName("li");

for (let i = 0; i < li.length; i++) {

let a = li[i].getElementsByTagName("a")[0];

if (a.innerHTML.toUpperCase().indexOf(filter) > -1 && a.innerHTML.toUpperCase()===filter) {

li[i].style.display = "";

}else{

li[i].style.display ="none";

}

}

Листинг Javascript поиск по имени

Приложение Д

<header>

<img src="logo.svg" id="log">

<h1 id="demo"></h1>

</header>

<nav>

<a href="course.html">Обзоры</a>

<a href="page1.html">Книги</a>

</nav>

<main>

<aside>

<input type="text" id="Search" placeholder="Поиск..." title="Type in a category"><button onclick="search()">Поиск</button>

<a href="drama.html">Драма</a>

<a href="detective.html">Детектив</a>

<a href="action.html">Боевик</a>

<a href="comedy.html">Комедия</a>

<a href="adventure.html">Приключения</a>

<div class="rek">

<h2>Рекомендовано к просмотру</h2>

<a href="film1.html"><img src="comedy.jpg"></a>

<a href="film1.html"><img src="test.jpg"></a>

<a href="film1.html"><img src="zod.jpg"></a>

</div>

</aside>

<article>

<ul id="list">

<li><a href="film1.html">Битва шефов</a>

<div class="rew">

<img src="test.jpg">

<div class="info">

<p>Режисер: Марк Форстер<p>

<p>В ролях: Том Хэнкс,Мариана Тревиньо,Рэйчел Келлер </p>

</div>

</div>

</li>

<li><a href="film1.html">Сплит</a>

<div class="rew">

<img src="test.jpg">

<div class="info">

<p>Режисер: Марк Форстер</p>

<p>В ролях: Том Хэнкс,Мариана Тревиньо,Рэйчел Келлер </p>

</div>

</div>

</li>

<li><a href="film1.html">Пираты карибского моря:На краю света</a>

<div class="rew">

<img src="test.jpg">

<div class="info">

<p>Режисер: Марк Форстер</p>

<p>В ролях: Том Хэнкс,Мариана Тревиньо,Рэйчел Келлер </p>

</div>

</div>

</li>

<li>

<a href="film1.html">Джуманжи</a>

<div class="rew">

<img src="test.jpg">

<div class="info">

<p>Режисер: Марк Форстер</p>

<p>В ролях: Том Хэнкс,Мариана Тревиньо,Рэйчел Келлер </p>

</div>

</div>

</li>

<li><a href="film1.html">Пиксели</a>

<div class="rew">

<img src="test.jpg">

<div class="info">

<p>Режисер: Марк Форстер</p>

<p>В ролях: Том Хэнкс,Мариана Тревиньо,Рэйчел Келлер </p>

</div>

</div>

</li>

<li><a href="film1.html">Мой ужасный сосед</a>

<div class="rew">

<img src="test.jpg">

<div class="info">

<p>Режисер: Марк Форстер</p>

<p>В ролях: Том Хэнкс,Мариана Тревиньо,Рэйчел Келлер </p>

</div>

</div>

</li>

</ul>

Листинг HTML главной страницы