Министерство образования и науки Российской Федерации

**Муромский институт (филиал)**

федерального государственного бюджетного образовательного учреждения высшего образования

**«Владимирский государственный университет**

**Имени Александра Григорьевича и Николая Григорьевича Столетовых»**

**(МИ ВлГУ)**

Факультет ИТР

Кафедра: ПИн

Курсовая работа

По: *WEB Технологии*

Тема: Онлайн киносервис

Руководитель



Рыбкин И. А.

(Оценка)

(фамилия, инициалы)

(подпись) (дата)

Члены комиссии

Студент ПИН-122

(группа)

Сергеев Д.Д.

(Подпись Ф.И.О.)

(фамилия, инициалы)

(Подпись Ф.И.О.)

(подпись) (дата)

Муром 2025

В рамках данной работы была создана информационная система хранения и поиска фильмов и модели данных. Система включает в себя базу данных, в которой содержится информация о фильмах, их режиссерах, актерах, пользователях, записях о бронировании и возврате фильмов, а также о запросах на добавление новых фильмов в коллекцию. Также было разработано приложение для администраторов, в котором можно удобно работать с данными в таблицах базы данных через графический интерфейс.

Работа состоит из вводной части, анализа технического задания, разработки моделей данных, разработки и реализации информационной системы, тестирования системы, заключения, списка литературы и приложений.

As part of this work, an information system for storage and search of movies and data models were created. The system includes a database that contains information about movies, their directors, actors, users, records of bookings and returns, as well as requests for adding new movies to the collection. An application for administrators was also developed, which allows for convenient interaction with the data in the database tables through a graphical interface.

The work consists of an introductory section, an analysis of the technical requirements, development of data models, development and implementation of the information system, system testing, conclusion, a list of references, and appendices.

**Содержание**

[**Введение** 5](#_Toc186159678)

[**1. Анализ технического задания** 5](#_Toc186159679)

[**2.** **Описание языков программирования, применяемых на сайте** 7](#_Toc186159680)

[**3.** **Выбор и обоснование архитектуры системы** 11](#_Toc186159681)

[**4. Выбор и обоснование алгоритмов** 13](#_Toc186159685)

[**5.** **Описание программы** 14](#_Toc186159686)

[**Заключение** 16](#_Toc186159689)

[**Список литературы** 17](#_Toc186159690)

[**Приложения** 18](#_Toc186159691)

[**Приложение А** 17](#_Toc186159692)

[**Приложение Б** 18](#_Toc186159693)9

[**Приложение В** 20](#_Toc186159694)

**Введение**

В условиях стремительного развития информационных технологий и увеличения числа пользователей интернета, онлайн-сервисы кино и видео содержания стали неотъемлемой частью современной культуры. Одной из наиболее динамично развивающихся сфер онлайн-контента является услуга по стримингу кино и видео, что обусловлено высоким спросом на легкую и удобную доступность контента и постоянным обновлением ассортимента. Онлайн-платформа кино и видео представляет собой удобную платформу для посетителей, позволяющую им в любое время и из любого места ознакомиться с широким ассортиментом кино и видео, сравнить цены и выбрать наиболее подходящую платформу под свои вкусы и бюджет.

Целью данной курсовой работы является разработка веб-приложения для онлайн-платформы кино и видео, которое обеспечит эффективное управление каталогом контента, автоматизацию процессов подачи и учета доступов, а также анализ поведения клиентов. В процессе работы будет рассмотрен функционал системы, включая интеграцию с платежными шлюзами и логистическими сервисами для обеспечения быстрого и простого доступа к содержанию, а также оптимизацию бизнес-процессов.  
Особое внимание будет уделено вопросам безопасности данных, конфиденциальности клиента и удобства пользовательского интерфейса, так как эти аспекты играют ключевую роль в формировании доверия со стороны клиентов и их лояльности к платформе. В результате реализации проекта планируется создать интуитивно понятный и функциональный онлайн-платформу, которая сможет удовлетворить потребности самых различных категорий клиентов и эффективно конкурировать на рынке онлайн-контента.  
Таким образом, данная курсовая работа не только охватывает теоретические аспекты разработки веб-приложений, но и предоставляет практические рекомендации по созданию успешной онлайн-платформы кино и видео, способной адаптироваться к быстро меняющимся условиям рынка онлайн-контента.

**1. Анализ технического задания**

Задание – разработать веб-приложение для онлайн киносервиса

1) Локальный сервер и программная оболочка

2) Создать базу данных текстов, информации об изображениях и разделах

3) Использовать загрузку и выгрузку информации в/из базы данных на страницу сайта

4) Создать кроссбраузерные, блочные шаблоны страниц (отражающие не менее трех информационных блока)

5) Динамическое изменение внешнего вида и/или информации на странице сайта

6) Обязательно использование комментариев к функциям динамического ввода/вывода

Для разработки веб-приложения потребуется локальный сервер, который будет эмулировать работу реального веб-сервера. Это может быть достигнуто с помощью таких решений, как XAMPP, WAMP или MAMP, которые позволяют разрабатывать и тестировать приложения на локальном компьютере.

Программная оболочка должна поддерживать необходимые языки программирования и технологии, такие как HTML, CSS, JavaScript для фронтенда и PHP, Python, Node,js для бэкенда. Это обеспечит возможность полноценной работы приложения и тестирования всех его функций без необходимости размещения в интернете. Необходимость создания базы данных подразумевает использование системы управления базами данных (СУБД), такой как MySQL или PostgreSQL.

База данных должна содержать таблицы для хранения информации о фильмах(название, описание, жанр, длительность), а также таблицы для хранения данных учетных записей пользователей, администраторов. Это позволит структурировать данные и упростить их управление. Решение задания выполняется с помощью различных технологий и инструментов, таких как HTML, CSS для верстки страниц, языков программирования JavaScript и PHP для создания функционала пользовательской и серверной частей, а также используется фреймворк React, позволяющий легко и быстро создавать пользовательский интерфейс. Для хранения данных выбрана СУБД MySQL.

Проект, состоящий из множества динамических страниц сайта, представляет из себя веб-приложение. При запуске приложения открывается домашняя страница со списком фильмов. Можно смотреть контент только после авторизации на сервисе, которая производится в окне профиля.

**2. Описание языков программирования, применяемых на сайте**

Данное веб-приложение разработано с помощью язык разметки HTML, языка стилей CSS, а также языков программирования JavaScript и PHP.

HTML - это стандартный язык разметки, используемый для создания и структурирования контента на веб-страницах. Он позволяет разработчикам определять различные элементы, такие как заголовки, абзацы, ссылки, изображения и другие медиафайлы. HTML использует теги (например, <h1>, <p>, <a>, <img>) для обозначения структуры документа. Основная задача HTML - обеспечить семантическую разметку и доступность контента для пользователей и поисковых систем. HTML был разработан в начале 1990-х годов Тимом Бернерсом-Ли, британским ученым, который считается одним из основателей Всемирной паутины. Первая версия HTML (HTML 1.0) была представлена в 1993 году. Она содержала базовые теги для разметки текста, ссылок и изображений.

CSS - это язык стилей, который используется для описания внешнего вида и форматирования HTML-документов. С помощью CSS разработчики могут управлять цветами, шрифтами, размерами, отступами и другими аспектами визуального представления веб-страниц. CSS позволяет создавать адаптивные и отзывчивые дизайны, которые хорошо выглядят на различных устройствах. Основные концепции CSS включают селекторы, свойства и каскадирование, что позволяет применять стили к элементам на основе их иерархии и специфичности. CSS был разработан в 1996 году Хоконгом Вимом Ли, который работал над улучшением внешнего вида веб-страниц. Первая версия CSS (CSS1) была опубликована в 1996 году. Она позволила отделить содержание (HTML) от представления (стили), что значительно упростило процесс разработки.

JavaScript - это высокоуровневый язык программирования, который позволяет добавлять интерактивность и динамическое поведение на вебстраницы. Он выполняется на стороне клиента (в браузере), что позволяет реагировать на действия пользователя, такие как клики, ввод данных и навигация. JavaScript используется для манипуляции DOM (Document Object Model), а также для работы с асинхронными запросами (AJAX), что позволяет загружать данные без перезагрузки страницы. С помощью JavaScript разработчики могут создавать интерактивные элементы, такие как формы, слайдеры и игры. JavaScript был создан Бренданом Эйхом в 1995 году, когда он работал в Netscape Communications Corporation. Изначально язык назывался Mocha, затем переименовывался в LiveScript и, наконец, стал JavaScript. JavaScript быстро стал популярным благодаря своей способности добавлять интерактивность на вебстраницы. В 1997 году он был стандартизирован под названием ECMAScript.

PHP - это серверный язык программирования, который широко используется для создания динамических веб-приложений. Он выполняется на сервере и генерирует HTML-код, который отправляется клиенту. PHP отлично подходит для работы с базами данных, обработки форм и управления сессиями. Он может быть встроен непосредственно в HTML-код, что делает его удобным для разработки веб-страниц с динамическим контентом. PHP поддерживает множество фреймворков (например, Laravel, Symfony), которые упрощают разработку сложных приложений. PHP был создан в 1994 году датским программистом Расмусом Лердорфом. Изначально он разработал набор скриптов для отслеживания посетителей своего личного сайта.

**3. Выбор и обоснование архитектуры системы**Для реализации проекта был выбран подход клиент-серверной архитектуры, который предполагает разделение приложения на две основные части:

Клиентская часть (Frontend): разработана с использованием React — популярной библиотеки для создания пользовательских интерфейсов. React позволяет создавать компоненты, которые легко переиспользовать и тестировать, что упрощает процесс разработки и упрощает поддержку приложения.

Серверная часть (Backend): реализована на PHP с использованием MySQL в качестве системы управления базами данных. PHP отлично подходит для обработки клиентских запросов и взаимодействия с базой данных, а MySQL обеспечивает надежное и эффективное хранение и управление данными.

Выбор данной архитектуры обусловлен следующими факторами:

Современные технологии: Использование React и PHP соответствует актуальным требованиям разработки веб-приложений;

Разделение ответственности: Архитектура клиент-сервер позволяет четко разделить логику представления и бизнес-логику, что значительно упрощает поддержку и развитие приложения;

Сообщество и поддержка: React и PHP имеют широкое сообщество разработчиков, что предоставляет доступ к множеству библиотек, инструментов и ресурсов для решения возникающих задач.

Клиентская часть сайта состоит из нескольких страниц:

Главная страница

На главной странице пользователи увидят карточки фильмов, каждая из которых содержит:

- Изображение: яркое и четкое изображение, позволяющее оценить визуальную составляющую контента;

- Ссылку на страницу с подробным описанием: текстовая ссылка, переводящая на отдельную страницу с детальной информацией;

- Кнопку добавления в избранное: удобная кнопка для быстрого добавления контента в личный список без перехода на страницу описания.

Также на главной странице находятся:

- Заголовок: привлекающий внимание заголовок с названием киноконтента;

- Кнопки входа и регистрации: ссылки для пользователей, которые уже зарегистрированы или хотят создать новый аккаунт;

- Переход в избранное: кнопка для быстрого перехода к списку избранных фильмов и сериалов;

- Label с именем пользователя: отображает имя пользователя, если он вошел в систему, создавая более персонализированный опыт.

Страница описания фильма/сериала

На странице описания пользователи могут найти:

- Изображение: крупное изображение, позволяющее детально рассмотреть контент;

- Описание: текстовая информация о характеристиках, жанре и особых моментах фильма или сериала;

- Рейтинг: чётко обозначенный рейтинг контента;

- Дата выхода и другие детали: информация, которая помогает пользователю решить о просмотре;

- Кнопка добавления в избранное: аналогичная кнопка, как на главной странице, для быстрой добавки.

Странцы входа и регистрации

Эти страницы предназначены для входа пользователей в систему и создания нового аккаунта. Включают в себя:

- Поля для ввода логина и пароля;

- Поле для ввода электронной почты: дополнительный метод идентификации пользователя;

- Кнопки для отправки учетных данных и входа в систему.

Страница избранного

На странице избранного пользователи могут видеть:

- Карточки избранных фильмов и сериалов: каждая карточка содержит информацию:

- Изображение: визуальное представление каждого добавленного контента;

- Ссылку на описание: возможность перейти к подробной информации;

- Кнопки удаления из избранного: позволяет пользователю убрать контент из избранного, если он передумал;

- Кнопка просмотра: для перехода к просмотру выбранного контента.

Серверная часть сайта состоит из базы данных, а также скриптов на PHP, обеспечивающих взаимодействие между клиентскими запросами и базой данных. Физическая диаграмма базы данных представлена в приложении Б.

**4. Выбор и обоснование алгоритмов**

При разработке онлайн-магазина смартфонов важным этапом является выбор алгоритмов, которые обеспечивают выполнение ключевых функций: управление данными о врачах, услугах, новостях, а также безопасное взаимодействие с пользователем. Выбранные алгоритмы позволяют достичь высокой производительности, надежности и простоты реализации.

Алгоритмы взаимодействия с базой данных

1) CRUD-операции (Create, Read, Update, Delete)

CRUD-операции являются основой взаимодействия с базой данных. Они реализуются с использованием SQL-запросов через API-эндпоинты.

- Добавление записей (Create): алгоритм принимает данные от пользователя, проверяет их и создает соответствующую запись в базе данных;

- Чтение записей (Read): используется для выборки данных с возможностью фильтрации и сортировки;

- Обновление записей (Update): предназначено для внесения изменений в существующие данные, например, обновления описания услуги;

- Удаление записей (Delete): удаляет устаревшие или ненужные данные из базы.

Алгоритмы валидации данных

Валидация данных - это процесс проверки данных на соответствие определённым критериям или правилам перед их использованием, или сохранением в базе данных. Этот процесс важен для обеспечения целостности и корректности данных, предотвращения ошибок и обеспечения безопасности приложений. Рассмотрим основные этапы валидации данных.

1) Сбор данных: на первом этапе данные собираются из различных источников, таких как формы ввода, API, базы данных и т.д.

2) Определение правил валидации: на этом этапе необходимо определить, какие правила будут применяться к данным. Правила могут включать:

- Формат (например, для email, телефона);

- Доменные ограничения (например, возраст должен быть в пределах 0-120);

- Обязательные поля (некоторые поля должны быть заполнены);

- Уникальность (например, логин или email должны быть уникальными).

3) Проверка данных: на этом этапе данные проверяются на соответствие

установленным правилам. Это может включать:

- Синтаксическую проверку (например, правильный формат email);

- Логическую проверку (например, дата начала не может быть позже даты окончания);

- Проверку на наличие значений (обязательные поля не должны быть пустыми).

4) Обработка ошибок: Если данные не прошли валидацию, необходимо предоставить пользователю информацию о том, какие ошибки были обнаружены и как их исправить. Это может быть сделано с помощью сообщений об ошибках или визуальных индикаторов.

5) Сохранение данных: Если данные прошли все проверки, они могут быть безопасно сохранены в базе данных или использованы в приложении.

6) Мониторинг и аудит: Важно периодически проверять качество и целостность данных, чтобы выявить возможные проблемы и улучшить процесс валидации.

**5 Описание программы**

Программа представляет собой динамический веб-сайт онлайн-киносервиса, разработанный с использованием современных технологий. Основная цель сайта - обеспечить пользователям удобный доступ к обширной библиотеке фильмов и сериалов, их описаниям, жанрам и рейтингу.

Сайт состоит из нескольких разделов, каждый из которых предназначен для выполнения определенной функции:

- Отображение списка доступных фильмов и сериалов с краткой информацией о каждом;

- Детальная информация о каждом фильме, включая описание, актерский состав, жанр, рейтинг и ссылку на трейлер;

- Возможность администраторам добавлять, изменять или удалять записи о фильмах и сериалах.

Для хранения данных используется реляционная СУБД MySQL. База данных состоит из следующих таблиц:

- Фильмы: содержит информацию о фильмах (название, описание, жанр, рейтинг, год выхода, изображение);

Пользователи: содержит информацию о зарегистрированных пользователях (имя, электронная почта, пароль);

Избранное: содержит информацию о избранных фильмах и сериалах каждого пользователя (идентификатор пользователя, список идентификаторов фильмов);

Для работы с базой данных используются CRUD-операции, реализованные с помощью API-эндпоинтов. Это обеспечивает:

- Добавление новых записей о фильмах и пользователях;

- Изменение существующих записей для обновления информации;

- Удаление устаревших данных.

Основные технологии программы:

Frontend (пользовательский интерфейс):

- React для создания динамичного пользовательского интерфейса;

- CSS для создания адаптивного и современного дизайна;

- Библиотеки компонентов для удобного управления элементами интерфейса.

Backend (серверная часть):

- Реализован с использованием PHP для обработки запросов к базе данных;

- Использование ORM для упрощения работы с MySQL.

Авторизация:

- Интеграция системы аутентификации для безопасной авторизации пользователей и администраторов.

Оптимизация:

- Использование инструментов для оптимизации загрузки страниц и повышения производительности сайта.

Проект приложения с исходным кодом можно посмотреть в репозитории по ссылке, указанной в приложении А.

**Заключение**

В рамках данной курсовой работы была поставлена задача разработки сайта онлайн-киносервиса, предоставляющего пользователям доступ к обширной библиотеке фильмов и телепередач, а также информации о жанрах, рецензиях и рекомендациях. Для достижения этой цели были рассмотрены принципы разработки веб-приложений, выбраны современные технологии и реализована система, соответствующая требованиям удобства, производительности и безопасности.

Результатом работы стал современный веб-сайт онлайн-киносервиса с простым и интуитивно понятным интерфейсом, адаптированным для различных устройств. Сайт обеспечивает пользователям возможность легко находить интересующие фильмы, фильтровать предложения по жанрам, годам выпуска и рейтингу, а также получать актуальную информацию о новинках, акциях и рекомендованных видео.

В ходе работы были достигнуты поставленные цели, продемонстрирована возможность использования выбранных инструментов для решения практических задач, связанных с разработкой веб-приложений. Созданный сервис обладает гибкостью, необходимой для дальнейшего расширения функционала, и полностью соответствует современным стандартам веб-разработки.

Разработанный проект может быть использован в реальных условиях, а полученные навыки и подходы могут быть применены в других аналогичных проектах. Таким образом, работа над сайтом онлайн-киносервиса не только удовлетворяет текущие потребности пользователей в качественном контенте, но и закладывает основу для будущих улучшений и адаптаций под изменяющиеся условия рынка онлайн-развлечений.

**Список литературы**

1. Чак, М. HTML и XHTML. Подробное руководство [Текст]: 6-е издание / М. Чак, Б. Кеннеди. – Москва: Символ-Плюс, 2008. – 748 с.
2. Мейер, Э.А. CSS. Каскадные таблицы стилей. Подробное руководство [Текст]: 3-е издание / Э.А. Мейер. – Москва: Символ-Плюс, 2008. – 575 с.
3. Дакетт, Д. Основы веб-программирования с использованием HTML, XHTML и CSS [Текст]: 2-е издание / Д. Дакетт. – Москва: Эксмо, 2010. – 768 с.
4. Флэнаган, Д. JavaScript. Подробное руководство [Текст] / Д. Флэнаган. – Москва: Символ-Плюс, 2008. – 1080 с.
5. Веллинг, Л. Разработка веб-приложений с помощью PHP и MySQL [Текст] / Л. Веллинг, Л. Томсон. – Москва: Вильямс, 2010. – 880 с

**Приложения**

**Приложение А**

Ссылка на репозиторий: <https://github.com/DiRasher04/Movie-service-web>

**Приложение Б**

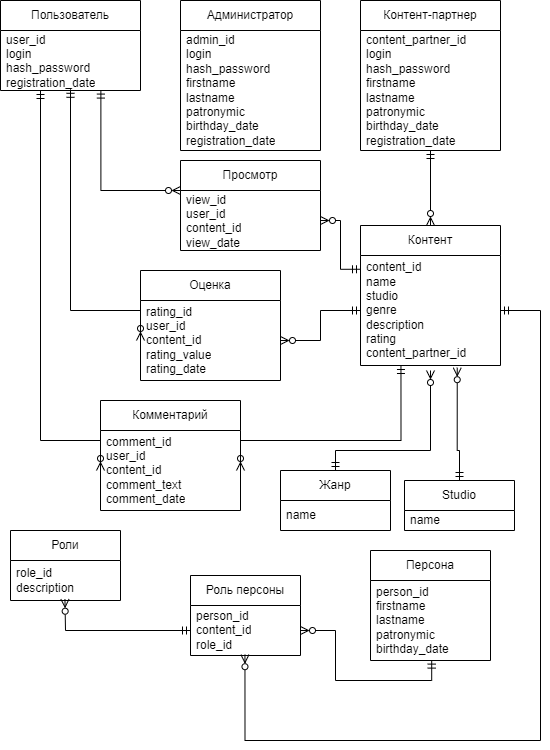
****

Рисунок 1 – Логическая модель данных

**Приложение В**

**Демонстрация приложения:**

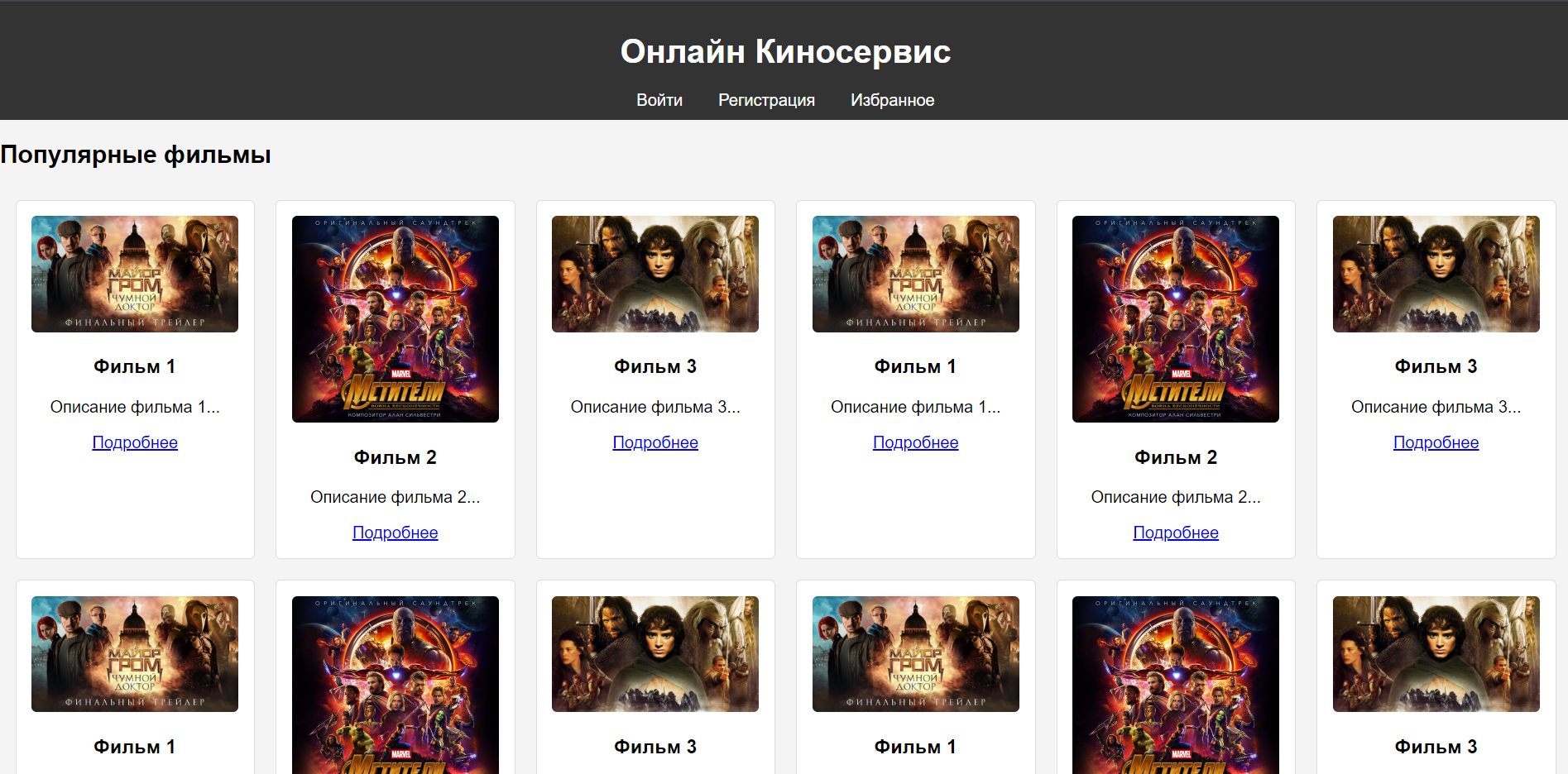
****

Рисунок В.1 – Главная страница

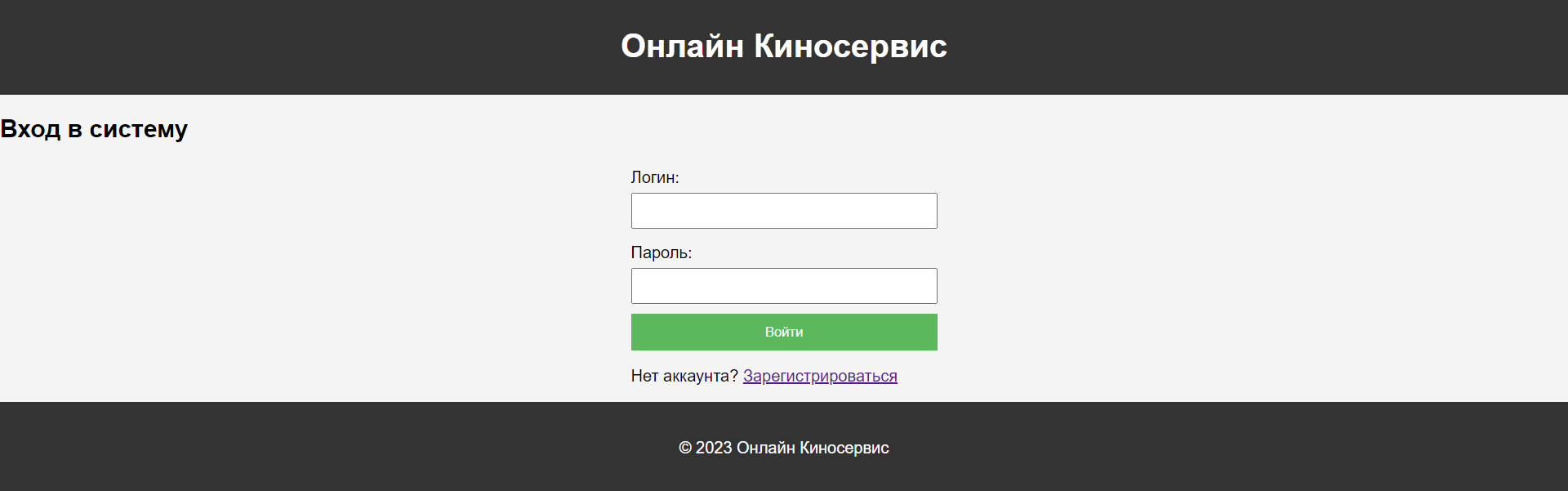
****

Рисунок В.2 – Страница авторизации

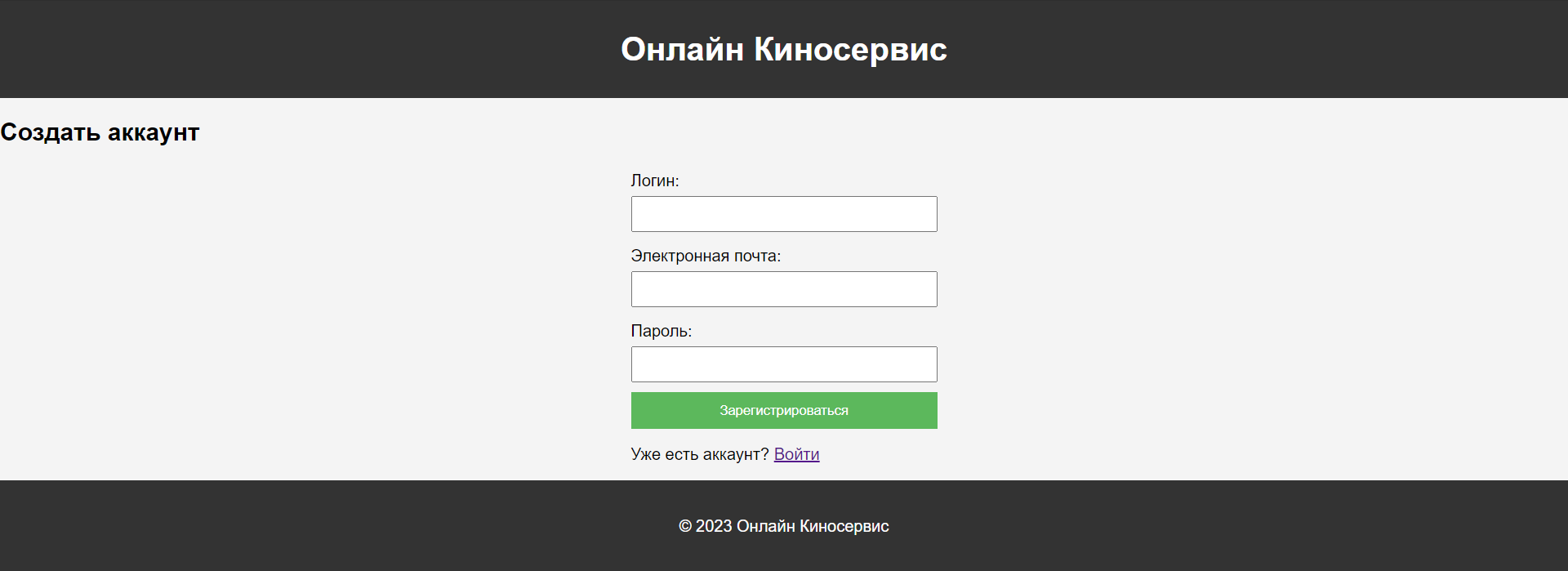
****

Рисунок В.3 – Страница регистрации

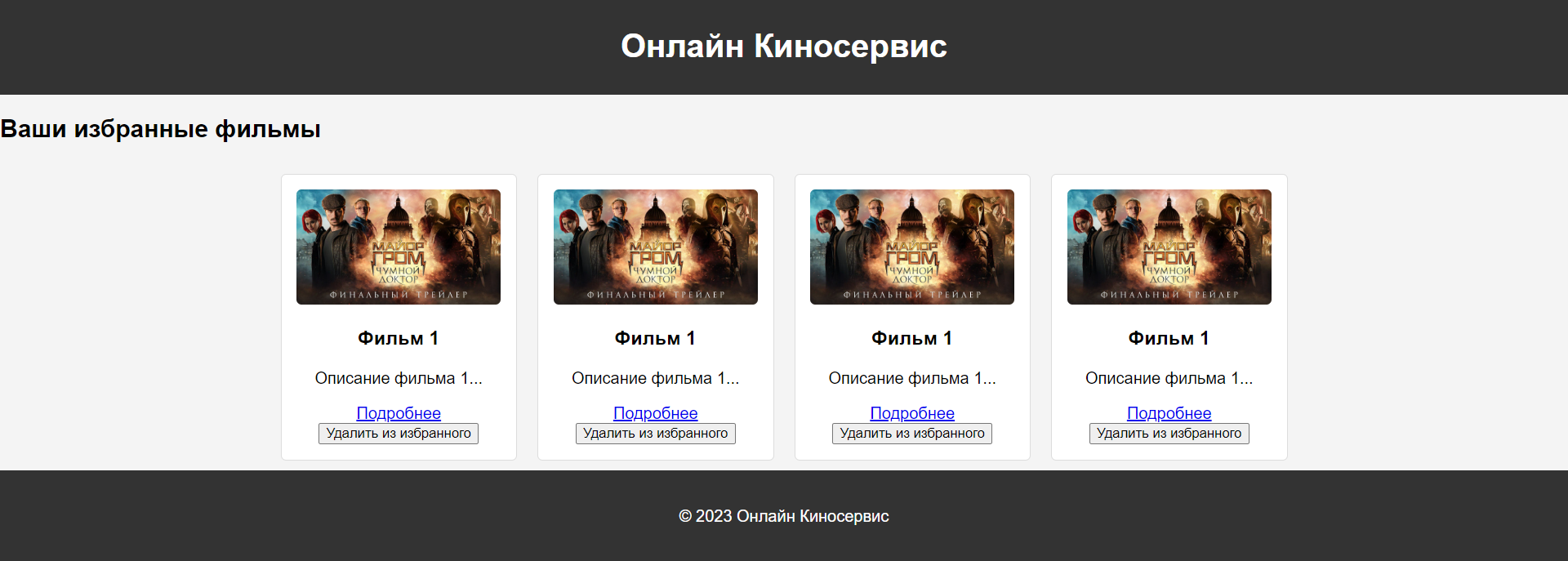


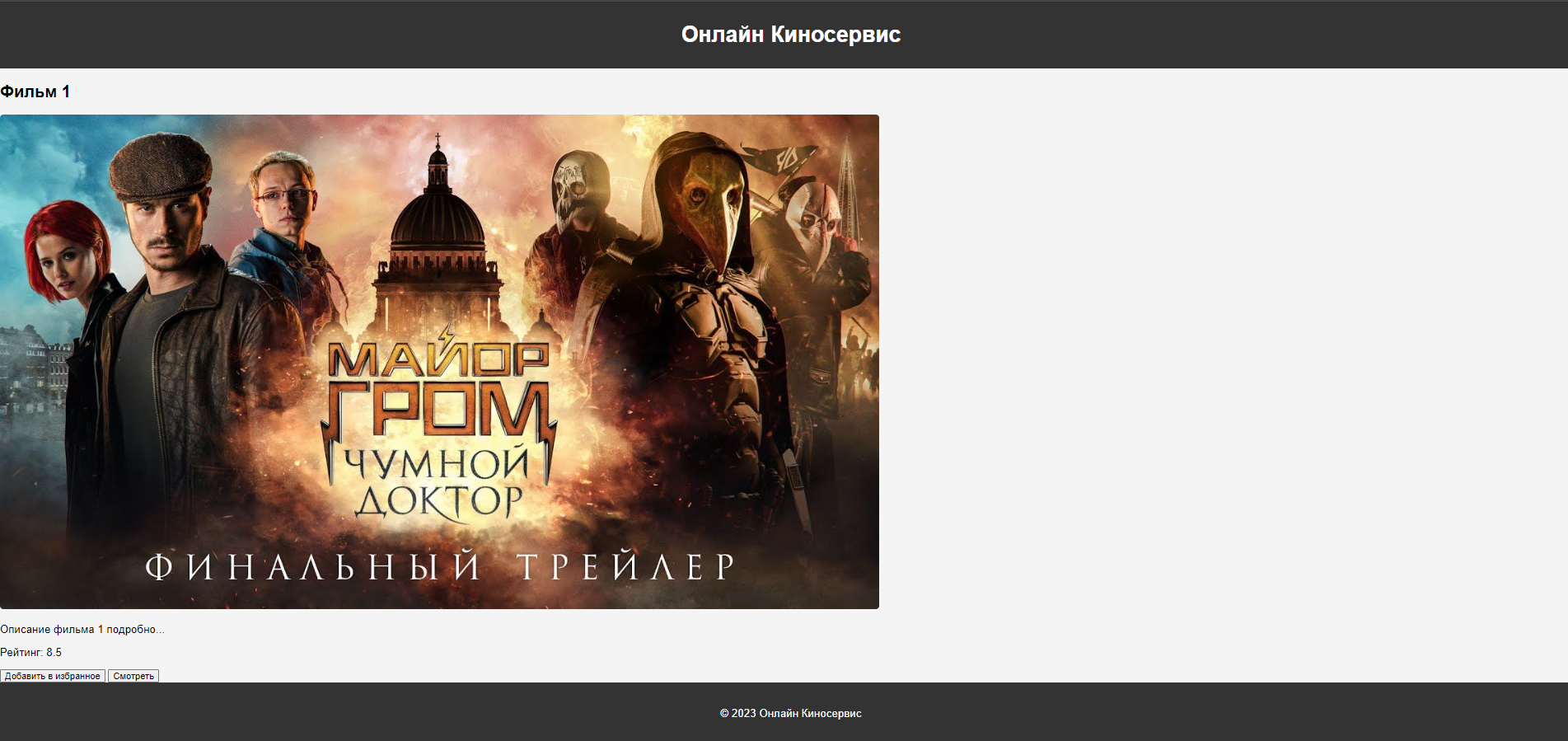
Рисунок В.4 – Страница избранного

Рисунок В.5 – Страница детального просмотра фильма