

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**СОЗДАНИЕ ПОЛНОФУНКЦИОНАЛЬНОГО КИНОСАЙТА С
СИСТЕМОЙ ПЕРСОНАЛИЗИРОВАННЫХ РЕКОМЕНДАЦИЙ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные
технологии
факультета КНиИТ
Дербина Даниила Михайловича

Научный руководитель
доцент, к. ф.-м. н.

М. И. Сафрончик

Заведующий кафедрой
доцент, к. ф.-м. н.

С. В. Миронов

Саратов 2025

ВВЕДЕНИЕ

В наши дни кино занимает важное положение в культурной сфере общества. Каждый год на экраны выходит множество новых фильмов разных направлений, и зрителям нужны удобные инструменты для поиска, анализа и оценки кинопродукции. Обычные способы получения информации о фильмах через печатные издания или телепередачи уступают место цифровым ресурсам, которые предлагают персонализированный и интерактивный опыт.

Актуальность данной работы обуславливается растущим интересом к созданию качественных веб-ресурсов, посвящённых киноиндустрии, которые не только предоставляют детальную информацию о фильмах, но и поддерживают сообщество любителей кино, способствуют обмену мнениями и помогают пользователям обнаруживать контент, который соответствует их вкусам и предпочтениям.

Этот проект представляет собой значительный шаг вперёд от исходного статического веб-сайта, который был создан в рамках предыдущих курсовых работ на основе элементарных технологий HTML, CSS и JavaScript. Если ранняя версия была лишь простой информационной страницей с ограниченной функциональностью, то новая реализация превращает концепцию в полноценное веб-приложение с современным дизайном, который включает серверную часть на базе Node.js с использованием Express.js, клиентскую часть на React.js и систему управления данными на MongoDB.

Основной целью бакалаврской работы является создание современного веб-портала, посвящённого кино, с расширенными функциями, среди которых система рекомендаций пользователям на основе их предпочтений и интеграция с внешними API для получения актуальной информации о фильмах.

Поставленная цель определила следующие задачи.

1. Необходимо провести всесторонний анализ существующих решений в сфере кинематографических веб-ресурсов, выявить их сильные и слабые стороны, а также определить целевую нишу для разрабатываемого продукта.
2. Следует изучить современные подходы к архитектуре веб-приложений и выбрать оптимальный набор технологий для выполнения поставленных задач.
3. Спроектировать архитектуру системы, которая обеспечит масштабируе-

мость, безопасность и эффективность приложения.

4. Реализовать серверную часть, основываясь на актуальных практиках разработки, включая систему аутентификации, API для работы с данными и интеграцию с внешними ресурсами.
5. Создать удобный и адаптивный интерфейс пользователя, позволяющий комфортно взаимодействовать с системой на различных устройствах.
6. Интегрировать систему персонализированных рекомендаций, которая должна опираться на анализ предпочтений пользователей и их активностей на платформе.

Бакалаврская работа состоит из введения, двух разделов, заключения, списка использованных источников и 5 приложений. Общий объем работы — 61 страниц, из них 44 страниц — основное содержание, включая 8 рисунков. Приложение включает USB-накопитель с исходным кодом. Список использованных источников — 23 наименования.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Фундаментальные аспекты проектирования ориентированных на киноиндустрию веб-систем» посвящен анализу основных игроков на рынке веб-ресурсов в сфере кино, их стратегий, а также исследованию особенностей создания и позиционирования продукта. Также рассматриваются ключевые технологические решения и библиотеки, используемые в современной веб-разработке для создания эффективных и масштабируемых кино-платформ.

Для того чтобы понять требования к современным кино-платформам, необходимо изучить существующие решения и определить их сильные и слабые стороны. IMDb является крупнейшим источником данных о кино, предоставляющим информацию о более 8 миллионах фильмов, сериалах и более 10 миллионах работников киноиндустрии. Платформа ориентирована на мировой рынок, что предоставляет пользователям доступ к самым разнообразным данным, включая рецензии, рейтинги и биографии актеров.

КиноПоиск представляет собой российский аналог, который объединяет информационные и стриминговые аспекты. Платформа выделяется качественной локализацией контента, подробными рецензиями профессиональных критиков и активной пользовательской базой. В отличие от IMDb, КиноПоиск фокусируется на русскоязычных пользователях и локализует контент в соответствии с их предпочтениями.

Архитектурная концепция разрабатываемой системы основывается на современных принципах веб-разработки, таких как разделение обязанностей, масштабируемость и удобство поддержки кода. Система разрабатывается как одностраничное приложение (SPA), что позволяет обеспечить высокий уровень интерактивности и улучшенный пользовательский опыт. Клиентская часть приложения будет реализована с использованием React, что позволяет создать переиспользуемые компоненты и централизованное управление состоянием, обеспечивая предсказуемость работы приложения и упрощая его отладку.

Серверная часть будет представлять собой RESTful API, разработанный на платформе Node.js с использованием фреймворка Express.js. Это решение позволяет разделить различные уровни приложения — презентацию, бизнес-логику и доступ к данным, что улучшает гибкость и упрощает тестирование и поддержку системы. MongoDB, как документо-ориентированная NoSQL база данных, была выбрана благодаря своей гибкости в хранении разнородных дан-

ных о фильмах и высокой производительности при манипуляциях с большими объемами информации.

Дизайн интерфейса будет следовать современным принципам веб-дизайна с акцентом на читаемость и визуальную иерархию. В интерфейсе будут использоваться темные оттенки для создания атмосферы кинотеатра, а яркие акценты привлекут внимание к важным элементам. Современные технологии и фреймворки, такие как React, Vue.js и Angular, позволяют создавать сложные и высокоэффективные интерактивные приложения, что дает возможность предложить пользователю непревзойденный опыт взаимодействия с контентом.

Выбранный технологический стек (JavaScript, Node.js, React.js, MongoDB) позволяет обеспечить единообразие в коде и эффективность разработки, так как один и тот же язык используется как на клиентской, так и на серверной стороне. Node.js предоставляет отличную производительность благодаря неблокирующей модели ввода-вывода, что важно для приложений с высокой нагрузкой, а также позволяет эффективно обрабатывать большое количество одновременных соединений.

Express.js, выбранный в качестве фреймворка для Node.js, упрощает создание и поддержку серверной части приложения, предлагая удобные средства для работы с HTTP-запросами, маршрутизацией и интеграцией с внешними API. Для взаимодействия с базой данных MongoDB используется Mongoose, что облегчает процесс работы с документами и моделями данных.

Для аутентификации и безопасности используется технология JWT (JSON Web Tokens), которая обеспечивает безопасную передачу данных между клиентом и сервером. Важным компонентом безопасности является Bcrypt — библиотека для хеширования паролей, которая помогает защитить данные пользователей от несанкционированного доступа.

Интеграция с API КиноПоиска позволяет предоставить пользователям актуальную информацию о фильмах без необходимости поддержания собственной базы данных. Это решение значительно упрощает разработку и уменьшает затраты на хранение данных.

Выбранный технологический стек Node.js, Express.js, React.js и MongoDB обеспечивает оптимальный баланс между продуктивностью разработчика, производительностью приложения и удобством в обслуживании. Современные инструменты разработки и архитектурные паттерны создают основу для создания

масштабируемых и надежных веб-приложений, способных адаптироваться под меняющиеся требования и растущую пользовательскую базу.

Второй раздел «Разработка портала о кино с рекомендательной системой» посвящен реализации серверной составляющей приложения, разработке пользовательского интерфейса, системе поиска и фильтрации, адаптивному дизайну для мобильных устройств и внедрению рекомендательной системы.

Разработка архитектуры приложения базировалась на трехуровневой модели, которая разделяет функциональные обязанности между клиентом, бизнес-логикой и уровнем хранения данных. Для интерфейса был выбран React, что позволило использовать компонентный подход для создания модульных и повторно используемых компонентов, которые обеспечивают взаимодействие с пользователем, отображение информации и связь с сервером через HTTP-запросы. Этот подход позволяет улучшить структуру кода, упростить поддержку и масштабируемость приложения. Компоненты, такие как карточки фильмов и фильтры, могут быть независимо обновлены или заменены без затрагивания остальной логики приложения.

Серверная часть приложения реализована с использованием Express.js, который выполняет роль связующего звена между клиентом и базой данных. В этом слое реализованы важнейшие функции, такие как аутентификация пользователей, валидация данных, взаимодействие с внешними API (например, для получения информации о фильмах) и алгоритмы рекомендательной системы. Express.js также предоставляет необходимую гибкость для организации и обработки маршрутов, контроллеров и middleware. Модульная структура приложения и возможность легко добавлять новые маршруты или изменять существующие маршруты обеспечивают гибкость при дальнейшем развитии приложения.

Для хранения данных была выбрана NoSQL база данных MongoDB, которая позволяет эффективно работать с гибкими структурами данных, такими как фильмы, пользователи и их предпочтения. В MongoDB используются схемы, определенные с помощью Mongoose, что упрощает работу с данными и их валидацию на уровне базы данных. Модели данных, такие как модель фильма или пользователя, содержат различные поля, включая текстовые строки, числа, массивы и объекты. Для улучшения производительности и снижения нагрузки на внешний API была внедрена система промежуточного кэширования. Это кэширование позволяет локально хранить информацию о фильмах с временными

метками, что облегчает актуализацию данных и уменьшает количество запросов к API при повторных запросах.

Система аутентификации была реализована с использованием JWT (JSON Web Tokens), что позволяет использовать stateless-аутентификацию. Это гарантирует, что данные о пользователях не будут храниться на сервере, а все необходимое будет содержаться в токенах, которые передаются между клиентом и сервером. JWT-токены включают в себя информацию о пользователе, такую как его ID и права доступа, и имеют ограниченный срок действия, что повышает безопасность. В случае истечения срока действия токена используется refresh-токен для получения нового access-токена без необходимости повторной аутентификации. Аутентификация через почту была дополнена функционалом отправки кода подтверждения на электронную почту пользователя, что добавляет еще один уровень безопасности, гарантируя, что процесс регистрации или восстановления пароля выполняется только пользователем, имеющим доступ к его почтовому ящику.

Система рекомендаций построена на базе алгоритма косинусного сходства, который вычисляет степень схожести между фильмами и профилем пользователя. Фильмы представлены в виде векторов с числовыми значениями, которые включают метаданные, такие как жанры, год выпуска и рейтинг. Профиль пользователя также представлен как вектор, составленный из предпочтений, основанных на фильмах, которые он оценил высоко. С помощью косинусного сходства система определяет наиболее релевантные фильмы, которые могут заинтересовать пользователя, и рекомендует их.

Система поиска и фильтрации фильмов была реализована для быстрого нахождения контента. Для этого был внедрен механизм автодополнения, который начинает работать после ввода хотя бы трех символов. Алгоритм поиска сначала ищет точные или частичные совпадения по названию фильма, а затем расширяет поиск на описание фильма. Результаты поиска отображаются с возможностью дальнейшей фильтрации по жанрам, рейтингу и году выпуска. Система поддерживает фильтрацию как на клиентской, так и на серверной стороне, где сервер обрабатывает запросы, делает фильтрацию данных и кэширует результаты для повышения производительности.

Важным аспектом является интеграция видеоплеера KinoBox, который работает по ID фильма, полученному из API КиноПоиска. Это позволяет воспро-

изводить видео прямо на сайте, интегрируя плеер с данными о фильме, такими как описание, постер и трейлеры. Плеер использует HTML5 Video API, что обеспечивает кроссплатформенную совместимость и высокое качество воспроизведения. Это решение позволяет пользователю не покидать сайт для просмотра контента, что повышает удобство использования. Встроенный плеер поддерживает управление воспроизведением, паузу, регулировку громкости, а также переключение в полноэкранный режим и перемотку на 10 секунд вперед и назад.

Кроме того, для поиска была реализована система дебаунсинга, которая позволяет минимизировать количество запросов к серверу, отправляя запросы только после небольшой паузы в 300 миллисекунд. Это снижает нагрузку на сервер и улучшает производительность приложения. Поиск фильмов также поддерживает пагинацию, что позволяет загружать результаты по частям (например, по 20 фильмов на страницу), улучшая пользовательский опыт и обеспечивая более быструю загрузку страниц.

Страница профиля пользователя была спроектирована с целью предоставить персонализированный опыт взаимодействия с порталом. Пользователи могут управлять своими настройками, просматривать историю просмотров, а также получать персонализированные рекомендации на основе их предпочтений. Для удобства навигации интерфейс профиля был разделен на несколько вкладок: общие настройки, безопасность, историю просмотров и персонализированные советы. Все изменения, такие как смена пароля или изменение темы интерфейса, проходят через проверку и защиту данных, чтобы обеспечить высокий уровень безопасности.

Для мобильных устройств был создан адаптивный дизайн, использующий подход «mobile-first». Это означает, что интерфейс сначала проектируется для малых экранов, а затем подстраивается для больших экранов с помощью CSS Media Queries. Адаптивный дизайн обеспечил быструю загрузку страниц и улучшение взаимодействия с пользователем. Все элементы, такие как фильтры, меню и карточки фильмов, были оптимизированы для мобильных устройств, что улучшает пользовательский опыт, делая приложение доступным и удобным для использования на разных устройствах.

Рекомендательная система работает с помощью векторных представлений фильмов и пользователей. Каждый фильм представлен в виде числового вектора, который включает в себя его основные характеристики, а профиль

пользователя создается на основе его предпочтений. Процесс преобразования фильмов и сериалов в векторную форму включает трансформацию качественных и количественных атрибутов контента в числовые векторы фиксированной длины. Для этого применяется сочетание различных методов кодирования в зависимости от характера данных.

Жанры фильмов кодируются с применением метода one-hot encoding, что приводит к созданию бинарного вектора, где каждому жанру выделено отдельное место. Если фильм соответствует конкретному жанру, соответствующий элемент вектора принимает значение 1, иначе — 0. Такой подход позволяет эффективно обрабатывать фильмы, относящиеся к нескольким жанрам одновременно, и обеспечивает равное влияние каждого жанра на итоговый вектор.

Год выпуска нормализуется для согласования со шкалой других атрибутов. Используется стандартная нормализация (z-score normalization), которая центрирует данные относительно среднего значения и масштабирует по стандартному отклонению. Это помогает избежать преобладания признаков с высокими числовыми значениями над альтернативными компонентами вектора.

Тип контента (фильм или сериал) обозначается бинарным значением, где фильмы получают 0, а сериалы — 1. Этот метод позволяет системе учитывать предпочтения пользователя относительно формата контента при создании рекомендаций.

Пользовательский профиль формируется путём агрегации векторов всех фильмов, которые понравились пользователю, с последующей L2-нормализацией. Это обеспечивает адекватное представление предпочтений пользователя. Взаимосвязь между пользователем и фильмами определяется с помощью косинусного сходства, которое оценивает косинус угла между векторами в многомерном пространстве:

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

, где A и B — это сравниваемые векторы, $\cdot$ означает скалярное произведение, а $\|\cdot\|$ — это евклидова норма вектора. Поскольку все векторы были предварительно нормализованы, вычисление упрощается до простого скалярного произведения.

Система вычисляет степень схожести между профилем пользователя и

всеми имеющимися фильмами, исключая уже просмотренные материалы. Финальный рейтинг формируется путём объединения косинусного сходства и нормализованного рейтинга: $\text{итоговыйРейтинг} = \alpha \times \text{косинусноеСходство} + \beta \times \text{рейтингФильма}$. Для обеспечения разнообразия в рекомендации реализовано чередование фильмов и сериалов в топ-30.

Таким образом, проект обеспечил комплексную реализацию всех необходимых функций для успешного функционирования кино-портала с рекомендательной системой, что включает в себя эффективное хранение данных, удобный интерфейс для пользователей и интеллектуальные алгоритмы для улучшения пользовательского опыта.

ЗАКЛЮЧЕНИЕ

В рамках написания выпускной квалификационной работы была успешно запущена современная онлайн-платформа для библиотеки фильмов с интеграцией системы персонализированных рекомендаций. Этот проект представляет собой полноценное веб-приложение, которое сочетает передовые технологии веб-разработки с современными алгоритмами машинного обучения для обеспечения высокого качества пользовательского опыта.

В ходе разработки все задачи были успешно выполнены. Была создана масштабируемая трехуровневая архитектура, которая включает клиентскую часть на React, сервер на Express.js для реализации бизнес-логики и MongoDB для хранения данных. Реализована надежная схема аутентификации и авторизации пользователей с использованием JWT-токенов, что гарантирует безопасность и архитектуру без состояния.

Особое внимание уделено разработке интуитивно понятного интерфейса, адаптивного под разные устройства. Интеграция с внешним API Кинопоиска предоставляет актуальные данные о фильмах, а система кэширования помогает оптимизировать производительность приложения.

Ключевым достижением стало создание рекомендательной системы, основанной на косинусном сходстве, которая оценивает предпочтения пользователей и предлагает индивидуализированный контент. Эта система демонстрирует высокую точность рекомендаций за счет комплексного анализа характеристик фильмов и поведения пользователей.

Созданное приложение обладает высокой практической ценностью и может использоваться как готовое решение для кино-порталов или адаптироваться к другим областям, в которых требуется каталогизация и рекомендации контента. Модульная архитектура и современный стек технологий обеспечивают легкость в масштабировании и развитии функциональных возможностей.

Проект иллюстрирует эффективное применение современных веб-технологий и алгоритмов машинного обучения в реальных приложениях, что подтверждает актуальность выбранного подхода к разработке индивидуализированных веб-сервисов.

Разработанная платформа имеет большой потенциал для дальнейшего роста. Возможные направления для улучшения включают внедрение более продвинутых алгоритмов машинного обучения для повышения точности рекоменда-

ций, добавление социальных функций для формирования пользовательского сообщества и расширение интеграции с дополнительными источниками данных о фильмах.

Таким образом, задача создания современного веб-портала с расширенными функциями была полностью осуществлена, и разработанное решение может стать основой для коммерческих кино-порталов или образовательных проектов в сфере веб-разработки.