

потребности пользователей из-за ограниченного функционала или неудобного интерфейса. В данном контексте особенно важна роль разработанного веб-приложения "Финансовый Помощник", которое объединяет все важные и удобные функции на одной платформе.

Приложение содержит две части: серверную, которая разработана с использованием платформы Node.js, фреймворка Express и MongoDB Atlas, и клиентскую, в которой использовался язык JavaScript с фреймворком React.js.

В ходе разработки получилось готовое приложение, в котором реализован удобный функционал для управления и анализа своих денежных потоков. Приложение позволяет создавать счета, добавлять расходы, доходы, ограничения на расходы, предоставляет возможность создавать финансовые цели, позволяет приглашать других пользователей для совместного управления счетами, получать уведомления при превышении ограничений по расходам, получать предсказание расходов на следующий месяц, просматривать и экспортировать статистику и отчёты о финансовом состоянии, предоставляет возможность расчёта инвестиций с помощью финансового калькулятора. Отчеты визуализированы в виде разнообразных диаграмм. Интерфейс приложения интуитивно понятный и адаптирован к различным устройствам.

ЛИТЕРАТУРА

1. Документация по ReactJS [Электронный ресурс]. – Режим доступа: <https://ru.reactjs.org/>. – Дата доступа: 15.04.2024.
2. О Node.js [Электронный ресурс]. / nodejs.org. – Режим доступа: <https://nodejs.org/ru/about>. – Дата доступа: 16.04.2024.
3. Express.js [Электронный ресурс]. / expressjs.com. – Режим доступа: <https://expressjs.com/ru>. – Дата доступа: 17.04.2024

УДК 004.43

Студ. Д.А. Зворыкин
Науч. рук. ст. преп. И.Г. Сухорукова
(кафедра программной инженерии, БГТУ)

ВЕБ-ПРИЛОЖЕНИЕ «ОНЛАЙН-КИНОТЕАТР»

Онлайн кинотеатры, стали привычной частью нашей жизни. Они позволяют пользователям пользоваться обширной библиотекой фильмов в удобное время и месте, расширяя аудиторию потенциальных зрителей за пределами географических ограничений традиционных кинотеатров.

Для разработки приложения с использованием ASP.NET Core и React, а также с микросервисной архитектурой, было выбрано несколько ключевых компонентов и паттернов. В приложении представлено четыре микросервиса: микросервис авторизации, рекомендации фильмов, основной микросервис, выполняющий бизнес-логику, и микросервис, реализующий паттерн Gateway с использованием пакета Ocelot.

Наибольший интерес с точки зрения реализации представляют система рекомендаций фильмов и система VOD (видео по запросу). Для реализации системы VOD - потокового вещания видео от сервера к пользователю без необходимости полностью загружать видео перед просмотром, а также адаптивным качеством видео с возможностью выбрать субтитры и аудиодорожку, был выбран протокол HLS и DASH. HLS отправляет видео- и аудиоконтент по сети в небольшие сегменты потокового мультимедиа на основе TCP. Поскольку HLS использует TCP, то он работает по принципу «качество важнее задержки», поэтому время задержки может быть высоким (например, в секундах, а не в миллисекундах), но для просмотра фильма это не важно [1].

Для реализации системы рекомендации было использовано машинное обучение. Система рекомендации фильмов работает на основе алгоритма матричной факторизации, который использует метод совместной фильтрации [2]. Этот метод предполагает, что если два пользователя оценили фильмы одинаково, то они, вероятно, будут иметь схожие мнения о других фильмах. Алгоритм разложения матрицы разбивает матрицу оценок на две матрицы, каждая из которых представляет собой набор признаков для пользователей и фильмов соответственно. Эти матрицы затем используются для предсказания оценок, которые пользователь мог бы дать фильму, основываясь на его прошлых оценках и оценках других пользователей.

ЛИТЕРАТУРА

1. SRT, HLS и MPEG-DASH – будущее потокового вещания [Электронный ресурс] – Режим доступа: https://www.epiphan.ru/articles/art_91.php (дата обращения 15.04.2024).

2. Документация по ML.NET [Электронный ресурс] – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/machine-learning/> (дата обращения 15.04.2024).