

Е. А. Плехоткин^{1✉}, Е. Ю. Воронкин¹

Разработка веб-ресурса «Дневник хроник моих интересов» на языке программирования Python

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: loldalender@gmail.com

Аннотация. В статье рассматривается процесс разработки веб-ресурса для публикации отзывов на различные категории и получения персонализированных рекомендаций на основе профиля интересов. Актуальность работы обусловлена необходимостью создания единой цифровой платформы для предоставления пользователям мультикатегорийных рекомендаций на основе его отзывов. В качестве основного фреймворка был выбран Django framework с применением встроенной базы данных SQLite и интегрированием GigaChat API. При разработке был разработан функционал публикации отзывов, просмотра, редактирования, удаления оставленных отзывов и формирование списка рекомендаций. Было уделено особое внимание обеспечению безопасности, масштабируемости и предоставления пользователям удобного интерфейса. В статье продемонстрировано использование современных актуальных технологий.

Ключевые слова: веб-программирование, Django, Python, ORM, HTML, CSS, Javascript, API, разработка веб-ресурсов, база данных, SQLite

E. A. Plehotkin^{1✉}, E. Yu. Voronkin¹

Development of the web resource «Diary of chronicles of my interests» in the Python programming language

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: loldalender@gmail.com

Abstract. The article discusses the process of developing a web resource for publishing reviews on various categories and receiving personalized recommendations based on the interests profile. The relevance of the work is due to the need to create a single digital platform for providing users with multicategory recommendations based on their feedback. The Django framework was chosen as the main framework, using the built-in SQLite database and integrating the GigaChat API. During the development, the functionality of publishing reviews, viewing, editing, deleting left reviews and creating a list of recommendations was developed. Special attention was paid to ensuring security, scalability and providing users with a user-friendly interface. The article demonstrates the use of modern relevant technologies.

Keywords: web programming, Django, Python, ORM, HTML, CSS, Javascript, API, web resource development, Database, SQLite

Введение

В современном мире пользователи зачастую проводят свободное время, потребляя медиаконтент или посещая различные заведения или мероприятия для отдыха и хорошего времяпрепровождения. В связи с этим, пользователи заинтересованы в поиске новых объектов увлечения, которые придется им по душе.

Однако из-за переизбытка информации и перенасыщенности контента, пользователям становится все труднее найти то, что соответствует их вкусам и интересам.

Лучшим вариантом для поиска являются отзывы и рекомендации других пользователей. Но в данный момент они распределены на по разным платформам, которые зачастую ограничены одной категорией контента, в следствии чего пользователям приходится тратить много времени на поиск того, что могло бы их заинтересовать. Кроме того, многие ресурсы, при составлении рекомендаций, не учитывают интересы пользователей в смежных категориях, и, следовательно, не могут предоставить более точные рекомендации.

Целью работы является разработка веб-ресурса «Дневник хроник моих интересов» на языке программирования Python. Для достижения поставленной цели были выполнены следующие задачи:

- анализ предметной области;
- анализ существующих веб-ресурсов отзывов и выявление их недостатков;
- проектирование структуры базы данных и создание интуитивного интерфейса;
- реализация основного функционала с использованием фреймворка Django;
- интеграция GigaChat API для ИИ-анализа отзывов и составления рекомендаций;
- тестирование и исправление выявленных ошибок.

Методы и материалы

Первым делом, были разработаны ключевые диаграммы, чтобы определить структуру базы данных и работу системы рекомендаций.

Была разработана диаграмма прецедентов, для определения действий, которые пользователь будет совершать в системе [1] (рис. 1).

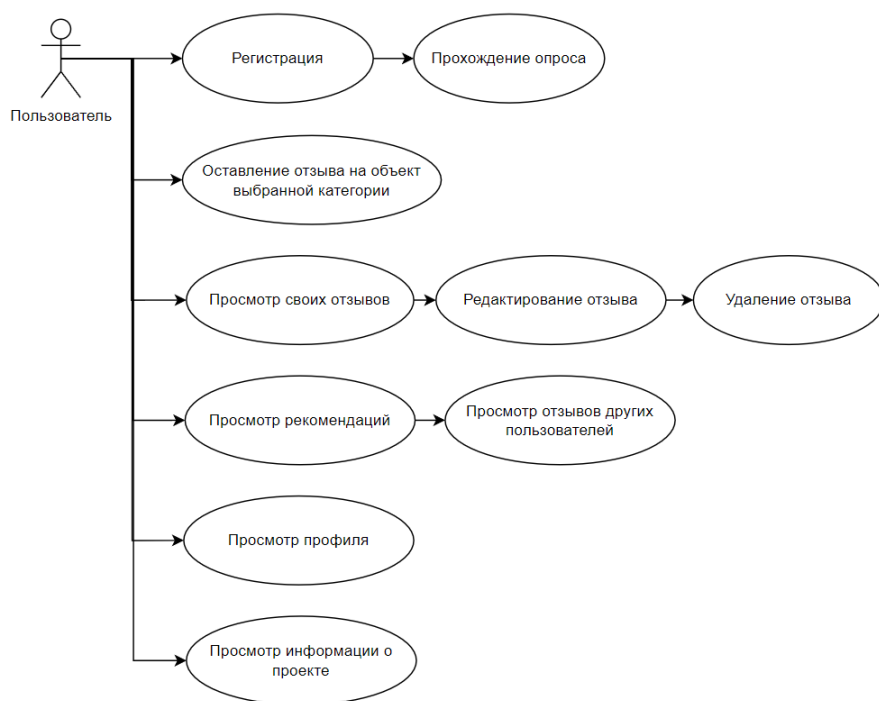


Рис. 1. Диаграмма прецедентов веб-ресурса

Следующей разработанной диаграммой стала диаграмма последовательности, которая демонстрирует процесс взаимодействия с веб-ресурсом от регистрации и оставления отзывов до получения пользователем персональных рекомендаций [2] (рис. 2).

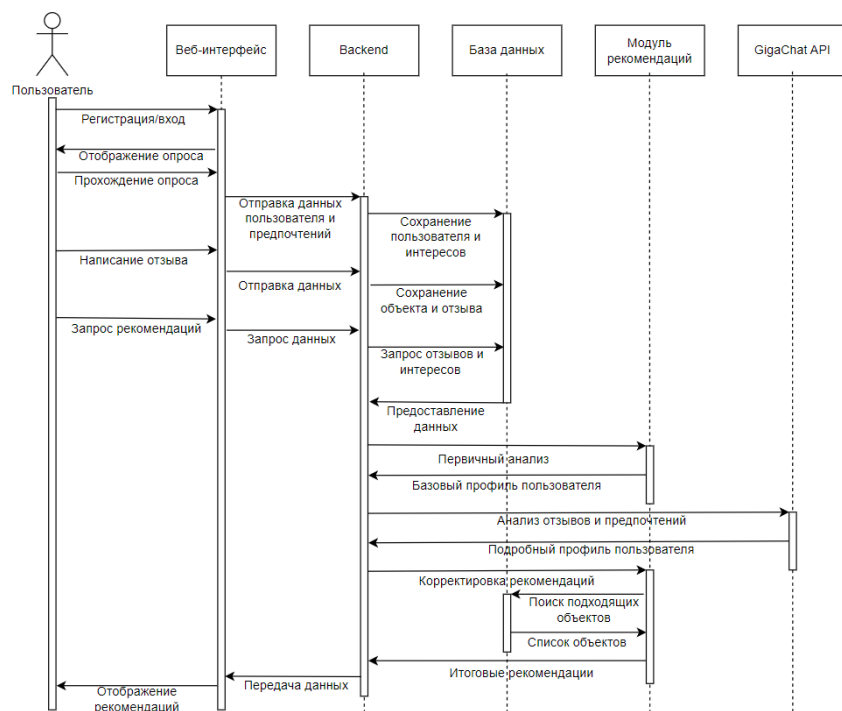


Рис. 2. Диаграмма последовательностей веб-ресурса

Для определения сущностей в базе данных, содержащейся в них информации и определения связей между сущностями была создана диаграмма связей сущностей [3] (рис. 3).

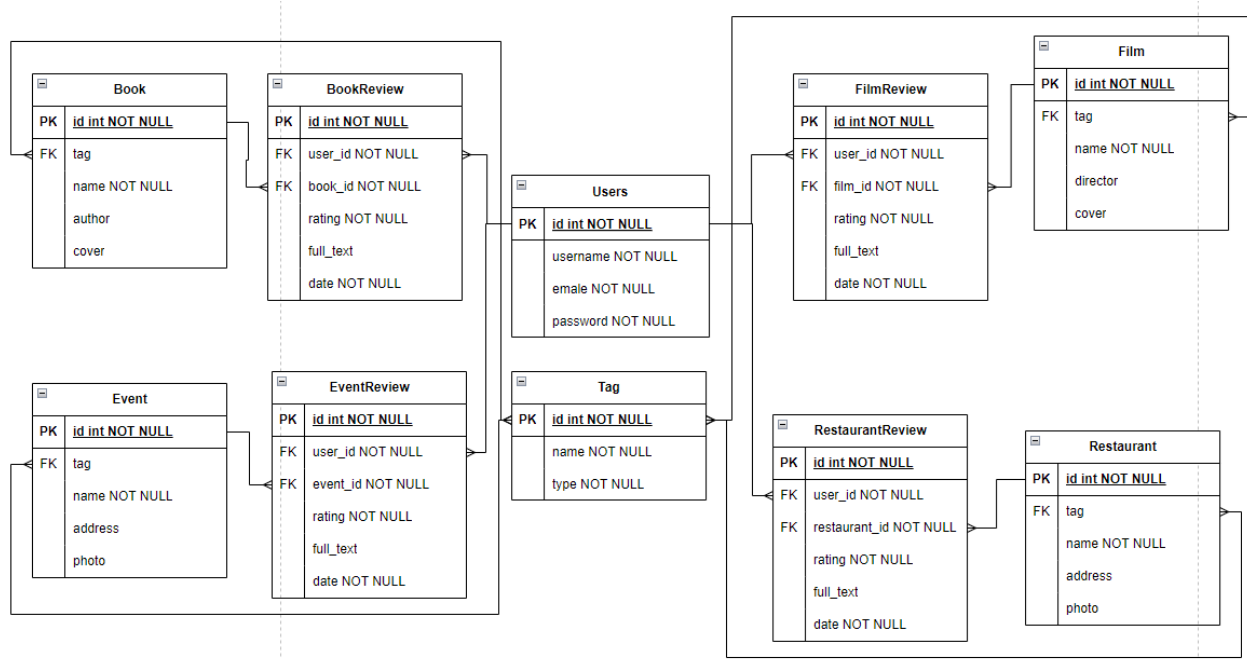


Рис. 3. Диаграмма отношения сущностей базы данных веб-ресурса

До начала разработки был создан макет интерфейса с помощью инструментов сервиса Figma (рис 4). Данный макет был использован как шаблон для верстки.

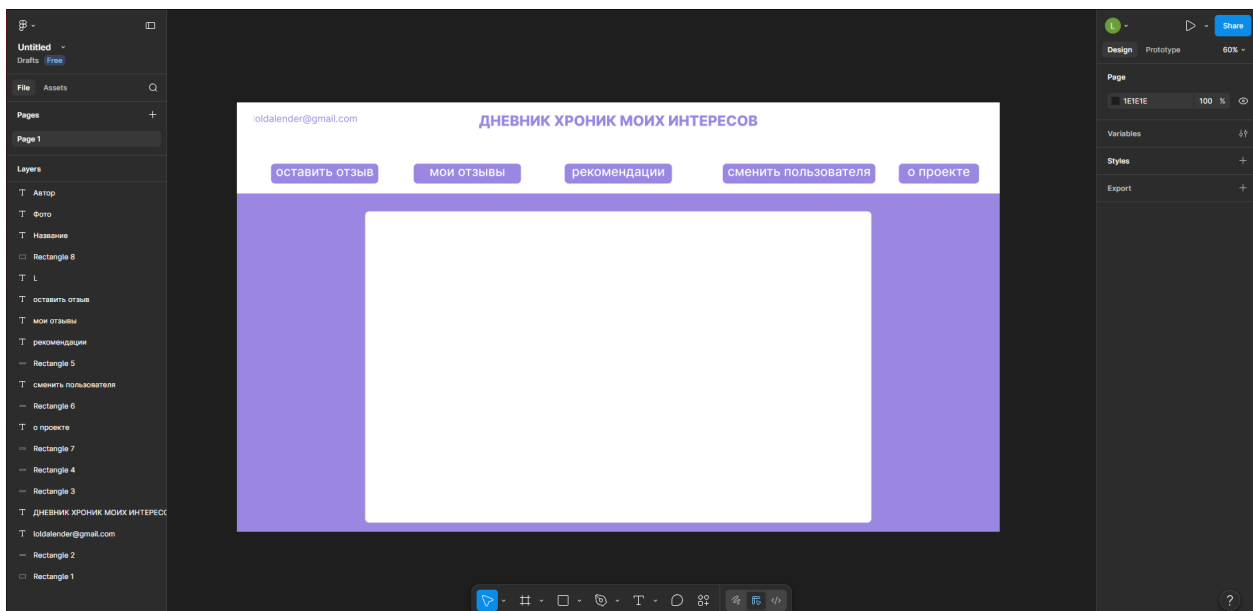


Рис. 4. Разработка дизайна шаблона веб-ресурса

Следующим шагом был выбран и обоснован стек технологий, к примеру фреймворк и СУБД.

Для разработки бэкэнд части был выбран фреймворк Django — распространенный веб-фреймворк с широким функционалом для разработки веб ресурсов. Для реализации интерфейса использовался метод MVT и HTML, CSS и JavaScript для фронтэнда.

Для хранения данных была выбрана встроенная СУБД SQLite, взаимодействие с которой осуществляется с помощью Django ORM [4].

Система рекомендаций реализована с интеграцией GigaChat API [5].

Результаты

Анализ, создание диаграмм для определения архитектуры и выбранный стек технологий (Django, SQLITE и GigaChat API) легли в основу реализации веб-ресурса.

Для взаимодействия с базой данных была написана функция home, которая является главным обработчиком, отвечающим за загрузку данных пользователя, получения списка отзывов, формирование рекомендаций и передачу информации в HTML-шаблон главной страницы.

Затем были созданы ORM модели сущностей базы данных и зарегистрированы в административной панели.

Среди моделей сущностей находятся модели объектов для каждой категории, модели отзывов на объекты для каждой категории и модель тегов, которая была вручную заполнена множеством тегов через административную панель (рис. 5).

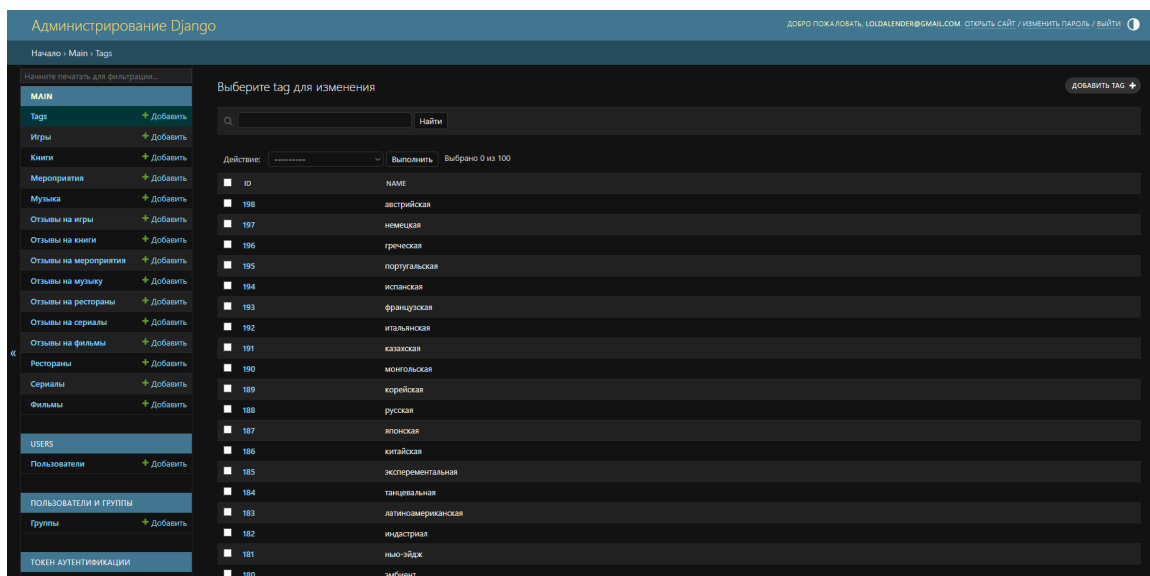


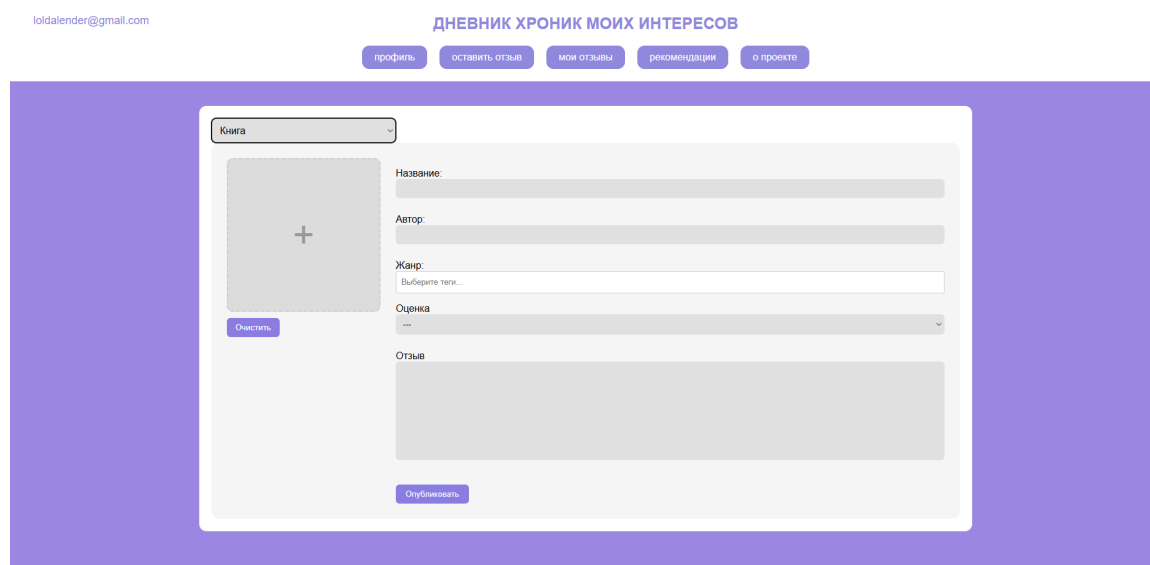
Рис. 5. Заполненная модель тегов в административной панели Django

Затем были разработаны формы ввода данных с использованием системы Django forms для централизованной проверки данных, генерации HTML-полей и

обработки загружаемых изображений. Формы были созданы для всех моделей объектов и отзывов.

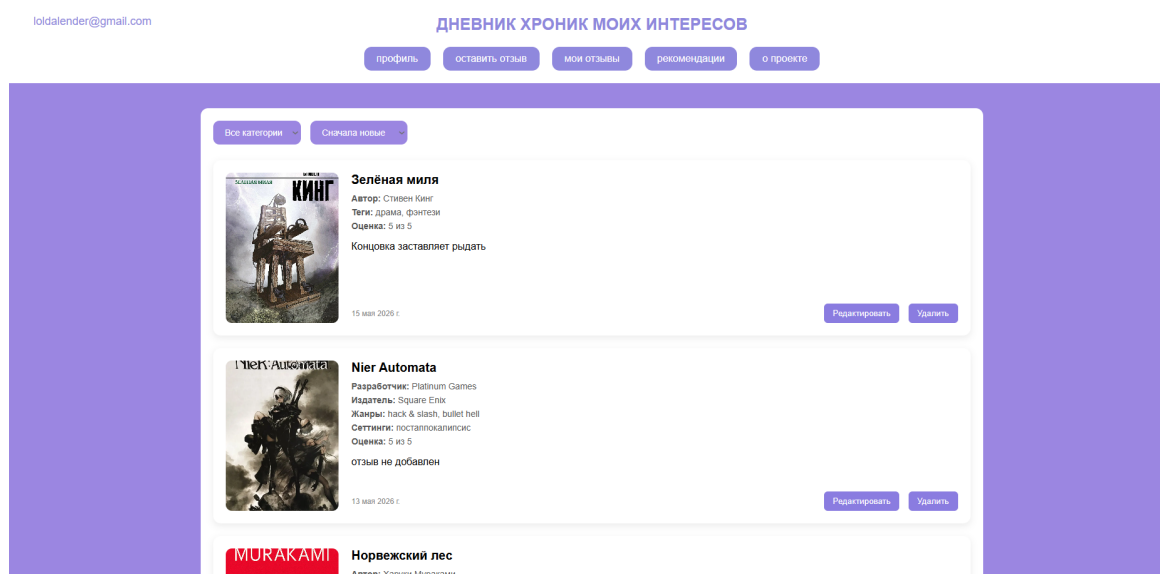
После форм был разработан обработчик создания отзывов `create_review`, который необходим для приема данных из HTML-форм и сохранения данных объекта и отзыва, которые пользователь введет, заполняя форму.

Для создания пользовательского интерфейса были разработаны HTML-шаблоны для формы написания отзыва (рис. 6), отображения списка оставленных отзывов с возможностью редактирования и удаления (рис. 7), и для отображения списка рекомендаций (рис. 8). Функциональная часть интерфейса разработана на языке JavaScript с использованием AJAX-запросов для создания динамично-обновляемого интерфейса.



The screenshot shows a web interface for a diary titled "ДНЕВНИК ХРОНИК МОИХ ИНТЕРЕСОВ". At the top, there are navigation buttons: "профиль", "оставить отзыв", "мои отзывы", "рекомендации", and "о проекте". Below these is a form for writing a review. The form has a dropdown menu for "Книга" (Book) with a plus icon. To the right of the dropdown are input fields for "Название:" (Title), "Автор:" (Author), and "Жанр:" (Genre). Below the genre field is a "Выберите теги..." (Select tags...) dropdown. There is a "Оценка" (Rating) dropdown set to "5". Below the rating is a large text area for "Отзыв" (Review). At the bottom of the form is a "Опубликовать" (Publish) button. The background is a solid purple color.

Рис. 6. Демонстрация формы написания отзыва



The screenshot shows a web interface for a diary titled "ДНЕВНИК ХРОНИК МОИХ ИНТЕРЕСОВ". At the top, there are navigation buttons: "профиль", "оставить отзыв", "мои отзывы", "рекомендации", and "о проекте". Below these is a list of reviews. The list has two tabs: "Все категории" (All categories) and "Сначала новые" (Newest first). The first review is for "Зелёная миля" (The Green Mile) by "Стивен Кинг" (Stephen King). It has a rating of "5 из 5" and the text "Концовка заставляет рыдать". The second review is for "Nier Automata" by "Platinum Games". It has a rating of "5 из 5" and the text "отзыв не добавлен". Each review entry has "Редактировать" (Edit) and "Удалить" (Delete) buttons. The background is a solid purple color.

Рис. 7. Список составленных ранее отзывов

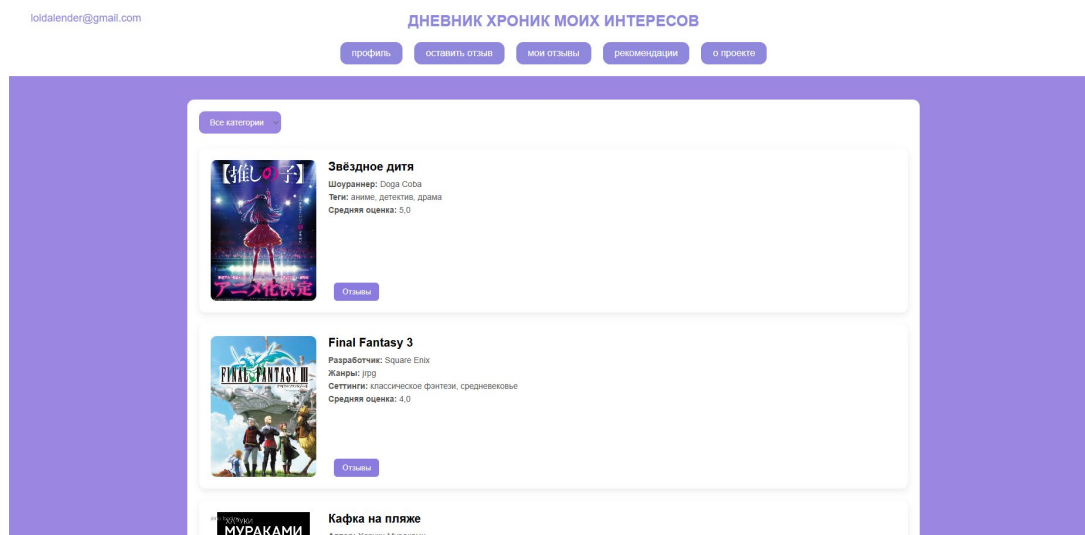


Рис. 8. Список рекомендаций

Для реализации системы рекомендаций был разработан отдельный модуль `recommendation.py`, отвечающий за анализ пользовательских предпочтений и подбор подходящих объектов. Основной функцией модуля является `get_user_recommendation(user)`, принимающая отзывы пользователя на объекты и возвращающая список рекомендаций.

Система рекомендаций работает следующим образом: из базы данных выбираются объекты, которым пользователь поставил высокую оценку. На основе этих данных система определяет предпочитаемые пользователем теги, жанры, сеттинги, авторов книг, режиссеров фильмов и т.д. Для фильтрации объектов используются ORM-запросы и выражения `Q`, позволяющие объединять несколько условий поиска. Из рекомендаций автоматически исключаются объекты, на которые пользователь уже оставлял отзывы. Дополнительно система рассчитывает среднюю пользовательскую оценку каждого объекта с помощью функции `Avg`.

Также для повышения качества рекомендаций дополнительно была реализована система ИИ-рекомендаций. Она анализирует не только оценки и теги объектов, но и текст отзывов пользователя. Основной функцией модуля является `get_ai_recommendations(user)`, которая получает все отзывы пользователя, формирует единый текстовый запрос и отправляет его в GigaChat по API.

В запросе передаются названия объектов, оценки, которые поставил пользователь и теги, на основе которых нейросеть анализирует предпочтения пользователя и формирует рекомендации среди всех объектов разных категорий, представленных в базе данных. Для взаимодействия с языковой моделью используется библиотека `gigachat`.

Заключение

В статье был рассмотрен процесс создания единого веб-ресурса для публикации мультикатегорийных отзывов и получения персональных рекомендаций. Це-

лью проекта стала разработка безопасного, удобного и масштабируемого веб-ресурса, предоставляющего пользователям возможность публиковать отзывы на объекты любимых категорий и получать наиболее точные рекомендации. В отличие от разработанного веб-ресурса – аналоги в виде IMDb или Отзовика являются узконаправленными, поддерживая лишь несколько категорий тематического контента или не предоставляют пользователям рекомендации.

Разработка велась на основе фреймворка Django с применением метода MVT, SQLite и GigaChat API для ИИ-анализа пользовательских отзывов и составления более точного списка рекомендаций. Были разработаны основные диаграммы (связей сущностей, прецедентов и последовательности), реализован функционал формы написания отзывов, списка оставленных отзывов и списка рекомендаций.

Итог: разработанный веб-ресурс предоставляет пользователям удобный сервис для поиска нового контента разных категорий через публикацию отзывов и получение персональных рекомендаций.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Использование диаграммы вариантов использования UML при проектировании программного обеспечения // Хабр URL: <https://habr.com/ru/articles/566218/> (дата обращения: 15.05.2024).
2. Диаграмма последовательности (sequence-диаграмма) // Хабр URL: <https://habr.com/ru/articles/814769/> (дата обращения: 15.05.2024).
3. ER-модель // Википедия URL: <https://ru.wikipedia.org/wiki/ER-модель> (дата обращения: 15.05.2024).
4. Django ORM для начинающих | Оптимизируем запросы // Хабр URL: <https://habr.com/ru/articles/503526/> (дата обращения 15.05.2026).
5. GigaChat API: Обзор | Документация для разработчиков // Sber Developer URL: <https://developers.sber.ru/docs/ru/gigachat/api/overview> (дата обращения 15.05.2026).

© Е. А. Плехоткин, Е. Ю. Воронкин, 2026