

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский политехнический университет»

Кафедра «Инфокогнитивные технологии»
Образовательная программа «Веб-технологии»

Отчет по курсовому проекту
по дисциплине «Основы серверной веб-разработки»

Тема: «Рекомендательная система по работе с книгами»

Выполнил:

Студент группы 221-321

Переверзев И. Д.

Подпись, дата

Принял:

Старший преподаватель:

Никишина И. Н.

Подпись, дата

МОСКВА 2023

Содержание

Содержание.....	2
Введение.....	3
Идея проекта.....	3
Актуальность.....	3
Аналоги.....	3
Основная часть.....	4
Проектирование.....	4
Выбор функционала.....	4
Подробное описание перечисленного функционала.....	5
Разработка и тестирование.....	12
Структура проекта.....	12
Описание моделей таблиц.....	12
Описание базы данных.....	12
Заключение.....	14
Источники.....	15

Введение

Идея проекта

Данный проект нацелен на изучение принципов работы с бесплатным фреймворком для создания веб-приложений Django. В качестве темы была выбрана «Рекомендательная система по работе с книгами». Это значит, что полученный в ходе работы продуктовый результат представлять из себя инструмент, на который пользователи могли бы опираться при выборе литературы различного толка: начиная от научных статей и заканчивая фантастическими романами.

Актуальность

Книги — одно из самых главных богатств человечества, поскольку они передают опыт, устремления и достижения мысли прошлых поколений будущим. Книги могут быть просто увлекательными, а могут быть и нашими друзьями и учителями. В цифровую эпоху они не утратили своей значимости, однако способы делиться любимыми произведениями претерпели значительные изменения. На смену книжным клубам пришли различные форумы, и теперь мнением о прочитанном в виде рецензий и отзывов делятся в сети.

Следовательно, задачей книжной рекомендательной системы (РС) является предложение пользователю тех произведений, которые с наибольшей вероятностью ему понравятся.

Аналоги

В ходе работы над проектом был проведен анализ некоторых платформ, выполняющих функции РС. Наиболее подходящей была выбрана платформа «Библа».

Данный сервис функционирует с 2009 года и помогает своим пользователям ориентироваться в огромном многообразии различной литературы. Осуществляется это путем предоставления следующего функционала:

1. Просмотр общей библиотеки
2. Добавление книг в различные разделы личного кабинета пользователя. Например, «Список прочитанных книг», «Хочу прочитать», «У меня есть» и т. д.
3. Удобный поиск по названию книг, их авторам или с помощью системы тегов.
4. Составление различных тематических подборок.

Основная часть

Проектирование

Выбор функционала

Основная задача рекомендательной системы заключается в том, чтобы помогать пользователю находить литературу, соответствующую его интересам. Современные сервисы реализуют данную задачу с различными способами.

Некоторые вводят системы тегов, присваивание которых помогает определить нужную книгу, даже не имея точного представления о ней. Некоторые добавляют функцию «жесткого» поиска по названию книги.

Однако, так как данный проект преследует цель познакомиться с фреймворком Django, для реализации был определен следующий функционал:

1. Просмотр всех книг, добавленных в базу данных (БД), на соответствующей странице.
2. Комфортное добавление книг, авторов и других составляющих в базу данных посредством системы администрирования (админки).
3. Просмотр всех авторов, добавленных в базу данных, на соответствующей странице.
4. Просмотр профиля конкретного автора на отдельной странице.
5. Просмотр всех книг конкретного автора на отдельной странице.
6. Поиск по названию книг с последующим отображением на отдельной странице.
7. Поиск по ФИО добавленных в БД авторов с последующим отображением на отдельной странице.
8. Возможность просмотра профиля конкретной книги, в котором будет представлена более подробная информация о ней. Просмотр профиля книги осуществляется на отдельной странице.
9. Возможность взаимодействия с «Черным списком»: добавление и удаление имеющихся в базе данных книг.
10. Возможность взаимодействия с списком «Любимых» книг: добавление и удаление имеющихся в базе данных книг.
11. Возможность просмотра чужих комментариев, оставленных другими пользователями к конкретной книге и добавления собственных. Автор имеет опцию удалить свой комментарий в случае надобности.
12. Оценивание каждой конкретной книги посредством кнопок «Нравится» и «Не нравится». На основе оценок пользователей складывается общий рейтинг книги.

13. Рекомендательная система должна предоставлять функционал для регистрации новых пользователей и авторизации старых. Авторизованные пользователи имеют больше возможностей взаимодействия с РС.

Подробное описание перечисленного функционала

1. Рекомендательная система должна предоставлять пользователю возможность просматривать все имеющиеся в ее базе данных книги (см. рис. 1). Каждая книга должна иметь собственную карточку, по нажатию на которую пользователь переходит в профиль выбранной книги. В случае, если книг слишком много для отображения на одной странице, система должна разделять их на несколько страниц.

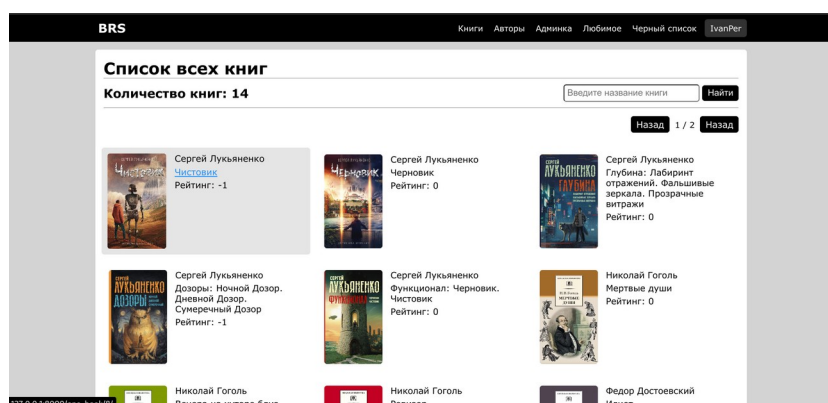


Рис. 1

2. Система администрирования рекомендательной системы должна предоставлять владельцу удобный способ добавления новых записей в базу данных. Для этого необходимо чтобы:
 1. в отображались админке все добавленные модели для взаимодействия с таблицами базы данных (см. рис. 2);

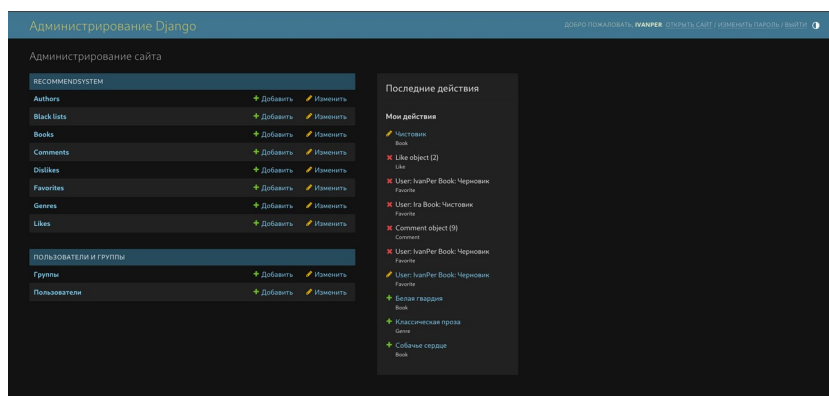


Рис. 2

2. при переходе к той или иной модели в админке отображались основные поля добавленных объектов для более комфортной навигации и быстрого поиска нужного значения. Также необходимо добавить поиск и фильтрацию по ключевым значениям. Продемонстрировано на примере модели книг (см. рис. 3)

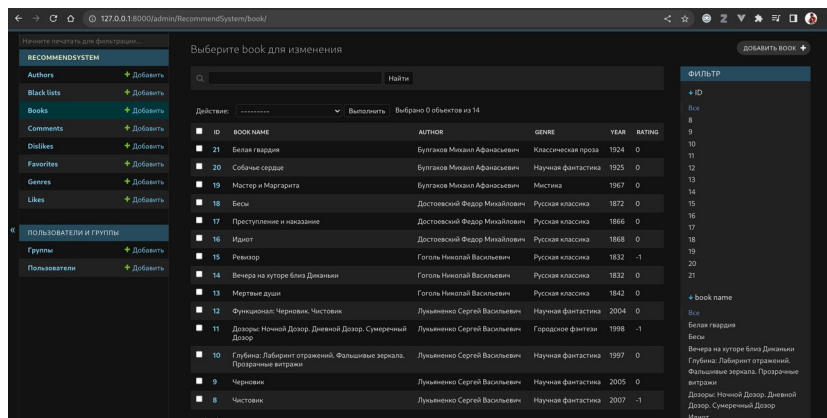


Рис. 3

3. При переходе к созданию нового экземпляра определенной модели пользователь должен видеть все необходимые поля. Каждое поле должно быть названо так, чтобы администратор сразу понял, какой тип данных нужно использовать. Продemonстрировано на примере модели книг (см. рис. 4)

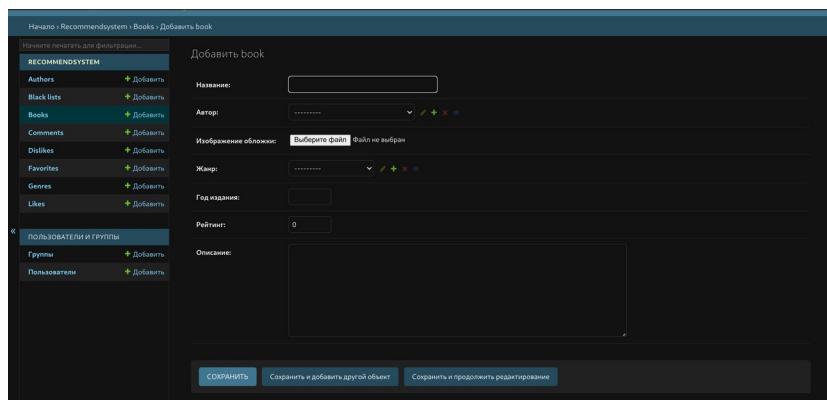


Рис. 4

3. РС должна предоставлять пользователю возможность посмотреть всех авторов, добавленных в базу данных. Каждому автору соответствует собственная карточка, по нажатию на которую пользователь переходит на страницу просмотра профиля выбранного автора. (см. рис. 5)

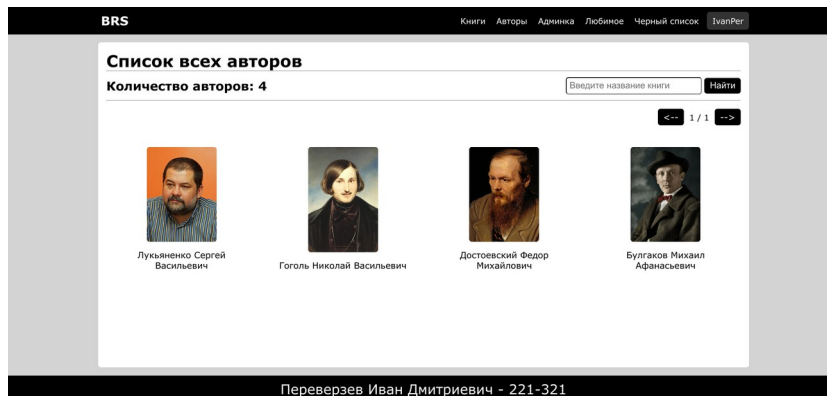


Рис. 5

4. По нажатии на карточку автора пользователь должен переходить на страницу просмотра профиля автора. Данная страница должна отображать загруженную фотографию выбранного автора, его краткое описание, а также ссылку, ведущую на страницу со всеми написанными им произведениями. (см. рис. 6)

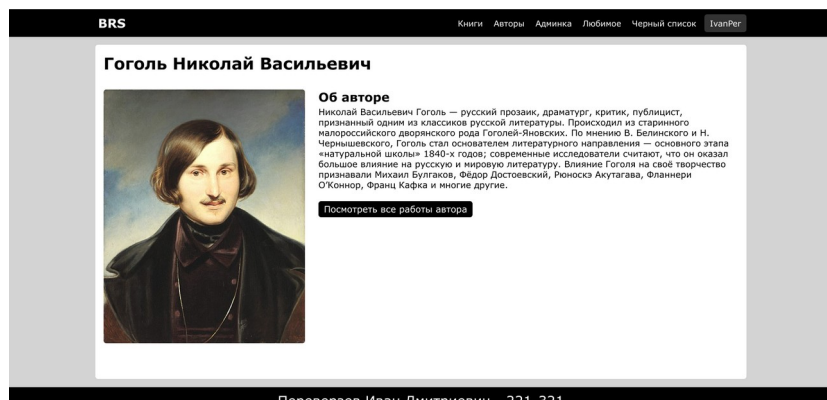


Рис. 6

5. По нажатии на соответствующую кнопку в профиле автора пользователь должен перейти к списку всех написанных выбранным автором книг (см. рис. 7). Требования к отображению найденных книг такие же, что и к отображению всех имеющихся в базе данных книг.

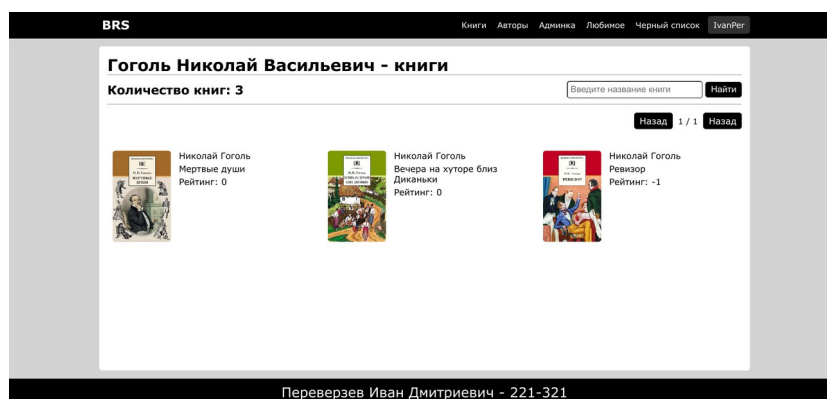


Рис. 7

6. Рекомендательная система должна предоставляться пользователям возможность поиска интересующей книги по ее названию. Соответствующее поле должно присутствовать на страницах, предназначенных для отображения списка книг. Требования к отображению списка найденных по запросу книг такие же, как и для отображения всех имеющихся в базе данных книг книг.



Рис. 8

7. Пользователь должен иметь возможность найти интересующего его автора по его фамилии, имени или отчеству. Поле для ввода поискового запроса должно присутствовать на страницах, предназначенных для отображения списка авторов. Требования к отображению найденных по запросу авторов такие же, как и для отображения всех имеющихся в базе данных авторов. (см. рис. 9)

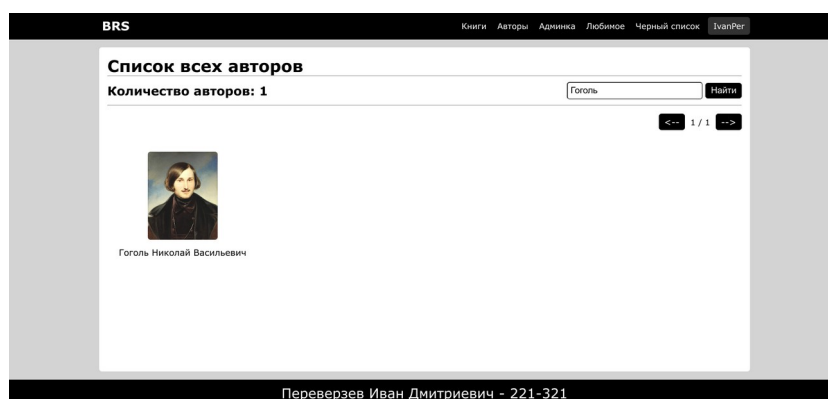


Рис. 9

8. По нажатии на карточку, выбранную из списка книг, пользователь должен переходить на страницу для просмотра профиля выбранной книги (см. рис. 10). Данный профиль должен отображать следующую информацию:
1. Автор книги (с ссылкой на профиль);
 2. Жанр;
 3. Год издания;
 4. Рейтинг;
 5. Описание
 6. Комментарии
 7. Кнопки взаимодействия (если пользователь авторизован) (см. рис. 11)

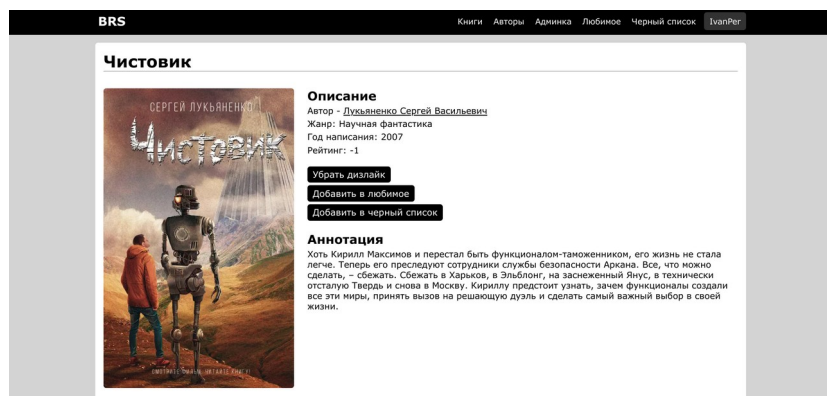


Рис. 10



Рис. 11

9. Пользователь должен иметь возможность добавлять книги, которые ему очень не нравятся, в черный список. Данное действие осуществляется через соответствующую кнопку в профиле книги (см. рис 10). Только авторизованные пользователи могут добавлять книги в черный список. Добавленные в черный список книги не будут отображаться в общих списках книг. Посмотреть их можно будет только в соответствующем списке (см. рис. 12). Если пользователь захочет удалить книгу из черного списка, он может сделать это через профиль книги, доступный через выше упомянутый список. Если книга уже была добавлена в список любимых, пользователь не сможет добавить ее в черный список. Для этого сначала понадобится удалить книгу из списка любимых.

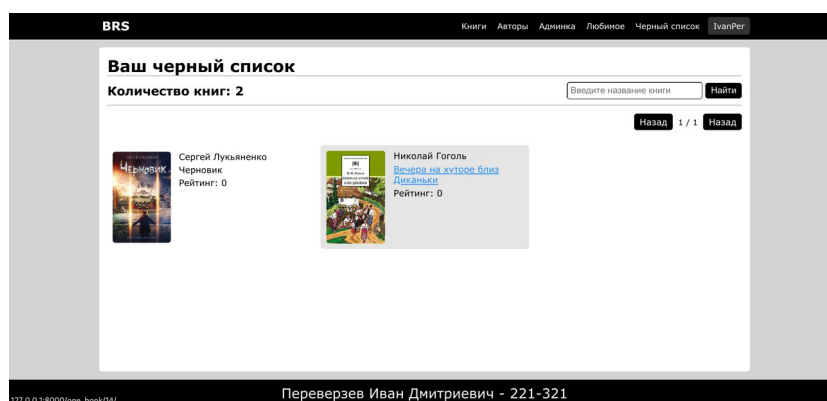


Рис. 12

10. Кроме взаимодействия с Черным списком, рекомендательная система должна позволять пользователю добавлять понравившиеся книги в список любимых.

Сделать это можно с помощью соответствующей кнопки в профиле книги (см. рис. 13). Добавленные в список любимых книги можно посмотреть на соответствующей странице (см. рис. 14). Если пользователь захочет удалить книгу из списка любимых, он может сделать это через профиль книги. Если книга уже была добавлена в черный список, пользователь не сможет добавить ее в список любимых. Для этого сначала понадобится удалить книгу из черного списка.

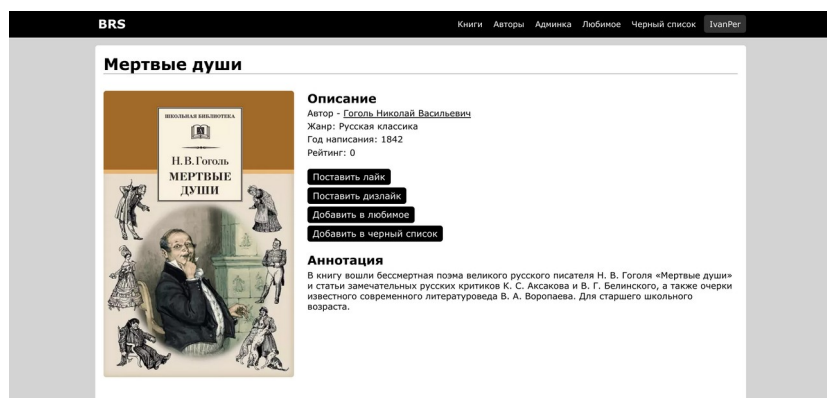


Рис. 13

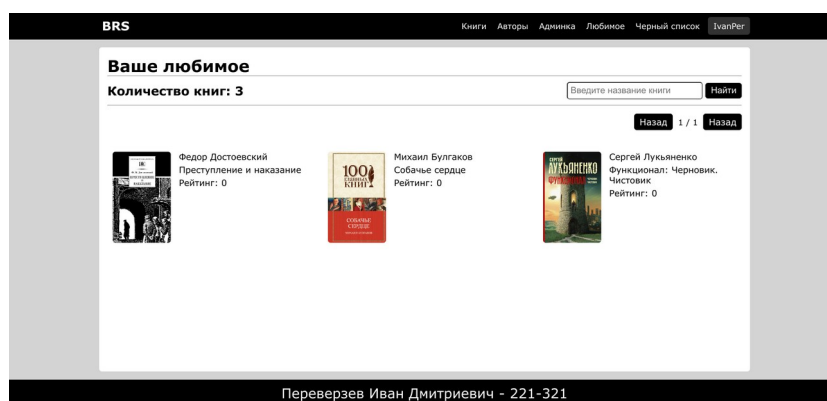


Рис. 14

11. Рекомендательная система должна предоставлять пользователю возможность просмотра комментариев, оставленных другими пользователями к определенной книге. Просматривать комментарии могут все пользователи, однако добавлять — могут только авторизованные. Для этого необходимо нажать кнопку «Оставить отзыв». Откроется форма по заполнению которой комментарий будет добавлен. Комментарии можно посмотреть в профиле книги, к которой они были оставлены. Если пользователь захочет, он может удалить свой комментарий. Пример приведен на рисунке 11.
12. Пользователь должен иметь возможность оценивать каждую отдельную книгу, тем самым влияя на ее общий рейтинг популярности. Сделать это можно в профиле книги с помощью соответствующих кнопок (см. рис. 13). Если пользователь изменит свое мнение о ранее оцененной книге, ему сначала придется отменить свою предыдущую оценку, а затем поставить новую (см. рис. 10).

13. Рекомендательная система должна предоставлять пользователю возможность зарегистрироваться в базе данных для получения новых привилегий. Для этого необходимо перейти на соответствующую страницу через навигационной меню. На странице регистрации представлена форма, при корректном заполнении которой пользователь будет зарегистрирован в сразу же авторизован системой (см. рис. 15). Если пользователь уже проходил регистрацию на сайте, он может авторизоваться перейдя на соответствующую страницу через навигационное меню. На странице также будет отображена форма, заполнение которой позволит пользователю авторизоваться в системе (см. рис. 16).

The screenshot shows a web application interface for registration. At the top, there is a black navigation bar with the text 'BRS' on the left and a menu with 'Книги', 'Авторы', 'Админка', 'Регистрация', and 'Вход' on the right. The 'Регистрация' item is highlighted. Below the navigation bar, the page title 'Авторизация' is displayed. The main content area contains a registration form with three input fields: 'Имя пользователя', 'Пароль', and 'Подтверждение пароля'. A black button labeled 'Отправить' is positioned to the right of the third field. At the bottom of the page, a black footer bar contains the text 'Переверзев Иван Дмитриевич - 221-321'.

Рис. 15

The screenshot shows a web application interface for login. It has the same layout as the registration page, with a black navigation bar at the top containing 'BRS' and a menu with 'Книги', 'Авторы', 'Админка', 'Регистрация', and 'Вход'. The 'Регистрация' item is highlighted. The page title 'Авторизация' is shown. The main content area contains a login form with two input fields: 'Имя пользователя:' and 'Пароль:'. A black button labeled 'Отправить' is located to the right of the second field. The black footer bar at the bottom contains the text 'Переверзев Иван Дмитриевич - 221-321'.

Рис. 16

Разработка и тестирование

Структура проекта

Как и многие фреймворки, для комфортной и эффективной разработки Django предлагает свою структуру проекта, которой необходимо следовать.

В ходе работы над данным проектом были созданы два django-приложения:

1. RecommendSystem — основное приложение, через которое осуществляется работа всего проекта. В нем содержатся все главные адреса (руты), все статические файлы (стили, изображения и шаблоны). Также именно в рамках данного приложения были реализованы основные модели для работы с таблицами базы данных.
2. Users — приложение для реализации функционала авторизации и регистрации пользователей. Функционал авторизации был реализован с помощью встроенных в django инструментов. Для добавления регистрации пользователей были созданы дополнительные форма и класс представления.

Описание моделей таблиц

В ходе реализации проекта были созданы следующие модели для работы с таблицами базы данных:

1. Genre -модель для работы с жанрами книг.
2. Author — модель для работы с авторами книг.
3. Book — модель для непосредственно работы с книгами.
4. Comment — модель для работы с комментариями.
5. Favorite — модель для работы с книгами, добавленными в список любимых.
6. BlackList — модель для работы с книгами, добавленными в черный список.
7. Like — модель для работы с работами с оценками «Нравится»
8. Dislike — модель для работы с оценками «Не нравится»

Описание базы данных

На основе выше перечисленных моделей были созданы следующие таблицы базы данных со следующими полями:

1. RecommendSystem_author:

Имя	Тип данных	Первичный ключ	Внешний ключ	Уникальность	Проверка	Не NULL	Сравнение	Сгенерированное	Значение по умолчанию
1 id	integer	?				☑			NULL
2 firstname	varchar (200)					☑			NULL
3 lastname	varchar (200)					☑			NULL
4 patronymic	varchar (200)					☑			NULL
5 description	text								NULL
6 image	varchar (100)					☑			NULL

2. RecommendSystem_blacklist:

Имя	Тип данных	Первичный ключ	Внешний ключ	Уникальность	Проверка	Не NULL	Сравнение	Сгенерированное	Значение по умолчанию
1 id	integer								NULL
2 book_id	bigint								NULL
3 user_id	integer								NULL

3. RecommendSystem_comment:

Имя	Тип данных	Первичный ключ	Внешний ключ	Уникальность	Проверка	Не NULL	Сравнение	Сгенерированное	Значение по умолчанию
1 id	integer								NULL
2 text	text								NULL
3 created_at	date								NULL
4 book_id	bigint								NULL
5 author_id	integer								NULL

4. RecommendSystem_book:

Имя	Тип данных	Первичный ключ	Внешний ключ	Уникальность	Проверка	Не NULL	Сравнение	Сгенерированное	Значение по умолчанию
1 id	integer								NULL
2 book_name	varchar (200)								NULL
3 year	integer								NULL
4 rating	integer								NULL
5 genre_id	bigint								NULL
6 author_id	bigint								NULL
7 image	varchar (100)								NULL
8 annotation	text								NULL

5. RecommendSystem_dislike:

Имя	Тип данных	Первичный ключ	Внешний ключ	Уникальность	Проверка	Не NULL	Сравнение	Сгенерированное	Значение по умолчанию
1 id	integer								NULL
2 book_id	bigint								NULL
3 user_id	integer								NULL

6. RecommendSystem_favorite:

Имя	Тип данных	Первичный ключ	Внешний ключ	Уникальность	Проверка	Не NULL	Сравнение	Сгенерированное	Значение по умолчанию
1 id	integer								NULL
2 book_id	bigint								NULL
3 user_id	integer								NULL

7. RecommendSystem_genre:

Имя	Тип данных	Первичный ключ	Внешний ключ	Уникальность	Проверка	Не NULL	Сравнение	Сгенерированное	Значение по умолчанию
1 id	integer								NULL
2 genre_name	varchar (200)								NULL

8. RecommendSystem_like:

Имя	Тип данных	Первичный ключ	Внешний ключ	Уникальность	Проверка	Не NULL	Сравнение	Сгенерированное	Значение по умолчанию
1 id	integer								NULL
2 book_id	bigint								NULL
3 user_id	integer								NULL

Каждая из представленных таблиц соответствует определенной модели, описанной в коде Django-приложения RecommendSystem.

Заключение

В ходе работы над проектом были изучены и применены многие аспекты работы с фреймворком Django. А именно:

1. Создание нового проекта Django.
2. Создание нового Django-приложения.
3. Формирование базы данных путем определения классов моделей.
4. Маршрутизация (рутинг).
5. Обработка запросов с помощью функций и классов представлений.
6. Шаблонизация html-страниц.
7. Настройка системы администрирования (админки) для комфортного управления проектом.
8. Создание тестов для отладки процессов.
9. Работа с пользователями.
10. Работа с миграциями базы данных.
11. Работа с кастомными формами.
12. Добавление и использование статических файлов.

Таким образом, я считаю, что поставленные перед этим проектом цели: разработка рекомендательной системы и знакомство с фреймворком Django, были выполнены.

Источники

В этой части приведены основные источники, использовавшиеся для реализации проекта:

1. Документация языка программирования Python - <https://docs.python.org/3/>
2. Документация фреймворка Django - <https://docs.djangoproject.com/en/4.2/>
3. Django tutorial - <https://docs.djangoproject.com/en/4.2/intro/tutorial01/>
4. Рекомендательный сервис книг «Библа» - <http://bibla.ru/about/>
5. Литрес — источник информации для заполнения БД - <https://www.litres.ru/>
6. Видеокурс по работе с фреймворком Django - https://www.youtube.com/watch?v=FyTL1bnUx5I&list=PLA0M1Bcd0w8xO_39zZl12u1lz_Q-Mwn1F&ab_channel=selfedu