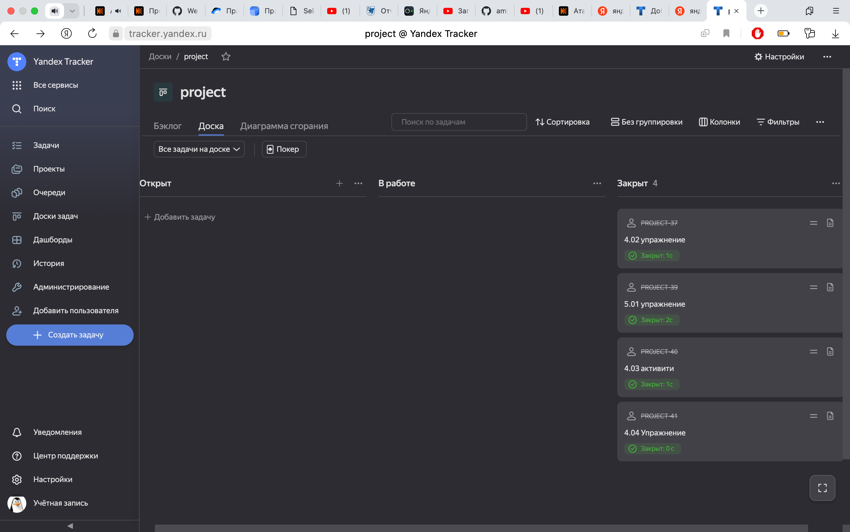
**Спринт 08.11 – 21.11**

**Газизова Амина**

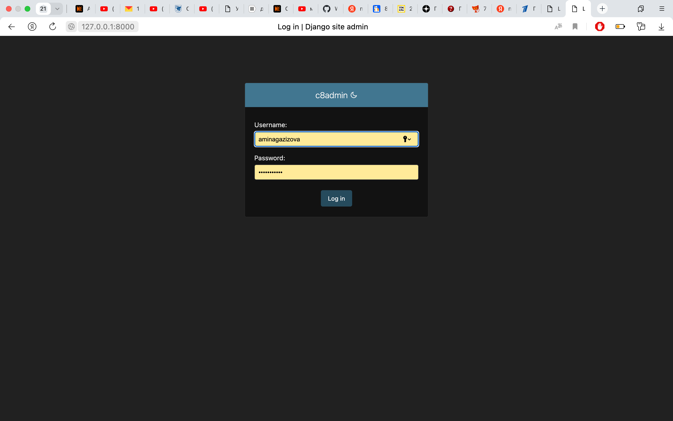
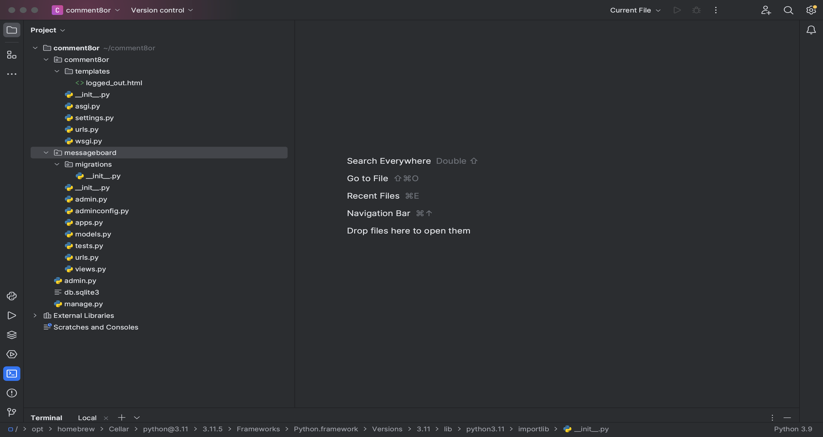
**Группа 11-311**

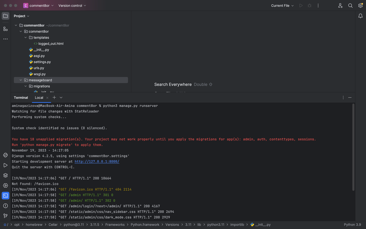
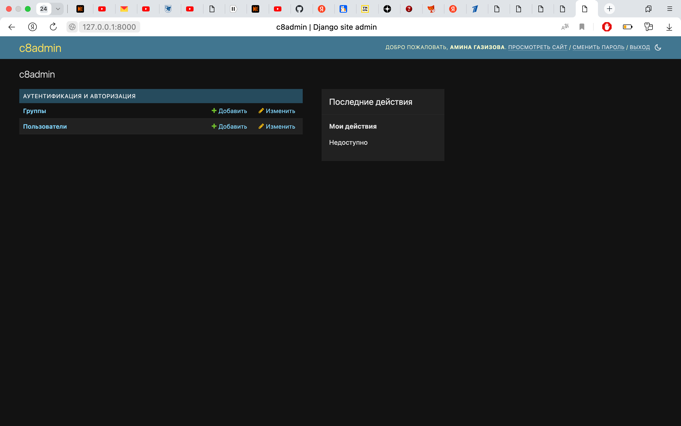
**177-226 страницы**

****

**Упражнение 4.02**

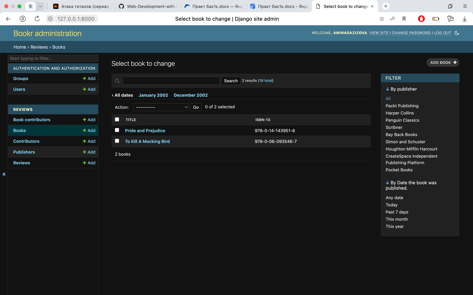
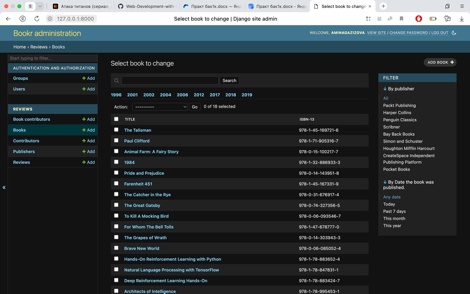
В этом упражнение я создала новый проект, используя все знания, которые я получила в изучении предмета. Создала проект под названием comment8or и приложение massageboard. Создала суперпользователя и зашла на сайт.

****

****

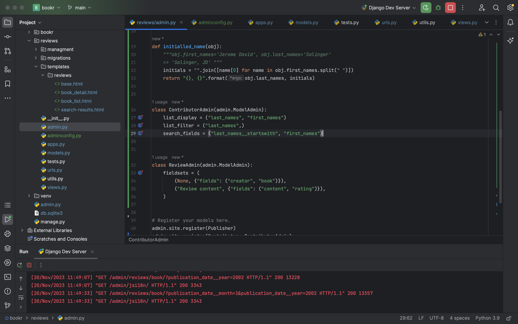
**Упражнение 4.04**

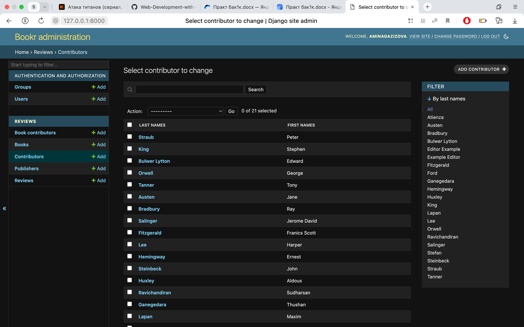
В этом упражнение я добавила фильтр даты.



**Активити 4.03**

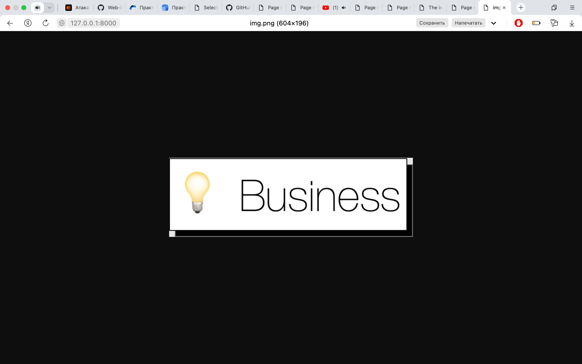
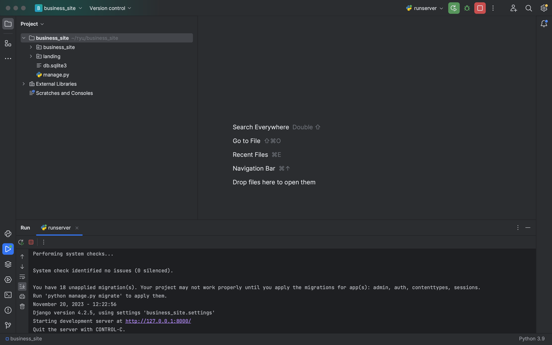
В этом упражнение я модифицировала Contributor и добавила класс ContributorAdmin.





**Упражнение 5.01**

В этом упражнение мы создали новый проект business\_site и добавили логотип на лицевую страницу сайта. Также мы создали новую конфигурацию runserver.



**Вывод:**

*Этот спринт мне понравился, я создала проект с нуля, научилась добавлять фотографии и модифицировала классы. При этом, у меня возникали трудности, из-за которых я долго сидела на одном месте, поэтому этот спринт я делала дольше обычного. В общем, мне понравилось делать этот спринт!*

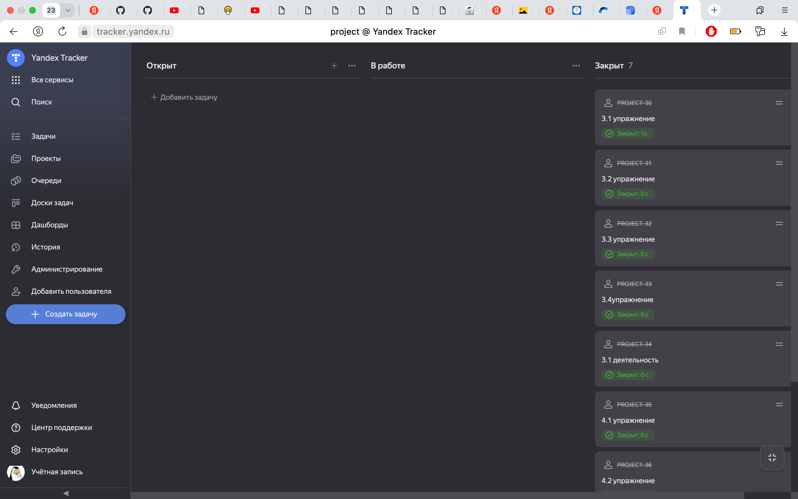
**Спринт 18.10-31.10**

**Газизова Амина**

**Группа 11-311**

**123-177 стр.**

**https://github.com/aminagazizova/Bookr/tree/main**

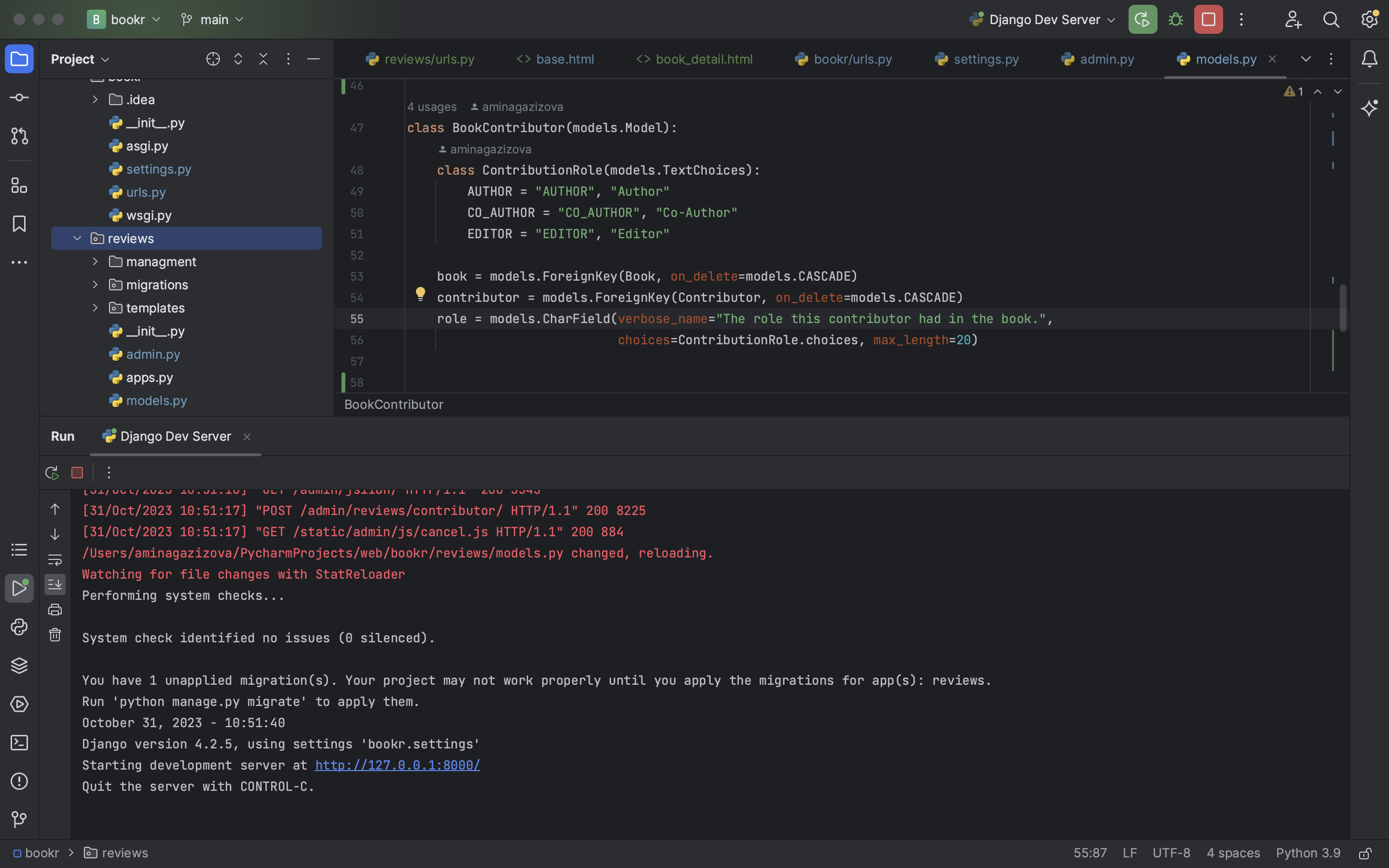
****

**Упражнение 3.1**

Я создала веб-страницу, с помощью которой отображается простое сообщение для пользователя и показывается какое количество книг находится в базе данных. Здесь у меня была проблема, но я ее решила, углубившись в упражнение.

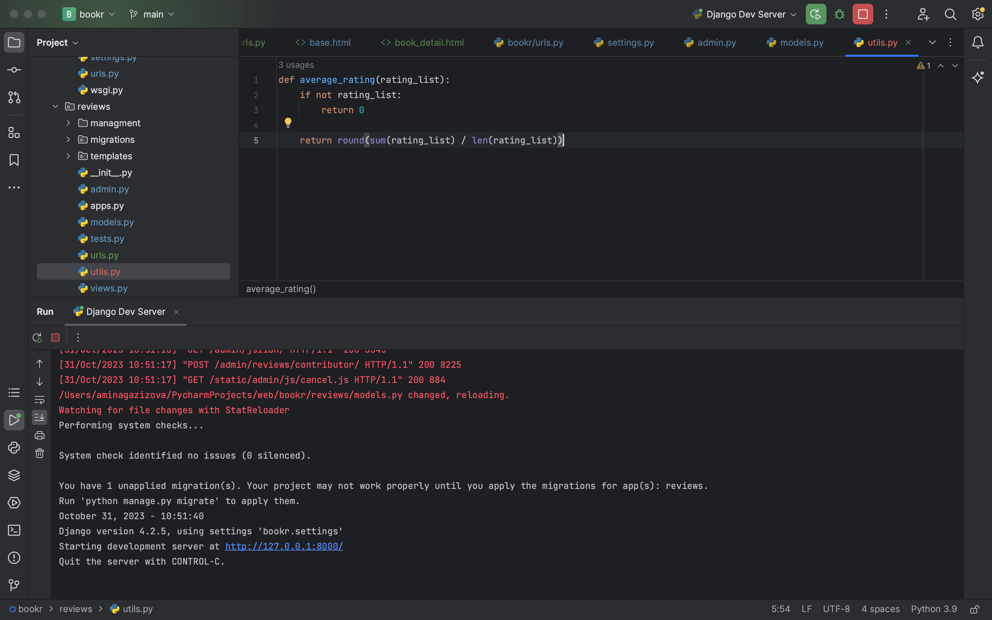
**Упражнение 3.2**

Здесь мы с помощью шаблона вывели фразу Welcome to Bookr!



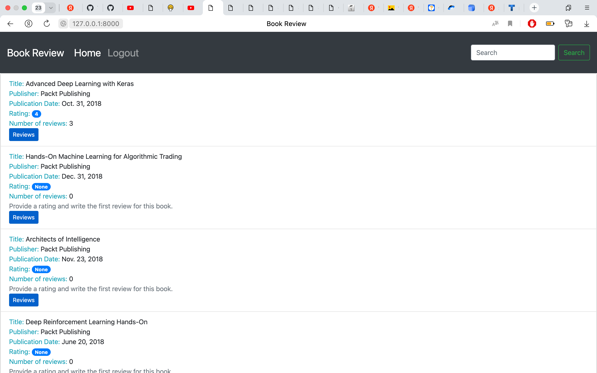
**Упражнение 3.3**

С помощью этого упражнения у меня получилось создать страничку, в которой будет тображаться список всех книг, их рейтинги и количество рецензий. Проблем в этом упражнение не было.



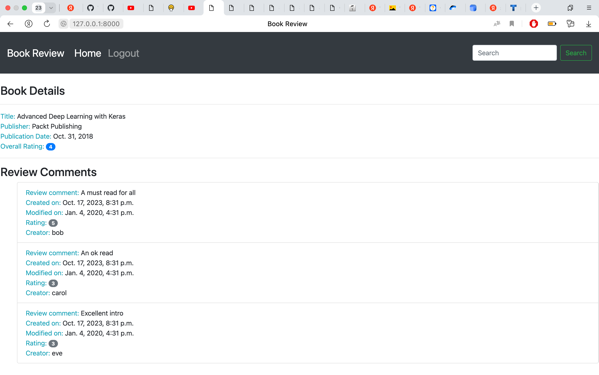
**Упражнение 3.04**

Здесь я сделала оформление с помощью Bootstrap.



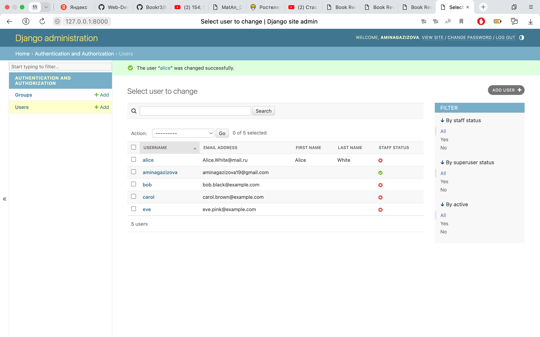
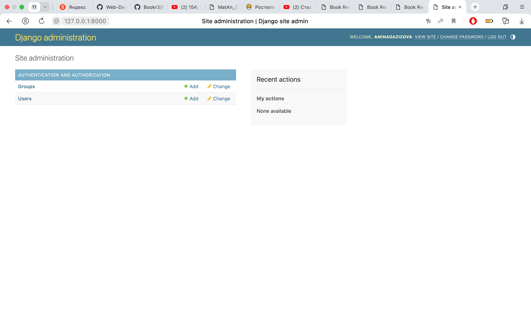
**Деятельность 3.01**

В этом упражнение я реализовала новое представление, шаблон и URL-маппинг для отображения следующих сведений о книге: название, издательство, дата публикации и общая оценка.



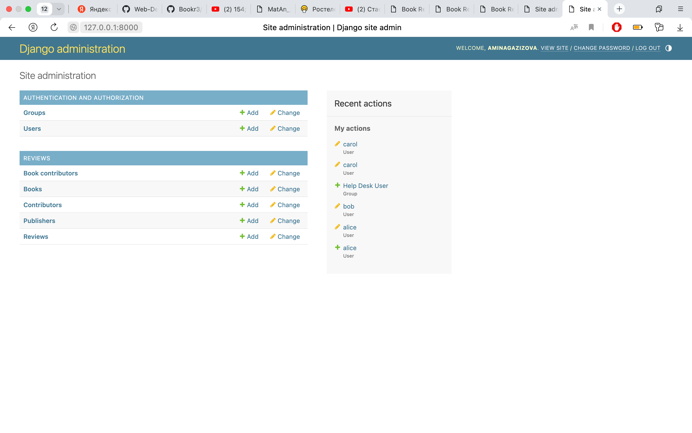
**Упражнение 4.01**

В этом упражнение я создала учетную запись для админа. Также я добавила еще пользователей в наш веб сайт.

****

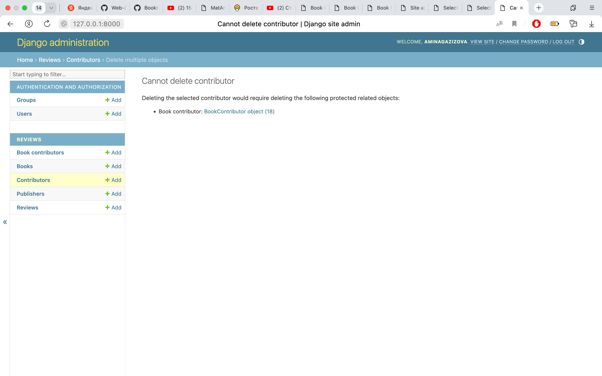
**Упражнение 4.02**

Здесь мы создали группу Help Desk User, добавили туда Carol и назначили его администратором группы.



**Упражнение 4.03**

Мы поменяли функцию удаления для удобства, а потом поменяли его обратно.



**Вывод:**

Этот спринт не показался мне сложным, он показался мне самым интересным среди всех. Мне кажется дело в том, что в этом упражнение мы работали напрямую с сайтом, то есть мы уже видим как наши старания вырисовываются в целостную картинку и это очень круто! Мне очень понравилось работать над этим спринтом.

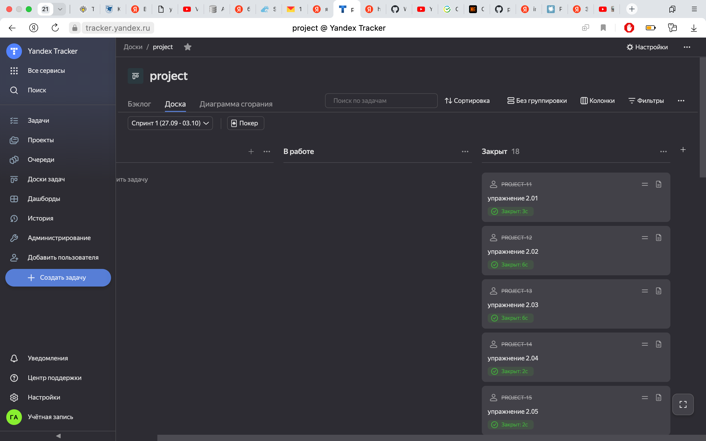
**Спринт 04.10-17.10**

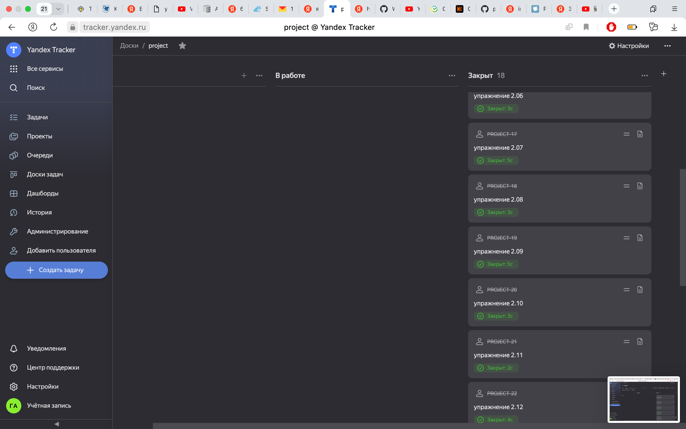
**Газизова Амина**

**Группа 11-311**

**65-121 стр.**

**https://github.com/aminagazizova/Bookr**

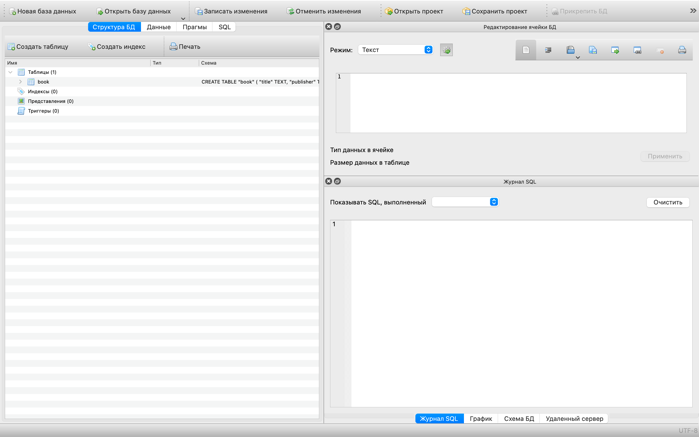
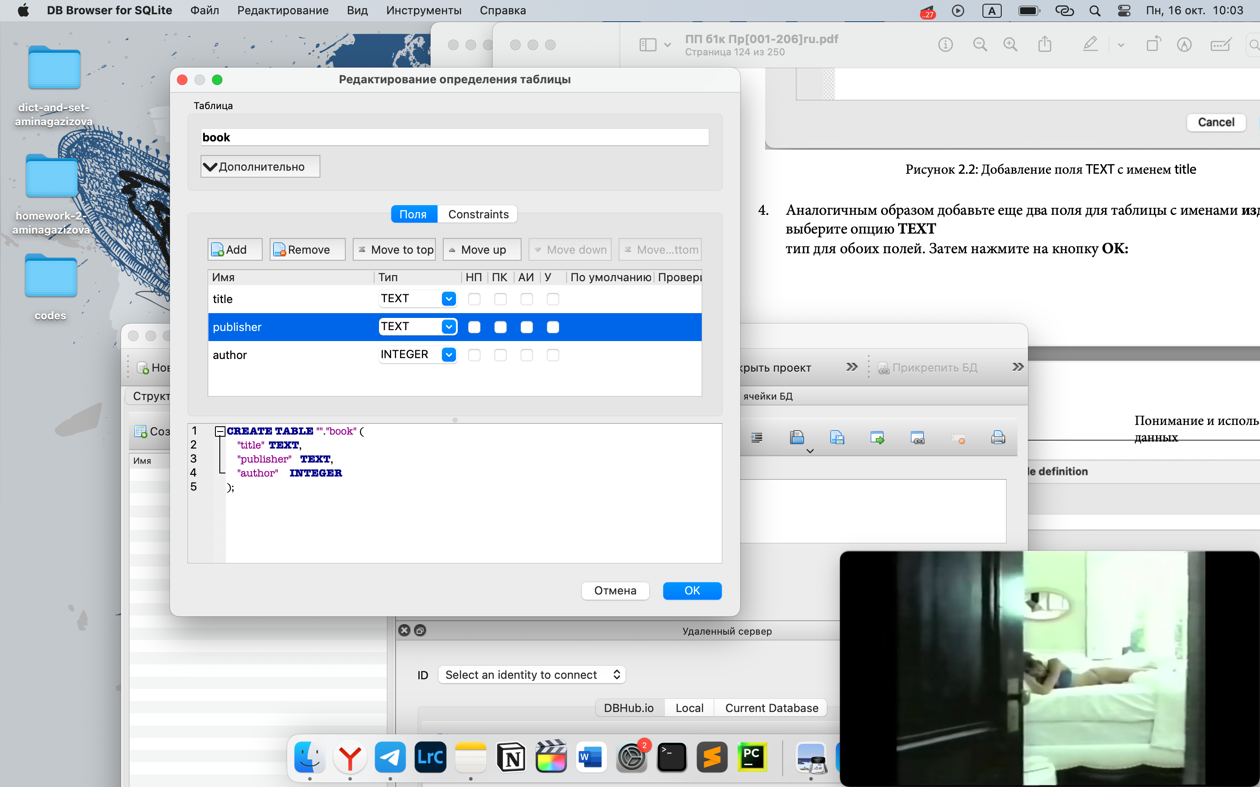
****

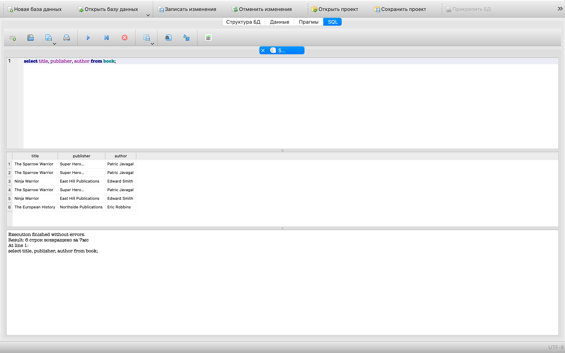
****

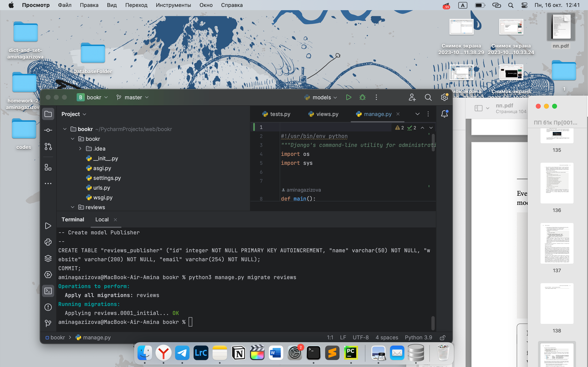
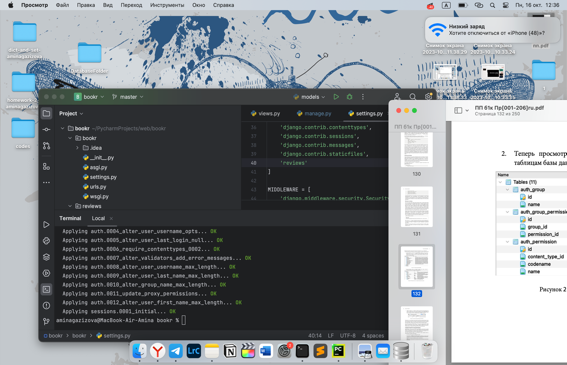
**\***скрин с яндекс трекера

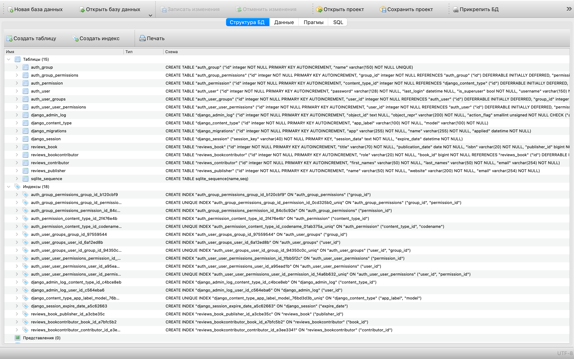
**Упражнение 2.01**

Здесь я создала базу данных книг для приложений. Были сложности с самым первым действием, я не могла создать таблицу, мне выдавало ошибку, я просидела несколько часов, а оказалось я просто не добавила папку, в которую нужно будет загрузить файл. Также я научилась выполнять операции в SQL. Я узнала, что такое миграция на Django, создала модели и миграции для Bookr. Я узнала, что такое первичные ключи, узнала, что бывают различные Relationships**,** такие как -Many-to-one, Many-to-many, One-to-one.



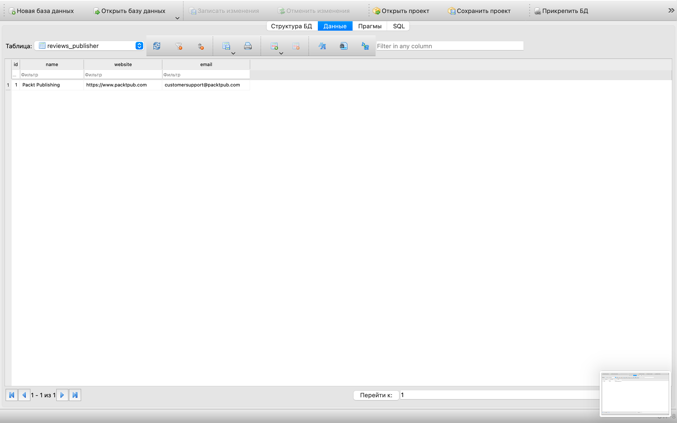
****

****

** \***скрины с первого упражнения

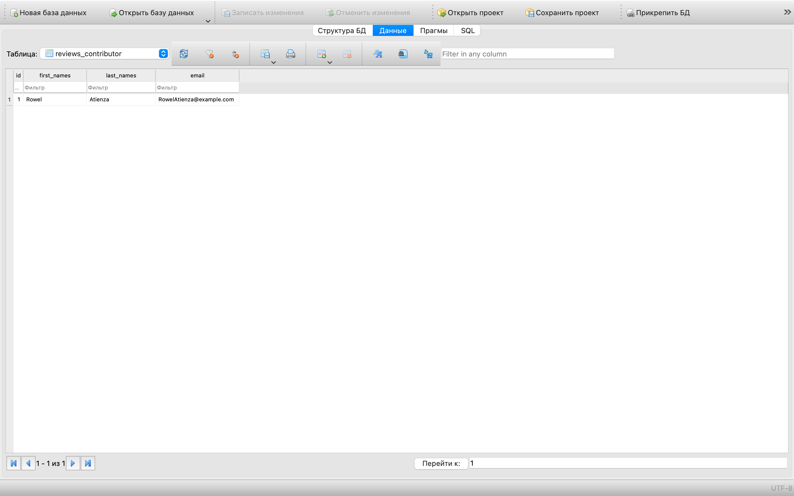
**Упражнение 2.02**

Здесь я создала запись в Publisher таблице в базе данных. Это упражнение не вызвало у меня каких-то проблем.



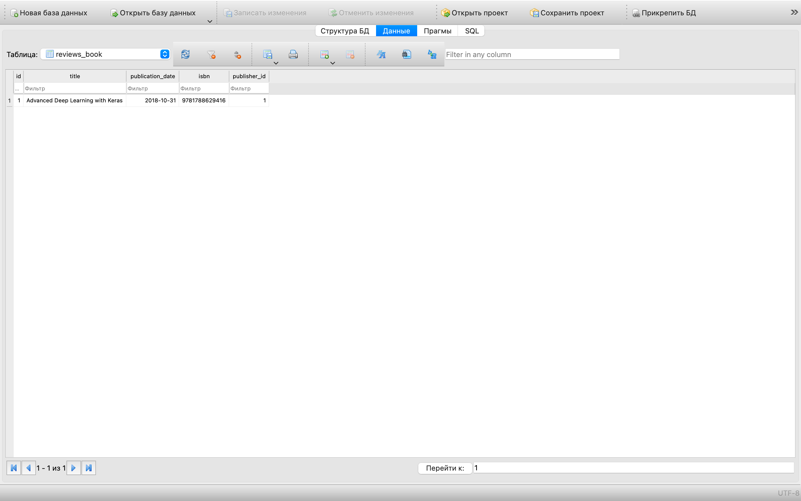
**Упражнение 2.03**

Здесь я создала запись в Contributor таблице, используя метод сreate().



**Упражнение 2.04**

Здесь я создала запись в таблице Book, используя метод **many-to-one.**

****

**Упражнение 2.05**

Здесь я создала отношение **many-to-many** между contributor и book.

Скрин забыла сделать…. Прстите…

**Упражнение 2.06**

Здесь я научилась создавать отношения **many-to-many,** но уже с помощью метода add().

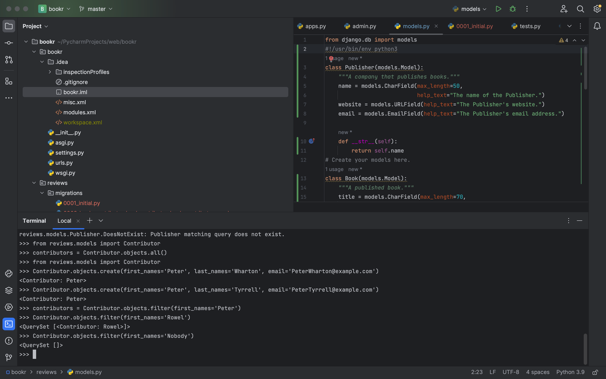
**Упражнение 2.07**

В этом упражнение я получила объект из базы данных с помощью метода get().

Здесь я тоже не сделала скрин, но в последующих скринах видно, что я все сделала правильно.

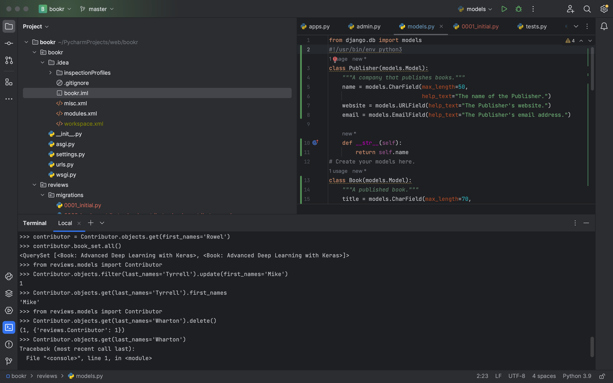
**Упражнение 2.08**

В этом упражнение я научилась работать с методом all().



**Упражнение 2.09**

Тут я узнала про метод filter() и делала разные упражнения с ним.



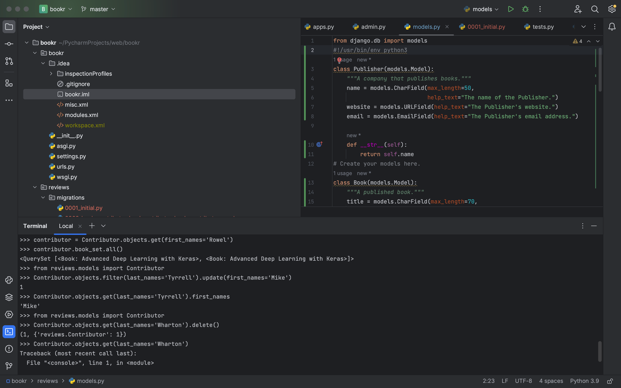
**Упражнение 2.10**

В этом упражнение я сделала запрос для получения всех авторов книги 'The Talisman’.

Я опять же не сделала скрин…

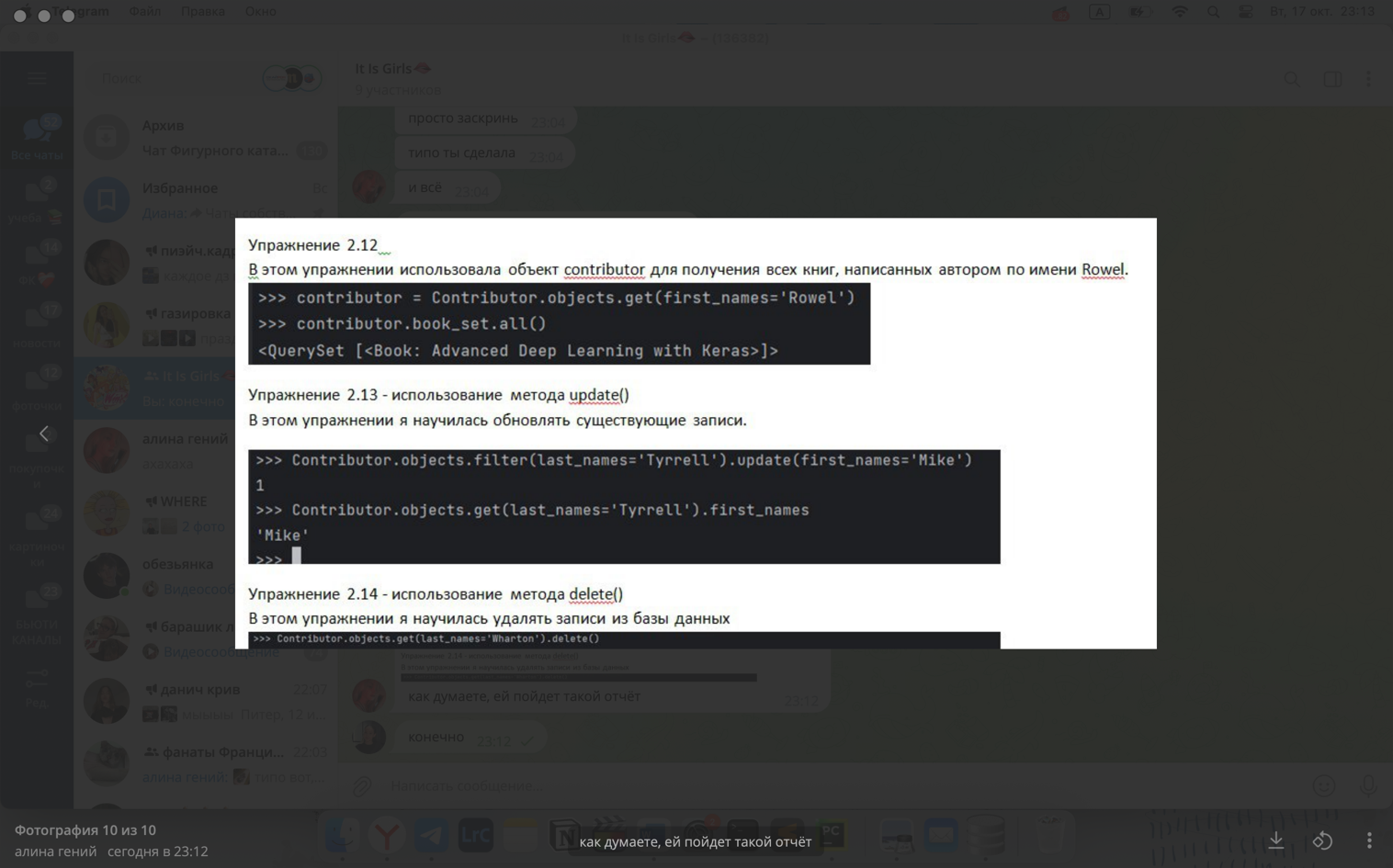
**Упражнение 2.11**

В этом упражнение я нашла всех авторов книги с помощью Book.



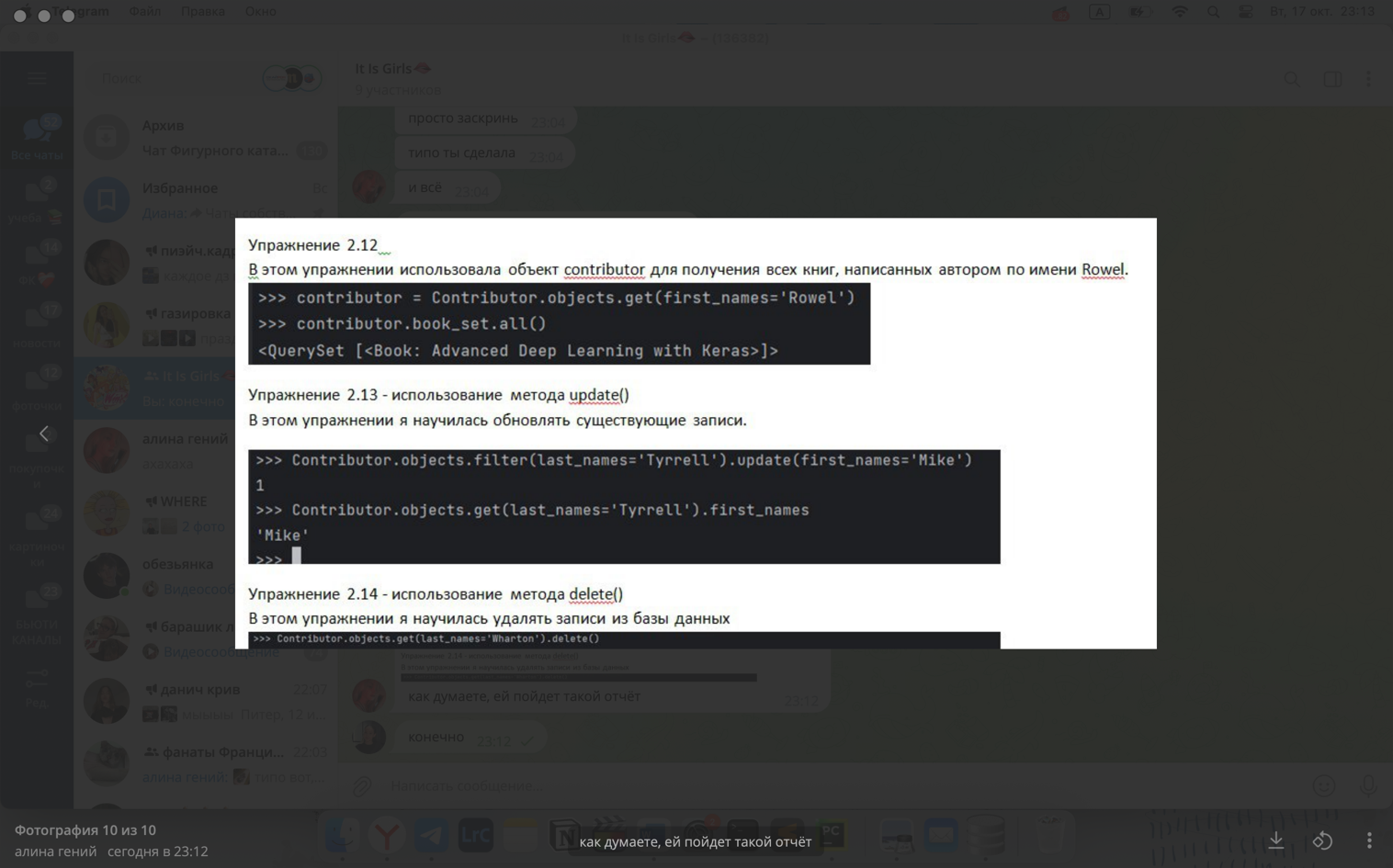
**Упражнение 2.12**

В этом упражнение я использовала contributor для получения всех книг, написанных Rowel.



**Упражнение 2.13**

В этом упражнение я познакомилась с методом update().

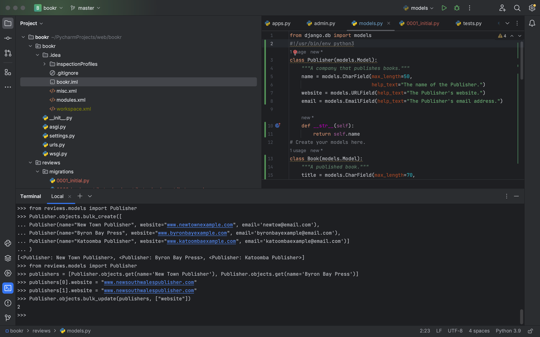


**Упражнение 2.14**

В этом упражнение я познакомилась с методом delete().

**Упражнение 2.15**

Здесь я научилась создавать несколько записей одновременно с помощью функции bulk\_create.



**Упражнение 2.16**

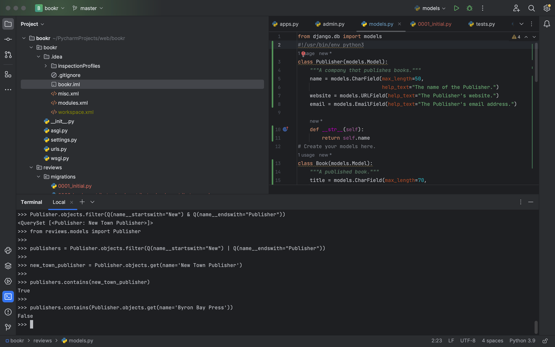
Здесь я делала то же самое, что и в 15 упражнение, только для нескольких записей

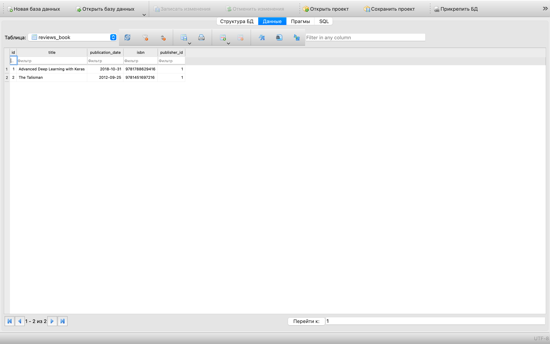
**Упражнение 2.17**

Здесь я научилась выполнять сложный запрос, с использованием Q.

**Упражнение 2.18**

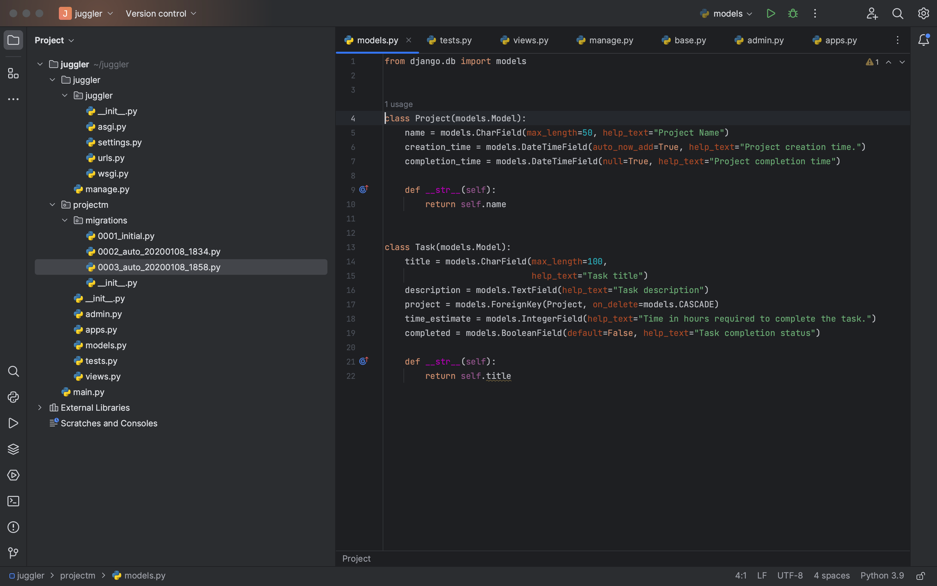
В этом упражнении я узнала, как с помощью метода contain() можно проверить, является ли объект частью набора запросов.





**Активити 2.01**

Здесь я создала проект Juggler, который отслеживает проекты. У меня возникли небольшие трудности с созданием баззы данных, мне выдавало ошибку, но вроде как все исправила.



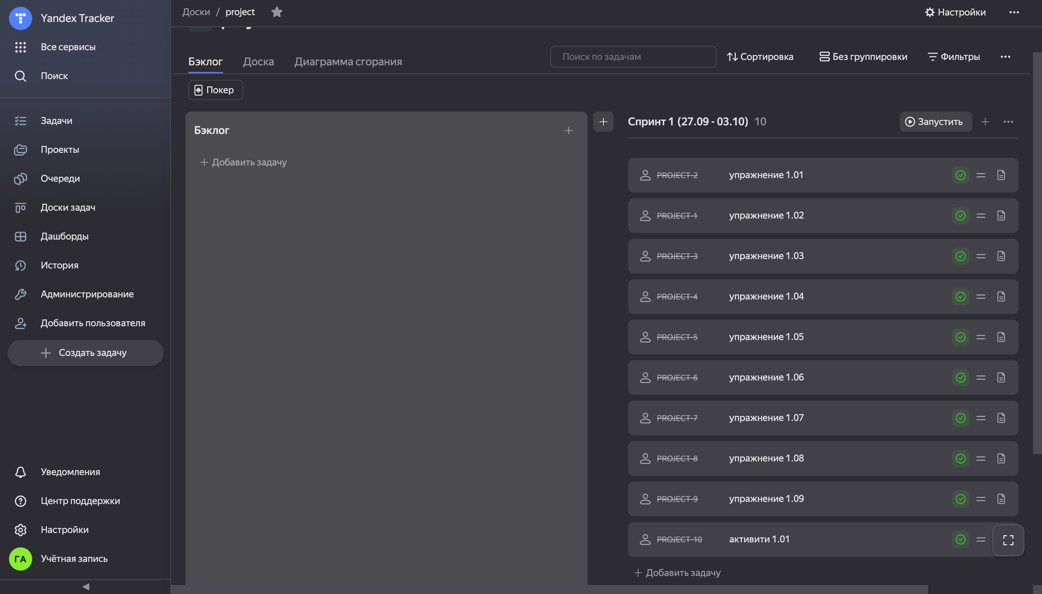
**Вывод :**

*У меня возникали трудности либо по моей невнимательности, либо из-за непонимания книги. Первое упражнение и последнее мне дались тяжелее всего. Я заранее извиняюсь, за то что так мало скринов, просто я делала не за один день и путалась, что я уже скринила, а что нет, поэтому у меня в библиотеке много повторяющихся фотографий заданий, а каких-то вообще нет. В следующий раз буду вести яндекс трекер немного по-другому, буду расписывать, что скринила, а что нет. Буду учиться на оишибках. Пожалуйста не снимайте за это баллы))*

**-Спринт 27.09-03.10**

Газизова Амина Альбертовна

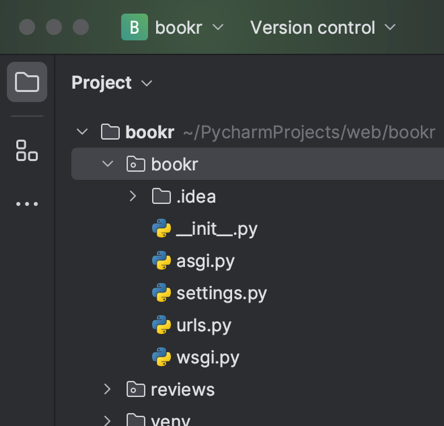
Группа 11-311

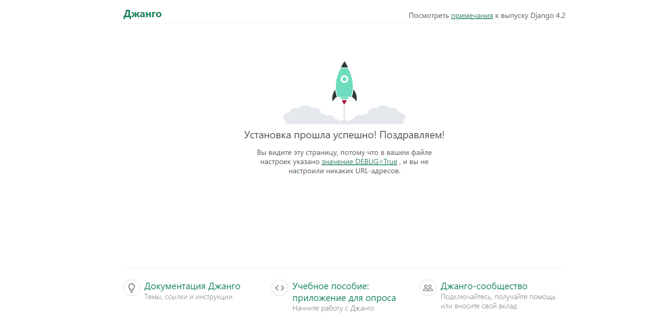


\*Скриншот с яндекс трекера

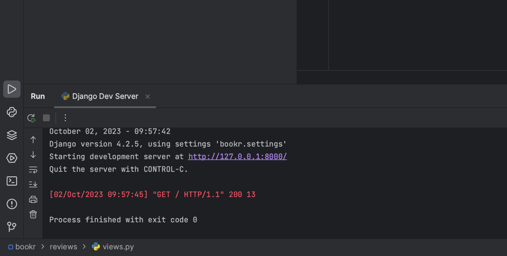
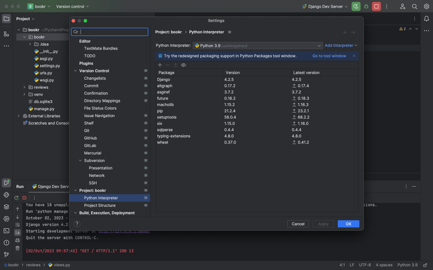
<https://github.com/aminagazizova/Bookr>

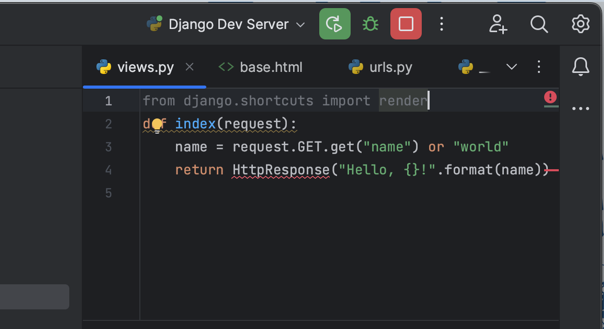
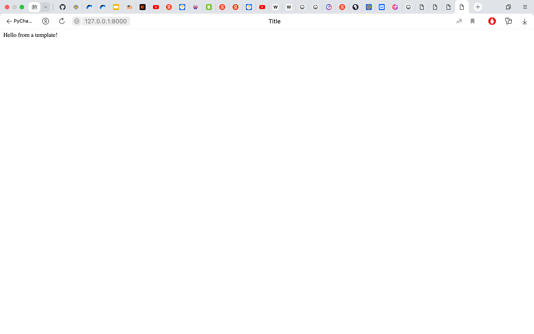
**1.01 УПРАЖНЕНИЕ**

****В этом упражнение я научилась создавать Django, проверила как оно работает и создала приложение 'reviews’



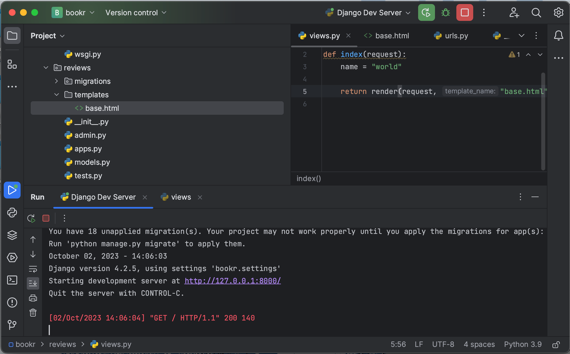
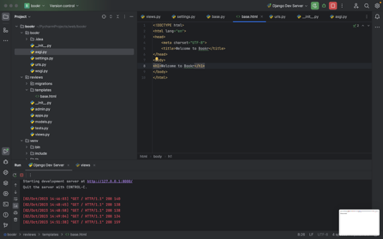
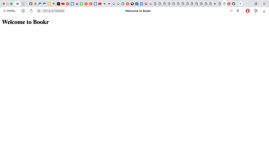
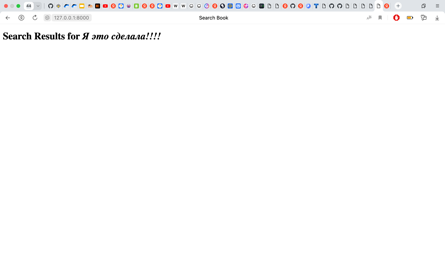
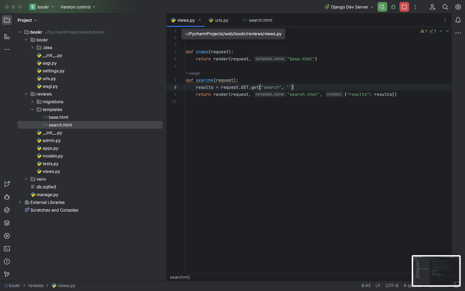
**1.02 упражнение**

****Так как у меня уже было все добавлено, мне оставалось только добавить конфигурацию

****

**Упражнение 1.07**

Здесь я смотрю как работает интерполяция переменных

****

**Вывод**

Все было несложно, трудности вызвало задание Активити 1.02

**Активити 1.02**

Здесь я создала базовую страничку Bookr

**Активити 1.01**

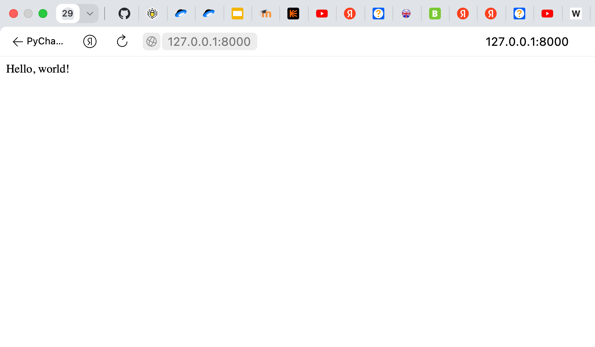
Здесь мы создали новую шапку приложения и экран приветствия

**Упражнение 1.09**

Здесь я запустила Django в отладчике, а затем остановила в функции view, для изучения переменных

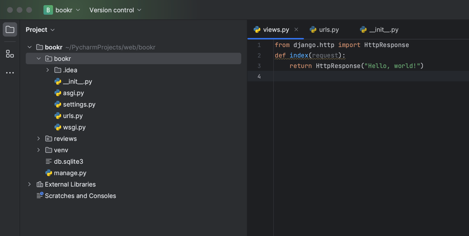
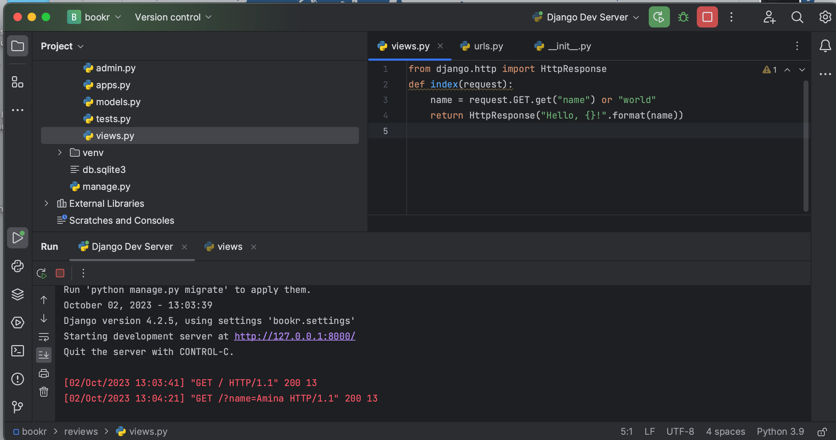
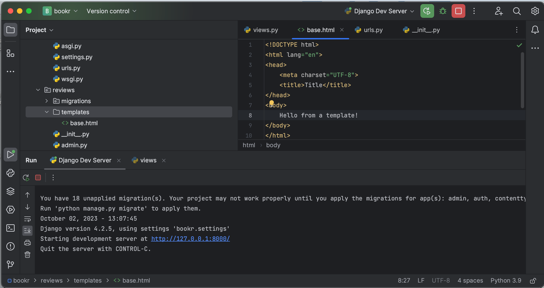
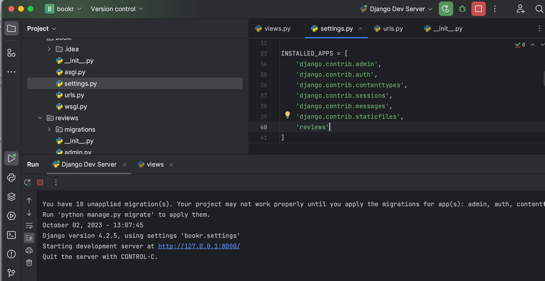
**Упражнение 1.08**

В этом упражнение я специально сделала ошибку в Django, вызвав пременную, не указанную в коде



**Упражнение 1.03**

Здесь я рассмотрела как написать представление и как добавить URL карту для перехода к представлению

****

**Упражнение 1.04**

В этом упражнении я научилась читать значения URL в атрибуте GET

**Упражнение 1.06**

В этом упражнении я обновила функцию index view для отображения HTML-шаблона

**Упражнение 1.05**

В этом упражнении я создала каталог шаблонов для приложения "*Reviews"*. Затем я добавила файл HTML-шаблона, который Django сможет отобразить в HTTP-ответ.