

Бронирование книг и отображение броней клиента

Задание 1

Реализовать view-функцию book_reserve

1. URL: /books/<int:book_id>/reserve/ (имя: book_reserve)

2. Логика:

- GET-запрос: Показывает форму бронирования (BookReservationForm).
- POST-запрос:

- Проверяет, доступна ли книга для брони (не забронирована другим читателем).
- Если книга недоступна – возвращает ошибку: *"Книга уже забронирована".
- Если доступна – создает бронь на **14 дней** от текущей даты.
- После успешного бронирования перенаправляет на список броней читателя.

3. Шаблон: reservations/create.html

- Форма с выбором читателя (если не передается автоматически).
- Кнопка "Забронировать".
- Вывод ошибок, если бронь невозможна.

Задание 2

Реализовать view-функцию и шаблон для бронирования книги и отображения списка броней конкретного читателя с визуальным выделением статусов и сортировкой.

Требования:

1. View-функция reader_reservations

- Принимает reader_id (ID читателя).
- Возвращает список броней, отсортированный:
 - **1-й приоритет:** Просроченные брони (дата окончания < текущей даты, статус active).
 - **2-й приоритет:** Активные брони (дата окончания ≥ текущей даты, статус active).
 - **3-й приоритет:** Закрытые брони (статус completed или canceled).
- Внутри каждой группы сортировка **от новых к старым** (по дате бронирования).

- Передает в контекст:
 - reservations – отсортированный QuerySet броней.
 - reader – объект читателя.
 - current_date – текущая дата (для сравнения в шаблоне).

2. Шаблон `reservations/reader_reservations.html`

- Таблица с колонками:
 - Книга (название + ссылка на детали книги).
 - Дата бронирования.
 - Дата окончания брони.
 - Статус.
- Стилизация строк:
 - Просроченные – красный фон (`style="background-color: #ffcccc;"`).
 - Активные (непросроченные) – стандартный фон.
 - Закрытые – желтый фон (`style="background-color: #ffffcc;"`).
- Если броней нет – вывести сообщение: `*"У читателя нет броней."*`

3. URL

- Путь: `/readers/<int:reader_id>/reservations/`.
- Имя: `reader_reservations`.

Сдача работы

1. Обновить репозиторий:

- Создать ветку `feature/reservations-list`.
- Коммиты должны начинаться с `"108."` (примеры ниже).

2. Формат коммитов:

- 108. Добавлена view-функция `reader_reservations`
- 108. Реализована сортировка броней по статусам
- 108. Добавлен шаблон с цветовой подсветкой
- 108. Обновлены URL для списка броней

3. Ссылка на PR: Отправить ссылку

Пример кода (подсказка)

View-функция (примерная логика)

```

from django.utils import timezone
from django.db.models import Case, When, Value, IntegerField

def reader_reservations(request, reader_id):
    reader = get_object_or_404(Reader, pk=reader_id)
    today = timezone.now().date()

    reservations = BookReservation.objects.filter(reader=reader).annotate(
        status_priority=Case(
            # Просроченные (активные, но дата окончания < сегодня)
            When(
                status='active',
                reservation_end__lt=today,
                then=Value(1)
            ),
            # Активные (непросроченные)
            When(
                status='active',
                reservation_end__gte=today,
                then=Value(2)
            ),
            # Закрытые
            When(
                status__in=['completed', 'canceled'],
                then=Value(3)
            ),
            output_field=IntegerField(),
        )
    ).order_by('status_priority', '-reservation_date')

    return render(request, 'reservations/reader_reservations.html', {
        'reservations': reservations,
        'reader': reader,
        'current_date': today,
    })

```

Шаблон (пример строки таблицы)

```

{% for reservation in reservations %}
<tr style="{% if reservation.status == 'active' and reservation.reservation_end
background-color: #ffcccc;
{% elif reservation.status in ['completed', 'canceled'] %}background
&lt;td&gt;{{ reservation.book.title }}&lt;/td&gt;
&lt;td&gt;{{ reservation.reservation_date }}&lt;/td&gt;
&lt;td&gt;{{ reservation.reservation_end }}&lt;/td&gt;
</pre>

```

```
<td>{{ reservation.get_status_display }}</td>
</tr>
{% empty %}
<tr><td colspan="4">У читателя нет броней.</td></tr>
{% endfor %}
```