

# Расширение библиотеки (Читатели и Бронирование книг)

## Часть 1: Модели

Задание: Спроектировать и добавить новые модели

### 1. Модель **Reader** (Читатель):

- Поля:
  - ФИО (CharField)
  - Дата рождения (DateField)
  - Адрес (TextField)
  - Номер телефона (CharField с валидацией)
  - Email (EmailField)
  - Дата регистрации (DateTimeField, auto\_now\_add)
- Методы:
  - `__str__` - должен возвращать ФИО
  - `get_active_reservations` - возвращает активные брони

### 2. Модель **BookReservation** (Бронь книги):

- Поля:
  - Книга (ForeignKey к Book)
  - Читатель (ForeignKey к Reader)
  - Дата бронирования (DateTimeField, auto\_now\_add)
  - Дата окончания брони (DateTimeField)
  - Статус (CharField с choices: 'active', 'completed', 'canceled')
- Методы:
  - `__str__` - должен возвращать строку "Бронь #id"
  - `is_active` - property, проверяет активна ли бронь

Требования к моделям:

- Добавить `related_name` для связей
- Реализовать методы моделей
- Добавить индексы для часто используемых полей

**Миграции:**

- После создания моделей выполните миграции

- Добавьте файлы миграций в репозиторий

## Часть 2: View-функции

### Задание 1: Управление читателями

#### 1. **reader\_list** - список всех читателей:

- GET-параметр `search` для поиска по ФИО (если ключа нет, показать всех читателей, если есть - показать список найденных)
- Шаблон: `readers/list.html`

#### 2. **reader\_create** - добавление нового читателя:

- Обработка GET (форма) и POST (сохранение)
- Использовать форму `ReaderForm`
- После сохранения - перенаправление на список читателей
- Шаблон: `readers/form.html`

#### 3. **Формы:**

- Создайте формы для `Reader` и `BookReservation`
- Добавьте валидацию данных

## Сдача работы

1. Обновить существующий репозиторий
2. Коммиты должны начинаться с "107." (например: "107. Добавлена модель Reader")

Пример структуры коммитов:

```
107. Добавлены модели Reader и BookReservation
107. Реализованы view-функции для читателей
107. Добавлены формы бронирования
107. Созданы шаблоны для управления читателями
107. Добавлены тесты для новых view-функций
```