Общая структура проекта

1. post — Управление публикациями (статьями) и комментариями.
2. userAcc — Регистрация новых пользователей.
3. userProfile — Управление профилями пользователей, включая расширенные права (модерация).

1. Приложение post (Публикации и комментарии)

Это ядро блога, отвечающее за основной контент.

Модели:

- Post: Модель для статей блога.
    - Связана с автором через ForeignKey на стандартную модель User.
    - Содержит заголовок, текст и дату создания.
- Comment: Модель для комментариев к статьям.
    - Связана со статьей (Post) и автором (User) через ForeignKey.

Формы:

- PostCreateForm: Форма для создания и редактирования статей.
- CommentForm: Форма для добавления комментариев.

Представления (Views) и их функционал:

- post_list: Отображает список всех статей на главной странице, отсортированных от новых к старым.
- post_detail: Показывает полный текст одной статьи, все связанные с ней комментарии и форму для добавления нового комментария.
- create_post (@login_required): Позволяет аутентифицированному пользователю создать новую статью.
- edit_post (@login_required): Позволяет автору статьи редактировать свой пост.
- delete_post (@login_required): Позволяет автору статьи удалить свой пост.
- delete_any_post (@user_passes_test): Позволяет модератору удалить **любую** статью.
- delete_any_comment (@user_passes_test): Позволяет модератору удалить **любой** комментарий.

Связи:

- **С** userAcc: Для создания поста или комментария пользователь должен быть зарегистрирован (через userAcc) и аутентифицирован.
- **С** userProfile: Проверка прав модератора (user.profile.is_moderator) для использования функций delete_any_post и delete_any_comment.

2. Приложение userAcc (Учетные записи)

Отвечает за первоначальную регистрацию пользователей в системе.

Модели:

- Использует стандартную модель Django User.

Формы:

- Registration: Кастомная форма регистрации, основанная на UserCreationForm. Добавляет обязательное поле email.

Представления (Views) и их функционал:

- register: Обрабатывает регистрацию нового пользователя. После успешной регистрации перенаправляет на страницу входа и показывает сообщение об успехе.

Связи:

- **С** userProfile: При сохранении нового пользователя (сигнал post_save), автоматически создается связанный с ним объект Profile.

3. Приложение userProfile (Профили пользователей)

Расширяет функционал учетной записи, добавляя профиль с дополнительной информацией и правами.

Модели:

- Profile: Связана с моделью User через OneToOneField.
    - Добавляет биографию (bio), аватар (avatar), а также флаги is_blocked (заблокирован) и is_moderator (является модератором).

Сигналы:

- signals.py: Автоматически создает объект Profile для каждого нового пользователя.

Формы:

- UserUpdateForm: Форма для редактирования стандартных данных пользователя (имя, email).
- ProfileUpdateForm: Форма для редактирования данных профиля (био, аватар).

Представления (Views) и их функционал:

- profile_view (@login_required): Показывает профиль текущего пользователя.
- profile_edit (@login_required): Позволяет пользователю редактировать данные своего профиля и учетной записи.
- toggle_block_user (@user_passes_test): Позволяет модератору или суперпользователю заблокировать или разблокировать любого пользователя..
- delete_profile (@login_required): Позволяет пользователю полностью удалить свой аккаунт.

Админ-панель:

- admin.py: Настраивает админку так, что модель Profile отображается "внутри" страницы редактирования User. Также добавляет колонки is_moderator и is_blocked в список пользователей.

Связи:

- **С** post: Предоставляет механизм проверки прав модератора для глобального управления контентом.
- **С** userAcc: Непосредственно расширяет функционал созданных учетных записей.

Сводная таблица связей и прав

Действие	Кто может выполнять	Связь между приложениями
Регистрация	Анонимный пользователь	userAcc -> (создает User) -> userProfile (сигнал создает Profile)

Создание поста	Аутентифицированный пользователь	post relies on userAcc
Редактирование/ удаление своего поста	Автор поста	post (проверка author=request.user)
Комментирование	Аутентифицированный пользователь	post relies on userAcc
Удаление любого поста/комментария	Модератор (is_moderator=True)	post relies on userProfile (проверка u.profile.is_moderator)
Просмотр/ редактирование профиля	Владелец профиля	userProfile
Блокировка пользователя	Модератор или Суперпользователь	userProfile (использует свои же флаги профиля)

=классический блог с разделением прав. Обычные пользователи могут вести свой блог, а модераторы — контролировать весь контент и пользователей в системе. Все приложения тесно связаны через модель пользователя и ее расширение Profile.