

Короче есть хуйня на нем пользователи у него есть возможность чекать ленту смотреть профиль, менять фотку на профиле, писать другим пользователям. Пользователь может создавать посты а туда может помещать видео, добавлять описание в пости, а другие пользователи могут лайкать и комментировать. Платформа для нахождения репетитора где пользователя которые работают репетитором выкладывают примеры своих обучающих роликов и люди ставят отзывы и лайки и говорят про флекс.

Микросервисная архитектура бекэнд - Джанго, фронт - Реакт, вспом - Редис и Кафка, дб - постгресс.

## **1. \*\*Сервис пользователей (User Service)\*\***

**\*\*Функции:\*\***

- Просмотр профилей спецов.
- Обновление информации в профиле, в том числе загрузка и изменение аватаров у спеца.
- Вход и регистрация для спецов и клиентов.
- Действующие спецы

**\*\*Хранилище:\*\***

- Постгресс

## **2. \*\*Сервис постов (Post Service)\*\***

**\*\*Функции:\*\***

Создание задачи(с указаниями требований к репетитору)и выдача подходящих репетиторов по категориям задачи.

**\*\*Хранилище:\*\*** Постгресс

## **4. \*\*Чат (Interaction Service)\*\***

**\*\*Функции:\*\***

Чат поддержки

Чат с репетитором

**\*\*Хранилище:\*\***

- Постгресс

## 6. **\*\*Сервис видео (Video Service)\*\***

**\*\*Функции:\*\***

Сохранение видео после изменения профиля

**\*\*Хранилище:\*\***

-S3

## 7. **\*\*Сервис отзывов (Review Service)\*\***

**\*\*Функции:\*\***

Все отзывы текущего репетитора

Лучшие отзывы (главную станицу)

Создание отзыва

**\*\*Хранилище:\*\*** s3

### **микросервисы:**

- Сервис пользователей
- Сервис постов
- Сервис чата
- Сервис видео
- Сервис отзывов

### ### 1. \*\*Сервис пользователей (User Service)\*\*

**\*\*Приложение:\*\*** `users`

**\*\*Модели:\*\***

- **\*\*User (Пользователь)\*\***

- Поля: `username`, `email`, `password`, `role` (репетитор/ученик),  
`profile\_picture`, `bio`, `created\_at`, `updated\_at`

- Описание: Основная модель для хранения информации о пользователе.

- **\*\*Tutor (Репетитор)\*\***

- Поля: `user` (ForeignKey на User), `subject`, `experience`, `rating`,  
`created\_at`, `updated\_at`

- Описание: Модель для хранения дополнительной информации о репетиторах (предметы, стаж, рейтинг).

- **Session (Сессия)**

- Поля: `user` (ForeignKey на User), `token`, `created\_at`, `expires\_at`
  - Описание: Модель для хранения активных сессий пользователей, чтобы обрабатывать их авторизацию.

- **ProfileSettings (Настройки профиля)**

- Поля: `user` (ForeignKey на User), `email\_notifications`, `privacy\_settings`
  - Описание: Модель для хранения настроек профиля пользователя.

**Функциональность:**

- Регистрация/аутентификация пользователей.
- Роли пользователей (репетитор или ученик).
- Изменение профиля (аватар, биография и т.д.).
- Управление сессиями.

---

## ### 2. Сервис постов (Post Service)

**Приложение:** `posts`

**Модели:**

- **Post (Пост)**

- Поля: `author` (ForeignKey на User), `content` (текст поста), `video\_url`, `created\_at`, `updated\_at`
  - Описание: Модель для хранения информации о постах, включая ссылки на видео и текстовое описание.

- **Video (Видео)**

- Поля: `post` (ForeignKey на Post), `video\_file`, `duration`, `created\_at`

- Описание: Отдельная модель для видеофайлов, связанных с постами.

- **Tag (Тэг)**

- Поля: `name`

- Описание: Модель для хранения тегов, которые можно прикреплять к постам.

- **PostTag (Тэг для поста)**

- Поля: `post` (ForeignKey на Post), `tag` (ForeignKey на Tag)

- Описание: Промежуточная модель для связи постов с тегами.

**Функциональность:**

- Создание и редактирование постов.

- Загрузка видео и текстового описания.

- Возможность добавления тегов.

---

### ### 3. Сервис ленты (Feed Service)

**Приложение:** `feed`

**Модели:**

- **FeedItem (Элемент ленты)**

- Поля: `post` (ForeignKey на Post), `user` (ForeignKey на User), `created\_at`

- Описание: Модель для хранения записей о том, какие посты отображаются в ленте пользователя.

- **Recommendation (Рекомендация)**

- Поля: `user` (ForeignKey на User), `post` (ForeignKey на Post), `reason` (текстовое описание причины рекомендации), `created\_at`

- Описание: Хранит рекомендованные посты для пользователя, основанные на его действиях.

**\*\*Функциональность:\*\***

- Формирование ленты для каждого пользователя (сортировка постов по времени, популярности).
- Механизм рекомендаций постов (на основе активности или тегов).

---

#### ### 4. **\*\*Сервис лайков и комментариев (Interaction Service)\*\***

**\*\*Приложение:\*\*** `interactions`

**\*\*Модели:\*\***

- **\*\*Like (Лайк)\*\***

- Поля: `user` (ForeignKey на User), `post` (ForeignKey на Post), `created\_at`
- Описание: Модель для хранения лайков пользователей на постах.

- **\*\*Comment (Комментарий)\*\***

- Поля: `author` (ForeignKey на User), `post` (ForeignKey на Post), `content` (текст комментария), `parent\_comment` (ForeignKey на Comment, опционально), `created\_at`, `updated\_at`
- Описание: Модель для хранения комментариев к постам. Поддержка вложенных комментариев (ответов).

- **\*\*CommentLike (Лайк на комментарий)\*\***

- Поля: `user` (ForeignKey на User), `comment` (ForeignKey на Comment), `created\_at`
- Описание: Лайки для комментариев.

**\*\*Функциональность:\*\***

- Лайки на посты и комментарии.

- Комментирование постов и ответов на комментарии.

---

### ### 5. \*\*Сервис уведомлений (Notification Service)\*\*

**\*\*Приложение:\*\*** `notifications`

**\*\*Модели:\*\***

- **\*\*Notification (Уведомление)\*\***

- Поля: `recipient` (ForeignKey на User), `content` (текст уведомления), `is\_read`, `created\_at`
- Описание: Модель для хранения уведомлений, отправленных пользователям (о новых лайках, комментариях и т.д.).

**\*\*Функциональность:\*\***

- Отправка уведомлений (пуш-уведомления, e-mail).
- Система уведомлений о действиях других пользователей (лайки, комментарии, новые посты).

---

### ### 6. \*\*Сервис видео (Video Service)\*\*

**\*\*Приложение:\*\*** `videos`

**\*\*Модели:\*\***

- **\*\*Video (Видео)\*\***

- Поля: `uploader` (ForeignKey на User), `video\_file`, `duration`, `format`, `created\_at`
- Описание: Основная модель для хранения и управления видеофайлами.



- **VideoConversionTask (Задача конвертации видео)**
  - Поля: `video` (ForeignKey на Video), `status`, `created\_at`, `updated\_at`
  - Описание: Модель для отслеживания задач конвертации видео.

**Функциональность:**

- Хранение и управление видеофайлами (например, конвертация в нужные форматы).
- Интеграция с CDN для быстрой доставки видео.

---

## ### 7. **Сервис отзывов (Review Service)**

**Приложение:** `reviews`

**Модели:**

- **Review (Отзыв)**
  - Поля: `author` (ForeignKey на User), `tutor` (ForeignKey на Tutor), `rating`, `content`, `created\_at`, `updated\_at`
  - Описание: Модель для хранения отзывов и рейтингов для репетиторов.
- **TutorRating (Рейтинг репетитора)**
  - Поля: `tutor` (ForeignKey на Tutor), `average\_rating`, `review\_count`
  - Описание: Модель для хранения агрегированных данных о репетиторе (средний рейтинг и количество отзывов).

**Функциональность:**

- Оставление отзывов о репетиторах.
- Подсчёт среднего рейтинга для каждого репетитора.

---

### ### 8. \*\*Сервис поиска (Search Service)\*\*

**\*\*Приложение:\*\*** `search`

**\*\*Модели:\*\***

- Для этого сервиса скорее всего не нужны традиционные Django модели, так как для поиска будет использоваться Elasticsearch или аналогичные решения для полнотекстового поиска.

**\*\*Функциональность:\*\***

- Полнотекстовый поиск по постам, репетиторам и тегам.
- Поиск по рейтингу и отзывам репетиторов.

---

### Примерные отношения между микросервисами:

- **\*\*Пользователь (User)\*\*** будет использоваться в большинстве сервисов, таких как `posts`, `reviews`, `comments` и `interactions`.
- **\*\*Сервис уведомлений (Notification)\*\*** будет взаимодействовать с сервисами комментариев, лайков и постов для генерации уведомлений.
- **\*\*Видео (Video Service)\*\*** будет интегрирован с сервисом постов для хранения и обработки видеоконтента.

Таким образом, вся система состоит из отдельных приложений и моделей, где каждое приложение выполняет свою специфическую задачу.

## Основные возможности платформы:

### 1. Пользовательский функционал:

- **Регистрация и авторизация:** Пользователи могут создавать аккаунты, входить в систему и редактировать свой профиль.
- **Просмотр ленты:** Пользователи могут просматривать ленту с публикациями репетиторов.
- **Просмотр профиля:** Возможность просмотра профилей других пользователей, включая их публикации и учебные материалы.
- **Редактирование профиля:** Пользователи могут загружать или менять фотографии профиля.
- **Создание постов:** Репетиторы могут создавать посты с учебными видео, добавлять описание и примеры учебных материалов.
- **Интерактивность:** Пользователи могут лайкать и комментировать посты, оставлять отзывы на контент, отмечать понравившиеся материалы.

### 2. Функции админа

=====