Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Московский политехнический университет»

Образовательная программа «Веб-технологии»

Направление подготовки: 09.01.01 «Информатика и вычислительная техника»

Техническое задание

по дисциплине «Веб-разработка»

**Выполнил:**

Студент группы 231-323

Жетписов А.Н.

24.08.2025

**Принял:**

Старший преподаватель кафедры ИКТ

Даньшина М.В.

Старший преподаватель кафедры ИКТ

Голубева И.В.

## 1. Введение

**Цель проекта.** Создать веб-приложение формата QA (аналог StackOverflow/Quora), в котором пользователи публикуют темы или вопросы, оставляют сообщения или ответы, комментарии и оценки.

**Задачи:**

1. Реализовать роли и права.
2. Описать и внедрить валидации бизнес-логики.
3. Оптимизировать запросы (select\_related / prefetch\_related).
4. Настроить сериализаторы (SerializerMethodField, передача контекста).
5. Добавить аннотации (вычисляемые поля).
6. Сделать фильтрацию (django-filter).
7. Подготовить Postman-коллекцию для тестирования API.

## 2. Роли и права доступа

**Роли:**

* **Пользователь (авторизованный).**  
  Минимум 3 функции (обязательные):
  1. Создание темы (вопроса) в выбранной категории.
  2. Создание сообщений (ответов) в теме.
  3. Редактирование и удаление **только собственных** тем и сообщений.  
     Дополнительно: комментирование, выставление оценок (+1/–1), редактирование профиля (аватар, сайт, описание).
* **Администратор.**  
  Минимум 3 функции (обязательные):
  1. Управление справочниками и контентом: создание/редактирование/удаление категорий и тем; модерация сообщений и комментариев.
  2. Управление пользователями: просмотр списка, блокировка/разблокировка.
  3. Доступ к базовой аналитике: самые активные темы/пользователи, отчёты.

**Общее правило прав:**  
– Редактировать/удалять объект может либо его автор, либо администратор.  
– Голосовать за собственные записи запрещено.

## 3. Основные функции сайта

* **Категории и темы:** просмотр списка категорий; создание и просмотр тем; поиск по ключевым словам.
* **Темы и сообщения:** создание, редактирование, удаление (с ограничением прав); просмотр и сортировки; счётчики активности.
* **Комментарии и рейтинги:** добавление комментариев к темам/сообщениям; голосование (+1/–1); агрегированные метрики.
* **Профиль:** просмотр и редактирование личных данных (аватар — ImageField, домашняя страница — URLField).
* **Поиск и фильтрация:** фильтры по категории, тегу, автору, дате, рейтингу; текстовый поиск (\_\_icontains).
* **Главная страница:** виджеты «Горячее» (по активности за период), «Новое», «Топ пользователи».

## 4. Требования к валидации

1. **Уникальность темы в пределах категории.**  
   При создании/редактировании тема с таким же названием в той же категории запрещена (валидатор на модель/сериализатор; уникальный (category, slug) или проверка по title с нормализацией).
2. **Права на редактирование/удаление.**  
   Только автор записи или админ может изменять/удалять тему/сообщение (permissions во viewset’ах/проверка в сервисном слое).
3. **Фильтрация запрещённых слов.**  
   В заголовках/тексте тем и сообщений нельзя использовать слова из стоп-списка (валидатор на сериализаторе/модели).
4. **Корректность контактной информации.**  
   email — форматная проверка при регистрации/редактировании; homepage — URLField с валидацией URL.

## 5. Оптимизация запросов (select\_related / prefetch\_related)

**Точки применения (минимум):**

* **Список тем:** select\_related('category', 'author') + prefetch\_related('tags').
* **Список сообщений:** select\_related('author', 'topic').
* **Комментарии:** select\_related('author') и при необходимости привязка к topic/post.
* **Главная страница (виджеты):** выборки «горячих»/«новых» — с нужными связями, чтобы избежать N+1.

## 6. Сериализаторы (DRF)

**Обязательные элементы:**

* **SerializerMethodField** — например, is\_editable (возвращает True, если текущий пользователь — автор или админ), preview (обрезка текста).
* **Передача контекста (context)** — передавать request в сериализатор (используется в is\_editable, условных полях, форматировании дат).
* Отдельные сериализаторы: список/деталь для тем; сериализатор сообщений; сериализатор профиля.

## 7. Аннотации (вычисляемые поля)

1. **Количество сообщений у темы** — posts\_count = Count('posts').
2. **Средний рейтинг темы** — avg\_rating = Avg('posts\_\_rating') или агрегирование по голосам.  
   Дополнительно: **последняя активность** (Max(created\_at/updated\_at) по постам); **суммарная активность за период**.

## 8. Фильтрация (django-filter)

**Критерии фильтрации:**

* По категории (category/category\_slug).
* По автору (author\_id/author\_username).
* По тегам (tag/несколько тегов).
* По периоду создания (created\_from, created\_to).
* По рейтингу (rating\_gte, rating\_lte).
* По тексту заголовка/контента (title\_\_icontains, body\_\_icontains).  
  Сортировка: ordering=-rating,-created\_at и т.п.  
  Пагинация — стандарт DRF.

## 9. API (обзор)

**Аутентификация:** JWT (simplejwt) — register, login, refresh, me.  
**Ресурсы (v1):**

* /categories — список/деталь, (CRUD под админом).
* /topics — список/создание; /topics/{id} — деталь/редактирование/удаление (автор/админ).
* /topics/{id}/posts — список/создание; /posts/{id} — деталь/редактирование/удаление.
* /topics/{id}/vote и /posts/{id}/vote — голосование (+1/–1) с проверками.
* /topics/{id}/comments, /posts/{id}/comments — комментарии.
* /profile/me — просмотр/редактирование профиля (аватар — ImageField, сайт — URLField).  
  Все списки — с фильтрами и оптимизациями из пп. 5–8.

## 10. Нефункциональные требования

* **Локализация админки:** LANGUAGE\_CODE='ru'.
* **Часовой пояс:** TIME\_ZONE='Europe/Moscow' (или по месту защиты).
* **Безопасность:** только автор/админ меняют записи; проверка типов/длин; запрет небезопасного HTML в текстах; базовая защита от массовых присваиваний.
* **Файлы и медиа:** хранение в MEDIA\_ROOT; ImageField/FileField для аватаров/вложений.
* **Производительность:** применение select\_related/prefetch\_related; индексы по FK и часто фильтруемым полям (category, author, created\_at, slug).
* **Логи и отладка:** стандартное логирование Django; опционально Silk/Sentry.
* **Сборка/запуск:** Docker Compose, .env для секретов.

## 11. Данные и наполнение

* Стартовые фикстуры: категории, теги, несколько пользователей, темы с постами и комментариями (для демонстрации).
* В каждой ключевой таблице — по 10+ записей для виджетов и фильтров.

## 12. Postman-коллекция

Подготовить QueStudio.postman\_collection.json со следующими группами:

* **Auth:** register, login, refresh, me.
* **Categories:** list, detail (CRUD — отдельно под админ-токеном).
* **Topics:** list (+фильтры/поиск), create, detail, update, delete, vote.
* **Posts:** list/create, detail/update/delete, vote.
* **Comments:** для тем и постов.  
  Каждый запрос — с примером успешного ответа и, где уместно, примерами ошибок (валидации/доступа).

## 13. Критерии приёмки

1. Реализованы функции ролей (минимум по 3 на роль) с корректными правами.
2. Срабатывают валидации из раздела 4 (минимум 3 шт. обязательно).
3. Списки и детали используют оптимизированные выборки (select\_related/prefetch\_related).
4. Сериализаторы содержат минимум один SerializerMethodField и используют context (request).
5. В API присутствуют поля/сортировки, основанные на аннотациях (Count, Avg, Max).
6. Фильтрация работает согласно разделу 8; параметры проверены в Postman.
7. Postman-коллекция импортируется и отрабатывает на локальном окружении.
8. Проект поднимается в Docker; админка на русском; загрузка/отдача медиа — исправна.

## 14. Карта соответствия «требование → где выполнено»

* **(1) Бизнес-логика, роли, ТЗ:** Разделы 2–4, 9.
* **(2) Валидации:** Раздел 4 (уникальность темы, права автора/админа, стоп-слова, email/URL).
* **(3) select\_related:** Раздел 5 (темы, сообщения, комментарии, главная).
* **(4) Сериализаторы:** Раздел 6 (SMF + context).
* **(5) Аннотации:** Раздел 7 (posts\_count, avg\_rating, last\_activity\_at).
* **(6) Filterset:** Раздел 8 (категория/теги/автор/дата/рейтинг/поиск).
* **(7) Postman:** Раздел 12 (структура коллекции).

## 15. Глоссарий

* **Тема (Topic)** — вопрос/дискуссия в категории.
* **Сообщение (Post)** — ответ внутри темы.
* **Комментарий (Comment)** — реплика к теме или сообщению.
* **Оценка (Vote/Rating)** — голос пользователя за тему/сообщение.
* **Аннотация** — вычисляемое поле в SQL (COUNT/AVG/MAX и т.д.).
* **Фильтр (django-filter)** — декларативная фильтрация QuerySet по параметрам запроса.