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»

Отчет по курсовой работе

по дисциплине «Веб-разработка»

Тема: «Сайт с вопросами и ответами»

Выполнил:

Студент группы 231-323

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Жетписов А.Н.

подпись, дата

Принял:

Старший преподаватель кафедры ИКТ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Даньшина М.В.

подпись, дата

Старший преподаватель кафедры ИКТ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Голубева И.В.

подпись, дата

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#__RefHeading___Toc742_4079733222)

[Глава 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 4](#__RefHeading___Toc744_4079733222)

[1.1 Анализ предметной области](#__RefHeading___Toc1789_4079733222) 4

[1.2 Анализ аналогов 4](#__RefHeading___Toc1791_4079733222)

[1.3 Анализ целевой аудитории](#__RefHeading___Toc1793_4079733222) 5

[1.4. Анализ функциональности проекта](#__RefHeading___Toc1795_4079733222) 5

[1.5 Структура базы данных](#__RefHeading___Toc1797_4079733222) 5

[Глава 2. ПРАКТИЧЕСКАЯ ЧАСТЬ](#__RefHeading___Toc746_4079733222) 7

[2.1. Административная часть](#__RefHeading___Toc1373_2528408596) 7

[2.2. Главная страница](#__RefHeading___Toc1803_4079733222) 7

[2.3. Детальная страница](#__RefHeading___Toc1805_4079733222) 7

[2.4. Работа с базой данных](#__RefHeading___Toc1807_4079733222) 8

[2.5 Оптимизация запросов](#__RefHeading___Toc1375_2528408596) 8

[ЗАКЛЮЧЕНИЕ](#__RefHeading___Toc748_4079733222) 9

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ](#__RefHeading___Toc750_4079733222) 10

# ВВЕДЕНИЕ

Современные профессиональные сообщества требуют удобных инструментов для обмена знаниями. Одним из популярных форматов являются веб-сервисы вопросов и ответов, где пользователи могут задавать вопросы и получать ответы от других участников. Подобные платформы позволяют формировать знания и ускорять процесс обучения.

Цель проекта: разработка веб-приложения «QueStudio» - сайта с вопросами и ответами, реализованного на фреймворке Django (backend) и Vue.js (frontend).

Задачи проекта:

1. Проанализировать предметную область и существующие аналоги
2. Спроектировать структуру базы данных с использованием связей 1:N и M:N
3. Реализовать административную часть с возможностью работы с сущностями
4. Создать главную страницу с основными виджетами («горячее», «новое», «топ пользователи») и поиском
5. Разработать детальную страницу темы с отображением ответов, комментариев, голосования и CRUD-операциями
6. Реализовать систему аутентификации пользователей и разграничение прав
7. Наполнить базу тестовыми данными для демонстрации

Ссылка на репозиторий: [sxd0/QueStudio](https://github.com/sxd0/QueStudio)

# Глава 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## 1.1 Анализ предметной области

Сервисы вопросов и ответов предназначены для организации взаимодействия пользователей в формате «вопрос - ответ». Пользователь публикует тему (вопрос), другие пользователи создают ответы и комментарии, возможна система голосования, которая определяет рейтинг.

Ключевые элементы подобных систем:

* Категории - позволяют структурировать вопросы по тематике.
* Теги - создают систему классификации.
* Рейтинг - обеспечивает обратную связь и стимулирует активность пользователей.
* Комментарии - дают возможность уточнить или дополнить ответы.

## 1.2 Анализ аналогов

StackOverflow – крупнейший англоязычный сайт для программистов. Отличается строгой модерацией, высокой культурой качества вопросов и ответов.  
Quora – более открытая платформа для вопросов на любые темы, делает акцент на социальные аспекты и обсуждения.

Сильные стороны:

* широкий охват аудитории;
* развитая система тегов и поиска;
* репутация и рейтинги.

Недостатки:

* высокая сложность для новичков (StackOverflow),
* низкая структурированность и шумность (Quora).

QueStudio решает учебные задачи: реализован базовый функционал, позволяющий студентам осваивать необходимый спектр тем.

## 1.3 Анализ целевой аудитории

Основными пользователями проекта выступают:

* студенты и начинающие разработчики, которым важно задавать вопросы и быстро получать ответы;
* преподаватели и наставники, которым необходима площадка для обсуждения заданий.

## 1.4. Функциональные требования проекта

* Регистрация и аутентификация (JWT).
* CRUD-операции для тем, постов и комментариев.
* Система голосования за темы и ответы.
* Фильтры и поиск (django-filter).
* Административная панель с расширенными возможностями.
* Главная страница с динамическими виджетами.
* Детальная страница с редактированием и удалением материалов.

## 1.5. Структура базы данных

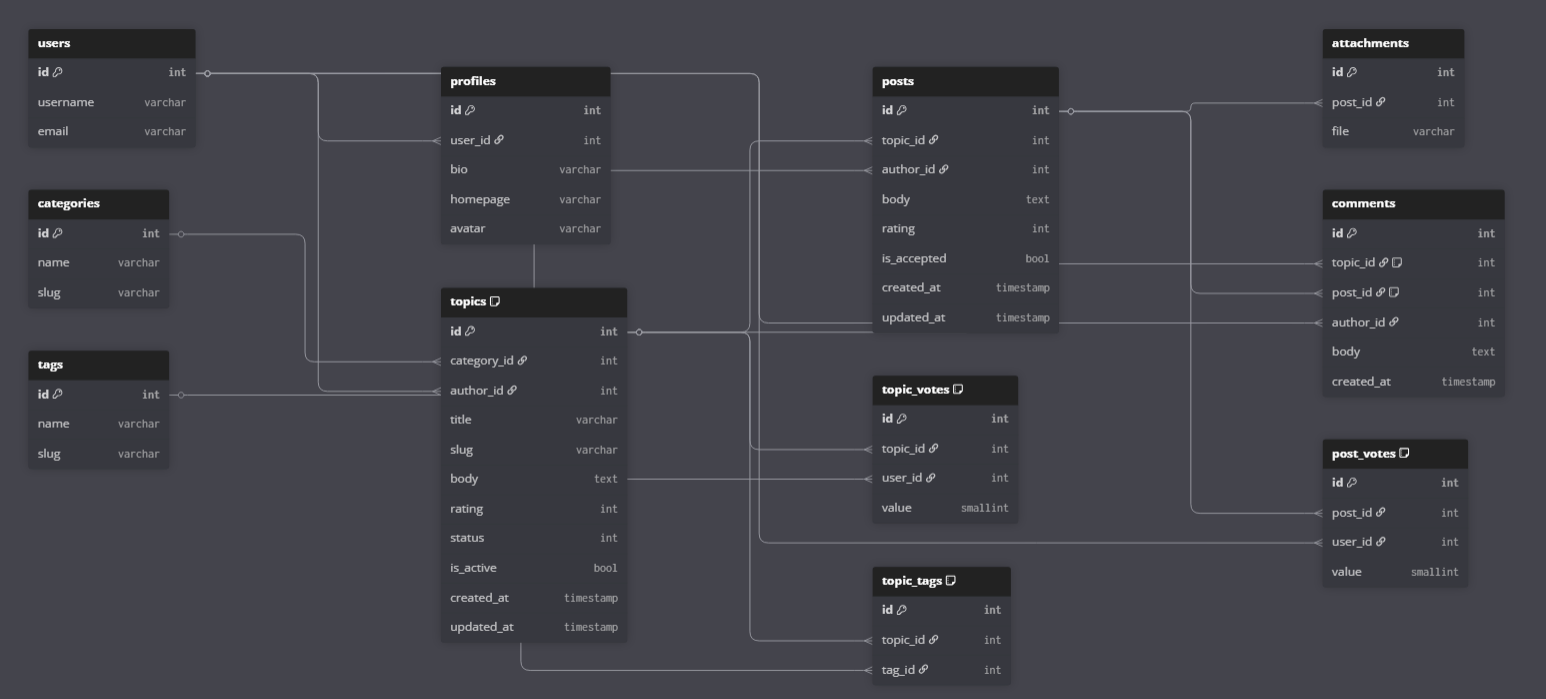
В проекте используются следующие основные сущности:

* User - пользователь системы;
* Category - категория вопросов;
* Topic - тема (вопрос);
* Tag - тег;
* TopicTag - связь M:N между темами и тегами;
* Post - ответ;
* Comment - комментарий.

Связи:

* User ↔ Topic (1:N);
* Topic ↔ Post (1:N);
* Post ↔ Comment (1:N);
* Topic ↔ Tag (M:N через TopicTag).

Дополнительно в системе реализованы служебные сущности:  
• Profile - профиль пользователя (1:1 к User, содержит аватар, биографию, URL).  
• Attachment - вложения к постам (1:N).  
• TopicVote **/** PostVote - голоса за темы и посты, обеспечивают рейтинг и уникальность голосов (M:N через отдельные таблицы).  
Эти таблицы расширяют функциональность, но в минимальном варианте проекта могли бы быть опущены.



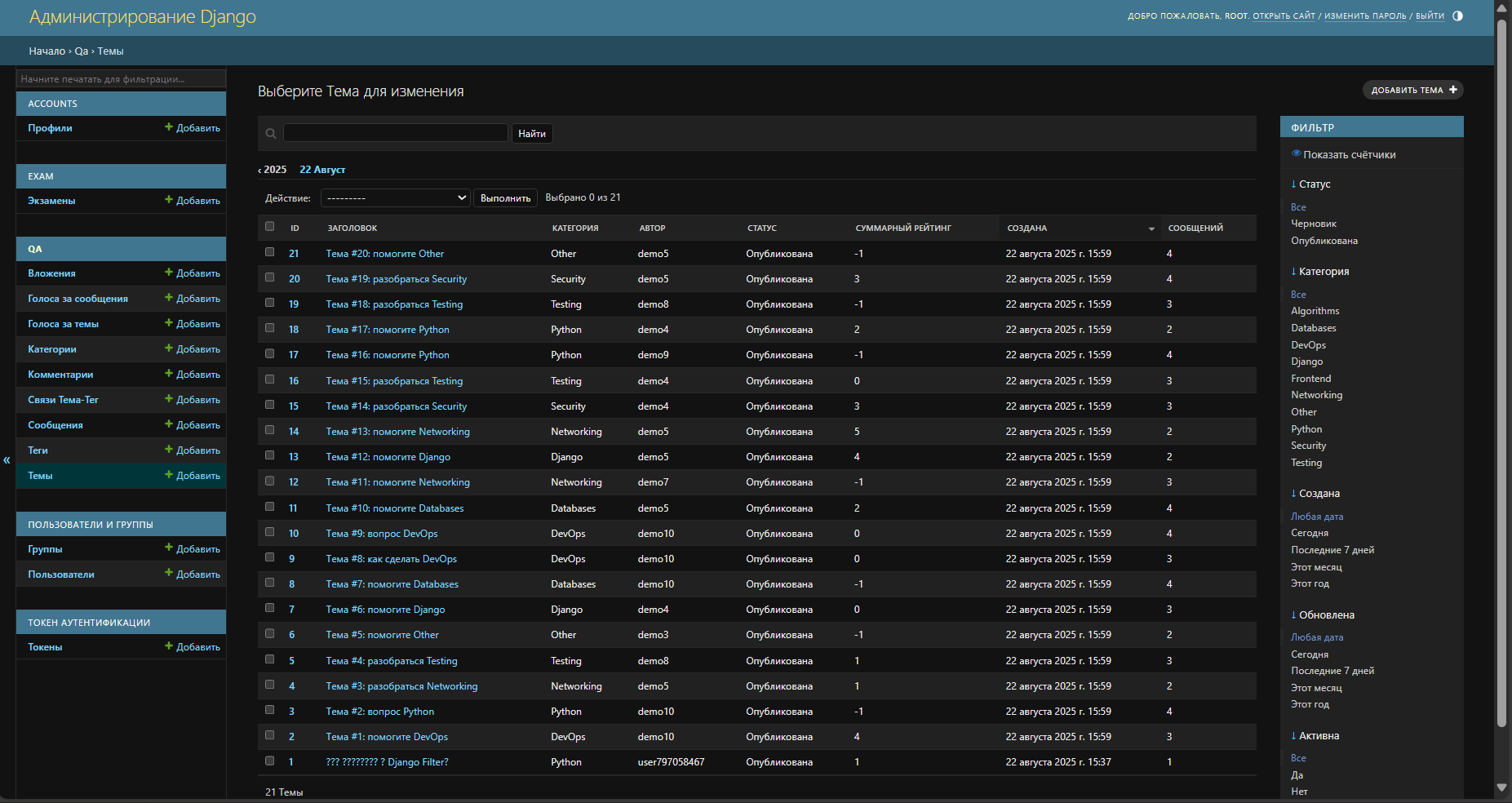
ER - диаграмма 1

# Глава 2. ПРАКТИЧЕСКАЯ ЧАСТЬ

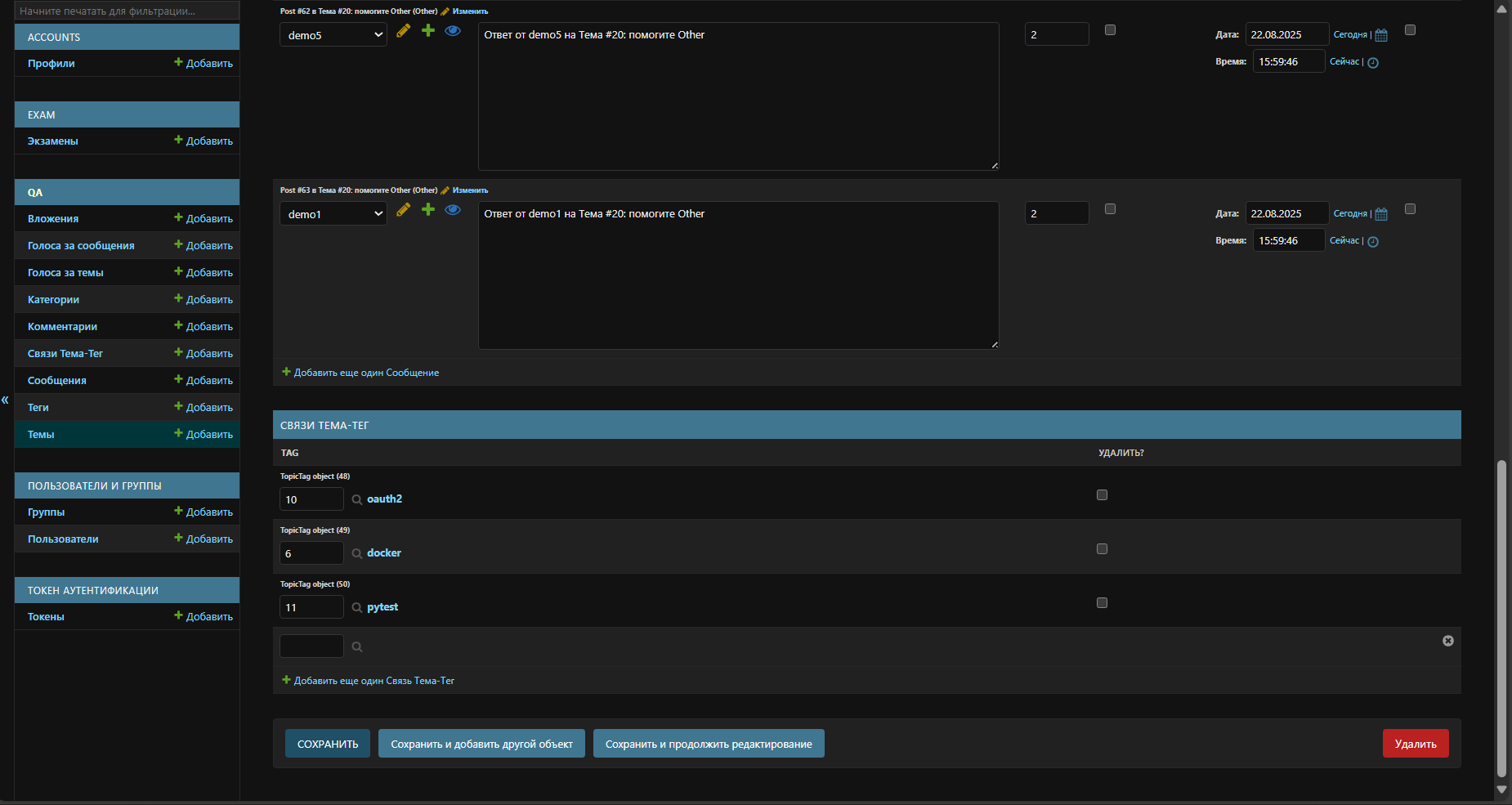
## 2.1. Административная часть

Для администрирования реализована кастомизация Django Admin:

* list\_display, list\_filter, search\_fields;
* date\_hierarchy и readonly\_fields;
* inlines для связи тем и постов;
* filter\_horizontal для M2M (теги).

**

Список тем в админке 1

**

Форма редактирования темы 1

## 2.2. Главная страница

Главная страница реализована на Vue.js. Отображаются:

* список «горячих» тем за 7 дней;
* список новых тем;
* топ-5 пользователей по рейтингу.

Реализован поиск по темам и пагинация.

## 

Главная страница с виджетами 1

## 2.3. Детальная страница

Детальная страница темы отображает:

* саму тему (название, текст, рейтинг, категория);
* ответы и их рейтинг;
* комментарии;
* формы для добавления новых ответов и комментариев;
* кнопки редактирования и удаления для автора.

## 

Детальная страница темы 1

## 2.4. Работа с базой данных

Для наполнения базы используется команда python manage.py seed\_demo. Она создаёт:

* 10 пользователей;
* 10 категорий;
* 12 тегов;
* 20 тем;
* 62 поста;
* 134 комментария.

В проекте активно применяются аннотации (Count, Avg, Max) и методы values(), values\_list().  
Отдельный эндпоинт /tags/cloud/ возвращает облако тегов с количеством тем.

*(Вставить рисунок 6 – пример ответа API /tags/cloud/)*

## 2.5 Оптимизация запросов

Для оптимизации работы с базой применяются select\_related и prefetch\_related.  
Сравнение произведено с помощью Django Silk:

* без оптимизаций - множество отдельных SQL-запросов;
* с оптимизациями - количество SQL уменьшено, общее время обработки снижено.

# 

silk 1

# ЗАКЛЮЧЕНИЕ

В рамках курсового проекта была достигнута цель - разработано веб-приложение QueStudio. Реализованы:

* административная панель с расширенными возможностями;
* главная страница с поиском и виджетами;
* детальная страница темы с CRUD и системой голосования;
* сидинг базы и Postman-коллекция для проверки API;
* оптимизация SQL-запросов с доказательствами через Django Silk.

Проект соответствует требованиям курсовой работы. В дальнейшем возможна доработка: внедрение Celery для фоновых задач, отправка e-mail уведомлений, деплой на сервер.

Все поставленные задачи решены, цель достигнута.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Django REST Framework - https://www.django-rest-framework.org/
2. Django Filter - https://django-filter.readthedocs.io/
3. Vue.js Documentation - https://vuejs.org/
4. StackOverflow - <https://stackoverflow.com/>
5. Quora - [https://www.quora.com/](https://www.quora.com/" \t "_new)
6. Документация Django - https://docs.djangoproject.com/en/5.0/
7. Документация Vite - https://vitejs.dev/