

Сопроводительное письмо к проекту

1 Краткое описание

"Chirk" - это аналог популярной социальной сети Twitter. Проект предоставляет платформу, на которой пользователи могут создавать посты, выражать свои мысли и делиться новостями со своей аудиторией. Проект "Chirk" предоставляет возможность пользователям выразить свои мысли, общаться и взаимодействовать в сети, а также обеспечивает механизмы модерации для поддержания порядка и контроля качества контента.

2 Описание команды

Наша команда состоит из 6-ти человек:

- Лукашев Олег - TeamLead, Backend Developer
- Пеканов Никита - Backend Developer, DevOps Engineer
- Грибанова Дарья - Backend Developer, Technical Writers
- Бабирье Джанат - QA Engineer
- Мануковский Илья - Frontend Developer
- Акмызрак Мердан - Frontend Developer/Product Analysts

3 Описание процесса разработки

1. Этап: Подготовка и планирование

- Разработка макетов приложения для визуализации интерфейса и функциональности.
- Составление UML диаграмм для описания структуры и взаимодействия компонентов.
- Написание технического задания, которое содержит требования, функциональность и описание работы приложения.

2. Этап: Реализация и разработка

- Разработка серверной части приложения, включая обработку запросов, взаимодействие с базой данных и бизнес-логику.
- Разработка клиентской части приложения, включая интерфейс пользователя, взаимодействие с сервером и обработку данных.
- Реализация основных модулей приложения, которые включают функциональность, такую как аутентификация, управление данными и т.д.
- Структурирование кода, использование подходящих архитектурных шаблонов и библиотек.

3. Этап: Тестирование и развертывание

- Тестирование приложения для обнаружения и исправления ошибок, проверка его работоспособности и соответствия требованиям.
- Развертывание приложения на выбранном хостинге, настройка необходимых серверных компонентов и базы данных.
- Написание курсового проекта, который документирует весь процесс разработки, используемые технологии, решенные задачи и результаты тестирования.

4 Описание основных функций проекта

Для неавторизованного пользователя:

- Просмотр общей ленты

Для авторизованного пользователя:

Включает в себя все возможности неавторизованного пользователя и расширяется:

- Возможность создать пост

- Возможность создать, который будет виден только 24 часа после создания
- Возможность ставить лайк/дизлайк на пост
- Просмотр данных своего профиля
- Возможность редактировать данные профиля
- Возможность редактирования пароля
- Просмотр понравившихся постов
- Просмотр не понравившихся постов
- Просмотр своих постов
- Возможность удалить свой пост
- Возможность скрыть свой пост

Для модератора:

Включает в себя все возможности авторизованного пользователя и расширяется:

- Возможность удалять посты других пользователей

Для администратора:

Включает в себя все возможности модератора и расширяется:

- Возможность назначать/отменять модераторов

5 Описание проблем, с которыми столкнулась команда

В ходе разработки мы столкнулись с основной проблемой - невозможностью добавления модераторов через мобильное приложение.

Для решения этой проблемы мы внесли следующие изменения:

- Создали новую роль - администратора. Администратор автоматически создается при старте сервера и добавляется в базу данных, если его там нет.
- Администратор получил расширенные права доступа, включая возможность присваивать и отзывать права модератора у авторизованных пользователей.

— Только администратор имеет возможность добавлять и удалять модераторов.

Для удобства использования мы разработали web-приложение. Веб-приложение состоит из двух страниц: "admin_authorization" и "admin_panel".

Страница "admin_authorization" представляет собой форму авторизации, доступную только администратору с соответствующими правами. Если обычный пользователь попытается авторизоваться, ему будет выведено сообщение об ошибке.

После успешной авторизации, администратор попадает на страницу "admin_panel", где отображается список зарегистрированных пользователей. Для удобства использования, реализована возможность фильтрации списка по фамилии и email пользователя через форму фильтрации.

На странице "admin_panel" также есть возможность назначать и отзывать права модератора, путем ввода email нужного пользователя в соответствующую форму.

Страница "admin_panel" доступна только администратору после успешной авторизации.

Таким образом, эти изменения и веб-приложение обеспечивают команде удобные средства для управления модераторами и пользователями.

Для разработки web-приложения мы выбрали язык программирования Python в сочетании с фреймворком Flask. Этот выбор основан на ряде причин.

Во-первых, Python - это мощный и гибкий язык программирования, который обладает простым и понятным синтаксисом. Это делает его идеальным выбором для разработки web-приложений, особенно в случае, когда команда разработчиков имеет различный уровень опыта.

Во-вторых, Flask является легковесным и гибким фреймворком для разработки web-приложений на языке Python. Он предоставляет минималистичный набор инструментов, позволяющих создавать эффективные и масштабируемые приложения. Flask также обладает обширной документацией и активным сообществом разработчиков, что упрощает

процесс разработки и обеспечивает доступ к множеству расширений и плагинов.

Кроме того, выбор Python и Flask позволяет нам быстро создать web-приложение с необходимыми функциональными возможностями. Мы можем легко обрабатывать запросы, взаимодействовать с базой данных и реализовывать логику авторизации и управления правами доступа.

Таким образом, выбор Python и Flask для разработки нашего web-приложения обусловлен их простотой, гибкостью, доступностью расширений и возможностью создания эффективного и масштабируемого приложения.