1. Меня зовут Халитов Мурад, я участник командного проекта "Социальная сеть". В рамках проекта я разрабатывал микросервис для управления постами и комментариями в социальной сети.

2. Микросервис отвечает за полный цикл работы с постами и комментариями, обеспечивая следующие ключевые функции:

- Создание, редактирование и удаление постов.
- Фильтрация и сортировка постов по различным параметрам.
- Работа с комментариями: добавление комментариев к постам, а также создание цепочек ответов на комментарии.
- возможность ставить и удалять лайки как к постам, так и к комментариям.

Особенностью данного микросервиса является поддержка вложенных комментариев , что обеспечивает удобство общения пользователей.

3. В разработке микросервиса использован следующий технологический стек:

- **Java 17**: реализация бизнес-логики.
- **Spring Boot 3**: создание микросервисов.
- **Spring Cloud**: построение микросервисной архитектуры.
- **Spring Security**: обеспечение безопасности.
- **Liquibase**: для управление изменениями в базе данных.
- **Spring Data JPA**: для работы с базой данных.
- **PostgreSQL**: для хранения постов, комментариев и лайков.
- **Docker**: контейнеризация для упрощения развертывания и масштабирования.
- **Kafka**: для асинхронного обмена сообщениями между сервисами.
- **OpenFeign**: REST API клиент для взаимодействия с микросервисами.

4. Проект структурирован по типичному паттерну MVC (Model-View-Controller), что облегчает поддержку и расширение функционала:

- **Контроллеры**: REST API для работы с постами и комментариями.
- **Сервисы**: Для обработки бизнес-логики, работы с данными и взаимодействие с внешними сервисами.
- **Репозитории**: слой доступа к данным, реализованный через Spring Data JPA, для работы с базой данных.

5. Микросервис Post активно взаимодействует с другими микросервисами проекта для выполнения дополнительных задач. Дальше я покажу взаимодействие сервиса Post с другими микросервисами при создании поста:

6. Клиент отправляет запрос на создание поста в сервис Post

7.

8. Он проходит через SecurityFilterChain где извлекается jwt токен

если спросят. SecurityFilterChain это цепочка фильтров, которая обрабатывает запросы в рамках системы безопасности Spring Security. Каждый фильтр в этой цепочке выполняет свою задачу, например:

- Аутентификация (проверка личности пользователя, например, на основе JWT токена),
- Авторизация (проверка прав доступа к ресурсам),
- Логирование или контроль сессий.

9. Затем токен отправляется в сервис Авторизации на валидацию.

10. После успешной валидации токена сервис переходит в метод создания поста createPost, где данные отправленные в запросе мапятся в сущность Post

11. И сохраняются в БД.

12. Микросервис Post не хранит информацию о пользователях (кроме их ID)

13. поэтому для получения данных о пользователе используется OpenFeign клиент. Сервис Post передает ID пользователя в сервис Account

14. и получает информацию о пользователе(а конкретно имя и фамилию) Затем формирует сообщение для телеграмм бота.

15. После чего отправляет его через кафку в топик который слушает телеграм бот.

16. А также в топик который слушает сервис уведомлений, чтобы уведомить пользователей на платформе о новом контенте.

17. Дополнительной задачей было разработать телеграм-бот для уведомления пользователей о новых постах. Бот получает сообщения от микросервиса Post через Kafka и отображает имя автора заголовок и текст поста. Это сообщение помогает пользователям своевременно получать информацию о новых постах через телеграм.

18. Для обеспечения высокого качества кода и минимизации ошибок мы используем SonarQube. Этот инструмент анализирует проект на ранних этапах, выявляя потенциальные проблемы и предоставляя рекомендации для улучшения структуры кода и безопасности.

19. В моем микросервисе я использую два типа тестов: модульные и интеграционные. Модульные тесты позволяют проверять отдельные методы и классы с помощью Junit. Интеграциооные тесты позволяет тестировать работу контроллеров, сервисов и репозиториев в условиях, максимально приближенных к реальной эксплуатации.

20. Дальше я покажу вам на фронте функционал моего сервиса.

На вкладке новости показываются собственные посты и посты друзей, idшники друзей через OpenFeign клиент получаем из сервиса Friends.

На вкладке Моя страница видим только свои посты

Создадим пост «Новый пост!» текст поста.

Отредактируем пост «Новый пост 2» текст поста 2, добавим тег «teg» добавим картинку.

Удалим пост.

Покажу что можно писать комментарии, на примере существующих.
Можно ставить лайки как к комментарию так и к посту.

Перейдем к фильтрации

если ввести в поле поисковика слово будет происходить поиск в по авторам и по тексту постов, нас интересуют посты,

можно искать по автору, при поиске по автору мой сервис обращается к сервису Account, можно вводить имя, имя и фамилию, только фамилию, можно ввести часть имени или фамилии
можно отфильтровать по периоду создания поста,
фильтрация по тегам тоже работает, но так как на фронте косяк и запрос с тегом не отправляется на бакенд я здесь не могу продемонстрировать этот фильтр. Но я тестировал его в постман он тоже работает, Так как я показал весь функционал сервиса у меня все, спасибо за внимание.