Приложение “Support”.

Технические требования:

1) Никакого фронта/темплейтов, только Backend API, стремимся к REST архитектуре.
2) Технологии:
   Django + Django Rest Framework, JWT авторизация, PostgreSQL, Docker (Docker-compose), PyTests (для тестов, лучше пару написать, просто чтобы понимать что это и зачем, и как с этим работать), Celery и Redis в качестве брокера сообщений.
3) Для код стайла:
   flake8(можно и другие либы юзать, если у вас с ними есть опыт, только поставьте ограничение длины строки на 120 символов), isort(для импортов).

Мои пожелания:

Стремитесь написать эффективный и понятный любому программисту код, документируйте код(без фанатизма, в местах со сложной логикой, можете писать комментарии, но если будете придерживаться SOLID принципов, все в принципе и так будет ясно), там где это надо, учитывайте, что почти на всех проектах вам надо работать в команде, и вы должны сделать так, чтобы им было с вами комфортно работать. Думайте о том что пишите, выносите логику в отдельные сервисы, пишите так, чтобы не приходилось сидеть и разбирать функцию на 500 строк, разбейте логику на небольшие задачи, это повысит читаемость и пере используемость. Главное не просто сделать, или там, сделать быстро, а сделать качественно, показать что вы можете,чтобы я посмотрел ваш код, и понял что хочу с вами работать(в идеале :D).

Описание бизнес задачи:

Базово:

Служба саппорта:

1) Пользователь пишет тикет и отправляет.
2) Саппорт видит решенные, нерешенные и замороженные тикеты (все по факту), может отвечать на них.
3) Пользователь может просмотреть ответ саппорта, и добавить новое сообщение(саппорт ответить на него).
4) Саппорт может изменять статусы тикетов.

Дополнительно:
-

По желанию:
Любые кастомные дополнения, чем лучше вы себя проявите, тем лучше.