**Фидбек по ТЗ - (советуем подтянуть):**

**Плюсы:**

Реализованы все функции, включая регистрацию, логин, логаут, обновление профиля и удаление аккаунта. Реализованы таблицы и модели для ролей, правил и привязок ролей к пользователям. Предусмотрены эндпоинты для управления правами только администратором. Используется кастомная JWT авторизация, хэширование паролей реализовано. API охватывает большинство требуемых операций. Swagger интегрирован для удобства.

**Минусы:**

- JWT exp ломается на Windows: int(dt.strftime(""%s"")) не работает на Win; приводит к падению. Лучше использовать int(dt.timestamp()).

Исправил

- Отсутствует подтверждение пароля на регистрации: Нет поля повтора пароля и проверки совпадения.

Исправил. Добавил поле и проверку.

- 401 vs 403: При отсутствии токена возвращается 403 Forbidden вместо 401 Unauthorized.

Исправил, переопределил в backend и permission

- Profile(role=ForeignKey(Role)) обязателен, но его не создают при регистрации. Без профиля проверки ролей падают на AttributeError и приводят к 403. Нигде не назначается роль пользователю (кроме админки).

Исправил. Добавил по умолчанию при регистрации в drf сразу роль Buyer, повышение ролей через drf update /user/

IsAdminOrNoAccess потребует профиль с ролью Admin — в реальности доступ к управлению доступами будет заблокирован.

В createsuperuser сразу добавляется drf role Админ.

- Процесс logout не завершен: токены не помещаются в блэклист, что делает процесс аннулирования токенов неполным.

Исправил, по выходу токены добавляются в блэклист, и проверяются при входе с удалением старого.

- Слабый ридми.