TEXNIK TOPSHIRIQ

Loyiha: Coffee Shop API — foydalanuvchilarni boshqarish bloki

Maqsad: Foydalanuvchilarni to‘liq boshqarish funksiyalarini amalga oshirish: ro‘yxatdan o‘tish, avtorizatsiya, autentifikatsiya va verifikatsiya. Loyihaning tuzilishini tavsiflash. Arxitektura kengaytiriladigan, production-ready bo‘lishi kerak.

FUNKSIONAL TALABLAR

1. Foydalanuvchilarni ro‘yxatdan o‘tkazish Yangi foydalanuvchini ro‘yxatdan o‘tkazuvchi endpointni amalga oshiring: - Email - Parol - Ism, familiya (ixtiyoriy) - Emailning noyobligini tekshirish. - Ro‘yxatdan o‘tganidan so‘ng foydalanuvchi “verifikatsiyadan o‘tmagan” (not verified) statusini oladi.

2. Autentifikatsiya va avtorizatsiya JWT (access va refresh tokenlar) dan foydalanilsin. Endpointlar: - POST /auth/signup — ro‘yxatdan o‘tish - POST /auth/login — avtorizatsiya (tokenlarni berish) - POST /auth/refresh — access-tokenni yangilash

3. Verifikatsiya Email yoki SMS orqali verifikatsiyani amalga oshiring: - Ro‘yxatdan o‘tganidan so‘ng tasdiqlash uchun kod yoki havola (link) yaratiladi (amaliyotda konsolga chiqarish mumkin — dummy). - POST /auth/verify — verifikatsiya tasdiqlash endpointi. - Muvaffaqiyatli verifikatsiyadan so‘ng foydalanuvchi “verifikatsiyalangan” (verified) statusini oladi.

4. Avtomatik tozalash - 2 kundan ortiq verifikatsiyadan o‘tmagan foydalanuvchilar avtomatik o‘chirib tashlanishi kerak. - Buni Celery orqali amalga oshirish yoki READMEda rejalashtirilgan logikani batafsil tavsiflash lozim.

5. Foydalanuvchi rollari Ikki rol mavjud: - User — oddiy foydalanuvchi - Admin — kengaytirilgan funksiyalarga ega foydalanuvchi Endpointlarga kirishni rollarga qarab cheklashni amalga oshiring.

6. Foydalanuvchilarni boshqarish endpointlari - GET /me — joriy foydalanuvchini olish - GET /users — foydalanuvchilar ro‘yxati (faqat admin uchun) - GET /users/{id} — ID bo‘yicha foydalanuvchini olish (faqat admin uchun) - PATCH /users/{id} — foydalanuvchi ma’lumotlarini qisman yangilash - DELETE /users/{id} — foydalanuvchini o‘chirish (faqat admin uchun)

NEFUNKSIONAL TALABLAR - FastAPI dan foydalanish. - Asinxron arxitektura. - ORM: SQLAlchemy. - Ma’lumotlar bazasi: PostgreSQL yoki SQLite. - Konteynerizatsiya: Docker — Dockerfile va docker-compose.yml. - Endpointlar hujjatlari: OpenAPI (Swagger UI). - Barcha kommentariy va tavsiflar ingliz tilida bo‘lishi kerak.

QO‘SHIMCHA TALABLAR - Loyiha jamoat GitHub repozitoriyasida joylashtirilishi kerak. - Email/SMS yuborish, JWT generatsiyasi kabi funksiyalar uchun uchinchi tomon kutubxonalaridan foydalanish mumkin. - Barcha endpointlar uchun o‘qilishi oson summary va description lar ingliz tilida bo‘lsin. - READMEda ishga tushirish bo‘yicha ko‘rsatmalar va modul arxitekturasi tasviri bo‘lsin. - Agar siz ba’zi qismlarni ongli ravishda soddalashtirsangiz, kod ichida izoh bilan qoldiring — “agar ko‘proq vaqt bo‘lsa, buni shunday amalga oshirardim” tarzida.