

مراحل اجرای برنامه:

۱- در مسیر `app>tracker` اسم فایل `env.example` را به `env` تغییر دهید.

۲- در دایرکتوری `app` ترمینال را باز کرده و دستور `docker-compose up --build` را وارد کنید.

انتخاب های مهم طراحی و ترید اف هایی که انجام دادید چه بود؟

برای طراحی سعی کردم تا برنامه تا حد امکان ایزوله باشه برای همین بیزنس لاجیک رو از ویوها جدا کردم. البته برنامه هنوز هم به orm وابسته هست. اگه زمان بیشتری باشه میشه یک لایه دیتا ترنسفر هم اضافه کرد تا برنامه دیگه به دیتابیس یا orm خاصی وابسته نباشه.

در بخش ردیابی بسته تصمیم گرفتم اطلاعات آب و هوایی ک از طریق API دریافت میکنم رو در دیتابیس ذخیره نکنم. به همین دلیل از ردیس استفاده کردم و هر درخواست کاربر رو به مدت دو ساعت کش میکنم. دلیل این کار کمتر کردن عملیات رایت روی دیتابیس اصلی و سریع تر بودن ردیس در برگرداندن اطلاعات بود. و اینکه فکر میکردم که نیازی به ذخیره اطلاعات نیست چون کاربر فقط در همان لحظه به اون اطلاعات نیاز داره.

از طرف دیگه وقتی بسته در حالت تحول داده شده قرار گرفت نیاز بود که کاربر تمام مسیری که بسته بسته طی کرده تا به مقصد برسه رو ببینه. برای همین زمانی که حمل کننده بسته تغییر مکانش رو توی سیستم وارد میکنه یک بار اطلاعات آب و هوایی مکان جدید گرفته میشه و در مدل اطلاعات بسته ذخیره میشه. پس کاربر بعد از اینکه بسته رو تحویل گرفت میتونه تمام مسیری که بسته رفته را همراه با اطلاعات آب و هوایی اون مناطقی که بسته در اونجا بوده رو ببینه.

برای استقرار این برنامه در پروداکشن چه چیزی لازم است؟

من در شروع پروژه سعی کردم که تا حد امکان محیط توسعه با پروداکشن شبیه به هم باشد. برای همین از postgres, nginx, gunicorn توای محیط توسعه استفاده کردم. اما بازم برای بهتر کردن پروژه بهتره که کارهایی مثل جداکردن فایل requirements توسعه و پروداکشن و جدا کردن داکر فایل دو محیط و بررسی نکات امنیتی مثل false بودن دیباگر جنگو قراردادن secret key جنگو در env تنظیم cross origin برای قبول کردن درخواست ها از ادرس خاص رو انجام داد. همچنین تنظیم پسورد سخت برای دیتابیس ها هم از نکاتیه که باید انجام بشه.

چه چیزی لازم است تا مقیاس این برنامه برای رسیدگی به 1000 درخواست در ثانیه انجام شود؟

بنظر من هات پوینت این اپلیکیشن یو آر ال track هستش که با اون میشه اطلاعات بسته همراه با اطلاعات آب و هوایی رو دید.

کاری که بنظر من برای بهینه کردن کد میشه انجام داد با کمک سلری بیت به صورت دوره ای اطلاعات آب و هوایی رو کش کنیم. تا کاربر همیشه اطلاعات آب و هوایی رو از کش بگیره و اتصال به API آب و هوا فقط از طریق سلری بیت انجام بشه. و همچنین میشه کوئری این بخش که اطلاعات بسته رو میگیره رو هم کش کرد تا ریکوئست به سرور تا حد امکان کم بشه. ایندکس کردن شماره بسته و اسم حمل کنند بسته در دیتابیس هم میتونه به سریع تر شدن کوئری ها کمک بکنه. برای بخش اپدیت کردن لوکیشن حامل بسته هم باید اطلاعات کل بسته ها به روز رسانی بشه. من از bulk create استفاده کردم اما بنظرم باید این کار رو به یک تسک سلری تبدیل کنم تا پرفورمنس بهتری داشته باشه.

از سمت دیگه میشه از تکنیک هایی مثل LOAD BALANCING به کمک انجین ایکس استفاده کرد تا ریکوست هارو بین سرورها پخش بکنه.

برای تشخیص مشکلات سیستم میشه اون رو زیر فشار برد. مثلاً از تست بار استفاده کرد و با مانیتورکردن و مشخص شدن مشکلات اونهارو برطرف کرد.