**آزمون بک پک ۵**

سلام!

می‌خواهیم با هم یک پلتفرم ساده برای ساخت و ویرایش متن طراحی کنیم. ایده‌اش خیلی شبیه سرویس‌های آنلاین ویرایش متن هست (مثل Google Docs اما در نسخه‌ی خیلی ساده‌تر). شما قراره **بک‌اند** این پلتفرم رو پیاده‌سازی کنید.

برای رسیدن به این نسخه ساده و اولیه، چند مرحله مشخص شده که می خواهیم گام به گام پیش برویم. در هر مرحله چالش هایی هم مطرح شده که اگر دوست داشتی مهارت خودت را به چالش بکشی حتما برو سراغشون.

پروژه رو به دو سرور تقسیم می‌کنیم که در ادامه شرایط هر سرور مشخص شده است.

---

**بخش ۱: سرور ویرایش متن**

این سرور قلب ماجراست و وظیفه‌اش کار با متن‌هاست. مستقیم با کلاینت‌ها حرف نمی‌زند، بلکه فقط از سرور **Proxy**درخواست دریافت می‌کند.

**امکانات مورد نیاز:**

۱. **ایجاد متن جدید** برای کاربر (هر متن با یک شناسه ذخیره شود و ممکن است برای چند کاربر در دسترس باشد).

۲. **حذف متن** قبلی توسط ایجاد کننده متن.

۳. **ویرایش متن** (در نسخه ساده می تواند کل متن جایگزین شود).

٤. **دریافت یک متن** برای نمایش ( در این حالت، یک مود read only هم به سلیقه خودتون اضافه کنید. وقت هایی که یک متن فقط می تواند دیده شود و قابلیت ویرایش ندارد ).

**چالش!**

انجام یکی از چالش های زیر اجباری است:

۱. قسمت ویرایش را اگر خواستید، حالت پیشرفته‌تر با تغییر کاراکترها پیاده کنید (یعنی فقط قسمت تغییر کرده و اطلاعات لازم برای سرور ارسال شود و در متن اصلی جایگذاری شود. نیازی به پوشش همه حالت های تغییر هم نیست).

۲. ویرایش همزمان

حالا فرض کنید دو نفر همزمان می‌خواهند یک متن را تغییر دهند. چطور جلوی تداخل را می‌گیرید؟

- ساده‌ترین روش: قفل کردن فایل وقتی یک نفر در حال ویرایش است.

- روش پیشرفته‌تر: نگه داشتن نسخه‌ها یا merge کردن تغییرات.

هر راه‌حلی ارائه کنید، به خلاقیت و منطق آن نمره اضافه داده می‌شود.

**بخش ۲: سرور پروکسی(Proxy Server)**

این سرور مثل یک واسطه جلوی سرور اصلی قرار می‌گیرد. کاربر مستقیم با این سرور حرف می‌زند وچند کار مهم را انجام می‌دهد:

۱. **اعتبارسنجی کاربر**

- هر کاربر یک **توکن ثابت** دارد.
- Proxy یک لیست از توکن‌های معتبر نگه می‌دارد.
- فقط درخواست‌هایی که توکن معتبر دارند، به سرور اصلی فرستاده می‌شوند.

نکته : قضیه رو اینجا سخت نکنید و دنبال مدیریت توکن و دیتابیس برای ذخیره سازی نباشید. یک راه ساده براش کشف کنید!

۲. **یکنواخت کردن درخواست‌ها**

- چون کلاینت‌ها ممکن است درخواست‌هایی با فرمت متفاوت بفرستند، Proxy باید همه‌ی درخواست‌ها را به یک فرمت یکسان تبدیل کند و برای سرور ویرایش بفرستد.

**چالش!**

۱. **لاگ گرفتن**

- زمانی که کاربر برای اولین بار تقاضای ایجاد متن میکنه، ما بهش می گیم فاز handshake. بعد از اون هر تغییری توی کاراکترها داد، فاز بعدی هستند. حالا میخوایم وقتی یک کاربر وارد فاز handshake شد، یک لاگ ثبت شود. یعنی با لاگ بگیم که فلان کاربر میخواد فلان متن رو بخونه یا ویرایش کنه!
- لاگ شامل: نام کاربر + نوع عملیاتی که می‌خواهد انجام دهد + (read, create, edit, delete) و زمان.
- توجه کنید: لاگ فقط در **handshake** ذخیره می‌شود، نه برای هر درخواست کوچک بعدی.

---

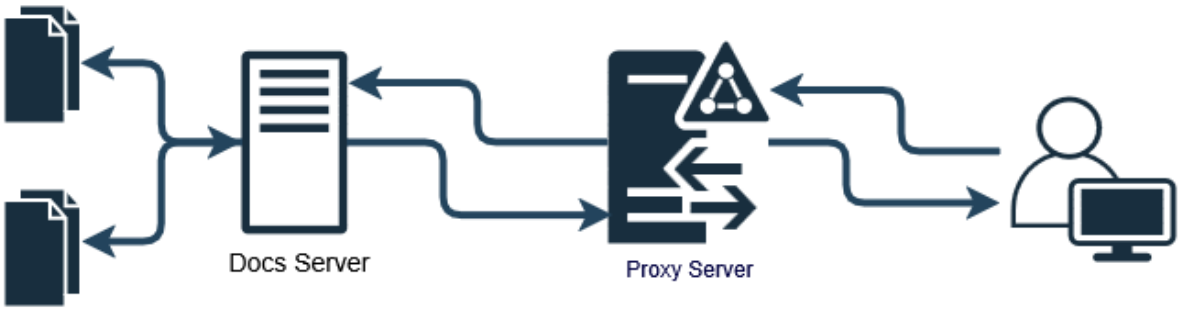
**ارتباط و پروتکل‌ها (اختیاری)**

- اگر به **WebSocket** مسلط هستید، می‌توانید ارتباط Client ↔ Proxy ↔ Main Server را با آن پیاده کنید (هر تغییر کوچک یا کاراکتر را بفرستید).
- اگر نه، خیلی ساده با **HTTP**پیاده‌سازی کنید (هر درخواست شامل یک بلوک متن یا مجموعه‌ای از تغییرات باشد).
- انتخاب با شماست.

---

**نکات مهم**

- پروژه باید با Node.js و ماژول‌های داخلی (fs, path, http, events) پیاده‌سازی شود.
- متن‌ها روی سرور در فایل‌ها ذخیره شوند (فرمت JSON یا TXT ).
- برای هر متن، حداقل { id, owner, content, createdAt, updatedAt} نگه دارید.
- در ازای هر درخواستی که از سمت سرور **Proxy** به سرور ویرایش متن زده می‌شود، پاسخ با فرمت مشخص برگردانده شود.
- به تمامی جزئیات مربوط به هر API را دقت و رعایت کنید.
- مدیریت خطا به درستی لحاظ شود.
- نکات مربوط به کدتمیز حتما رعایت شود.
- موارد مربوط به گیت رعایت شود.
- **تست چند API اصلی با Mocha یا Jest نوشته شود.**




موفق باشید.