

OSOBNÍ REFLEXE PROJEKTU EDUAGENT

ZVOLENÉ ŘEŠENÍ A TECHNOLOGICKÝ STACK

Python jsem zvolil záměrně kvůli rozšíření obzorů a hlubším znalostem v oboru AI. Zpětně ale hodnotím volbu Pythonu a odděleného backendu jako komplikaci – hlavním problémem byla zbytečná reže a složitost separátního API a klienta. Musel jsem re-implementovat mnoho funkcí, které moderní full-stack frameworky (např. Next.js) již řeší nativně.

Komplikací byl i LangChain a jeho komplexní abstrakce, zejména u streamování. Ačkoliv jsem znal efektivnější řešení jako Vercel AI SDK, zvolil jsem LangChain kvůli návaznosti na Python. Zpětně vidím, že setrvání u osvědčených TS nástrojů by bylo efektivnější než vrstvení nad LangChainem.

Naopak velmi pozitivním objevem byl nástroj „uv“, který mi výrazně pomohl se správou tohoto monorepozitáře. Jeho rychlost a efektivita při práci se závislostmi v Pythonu mi ušetřila mnoho času a zpříjemnila práci v prostředí, které pro mě bylo v tomto rozsahu nové.

VÝVOJOVÉ FÁZE A UI/UX

Zklamáním byla vizualizace myšlenkových map. Knihovny jako React Flow neumožnily interaktivní rozhraní podobné NotebookLM a mapy zůstaly statické a nepřehledné. Kvůli času jsem ponechal funkční, ale ne ideální řešení.

Naopak design UI považuji za velmi zdařilý. Čisté prostředí a plynulé streamování chatu poskytují uživateli skvělý zážitek, což je jeden z nejsilnějších bodů aplikace.

INFRASTRUKTURA A NÁKLADY

Celou infrastrukturu v Azure jsem spravoval pomocí nástroje Terraform. Práce s ním pro mě byla velkým přínosem a naučil jsem se díky němu efektivně definovat a nasazovat cloudové prostředky jako kód (Infrastructure as Code).

Absence SDK pro Azure Content Understanding mě donutila generovat klienta z OpenAPI specifikace. Práce přímo s REST API mi ve výsledku poskytla větší kontrolu a flexibilitu při integraci služby.

V prosinci jsem chyboval v analýze nákladů Azure Storage Queue. Neustálé operace mě stály 30 USD v kreditech, než jsem upravil logiku. Byla to drahá, ale užitečná lekce o cenotvorbě cloudů a nutnosti monitoringu.

CO BYCH PŘÍŠTĚ UDĚLAL JINAK

- **Stack:** Příště zvolím Node.js/Next.js/Tanstack Start – ušetřil bych si mnoho práce s backendem
- **UI rešerše:** Věnoval bych více času hledání knihoven pro klíčové prvky (mind mapy) před samotnou implementací
- **Cloud monitoring:** Nasadil bych rozpočtové limity od prvního dne

ZÁVĚR

EduAgent mě naučil pracovat s nejmodernějšími AI technologiemi a prohloubil mé znalosti cloudové infrastruktury. I přes technologické překážky se mi podařilo doručit funkční a esteticky zdařilý produkt, který splňuje stanovené cíle.