BÄRNHÄCKT

Bärnstormers PostFinance: SpendCast

Technische Informationen für die Jury

Technische Informationen für die Jury

Unsere Vision ist es, das persönliche Finanzmanagement von einer lästigen Pflicht in ein motivierendes, interaktives Erlebnis zu verwandeln. Wir entwickeln dafür eine mobile Finanz-Assistentin, die auf einem persönlichen Gespräch und Gamification-Elementen basiert. Über einen Chatbot stellen Nutzer*innen Fragen; das System extrahiert die relevanten Informationen aus einem Wissensgraphen und beantwortet sie mithilfe eines LLM (z.B. OpenAI ChatGPT). Zusätzlich erhalten Nutzer*innen eine monatliche Ausgabenübersicht im Stil eines „Spotify Recaps“, können Abzeichen für verantwortungsbewusstes Konsumverhalten sammeln (z.B. regionale oder gesunde Produkte) und ihr Ausgabeverhalten mit Vormonaten vergleichen. Auf Basis der letzten Monate wird außerdem ein Quiz generiert,

https://github.com/TheArchbishopOfDjentinbury/Baernstormers

Ausgangslage

- **Worauf habt ihr euch fokussiert?**

  Wir haben uns strategisch für einen **Audio-First-Ansatz** entschieden, anstatt der ursprünglich angedachten Video-Generierung. Unsere Analyse ergab, dass wir innerhalb des Hackathon-Zeitrahmens ein qualitativ hochwertiges und immersives Audio-Erlebnis schaffen können. Dies ermöglicht zudem unser Kern-Feature: einen **persönlichen Audio-Begleiter**, der die Nutzer*innen durch ihre Finanzen führt – eine weitaus persönlichere und skalierbarere Lösung als generisches Video.

  Zudem setzen wir auf eine **LLM-agnostische Architektur**. Das von uns genutzte Framework und MCP-Server-Protokoll funktionieren unabhängig vom verwendeten LLM. Das erlaubt uns, mit verschiedenen Modellen zu experimentieren und schafft eine zukunftssichere Grundlage für die Skalierung.

- **Welche technischen Grundsatzentscheide habt ihr gefällt?**
- **Audio-First statt Video:** Diese Entscheidung ermöglichte unser Kern-Feature des persönlichen Begleiters und sicherte eine hohe Umsetzungsqualität innerhalb des Zeitrahmens.
- **Architektur nach dem RAG-Muster (Retrieval-Augmented Generation):** Der Kern unserer Lösung ist die Anbindung des LLM an den PostFinance-Wissensgraphen. Anstatt dem LLM unkontrolliert Anfragen zu überlassen, nutzen wir den Graphen als alleinige „Source of Truth“. Dieser RAG-Ansatz **verhindert Halluzinationen** und stellt sicher, dass alle Antworten auf validen, kundenspezifischen Daten basieren – ein Muss für eine Finanzanwendung.
- **Modulare, LLM-agnostische Service-Architektur:** Durch die Kapselung des LLM-Zugriffs hinter unserem MCP-Server-Protokoll vermeiden wir einen Vendor-Lock-in. Dies erlaubt uns, jederzeit das kosteneffizienteste oder leistungsfähigste Modell (z.B. OpenAI, ein Open-Source-Modell oder Groq) zu integrieren, ohne die Kernanwendung zu verändern.

Technischer Aufbau

- **Welche Komponenten und Frameworks habt ihr verwendet?**
- LangChain zur Orchestrierung des RAG-Workflows (Abfragen, Kontext-Erstellung, Prompting).
- **MCP-Server-Protokoll** als standardisierte, modellunabhängige Schnittstelle zu LLMs.
- **FastAPI** als performantes Backend-Framework.
- **LLM-Anbindung** (aktuell OpenAI ChatGPT, flexibel austauschbar).
- **PostFinance Wissensgraph** als primäre Datenquelle.

- Wozu und wie werden diese eingesetzt?

Der Prozess folgt einem klaren RAG-Workflow, orchestriert von LangChain:

- **Nutzerfrage:** *"Wofür habe ich letzten Monat am meisten Geld ausgegeben?"*
- **LangChain (Retrieval):** Analysiert die Frage und übersetzt sie in eine präzise Cypher-Abfrage für den Wissensgraphen. z.B.

```
MATCH (u:User {id: 'xyz'})-[:MADE]->(t:Transaction) WHERE t.date >= '2025-07-01' ...
RETURN t.category, sum(t.amount) ORDER BY sum DESC LIMIT 1
```

- **Kontext-Erstellung (Augmentation):** LangChain erhält das Abfrageergebnis (z.B. {'category': 'Wohnen', 'sum': 1200 CHF}) und fügt es in einen Prompt-Template ein.
- **LLM (Generation):** Das LLM erhält nur diesen minimalen, faktenbasierten Kontext und die Anweisung, eine freundliche Antwort zu formulieren. Es generiert: *"Letzten Monat ging der grösste Teil deiner Ausgaben mit 1200 CHF für das Wohnen drauf."*

Diese Methode ist extrem effizient und stellt sicher, dass das LLM **nur als „Sprachrohr" für die validen Daten des Graphen** agiert.

Implementation

- Gibt es etwas Spezielles, was ihr zur Implementation erwähnen wollt?
- **Direkte Anbindung an den Wissensgraphen:** Unsere Chatbot-Interaktion greift nicht auf eine Kopie, sondern direkt und in Echtzeit auf den PostFinance-Wissensgraphen als „Single Source of Truth" zu.
- **Personalisierte Audio-Companions:** Wir planen auswählbare Stimmen bzw. Avatare, die persönliche Tipps und Gamification-Feedback geben.

- Was ist aus technischer Sicht besonders cool an eurer Lösung?

Der entscheidende technische Vorteil unserer Lösung ist die **extreme Effizienz und Skalierbarkeit.** Indem wir dem LLM über LangChain nur winzige, kontextbezogene Snippets aus dem Wissensgraphen bereitstellen, minimieren wir die Token-Nutzung drastisch. Dies ermöglicht den Einsatz von blitzschnellen und kostengünstigen Inferenz-Engines wie **Groq**. Unsere Architektur ist damit nicht nur ein Prototyp, sondern eine **produktionsreife Grundlage**, die für tausende von Nutzern skaliert, ohne die Betriebskosten explodieren zu lassen.

Abgrenzung / Offene Punkte

- Welche Abgrenzungen habt ihr bewusst vorgenommen und damit nicht implementiert? Weshalb?

Videogenerierung: Aus Gründen der Komplexität und der Qualitätsanforderungen bewusst weggelassen, zugunsten eines hochwertigen Audio-Erlebnisses.