

Rendszerspecifikáció: ReceptSéf Alkalmazás

1. Bevezetés

A **ReceptSéf** egy mesterséges intelligenciával támogatott webalkalmazás, amelynek célja, hogy megoldást nyújtson a minden nap "mit főzzek?" problémára. Az alkalmazás lehetővé teszi a felhasználók számára, hogy a rendelkezésükre álló alapanyagok megadásával személyre szabott recepteket generáljanak egy Large Language Model (LLM) segítségével. A rendszer nemcsak recepteket kínál, hanem támogatja a bevásárlási folyamatot is a hiányzó hozzávalók listázásával.

2. Felhasználói Szerepkörök

Az alkalmazásban egyetlen kiemelt szerepkört definiálunk a fejlesztés jelenlegi fázisában:

- Regisztrált Felhasználó:**

- Jogosult a rendszerbe belépni.
- Képes alapanyagok alapján receptet generáltatni az MI-vel.
- Elmentheti a generált recepteket a saját profiljába ("Kedvencek").
- Kezelheti a bevásárlólistáját.
- Megtekintheti a korábban mentett receptjeit.

3. Funkcionális Követelmények

3.1 Frontend (Kliens oldali) Követelmények

A frontend felületet **React (TypeScript)** technológiával valósítjuk meg.

- F01 - Alapanyag Bevitel:** A felhasználó egy szöveges beviteli mezőben vagy címkék (tags) hozzáadásával felsorolhatja a rendelkezésre álló alapanyagokat (pl. "tojás, liszt, tej").
- F02 - Recept Generálás Indítása:** Egy gombnyomásra a kliens elküldi az adatokat a backendnek, és várakozó állapotot (loading spinner) jelenít meg a válasz megérkezéséig.
- F03 - Recept Megjelenítés (Recept Kártya):** A szervertől kapott strukturált választ esztétikusan jeleníti meg:
 - Étel neve
 - Elkészítési idő
 - Hozzávalók listája
 - Elkészítési lépések
- F04 - Mentés Kedvencekbe:** Lehetőség a generált recept tartós mentésére az adatbázisba.
- F05 - Bevásárlólista Kezelés:** A recept nézetben a hiányzó hozzávalók kijelölése és hozzáadása egy globális bevásárlólistához. A lista elemei pipálhatók (kész) vagy törölhetők.

- **F06 - Autentikáció:** Regisztrációs és bejelentkezési űrlapok kezelése.

3.2 Backend (Szerver oldali) Követelmények

A szerver oldali logikát **Python (FastAPI)** keretrendszerrel fejlesztjük.

- **B01 - REST API Végpontok:** Szabványos HTTP végpontok biztosítása a frontend számára (GET, POST, DELETE műveletek).
- **B02 - MI Integráció (Recept Generátor):** A rendszer egy külső MI szolgáltató (pl. OpenAI GPT-4o vagy Google Gemini API) segítségével dolgozza fel a bemeneti alapanyagokat.
 - *Bemenet:* Alapanyagok lista + Rendszer prompt (instrukció, hogy recept formátumban válaszoljon).
 - *Kimenet:* Strukturált JSON adat (Név, Hozzávalók, Lépések).
- **B03 - Adatbázis Kezelés:** PostgreSQL adatbázis használata a perzisztens tároláshoz.
- **B04 - Felhasználókezelés:** Jelszavak hashelt tárolása, token alapú hitelesítés (JWT) a védett végpontokhoz.

4. Nem-funkcionális Követelmények

- **Teljesítmény:** Az MI generálás válaszideje nem haladhatja meg a külső szolgáltató átlagos válaszidejét + 1 másodpercet.
- **Biztonság:** Az API kulcsokat (MI service) környezeti változókban kell tárolni, nem a forráskódban.

5. Adatmodell Terv

Az adatbázis logikai felépítése a következő entitásokat tartalmazza:

1. User (Felhasználó):

- id: Egyedi azonosító
- username: Felhasználónév
- password_hash: Titkosított jelszó

2. Recipe (Recept):

- id: Egyedi azonosító
- user_id: Külső kulcs a létrehozóhoz (tulajdonos)
- title: Recept neve
- ingredients: Hozzávalók lista (JSON vagy szöveges)
- instructions: Elkészítési lépések (Text)
- created_at: Létrehozás dátuma

3. ShoppingItem (Bevásárlólista Elem):

- id: Egyedi azonosító

- user_id: Külső kulcs a felhasználóhoz
- item_name: Termék neve
- is_purchased: Logikai változó (megvéve vagy sem)

6. Technológiai Stack

- **Frontend:** React, Vite, TypeScript, Custom CSS.
- **Backend:** Python 3.11+, FastAPI, Pydantic.
- **Adatbázis:** PostgreSQL.
- **ORM:** SQLAlchemy.
- **MI Szolgáltató:** Google Gemini API
- **Verziókezelés:** Git & GitHub.

7. MI Használata a Fejlesztés Során

A projekt fejlesztése során a mesterséges intelligencia (MI) használata kötelező elem. Az alábbi területeken kerül alkalmazásra:

1. **Alkalmazás Funkció:** Maga az alapfunkció (recept generálás) MI alapú.
2. **Kódolás Támogatás:** GitHub Copilot / ChatGPT használata a boilerplate kódok (API váz, React komponensek) generálására és refaktorálására.
3. **Dokumentáció:** A specifikáció és a technikai dokumentáció vázlatának elkészítése LLM segítségével.
4. **Hibakeresés:** Stack trace elemzése és hibajavítási javaslatok kérése MI asszisztenstől.

A felhasznált promptokat és eszközöket a projekt README.md fájljában naplózzuk.