`MealPlanr Lite`

L’idea è semplice ma realistica: creare un’applicazione in **Next.js** che vi permetta di **cercare ricette**, vedere i dettagli, e organizzare un **piano settimanale dei pasti**.

Obiettivo

L’obiettivo non è solo avere un’app funzionante, ma **esercitarvi con le principali caratteristiche di Next.js** e con le buone pratiche di un’app reale: fetch di dati da un’API, gestione dello stato, persistenza in `localStorage`, interfaccia chiara e con feedback all’utente.

Cosa andrete a realizzare

1. **Home page**
    - In alto ci sarà una **barra di ricerca** per scrivere il nome di una ricetta. La ricerca deve essere intelligente, quindi non chiamate l’API ad ogni singolo tasto, ma usate un **debounce** (per esempio 300–400 ms di pausa).
    - Ci sarà anche un **menu a tendina per filtrare le ricette per categoria** (es. “Pasta”, “Dessert”, ecc.), usando le categorie fornite dall’API.
    - I risultati verranno mostrati in una **griglia di card** con immagine, titolo e categoria. Ogni pagina mostrerà massimo 12 ricette, con pulsanti per andare avanti/indietro (paginazione client-side).
2. **Pagina di dettaglio `/recipes/[id]`**
    - Quando l’utente clicca una ricetta, si apre la pagina di dettaglio. Qui mostrerete:
        - Titolo e immagine principale.
        - Categoria e area geografica della ricetta.
        - Lista degli **ingredienti con quantità**.
        - Istruzioni passo passo.
    - Sulla stessa pagina ci saranno due pulsanti:
        - **Aggiungi ai preferiti** → salva l’ID della ricetta in `localStorage`.
        - **Aggiungi al piano settimanale** → l’utente sceglie un giorno e se inserirla a **Pranzo o Cena**, e la ricetta viene salvata nel piano settimanale (sempre in `localStorage`).
3. **Piano settimanale `/plan`**
    - Una tabella con 7 colonne (Lunedì–Domenica) e 2 righe (Pranzo, Cena).
    - Ogni cella può contenere una ricetta. L’utente può rimuovere o cambiare una ricetta in qualsiasi momento.

- Il piano rimane salvato in `localStorage`, quindi anche aggiornando la pagina o tornando il giorno dopo resta tutto com'era.

API che userete

Non serve nessuna chiave API. Usiamo TheMealDB:

- Ricerca: `https://www.themealdb.com/api/json/v1/1/search.php?s=pasta`
- Dettaglio: `https://www.themealdb.com/api/json/v1/1/lookup.php?i=52772`
- Categorie: `https://www.themealdb.com/api/json/v1/1/list.php?c=list`

Dovrete imparare a "normalizzare" i dati: ad esempio gli ingredienti arrivano in campi numerati (`strIngredient1..20`). Servirà un piccolo ciclo per estrarli in un array più comodo da usare.

Persistenza locale

Per non perdere i dati, userete **`localStorage`**:

- Un array con gli ID delle ricette preferite.
- Una struttura a oggetto per il piano settimanale (giorno → { lunch, dinner }).

In questo modo, aggiornando la pagina o tornando più tardi, i dati saranno ancora lì.

Indicazioni di lavoro

- Usate **Server Components** per il fetch dei dati (quando basta mostrare dati).
- Usate **Client Components** (`"use client"`) solo quando c'è interattività: la barra di ricerca, i pulsanti dei preferiti, il piano settimanale.
- Create un file `lib/meals.js` per tutte le funzioni di fetch (così se cambiamo API, il resto del codice non cambia).
- Usate un file `lib/storage.js` per le funzioni di lettura/scrittura in localStorage.
- Gestite loading ed errori con `loading.js` ed `error.js`.

Extra (facoltativi, per chi vuole mettersi alla prova)

- Creare una pagina `/favorites` che mostra tutte le ricette salvate come preferite.
- Implementare una **dark mode** salvata in `localStorage`.
- Aggiungere un pulsante "Copia lista piano" che mette negli appunti tutte le ricette pianificate.

Conclusione

Alla fine avrete un’app che:

- Vi permette di scoprire nuove ricette,
- Organizzare un piano settimanale dei pasti,
- Salvare preferiti e piano anche dopo il refresh,
- Ha un’interfaccia chiara con ricerca, paginazione e gestione errori.

È un progetto “vero”, che potrebbe tranquillamente stare come demo su un portfolio.

Deploy su Vercel

Al termine dello sviluppo, l’app va messa online con **Vercel**:

1. **Mettere il progetto su GitHub**

   ```
   git init
   git add .
   git commit -m "First commit MealPlanr Lite"
   git branch -M main
   git remote add origin https://github.com/tuo-utente/mealplanr-lite.git
   git push -u origin main
   ```

2. **Accedere a vercel.com** (login con GitHub).
   - Cliccare **Add New Project → Import repository**.
   - Selezionare il repo.
   - Vercel riconosce Next.js automaticamente.
3. **Deploy**
   - Premere **Deploy**.
   - Dopo pochi secondi si avrà un link pubblico tipo:

     `https://mealplanr-lite.vercel.app`

Consegna finale

- Repository su GitHub con il codice.
- Link all’app online su Vercel.
- Tutte le funzionalità MVP completate e testate.
- Extra facoltativi se volete migliorare il progetto e mostrare creatività.