Working with AI– Recipe Ideas Application

1. Understanding the Problem

The goal was to build a simple application that fetches and displays recipe ideas using a free public API.

The user need is straightforward:

>A user wants to quickly search for recipes and get suggestions.

>The app should be lightweight, fast, and easy to use.

So I extracted the following key requirements:

>Input box for searching recipes.

>Fetch recipe ideas from a free API (no auth).

>Display them with essential details (name, ingredients, maybe an image).

>Keep it responsive and user-friendly.

2. My Approach & Thought Process

Before starting development, I broke down the solution into steps:

1. Framework - I selected React since I'm comfortable with it and it's widely used for building modern apps.
2. Styling - I chose bootstrap.
3. API - I used a public recipe API as given in task, since it provides recipes without requiring signup or keys.
4. State Management - React's built-in useState and useEffect were enough for handling API data.
5. App Flow:

>User enters a search query like "chicken".

>API returns a list of recipes.

>Recipes are displayed with images, names, and links for more details.

3. Working with AI

I used AI (Google AI Studio) to guide me at different stages:

>Getting clarity on which free recipe APIs I could use without registration.

>Understanding how to fetch API data in React with fetch() and useEffect.

>Structuring the app into components (App.jsx, RecipeList.jsx, RecipeCard.jsx).

>Improving the UI/UX design ideas (card layout, search bar placement, responsive grid).

>Handling edge cases like no results found or invalid input.

4. Planned Design and Flow

>App.jsx - main entry point, includes search bar and renders results.

>RecipeList.jsx - handles rendering all recipe results.

>RecipeCard.jsx - displays each recipe with image + name.

>Search Functionality - connected to API calls.

>Responsive Layout - Bootstrap.

5. Example Code Snippet

```jsx
const [query, setQuery] = useState("");
const [recipes, setRecipes] = useState([]);
async function fetchRecipes() { try {
const res = await fetch(
`https://www.themealdb.com/api/json/v1/1/search.php?s=${query}`
);
const data = await res.json();
setRecipes(data.meals || []);
} catch (error) {
console.error("Error fetching recipes:", error);
}
}
```

6. Reflection

Working with AI gave me confidence to take the right approach.
Instead of spending hours figuring out API options or UI patterns, I could quickly check for ideas and get working code snippets.
It also helped me think about deployment on CodeSandbox, StackBlitz and vercel
Overall, AI made the process more efficient and structured.