🛍️ Task 10: Product Search Based on User Preferences

In this task, you'll learn how to build a **product filtering system** using **OpenAI's function calling** capabilities. Instead of manually writing filtering logic, you'll define a function schema, describe user preferences, and let the model invoke the function with the appropriate structured arguments.

📚 Theory

Traditional product filtering relies on **conditional logic written in code**. But with **OpenAI's function calling** [documentation], you can hand off this logic to the model.

Function calling lets the model decide **when and how to invoke a predefined function**, using structured reasoning over both **user preferences** and **JSON-formatted data**. You define the function signature, and the model fills in the arguments based on **natural language input** and **dataset context**.

This approach allows for **flexible, natural filtering behavior**—while keeping full control over **structure, safety**, and **post-processing**.

🧠 AI Technique

This challenge combines several AI techniques:

- **OpenAI Function Calling** – Let the model select matching products and call your function with structured arguments
- **Natural Language-to-Structure Conversion** – Translate input like "under $200 and in stock" into clean, typed JSON

- **Reasoning over Datasets** – Guide the model to extract relevant products from a JSON dataset based on user intent

---

📝 Task

Introduction

User Preferences: Inputs provided by the user that define the filtering criteria for a search query. Examples include:

- Maximum price
- Minimum rating
- Specific product categories (Electronics, Fitness, Kitchen, Books, Clothing)
- Stock availability

Filtering Logic: The mechanism or algorithm used to narrow down a dataset based on user-defined conditions. In this task, filtering logic is performed by leveraging:

- Hardcoded datasets (products.json) and queries (user input)
- OpenAI API for natural language filtering

Dataset: A structured collection of data in **JSON format** used as the source for product searches. Each item typically contains attributes like:

- name
- category
- price
- rating
- in_stock

Your task is to create a **console-based product search tool** that:

- Accepts user preferences (e.g., category, max price, min rating, in-stock status) in **natural language**

e.g., "I need a smartphone under $800 with a great camera and long battery life"

- Calls the **OpenAI API using** function calling
- Searches the given dataset for requested items
- Returns the final **filtered product list** in structured format (see example below)

📦 Use the dataset from products.json as an input parameter for this task.

💬 Example Response

Filtered Products:

1. Wireless Headphones – $99.99, Rating: 4.5, In Stock
2. Smart Watch – $199.99, Rating: 4.6, In Stock

⚠️ You must use **OpenAI's function calling** to extract and return matching products. **Manual filtering logic is not allowed.**

📌 Requirements

- The app must include **calls to OpenAI API**
- The app must use the **function calling mechanism**
- The app must accept **input from the console in natural language**
- README.md should contain **clear and detailed instructions** on how to run the application
- sample_outputs.md must include at least **two sample runs** of your application for different user requests
- Output must be **clear, properly formatted**, and align with all the requirements stated in task description