**Product Autocomplete Backend – Design Documentation**

**API Structure and Endpoints**

The backend exposes a REST API at /api/products/search that supports product autocomplete for a search feature.

* **Endpoint:** GET /api/products/search
* **Query Parameters:**
  + q (string, required): Search query (minimum 2 characters)
  + limit (integer, default 10, max 50): Number of results per page
  + skip (integer, default 0): Number of results to skip (for pagination)
* **Behavior:** Returns a paginated list of products where the title or brand contains the query string ( This is case-insensitive).

**Data Schema**

The products are seeded from [DummyJSON](https://dummyjson.com/docs/products) and stored in a local JSON file(products.json). Each product includes:

* id: integer
* title: string
* brand: string
* category: string
* price: number

**Pagination Logic and Edge Cases**

* Uses limit and skip to paginate the results after filtering and scoring.
* If skip exceeds the number of matching products, returns an empty list.
* If q is missing or too short, returns 400 with a descriptive error.
* If no matches, returns an empty array and total: 0.

**Assumptions and Limitations**

* The backend uses a static JSON file as its data source (no live DB).
* Only title and brand fields are searched.
* No authentication or rate-limiting.

**Bonus: Scoring and Ranking Logic**

* Results are scored to improve relevance:
  1. Products whose **title starts with** the query (highest score)
  2. Products whose **brand starts with** the query
  3. Products where the query is a substring of the title
  4. Products where the query is a substring of the brand
* Within each group, the results are sorted alphabetically by title.