

Take-Home Exercise (Backend Intern/Fresher)

Note: Please do not use AI coding agents to generate the solution. You may refer to the internet for documentation/examples.

You can use **any language** to code this.

Problem Statement

At **Streamoid**, we work with many online sellers who want to list their products on marketplaces (like Myntra, Flipkart, Amazon).

These sellers usually maintain their product catalog in a **CSV/Excel file** with details like:

- SKU (unique product code)
- Name (product title)
- Brand
- Color, Size
- MRP, Price
- Quantity

But sellers often want to **view, filter, and validate** their product data before sending it to marketplaces.

Your task is to build a **backend service** that simulates this process:

- Upload a **CSV file of products**.
- Parse the file and validate each row (check for missing/invalid values).
- Store valid products in a database.

- Provide APIs to **list all products** and **search/filter products**.

This exercise will test your ability to:

- Parse CSV files
- Validate data
- Design simple REST APIs
- Work with a database

Product Data (CSV File) example:

sku,name,brand,color,size,mrp,price,quantity

TSHIRT-RED-001,Classic Cotton T-Shirt,StreamThreads,Red,M,799,499,20
TSHIRT-BLK-002,Classic Cotton T-Shirt,StreamThreads,Black,L,799,549,12
POLO-GRN-003,Heritage Polo,StreamThreads,Green,XL,1299,999,8
JEANS-BLU-032,Slim Fit Jeans,DenimWorks,Blue,32,1999,1599,15
JEANS-BLK-030,Slim Fit Jeans,DenimWorks,Black,30,1999,1499,18
DRESS-PNK-S,Floral Summer Dress,BloomWear,Pink,S,2499,2199,10
DRESS-YLW-M,Floral Summer Dress,BloomWear,Yellow,M,2499,1999,7
SHOE-WHT-7,Everyday Sneakers,StrideLab,White,UK7,2999,2499,25
SHOE-NVY-8,Everyday Sneakers,StrideLab,Navy,UK8,2999,2499,19
BAG-TOTE-BEI,Canvas Tote Bag,CarryCo,Beige,OneSize,899,699,35
BELT-BRN-38,Leather Belt,CarryCo,Brown,38,1199,899,40
SAREE-RED-001,Banarasi Silk Saree,Ethniq,Red,Free,6999,5999,5
KURTA-BLU-M,Cotton Kurta,Ethniq,Blue,M,1599,1299,22
JKT-OLV-L,Utility Jacket,UrbanEdge,Olive,L,3499,2999,6
TSHIRT-GRY-S,Graphic Tee,UrbanEdge,Grey,S,899,699,30
TSHIRT-WHT-XS,Graphic Tee,UrbanEdge,White,XS,899,649,14
SHIRT-CHK-M,Checked Casual Shirt,ButtonUp,Multi,M,1799,1399,16

SHIRT-PLN-L,Plain Oxford Shirt,ButtonUp,Blue,L,1899,1499,12

HOODIE-CHR-XL,Cozy Hoodie,SnugWear,Charcoal,XL,2199,1799,11

HOODIE-CRM-M,Cozy Hoodie,SnugWear,Cream,M,2199,1699,9

Requirements

1. CSV Upload & Storage

- **API:** POST /upload
- Accept a CSV file.
- Parse and validate rows. Rules:
 - price must be \leq mrp
 - quantity ≥ 0
 - All required fields (sku, name, brand, mrp, price) must be present.
- Store valid rows in a database (SQLite or Postgres).

2. List Products

- **API:** GET /products
- Returns all stored products with pagination support (page, limit).

3. Search Products

- **API:** GET /products/search
- Filters:
 - By brand \rightarrow /products/search?brand=StreamThreads
 - By color \rightarrow /products/search?color=Red
 - By price range \rightarrow /products/search?minPrice=500&maxPrice=2000

Deliverables

- A **GitHub repo** containing:
 - **Source code (Backend APIs)**
 - **README.md** with:
 - API documentation (endpoints, sample requests/responses)
 - Setup and testing instructions

Evaluation Criteria

- Correctness of APIs (upload, list, search)
- Clean, modular, and testable code.
- ★ Bonus:
 - Unit tests for CSV parsing, validation, and search filters
 - Dockerized solution (easy to run in any environment)

Example API Flows

Upload CSV

API

curl -X POST -F "file=@products.csv" <http://localhost:8000/upload>

Response

```
{  
  "stored": 3,  
  "failed": []  
}
```

List Products

API

GET /products

Response

```
[
  {
    "sku": "TSHIRT-RED-001",
    "name": "Classic Cotton T-Shirt",
    "brand": "StreamThreads",
    "color": "Red",
    "size": "M",
    "mrp": 799,
    "price": 499,
    "quantity": 20
  },
  {
    "sku": "JEANS-BLU-032",
    "name": "Slim Fit Jeans",
    "brand": "DenimWorks",
    "color": "Blue",
    "size": "32",
    "mrp": 1999,
    "price": 1599,
    "quantity": 15
  }
]
```

Search Products

API

GET /products/search?brand=BloomWear&maxPrice=2500

Response

```
[
  {
    "sku": "DRESS-PNK-S",
    "name": "Floral Summer Dress",
    "brand": "BloomWear",
    "color": "Pink",
    "size": "S",
    "mrp": 2499,
    "price": 2199,
    "quantity": 10
  }
]
```