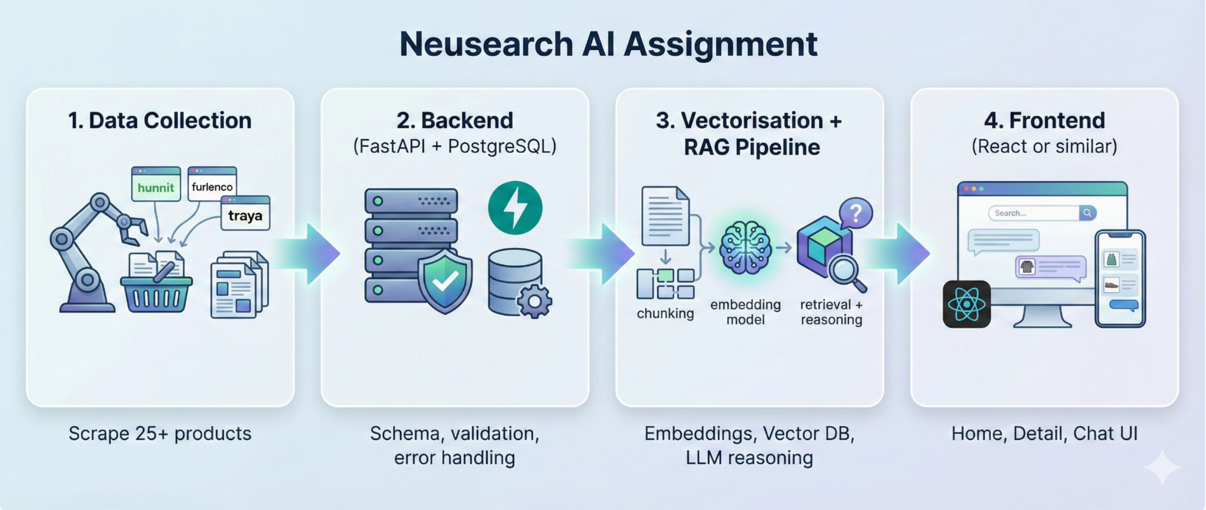
Neusearch AI

AI Engineering Intern - Technical Assignment

Overview

Your task is to build a **mini AI-powered product discovery assistant** that can recommend the right products based on open-ended and abstract user queries. In order to build this, you will have to scrape product data, store it in a database, vectorize it and use retrieval + reasoning.



This assignment evaluates how well you can:

- build end-to-end systems
- work with external APIs and vector databases
- structure data pipelines
- think proactively and product-first
- document your work like a real engineer

The goal is not perfection - but clarity, ownership and a working project.

What You Will Build

1. Data Collection Pipeline

- Scrape product data from **ONE of the 3 websites** below -
    1. hunnit.com
    2. furlenco.com
    3. Traya.health
- Scrape a minimum of **25 products** and store them on a PostgreSQL DB
- Each product/SKU scraped must include the following properties:
    - Title
    - Price
    - Description
    - Features / attributes
    - Image URL
    - Category
    - Any other properties you can extract
- Focus on clean, consistent and reliable data.

Note : Feel free to use any 3rd party scraper API services that you find online, in order to achieve this. It should be implemented as a FastAPI service.

2. Backend (FastAPI + PostgreSQL)

Build **Python FastAPI** backend with **PostgreSQL** as a database, that houses the below vectorisation and RAG architecture. Your backend should include:

- Schema design
- Input validation
- Error handling
- Clean code structure

3. Vectorisation + RAG Pipeline

Build a retrieval pipeline that powers the chatbot:

- Chunk product data
- Generate embeddings using any open-source or hosted model
- Store vectors in a vector DB (Chroma, PgVector, Pinecone, Weaviate, etc.)
- Retrieve relevant products for any user query
- Use an LLM (OpenAI, Gemini, HuggingFace models, etc.) to:
    - interpret abstract queries
    - ask clarifying questions when needed
    - recommend products with explanations

The bot should be able to handle abstract and nuanced queries like:
 - *“Looking for something I can wear in the gym and also in meetings.”*

- *“Looking to rent furniture for my 2bhk apartment.”*
- *“I have a dry scalp. What products can improve my hair density?”*

This is the core of the assignment.

4. Frontend (React or similar)

Build a **SIMPLE** ecommerce website UI consisting of:

A. **Home Page**
   - list all scraped products
   - basic grid or list view
   - fetch from backend API

B. **Product Detail Page**
   - product title, price, features, images
   - URL routing required

C. **Chat Interface**
   - message bubbles
   - display product cards when the bot recommends items

Note : Prioritize a basic clean UI/UX and usability over aesthetics.

5. Deployment

Deploy both the frontend and backend online using any stack of your choice:

- Render
- Railway
- Fly.io
- Vercel + Supabase
- AWS Lightsail
- DigitalOcean
- Any other platform of your choice

Bonus points for:

- Docker configuration
- environment variables
- production-like setup

Submission Requirements

Submit the following by filling up **this google form**:

1. **A link to a the public GitHub repository** (both frontend + backend)
    a. The Github must contain a **README** that explains:
        - how to run the project locally
        - architecture and decisions
        - scraping approach
        - RAG pipeline design
        - challenges + trade-offs
        - *[Bonus]* A short note on what improvements you'd make to the submitted project if you had more time
2. **Link to the live deployed project**
3. **A link to a short Loom video demo (2-3mins)**
    Show the product in action and walk through

Evaluation Criteria

Technical Skills (50%)

- scraping quality
- backend structure
- vectorisation + RAG accuracy
- integration of APIs/libraries
- deployment completeness

Product Thinking (20%)

- quality of assumptions
- relevance of recommendations
- clarity of chatbot flow

Ownership & Proactivity (20%)

- documentation quality
- extra effort beyond instructions
- handling edge cases
- initiative in design

Communication (10%)

- clarity in README
- clarity in Loom walkthrough

Timeline for submission

You will have **4 days** from the moment you receive this assignment.

If you require any clarification, feel free to contact me at rahul@neusearch.ai.