

SpecSense AI – Project Specification Document

1. Project Overview

SpecSense AI is a modular AI-driven product recommendation system designed to interpret natural language user queries and generate structured, explainable product recommendations.

The system bridges the gap between free-form user preferences and structured product catalog data to deliver intelligent, transparent, and user-aligned recommendations.

2. System Objective

To build an AI system capable of:

- Understanding free-form product requirement queries
 - Extracting structured constraints and soft preferences
 - Matching user intent against catalog data
 - Providing explainable and transparent recommendations
-

3. Input Data Requirements

3.1 Product Catalog Data

Each product in the catalog must include:

- **Name**
- **Specifications**

- **Category**
- **Description**

This structured dataset will serve as the searchable database for matching user intent.

3.2 User Profile Data

User data may include:

- **Age**
- **Budget**
- **Context Tags** (e.g., travel, gaming, office use)
- **Product History**

This profile enhances personalization and recommendation accuracy.

4. Core Functionalities

4.1 Natural Language Understanding (NLU)

The system must:

- Accept free-form product preference queries
- Example:

“Need a lightweight laptop under ₹30k for travel and battery life.”

- Interpret user intent without requiring structured input
-

4.2 Constraint Extraction

The system must extract:

A. Structured Constraints (Hard Requirements)

- Budget (e.g., under ₹30,000)
- Weight limits
- Product category
- Intended purpose

B. Soft Preferences

- Long battery life
 - Travel-friendly
 - Performance expectations
 - Design preferences
-

4.3 Intent-to-Catalog Matching

The system should:

- Map extracted constraints to structured catalog fields
 - Filter products based on hard constraints
 - Rank products based on soft preference alignment
-

4.4 Explainable Recommendation Engine

For each recommended product, the system must clearly explain:

1. **Criteria That Were Fully Met**

- Example: Budget constraint satisfied
- Weight within desired range

2. **Trade-offs**

- Example: Slightly lower battery life than requested
- Heavier than ideal but within acceptable range

3. **Partially Fulfilled Soft Preferences**

- Example: Moderate battery life instead of long battery life

This transparency builds trust and improves user satisfaction.

5. **System Architecture (High-Level)**

SpecSense AI should be modular and include:

1. **Input Processing Module**

- Handles user query intake
- Preprocessing and normalization

2. **Intent & Constraint Extraction Module**

- NLP-based parsing
- Structured constraint extraction

3. **Matching Engine**

- Hard constraint filtering
- Soft preference scoring

4. Recommendation Explanation Module

- Generates human-readable breakdown
 - Highlights matches and trade-offs
-

6. Expected Output Format

For each recommendation:

- Product Name
 - Key Specifications
 - Why it was recommended
 - What constraints were satisfied
 - Any trade-offs
 - Alignment with soft preferences
-

7. Key System Characteristics

- Modular design
- Explainable AI approach
- Personalization-ready
- Structured + unstructured data integration
- Transparent decision logic

Design:

User Query



LLM → Extract Requirements



Python → Filter CSV



Python → Rank



LLM → Generate Explanation



Structured Response

Flow

User Query



RequirementParser → LLM → structured JSON



FilterEngine → Python → filtered products



RecommendationExplainer.build_prompt()
(just builds text string)



RecommendationExplainer.explain()
→ LLM generates explanation

We use LLM first to understand intent.

We use Python to deterministically choose top 5 products.

We use LLM again only to explain those choices.