# SpecSense AI: Problem Statement

**Overview:** In digital commerce and service ecosystems, matching users to the right products is a critical driver of engagement and retention. While traditional recommendation systems rely heavily on rigid filters or historical behavior, they often fail to capture the real-time context and intent behind a user's query.

SpecSense AI bridges this gap by leveraging Generative AI to process a user's natural language preferences and context and intelligently match them to the most suitable product(s) from a company-provided catalog. Instead of overwhelming users with irrelevant choices, filters, or context-unaware, explainable recommendations based only on limited product features and customer profile metadata.

This solution is especially valuable for enterprises that already possess clean product and customer datasets and want to enhance personalization without redesigning their entire platform logic.

SpecSense AI solves this by building a Generative AI-based system that:

- Understands a user's free-form product preference query.
- Matches it with structured company-provided product data.
- Generates explainable, insight-driven product recommendations.

---

## Problem Context:

Most recommendation systems struggle to support:

- **Flexible Input:** Users can't express needs like "I want something light, with great battery and under $30K."
- **Transparent Recommendations:** Why a product was recommended isn't made clear.
- **Hybrid Filtering:** Combining structured product specs with soft preferences (e.g., "good to use in low-light" or "eco-friendly").

Companies have the data but lack the intelligence layer to connect the data using natural language understanding, real-time context, and personalized matching.

This leads to:

- Customer frustration or drop-offs
- Distrust due to irrelevant recommendations
- Misalignment with true user intent

---

## SpecSense AI helps businesses:

- Deliver human-like, conversational product matching
- Make product discovery faster and more trusted
- Convert vague user intent into concrete decisions
- Reduce cognitive load on customers

---

## Data:

- **Product Catalog:** Structured CSV/JSON with name, specs, category, description
- **User Profile:** Age, budget, context tags, product history

---

## What SpecSense AI Does:

SpecSense AI is a modular system that:

1. Understands free-form product preference queries (e.g., “Need a lightweight laptop under $25k for travel and battery life”)
2. Extracts structural constraints (e.g., budget, weight, purpose) and soft preferences (e.g., long battery life, eco-friendly).
3. Matches user intent with structured catalog data

4. Generates explainable product recommendations, clearly breaking down:
    - What criteria were met
    - Where trade-offs exist
    - What soft preferences were partially fulfilled
5. Highlights gaps (if no perfect match is found, and suggests the closest alternatives with rationale).
6. Enriches results with real-world insights, such as customer review snippets, availability status, and current offers.
7. Evaluates recommendation quality, explainability, and user satisfaction for continuous improvement.

---

## User Flow:

1. **User Query Input (Natural Language)**
    - e.g., “Looking for a lightweight laptop under $30k, good for coding and battery backup.”
2. **Intent Understanding & Constraint Extraction**
    - Extract structured constraints (e.g., budget, specs)
    - Soft needs: “good for coding”, “battery backup”
3. **Matching Engine**
    - Match structured data (e.g., specs like RAM, battery life)
    - Soft match using LLMs (e.g., associate “good for coding” with CPU + RAM + reviews)
4. **Recommendation + Transparency Layer**
    - For each product, show:

- Hard matches
- Soft matches
- Trade-offs
- Unmet needs

5. **Gap Detection Layer (if needed)**
   - “No products meet all criteria. Showing best alternatives…”
6. **Enrichment with Real-World Insights**
   - Add relevant customer review snippets
   - Mention availability/offers (if data exists)
7. **Evaluation Layer**
   - For testing: total quality of match, correctness of rationale, and explainability

---

## Technical Requirements:

- Python 3.12+
- Gemini API (Google Generative AI) or OpenAI API (optional)
- Pandas (for parsing and structuring reviews)
- Streamlit (for interactive UI)
- Python-dotenv (for environment config)
- Pytest (for scenario validation)

---

## By solving a real-world consumer problem, this project gives learners the skills to:

- Design practical LLM pipelines
- Apply advanced prompting techniques
- Engineer products for interpretability and usability
- Build scalable, modular systems for real-world deployment