International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 03 | March - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

DecisionBuddy – AI-Powered Product Comparison with Real-Time Data: A Technical Deep Dive

Yogesh Dutt, Suman Agrawal

ABSTRACT

This paper explores the development and implementation of *DecisionBuddy*, an AI-powered product comparison engine that leverages real-time data scraping, Natural Language Understanding (NLU), and Generative AI (GenAI). Aimed at overcoming decision fatigue in e-commerce and product selection, the platform demonstrates a scalable architecture and efficient query-to-recommendation pipeline. We provide an end-to-end technical breakdown of its architecture, implementation challenges, intelligent query parsing, and GPT-powered summarization process. A working Proof of Concept (POC) validates the system's efficiency and accuracy.

Live Url : http://decissionbuddy.yogeshdutt.in/

Keywords: Product Comparison, Real-Time Scraping, GPT-4, Natural Language Understanding, AI Recommendations

---

1. Introduction

Modern consumers are inundated with options while shopping online, often struggling with inconsistent reviews and fragmented product data. DecisionBuddy was built to simplify this experience by converting natural language product queries into structured searches, fetching real-time product data, and using GenAI to deliver concise, tailored recommendations. This paper presents the core technologies and implementation strategy that power the DecisionBuddy system.

---

2. Core Challenges

2.1 Data Fragmentation

• Product details and user reviews exist in diverse formats across marketplaces (e.g., Flipkart, Amazon).
• No unified structure for comparison—static tables and affiliate content dominate existing platforms.

2.2 Real-Time Data Extraction

• Lack of standardized APIs across e-commerce platforms.
• Headless browser scraping is required to ensure updated results.

2.3 Natural Language Complexity

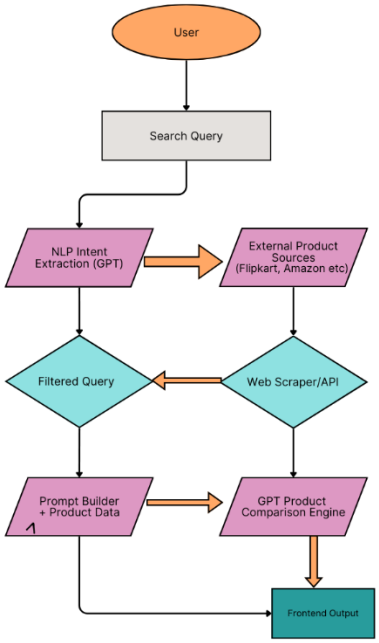
• User queries are unstructured, with implicit preferences (e.g., "cheap phone with good camera").
• Requires intent parsing, entity extraction, and context understanding.

2.4 Personalized Output Generation

• Mapping scraped product specs to user intent requires structured representations.
• Summarization must balance objectivity with personalization.

---

3. System Architecture and Implementation



3.1 End-to-End Flow

1. **User Query Parsing**
   Using GPT-4-based prompting, the user's natural language input is transformed into structured parameters.
2. Input: "Best phone under 20000 with great camera"
3. Output:

```
{
  "category": "smartphone",
  "budget": 20000,
  "priority": ["camera"]
}
```

4. **Real-Time Data Collection**
   Headless scraping (Puppeteer) or APIs fetch live product listings and user reviews.
5. **Information Extraction from Reviews**
   GPT models structure reviews by mapping sentiment to specific features (battery, camera, performance).

6. **Recommendation Generation**
   GPT compares multiple products using structured data and provides a human-like, context-aware recommendation.

---

4. Step-by-Step Implementation

Step 1: Parse Natural Language Query

Utilizes GPT-4 in a zero-shot or few-shot setup for intent detection and entity extraction. This bridges human input and machine-readable search parameters.

Step 2: Scrape Product Data

Puppeteer scrapes top results from platforms like Flipkart or Amazon. Selectors extract:

- Product title
- Price
- Feature list
- User reviews

Step 3: Structure User Reviews

Example:

Input: “Battery lasts all day, camera is average at night.”

Output:

```
{
  "battery": "excellent",
  "camera": "average night, good daylight",
  "sentiment": "mixed"
}
```

Step 4: AI Recommendation Prompt

Products are compared based on structured features and summarized in conversational language by GPT.

Step 5: Summarized Output to User

“Redmi Note 12 offers the best camera quality under ₹20,000 and strong battery performance, making it ideal for your needs.”

---

5. Technology Stack

Layer	Technology Used
Frontend UI	React.js
Backend API	Node.js + Express
Database	MongoDB
Scraping Engine	Puppeteer + Cheerio
GenAI Engine	OpenAI GPT-4
Hosting	Vercel, Render, AWS

---

6. API Design

- POST /query
  Accepts user query string
  → { "query": "Phone under 20000 with good camera" }
- POST /parse-query
  Returns structured parameters
  → { "category": "smartphone", "budget": 20000, "priority": ["camera"] }
- GET /fetch-products
  Scrapes product data
  → Returns array of matching products
- POST /summarize-products
  GPT-generated recommendation summary
  → { "summary": "Redmi Note 12 is recommended..." }

---

7. Proof of Concept and Validation

- ✓ Natural language query parsed with high accuracy
- ✓ Live data scraped from Flipkart
- ✓ GPT-4 generated meaningful comparisons
- ✓ Total response time ~7 seconds

---

7. Key Outcomes & Differentiators

Feature	Traditional Platforms	DecisionBuddy
Real-Time Data	Static	Live Scraping
Personalized Comparison	Generic	User-Specific
Natural Language Input	No Support	Fully Supported
AI-Generated Summary	Manual/None	GPT-Driven
Structured Review Mapping	Unstructured	GPT-Based

9. Future Enhancements

- Browser extension for instant overlays on shopping sites
- Voice-based assistant for hands-free use
- Preference learning via user feedback loop
- Multilingual support for regional personalization

---

10. Conclusion

DecisionBuddy integrates cutting-edge GenAI with real-time web scraping to eliminate decision fatigue. By transforming user intent into structured, explainable, and contextual product recommendations, it redefines how consumers navigate choices in e-commerce and beyond. The POC confirms its feasibility, scalability, and relevance in today's AI-driven decision support landscape.
