

# **Project Canvas prepared by Kevin Selig October 05, 2025**

## **1. Problem Statement**

The project aims to build an intelligent Amazon Shopping Assistant that helps users efficiently locate and order products from a vast inventory. Traditional keyword search methods often fail to capture nuanced queries, leading to user frustration and time loss. By integrating AI-driven retrieval and reasoning, this assistant will understand intent, suggest relevant items, and consider geographic stock availability. Without such a solution, both customers and distribution networks face inefficiencies in product discovery and fulfillment accuracy.

## **2. Data & Knowledge**

The system will use Amazon's publicly available electronics dataset (<https://amazon-reviews-2023.github.io/main.html>), which includes product metadata and customer reviews. Data will be filtered to focus on high-volume and recent products. JSON files will be preprocessed to extract key fields such as title, description, category, rating, and location-related metadata. A Retrieval-Augmented Generation (RAG) pipeline will be developed using Qdrant for vector storage and retrieval, ensuring relevant and efficient query responses.

## **3. AI Approach & Methodology**

The assistant will use a LLM + RAG approach to combine general reasoning with accurate data retrieval. Open-source and cost-efficient models (e.g., from OpenAI, Groq, and Google) will power the language understanding, while embeddings from **Instructor** or similar models will drive semantic search. Development will use LangGraph or CrewAI to build multi-agent workflows capable of handling product discovery, summarization, and recommendation subtasks. The focus will be on minimizing API costs while maintaining robust and explainable results.

## **4. Performance Metrics & Evaluation Rules**

Success will be measured by three main metrics:

- Retrieval Precision: The percentage of relevant products in top results.
- Response Relevance Score: Human-rated accuracy of the assistant's recommendations.
- Latency: Average response time under 3 seconds for typical queries.  
Ongoing monitoring will use LangSmith for observability, ensuring consistency and detecting hallucinations or irrelevant results. Evaluation benchmarks will compare AI outputs against manual search baselines.

## 5. Resources & Stakeholders

The project team includes an AI engineer, data engineer, and UX developer all rolled into one. Required tools include Python, Streamlit, Docker, Qdrant, and access to LLM APIs. GitHub Actions will manage CI/CD, and AWS EKS and ECR will host and deploy services. The main stakeholder is the distribution network owner.

## 6. Risks & Mitigation

Key risks include large dataset processing overhead, potential API cost overruns, and hallucination in AI responses. To mitigate these, data will be subsetted early for scalability testing, fallback logic will route to cheaper models via LiteLLM, and GuardrailsAI will validate output formats. Ethical risks around bias in reviews will be addressed through balanced sampling and bias detection checks.

## 7. Deployment & Integration

The system will be containerized using Docker and deployed on AWS EKS for scalability. APIs will be documented for integration into existing logistics dashboards or e-commerce tools. Data flow will use standardized JSON and REST interfaces. Version control via GitHub will include rollback support, and Streamlit will serve as a lightweight demo UI. Future extensions could integrate MCP and A2A for agent-to-agent interoperability.

## 8. Timeline & Milestones (8 Weeks)

Sprint	Duration	Key Deliverables
<b>Sprint 0</b>	Week 1	Problem framing, AWS & Docker setup, dataset access
<b>Sprint 1</b>	Week 2	Initial RAG prototype
<b>Sprint 2</b>	Week 3	Improve retrieval quality & add context engineering
<b>Sprint 3</b>	Week 4	Agents and Agentic Systems
<b>Sprint 4</b>	Week 5	Upgrade to agentic RAG
<b>Sprint 5</b>	Week 6	Expand to multi-agent systems
<b>Sprint 6</b>	Week 7	Deployment, optimization and reliability build
<b>Sprint 7</b>	Week 8	Final build and delivery

## Conclusion

This eight-week project delivers a scalable, cost-efficient Amazon Shopping Assistant powered by agentic AI and retrieval-augmented reasoning. The approach balances cutting-edge methodologies with practical constraints, providing a strong foundation for future extensions into broader e-commerce intelligence systems.