

Northstar Smart Living AI Product Concierge

A Streamlit web app that combines RAG, ChromaDB, Ollama, and a ReAct agent to help customers compare smart-home products and understand policies.

MGS 636 Applied AI for Managers

**Presentation
Deck**



Customers need more than static product pages

Smart-home buyers often have questions that generic product pages cannot answer clearly.

1

Too many choices

Customers compare features, prices, bundles, and use cases across multiple smart-home devices.

2

Policy confusion

Return, warranty, delivery, and compatibility questions often need quick, trusted answers.

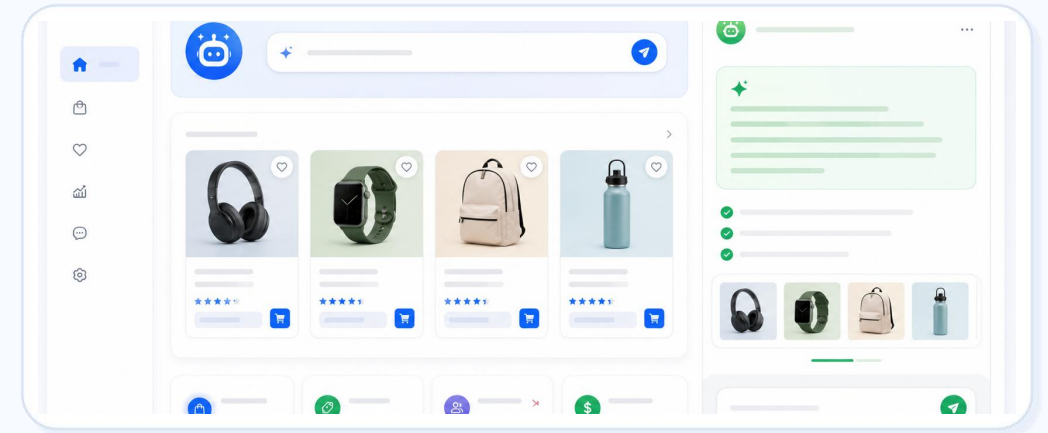
3

Support workload

Basic questions repeat again and again, taking time away from higher-value support tasks.

Opportunity

- Give personalized answers in normal language
- Use company documents instead of guessing
- Support product discovery and purchase confidence



AI concierge for smart-home product decisions

The app works like a helpful product assistant for Northstar customers.



What the assistant can do



Recommend products

Suggest devices based on customer needs and budget.



Compare options

Explain differences between products in simple language.



Answer policy questions

Use company policy documents for grounded answers.



Calculate prices

Apply quantities, discounts, and basic math tools.

RAG foundation: answers from company documents

Lab 4 is used to build a local knowledge base with ChromaDB and embeddings.



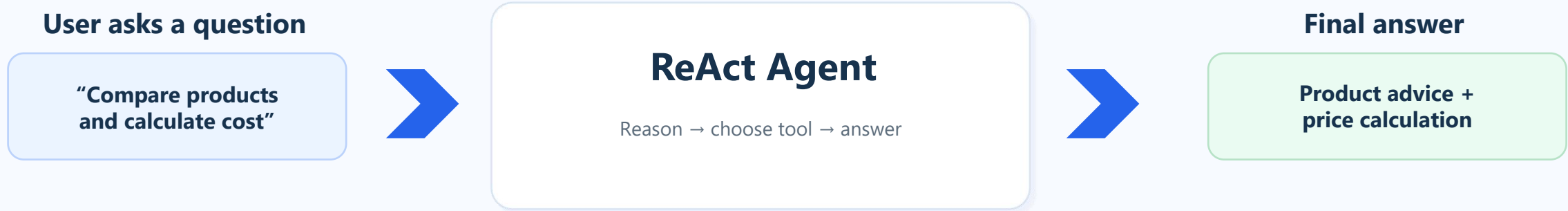
RAG flow in simple words

- Documents are loaded into the project
- Text is converted into embeddings
- Embeddings are stored in ChromaDB
- User questions retrieve the best matching content
- The LLM uses that content to answer

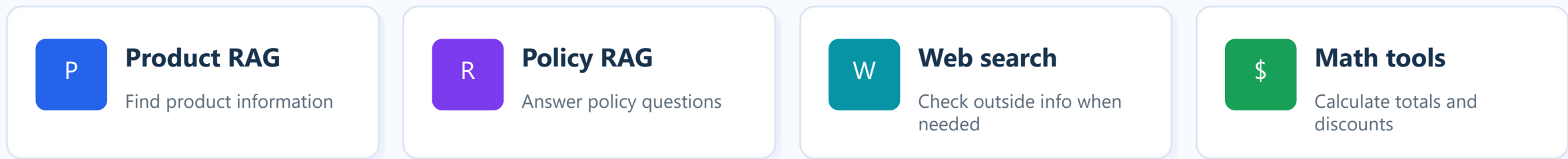
Key files: `ragcreate.py`, `ragrun.py`, `rag_utils.py`

ReAct agent: chooses the right tool for each question

Lab 5 adds an agent that can reason and use tools instead of giving only one direct answer.

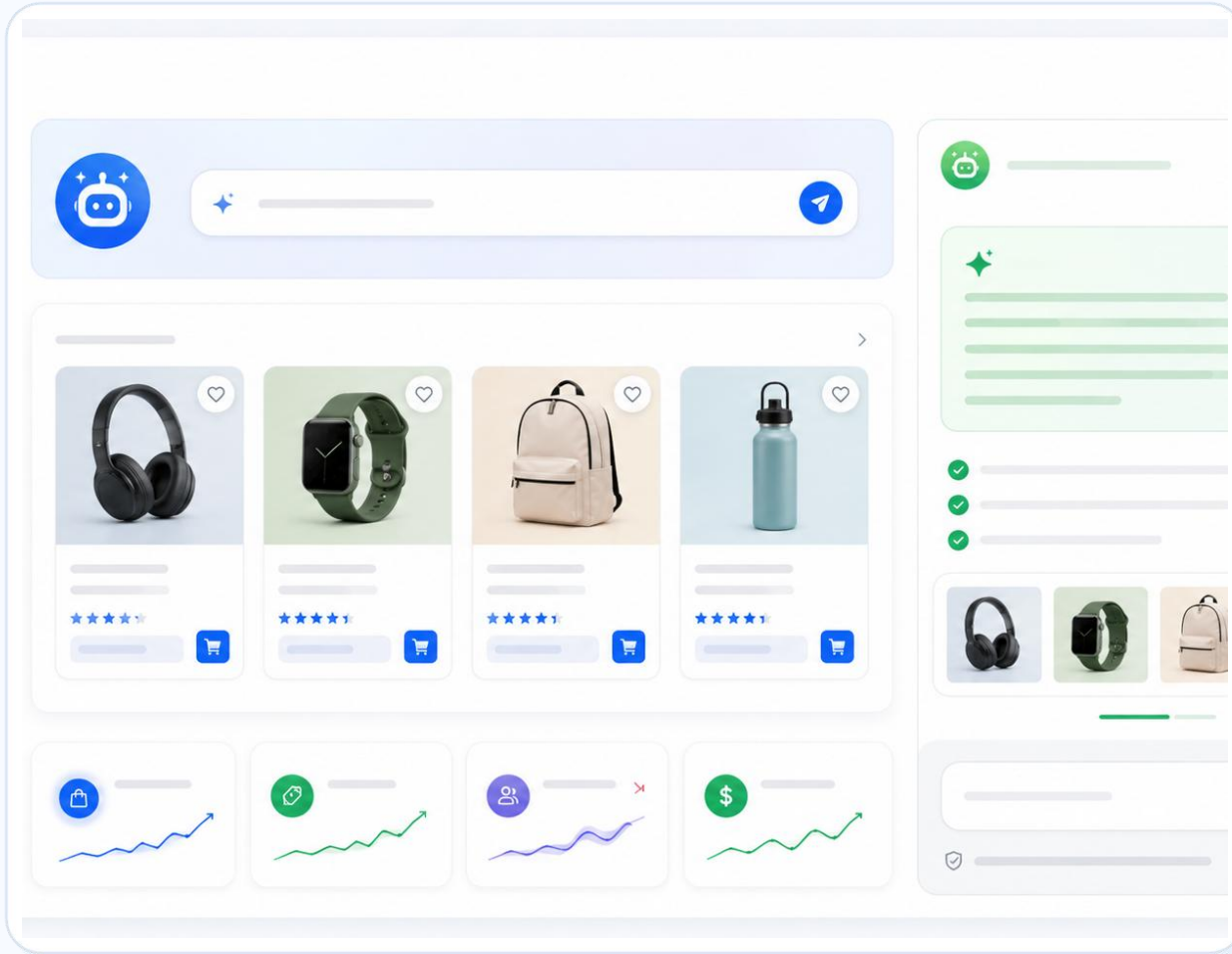


Agent tools



Streamlit turns the agent into a usable web app

The web app gives users a simple interface instead of running code manually.



Key file: webapp.py

1

Question box

Users can type product, policy, or pricing questions.

2

Sample prompts

Buttons help users test the system quickly during demo.

3

Conversation history

The app keeps previous questions and answers visible.

4

Transcript download

Users can download the session for reporting or review.

Demo plan: show three realistic customer questions

These demo prompts prove that the app can handle product, policy, and calculation tasks.

1

Product recommendation

“Which doorbell is best for deliveries under \$200?”

Shows product RAG

2

Policy answer

“What is the return policy for SafeEntry products?”

Shows policy RAG

3

Compare + calculate

“Compare Aura Mini Speaker Plus and Max, then calculate the cost for 3 devices.”

Shows agent + math tools

Demo tip: first explain the question, then show the app answer, then connect it back to Lab 4 or Lab 5.

Business value: faster answers and better buying confidence

The project is not only technical; it solves a real customer-support and product-discovery problem.

1

24/7 support

Customers get help any time without waiting for a support agent.

2

Better discovery

Users find products based on needs, budget, and use case.

3

Reduced workload

Common product and policy questions can be handled automatically.

4

Higher confidence

Clear comparisons help customers make decisions faster.

Main business impact: move from static browsing to interactive, document-grounded product support.

What we learned and what we would improve next

The project successfully connects RAG, tools, and a web interface, but there is room to improve.

What worked well

- RAG made answers more grounded
- Agent tools made the app more useful
- Streamlit made the project easy to demo
- Local setup helped us understand the full pipeline

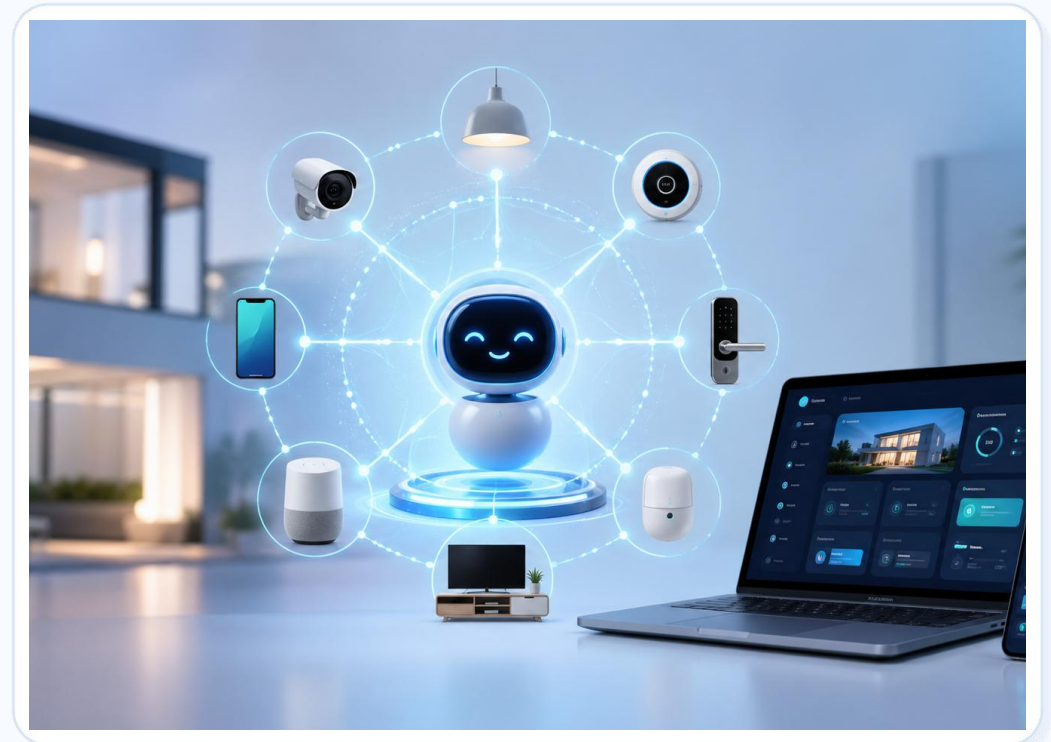
Next improvements

- Improve UI design and navigation
- Add more product and policy documents
- Add stronger error handling
- Test with more customer questions

Final takeaway

Northstar Smart Living AI Product Concierge shows how RAG, agents, tools, and a web app can work together in a real business use case.

- Lab 4: document-based RAG answers
- Lab 5: ReAct agent with tools
- Final app: Streamlit customer assistant



Thank you

Questions?