

ATLAS: AI-POWERED CUSTOMER SUPPORT AUTOMATION FOR E-COMMERCE

Presented by:

Chandana Rondla
Himaja Sree Thota
Suharsha Cheedala
Meetkumar Patel

Introduction

Problem
E-commerce support needs fast, accurate, and empathetic answers. Pure LLM chatbots can hallucinate order status, product facts, or policy promises. Rule-based bots are reliable but often feel rigid.

Atlas Solution

- LLM conversation plus deterministic business logic
- PostgreSQL stores order, customer, refund, stock, and message truth.
- Qdrant RAG searches about 1.4M Amazon products and company policies.
- Deterministic routing escalates frustrated customers to humans.

Requirements and Deliverables

- Real-time order lookup, refunds, account information, and policy/product retrieval.
- Identity checks before private order or account data is returned.
- Auditable human escalation when customer sentiment deteriorates.
- Constraints: FastAPI backend, PostgreSQL schema, Qdrant vector store, deterministic agent core, LLaMA adapters, and web UI.

Data Engineering

- Static data: Amazon Products 2023 listing and company policy CSV files.
- Dynamic data: Synthetic order conversations, orders, refunds, and accounts for testing.
- Data ingestion: In-house generator and deterministic dataset seed v0.046.
- Text is embedded with all-MiniLM-L6-v2 into 384-dimensional vectors.
- Qdrant uses cosine similarity; PostgreSQL enforces ACID transactional consistency.

Backend Safety Tools

- Lookup_order verifies customer ownership and returns ordertracking facts.
- Refunds_refund uses SQLAlchemy ORM, returns flag for idempotency.
- Search_products knowledge and search_policy knowledge ground answers in Qdrant.
- Escalate_to_human opens a support ticket when needed.

Methodology and Architecture

Atlas was developed with an evaluation-driven cycle: containerize the stack, seed relational data, ingest vector knowledge, connect the fine-tuned model, test support scenarios, and revise weak flows.



Runtime Flow



- Deterministic routing calls backend tools using regex and keyword detection.
- Verified tool outputs are injected into the prompt before the LLM responds.
- Feedback templates replace stalled or incomplete responses with safe fact-based answers.

Model Development

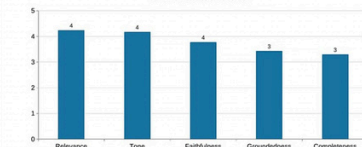
- Candidate architectures: Phi-4-mini, LLaMA, Qwen, and GPT4o.
- Fine-tuning used LLaMA-2-7B-Adapter for customer support language and tool-output routing.
- The primary end-to-end configuration is fine-tuned Phi-4-mini + RAG.
- RAG was selected because customer support cannot guess modular facts such as refund amount, policy terms, or tracking status.

Phi-4 Primary Model	384 Vector Dims	1.4M Product Rows
RAG Grounding	Docker Portable Stack	7 Use Cases

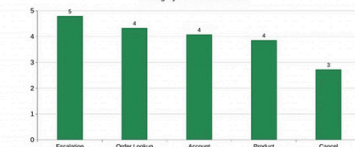
Analysis and Results

50 Test Cases	0.723 Task Success	0.191 ROUGE-L	3.79/5 Avg. G-Eval	93% Escalation Success
------------------	-----------------------	------------------	-----------------------	---------------------------

G-Eval Dimension Scores



Category-Level Performance



Key interpretation: escalation and order/account workflows are strong because backend facts are available and explicitly injected. Cancellation is the weakest category and should be improved through targeted retraining and stricter business-rule handling.

Demo and Visualization

- The chat response includes reply text, sentiment, intent, escalated flag, tools called, and rag sources.
- The web UI also shows product image, escalation reasons, and messages received while waiting for the agent workflow.
- Demo endpoints: localhost:8080 for chat, localhost:8025 for testing email preview, health and insights for diagnostics.

Project Snapshot

- Goal: reliable AI customer support for e-commerce.
- Core approach: fine-tuned LLM + deterministic tools + RAG.
- Stack: FastAPI, PostgreSQL, Qdrant v0.10.10, SQLAlchemy, all-MiniLM-L6-v2, Phi-4-mini.
- Channels: web chat and inbound email webhooks.
- Evaluation: 50 handcrafted support cases across 7 categories.

Implementation Highlights

- FastAPI constraints: chat state, identity checks, SQL queries, vector retrieval, and LLM calls.
- SQLAlchemy maps support entities: Customer, Order, Refund, Ticket, Chat Session, and Message.
- Deterministic routing and fallbacks reduce hallucination risks for order, refund, and escalation workflows.
- Docker Compose starts API, database, vector store, and email preview services reproducibly.

Summary / Conclusions

- Atlas proves that support automation is safer when LLMs are grounded in verified data.
- The system answers product and policy questions without relying on every coding or policy option.
- Deterministic routing and fallbacks reduce hallucination risks for order, refund, and escalation workflows.
- Results show strong escalation handling and clear future work for cancellation workflows.

Future Work

- Connect live refund actions to payment providers such as Stripe or Adyen.
- Add cancellation-focused training data.
- Add cross-encoder re-ranking for stronger retrieval precision.

References and Acknowledgements

- [1] Lewis et al., Retrieval-Augmented Generation, 2020.
- [2] Llama 3, LLaMA, 2023.
- [3] Qdrant, Qdrant, 2023.
- [4] OpenAI GPT-4o, OpenAI, 2024.
- [5] Hugging Face all-MiniLM-L6-v2 and sentence-transformers.

Thanks to the Applied Data Science Department, DATA 208 faculty, project advisor, and observers for project feedback.

Interested in partnering with us on a capstone project or exploring other ways to collaborate with the College of IDS? Contact CIDS_CAP@sjsu.edu to discuss potential projects and partnership opportunities.