
Proud Project #1 — Research Centre (LLM EHR Extraction)

Situation:
At the Research Centre, I was leading a project to automate data extraction from unstructured electronic health records (EHRs) using LLMs. *(Ownership, Customer Obsession)*
Clinicians needed structured data for oncology trials, but manual curation was slow and error-prone. *(Dive Deep, Insist on the Highest Standards)*

Task:
My goal was to design and deploy a system that could reliably extract and structure information like tumor type, stage, and treatment from clinical notes with over 90% accuracy. *(Deliver Results, Bias for Action)*

Action:
I developed a multi-stage NLP pipeline using **LangChain**, **PyTorch**, and **MySQL** to extract and validate key entities. *(Invent and Simplify, Dive Deep)*
To ensure robustness, I fine-tuned domain-specific LLMs and built an evaluation module with statistical metrics. *(Are Right, A Lot; Insist on the Highest Standards)*
I collaborated with oncologists to validate the model outputs and iteratively refined the prompts to improve recall. *(Customer Obsession, Earn Trust)*
I also supported the team's grant writing by generating statistical summaries and visualizations of extracted data. *(Ownership, Deliver Results)*

Result:
The system automated over 70% of the data extraction workload, improving research turnaround time by 40%. *(Deliver Results)*
Our outputs supported successful funding proposals and an upcoming journal publication. *(Think Big, Earn Trust)*

Proud Project #2 — Personalized Patient System

Situation:
At Health Services, I developed a **Personalized Patient System** to support real-time treatment ranking for precision medicine. *(Ownership, Customer Obsession)*
Data was fragmented across structured and unstructured EHR sources, making unified retrieval difficult. *(Dive Deep)*

Task:
I needed to build an ML pipeline that could process real-time EHR streams, rank treatment recommendations, and scale to thousands of patient queries daily. *(Deliver Results, Bias for Action)*

Action:
I developed real-time ETL pipelines using **Kafka**, **Spark**, and **Flink** to unify incoming medical data. *(Invent and Simplify, Dive Deep)*
Then I trained **GBDT + LR** models with note embeddings and applied **Learning-to-Rank** for treatment prioritization. *(Learn and Be Curious, Deliver Results)*
To support inference at scale, I deployed the models via **MLflow on Databricks** and integrated Redis caching for fast retrieval. *(Bias for Action, Are Right, A Lot)*
I also collaborated with clinical data scientists to validate treatment predictions using retrospective data. *(Earn Trust, Insist on the Highest Standards)*

Result:
The model improved treatment ranking precision by **19%** and handled over **10K daily patient queries** in real time. *(Deliver Results)*
The system became the foundation for a new provincial precision medicine initiative. *(Think Big, Ownership)*

Conflict Story — Route Optimization

Situation:
At SHIPPING, I worked with the product team on a route optimization engine. *(Ownership)*
The product lead preferred a simple rule-based system, while I advocated for an ML-based recommender approach. *(Have Backbone; Disagree and Commit)*

Task:
I needed to demonstrate that a machine learning solution could yield measurable gains without compromising reliability. *(Customer Obsession, Are Right, A Lot)*

Action:
I ran controlled **A/B tests** comparing the ML model against rule-based methods, using click-through rate (CTR) and reliability as KPIs. *(Dive Deep, Deliver Results)*
The ML model showed a **1% CTR improvement** while maintaining system stability. *(Deliver Results, Insist on the Highest Standards)*
I presented findings to leadership, proposing a **hybrid deployment** that combined ML ranking with rule-based safety checks. *(Invent and Simplify, Earn Trust)*
Although initially skeptical, the product team agreed to adopt the hybrid model. *(Have Backbone; Disagree and Commit)*

Result:
The hybrid approach improved customer engagement and built lasting trust between product and engineering teams. *(Earn Trust, Deliver Results)*

Mistake Story — QA System Latency Issue

Situation:
Early in development of the **Question Answering System**, our prototype had high latency in production. *(Ownership)*
The bottleneck risked failing a key demo for clinical stakeholders. *(Customer Obsession)*

Task:
I needed to quickly diagnose and fix the issue to meet the performance SLAs before deployment. *(Bias for Action, Deliver Results)*

Action:
I analyzed logs and found that synchronous **Elasticsearch** requests were blocking the Flask API. *(Dive Deep)*
I redesigned the pipeline using **asynchronous tasks** and introduced **Redis caching** for repeated queries. *(Invent and Simplify)*
I benchmarked the revised system using **LangSmith** to measure average latency and throughput under load. *(Are Right, A Lot; Learn and Be Curious)*

Result:
Latency was reduced by **50%**, and the demo succeeded, securing project approval. *(Deliver Results)*
I learned to include scalability testing early in the pipeline lifecycle. *(Learn and Be Curious, Ownership)*

Tight Deadline Story – Research Grant Demo

Situation:
At health authority, we had a **two-week deadline** to deliver a demo of our QA system for a grant review. *(Bias for Action, Deliver Results)*
Missing the deadline could have impacted research funding. *(Customer Obsession)*

Task:
I was responsible for ensuring the prototype met functional and performance expectations in time. *(Ownership)*

Action:
I prioritized the **core retrieval and generation pipeline** over non-essential UI polish. *(Are Right, A Lot; Invent and Simplify)*
I coordinated two async workstreams: one focused on fine-tuning RAG models, another on evaluation scripts. *(Bias for Action, Dive Deep)*
I set up automated evaluation to track accuracy daily, focusing on the most critical question sets. *(Deliver Results, Insist on the Highest Standards)*

Result:
We delivered the demo on time, the reviewers approved the next funding phase, and the system later went live in production. *(Deliver Results, Ownership)*

Multitasking Story – Handling Dual Systems

Situation:
While maintaining the **Personalized Patient System**, I was simultaneously building the **QA retrieval engine**. *(Ownership)*
Both had overlapping deadlines and required cross-functional coordination. *(Bias for Action)*

Task:
I had to ensure that progress on one project didn't block the other, while maintaining production reliability. *(Deliver Results, Insist on the Highest Standards)*

Action:
I divided work into daily sprints and used **CI/CD pipelines** to automate integration tests across both projects. *(Invent and Simplify, Dive Deep)*
I offloaded batch ETL processes to **nighttime jobs** and used monitoring tools to ensure stability. *(Are Right, A Lot; Deliver Results)*
I tracked progress via Jira dashboards and communicated risks proactively to both teams. *(Earn Trust, Ownership)*

Result:
Both systems were delivered successfully, meeting their respective milestones. *(Deliver Results)*
The dual success led to my recognition as a key engineer driving the AI strategy. *(Ownership, Think Big)*

Summary of LP Coverage:

Leadership Principle	Covered In Stories
Customer Obsession	#1, #2, #4, #5
Ownership	#1–#6
Invent and Simplify	#1, #2, #4–#6
Are Right, A Lot	#1, #4–#6
Learn and Be Curious	#2, #4
Hire and Develop the Best	(implied in collaboration – #1, #2)
Insist on the Highest Standards	#1, #2, #4–#6
Think Big	#1, #2, #6
Bias for Action	#2, #5, #6
Frugality	(implicit in optimization choices – #2, #4)
Earn Trust	#1–#3, #6
Dive Deep	#1–#6
Have Backbone; Disagree and Commit	#3
Deliver Results	#1–#6
Strive to be Earth's Best Employer	(lightly touched through collaboration – #1, #2)
Success and Scale Bring Broad Responsibility	(implied through healthcare/AI ethics – #1, #2)
