Optimizing AI Developer Efficiency

Bryan Kohler, Jason Birdsong, Luke Cerminaro, and Ryan Eagan

Team Introductions

Bryan Kohler - Team Lead, Computer Science

Luke Cerminaro - Frontend Developer, Computer Science and Political Science

Jason Birdsong - Backend Developer, Computer Science

Ryan Eagan - Database Developer, Computer Science

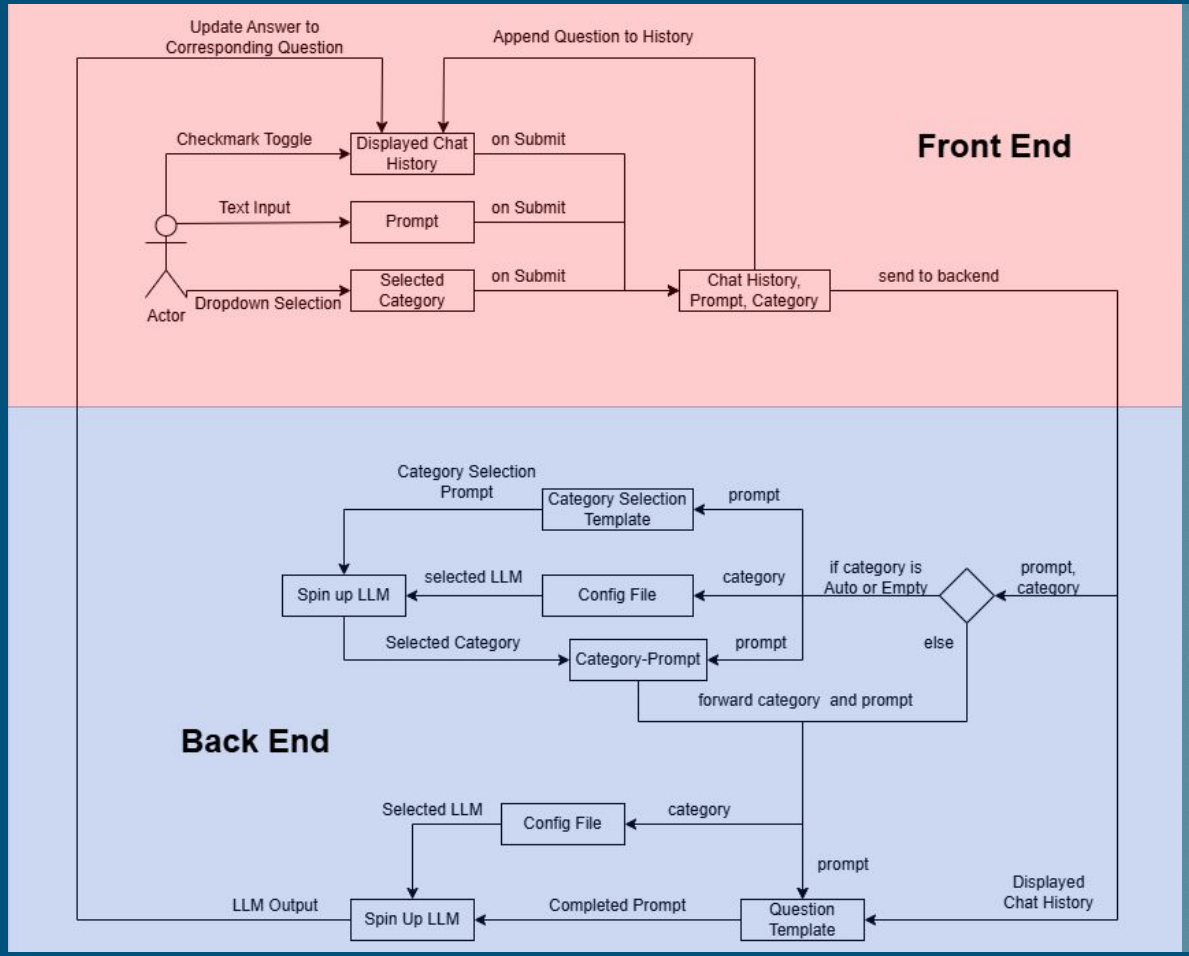
Sponsor Information

- Nonprofit, government contracted research organization
- Founded in NYC in 1967 to research electromagnetic science
- Current research now includes topics like: modeling, simulation, resilient systems, and AI
- Josie and Tristan
- www.riversideresearch.org

Project Description

There are many different Large Language Models (LLMs) available, and all have varying degrees of competency in different fields. In order to maximize accuracy and efficiency, and minimize waste, Riverside Research wants a way to choose the best LLM based on their various coding proficiencies. This will allow for programmers to have one modular, convenient tool that will effectively act as the best LLM for each question, to make development in other projects faster.

Design Overview + Technical Details



TECH STACK

Development Timeline - Month 1

Progress:

- Project architecture and timeline planned (no database required)
- Initial research into tech stack
- Research into LLMs, benchmarks, and potential categories

Challenges:

- Delayed roadmapping with Riverside Research
- Scarce academic research on open-source LLMs
- Speed of LLM development vs research

Development Timeline - Month 2

Progress:

- Finalized benchmark and model selections
- Wrote final research paper over whole research process/findings
- Built backend using FastAPI with full LLM routing capabilities

Challenges:

- Inconsistent benchmark scores across models
- Llama3’s over performance

Development Timeline - Month 3

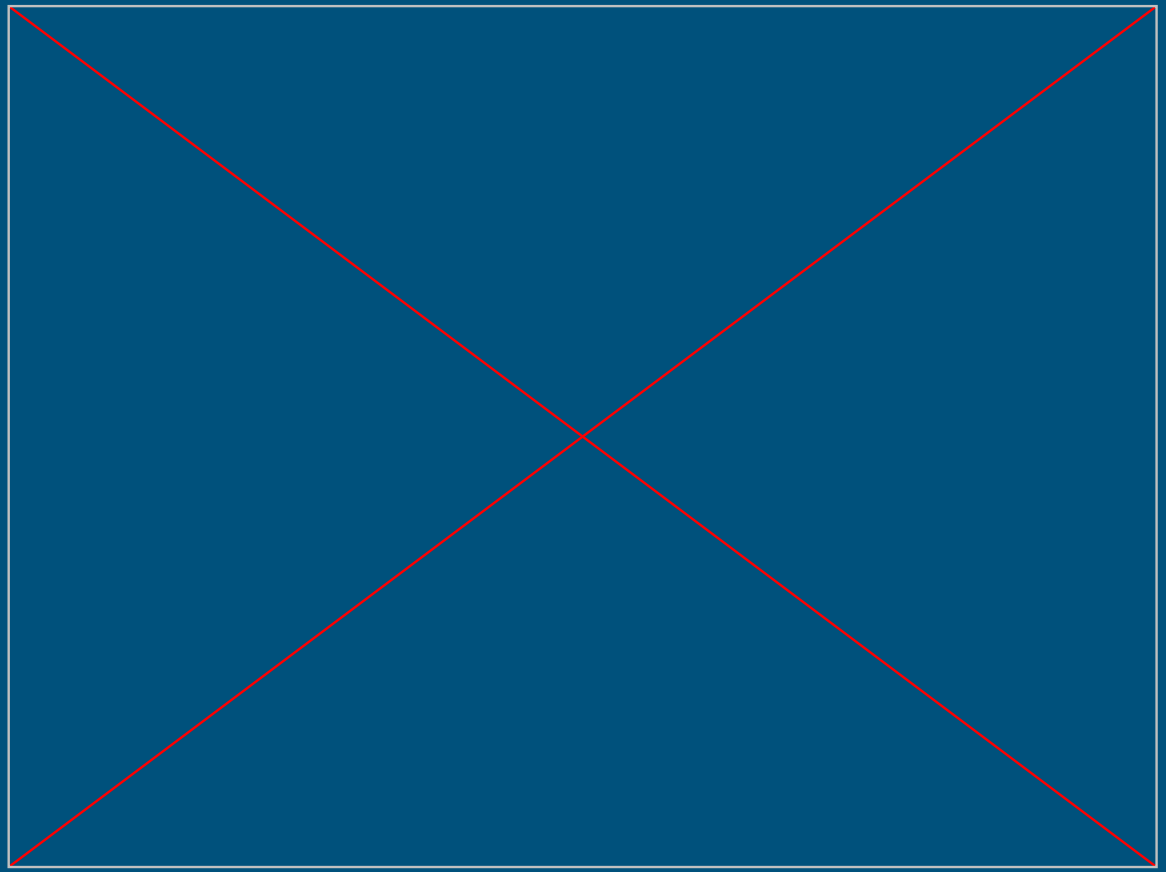
Progress:

- Built Vue frontend
- Added “Auto” category routing
- Implemented chat history display with message visibility control
- Improved UI/UX: formatted output, loading spinners, user instructions

Challenges:

- Connecting front end to backend
- Balancing capstone workload with other commitments.

Demo



Project In Review

Fulfilled Requirements:

- Intelligent LLM routing system implemented
- Functional backend with FastAPI and Ollama integration
- Working, easy to understand frontend
- All stretch goals completed ahead of schedule: persistent history, “auto” category selection

Unfulfilled / Partially Fulfilled:

- Original goal of widely improved efficiency undercut by Llama3, benchmark sparsity
- Currently routes to 2 LLMs instead of 3

Analysis

What We Would Do Differently:

- More frequent communication would have enabled more collaborative decisions on stretch goals
- Widening scope of researched models
- Run our own benchmark tests for more rigorous data.

Contributions

Jason:
- Backend testable with Swagger Docs
- Persistent history between models
- Chat question/answer display
- Toggling/deleting history

Luke:
- frontend and UI design
- Vue framework
- Drop down selection, output container
- Tutorial blurb

Ryan:
- Connected API to backend
- Debugging and testing features
- Loading spinner
- Input textbox

Bryan:
- Team lead responsibilities
- Configuration file
- Auto category selector
- Documentation

All:
- Benchmark and LLM research
- Categorizing benchmarks
- Calculating and Selecting LLMs
- Final Research Report

Questions?