AI Risk Repository Chatbot: User Feedback & Performance Improvement Plan

The AI Risk Repository Chatbot has been integrated as a proof-of-concept into the airisk.mit.edu mirror via Webflow. To optimize user experience and system performance, ensuring the tool meets researcher and practitioner needs, I propose a structured feedback and performance testing gathering initiative within the FutureTech team and adjacent stakeholders. Let this be the document outlining the methodology for collecting actionable insights and performance metrics.

# Current State & Challenge

## What We Have

• Chatbot embedded in airisk.mit.edu

• Direct navigation from main site "Chatbot" tab to /chat interface (no separate homepage for the chatbot)

• Citation system linking to source documents

• Diverse query processing averaging 2-3 second response times

## The Challenge

The "curse of knowledge", of understanding the system's capabilities and limitations. We need fresh perspectives to understand:

• What new users expect vs. what they experience

• Whether the current interface supports user goals effectively

• What barriers prevent successful task completion

• Which queries cause performance bottlenecks

• How system accuracy varies by query type

# Key Questions to Answer Internally

## 1. Onboarding & First Impressions

• Do users understand what the chatbot can/cannot do from first impression?

• Are example queries or guided options needed?

• Does initial query response time affect user engagement?

## 2. Interface Design Decisions (not mutually exclusive)

Current: Full-page embedded chat via iframe

Alternative A: Floating chat widget that expands

→ Idea: implement Alternative A as well, with option to expand to full-page, and see how the testing for that goes

→ Measure load time differences between implementations

## 3. User Intent & Success

• What are users actually trying to accomplish?

• Can they complete their intended tasks?

• What queries fail most often?

• What is the accuracy rate for different query categories?

• How many retries occur due to unsatisfactory responses?

## 4. Trust & Understanding

• Do users trust the responses?

• Do users understand the citations and use them?

• Are citations accurate and do links work properly?

• What percentage of responses contain verifiable facts?

# Feedback Methodology

## Phase 1: User Research (Weeks 1-2?)

### Structured User Testing Sessions

Aim for 6 people? Plus automated testing on 100+ queries

Specific test tasks to fulfill using the chatbot:

1. Find information about AI privacy risks

2. Understand how the repository categorizes risks

3. Explore employment impacts of AI

4. Locate specific statistics about AI safety

Performance metrics for each task:

• Response time (target <2s)

• Factual accuracy (target >95%)

• Citation validity (100% working links)

Non-specific test tasks:

1. 4-6 prompts/tasks at the liberty of the tester (with eventual short follow-ups)

What we will measure:

• Task completion rates

• Understanding of system capabilities

• Points of confusion or abandonment

• Time to first meaningful interaction? (first meaningful question asked and properly responded)

• User satisfaction with the responses

• Query-to-response latency for each interaction

• Token usage per query type

• Error rates and timeout frequencies

How we will measure:

Deploying survey after users interact with the chatbot:

1. Did you find what you were looking for? (Yes/Partially/No), for each question posed to the chatbot

2. What was confusing about the interface? (open text)

3. Can we better convey what the system knows/what the system can do?

4. What would make this tool more useful? (open text)

5. How likely are you to recommend this to someone of a similar occupation to you? What about to a family member or friend of a totally different occupation? (1-10 scale)

6. Was the response speed acceptable? (Yes/No/Sometimes)

7. Did you encounter any errors or timeouts? (Yes/No + details)

+ Reporting the distribution/statistics of answers to numerical scale question

+ Automated performance metrics dashboard with real-time monitoring

+ IF we videotape: empirical measurement of understanding of system capabilities & measurement of time to first meaningful interaction

+ Screen recordings to correlate UI actions with performance spikes

### Secondary for now, important in the long run

1. How well does the iframe embed work on phones/tablets?

→ Mobile performance benchmarks

2. Should we add autocomplete?

→ Impact on query formulation time

3. Should we add conversation history/export?

→ Session memory usage implications

Additional Performance Testing:

• Load testing with 10, 50, 100 concurrent users

• Stress testing with edge cases and malformed queries

• Database query optimization opportunities

• Cache effectiveness measurement

## Phase 2: Analysis and Implementation (Week 3-4?)

• Synthesizing findings into priority recommendations

→ Performance bottleneck identification

• And implement priority features

• Fix identified pain points

→ Optimize slow query patterns

• Implement high-priority features

→ Deploy performance improvements

Performance Success Criteria:

• Response time <2s (P95)

• Accuracy >95% for repository facts

• Error rate <2%

• All citations functional

• Support 100+ concurrent users