*PostHog Analytics: See project dashboard at* [*https://app.posthog.com/*](https://app.posthog.com/)

*Note: Project API key is already configured in frontend code (frontend/src/utils/analytics.ts)*

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**

• **Floating chat widget** on all airisk.mit.edu pages with burgundy branding

• **Full-page chat interface** accessible via /chatbot URL

• **Session transfer system** enabling conversation continuity between widget ↔ full page

• **Session persistence** via backend API (/api/session/{id}/messages endpoints)

• **Citation system** with clickable RID links to source documents

• **Real-time status indicators** during query processing (🔄 Initializing, 🧠 Classifying, 🔍 Searching, ✨ Generating)

• **PostHog analytics integration** tracking queries, responses, citations, errors, and session metrics

• Response times: 2-10 seconds typical, 15-30 seconds on first query (embedding initialization)

## **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?

• (minor) Does initial query response time affect user engagement?

## **2. Interface Design & User Preference**

**Current Implementation:**

• Floating widget (popup) on main site pages • Full-page interface at /chatbot • Session transfer between both interfaces

**Questions to Answer:**

• Which interface do users prefer for quick questions vs. in-depth research? • Is the widget discoverable enough on the main page? • Does session transfer work reliably from user perspective? • Do users understand they can switch between interfaces? • What percentage of users try both vs. stick to one?

## **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?5.

**5. Deployment Threshold**

* What do we want to be a performance for deployment?

# **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: <5s typical, <15s max excluding first-time init)

• Factual accuracy (target >95%)

• Citation validity (target: 100% working links)

• Status indicator appearance (target: 100% of queries show progress)

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

**Widget & Session Transfer Metrics:**

• Widget button click rate (how many visitors engage) • Widget → Full Page transition rate • Session transfer success rate (messages preserved correctly) • Session ID copy attempts and success • Average time spent in widget vs. full page • Query count comparison: widget-only vs. full-page-only vs. mixed users

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)

8. Did you try both the widget and full page versions? (Yes, both / Only widget / Only full page)

9. If you tried both, which did you prefer? (Widget / Full page / No preference) Why?

10. Was the floating chat button easy to find on the main page? (Yes, immediately / Yes, after looking / No, missed it)

11. Did the conversation transfer correctly when switching to full page? (Yes, perfectly / Mostly / No, messages were lost / Didn't try)

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

+ Automated performance metrics dashboard with real-time monitoring, through Railway

**PostHog Events Being Tracked:**

• chatbot\_opened - Initial widget/page load • query\_submitted - User sends message • response\_received - Bot responds (includes latency\_ms, citations\_count, response\_length) • citation\_clicked - User clicks RID link • error\_occurred - Any errors during query processing • session\_ended - Session summary with total queries and duration • session\_transferred - Widget → Full Page transition *(to be added)*

See frontend/src/utils/analytics.ts for implementation details.

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

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

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

2. Should we add autocomplete?

→ Impact on query formulation time

3. Should we add conversation history/export?

→ Session memory usage implications

Additional Performance Testing:

• Stress testing with edge cases and malformed queries

• Database query optimization opportunities

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

• Synthesizing findings into priority recommendations

• And implement priority features

• Fix identified pain points

→ Optimize slow query patterns

• Implement high-priority features

→ Deploy performance improvements