Hackathon Problem Statement

AI-Powered Evidence-on-Demand Bot

Background

During audits, compliance checks, or incident investigations, engineering and operations teams often spend hours searching for the proper proof — log files, configurations, access approvals, or historical records. These proofs exist across various systems such as Slack, Bitbucket, Jira, cloud provider APIs, and internal databases, but are scattered, hard to locate, and require manual collection.

The goal is to create an AI-driven assistant that instantly fetches and formats the required evidence when requested in natural language. This will drastically reduce the time spent on evidence gathering, help meet auditor requests faster, and improve overall audit readiness.

Challenge

Develop an **AI-powered Evidence-on-Demand Bot** that can:

- Accepts natural language queries from an auditor (via an input box in the UI).
- Fetch evidence from APIs (e.g., GitHub PR statuses, JIRA workflow - ***participants can pick one***) and internal data sources.
    - Example input
        - Why was PR #456 merged without approval? - Github integrated
        - Access was given to Jane to read access to Prod DB - JIRA integrated (Validate the access managed by JIRA, approval was taken in the JIRA project using workflow on each ticket - Ex. TODO → approved → Done)
- For systems without APIs, intelligently reference pre-stored documents (e.g., invoices, asset registers) and share relevant details in structured formats like CSV/XLS.
    - Example input- “Share the current count of laptops in the office and export details to XLS or CSV format”

Participants are invited to build a tool that leverages AI for intelligent retrieval, formatting, and delivery of evidence across multiple data sources. The bot should support natural language queries and provide accurate results.

Requirements

1. Natural Language Query Handling *(mandatory)*

- Accept queries from an auditor (via an input box in the UI). (e.g., “Why was PR #456 merged without approval?”).

2. Evidence Retrieval from APIs *(mandatory)*

- Integrate with APIs from Change Management providers like GitHub or Ticketing providers such as Jira, etc. (You need to integrate one provider)
- Pull specific records, logs, or configurations based on query parameters.
- Format results into human-readable summaries or downloadable files.

3. Evidence Retrieval Without APIs *(mandatory)*

- For data sources without APIs (e.g., office asset register, invoices), retrieve data from a pre-stored document repository. [Sh*ould we ask for a fallback if the required information wasn't found - escalate to someone?*]
- Parse files (PDF, Excel, CSV) and extract relevant information.

4. Response Formatting & Delivery *(mandatory)*

- Display results in a structured, auditor-friendly format (tables, JSON, or downloadable file).
- Allow copy/export of evidence for audit reporting.

5. Scalability & Performance *(good to have)*

- Optimise integrations to avoid API rate limits.

Bonus Points

- **AI-powered query rewriting**: Interpret vague queries and clarify them with the user.
- **Multi-source aggregation**: Combine data from multiple systems into one consolidated report.

Stretch Goal (Optional)

- **Slack integration**: allow the same queries via Slack command/DM and return results or a link back to the UI.

Evaluation Criteria

1. **Innovation and AI Utilisation**
    - Does the bot use AI to truly simplify evidence retrieval?
    - Are the integrations and use cases unique compared to existing tools?
2. **User Experience**
    - Is the UI simple and auditor-friendly?
3. **Functionality and Completeness**
    - Are both API-based and non-API evidence retrievals handled well?
    - Does the solution include formatting, access control, and logging?
4. **Technical Excellence**
    - Code quality, integration handling, and adherence to security best practices.
5. **Presentation and Pitch**
    - Clear demo of how the bot solves real-world auditor requests.

Submission Requirements

- A working prototype of the bot with at least 1 API integrations and 1 non-API document parsing feature.
- A presentation deck outlining the problem, solution, and demo flow.
- (Optional) A short video demo showing example queries and responses.