**Technology Hackathon**

**Gen Al for Platform Support – Integrated Platform Environment**

**Challenge Overview**

Currently Technology organization has a large platform support system operations organization that provides application and infrastructure support across L1/L2/L3 levels for our application and infrastructure platforms. Platform support requires troubleshooting, accessing variety of KB (Knowledge base) articles, running ansible automation scripts, reviewing telemetry /observability metrics as well as leveraging foundational information about the Cls (Configuration items) including their relationships, dependencies, health etc. etc. All these tools and products require frequent context switching, cause significant time/effort overhead in accessing different tools, portals to provide the required platform support for the technology environment.

**Problem Statement**

Develop a Gen-Al enabled Integrated Platform Environment (IPE) that provides an integrated console to our platform support teams providing them following capabilities (and more!)

1. Provide agentic capabilities for incident resolution: should provide the platform support operator access to agentic tools to run automations, summarize RCAs, launch health check, pull related incidents etc. etc. from the IPE (Integrated Platform Environment) console
2. Al chatbot to contextually chat with a GPT backend on the incident/issue that they are resolving at the moment
3. Contextual recommendations with telemetry, related incidents etc. should provide proactive recommendation engine mapping to telemetry, related incidents etc.
4. Ability to leverage enterprise information for troubleshooting: should have additional data source integration using MCP (Model context protocol) based integrations based on the need of the platform support person
5. Context based data extraction: Extract fields like connectivity information, upstream and downstream dependency based on simple query for the Cl in question

The Integrated Platform Environment should be the one-stop environment for all platform support users/teams for identifying, resolving and responding to all incidents.

**Expectations from Participants**

Participants will develop an end-to-end solution that:

Provide a Ul view of the Integrated Platform Environment (IPE) for the platform teams with the capabilities identified in the problem statement. Given the time, it is fine if the specific capabilities are a mix of wire-frames vs. working prototype

Utilizes LLMs (such as OpenAl’s GPT, LLAMA, or Gemini) to provide the required capabilities mentioned in the problem statement. The participants can use sample operational data from sources such as hugging face etc. for the purpose of their demo.

Be able to showcase an end-to-end journey of an incident/issue and how leveraging GenAl and agentic capabilities the management of the same will be simplified

Ensure the solution is scalable, explainable, and efficient allowing seamless integration into existing Wells Fargo applications currently used

**Technical Constraints**

The solution must be built using freely available tools (e.g., OpenAl’s free-tier API, Hugging Face models, Scikit-learn, Pandas, LangChain, etc.).

The dataset used for by the participant can be from any of the open-source sample data sets

The solution should incorporate both the mock-up/wireframe for the target IPE (Integrated Platform Environment) as well provide working prototype for one or more agentic flows

Below to test the solution:

**Evaluation Criteria**

Submissions will be evaluated based on:

1. Accuracy & Effectiveness (30%)

How well does the model interpret the emails and documents?

* Are the generated fields, request type and duplicate indicators accurate and contextually relevant?
* Does the solution minimize the false positives/negatives in classification and duplicate detection

1. Innovation & Technical Approach (30%)

* Creativity in using LLMs

Effectiveness of using prioritization of request type, handling emails with multiple request types, ensuring the correct primary intent is identified.

Confidence scoring-Approach to assign confidence levels to extracted fields and

Request type classification.

Rationalization of documents- how well the solution decides which content to prioritize for extraction

1. Scalability & Efficiency (20%)

• Can the solution scale to large datasets?

Performance of the pipeline in terms of runtime and efficiency.

1. Explainability & Interactivity (20%)

Clarity in explaining the approach.

Well-documented code and methodology.

**Submission Requirements**

**Participants must submit:**

A GitHub repository with the code, including a README file with setup instructions.

A presentation (PDF/Slides) explaining the approach, challenges, and results.

A demo working protype (optional but highly recommended) showcasing the solution in action.

**Tools & Resources**

Participants can use any freely available tools:

**Conclusion**

This challenge will push participants to explore the cutting elige of Gen Al, providing a real-world application of Al-powered automation in loan servicing. The best solutions could potentially be adapted for real-world banking applications

.