# Hackathon Challenges

**1. DATA CLEANSER**

**Hack Title:** Data Quality Optimization

**Challenge:**  
Participants are tasked with developing a solution to clean and optimize master data (e.g., customer, vendor, and material data) to ensure its quality and consistency. The solution should address typical data issues such as duplicates, missing fields, and inconsistencies.

**Goals:**

* Migrate data for AI usage and master data integrity
* Identify and remove duplicate records
* Fill in missing fields with appropriate values
* Resolve data inconsistencies to ensure uniformity

**Technical Requirements:**

* **Tools and Software:**
  + Choose the tools that best suit your approach (e.g., Python, AI models, data manipulation libraries)
  + Optional: Microsoft Fabric, Power BI for data visualization and reporting
  + Optional: SAP Data Services for ETL processes
  + Optional: Cloud-based tools for data handling
  + You may use any other tools and platforms you deem fit

**Data Requirements:**

* Sample datasets containing customer, vendor, and material data
* Participants will need to clean and preprocess the data
* Uploading the data into a SAP system after successful cleansing is required

**Hardware Requirements:**

* Computer with internet access
* Required software installed (e.g., Python, libraries)
* VPN (if accessing certain tools or platforms like SAP Playground)

**Submission Requirements:**

* Final solution (e.g., GitHub repo, cloud-hosted solution)
* Brief documentation describing your approach and methodology
* Demo video or presentation

**Judging Criteria:**

* **Innovation:** How creative is the solution?
* **Impact:** Does the solution address the data cleansing challenges effectively?
* **Technical Execution:** Quality of code and solution implementation.
* **Usability:** How easy is it for end-users (data analysts, system users) to use the solution?

**2. SERVICE DESK ALERTS CHECKER**

**Hack Title:** AI Email Categorization and Routing

**Challenge:**  
Participants are tasked with developing an AI solution to filter incoming tasks/mails of service desk alerts and route them to the appropriate recipient. The solution should minimize unnecessary disturbances to service desk consultants by categorizing incoming emails from the service desk Outlook inbox and sending a response with the categorized information back to Outlook.

**Goals:**

* Categorize incoming emails based on content
* Route emails to the appropriate recipient/user group
* Minimize disruptions to service desk consultants during night hours

**Technical Requirements:**

* **Tools and Software:**
  + Optional: Microsoft Power Platform (Power Automate, AI Builder)
  + Optional: Any AI models or frameworks for text classification
  + You may use any platform (e.g., Python, Low-code platforms)
* **Integration:**
  + Optional: Integration between Power Automate and Outlook mailbox
  + API integration may be required depending on your approach

**Hardware Requirements:**

* Computer with access to Outlook mailbox (Test Environment)
* VPN and internet access to use required platforms (e.g., Power Automate)

**Submission Requirements:**

* Final solution (e.g., a flow created in Power Automate, GitHub repo)
* Brief documentation on how the classification and routing work
* Demo video or presentation

**Judging Criteria:**

* **Innovation:** How creative and efficient is the email classification system?
* **Impact:** Does it reduce unnecessary disturbances during off-hours effectively?
* **Technical Execution:** Quality of implementation and integration.
* **Usability:** Is the solution easy to use, especially for non-technical users?

**3. SAP MOVEMENT WITH PROMPTING**

**Hack Title:** Simplified Warehouse Movement via Natural Language

**Challenge:**  
Participants are tasked with developing a solution to create or modify an existing UI5 Dialog within SAP to allow natural language commands to trigger movements within the SAP EWM system. This solution aims to simplify warehouse interactions and reduce the need for extensive system knowledge.

**Goals:**

* Enable natural language commands to trigger movements within SAP (EWM)
* Simplify user interactions for warehouse employees
* Reduce the need for in-depth SAP system knowledge

**Technical Requirements:**

* **Tools and Software:**

**Mandatory:**

* SAP UI5, SAP S/4 HANA for system integration.
* API interaction using CL\_HTTP\_CLIENT ABAP-side interface.

**Optional:**

* Natural Language Processing (NLP) models like **Llama3** (preferred) or **Llama2**.
* Ollama Connector for ABAP:
  + [GitHub: b-tocs/abap\_btocs\_Ollama](https://github.com/b-tocs/abap_btocs_Ollama).
* Cloud-based AI tools for enhanced NLP capabilities

**Documentation Resources:**

In PG3 the following Tooling is used for the connection to the Ollama instance.

* <https://github.com/b-tocs>

More specifically, the Ollama Connector:

* [https://github.com/b-tocs/abap\_btocs\_Ollama?tab=readme-ov-file](https://github.com/b-tocs/abap_btocs_ollama?tab=readme-ov-file)

Official ACN-Internal Documentation can be found:

* <https://asgixpo.atlassian.net/wiki/spaces/itservicewu/pages/310837406/SAP+Gen+AI#KI-f%C3%BCr-ABAP---onPremise>
* Main Model in use: *llama3*, alternatively *llama2* is also available.

Destination

At the base of it, you can simply imagine API calls made via the CL\_HTTP\_CLIENT ABAP-side API Interface. The following **SM59** destination was set up accordingly:

* Point of Contact: Lars Zillger
* <https://asgixpo.atlassian.net/wiki/spaces/itservicewu/blog/2024/07/08/564101126/News-2024-07-08> → Here the addition of the Ollama Models was announced.

At the time of writing, the RFC Destination Z\_OLLAMA is in use. Sometimes this may cause issues and therefore the alternative, Z\_OLLAMA\_2 is available. The model in use is llama3, and llama 2 as alternative (however, the latter is not available via the alternative RFC destination – Z\_OLLAMA\_2)

A screenshot of a computer

Description automatically generated

Main Package(s)

* $BTOCS\_OLLAMA which includes:
  + $BTOCS\_OLLAMA\_RWS (Remote Web Service) and
  + $BTOCS\_OLLAMA\_GUI

Testing locally

You may test the endpoint manually, using the following Report: “ZBTOCS\_OLLAMA\_GUI\_RWS\_DEMO”. More information & Documentation can be found in the corresponding repository: [https://github.com/b-tocs/abap\_btocs\_Ollama](https://github.com/b-tocs/abap_btocs_ollama)

Example Usage(s) – PG3

One of the current usages on PG3 - for the Ollama Model - is the “Weißwurst Manager”, here the OData V2 Service “Z\_CHATBOT\_AI” can be found & inspected within the SEGW. The use case here was, to interpret prompt input from the user via a UI5 based user interface and create or cancel “orders” based on that.

**Data Requirements:**

* Access to a test SAP EWM environment (or equivalent)
* Sample data for warehouse movements

**Hardware Requirements:**

* Access to the Internet
* Access to SAP Playground (or equivalent)

**Submission Requirements:**

* Final solution (e.g., a working UI5 Dialog or GitHub repo)
* Brief documentation describing your approach to integrating NLP with SAP
* Demo video or presentation

**Judging Criteria:**

* **Innovation:** How innovative and user-friendly is the solution?
* **Impact:** Does the solution effectively simplify warehouse operations?
* **Technical Execution:** Quality of code, integration with SAP, and NLP implementation.
* **Usability:** How easy is it for warehouse employees to use the solution?

**4. AI Chatbot for IT Service Desk and Incidents**

**Hack Title:** Automated IT Incident Resolution

**Challenge:**  
Develop a GenAI-powered chatbot to automate IT incident resolutions. The solution should handle common IT issues such as login problems, credential resets, and application troubleshooting. It should streamline ticket creation and routing while reducing the workload for IT support teams.

**Goals:**

* Provide instant solutions to common IT incidents.
* Automate ticket creation, assignment, and management.
* Reduce IT workload and improve response times.

**Technical Requirements:**

* Use of AI/GenAI frameworks (e.g., OpenAI, Dialogflow).
* Integration with ticketing systems like Jira, ServiceNow, or custom ITSM platforms.
* Optional: Multi-language support and voice-to-text functionality.

**Hardware Requirements:**

* Computer with internet access.
* Required tools for development (e.g., Python, Node.js).
* VPN access if integration with internal systems is required.

**Submission Requirements:**

* A working chatbot prototype (e.g., GitHub repo or live demo).
* Documentation describing functionality, architecture, and implementation.
* A demo video or presentation showcasing the chatbot in action.

**Judging Criteria:**

* Innovation in chatbot capabilities and features.
* Effectiveness in reducing ticket resolution time.
* Ease of use for IT support teams and non-technical users.

**5. Brownfield Project AI**

**Hack Title:** AI-Driven Transition for Legacy Projects

**Challenge:**  
Build an AI-powered solution to streamline the transition of brownfield projects by understanding unknown codebases and documentation. The solution should assist DevOps teams with code analysis, troubleshooting, and maintaining project integrity.

**Goals:**

* Shorten transition phases by creating a centralized knowledge base.
* Enhance troubleshooting efficiency and maintainability.
* Improve documentation and code insights for smoother project takeovers.

**Technical Requirements:**

* Use AI/ML tools for code analysis and documentation linking.
* Integration with version control systems like GitHub or GitLab.
* Optional: Natural language processing (NLP) for querying documentation and codebases.

**Hardware Requirements:**

* Computer with access to development tools.
* VPN for secure access to private repositories if required.

**Submission Requirements:**

* A centralized knowledge base or analysis tool prototype (e.g., GitHub repo).
* Documentation of the approach and methodology used.
* A demo video or presentation showcasing the solution’s features.

**Judging Criteria:**

* Reduction in transition time and troubleshooting delays.
* Innovation in understanding legacy systems.
* Usability for DevOps and engineering teams.

**6. GenAI Copilot Framework for ServiceNow**

**Hack Title:** GenAI for Streamlined ServiceNow Workflows

**Challenge:**  
Create a GenAI-powered framework to enhance user adoption and efficiency within ServiceNow. The solution should automate routine tasks, provide cost analysis, and support incident resolution.

**Goals:**

* Automate daily workflows and improve user acceptance.
* Enable quick decision-making through intelligent recommendations.
* Integrate seamlessly with ServiceNow workflows.

**Technical Requirements:**

* Integration with ServiceNow APIs and GenAI models.
* Use of cloud-based AI services like OpenAI or Azure AI.
* Optional: Additional integrations for advanced reporting and visualization.

**Hardware Requirements:**

* Computer with internet access.
* Access to ServiceNow test environments.

**Submission Requirements:**

* Functional GenAI-powered framework prototype (e.g., GitHub repo).
* Documentation on setup and capabilities.
* Demo video or live presentation showcasing its use.

**Judging Criteria:**

* Innovation in workflow automation and AI integration.
* Efficiency in reducing manual tasks.
* Usability for ServiceNow users.

**7. SurViVor Control Tower**

**Hack Title:** AI-Powered Supply Chain Resilience

**Challenge:**  
Develop an AI-driven system to monitor and improve supply chain resilience in real-time. The solution should track disruptions, analyze resilience scores, and recommend actions for unforeseen events.

**Goals:**

* Monitor disruptions and inventory levels in real-time.
* Provide scenario-based recommendations for resilience.
* Improve decision-making and operational efficiency.

**Technical Requirements:**

* Use AI/ML tools for disruption analysis and resilience scoring.
* Integration with real-time data sources or APIs.
* Optional: Advanced visualization capabilities for KPIs.

**Hardware Requirements:**

* Computer with internet access.
* Access to sample supply chain data.

**Submission Requirements:**

* A functional dashboard or prototype (e.g., GitHub repo).
* Documentation explaining the system’s resilience analysis.
* Demo video showcasing scenario-based recommendations.

**Judging Criteria:**

* Accuracy of resilience score and disruption predictions.
* Effectiveness of recommendations.
* Usability for supply chain stakeholders.

**8. Talk to The Factory**

**Hack Title:** Automated Malfunction Reporting and Resolution

**Challenge:**  
Create a GenAI-powered solution for efficient malfunction reporting and resolution in factories. The system should enable machinists to report issues via apps or voice inputs and receive AI-generated solution suggestions.

**Goals:**

* Automate malfunction report creation and analysis.
* Reduce machine downtime and operational costs.
* Provide faster and more accurate resolutions.

**Technical Requirements:**

* Use of AI models for malfunction analysis and solution recommendations.
* Integration with ITSM systems like Jira or ServiceNow.
* Optional: Mobile or web-based reporting interface.

**Hardware Requirements:**

* Computer with internet access.
* Tools for app development if applicable.

**Submission Requirements:**

* Functional malfunction reporting system (e.g., GitHub repo).
* Documentation explaining the workflow and AI logic.
* Demo video showcasing issue reporting and resolution.

**Judging Criteria:**

* Reduction in machine downtime.
* Innovation in reporting and resolution mechanisms.
* Accuracy of AI recommendations.

**9. Outbound Logistics: Freight Cost Comparison**

**Hack Title:** AI for Freight Cost Optimization

**Challenge:**  
Develop an AI-powered solution to automate the comparison of freight costs from multiple carriers. The system should extract and analyze rates from different formats (e.g., PDFs, Excel files) and recommend the most cost-effective carrier for shipments.

**Goals:**

* Automate freight rate extraction and analysis.
* Identify the most cost-effective carriers based on shipment details.
* Streamline decision-making for logistics teams to optimize shipping costs.

**Technical Requirements:**

* Use AI/ML models to process and analyze freight rates.
* Support multi-format data extraction (e.g., PDF, Excel).
* Optional: Real-time recommendation engine for carriers based on shipment needs.

**Hardware Requirements:**

* Computer with internet access.
* Tools for data extraction and analysis (e.g., Python libraries, AI platforms).

**Submission Requirements:**

* Freight cost comparison tool (e.g., GitHub repo).
* Documentation explaining data extraction, analysis, and decision-making workflows.
* Demo video or presentation showcasing cost optimization features.

**Judging Criteria:**

* Innovation in freight cost analysis and automation.
* Accuracy of AI-generated recommendations.
* Usability for logistics teams.

**10. Outbound Logistics: Efficient Use of Truck Loading Space**

**Hack Title:** AI-Driven Truck Loading Optimization

**Challenge:**  
Build an AI solution to optimize truck loading by generating 3D visualizations of space usage. The system should maximize truck space utilization, streamline loading processes, and ensure compliance with handling regulations.

**Goals:**

* Maximize truck space utilization with AI-powered optimization.
* Reduce loading times by providing detailed 3D visualizations of truck layouts.
* Ensure compliance with regulations for handling special goods.

**Technical Requirements:**

* Use AI for optimization algorithms to calculate efficient space usage.
* Tools for 3D visualization (e.g., WebGL, Python 3D libraries).
* Integration with logistics systems for input data and results.

**Hardware Requirements:**

* Computer with required software for 3D modeling and optimization.
* Internet access for testing and integration.

**Submission Requirements:**

* Truck loading optimization prototype (e.g., GitHub repo).
* Documentation describing the algorithms and compliance mechanisms.
* Demo video showcasing the 3D visualization and loading process.

**Judging Criteria:**

* Innovation in space-saving techniques.
* Reduction in loading times and operational efficiency.
* Usability for logistics personnel and adaptability to real-world scenarios.

**General Hackathon Information**

**Timeframe / Deadlines:**

* **Project Deadline:** Final submissions must be made by 5pm.
* **Presentation:** Each team will have 10 minutes to present their solution.
* **Progress Check-ins:** Optional mid-way check-ins for feedback, if needed.

**Team Collaboration:**

* Participants can work solo or in teams. You are encouraged to collaborate with others to combine skill sets and creativity.

**Resources & Mentorship:**

* Access to APIs, sample datasets, and other resources will be provided.
* Mentors will be available throughout the hackathon to offer guidance on technical challenges and platform usage.

**Prizes & Incentives:**

* Prizes will be awarded to the top solutions in each category.
* Winning teams will receive recognition points as prizes.

**Ethical Guidelines / Data Privacy:**

* Ensure all data used in your solution is anonymized, and avoid using sensitive or personal data unless properly consented.
* Solutions should adhere to ethical AI guidelines, ensuring that user privacy and data security are respected.

**Good Luck to All Participants!**  
Let your creativity shine and choose the tools that will help you deliver the best solutions!