🚀 Hackathon 2 – Small Business Auto-Bookkeeper (Local AI Edition)

Local AI · Vertex AI · CORD Dataset · Full-Stack Automation Challenge

🧩 Context

Your team will build a production-grade prototype of a **Local-AI-powered bookkeeping & analytics system** for a fictional small business (SME).
The SME receives lots of receipts every month and wants automated:

- extraction
- categorization
- anomaly detection
- analytics
- and an **AI Auditor** they can chat with

You will combine **local AI**, **RAG**, a **real dashboard UI**, and optional **Vertex AI** services.

Teams: **3 members each**.

🎁 Hackathon Starter Kit

Each team receives access to a prepared Drive folder containing:

1. Premium MUI Dashboard Template (React + TS)

A fully designed enterprise-grade admin template including:

- layout system (sidebar, navigation, header)
- tables, lists, cards
- charts (bar, line, pie)
- form components
- dialogs, modals
- theming system

This ensures every team can build a professional-looking SaaS dashboard without wasting time on basic UI scaffolding.

---

2. Mock API Dev Server

A lightweight Node/Python dev server that simulates the backend API.
This exists to help teams build and test the frontend early and understand the data contracts.

The mock server includes endpoints such as:

Example endpoints (will match the provided overview):

- `GET /api/receipts`
- `GET /api/receipts/:id`
- `GET /api/analytics/summary`
- `GET /api/analytics/vendors`
- `GET /api/audit`

- `POST /api/chat/query`
- `POST /api/ingest`

The mock server:

- returns realistic sample data
- matches the database schema you are expected to implement
- can be replaced by your real backend once you build it

This gives teams freedom to evolve the backend while keeping a stable API contract for the frontend.

---

3. Dataset – CORD Receipt Dataset

A CC-BY 4.0 licensed dataset including:

- receipt images
- OCR text
- bounding boxes
- multi-level semantic labels
- line items and totals

You will use this dataset to simulate **receipt ingestion** and extraction.

---

4. Local AI Model Freedom

Teams may run **any local model**, including:

- Ollama / LM Studio LLMs

- Donut (OCR-free document understanding)
- Local embedding models
- Local text-classification models

---

5. Optional Access to Vertex AI & GCP Tools

Teams may choose to use Google Cloud for:

- RAG
- embeddings
- vector search
- agents
- hosting
- model deployment
- data pipelines

the available Vertex AI capabilities include:

🔹 Vertex AI Studio

Try and test model behaviors interactively.

🔹 Model Garden

Use:

- Gemini models
- open-weights models
- task-specific models

- or import your own

🔹 GenAI Evaluation

Evaluate prompt quality, grounding, output correctness.

🔹 Agent Builder Suite

Includes:

- **Agent Garden** – prebuilt agent blueprints
- **Agent Engine** – runtime for multi-step agent flows
- **RAG Engine** – high-level retrieval-augmented generation pipelines
- **Vertex AI Search** – enterprise search
- **Vector Search** – scalable vector storage

🔹 Model Registry

Upload trained/fine-tuned models and deploy them.

🔹 Datasets & Pipelines

Automate ingestion jobs or receipt processing workflows.

🔹 Colab Enterprise / Workbench

GPU notebooks for LLM fine-tuning or embedding generation.

🔹 Provisioned Throughput

For high-volume vector search or embedding workloads.

🔧 How to Request GCP Resources

Teams simply message the mentors on **Discord** with:

- team name
- resource type needed (e.g. Vertex AI Search index, RAG Engine instance, GPU notebook)

We will create the resource on our GCP instance and grant you access.

🎯 What You Must Build

The final product consists of **three major pillars**.

Pillar 1 — Auto-Bookkeeper Engine

Required:

- Load & ingest receipts (JSON from CORD dataset)
- Local or Vertex AI extraction of:
  - vendor
  - date
  - total
  - VAT
  - payment method
  - line items
- Expense classification
- Merchant & category normalization

- Error flags when totals do not match item sums

Optional Enhancements:

- OCR-free parsing with Donut
- VAT number validation
- Automatic deduplication across receipts

Pillar 2 — Financial Command Center (Dashboard)

Using the MUI template, teams should implement:

Pages Required

- **Receipts Page**
  - table view
  - filtering
  - detail drawer/page
- **Analytics Dashboard**
  - category spending
  - monthly spend
  - vendor summary
- **Vendor Analytics**
  - top merchants

    - spending trends
- **Audit Findings**
    - duplicates
    - mismatched totals
    - suspicious categories (alcohol, tobacco)
    - missing VAT

Pillar 3 — AI Auditor Chat

A natural-language accountant assistant.

Must support queries like:

- “Show me all travel receipts above 100 EUR.”
- “How much did we spend on hardware last quarter?”
- “Which receipts contain alcohol?”
- “Find duplicates.”

Technical Requirements:

- RAG pipeline using:
    - local embeddings, OR
    - Vertex Vector Search, OR
    - Vertex RAG Engine
- Query planning (structured query extraction)

- Response referencing specific receipt IDs

🛠️ Technical Requirements

Frontend:

- React + TS
- MUI Premium Template
- Pages:
  - Receipts
  - Receipt Details
  - Analytics
  - Vendors
  - Audit
  - Chat

Backend:

- Replace or extend the mock server
- Implement real endpoints
- Either:
  - local Node/Python server, or
  - Cloud Run

Local AI:

- At least one local model must be integrated
- You may combine local + Vertex workload splitting

📦 Deliverables

Teams must submit:

- Git repository
- README with:
    - architecture
    - setup instructions
    - local AI explained
    - which GCP/Vertex tools were used
    - dataset attribution
- Running demo (local or hosted)
- Optional: 2–3 minute demo video

🧮 Evaluation Criteria

1/3 – Frontend & UX

- Quality of UI
- Presentation of analytics

- Usability

1/3 – Backend & Data Pipeline

- Clean architecture
- Data normalization
- Extraction quality
- Audit logic

1/3 – AI Intelligence & RAG

- Proper use of local models
- Creative use of Vertex tools
- Accuracy of RAG answers
- Quality of anomaly detection

🎤 Closing Note

This hackathon simulates a high-impact real-world AI automation challenge.
You are free to innovate, expand scope, and build a polished, production-ready prototype.