SHAASTRA2026

VISA 24HRS AI HACKATHON

AI-POWERED MODERNIZATION OF PAYMENT REPORTING SYSTEMS

Team Name: Aesthetic Agents

Team Members :

- Hari Sudharsan G.
- Mahadev M.
- Susanth Mohan Kamala
- Nithin S.

BREIF ABOUT THE IDEA

We propose an intelligent, agentic orchestration platform designed to modernize the financial reporting lifecycle. Unlike traditional systems that rely on static dashboards or manual Excel reconciliation, our solution deploys a Multi-Agent System (MAS) that acts as an autonomous financial analyst. By leveraging secure, on-premise Large Language Models, the platform transforms static banking logs into a dynamic, queryable intelligence layer that provides real-time insights, automated reconciliation, and predictive forecasting.

- **Autonomous Financial Analyst:** Replaces manual Excel work with a Multi-Agent System that plans, executes, and audits financial tasks independently.
- **Secure "Edge-AI" Architecture:** Utilizes specialized LLMs tailored for on-premise deployment, ensuring sensitive financial data never leaves the institution's secure infrastructure.
- **Complex Legacy Ingestion:** Seamlessly ingests and parses complex multi-format reports (Visa/Mastercard PDFs, Mainframe Fixed-Width logs) that traditional OCR tools fail to read.
- **Deterministic Accuracy:** Decouples reasoning from calculation—using LLMs to understand intent and SQL engines to perform zero-error mathematical reconciliation.
- **Natural Language Intelligence:** Empowers non-technical staff to interrogate massive datasets using plain English (e.g., "Why did settlement drop on Tuesday?"), bridging the gap between data and decision-making.

HOW IT IS DIFFERENT?

Most Generative AI solutions in finance are simple "Chat with PDF" wrappers that fail because they hallucinate numbers and cannot read complex tables. Our solution differentiates itself through three core architectural decisions:

Deterministic Accuracy: We do not ask the LLM to do math. Instead, we use a "Code Interpreter" approach where the AI writes SQL queries executed by a high-performance OLAP engine (DuckDB), ensuring 100% mathematical precision for reconciliation.

Hybrid Memory Architecture: We utilize a "Split-Brain" storage strategy—Vector Storage for semantic understanding (rules/compliance) and Structured SQL Storage for analytical memory (transaction logs)—solving the context window limitation for massive datasets.

Fidelity-First Ingestion: Instead of generic text extraction, we employ deep-learning-based OCR and specialized fixed-width parsers to preserve the structural integrity of banking tables, ensuring no row or column is misread from legacy reports.

HOW IT SOLVES THE PROBLEM?

Traditional financial reporting is reactive, fragmented, and heavily reliant on manual interpretation. Our solution bridges the gap between legacy formats and modern decision-making by deploying an intelligent orchestration layer. By treating reports as structured data sources, the platform reduces time-to-insight from days to seconds while ensuring mathematical precision.

- **Transforms Static Data to Live Intelligence:** Converts "dead" data in PDFs and mainframe logs into a live, SQL-queryable database for instant visualization and dynamic querying.
- **Automates Forensic Reconciliation:** Instantly cross-references thousands of transactions between Authorization and Settlement logs to surface hidden discrepancies and fee anomalies.
- **Demystifies Technical Jargon:** Decodes complex banking standards (e.g., Visa TC33) using Semantic Search (RAG) to explain rules and errors in plain English.
- **Enables Proactive Forecasting:** Shifts focus from reactive reporting to planning by using ML models to predict future cash flow and liquidity needs based on historical trends.
- **Unifies Fragmented Data Silos:** Ingests diverse formats (CSV, Text, PDF) into a single analytical memory, creating a holistic view of the entire payment lifecycle.

KEY FEATURES OF THE PROPOSED SOLUTION

- **Multi-Format Intelligent Ingestion:** Features a "Fidelity-First" parsing engine that seamlessly ingests diverse formats simultaneously—utilizing Deep Learning for PDF tables, Schema Parsing for Mainframe Fixed-Width logs, and standard parsers for CSVs—to preserve data integrity.

- **Smart Context Routing:** Deploys a specialized "Supervisor Agent" that classifies user intent in real-time. It intelligently routes queries to the most efficient sub-system—sending math tasks to the SQL engine, policy questions to the Vector Store, and trend analysis to the Forecasting module.

- **Cyclic Agentic Orchestration:** Utilizes a non-linear reasoning loop (LangGraph) where the AI plans, executes, and self-corrects. If a query fails or data is missing, the agent autonomously retries or requests clarification rather than hallucinating an answer.

- **Deterministic Reconciliation Engine:** Decouples reasoning from calculation by leveraging an embedded OLAP SQL engine (DuckDB). This ensures zero-error cross-referencing of thousands of transaction rows, identifying discrepancies with mathematical precision.

- **Automated Predictive Analytics:** Goes beyond historical reporting by integrating Python-based Machine Learning modules. The system runs regression models on settled data to forecast future cash flow requirements and detect emerging anomalies.

TECHNOLOGY STACK & ARCHITECTURAL COMPONENTS

Frontend & Interaction Layer

- Interface: Chainlit (Conversational UI for interactive dashboards & chat).
- Visualization: Plotly / Matplotlib (Dynamic rendering of financial trends & graphs).

Orchestration & Intelligence Layer

- LLM: Llama 3.2 8B (Optimized for edge deployment via Ollama).
- Agent Framework: LangGraph (Cyclic state management for planning, execution, and self-correction).
- Guardrails: Custom Relevance Classifier (Input validation & safety checks).

Memory & Data Storage ("The Trinity")

- Short-Term Memory: SQLite (Session state & context persistence).
- Semantic Memory: ChromaDB (Vector store for RAG / Policy retrieval).
- Analytical Memory: DuckDB (High-performance in-process SQL OLAP engine for structured logs).

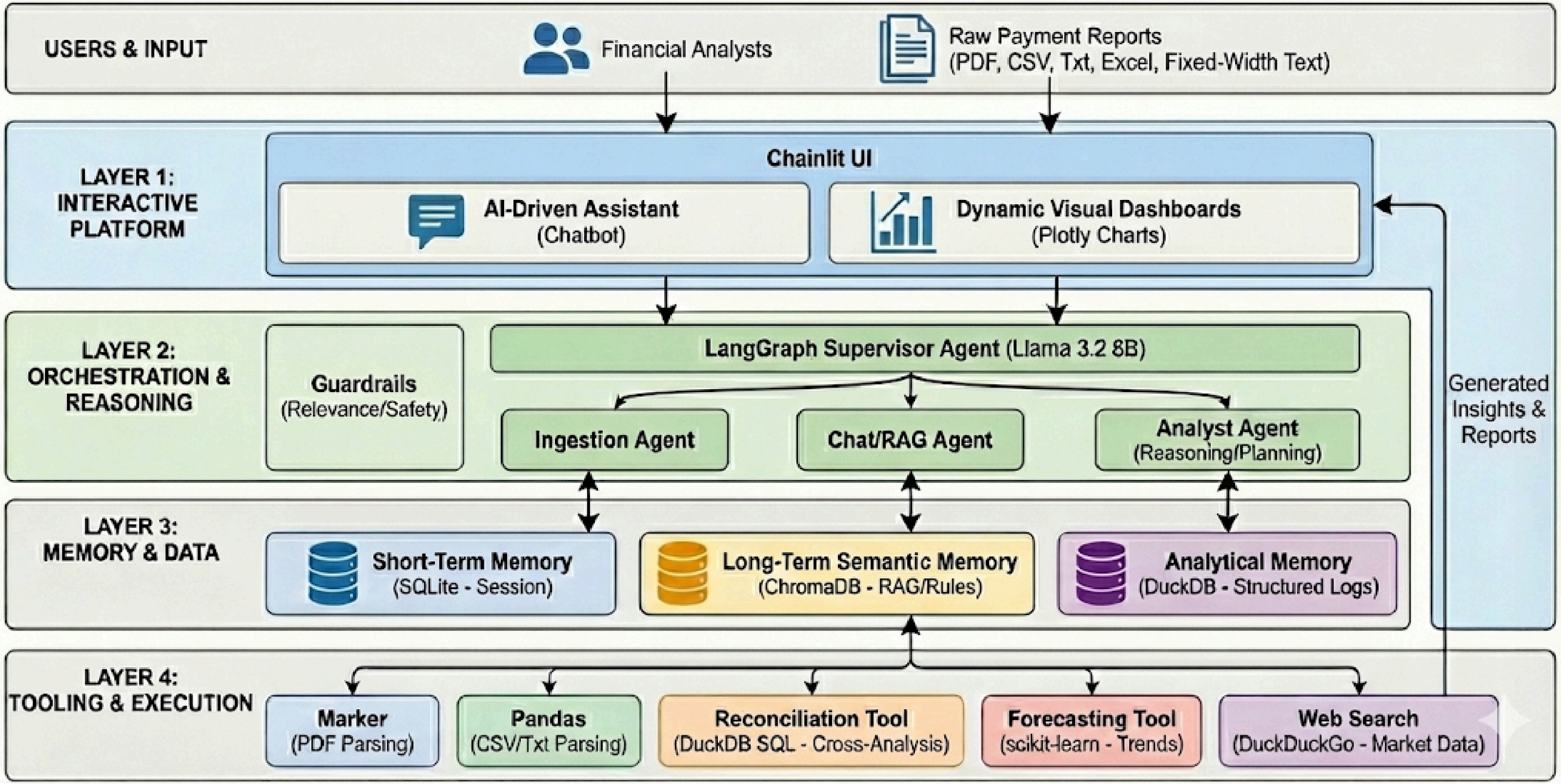
Ingestion & Parsing Layer

- PDFs: Marker (Deep Learning OCR for precise table extraction).
- Structured Logs: Pandas (CSV & Fixed-Width/Mainframe parsing).
- Unstructured Data: Unstructured.io (Handling messy text/notes).

Analysis & Execution Tools

- Reconciliation: DuckDB SQL Wrapper (Deterministic cross-reference analysis).
- Forecasting: Scikit-learn (Python REPL for regression & trend prediction).
- Market Data: DuckDuckGo Search (Live external rate/fee retrieval).
- Reporting: FPDF (Automated PDF report generation).

ARCHITECTURE DIAGRAM



WORKFLOW

Ingestion & Architecture

Phase 1: Intelligent Ingestion & Hybrid Memory Architecture

- **The Input Challenge:** Financial institutions handle fragmented data formats: high-volume Authorization Logs (CSV), legacy Clearing Files (Fixed-Width/Mainframe), and complex Settlement Reports (PDF).
- **Layer 1:** Security Guardrails: A Relevance Classifier acts as the first line of defense, scanning file headers to validate legitimacy and instantly rejecting unsafe or irrelevant uploads before processing.
- **Layer 4:** "Fidelity-First" Parsing: The system employs intelligent routing to select the optimal parser:
  - Deep Learning OCR (Marker): Extracts tables from Settlement PDFs while preserving structural integrity (rows/columns).
  - Schema-Based Parsing (Pandas): Decodes legacy fixed-width byte strings from mainframe logs into readable columns.
- **Layer 3:** "Split-Brain" Storage Strategy:
  - Analytical Memory (DuckDB): Stores quantitative data (amounts, dates, IDs) in a high-performance SQL engine for zero-error mathematical operations.
  - Semantic Memory (ChromaDB/Vector Store): Indexes qualitative data (compliance rules, error codes) for context-aware retrieval (RAG).

WORKFLOW

Orchestration & Insight Generation

Phase 2: Agentic Orchestration & Deterministic Analysis

- **Layer 2:** The "Supervisor" Logic: A Llama 3.2 Supervisor Agent acts as the central traffic controller, analyzing user intent to route tasks to specialized sub-agents rather than a single monolithic model.
- **Workflow A:** Automated Reconciliation ("The Accountant"):
  - **Trigger:** User requests discrepancy analysis (e.g., "Compare Auth vs. Settlement").
  - **Action:** The Analyst Agent instructs the SQL engine to execute a deterministic LEFT JOIN, identifying missing Transaction IDs with 100% mathematical precision instead of LLM guessing.
- **Workflow B:** Contextual Understanding ("The Librarian"):
  - **Trigger:** User asks about specific rules (e.g., "Explain Error Code 51").
  - **Action:** The RAG Agent retrieves exact definitions from the Vector Store (Visa Guidelines) and combines them with transaction data to explain failures in plain English.
- **Workflow C:** Market Intelligence ("The Researcher"):
  - **Trigger:** User asks about external factors (e.g., "2025 Visa Fee Changes").
  - **Action:** The system detects missing internal data and activates the Web Search Tool to fetch live market rates.

INFERENCE & ORCHESTRATION LAYER

Tech Stack: Llama 3.2 8B (Ollama), LangGraph.
General Utility:

- **Llama 3.2 8B:** Lightweight LLM optimized for edge deployment and function calling.
- **LangGraph:** State machine library for building cyclic, multi-step agent workflows.

Solution:
State Management: LangGraph manages the "reasoning loop," allowing the agent to retry failed SQL queries or request missing information before answering.

Use Case: The system identifies a user's request for "trend analysis," routes it to the specific Analyst Agent, and maintains the conversation state while tools are executing.

MEMORY ARCHITECTURE

Tech Stack: SQLite, ChromaDB, DuckDB.

General Utility:

- SQLite: Relational database for session management.
- ChromaDB: Vector database for semantic text retrieval.
- DuckDB: In-process OLAP database for high-performance analytical queries.

Solution:

- Context Management: SQLite stores the active chat history (Short-Term Memory).
- RAG: ChromaDB retrieves unstructured text like compliance rules from PDFs (Long-Term Semantic Memory).
- Analytics: DuckDB executes SQL on parsed logs, ensuring mathematical accuracy that vector stores cannot provide (Analytical Memory).

Use Case: To explain a "failed transaction," the system queries DuckDB for the transaction amount and ChromaDB for the definition of the error code.

EXTERNAL DATA RETRIEVAL

Tech Stack: DuckDuckGo Search, LangChain Community.

General Utility: API-free tool for anonymous web searching.

Solution:

- Dynamic Context: Supplements static uploaded reports with real-time market data that is otherwise unavailable in historical logs.
- Rate & Policy Checks: Verifies current interchange fees or compliance updates referenced in the user query.

Use Case: User asks about "2025 Visa Fee changes"; the agent fetches live data from the web to compare against the historical rates in the uploaded CSV.

DOCUMENT INGESTION

Tech Stack: Marker (PDF), Pandas (CSV/Fixed-Width), Unstructured (General).
General Utility: Converting non-standard file formats into structured machine-readable data.

Solution:

- **Table Preservation:** Marker uses deep learning to correctly extract multi-column tables from PDFs, which is critical for Settlement Reports.
- **Legacy Support:** Pandas read_fwf handles fixed-width mainframe logs (e.g., Visa TC33) commonly used in banking.

Use Case: A "Global Settlement PDF" is uploaded; Marker extracts the transaction table into a DataFrame for DuckDB, while Unstructured extracts the footnotes for ChromaDB.

RECONCILIATION ENGINE

Tech Stack: DuckDB SQL Tool.

General Utility: Executing SQL queries on local dataframes without a server.

Solution:

- **Deterministic Logic:** Uses standard SQL JOIN operations to compare datasets, guaranteeing 100% accuracy for financial balancing (unlike LLM-based comparison).
- **Scalability:** Efficiently processes thousands of rows locally.

Use Case: The tool executes a LEFT JOIN between the Authorization Log and Settlement Log to identify specific Transaction IDs present in one but missing in the other.

FORECASTING & REGRESSION

Tech Stack: Python REPL, Scikit-learn.

General Utility: Sandboxed environment for executing Python code and ML algorithms.

Solution:

- **Computational Analysis:** Allows the agent to perform linear regression and trend analysis rather than hallucinating numbers.
- **Visualization:** Generates the data points required for plotting graphs.

Use Case: The agent writes and executes a Python script to calculate a 7-day moving average of settlement volumes and predicts next week's cash flow requirements.

AUTOMATED REPORTING

Tech Stack: FPDF.

General Utility: Programmatic generation of PDF documents.

Solution:
Artifact Generation: Converts chat outputs, SQL query results, and generated insights into a downloadable, professional file format for external distribution.

Use Case: User requests a summary of discrepancies; the system compiles the SQL findings and executive summary into a "Reconciliation_Report.pdf" file.

INPUT GUARDRAILS

Tech Stack: Custom Relevance Classifier.

General Utility: Pre-processing middleware to validate input data quality and safety.

Solution:

- **Domain Validation:** Checks file headers to ensure only valid financial reports (not random user files) are ingested.
- **Prompt Safety:** Filters out malicious or irrelevant queries before they consume inference resources.

Use Case: A user uploads a generic "HR_Policy.docx"; the classifier detects the lack of payment keywords and rejects the file, preventing database pollution.

THANK YOU