GenAI-Powered Assistant for Credit Card Onboarding

Srikanth Suryawanshi

Aug 2025

Table of Contents

1 OBJECTIVE 3

2 PROBLEM STATEMENT 4

3 SOLUTION OVERVIEW 5

4 SYSTEM ARCHITECTURE 6

5.1 Query Assistant Agent (RAG Powered Policy Operator) 7
5.2 Input Checker Agent (Personal Detail Validator) 8
5.3 Document Extraction Agent (OCR + Field Structuring) 9
5.4 Application Validation Agent (Input-Document Cross-Verification) 10
5.5 Eligibility and Recommendation Agent (Rule-Based Product Matching) 11
5.6 Response Agent (Final Output Generator) 12

6 PROTOTYPE RESULTS AND DEMONSTRATION 13

6.1 Query Assistant (RAG-Based Q&A) 13
6.1.1 Knowledge Base and Query Design 13
6.1.2 Prompt and Instruction Design 14
6.1.3 Evaluation and Experiments 15
6.1.4 Limitations and Risks 15
6.2 Document Intelligence 18

7 EVALUATION AND KEY INSIGHTS 20

8 BUSINESS IMPACT AND FUTURE SCOPE 21

9 CONCLUSION 22

1 Objective

This project aims to design and prototype an AI-powered onboarding assistant that leverages large language models (LLMs) and agent-based orchestration to automate and optimize the credit card application journey. The system is designed to activate intelligent agents such as policy interpreters, input validators, and document analysers in a stage-aware manner, aligning directly with how applicants explore, fill, and submit credit card forms. By combining retrieval-augmented generation (RAG) with modular document intelligence and rule-driven validation, the assistant reduces friction, improves policy clarity, and enhances operational scalability. The solution is extensible across Indian and UAE banking contexts and serves as a reference model for applying generative AI to regulated, high-volume financial workflows.

2 Problem Statement

Financial institutions are increasingly expected to offer complex, feature-rich credit card products while maintaining fast, compliant onboarding experiences. However, product complexity has introduced a new class of operational challenges. Credit card offerings often span multiple variants with varying fees, eligibility criteria, cashback rules, reward tiers, and regional benefits. This information is typically embedded in static brochures, lengthy terms and conditions documents, or fragmented internal content systems.

For relationship managers, customer support teams, and even applicants themselves, retrieving accurate and up-to-date answers to specific product-related queries such as fee waivers, eligibility thresholds, or usage caps requires manual interpretation of policy documents. This not only introduces inefficiencies and delays but also carries the risk of inconsistent communication, misinterpretation, and downstream compliance exposure.

Generative AI systems, particularly those built using retrieval-augmented generation (RAG) frameworks, offer a compelling solution. These systems can combine semantic retrieval over institution-approved content with large language model (LLM) reasoning to generate accurate, traceable responses to user questions. When designed with appropriate grounding, instruction constraints, and citation logic, such systems can reduce dependency on human expertise for routine product support and improve response quality at scale.

Beyond query handling, another friction point in the onboarding process lies in validating applicant-submitted information against supporting documentation. Across both Indian and UAE banking contexts, applicants must upload identification, income, and residency documents, which vary in structure and formatting. Operations teams must manually extract relevant fields and verify their consistency with user-submitted forms. This process is prone to errors and inefficiencies, especially when data entry discrepancies, formatting issues, or missing values are present.

A unified solution that combines large language model based policy interpretation with automated document field extraction and rule-based validation offers a pathway to dramatically streamline onboarding workflows. By reducing manual interpretation, minimizing data mismatches, and ensuring consistency in policy communication, such a system has the potential to improve both operational efficiency and customer experience while laying the groundwork for scalable, explainable AI adoption across the enterprise.

3 Solution Overview

The proposed solution introduces a stage-aware, agent-oriented AI architecture to modernize and streamline the credit card onboarding process. Rather than treating user input, document validation, and product policy clarification as isolated tasks, the system orchestrates these functions through a pipeline of intelligent agents each responsible for a specialized capability and activated based on the applicant's progression through the onboarding journey.

At its core, the system is designed to improve both the **efficiency** and **transparency** of onboarding by leveraging large language models (LLMs) for reasoning over institutional policy documents, while simultaneously automating field-level validation of user-submitted information and documents. The architecture supports both Indian and UAE banking contexts and is built to be modular, allowing future expansion to support other geographies or products.

The process begins with a **Query Assistant Agent**, which uses a retrieval-augmented generation (RAG) framework to interpret user queries about credit card features, benefits, fees, and eligibility requirements. This agent performs semantic retrieval over bank-issued brochures, terms and conditions, and product FAQs, grounding LLM-generated answers in retrieved content to ensure accuracy and auditability. This step supports customers during their discovery phase, allowing them to explore product options with minimal manual intervention from customer service teams.

As users proceed to enter their personal and financial information, the **Input Checker Agent** performs local validations such as checking national ID formats (e.g., PAN, Emirates ID), income formatting, and required fields.

Once documents are uploaded, the **Document Extraction Agent** performs layout-aware OCR and entity extraction, converting semi-structured inputs (like salary slips, passports, Aadhaar, Emirates ID) into normalized, machine-readable fields.

These extracted fields are routed to the **Application Validation Agent**, which performs fuzzy matching and rule-based validation against the user-declared inputs. This allows the system to flag mismatches (e.g., name or date of birth differences) and assess document quality. If discrepancies are within tolerance, users are informed and permitted to proceed; otherwise, they are prompted to re-upload or provide additional clarification.

Finally, the **Eligibility and Recommendation Agent** applies bank-specific onboarding rules to determine whether the applicant qualifies for the selected product. In cases where the applicant is ineligible due to salary thresholds or document constraints, the system may recommend an alternative product that better matches the applicant's profile, providing reasoning in natural language. All results are compiled by the **Response Agent**, which generates a unified output including validation summaries, query responses, eligibility insights, and next-step recommendations.

By segmenting responsibilities across independent agents and coordinating them through a lightweight orchestration layer, the system ensures high explainability, low interdependency, and future extensibility. It also enables granular observability, compliance enforcement, and modular testing, making the solution enterprise-ready and compliant with evolving regulatory and product complexity in the financial services domain.

4 System Architecture

The system is built using an agent-oriented architecture that mirrors the natural flow of a credit card applicant's journey from product exploration to final eligibility assessment. At the core is a lightweight orchestration layer that activates context-aware agents based on user interaction at each stage. These agents are modular and independently responsible for tasks such as policy question answering, document extraction, input validation, and eligibility checking. This design ensures transparency, testability, and scalability across geographies and product types. The architecture supports gradual adoption in enterprise environments and lays the foundation for more advanced agent-based coordination in the future.

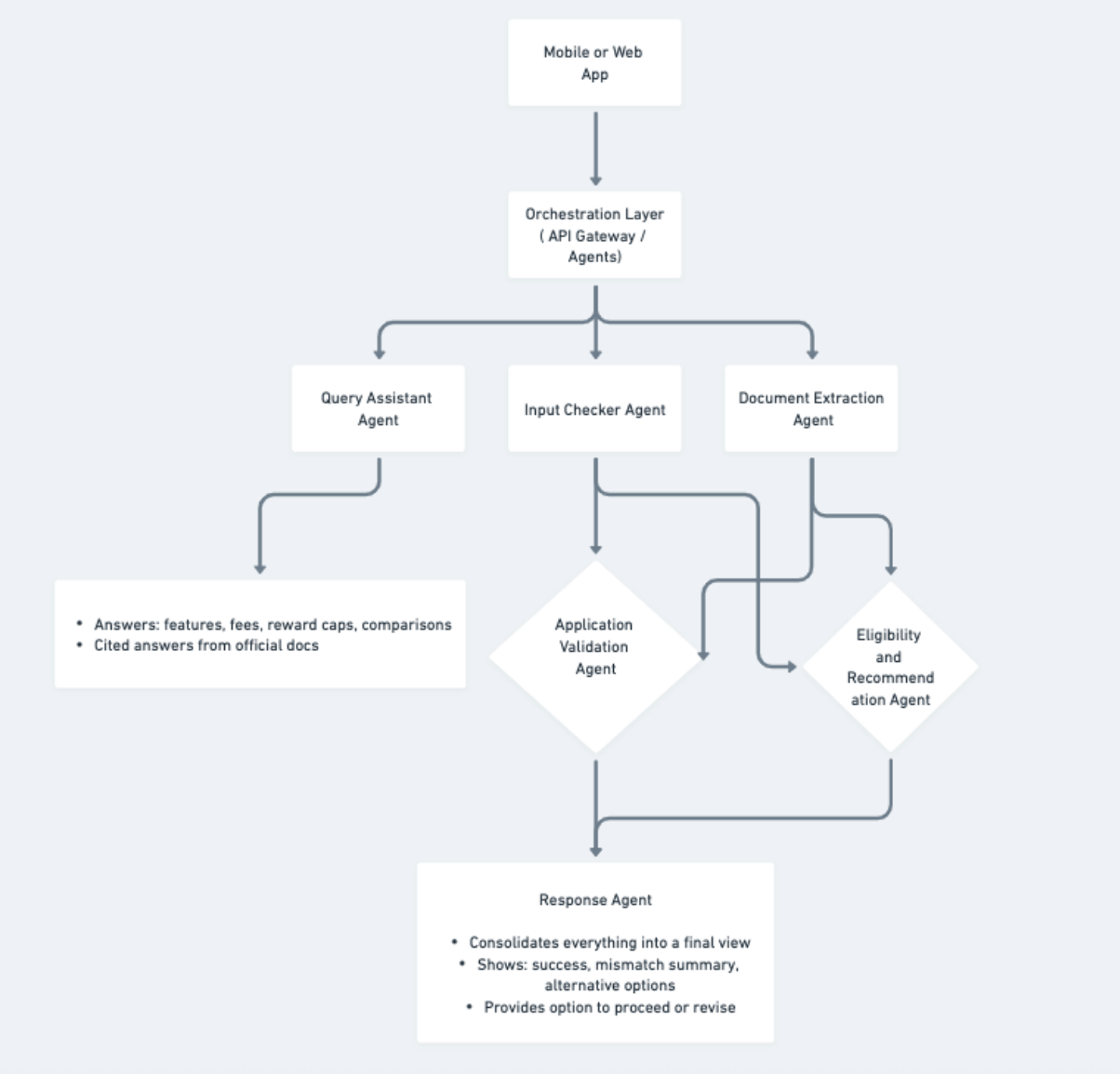


Figure 1Agent-Oriented Architecture for Credit Card Onboarding

5.1 Query Assistant Agent (RAG Powered Policy Operator)

The Query Assistant Agent is the first active intelligence component in the system and plays a critical role during the applicant's discovery and pre-application stage. Its core function is to serve as a grounded, explainable interface for interpreting product-specific policy documents such as credit card brochures, eligibility guidelines, reward structures, and terms & conditions.

This agent leverages a Retrieval-Augmented Generation (RAG) framework, which combines the linguistic fluency of a large language model (LLM) with the factual reliability of an enterprise-controlled knowledge base. Rather than generating answers from pre-trained knowledge alone, the RAG agent performs a semantic search over a document corpus using dense vector embeddings. This search retrieves relevant context passages which are then passed into the LLM along with the user's query. The model is further instruction-tuned to generate responses strictly grounded in retrieved content, reducing the risk of hallucination and improving auditability.

The knowledge base is constructed from institution-issued credit card documents brochures, product matrices, terms & conditions, fee sheets—converted into structured chunks (e.g., per section, paragraph, or table) and indexed using sentence embedding models such as all-*MiniLM-L6-v2 or bge-small-en.* A vector database (*e.g., FAISS or Qdrant)* is used to store and retrieve these chunks with high semantic fidelity. Re-ranking strategies may also be applied to improve the quality of the top-k retrieved results.

Upon receiving a query (e.g., "What is the salary eligibility for the Platinum card?" or "Is there a joining fee waiver?"), the Query Assistant Agent:

1. Converts the query into an embedding vector
2. Performs semantic search over the vector store to retrieve the most relevant passages
3. Assembles a retrieval context (with citation metadata)
4. Prompts the LLM with the query and context using a constrained instruction template
5. Returns a grounded answer with inline or post-text citations

This process allows the agent to act as a policy explainer, product comparison advisor, and eligibility clarifier, depending on the query context. It reduces dependency on frontline staff for answering common questions and provides consistent, regulatory-aligned responses across all applicant touchpoints.

The architecture is designed to be LLM-agnostic, supporting both proprietary models (*e.g., OpenAI GPT-4 via Azure*) and open-source models (*e.g., LLaMA, Mistral, or Falcon)* hosted locally or via managed endpoints. The agent supports prompt logging and result caching, enabling optimization through feedback loops and usage analytics.

5.2 Input Checker Agent (Personal Detail Validator)

The Input Checker Agent is responsible for validating the structure, format, and completeness of applicant-provided information at the early stage of onboarding. Activated once a user begins entering personal and financial details, this agent ensures that inputs such as name, date of birth, identification number, income, and contact information conform to institutional and regional standards.

The agent performs a sequence of validations ranging from field completeness and format adherence to light logical consistency checks. National ID formats such as PAN, Aadhaar (India), and Emirates ID (UAE) are verified using regular expression patterns specific to each document type. Inputs like mobile numbers and email addresses are validated for both syntax and plausibility.

To prepare for downstream document comparison, the agent also normalizes key fields removing extra whitespace, enforcing consistent date formats, standardizing capitalization, and sanitizing special characters. In cases of invalid or missing inputs, real-time feedback is provided to the applicant, prompting corrections before continuing to the next stage.
By proactively catching format errors and preparing clean, structured records for later verification, the Input Checker Agent significantly reduces mismatch errors during document validation. Its rule engine is modular and customizable to support institution-specific policies, while maintaining consistent behaviour across both Indian and UAE applicant flows.

5.3 Document Extraction Agent (OCR + Field Structuring)

The Document Extraction Agent is responsible for transforming user-uploaded identity and income documents into structured, machine-readable data that can be used for validation and eligibility checks. This agent is typically activated after the applicant has completed their form and proceeds to upload supporting documents such as Aadhaar, PAN, Emirates ID, passport, salary slips, or employment letters.

At the core of this agent is a layout-aware Optical Character Recognition (OCR) engine capable of handling complex document structures including multi-column layouts, embedded tables, and mixed language text. This layer uses AI-powered document intelligence services (e.g., Azure Document Intelligence, Google Document AI, or open-source alternatives like LayoutParser or Tesseract with layout classifiers) to extract key-value pairs and tabular data.

Once raw text is extracted, a post-processing pipeline is applied to:

- Identify and classify fields such as name, date of birth, ID number, employer name, salary amount, and document type
- Normalize extracted values into standardized formats for consistency (e.g., unifying date formats, removing currency symbols from salary)
- Preserve spatial and semantic context by maintaining relationships between labels and values, even in non-standard templates
- Assign confidence scores to extracted fields, enabling downstream agents to handle uncertainty in a controlled manner

For example, in a scanned salary slip from the UAE, the system identifies "Basic Salary", "Employer Name", and "Month" as labels and aligns them with their corresponding values. In an Aadhaar card, the system locates the applicant’s name, masked Aadhaar number, and date of birth with bounding box references.

To support compliance and traceability, all extracted fields are stored along with their source coordinates, OCR confidence levels, and original cropped image snippets. This enables human verification when needed and supports auditability for regulated environments.

The Document Extraction Agent is designed to support a variety of templates across both Indian and UAE markets, and is extensible to handle new document formats via training or configuration. It outputs a structured JSON-like object containing normalized field-value pairs, which is then passed to the Application Validation Agent for comparison against user-declared input.

This agent plays a critical role in bridging the unstructured nature of real-world documents with the structured logic of automated onboarding, enabling data-driven validation and intelligent decision-making.

5.4 Application Validation Agent (Input-Document Cross-Verification)

The Application Validation Agent serves as the critical bridge between user-declared inputs and the structured information extracted from supporting documents. Its primary role is to verify the consistency and alignment between what the applicant submits through the onboarding form and what is detected from documents such as Aadhaar, Emirates ID, passport, or salary slips. This cross-verification process is essential to ensuring the authenticity of the application and reducing the likelihood of downstream rejections due to data discrepancies.

Once the Document Extraction Agent has completed its task, the Application Validation Agent receives a normalized set of fields and performs a comparison against user-provided entries. Rather than relying on exact string matches, the agent uses flexible fuzzy-matching strategies that accommodate minor formatting variations, abbreviations, and typos that are common in real-world scenarios. For example, an applicant might enter "HDFC Bank Limited" as their employer, while the salary slip states "HDFC Bank Ltd."; the agent uses a similarity score model to determine whether the match is acceptable based on pre-defined confidence thresholds.

In cases where mismatches are detected, the system doesn't immediately reject the application. Instead, the agent generates a structured validation summary indicating which fields matched, which were partially matched, and which failed altogether. This summary is accompanied by explanatory feedback that allows the user to either correct the information or acknowledge and proceed with a justification. Importantly, the agent also considers document quality and OCR confidence scores when evaluating mismatches, helping distinguish between genuine inconsistencies and extraction noise.

All validation operations are logged in a structured format for auditability, including the original input, extracted value, similarity score, and final status. This ensures transparency in how eligibility decisions are formed, and provides a solid foundation for human review if escalations are required. The output of this agent plays a vital role in informing the downstream Eligibility and Recommendation Agent, which interprets the results in the context of specific card product rules.

5.5 Eligibility and Recommendation Agent (Rule-Based Product Matching)

The Eligibility and Recommendation Agent interprets the validation results and determines whether the applicant meets the criteria for the selected credit card product. It evaluates the combined outcome of form inputs and document-based verification against a predefined ruleset associated with each product, such as minimum income requirements, accepted forms of identification, employment categories, age limits, and region-specific constraints.

This process is managed through a configurable rule engine that maps applicant data to product eligibility parameters. For instance, if a premium card requires a minimum monthly income of ₹80,000 and the applicant declares ₹45,000, the agent will classify the applicant as ineligible for that product. However, instead of simply flagging the rejection, the agent proactively searches for a more suitable alternative card that aligns with the applicant's profile. This recommendation process is powered by a weighted scoring system that ranks other products based on rule compatibility and highlights one or more viable alternatives.

Crucially, the agent does not treat this as a binary pass/fail decision. It accounts for soft failures such as a minor shortfall in income or missing optional documentation and classifies the result as "eligible with conditions," allowing the user to proceed if they acknowledge and accept the trade-offs. Each decision is accompanied by a detailed explanation in natural language, ensuring that the user or relationship manager understands the rationale.

For example, an applicant may be informed: *"You do not meet the salary criteria for the Platinum Card, but based on your current profile, we recommend the RewardsPlus Card, which requires a minimum monthly salary of ₹40,000 and offers similar benefits."*

All eligibility logic is version-controlled and region-aware, enabling the same system to function across different markets with local configurations. The agent contributes significantly to reducing drop-offs in the onboarding funnel by providing guided redirection, rather than dead-end rejections, and helps ensure that the right applicants are matched with the right products.

5.6 Response Agent (Final Output Generator)

The Response Agent serves as the final synthesis layer in the onboarding pipeline, responsible for consolidating and formatting the outputs generated by all preceding agents into a coherent, human-readable response. Rather than acting as a decision-maker itself, the Response Agent functions as a formatter, translator, and communicator ensuring that both users and internal stakeholders receive clear, context-aware, and actionable summaries of the system's findings.

Once the Eligibility and Recommendation Agent completes its evaluation, the Response Agent gathers key outputs from the input validator, document extraction layer, validation summary, and eligibility engine. It constructs a narrative that reflects the applicant's current status, highlights any issues or mismatches identified along the way, and outlines next steps in plain language. For applicants, this may include a recommendation to upload clearer documents, a note about minor discrepancies, or a suggestion to consider an alternative card product. For back-office relationship managers or case workers, the same response can be enhanced with traceable metadata, match confidence scores, and links to document crops or OCR logs.

The agent is designed to support multiple output modes either inline display within the onboarding UI, structured JSON responses for API integrations, or stylized PDF reports for manual review workflows. By keeping the formatting layer separate from validation logic, the architecture allows institutions to tailor responses for different channels (e.g., chatbot, mobile app, RM dashboard) without altering the underlying logic.

Additionally, the Response Agent ensures that all system-generated feedback is both transparent and audit-friendly. Wherever applicable, it includes source references (e.g., "as extracted from your uploaded salary slip"), explains why certain decisions were made (e.g., "your declared income falls below the minimum threshold for this card"), and presents alternate options when the application fails to qualify for the selected product. This attention to clarity and context improves user trust, reduces the need for human intervention, and enables smoother handoffs in hybrid onboarding scenarios.

In the broader architecture, the Response Agent also acts as a natural integration point for downstream systems such as CRM, core banking, or escalation queues—making it a strategic bridge between AI-powered reasoning and real-world operational workflows.

6 Prototype Results and Demonstration

This section presents the results from a working prototype that validates the feasibility of the proposed architecture. The focus was on two key capabilities: document intelligence for extracting structured fields from real-world uploads, and retrieval-augmented generation (RAG) for answering user queries grounded in official product documents. Together, these modules serve as foundational enablers for the larger agentic system.

6.1 Query Assistant (RAG-Based Q&A)

There are four broad approaches to designing a Generative AI solution for answering product-specific queries: training a model from scratch, fine-tuning an existing model, using prompting alone, and Retrieval-Augmented Generation (RAG).

Training a model from scratch is impractical for this use case because the task requires general language understanding but domain grounding in bank-specific documents. Fine-tuning could improve style and formatting but risks embedding outdated information if fees and offers change, requiring frequent re-training. Pure prompting without retrieval can produce fluent answers but lacks guarantees that content comes from up-to-date, official sources.

RAG was chosen because it keeps authoritative knowledge in a refreshable KB, letting the model generate answers only from retrieved, verified content. This design reduces compliance risk, ensures freshness when documents are updated, and avoids high costs associated with retraining.

The proof of concept was built in Azure AI Studio using:

- Model: GPT-4.1-mini
- Retrieval: Azure File Search (hybrid keyword + vector search)
- Document Format: PDFs (Product Information + T&C for each card)
- Citations: Enabled for compliance and traceability
- Grounding: Strict prompt to use only KB content unless clearly marked as interpretation

Assumptions:

- PDFs are bank-approved, text-selectable (no OCR errors), and include effective dates.
- All answers are in English.
- Product pages from the official HSBC website were converted to PDF to ensure consistent formatting with T&C docs.

6.1.1 Knowledge Base and Query Design

The knowledge base for this project was built from official HSBC India sources to ensure accuracy and relevance. For each credit card, two documents were included: a Product Information brochure and a Terms & Conditions PDF.

The Terms & Conditions documents were directly downloaded from HSBC India's official website. For the Product Information documents, the details were sourced from the credit card product pages on the official HSBC India site. The relevant sections covering benefits, eligibility, fees, cashback rules, and other features were copied into a Word document, formatted to match the structure of an official brochure, and then exported as PDF. This ensured the KB had a consistent document format across both marketing-level descriptions and legal/policy details.

The KB creation followed a staged approach:

Stage 1 – Initial KB (Live+ only)

The first version of the KB included only the HSBC Live+ Credit Card documents. This setup enabled the assistant to handle Live+-specific queries such as:

- "What is the joining fee for HSBC Live+ and when is it waived?"
- "List all cashback categories for HSBC Live+ with rates, monthly caps, and exclusions."

Stage 2 – Expanded KB (Live+ + TravelOne)

To enable multi-product comparisons and answer queries for more than one card, the KB was updated with the HSBC TravelOne Credit Card documents (both Product Information and Terms & Conditions). This expansion allowed the assistant to answer cross-card queries such as:

- "Compare the lounge access benefits of HSBC Live+ and HSBC TravelOne."
- "Which card is better for frequent international travellers?"

By structuring the KB in two stages, the evaluation was able to demonstrate both the coverage gap when documents are missing and the improvement in capability once they are added.

6.1.2 Prompt and Instruction Design

Initial instructions were intentionally strict: *answer only from the provided sources; if absent, say "Not found."* This eliminated hallucinations but also suppressed helpful scenario answers (e.g., "Which card suits frequent flyers?"). I then revised the instruction to allow careful interpretation: the model may *combine and apply facts* from retrieved passages to a user's scenario, while clearly signaling that this is interpretation and still citing the underlying facts.

The effect was noticeable. Before the change, the assistant refused or gave terse responses to scenario prompts. After the change, it produced grounded comparisons e.g., citing exact lounge benefits for each card and explaining, "for four flights per month, TravelOne's lounge access provides greater value," with the specific pages referenced. This demonstrates that instruction tuning is as important as retrieval quality for practical usefulness, provided citations remain mandatory.

I also explored model settings. At temperature 0.0, answers were crisp and repeatable ideal for compliance-sensitive contexts. At temperature 1.0, the tone became more conversational and explanatory, which can be desirable for user engagement but should be monitored.

Changes to Top-P produced subtler differences because the retrieved passages strongly constrain the token distribution; nevertheless, at higher temperatures, lowering Top-P reduced stylistic variance.

6.1.3 Evaluation and Experiments

Testing proceeded in three phases:
Phase 1 – Live+ only:

- All Live+ queries answered accurately.
- TravelOne queries returned "Not found," confirming strict grounding.

Phase 2 – Live+ + TravelOne:

- Same TravelOne queries now answered with correct details.
- Cross-card comparisons generated with citations from both products.

Phase 3 – Model settings impact:

- Temperature 0.0 → precise, consistent answers suitable for compliance.
- Temperature 1.0 → more conversational tone, useful for customer-facing marketing.
- Top P adjustments had limited visible effect but slightly influenced phrasing variety when temperature was higher.

6.1.4 Limitations and Risks

The system's reliability depends on KB freshness. If a fee changes and the PDF is not refreshed, the assistant will confidently cite outdated numbers. A basic mitigation is a monthly document review with hash/version checks and effective dates embedded in metadata. Poorly formatted or scanned PDFs can lead to extraction errors; where scanning is unavoidable, high-quality OCR and human spot-checks are recommended especially for fee tables.

From a cost and latency perspective, RAG adds a retrieval step but avoids re-training costs. In Azure AI Studio, typical end-to-end latency for this workload was within a few seconds; in a tuned deployment with caching and lightweight re-rankers, p95 response times under ~3–6 seconds are realistic for short answers. To control costs, use a smaller model for intent detection/query rewrite, reserve GPT-4.x for final answer synthesis, and cache frequent queries (e.g., "joining fee" or "late payment fee" for popular cards).

Finally, compliance requires citation-first answers, conservative temperature settings in production, and a clear disclaimer that the assistant provides information, not legal or personalized financial advice.

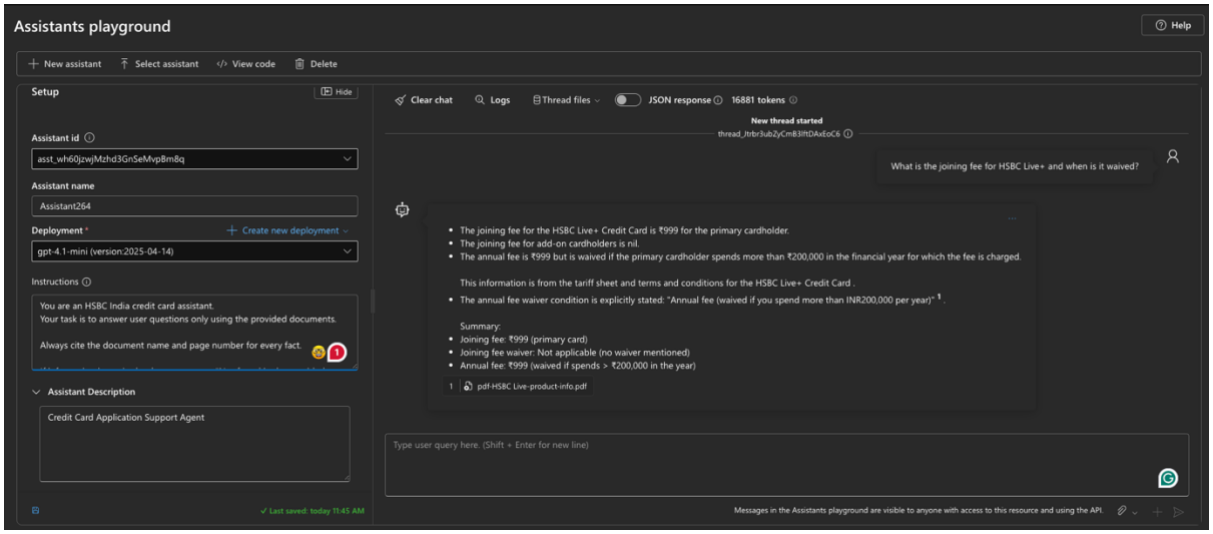
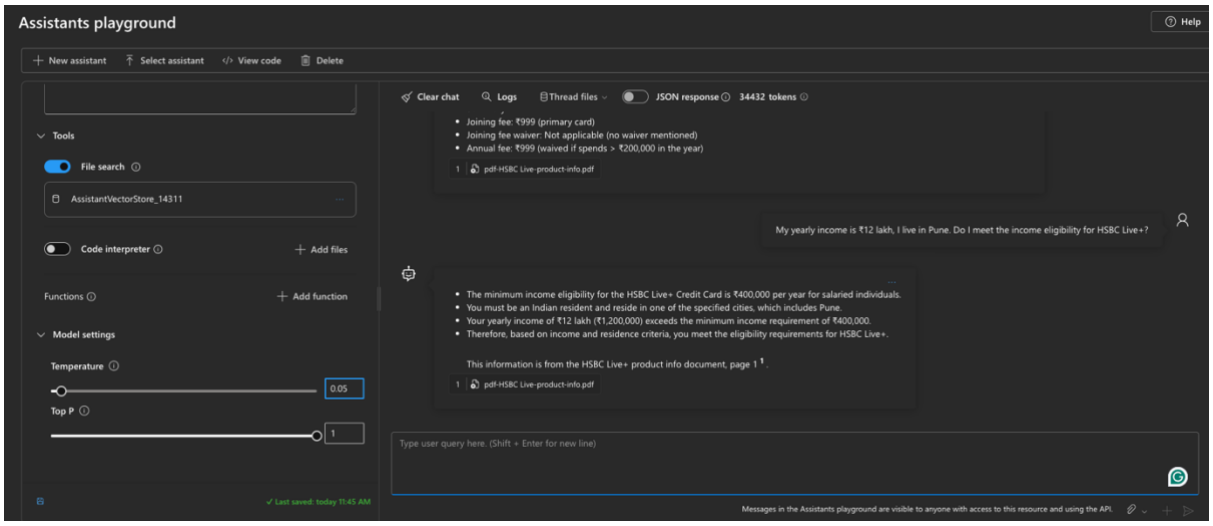
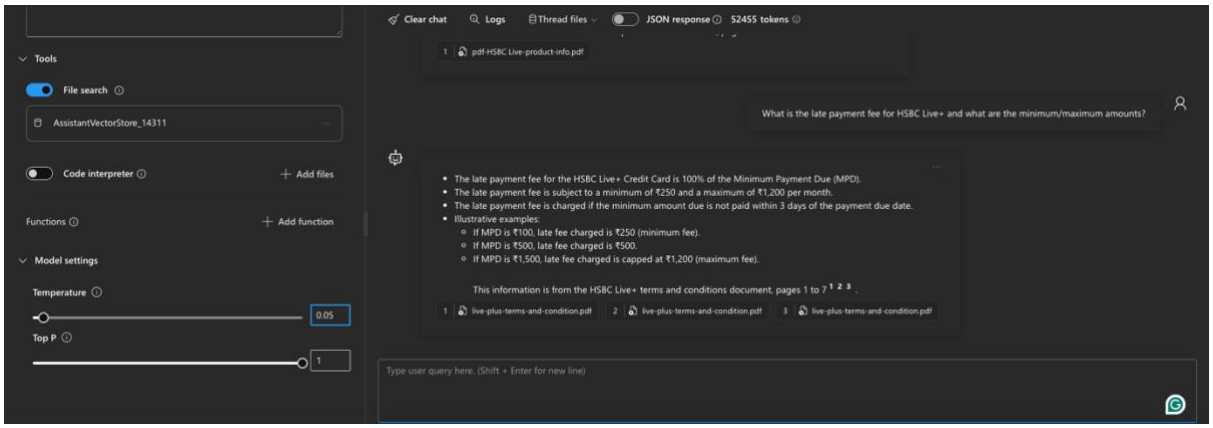
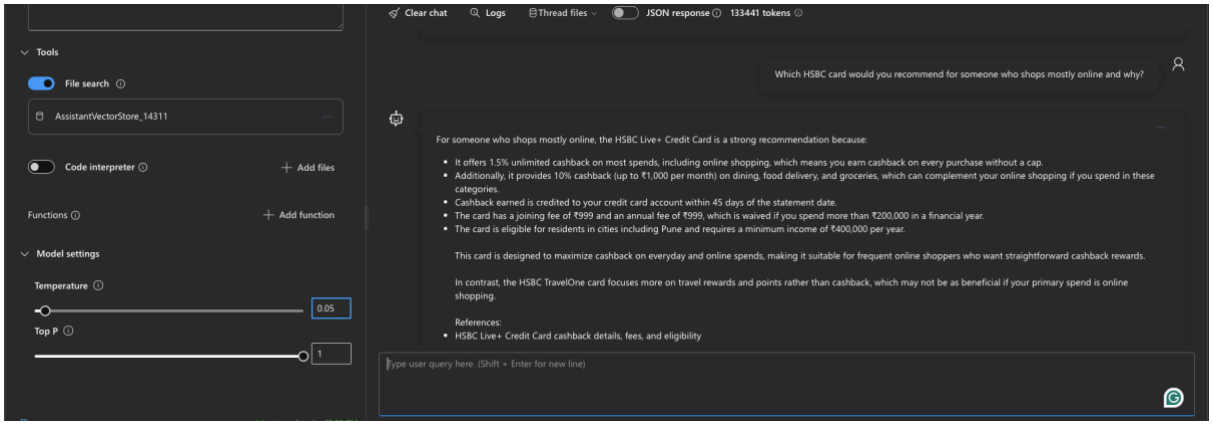


Figure 6.1 RAG Based Query and Answer






Clear chat
Logs
Thread files
JSON response
133441 tokens
Tools
File search
AssistantVectorStore_14311
Code interpreter
Add files
Functions
Add function
Model settings
Temperature
0.05
Top P
1
Which HSBC card would you recommend for someone who shops mostly online and why?
For someone who shops mostly online, the HSBC Live+ Credit Card is a strong recommendation because:
It offers 1.5% unlimited cashback on most spends, including online shopping, which means you earn cashback on every purchase without a cap.
Additionally, it provides 10% cashback (up to ₹1,000 per month) on dining, food delivery, and groceries, which can complement your online shopping if you spend in these categories.
Cashback earned is credited to your credit card account within 45 days of the statement date.
The card has a joining fee of ₹999 and an annual fee of ₹999, which is waived if you spend more than ₹200,000 in a financial year.
The card is eligible for residents in cities including Pune and requires a minimum income of ₹400,000 per year.
This card is designed to maximize cashback on everyday and online spends, making it suitable for frequent online shoppers who want straightforward cashback rewards.
In contrast, the HSBC TravelOne card focuses more on travel rewards and points rather than cashback, which may not be as beneficial if your primary spend is online shopping.
References:
HSBC Live+ Credit Card cashback details, fees, and eligibility
Type user query here. (Shift + Enter for new line)

6.2 Document Intelligence

To evaluate the feasibility of document understanding within the proposed architecture, Azure's Document Intelligence Studio was used to process a diverse set of document types including government-issued identity cards and financial documents such as salary slips and bank statements. The aim was to test the accuracy, structure, and confidence levels of key-value pair extraction that would later power the Application Validation and Eligibility Agents.

In the case of identity documents, the system successfully extracted structured fields such as First Name, Last Name, Date of Birth, Nationality, and Document Number, all with confidence scores ranging from 65% to 76%. These fields were extracted with semantic labels and bounding boxes, allowing for precise visual verification and traceability.

For example, as seen in *Figure 6.2.1*, the fields for "Cameron Savage" and ID number "UM3H4BW84" were correctly tagged along with their respective positions and confidence levels. This confirms the engine's capability to handle visual layouts typical to Aadhaar cards, Emirates IDs, and other national IDs from both India and UAE, supporting dual-region applicability.

Similarly, for financial documents, we tested a *US-format salary slip and bank statement* to simulate a common onboarding scenario. In the bank statement case (*Figure 6.2*), the engine extracted key fields such as Account Holder Name, Account Number, Statement Period, and critical numeric fields like Beginning Balance, Ending Balance, and Total Deposits.

Confidence scores ranged from 62% to 89%, and semantic labelling enabled direct mapping to application form fields such as declared income or employment history.

These extractions directly feed into the Input Checker Agent and Application Validation Agent pipelines. By comparing user-declared fields (e.g., monthly income, address, DOB) against the extracted ones, the system can detect inconsistencies, flag mismatches, and route cases for additional verification or correction. For instance, if the user enters "₹85,000" as net monthly income but the document extract returns "₹68,000," the validation logic can flag it for user attention or soft rejection thresholds.

What's most valuable is the structure and consistency of the output, making the results easily consumable by downstream agents. The prototype demonstrates that once uploaded, documents do not require complex post-processing — and can be used immediately for reasoning, validation, and recommendation in the larger onboarding pipeline.

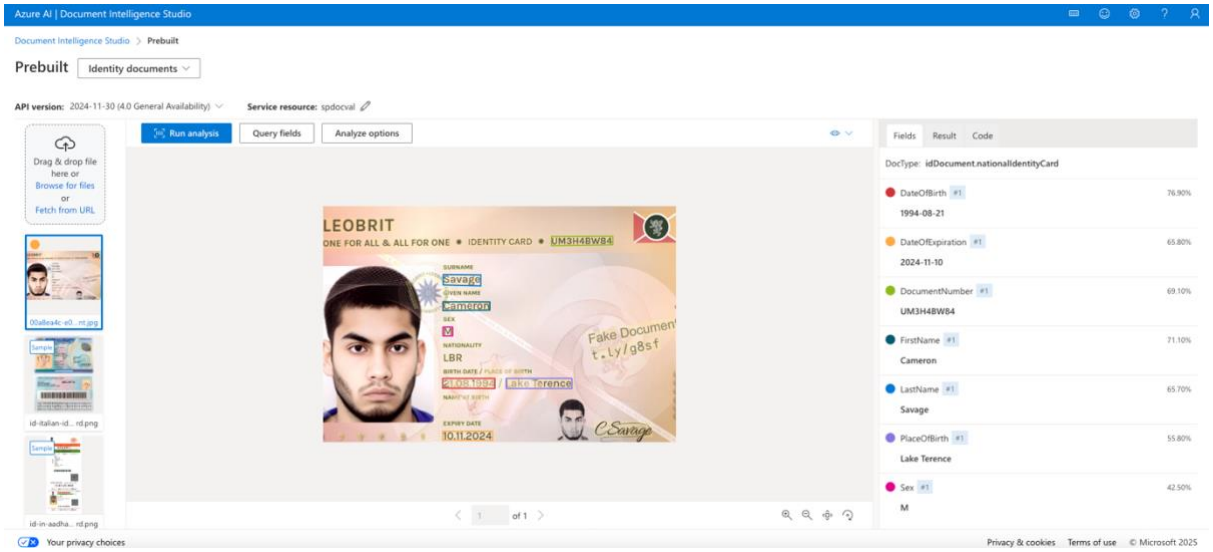


Figure 6.2.1 Identity Document Extraction using Azure Document Intelligence

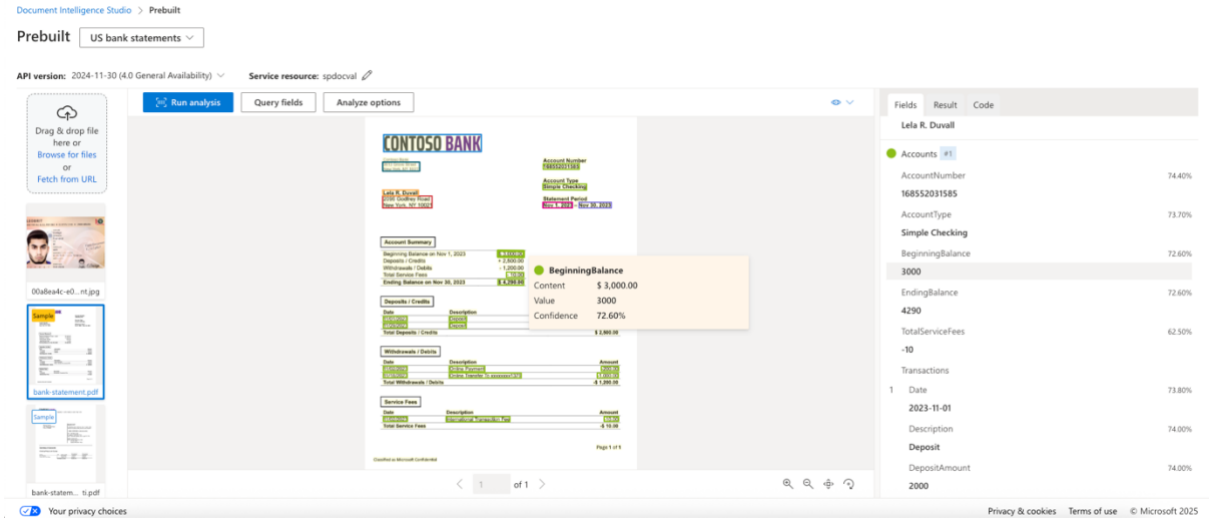


Figure 6.2.2 Financial Document Parsing - Bank Statement View

7 Evaluation and Key Insights

The prototype implementation across document intelligence and retrieval-based Q&A agents yielded promising outcomes that validate the technical feasibility and business relevance of the proposed AI-driven onboarding system. The Azure Document Intelligence results demonstrated consistent accuracy in parsing structured information from identity documents and financial statements, with confidence scores for most fields ranging between 70% to 89%. This indicates that a significant portion of user-submitted documents when clear and legible can be automatically processed with minimal human intervention, thereby accelerating the onboarding workflow.

One critical insight that emerged during document validation was the importance of document clarity. Images with blurriness, poor lighting, skewed orientation, or background noise significantly affected field recognition and reduced confidence scores. This reinforces the need for a user-facing advisory mechanism either through inline UI feedback or soft validation layers that can guide users to upload higher-quality images. Simple nudges like “Please ensure the text is readable and the document is well-lit” can dramatically improve extraction results and downstream validation accuracy.

Another key consideration was the handling of multilingual documents, especially given the dual market focus on UAE and India, where documents often include Arabic, Hindi, Marathi, or regional scripts. Current off-the-shelf models show stronger performance for Latin-script documents. To address this, future iterations could integrate custom-trained OCR pipelines or use translation bridges via LLMs to normalize and map multilingual fields into a unified format for validation. Additionally, prompt engineering within the RAG-based Q&A agent can be tuned to handle bilingual queries, allowing users to ask questions in their preferred language while maintaining answer consistency.

Overall, the prototype reinforces that with structured document input, high-quality image uploads, and controlled retrieval domains, the system can reliably extract, validate, and reason across applicant data. These learnings form a strong foundation for building an intelligent, autonomous credit card onboarding pipeline where user confidence is increased, processing cost is reduced, and decision latency is minimized.

8 Business Impact and Future Scope

The proposed GenAI-powered credit card onboarding assistant introduces measurable improvements across operational efficiency, customer experience, and compliance accuracy. By automating large portions of the information collection and validation pipeline, banks and financial institutions can significantly reduce manual effort in handling customer inputs, verifying documents, and ensuring application completeness. This directly translates to faster onboarding cycles, lower customer drop-off rates, and improved SLA adherence particularly valuable in competitive markets like India and the UAE, where customer acquisition speed is critical.

From a compliance and risk management standpoint, the system enhances transparency by maintaining traceable, explainable AI-driven decision points such as document mismatch alerts, eligibility clarifications, and cited responses from official product documents. This audit-friendly behaviour aligns well with financial sector mandates like DPDP, GDPR, and local KYC regulations, further increasing the trust quotient for regulators and customers alike.

The multi-agent orchestration allows the architecture to evolve modularly. New agents such as fraud risk detection, biometric matching, or sentiment analysis can be plugged in as needed without re-architecting the platform. This future-ready design ensures the bank can extend the solution across other financial products, such as loan applications, insurance claims, or mutual fund KYC flows.

In the short term, the solution can be used as an internal tool to aid customer service teams, reduce query response times, and flag incomplete applications. In the long term, it can be made customer-facing through mobile or web integrations, offering an intelligent and interactive onboarding journey with proactive recommendations and feedback loops.
By aligning business goals with cutting-edge GenAI and agentic technologies, the proposed assistant not only reduces costs but also elevates customer engagement and operational excellence, positioning the bank as an innovation leader in digital financial services.

9 Conclusion

This document presented a modular, agent-based architecture for building an intelligent, GenAI-powered credit card onboarding assistant capable of processing customer queries, validating user inputs, extracting document data, and offering eligibility-based recommendations. The solution was designed with practical banking workflows in mind, drawing from real-world requirements in markets such as India and the UAE, and built using proven technologies including large language models (LLMs), retrieval-augmented generation (RAG), and layout-aware document intelligence platforms.

Prototype results across multiple stages document extraction, input validation, and RAG-driven Q&A validated the core feasibility of the architecture. By leveraging structured AI agents and explainable logic layers, the system offers a scalable, auditable, and user-friendly path to intelligent onboarding. It not only streamlines operations and reduces manual overhead but also enhances user trust by providing contextual guidance, transparent mismatch reporting, and grounded responses based on official policy documents.

Importantly, this solution has been architected with long-term adaptability in mind. As document formats evolve and regulations shift, the agentic structure allows for incremental upgrades without reengineering the full pipeline. Future enhancements could include multilingual support, behavioural fraud detection, image quality scoring, or deeper integration with CRM and analytics systems.

In a rapidly transforming financial landscape, where personalization, speed, and compliance coexist as strategic priorities, this GenAI-led onboarding assistant offers a future-ready blueprint for customer-centric digital banking. By bridging the gap between unstructured documents, structured inputs, and intelligent decision systems, it sets the foundation for faster, fairer, and more transparent credit access.