

Capital One Launchpad Hackathon: Synopsis Submission

1. Team Details

Team Name: Money Farmers

Team Members:

1. Aarya Pakhale
 2. Nabayan Saha
 3. Shamik Bhattacharjee
 4. Anshul Trehan
-

2. Theme Details

Theme Name: Exploring and Building Agentic AI Solutions for a High-Impact Area of Society: Agriculture

Theme Benefits:

The agriculture sector faces complex challenges such as unpredictable weather patterns, crop diseases, fluctuating market prices, and rapidly changing government policies. By exploring and building agentic AI solutions tailored for agriculture, we can empower farmers, suppliers, and agri-businesses with real-time, context-aware insights that directly improve decision-making and productivity. These solutions can deliver weather-aware decision support, enabling accurate forecasts and localized alerts for better planning of irrigation, harvesting, and pest control. Vision-enabled AI agents can monitor crop health and detect diseases early, allowing preventive action to reduce crop loss. Market intelligence features can provide price trends and demand forecasts, helping farmers choose the most profitable time and place to sell their produce. Integrated policy and scheme navigation can simplify access to government programs and subsidies, breaking down information barriers in rural areas. With multi-channel accessibility through web, WhatsApp, SMS, and voice interfaces (including call-based voice assistance), the system ensures inclusivity for users with different levels of technological access. A continuous learning loop further refines recommendations based on evolving conditions, ensuring that the solution remains relevant and impactful over time. Ultimately, this approach supports sellers and producers in making profitable, data-driven decisions while contributing to food security and rural economic stability.

3. Synopsis

Solution Overview:

The proposed solution is an **AI-driven multi-domain query and data processing platform** that integrates diverse data sources, processes them through a unified ingestion pipeline, and provides intelligent, context-aware responses to end users which also integrates **SMS** and **Phone call** based assistance for rural India.

1. Data Sources

The system ingests data from multiple sources including:

- **Weather APIs** (real-time meteorological data)
- **Market Data feeds** (commodity prices, trends)
- **Government Portals** (policies, schemes, official announcements)
- **Google Drive** (user-provided documents, spreadsheets, images)

- **Bank Documents** (Current interest rates and loan offers of banks)

2. Ingestion & Indexing Layer

Data from all sources flows into the **Ingestion Pipeline**, where it is parsed, processed via OCR if needed, chunked, embedded, and indexed into a **Vector Database** and **Knowledge Graph**. The **Drive Connector** and **Drive Sync Manager** enable secure and continuous ingestion from Google Drive. Also for people with **no internet** connection we've integrated **SMS services** with our Backend along with a **Voice Assistant on Call**.

3. Frontend Query Flow

Users can submit **queries or uploads** via text or voice. The query undergoes **language detection and translation, refinement**, and routing to the appropriate domain agent. Also it contains two more features one is to **mark** your land by walking around your farm and also it has a feature to **upload images of crop** to detect the Issues in the crop.

4. Intelligent Agent Layer

Domain-specific agents handle specialized queries:

- **Weather Agent** – weather forecasts and advisories
- **Crop Agent** – agricultural insights, disease detection, and land marking tools
- **Market Agent** – market trends and pricing
- **Policy Agent** – policy interpretation and scheme recommendations
- **Generic Fallback Agent** – handles unclassified queries

The **Crop Agent** also triggers crop-specific tools like **Visual Language Models (VLM)** for plant health scanning and a **Field Marking UI** for land demarcation.

5. Context & Response Generation

All agent outputs are aggregated in the **Agent Aggregator** and enriched using the **Context Builder**, which pulls relevant information from the Vector DB and Knowledge Graph. A **Fact Checker** validates the information before passing it to the **Response Generator**. If confidence is low, **Fallback Logic** is applied.

6. Output & Continuous Learning

Responses are formatted for **text or voice output**, delivered to the user, and user feedback is collected. A **Continuous Learning** loop updates the Vector DB, Knowledge Graph, and agents to improve future responses.

Technical Stack:

The solution will leverage a combination of AI orchestration frameworks, cloud integrations, and multi-platform frontends to ensure scalability and flexibility. While the exact tools may evolve during development, the current planned stack includes:

1. AI Orchestration & Agent Framework

- **LangGraph** – for building and managing the multi-agent workflows, domain-specific agents, and inter-agent communication.
- **API integrations** – for image analysis, crop health detection, and other audio and visual tasks.

2. Data Storage & Indexing

- **Google Drive** – primary source for user-uploaded documents, images, and spreadsheets.
- **MongoDB** – primary structured data store for ingestion metadata, user profiles, and query logs.
- **Vector Database (FAISS)** – for semantic search and context retrieval.
- **Knowledge Graph** – for entity and relationship modeling.

3. Backend Processing & APIs

- **Python** – core backend language for ingestion pipelines, OCR, embedding generation, and agent execution.
- **FastAPI** – for API endpoints and integration with frontend and WhatsApp bot.
- **OCR Libraries** (Tesseract) – for text extraction from images/PDFs.

4. Frontend & User Interaction

- **React.js** – primary web frontend for query input, visualizations, and land marking tools.
- **WhatsApp Bot** – for conversational query submission and receiving responses
- **SMS Bot** – for people with no internet connection or very slow internet connection
- **Voice Assistant on Call** – Easier to use for communication for people with low technical knowledge

5. Cloud & Infrastructure

- **Docker** – containerization for deployment consistency.
- **Cloud Hosting** (Azure) – to host APIs, databases, and vector stores.

Decision Rationale:

The technology stack was selected to balance scalability, modularity, and adaptability, with an emphasis on enabling rapid prototyping while keeping the door open for future optimizations. LangGraph was chosen as the agent orchestration framework because of its ability to manage multi-agent workflows, maintain stateful interactions, and integrate seamlessly with both structured and unstructured data pipelines. Google Drive was selected as a primary ingestion source due to its widespread use among stakeholders and ease of integration through existing APIs, while MongoDB was preferred for its flexibility in storing semi-structured data such as ingestion metadata, agent logs, and query results.

For vector search, the plan includes integrating a specialized vector database to enable fast and contextually relevant retrieval, ensuring that semantic search and context building remain performant even as the dataset scales. React was selected for the frontend to create an interactive, responsive web interface, while the addition of a WhatsApp bot ensures accessibility for users with limited internet bandwidth or technical literacy. The backend will be built in Python with frameworks such as FastAPI or Flask for rapid API development, making it straightforward to integrate vision APIs for crop health detection and other visual tasks.

Key assumptions include consistent availability of core APIs (weather, market data, government portals) and reliable data sync from Google Drive. Constraints include the need to optimize for rural network conditions, ensure multilingual support for diverse user bases, and maintain low latency for real-time decision support. Some technology decisions are intentionally kept flexible, allowing substitution of components such as the vector database, hosting provider, or OCR engine if performance benchmarks or project requirements evolve during implementation.

Innovation Highlights:

SMS Feature for Low Internet Connection – The platform integrates an SMS-based communication system that works even in low-connectivity areas. Users can receive alerts, send commands, and interact directly with AI agents via SMS, ensuring operational continuity without relying on high-bandwidth internet.

Voice Assistance Over Call – The system includes an automated voice-response interface that allows authorized personnel to call a dedicated line and receive real-time traffic updates, congestion alerts, and predicted traffic conditions. This feature supports on-ground teams who may not have access to screens or internet-connected devices.

Multi-Agent AI Orchestration – The solution uses LangGraph to coordinate specialized domain agents for weather, crop, market, and policy insights, enabling them to collaborate dynamically based on the user's query and context. This orchestration ensures that responses are not siloed but instead combine expertise across domains for more accurate and actionable outputs.

Vision-Enabled Field Intelligence – By integrating vision-enabled agents for crop health detection and land marking, the platform introduces proactive, on-the-ground intelligence. This allows early detection of diseases and accurate field mapping, reducing the time between issue identification and intervention.

Multi-Modal, Multi-Channel Interaction – Users can interact with the system through a responsive web app, voice interface, or WhatsApp bot, ensuring accessibility even in low-connectivity rural areas. This inclusivity maximizes adoption among diverse user groups with varying levels of technological literacy.

Intelligent Data Ingestion & Retrieval – The ingestion pipeline processes both structured and unstructured data, applying OCR, chunking, and embedding before indexing it into a vector database and knowledge graph. This enables high-speed semantic search and context-rich query resolution.

Continuous Learning and Adaptation – Feedback loops update the system's vector database, knowledge graph, and agent behavior over time, ensuring that recommendations evolve with changes in environmental conditions, market trends, and policy landscapes.

Feasibility and User-Friendliness:

The solution is designed with both technical feasibility and end-user adoption in mind, making it realistic to implement and scalable for long-term use. By leveraging mature, well-documented frameworks like LangGraph for agent orchestration, React for frontend development, and MongoDB for flexible data storage, the development process benefits from proven technologies with active community support. The modular architecture allows incremental deployment, meaning core features such as query handling and weather insights can be launched early, with additional modules like vision-based crop health detection added progressively. This phased approach ensures quick wins while managing complexity.

From a user adoption perspective, the system focuses on minimizing barriers to entry. Farmers and agricultural stakeholders can access it through familiar channels like WhatsApp, alongside a responsive web application for richer interactions. Multilingual support and voice output make the platform inclusive for users with varying literacy levels. The integration of intuitive tools like the Field Marking UI and automated translation ensures that even non-technical users can engage with advanced AI features without requiring training.

Operational efficiency is achieved through an optimized ingestion pipeline, fast semantic retrieval via a vector database, and domain-specific agents that reduce unnecessary processing. The feedback-driven continuous learning loop keeps the system's advice relevant over time, increasing trust and repeat usage. These factors, combined with the adaptability of the tech stack, position the solution for lasting success and widespread adoption in the agricultural sector.

Success Metrics:

User Satisfaction – Measured through feedback scores, repeat usage rates, and Net Promoter Score (NPS), this metric reflects how well the platform meets the needs of farmers, sellers, and other stakeholders. High satisfaction levels will indicate that the system is delivering valuable, relevant, and easy-to-understand insights that users trust and return to.

System Performance – Tracked via response latency, query resolution accuracy, and uptime percentage, this ensures the platform operates reliably even under high loads or in low-connectivity environments. Consistent, fast, and accurate responses will be critical to maintaining user confidence in the solution.

Adoption and Engagement – Monitored through the number of active users across channels (web, WhatsApp, and voice), frequency of queries, and diversity of features used. Strong adoption rates and sustained engagement will demonstrate the platform's real-world relevance and usability across diverse user groups.

AI Effectiveness – Assessed through precision and recall in information retrieval, accuracy of crop health detection via vision agents, and the success rate of providing contextually correct answers in multi-agent orchestration. Higher AI effectiveness translates into more actionable and trustworthy insights for decision-making.

Maintainability and Scalability – Evaluated through update turnaround time, ease of integrating new data sources or agents, and system stability after updates. A maintainable and scalable system ensures long-term adaptability, allowing the platform to evolve alongside technological advancements and changing agricultural needs.

4. Methodology/Architecture Diagram

The architecture of the solution is designed around a modular, agent-based workflow that connects multiple data sources to intelligent domain-specific agents, context enrichment modules, and multi-channel user interfaces. The diagram below represents the end-to-end data and interaction flow:

Links:

1. <https://drive.google.com/file/d/1uCWfVAlIcMskm79U9eEyQUkMTseMiecw/view?usp=sharing>
 2. (Mermaid) <https://www.mermaidchart.com/app/projects/683644a2-e0dd-48ed-90a5-315c89853f6d/diagrams/3abac31e-c837-48e2-85c4-62bc2bc0af06/version/v0.1/edit>
-