**Project Title:** SIP Goal Planning & Fund Recommendation Platform  
**Author/Team:** Shiv Shankar Dutta  
**Date:** 20 Oct 2025

**1. Executive Summary**

The **SIP Goal Planning & Fund Recommendation Platform** is an **AI-powered financial advisory solution** that provides users with end-to-end support for:

* Planning **Systematic Investment Plans (SIP)**
* Receiving **personalized mutual fund recommendations**
* Viewing **consulting-grade, interactive reports** with allocations, projections, rebalancing, and tax strategies

The solution is built using a **multi-agent orchestration framework (AgentLoop with Gemini LLM)**, providing **explainability** through stepwise logs and **session-aware reporting**.

Technologies used:

* **Frontend:** React 18 for dynamic user interaction
* **Backend:** FastAPI for orchestration & APIs
* **Agents:** Domain-specific (SIP & Fund) + Base Agents (Planner, QA, Formatter, Report Generator, etc.)
* **Deployment:** Azure App Service with Azure Container Registry for CI/CD

This platform demonstrates how **agentic LLM systems** can enhance personal finance advisory services by delivering **transparent, explainable, and scalable** recommendations beyond static calculators.

**2. Objectives**

1. **Personalized Financial Planning** – Enable users to generate SIP projections tailored to their age, goal, horizon, and risk appetite.
2. **Intelligent Fund Recommendation** – Rank mutual funds based on risk-weighted scoring, overlap reduction, and performance heuristics.
3. **Explainability via Agents** – Provide real-time logs showing agent-level reasoning and validations.
4. **Interactive Reporting** – Generate HTML reports stored per session with charts, allocations, and narratives.
5. **Scalable Deployment** – Ensure portability and scalability through containerized Azure deployment with CI/CD pipelines.

**3. Problem Statement**

Existing SIP calculators and advisory tools face three major shortcomings:

* **Limited Personalization:** Most calculators use static formulas, ignoring user-specific contexts like tax planning, rebalancing goals, or investment horizon shifts.
* **Lack of Transparency:** Outputs are presented without showing intermediate steps, reasoning, or validation.
* **Scalability Issues:** Legacy deployments struggle with cloud-native scalability and automation.

This project addresses these gaps by combining:

* **AI Agents for reasoning & explainability**
* **Streaming logs for transparency**
* **Session-aware storage for personalization**
* **Cloud-native deployment for scalability**

**4. Literature Review**

* **Traditional Bank SIP Tools:** Primarily formula-driven, provide simple projections without context or advanced analytics.
* **AI in Finance:** Research highlights the potential of Large Language Models (LLMs) to provide reasoning, summarization, and data-driven recommendations.
* **Gap Identified:** No modular, multi-agent systems currently exist that combine SIP planning, fund recommendations, explainable workflows, and real-time reporting.

**5. System Architecture Breakdown**

**Diagram**

A diagram of a computer

AI-generated content may be incorrect.

**Component Layers**

**1. User Layer**

* End users log in and submit SIP planning requests via the **React UI**.
* Inputs: goal type, age, investment horizon, risk appetite, and currency.

**2. Frontend Layer (React)**

* **Login.jsx:** Captures user ID and session metadata.
* **react\_sip\_form.js:** Collects SIP details, triggers API requests, handles **Server-Sent Events (SSE)** for logs.
* Displays interactive HTML reports and allows navigation between SIP & Fund results.

**3. Backend Layer (FastAPI)**

* **fastapi\_sip\_service.py:** Handles SIP calculations, fund recommendations, and report serving.
* **agent\_stream\_service.py:** Orchestrates AgentLoop workflows with live streaming logs.
* Integrated with **Azure App Service** and **Azure Container Registry** for deployment.

**4. Orchestrator Agent**

* Acts as the **“conductor”** of the multi-agent workflow.
* Routes requests between SIP vs Fund agents.
* Ensures error handling, retries, and task sequencing.

**5. Domain-Specific Agents**

* **SIP Agent:**
  + Projects investment returns
  + Generates allocation plans
  + Suggests rebalancing and tax optimization strategies
* **Fund Agent:**
  + Ranks funds + LLM-based reasoning
  + Screens for portfolio overlap
  + Produces ranked fund lists for recommendations

**6. Base Agents (Reusable AI Agents)**

* **Planner Agent:** Breaks down complex user queries into smaller tasks
* **Coder Agent:** Automates code and formula execution
* **Formatter Agent:** Prepares structured outputs in JSON/HTML
* **Report Generator Agent:** Creates full HTML reports
* **Clarification Agent:** Resolves ambiguous user inputs
* **QA Agent:** Validates results before delivery
* **Thinker Agent:** Provides logical reasoning and multi-step problem solving

**7. Storage Layer**

* Reports stored per session under media/generated/{session\_id}/
* Two report types:
  + SIP Goal Planning Report
  + Fund Recommendation Report
* Reports include: charts, projections, tax optimization, allocations

**6. Features**

* **FastAPI Backend:** APIs for SIP & fund workflows, report serving
* **Multi-Agent Orchestration:** Modular execution of agents with distinct roles
* **Retrieval & Knowledge Integration:** FAISS index for semantic search
* **Streaming Logs:** Real-time transparency from backend to UI
* **Session Management:** Unique session IDs ensure secure, personalized storage
* **React UI:** SIP form, login, streaming progress, and HTML report previews
* **Deployment Scripts:** PowerShell scripts for full deployment (deploy.ps1) and incremental syncs (sync-up.ps1)

**7. Implementation Details**

**Backend (FastAPI)**

* **Endpoints:**
  + /api/calculate-sip → Runs SIP workflow
  + /api/fund-recommendation → Runs Fund workflow
  + /api/reports/{filename} → Serves HTML reports
* **Agent Stream Service:** Executes workflows and sends SSE logs

**Orchestration (AgentLoop)**

* **PlannerAgent → SIPGoalPlannerAgent/FundRecommendationAgent → Base Agents**
* SIP Orchestrator prompts dynamically filled via Jinja2 templates and Fund Orchestrator prompts is dynamically generated by filling placeholder using LLM
* **ReportGeneratorAgent** consolidates outputs → HTML

**Frontend (React)**

* **Login.jsx:** Captures identity & session
* **react\_sip\_form.js:** Collects SIP details, fetches reports, streams logs
* Provides **navigation controls** for switching between SIP & Fund reports

**8. Project Structure**

backend/ → FastAPI services

agentLoop/ → Agents and orchestration logic

prompts/ → Prompt templates for each agent

frontend/ → React UI

media/generated/ → Session reports

requirements.txt, package.json

**9. Testing & Validation**

* **Unit Tests:** Validated SIP formulas, JSON parsing, report serving
* **Integration Tests:** End-to-end SIP → Fund workflows tested
* **UI Tests:** Form validation, progress tracking, report previews
* **Performance Testing:** Azure App Service tested for load handling
* **Known Issues:**
  + Azure Basic plan instability under heavy load
  + Gemini API key expiry (daily resets observed)

**10. Results**

* Users can:
  + Plan SIP goals with personalized projections
  + Generate fund recommendations ranked by risk & performance
  + View session-specific HTML reports with interactive charts
* Reports generated with **charts, allocations, projections, and tax strategies**
* Multi-agent reasoning successfully demonstrated via streamed logs

**11. Deployment**

* **Scripts:**
  + deploy.ps1 → Builds & deploys containerized app to Azure
  + sync-up.ps1 → Pushes incremental updates
* **Flow:**
  + Docker image built locally → Pushed to ACR → Deployed to App Service Plan
* **Monitoring:** Logs tailed using az webapp log tail

**12. Future Enhancements**

* **Workflow Orchestration:** Integrate with Airflow / Kafka for scheduling and event-driven workflows
* **Fault Tolerance:** Implement retries, checkpointing, and circuit breakers
* **Retriever Optimization:** RAG caching with delta updates
* **Performance Improvements:** Native SIP math + chart rendering in optimized code
* **Memory:** Add short-term & persistent memory for personalization
* **Admin Dashboard:** Monitoring, compliance, and audit trails
* **Domain Collaboration:** Fine-tuning with financial experts for better heuristics

**13. Conclusion**

The project successfully demonstrates a **modern, AI-driven financial advisory system** combining:

* **Explainable multi-agent reasoning**
* **Personalized SIP & fund workflows**
* **Interactive reporting**
* **Cloud-native deployment on Azure**

It provides a foundation for a future **robo-advisory ecosystem** that is **transparent, scalable, and user-centric**.

  ❤️✨ Made with love by ❤️Claude , Gemini 2.5 Pro and Azure Cloud 🤖💡