**Phase 2: Portal Development & Documentation**

**Team Members:** Abdulaziz, Hafsa Nawaz, Grace Sam, Riddhima Singh

# **Scope**

Consumer-Oriented Financial Planning Portal – focusing specifically on Investment Planning and Cash Flow Management.After studying popular financial tools like YNAB (You Need A Budget) and Empower Personal Dashboard, it became clear that ordinary users, not financial experts, form the largest and most underserved user base.  
These users face practical challenges like:

* Planning monthly savings to meet goals (e.g., buying a home, saving for education, building an emergency fund).
* Managing their cash inflows and outflows without professional financial advisors.
* Understanding the impact of daily spending habits on their long-term wealth-building goals.

While mortgage optimization, insurance products, and advanced investing tools serve niche or high-income audiences, cashflow tracking and savings planning are universal needs for a much wider demographic.

1. **Portal Focus Areas**

* Short-Term Financial Planning: Helping users save for specific goals within 1-3 years (e.g., $10,000 savings in 12 months).
* Long-Term Financial Planning: Structuring simple investment advice aligned with longer horizons (e.g., retirement savings, wealth accumulation).
* Daily/Monthly Budgeting: Offering users dynamic spending and saving advice based on real transaction patterns.
* Accessible AI Advisory: Instead of complex financial jargon, providing clear, simple, and actionable advice generated by specialized agents.

This focus ensures that the portal:

* Remains easy to use for non-expert users.
* Prioritizes practical, day-to-day financial actions.
* Bridges the gap between cash flow awareness and investment decision-making.

# **Conceptual Design of Portal Workflow**

[User Inputs] -> [Sales Agent -> Bank Summarizer Agent] -> [Budget Planner Agent] ->

[Investment Planner Agent -> Reviewer Agen -> Refiner Agent] ->

[Final Summary Display]

# Agents specialize in individual financial tasks but collaborate seamlessly through multi-agent communication.

# Details on agent roles

1. User Inputs & Sales Agent

* Actor: User  
  Goal: Provide short-term and long-term financial goals, along with current financial data.
* Flow:
  + User enter short-term and long-term financial goals into a form.
  + User uploads a CSV file of their recent bank transactions.
  + The Sales Agent summarizes the bank statement using LLM summarization patterns (Summarizer Agent).
  + The information is stored and prepared for downstream analysis.
* Design Patterns Involved: Task Planning, Summarization, RAG (Retrieval if external data were added)

1. Investment Planner Agent

* Goal: Create two investment strategies (aggressive and conservative) tailored to their goals and risk tolerance.
* Flow:
  + Investment Agent reads the user profile and financial goals.
  + Generates two portfolio suggestions (aggressive vs conservative) using financial assumptions.
  + Reviewer Agent critiques the investment suggestions based on financial best practices.
  + Refiner Agent adjusts the recommendations based on reviewer feedback.
  + Final investment plan is presented to the user.
* Design Patterns Involved: Critiquing, Iterative Refinement, Planning, Human Reflection (optional if user feedback loop is added).

1. Budget Planner Agent

* Goal: Create a budget plan that balances current expenses while saving toward goals.
* Flow:
  + The Budget Agent analyzes the user's summarized financial profile.
  + Creates a budgeting strategy prioritizing recurring costs and suggesting expense reductions.
  + Reviewer Agent checks the practicality and realism of the budget.
  + The Refiner Agent finalizes and polishes the plan before presenting it to the user.
* Design Patterns Involved: Planning, Critiquing

1. Final Summary Display

* Goal: View an integrated summary of both investment and budgeting strategies.
* Flow:
  + Streamlit orchestrator checks if both investment and budget plans are generated.
  + Combines short-term (budget) and long-term (investment) strategies into a coherent final financial plan.
  + Displays the final recommendations to the user with explanations.
* Design Patterns Involved: Task Planning, Multi-Agent Collaboration.

# **D. Pattern Implementation Rationale**

After reviewing the Agent Design Pattern Catalogue, we selected **seven** agent patterns to implement based on their ability to improve reasoning, self-checking, collaboration, and modularization:

### 1. Multi-Agent Collaboration Pattern

* Usage in Portal: Our system employs multiple specialized agents, such as the Sales Agent, Budget Planner, Investment Planner, Reviewer, and Refiner, that work collaboratively to achieve the overall goal of providing personalized financial planning.
* Role: This pattern facilitates the decomposition of complex tasks into subtasks, each handled by different agents, promoting modularity and scalability.​

### 2. Planning Pattern

* Usage in Portal: The Investment Planner and Budget Planner agents develop step-by-step plans to meet user-defined financial goals.
* Role: This pattern enables agents to formulate and execute multi-step plans, allowing for dynamic and goal-oriented behavior.​

### 3. Reflection Pattern

* Usage in Portal: The Reviewer Agent evaluates the outputs of the Investment Planner, providing feedback for improvement.
* Role: This pattern allows agents to assess their own performance, facilitating self-improvement and error correction.​

### 4. Tool Use Pattern

* Usage in Portal: Agents utilize external tools and APIs, such as financial data retrieval services, to enhance their capabilities.
* Role: This pattern enables agents to extend their functionality by integrating external tools, allowing for more informed decision-making.​

### 5. User Proxy Pattern

* Usage in Your Portal: The Sales Agent acts on behalf of the user, coordinating interactions among various agents to fulfill user requests.
* Role: This pattern allows an agent to represent the user within the system, managing tasks and communications to achieve user goals.

# Why These Patterns?

# Compared to simpler direct-answer agents, this design gives better quality of responses (and hence better user experience) by:

# Mimicking real-world financial advising workflows.

# Improving transparency and explainability.

# Handling the user needs modularly and adaptively.

# Aligning with industry-grade AI agent system design trends.

**E. Reflection**

Building the Personal Financial Planning Portal offered an insightful journey through the landscape of agent-based design, LLM orchestration, and real-world tool selection. The project involved deliberate decision-making around technology choices and iterative learning that shaped the final product.

Tool Selection Trade-Offs

Selecting the right tools was crucial to the portal’s success. We evaluated several frameworks: Autogen (Python), LangChain and CrewAI. Ultimately, we chose Autogen due to its strong support for multi-agent workflows and for learning purposes.

However, there were trade-offs. Autogen’s default Docker dependency complicated local deployment, requiring manual configuration (use\_docker=False). Autogen provided more fine-grained control over agent conversations and behavior — at the cost of higher development time and complexity. In hindsight, for a simpler portal, a no-code tool might have accelerated development, but Autogen’s flexibility was necessary for implementing advanced design patterns like Critiquing and Iterative Refinement.

Streamlit was chosen for the front-end interface because of its simplicity, integration with Python, and ability to quickly deploy interactive UIs.

Ethical Concerns

Several ethical issues surfaced while developing a financial portal. First, accuracy and reliability of model outputs were critical. A hallucinating LLM giving poor investment advice could harm users financially. To mitigate this, we added a Reviewer Agent and Refiner Agent — layers of validation that mimic human oversight.

Secondly, privacy and data security were significant concerns. Users upload sensitive financial data (bank statements). Although the portal runs locally in the prototype, any production system would require encryption, secure data storage practices, and transparent data usage policies to maintain trust and comply with regulations.

Bias was another ethical issue. Pre-trained LLMs can carry implicit biases, which could impact investment recommendations unfairly. Full mitigation would require fine-tuning models on diverse financial datasets — a longer-term solution.

Finally, user agency was emphasized. Instead of dictating choices, the portal provides multiple options (aggressive vs conservative investment portfolios) and explanations, ensuring the user retains final decision-making authority.

Lessons Learned

This project emphasized the real-world gap between theoretical design and practical engineering. Implementing even well-known agent patterns (e.g., Critiquing, Refinement) required careful coordination between state management (via st.session\_state) and agent message flows. It became evident that modular, loosely coupled agents are easier to debug, maintain, and extend than tightly interwoven logic.

Another lesson was about tool maturity. Open-source LLM models like Mistral-7B provided a good balance of cost (free) and capability, but careful system prompting was essential to avoid vague or overly generic outputs. Fine-tuning, while ideal, was outside the project scope and highlights the need for future model refinement.

Finally, this project reinforced the importance of human-centered design. Financial tools must not only be technically sound but also ethical, user-friendly, and supportive of real human needs. Adding reviewer and refiner agents improved the system's reliability - but also demonstrated that fully automated financial advice still needs human oversight.

Overall, this project was a deep dive into the exciting potential - and real responsibilities - of agentic AI design.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Citation*

*Code and report was generated word-to-word using GenAI with human oversight on final output.*

*Link -* [*https://chatgpt.com/share/680ee0cf-d740-8001-a305-23a887fe2ba6*](https://chatgpt.com/share/680ee0cf-d740-8001-a305-23a887fe2ba6)