INTERVIEW KICKSTART

Empowering Learners to Build Production - Ready AI Agents

Applied Agentic AI for SWEs

Capstone Project : AI Finance Assistant

Democratizing Financial Literacy Through Intelligent Conversational AI

Project Milestones

Core Milestones

1. **Initial Research & Architecture Design:**
    - Study existing financial education platforms and robo-advisors
    - Research LangChain/LangGraph/CrewAI multi-agent architectures
    - Design system architecture and agent communication protocols
    - Set up development environment and project structure

2. **Knowledge Base Development:**
    - Curate 50-100 financial education articles covering basics
    - Structure content by categories (stocks, bonds, ETFs, etc.)
    - Create glossary of financial terms
    - Prepare sample portfolio data for testing

3. **Core Agent Implementation:**
    - Implement base agent class with common functionality
    - Develop Finance Q&A Agent with RAG integration
    - Build Portfolio Analysis Agent with calculation capabilities
    - Create Market Analysis Agent with API integration

4. **Advanced Agent Development:**
    - Implement Goal Planning Agent with projection algorithms
    - Build News Synthesizer Agent (if time permits)
    - Develop Tax Education Agent (stretch goal)
    - Test inter-agent communication

5. **Workflow Orchestration:**
    - Implement LangGraph workflow for agent routing
    - Build conversation state management

- Create fallback mechanisms for errors
- Add conversation memory and context preservation

6. **RAG System Implementation:**
   - Set up FAISS vector database
   - Implement document chunking and embedding
   - Build retrieval pipeline with relevance scoring
   - Add source attribution to responses

7. **User Interface Development:**
   - Build conversational chat interface
   - Implement portfolio analysis dashboard
   - Add market overview visualizations

8. **Real-time Data Integration:**
   - Integrate Alpha Vantage API
   - Implement caching strategy
   - Handle rate limits and failures
   - Add data freshness indicators

9. **[Optional] Testing & Quality Assurance:**
   - Write comprehensive unit tests
   - Implement integration tests
   - Conduct user acceptance testing
   - Performance optimization

10. **Documentation & Deployment:**

- Create comprehensive documentation
- Record demo video
- Prepare deployment artifacts
- Final polish and bug fixes

11. **[STRETCH] MCP Server Implementation:**
    - Build Model Context Protocol server
    - Integrate with Claude Desktop
    - Document protocol usage

FAQs

Technical Implementation

1. **Which LLM should I use for the agents?**
   Google Gemini 2.0 Flash is recommended for its balance of performance and cost. The free tier (60 requests/minute) is sufficient for development. Ensure you implement proper rate limiting.

2. **How should I structure the multi-agent communication?**
   Use LangGraph's StateGraph for orchestration. Each agent should have clearly defined inputs/outputs. The workflow router should decide which agent(s) to invoke based on the user query classification.

3. **What if I can't implement all six agents?**
   Focus on quality over quantity. Implement at least 4-5 agents well rather than rushing

through all six. The core agents (Q&A, Portfolio, Market, Goal) should be prioritized.

4. **How do I handle API rate limits?**
   Implement caching for market data (30-minute TTL recommended), use exponential backoff for retries, and provide fallback mechanisms with cached or mock data when APIs are unavailable.

Domain Knowledge

1. **Do I need deep financial expertise?**
   Basic understanding is sufficient. Focus on common concepts like diversification, compound interest, and major investment types. Use reputable sources like Investopedia for reference.

2. **How do I ensure financial information accuracy?**
   Always cite sources, stick to widely accepted principles, avoid specific investment advice, and include clear disclaimers that this is for educational purposes only.

3. **What portfolio metrics should I calculate?**
   Essential metrics include: total value, allocation percentages, expense ratios, basic diversification score, and simple risk assessment based on asset types.

User Interface

1. **How complex should the Streamlit interface be?**
   Keep it clean and intuitive. Focus on 3-4 main tabs (Chat, Portfolio, Market, Goals). Ensure responsive design and clear navigation for beginners.

2. **What visualizations are most important?**
   Portfolio pie charts, allocation bar graphs, market trend lines, and goal projection charts. Keep visualizations simple and clearly labeled.

3. **How do I handle user sessions?**
   Use Streamlit's session state for conversation memory. You don't need complex authentication - simple session-based identification is sufficient.

Testing & Documentation

1. **What should my test coverage include?**
   Aim for 80%+ coverage. Test individual agents, workflow routing, RAG retrieval, error handling, and integration points. Include edge cases like malformed queries.

2. **How detailed should documentation be?**
   Very detailed. Include architecture diagrams, setup instructions, API documentation, usage examples, and troubleshooting guides. Think of it as documentation for a production system.

Submission & Grading

1. **What's most important for grading?**
   The multi-agent architecture and workflow orchestration (40% of grade). Focus on demonstrating sophisticated AI system design rather than UI polish.

2. **Can I use different technologies?**
   Yes, but document your choices. For example, you could use ChromaDB instead of FAISS, or FastAPI instead of Streamlit, but explain your reasoning.

3. **How do I demonstrate the system effectively?**
   Create a 5-10 minute video showing multi-turn conversations, portfolio analysis, real-time market data, and goal planning. Show how agents work together.

4. **What about deployment?**
   Local deployment is sufficient, but containerization (Docker) and cloud deployment readiness earn bonus points. Focus on making the system easily runnable by

evaluators.

📚 Resources & References

Technical Resources

- LangChain Documentation
- LangGraph Guide
- Streamlit Documentation
- FAISS Documentation
- yfinance API
- Alpha Vantage API

Financial Education Resources

- Investopedia for concept definitions
- Bogleheads wiki for investment principles
- SEC investor.gov for regulatory guidance
- Modern Portfolio Theory basics