Project Ideation Chatbot - Technical Architecture Assignment

Project Overview

You're designing an AI-powered conversational assistant for an education consulting company that helps high school students gain admission to top-tier US/UK colleges. This is an **internal tool for mentors** who need intelligent, personalized project recommendations for their assigned students.

Key Context

- **Domain**: Education consulting for high school college admissions
- **Users**: Internal mentors (100 total users, 50 concurrent at peak)
- **Experience**: Natural conversational interface - no buttons or forms, pure chat experience
- **Data Sources**: Airtable trackers, Google Drive documents, onboarding questionnaires, psychometric test reports
- **Scope**: Mentor-student relationships are predefined; mentors should only access their assigned students' data

---

Core Requirements

1. Conversational Intelligence

- Natural chat interface that feels like talking to an expert mentor
- Multi-turn conversations that build context and refine understanding
- Proactive follow-up questions to clarify vague requests
- Context switching capabilities ("Actually, let me ask about Sarah instead of John")

2. Data Integration & Security

- Integrate with existing data sources (Airtable ~50% of data, Google Drive for rest)
- Strict access control - mentors only see their assigned students
- Real-time data access with intelligent caching

3. Personalized Recommendations

- Highly personalized suggestions based on student profiles, interests, skills, psychometric data
- Trained on proprietary "Athena frameworks" and historical project success data
- Source attribution for all recommendations

4. Self-Learning Capabilities

- Continuous improvement based on mentor feedback and interaction patterns
- Adaptation to successful project outcomes
- A/B testing capabilities for prompt optimization

Technical Architecture Challenge

Section A: Conversational Interface Design

1. **How will you architect the real-time chat experience?**
    - WebSocket vs. polling strategies
    - Message queuing and processing pipeline
    - Handling concurrent conversations
2. **What's your approach to natural language understanding for complex project ideation queries?**
    - Intent recognition and entity extraction
    - Handling ambiguous or incomplete requests
    - Multi-turn conversation state management

3. **How will you design conversation flows that guide mentors effectively?**
    - Proactive questioning strategies
    - Context preservation across topic switches
    - Conversation repair when the AI doesn't understand

Section B: System Architecture & Data Management

4. **Where and how will you host this system?**
    - Cloud platform selection and justification
    - Containerization and scaling strategy
    - Load balancing and fault tolerance
5. **How will you retrieve and process data from multiple sources?**
    - Data pipeline architecture (Airtable, Google Drive integration)
    - Real-time vs. batch processing decisions
    - Data transformation and harmonization strategies
6. **What's your data hosting and storage strategy?**
    - Database selection for different data types
    - Vector database strategy for semantic search
    - Data lake vs. data warehouse considerations
7. **How will the bot understand user identity and enforce access controls?**
    - Authentication and authorization mechanisms
    - Row-level security implementation
    - Session management and security
8. **How will you handle context within conversations and across chat sessions?**
    - Short-term conversation memory
    - Long-term mentor preference learning
    - Cross-session context preservation
9. **What's your strategy for live data access during conversations?**
    - Caching layers and invalidation strategies

- Real-time data synchronization
- Handling data source failures gracefully

10. **How will you manage API rate limits and costs?**

    - Token optimization strategies
    - Intelligent caching to reduce API calls
    - Cost monitoring and budget controls

Section C: AI Quality & Safety

11. **How will you prevent AI hallucinations and ensure factual accuracy?**

    - Source verification and attribution mechanisms
    - Confidence scoring for recommendations
    - Fallback strategies for uncertain responses

12. **What's your approach to handling incomplete or inconsistent data?**

    - Graceful degradation strategies
    - Data quality assessment and reporting
    - Managing edge cases in student profiles

13. **How will you ensure response relevance and appropriateness?**

    - Content filtering and safety measures
    - Age-appropriate recommendation filtering
    - Bias detection and mitigation strategies

Section D: Learning & Optimization

14. **How will you implement the self-learning capability?**

    - Feedback collection mechanisms
    - Success metrics and outcome tracking
    - Model retraining and prompt optimization workflows

15. **What analytics will you implement to measure success?**

    - User engagement and satisfaction metrics
    - Recommendation effectiveness tracking

    - System performance and reliability monitoring
16. **How will you A/B test and iterate on the conversation experience?**
    - Experimentation framework design
    - Statistical significance testing
    - Feature flag implementation for gradual rollouts

Expected Deliverables

Technical Solution Design (3-4 pages max)

- **Architecture Overview**: System components and how they connect
- **Key Technical Decisions**: Technology choices with reasoning
- **Data Flow**: How information moves from sources to user responses
- **Critical Implementation Details**: Code patterns, database design, API structure for core features

---

Evaluation Criteria

Technical Excellence (40%)

- Architectural soundness and scalability considerations
- Technology choices appropriateness and justification
- Understanding of AI/ML best practices and limitations

Product Thinking (25%)

- User experience design for conversational interfaces
- Understanding of real-world constraints and edge cases
- Business value alignment and practical implementation concerns

System Design Maturity (20%)

- Security, privacy, and compliance considerations

- Production readiness and operational concerns
- Error handling and fault tolerance planning

Innovation & Problem-Solving (15%)

- Creative approaches to complex technical challenges
- Self-learning implementation sophistication
- Quality assurance and continuous improvement strategies

Submission Guidelines

- **Format**: Technical solution document (3-4 pages)
- **Focus**: Architecture decisions and implementation approach - not project management
- **Include**: Code snippets, database schemas, API designs where relevant
- **Timeline**: 24 hours

Additional Notes

- Assume access to modern AI APIs (OpenAI, Anthropic, etc.) and cloud services
- Consider data privacy regulations and student data protection requirements
- Think beyond MVP - this system needs to scale and evolve with the business
- Remember: This is a conversational AI, not a traditional web application

---

Questions? Feel free to ask clarifying questions if any requirements need elaboration.