PART 1 — SYSTEMS THINKING & DESIGN (Primary Signal)

Architecture

High level Architecture

Frontend → Backend API → Conversation Logic Layer → Database → Integrations (CRM, Analytics, etc)

Core Components

1. **Frontend (web chat)**
    - Chat interface
    - Quick reply buttons (yes/no, budget range, timeline, etc)
    - Simple feedback buttons (👍/👎)
    - Book a demo or get a quote

The goal for this would be to educate them and collect a structured data and also keep it simple for the user

2. **Backend API**
    - Manage sessions
    - Store the history of the chat
    - Call the logic layer
    - Send the data to the CRM
    - Log events
    - Every user will get a session_id
3. **Conversation Logic layer**
    - **Intent detection**
        - Low intent (browsing only)
        - Medium intent (asking feature questions, from where we collect data)
        - High intent (when they fill out “book a demo”, timelines and what their budget is)

- **Structured Signals**
  - We extract use case, budget, timeline, company size, the urgency
- **Response Generation**
  - Templates based on response
  - Give them personalized recommendations/results

4. **Data Storage**
   - **We can have 2 tables**
     - Sessions
       1. Session_id
       2. Intent_score
       3. All the gathered data like the fields
       4. Stages (educated them/ qualify/ convert)
     - Events
       1. Message
       2. Time
       3. Topic
       4. Intent score
       5. Experiment variant
5. **Integrations**
   - When the user has high intent
     - Send lead + structured signals to CRM
     - Log events to analytics
     - Notify sales if needed

Experiment & Measurement

What to test - opening style questions, when to ask for email, CTA wording, make user life easy when educating them

Metrics that would matter and why

1. Engagement
    a. Messages per session
    b. Feedback score
2. Qualifying questions
    a. % sessions with budget + timeline captured
    b. Accuracy of high intent detection
3. Business
    a. Demo requests
    b. Conversion rate

Measuring quality and not just how many people click on it, testing one major change at a time would help avoid noisy experiment according to me

Decision Logic

Flow:

User input → classify topic → intent → extract slots → choose next action

Rule vs models

Rules: routing, compliance, CRM triggers

Models: Understand free text (LLMs or DistilBERT) , extract use cases (BERT-NER), estimate intent (LR, SVM)

Tradeoffs & Risks

Technical

- Hallucination → use structured knowledge
- Latency → limit model calls
- CRM failures → async + retries

Product

- Asking for contact info too early
- Incorrect intent detection
- Optimizing for clicks over lead quality