

1) Personalized Banking & Insurance AI Advisor

Theme

Build an intelligent AI chatbot advisor that provides personalized banking and insurance recommendations for Indian consumers. The system must understand user risk profiles, analyze financial needs, and recommend appropriate BFSI products with clear rationale - all within the Indian financial landscape.

Core Challenge: Context Engineering

Participants must create an AI agent that:

- **Understands** individual financial profiles and risk tolerance
- **Analyzes** personal financial situations in the Indian context
- **Recommends** tailored BFSI products with clear reasoning
- **Explains** product benefits specific to user's circumstances

Target Audience

Primary User: Average Indian adult consumer seeking personalized financial guidance

- **Age Range:** 25-45 years
- **Income Levels:** ₹3L - ₹15L annually
- **Financial Goals:** Savings, insurance, investments, retirement planning
- **Language:** English (primary requirement)

Product Categories to Cover

1. Insurance Products

- **Life Insurance:** Term, endowment, ULIP policies
- **Health Insurance:** Individual, family floater, critical illness
- **General Insurance:** Motor, home, travel insurance

2. Investment Products

- **Mutual Funds:** Equity, debt, hybrid funds
- **Fixed Deposits:** Bank FDs, corporate FDs
- **PPF/EPF:** Tax-saving instruments
- **NPS:** Retirement planning

3. Banking Products

- **Savings Accounts:** Regular, salary, zero-balance
- **Credit Cards:** Cashback, rewards, travel cards
- **Personal Loans:** Instant loans, gold loans
- **Home Loans:** Interest rates, eligibility

Conversational Flow

Advisor: "Hello! I'm your personal financial advisor. Let me understand your profile

first." User: "I'm 28, earning ₹60,000/month, want to buy term insurance"

Advisor: "Perfect! For a 28-year-old with ₹60K income, I recommend ₹75L-₹1Cr term coverage. Here's why:

- Your annual income: ₹7.2L
- Recommended coverage: 10-15x income
- Premium: ~₹8,000/year
- Top options: HDFC Click 2 Protect, LIC Tech Term

Sample Implementation Requirements

Core Architecture Expectations

Semantic Search Implementation:

- Use embeddings to find relevant financial advice from knowledge base
- Implement similarity matching with confidence scoring
- Enhance queries with user profile context for better results

Personalized Response Generation:

- Generate contextual advice using LLM capabilities
- Integrate user profile data into response formulation
- Maintain professional financial advisor tone and expertise
- Include specific product recommendations with reasoning

Expected System Flow:

1. User profile analysis and risk assessment

2. Query understanding and context enhancement
3. Knowledge base search with personalization
4. Response generation with Indian financial context
5. Confidence scoring and source attribution

Key Deliverables

1. Working demo Chatbot Video (max 3min):
2. Github repo
3. Hosted Website url(if done)
4. PPT

2) Personalized E-Commerce Recommender Bot



Theme

Build an AI-powered recommendation assistant for Indian D2C food and fashion brands that delivers hyper-personalized product suggestions based on user context and preferences. The system must understand natural language queries, extract user preferences, and provide contextually relevant product recommendations with clear reasoning.

Core Challenge: Context Engineering

1. **Understands User Context:** Extracts key preferences (dietary needs, style preferences, budget) from natural language queries
2. **Personalized Recommendations:** Matches products to user profiles using hybrid filtering approaches
3. **Explains Recommendations:** Provides clear, brand-aligned reasoning for product suggestions
4. **Adapts to Indian Market:** Considers local pricing (₹), seasonal trends, and cultural preferences

Conversational Interface

Shopping Assistant: "Hi! I'm your personal shopping assistant. What are you looking for today?" User: "I need vegan snacks under ₹300"

Shopping Assistant: "Perfect! Here are my top recommendations for you: Plant-Based Protein Cookies - ₹299
95% Match

Why it's perfect: These gluten-free cookies fit your vegan diet and budget perfectly. With 12g plant protein per serving, they're both healthy and delicious!

Quinoa Energy Bites - ₹249

87% Match

Why it's perfect: Vegan superfood snacks under your budget with natural sweetness from dates and coconut."

Sample Queries & Expected Outputs

User Query	Top Recommendation	Confidence	Justification
"Vegan snacks under ₹300"	Plant-Based Protein Cookies (₹299)	95%	"Perfect vegan option within budget with high protein content for health-conscious snacking"
"Light ethnic wear for summer"	Handblock Print Kurta (₹1499)	92%	"100% cotton kurta keeps you cool while maintaining traditional elegance for summer occasions"
"Protein-rich breakfast options"	Oats & Chia Seed Bars (₹349)	88%	"15g plant protein per serving makes this ideal for active lifestyles and morning nutrition"
"Trendy casual wear under ₹1000"	Oversized Cotton Tee (₹799)	85%	"On-trend oversized fit in premium cotton, perfect for casual outings within your budget"

Key Deliverables

1. Working Demo Video (max 3 minutes):
2. GitHub Repository with:
3. PPT
4. Hosted Website url(if done)

3) Automated Data Filler Agent

Theme

Build an intelligent AI agent that automatically processes sales meeting summaries and populates CRM systems with structured data. The system must extract key information from unstructured text, understand sales context, and create accurate CRM entries - all within the B2B sales workflow.

Core Challenge: Context Engineering This is a Context Engineering Hackathon where participants must create an AI agent that:

- Understands unstructured sales meeting summaries and conversations
- Extracts key business entities, relationships, and deal information
- Structures data into CRM-compatible formats with proper categorization
- Integrates seamlessly with CRM platforms while maintaining data accuracy

Target Audience Primary User: B2B Sales Representatives and Sales Managers

- Company Size: SMEs to Enterprise (50-5000 employees)
- Industries: SaaS, Manufacturing, Consulting, Financial Services
- Current Pain: Spending 2-3 hours daily on manual CRM data entry
- CRM Platform: HubSpot, Salesforce, or Zoho

CRM Data Categories to Extract 1. Contact Information

- Personal Details: Name, designation, email, phone number
- Professional Info: Decision-making authority, department, seniority
 - Communication: Preferred contact method, availability, timezone
- Company Information
- Basic Details: Company name, industry, size, website
- Business Context: Current challenges, pain points, technology stack

- Market Position: Competitors, budget range, growth stage
- Deal Information
- Opportunity Details: Deal value, timeline, decision criteria
- Sales Stage: Current stage, next steps, proposal requirements
- Competitive Landscape: Alternatives being evaluated, key differentiators

Conversational Flow Agent: "Please paste your meeting summary, and I'll extract the CRM data automatically."

User: "Had a call with Sarah Johnson, Marketing Director at GrowthTech. They need marketing automation for 50-person team. Budget: \$30K annually. Evaluating HubSpot vs our solution.

Next: Demo scheduled for Friday."

Agent: "Perfect! I've extracted and structured your CRM data:

Contact Created:

- Name: Sarah Johnson
- Title: Marketing Director

- Company: GrowthTech

• Stage: Qualified

Lead Company

Created:

- GrowthTech (Marketing Industry)
- Team Size: 50 employees • Budget: \$30,000

annually Deal Created:

- Marketing Automation Implementation

- Value: \$30,000

- Stage: Demo Scheduled

- Competitor: HubSpot

- Next Action: Demo on Friday"

Sample Implementation Requirements Core Architecture Expectations Text Processing Implementation:

- Use NLP to extract entities (names, companies, amounts, dates)
- Implement relationship mapping between contacts, companies, and deals
- Handle various meeting summary formats and writing styles
- Validate extracted data against CRM field requirements

CRM Integration Implementation:

- Generate API calls to create contacts, companies, and deals
- Handle duplicate detection and data merging
- Implement error handling for failed API calls
- Maintain data consistency across CRM entities

Expected System Flow:

1. Meeting summary input and preprocessing
2. Entity extraction and relationship mapping
3. Data validation and CRM field mapping
4. API integration and CRM data creation

5. Confirmation and error handling with user feedback

Key Deliverables

1. Working Demo Video (max 3 minutes):
2. GitHub Repository with:
3. PPT
4. Website link (If deployed)

4) Personalized Learning Recommender for Coding Students

Theme

Build an intelligent AI-powered learning coach that analyzes coding solutions and provides personalized learning recommendations for college students practicing on platforms like LeetCode. The system must understand individual coding patterns, identify knowledge gaps, and create tailored learning paths with relevant resources - all focused on improving programming skills and interview readiness.

Core Challenge: Adaptive Learning Intelligence

Participants must create an AI agent that:

- **Analyzes** code quality, patterns, and problem-solving approaches
- **Identifies** specific knowledge gaps and areas for improvement
- **Recommends** personalized learning resources and study plans
- **Tracks** progress and adapts recommendations based on learning patterns
- **Generates** step-by-step improvement roadmaps for coding skills

Target Audience

Primary User: College students preparing for coding interviews and improving programming skills

- **Education Level:** Undergraduate/Graduate Computer Science students/ ANYONE who's curious
- **Experience Range:** Beginner to Intermediate programmers
- **Goals:** FAANG interview preparation, competitive programming, skill development
- **Platform Focus:** LeetCode, HackerRank, CodeChef problem solving

Problem Categories to Analyze

1. Data Structures & Algorithms

- **Array/String Manipulation:** Two pointers, sliding window, prefix sums
- **Linked Lists:** Traversal, reversal, cycle detection
- **Trees & Graphs:** DFS, BFS, traversal patterns
- **Dynamic Programming:** Memoization, tabulation, optimization
- **Sorting & Searching:** Binary search, merge sort, quicksort

2. Coding Patterns

- **Two Pointers:** Fast/slow pointer, left/right pointer techniques
- **Sliding Window:** Fixed/variable window problems
- **Backtracking:** Permutations, combinations, constraint satisfaction
- **Greedy Algorithms:** Optimization problems, scheduling
- **Divide & Conquer:** Recursive problem solving

Sample Conversational Flow

AI Coach: "Hello! I'm your personalized coding mentor. Let me analyze your recent LeetCode submissions to create a tailored learning plan."

User: *[Uploads code solution for "Two Sum" problem]*

AI Coach: "Great! I analyzed your Two Sum solution. Here's what I found:

Strengths: • Clean variable naming and good code structure • Correct algorithm implementation with $O(n)$ time complexity • Proper handling of edge cases

Areas for Improvement: • **Hash Map Patterns:** You solved this correctly, but similar problems like 'Three Sum' require advanced hash map techniques • **Space-Time Tradeoffs:** Consider when to optimize for space vs time

Personalized Learning Plan:

1. **Next Challenge:** Try 'Three Sum' (LeetCode #15) - builds on your hash map knowledge
2. **Concept Deep Dive:** Hash map collision handling and optimization
3. **Resources:**
 - YouTube: 'Advanced Hash Map Techniques' by NeetCode
 - Article: 'Hash Table Implementation Details'
 - Practice: 5 similar hash map problems

Weekly Goal: Master hash map patterns (estimated 3-4 hours)"

Key Deliverables

1. Working Demo Video (max 3 minutes):
2. GitHub Repository with:
3. PPT
4. Website link (If deployed)

5) Personalized Learning Recommender for Math & English (Grades 1-5)

Theme

Build an intelligent AI-powered learning coach that provides personalized learning recommendations for elementary school students (Grades 1-5) in Math and English, specifically designed for CBSE/NCERT curriculum. The system must assess student knowledge levels, identify learning gaps, and create adaptive daily/weekly learning plans to improve foundational skills - with special focus on Rural India's educational needs.

Core Challenge: Adaptive Elementary Education Intelligence

Participants must create an AI agent that:

- **Assesses** student knowledge levels through adaptive questioning
- **Identifies** specific learning gaps in Math and English fundamentals
- **Generates** personalized daily/weekly learning challenges and activities
- **Tracks** progress and adapts difficulty levels based on performance
- **Provides** engaging, age-appropriate learning experiences for young learners
- **Advanced Operations:** Large Numbers, Factors, H.C.F And L.C.M., Tests Of Divisibility
- **Decimals & Fractions:** Fractions, Decimals, Averages, Percentages
- **Practical Math:** Simple Interest, Profit and Loss, Time, Unitary Method
- **Geometry:** Mensuration, geometry, weights and masses, symmetry
- **Measurement:** Metric Measures, Bills, Temperature

Sample Learning Journey

Initial Assessment Flow: AI Tutor: "Hi! I'm your learning buddy! Let's start with some fun questions to see what you already know!"

Math Assessment (Grade 3 Example):

- "Can you solve: $25 + 17 = ?$ "
- "Which is bigger: $1/2$ or $1/4$?"
- "How many sides does a triangle have?"

Personalized Learning Plan: AI Tutor: "Great job! I can see you're good at addition but let's work on fractions together. Here's your learning plan for this week:

Monday - Wednesday: Fraction Fun • Activity: Pizza slice fractions game • Challenge: Compare $1/2$ and $1/4$ using pictures • Practice: 5 fraction comparison problems

Thursday - Friday: English Reading • Story: 'The Brave Little Mouse' (150 words) • Activity: Find 10 new words and their meanings • Writing: Write 3 sentences about your favorite animal

Weekend Challenge: Math + English Mix • Create a story problem using fractions • Example: 'Ram ate $1/4$ of a pizza. How much pizza is left?'"

- **Offline Capabilities:** Downloadable content for areas with limited connectivity
- **Regional Language Support:** Hindi translations and voice support
- **Cultural Relevance:** Examples and contexts familiar to rural students

Key Deliverables

1. Working Demo Video (max 3 minutes):
2. GitHub Repository with:
3. PPT
4. Website link (If deployed)

6) Prompt Analyser

Theme Build an intelligent AI agent that monitors brand visibility across major AI platforms and search engines. The system must generate relevant prompts, analyze brand mentions in AI responses, track competitor performance, and provide actionable visibility insights all within the evolving AI-powered search landscape.

Core Challenge: Context Engineering This is a **Context Engineering Hackathon** where participants must create an AI agent that:

- **Generates** industry-specific prompts that users commonly search
- **Analyzes** brand mentions and rankings across multiple AI platforms
- **Compares** brand visibility against key competitors
- **Scores** overall brand presence with actionable insights and recommendations

Target Audience Primary User: Brand Managers and Digital Marketing Teams

- **Company Size:** Startups to Enterprise (50-10,000+ employees)
- **Industries:** FinTech, E-commerce, SaaS, Healthcare, EdTech
- **Marketing Budget:** ₹5L - ₹50L annually on digital marketing
- **Pain Point:** Unknown brand visibility in AI-generated search results

AI Platform Categories to Monitor

1. Conversational AI Platforms

- **ChatGPT:** OpenAI's flagship conversational AI
- **Google Gemini:** Google's advanced AI assistant
- **Perplexity AI:** AI-powered search and answer engine

2. Search-Enhanced AI

- **Microsoft Copilot:** Bing-integrated AI assistant
- **Meta AI:** Facebook's conversational AI platform

Conversational Flow

Analyzer: "Welcome! Let's analyze your brand's AI visibility. What's your brand name and industry?"

User: "PolicyBazaar, insurance comparison platform"

Analyzer: "Perfect! I'll analyze PolicyBazaar's visibility across ChatGPT, Gemini, and Perplexity using insurance-related prompts.

Generated Prompts (20 samples):

- Best insurance comparison platforms in India
- Where to buy car insurance online
- Compare health insurance policies

Visibility Analysis:

- **ChatGPT:** 7/10 prompts mention PolicyBazaar (Rank #2 avg)
- **Gemini:** 5/10 prompts mention PolicyBazaar (Rank #3 avg)

Competitor Analysis: • Bajaj Allianz: 6/10 average mentions • HDFC ERGO: 4/10 average mentions • ICICI Lombard: 5/10 average mentions

Overall Visibility Score: 67/100"

Sample Implementation Requirements Core Architecture Expectations

Automated Prompt Generation:

- Generate 20-30 industry-specific prompts based on brand category
- Include variations: direct questions, comparison queries, recommendation requests
- Simulate real user search patterns and intent
- Cover different stages of customer journey (awareness, consideration, purchase)

Multi-Platform Analysis Implementation:

- Integrate with 3+ AI platforms via APIs or web scraping
- Extract brand mentions, rankings, and context from responses
- Identify competitor mentions and relative positioning
- Track sentiment and recommendation strength

Expected System Flow:

1. Brand profile setup and competitor identification
2. Automated prompt generation based on industry keywords
3. Multi-platform query execution and response collection
4. Brand mention extraction and ranking analysis
5. Competitor comparison and visibility scoring with actionable insights

Key Deliverables

1. **Working Demo Video (max 3 minutes)**
2. **GitHub Repository**
3. **Hosted Website url(if done)**
4. **PPT**

Submission Form Link :

[https://docs.google.com/forms/d/e/1FAIpQLSdvaZHzHXWD
K0E1p-RarOJbcT4WgZCf_SrdB7e2K0PegbMd0w/viewform
?usp=publish-editor](https://docs.google.com/forms/d/e/1FAIpQLSdvaZHzHXWDK0E1p-RarOJbcT4WgZCf_SrdB7e2K0PegbMd0w/viewform?usp=publish-editor)