



**តែងបទ Case Interview >>**

**"AI Requirement  
Interview Assistant"**



## เกริ่นนำ

สำหรับผู้สมัครที่ผ่านการสัมภาษณ์รอบแรกแล้ว เพื่อยกระดับกระบวนการคัดเลือกบุคลากรด้านเทคโนโลยี iReadCustomer จึงออกแบบ Case Interview ฉบับนี้ขึ้นมาโดยจำลองสถานการณ์ “สร้างระบบ AI Requirement Interview Assistant” ตั้งแต่ Frontend (React 19) จนถึง Backend (Python/FastAPI) พร้อมการบรรจุด้วย Docker-Compose ทั้งหมดต้องพัฒนาให้เสร็จภายในเวลา 120 ชั่วโมง

### วัตถุประสงค์

- วัดความสามารถด้าน System Design (แยกเลเยอร์ ชัดเจน ขยายได้)
- ประเมินทักษะ AI Orchestration (ออกแบบ Agent หลายตัว + Routing Logic)
- ตรวจสอบคุณภาพ Frontend UX (Design Token, Responsive, A11y, Streaming)

ท้ายที่สุด เราอยากเห็น “เหตุผล” เบื้องหลังการตัดสินใจทุกข้อ—สั้น กระชับ แต่ชัดเจน ความสามารถในการสื่อสารและจัดลำดับความสำคัญคือหัวใจของงาน

### Resource

- Design system ([Click here](#))
- Landing page website (<https://ireadcustomer.com/>)

## 1. บทนำและบริบทธุรกิจ

InsightForge เป็นบริษัทที่ปรึกษาด้านข้อมูลขนาดกลางในไทย บริษัทให้บริการวางยุทธศาสตร์ข้อมูลและทำวิจัยตลาดเชิงลึกแก่ลูกค้าระดับกลาง-ใหญ่ทั่วเอเชียตะวันออกเฉียงใต้ ลูกค้าหลักกระจายอยู่ในกลุ่ม FMCG, Fin-Tech, Retail Omni-channel และ Startup Series B ขึ้นไป

InsightForge เติบโตเฉลี่ย CAGR 28 % ต่อปี แต่เริ่มสะดุดจากการแข่งขันของผู้ให้บริการรายใหม่ที่ใช้เครื่องมือ AI เข้ามาแย่งเวลาการทำงาน Presales ได้ดีกว่า จึงเกิดโครงการ ARIA เพื่อรักษาความสามารถทางการแข่งขัน

### 1.1 ปัญหาปัจจุบัน

ทีม Presales ใช้เวลาสัมภาษณ์ลูกค้า 3-5 ชม. เพื่อเก็บ Requirement แต่ผลลัพธ์ไม่สม่ำเสมอ ทำให้เกิด Scope Creep และประเมิน Manday คลาดเคลื่อน

- Discovery Call / Workshop (3-5 ชม.) ที่ปรึกษาเก็บ Requirement ผ่านการสัมภาษณ์สดหรือวิดีโอคอล
- Manual Note-Taking - ใช้ Google Docs/Sheet ไม่มีเกมเพลตกลาง
- ส่งต่อทีมวิเคราะห์ - สรุป Scope ใน PowerPoint หรือ Excel แต่ละคนมีรูปแบบไฟล์ต่างกัน
- Pain Point - ข้อมูลสับสน, ความหมายไม่ตรงกัน, ใช้เวลาปรับเอกสารซ้ำ ทำให้เกิด Scope Creep และ Manday บวมเฉลี่ย 20 %

### 1.2 โครงการเชิงกลยุทธ์

สร้าง AI Requirement Interview Assistant (ARIA) เพื่อ

- เก็บข้อมูลเชิงโครงสร้างผ่านแชต (Chat Message)
- จัดหมวด Purpose ของโปรเจกต์
- ตั้งคำถามเชิงลึกอัตโนมัติ ตามบริบทอุตสาหกรรมและขนาดโครงการ
- สกัดข้อมูลเชิงโครงสร้าง เช่น Company Info, Business Problem, KPI, Budget Band
  - { 'COMPANY\_PROFILE', 'BUSINESS\_PROBLEM', 'BUDGET', 'PURPOSE\_OF\_PROJECTS' }
- โดยจัดหมวด PURPOSE\_OF\_PROJECTS เป็น PRODUCT\_DEV, MARKETING, MARKET\_RESEARCH อย่างใดอย่างหนึ่ง (เช่น เลือก PRODUCT\_DEV)

Driver	เป้าหมายเชิงปริมาณ	เหตุผลทางธุรกิจ
Presales Efficiency	ลดระยะเวลา Discovery $\geq 80$ %	ลดต้นทุนแรงงาน และเพิ่ม Throughput ต่อที่ปรึกษา
Quote Accuracy	คลาดเคลื่อน Manday $\leq \pm 10$ %	รักษาอัตรากำไรขั้นต้น & ลด Scope Creep
Win Rate	$= +15$ % YoY	ปิดดีลได้เร็วขณะลูกค้ายัง "ร้อน"
Knowledge Retention	เก็บ Transcript คสว 100 %	ต่อเติมเป็น Knowledge Graph ภายใน 2026



## 2. เอกสารความต้องการทางธุรกิจ (BRD)

### 2.1 วัตถุประสงค์

- เก็บ Requirement ผ่าน Conversational AI + AI Follow up
- จัดหมวด Purpose เป็น PRODUCT\_DEV, MARKETING, MARKET\_RESEARCH
- สร้าง Scope Summary (JSON) และร่าง SoW (PDF)
- ประเมิน Manday จาก Matrix ที่แก้ไขได้ (YAML)

### 2.2 คุณสมบัติในขอบเขต (In-Scope)

รหัส	ฟีเจอร์	เหตุผล
BR-01	ล็อกอิน JWT	ปกป้องข้อมูลลูกค้า
BR-02	Web Chat Real-time	UX คืบเคย
BR-03	Agentic Pipeline	โมดูลาร์ / ขยายง่าย
BR-04	Manday Matrix (YAML)	ปรับสูตรได้ตามบิงบ
BR-05	ส่งออก JSON + PDF	ป้อน CRM / ส่งใบเสนอราคา

### 2.3 นอกขอบเขต (MVP)

สิ่งที่ไม่จำเป็นต้องทำ : ระบบจ่ายเงิน, e-Signature, BI Dashboard เชิงลึก, Multi-tenant Isolation

## 3. ภาพรวมโจทย์ (Problem Overview)

ผู้สมัครต้องพัฒนา แอปพลิเคชันเว็บ ที่มี 2 ส่วน:

- Frontend SPA (React 19)
  - หน้า "Interview Room" — แชนบ็อตถาม-ตอบ,
- Backend API (FastAPI)
  - JWT Auth, File Upload
  - Agentic Pipeline + Router
  - Endpoint /ask สดริ่มคำตอบ + Citation



## 4. ขอบเขตงาน

### 4.1 MVP

หมวด	รายละเอียดขั้นต่ำ
Auth	Email / Password → JWT cookie (httpOnly)
Chat UI	React SPA + Bubble, Typing Indicator, Design Token
Agents	มี AI Model ที่ทำหน้าที่ ตามให้ได้รับ Scope ของงานตามที่กำหนด , Follow up และ ตรวจสอบ
Summary Modal	ตาราง Scope
Containerisation	Dockerfile (front/back) + docker-compose.yml one-liner

### 4.2 Stretch Goals (+1 คะแนน/ข้อ ≤10 คะแนน)

Rate-limit, Admin UI, SSE Streaming, CI Pipeline, ≥ 80 % Coverage, PDF Export, Multi-tenant, Prometheus, Public Demo, Transcript Search

## 5. ข้อกำหนดทางเทคนิค (Technical Requirements)

ด้าน	ข้อบังคับ	หมายเหตุ
Backend	Python 3.12 + FastAPI (async)	Formatter = black, Linter = ruff
Frontend	React 19 + Next.js	Tailwind/MUI/Chakra ใช้ได้
AI Model	OpenAI / Gemini / Llama 3 API	อธิบายต้นทุนใน ADR
Vector Store	ไม่บังคับ (in-mem / Chroma / pgvector)	
Testing	Pytest, React Testing Library/Vitest	ไม่ต้อง E2E Cypress
Docs	README ≤ 800 คำ, ADR ≤ 1 หน้า	Diagram เก็บใน /docs

- Svelte / Angular สามารถใช้ได้หากผู้ทำโจทย์มีความถนัดมากกว่า

## 6. ข้อมูลที่ระบบต้องรวบรวม (Required Fields)

Field	คำอธิบาย	ตัวอย่าง
COMPANY_PROFILE	ชื่อบริษัท	"ACME Foods Ltd."
BUSINESS_PROBLEM	ปัญหาที่ต้องการแก้	"ยอดขายสินค้าใหม่ต่ำ"
PURPOSE_OF_PROJECTS	PRODUCT_DEV, MARKETING, MARKET_RESEARCH	MARKET_RESEARCH
BUDGET	ช่วงงบ	"≤ 1 ล้านบาท"

Agent ต้องถามจนได้ข้อมูลครบทุก Field ในรูปแบบ JSON

## 7. Competencies ที่จะวัดในการสัมภาษณ์

- Frontend Design** — Design Token, Responsive, A11y
- Agentic Pipeline Engineering** — Agent, Prompt, Router Logic
- Architecture & Code Quality** — Layering, ADR, CI/CD, ความเป็น Lean ของตัว Code



## 8. ผลลัพธ์ที่ต้องส่ง (Deliverables)

#	รายการ	เกณฑ์ผ่าน
1	Git Repo	Branch หลัก main
2	Diagram	ภาพรวมโครงสร้าง Diagram
3	ตัวอย่าง JSON Summary	ชื่อไฟล์ sample_summary.json
4	ADR-01.md	อธิบาย Agent Routing
5	ระบุ IDE ที่ใช้	VSCode, PyCharm, Xcode, Cursor, Windsurf etc.