SECTION 1: BASIC DETAILS

Name: Jarvis

AI Agent Title / Use Case:

Voice-enabled AI Assistant for conversational task automation, content retrieval, and intelligent dialogue management across various domains.

SECTION 2: PROBLEM FRAMING

1.1. What problem does your AI Agent solve?

Jarvis enables natural voice-based interaction for retrieving, summarizing, and answering complex user queriesespecially across dynamic sources like web content or document repositorieswithout requiring a screen or keyboard.

1.2. Why is this agent useful?

It provides hands-free, real-time conversational assistance with long-form memory, contextual understanding, and task continuity. Ideal for accessibility, multitasking, and smart interface applications.

1.3. Who is the target user?

- Knowledge workers needing hands-free research
- Students using voice to learn, revise, or generate notes
- Elderly or visually impaired users
- Anyone who prefers interacting with an intelligent assistant via voice

1.4. What not to include?

- No facial recognition or computer vision features
- No complex physical automation (e.g., IoT device control)
- No deeply personalized behavioral tracking (initial version avoids this for privacy reasons)

SECTION 3: 4-LAYER PROMPT DESIGN

3.1 INPUT UNDERSTANDING

Prompt: "Convert this spoken input into a clean, structured query and identify intent."

Purpose: Parses voice input into clear, actionable intent with minimal hallucination.

Example Input + Output:

Input: What happened in the Indo-Bangladesh fencing case?

Output: Query mapped to: Summarize Supreme Court Committee visit regarding Indo-Bangladesh border fencing

3.2 STATE TRACKER

Prompt: Maintain the users previous context, tasks, and feedback throughout the session. Summarize session state in every interaction.

Purpose: Simulates memory of user preferences and intent continuity.

Memory Method: Maintained in system messages or persistent context state (e.g., via LangChain memory or session variables)

3.3 TASK PLANNER

Prompt: Break the input into modular tasks (e.g., search, summarize, respond) and plan the most efficient response path.

Planning Strategy:

- Chaining used for multi-turn interactions
- Branching for follow-up handling (conditional memory recall)
- Task breakdown: Parse intent Choose tool (LLM, search, doc lookup) Summarize/Answer Confirm Respond

3.4 OUTPUT GENERATOR

Prompt: Respond conversationally with clear structure. Use markdown formatting for display when applicable. Ensure response can be spoken naturally via TTS.

Behavior:

- Voice-friendly phrasing
- Summary-first, detail-second structure
- Context-aware closing ("Would you like more details?", "Should I explain the background too?")

SECTION 4: CHATGPT EXPLORATION LOG

Attempt | Prompt Variant | What Happened | What You Changed | Why You Changed It

1 | Summarize this news article | Gave full page scrape | Added cleaner summary constraint | Needed shorter output

2 | Answer this follow-up question | Lost context | Added system memory recall prompt | Needed continuity

3 | Explain like Im 12 | Used technical terms | Tuned tone to conversational | Improve accessibility

4 | Voice summarize this PDF | Gave text output | Integrated TTS playback module | Support hands-free use

5 | What happened next? | No thread memory | Included user session chain ID | For multi-turn chat thread

SECTION 5: OUTPUT TESTS

Test 1: Normal input

Input: What did the Supreme Court committee do about the Indo-Bangladesh border fencing?

Output: The committee visited outposts in Meghalaya to inspect fencing progress. Led by A.K. Mangotra, it emphasized tighter border management...

Test 2: Vague input

Input: Tell me about the fence issue

Output: Do you mean the Indo-Bangladesh border fencing case? Would you like a recent summary or background?

Test 3: Invalid input

Input: or mumble

Output: Sorry, I couldnt catch that. Can you repeat or rephrase?

SECTION 6: REFLECTION

6.1. Hardest part: Managing memory for contextually rich voice-based conversations.

6.2. Most enjoyable: Designing the memory-aware agent that feels like a human assistant.

6.3. Improvements: Add tone detection, better noise resilience, and summarization under time limits.

6.4. Learned: Role-specific prompts and memory simulation are key to natural dialogue.

6.5. Challenges: Follow-up breakdowns, solved with session chunking and memory tagging.

SECTION 7: HACK VALUE

- Integrated voice input/output with Whisper + ElevenLabs
- Chained search + summarization + explanation dynamically
- Simulated conversational memory using state containers
- Added semantic clarification for vague prompts
- Optimized for voice tone, pacing, and flow for TTS delivery
- Agent-ready for voice-first interfaces (phones, smart assistants, etc.)