

Project Summary:

Jarvis is an AI assistant. It helps you do repetitive tasks to remove all the fluff in your day and get you straight to work. The longer you use Jarvis, the more efficient it gets because it learns from what you do. For example, if you go on your PC every single day and open up Spotify that's the first thing you do, Jarvis learns from that and opens up Spotify and gets you going.

Problem Statement:

Most users inefficiently juggle apps and information across their laptop/PC and they spend too long doing physical tasks. Jarvis Allows you to do these things by itself, first by automating simple tasks, and later on by becoming a tangible holographic system that provides a real-time visualization and assistance. The goal is to make technology feel natural, interactive, and seamlessly integrated into your everyday life.

Use Case:

Anyone that uses a laptop/PC will be able to use Jarvis. It'll be used by those who can comprehend higher-level tasks, so that they are able to understand what Jarvis is doing. Later on, it will be able to be used by everyone worldwide because it'll become simpler and easier to understand and use. People will use my product because it allows you to take out every repetitive task, and the more you use it, the more efficient your life gets.

Two specific goals for Jarvis are:

1. That it learns from you
2. That it takes what it learns from you and does it for you if it's able to do it.

Key Features:

- Wake Word Activation: “Jarvis” activates the system hands-free.
- Proactive Behavior: Suggests reminders, opens study materials, or plays focus music automatically.
- Voice Commands: Execute tasks like sending messages, managing tasks, or checking markets.
- Mood-Based Interaction: Uses tone detection to select playlists or adjust responses.
- Learning Capability: Adapts to user preferences over time.
- Cross-Platform Control: Syncs across desktop and mobile environments.

Tech Stack:

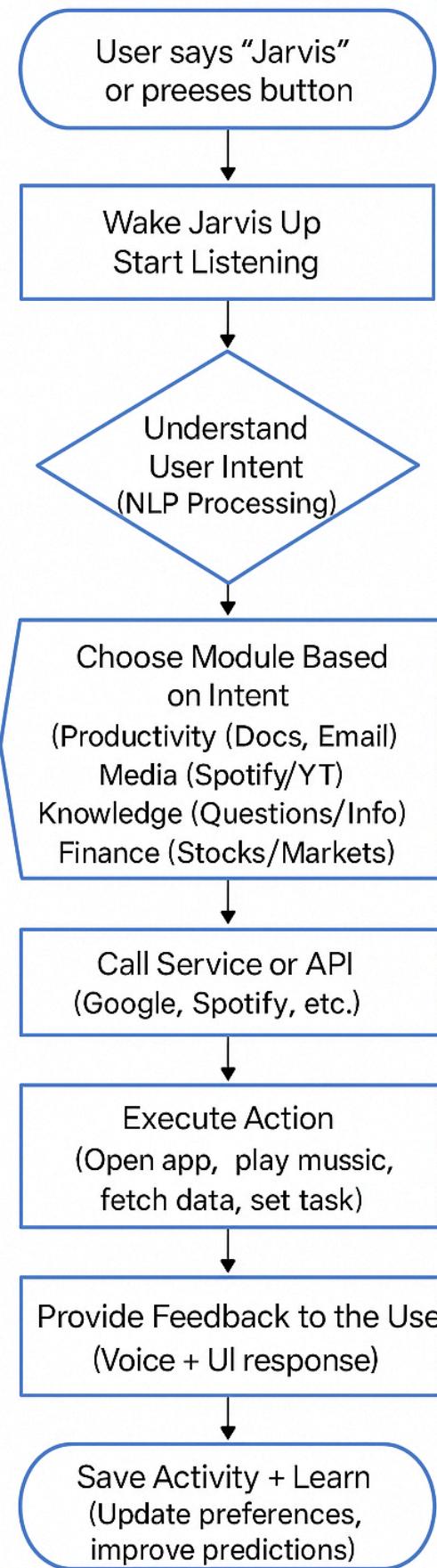
- Frontend: React / Next.js with Tailwind CSS
- Backend: Python (Flask or FastAPI)
- Database: PostgreSQL

- Integration APIs: OpenAI/Gemini, Spotify, Google APIs, TradingView, etc.
- Voice System: SpeechRecognition + pyttsx3 or custom neural TTS
- Additional Components: XML/XSL for customizable settings, and a proactive reminder engine using user-specific behavioral data.

Algorithm:

1. User activates Jarvis
 - a. The user says the wake word “Jarvis” or taps the Jarvis button.
2. Jarvis starts listening
 - a. Jarvis turns on the microphone and records what the user says.
3. Convert speech to text
 - a. The voice recording is sent to the speech recognition system, which turns it into text.
4. Understand the command
 - a. Jarvis analyzes the text to figure out what the user wants (ex: play music, open a doc, check a stock, set a reminder).
5. Choose the correct module
 - a. Based on the command, Jarvis picks the right feature:
 - i. Productivity (docs, email, reminders, calendar)
 - ii. Media (Spotify, YouTube, podcasts)
 - iii. Knowledge (questions, definitions, info)
 - iv. Finance (stock prices, market summary)
 - v. System tasks (open apps, timers, settings)
6. Call the needed service or API
 - a. Jarvis connects to Google, Spotify, finance APIs, or local apps to do the task.
7. Execute the action
 - a. The requested action is performed (for example: create a new Google Doc, play a playlist, get Tesla’s stock price, or set a reminder).
8. Give feedback to the user
 - a. Jarvis speaks back the result and (if needed) shows it on the UI, confirming what was done.
9. Save activity and learn
 - a. Jarvis logs the command and updates user preferences so future suggestions and responses are more personalized.
10. Return to idle mode
 - a. Jarvis stops active listening, goes back to low-power background mode, and waits for the next “Jarvis” command.

Flowchart:



Timeline:

The project will be split into two sections, section one is all the software anything that needs to be developed for example all of JARVIS will be built except the hardware, section two is when the hologram, a physical assistant will be built (think alexa combined with a hologram). Section one is expected to be completed by February and the physical product to be completed by June.

Risk Mitigation:

- Risks:
 - Jarvis may mishear or misunderstand voice commands, especially in noisy environments, leading to wrong actions or user frustration.
 - Users may worry about Jarvis constantly listening, storing voice data, or accessing sensitive information.
 - With many APIs (Google, Spotify, TradingView, Gmail), a single API outage or authentication issue can break parts of the system.
 - If Jarvis runs too many processes (voice, NLP, emotions, APIs) at once, it may respond slowly, making the assistant feel unusable.
- Mitigations
 - Use a high-accuracy speech recognition model (Whisper / Picovoice)
 - Keep wake-word detection local, allow users to toggle between “always-on” and “push-to-talk”
 - Use modular API handlers, include fallback behaviors, and log integration failures so Jarvis continues working even if one service goes down. Cache key data locally to prevent full feature downtime.
 - Optimize by offloading heavy processing to background threads, caching frequent responses, and running real-time components locally. Test performance regularly on low-end devices.

Eval Criteria:

- Jarvis should correctly understand and execute at least 90% of user voice commands in normal room conditions.
- The assistant should respond to commands within 1–2 seconds for local tasks (music, reminders, opening apps) and under 4 seconds for tasks involving external APIs (Google

Docs, TradingView, Spotify). A fast response indicates a smooth and natural user experience.

- Google Docs, Gmail, Spotify, TradingView, and other APIs should work consistently with less than 5% error or downtime during regular testing. If an integration fails, Jarvis should gracefully provide a fallback or error message.

Future Considerations:

The long-term plan is to expand beyond the software version and move toward a real-world holographic device—a physical assistant capable of projecting interactive 3D visuals, schematics, and data panels. This will bring Jarvis from the screen into the user's environment, creating a more immersive and hands-free productivity experience.