🧩 Frontend (User-Facing Layer)

Framework: Next.js 15 (React + TypeScript) - Real-time UI for dictation, editing, and chat-like interaction. - Extension-ready for Chrome/Edge (MV3) and as a standalone web dashboard.

Realtime Communication: Socket.IO / WebRTC - Streams voice and AI responses with low latency.

Styling: TailwindCSS + Shadcn UI - Provides clean, modern, and responsive design.

Speech Capture: Web Speech API (fallback) - Enables quick speech-to-text in-browser when Whisper isn't available.

Build Targets: - Chrome/Edge Extension - Progressive Web App (PWA)

Backend (AI & Logic Layer)

Framework: FastAPI (Python 3.12) - Async architecture for concurrent AI calls (LLM, Whisper, TTS).

Speech-to-Text (STT): Whisper.cpp (local) / OpenAI Whisper API - Converts user voice input to text.

Language Model (LLM): DeepSeek 1.3B / Mistral 7B / IBM Granite 8B - Handles command understanding, rewriting, grammar correction, summarization, and dialogue.

Text-to-Speech (TTS): OpenAI TTS / Edge-TTS / ElevenLabs - Converts AI or system responses to realistic spoken voice.

Memory Store: Valkey (Redis) - Maintains live session memory, last edits, and user context.

Database: PostgreSQL (SQLAlchemy / Prisma) - Stores history, configurations, and user preferences.

Containerization: Docker + Docker Compose + Traefik - Microservice orchestration for frontend, backend, and Redis.

Infrastructure & Deployment

Frontend Hosting: Vercel / Render **Backend Hosting:** AWS Fargate / GCP Cloud Run / Modal **GPU Compute:** Lambda Labs / RunPod / AWS EC2 GPU Instances **Monitoring:** Grafana + Prometheus / UptimeRobot **CI/CD:** GitHub Actions → Docker Build → Automated Deployment
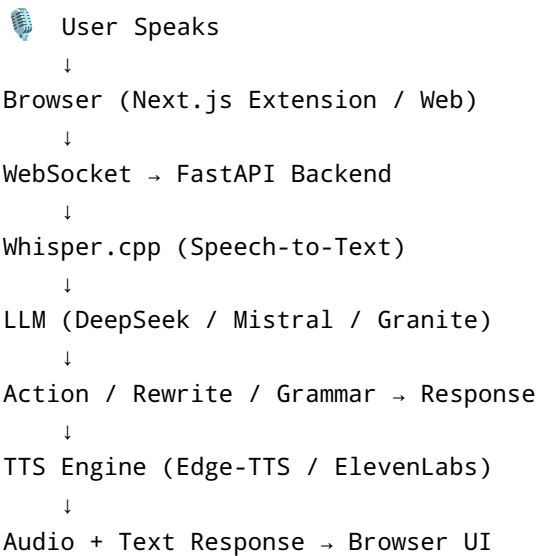

🧩 Phase 2 Integrations

Discord Bot: discord.py / discord.js (voice + text synchronization) - Joins voice channels, transcribes speech, and responds naturally.

Desktop Integration: Electron App - Brings Jarvis Dictator to MS Word, VS Code, and Notion with system-wide shortcuts.

RAG Support: LangChain / LlamaIndex - Provides context-aware writing using personal style and document memory.

🎙️ Architecture Flow

```
🎙️  User Speaks
    ↓
Browser (Next.js Extension / Web)
    ↓
WebSocket → FastAPI Backend
    ↓
Whisper.cpp (Speech-to-Text)
    ↓
LLM (DeepSeek / Mistral / Granite)
    ↓
Action / Rewrite / Grammar → Response
    ↓
TTS Engine (Edge-TTS / ElevenLabs)
    ↓
Audio + Text Response → Browser UI
```

Summary:
The Jarvis Dictator stack combines the best of modern AI infrastructure and real-time web frameworks:
React + FastAPI + Whisper + LLM + TTS + Redis = an intelligent, conversational, voice-driven writing companion that works anywhere — browser, desktop, or Discord.