TECH STACK	SUPPORTING REASON
FLUTTER	I used **Flutter** to ensure a **unified codebase** for both Android and iOS platforms, reducing development and maintenance effort. Its **high performance and native-like UI** made it ideal for building a responsive mobile interface for site engineers, also sInce the app is not going to be too complex.
REACT	I used **React** to build a **dynamic and responsive dashboard** for architects and managers on the web. Its **component-based architecture** and rich ecosystem enabled rapid development, scalability, and seamless integration with backend APIs.
WHISPER	I used **Whisper** for its **state-of-the-art transcription accuracy**, especially in noisy, real-world environments like construction sites. Its **open-source nature and multi-language support** made it cost-effective, customizable, and ideal for integrating into my AI pipeline.
GPT-4o-Mini	I used **GPT-4o mini** for its **balanced trade-off between speed, cost, and reasoning ability**, making it ideal for real-time, on-device or API-based AI tasks. Its **strong performance in text understanding and summarization** ensured high-quality extraction of insights from transcribed audio. We can also implerment safeguards when Whisper fails.
AWS SQQ	I used **AWS SQS** to enable **reliable, decoupled communication** between services like audio upload, transcription, and processing. [illegible]
Prometheus + Grafana	I used **Prometheus** and **Grafana** for **real-time monitoring and visualization** of system performance, ensuring uptime and observability.
PostgreSQL	**PostgreSQL** was chosen for its **robust relational capabilities,** making it ideal for storing structured reports, user data, and metadata with ACID compliance.
REST	I used **REST APIs** to ensure **clear, stateless communication** between frontend and backend services. Their **widespread adoption and simplicity** made integration with mobile, web, and third-party systems seamless and scalable.