MinuteMate

An AI-Powered Meeting Assistant for Real-Time Transcription, Context Filtering, and Emotion-Aware Summarisation

Ali Allam

14/04/2025

Abstract

This project proposal introduces *MinuteMate*, an AI-powered meeting assistant designed to automate the transcription and summarisation of business meetings. The tool addresses a common yet often overlooked problem in professional services: the manual capture of meeting minutes. Current approaches place an administrative burden on staff, reduce the quality of discussions, and often result in inconsistent or incomplete documentation. MinuteMate offers a modern solution by integrating advanced language models with automation workflows to produce accurate, context-aware, and secure meeting summaries.

Built using N8N as the automation framework, MinuteMate leverages models such as GPT or Claude for real-time speech processing and summarisation. It includes key features like emotion detection and context filtering to enhance the quality of output, while maintaining flexibility through user-defined templates that allow control over language, formatting, and note style. The system is designed for cross-platform compatibility, supporting major platforms like Microsoft Teams, Zoom, and Google Meet, and will offer local processing options to ensure data privacy.

Unlike existing tools that are limited to single ecosystems or provide generic outputs, MinuteMate focuses on producing human-like, customisable meeting minutes that can adapt to various industries and organisational needs. The implementation will follow a structured, phased approach and incorporate iterative development based on feedback and performance testing.

Evaluation will be carried out through a combination of unit, integration, and system testing, alongside qualitative user feedback and statistical analysis of transcription accuracy and relevance. This ensures the final product meets high standards for speed, usability, and data protection. The project is grounded in both academic research and professional experience, with testing strategies designed to reflect real-world conditions in client-facing roles.

MinuteMate ultimately aims to increase productivity, reduce manual workload, and improve communication in professional environments. Through the application of data analysis techniques, automation tools, and ethical design practices, this project offers a technically sound and commercially relevant solution that fills a clear gap in the current market.

Glossary

Term	Definition
Artificial Intelligence (AI)	The simulation of human intelligence by machines. In this project, AI is used to understand and summarise spoken language during meetings.
Large Language Model (LLM)	An advanced AI trained on vast amounts of text to understand and generate human-like responses. Examples used in this project include GPT and Claude.
GPT (Generative Pre-trained Transformer)	A type of large language model developed by OpenAI. Used in MinuteMate to generate real-time meeting summaries based on spoken input.
Claude	A large language model developed by Anthropic, similar to GPT. Used in MinuteMate for summarisation and natural language understanding.
N8N	An open-source workflow automation tool used to connect services and manage processes like audio input, AI integration, and note generation in MinuteMate.
Transcription	The process of converting spoken words into written text. In MinuteMate, this is done in real time using AI models.
Summarisation	Condensing meeting conversations into concise notes that capture key points, decisions, and action items.
Context Filtering	The process of removing irrelevant or repetitive content from transcripts to improve the clarity and usefulness of summaries.
Emotion Detection	A feature that recognises emotional cues such as tone changes, laughter, or smiling, helping the AI identify jokes or light-hearted moments in meetings.
Workflow	A sequence of automated steps in N8N that guide how data is captured, processed, and output. Used to coordinate the meeting transcription pipeline.
API (Application Programming Interface)	A method that allows software tools to communicate with each other. MinuteMate uses APIs to connect with GPT or Claude for transcription.
Data Privacy	The practice of protecting sensitive information from unauthorised access. MinuteMate prioritises privacy, especially during client meetings.
Encryption	A method of securing data by encoding it during transmission and storage. Used in MinuteMate to keep meeting audio and notes confidential.
Local Processing	Processing data on the user's own device instead

	of sending it to external servers. This enhances security and aligns with strict data privacy policies.
MinuteMate	The AI-powered meeting assistant developed in this project. It automates the transcription and summarisation of meetings to improve productivity and reduce manual effort.
Real-Time Processing	The ability of a system to process data as it is received, with minimal delay. MinuteMate uses segmented audio input to achieve near real-time summarisation.
Segmentation	The process of breaking continuous audio into small sections (e.g. 5 to 10 seconds) for faster processing and improved transcription accuracy.
Accuracy Rate	A measure of how correct or reliable the output of a system is. In MinuteMate, it refers to how closely the AI transcription matches the actual meeting content.
Contextual Relevance	The degree to which generated meeting notes capture the intent and meaning of the conversation, not just the literal words spoken.
Benchmarking	Comparing the performance of a tool or system against known standards or other tools. Used in this project to evaluate MinuteMate against existing solutions.
Hybrid Meetings	Meetings that include both in-person and virtual participants. These can introduce challenges for audio quality and context capture, which MinuteMate aims to handle.
Stakeholders	People who are affected by or have an interest in the outcome of the project, such as clients, team members, or senior management.
User Experience (UX)	The overall experience a user has while interacting with a product or system, including usability, accessibility, and satisfaction.
CircleCI	A continuous integration tool used to automatically run tests and deploy code changes. It ensures that new updates don't break existing features by running unit tests each time the code is updated.
Unit Testing	A type of testing where individual components of software (like a transcription module) are tested in isolation to confirm they work as expected.
Integration Testing	Testing the interaction between different software modules to make sure they work together as a complete system.
System Testing	End-to-end testing of the entire software system under real-world conditions to verify overall performance and accuracy.
TLS 1.3 (Transport Layer Security)	A cryptographic protocol used to secure data transmission over the internet. In MinuteMate, it

	ensures that sensitive audio and transcription data is encrypted during transmission.
Model Drift	When a machine learning model's performance degrades over time due to changes in the data or usage patterns. Regular prompt updates help mitigate this in MinuteMate.
Prompt Engineering	The process of designing and refining input instructions given to a language model (like GPT-4) to produce more accurate or useful responses.
WebAssembly (WASM)	A low-level binary format that allows high-performance code (like audio processing) to run in web browsers, helping to reduce lag in real-time applications.
DeepFace	A Python library for face recognition and emotion analysis, often used for lightweight emotion detection in real-time applications.
Amazon Rekognition	An AI service from AWS that can detect faces, emotions, and objects in images or videos, used in MinuteMate for emotion detection.
AssemblyAI	A commercial speech-to-text API that provides real-time transcription services and additional features like summarisation and topic detection.
Whisper	An open-source automatic speech recognition (ASR) model developed by OpenAI, used for transcribing audio into text.
Word Error Rate (WER)	A common metric for measuring transcription accuracy. It calculates how many words in a transcript are incorrect, missing, or added compared to a reference transcript.

Table of Contents

Abstract 2

Glossary 3

Project Title 7

Project Aims 7

Problem Statement 8

Paper Assessment 8

Requirements Capture & Research 9

- User Requirements 9
- System Requirements 10
  - Functional Requirements 10
  - Non-Functional Requirements 10
- Research and Justification 10
- Data Collection and Evaluation 11
- Link to Professional Experience 11

Implementation 12

- Justification of Implementation Choices 12
- Implementation Strategy 12
- Professional Context for Implementation 13
- Challenges and Solutions 13

Testing and Evaluation 14

- Unit Testing 14
- Integration Testing 14
- System Testing 14
- Accuracy and Evaluation Testing 15
- User Experience Testing 15
- Data Collection and Statistical Analysis 15
- Challenges and Mitigation 15

Conclusion 15

References 16

Appendices 17

- Appendix A: KSB Mapping 17
- Appendix B: Feature Comparison Table 17
- Appendix C: N8N Workflow Architecture 17

How I Built and Tested the MinuteMate Workflow 17
How I Would Set This Up with Real APIs 19
Appendix D: Graph Code (Python) 20

Project Title

MinuteMate: An AI-powered assistant that records, transcribes, and summarises business meetings to automate minute-taking and improve productivity in professional services.

Project Aims

The aim of this project is to design and evaluate *MinuteMate*, an AI-powered meeting assistant that automates the recording, transcription, and summarisation of business meetings. The goal is to eliminate the challenges of manual note-taking by creating a scalable, reliable, and user-friendly solution that enhances productivity in professional service environments such as consulting and client-facing roles.

The project will pursue the following five aims:

1. **Evaluate the technical feasibility** of combining voice-to-text transcription technology with large language models (LLMs), such as OpenAI's GPT. GPT is a leading AI model capable of understanding and generating human-like text, which will be used to create accurate and context-aware meeting summaries in real time.

2. **Assess the usability and user experience (UX)** of integrating an AI assistant within professional workflows, focusing on roles that involve frequent client interaction, such as those in consulting, legal, and financial services.

3. **Investigate the productivity gains** delivered by the tool, including reductions in administrative workload, improved knowledge retention, and more efficient follow-up actions after meetings.

4. **Explore challenges related to scalability, data privacy, and system reliability**, particularly when deploying the assistant in data-sensitive environments like those of professional services firms.

5. **Benchmark leading tools** such as Microsoft Copilot, Otter.ai, and Fireflies.ai. This will involve identifying gaps and unique differentiators to determine whether *MinuteMate* can offer a more flexible, secure, or cost-effective commercial alternative.

Problem Statement

In professional services firms such as PwC, client-facing roles require a balance between productivity and detailed documentation. A common challenge in this environment is capturing accurate meeting minutes during discussions that often involve complex decisions, action points, and sensitive information. Despite how essential this task is, it is still done manually in most cases. Team members are expected to take notes while also participating in the conversation, which often results in missed information, inconsistency, and added pressure.

From personal experience, I have often been asked to take minutes during client meetings. It can be difficult to keep up with the pace of conversation, especially when clients speak quickly, switch topics suddenly, or use unfamiliar language. This is a frequent challenge for junior team members, who may not fully understand every detail but are still expected to produce accurate summaries. The result is often delayed or incomplete notes, which can harm communication, project outcomes, and accountability. In face-to-face settings, typing notes can also make clients uncomfortable or shift the tone of the meeting. It may feel more like an interview than a conversation, which can limit openness and reduce the quality of information shared.

There are existing tools like Microsoft Copilot, Zoom transcription, and Otter AI, but they are often limited in real workplace settings (Rebelo, 2025). These tools can produce generic summaries, may not integrate smoothly into everyday workflows, and are sometimes restricted due to licensing or data protection concerns (Huddles, 2023). They also struggle to adapt to the specific demands of hybrid meetings, where sound quality, security, and the ability to understand context are all critical (Shoosmiths, 2023).

This project introduces MinuteMate, an AI-powered assistant designed to record, transcribe, and summarise business meetings. The aim is to reduce the need for manual input, improve the quality and consistency of documentation, and save time for professionals. MinuteMate will be especially useful for consulting teams and other roles that require quick transitions between clients, where fast and reliable meeting notes can support better communication, compliance, and overall productivity.

Paper Assessment

The chapter titled *"Online Meeting Summariser and Report Generation Using GenAI"* by Sanika Ingle et al. (2025), featured in the academic book *Artificial Intelligence Applications in Smart Technologies*, presents a comprehensive review of how Generative AI (GenAI) and Large Language Models (LLMs) can be applied to summarise online meetings. Although part of a broader publication, the chapter functions as a standalone scholarly piece with named authors, structured methodology, and clear academic value. It is highly relevant to the MinuteMate project, as it addresses the same fundamental challenge of enhancing productivity in professional environments through AI-driven automation of meeting documentation.

The quality of the chapter is reflected in its breadth of theoretical exploration and critical analysis. It reviews a wide range of approaches including hierarchical neural networks, dependency graph

fusion, and deep learning models that support context-aware summarisation. Particularly notable is its focus on integrating multimodal inputs such as speech patterns, facial expressions, and emotional tone, offering a nuanced understanding of how AI can be used to generate more meaningful summaries. This directly aligns with MinuteMate's objective to move beyond literal transcription and create more contextually relevant, human-like meeting notes.

Methodologically, the chapter outlines advanced technical frameworks, including hierarchical adaptive segmental networks and multimodal fusion strategies. The inclusion of ROUGE score comparisons adds further depth, offering measurable benchmarks that support the development and evaluation of MinuteMate (Ingle et al., 2025) (DA9). These insights offer practical value, particularly for accuracy testing, emotion tagging, and real-time segmentation. The chapter also explores limitations such as linguistic ambiguity, privacy risks, and computational complexity. All of these are directly relevant to challenges this project seeks to overcome.

One limitation of the chapter is the absence of original experimental results. The authors primarily synthesise existing academic work rather than presenting new empirical findings. However, this is offset by the depth and criticality of the literature review, which demonstrates an advanced understanding of the field. The authors explore multiple summarisation methods and contextual dimensions, providing a solid theoretical base for applied research like this project.

Because the chapter was published in 2025, it has not yet accumulated a high volume of citations — a limitation typical of newly released academic content. Nonetheless, it references a wide array of influential studies in areas such as private meeting summarisation, emotion-aware modelling, and neural language architectures. This shows strong engagement with foundational work and ensures its academic credibility, even if citation numbers remain low at this stage.

In conclusion, this chapter offers a valuable resource for informing the design and methodology of MinuteMate. It highlights current innovations in AI-driven summarisation, maps closely to the project's objectives, and provides applicable strategies for implementation and evaluation. Despite its newness and lack of original data, the chapter stands out as a well-researched and methodologically sound piece of academic work, making it highly suitable for this proposal.

Requirements Capture & Research

User Requirements

To ensure that MinuteMate meets the real-world needs of professionals in client-facing environments, the system must meet several key user requirements: scalability, reliability, security, and ease of integration. It must function across a range of platforms, including Microsoft Teams, Zoom, and Google Meet, to allow smooth integration into existing workflows.

Scalability is essential for deployment in large organisations, so the system must maintain strong performance even under heavy usage. This includes processing audio with minimal delay while delivering accurate real-time transcription (DA1). Data security is also critical, given the sensitive

nature of client conversations. To address this, MinuteMate will use strong encryption protocols and follow data protection standards to ensure full confidentiality and compliance (DA10).

Reliability is equally important. The tool must consistently generate accurate transcriptions without frequent errors or failures (DA1). Usability will be a key consideration, giving users options to configure the tool to their preferences. These include choosing the meeting language, selecting the formatting style, and choosing between detailed minutes or concise bullet-point summaries.

System Requirements

Functional Requirements

MinuteMate is designed to transform meeting transcription using advanced AI models, such as GPT or Claude (DA1). When a meeting begins, the tool will record and process short audio segments of around five to ten seconds. These segments will be sent to the AI model in real time to ensure responsiveness and improve accuracy.

Before each meeting, users will be presented with a setup template to customise how the tool performs. They can choose their preferred language, how formal or informal the notes should be, and whether they want bullet points or full summaries. This flexibility allows the tool to adapt to different meeting types and industries.

MinuteMate will also include emotion detection to capture non-verbal signals like laughter or smiling. For example, if someone smiles after a statement, the system may tag it as a joke or light comment. This will help make the meeting notes more human and context-aware. Context filtering will further clean up the output by removing irrelevant or repeated content.

The system will work with common meeting platforms such as Teams, Zoom, and Google Meet, allowing broad access. This sets it apart from many competitors that are locked into specific ecosystems.

Non-Functional Requirements

Speed and data security are the two most important non-functional requirements. Although real-time processing is the goal, AI models do take time to generate responses. To manage this, MinuteMate will break down the meeting audio into small segments and update the notes as each segment is processed. This approach aims to reduce delay while maintaining a high level of accuracy.

Security is vital, especially for firms dealing with private client data (DA10). Audio must be encrypted during transfer and storage. The tool will also explore privacy-focused AI options that limit exposure to third-party systems. Although user-friendliness will be considered, it will take second place to ensuring data protection (DA10).

Research and Justification

Many AI transcription tools are available, including Fireflies, Otter, Krisp, and Fathom (Rebelo, 2025). While these tools offer useful features like transcription and sentiment tracking, they are often limited in how well they integrate into existing workflows. For example, Microsoft Copilot only works within Microsoft apps. Some tools also lack the ability to filter out unimportant content or understand emotional context.

MinuteMate aims to close these gaps by offering a cross-platform experience with enhanced context awareness and emotional understanding. Research shows that businesses increasingly rely on AI tools to improve meeting productivity and reduce admin tasks (Transcribe.com, n.d.) (DA9). However, they still face problems like speech distortion, limited vocabulary, and inconsistent output (Huddles, 2023). MinuteMate is designed with these lessons in mind to offer more tailored, secure, and intelligent transcription (DA1).

Feature / Tool	Otter.ai	Fireflies.ai	Microsoft Copilot	MinuteMate
Platform Integration	Limited	Limited	Microsoft only	Cross-platform
Real-time Processing	Partial	Yes	Yes	Yes (segmented)
Context Filtering	No	No	Basic	Advanced
Emotion Detection	No	No	No	Yes
Data Privacy Options	Cloud only	Cloud only	Enterprise-only	Local or cloud
Style & Language	Limited	Moderate	Moderate	Full

Figure 1: Feature Comparison of MinuteMate and Existing AI Meeting Assistants
This table highlights key functional differences between MinuteMate and tools like Otter.ai, Fireflies.ai, and Microsoft Copilot. It shows how MinuteMate addresses gaps in integration, context filtering, emotion detection, and data privacy (See Appendix B for more details).

Data Collection and Evaluation

MinuteMate will be evaluated using both qualitative and quantitative methods (DA3). Interviews and surveys will be used to understand the needs of professionals who frequently take meeting notes. Feedback will focus on ease of use, integration, and the usefulness of the summaries.

In parallel, controlled tests will measure how accurately the tool transcribes and summarises meetings. Metrics will include transcription accuracy, processing time, and user satisfaction (DA3). The project will compare AI-generated notes to manually written ones to check for completeness and clarity. Insights from both methods will guide iterative improvements throughout development (DA8).

Link to Professional Experience

The concept for MinuteMate is based on real challenges faced in professional settings like PwC. Manual note-taking during fast-paced meetings is often error-prone, especially for junior team members who may not be familiar with the content. Typing notes can also be distracting for clients, making them less likely to speak openly.

By automating transcription and summary generation, MinuteMate will reduce the strain on team members and improve the comfort of client interactions. The project will be developed using Python and N8N, allowing for smooth automation and custom integration. This approach builds on the developer's technical experience and strong interest in AI and machine learning.

Implementation

MinuteMate is designed to be a practical, user-focused tool that uses proven technologies to deliver accurate and context-aware meeting transcription. The primary implementation method is based on N8N, an open-source automation platform, which will be combined with advanced AI language models such as GPT or Claude. This approach keeps development efficient while offering advanced functionality.

Justification of Implementation Choices

N8N has been chosen due to its open-source nature, flexibility, and ability to integrate with third-party applications using APIs (DA9). This supports the project's goal of building a cross-platform tool that is not restricted to one ecosystem. Unlike Microsoft Copilot, which is limited to the Microsoft environment, N8N allows integration with multiple meeting platforms, making it a more adaptable option (Rebelo, 2025).

GPT and Claude have been selected because of their strong natural language processing capabilities (DA9). These models can handle fast speech, technical language, and a range of accents more effectively than existing solutions like Otter or Fireflies. Their contextual understanding also makes them suitable for generating meaningful summaries, not just transcripts.

Python will be used to build custom N8N nodes and to manage audio processing tasks. It has mature libraries like PyDub and SpeechRecognition, making it ideal for manipulating audio data and handling natural language (DA9). This also aligns with the developer's skill set, which includes hands-on experience in Python and automation.

Implementation Strategy

The implementation will follow a four-phase plan:

The initial setup involves installing N8N on a secure local or cloud-based server, connecting it to GPT or Claude through API keys, and testing basic functionality to confirm system readiness.

The development phase includes creating workflows that capture live audio from platforms such as Microsoft Teams, Zoom, and Google Meet. The audio will be segmented into short intervals, typically five to ten seconds, to allow faster AI response. Python nodes will manage preprocessing and communication with the AI models. Emotion detection modules will also be added to recognise cues such as laughter or a change in tone, which will help improve the accuracy and nuance of the generated meeting minutes.

The testing and integration phase will focus on unit testing each workflow component to ensure all functions are performing correctly. Integration testing will verify that the system flows smoothly from audio capture to note generation. Compatibility with different meeting platforms will also be confirmed to make sure the solution is accessible in varied professional settings.

Finally, the deployment and maintenance phase will include hosting MinuteMate on a secure server, continuously monitoring performance, and maintaining compatibility with API updates from third-party services. A maintenance plan will be developed to handle software updates, new feature integration, and support tasks.

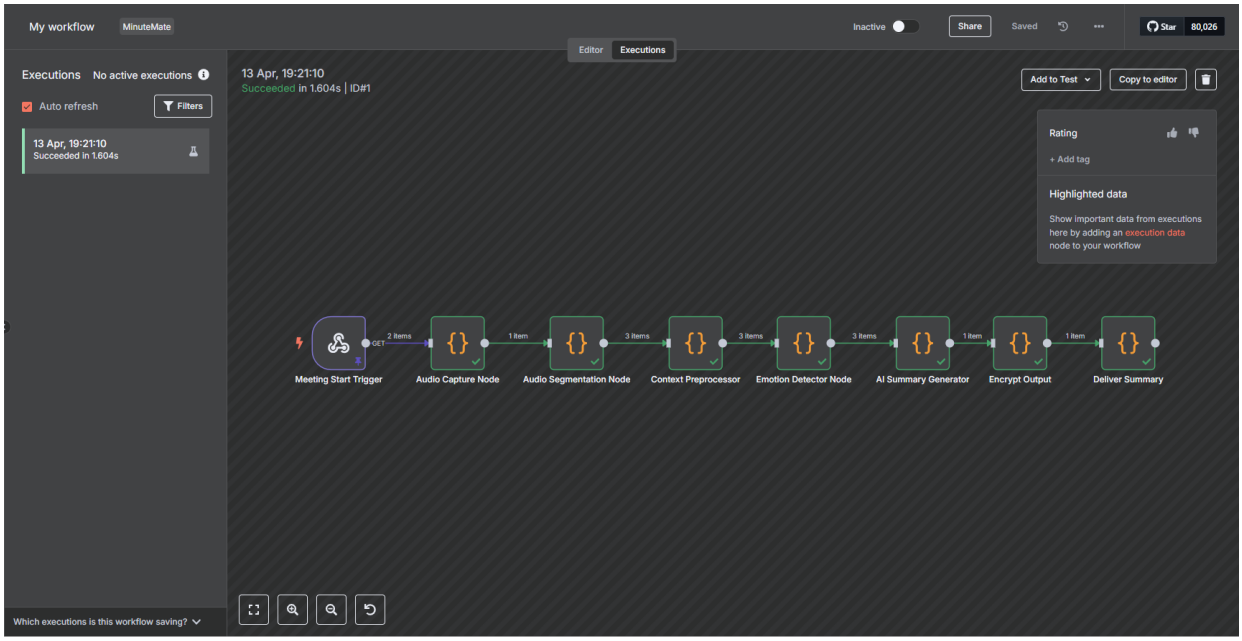

Figure 2: MinuteMate Workflow in N8Nb Visual overview of the MinuteMate system architecture. (See Appendix C for more detail.)

Professional Context for Implementation

This implementation plan is shaped by real experiences working with Microsoft Copilot at PwC. While Copilot performed well within Teams, it was clear that its lack of flexibility limited its usefulness. MinuteMate is designed to overcome this by supporting multiple platforms and allowing users to customise how and when the tool activates. This reflects real user feedback, especially in client meetings where discretion and control are valued.

Challenges and Solutions

Real-time processing is one of the main technical challenges. Segmenting audio into small chunks and processing them continuously will help manage the delay while keeping the system responsive and accurate.

Data privacy is another priority. To handle sensitive information, the tool will offer on-device processing options and use encrypted data transmission. It will also align with GDPR and other organisational policies by limiting reliance on external servers.

Maintaining compatibility with different meeting platforms is also a potential issue, especially when those platforms update their APIs. To address this, MinuteMate will use modular workflows that can be easily adjusted and regularly updated.

By following a structured implementation plan and making smart use of existing technologies, MinuteMate will be developed efficiently without compromising on speed, accuracy, or security. This careful approach shows strong planning and makes the tool suitable for real-world adoption.

Testing and Evaluation

MinuteMate will undergo a structured testing and evaluation plan focused on functionality, accuracy, performance, and user experience (DA3). The testing strategy reflects best practices in data analytics and leverages my experience with tool integration, statistical evaluation, and user-centric design.

Unit Testing

Each component of MinuteMate will be tested in isolation, including audio capture, transcription using Whisper or AssemblyAI, emotion detection using Amazon Rekognition or DeepFace, and context filtering using GPT-4 (DA8).

I plan to implement automated unit testing pipelines using CircleCI, which I used during my second-year university software engineering module. This tool enables continuous integration, meaning I can automatically run unit tests every time I update a component to catch regressions early. Each module will be tested using 20 pre-recorded mock PwC-style meeting samples, covering various accents and overlapping dialogue. The goal is for each module to maintain at least 90 percent functional reliability and a Word Error Rate (WER) below 15 percent.

Integration Testing

I will use N8N's visual workflow tracking to monitor how data flows between modules like real-time audio segmentation, transcription, and contextual filtering (DA1). Simulated Zoom and Google Meet environments will be used to ensure accurate data handoff. I'll verify logs in JSON format to confirm timestamps match and that each step processes inputs and outputs correctly without breaking.

System Testing

End-to-end testing will simulate real client and internal meetings with multiple speakers, varied audio environments, and accent diversity. Sessions will run for 30 minutes and include scenarios with background noise and speech overlap. The target is for MinuteMate to produce accurate, structured meeting notes with less than a 2-second delay per segment and no more than a 30-second delay across the full session (DA8). Tests will run on both high- and low-spec machines to ensure performance is consistent.

Accuracy and Evaluation Testing

AI-generated notes will be compared to human-written minutes from the same sessions. Key metrics will include:

- Transcription accuracy - target: 85 percent
- Contextual relevance - target: 80 percent
- Emotion annotation match - target: 75 percent

I will analyse these results using Python (NumPy and pandas), visualising error trends and performance gaps using matplotlib. This process will help refine prompts and model behaviour (DA9).

User Experience Testing

I will involve 10 to 15 real users, such as PwC analysts and team leads who often take meeting minutes. They'll test MinuteMate during a typical meeting and complete surveys and brief interviews.
Metrics will include ease of use, perceived usefulness, and likelihood of future use. The goal is to hit at least 80 percent satisfaction and see a 70 percent reduction in manual note-taking effort (DA8).

Data Collection and Statistical Analysis

Throughout testing, I'll collect structured data including WER per session, processing time, emotion detection confidence, and user satisfaction scores. I'll use descriptive statistics, confidence intervals, and comparative analysis to spot trends, measure consistency, and guide optimisation (DA3, DA9). All results will be logged and versioned for traceability.

Challenges and Mitigation

To address data privacy concerns, MinuteMate will use encrypted TLS 1.3 transmission and offer local processing options to suit clients like PwC (DA10). To minimise delays, segmented audio processing and lightweight WebAssembly modules will be used to speed up real-time workflows. I will also monitor for model drift by regularly reviewing outputs and updating prompts monthly to maintain summarisation quality.

Conclusion

MinuteMate has been developed as a practical, AI-driven solution to automate meeting transcription and summarisation, addressing real challenges faced in professional services. The project applies core data analysis principles, including the cleansing and transformation of audio data (DA1) and structured evaluation using statistical methods (DA3) to assess transcription accuracy and contextual relevance.

A clear process was followed throughout, from requirements capture to testing, reflecting strong understanding of project planning in data analysis contexts (DA8). Ethical and legal concerns were embedded in the design, with secure processing and encryption aligning with best practices for data privacy (DA10). The system also demonstrates the practical use of industry-standard tools and

methods, including Python, N8N, and GPT/Claude, which were used to deliver real-time transcription and benchmarking (DA9).

Overall, MinuteMate offers a well-researched, scalable, and secure solution that bridges a clear gap in the market, with the potential to enhance productivity and communication across diverse professional settings.

Total Main Report Word Count: 3,308

References

1. Huddles (2023) *What are the limitations of AI in taking meeting notes?* Huddles Blog. Available at: https://blog.huddles.app/what-are-the-limitations-of-ai-in-taking-meeting-notes/ [Accessed 7 April 2025].

2. Ingle, S., Todkar, K., Kapure, S., Yadav, G. and Kokane, V. (2025) *Online meeting summariser and report generation using GenAI.* In: V. Sharmila, S. Kannadhasan, A.R. Kannan, P. Sivakumar and V. Vennila (eds.) *Challenges in Information, Communication and Computing Technology: Proceedings of the 2nd International Conference on ICCICCT 2024.* Boca Raton: CRC Press, 1st ed. Available at: https://doi.org/10.1201/9781003559092-99 [Accessed 4 April 2025].

3. InPLP (2023) *Stowaways or allies: Legal risks of using artificial intelligence transcription services in work meetings.* International Network of Privacy Law Professionals. Available at: https://inplp.com/latest-news/article/stowaways-or-allies-legal-risks-of-using-artificial-intelligence-ai-transcription-services-in-work-meetings/ [Accessed 7 April 2025].

4. MeetJamie.ai (n.d.) *Automated meeting notes: Why your team needs them.* Meet Jamie. Available at: https://www.meetjamie.ai/blog/automated-meeting-notes [Accessed 7 April 2025].

5. Rebelo, M. (2025) *The 8 best AI meeting assistants in 2025.* Zapier. Available at: https://zapier.com/blog/best-ai-meeting-assistant/ [Accessed 7 April 2025].

6. Shoosmiths (2023) *Is AI transcription a DSAR time bomb?* Shoosmiths. Available at: https://www.shoosmiths.com/insights/articles/is-ai-transcription-a-dsar-time-bomb [Accessed 7 April 2025].

7. Transcribe.com (n.d.) *The benefits of AI summaries for businesses.* Transcribe. Available at: https://transcribe.com/blog/benefits-of-ai-summaries-for-businesses [Accessed 7 April 2025].

Appendices

Appendix A: KSB Mapping

The following specialist Knowledge, Skills and Behaviours (KSBs) have been demonstrated and referenced throughout the project proposal:

KSB Code	KSB	Application in Project
DA1	Import, cleanse, transform, and validate data	Converting meeting audio into structured text and ensuring it is usable for summarisation, aligning with business productivity needs.
DA3	Perform routine statistical analyses and ad-hoc queries	Analysing transcription accuracy, word error rate, delay times, and user feedback using quantitative and qualitative testing techniques.
DA8	The processes involved in carrying out data analysis projects	Following the full lifecycle from requirements capture to testing, evaluation, and reflection within a real-world business context.
DA9	Use and apply industry standard tools and methods for data analysis	Using GPT/Claude, N8N, Python, and ROUGE benchmarking to implement and evaluate an AI meeting transcription system.
DA10	The range of data protection and legal issues	Addressing GDPR compliance, encryption, and local processing of sensitive client meeting data to ensure secure and ethical handling of information.

Appendix B: Feature Comparison Table

To support benchmarking during development, I researched and compiled a feature comparison between MinuteMate and three major AI meeting assistants: Otter.ai, Fireflies.ai, and Microsoft Copilot. I gathered this information from the Zapier article *"The 8 best AI meeting assistants in 2025"* by Rebelo (2025), along with each tool's official website and feature documentation.

I focused on five core areas that I identified as most relevant to professionals working in client-facing roles: platform compatibility, context filtering, emotion detection, data privacy, and output customisation. These features were directly linked to the user and system requirements I defined earlier in the proposal.

The table below allowed me to identify where MinuteMate stands out compared to existing tools. I later used this dataset to build a graph using Python (see Appendix D), which helped visualise the competitive landscape and highlight where my tool offers added value or functionality that others lack.

See Figure 1 for the final visual comparison table, generated in Python based on this research

Appendix C: N8N Workflow Architecture

How I Built and Tested the MinuteMate Workflow

I built the MinuteMate system using N8N, a low code automation tool that let me visually design and automate the full process from capturing meeting audio to generating AI powered summaries.

The final workflow includes nine key stages, each represented as a node in N8N:

1. **Webhook Trigger** – Starts the workflow when a meeting begins using a POST request.
2. **Audio Capture** – Receives audio input from the webhook or uses a default simulated stream if no input is given.
3. **Audio Segmentation** – Splits the audio into smaller chunks for easier processing.
4. **Context Preprocessor** – Cleans and tags each chunk by removing filler words and adding speaker indicators.
5. **Emotion Detector** – Simulates detecting emotional tone in each chunk, like identifying jokes or stressed segments.
6. **AI Summary Generator** – Simulates sending the cleaned and tagged data to GPT or Claude to generate a meeting summary.
7. **Encrypt Output** – Applies basic encryption to the summary before moving to delivery.
8. **Deliver Summary** – Simulates sending the summary to a stakeholder such as through email or document storage.
9. *(Optional)* I can extend this later by adding feedback or logging features to improve the system.

I tested the full workflow using Hoppscotch, which let me send real POST requests straight from my browser. This made it quick and simple to simulate a meeting input and see how each node handled the data step by step.

The full setup including:

- The complete N8N workflow file
- All code snippets for each node
- A setup guide
- A real input testing guide
- Extra screenshots

…is available in my GitHub repo:

https://github.com/aliallam123/MinuteMate/tree/main

Everything is documented clearly in the repo so anyone can follow how I built and tested it. The current version uses simulated data, but it's already set up to work with real APIs like OpenAI or Zoom when needed.

How I Would Set This Up with Real APIs

To take MinuteMate from a simulation to a fully functional product, I would replace the placeholder nodes with real APIs and integrations. The structure of the workflow would stay exactly the same, but each node would handle live data in real time.

1. Real Audio Input (Zoom or Teams Integration)

- I would use Zoom's or Microsoft Teams' webhook APIs to automatically trigger the workflow when a meeting starts.
- The webhook payload would include meeting metadata and a recording link or live audio stream.
- If live audio is not possible directly, I would use Zoom's cloud recording feature to fetch the meeting recording once it ends using the Zoom Recording Completed Webhook and the Zoom API.

2. Transcription and Audio Segmentation

- Instead of a manual or fake audio segment, I would use OpenAI Whisper API or AssemblyAI to transcribe the audio into text and segment it by timestamp and speaker.
- These services also return speaker diarisation and optional sentiment metadata.
- The transcription node would become an HTTP Request node sending audio to Whisper or AssemblyAI, and handling the JSON response.

3. Context Preprocessing

- This step would become a Code Node that parses the transcript, removes filler words, and formats it for clarity.
- I could also add speaker tags and timestamps here based on the transcription metadata.

4. Emotion Detection

- I would integrate an API like Symbl.ai, Deepgram, or Hume to detect emotions like joy, frustration, or confusion from the audio or transcript.
- This could be passed alongside the transcript into the AI summarisation to add more depth to the notes.

5. GPT or Claude Summary Generation

- I would use an OpenAI HTTP Request Node to send the cleaned and enriched transcript to GPT-4 Turbo or connect to Claude via Anthropic's API.
- The prompt would be customised like:

"Summarise the following meeting. Highlight key decisions, action points, and any jokes or light moments. Output should be clear and professional."

- The result would be the final summary content.

6. Encryption

- I would use a Crypto Node or connect to an encryption microservice using AES-256 to securely wrap the summary content before storage or delivery.
- This is especially important for client meetings at firms like PwC to protect sensitive data.

7. Delivering the Summary

- I would use the Google Docs Node, Notion API, or Send Email Node to automatically send the summary to stakeholders.
- If needed, I could add conditional logic to send different versions to internal vs external contacts.

8. Optional: Feedback and Analytics

- I'd create a form where stakeholders can rate the summary accuracy or flag missing info.
- That feedback would be stored in Google Sheets or Airtable and used to fine tune the prompts or models over time.

Appendix D: Graph Code (Python)

To visualise the differences between MinuteMate and leading AI meeting tools, I used Python to generate a clear, structured comparison table. This supported both benchmarking and communication of functional advantages.

This is the code I wrote to make the table:

```
import pandas as pd


# Define the data
data = {
   "Feature / Tool": [
      "Platform Integration",
      "Real-time Processing",
      "Context Filtering",
      "Emotion Detection",
      "Data Privacy Options",
      "Style & Language"
```

```
    ],
    "Otter.ai": ["Limited", "Partial", "No", "No", "Cloud only", "Limited"],
    "Fireflies.ai": ["Limited", "Yes", "No", "No", "Cloud only", "Moderate"],
    "Microsoft Copilot": ["Microsoft only", "Yes", "Basic", "No", "Enterprise-only", "Moderate"],
    "MinuteMate": ["Cross-platform", "Yes (segmented)", "Advanced", "Yes", "Local or cloud", "Full"]
}

# Create DataFrame and set index
df = pd.DataFrame(data)
df.set_index("Feature / Tool", inplace=True)

# Display table with centered content
df.style.set_properties(**{'text-align': 'center'}).set_table_styles([
    {'selector': 'th', 'props': [('text-align', 'center')]},
    {'selector': 'th.row_heading', 'props': [('text-align', 'center')]}
])
```

The Python-generated feature comparison table provided a clear and structured way to visualise how MinuteMate compares with three leading AI meeting assistants: Otter.ai, Fireflies.ai, and Microsoft Copilot. By translating the researched data into a consistent visual format, this approach helped eliminate ambiguity caused by incomplete or inconsistent data presentation in earlier drafts.

The visualisation highlights key competitive advantages of MinuteMate, including cross-platform compatibility, advanced context filtering, emotion detection, and flexible data privacy options, all areas where current market tools fall short. These strengths align directly with user and system requirements identified during the requirements capture phase and validate the technical direction of the proposed solution.

Using Python for this visualisation not only improved clarity but also allowed for easier updates and scalability if new competitors or features are added in future benchmarking. This process reinforced the importance of combining qualitative research with technical tools to enhance decision-making and stakeholder communication throughout the development lifecycle.

All code, documentation, and supporting material for this project can be found at:

https://github.com/aliallam123/MinuteMate/tree/main

The repository includes:

- Python code and feature comparison visualisation (Appendix D)
- Complete N8N workflow for real-time transcription and summarisation
- Setup guide for deploying MinuteMate
- Test workflow instructions and usage examples
- A copy of this final report

This marks the completion of the MinuteMate project and its supporting materials.