Soundwave Team: Matthew, Emily, Naima, Rachel, Julian, Yosief, James

Product Overview

Podcast production today is fragmented and inefficient. Small podcast teams and distributed co-hosts spend 5-12 hours per week managing files across disconnected tools, recording in Riverside, editing in Descript, sharing through Drive and Slack, creating version control chaos and waiting days for feedback. User research reveals that collaboration and version control, not advanced editing features, are top pain points, especially for podcasters without entire production teams. As one podcast creator explained: "Version control was a major pain point... we need channels for immediate feedback and easier ways to convey a vision—it would save us a lot of back and forth." Teams consistently rank real-time collaboration above editing speed or sophisticated audio controls, yet no existing platform enables true simultaneous editing for long-form audio content. SoundWave is launching **a cloud-based collaborative audio studio purpose-built for podcast teams.** By bringing recording, editing, commenting, and versioning into a single real-time platform (analogous to how Google Docs transformed document collaboration) SoundWave eliminates tool-switching friction and enables distributed teams to work on episodes simultaneously.

Unlike professional DAWs built for music production (which are the status quo for professional podcasters) or AI-simplified editors that sacrifice precision (which are common among beginners), SoundWave focuses on the collaborative workflow needs of spoken-word content creators. This product represents a strategic pivot from SoundWave's existing music-focused DAW, leveraging our proven real-time collaboration infrastructure to build a moat in the podcast production market where teams value quality and workflow efficiency over solo speed.

Scope and Objectives

This MVP leverages SoundWave's proven real-time collaboration technology to deliver a complete record-to-export podcasting workflow within a single collaborative environment. This product will not include advanced music production capabilities (VST plugins, complex mixing boards), AI-powered features, mobile applications, video editing, or third-party integrations with podcast hosting platforms. By constraining scope to the essential collaborative podcast editing loop, we can ship a focused MVP in 4 weeks that directly addresses validated user pain points while maintaining technical feasibility.

Primary Objectives

1. Deliver Core Collaborative Editing Platform: Launch a fully functional web-based studio within 4 weeks that enables real-time, Google Docs-style simultaneous editing with text-based waveform editing, timestamped comments, change syncing, and cloud versioning, eliminating the file export/import cycle entirely.
2. Validate Product-Market Fit with Podcast Teams: Onboard 10+ active podcast teams (3-5 professional teams plus amateur creators), in a month after the MVP is deployed, who complete at least one full episode in SoundWave, demonstrating the platform solves real workflow problems for collaborative creators.
3. Prove Collaboration Adoption: Achieve ≥40% of users actively using collaborative features (simultaneous editing, comments, shared sessions) compared to 15% collaboration usage in our music DAW.
4. Establish Conversion and Retention Benchmarks: Achieve 10% trial-to-paid conversion within 90 days, demonstrating users find sufficient value in the collaborative workflow.

Secondary Objectives

1. Build Scalable Infrastructure Foundation: Ensure the platform reliably supports 4+ concurrent editors per session without performance degradation, addressing known limitations from our music DAW and establishing technical credibility for future growth.
2. Gather User Feedback for Roadmap Prioritization: Conduct additional in-depth usability sessions with podcast teams to identify highest-value feature additions (e.g., advanced audio tools, publishing integrations, mobile apps) and inform future development.
3. Establish Architecture for Future Integrations: Design API and data export capabilities that enable future connections to podcast hosting platforms (Spotify, Apple Podcasts, RSS feeds)

Out-of-Scope/Non-Goals

- AI and Advanced Audio Features: No advanced AI-powered cleanup, mastering, or content generation. No VST plugin support or music production-grade mixing capabilities. Basic manual audio tools (EQ, compression, volume normalization).
- Multi-Platform Distribution: No mobile apps (iOS/Android). No direct publishing to podcast hosting platforms or RSS feed generation. Users export MP3/WAV files and handle distribution through existing channels.
- Third-Party Integrations: No integrations with Google Drive, Dropbox, Slack, etc.
- Video and Multi-Format Support: No video editing, no synchronization with video timelines, no support for formats beyond standard audio files.

User Personas

Based on our user research regarding fragmented workflows and version control chaos, we have identified three primary user personas. The core conflict SoundWave solves is the friction between the **Creator** (who cares about content/vision) and the **Producer** (who cares about

technical execution/editing)—even when these responsibilities are handled by the same person.

Persona 1: Creative Casey (The Host/Content Creator)

“I want to focus on the conversation and the story, not file management. I hate sending long emails with timestamps just to get a stutter removed.”

- **Demographics:** Age 25–40. Journalist, Subject Matter Expert, or Co-host. Non-technical.
- **Role:** Content creation, interviewing, storytelling.
- **Behaviors:**
    - Records remotely using Riverside or Zoom.
    - Listens to drafts on Google Drive and writes notes in a separate doc or Slack.
    - Often feels "locked out" of the editing process until the final cut.
- **Pain Points:**
    - **The Feedback Loop:** Frustrated by the delay between giving feedback and hearing the change.
    - **Tech Anxiety:** Fears losing recordings or messing up file transfers.
    - **Vision Loss:** Feels the editor sometimes cuts the "soul" out of a conversation because they can't collaborate in real-time.
- **Goals:**
    - Record high-quality audio without complex software.
    - Make "text-based" edits (delete this sentence) without needing to know how to use a waveform or listen to the entire recording.
    - Review and approve episodes faster to meet weekly deadlines.

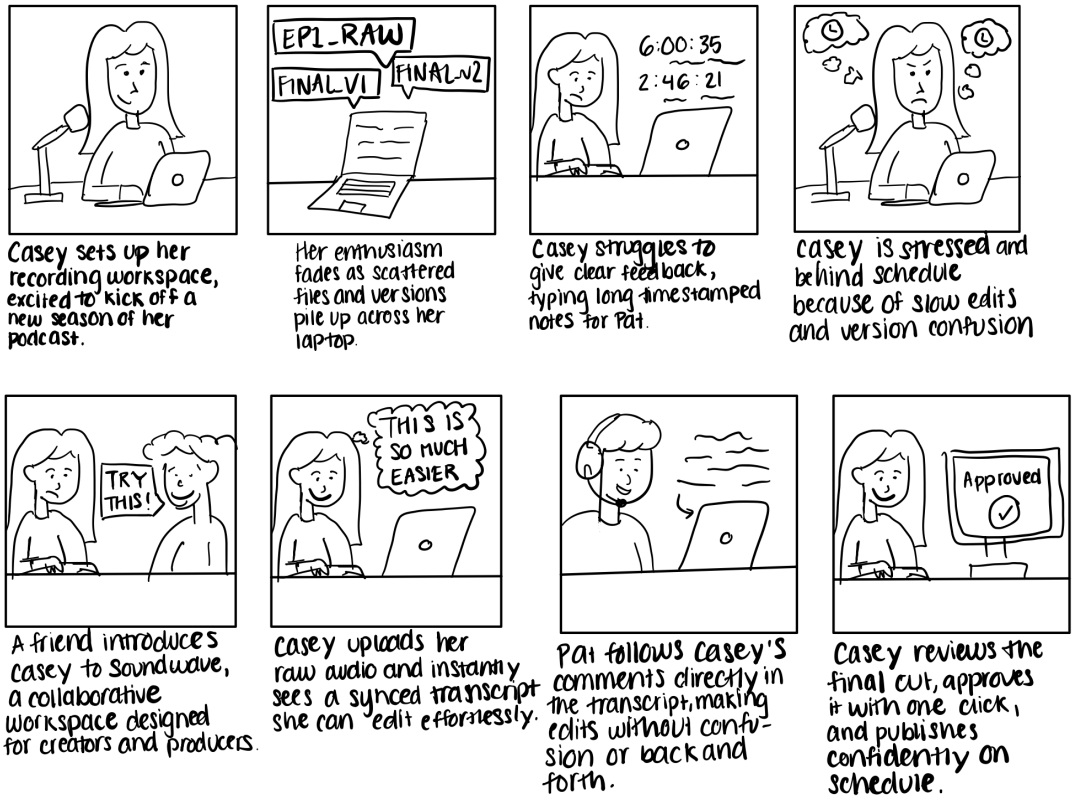
Persona 2: Producer Pat (The Editor/Audio Engineer)

“I spend more time managing versions and deciphering client emails than I do actually editing. I need a tool that lets me work faster, not harder.”

- **Demographics:** Age 20–45. Freelance Audio Engineer or Designated "Tech Savvy" Co-host.
- **Role:** Editing, mixing, noise reduction, final export.
- **Behaviors:**
    - Currently uses Reaper, Audacity, or Descript.
    - Juggles 3-4 different file versions (e.g., Ep1_Final.mp3, Ep1_Final_REAL.mp3).
    - Uses distinct tools for recording vs. editing.
- **Pain Points:**

    - **Version Control Chaos:** Managing files from multiple hosts sent via disparate channels (Drive, Dropbox, Email).
    - **Vague Feedback:** Receiving feedback like "fix the sound at the beginning" without a timestamp.
    - **Tool Switching:** Exporting from a recording platform, importing to a DAW, exporting for review, re-editing, exporting again.
- **Goals:**
    - A single source of truth for the project file.
    - Precise audio control (EQ, Compression) combined with the speed of transcript editing.
    - Immediate client sign-off without rendering multiple draft files.

User Stories



Jobs To Be Done (JTDB)

When...	... I want to ...	... so that I/we can ...
When I record a new episode with a guest,	I want to capture high-quality audio without having to use complex tools	So that I can focus on the conversation and story instead of technology.
When I review an edit as a podcast host,	I want to make text-based edits, such as “delete this sentence,” directly on the transcript	So that I can quickly edit clips without having to watch all the hours of footage repeatedly.
When I edit as a producer,	I want to see the host’s comments anchored to exact words in the transcript	So that I can understand exactly what they want changed instead of deciphering vague feedback.
When a new collaborator joins the team mid-season,	I want to give them access to past episodes, comments, and decisions in the same workspace	So that we can start working together immediately without digging through scattered files.
When I’m worried about losing assets,	I want to be able to safely store recordings and edits in the cloud	So that I can edit without fear knowing our work is backed up.
When I want to refine the creative vision over time,	I want to easily search transcripts across episodes	So that I can repurpose clips without re-listening to hours of audio.
When we deliver the final cut,	I want to get quick sign-off from all stakeholders directly in the project	So that I can complete the episode without rendering and sending multiple “final” files.

Functional Requirements

The core functionality of our collaborative podcast platform centers on supporting the full workflow that podcast teams follow when creating an episode together. Since our research showed that the biggest problems today come from scattered tools, unclear versions, and slow feedback cycles, the most important requirement is a shared editing space where multiple creators can work at the same time. This real-time environment must allow users to see edits, selections, and structural changes from their teammates as they happen.

The other requirements are listed in order of descending priority, starting with this shared space. Within this shared space, creators also need the ability to record directly inside the browser. Many podcast teams record remotely, and moving large files between different applications slows the entire process. This feature depends on browser based audio capture and cloud storage but does not require advanced processing, making it achievable within our timeline.

Once audio has been captured, editing becomes the main task. Spoken word editing is largely driven by removing filler words, tightening pacing, and restructuring conversations, so being able to modify the transcript and see those edits reflected in the audio is essential. This functionality relies on accurate speech-to-text conversion and must integrate smoothly with the shared editor so that multiple users can make changes without conflicts. Another problem identified in interviews was the difficulty of giving feedback across multiple apps. To solve this, the platform must include a built-in commenting system tied to specific timestamps. This allows creators to ask questions, note improvements, and direct others' attention to moments inside the episode without leaving the editing environment. Because comments need to remain synchronized with the episode, this feature depends on the stability of the collaborative editor. Similarly, the platform must maintain a clean and understandable version history; Soundwave should automatically save meaningful changes and allow users to return to earlier points in the editing process (similar to Google Docs). This ensures that no one has to duplicate work or worry about overwriting someone else's progress.

Finally, the platform needs to support importing and exporting audio so that creators can bring in tracks recorded elsewhere or export the final product for distribution. Simple audio tools such as trimming, adjusting volume, and adding basic fades are enough to support high-quality spoken word editing. These controls build on our existing audio engine and do not require the complexity of a music production suite. Together, these functional requirements establish a complete workflow for teams who want to collaborate smoothly from recording to publication without relying on a patchwork of external tools.

Non-Functional Requirements

The non-functional requirements determine the level of quality and reliability the platform must provide in order to support real podcast production. Because the central promise of the product is real-time collaboration, the editor needs to remain responsive even when several people are working together on an episode, and changes should appear for collaborators within a near invisible delay. The system must also handle full-length podcast episodes, which, based on interviews, typically extend up to ninety minutes. Podcast teams also store multiple recordings, drafts, and versions, so the platform must grow with their usage without slowing down. We should be able to scale the product to more users with larger teams without requiring architectural redesigns.

Security is an important concern because podcast creators often work with unreleased or sensitive material. All audio files, transcripts, and session data must be stored securely, and communication between clients and servers should use encrypted channels. Although our product does not operate under industry-specific regulations like food safety or medical compliance, it still must meet general expectations for data privacy. These expectations are especially important for creators who release episodes on fixed editorial schedules and need confidence that the platform will protect their work.

Finally, the platform must be reliable. Recordings should be protected by local backup audio when possible so that connection issues do not result in lost content. The editor should work consistently across major desktop browsers, and when conditions are suboptimal, the platform should degrade gently rather than failing outright. Together, these non-functional requirements help ensure that the product is not only feature complete but also stable, secure, and dependable enough for teams to rely on during weekly production cycles.

Prototypes

Live Prototype

Demo Video

Constraints and Assumptions

Constraints

1. Web-Based Only (MVP): The product must function entirely in the browser. No mobile applications (iOS/Android) or desktop clients will be developed for the MVP.

2. Use of Existing Infrastructure: The MVP must leverage SoundWave's current cloud-based DAW infrastructure, including real-time collaboration technology, WebRTC architecture, and cloud storage systems.
3. Four-Week Timeline: The core collaborative editing platform must be launched within four weeks, requiring focused scope and minimal technical deviation (See "Scope and Objectives").
4. Feature Scope Limitations The MVP is restricted to podcast and spoken-word editing. It will not include:
    - Advanced music production tools (e.g., VST plugins, complex mixing boards)
    - AI-powered cleanup, mastering, or content generation
    - Direct publishing or third-party integrations (e.g., Spotify, Google Drive, Dropbox, Slack)

Assumptions

1. Simplicity Over Complexity for Podcast Teams: Podcast editors prioritize a streamlined, easy-to-use workflow over highly advanced audio engineering features. User research supports this assumption: version control, real-time feedback, and collaboration are their primary pain points, not sophisticated editing tools.
2. Existing Collaboration Infrastructure Is Sufficient: We assume SoundWave's existing real-time collaboration architecture is stable enough to support 4+ concurrent editors with acceptable latency. *Note:* We could not directly test this because we do not have access to a fully testable version of the original SoundWave DAW.
3. Team-Based Podcast Production Represents a Viable Market: We assume collaborative podcast teams represent a sufficiently large and underserved segment to justify a dedicated product focused on workflow efficiency. *Note:* This assumption was preliminarily explored in our early TAM/SAM/SOM analysis but requires further validation.

Ethical Implications

1. Data Privacy: Podcast recordings often contain sensitive or unreleased content. We must ensure encrypted storage, clear data retention policies, and explicit consent mechanisms for all recording participants.
2. Accessibility: While transcript-based editing improves accessibility for deaf/hard-of-hearing users, the interface must meet WCAG 2.1 AA standards for all users with disabilities.

Milestones and Timeline

Phase 1: Initial Prototyping

Month 1:

- Development and refinement of initial prototypes (MVP), including early wireframes for main user flow and key user interfaces.
- Refactoring of existing SoundWave DAW codebase (removing music-specific features and enabling more text-based collaborative editing)
- Test with a small group of initial users to guide more high-fidelity prototype
    - No more than 3 people collaborating on a podcast at once

Month 2-3:

- Development of high-fidelity prototypes and core editing interactions
- Additional testing for small group editing experience
- Begin expanding and testing collaboration model on larger groups (4 - 8 people)

Phase 2: In-Depth Customer Research and MVP Refinement

Month 4-6:

- Begin marketing push, targeting podcasters, audio engineers, and small production teams.
    - Engage podcasting communities on Facebook, Slack, and other online groups
    - Build landing page & waitlist
    - Acquire users who want to beta test
- Conduct interviews with target users

Month 7-12:

- Continue marketing push (potentially hiring a designated PMM)
- Large scale QA
- Prepare for soft launch with analytical tools, evaluation criteria, customer support, etc
- Establish objectives and OKRs

Phase 3: Soft Launch and Iterative Feedback

- Soft launch with 10-20 podcasting teams (likely those involved with earlier research or on the waitlist)
- Gather data on workflow friction and latency

- Rapid iterations cycles: fix crashes, push bug patches, improve performance

Phase 4: Hard Launch and Market Expansion

First half:

- Hard launch of Soundwave to the public
- Broader marketing push across Youtube tutorials, creator sponsorships, brand partnerships, and targeted advertisements.

Second half:

- Evaluation of user data and feedback post-launch for further product refinement
- Exploration and initiation of new features, such as AI-generated chimes, multi-platform distribution, third-party integrations
- Expand to more diverse markets via outreach to small podcasting studios and podcast networks.

Resource Requirements

1. Human Resources

Rationale: Building Soundwave will require heavy frontend logic for intuitive UI/UX and a robust backend for real-time synchronization.

Roles (1 person for each role in the first few months)	**Responsibilities & Focus**
Product Manager	Defining the strict MVP scope (cutting non-essential features based on needfinding and user testing), managing the beta launch with 10+ teams , and making sure that the team is on track with the timeline. Since we are resource constrained, it may be necessary also to have the PM take on some PMM responsibilities so that we do not have to hire a separate marketing specialist.

Solution Architect	Adapting the existing "music DAW" backend to support long-form podcast data. Responsible for architecture stability and ensuring latency.
Frontend Engineers	Implementing the complex "text-based editing" interface where deleting text deletes audio and audio is also searchable via text search. Handling the visual waveform rendering and browser-based recording. To meet the <200ms latency requirement, the frontend should also focus on being able to register user comments *immediately* to the creator while simultaneously sending the data to the backend.
Backend/DevOps Engineer	Managing the cloud storage of large audio files (90-min episodes), API integrations (Speech-to-Text), and version control database logic. When a producer marks a comment as "Done," the backend updates the status and broadcasts the change so the comment fades out or archives for all users simultaneously.
UI/UX Designer	Designing the "streamlined" interface to differentiate from complex tools like Reaper/Audacity. Creating intuitive flows for commenting and conflict resolution. The goal is to have a drastically easier onboarding experience than existing pro tools. The UI/UX Designer will also work with the PM on UX research by leading needfinding and user testing interviews.
QA Engineer	Executing the stress tests (10 concurrent users) and verifying the "99.9% Audio Integrity" requirement.

2. Technical Resources & Tech Stack

Category	**Resource / Tool**	**Justification from PRD**

Core Infrastructure	**Cloud Provider (AWS/GCP)**	Scalable compute for the web app and Object Storage (S3/GCS) for large, uncompressed audio files and version history.
Real-Time Tech	**WebRTC / WebSocket Cluster**	The backbone of the product. Required to sync cursor movements, text selections, and comments in "real-time".
AI/ML Services	**Speech-to-Text API**	An external enterprise-grade API (e.g., OpenAI Whisper) is required to power the "transcript-based editing" feature. This may be a significant cost depending on how much content users are editing with the speech-to-text feature.
Browser APIs	**Web Audio API**	Required to capture high-quality microphone input directly in the browser without plugins.
Collaboration	**Figma Make / Prototyping**	For rapid iteration of the UI/UX, specifically ensuring the "collaborative" visual cues (cursors, comment bubbles) are intuitive.

Testing and Quality Assurance

To ensure platform reliability for the 4-week MVP launch, our QA strategy focuses on three core pillars:

- **Unit Testing:** Validation of individual component logic, prioritizing the Audio Engine (cutting/fading accuracy), Backend Storage (large file uploads >1hr), and Collaboration Syncing (merge logic).
- **Real-Time Stress Testing:** Simulating sessions with 10 concurrent users to verify conflict resolution, ensure zero data loss under load, and validate the <200ms latency target.
- **Integration & Beta Testing:** End-to-end workflow validation (Import $\rightarrow$ Edit $\rightarrow$ Export) conducted with a closed beta group of podcasters to identify edge cases and usability friction before public release.

Metrics and Success Criteria

To measure the effectiveness of the SoundWave collaborative studio, we will focus on metrics that validate engagement with collaborative features rather than just raw user acquisition. We need to prove that users are not just editing, but editing *together*.

Key Performance Indicators (KPIs)

Metric Name	Definition	Why
Weekly Active Collaborative Sessions (WACS)	Number of editing sessions per week where multiple users interact within the same project.	Proves the core value prop (collaboration) is being used.
Feature Utilization: Comments	Average number of comments created per episode project.	Indicates that the feedback loop is moving out of Slack/Email and into SoundWave.
Time-to-First-Export	Average time elapsed from account creation to the first completed audio export.	Measures the ease of onboarding and the "Record-to-Export" workflow efficiency.
Project Continuity Rate	Percentage of teams who start a second episode within 14 days of finishing their first.	Validates that the workflow is sustainable for recurring content creators.
Trial-to-Paid Conversion	% of users who convert to a paid subscription after the MVP trial period.	Validates perceived value and willingness to pay.

Success Criteria (MVP Launch)

We will consider the MVP launch a success if we achieve the following benchmarks within 30 days of public release:

1. **Adoption Benchmark:**
    - Onboard **10+ active podcast teams** (defined as teams producing at least one episode per week).
    - Achieve **≥40% rate of collaborative sessions** (as opposed to solo editing), validating the hypothesis that users prefer real-time teamwork.
2. **Performance Benchmark:**
    - Platform successfully supports **3 concurrent editors** in a single session with <200ms latency on visual updates (cursor movements, text selection).
    - **99.9% Audio Integrity:** Zero reported instances of data loss or file corruption during cloud syncing.
3. **Workflow Validation:**
    - **7 out of 10** beta users report that SoundWave saved them at least **2 hours** of production time per episode compared to their previous stack.
4. **Business Viability:**
    - Achieve a **10% conversion rate** from free beta/trial to paid tier (or intent-to-pay waitlist if billing is not ready).
    - Users report a Net Promoter Score that is higher than 6.

Go-to-Market Strategy

Pre Launch

MVP Soft Launch (Beta)

We will soft-launch with 10–20 waitlist teams to validate collaboration usage and identify workflow friction. We will push weekly bug fixes to refine our MVP as we prepare for public launch.

Distribution Strategy

We will distribute online through a landing page + waitlist, podcasting Slack/Facebook communities, university programs, and partnerships with freelance editors and small agencies. Early users will come primarily from research contacts and the waitlist.

Marketing & Promotion - Messaging

“SoundWave: Edit podcasts together, in real time, with no files, no exports, no chaos.”

Messaging focuses on collaboration, simplicity, and time saved through a single shared workspace.

Launch

Hard Launch to Public

Set up a Freemium pricing model to bring in first revenue. Continue engineering efforts to fix bugs and respond to customer service requests.

Customer Support & Service
Support includes an in-app help center, email support, feedback prompts, and optional onboarding for weekly-production teams. The focus is fast issue resolution and capturing insights for iteration.

Distribution Channels
Announce product launch across multiple channels, i.e. Product Hunt debut, Slack/Facebook groups and newsletters, Youtube tutorials, Soundwave waitlist.

Marketing & Promotion
Ramp up marketing & promotion, especially across 2 most successful channels. Expand partnerships with podcast studios and launch Google/Youtube targeted search ads. Gather and post customer testimonials.

Funnel:
Top: demos, community posts, waitlist
Middle: landing page → onboarding
Activation: first recording/upload + collaborative edits
Retention: full-episode completion and repeat use

Post Launch

Evaluation Plan: Pivot vs. Persevere

- **Persevere:** If WACS (North Star) is high and retention is strong, we proceed to Phase 2 (Mobile App & Advanced Audio Tools).
- **Pivot:** If users are signing up but *only* using the tool for solo recording (low collaboration usage), we may need to pivot the feature set to focus more on AI-enhancement or solo-ease-of-use rather than team collaboration.
- **Perish:** If we cannot maintain audio sync reliability (technical failure) or if very few teams complete a full episode

References

[1] https://edu.soundtrap.com/what-is-a-daw-and-why-does-it-matter

[2] https://www.gminsights.com/industry-analysis/podcasting-market

[3] https://www.linkedin.com/pulse/2024-podcast-industry-report-market-size-growth-trends-nnkdc

[4] https://www.castmagic.io/post/the-benefits-of-transcribing-podcasts-for-accessibility-and-seo

[5] https://contentallies.com/learn/top-podcast-collaboration-tools

[6] https://www.tracklib.com/blog/benefits-online-daw

[7] https://www.podcastvideos.com/articles/riverside-vs-descript-vs-podcastle-2025

[8] https://www.lucidlink.com/blog/podcast-editing

[9] https://www.reddit.com/r/podcasting/comments/1djpfo9/transcriptionbased_editing_in_descript_vs

[10] https://docs.google.com/document/d/1bVD3JIussthj8R0OjBSq6AlIt3s_pSEsXAvRF29X9xk/edit?tab=t.0#heading=h.lxs2a486aevk

[11] https://www.deviqa.com/blog/qa-for-startups-the-key-to-sustainable-success/

[12] https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API

[13] https://coachchristinanm.medium.com/i-used-riverside-fm-to-start-a-podcast-here-is-my-review-9543aed73e84

[14] https://www.musicradar.com/news/the-best-daws-the-best-music-production-software-for-pc-and-mac

[15] https://www.atlassian.com/git/tutorials/what-is-version-control

[16] https://www.proofhub.com/collaboration/types/real-time

Our team also used Claude and Gemini to brainstorm ideas.