

Request for

Proposal

Clario

September 2025

Team Member:   
Cayleigh Leishman -> [cayleigh.leishman@gmail.com](mailto:cayleigh.leishman@gmail.com)

Instructor/Advisor:

Robert Kumar -> [kumarr@byui.edu](mailto:kumarr@byui.edu)

Stakeholders:

* Deaf and Hard of Hearing Clients - People who need live notes for meetings
* Transcribers - People who type the notes live
* Instructor - Oversees the Project and Grades Progress
* Accessibility Unit - People in charge of managing accessibility at universities or businesses.

# 

# Introduction & Background

We are making a website with two main parts, one for each kind of user:

1. **Client’s Page** – For the person who needs notes. They can log in, see past notes, and manage their meetings.
2. **Transcriber’s Page** – For the person who types the notes. They log in, join the meeting, listen live, and send finished notes back to the Client.

To make Clario really useful, we’re focusing on four main goals that guide how the site will work and look.

### Four Main Goals

We will focus on these four goals to make Clario work well:

We want Clario to really help people, so we are focusing on four main goals:

1. **Help People in Real Time:** Clients should get their notes during meetings without any delay. Clario’s main job is to be useful exactly when it’s needed.
2. **Keep Data Safe:** All notes and personal info must stay private. Only the right people can see what they’re supposed to.
3. **Make It Easy to Use:** The website should be simple and clear for both clients and transcribers. Buttons, pages, and instructions should be easy to understand.
4. **Work for Everyone:** Clario should be easy to use for people with hearing, vision, or other difficulties. Following accessibility standards helps everyone get the help they need.

By achieving these goals, Clario will not only help users, but also show the skills I’m learning in a professional, resume-ready way.



# Project Description

The purpose of this project is to build a new website called Clario.

* Make a Working Site: We need to finish a website that actually works and shows off the important features (like typing notes and logging in).
* Use Good Tools: We want to show that our special building tools (SvelteKit and Supabase) are fast and safe for making the website.
* Be Easy to Use: The whole website must be clean, simple, and easy for everyone to use when getting notes or typing notes.

The description of the project is as follows:

1. Two Clear Roles: The site will know if you are the Client (the one who needs the notes) or the Transcriber (the one who types the notes). Right from the login, you will see your own special page for what you need to do.
2. Modern and Clean Look: The website will look new and clean. It will have a clear menu on the side (for things like Notes or Settings) and will implement accessible features as it’s called for.
3. Safe Connections (and Storage): We will use Supabase to handle all the logins, user names, and the real-time note-taking. This keeps all your information and the notes safe and working smoothly.

# 

# Significance/Resume Value Statement

This project shows that I can take an idea from start to finish and make it work. By using modern tools like SvelteKit and Supabase, I will show that I understand both front-end design and back-end data storage.

Finishing Clario will also be proof that I can make websites that help people with different needs and keep private data safe. These are the kinds of skills employers look for in entry-level developers.

# 

# New Computing Science Concepts:

* **Real-time syncing** – Making notes update instantly between two users using Supabase.
* **Secure authentication** – Making sure only the right person can log in and see notes.
* **Audio streaming** – Sending the client’s voice to the transcriber through the browser.
* **Accessibility standards** – Learning how to follow WCAG rules so people with disabilities can use the site.
* **Responsive design** – Making sure the site works well on phones, tablets, and computers.

To make this project work and be meaningful, I’ll need to learn some these computer science concepts.

# Background/Prior Knowledge

I already know how to build websites using HTML, CSS, and JavaScript. I have started learning SvelteKit to make faster and more modern pages, and Supabase to handle logins and saving data securely.

I have also practiced making websites that are clear and simple, but this project will be my first time building real-time features and testing accessibility. It is a chance for me to use what I know and learn new tools at the same time.

# Interestingness/Motivation :

I’m excited to build Clario because it’s something I wish I had during my time at BYU–I. There were times when keeping up with notes in class or meetings was hard, and having a tool like this would have made a big difference.

Building Clario gives me the chance to practice real-world skills like planning, designing, and creating a working prototype. It’s not just another project for me — it’s a way to make something modern, safe, and easy to use that could help people in the same situation I was in.

To turn this idea into a working site, I’ll be using a set of tools and resources that help me build everything efficiently

# Resources:

* **SvelteKit & Vite** – These help me make the website fast and modern. They let me build pages quickly without starting from scratch.
* **Supabase** – This handles logins, keeps notes safe, and lets updates happen in real-time. It’s like the website’s memory and security guard all in one.
* **Simple-Peer** – Makes sure the audio from the Client can reach the Transcriber without too much lag or trouble.
* **RecordRTC** – Lets me record audio sessions if needed.
* **TypeScript & ESLint/Prettier** – Help me write code that’s easier to read, less likely to break, and looks consistent.
* **Other development tools** – This includes Svelte, plugins, and extra packages that help the website run smoothly and make sure I’m following best practices.
* **VS Code** – Where I write all the code.
* **GitHub** – Where I save my work so it’s safe and I can share progress with my instructor.
* **Figma** – To make simple designs and diagrams before building pages.
* **Browser APIs** – These let me handle audio and video right in the browser without extra software.

These tools give me everything I need to make the first version of Clario. They help me make the website work fast, keep data safe, and make it easy for everyone to use. Even though some features are tricky, these resources help me manage the risks and make sure the site works well. These tools are the foundation, but we also rely on several technical packages to make everything run smoothly

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# Dependencies:

SvelteKit & Vite – For building the frontend quickly and efficiently.

* **SvelteKit & Vite** – These are the backbone of the frontend, letting me build pages quickly and efficiently while keeping them modern and responsive.
* **Supabase (@supabase/supabase-js, @supabase/ssr, @supabase/realtime-js)** – Handles authentication, real-time updates for live notes, and secure data storage.
* **Simple-Peer** – Manages WebRTC connections for live audio streaming and optional one-way video from Client to Transcriber.
* **RecordRTC** – Enables optional recording of audio sessions if needed for reference.
* **TypeScript & ESLint/Prettier** – Ensures type safety, clean code, and consistent formatting across the project.
* **Other dev dependencies** – Includes Svelte, Svelte plugins, and supporting packages for formatting, linting, and code checking.

These dependencies are essential for making Clario fast, reliable, and secure. Each package supports the real-time features, safe authentication, and overall usability of the website, helping ensure that Clients and Transcribers have a smooth experience. Even with the right tools and packages, there are some challenges we need to watch out for.

# Risks:

* **WebRTC complexity** – Setting up reliable one-way audio/video streams can be tricky, especially handling network interruptions or multiple browsers.
* **Real-time sync issues** – Using @supabase/realtime-js for live note updates may introduce latency or data conflicts if not handled carefully.
* **Browser compatibility** – Features like WebRTC and audio recording may work differently across browsers, requiring additional testing and possible workarounds.
* **Learning curve** – I haven’t fully implemented WebRTC or recordrtc before, so figuring out how to connect audio/video correctly and securely may take extra time.
* **Security** – Handling user audio and possibly video data must be done safely, including authentication and permissions, to protect privacy.
* **Scope creep** – Adding features like one-way video or audio recording could expand the project beyond the planned timeline if not carefully managed.

# 

# Submission Guidelines & Requirements

* We will turn in all our work on the school’s website (I-Learn/Canvas).
* We will send our files or links to gihub files under “developerNotes”
* Each file will have our project name, date, and version on it so it’s easy to know what it is.
* If the teacher needs a password to see something, we will give it.
* We will turn everything in on time so we don’t lose points.
* If we use help from other websites, tools, or tutorials, we will write where we got it.



# Project Scope

To finish the Clario website and call the project a success, all these six rules must be true:

1. **Fast Website:** The front part of the site must be built with the SvelteKit tool so it is very quick and fast when people use it.
2. **Safe Login:** The login and log-out buttons must be safe and actually work for everyone using our Supabase tool.
3. **Good Storage:** The database (the website's big memory) must be set up correctly to save and find all the notes and user names we need.
4. **Right Door:** When a Client logs in, they must go to the Client's special page. When a Transcriber logs in, they must go to the Transcriber's special page.
5. **Color Switch:** The website must easily change between a light (day) look and a dark (night) look based on what the user wants.
6. **Working Connection:** We need to clearly show that one Client and one Transcriber can securely connect to each other for note-taking.

Here’s the plan for when each part of the project will be done



# RFP & Project Timelines

## The Request for Proposal timeline is as follows:

|  |  |
| --- | --- |
| **Project Set up and Backend Foundation**  The core technical environment will be set up with SvelteKit and Supabase, establishing the initial database schema and operational login functions. | 10/04/2025  (1 week) |
| **Front-End Development & UI completion** All necessary pages and functional navigation will be completed for both user roles, including buttons and space for accessibility features. | 10/18/2025  (2 weeks) |
| **Core Features**  The system will integrate note-taking and data persistence with Supabase, placing the audio feature and starting accessibility improvements. | 11/08/2025 (3 Weeks) |
| **Design Refinement & Advanced Features** The application will achieve full responsiveness and complete its accessibility review, along with the implementation of a functional user notification system. | 11/22/2025  (2 weeks) |
| **Testing, Debugging and Stabilization**  A final quality assurance pass will resolve all known bugs related to login and data saving to deliver a stable, finalized prototype. | 12/07/2025  (2 weeks) |

See Github Milestones link: <https://github.com/CayleighLeishman/Clario/milestones>  
Since this is a school project, we don’t need a big budget, but here’s what we’re starting with.



# Budget

**No Money**: This is a school project, so we have no budget (no money).

**Small Start:** Because of this, the first version will only let 4 Clients and 5 Transcribers use it.

**Ready to Grow**: Even though it starts small, I’m planning on building it in a smart way so it can be easily made bigger later if it ends up getting used by more people.