**Maintenance Guide**

**System Dependencies:**

First, we will cover the essential dependencies for both the client-side (react web application) and server-side components of our project. We'll highlight key libraries and dependencies used in each environment.

**Client**

The following libraries are essential for running the client-side of our application:

* **React:** A JavaScript library for building user interfaces. (Version: ^18.2.0)
* **React DOM:** Provides DOM-specific methods for React. (Version: ^18.2.0)
* **React Router DOM:** Declarative routing for React. (Version: ^6.22.0)
* **Tailwind CSS:** A utility-first CSS framework for rapid UI development. (Version: ^3.4.1)
* **SweetAlert2:** JavaScript library for creating customizable elegant and responsive alerts (Version: ^ 11.10.5)

**Server**

The server-side of our application relies on the following dependencies:

* **Node.js:** A JavaScript runtime for server-side development. (Engine Version: 20.x)
* **Express:** Fast, unopinionated, minimalist web framework for Node.js. (Version: ^4.17.1)
* **Firebase Admin:** Firebase Admin SDK for accessing Firebase services from privileged environments. (Version: ^12.1.0)
* **Microsoft Cognitive Services Speech SDK**: SDK for integrating speech capabilities into applications. (Version: ^1.36.0)
* **Nodemailer:** Module for sending emails with Node.js. (Version: ^6.9.10)
* **FFmpeg and FFprobe Installers:** Tools to install FFmpeg and FFprobe, essential for audio and video processing and editing. (Versions: ^1.1.0 FFmpeg, ^2.1.2 FFprobe)
* **dotenv:** Loads environment variables from a .env file into process.env. (Version: ^16.4.4)

The complete list of dependencies can be located within the package.json file of each environment.

**Managing Tools and Subscriptions of The Project**

In this section, we will provide a brief overview of the tools, APIs, and services utilized in our project, highlighting their functionalities, usage costs, and maintenance requirements.

**OpenAI**

OpenAI provides access to the ChatGPT AI model, in our project we are using ChatGPT 3.5 turbo model, allowing for natural language processing and conversation generation. Calls to ChatGPT API incur costs per usage, we deposited a small amount of money for the account in order to get access to OpenAI, it's important to monitor the account balance to ensure continued access to the AI model.

**Firebase**

Firebase is currently used for a not mandatory user authentication (username, email and password). Its has the potential to scale with additional Google Cloud services such as Google Cloud Storage (Google Buckets) for storing user-generated videos and metadata. Cloud Storage incur costs based on storage usage and data retrieval, monitoring this costs is essential when scaling the project.

**Render**

Render serves as the project's deployment platform, offering free reliable hosting and enabling seamless deployment and management of our applications. However, to access advanced capabilities and achieve optimal performance, such as efficient video editing, a paid subscription is crucial. Without a payment plan, video generation can take up to 10 minutes, whereas with a paid plan, the same operation can be completed in about 1 minute. This underscores the significant performance improvements achievable through Render's monthly paid subscription, ensuring efficient and responsive application functionality.

**Azure**

Azure plays a crucial role in our project by enabling essential Text-to-Speech (TTS) and Speech-to-Text (STT) functionalities, enhancing user interaction and accessibility within the application. Currently, we leverage Azure services through our Azure account at Braude College without any additional costs. However, it's important to note that there may be a need to transition to a new account in the future if the college account becomes unavailable. Regularly monitoring usage and account status is essential to avoid interruptions in the application service. Azure's robust capabilities empower our project with advanced audio processing features, contributing to a richer user experience.

**Gmail Account**

Our project utilizes the Gmail account vidwizardweb@gmail.com to receive emails from the 'Contact Us' page. It's important to keep this email active and regularly monitor notifications to maintain effective communication with our users. In the future, this account can also serve to send updates about new features and announcements to our user base, enhancing engagement and communication.

Guidance for running and deploying the project can be found in the README file or in the User Guide among other useful information.