WhisperSend - Maintenance Guide

# Purpose:

This guide aims to ensure that WhisperSend can be maintained, updated, and improved even after the initial project completion. It covers the installation, setup, and key components required for the system's operation, as well as instructions for applying changes, updates, and improvements.

# System Overview:

WhisperSend is a peer-to-peer file-sharing application that allows encrypted file transfers and communication using asymmetric encryption (public/private keys). It’s designed for maximum privacy and security without relying on a central server.

# System Dependencies:

## Client-Side (Frontend):

The client-side is built using Preact with additional libraries that are essential for the application's functionality:  
- Preact: A lightweight alternative to React. (Version: ^10.23.1)  
- React: Compatibility layer provided by preact/compat. (Version: ^17.1.2)  
- React Router DOM: Declarative routing for Preact. (Version: ^6.22.0)  
- Tailwind CSS: Utility-first CSS framework for rapid UI development. (Version: ^3.4.1)  
- PeerJS: Used for peer-to-peer communication. (Version: ^1.3.2)  
- ethers: Used for cryptography and wallet generation. (Version: ^5.7.2)  
- react-icons: Icons for UI design. (Version: ^4.8.0)  
- react-toastify: Provides notifications. (Version: ^9.0.0)  
- emailjs-com: For sending emails. (Version: ^3.2.0)  
- emoji-mart: Emoji picker for chat functionality. (Version: ^5.3.0)

## Server-Side (Backend):

The backend leverages Node.js and related libraries for the server-side functionality:  
- Node.js: JavaScript runtime environment. (Engine Version: 20.x)  
- PeerJS Server: Manages peer-to-peer connections. (Version: ^1.0.2))

# Installation and Deployment:

1. **Local Development Setup**  
     
   1.1 Clone the Repository:  
    git clone https://github.com/jessicamirov/WhisperSend.git  
    cd WhisperSend  
     
   2. Install Dependencies:  
    npm install  
     
   3. Running the Development Server:  
    npm run dev  
    The app will run on `http://localhost:3000`.

# Deployment:

WhisperSend can be deployed on services such as Vercel. You can watch the deploy online on the vercel server by clicking this link: ' https://whispersend.vercel.app/'

# Managing Tools and Services:

- **PeerJS:** Ensure that the PeerServer is properly configured and running. This server manages the signaling process and should be monitored for connectivity issues.  
- **Gmail Account:** Users can access the Contact Us section on the website, which is linked to a dedicated email address. When a user sends an email it is automatically sent to the support team using emailjs-com, in our mail address: whispersendorg@gmail.com

# Key Features to Maintain:

1**. Peer-to-Peer File Sharing**: Ensure PeerJS connections are functional, and the encryption/decryption processes remain secure.  
2. **Encryption/Decryption**: WhisperSend uses asymmetric encryption with public and private keys. The encryption protocol should be updated as needed to maintain security.  
3. **Real-Time Chat**: Maintain the chat interface for smooth user experience, ensuring emoji support and message encryption remain functional.  
4. **Mnemonic Management**: Ensure that users’ mnemonic phrases are properly generated and stored securely for future use in encryption and decryption.  
5. **Browser Compatibility**: Since WhisperSend is a WebRTC-based peer-to-peer application, it's important to ensure compatibility with modern web browsers:

* Google Chrome
* Mozilla Firefox
* Safari
* Microsoft Edge

Make sure to track browser updates to maintain WebRTC functionality.

# Future Updates and Improvements:

- Security Updates: Regularly check the dependencies for vulnerabilities, especially the cryptographic libraries like `ethers` and `tweetnacl`, and apply updates when necessary.  
- Performance Enhancements: As user traffic grows, consider scaling the PeerJS signaling server and optimizing the chat interface for better performance on low-resource devices.  
- User Feedback Integration: Continue monitoring the Gmail account and apply user-suggested improvements to ensure WhisperSend remains user-friendly.

# Conclusion:

This maintenance guide provides the necessary information to continue the development and support of WhisperSend. By following these steps, updates and improvements can be implemented seamlessly, ensuring the platform remains secure, user-friendly, and functional.