Simplify Web App with Speak Feature

# 1. Solution Details

This is finance advisor web app with enhancement adds differentiated behavior between voice-input and typed-input in the chat app:  
- If the user speaks a query using the microphone button, the bot’s reply is automatically read aloud.  
- If the user types a query, the reply is displayed as text only (no auto-speak).  
- In both cases, the user can still use the '🔊 Read' / '🔇 Stop' button to manually toggle text-to-speech.

Key implementation details:

-We need OpenAI API key in .env file in backend or remove the code of query api but a hardcoded string in return value.  
- Introduced `lastInputMethodRef` (voice | text) to track how the last query was entered.  
- Updated `handlechange` (typing) and `rec.onresult` (voice input) to update this reference.  
- Modified `dothis()` to set a per-message flag `shouldAutoSpeak`.  
- Auto-read effect checks if the message has `shouldAutoSpeak` before initiating speech synthesis.  
- Manual TTS toggle still works in both cases.

# 2. Deployment Notes

To deploy this updated feature:  
1 We need to run two Sever one for backend and Frontend. We need to put the backend api query in frontend.  
2. Ensure that dependencies are installed:  
 - `react-markdown`  
 - Browser support for Web Speech API (`window.SpeechRecognition` and `window.speechSynthesis`).

- pip install -r requirements.txt  
3. Rebuild the project with your package manager:  
 - `npm run build` or `yarn build`  
4. Deploy the build to your hosting service (e.g., Netlify, Vercel, Render).

# 3. Execution Notes

Steps to execute locally:

1. You need to Have Node and Python install.  
2. Start your backend service on port 5000 (as the frontend fetches from `http://127.0.0.1:5000/query`). Use python main.py to run backend and be in backend folder before running it.  
2. Run the frontend locally be in react folder:  
 - npm install and npm run dev  
3. Test the following scenarios:  
 - Speak a query → Response should auto-speak.  
 - Type a query → Response should remain text-only.  
 - In both cases → Test the 🔊 / 🔇 button to confirm manual control works.  
4. Confirm behavior across different browsers (Chrome recommended as it supports the Web Speech API).

# 4. Challenges Faced

I had to give fast voice response, and I am using Window Speech Recognition. Now having option available for past query to have voice response so I keep every turn in state and use Read button to allow them to speak. Another was to reply in the language asked and reply when question is asked by voice command and for that I asked in prompt to give reply in the language asked and reply in what language has user asked the question and then from frontend speak in that language.

# Note

It can reply both in Hindi and English and can speak Hindi and English. I saw in simplify app that we can ask questions in Hindi, and I made it that way.

A screenshot of a chat

AI-generated content may be incorrect.