나현웅 0724 진행사항

1. 개발 환경 준비 : 깃 init 완료
2. 오디오 전처리 전략 수립 완료
   1. Audio Web API : 게이트, 컴프레서, 리미터
   2. webRTC APM : 로우패스필터, 하이패스필터, 노이즈 필터, 에코 제거 필터, 볼륨 최적화 필터
   3. webRTC VAD : 음성 인지 자동 분할 VAD 서비스
3. 단순 오디오 녹음 테스트 코드 작성 완료
4. <!DOCTYPE html>
5. <html lang="en">
6. <head>
7. <meta charset="UTF-8">
8. <meta name="viewport" content="width=device-width, initial-scale=1.0">
9. <title>Audio Recorder</title>
10. <style>
11. /\* Add some basic styling \*/
12. body {
13. display: flex;
14. justify-content: center;
15. align-items: center;
16. height: 100vh;
17. }
18. button {
19. padding: 10px 20px;
20. font-size: 16px;
21. }
22. </style>
23. </head>
24. <body>
25. <h1>Audio Recorder</h1>
26. <button id="startButton">Start Recording</button>
27. <button id="stopButton" disabled>Stop Recording</button>
28. <a id="downloadLink" style="display:none">Download Recorded Audio</a>
29. <script>
30. // Variables for recording and audio stream
31. let mediaRecorder;
32. let recordedChunks = [];
33. // Get access to the microphone and start recording
34. async function startRecording() {
35. try {
36. const stream = await navigator.mediaDevices.getUserMedia({ audio: true });
37. mediaRecorder = new MediaRecorder(stream);
38. // Event handlers for mediaRecorder
39. mediaRecorder.ondataavailable = handleDataAvailable;
40. mediaRecorder.onstop = handleRecordingStop;
41. // Start recording
42. mediaRecorder.start();
43. // Disable/enable buttons accordingly
44. document.getElementById("startButton").disabled = true;
45. document.getElementById("stopButton").disabled = false;
46. } catch (error) {
47. console.error("Error accessing microphone:", error);
48. }
49. }
50. // Handle recorded audio data
51. function handleDataAvailable(event) {
52. if (event.data.size > 0) {
53. recordedChunks.push(event.data);
54. }
55. }
56. // Stop recording and create audio file to download
57. function stopRecording() {
58. mediaRecorder.stop();
59. }
60. // When recording is stopped, create audio file and enable buttons
61. function handleRecordingStop() {
62. const audioBlob = new Blob(recordedChunks, { type: 'audio/wav' });
63. recordedChunks = [];
64. // Create a URL for the audio file and enable download link
65. const audioURL = URL.createObjectURL(audioBlob);
66. const downloadLink = document.getElementById("downloadLink");
67. downloadLink.href = audioURL;
68. downloadLink.download = "recorded\_audio.wav";
69. downloadLink.style.display = "block";
70. // Disable/enable buttons accordingly
71. document.getElementById("startButton").disabled = false;
72. document.getElementById("stopButton").disabled = true;
73. }
74. // Event listeners for buttons
75. document.getElementById("startButton").addEventListener("click", startRecording);
76. document.getElementById("stopButton").addEventListener("click", stopRecording);
77. </script>
78. </body>
79. </html>