

```

import React, { useState, useEffect } from 'react';
import './App.css';

const App = () => {
  const [audioFiles, setAudioFiles] = useState([]);
  const [currentAudio, setCurrentAudio] = useState(null);
  const [currentIndex, setCurrentIndex] = useState(0);
  const [currentTime, setCurrentTime] = useState(0);

  useEffect(() => {
    const storedAudioFiles =
JSON.parse(localStorage.getItem('audioFiles'));
    if (storedAudioFiles) {
      setAudioFiles(storedAudioFiles);
    }

    const storedCurrentAudio =
JSON.parse(localStorage.getItem('currentAudio'));
    if (storedCurrentAudio) {
      setCurrentAudio(new
Audio(URL.createObjectURL(storedCurrentAudio)));
      setCurrentTime(storedCurrentAudio.currentTime);
    }

    const storedCurrentIndex =
JSON.parse(localStorage.getItem('currentIndex'));
    if (storedCurrentIndex !== null) {
      setCurrentIndex(storedCurrentIndex);
    }
  }, []);

  const handleFileUpload = (e) => {
    const file = e.target.files[0];
    if (file) {
      setAudioFiles([...audioFiles, file]);
    }
  };

  const handlePlay = () => {
    if (currentAudio) {
      currentAudio.play();
    }
  };

  const handlePause = () => {
    if (currentAudio) {
      currentAudio.pause();
    }
  };

  const handleNext = () => {
    if (currentIndex < audioFiles.length - 1) {
      setCurrentIndex(currentIndex + 1);
    }
  };

  const handlePrevious = () => {
    if (currentIndex > 0) {

```

```

        setCurrentIndex(currentIndex - 1);
    }
};

useEffect(() => {
    if (currentAudio) {
        currentAudio.currentTime = currentTime;
        currentAudio.addEventListener('ended', handleNext);
        currentAudio.addEventListener('timeupdate', (e) => {
            setCurrentTime(e.target.currentTime);
        });
    }

    return () => {
        if (currentAudio) {
            currentAudio.removeEventListener('ended', handleNext);
        }
    };
}, [currentAudio, currentTime]);

useEffect(() => {
    if (currentIndex !== null) {
        const file = audioFiles[currentIndex];
        if (file) {
            const audio = new Audio(URL.createObjectURL(file));
            audio.currentTime = currentTime;
            setCurrentAudio(audio);
            localStorage.setItem('currentAudio', JSON.stringify(file));
            localStorage.setItem('currentIndex',
JSON.stringify(currentIndex));
        }
    }
}, [currentIndex, audioFiles, currentTime]);

return (
    <div className="App">
        <input type="file" accept="audio/*" onChange={handleFileUpload} />
        <button onClick={handlePlay}>Play</button>
        <button onClick={handlePause}>Pause</button>
        <button onClick={handlePrevious}>Previous</button>
        <button onClick={handleNext}>Next</button>
        <div>
            {audioFiles.map((file, index) => (
                <div key={index}>
                    {file.name}
                </div>
            ))}
        </div>
        {currentAudio && (
            <audio ref={audio} => audio && audio.srcObject !==
currentAudio.srcObject && (audio.srcObject = currentAudio.srcObject)}
controls />
        )}
    </div>
);
};

export default App;

```