CODE EXERCISE
JAVASCRIPT AUDIO

1) Create an HTML file called "player.html" and add the following code:

```html
<!DOCTYPE html>
<html>
<head>
    <title>JavaScript MP3 Player</title>
</head>
<body>
    <audio id="audio-player" controls>
        <source src="" type="audio/mpeg">
        Your browser does not support the audio element.
    </audio>
    <button id="play-button">Play</button>
    <button id="pause-button">Pause</button>
    <button id="stop-button">Stop</button>
    <script src="player.js"></script>
</body>
</html>
```

This creates a basic HTML page that contains an audio element with the id "audio-player" and the controls attribute. It also contains three buttons that will be used to play, pause, and stop the audio. We also include a link to a JavaScript file called "player.js" which will contain our JavaScript code.

2) Create a file called "player.js" in the same directory as "player.html" and add the following code:

```javascript
const audioPlayer = document.getElementById("audio-player");
const playButton = document.getElementById("play-button");
const pauseButton = document.getElementById("pause-button");
const stopButton = document.getElementById("stop-button");

playButton.addEventListener("click", () => {
    audioPlayer.play();
});

pauseButton.addEventListener("click", () => {
    audioPlayer.pause();
});

stopButton.addEventListener("click", () => {
```

JAVASCRIPT AUDIO

```
    audioPlayer.pause();
    audioPlayer.currentTime = 0;
});
```

This code uses the getElementById() method to select the audio element, and the playButton, pauseButton, stopButton element. Then, it uses the addEventListener() method and arrow function to attach a "click" event listener to the buttons. When the playButton is clicked, the anonymous function passed to playButton.addEventListener() is executed, which will use the play() method of the audioPlayer object to start playing the audio. Similarly, the pause button will call the pause() method and the stop button will call the pause() method and set the currentTime property to 0.

3) Now, You need to set the source of the audio file to play. You can do this by setting the src attribute of the audioPlayer object:

```
audioPlayer.src = "path/to/your/audio.mp3";
```

You can set the path of the mp3 file you have.

4) Open "player.html" in a web browser.

5) Once the page is loaded, the Play button should work and you should be able to play, pause and stop the audio file.

To add a volume control and display the time elapsed in the audio file, you can follow these additional steps:

6) Add a volume control input to your HTML file. Here's an example of what it might look like:

```
<input type="range" id="volume-control" min="0" max="1" step="0.1" value="1">
```

This input element is a range slider with an id "volume-control", a minimum value of 0, a maximum value of 1, and a step of 0.1. The current value is set to 1.

7) In your JavaScript file, you can add an event listener to the volume control input and use the volume property of the audioPlayer object to set the volume:

const volumeControl = document.getElementById("volume-control");

CODE EXERCISE
JAVASCRIPT AUDIO

```
volumeControl.addEventListener("input", (e) => {
    audioPlayer.volume = e.target.value;
});
```

8) To display the time elapsed, you can add a div or p element with an id like "time-elapsed" to the HTML.

```
<div id="time-elapsed"></div>
```

9)In the JavaScript file, you can use the currentTime property of the audioPlayer object to get the current time of the audio file and update the content of the element with the id "time-elapsed" with the current time in the format of mm:ss:

```
const timeElapsed = document.getElementById("time-elapsed");
audioPlayer.addEventListener("timeupdate", () => {
    let minutes = Math.floor(audioPlayer.currentTime / 60);
    let seconds = Math.floor(audioPlayer.currentTime % 60);
    timeElapsed.innerHTML = `${minutes}:${seconds}`;
});
```

Finally, test your player in the browser, you should be able to adjust the volume and see the time elapsed of your audio file.