

1. a) Code snippet:

```
<form>
  <label>Enter your message:</label>
  <input required>
  <input type="submit">
</form>
```

b) Current outcome:



Currently, this code snippet from A2 displays a basic textbox with a submit button next to it which does nothing when pressed, other than outputting a “required” message if submitting an empty field.

c) Desired outcome:

A preferred outcome would be to display a notification or an alert to signal any input error or to respond to user when submit button is clicked. Also, replace the normal text input with a larger textarea with placeholder text to give the user an idea of what to write.

d) Code snippet of desired outcome:

```
<div id="message">
  <p>Leave a message below or email me at <a href="mailto:tahseen0904@gmail.com">tahseen0904@gmail.com</a></p>
  <form onsubmit="notifOnSubmit()">
    <p><label for="text-area">Enter your message:</label></p>
    <textarea type="text" id="text-area" placeholder="Give any feedback!" rows="10" cols="100"></textarea>
    <input type="submit" onclick="notifOnSubmit">
  </form>
</div>
```

```
#text-area{
  border: 5px solid;
  border-radius: 10px;
  border-color: forestgreen;
  font-size: 20px;
}

::placeholder{
  color : forestgreen;
}
```

```
function notifOnSubmit(){
  const message = document.getElementById("text-area").value;
  if(message.length < 1){
    alert("You submitted an empty message!");
  } else {
    alert("Thank you for submitting feedback");
  }
}
```

The javascript function notifOnSubmit() retrieves the input value from the user and based on the number of characters entered, displays a notification alert accordingly.

e) Desired outcome:

Leave a message below or email me at [tahseen0904@gmail.com](mailto:tahseen0904@gmail.com)

Enter your message:

Give any feedback!

Submit

Thank you for submitting feedback

OK

Did you like the website?

- ☐ Yes
- ☐ Yes

Submit

Leave a message below or email me at [tahseen0904@gmail.com](mailto:tahseen0904@gmail.com)

Enter your message:

Hello!

Submit

2. a) Code snippet:

```
<audio controls>
  <source src="Audio/IntroAudio.mp3">
</audio>
```

b) Current outcome:



Currently, the audio has no labels for the functionality and the controls are built-in, with a pause and play button and a volume controller.

c) Desired outcome:

With javascript, I can create separate buttons that are labelled and match the color scheme of the website, placing them under the main audio control for contrast. This allows user a different variation of control for the audio.

d) Code snippet for desired outcome:

```
<audio controls id="audio" src="Audio/IntroAudio.mp3"></audio>
<div>
  <button class="audio-button" id="play" onclick="document.getElementById('audio').play()">Play</button>
  <button class="audio-button" id="pause" onclick="document.getElementById('audio').pause()">Pause</button>
  <button class="audio-button" id="volUp" onclick="document.getElementById('audio').volume += 0.1">Vol +</button>
  <button class="audio-button" id="volDn" onclick="document.getElementById('audio').volume -= 0.1">Vol -</button>
</div>

.audio-button{
  background-color: royalblue;
  border : none;
  border-radius: 5px;
  color: lawngreen;
  padding: 5px 5px;
  text-align: center;
  text-decoration: bold;
  display: inline-block;
}

.audio-button:hover{
  opacity: 0.8;
}
```

This code shows the usage of buttons scripted with the ability to control an audio file with built-in functions such as play and pause. The CSS styles those buttons to fit in with the overall theme of the webpage.

e) Desired outcome:

