Learnings –

* Basic implementation of buttons, paragraphs,
* <div id="drop-zone" ondrop="handleDrop(event)" ondragover="handleDragOver(event)">

<p>Drag and drop or click to upload images:</p>

<input type="file" id="image-upload" accept="image/\*" multiple onchange="previewImages()">

</div>

This is for drag and drop functionality.

<input type="file" accept="audio/\*" id="background-music"> - To upload background music

* Flex box- to align and distribute space-

1. display: flex: This makes the body a flex container, meaning all of its child elements (such as #landing-page and any other content) become flex items.
2. flex-direction: column: This sets the main axis to run vertically, meaning the content is stacked in a column.
3. justify-content: center: This centers the flex items along the main axis (vertically in this case).
4. align-items: center: This centers the flex items along the cross axis (horizontally in this case).

* <div> element with the id="drop-zone". The key aspects of the code that enable this functionality are:
  + - ondragover Event- This event is triggered when a draggable item is dragged over the target area (#drop-zone). By default, drag-and-drop actions are not allowed in certain areas of the browser. To enable the drop functionality, the handleDragOver() function must prevent the default behavior using event.preventDefault().
    - **ondrop Event**- This event is triggered when the user releases (drops) the draggable item into the drop zone (#drop-zone). The function handleDrop(event) is called. The handleDrop() function should also prevent the default behavior and process the dropped files (in this case, the image files). Access the dropped files via event.dataTransfer.files.
* div- The primary purpose of the <div> is to group related elements together so they can be styled or manipulated together using CSS or JavaScript.