**Music Mixer Assignment**

The focus of this project is to collaboratively research, experiment with, and ultimately design and build a multimedia application using design and motion tools, as well as HTML, CSS, and JavaScript. Your team will implement drag-and-drop functionality to load audio clips, create a dynamic audio experience, trigger animations, and play a compiled audio track. Reference: [Incredibox](https://www.incredibox.com/).

Try to implement this project in **sprints** – part of an agile workflow best practice. A sprint consists of a scoped mini-project that focuses on one particular deliverable or feature. The work you do should live on a **specific branch**, and everything required to complete that deliverable must be done on that branch and merged into the **main branch** once finished. Then move on to the next sprint!

**Assignment Requirements**

**Sprint 1: Setup, Analysis, and Initial Design**

1. **Initial Repository Setup**
   * Create a GitHub repository with a detailed **README.md** (not just the project name – refer to good README examples online).
   * Create **design** and **development** branches to separate tasks and organize collaboration.
2. **Team Collaboration**
   * **Designers:**
     + Create the initial layout (HTML and CSS).
     + Work directly in the GitHub repository (use your own design branch).
   * **Developers:**
     + Research the functionality you need to implement.
     + Create JavaScript files and begin testing events (e.g., loading audio, prototyping features).
     + Update the HTML as necessary by adding classes or IDs to support interactivity.
3. **Initial Prototyping:**
   * Experiment with layouts and functionality to define how the app will look and work.

**Sprint 2: Drag-and-Drop, Audio, and Design Improvements**

1. **Core Functionality:**
   * Implement drag-and-drop functionality for loading audio files.
   * Synchronize the audio playback feature and ensure it works correctly.
2. **Refined Design:**
   * Replace "for position only" (FPO) assets with final artwork.
   * Refine the design based on feedback or new ideas.

**Sprint 3: Final Touches and Improvements**

1. **Final Fixes:**
   * Identify and fix any remaining bugs.
   * Refine final design and functionality details.
2. **Extensive Testing:**
   * Test the application on multiple devices and screen sizes to ensure responsiveness.
   * Apply necessary adjustments based on test results.

**Submission Requirements**

1. **Deadlines and Structure:**
   * The project must be submitted during the **Week 12 class time**.
   * Only the **main branch** will be graded.
2. **Repository Structure:**
   * The **main branch** should contain the final version of the project.
   * Separate branches for **design** (e.g., des.tvr.artwork) and **development** (e.g., dev.tvr.script).
   * Organized folder structure:
     + **css/**: CSS files
     + **js/**: JavaScript files
     + **images/**: Images (only SVG, PNG, or JPG files – no AI or PSD files allowed)
     + **audio/**: Audio files
     + **index.html**: Main application page
3. **GitHub Workflow:**
   * All team members must actively contribute to the repository.
   * Designers must work directly in GitHub by committing changes to the **design branch** (not via social media or other methods).

**Checkpoint (Week 6)**

* A graded checkpoint will take place in **Week 6** (see page 8 of the course material for details).

**Summary of Agile Workflow (Sprints)**

1. **Sprint 1:** Initial project setup, research, and prototyping of the design and functionality.
2. **Sprint 2:** Implementation of core functionality (drag-and-drop and audio playback) and design improvements.
3. **Sprint 3:** Final refinements, bug fixes, and extensive testing.

**Important Notes**

* Strictly follow agile workflow practices and use Git/GitHub properly.
* Non-compliance (e.g., a designer not contributing directly through GitHub) will result in significant grade penalties