**CCT College Dublin**

**Assessment Cover Page**

|  |  |
| --- | --- |
| **Module Title:** | Cross platform Development, Interactive Application Development |
| **Assessment Title:** | Music Player |
| **Lecturer Name:** | David González, Sam Weiss |
| **Student Full Name:** | Rhuan Mendanha Eli Raimundo (2023186) Henrique Queiros de Morais (2023288) |
| **Student Number:** | Rhuan Mendanha Eli Raimundo (2023186) Henrique Queiros de Morais (2023288) |
| **Assessment Due Date:** | 07/05/2025 |
| **Date of Submission:** |  |

**Declaration**

|  |
| --- |
| By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution. |

**Table of Contents**

[How to Run the Application 3](#_Toc197196169)

[Prerequisites 3](#_Toc197196170)

[Installation & Execution: 3](#_Toc197196171)

[Challenges Encountered and Solutions Implemented 3](#_Toc197196172)

[Playback Control and Sync 4](#_Toc197196173)

[UI Responsiveness 4](#_Toc197196174)

[Testing Setup 4](#_Toc197196175)

[Reflections on the Development Process 4](#_Toc197196176)

[Potential Future Improvements 4](#_Toc197196177)

# How to Run the Application

MyMusicApp is an Electron-based desktop music player. To run the application, follow these steps:

# Prerequisites

* Node.js (v18 or later)
* npm (comes with Node.js)
* Git (for cloning the repository)

# Installation & Execution:

git clone https://github.com/CCT-Dublin/music-player-ca1-the-cria.git

cd MyMusicApp

npm install

npm start

This will launch the Electron window and start the app locally. The UI will allow users to load local audio files and control playback through a simple interface.

# Challenges Encountered and Solutions Implemented

File Handling in Electron:

**Challenge:**  
Accessing local files (audio) securely in the renderer process was restricted by Electron's sandboxing policies.

**Solution:**  
Implemented a preload.js script to safely bridge between the Electron main process and the renderer using contextBridge and ipcRenderer. This allowed secure file selection without exposing Node APIs to the frontend.

# Playback Control and Sync

**Challenge:**  
Ensuring the play/pause state and progress bar stayed synchronized between UI and the audio element.

**Solution:**  
Encapsulated playback logic in player.js and used event listeners (timeupdate, ended, etc.) to constantly update UI elements via renderer.js.

# UI Responsiveness

**Challenge:**  
Making the UI responsive across screen sizes and DPI settings.

**Solution:**  
Used flexible layouts with CSS Grid and Flexbox, ensured scalable vector icons, and tested on different resolutions.

# Testing Setup

**Challenge:**  
Running automated UI tests in an Electron environment was initially difficult.

**Solution:**  
Used Spectron for E2E testing and Jest for unit testing of core logic, separating DOM manipulation from logic-heavy code to ease testability.

# Reflections on the Development Process

* **Team Coordination:**  
  Effective division of labour into frontend, backend (Electron setup), and testing helped parallelize development and stay on schedule.
* **Learning Curve:**  
  Setting up Electron and understanding IPC (Inter-Process Communication) required a deep dive into desktop security models and event-driven architecture.
* **Design Iterations:**  
  Initial UI drafts were simplified based on usability testing, improving the experience for non-technical users.
* **Documentation:**  
  Maintaining a structured folder (docs/, tests/, reflections/) helped with clarity and academic deliverables.

Overall, the project strengthened our skills in cross-platform development, testing, and teamwork.

# Potential Future Improvements

* **Playlist Support:**  
  Allow users to queue and manage playlists, rather than single file playback.
* **Drag-and-Drop Interface:**  
  Add intuitive drag-and-drop for loading audio files directly into the player.
* **Settings Menu:**  
  Let users customize themes, audio output device, and shortcuts.
* **Improved Testing Coverage:**  
  Expand E2E tests to cover window resize events, file error handling, and multi-platform compatibility.
* **Packaging & Distribution:**  
  Add one-click installers for Windows/macOS/Linux using electron-builder.
* **ID3 Metadata Display:**  
  Display track metadata (artist, album, cover) using an ID3 parsing library.