ASSIGNMENT 1

RICH MEDIA DEVELOPMENT (MAD 110)

SUBMITTED TO – SHAHID AHMAD

SUBMITTED BY-

KIRAN BHALALA (A00182454)

PRIYA BATRA (A00184500)

SIJAN NEUPANE (A00185437)

DOCUMENTATION

Title: Music Player Web Application Documentation

Introduction: To give consumers a rich music playing experience, the Music Player Web Application is a straightforward yet sophisticated solution. The programme, which was created with HTML, CSS, and JavaScript, has a visually appealing interface that skillfully combines aesthetics and usefulness. The features that have been put into practice, design choices, difficulties encountered during development, and an overview of the codebase will all be covered in this documentation.

Features Implemented:

1. **User Interface:**
- HTML and CSS are used to create a responsive layout for the user interface.
- The Music Player is presented with an eye-catching welcome screen that features a floating animation and a gradient background.
2. **Music Player Features:**
- Users can pause, play, and go to the previous or next track with this programme.
- Track details are dynamically shown, including artist and name.
- User control is provided by a slider for panning through the track and changing the volume.
- The addition of random play functionality gives the user experience a lively and captivating touch.
3. **Playlist and Music Selection:**
- An album image and details are presented alongside a list of tracks.
- From the playlist, users can choose which individual song to play.
- The playlist's currently playing track is indicated.
4. **Visual Enhancements**:
- An eye-catching experience is offered by the revolving album art and animated equaliser that operate while the music is playing.
- With every track, the background colour varies dynamically, resulting in a lively environment.
5. **Design Responsiveness:**

- The design is flexible, allowing for a smooth experience across a range of devices and screen sizes.
- For displays that are narrower than 768 pixels, a mobile-friendly layout is used.

6. **Code Modularity:**

- To make the code more readable and maintainable, it is divided into distinct HTML, CSS, and JavaScript files.
- Effective integration of external libraries (Font Awesome) improves design components.

7. **Controls for Play, Pause, and Navigation:**

- The application features a central play button that users can utilise to play or pause the music.
- The buttons for the next and previous tracks make it simple to go through the playlist.

8. **Information on Dynamic Tracks:**

- Name and artist of the track that is now playing displayed in real time.
- The album art's visual depiction improves the user experience.

9. **Volume and Seek Sliders:**

- The seek slider helps the user to adjust the playback position.
- The audio volume can be changed using the volume slider.

10. **Functionality of Random and Repeat:**

- Users have the option to switch between random play mode and random track shuffle.
- The current music can be continuously played back thanks to the repeat option.

Design decisions:

1. **Visual Aesthetics:**

- An eye-catching and captivating user interface is enhanced by the gradient background and colour scheme selection.
- Playing music becomes more immersive when the background colour changes and album graphics are combined.

2. **User Interaction:**

- By giving consumers recognisable controls, the play, pause, and skip buttons improve usability.
- A creative touch is added by the welcome screen's floating animation.

3. **Code Structure:**
- To facilitate readability and ease of maintenance, the code is divided into distinct functions for each functionality.
- The entire visual experience is improved by the usage of keyframes and animations.

4. **Animation:**
- An interface that is both active and aesthetically pleasing is enhanced by floating animations and equalisation effects.

5. **Designing with responsiveness:**
- Because to the responsive design, using it on different devices is effortless.

6. **Utilising Outside Libraries:**
- The player's aesthetic appeal and functionality are enhanced with FontAwesome and additional CSS libraries.

Challenges Faced:

1. **Cross-Browser Integrated:**
- Thorough testing and tweaking were necessary to guarantee consistent behaviour across various web browsers.

2. **Audio Playback and Loading:**
- There were difficulties with controlling the asynchronous loading and playing of audio files, but these were resolved by extensive testing and optimisation.

3. **Audio Synchronisation:**
- It was difficult to ensure precise synchronisation of the audio track with the slider and timer; much testing and adjustment were necessary.

4. **Dynamic backdrop Colour:**
- Finding a balance between readability and a seamless backdrop colour transition for every track was a challenge.

5. **Mobile responsiveness:**
- To preserve a positive user experience, the layout had to be carefully adjusted for smaller devices without sacrificing functionality.

6. **Random Play Logic:**
- Careful algorithm design was required to ensure a balanced distribution of music without repetition and to implement random play capability.

To sum up, the Music Player Web Application offers an engaging music playback experience through a tasteful combination of functionality and design. An interesting interface is influenced by elements such as dynamic track information, user control sliders, and aesthetic improvements. The user experience was optimised by design choices like the use of external libraries and a responsive layout. Notwithstanding difficulties in guaranteeing audio playback and cross-browser compatibility, the development approach produced a feature-rich programme that successfully achieves its goal. The documentation offers a glimpse into the decisions, efforts, and processes involved in developing this web application for a music player.