

PROJECT ABSTRACT

Music Player Web App with Playlist Management

A web music player with a now-playing view, animated visualizer, up-next queue, playlist management, a song library and a persistent bottom player bar. Delivered built-to-order with working demo, source code, report, PPT and viva kit.

01. Title

Music Player Web App with Playlist Management

02. Introduction / Background

Streaming apps set a high bar for media UI: now-playing screens, queues, playlists and persistent mini-players that survive navigation. This project builds Waveform, a web music player with an album-art now-playing view, an animated audio visualizer, transport controls with play/pause toggle, an up-next queue, six curated playlists with cover art, a full song library with genre and duration metadata, playlist creation, and a fixed bottom player bar. Eight realistic tracks with artists, genres and durations populate every list, and the visualizer animates in sync with the play state to feel alive.

03. Problem Statement

Media-player demos often fake playback with a static progress bar. This project implements the full player chrome — stateful play/pause, queue, playlists, library, persistent bar — over a realistic track catalog, so the student demonstrates genuine media-app architecture rather than a decorative mockup.

04. Objectives

- Build a now-playing view with album art, visualizer and transport controls.
- Manage an up-next queue and a filterable song library.
- Support playlist browsing and creation with cover art.
- Keep a persistent bottom player bar across all views.
- Use realistic track metadata throughout the catalog.

05. Proposed Methodology

The player is structured in three parts. First, the catalog model: tracks with title, artist, genre, duration and liked state; playlists with covers and song counts. Second, the playback model: current track, playing/paused state and progress, driving the

visualizer animation, transport buttons and both player bars. Third, the views: Now Playing, Playlists (grid plus creation form plus recently played) and Library, all rendered from the same catalog.

06. System Architecture / Working

- Catalog layer: track and playlist objects with full metadata.
- Playback layer: play-state machine driving visualizer, controls and progress.
- Now Playing view: art, visualizer canvas, controls, queue.
- Playlists view: cover grid, creation form, recently played.
- Library view: full track table; bottom bar persists across views.

07. Software Requirements

- Any modern web browser; no dependencies.
- Static hosting for deployment.
- Audio files (MP3) if the student wires real playback via the HTML5 audio element.
- Text editor for extension.

08. Expected Outcomes

- A working player UI with stateful play/pause and progress.
- Queue, playlists and library views over a realistic 8-track catalog.
- Animated visualizer tied to playback state.
- Single-file source ready for real audio wiring.

09. Applications

- Front-end for college radio or event music portals.
- Teaching example for media-app state management.
- Portfolio piece demonstrating rich interactive UI.
- Base for a real player with streaming or local files.

10. Future Scope

- Real audio playback with the Web Audio API and analyzers.
- Search, filters and smart playlists.
- User accounts with synced libraries.
- Offline caching via service workers.