MINI PROJECT REPORT

ON

MUSIC PLAYER

Submitted by

**NAME: VARSHA V S**

**ROLL NO: NC.SC.U4CSE24049**

For

23CSE113-User Interface Design

II Semester

B.Tech. CSE

School of Computing

Amrita Vishwa Vidyapeetham, Nagercoil

Index

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Chapter Name** | **Page No.** |
|  | Introduction |  |
|  | Problem Statement |  |
|  | Objectives |  |
|  | System Design |  |
|  | Modules of project |  |
|  | Code |  |
|  | Output Screenshots |  |
|  | Application of the project |  |
|  | Limitations of the project |  |
|  | Bibliography(5 lines) |  |
|  | GitHub link of the project |  |

**1. Introduction**

This web-based music player is designed as a frontend project to demonstrate the core concepts of web development. It allows users to play audio files, navigate through songs, and interact with a sleek, modern interface similar to commercial music apps.

**2. Problem Statement**

With growing demand for streaming services, many lightweight, offline solutions are overlooked. This project addresses that gap by creating a browser-based music player that functions without relying on internet or backend services, providing users with a fast and accessible music experience.

**3. Objectives**

Build a responsive and attractive UI using HTML and CSS.

Implement music control functionality using JavaScript.

Support features like marking songs as favorites and downloading tracks.

Offer a smooth user experience that works offline.

**4. System Design**

Frontend Only: Built using HTML5, CSS3, and vanilla JavaScript.

Audio Interface: Uses the HTML5 <audio> tag for playback.

File Handling: Assumes songs are stored in a project directory.

UI Design: Mimics dark-themed streaming interfaces, including album art, song details, and buttons for interaction.

**5. Modules of the Project**

UI Display Module: Manages the card layout, song info, and album art.

Audio Control Module: Handles playing, pausing, skipping, and progress tracking.

Interaction Module: Processes user input from buttons (favorite, download, etc.).

Playback Logic: Syncs song duration and updates progress bars in real-time.

**6.Code**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8" />**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

**<title>Now Playing</title>**

**<link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600&display=swap" rel="stylesheet" />**

**<style>**

**\* {**

**margin: 0;**

**padding: 0;**

**font-family: 'Inter', sans-serif;**

**box-sizing: border-box;**

**}**

**body {**

**background-color: #121212;**

**color: white;**

**display: flex;**

**flex-direction: column;**

**align-items: center;**

**justify-content: center;**

**min-height: 100vh;**

**padding: 20px;**

**position: relative;**

**overflow: hidden;**

**}**

**body::before {**

**content: '';**

**position: absolute;**

**top: 0;**

**left: 0;**

**width: 100%;**

**height: 100%;**

**background-image: url('https://images.unsplash.com/photo-1511671782779-c97d3d27a1d4?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fit=crop&w=2070&q=80');**

**background-size: cover;**

**background-position: center;**

**opacity: 0.3;**

**z-index: -1;**

**}**

**.player-container {**

**background: rgba(30, 30, 30, 0.9);**

**padding: 30px;**

**border-radius: 20px;**

**box-shadow: 0 0 20px rgba(0, 0, 0, 0.8);**

**max-width: 400px;**

**width: 100%;**

**text-align: center;**

**-webkit-backdrop-filter: blur(10px);**

**backdrop-filter: blur(10px);**

**border: 1px solid rgba(255, 255, 255, 0.1);**

**}**

**.cover img {**

**width: 100%;**

**max-height: 300px;**

**object-fit: cover;**

**border-radius: 15px;**

**margin-bottom: 20px;**

**box-shadow: 0 10px 20px rgba(0, 0, 0, 0.3);**

**}**

**.song-title {**

**font-size: 24px;**

**font-weight: 600;**

**margin-bottom: 5px;**

**}**

**.song-artist {**

**font-size: 16px;**

**color: #aaa;**

**margin-bottom: 20px;**

**}**

**audio {**

**width: 100%;**

**margin: 20px 0;**

**filter: sepia(20%) saturate(70%) grayscale(1) contrast(99%) invert(12%);**

**}**

**audio::-webkit-media-controls-panel {**

**background-color: #333;**

**}**

**.buttons {**

**display: flex;**

**justify-content: space-around;**

**margin-top: 20px;**

**gap: 10px;**

**}**

**.buttons button {**

**background-color: #1db954;**

**color: white;**

**border: none;**

**padding: 10px 20px;**

**border-radius: 50px;**

**cursor: pointer;**

**font-weight: bold;**

**transition: all 0.3s;**

**flex: 1;**

**}**

**.buttons button:hover {**

**background-color: #1ed760;**

**transform: scale(1.05);**

**}**

**.buttons button:active {**

**transform: scale(0.98);**

**}**

**.nav-buttons {**

**display: flex;**

**justify-content: space-between;**

**margin-top: 20px;**

**gap: 10px;**

**}**

**.nav-buttons button {**

**background-color: rgba(51, 51, 51, 0.7);**

**color: white;**

**border: none;**

**padding: 10px 15px;**

**cursor: pointer;**

**font-weight: bold;**

**border-radius: 50px;**

**transition: all 0.3s;**

**flex: 1;**

**}**

**.nav-buttons button:hover {**

**background-color: rgba(85, 85, 85, 0.7);**

**transform: scale(1.05);**

**}**

**.nav-buttons button:active {**

**transform: scale(0.98);**

**}**

**.progress-container {**

**width: 100%;**

**height: 6px;**

**background: #333;**

**border-radius: 5px;**

**margin: 15px 0;**

**cursor: pointer;**

**}**

**.progress {**

**height: 100%;**

**background: #1db954;**

**border-radius: 5px;**

**width: 0%;**

**transition: width 0.1s linear;**

**}**

**@media (max-width: 480px) {**

**.player-container {**

**padding: 20px;**

**}**

**.song-title {**

**font-size: 20px;**

**}**

**.buttons button,**

**.nav-buttons button {**

**padding: 8px 12px;**

**font-size: 14px;**

**}**

**}**

**</style>**

**</head>**

**<body>**

**<div class="player-container">**

**<div class="cover">**

**<img id="cover" src="" alt="Album Cover">**

**</div>**

**<div class="song-title" id="title"></div>**

**<div class="song-artist" id="artist"></div>**

**<div class="progress-container" onclick="setProgress(event)">**

**<div class="progress" id="progress"></div>**

**</div>**

**<audio id="audio" controls autoplay></audio>**

**<div class="buttons">**

**<button onclick="addToFavorites()">❤ Favorite</button>**

**<button onclick="downloadSong()">⬇ Download</button>**

**</div>**

**<div class="nav-buttons">**

**<button onclick="previousSong()">⏮ Prev</button>**

**<button onclick="nextSong()">Next ⏭</button>**

**</div>**

**</div>**

**<script>**

**const songs = [**

**{**

**title: "Perfect",**

**artist: "Ed Sheeran",**

**file: "https://www.soundhelix.com/examples/mp3/SoundHelix-Song-1.mp3",**

**cover: "https://i.scdn.co/image/ab67616d0000b273ba5db46f4b838ef6027e6f96"**

**},**

**{**

**title: "Shape of You",**

**artist: "Ed Sheeran",**

**file: "https://www.soundhelix.com/examples/mp3/SoundHelix-Song-2.mp3",**

**cover: "https://images.genius.com/86dcab92e27599f8556b5b8f21932449.1000x1000x1.png"**

**},**

**{**

**title: "Night Changes",**

**artist: "One Direction",**

**file: "https://www.soundhelix.com/examples/mp3/SoundHelix-Song-7.mp3",**

**cover: "https://i.scdn.co/image/ab67616d0000b273d71fd77b89d08bc1bda219c7"**

**},**

**{**

**title: "Poker Face",**

**artist: "Lady Gaga",**

**file: "https://www.soundhelix.com/examples/mp3/SoundHelix-Song-16.mp3",**

**cover: "https://i.scdn.co/image/ab67616d0000b2738265a736a1eb838ad5a0b921"**

**},**

**{**

**title: "Heartbreak Anniversary",**

**artist: "Giveon",**

**file: "https://www.soundhelix.com/examples/mp3/SoundHelix-Song-17.mp3",**

**cover: "https://th.bing.com/th/id/OIP.2tdxFSbYEp4m6gvaiq9YbAHaHa?w=184&h=184&c=7&r=0&o=5&dpr=1.3&pid=1.7"**

**}**

**];**

**let currentSongIndex = 0;**

**const audio = document.getElementById('audio');**

**const progress = document.getElementById('progress');**

**function loadSong(index) {**

**const song = songs[index];**

**document.getElementById("title").textContent = song.title;**

**document.getElementById("artist").textContent = song.artist;**

**document.getElementById("cover").src = song.cover;**

**audio.src = song.file;**

**audio.play();**

**}**

**function nextSong() {**

**currentSongIndex = (currentSongIndex + 1) % songs.length;**

**loadSong(currentSongIndex);**

**}**

**function previousSong() {**

**currentSongIndex = (currentSongIndex - 1 + songs.length) % songs.length;**

**loadSong(currentSongIndex);**

**}**

**function addToFavorites() {**

**alert("Added to Favorites!");**

**}**

**function downloadSong() {**

**const song = songs[currentSongIndex];**

**const link = document.createElement('a');**

**link.href = song.file;**

**link.download = ${song.title} - ${song.artist}.mp3;**

**link.click();**

**}**

**function updateProgress() {**

**const percent = (audio.currentTime / audio.duration) \* 100;**

**progress.style.width = percent + '%';**

**}**

**function setProgress(e) {**

**const width = e.currentTarget.clientWidth;**

**const clickX = e.offsetX;**

**const duration = audio.duration;**

**audio.currentTime = (clickX / width) \* duration;**

**}**

**audio.addEventListener('timeupdate', updateProgress);**

**audio.addEventListener('ended', nextSong);**

**window.onload = () => loadSong(currentSongIndex);**

**</script>**

**</body>**

**</html>**

**7.Output screenshot**

|  |
| --- |
| **C:\Users\DeLL\Downloads\WhatsApp Image 2025-04-24 at 22.22.41.jpeg** |
| **C:\Users\DeLL\Downloads\WhatsApp Image 2025-04-24 at 22.22.11.jpeg** |
| **C:\Users\DeLL\Downloads\WhatsApp Image 2025-04-24 at 22.22.27.jpeg** |

**8. Application of the Project (Extended)**

Educational Use: Ideal for learning and demonstrating frontend skills like DOM manipulation, event handling, and CSS styling.

Offline Music Player: Functions as a local music player, especially useful in environments with no internet access.

Portfolio Project: Great addition to any developer’s portfolio to showcase frontend capabilities and attention to UI/UX.

Base for Full Stack Projects: Can be expanded into a full-stack music streaming service by adding a backend, user login, and database.

Customization Sandbox: Users can customize album covers, song lists, or even add animations, making it perfect for experimenting with web design trends.

Event or Kiosk Use: Can be used in exhibitions or local events to play themed music without relying on internet-based streaming.

**9. Limitations of the Project**

Only works with locally stored audio files; no streaming support.

Does not support dynamic playlist creation or user-specific libraries.

No backend, so favorites/downloads aren’t stored permanently.

Limited audio formats supported based on browser compatibility.

Lacks search and shuffle functionalities.

**10. Bibliography (5 lines)**

freeCodeCamp – for introductory web development tutorials.

MDN Web Docs – for HTML5 Audio API documentation.

CSS-Tricks – for styling and responsive UI tips.

Stack Overflow – for troubleshooting bugs and logic issues.

GitHub – for studying similar open-source music app projects.

**11. GitHub Link of the Project**

🔗 GitHub Repository (Add your actual link here once uploaded)