

ỨNG DỤNG CHƠI NHẠC ONLINE (MUSIC PLAYER)

1. Giới thiệu

Ứng dụng Chơi Nhạc Online là một ứng dụng web giúp người dùng phát nhạc từ một danh sách bài hát có sẵn. Người dùng có thể phát, tạm dừng, chuyển bài hát, điều chỉnh âm lượng và tạo danh sách phát.

2. Mục tiêu

- Giúp sinh viên làm quen với lập trình JavaScript tương tác (DOM, Audio API).
- Phát triển kỹ năng thiết kế giao diện với HTML, CSS.
- Xây dựng một ứng dụng web hoàn chỉnh với tính năng đa phương tiện.
- Hoàn thành ứng dụng trong vài ngày.

3. Yêu cầu chức năng

- Hiển thị danh sách bài hát.
- Phát, tạm dừng và chuyển bài hát.
- Hiển thị tên bài hát đang phát.
- Điều chỉnh âm lượng.
- Hiển thị thanh tiến trình của bài hát.
- Cho phép tạo danh sách phát (playlist).

4. Yêu cầu kỹ thuật

- Frontend:
 - HTML, CSS (Bootstrap cho giao diện đẹp).
 - JavaScript (DOM Manipulation, Audio API).

- Backend:
 - Không yêu cầu (chạy hoàn toàn trên frontend).

5. Phân chia công việc và deadline

- **Ngày 1:** Thiết kế giao diện với HTML, CSS (Bootstrap).
- **Ngày 2:** Viết code JavaScript để phát nhạc và điều khiển trình phát nhạc.
- **Ngày 3:** Hoàn thiện tính năng danh sách phát (playlist) và kiểm tra.

6. Tiêu chí đánh giá

- Giao diện đẹp, dễ sử dụng.
- Chức năng phát nhạc hoạt động ổn định.
- Danh sách phát có thể quản lý dễ dàng.
- Mã nguồn rõ ràng, có chú thích.

7. Hướng dẫn từng bước (Step-by-Step)

Bước 1: Chuẩn bị môi trường

1. Tạo thư mục dự án:

```
mkdir music-player-app
```

```
cd music-player-app
```

2. Tạo cấu trúc thư mục:

```
music-player-app/
```

```
    └── index.html
```

```
    └── style.css
```

```
    └── script.js
```

```
└── music/
    ├── song1.mp3
    ├── song2.mp3
    └── song3.mp3
```

3. Thêm các bài hát vào thư mục music/.

- o Đảm bảo các file nhạc là định dạng .mp3.

Bước 2: Thiết kế giao diện người dùng (HTML + CSS)

1. Thiết kế giao diện HTML (index.html)

```
<!DOCTYPE html>

<html>
  <head>
    <title>Music Player</title>
    <link rel="stylesheet" href="style.css">
    <link rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css">
  </head>
  <body>
    <div class="container mt-5 text-center">
      <h2>♪ Music Player ♪</h2>
      <div id="player">
        <div id="current-song">Select a song</div>
        <audio id="audio" src=""></audio>
        <div class="controls">
          <button id="prev" class="btn btn-primary">◀</button>
```

```
<button id="play-pause" class="btn btn-success"> </button>

<button id="next" class="btn btn-primary"> </button>

</div>

<div class="mt-3">
  <input type="range" id="progress" value="0" max="100" class="form-range">
</div>

<div class="mt-3">
  <label>Volume:</label>
  <input type="range" id="volume" value="100" max="100" class="form-range">
</div>
</div>

<h3 class="mt-4">Playlist</h3>
<ul id="playlist" class="list-group">
  <li class="list-group-item" onclick="selectSong(0)">Song 1</li>
  <li class="list-group-item" onclick="selectSong(1)">Song 2</li>
  <li class="list-group-item" onclick="selectSong(2)">Song 3</li>
</ul>
</div>

<script src="script.js"></script>
</body>
</html>
```

2. Thiết kế giao diện CSS (style.css)

```
body {  
background-color: #f0f0f5;  
font-family: Arial, sans-serif;  
}
```

```
#player {  
background-color: #fff;  
padding: 20px;  
border-radius: 10px;  
box-shadow: 0 0 15px rgba(0, 0, 0, 0.1);  
}
```

```
.controls button {  
    margin: 0 5px;  
    font-size: 24px;  
}
```

```
#playlist {  
    margin-top: 15px;  
}  
}
```

```
#playlist li {  
    cursor: pointer;
```

```
}
```

```
#playlist li:hover {  
    background-color: #e0e0e5;  
}
```

Bước 3: Viết logic JavaScript (script.js)

```
const audio = document.getElementById('audio');  
  
const playPauseBtn = document.getElementById('play-pause');  
  
const progressBar = document.getElementById('progress');  
  
const volumeBar = document.getElementById('volume');  
  
const currentSong = document.getElementById('current-song');  
  
const playlist = [  
    'music/song1.mp3',  
    'music/song2.mp3',  
    'music/song3.mp3'  
];  
  
let currentIndex = 0;  
  
  
// Phát nhạc  
  
function playSong() {  
    audio.src = playlist[currentIndex];  
  
    currentSong.textContent = "Playing: " + playlist[currentIndex];  
  
    audio.play();
```

```
playPauseBtn.textContent = "⏸";  
}  
  
// Dừng nhạc  
function pauseSong() {  
    audio.pause();  
    playPauseBtn.textContent = "▶";  
}  
  
// Bật/tắt nhạc  
playPauseBtn.addEventListener("click", () => {  
    audio.paused ? playSong() : pauseSong();  
});  
  
// Chuyển bài hát  
document.getElementById("next").onclick = () => {  
    currentIndex = (currentIndex + 1) % playlist.length;  
    playSong();  
};  
document.getElementById("prev").onclick = () => {  
    currentIndex = (currentIndex - 1 + playlist.length) % playlist.length;  
    playSong();  
};
```

```
// Chọn bài hát trong playlist  
  
function selectSong(index) {  
    currentIndex = index;  
  
    playSong();  
  
}  
  
  
// Điều chỉnh âm lượng  
  
volumeBar.addEventListener("input", () => {  
    audio.volume = volumeBar.value / 100;  
  
});  
  
  
// Cập nhật tiến trình  
  
audio.addEventListener("timeupdate", () => {  
    progressBar.value = (audio.currentTime / audio.duration) * 100;  
  
});  
  
progressBar.addEventListener("input", () => {  
    audio.currentTime = (progressBar.value / 100) * audio.duration;  
  
});
```

Bước 4: Kiểm tra và hoàn thiện

1. Mở file index.html trong trình duyệt.
2. Chọn bài hát từ danh sách và thử phát nhạc.
3. Kiểm tra các chức năng: phát, tạm dừng, chuyển bài, điều chỉnh âm lượng.

8. Các xử lý JavaScript nâng cao (tùy chọn 3 chức năng):

- **Hiệu Ứng Visualizer (Thanh sóng nhạc động)**
 - Sử dụng API Web Audio của JavaScript để tạo hiệu ứng trực quan hóa âm thanh (visualizer).
 - Tạo các thanh sóng hoặc hiệu ứng dạng hình tròn chuyển động theo nhịp bài hát.
- **Chế độ Hẹn Giờ Tắt Nhạc (Sleep Timer)**
 - Thêm một nút để người dùng thiết lập thời gian (5, 10, 30 phút) để tự động dừng phát nhạc.
 - Sử dụng setTimeout để dừng nhạc khi hết thời gian.
- **Lưu Playlist Của Người Dùng (Local Storage)**
 - Lưu danh sách phát của người dùng vào localStorage để khi người dùng mở lại ứng dụng, danh sách phát của họ sẽ được tự động khôi phục.
 - Kết hợp với tính năng thêm bài hát từ URL.
- **Thêm Playlist Nhiều Danh Mục**
 - Cho phép người dùng tạo nhiều playlist khác nhau, chuyển đổi giữa các playlist.
 - Ví dụ: Playlist Pop, Playlist Rock, Playlist User's Favorites.
- **Tính Năng Tua Nhanh & Tua Lại (Seek Forward/Backward)**
 - Thêm nút tua nhanh (10 giây) và tua lại (10 giây).
 - Sử dụng phương thức audio.currentTime để điều khiển thời gian phát.
- **Chỉnh Tốc Độ Phát (Playback Speed)**
 - Cho phép người dùng chọn tốc độ phát (0.5x, 1x, 1.5x, 2x).
 - Sử dụng thuộc tính audio.playbackRate để điều chỉnh.
- **Thông Báo Đang Phát Nhạc (Notification API)**

- Khi bài hát chuyển sang bài mới, hiện thông báo trên trình duyệt với tên bài hát mới.
- Sử dụng Notification API của trình duyệt.
- **Hiển Thị Lời Bài Hát (Lyrics)**
 - Cho phép người dùng nhập lời bài hát cho từng bài.
 - Khi phát bài hát, lời bài hát sẽ tự động cuộn theo thời gian (Lyric Sync).
- **Giao Diện Tùy Chỉnh (Dark Mode/Light Mode)**
 - Thêm nút chuyển đổi giữa chế độ nền tối (Dark Mode) và nền sáng (Light Mode).
 - Lưu lựa chọn của người dùng vào localStorage.
- **Chia Sẻ Bài Hát Qua Link (Shareable URL)**
 - Khi người dùng đang nghe một bài hát, họ có thể sao chép URL của bài hát đó để chia sẻ với người khác.
 - Sử dụng window.location.href để tạo URL chứa thông tin bài hát.