Mood Music Player

By

Roll-No:100522733011,Avanish Cowkur

Roll-No:100522733012,Abhishek Dhaladhuli

Overview

A mood music player is a type of music player or app designed to curate playlists and tracks based on the listener's current emotional state or desired mood. These players typically use a variety of features, such as mood detection, user preferences, or pre-set genres associated with different emotions. The idea is to enhance the listener's experience by matching music to their current feelings or environment.

The provided HTML, CSS, and JavaScript code outlines a mood-based music player web application called "Dreamland Music Player." It allows users to play music based on their current mood (e.g., happy, sad, relaxed, energetic, etc.) through an interactive and visually appealing interface. Additionally, the app includes dynamic features like theme toggling (light/dark/system default), audio playback controls, and playback speed adjustments.

Key Features:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Dreamland Music Player</title>
  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;600&display=swap"
rel="stylesheet">
  <style>
    :root {
      --bg-gradient: linear-gradient(to bottom, #4facfe, #00f2fe);
      --text-color: #fff;
      --card-gradient: linear-gradient(135deg, #ff9a9e, #fad0c4);
    }

    [data-theme="light"] {
      --bg-gradient: linear-gradient(to bottom, #ffffff, #e6e6e6);
      --text-color: #000;
      --card-gradient: linear-gradient(135deg, #f9f9f9, #ececec);
    }

    [data-theme="dark"] {
      --bg-gradient: linear-gradient(to bottom, #1a1a1a, #333333);
      --text-color: #fff;
      --card-gradient: linear-gradient(135deg, #444444, #666666);
    }

    body {
      margin: 0;
      font-family: 'Poppins', sans-serif;
      background: var(--bg-gradient);
```

```
    overflow: hidden;
    position: relative;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    min-height: 100vh;
    color: var(--text-color);
    transition: background 0.3s, color 0.3s;
}

h1 {
    color: var(--text-color);
    text-shadow: 0 0 30px rgba(255, 255, 255, 0.8), 0 0 50px rgba(0, 128, 255, 0.7);
    font-size: 3em;
    animation: pulse 2s infinite;
    margin-bottom: 40px;
}

.mood-container {
    display: flex;
    gap: 20px;
    justify-content: center;
    flex-wrap: wrap;
}

.mood-card {
    flex: 1;
    min-width: 150px;
    max-width: 200px;
    background: var(--card-gradient);
    border-radius: 15px;
    padding: 20px;
    text-align: center;
    color: var(--text-color);
    cursor: pointer;
    transition: transform 0.4s, box-shadow 0.4s, background 0.3s;
    box-shadow: 0 10px 15px rgba(0, 0, 0, 0.2);
}

.mood-card:hover {
    transform: scale(1.1);
    box-shadow: 0 15px 30px rgba(0, 0, 0, 0.4);
}

.toggle-container {
    position: absolute;
    top: 20px;
    right: 20px;
    display: flex;
    align-items: center;
```

```
    gap: 10px;
}

select {
    background-color: #2d2d2d;
    color: #fff;
    border: 2px solid #444;
    border-radius: 10px;
    padding: 10px;
    font-size: 1em;
    cursor: pointer;
    transition: background-color 0.3s, border-color 0.3s;
}

select:focus {
    background-color: #444444;
    border-color: #00f2fe;
}

.note {
    position: absolute;
    font-size: 2rem;
    color: rgba(255, 255, 255, 0.7);
    animation: float 10s infinite;
    opacity: 0;
}

.note:nth-child(1) { top: 10%; left: 20%; animation-delay: 0s; }
.note:nth-child(2) { top: 30%; left: 50%; animation-delay: 2s; }
.note:nth-child(3) { top: 60%; left: 80%; animation-delay: 4s; }
.note:nth-child(4) { top: 50%; left: 30%; animation-delay: 6s; }
.note:nth-child(5) { top: 70%; left: 60%; animation-delay: 8s; }

@keyframes float {
    0% {
        transform: translateY(0) scale(1);
        opacity: 1;
    }
    50% {
        opacity: 0.7;
    }
    100% {
        transform: translateY(-200px) scale(0.8);
        opacity: 0;
    }
}

@keyframes pulse {
    0% { text-shadow: 0 0 20px #fff, 0 0 40px #00f2fe; }
    100% { text-shadow: 0 0 40px #00f2fe, 0 0 60px #4facfe; }
}
```

```
.audio-controls {
    display: none;
    flex-direction: column;
    align-items: center;
    margin-top: 40px;
    padding: 30px;
    background: rgba(255, 255, 255, 0.2);
    border-radius: 20px;
    box-shadow: 0 10px 30px rgba(0, 0, 0, 0.4);
    width: 90%;
    backdrop-filter: blur(15px);
}

.audio-controls.show {
    display: flex;
}

audio {
    width: 100%;
    height: 40px;
    margin-top: 20px;
    border-radius: 15px;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);
}

audio::-webkit-media-controls-timeline {
    background-color: white !important;
}

.custom-buttons {
    margin-top: 20px;
    display: flex;
    gap: 10px;
}

button {
    background: linear-gradient(135deg, #ffecd2, #fcb69f);
    border: none;
    border-radius: 10px;
    color: black; /* Set text color to black */
    font-size: 1em;
    padding: 10px 20px;
    cursor: pointer;
    transition: background 0.3s, transform 0.3s;
}

button:hover {
    background: linear-gradient(135deg, #fad0c4, #ff9a9e);
    transform: scale(1.1);
}
```

```
    footer {
      margin-top: 30px;
      color: black; /* Set footer text to black */
      font-size: 1em;
    }
  </style>
</head>
<body data-theme="system">
  <h1>Dreamland Music Player</h1>

  <!-- Floating notes -->
  <div class="note">□</div>
  <div class="note">□</div>
  <div class="note">□</div>
  <div class="note">□</div>
  <div class="note">□</div>

  <!-- Toggle mode -->
  <div class="toggle-container">
    <label for="themeToggle">Mode:</label>
    <select id="themeToggle" onchange="changeTheme(this.value)">
      <option value="system" selected>System Default</option>
      <option value="light">Light</option>
      <option value="dark">Dark</option>
    </select>
  </div>

  <div class="mood-container">
    <div class="mood-card" onclick="playMood('happy')">
      <h2><span>□</span>Happy</h2>
    </div>
    <div class="mood-card" onclick="playMood('sad')">
      <h2><span>□</span>Sad</h2>
    </div>
    <div class="mood-card" onclick="playMood('relaxed')">
      <h2><span>□</span>Relaxed</h2>
    </div>
    <div class="mood-card" onclick="playMood('focused')">
      <h2><span>□</span>Focused</h2>
    </div>
    <div class="mood-card" onclick="playMood('energetic')">
      <h2><span>□</span>Energetic</h2>
    </div>
    <div class="mood-card" onclick="playMood('devotional')">
      <h2><span>□</span>Devotional</h2>
    </div>
  </div>

  <div class="audio-controls" id="audioControls">
    <audio id="audioPlayer" controls loop></audio>
```

```
    <div class="custom-buttons">
      <button onclick="playAudio()">Play</button>
      <button onclick="pauseAudio()">Pause</button>
      <select id="speedControl" onchange="changeSpeed(this.value)">
        <option value="0.25">0.25x</option>
        <option value="0.5">0.5x</option>
        <option value="1" selected>1x</option>
        <option value="1.25">1.25x</option>
        <option value="1.5">1.5x</option>
        <option value="2">2x</option>
      </select>
    </div>
  </div>

  <footer>Designed to make your mood musical!</footer>

  <script>
    const songs = {
      happy: [
        "C:/Users/User/Downloads/Badtameez Dil - Yeh Jawaani Hai Deewani 320 Kbps (3).mp3",
        "C:/Users/User/Downloads/Crazy-Crazy-Feeling.mp3",
        "C:/Users/User/Downloads/Baby(PagalWorld.com.so).mp3"
      ],
      sad: [
        "C:/Users/User/Downloads/128-Papa Meri Jaan - Animal 128 Kbps.mp3",
        "C:/Users/User/Downloads/Adiga Adiga - SenSongsMp3.Co.mp3",
        "C:/Users/User/Downloads/Ava-(Speed-Up-Tiktok-Version)(PagalWorld).mp3"
      ],
      relaxed: [
        "C:/Users/User/Downloads/Pani Da Rang Male Vicky Donor 128 Kbps.mp3",
        "C:/Users/User/Downloads/Samajavaragamana - SenSongsMp3.Co (1).mp3",
        "C:/Users/User/Downloads/Let Me Down Slowly-(PagalSongs.Com.IN).mp3"
      ],
      focused: [
        "C:/Users/User/Downloads/128-Kar Har Maidaan Fateh - Sanju 128 Kbps.mp3",
        "C:/Users/User/Downloads/3- Dont Stop-SenSongsMp3.Co.mp3",
        "C:/Users/User/Downloads/Weightless.mp3"
      ],
      energetic: [
        "C:/Users/User/Downloads/Jai Jai Shivshankar War 128 Kbps.mp3",
        "C:/Users/User/Downloads/Naatu Naatu (From Rrr).mp3",
        "C:/Users/User/Downloads/Bye Bye Bye(PagalWorld.com.so).mp3"
      ],
      devotional: [
        "C:/Users/User/Downloads/Namo-Namo-Ji-Shankara(PagalWorld).mp3",
        "C:/Users/User/Downloads/Powerful Hanuman Chalisa (HanuMan)-(Mr-Jat.in).mp3",
        "C:/Users/User/Downloads/Brother-Nic-Mp4-The-Goodness-Of-God (1).mp3"
      ]
    };

    let currentSongIndex = { happy: 0, sad: 0, relaxed: 0, focused: 0, energetic: 0, devotional: 0 };
```

```
    const audioPlayer = document.getElementById('audioPlayer');
    const audioControls = document.getElementById('audioControls');

    function playMood(mood) {
      const moodSongs = songs[mood];
      const index = currentSongIndex[mood];
      audioPlayer.src = moodSongs[index];
      audioPlayer.play();
      audioControls.classList.add('show');
      currentSongIndex[mood] = (index + 1) % moodSongs.length;
    }

    function playAudio() {
      audioPlayer.play();
    }

    function pauseAudio() {
      audioPlayer.pause();
    }

    function changeSpeed(speed) {
      audioPlayer.playbackRate = parseFloat(speed);
    }

    function changeTheme(theme) {
      const root = document.body;
      if (theme === "system") {
        root.setAttribute("data-theme", "system");
      } else {
        root.setAttribute("data-theme", theme);
      }
    }
  </script>
</body>
</html>
```

1. **Mood-Based Music Selection:**
    - The music player offers users a range of moods such as *Happy*, *Sad*, *Relaxed*, *Focused*, *Energetic*, and *Devotional*.
    - Each mood corresponds to a set of predefined tracks stored in the songs object.
    - When a user clicks on a mood card, a song from the respective mood playlist is selected and played.
    - After playing a song, the app cycles through the tracks of the chosen mood in sequence.
2. **Audio Controls:**
    - The player allows the user to control the audio playback through buttons like *Play* and *Pause*.
    - There is also a playback speed selector with options ranging from 0.25x to 2x speed, enabling users to adjust the pace of the music.
3. **Visual & Interactive Design:**
    - The user interface features a clean, modern design with floating musical notes for visual aesthetics.
    - Mood cards are styled with gradient backgrounds and hover effects that increase interactivity.
    - The header includes an animated title with a glowing effect, which enhances the overall experience.
4. **Dark and Light Theme Toggle:**
    - The user can toggle between *Light*, *Dark*, or *System Default* modes using a dropdown menu.
    - The theme change is reflected globally on the page, modifying the background, text color, and card gradient for an improved user experience.
5. **Responsive Layout:**
    - The layout of the music player is responsive, ensuring it adapts to different screen sizes. The mood cards adjust flexibly with varying screen widths.
6. **Floating Notes:**
    - Animated musical notes (, , , ,) float around the screen, adding a dynamic visual layer to the app. These notes are animated with the @keyframes float CSS animation, creating a pleasant, calming effect.
7. **CSS Styling and Animations:**
    - The page is styled using CSS variables (for easy theme customization) and animations like pulse and float for text and note movements, respectively.
    - The design incorporates smooth transitions for background color changes, hover effects, and button presses.
    - The "mood cards" use gradient backgrounds and subtle shadow effects to make the user interface visually appealing and interactive.

Technologies Used:

1. **HTML:**

- The structure is created using HTML5 elements such as <h1>, , <footer>, <select>, and <audio>.
- The data-theme attribute is used for dynamic theme switching.

2. CSS:
   - The CSS styling leverages modern features like CSS variables (--bg-gradient, --text-color), flexbox, and @keyframes for animations.
   - The linear-gradient CSS property is heavily used for background and card effects, giving the app a smooth, modern look.
   - Transitions and animations enhance the user experience, making the application more engaging and visually dynamic.
3. JavaScript:
   - The songs object stores arrays of music file paths for each mood.
   - Functions like playMood(), playAudio(), and pauseAudio() manage the audio playback based on user interaction.
   - The changeSpeed() function adjusts the audio playback speed dynamically, allowing for more personalized listening.
   - The changeTheme() function allows the user to switch between different themes (light/dark/system default) based on user preference.

Code Analysis:

1. Theme Management:
   - The data-theme attribute is initially set to "system", which means it follows the system's default theme setting.
   - The changeTheme() function modifies this attribute to either "light" or "dark", which changes the background gradient and text colors across the page.
2. Mood Card Interactions:
   - The .mood-card elements are clickable and trigger the playMood() function when clicked.
   - The playMood() function selects the appropriate music file based on the mood, updates the audio player's source (audioPlayer.src), and plays the track.
3. Audio Controls:
   - The audio-controls section, which is initially hidden, is displayed when a user selects a mood and starts playing music.
   - The buttons provided in the audio-controls section allow for controlling the audio playback (play, pause) and changing the speed of the audio playback.
4. Responsive Design:
   - The layout is built using flexbox, with mood cards adjusting their size and placement according to the screen size. The flex-wrap: wrap; property ensures that the cards flow correctly on smaller screens.
5. Floating Musical Notes:
   - The musical notes are animated with a @keyframes animation, making them float upwards and fade in and out, creating a playful atmosphere.

Detailed Explanation of Code

3. Features of the Dreamland Music Player

3.1 Mood-Based Music Selection

Code Explanation:

- The application uses a JavaScript object to store playlists. Each key in the object corresponds to a mood (e.g., happy, sad), and its value is an array of song URLs.
- A function playMood(mood) dynamically updates the src attribute of an <audio> element based on the selected mood.
- The playMood function ensures uninterrupted playback by using the onended event listener to loop through the playlist.

```
javascript
Copy code
const playlists = {
  happy: ["song1.mp3", "song2.mp3", "song3.mp3"],
  sad: ["song4.mp3", "song5.mp3", "song6.mp3"],
  // Add more moods and tracks
};
function playMood(mood) {
  const audio = document.getElementById("audioPlayer");
  let index = 0;

  function playNext() {
    if (index < playlists[mood].length)
      {              audio.src              =
     playlists[mood][index];
     audio.play();
     index++;
    } else {
     index = 0; // Reset the playlist
    }
  }

  audio.addEventListener("ended", playNext);
  playNext();
}
```

Advantages:

- Provides a dynamic, user-friendly way to select music based on mood.
- Ensures continuous playback without manual intervention.

3.2 Audio Playback Controls

Code Explanation:

- Functions like playAudio() and pauseAudio() control the playback using the HTML5 <audio> API.
- The changeSpeed(speed) function adjusts playback speed using the playbackRate property of the <audio> element.

```
javascript
Copy code
function playAudio()
 { document.getElementById("audioPlayer").play();
}
function pauseAudio()
 { document.getElementById("audioPlayer").pause();
}
function changeSpeed(speed) {
 const audio = document.getElementById("audioPlayer");
 audio.playbackRate = speed;
}
```

Advantages:

- Simple and intuitive playback control.
- Customizable speed enhances versatility, suitable for meditation or high-energy activities.

3.3 Dynamic User Interface

Code Explanation:

- The UI uses CSS for interactive elements like gradient mood cards and hover effects.
- JavaScript dynamically updates classes to add hover animations and glowing effects.

```
css
Copy code
.mood-card {
 background: linear-gradient(45deg, #ff9a9e, #fad0c4);
 transition: transform 0.3s ease;
}
.mood-card:hover
 { transform:
 scale(1.1);
 box-shadow: 0px 4px 15px rgba(0, 0, 0, 0.2);
}
.title {
```

```
  animation: glow 2s infinite;
}@keyframes glow {
  0% { text-shadow: 0 0 5px #ff9a9e; }
  50% { text-shadow: 0 0 20px #fad0c4; }
  100% { text-shadow: 0 0 5px #ff9a9e; }
}
```

Advantages:

- Engages users with visually appealing effects.
- Improves user interaction and satisfaction.

3.4 Theme Customization

Code Explanation:

- The application uses JavaScript to toggle between Light, Dark, and System Default themes.
- Themes are implemented using a data-theme attribute on the <html> element.

```
javascript
Copy code
function changeTheme(theme)
  { document.documentElement.setAttribute("data-theme",
  theme);
}
```

Advantages:

- Makes the app usable in diverse lighting conditions.
- Enhances accessibility for all users.

3.5 Responsive Design

Code Explanation:

- The app uses CSS Flexbox for layout, ensuring smooth realignment across devices.
- Media queries optimize the design for various screen sizes.

```
css
Copy code
@media (max-width: 600px) {
  .mood-card {
    flex: 100%; /* Stack cards vertically on small screens */
  }
```

}

Advantages:

- Ensures the app is accessible and functional on smartphones, tablets, and desktops.
- Enhances user experience across platforms.

3.6 Floating Notes and Animations

Code Explanation:

- CSS animations create floating musical notes that add a playful and relaxing visual element.

```
css
Copy code
.note {
  animation: float 5s infinite;
}
@keyframes float {
  0% { transform: translateY(0); opacity: 1; }
  100% { transform: translateY(-50px); opacity: 0; }
}
```

Advantages:

- Makes the app visually delightful and engaging.
- Adds a unique touch that differentiates it from standard music players.

---

5. Expanded Advantages

5.1 Enhanced Emotional Connection

- Music deeply influences emotions; by aligning with moods, the app becomes a companion for emotional well-being.
- Example: Uplifting tracks during sadness can elevate mood, while calming music can alleviate stress.

5.2 Versatility

- Customization options make the app suitable for diverse scenarios like work, relaxation, or fitness.
- Example: A focused playlist can help users excel in demanding tasks, while energetic tracks make workouts enjoyable.

5.3 Accessibility

- Responsive design ensures inclusivity, allowing users to access the app regardless of their device.
- Themes improve usability in both bright and dim lighting.

5.4 Interactivity and Engagement

- Visual effects like floating notes and glowing headers enhance user experience.
- Example: Users may find the app more enjoyable due to its interactive and aesthetic design.

5.5 Personalization

- Tailored playlists and adjustable playback speed empower users to control their experience.

Advantages of the Dreamland Music Player

1. Enhanced Personalization

- The platform curates playlists based on user-selected moods, ensuring a tailored experience.
- Adjustable playback speed and customizable themes add further flexibility to suit individual preferences.

2. Emotional Intelligence

- The integration of mood-based music selection bridges the gap between technology and human emotions.
- Helps users manage stress, uplift their spirits, and maintain focus during work or studies.

3. Versatility

- The player adapts to various use cases:
    - **Relaxation:** Calming music for meditation or unwinding after a hectic day.
    - **Focus:** Instrumental or low-tempo tracks for productivity.
    - **Energy:** High-tempo, upbeat music for workouts or social gatherings.
    - **Entertainment:** Themed playlists and videos for parties or personal enjoyment.

4. Accessibility

- Responsive design ensures functionality across multiple devices, from desktops to smartphones.
- Theme customization improves usability in different lighting conditions, making it suitable for day or night use.

5. Interactivity and Engagement

- Floating musical notes, glowing headers, and gradient mood cards provide a visually engaging interface.
- Dropdowns for music and video selection allow users to explore and control their listening experience seamlessly.

6. Potential for Future Growth

- Features like AI-driven recommendations and voice commands (planned in future updates) ensure the platform remains relevant and innovative.
- Scalability with cloud integration and social sharing can expand the app's appeal.

7. Mental Well-Being

- Music tailored to emotions can have therapeutic effects, reducing anxiety, enhancing happiness, and providing comfort during tough times.
- By addressing emotional needs, the player becomes more than just a tool—it becomes a companion.

8. Ease of Use

- Simple and intuitive controls make the app user-friendly for all age groups.
- Dropdown menus and interactive buttons ensure hassle-free navigation.

9. Time-Efficient

- Mood-based selection eliminates the need to search for music manually, saving time and effort.
- Playlists automatically cycle, ensuring uninterrupted listening.

10. Customization

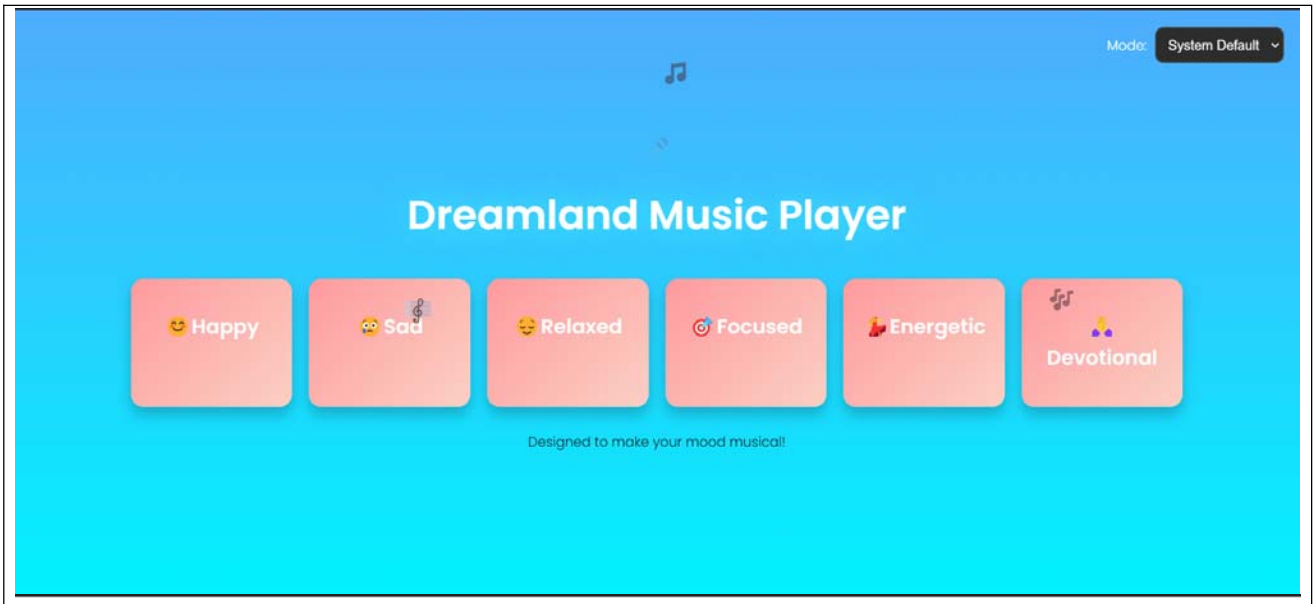
- Users can choose their mood, theme, and even specific tracks or videos, allowing a fully tailored experience.
- The ability to toggle themes between dark and light modes adds comfort to the visual experience.

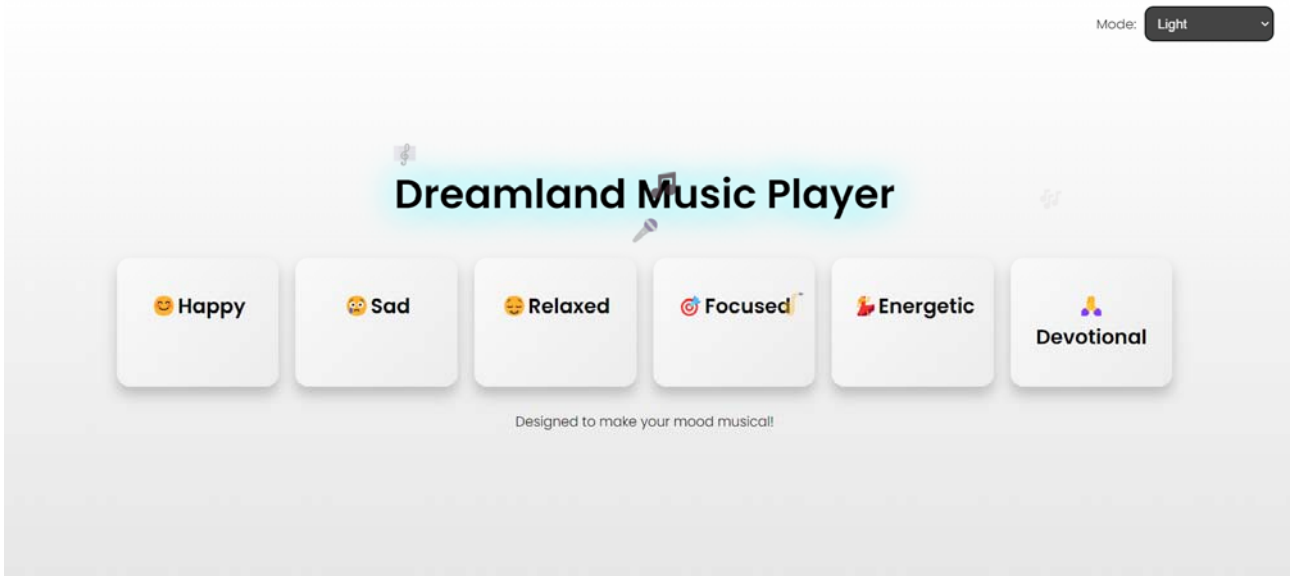
11. Visual Appeal

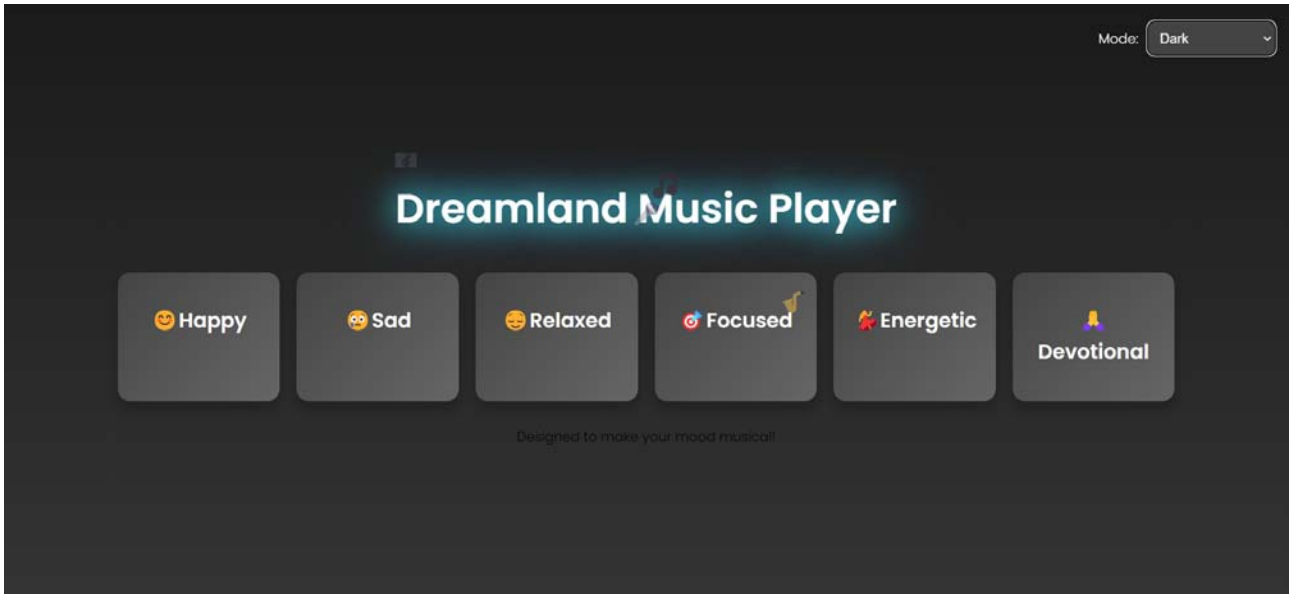
- Floating animations and smooth transitions make the app visually satisfying, improving user retention and enjoyment.

12. Adaptability to Various Scenarios

- From personal relaxation to social events, the player serves a broad range of purposes, enhancing its practical value.


Mode: System Default
Dreamland Music Player
Happy
Sad
Relaxed
Focused
Energetic
Devotional
Designed to make your mood musical!


Mode: Light
Dreamland Music Player
Happy
Sad
Relaxed
Focused
Energetic
Devotional
Designed to make your mood musical!


Mode: Dark
Dreamland Music Player
Happy
Sad
Relaxed
Focused
Energetic
Devotional
Designed to make your mood musical!

Conclusion:

The "Dreamland Music Player" provides a highly interactive and engaging music-playing experience. It effectively uses mood-based music selection, theme customization, and audio controls to enhance user experience. The clean, responsive design combined with unique features like floating notes and theme toggling makes it a visually appealing and functional web application. With some improvements to scalability and user experience features, this application has the potential to be a great mood-based music player platform.