**Maulana Abul Kalam Azad**

University Of Technology

**MAKAUT WB**

**PROJECT TITLE: ONLINE MUSIC PLAYER**

|  |
| --- |
| Submitted by :  Name : Sumanta Bhattacharya  Roll No : 30001222180  Course : BCA  Department : Computer Applications |

The project report submitted in partial fulfillment of the requirement for the BCA of Second year for the session 2022 to 2025 of

**Maulana Abul Kalam Azad University of Technology**

**MAKAUT WB**

**DECLARATION**

I Sumanta Bhattacharya, student of **Maulana Abul Kalam Azad University of Technology**, **Kalyani** here declare do hereby declare that the project work entitled "ONLINE MUSIC PLAYER" is the result of an original research work based on my own experience and study.

I further declare that it has not been previously submitted either in part or full to this or to any other university for any degree or diploma.

Place: **Kalyani**

Date:

Sumanta Bhattacharya

**MAKAUT WB**

**CERTIFICATE**

This is to certify that the project work entitled "ONLINE MUSIC PLAYER" submitted by us for the partial fulfillment of the requirement for the subject Software Engineering of BCA Second year is based upon my work carried out and that neither this work nor any part of it has been submitted before for any degree or diploma anywhere/elsewhere.

Counter signature by the supervisor

**MAKAUT WB**

**AKNOWLEDGEMENT**

* It is not possible to carry out any study or project without the guidance and help of the experts and supporters. In the same way, there are also people without which our research would not have been possible.
* I like to express my sincere gratitude to my teachers for their teachings, valuable guidance and support for completion of this project.

**MAKAUT WB**

DECLARATION

CERTIFICATE

ACKNOWLEDGEMENT

CONTENTS:

Introduction PAGE NO: 6

System Requirement Analysis PAGE NO: 7

Objectives PAGE NO: 8

Key Features PAGE NO: 9

File Structure PAGE NO: 10

System Specification PAGE NO: 11

Project Works PAGE NO: 12

Design PAGE NO: 13

Flowchart PAGE NO: 14-15

Source Code:

HTML Source Code PAGE NO: 16

CSS Source Code PAGE NO: 17-19

JS Source Code PAGE NO: 20-22

Deployment and Maintenance PAGE NO: 24

Conclusion PAGE NO: 22

**MAKAUT WB**

**INTRODUCTION**

This document outlines the Software Development Life Cycle

(SDLC) for developing a translator application using JavaScript.

It details each phase of the SDLC, including planning, analysis,

design, development, testing, deployment, and maintenance,

along with additional considerations for security, scalability, and

user experience.

Welcome to the Online Music Player project, a web-based solution designed to bring music to your ear seamlessly. ONLINE MUSIC PLAYERis a project named Web Based online Music Player. This project is done using HTML, CSS, and JAVASCRIPT. If you are searching for a simple and lightweight solution to enable music on your website, regardless of the user's internet browser this is the best solution.

This website allows you to play copyrighted music that you can listen on our site without fear of legal action.

Experience the harmony of technology and music with this Online Music Player project. Immerse yourself in a world of melodies and rhythms, all at your fingertips.

SYSTEM REQUIREMENT ANALYSIS

●**Planning** is an objective of each and every activity, where we want to discover things that belong to the project. An important task in creating a software program is extracting the requirements or requirements analysis.

Record analysis and specification phases to gather the requirements from the stakeholder and document them properly.

Feasibility study is a study to determine whether a project is feasible or not.

It evaluates whether the project is Cost effective or not.

It evaluates whether the project can be implement using available technologies and resources.

• **Implementation, testing and documenting**

Implementation is the part of the process engineers program the code for the project.

Software testing is an integral and important phase of the software development process. This part of the process ensures that defects are recognized as soon as possible.

Documenting the internal design of software for the purpose of future maintenance and enhancement is done throughout development.

Objectives

The objective of this:

1. **Make it with a simple feature and run smoothly**

By using this mp3 music player will make users fell comfortable and relaxed because it will pay more attention to the features commonly used by users, excluding some rarely used features that occupy a large of system processors, making the music player lightweight, simple, but also has powerful basic features.

1. **Ensure Cross-Browser Compatibility**:
   * The music player is developed using HTML, CSS, and JavaScript, guaranteeing compatibility across all major web browsers. This ensures a consistent and reliable experience for all users, regardless of their preferred browser.

3**. Provide Legal Music Streaming**:

* + The platform allows users to play copyrighted music without the fear of legal repercussions. All music available on the site is fully licensed, ensuring compliance with copyright laws and providing a safe and legal way to enjoy music.

1. **Optimize Performance and Efficiency**:

By minimizing the use of system resources and focusing on essential features, the music player ensures optimal performance. This results in faster load times, smooth playback, and a better overall user experience.

**Key Features**

* **Cross-Browser Compatibility**:

The music player works flawlessly on all major web browsers, ensuring that users have a consistent experience regardless of their choice of browser.

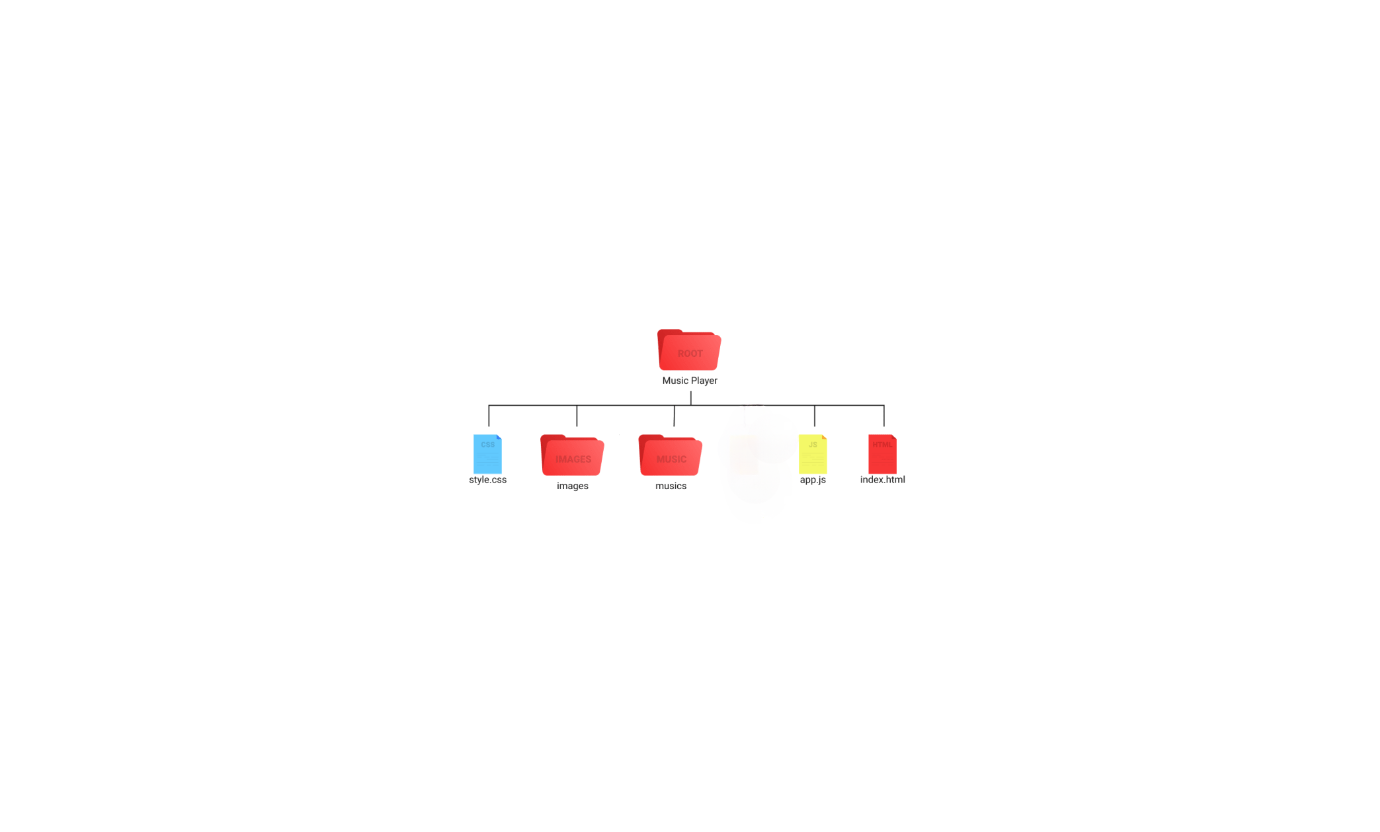
* **Lightweight Design**:

The player is designed to be lightweight, ensuring quick load times and smooth performance without burdening your website’s resources.

* **Legal Compliance**:

The music available through our platform is fully licensed, allowing you to stream copyrighted music legally without the fear of legal repercussions.

**File structure**



The file structure of the "Music Player" project is

1. **HTML File: index.html**
   * Located directly under the root directory.
   * The main HTML file that structures the content of the music player web page. It includes references to CSS for styling and JavaScript for functionality.
2. **CSS File: style.css**
   * Located directly under the root directory.
   * Contains the CSS styles that define the layout, appearance, and design of the music player.
3. **JavaScript Files:**
   * **app.js**: Located directly under the root directory.

Contains the main JavaScript code that implements the functionality of the music player, including handling user interactions, controlling playback, and updating the UI.

SYSTEM SPECIFICATION

**Hardware Specification**

* **Processor**: Intel i3 (3rd Generation) or equivalent/higher
* **RAM**: 2 GB or higher
* **Hard Disk Space**: 20 GB or higher

**Software Specification**

* **Operating System**: Windows family, macOS, Android, iOS
* **Front End Technologies**: HTML, CSS, JavaScript
* **Back End Technology**: JavaScript

PROJECT WORKS

MUSIC PLALYER This is the main module; users can play their favorite uploaded music from this page. First, by default, we will display the album with song name and image. User can click on any of their favorite music to play.

This allows for personalized listening experiences based on individual preferences. This visual representation helps users quickly locate their desired music

When user clicks on the play button, we will play the first song of that album, by default. Users can select any song to play. User can play, pause, and play next and previous songs of the list.

* **Playback Controls**: The player includes essential controls such as:
  + **Play**: Start or resume playing the current track.
  + **Pause**: Temporarily halt the playback of the current track.
  + **Next**: Skip to the next song in the album.
  + **Previous**: Go back to the previous song in the album.

These controls provide users with full command over their music playback.

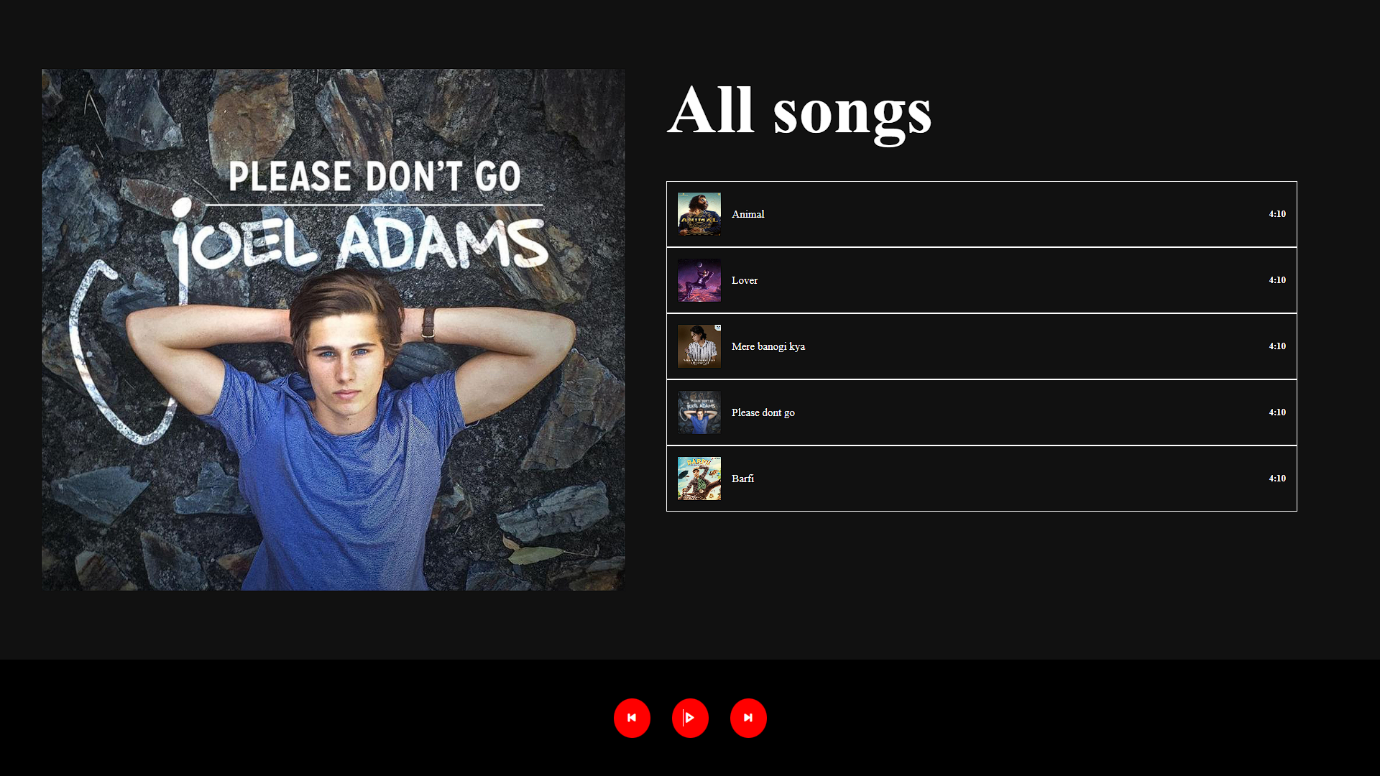
Design: design is the major step in moving from problem domain to solution domain.

#### Objectives:

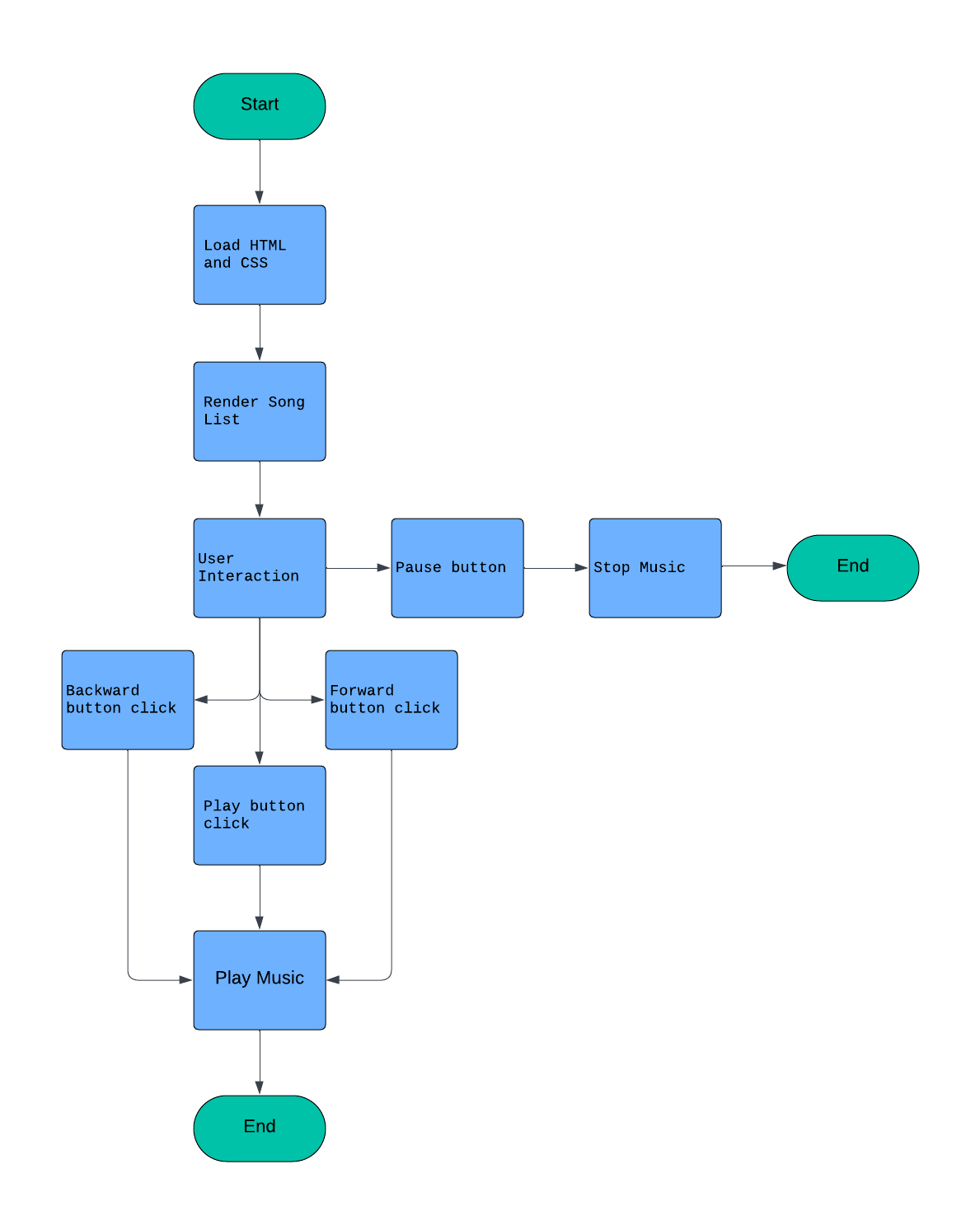
* Design the architecture of the application.
* Create wireframes and prototypes for the user interface.
* Plan the database schema to store songs and user data.
* Outline the flow of the application based on the flowchart.

#### Deliverables:

* System architecture design
* UI/UX design mockups
* Database schema
* Detailed flowchart



Flowchart



This flowchart provides a high-level overview of how the music player application works.

1. **Start**: The process begins here. It’s the entry point for your music player application.
2. **Load HTML and CSS**: This step involves loading the necessary HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) files. These files define the structure and styling of your music player interface.
3. **Render Song List**: After loading the HTML and CSS, the application renders the list of songs available to the user.
4. **User Interaction**:
   1. **Backward Button**: If the user clicks the backward button, the application responds by moving to the previous song in the list.
   2. **Forward Button**: Similarly, the forward button moves to the next song.
   3. **Play Button**: Clicking the play button starts playing the current song.
   4. **Pause Button**: Pausing the music temporarily stops playback.
   5. **Stop Music**: Stopping the music ends playback altogether.
5. **Play Music**: When the user interacts with the play button, the application plays the selected song.
6. **End**: The flowchart concludes here. It represents the termination point of the application.

Coding: In this phase, write code according to the design specifications using a programming language. Coding converts the design into code in specific language. Coding phase affects both testing and maintenance.

SOURCE CODE

HTML SAMPLE CODE

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

    <link rel="stylesheet" href="style.css">

    <link

    href="https://cdn.jsdelivr.net/npm/remixicon@4.2.0/fonts/remixicon.css"

    rel="stylesheet"

/>

</head>

<body>

    <div class="main">

        <div class="left"></div>

        <div class="right">

            <h3>All songs</h3>

            <div class="all-songs">

            </div>

        </div>

    </div>

    <div class="player">

        <h3 id="backward"><i class="ri-skip-back-mini-fill"></i></h3>

        <h3 id="play"><i class="ri-play-fill"></i></h3>

        <h3 id="forward"><i class="ri-skip-forward-mini-fill"></i></h3>

        <!-- <i class="ri-pause-circle-line"></i> -->

    </div>

    <script src="app.js"></script>

</body>

</html>

CSS SAMPLE CODE

\*{

    padding: 0;

    margin: 0;

    box-sizing: border-box;

    color: white;

}

html,body{

    height: 100%;

    width: 100%;

    position: relative;

}

.main{

    height: 85%;

    width: 100%;

    background-color: #111;

    display: flex;

    padding: 5vw 3vw;

}

.left{

    height: 100%;

    width: 45%;

    background-color: #333;

    /\* background-image: url("https://bollywoodmizaz.com/wp-content/uploads/2023/12/Pehle-Bhi-Main-Lyrics-1.jpg"); \*/

    background-repeat: no-repeat;

    background-size: cover;

    background-position: center;

}

.right{

    height: 100%;

    width: 55%;

    padding: 0 3vw;

}

.right h3{

    font-size: 5vw;

}

.all-songs{

    height: 70%;

    margin-top: 3rem;

    overflow-y: auto;

}

.all-songs::-webkit-scrollbar{

    display: none;

}

.song-card{

    display: flex;

    align-items: center;

    justify-content: space-between;

    border: 1px solid white;

    padding: 1rem 1rem;

}

.song-card:hover{

    background-color: #8f0808;

}

.song-card .part1{

    display: flex;

    align-items: center;

    pointer-events: none;

}

.song-card img{

    height: 4rem;

    width: 4rem;

    margin-right: 1rem;

}

.song-card h2{

    font-size: 1rem;

    font-weight: 500;

}

.song-card h6{

    font-size: 0.85rem;

}

.player{

    height: 15%;

    width: 100%;

    background-color: #000;

    display: flex;

    align-items: center;

    justify-content: center;

    gap: 2rem;

}

.player h3{

    background-color: red;

    padding: 1rem;

    border-radius: 50%;

    font-size: 1.4rem;

}

.player h3:active{

    scale: 0.9;

}

Js SAMPLE CODE

var arr = [

  {

    songName: "Animal",

    url: "./song/Animal.mp3",

    imageUrl: "./images/Ani.jpg",

  },

  {

    songName: "Lover",

    url: "./song/lover.mp3",

    imageUrl: "./images/diljit.jpg",

  },

  {

    songName: "Mere banogi kya",

    url: "./song/mere banagi kya.mp3",

    imageUrl: "./images/mere.jpg",

  },

  {

    songName: "Please dont go",

    url: "./song/hmmmm.mp3",

    imageUrl: "./images/image.png",

  },

  {

    songName: "Barfi",

    url: "./song/barfi.mp3",

    imageUrl: "./images/barf.jpg",

  },

];

var poster = document.querySelector(".left");

var audio = new Audio();

var selectedSong = 0;

function mainFunction() {

  var clutter = "";

  arr.forEach(function (e, i) {

    clutter += `

        <div class="song-card" id=${i}>

            <div class="part1">

                <img src="${e.imageUrl}" alt="${e.songName}">

                <h2>${e.songName}</h2>

            </div>

            <h6>4:10</h6>

        </div>`;

  });

  poster.style.backgroundImage = `url("${arr[selectedSong].imageUrl}")`;

  audio.src = arr[selectedSong].url;

  document.querySelector(".all-songs").innerHTML = clutter;

}

mainFunction();

document.querySelector(".all-songs").addEventListener("click", function (dets) {

  selectedSong = dets.target.id;

  mainFunction();

  audio.play();

  play.innerHTML = '<i class="ri-pause-line"></i>';

  flag = 1;

});

var play = document.querySelector("#play");

var backward = document.querySelector("#backward");

var forward = document.querySelector("#forward");

var flag = 0;

play.addEventListener("click", function () {

  if (flag == 0) {

    play.innerHTML = '<i class="ri-pause-line"></i>';

    flag = 1;

    mainFunction();

    audio.play();

  } else {

    play.innerHTML = '<i class="ri-play-line"></i>';

    mainFunction();

    audio.pause();

    flag = 0;

  }

});

forward.addEventListener("click", function (e) {

  if (selectedSong < arr.length - 1) {

    selectedSong++;

    mainFunction();

    audio.play();

    forward.style.opacity = 1;

  } else {

    forward.style.opacity = 0.43;

  }

});

backward.addEventListener("click", function (e) {

  if (selectedSong > 0) {

    selectedSong--;

    mainFunction();

    audio.play();

    backward.style.opacity = 1;

  } else {

    backward.style.opacity = 0.43;

  }

});

Deployment and maintenance

●Deployment starts after the code is appropriately tested, approved for release then its ready to be distributed into a production environment. This may involve installation, customization (such as by setting parameters to the customer's values), testing, and possibly an extended period of evaluation.

Maintaining and enhancing software to cope with newly discovered faults or requirements can take substantial time and effort, as missed requirements may force redesign of the software.

**CONCLUSION**

* This chapter is a concluding one which deals with the summary and findings of the present study. In this study I have made an attempt to see the result of our study but it also reminds us that this Research work has further scope of study.
* In this project I have made an attempt to see the result of my study and throughout this project, I have diligently worked to create a digital space that mirrors the energy and uniqueness of a Online Music Player.

**Overall Findings of the Study:**

* This project comprises of these technologies HTML, CSS and JavaScript.
* Intuitive controls allow users to play, pause, and navigate through tracks easily.
* It provides a user-friendly and lightweight music player for seamless music enjoyment.
* The project fulfils the requirements of the Software Engineering course.