

# **TULA'S INSTITUTE**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**Practical file of**  
**Software Engineering (BEC-403)**

---

---

Session: - 2021-22

SUBMITTED TO: -

DR.BHARTI KALRA  
ASSOCIATE PROFESSOR

SUBMITTED BY: -

SHIVAM SHUKLA (202004038)

## Table Of Contents

[illegible]

# **EXPERIMENT NO. 1**

**AIM:** TO BUILD A MUSIC PLAYER ON A WED PAGE

## **REQUIREMENTS:**

### **Hardware Interfaces**

- Pentium(R) i5, 2GB RAM

### **Software Interfaces**

- Any window-based operating system (Windows 8/10)
- WordPad or Microsoft Word

## **EXPECTED WORKING**

The software is expected to play songs provided in the menu of the webpage and then be able to play, pause, next and previous song with the help of control provided at the bottom of the page.

## **Introduction**

### **Goals and Requirements**

The document addresses the following goals and functional requirements for designing a simulate portable media player.

Access audio collection

Repeat track

Display track information

Play songs in queue

Play/Pause button

artwork images

### **Product Scope**

This product will be used by individual who need to store and listen to their music files on a portable device. The objective is to design a simple, interactive interface for playing music.

## **References**

Took help from a web developer intern.

## Design Overview

### System Overview

This simulated portable media player will be implemented as a website using html, CSS, javascript.

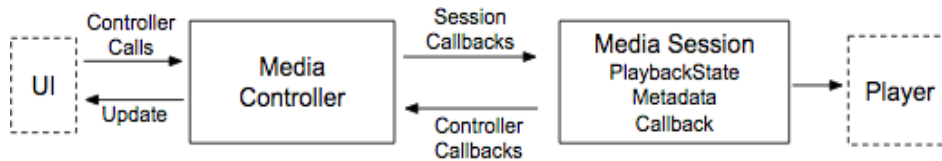


Figure 1:Media Player Architecture

### General Constraints

The player's display area is limited to 320x372 pixels screen size and 320x372 pixels for interactive controls[3].

### User Interface Design

This section illustrates the user interface for each case and elaborates on the design decisions for the simulated portable media player. Since user for usability study used their media player solely for listening to audio and particularly music, this interface design focuses mainly on the audio features. The control area consists of three buttons. Two are clearly labelled directional buttons for navigating the menus, the center button for play/pause .



### Graphical Interface:

#### Menu Bar:

**Description:** The menu bar will be displayed at the lower position of the media player where all the option regarding the software and the files will be given. For a particular task the user have to choose the proper option only.

**Input:** User need to click the option button as displayed in the software.

**Output:** The specific task will be performed.

#### Status Bar:

**Description:** The status bar use for displaying the duration of the running file.

**Input:** The mouse click on any position on the status bar.

**Output:** The running file will be seek to that time duration.

## **TECHNOLOGY BUCKET**

JavaScript, HTML, CSS and web application development

## **CONCLUSION**

On successful completion and execution, this software is supposed to play songs without any error and provide run perfectly.

## EXPERIMENT NO. 2

**Aim:** Understanding an SRS.

**Requirements:**

### Hardware Requirements:

Computer and processor	Dual-Core processor (Quad-Core or higher is recommended, 2.0 GHz or faster)
Memory	2 GB (minimum)
Free disk space	2 GB for installation and 20 GB for operation
Display	1366x768 or higher resolution
Audio	Sound card and microphone (for narration recording)
Video	Built-in or external webcam (for video recording)
Windows	Mozilla Firefox 45 or higher Google Chrome 48 or higher Microsoft Edge
Macintosh	Safari 11 or higher
Linux	Mozilla Firefox 45 or higher Google Chrome 48 or higher

### Software Requirements:

Software
----------

Operating System	Microsoft® Windows® 7/8/10 (32- and 64-bit)
Microsoft Office	Microsoft® Office® 2007 or higher (32- and 64-bit)
Microsoft PowerPoint	Microsoft® PowerPoint® 2007/2010/2013/2016/2019/365 (32- and 64-bit) (ISO (Non-Windows Store) versions)
Microsoft Word	Microsoft® Word® 2007/2010/2013/2016/2019/365 (32- and 64-bit) (ISO (Non-Windows Store) versions)
Adobe Flash	Adobe® Flash® Player ActiveX 12 or higher

## EXERCISE NO. 3

### AIM :-

To draw a sample ENTITY RELATIONSHIP DIAGRAM diagram

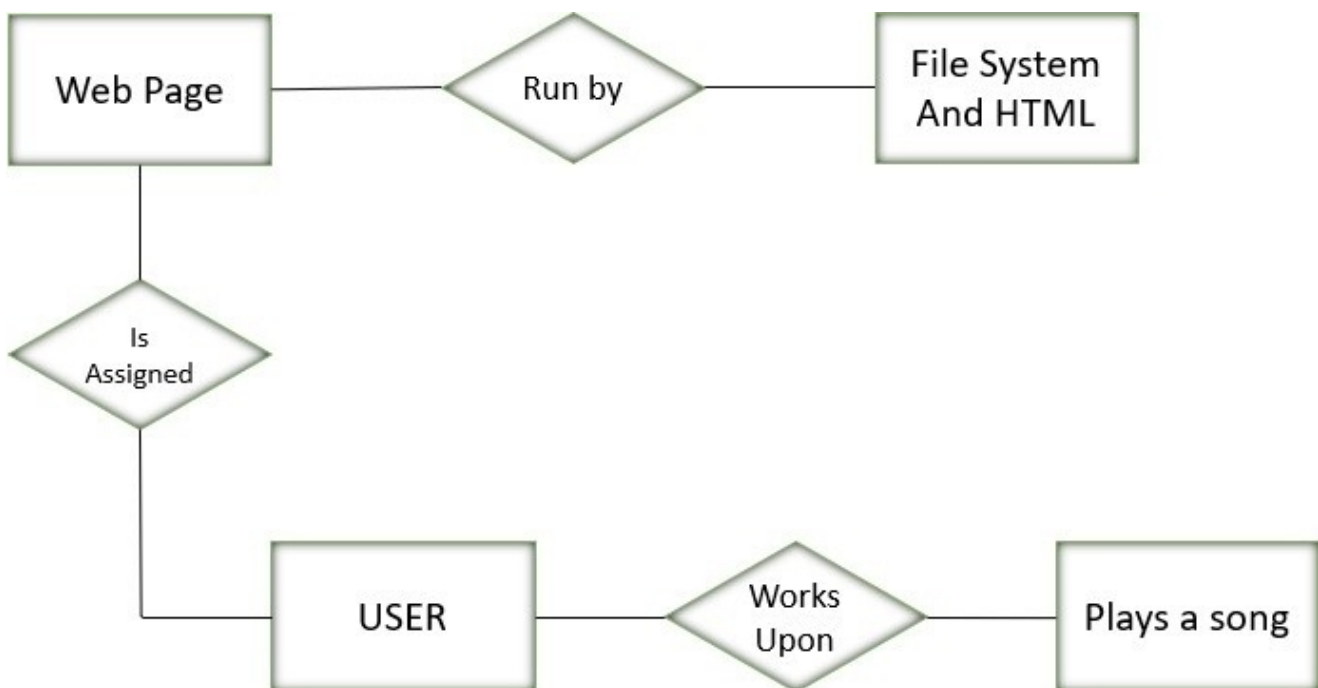
### Hardware Requirements:

- Pentium(R) i5, 2GB RAM.

### Software Requirements:

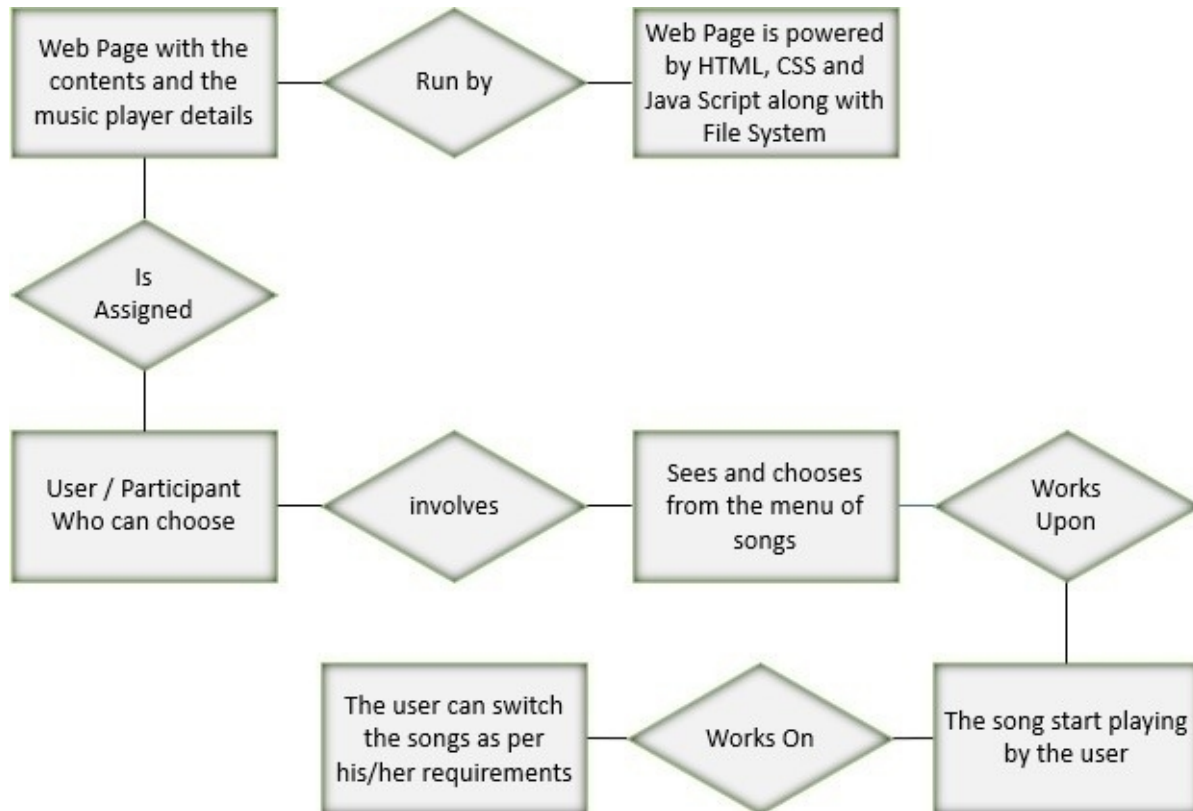
Visual Studio 2022, Visual Paradigm online, Windows 10/11.

### Rough ER diag:





### Fully Attributed ERD:



**Conclusion:** The entity relationship diagram was made successfully by following the steps described above

## EXERCISE NO. 4

**AIM:** To prepare DATA FLOW DIAGRAM

### REQUIREMENTS:

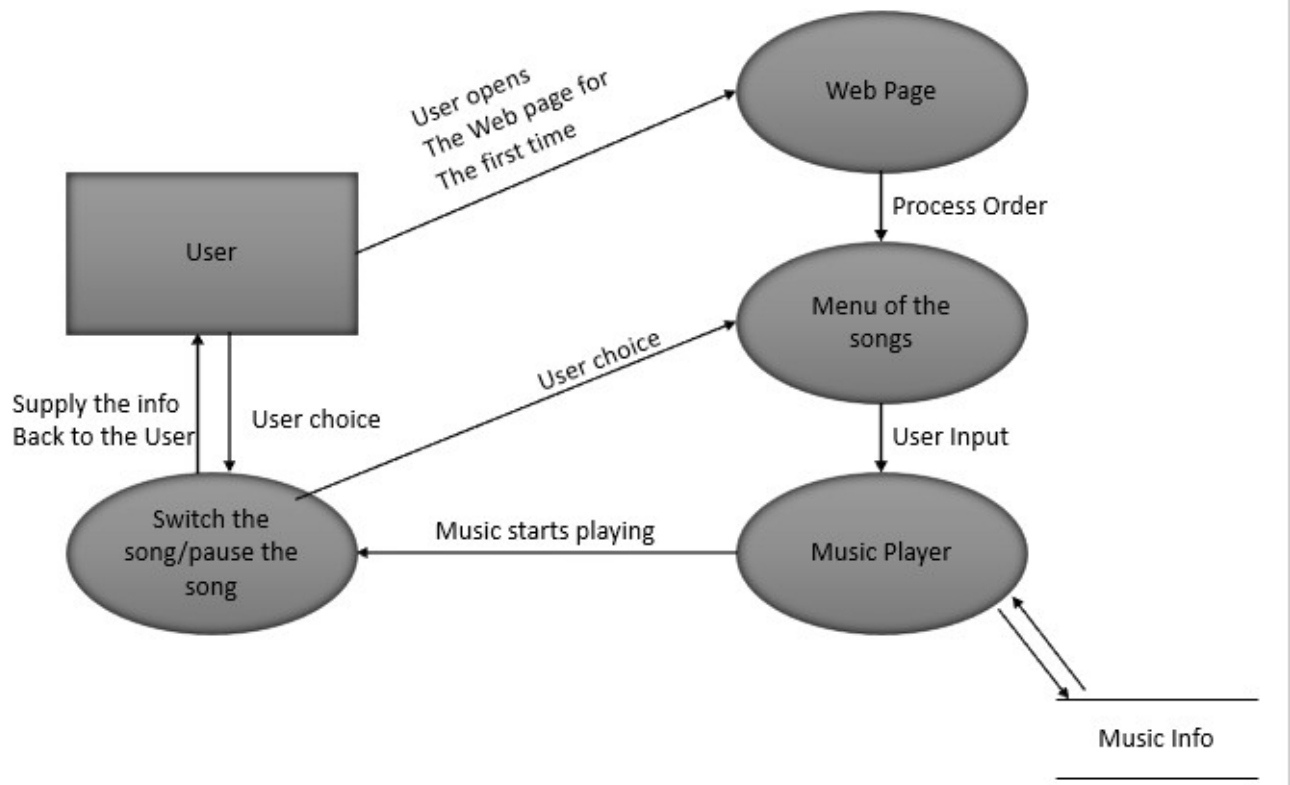
#### Hardware Interfaces

- Pentium(R) i5, 2GB RAM.

#### Software Interfaces

- Any window-based operating system (Windows 11/10)
- WordPad or Microsoft Word, Visual Studio 2022, Visual Paradigm online

#### Dataflow Diag:



#### Brief explanation:

The user when runs the software he opens the web page for the first time to run the music player

online. Then he is showed a menu of songs to chose from, the user selects from the menu a song to be played and he click the corresponding play button to play the song. When required the user can switch the song and pause the song and this information is shared back to user if the song is paused or resumed.

**Conclusion:** The dataflow diagram was made successfully by following the steps in the above dataflow diagram

## EXERCISE NO. 5

### Aim:

To draw the Use Case Diagram

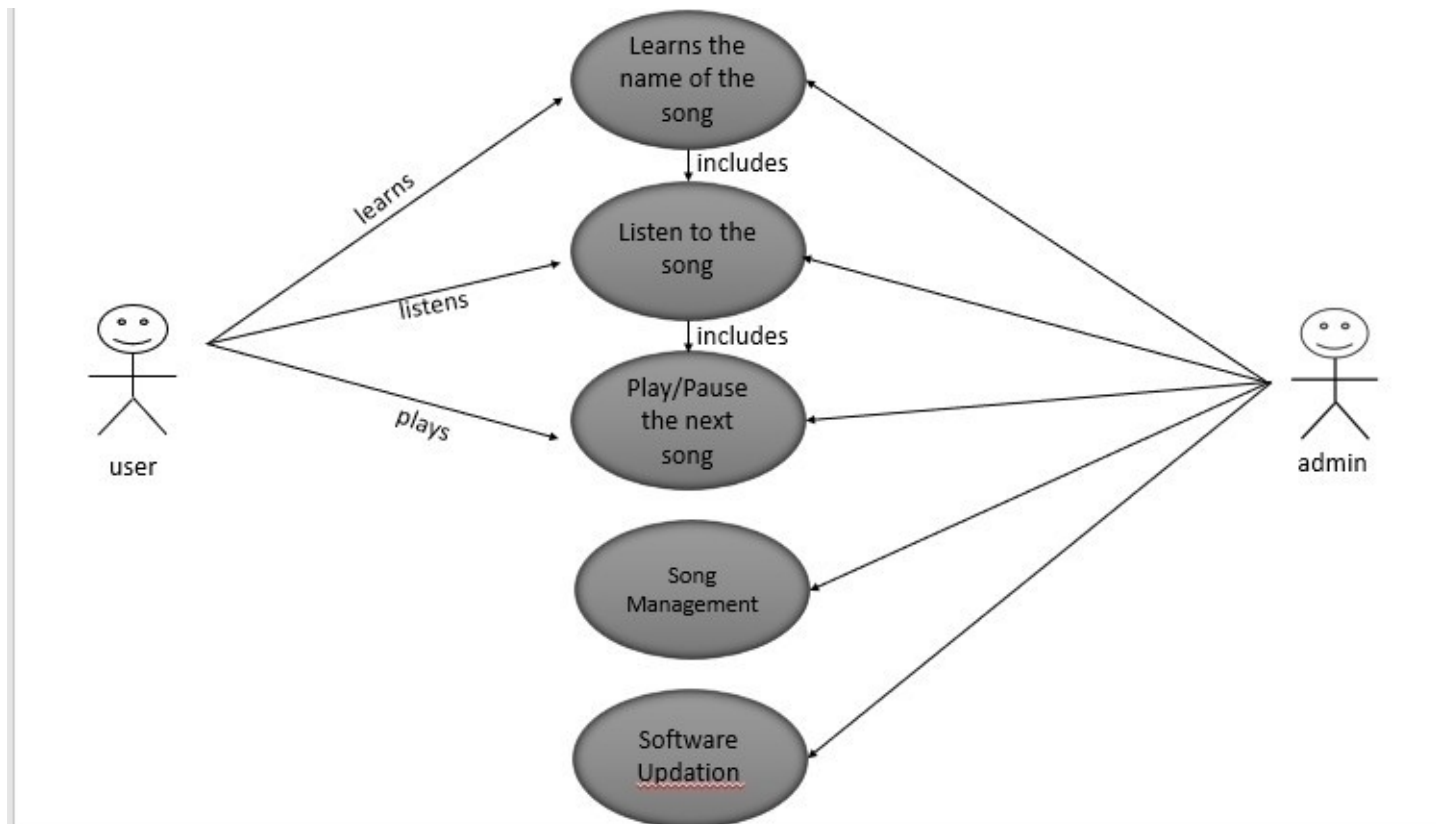
### Hardware Requirements:

- Pentium(R) i5, 2GB RAM.

### Software Requirements:

- Any window-based operating system (Windows 11/10)
- WordPad or Microsoft Word, Visual Studio 2022, Visual Paradigm online

### Use Case Diagram:



## **Use Cases:**

The user see the names of the songs, he then make a judgement to play a particular song and selects the song . He can play/pause the song as required with the click of a button. The user is a actor, organisation, or external system that plays a role in one or more interaction with the system.

The admin only has the access to all the functions of the software that a user has, including that he can also manage the menu of the songs where he can add or delete a song from the database or push a software update with redesigned structure of the application or new features increasing usability.

**Conclusion:** The use case diagram was made successfully by following the steps described above.

## EXERCISE NO. 6

**AIM :-** To draw a sample activity diagram

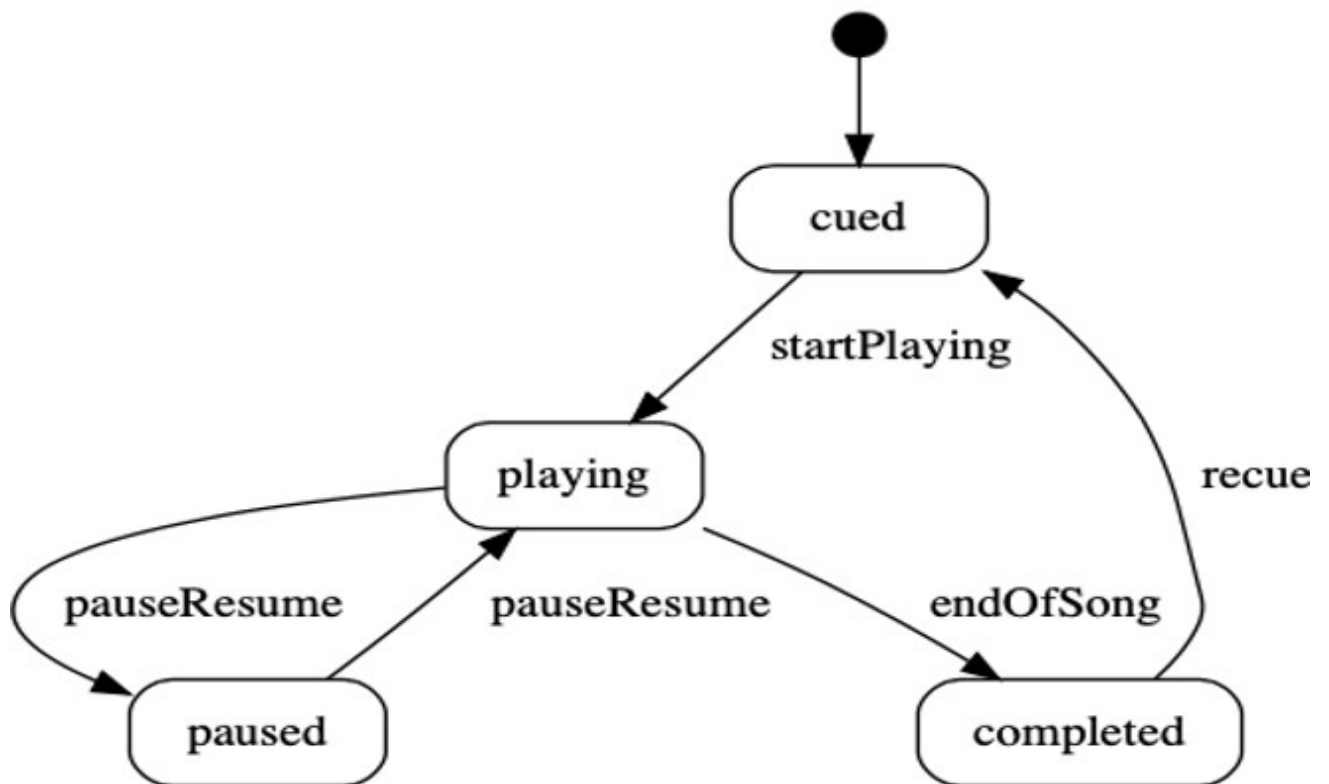
### Hardware Requirements:

- Pentium(R) i5, 2GB RAM.

### Software Requirements:

- Any window-based operating system (Windows 8/10)
- WordPad or Microsoft Word, Visual Studio 2022, Visual Paradigm online

**UML diagram:**



**Conclusion:** The activity diagram was made successfully by following the steps described above.

## EXERCISE NO. 7

**AIM:** To prepare STATE CHART DIAGRAM for any project.

### REQUIREMENTS:

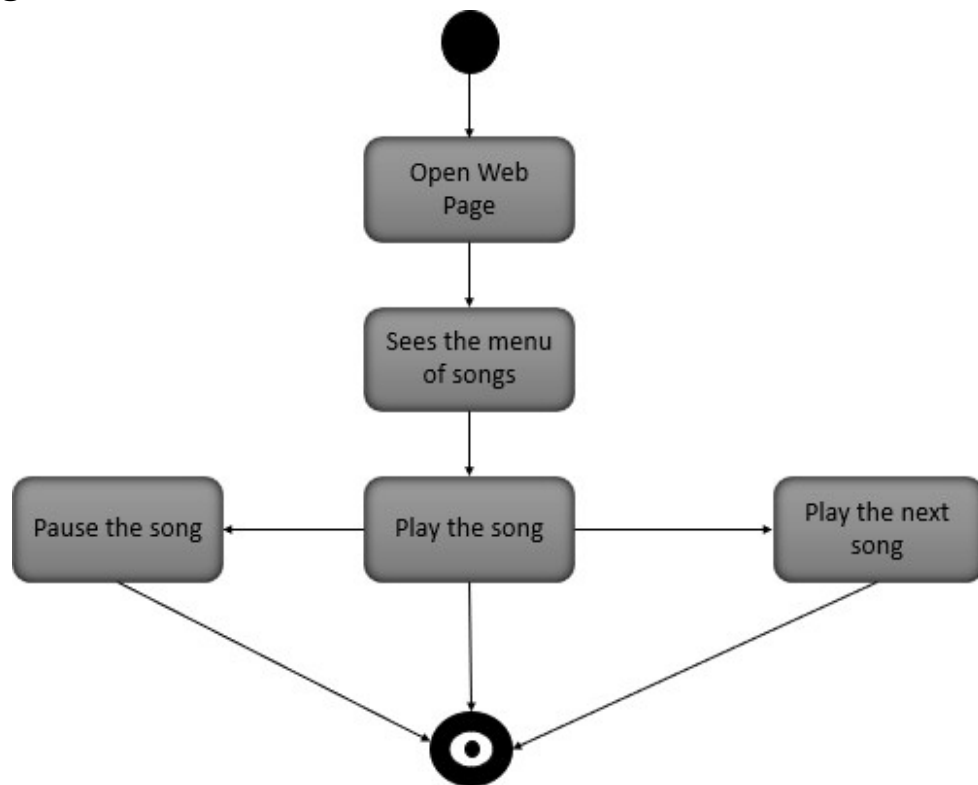
#### Hardware Interfaces

- Pentium(R) i5, 2GB RAM.

#### Software Interfaces

- Any window-based operating system (Windows 11/10)
- WordPad or Microsoft Word, Visual Studio 2022, Visual Paradigm online

### State chart diagram:



**Conclusion:** The state chart diagram was made successfully by following the steps described above.

## EXERCISE NO. 8

### Aim:

To draw the Sequence Diagram

### Hardware Requirements:

- Pentium(R) i5, 2GB RAM.

### Software Requirements:

- Any window-based operating system (Windows 11/10)
- WordPad or Microsoft Word, Visual Studio 2022, Visual Paradigm online

### Sequence diag:



**Conclusion:** The sequence diagram was made successfully by following the steps described above.



## EXERCISE NO. 9

**Aim:**

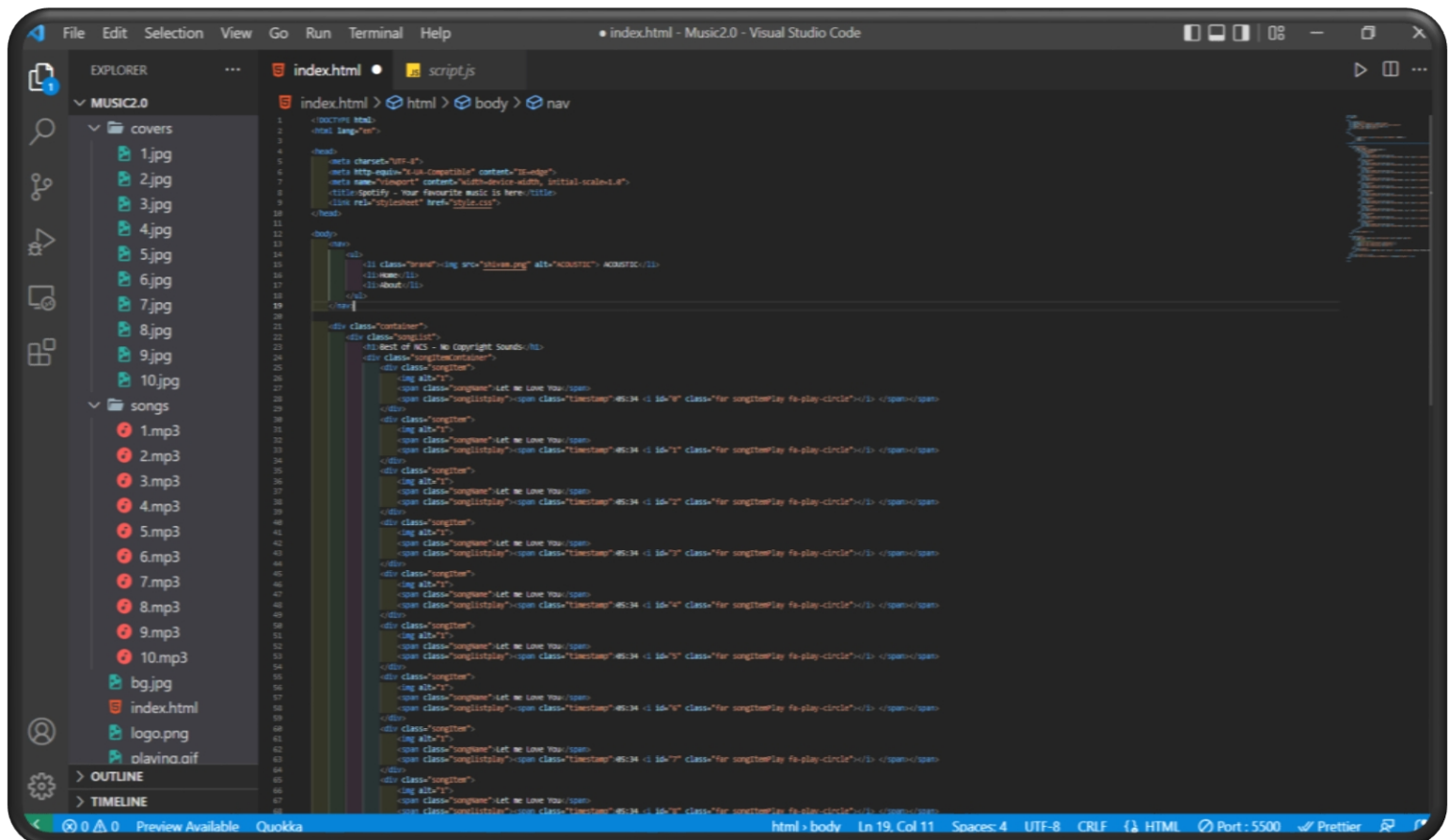
To present the code for your project.

## Hardware Requirements:

- Pentium(R) i5, 2GB RAM.

## Software Requirements:

- Any window-based operating system (Windows 11/10)
- WordPad or Microsoft Word, Visual Studio 2022, Visual Paradigm online



script.js - Music2.0 - Visual Studio Code

EXPLORER

MUSIC2.0

- 5.jpg
- 6.jpg
- 7.jpg
- 8.jpg
- 9.jpg
- 10.jpg
- songs
  - 1.mp3
  - 2.mp3
  - 3.mp3
  - 4.mp3
  - 5.mp3
  - 6.mp3
  - 7.mp3
  - 8.mp3
  - 9.mp3
  - 10.mp3
- bg.jpg
- index.html
- logo.png
- playing.gif
- script.js
- shivam.png
- style.css
- YASH.png

OUTLINE

TIMELINE

```
1 console.log("Welcome to Spotify");
2
3 // Initialize the variables
4 let songIndex = 0;
5 let audioElement = new Audio('songs/1.mp3');
6 let masterPlay = document.getElementById('masterPlay');
7 let myProgressbar = document.getElementById('myProgressbar');
8 let gif = document.getElementById('gif');
9 let masterSongName = document.getElementById('masterSongName');
10 let songItems = Array.from(document.getElementsByClassName('songitem'));
11
12 let songs = [
13   {songName: 'Harriso - Mortals (MC5 Release)', filePath: 'songs/1.mp3', coverPath: 'covers/1.jpg'},
14   {songName: 'Cielo - Huma-Huma', filePath: 'songs/2.mp3', coverPath: 'covers/2.jpg'},
15   {songName: 'Dear Rev - Imincible (MC5 Release) 128k', filePath: 'songs/3.mp3', coverPath: 'covers/3.jpg'},
16   {songName: 'Different women 8 bit02 - My Heart (MC5 Release)', filePath: 'songs/4.mp3', coverPath: 'covers/4.jpg'},
17   {songName: 'Jangli-heros-tonight-feat-Johnnig-MC5-Release', filePath: 'songs/5.mp3', coverPath: 'covers/5.jpg'},
18   {songName: 'Kabele - Salam-e-Ishq', filePath: 'songs/6.mp3', coverPath: 'covers/6.jpg'},
19   {songName: 'Fakhriam - Salam-e-Ishq', filePath: 'songs/7.mp3', coverPath: 'covers/7.jpg'},
20   {songName: 'Whula Dena - Salam-e-Ishq', filePath: 'songs/8.mp3', coverPath: 'covers/8.jpg'},
21   {songName: 'Tunhai Karam - Salam-e-Ishq', filePath: 'songs/9.mp3', coverPath: 'covers/9.jpg'},
22   {songName: 'Ma Jaha - Salam-e-Ishq', filePath: 'songs/10.mp3', coverPath: 'covers/10.jpg'},
23 ]
24
25 songItems.forEach((element, i) => {
26   element.getElementsByTagName("img")[0].src = songs[i].coverPath;
27   element.getElementsByClassName("songname")[0].innerText = songs[i].songName;
28 })
29
30 // Handle play/pause click
31 masterPlay.addEventListener('click', () => {
32   if(audioElement.paused || audioElement.currentTime < 0){
33     audioElement.play();
34     masterPlay.classList.remove('fa-play-circle');
35     masterPlay.classList.add('fa-pause-circle');
36     gif.style.opacity = 1;
37   }
38   else{
39     audioElement.pause();
40     masterPlay.classList.remove('fa-pause-circle');
41     masterPlay.classList.add('fa-play-circle');
42     gif.style.opacity = 0;
43   }
44 })
45
46 // listen to events
47 audioElement.addEventListener('timeupdate', () => {
48   // Update Seekbar
49   progress = parseInt(audioElement.currentTime/audioElement.duration)* 100;
50   myProgressbar.value = progress;
51 })
52
53 myProgressbar.addEventListener('change', () => {
54   audioElement.currentTime = myProgressbar.value * audioElement.duration/100;
55 })
56
57 const makeAllPlays = () => {
58   Array.from(document.getElementsByClassName('songitemplay')).forEach((element) => {
59     element.classList.remove('fa-pause-circle');
60     element.classList.add('fa-play-circle');
61   })
62 }
63
64 Array.from(document.getElementsByClassName('songitemplay')).forEach((element) => {
65   element.addEventListener('click', (e) => {
66     makeAllPlays();
67     songIndex = parseInt(e.target.id);
68     e.target.classList.remove('fa-play-circle');
69   })
70 })
71
72 Array.from(document.getElementsByClassName('songitemplay')).forEach((element) => {
73   element.addEventListener('click', (e) => {
74     makeAllPlays();
75     songIndex = parseInt(e.target.id);
76     e.target.classList.remove('fa-play-circle');
77   })
78 })
79
80 document.getElementById('next').addEventListener('click', () => {
81   if(songIndex < 0){
82     songIndex = 0
83   }
84   else{
85     songIndex += 1;
86   }
87   audioElement.src = 'songs/' + songIndex + '.mp3';
88   masterSongName.innerText = songs[songIndex].songName;
89   audioElement.currentTime = 0;
90   audioElement.play();
91   masterPlay.classList.remove('fa-play-circle');
92   masterPlay.classList.add('fa-pause-circle');
93 })
94
95 document.getElementById('previous').addEventListener('click', () => {
96   if(songIndex < 0){
97     songIndex = 0
98   }
99   else{
100     songIndex -= 1;
101   }
102   audioElement.src = 'songs/' + songIndex + '.mp3';
103   masterSongName.innerText = songs[songIndex].songName;
104   audioElement.currentTime = 0;
105   audioElement.play();
106   masterPlay.classList.remove('fa-play-circle');
107   masterPlay.classList.add('fa-pause-circle');
108 })
109 })
```

script.js - Music2.0 - Visual Studio Code

EXPLORER

MUSIC2.0

- 5.jpg
- 6.jpg
- 7.jpg
- 8.jpg
- 9.jpg
- 10.jpg
- songs
  - 1.mp3
  - 2.mp3
  - 3.mp3
  - 4.mp3
  - 5.mp3
  - 6.mp3
  - 7.mp3
  - 8.mp3
  - 9.mp3
  - 10.mp3
- bg.jpg
- index.html
- logo.png
- playing.gif
- script.js
- shivam.png
- style.css
- YASH.png

OUTLINE

TIMELINE

```
57 const makeAllPlays = () => {
58   Array.from(document.getElementsByClassName('songitemplay')).forEach((element) => {
59     element.classList.remove('fa-pause-circle');
60     element.classList.add('fa-play-circle');
61   })
62 }
63
64 Array.from(document.getElementsByClassName('songitemplay')).forEach((element) => {
65   element.addEventListener('click', (e) => {
66     makeAllPlays();
67     songIndex = parseInt(e.target.id);
68     e.target.classList.remove('fa-play-circle');
69     e.target.classList.add('fa-pause-circle');
70     audioElement.src = 'songs/' + songIndex + '.mp3';
71     masterSongName.innerText = songs[songIndex].songName;
72     audioElement.currentTime = 0;
73     audioElement.play();
74     gif.style.opacity = 1;
75     masterPlay.classList.remove('fa-play-circle');
76     masterPlay.classList.add('fa-pause-circle');
77   })
78 })
79
80 document.getElementById('next').addEventListener('click', () => {
81   if(songIndex < 0){
82     songIndex = 0
83   }
84   else{
85     songIndex += 1;
86   }
87   audioElement.src = 'songs/' + songIndex + '.mp3';
88   masterSongName.innerText = songs[songIndex].songName;
89   audioElement.currentTime = 0;
90   audioElement.play();
91   masterPlay.classList.remove('fa-play-circle');
92   masterPlay.classList.add('fa-pause-circle');
93 })
94
95 document.getElementById('previous').addEventListener('click', () => {
96   if(songIndex < 0){
97     songIndex = 0
98   }
99   else{
100     songIndex -= 1;
101   }
102   audioElement.src = 'songs/' + songIndex + '.mp3';
103   masterSongName.innerText = songs[songIndex].songName;
104   audioElement.currentTime = 0;
105   audioElement.play();
106   masterPlay.classList.remove('fa-play-circle');
107   masterPlay.classList.add('fa-pause-circle');
108 })
109 })
```

## EXERCISE NO. 10

### Aim:

To present the screenshots of working project.

### Hardware Requirements:

- Pentium(R) i5, 4GB RAM.

### Software Requirements:

- Any window-based operating system (Windows 11/10)
- WordPad or Microsoft Word, Visual Studio 2022, Visual Paradigm online

**Working model link:**<https://extraordinary-lily-1d97d2.netlify.app/>

