

Music Library



General Information

Project compound by two parts, one is the part of the search engine, and the other of the results.

Search Engine

The search engine implement the following options:

- **Search songs** by name
- **Search artists** by name
- **Search albums** by name
- **Search music** videos by name

Search engine optional parameters:

- **Filter by country**
 - ❖ Specific third-party API to obtain the necessary information to apply the filter: <https://api.printful.com/countries>
- **Filter by explicit content**
 - ❖ “Yes” or “No” options.
- **Filter by results limit**
 - ❖ Range from 1 to 200.

Results

Depending on the search performed, is possible to see for each item information:

- **For a song:**
 - ❖ Cover
 - ❖ Name of the song
 - ❖ Artist name
 - ❖ Album name
 - ❖ Song Price
 - ❖ Release date
 - ❖ Song length
 - ❖ Musical genre
 - ❖ Audio sample of the song
 - ❖ Song Link in iTunes
- **For an artist:**

- ❖ First name
- ❖ Musical genre
- ❖ Artist link on iTunes
- **For an album:**
 - ❖ Cover
 - ❖ Album name
 - ❖ Artist name
 - ❖ Album price
 - ❖ Number of songs
 - ❖ Release date
 - ❖ Musical genre
- **For a music video:**
 - ❖ Cover
 - ❖ Name of the song
 - ❖ Artist name
 - ❖ Song Price
 - ❖ Release date
 - ❖ Song length
 - ❖ Musical genre
 - ❖ Clip video sample
 - ❖ Link of the music video on iTunes

Project Specifications

- Favorite items should be stored in [localStorage](#), can be deleted from favorites.
- Use **jQuery** for the entire project development
- The web will have to be [responsive](#) and **compatible with the main browsers in the market**
- Any item found by a search can be added to favorites and will be displayed in the favorites section (example in the wireframe)
- Use a code style like camelCase
- Additional search parameters will be optional when performed, may or may not be applied individually (visual example in the wireframe)
- All code included comments need to be write in English
- Implement all functionalities on the same screen dynamically
- Make use of third-party API's to solve the project
- The search must be carried out each time the input changes, that is, each time a character is entered, the page will show the content found.
- You must create a correctly documented README.md file in the root directory of the project (see guidelines in **Resources**)
- Create a clear and orderly directory structure
- For the project documentation a PDF version is required within the repository
- You must capture the functionalities in a [use case diagram](#).

Risk Management Plan

ID	Risk Identification	Risk	Risk Mitigation	Risk Evaluation
1	Healthy	Less productive. Project not finish in the time.	Eat good meals	Medium
2	Hardware	May cause the project not to be completed. Loss of all project data.	Always take care about your hardware. Systematically commit your progress in git.	Low
3	Time management	Bad time management for every task. Project not finish in the time.	Organize the task time synchronously to your knowledge.	High
4	The Design and Construction Process	Bad design can hinder the project implementation process.	Organize tasks with Gantt Chart.	High

Quality Control Measurements

- HTML and CSS validate by W3C
- JavaScript validate by ESLint
- Web application is responsive
- Web application is compatible with main browsers in market
 - ❖ Google Chrome
 - ❖ FireFox
 - ❖ IE
 - ❖ Safari

Tools

- Visual Studio Code
- Git
- ESLint
- Browser Developer Tools
- W3C Validator

Structural Design of the Project

1. Project Architecture Design

The project will contain the following files:

.git/ - git folder

src/

js/ - folder for JavaScript files

css/ - folder for Cascading Style Sheets

Index.html - Main page starter

README.md - Instructions about how to run the project

JAVASCRIPT

2. API.JS

Import elements

```
import Song from './song.js';
import MusicV from './musicV.js';
import display from './display.js';
import Artist from './artist.js';
import Album from './album.js';
import getCountry from './countryAPI.js';
```

EventListener when HTML is ready

Time: 5H

Difficult Level : 3

```
$('#document').ready(function () {
    let selectValu = null;
    let textValu = null;
    let country = null;
    let explicit = null;
    let range = null;
    // Display elements when html is done
    getData("michael", "song", country, explicit, 25);
    getCountry();

    // LIMIT RANGE
    $('#range').change(function () {
        $('#range0').html(this.value);
        $('#conCard').html("");
    });
});
```

```

        getData(textValu,selectValu, country,explicit,this.value);
    });
    // COUNTRY SELECTOR
    $("#sel2").change(function () {
        $("#conCard").html("");

        getData(textValu,selectValu,this.value,explicit,range);

    });
    // EXPLICIT SELECTOR
    $("#sel3").change(function () {
        $("#conCard").html("");

        getData(textValu,selectValu,country,this.value,range);

    });

```

- I create global variables for have always last elements is memory
- I call for functions getdata for display elements when html is ready and getcountry for get date in country selector
- I create limit ranger with method change I display range in html and after change range I clean content for display new.
- I create country selector and I clean content for display new.
- I create explicit selector and I clean content for display new.

Button Search Event

Time: 2H

Difficult Level : 3

```

/ SEARCH BUTTON
$("#btn").click(function () {
    $("#conCard").html("");
    selectValu = $("#sel1").val().toLowerCase();
    textValu = $("#search").val().toLowerCase();
    country = $("#sel2").val();
    explicit = $("#sel3").val();
    range = $("#range").val();

```

```

        // validation
        let val = $("#search").val();
        if (val.length == 0) {
            $("#search").addClass('invalid');
        } else {
            $("#search").removeClass('invalid');
        }

        if (explicit == '--Select--') {
            explicit = 'Yes';
        }
        if (country == '--Select--') {
            country = 'PL';
        }
        if (selectValu == 'song') {
            selectValu = 'song';
        }
        if (selectValu == 'album') {
            selectValu = 'album';
        }
        if (selectValu == 'artist') {
            selectValu = 'musicArtist';
        }
        if (selectValu == 'music video') {
            selectValu = 'musicVideo';
        }
        getData(textValu, selectValu, country, explicit,
range);
        $("#search").val("");
    })
})

```

- I create click event listener I clean content and set attributes to variables.
- I create validation if input is not empty
- I create if conditions for my selectors
- I call function getData with arguments

- I clean input search

Connect to itunes

Time: 4H

Difficult Level : 3

```
function getData(term, entity, country, explicit, range) {  
  $.ajax({  
    url: "https://itunes.apple.com/search?",  
    data: {  
      "explicit": explicit,  
      "limit": range,  
      "country": country,  
      "term": term,  
      "entity": entity,  
    },  
    type: "GET",  
    dataType: "jsonp",  
    success: function (data) {  
      const results = data.results;  
      const resultCount = data.resultCount;  
      console.log(data);  
      getInfoSong(results, entity);  
      getInfoArtist(results, entity);  
      getInfoAlbum(results, entity);  
      getInfoMusicV(results, entity);  
    }  
  })  
};
```

- First step I create function with parameters from itunes, using jQuery and Ajax.
- Second step when connection is success I create methods to each class for get data.

Problems:

- Problem was with dataType : jsonp
- JSON was worrying about cross-domain.

Class Song

Time: 1H

Difficult Level : 2

```
class Song{
```

```

    constructor(artistName, trackName, releaseDate, artworkUrl100) {
        this.artistName = artistName,
        this.trackName = trackName,
        this.releaseDate = releaseDate,
        this.artworkUrl100 = artworkUrl100
    }
}
export default Song;

```

- After connecting to API, I design class song with constructor and parameters what I'm interested :
 1. artistName
 2. trackName
 3. releaseDate
 4. artworkUrl100

Problems:

- Problem was with understanding how to import and export functions and classes between files.

Function getInfoSong

Time: 7H

Difficult Level : 4

```

// Function for get informations about Song.
function getInfoSong(test, entity) {

    if (entity == 'song') {
        test.forEach(function attribute(item) {
            const song = new Song(item.artistName,
                item.trackName,
                item.releaseDate,
                item.artworkUrl100, );
            display.element(song, entity);
        })
    }
}

```

- I create function with parameters test and entity:
 1. Parameter test is for find interesting us object in our case is text from input search.

2. Parameter entity is for choice with group we gonna searching in our case we have 4 groups song,album,artist and music video.
- I create condition if for confirm that entity for sure is 'song'.
 - By method .forEach I iterate all results from api what I get from parameter test.
 - Inside method .forEach I create variable song what is equal to new class Song with 4 items.
 - When I create class Song I gonna send this class with all attributes to class display method element with attributes song and entity.

Problems:

- Problem was with understanding for to send elements to the other class
- Problem was with understanding how to import and export functions and classes between files.

Class Artist

Time: 1H

Difficult Level : 2

```
class Artist{

constructor(artistName,primaryGenreName,artistLinkUrl){
    this.artistName = artistName;
    this.primaryGenreName = primaryGenreName;
    this.artistLinkUrl = artistLinkUrl;
}

}

export default Artist;
```

- I design class Artist with constructor and parameters what I'm interested :
 1. artistName
 2. primaryGenreName
 3. artistLinkUrl

Problems:

Problem was with understanding how to import and export functions and classes between files.

Function getInfoArtist

Time:1H

Difficult Level : 2

```
// Function for get informations about Artist.
function getInfoArtist(test, entity) {
    if (entity == 'musicArtist') {
        test.forEach(function attribute(item) {
            const artist = new Artist(item.artistName,
```

```

        item.primaryGenreName,
        item.artistLinkUrl);
    display.element(artist, entity);
  })
}
}

```

- I create function with parameters test and entity:
 1. Parameter test is for find interesting us object in our case is text from input search.
 2. Parameter entity is for choice with group we gonna searching in our case we have 4 groups song,album,artist and music video.
- I create condition if for confirm that entity for sure is 'musicArtist'.
- By method .forEach I iterate all results from api what I get from parameter test.
- Inside method .forEach I create variable artist what is equal to new class Artist with 3 items.
- When I create class Artist I gonna send this class with all attributes to class display method element with attributes artist and entity.

Problems:

Class Album

Time: 1H

Difficult Level : 2

```

class Album{
  constructor(artistName,trackCount,collectionPrice,releaseDate,primaryGenreName,collectionName,artworkUrl100){
    this.artistName = artistName;
    this.trackCount = trackCount;
    this.collectionName = collectionName;
    this.collectionPrice = collectionPrice;
    this.releaseDate = releaseDate;
    this.primaryGenreName = primaryGenreName;
    this.artworkUrl100 = artworkUrl100;
  }
}
export default Album;

```

- I design class Artist with constructor and parameters what I'm interested :
 1. artistName
 2. primaryGenreName
 3. trackCount
 4. collectionName
 5. collectionPrice
 6. releaseDate
 7. artworkUrl100

Problems:

- Problem was with understanding how to import and export functions and classes between files.

Function getInfoAlbum

Time: 1H

Difficult Level : 2

```
//Function for get information about Album.
function getInfoAlbum(test, entity) {
  if (entity == 'album') {
    test.forEach(function attribute(item) {
      const album = new Album(item.artistName,
        item.trackCount,
        item.collectionPrice,
        item.releaseDate,
        item.primaryGenreName,
        item.collectionName,
        item.artworkUrl100);
      display.element(album, entity);
    })
  }
}
```

- I create function with parameters test and entity:
 1. Parameter test is for find interesting us object in our case is text from input search.
 2. Parameter entity is for choice with group we gonna searching in our case we have 4 groups song,album,artist and music video.
- I create condition if for confirm that entity for sure is 'album'.
- By method .forEach I iterate all results from api what I get from parameter test.
- Inside method .forEach I create variable album what is equal to new class Album with 7 items.
- When I create class Album I gonna send this class with all attributes to class display method element with attributes artist and entity.

Problems:

Class MusicV

Time: 1H

Difficult Level : 2

```
class MusicV {

  constructor(artworkUrl100,artistName,trackName,trackTimeMillis,releaseDate,primaryGenreName,trackPrice,trackViewUrl,previewUrl) {
    this.artworkUrl100 = artworkUrl100;
  }
}
```

```

this.artistName = artistName;
this.trackName = trackName;
this.trackTimeMillis = trackTimeMillis;
this.releaseDate = releaseDate;
this.primaryGenreName = primaryGenreName;
this.trackPrice = trackPrice;
this.trackViewUrl = trackViewUrl;
this.previewUrl = previewUrl;
}
}
export default MusicV;

```

- I design class Artist with constructor and parameters what I'm interested :

1. artistName
2. artworkUrl100
3. trackName
4. trackTimeMillis
5. releaseDate
6. primaryGenreName
7. trackPrice
8. trackViewUrl
9. previewUrl

Problems:

- Problem was with understanding how to import and export functions and classes between files.

Function getInfoMusicV

Time: 1H

Difficult Level : 2

```

//Function for get information about music video.
function getInfoMusicV(test, entity) {
  if (entity === 'musicVideo') {
    test.forEach(function attribute(item) {
      const musicV = new MusicV(item.artworkUrl100,
        item.artistName,
        item.trackName,
        item.trackTimeMillis,
        item.releaseDate,
        item.primaryGenreName,
        item.trackPrice,
        item.trackViewUrl,
        item.previewUrl);
      display.element(musicV, entity);
    });
  }
}

```

```

    })
  }
}

```

- I create function with parameters test and entity:
 1. Parameter test is for find interesting us object in our case is text from input search.
 2. Parameter entity is for choice with group we gonna searching in our case we have 4 groups song,album,artist and music video.
- I create condition if for confirm that entity for sure is 'musicVideo'.
- By method .forEach I iterate all results from api what I get from parameter test.
- Inside method .forEach I create variable musicV what is equal to new class MusicV with 9 items.
- When I create class MusicV I gonna send this class with all attributes to class display method element with attributes artist and entity.

Problems:

3. countryAPI.js

Connect to CountryAPI

Time: 2H

Difficult Level : 3

```

import Country from './country.js';
export default function getCountry() {
$.ajax({
  "async": true,
  "crossDomain": true,
  "url": "https://restcountries-v1.p.rapidapi.com/all",
  "method": "GET",
  "headers": {
    "x-rapidapi-host": "restcountries-v1.p.rapidapi.com",
    "x-rapidapi-key":
"5tCHtG27Whmsh1VlJXVPtO5J3jslp1VU3bQjsnuIbIJMIznue0"
  },
  type: "GET",
  dataType: "json",
  success: function (data) {
    getCountryName(data);
  }
})
}

```

- I create function getCountry then I connect by ajax to url of API.

- I create method getCountryName with parameter data from API.
- I export function getCountry to api.js.
- I import class Country from ./country.js

Class MusicV

Time: 1H

Difficult Level : 2

```
class Country {
  constructor(name, nameIso) {
    this.name = name;
    this.nameIso = nameIso;
  }
}
export default Country;
```

- I create class Country with constructor of 2 parameters
- I export my Class to countryAPI.js

Function getCountryName

Time: 2H

Difficult Level : 3

```
function getCountryName(name) {
  name.forEach(function name(item) {
    const country = new Country(item.name, item.alpha2Code);
    let arrayCountry = [];
    arrayCountry.push(country);
    pushName(arrayCountry);
  })
}
```

- I create function with parameter name:
- By method .forEach I iterate all results from api what I get from parameter name.
- Inside method .forEach I create variable country what is equal to new class Country with 2 items.
- I create Array and I push my Country class to array.
- I create callback pushName with argument arrayCountry.

Function getCountryName

Time: 2H

Difficult Level : 3

```
function pushName(name) {
  name.forEach(function push(item) {
```

```

let ele = $("

```

- I create function pushName with parameter name what is my arrayCountry from function getCountryName
- I iterate parameter name by method forEach, I create new variable ele for create tags <option> with country name.
- I append my tags <option> to selector "#sel2" in index.html.
- I give to each option vale of short name country like "PL".

4. DISPLAY.JS

Class Display

Time: 2H

Difficult Level : 2

```

let i=0;
class Display{
  element(element,entity){
    //console.log(entity);
    if(entity == 'song'){
      songCard(element,entity);
    }
    if(entity == 'album'){
      albumCard(element,entity);
    }
    if(entity == 'musicArtist'){
      artistCard(element,entity);
    }
    if(entity == 'musicVideo'){
      videoCard(element,entity);
    }
  }
}

```

```

    }
  }
}
let display = new Display;

export default display;

```

- I create class Display with method element with 2 parameters element and entity
- I use conditions if for display element by entity attribute.
- Each condition call a function.
- I create variable i like a counter.
- I create variable display like a new class
- I export default my variable display to api.js

Function songCard

Time: 4H

Difficult Level : 3

```

function songCard(element,entity) {

  let div = $("<div>",{id:"card",class:"card",style:"width:400px;"});

  let row = $("<div>",{id:"row",class:"p-2 d-flex align-content-center flex-wrap row no-gutters"});

  let divCol = $("<div>",{id:"divCol",class:"col-lg-4"});

  let divimg = $("<img>",{class:"p-2 card-img",src:element.artworkUrl100});

  let divCardCont =
  $("<div>",{id:"cardCont",class:"justify-content-center col-lg-8"});

  let divCardBody = $("<div>",{id:"cardBody",class:"card-body"});

  let cardTitle =
  $("<h5>",{id:"cardT",class:"card-title",style:"text-align:center;"}).html(element.artistName);

```



```

    let cardInfo =
    $("<p>", {class:"card-text", style:"text-align:center;font-size:16px"}).html(element.trackName);

    let cardDate =
    $("<p>", {class:"card-date", style:"text-align:center;font-size:14px"}).html('Release Date: '+element.releaseDate.substring(0,10));

    let btn = $("<button>", {id:"fbtn", class:"btn-fav btn mt-auto btn-primary"}).attr('t',i).attr("entity",entity).text("Favorite");

    //one way is to set the name of the button that youre clicking the artist name and get this attr later when you try to add to favourites

    //the better way is to get the sibling div element value of the button that youre clicking. You can search for term childNodes() that is vanilla js or siblings() from jquery

    $("#conCard").append(div);

    $(div).append(row);

    $(row).append(divCol);

    $(divCol).append(divimg);

    $(row).append(divCardCont);

    $(divCardCont).append(divCardBody);

    $(divCardBody).append(cardTitle);

    $(divCardBody).append(cardInfo);

    $(divCardBody).append(cardDate);

    $(div).append(btn);

    i++;
}

```

- I create function songCard with two parameters element and entity
- I create DOM html for display dynamic content with attribute of element
- For each display i++;
- I create button favorite with attribute of variable i and entity.

Function albumCard

Time: 4H

Difficult Level : 3

```
function albumCard(element ,entity){
  let div = $("<div>",{id:"card",class:"card m-2",style:"width:400px;"});
  let row = $("<div>",{id:"row",class:"justify-content-center row"});
  let divCol = $("<div>",{id:"divCol",class:"mt-5 col-lg-4"});
  let divimg =
  $("<img>",{id:"cardImg",class:"card-img",src:element.artworkUrl100});
  let divCardCont = $("<div>",{id:"cardCont",class:"p-2 col-lg-6"});
  let divCardBody = $("<div>",{id:"cardBody",class:"card-body"});
  let cardTitle =
  $("<h5>",{id:"titleCard",class:"card-title"}).html(element.artistName);
  let collectionName =
  $("<p>",{id:"collection",class:"card-text"}).html(element.collectionName);
  let collectionPrice = $("<p>",{id:"collection",class:"card-text card-price"}).html('Price: '+element.collectionPrice+'$');
  let collectionGenere = $("<p>",{id:"collection",class:"card-text card-type"}).html('Type: '+element.primaryGenreName);
  let trackInfo = $("<p>",{id:"cardInfo",class:"card-text num-track"}).html('Number of track: '+element.trackCount);
  let cardDate = $("<p>",{id:"cardDate",class:"card-text card-date"}).html('Release Date: '+element.releaseDate.substring(0,10));
  let btn = $("<button>",{id:"fbtn",class:"btn-fav btn mt-auto btn-primary"}).attr('t',i).attr("entity",entity).text("Favorite");
  $("#conCard").append(div);
  (div).append(row);
  (row).append(divCol);
  (divCol).append(divimg);
  (row).append(divCardCont);
  (divCardCont).append(divCardBody);
```

```

$(divCardBody).append(cardTitle);
$(divCardBody).append(collectionName);
$(divCardBody).append(trackInfo);
$(divCardBody).append(collectionPrice);
$(divCardBody).append(collectionGenere);
$(divCardBody).append(cardDate);
$(div).append(btn);
i++;
}

```

- I create function albumCard with two parameters element and entity
- I create DOM html for display dynamic content with attribute of element
- For each display i++;
- I create button favorite with attribute of variable i and entity.

Function artistCard

Time: 4H

Difficult Level : 3

```

function artistCard(element,entity){
  let div = $("<div>",{id:"card",class:" card m-2",style:"width:400px"});
  let row = $("<div>",{id:"row",class:"justify-content-center row"});
  let divCardCont = $("<div>",{id:"cardCont",class:"p-2 col-lg-6"});
  let divCardBody = $("<div>",{id:"cardBody",class:"card-body"});
  let cardTitle =
  $("<h5>",{id:"titleCard",class:"card-title"}).html(element.artistName);
  let cardInfo = $("<p>",{id:"cardInfo",class:"card-text card-type"}).html(element.primaryGenreName);
  let smallDate =
  $("<a>",{id:"smallDate",class:"url"}).html(element.artistLinkUrl);
  let btn = $("<button>",{id:"fbtn",class:"btn-fav btn mt-auto btn-primary"}).attr('t',i).attr("entity",entity).text("Favorite");

  $("#conCard").append(div);
  $(div).append(row);
  $(row).append(divCardCont);
  $(divCardCont).append(divCardBody);
  $(divCardBody).append(cardTitle);
  $(divCardBody).append(cardInfo);

```

```

$(divCardBody).append(smallDate);
$(div).append(btn);
i++;
}

```

- I create function artistCard with two parameters element and entity
- I create DOM html for display dynamic content with attribute of element
- For each display i++;
- I create button favorite with attribute of variable i and entity.

Function videoCard

Time: 4H

Difficult Level : 3

```

function videoCard(element,entity){
  let div = $("<div>",{id:"card",class:" card m-2",style:"width:400px;"});
  let row = $("<div>",{id:"row",class:"justify-content-center row"});
  let divCol = $("<div>",{id:"divCol",class:"mt-5 col-lg-4"});
  let divimg =
  $("<img>",{id:"cardImg",class:"card-img",src:element.artworkUrl100});
  let divCardCont = $("<div>",{id:"cardCont",class:"p-2 col-lg-6"});
  let divCardBody = $("<div>",{id:"cardBody",class:"card-body"});
  let cardTitle =
  $("<h5>",{id:"titleCard",class:"card-title"}).html(element.artistName);
  let videoName =
  $("<p>",{id:"collection",class:"card-text"}).html(element.trackName);
  let trackPrice = $("<p>",{id:"collection",class:"card-text card-price"}).html('Price: '+element.trackPrice+'$');
  let collectionGenere = $("<p>",{id:"collection",class:"card-text card-type"}).html('Type: '+element.primaryGenreName);
  let trackUrl = $("<p>",{id:"cardInfo",class:"card-text card-url"}).html('Url: '+element.trackViewUrl);
  let cardDate = $("<p>",{id:"cardDate",class:"card-text card-date"}).html('Release Date: '+element.releaseDate.substring(0,10));
  let btn = $("<button>",{id:"fbtn",class:"btn-fav btn mt-auto btn-primary"}).attr('t',i).attr("entity",entity).text("Favorite");

  $("#conCard").append(div);
  $(div).append(row);
}

```

```

$(row).append(divCol);
$(divCol).append(divimg);
$(row).append(divCardCont);
$(divCardCont).append(divCardBody);
$(divCardBody).append(cardTitle);
$(divCardBody).append(videoName);
$(divCardBody).append(trackUrl);
$(divCardBody).append(trackPrice);
$(divCardBody).append(collectionGenere);
$(divCardBody).append(cardDate);
$(div).append(btn);
i++;
}

```

- I create function videoCard with two parameters element and entity
- I create DOM html for display dynamic content with attribute of element
- For each display i++;
- I create button favorite with attribute of variable i and entity.

Button Favorite Event

Time: 14H

Difficult Level : 5

```

//##### BUTTON FAVORITE CLICK #####
$(document).on('click', ".btn-fav", function() {
    let t = $(this).attr('t');
    let entity = $(this).attr('entity');
    // IF ENTITY IS SONG
    if(entity == 'song'){
        let items = {
            title : $(this).siblings().find(".card-title").html(),
            track : $(this).siblings().find(".card-text").html(),
            date : $(this).siblings().find(".card-date").html(),
        }
        //NUMBER OF COUNTER i
        let we = ("items"+t);
        //LOCALSOTORAGE
        localStorage.setItem(we, JSON.stringify(items));
        //GET LOCALSTORAGE
        let ite = JSON.parse(localStorage.getItem(we));
        //CALLBACK
    }
}

```

```

        setItem(ite,t,entity);
    }

    // IF ENTITY IS ALBUM
    if(entity == 'album'){
        let items = {
            title : $(this).siblings().find(".card-title").html(),
            date : $(this).siblings().find(".card-date").html(),
            price : $(this).siblings().find(".card-price").html(),
            type : $(this).siblings().find(".card-type").html(),
            numTrack : $(this).siblings().find(".num-track").html(),
        }

        //NUMBER OF COUNTER i
        let we = ("items"+t);
        //LOCALSOTORAGE
        localStorage.setItem(we,JSON.stringify(items));
        //GET LOCALSTORAGE
        let ite = JSON.parse(localStorage.getItem(we));
        //CALLBACK
        setItem(ite,t,entity);
    }

    // IF ENTITY IS musicArtist
    if(entity == 'musicArtist'){
        let items = {
            title : $(this).siblings().find(".card-title").html(),
            type : $(this).siblings().find(".card-type").html(),
            url : $(this).siblings().find(".url").html(),
        }

        //NUMBER OF COUNTER i
        let we = ("items"+t);
        //LOCALSOTORAGE
        localStorage.setItem(we,JSON.stringify(items));
        //GET LOCALSTORAGE
        let ite = JSON.parse(localStorage.getItem(we));
        //CALLBACK
        setItem(ite,t,entity);
    }

    // IF ENTITY IS musicVideo
    if(entity == 'musicVideo'){
        let items = {
            title : $(this).siblings().find(".card-title").html(),

```

```

        trackName : $(this).siblings().find(".card-text").html(),
        price : $(this).siblings().find(".card-price").html(),
        type : $(this).siblings().find(".card-type").html(),
        url : $(this).siblings().find(".card-url").html(),
        date : $(this).siblings().find(".card-date").html(),
    }
    //NUMBER OF COUNTER i
    let we = ("items"+t)
    //LOCALSOTORAGE
    localStorage.setItem(we,JSON.stringify(items));
    //GET LOCALSTORAGE
    let ite = JSON.parse(localStorage.getItem(we));
    //CALLBACK
    setItem(ite,t,entity);
    }
});

```

- I create if condition for check which entity is for create different DOM html to each condition
- I create object items with attributes I use method siblings to get attributes from nodes
- I get attr from button what is my counter i and entity
- I use counter i for localStorage name of items
- I get items from localStorage and I send to callback setItem with my i counter and entity

Problems:

- Big problems was with get attributes from dynamic HTML elements and send to localStorage
- Big problems was with get attributes from localStorage and display them only one time after click the button.

Function setItem

Time: 10H

Difficult Level : 5

```

// FUNCTION SETITEM IS FOR DISPLAY DYNAMIC CONTENT OF FAVORITES
function setItem(items,t,entity){
    if(entity == 'song'){
        let div = $("<div>",{id:"favDiv"+t,class:"card d-flex justify-content-center m-2 sticky-top",style:"width:300px"});

```

```

        let ul = $("<ul>", {id:"favUl",class:"list-group
list-group-flush d-flex justify-content-center m-0
p-0",style:"min-height:150px"});

        let title = $("<li>", {id:"fav0",class:"tes
list-group-item",}).text(items.title);

        let track =
$("<li>", {id:"fav1",class:"list-group-item",}).text(items.track);

        let date =
$("<li>", {id:"fav2",class:"list-group-item",}).text(items.date);

        let btnX =
$("<button>", {id:"btnX",type:"button",class:"close",style:"width:50px"}
).attr('i',t);

        let span =
$("<span>", {id:"spanBtn",}).attr("aria-hidden","true").text("x");

        if($("#favDiv"+t).length);
    else{
        $("#favtest").append(div);
        $(title).append(btnX);
        $(btnX).append(span);
        $(div).append(ul);
        $(ul).append(title);
        $(ul).append(track);
        $(ul).append(date);
    }
}

if(entity == 'album'){
    let div = $("<div>", {id:"favDiv"+t,class:"card d-flex
justify-content-center m-2 sticky-top",style:"width:300px"});

    let ul = $("<ul>", {id:"favUl",class:"list-group
list-group-flush d-flex justify-content-center m-0
p-0",style:"min-height:150px"});

    let title = $("<li>", {id:"fav0",class:"tes
list-group-item",}).text(items.title);

    let track =
$("<li>", {id:"fav1",class:"list-group-item",}).text(items.numTrack);

    let price =
$("<li>", {id:"fav1",class:"list-group-item",}).text(items.price);

    let type=
$("<li>", {id:"fav1",class:"list-group-item",}).text(items.type);

```



```

        let date =
$(("<li>",{id:"fav2",class:"list-group-item"}).text(items.date);
        let btnX =
$(("<button>",{id:"btnX",type:"button",class:"close",style:"width:50px"}
).attr('i',t);
        let span =
$(("<span>",{id:"spanBtn"}).attr("aria-hidden","true").text("x"));
        if($("#favDiv"+t).length);
    else{
        $("#favtest").append(div);
        $(title).append(btnX);
        $(btnX).append(span);
        $(div).append(ul);
        $(ul).append(title);
        $(ul).append(price);
        $(ul).append(track);
        $(ul).append(type);
        $(ul).append(date);
    }
}

if(entity == 'musicArtist'){
    let div = $("<div>",{id:"favDiv"+t,class:"card d-flex
justify-content-center m-2 sticky-top",style:"width:300px"});
    let ul = $("<ul>",{id:"favUl",class:"list-group
list-group-flush d-flex justify-content-center m-0
p-0",style:"min-height:150px"});
    let title = $("<li>",{id:"fav0",class:"tes
list-group-item",}).text(items.title);
    let type =
$(("<li>",{id:"fav1",class:"list-group-item"}).text(items.type);
    let url =
$(("<li>",{id:"fav1",class:"list-group-item"}).text(items.url);
    let btnX =
$(("<button>",{id:"btnX",type:"button",class:"close",style:"width:50px"}
).attr('i',t);
    let span =
$(("<span>",{id:"spanBtn"}).attr("aria-hidden","true").text("x"));
    if($("#favDiv"+t).length);
    else{
        $("#favtest").append(div);

```

```

        $(title).append(btnX);
        $(btnX).append(span);
        $(div).append(ul);
        $(ul).append(title);
        $(ul).append(type);
        $(ul).append(url);
    }
}

if(entity == 'musicVideo'){
    let div = $("<div>",{id:"favDiv"+t,class:"card d-flex justify-content-center m-2 sticky-top",style:"width:300px"});
    let ul = $("<ul>",{id:"favUl",class:"list-group list-group-flush d-flex justify-content-center m-0 p-0",style:"min-height:150px"});
    let title = $("<li>",{id:"fav0",class:"list-group-item",}).text(items.title);
    let type = $("<li>",{id:"fav1",class:"list-group-item"}).text(items.type);
    let trackName = $("<li>",{id:"fav1",class:"list-group-item"}).text(items.trackName);
    let price = $("<li>",{id:"fav1",class:"list-group-item"}).text(items.price);
    let date = $("<li>",{id:"fav1",class:"list-group-item"}).text(items.date);
    let url = $("<li>",{id:"fav1",class:"list-group-item"}).text(items.url);
    let btnX = $("<button>",{id:"btnX",type:"button",class:"close",style:"width:50px"}).attr('i',t);
    let span = $("<span>",{id:"spanBtn"}).attr("aria-hidden","true").text("x");
    if($("#favDiv"+t).length);
    else{
        $("#favtest").append(div);
        $(title).append(btnX);
        $(btnX).append(span);
        $(div).append(ul);
        $(ul).append(title);
        $(ul).append(trackName);
        $(ul).append(price);
    }
}

```

```

        $(ul).append(type);
        $(ul).append(url);
        $(ul).append(date);

    }

}

}

```

- I create setItem function with if condition to each condition is different DOM html.
- I create another if condition with ID of my div in index.html for eliminate problem with duplicate elements in favorite list

Problems:

- Problem with duplicate elements in favorite list I spend more than 6 hours for solved problem
- Problem is with musicArtist content is grow up with favorite list. Time don't let me solved a problem.

Button Close Event

Time: 15min

Difficult Level : 1

```

$(document).on("click", ".close", function() {
    let i = $(this).attr("i");
    $("#favDiv"+i).remove();
})

```

- I create event click for button close
- I get attribute from button "i" counter
- After click remove all div