Coursework Report

HONG HONG
40333300@napier.ac.uk
Edinburgh Napier University - Advance Web Technology (SET09103)

Abstract

In order to test to test the proficiency in the use of python, html, css and javascript. A music website was planned to be established. This report will introduce frame of this website at first. Then it will describe the codes of each interface which compose this website. Finally, this report will sum up problems the web-app has and the solutions.

Keywords – web, Python, CSS, JavaScript, Music, HTML

1 Introduction

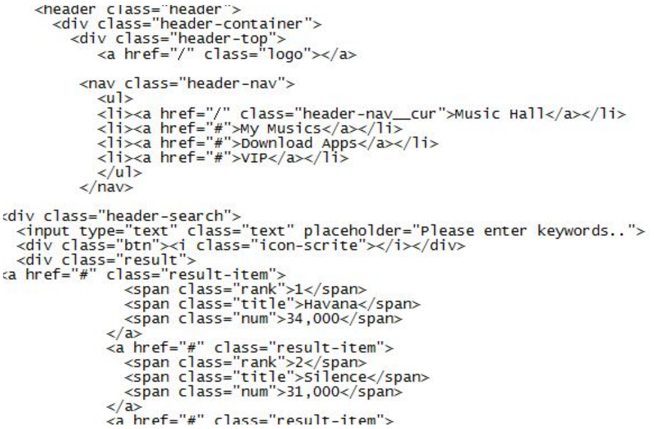
Referencing A complete music website should have several features:1.It must be able to collect recent hottest songs and have a big enough database of different types of music materials. 2.It should have a database to save users' information. Some prime examples of this is the user's search and download history, the music website can recommend based on these history. 3.It should has a simple and beautiful appearance. And each part need a striking title to help people find what they want.
At present, the music website has implemented part those features. It has clear classifications and beautiful appearance. A complete music list page and music video page have been built. It also has a login interface. The rest part can not be implemented because it does not use the database technology.
Here, we describe the processes of establishing this website and the main codes.

2 Frame

First of all, a frame of this web-app had been built. It contains a navigation bar, new songs, recommendation, leader boards, hot lists and premier mv. The navigation bar has contained most features a music website need. It has a logo at the head and it link to the home page. There are seven class had been put in the navigation bar,Music Hall, My Music, Download Apps, VIP, search box, Login and payment interface.

2.1 Search and Login

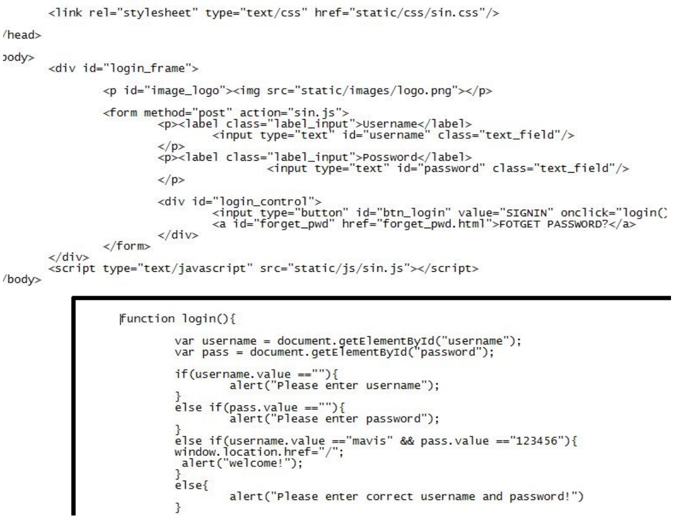
Among them, only the search box and login interface have some functions. The search box will show some search history when the cursor click on, the history includes the name of the song and the times it is searched.



```
    <header class="header">
      <div class="header-container">
        <div class="header-top">
           <a href="/" class="logo"></a>

         <nav class="header-nav">
           <ul>
           <li><a href="/" class="header-nav__cur">Music Hall</a></li>
           <li><a href="#">My Musics</a></li>
           <li><a href="#">Download Apps</a></li>
           <li><a href="#">VIP</a></li>
           </ul>
         </nav>

<div class="header-search">
  <input type="text" class="text" placeholder="Please enter keywords..">
  <div class="btn"><i class="icon-scrite"></i></div>
  <div class="result">
<a href="#" class="result-item">
             <span class="rank">1</span>
             <span class="title">Havana</span>
             <span class="num">34,000</span>
           </a>
           <a href="#" class="result-item">
             <span class="rank">2</span>
             <span class="title">Silence</span>
             <span class="num">31,000</span>
           </a>
           <a href="#" class="result-item">
```

Figure 1: **Navbar** - Part of the Code of navbar

The Login interface will link to the login page. The login page has a logo and two ¡div class¿ to build frame. First ¡div class¿ is for two sets of label and input text, for users to enter their username and password. The second ¡div class¿ is involved a "login" button and a href. After the establishment of frame, this page also has a css file and a js file to decorate and control. In the js file, the correct value of username and password had been set. After users enter the username and password into, it will be compared with the correct value. If they are the same, the website will show 'welcome' and link to the home page. If they are different, the website will request users to enter correct username and password.



```
        <link rel="stylesheet" type="text/css" href="static/css/sin.css"/>
/head>
body>
        <div id="login_frame">

                <p id="image_logo"><img src="static/images/logo.png"></p>

                <form method="post" action="sin.js">
                        <p><label class="label_input">Username</label>
                                <input type="text" id="username" class="text_field"/>
                        </p>
                        <p><label class="label_input">Possword</label>
                                        <input type="text" id="password" class="text_field"/>
                        </p>

                        <div id="login_control">
                                <input type="button" id="btn_login" value="SIGNIN" onclick="login(
                                <a id="forget_pwd" href="forget_pwd.html">FOTGET PASSWORD?</a>
                        </div>
                </form>
        </div>
        <script type="text/javascript" src="static/js/sin.js"></script>
/body>
```

```
function login(){

        var username = document.getElementById("username");
        var pass = document.getElementById("password");

        if(username.value ==""){
                alert("Please enter username");
        }
        else if(pass.value ==""){
                alert("Please enter password");
        }
        else if(username.value =="mavis" && pass.value =="123456"){
        window.location.href="/";
         alert("welcome!");
        }
        else{
                alert("Please enter correct username and password!")
        }
```

Figure 2: **Login** - Part of the Code of login

2.2 The error page

Each website need a 404 error page to face the wrong route. This page had been designed like a no signal television. A strong '404' will move around the centre of the page when the cursor moved. It also has a 'HOME' button in the upper left corner, users can come back to the home page by clicking this button.

Listing 1: Hello World! in

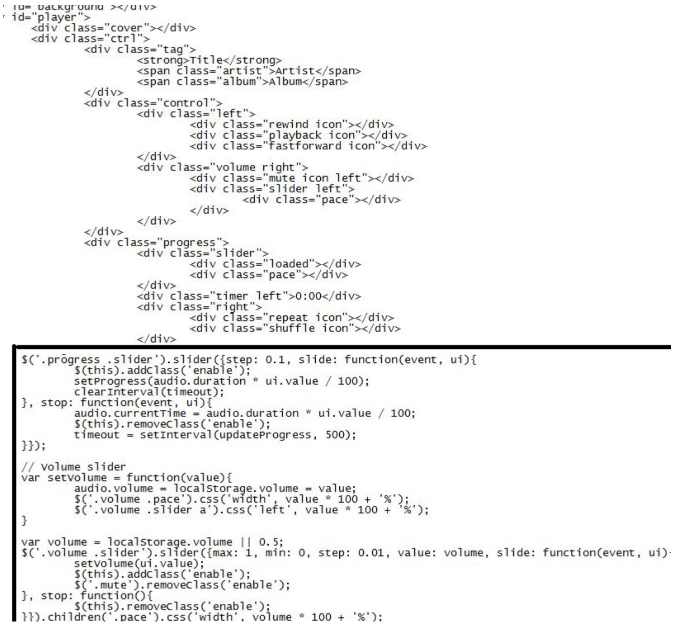
```
for (let pixel = 0; pixel < TOTAL; pixel++) {
  const X = pixel % WIDTH;
  const Y = Math.floor(pixel / WIDTH);
  const generateColor = () => {
    let base = Math.floor(Math.random() * 255).toString(16);
    return `#${base}${base}${base}`;
  };
  const color = generateColor();
  // CONTEXT.fillStyle = `#${Math.floor(Math.random() * 16777215).toString(16)}`
  CONTEXT.fillStyle = color;
  CONTEXT.fillRect(X, Y, 1, 1);
}

BODY.style.background = `url(${CANVAS.toDataURL()})`;

const update = () => {
  const X = Math.floor(Math.random() * WIDTH);
  const Y = Math.floor(Math.random() * HEIGHT);
  BODY.style.backgroundPosition = `${X}px ${Y}px`;
  RAF(update);
};
RAF(update);

const root = document.documentElement;

const move = e => {
  if (e.acceleration && e.acceleration.x !== null) {
    root.style.setProperty("--translateX", e.acceleration.x);
    root.style.setProperty("--translateY", e.acceleration.y);
  } else {
    root.style.setProperty("--translateX", e.pageX / innerWidth - 0.5);
    root.style.setProperty("--translateY", e.pageY / innerHeight - 0.5);
  }
};
```

2.3 List and video player page

There are 'List' and 'MV' components on the home page. And they are linked to a list page and video player page respectively. The list page has four classes for different functions. It used css to designed and be able to show the title, artist, album and picture of the song. And some necessary icons had been used. Those icons can control to pause and continue, adjust the process and volume by corresponding javascript file. There are several songs in the list, users can select which thy like. Similarly, the video player page has a video play box, and some icons work for it. The video is set to play at start time, and it can also choose to pause or continue, and adjust the process and volume.

3 Conclusion

In conclusion, this website has implemented some necessary functions. It has a simple and beautiful home page. And each part need a striking title to help people find what they want. The login page has implemented the basic functions, it can judge the username and password users entered are correct or incorrect and show corresponding text. The error page can let people know that they have entered wrong route directly. And it has a 'HOME' button to help people return to the home page. The music list and video player page has achieved the most important functions. They are able to control to play or pause. It can also adjust the progress and volume of music or video.



```
  id="background"></div>
  id="player">
    <div class="cover"></div>
    <div class="ctrl">
            <div class="tag">
                    <strong>Title</strong>
                    <span class="artist">Artist</span>
                    <span class="album">Album</span>
            </div>
            <div class="control">
                    <div class="left">
                            <div class="rewind icon"></div>
                            <div class="playback icon"></div>
                            <div class="fastforward icon"></div>
                    </div>
                    <div class="volume right">
                            <div class="mute icon left"></div>
                            <div class="slider left">
                                    <div class="pace"></div>
                            </div>
                    </div>
            </div>
            <div class="progress">
                    <div class="slider">
                            <div class="loaded"></div>
                            <div class="pace"></div>
                    </div>
                    <div class="timer left">0:00</div>
                    <div class="right">
                            <div class="repeat icon"></div>
                            <div class="shuffle icon"></div>
                    </div>
```

```
$('.progress .slider').slider({step: 0.1, slide: function(event, ui){
        $(this).addClass('enable');
        setProgress(audio.duration * ui.value / 100);
        clearInterval(timeout);
}, stop: function(event, ui){
        audio.currentTime = audio.duration * ui.value / 100;
        $(this).removeClass('enable');
        timeout = setInterval(updateProgress, 500);
}});

// Volume slider
var setVolume = function(value){
        audio.volume = localStorage.volume = value;
        $('.volume .pace').css('width', value * 100 + '%');
        $('.volume .slider a').css('left', value * 100 + '%');
}

var volume = localStorage.volume || 0.5;
$('.volume .slider').slider({max: 1, min: 0, step: 0.01, value: volume, slide: function(event, ui)
        setVolume(ui.value);
        $(this).addClass('enable');
        $('.mute').removeClass('enable');
}, stop: function(){
        $(this).removeClass('enable');
}}).children('.pace').css('width', volume * 100 + '%');
```

Figure 3: **Login** - Part of the Code of list

But there are still many areas that need improvement. For example, it should use database to store and classify a large number of musics. And another database for users to save their username and password. so that each user can have own sub-database to store their personal information, all music or video they played and downloaded will be saved in the sub-database. The recommendation should change based on these information.