**SOLENT UNIVERSITY** **in Partnership with QA (QAHE)**  
School of Media Arts and Technology

BSc (Hons) Computing

Academic Year 2020-2021

University ID: 15078949

Daniel Grabinski

**Unit Title - Web Technologies**

**Assessment Title – Virtual Event**

Supporting Documentation - QHO431

**AE1 (100%)**

Unit Leader: Amjad Alam

Contents

[Table of Figures 2](#_Toc82781581)

[Four pages (Festival, FAQ, Contact Us, Fun) 3](#_Toc82781582)

[Responsiveness 3](#_Toc82781583)

[Functionality 7](#_Toc82781584)

[Fun section 9](#_Toc82781585)

[Database MySQL 11](#_Toc82781586)

[W3C Standards (online validator) 14](#_Toc82781587)

[Index.html 14](#_Toc82781588)

[Index.css 15](#_Toc82781589)

[FAQMusicFestival.html 15](#_Toc82781590)

[FAQMusicFestival.css 16](#_Toc82781591)

[MusicFestivalFestival.html 16](#_Toc82781592)

[Festival.css 16](#_Toc82781593)

[ContactUsMusicFestival.html 17](#_Toc82781594)

[ContactUsMusicFestival.css 17](#_Toc82781595)

[funzonemusicfestival.html 17](#_Toc82781596)

[funzonemusicfestival.css 18](#_Toc82781597)

[Compatibility with different browsers 18](#_Toc82781598)

[Internet Explorer: 18](#_Toc82781599)

[Google Chrome: 19](#_Toc82781600)

[Microsoft Edge: 19](#_Toc82781601)

# Table of Figures

[Figure 1 When the resolution is less than 700px 4](#_Toc82778778)

[Figure 2 Hoover effect 5](#_Toc82778779)

[Figure 3 Hoover for the menu 6](#_Toc82778780)

[Figure 4 All artists 6](#_Toc82778781)

[Figure 5 Artist for Main Stage 7](#_Toc82778782)

[Figure 6 Database and table contacts 9](#_Toc82778783)

[Figure 7 Rows in the table contacts 9](#_Toc82778784)

[Figure 8 The form in the database, after submitting 10](#_Toc82778785)

[Figure 9 Database error message 11](#_Toc82778786)

# Four pages (Festival, FAQ, Contact Us, Fun)

The web page consists of the main page called index.html and CSS file index.css. The project has four subpages. The webpage is accessible, thanks to alternative text on images. The website conforms to W3C Standards. The first called MusicFestivalFestival.html includes information about the music festival. On the Festival page, after choosing buttons with stages Main, Club or Disco, It will display pictures of artists who play only on that stage.

The event is online and consists of three stages and channels. Additionally, at the end of the page, I added details of all artists occurring at the Festival, using JavaScript. All HTML and CSS files have been tested through the W3school validator in line with standards. All pictures and graphics I downloaded from pexels.com and pngitem.com. I used different types of fonts on my webpage that I took from Google Fonts.

The second subpage is used for FAQ (faqmusicfestival.html and faqmusicfestival.css) on which I display the most common questions about the online festival. The section with questions I created with JavaScript.

The third subpage is called Contact Us (contactusmusicfestival.html and contactusmusicfestival.css), on which I made the form. The users can ask organizers questions and contact them. After submitting the form, the details from that form are stored in the database.

The fourth subpage is called Fun, (funzonemusicfestival.html and funzonemusicfestival.css). On this page, I create a game in which the user has to find two pictures of the same item.

# Responsiveness

The webpage is responsive for devices with smaller resolutions. For that, I used media queries, and I set the maximum resolution for 700px. By the use of display block, the arrangement of different parts of the website changes to block (one under the other) when the resolution is below 700px.In HTML file I used: <meta name="viewport" content="with=device-width, initial-scale=1.0"> , which gives informations for the browser how to control scale and size of the webpage. The part with=device-width is in charge of adjusting the resolution of the web page according to the device display size. The initial-scale=1.0 part is responsible for

approximation and adjusting of the page content to the device.

While resolution changes, the menu changes as well, and it appears on the side of the page. I used the function showMenu() and hideMenu(), which hides the menu when is needed. The icons I used from the page fotawesome.com and used the source (<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">) in the head section. To hides the icons for the full resolutions I used nav .fa{ display: none;}. above media query in CSS file.

Graphical user interface, website

Description automatically generated Graphical user interface, website

Description automatically generated

Figure 1 When the resolution is less than 700px

Text

Description automatically generatedGraphical user interface

Description automatically generated

# Functionality

On the main page, by the use of CSS (background-image:linear-gradient(rgba(4,9,30,0.7),rgba(4,9,30,0.7)),url(../images/festivalbanner.jpg); , I used a picture as a background. For the picture, I used position: relative to be under the section for the text. For the text (text-box) I used position: absolute. Under the text, I used a button which after hovering the cursor, changes the color. I used in CSS file hero-btn: hover{}.

Graphical user interface, website

Description automatically generated

Graphical user interface, website

Description automatically generated

Figure 2 Hoover effect

After clicking on the button "VisitUsToKnowMore" we are being transferred to the subpage Festival. In the navigation section, after hoovering on one of the names of the pages, we can notice a line. For this effect, I used CSS in function .nav-links ul li a{}.

nav-links ul li a{

color: #fff;

text-decoration: none;

font-size:14px;

font-family: 'Monoton', cursive;

}

.nav-links ul li::after{

content: ' ';

width: 0%;

height: 2px;

background: #dbc78f;

display: block;

margin: auto;

transition: 0.5s;

}

.nav-links ul li:hover::after{

width:100%;

}

Because of the width:0%, the line is not appearing. After using hoover, the line will appear only when the user moves the cursor on one of the menu names.

Graphical user interface, text

Description automatically generated

Figure 3 Hoover for the menu

Because of the width:0%, the line is not appearing. After using hoover, the line will appear only when the user moves the cursor on one of the menu names.

Similar style method I used on the other pages. On the subpage Festival, under the banner, they are three pictures. When We hoover on any of them, we can notice a backlight. Additionally, after clicking on the stage name, we are being moved to the description of the stage. Under the picture, I added #Mainstage, and in the description section of this stage, id=mainstage. That makes the automatic movement to the description.

On the subpage Festival, I added a section that displays 20 artists performing on three stages. I used data-target to designate three groups (all artists, main stage, club stage and disco stage). I used data-id to adjust the stage for each artist. After clicking on Main, it displays the only artist for stage Main. Similar to other stages.

A picture containing text, screenshot, different, colorful

Description automatically generated

Figure 4 All artists

A collage of people playing instruments

Description automatically generated with medium confidence

Figure 5 Artist for Main Stage

# Fun section

In subpage Fun, I created a game that checks if the two cards are the same (memory game). I started by creating a list with 12 cards in HTML. I used pictures from pexels.com. I needed 6 pictures of instruments and one front picture that covers cards in case of error.

<div class="card" data-framework="Guitar">

      <img class="front-face" src="../images/guitar.jpg" alt="Guitar" />

      <img class="back-face" src="../images/front.jpg" alt="Front"  /></div>

    <div class="card" data-framework="Guitar">

      <img class="front-face" src="../images/guitar.jpg" alt="Guitar" />

      <img class="back-face" src="../images/front.jpg" alt="Front" /></div>

To make sure jvascript code works on created list, I used const cards=document.querySelectorAll('.card), mean for each <div clss="card"...> it will work JS. For each card, after click on card program execute flipCard. In CSS I added .cardflip { transform: rotateY(180deg);} and added other parameters and hided front-face.

backface-visibility: hidden;

 .front-face,

.back-face {

  width: 100%;

  height: 100%;

  padding: 10px;

  position: absolute;

  border-radius: 6px;

  background: #030224;

**backface-visibility: hidden;**

**After that I execute first clink and second:**

if (!hasFlippedCard) {

    hasFlippedCard = true;

    firstCard = this;

    return;

  }

  secondCard = this;

  checkForMatch();

}

**If the cards are the same:**

function checkForMatch() {

  let isMatch = firstCard.dataset.framework === secondCard.dataset.framework;

  isMatch ? disableCards() : unflipCards();

}

**If there is no match, the cards are flipped back, I set the time for 1500 by using function setTimeout()**

setTimeout(() => {

    firstCard.classList.remove('flip');

    secondCard.classList.remove('flip');

    resetBoard();

  }, 1500);

}

**Every time page is refreshed, the cards are in different places.**

  cards.forEach(card => {

    let randomPos = Math.floor(Math.random() \* 12);

    card.style.order = randomPos;

  });

# Database MySQL

Graphical user interface, text, application, Word, email

Description automatically generated

Figure 6 Database and table contacts

On the subpage Contact US, I placed the form. To the form, I connected the file dbconnect.php which makes that data is store in the database. In http://localhost/phpmyadmin I created a new project which I called contact (http://localhost/phpmyadmin/db\_structure.php?server=1&db=contact )

After that, I created a table of contacts, consists of six columns. The first column (id) is a Primary Key. The next three (type Varchar) can include only 50 characters. The email address 20 characters, and the message is less than 400 characters.

Graphical user interface, application

Description automatically generated

Figure 7 Rows in the table contacts

To send the form to the database, I used <input> in the HTML file and I added name-id, the same label as the one in the table, and I added type. In my form it was a text for label email : <input type="text" class="form-control" id="email" name="email"/> and then in a file dbconnect.php. I defined all variables in the super global variable POST.

$firstname = $\_POST['firstname'];

$lastname = $\_POST['lastname'];

$city = $\_POST['city'];

$email = $\_POST['email'];

$Message = $\_POST['Message'];

Afterwards, I executed connection $conn = new mysqli('localhost','root','','contact');. If any error occurs, display an error "Connection Failed". If the form was filled in correctly, On the screen should appear the message "Thank You for contacting us!". I tested the form, and below we can see the form details stored in the database.

Graphical user interface, text, application, email

Description automatically generated

Figure 8 The form in the database, after submitting

In the subpage FAQ, I used javascript to automate the section with questions. When we click on the "+" icon, the answer for the question will roll down. I used a loop in javascript (i = 0; i < faq.length; i++) to check if the question includes the answer, roll down the answer (addEventListener). The answer is displayed by the use of var body=addEventListener, which means next FAQ-body appearing after FAQ-question.

One of the biggest issues that I faced during writing a code for databases was an error while sending the form to the database. The error was caused by the wrong table name, where I missed a letter.

$stmt = $conn->prepare("insert into **contac**(firstname, lastname, city, email, Message) values(?, ?, ?, ?,?)");

It should be **contancts**

Graphical user interface, text, application, email

Description automatically generated

Figure 9 Database error message

# W3C Standards (online validator)

## Index.html

Graphical user interface, text, application, email

Description automatically generated

## Index.css

Graphical user interface, text, application, email

Description automatically generated

## FAQMusicFestival.html

Graphical user interface, text, application, email

Description automatically generated

## FAQMusicFestival.css

Graphical user interface, text, application, email

Description automatically generated

## MusicFestivalFestival.html

Graphical user interface, text, application, email

Description automatically generated

## Festival.css

Graphical user interface, text, application, email

Description automatically generated

## ContactUsMusicFestival.html

Graphical user interface, text, application, email

Description automatically generated

## ContactUsMusicFestival.css

Graphical user interface, text, application, email

Description automatically generated

## funzonemusicfestival.html

Graphical user interface, text, application, email

Description automatically generated

## funzonemusicfestival.css

Graphical user interface, text, application, email

Description automatically generated

# Compatibility with different browsers

## Internet Explorer:

Graphical user interface, website

Description automatically generated

## Google Chrome:

Graphical user interface, website

Description automatically generated

## 

## Microsoft Edge:

Graphical user interface, website

Description automatically generated