**Hong Kong Diploma of Secondary Education Examination 2017**

**Information and Communication Technology**

**School-based Assignment**

**Elective C**

**Hong Kong Taoist Association Tang Hin Memorial Secondary School**

**Name: Yeung Sum Ho**

**Class: 6D**

**Class no: 9**

**Project title: Class website**

**Website:**

Table of Content

1. Background and objective
   1. Background
   2. Objective
2. Design (CSS , main page ,page layout ,banner, button ,program design)
   1. Page layout

2.1.1Navigation bar~

2.1.2 Main-page layout~

2.2 Graphic design

2.2.1 Logo~

2.3 Program design

2.3.1 login and logout function

2.3.1.1 introduction~

2.3.1.2 layout(html)~

2.3.1.3 cookies~

2.3.1.4 database~

2.3.2 Gallery

2.3.2.1 introduction~

2.3.2.2layout(html)~

2.3.3 up-coming events

2.3.3.1 introduction~

2.3.3.2 layout(html)~

2.3.3.3database~

2.3.4 student profile

2.3.4.1 introduction~

2.3.4.2 layout(html)~

1. Implementation

3.1Login and logout

3.1.1Login~

3.1.2Logout~

3.2Gallery

3.2.1upload~

3.2.2filtering~

3.2.3viewing~

3.3Up-coming event

3.3.1upload~

3.3.2 delete & edit~

3.4student profile

3.4.1viewing~

3.4.2edit~

1. Testing and Evaluation~
2. Discussion and Conclusion
3. References
4. **Design** (CSS , main page ,page layout ,banner, button ,program design)
   1. **Page layout**
      1. **Navigation bar**

The navigation system allows user access different pages of the website easily. The navigation bar is designed horizontal and locate on the top of the page because the top sections of pages are more successful in drawing the eyes than left sidebars. Also, it can save more content area. The figure shows the design of the navigation bar.



The main theme of the navigation bar is orange, white and black so that the webpage would be more eye-catching and it also fit the main theme of the site. The logo in the navigation bar links to the homepage of the site.

The page title of page which users are accessing will become black in background color so that users would know which page they are in. When the cursor points to different titles on the navigation bar, the pointed title will become black in background color too. The figure below shows the effect when user is in ‘HOME’ and his cursor points to ‘GALLERY’.



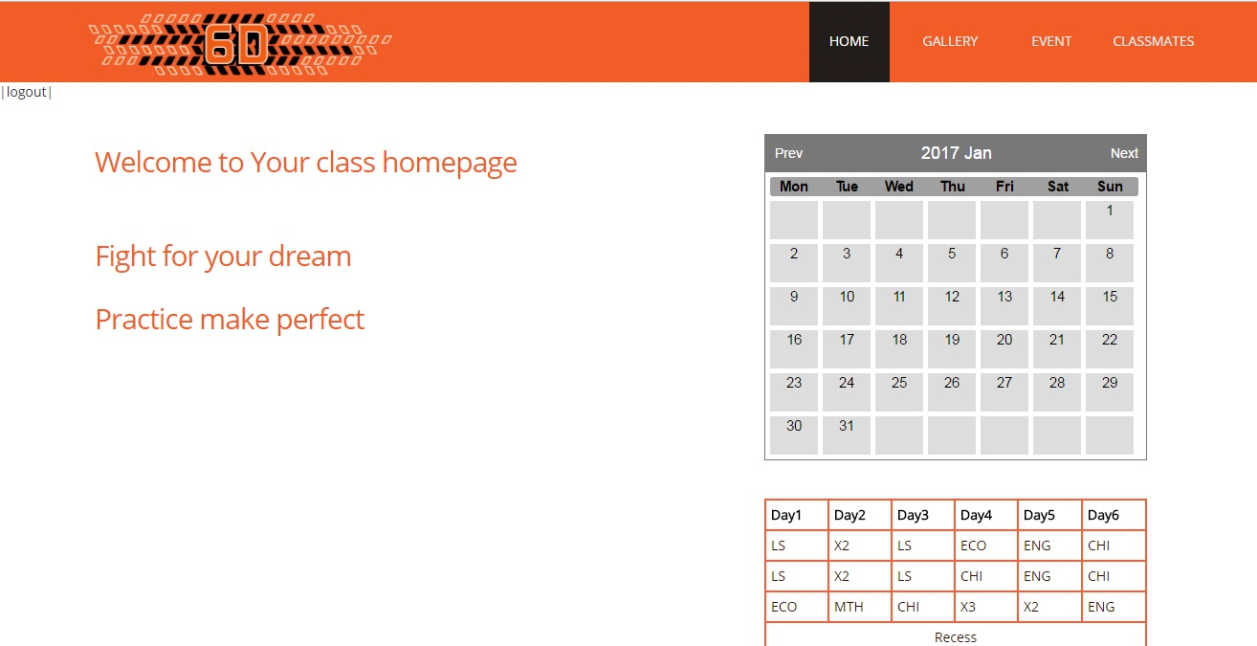
Here is the CSS

|  |
| --- |
| }  .nav-collapse\_ .nav > li.sfHover > a,  .nav-collapse\_ .nav > li.sfHover > a:hover,  .nav-collapse\_ .nav > li > a:hover,  .nav-collapse\_ .nav > li.active > a,  .nav-collapse\_ .nav > li.active > a:hover {  background:url("../img/bg-content.jpg") repeat scroll 50% 50% transparent;  color: #ffffff;  -webkit-box-shadow: none;  -moz-box-shadow: none;  box-shadow: none;  } |

Here is the html code

|  |
| --- |
| <div class="navbar navbar\_">  <div class="container">  <h1 class="brand brand\_">  <div class="nav-collapse nav-collapse\_ collapse">  <li class="active"><a href="homepage.php">Home</a></li>  <li><a href="photoalbum.php">Gallery</a></li>  <li><a href="event.php">Event</a></li>  <li><a href="classmates.php">classmates</a> </div>  </div>  </div> |

* + 1. Main-page layout



For the main page, there is a navigation bar with website logo on the header of the page. Logout button locates left hand side below the navigation bar. The main content is divided into two parts. The text content is on the left and some static functions are on the right. It is divided by the Bootstrap grid system.

The default Bootstrap grid system utilizes  **12 columns**, making for a 940px wide container without  [responsive features](http://getbootstrap.com/2.3.2/scaffolding.html#responsive)  enabled. The ‘.offset’class is used to move columns to the right. Each class increases the left margin of a column by a whole column. It can create the margin for the main content and separate two different contents side by side in certain distance.

Here is the Bootstrap code

|  |
| --- |
| <div class="row">  <div class="row">  <div class="span6 offset1">  <h3> Welcome to Your class homepage <h3>  <div class="container">  <h3>Fight for your dream</h3>  <h3>Practice make perfect</h3>  </div>  </div>  <div class="span4 offset1">  <br />  <?php  include 'calendar.php';  $calendar = new Calendar();  echo $calendar->show();  ?>  </br>  </div>  </div> |

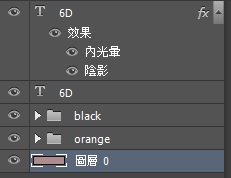
There are total 6 columns for the content on the left and 1 column of left margin. There are 4 columns for the content on the right and 1 column of its left margin.

**2.2 Graphic design**

**2.2.1Logo**



I designed the logo of my class website myself. I created it by using the graphic editing software ‘Adobe Photoshop CS6’.The resolution of the logo is 322pixels x 65pixels which is fit for the website. I used the function ‘stamp tool’, ‘layer mask’ to create the logo. By modifying the layer styles, I have added same effects which are ‘shadow’ and ‘inner grow’ to the text ‘6D’ so that it would be more eye catching.



The image are distributed to different layers , so I can apply different effects to the images in different layers. The logo was stored into PNG file format which is lossless compression file format with high image quality.

2.3 **Program Design**

**2.3.1 login and logout function**

**2.3.1.1 Introduction**

The login and logout functions are used to ensure that the users who access the website are 6D classmates of the school. Users can only access the page content and use the functions of the website when they login. The function of account register is not provided because the accounts are assigned by web designer.

**2.3.1.2 function layout**

**Login**

|  |
| --- |
|  |

There are two textbox for inputting the username and password .The reset button is for resetting the two textbox into blank .Clicking the submit button to login.

The HTML code

|  |
| --- |
| <article class="span12">  <h4 align = "center">Please Login :0)</h4>  </article>  <form align="center" name="myForm" action = "login.php" onsubmit="return validateForm()" method="get">  <h3> Username : </h3>  <input type = "text" name = "un"><br>  <h3> Password : </h3>  <input type = "password" name = "pw"><br>  <input type = "submit" class="btn btn-1" value = "Submit" >  <input type = "reset" class="btn btn-1" value = "reset">  </form> |

The inputted data will be collected when the button is clicked. The data will be passed to the server.

CSS setting

|  |
| --- |
| h3 {  font-size: 30px;  line-height: 34px;  margin: 32px 0 15px;  font-weight: normal;  }  h4 {  font-size: 48px;  line-height: 50px;  margin: 21px 0 20px;  }  .btn-1 {  color: #fff;  background: #F25C27;  padding: 5px 14px;  -webkit-border-radius: 0;  -moz-border-radius: 0;  border-radius: 0;  text-shadow: none;  border: none;  -webkit-box-shadow: none;  -moz-box-shadow: none;  box-shadow: none;  -webkit-transition: all 0.3s ease;  -moz-transition: all 0.3s ease;  -o-transition: all 0.3s ease;  transition: all 0.3s ease;  }  .btn-1:hover {  color: #fff;  background:#DB420E;  -webkit-transition: all 0.3s ease;  -moz-transition: all 0.3s ease;  -o-transition: all 0.3s ease;  transition: all 0.3s ease;  text-decoration: none; |

For the submit and reset button , CSS ’ class="btn btn-1"’is used. The button will be like that,.There will be transition effect when the cursor point to the button and it changes the background color from #F25C27 to #DB420E .Here is the effect when cursor is point to submit button.

**Logout**

It allows users to logout their account when finish using the website. The logout button locates left hand corner below the page logo. It exists on the same location every page except login page (index.html).

The HTML Code

|  |
| --- |
| <a href="logout.php">|logout|</a> |

The” logout.php” will be called .It deletes the cookies for logging out.

CSS code

|  |
| --- |
| a {  color: #F25C27;  text-decoration: none;  outline:none;  }  a:hover {  color:#ffffff;  text-decoration: none;  } |

The logout text button is in color #F25C27 and it will becomes white in color #ffffff when mouse is point at it.

**2.3.1.3 Cookies**

Cookies is set for specify the users of the website when users log in. Users can access the content and use the function of the site when the cookies is set. Here is the code of setting cookies:

|  |
| --- |
| {  setcookie("user", $un, time()+3600);  header("Location: homepage.php");  } |

It sets the cookies that last for 3600 seconds. After cookies is set “homepage .php”will be called and users can access the pages.

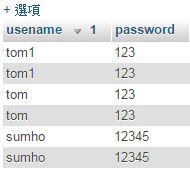
After 3600 seconds , the cookies will be deleted automatically. The login page“index.html”will be called and allow users login again. It can reduce the risk that others access the pages through users account if users forget logging out. Here is the code:

|  |
| --- |
| <?php  if (!isset($\_COOKIE["user"]))  {  header("Location: index.html");  }  ?> |

Also, the cookies will be deleted when users click the logout button. Here is the PHP code

|  |
| --- |
| <?php  setcookie("user", $un, time()-36000);  header("Location: index.html");  exit;  ?> |

**2.3.1.4database design**

I used free SQL database db4free.nt to create my own database. For the login and logout function, I create a table called “account”. It contains 2 fields that they are “username”and “password”. Here is the table “account”of the database.

The table contains several set of username and password. 1 username and 1 password form a record. It is used to check whether the inputted values in the textbox of the login page match with the data in the database. It can ensure that the users are my classmates.



Both username and password are in text data type

**2.3.2 Gallery**

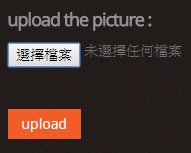
**2.3.2.1 introduction**

The photo album is designed for classmates to upload and share the photos. Users can upload and view the photo there. Students can view specific kind of photo on the photo album by using the filter.

**2.3.2.2 layout(html)**

The background color is set into black in color.

**On the upper part of the Gallery page**, there is an area for uploading photo. It allows users to upload the photo they want.



There would be default text “未選擇任何檔案”before users select the file which is ready file uploading. After selecting the file ,the text will change into the name of the selected file and it looks like  .Users can check whether it is the photo that they want to upload.

The HTML code

|  |
| --- |
| <form method="post" enctype ="multipart/form-data">  <br/>  <h5> upload the picture : </h5>  <input type = "file" class="custom-file-input" name = "image"><br>  <br/><br/>  <button type = "submit" class="btn btn-1" name="submit" value = "Upload" >upload </button>  </form> |

CSS

|  |
| --- |
| h5 {  font-size: 18px;  color: #939393;  font-weight: 600;  letter-spacing: -1px;  }  .btn-1 {  color: #fff;  background: #F25C27;  padding: 5px 14px;  -webkit-border-radius: 0;  -moz-border-radius: 0;  border-radius: 0;  text-shadow: none;  border: none;  -webkit-box-shadow: none;  -moz-box-shadow: none;  box-shadow: none;  -webkit-transition: all 0.3s ease;  -moz-transition: all 0.3s ease;  -o-transition: all 0.3s ease;  transition: all 0.3s ease;  }  .btn-1:hover {  color: #fff;  background:#DB420E;  -webkit-transition: all 0.3s ease;  -moz-transition: all 0.3s ease;  -o-transition: all 0.3s ease;  transition: all 0.3s ease;  text-decoration: none;  } |

The effect of the upload button is the same as login and reset button. The color will change with cursor points to it.<h5> is applied to the text that the content become grey in color and font size is 18px

**On the lower part of the Gallery page, users can view the uploaded photo.**



There is a 4x2 gallery and it can display 8 photos which are 270px \* 192px in each Gallery page. There are two Gallery pages. There is a pages system  at the bottom the Gallery pages. There will be underline effect on the page number of the current page so that users can know which page they are accessing.

The HTML code

|  |
| --- |
| <div align="center" class="navigationBar">  <u><a href="./photoalbum.php">1</a></u>  | <a href="./photoalbum1.php">2</a>  | <a href="./photoalbum1.php">Next</a>  | <a href="./photoalbum1.php">Last</a>  </div>  -------------------------------------------------------------  <div align="center" class="navigationBar">  <a href="./photoalbum.php"rel="prev">Previous</a>  | <a href="./photoalbum.php">1</a>  | <u><a href="./photoalbum1.php">2</a></u>  | <a href="./photoalbum1.php">Last</a>  </div> |

Relative links are used because those pages are located on the current website. However, I cannot use it if I link to a location on another website.<u>tag is used to underline the page number.

When the cursor points to the small photos of the Gallery pages, the photos will become transparent and there will be an image of magnifier on the middle of the photos. This effect tells users that they can magnify the photos.



When users click on the photos, the photos will be magnified and displayed on the middle of the screen. There are two arrows on two sides of the page foe shifting the photos one by one until the last photo of the current page. Users can also view previous photo by using this function

The HTML code

|  |
| --- |
| <div class="clear"></div>  <ul class="portfolio clearfix">  <li class="box"><a href="img/1/classphoto1.jpg" class="magnifier" ><img alt="" src="img/1/classphoto1.jpg" width="270px" height="192px" ></a></li>  <li class="box"><a href="img/1/classphoto2.jpg" class="magnifier" ><img alt="" src="img/1/classphoto2.jpg" width="270px" height="192px" ></a></li>  <li class="box"><a href="img/1/classphoto3.jpg" class="magnifier" ><img alt="" src="img/1/classphoto3.jpg" width="270px" height="192px" ></a></li>  <li class="box"><a href="img/2/event1.jpg" class="magnifier" ><img alt="" src="img/2/event1.jpg" width="270px" height="192px" ></a></li>  <div class="clear"></div>  <li class="box"><a href="img/3/butterfly.png" class="magnifier" ><img alt="" src="img/3/butterfly.png" width="270px" height="192px" ></a></li>  <li class="box"><a href="img/3/cat\_like\_dragon.png" class="magnifier" ><img alt="" src="img/3/cat\_like\_dragon.png" width="270px" height="192px" ></a></li>  <li class="box"><a href="img/3/four\_colors.png" class="magnifier" ><img alt="" src="img/3/four\_colors.png" width="270px" height="192px" ></a></li>  <li class="box"><a href="img/3/lightning\_bolt.png" class="magnifier" ><img alt="" src="img/3/lightning\_bolt.png" width="270px" height="192px" ></a></li>      </div> |

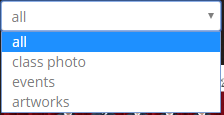
JPEG file format is used because it supports lossy compression with high compression ratio and most photos are in high resolution. File size of the photos will be small. It saves the resources of the server.

CSS Script

|  |
| --- |
| .magnifier {  opacity: 1;  position: relative;  height: auto;  display: block;  -webkit-transition: all 0.3s ease;  -moz-transition: all 0.3s ease;  -o-transition: all 0.3s ease;  transition: all 0.3s ease;  }  .magnifier:hover {  opacity: 0.7;  }  .magnifier:after {  content: '';  display: block;  width: 100%;  height: 100%;  position: absolute;  opacity: 0;  left: 0;  top: 0;  background: url('../img/magnifier.png') center center no-repeat #000000;  -webkit-transition: all 0.3s ease;  -moz-transition: all 0.3s ease;  -o-transition: all 0.3s ease;  transition: all 0.3s ease;  }  .magnifier:hover:after {  opacity: 0.6;  } |

Different CSS properties are applied. The opacity is 1 by default that the images are absolute solid without transparency. When the cursor points to the images, opacity will become 0.7 transparent with 0.7 degree solid. Then, the image “magnifier.png”will exist on the middle of the image if the cursor stays there. Those transition effects will operate and complete in 0.3 second. The kwicks-slider is activated when users click on the photos .

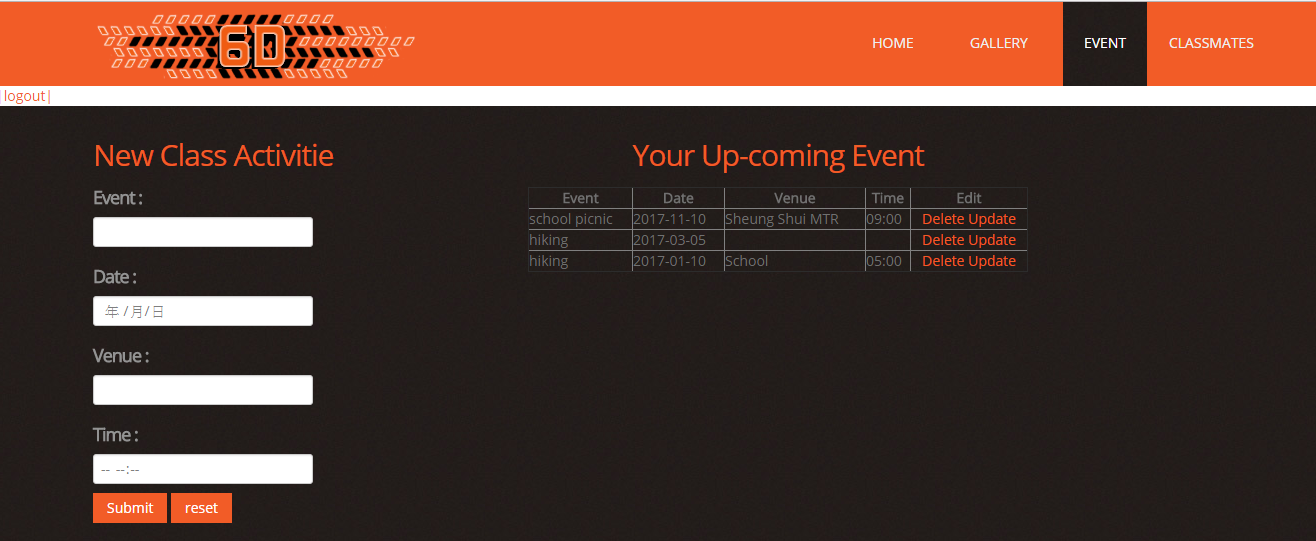
**There is a filtering function by selecting specific kind of photos.**

The dropdown list is used for filtering. When Users click on the list, the list will expand and several choices will be shown: all, class photo, events, artworks. Users can choose the type of photos that they want to view. After they pick the choice, only the chosen kind of photos will be displayed. It is convenient for users to find and view photos if there are large amount of photos in one Gallery pages.

**2.3.3 up-coming events**

**2.3.3.1 introduction**

Up-coming events is a function that allow classmates to update the information of coming class activities. Users can upload, update, delete the information the events. Students can know about the coming events when they access this page. Users can delete or edit the record if there is any adjustment.

**2.3.3.2** **layout(html)**

The up-coming events page is divided into two parts side by side. The left part is the form of inputting the information of the class activities. The right part is the uploaded information of the events. Here is the HTML code:

|  |
| --- |
| <article class="span4">  ~~~~~~~~~~~~~<content>~~~~~~~~~~~~  <article class="span6">  ~~~~~~~~~~~~~<content>~~~~~~~~~~~~ |

CSS

|  |
| --- |
| .span6 {  width: 460px;}  .span4 {  width: 300px;  } |

# Bootstrap Grid System is used for divided the main content of the up-coming events page into the two part. The left column is 300px width .The right column is 460px width.

# The fill-in form

# There are 4 input box to collect the information of the coming

class activities. They are Event, Date, Venue, Time. For Event and Venue, textbox is used for collecting those information. For Date and Time, the data types, date and time are used representatively. It can standardize the format of date to (year/month/day) . It can standardize the format of time too.

Here is the HTML code of fill-in form

|  |
| --- |
| <h3> New Class Activitie </h3>  <form action = "uploadevent.php" method="post">  <h5> Event : </h5>  <input type = "text" name = "event"><br>  <h5> Date : </h5>  <input type = "date" name = "date"><br>  <h5> Venue : </h5>  <input type = "text" name = "venue"><br>  <h5> Time : </h5>  <input type = "time" name = "time"><br>  <input type = "submit" class="btn btn-1" value = "Submit">  <input type = "reset" class="btn btn-1" value = "reset">  </form> |

CSS

|  |
| --- |
| h1,  h2,  h3,  h4,  h5,  h6 {  text-transform: none;  color: #F25C27;  font-family: 'Open Sans', sans-serif;  font-weight: 300;  letter-spacing: -1px;  h3 {  font-size: 30px;  line-height: 34px;  margin: 32px 0 15px;  font-weight: normal;  }  h5 {  font-size: 18px;  color: #939393;  font-weight: 600;  letter-spacing: -1px; |

With tag</h3>,the text will be orange with 30px in font size and typeface Open Sans. With tag</h5>,the text will be light grey with 18px in font size and typeface open Sans.

The table with event’s information



The table contains 5 columns .There are 4 data field and 1 function field. The imputed data will be shown on the table after the data store into database. The latest information is on the bottom of the table.

The HTML code

|  |
| --- |
| <table align = "center" border="1" cellspacing = "0" cellpadding="0" width= "500">  <caption><h3> Your Up-coming Event </h3></caption>  <tr>  <th>Event</th>  <th>Date</th>  <th>Venue</th>  <th>Time</th>  <th>Edit</th>  </tr>  <?php  $i=1;  while($row = mysqli\_fetch\_array($result))  {  echo"<tr>  <td> ".$row['event']."</td>  <td> ".$row['date']."</td>  <td> ".$row['venue']."</td>  <td> ".$row['time']."</td>  <td align = 'center'>  <a href='#'>Delete</a>  <a href='#'>Update</a>  </td>  </tr>";  $i++;  }  ?>  </table> |

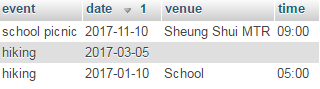
It will output the record after the server side program is excuted.

CSS

|  |
| --- |
| .table {  width: 100%;  margin-bottom: 20px;  }  .table th,  .table td {  padding: 8px;  line-height: 20px;  text-align: left;  vertical-align: top;  border-top: 1px solid #dddddd;  }  .table th {  font-weight: bold; |

Tag<th> is applied to field names and tag<td> is applied to field data. The styles are similar but the texts with tag<th> are bold. It can make the field name sharper.

2.3.3.3 database



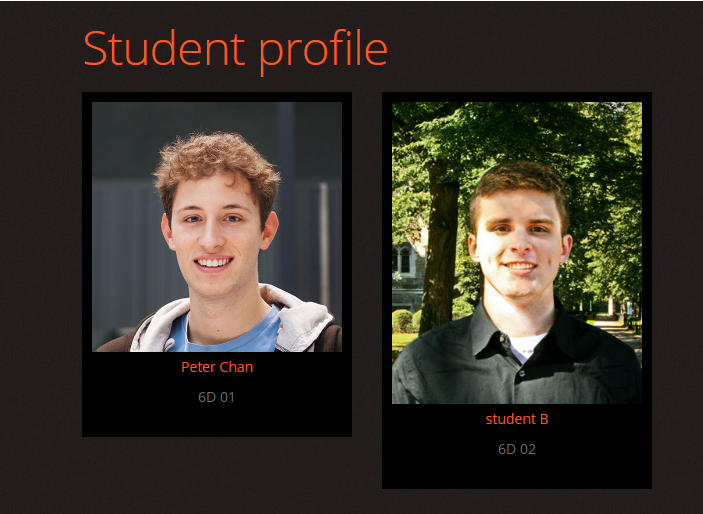


4 fields are created for 4 inputted data of the fill-in table. The data types of all fields in database are set as text. The data formats of date and time have been standardized during the input section.

**2.3.4 student profile**

**2.3.4.1introduction**

Student profile page contains personal information of my classmates, such as profile picture, mobile phone number, e-mail address. It helps classmates contact with each other. Users can edit their own profile information but they cannot modify others’profiles.



Name, class, class number are shown in the student profile page .Those elements store into container They list side by side and row by row. Here is the HTML code.

|  |
| --- |
| <ul class="thumbnails thumbnails-1 list-services">  <li class="span3">  <div class="thumbnail thumbnail-1"> <img src="img/4/stud1.jpg" width="270px" height ="192px" alt="">  <section> <a href="./profile.html" class="link-1">Peter Chan </a>  <p>6D 01</p>  </section>  </div>  </li>  <li class="span3">  <div class="thumbnail thumbnail-1"> <img src="img/4/stud2.jpg" width="270px" height ="192px" alt="">  <section> <a href="#" class="link-1">student B </a>  <p>6D 02</p>  </section>  </div>  </li>  </ul> |

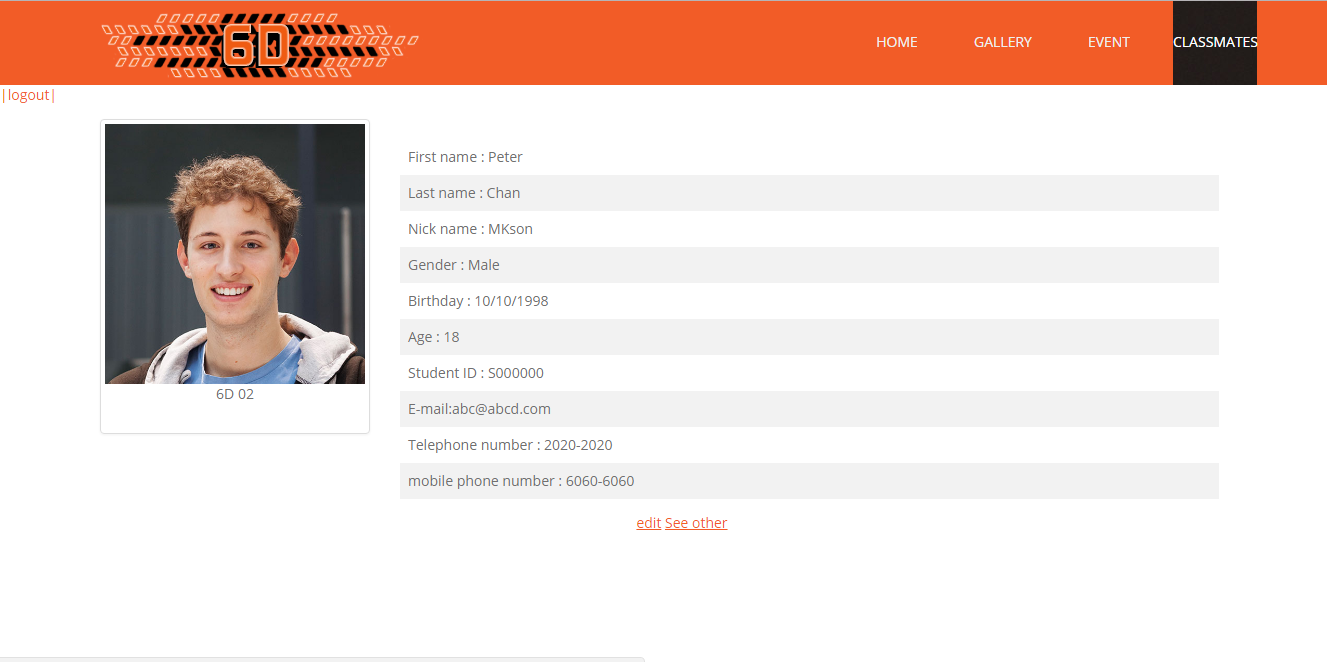
CSS

|  |
| --- |
| .list-services .thumbnail-1 {  background: #000000;  padding: 10px;  }  .list-services a {  margin-bottom: 10px;  display: inline-block;  }  .list-services p {  padding-bottom: 20px;  }  .thumbnail {  display: block;  padding: 4px;  line-height: 20px;  border: 1px solid #ddd;  -webkit-border-radius: 4px;  -moz-border-radius: 4px;  border-radius: 4px;  -webkit-box-shadow: 0 1px 3px rgba(0, 0, 0, 0.055);  -moz-box-shadow: 0 1px 3px rgba(0, 0, 0, 0.055);  box-shadow: 0 1px 3px rgba(0, 0, 0, 0.055);  -webkit-transition: all 0.2s ease-in-out;  -moz-transition: all 0.2s ease-in-out;  -o-transition: all 0.2s ease-in-out;  transition: all 0.2s ease-in-out;  }  }  .thumbnail > img {  display: block;  max-width: 100%;  margin-left: auto;  margin-right: auto; |

The first three setting is for the black container. ‘.thumbnail’set the area of the profile picture. ‘.thumbnail > img’set the images to fully fill up the provided area.



Users can click the name to access the student’s profile because it is a text link. It links to "./profile.html"



The HTML code

|  |
| --- |
| <div class="row-1">  <div class="row">  <div class="span3 offset1">  <div class="thumbnail thumbnail-1"> <img src="img/4/stud1.jpg" alt="">  <section>  <p align = "center">6D 02</p>  </section>  </div>  </div>  <div class="span9 ">  <br />  <div class="container">  <table>  <tr>  <td>First name : Peter</td>  </tr>  <tr>  <td>Last name : Chan</td>  </tr>  <tr>  <td>Nick name : MKson</td>  </tr>  <tr>  <td>Gender : Male</td>  </tr>  <tr>  <td>Birthday : 10/10/1998</td>  </tr>  <tr>  <td>Age : 18</td>  </tr>  <tr>  <td>Student ID : S000000</td>  </tr>  <tr>  <td>E-mail:abc@abcd.com </td>  </tr>  <tr>  <td>Telephone number : 2020-2020</td>  </tr>  <tr>  <td>mobile phone number : 6060-6060</td>  </tr>  </table>  </div>  </div>  </div>  </div>  </div>  <div align="center" class="navigationBar">  <u><a href="">edit</a></u>  <u><a href="./classmates.php">See other</a></u> |

Bootstrap Grid system is used again for dividing the contain into two part.



‘Edit’ button allow students edit their own profile. ‘See other’brings users back to the main profile page.

1. **Implementation**

**3.1Login and logout**

**3.1.1Login**

|  |
| --- |
|  |

When users access the website, they need to input the username and password. After

inputting the username and password, ‘login.php’ will be called

The html code which calls ‘login.php’

|  |
| --- |
| <form align="center" name="myForm" action = "login.php" onsubmit="return validateForm()" method="get"> |

Then, it will get the values of inputted data and compare them with the data at the database.

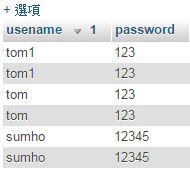
Here is the code of logging in the database

|  |
| --- |
| $servername = "85.10.205.173";  $dbusername = "sumhohoho04";  $dbpassword = "THMSSsumho04";  $dbname = "sumho2017";  $conn = mysqli\_connect($servername, $dbusername, $dbpassword,$dbname,3306);  if (mysqli\_connect\_errno()) {  echo "Failed to connect to MySQL: " . mysqli\_connect\_error();  } |

‘mysqli\_connect ‘ sets up connection to the database according to the server name, database name, username and password. $servername is the IP of the server. If there is a problem of connection, ‘ Failed to connect to MySQL: " . mysqli\_connect\_error()’ will be showed

The operation of comparing between data will start after connect to the database. It will compare with the data by following the SQL

|  |
| --- |
| $sql = "select \* from account where usename = '".$un."'and password='".$pw."' "; |



It checks the inputted value exist in the database on the left.

The code below will check whether the inputted data match with the username and password in the table “account” of the database ‘sumho2017’. If the inputted username and password can be found, the number of row will increase to 1 ; If the inputted username and password cannot be found, the number of row will be 0. It shows that ‘if ($result->num\_rows > 0)’carry out the identity verification.

|  |
| --- |
| $result = $conn->query($sql);  if ($result->num\_rows > 0) |

The cookie will be set if the inputted values are correct. The first attribute will specify the name of the cookie. Second attribute will specify the username as the content. Third one is the time that the cookie will last for. Users need to login again after 1 hours for the below setting.

|  |
| --- |
| setcookie("user", $un, time()+3600); |

‘Header’ will redirect to user the other location .User will be redirect to ’homepage.php’ if login is success. If the inputted data is invalid, it will go back to ‘index.html’ so that users could try again.

|  |
| --- |
| if ($result->num\_rows > 0)  {  setcookie("user", $un, time()+3600);  header("Location: homepage.php");  }  else  {  header("Location: index.html");  } |

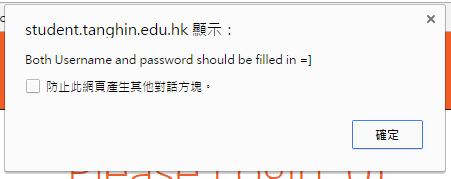
Field present check

Besides, if the login form is submitted with empty input box, an alert box will pop out and ask users to input the username and password. Here is the JavaScript and html code for doing validation.

The script will store the value of “un” and “pw” into variable x and y. ‘ if (x == "" || y =="") ‘ will check whether x or y is equal to ””(empty). “alert()” will call the alert box to draw the attention of users. If x or y is equal to nothing, an alert box will pop out.

|  |
| --- |
| <script>  function validateForm() {  var x = document.forms["myForm"]["un"].value;  var y = document.forms["myForm"]["pw"].value;  if (x == "" || y =="") {  alert("Both Username and password should be filled in =] ");  return false;  }  }  </script>  <form align="center" name="myForm" action = "login.php" onsubmit="return validateForm()" method="get">  <h3> Username : </h3>  <input type = "text" name = "un"><br>  <h3> Password : </h3>  <input type = "password" name = "pw"><br>  <input type = "submit" class="btn btn-1" value = "Submit" >  <input type = "reset" class="btn btn-1" value = "reset">  </form>validateForm()" method="get"> |

The alert box will pop out like this when users have not fill in the login form completely.



**3.1.2Logout**

If users want to log out, they need to click on the logout button on the upper left corner. The cookie which has been set will be deleted because the attribute ‘time()-36000’will set the time availability into 0.Then, ‘header’ will redirect users to login page ‘index.html’ Here is the code for logging out.

|  |
| --- |
| <?php  setcookie("user", $un, time()-36000);  header("Location: index.html");  exit;  ?> |

**3.2Gallery**

**3.2.1upload(prototype)**

For selecting the image file, users need to click on the button .Then, when users click on the button . After that it will connect to the database.

|  |
| --- |
| $servername = "85.10.205.173";  $dbusername = "sumhohoho04";  $dbpassword = "THMSSsumho04";  $dbname = "sumho2017";  $conn = mysqli\_connect($servername, $dbusername, $dbpassword,$dbname,3306); |

Here is the code for uploading the file.

|  |
| --- |
| <form method="post" enctype ="multipart/form-data">  <br/>  <h5> upload the picture : </h5>  <input type = "file" class="custom-file-input" name = "image">  <br>  <br/><br/>  <button type = "submit" class="btn btn-1" name="submit" value = "Upload" >upload </button>  </form> |

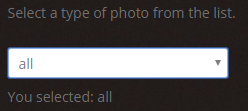
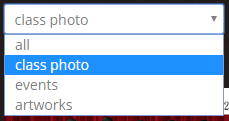
“$\_FILES”will get the information of the selected file. There are two attribute in

$\_FILES["image"]["name"].The first attribute is used to point to the selected file. The second attribute is the parameter which we need. Therefore, $imagename stores the name of the image.

'tmp\_name' stores the temporary location of the file. “file\_get\_contents()” read the contents of the image file into a string. The content will be stored into the variable $imagetmp. The value of the two variables will be transmitted to the database.

|  |
| --- |
| <?php  $imagename=$\_FILES["image"]["name"];  $imagetmp=addslashes (file\_get\_contents($\_FILES['image']['tmp\_name']));  $insert\_image= "INSERT INTO image\_table VALUES('$imagetmp','$imagename')";  mysql\_query($insert\_image);  ?> |

**3.2.2filtering(prototype)**

When users select the type of image, selected type of image will be shown on the photo album. For example, users choose type ”class photo”, only class photo will be show.

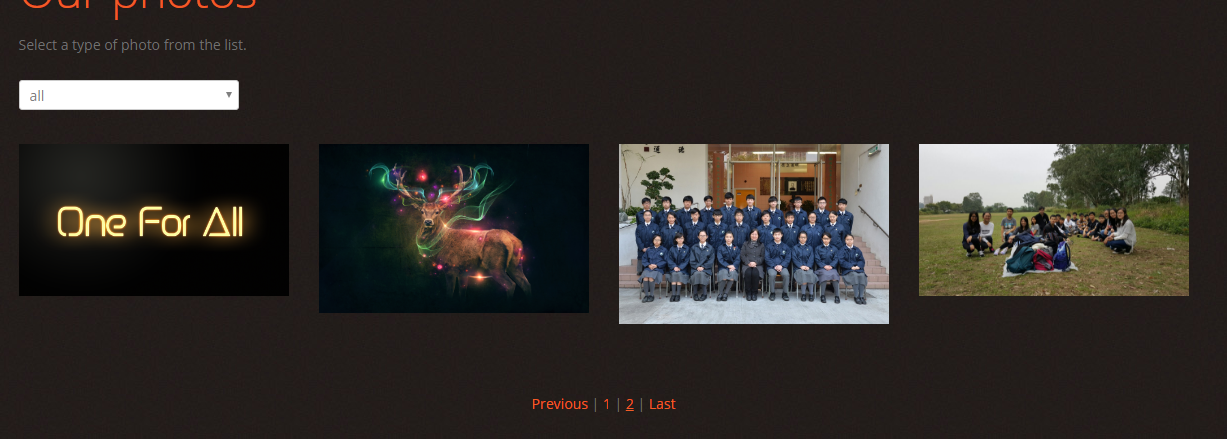
**3.2.3viewing**



Users can review the small photos when they come to Gallery page. Users can turn to the other page by clicking on the button on the footer of the page.



Users can click on “Next” button or “2” to access page 2 of Gallery. When the buttons are clicked, they will redirect users to “photoalbum1.php”. Page 2 is showed below:



The button will become , users can click “Previous” button or “1”

To go back to page 1 of Gallery page (photoalbum.php ) . Beside, users can shift to the last page of the Gallery by using “Last” button.

Here is the code of pages function in “photoalbum.php”

|  |
| --- |
| <div align="center" class="navigationBar">  <u><a href="./photoalbum.php">1</a></u>  | <a href="./photoalbum1.php">2</a>  | <a href="./photoalbum1.php">Next</a>  | <a href="./photoalbum1.php">Last</a>  </div> |

Here is the code of pages function in “photoalbum1.php”

|  |
| --- |
| <div align="center" class="navigationBar">  <a href="./photoalbum.php"rel="prev">Previous</a>  | <a href="./photoalbum.php">1</a>  | <u><a href="./photoalbum1.php">2</a></u>  | <a href="./photoalbum1.php">Last</a>  </div> |

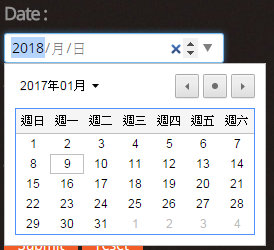
The attribute “href” specifies the link’s destination. The code above shows that different buttons link to different pages.

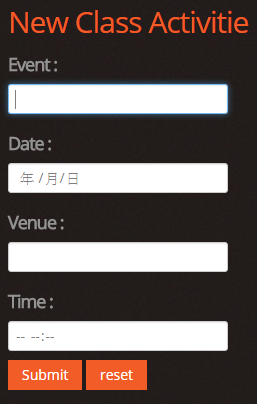
For viewing the large photos, users have to click on the photos to enlarge the photos. Users can click the black margin to quit the image slider.



|  |
| --- |
| <div class="clear"></div>  <ul class="portfolio clearfix">  <li class="box"><a href="img/1/classphoto1.jpg" class="magnifier" ><img alt="" src="img/1/classphoto1.jpg" width="270px" height="192px" ></a></li>  <li class="box"><a href="img/1/classphoto2.jpg" class="magnifier" ><img alt="" src="img/1/classphoto2.jpg" width="270px" height="192px" ></a></li>  <li class="box"><a href="img/1/classphoto3.jpg" class="magnifier" ><img alt="" src="img/1/classphoto3.jpg" width="270px" height="192px" ></a></li>  <li class="box"><a href="img/2/event1.jpg" class="magnifier" ><img alt="" src="img/2/event1.jpg" width="270px" height="192px" ></a></li>  <div class="clear"></div>  <li class="box"><a href="img/3/butterfly.png" class="magnifier" ><img alt="" src="img/3/butterfly.png" width="270px" height="192px" ></a></li>  <li class="box"><a href="img/3/cat\_like\_dragon.png" class="magnifier" ><img alt="" src="img/3/cat\_like\_dragon.png" width="270px" height="192px" ></a></li>  <li class="box"><a href="img/3/four\_colors.png" class="magnifier" ><img alt="" src="img/3/four\_colors.png" width="270px" height="192px" ></a></li>  <li class="box"><a href="img/3/lightning\_bolt.png" class="magnifier" ><img alt="" src="img/3/lightning\_bolt.png" width="270px" height="192px" ></a></li>  </div> |

**3.3Up-coming event**

**3.3.1upload**



Users can select the date at the calendar provided. Users are no need to type the date, so it can reduce the typing error. Also, It can standardize the format of the date to (year/month/day) .

Users can adjust the time by clicking the arrows on the right of the input box. It can standardize the format of collected

After filling up the form and clicking on the submit button. “uploadevent.php” will be called. The form data is set for processing to a PHP file “uploadevent.php”with the HTTP POST mothod. Information sent from a form with the POST method is invisible to others because all values are embedded within the body of the HTTP request and has no limits on the amount of information to send. Here is the code.

|  |
| --- |
| form action = "uploadevent.php" method="post">  <h5> Event : </h5>  <input type = "text" name = "event"><br>  <h5> Date : </h5>  <input type = "date" name = "date"><br>  <h5> Venue : </h5>  <input type = "text" name = "venue"><br>  <h5> Time : </h5>  <input type = "time" name = "time"><br>  <input type = "submit" class="btn btn-1" value = "Submit">  <input type = "reset" class="btn btn-1" value = "reset">  </form> |

In the PHP file “uploadevent.php”, the inputted data will be stored as variable $event, $date, $venue, $time. “$conn = mysqli\_connect()” will set up the connection of the server. The SQL code will insert the values of those variables one by one.

“if($conn->query($sql)==True)”will check whether the process of of inserting data is complete. If “if ($conn->query($sql)”is equal to False, it fails to insert the data to the database. The page will redirect users back to “event.php”: If ($conn->query($sql)”is equal to True, it inserts the data to the query successfully. The page will redirect users back to “event.php”.

|  |
| --- |
| $sql = "insert into event values('".$event."','".$date."','".$venue."','".$time."')";  echo $sql;  if($conn->query($sql)==True) {  echo "Record updated successfully";  header("Location:event.php");  }  else  {  echo "Error updating record : " . $conn->error;  header("Location:event.php");  } |

After that , the php below will connect to the database. “ "Failed to connect to MySQL: " . mysqli\_connect\_error()”will be shown if there is connection error. When there is not error, “$result = mysqli\_query()” performs a query against the database. The SQL statement "SELECT \* FROM event" will select all data from the table “event” in the database.

|  |
| --- |
| <?php  $con=mysqli\_connect("85.10.205.173","sumhohoho04","THMSSsumho04","sumho2017");  if (mysqli\_connect\_errno())  {  echo "Failed to connect to MySQL: " . mysqli\_connect\_error();  }  $result = mysqli\_query($con,"SELECT \* FROM event");  ?> |

“($row = mysqli\_fetch\_array($result)”will fetch the data from the database. The data will be stored into the variable $row. The data of the field “event” will be shown on the first column. The second one will be the data of the field ‘date’. The third one is the data of the field “venue”. The fourth one is the data of the field “time”. There will be two link button for deleting and update the data.

<tr>tag create a row of a table.<td>tag create a column of a table.

|  |
| --- |
| <?php  while($row = mysqli\_fetch\_array($result))  {  echo"<tr>  <td> ".$row['event']."</td>  <td> ".$row['date']."</td>  <td> ".$row['venue']."</td>  <td> ".$row['time']."</td>  <td align = 'center'>  <a href='#'>Delete</a>  <a href='#'>Update</a>  </td>  </tr>";  }  ?> |

**3.3.2 delete & edit**

When users click on the Delete button on the table, the data will be deleted from the database. Then, it will redirect users back to “event.php” and the SQL statement will fetch the data from database again. It will generate a new table.

When users click on the Update button, it will redirect users to other page for editing the data of the up-coming event table. There will be the original record of selected row. Also, there will be 4 input box with 4 submit buttons for editing the data. Users can edit the data they want. The data in the database will change. There will be a “back” button for redirecting users to “event.php”.

**3.4 student profile**

**3.4.1viewing**

For the student profile, there will be a link button for linking the users to profile.html”When users clicking on the name of the student, it will redirect users to the student’s profile. The users can view classmates’s information such as contact numbers.

|  |
| --- |
| <ul class="thumbnails thumbnails-1 list-services">  <li class="span3">  <div class="thumbnail thumbnail-1"> <img src="img/4/stud1.jpg" width="270px" height ="192px" alt="">  <section> <a href="./profile.html" class="link-1">Peter Chan </a>  <p>6D 01</p>  </section>  </div>  </li>  <li class="span3">  <div class="thumbnail thumbnail-1"> <img src="img/4/stud2.jpg" width="270px" height ="192px" alt="">  <section> <a href="#" class="link-1">student B </a>  <p>6D 02</p>  </section>  </div>  </li>  </ul> |

Users can go back to the classmates by clicking the button “see other”. When the button is clicked on, href="./profile.html" will redirect users to “profile.html”

3.4.2edit

Users can click on the edit button, if they want to modify the profile information. The Edit button will only be activated when they accessing their own profile. Therefore, they can only edit their own information. The database will store their information and the information will change when classmates modify their information. When they click on the button, a fill in form will pop out. Users can fill in the profile again, the new information will be sent to the database. Then, the modification is done.

1. **Testing and Evaluation**

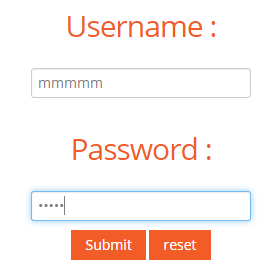
**4.1Testing**

**4.1.1Login and logout**

**4.1.1.1login**

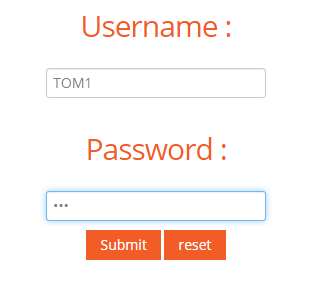
Login function is used to prevent the others who are not 6D classmates from accessing the website. Invalid data and valid data will be inputted for checking whether the function works. First invalid data will be inputted.

First set of data would be username: mmmmm and password:11111



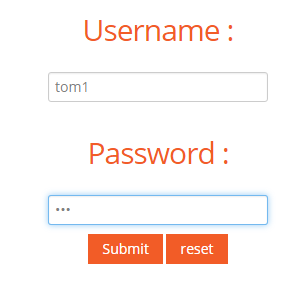
It is classified as invalid login.

Second set of data would be username:TOM1 and password:123



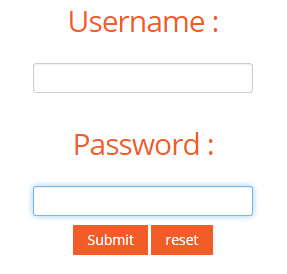
It should be an invalid login because it is in upper case. However, it is classified as a valid login and redirect users to “homepage.php”.

The third set of data is username: tom1and password:123 that it is the login account in the database.

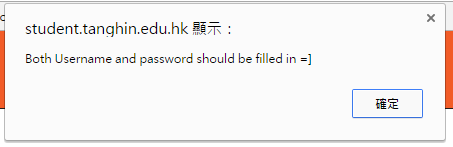


It is classified as valid login and redirect users to “homepage.php”.

For the fourth test, I leave the login form blank.



An alert box pops out and asks users to fill in all input box.



**4.1.1.2logout**

When the button  is click on, it redirects ours to login page (index.html)

When users click on the back button of the browser, it can still redirect users to previous page( homepage.php). When users refresh or access other pages, it redirects users to login page(index.html).Users can still access previous page by using the back button of the browser.

**4.1.2Gallery**

**4.1.2.1upload**

**4.1.2.2viewing~**

**4.1.3Up-coming event**

**4.1.3.1upload~**

The inputted data should be reasonable. For example, the input box “Event” and venue should not be only numbers. Also, all input box should be filled before being submitted.



For the first test, I left all input box blank.



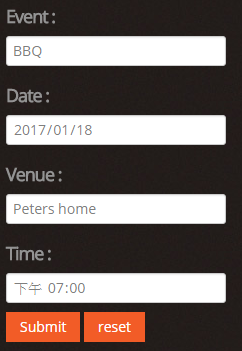
New record is generated and the record is empty. It shows that there is no default field present check.



For the second test, I inputted the Chinese words.



The record is showed correctly because the codec applied in the field is Unicode.



For the third test, I inputted English letters.



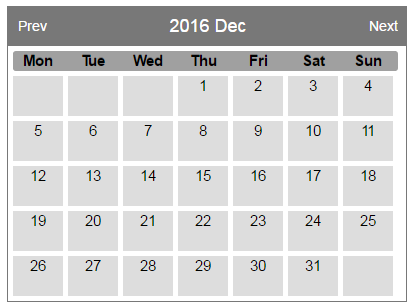
The data is showed successfully.

New record will be generated on the up-coming event table in the test. It shows that there is insufficient data validation. Although the uploading function works, there is no data validation that the output will be unreasonable or meaningless.

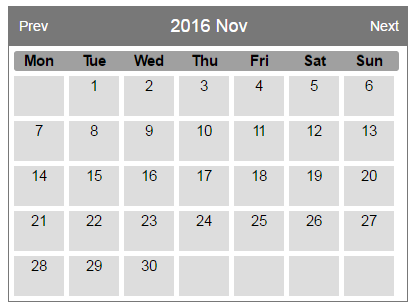
**4.4student profile**

**4.4.1viewing**

**4.5 calendar**



When users click on the button Prev and Next, the calendar will change into the calendar of previous or next month.



**4.2 Evaluation**

**4.2.1 A lack of data validation**

In order to ensure that the inputted data is valid, reasonable and meaningful for administrator and users, data validation is important because it can prevent users from inputting invalid data, for instance they may leave the input box blank by accident. This website contains few data validation. Therefore, the number of case of garbage-in-garbage-out will be high.

**4.2.2 No Captcha**

Captcha is a type of challenge-response test used in computing to determine whether or not the user is human. It prevents automated logging in that may increase the workload of the server. Also, the information may be modified by the artificial intelligent. It brings undesirable effects. Therefore, there should be a Captcha at the login section.

**4.2.3Unstable database**

I register a database from db4free.It is free of charge, but the connection is unstable. Sometimes, the loading time of logging in or uploading is relatively long. It will be a big problem if the is a lot of data in the database. There waiting will be much more longer.

4.2.4The latest event is at the bottom for the up-coming event

To catch users eyes, the most importance content should be locate on the middle top of the content section. However, the latest information of the coming activities is on the bottom of the table. The readability of the webpage will be low. It will reduce the effectiveness of the function.

4.2.5