

ADVANCED WEB PUBLISHING APPLICATIONS

Assignment I 2022

Technical Tips

To implement this page you can use 2 or more database tables (one for the users and one for the messages, and possibly one for avatars).

You also will need:

1. The page that enters/views the messages. This should have two independent components:
 - A. This should include a script that generates the HTML for the radio buttons/emotion names. I.e. this will be a PHP page but with a lot of client-side code. It should collect the data and send it to (2) below.
 - B. It should contain JavaScript that loads the content from (3) below and adds it to the page.
2. A PHP script for adding a message to the database. Used by (1)
3. A PHP script that returns the HTML for the 20 most recent messages (which can be added to the page in part 1 via Ajax). Used by (1)

Additionally, for the other parts, you will need the following pages:

4. A page to allow user to enter an emotion name and choose an avatar in order to create a new emotion. It will send data to (5)
5. A page that adds the user data to the database (collected by 4)

You can use tables similar to these. You may use an extra table for avatars if you wish.

Table: liveblogemotions

Select data **Show structure** Alter table New item

Column	Type	Comment
id	int(11) <i>Auto Increment</i>	
image	varchar(100) <i>NULL</i>	
name	varchar(100)	

Table: liveblog

Table has been altered. 22:09:41 [SQL command](#)

Select data **Show structure** Alter table New item

Column	Type	Comment
id	int(11) <i>Auto Increment</i>	
message	text	
emotion_id	int(11)	
time	datetime	

[JavaScript] URL encoding

When creating URL encoded query strings for sending data with Ajax you should escape the strings to remove any offending characters (regular HTML form submission did this automatically, but with Ajax you must explicitly do it yourself.).

E.g.

```
str= "newstatus="+escape(data) ;
```

PHP will automatically unencode the string.

[JavaScript] HTTP Method

When sending data to the PHP script the POST HTTP method may be more suitable (especially for larger amounts of text).

[JavaScript] Example of sending POST data

```
// Sample query string (yours will come from the form)
var querystring = "datatest=1234&moredata=5678";

var xmlho = new XMLHttpRequest();

: : :

xmlho.open("POST", "savemessage.php");

xmlho.setRequestHeader("Content-Type",
                        "application/x-www-form-urlencoded");

: : :

xmlho.send(querystring);
```

[PHP] HTTP Method

If using POST do this don't forget to extract the data from the `$_POST` array in your PHP (i.e. instead of the `$_GET` array)

```
$_POST['datatest']
```


[PHP] Security

Whenever we get input from a user we should assume it is malicious. If using text input from a form it is useful to convert all special characters to character entities so that they will appear as intended on a web page but cannot be used for malicious purposes.

E.g.

```
$safe_data = htmlentities($incoming_data);
```

You can also escape characters for the purposes of using them safely in MySQL.

```
$safeuser = mysqli_real_escape_string($db, $_POST['user']);
```

For this to work you may also have to set the character set for the connection.

[PHP] Users menu

For the radio buttons and emotion names you need a script that generates the necessary HTML. It can use the **id** from the database table as the value as you only need send this **id** to identify the emotion for the post.

```
<input type = "radio" name = "username" value = "1">Happy  
<input type = "radio" name = "username" value = "2">Sad  
<input type = "radio" name = "username" value = "3">Angry
```

etc.

[JavaScript] Finding Selected Radio Button

To find which radio button was selected you can loop through them.

```
var rbs = document.querySelectorAll('input[name="username"]');  
  
var selection = null;  
  
for (var i = 0; i < rbs.length; i++)  
{  
    if (rbs[i].checked)  
    {  
        selection = rbs[i].value;  
        break;  
    }  
}
```

[PHP] Timestamps

You need to store the time a message was sent. You can do this by storing the time and date separately. But you can also store a combined value.

The type you use (when creating the field) could be **datetime**.

Column	Type	Comment
id	int(11) <i>Auto Increment</i>	
message	text	
emotion_id	int(11)	
time	datetime	

To fill this field automatically you can use the **now()** SQL function.

```
$query = "insert into messages (message, emotion_id, time)  
        values ('$newmessage', $id, now());"
```

[PHP] Date time Format

If you store the time each message was added in the **datetime** format in the database, you can easily convert this to a PHP **datetime** object.

E.g. Once you retrieve the date from the MySQL results you can create a DateTime object.

```
$temptime = $row['time'];
```

```
$time = strtotime($temptime);
```

or

```
$time = new DateTime($temptime);
```

You can extract individual elements (as strings) from the date using the **date()** function, along with the datetime object, and a format string.

```
$datestring = date('d', $time);
```

```
$yearstring = date('y', $time);
```

There's more information here:

<https://www.php.net/manual/en/datetime.format.php>

[PHP] Checking for an Empty Field

In your PHP code you can check if a field is empty (e.g. the field for the user's avatar) by using **isset()** (assuming the **mysqli_fetch_array()** function stored the current row in the **\$row** variable in this example).

E.g.

```
if (isset($row['avatar'])) {  
  
}
```

[CSS/HTML] Overflow

In order to fit content into a finite space on a web page, you can specify its **height** and **width** explicitly (it will expand downwards to fit the content otherwise) and also specify its **overflow** property to **scroll** which adds scrollbars (or the **auto** property which adds scrollbars only when necessary).

E.g.

```
#display {  
  
    height: 500px;  
    width: 400px;  
  
    overflow: auto;  
}
```

Tips for Creating New Emotions

[PHP] Showing Avatars

To show the list of emotions on the **add Emotion** page you can do one of the following.

- A) Put the image files in a directory and display all the images in that directory (see later notes). When selecting the image the filename can be stored along with the user's name in a database table
- B) Store the avatar filenames in a separate table and loop through the table contents to display the images. The id of the appropriate row in that table can be stored along with the new username.

Scanning Directories

You can create a folder on the server that you can copy the image files into it.

The PHP page can then display an `` tag that displays the image automatically

You can use the following code to extract the file extension from a filename.

```
function findexts ($filename)
{
    $ext = pathinfo($filename, PATHINFO_EXTENSION) ;

    return $ext;
}
```

You can use the **scandir** function to get an array of filename from a specified directory.

```
$dir = 'images/';  
$filesInDir = scandir($dir);
```

<https://www.php.net/manual/en/function.scandir.php>

We can loop through these file names

```
foreach ($filesInDir as $item)
{
```

Display the image in \$item

```
}
```


Inside the loop we must make sure we only try to display images.

One way to do this is to create an array of acceptable extensions.

```
$imageFormats = ["png", "jpg", "jpeg", "gif"];
```

Then we can check if the file extension for the current file name we are trying to display is in that array before we try to display it.

```
$tempext = findexts($currentFile);  
  
if (in_array($tempext, $imageFormats))  
{
```

CODE TO DISPLAY IMAGE WITH TAG

```
}
```

Then the filename of the emotion image can be stored along with the emotion's name in a new table. You can leave the field blank to indicate no image.