SILESIAN UNIVERSITY OF TECHNOLOGY

FACULTY OF AUTOMATIC CONTROL, ELECTRONICS AND COMPUTER SCIENCE

Internet Technologies – project work

Brewing

Authors: Damian Pilny, Wojciech Lindenheim-Locher

Year: 2018/2019, semester: V, group: 4, section: 1

Project supervisor: dr inż. Szymon Ogonowski

Gliwice, December 2018

Table of Content

1. Introduction

The idea for this project arose because we have been interested in craft beer for a long time, but we could not find a webpage that would contain all the information and functionalities that were useful for us. Nowadays, the subject of kraft beer is becoming more and more popular. The creation of such a website will be a response to the needs of the market. That's why we decided to try to create the perfect portal dedicated to brewing.

It is directed mainly to beer lovers, however, the ordinary user will also find there a lot of interesting information. Users have the option of choosing the type of beer they are interested in, then they are redirected to the subpage where they can find details about the selected beer. To encourage users to create an account on our website for registered users, we have created the opportunity to evaluate beer and preview of your favorite beers in a separate bookmark. Unregistered user can only see the number of like and unlike, but there is no way to add rate what is to prevent spam. The website has a separate space for advertising, which will provide a future budget for further development of the portal.

Ultimately, we intend to add a forum to the website where users can share their thoughts on brewing and the brewing section. There you will find recipes for given beer and guides on how to make your own beer step by step.

The main advantage of the website is its simple and intuitive interface. It's easy to find on our website. The site is devoid of unnecessary graphic effects so that we can easily use it even on older devices. Knowing the needs of users, we took care of the compatibility of the website with smaller screens.

2. Aim and scope of the project

The aim of the project was to create a website that will be a compendium of knowledge about craft beer. On our website, we want to combine the advantages of other websites devoted to this subject and create an independent service for beer lovers. Here, every user will be able to find the necessary information on kraft beer and share their opinion on a selected beer type and confront it with other people.

The secondary objective was to broaden our knowledge about web design and group work. The project taught us how to divide tasks among team members and we could learn about many new technologies used to creating websites.

Stages of the project:

- Selection of a project theme related to the needs of the market and our hobby
- Analysis and selection of methods and technologies of project implementation After analyzing the available solutions, we decided to create a website. We did not have experience in making websites so at the beginning we had to familiarize with selected technologies and programming languages: HTML, PHP, CSS, MySQL, Apache server. We chose them because they offer great development opportunities and they are easy to learn by a novice programmer. To edit the codes, we decided to use the Notepad++ application which enabled us to work comfortably on the source code.
- Implementation of the project based on the assumed schedule and publication of the work progress on our blog once a week. First, we created a database template to enable us to develop the website with additional features in the future. Then we went to create the page.
- Website testing and error repair
- Possibilities of website development
    - Making a dedicated smartphone application
    - Adding a forum where users will be able to exchange their knowledge about brewing
    - Making a new subpage dedicated to brewing beer, where the user will find recipes and how to make a beer step by step. Adding a search engine to be able to find what kind of beer we can make from our ingredients

3. Schedule

3.1.Schedule approved at the beginning

12-19.10.2018

- Familiarization with the construction and good practices of creating a
- website
- Preparing the initial outline of the website
- Creating a basic database

19-26.10.2018

- Expansion of the database
- Designing the graphic design of the website

26-09.11.2018

- Making communication with the database
- Creating an Internet domain

09-16.11.2018

- Creating the initial possibility to navigate the website

16-23.11.2018

- Recognition of communication methods with the microcontroller
- The basic implementation of the microcontroller functionality to the
- website

23-30.11.2018

- Implementing communication with a microcontroller
- Getting to know the possibility of communication of the website with
- the functions of the microcontroller

30-07.12.2018

- Complementing the database
- Communication test of the microcontroller with the website

07-14.12.2018

- Final work, refining the details of the project
- Test of the created website

3.2.Schedule reflecting actual wok

12-19.10.2018

- Familiarization with the construction and good practices of creating a website
- Preparing the initial outline of the website
- Creating a basic database

19-26.10.2018

- Expansion of the database
- Designing the graphic design of the website

26-09.11.2018

- Making communication with the database
- Creating subpages by using Ajax

09-16.11.2018

- Creating the initial possibility to navigate the website
- Implementation of the registration module

16-23.11.2018

- Implementation of the new login and registration module
- Error repair with Ajax

23-30.11.2018

- Adding the opportunity to evaluate beer
- Fixing bugs with login module

30-07.12.2018

- Complementing the database
- adding bookmark for logged-in users where they can find types of beer which they clicked "like" button

07-14.12.2018

- Adding slider for ads to the sidebar
- Final work, refining the details of the project
- Test of the created website

4. Software implementation

The aim of the project was to create a portal where you can find all the important information about kraft beer.

The website we want to do will have:

- main page where we briefly describe what guests can find on the website
- over a dozen subpages with different types of beer
- registration and login module
- additional functions for logged in users
- the opportunity to evaluate the selected beer type
- database where the data of registered users will be stored, beers' rates and information about beer species
- slider where ads will be placed in the future

4.1. Defining the problem

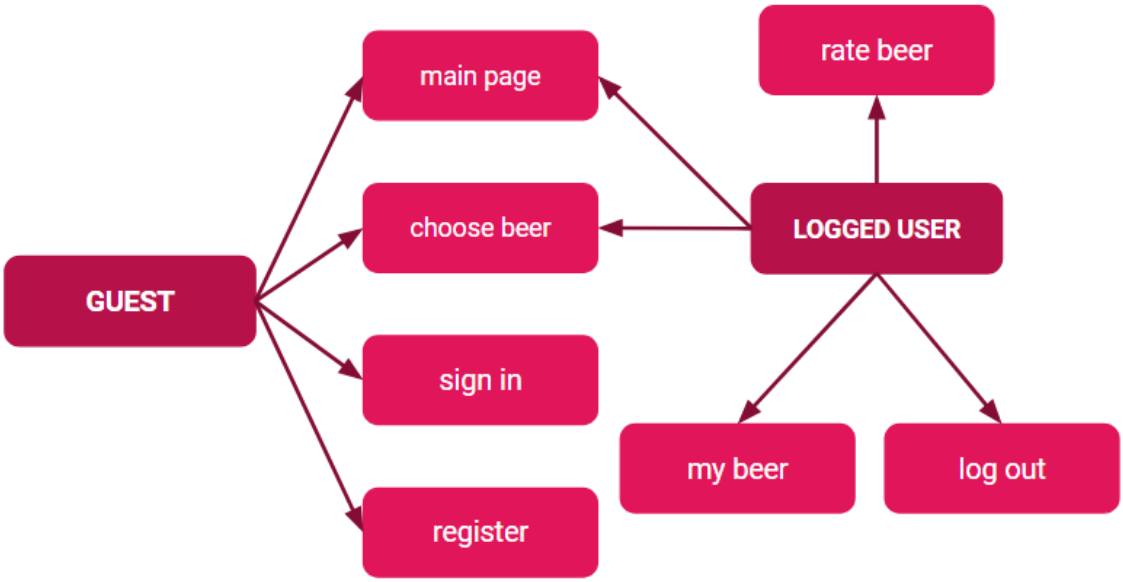

Figure 1. Website diagram

Our site has been designed in such a way that the user who is not logged in and logged in has access to various parts of the site. This is to encourage visitors to set up an account on our website.

4.2. Analysis of possible solutions

Our first task was to choose technology. Due to the fact that we did not have much experience in designing websites, we decided to use the most popular ones: HTML, PHP, MySQL. After analyzing our needs, it turned out that in order to achieve the intended goal, we would also have to use JavaScript.

As a server of our site, we used the free Xampp package. The tool we chose to create the database was phpMyAdmin. Using it, we made the database project easily and quickly. Data entry into the table was the most time-consuming.

4.3. Implementation

4.3.1. Subpages

We have several subpages on our website. Because of this, there was a problem how to do it. Initially, the only solution we knew was to make subpages in separate files. This solution did not seem to be optimal. We started to look for alternatives.

The best solution to our problem turned out to be Ajax. AJAX stands for Asynchronous JavaScript and XML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script. Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display.

This solution allowed us to refresh only the division with the content after selecting the subpage without having to refresh the entire page, which significantly improved the speed of the website. Another advantage was the lack of the need to create new files for each subpage. The content of div content is automatically retrieved from the database.

```
<script>
function load_page_details(id)
{
    $('.home_content').hide();

 $.ajax({
  url:"fetch.php",
  method:"POST",
  data:{id:id},
  success:function(data)
  {
   $('#page_details').html(data);
  }
 });
}

$('.navd a').click(function(){
 var id_rodzaju_skladnika = $(this).attr("id");
 load_page_details(id_rodzaju_skladnika);
});
</script>
```

Figure 2. Ajax code to load subpages

$ ('. home_content'). hide () - hides the content placed on the main page in the division "content".

$ ('# page_details'). html (data) - .html() is used to get the contents of '#page_details' element.

$ ('. navd a'). click (function () {} - here we determine the response to the click.

In the fetch.php file, we send a query to the database $ query = "SELECT * FROM types WHERE subsystem_ID =" ". $ _ POST [" id "]." '". This is to select all information from the "types" table on the type of beer selected by clicking the mouse. Then we select the columns we are interested in and by the function "echo $ output;" we send them to the Ajax code.

```php
$query = "SELECT * FROM rodzaje WHERE id_rodzaju_skladnika = '".$_POST["id"]."'";
$result = mysqli_query($connect, $query);
$output = '';
while($row = mysqli_fetch_array($result))
{
 $output .= '
 <h1>'.$row["nazwa"].'</h1>
 <p>'.$row["opis"].'</p>
 <h3>Smak</h3>
 <p>'.$row["smak"].'</p>
 <h3>Kraj pochodzenia</h3>
 <p>'.$row["kraj_pochodzenia"].'</p>
 ';
}
echo $output;
}
```

Figure 3. fetch.php - communication with database

4.3.2. Database

The basic element thanks to which our website works is a database. Its correct design was particularly important for the possibility of further development of the site. This caused us a lot of problems because previously we did not design a complex database. At this stage, it was necessary to think carefully what we want to have on the website and how we will use the data collected in the database. Finally, we designed a database based on five tables, which will allow us to add new functionalities in the future.

We used PHPMyAdmin to create it. The software is released under the GNU (General Public License) and allows, among other things, to create or delete databases, add or delete relations and edit their structure and content.

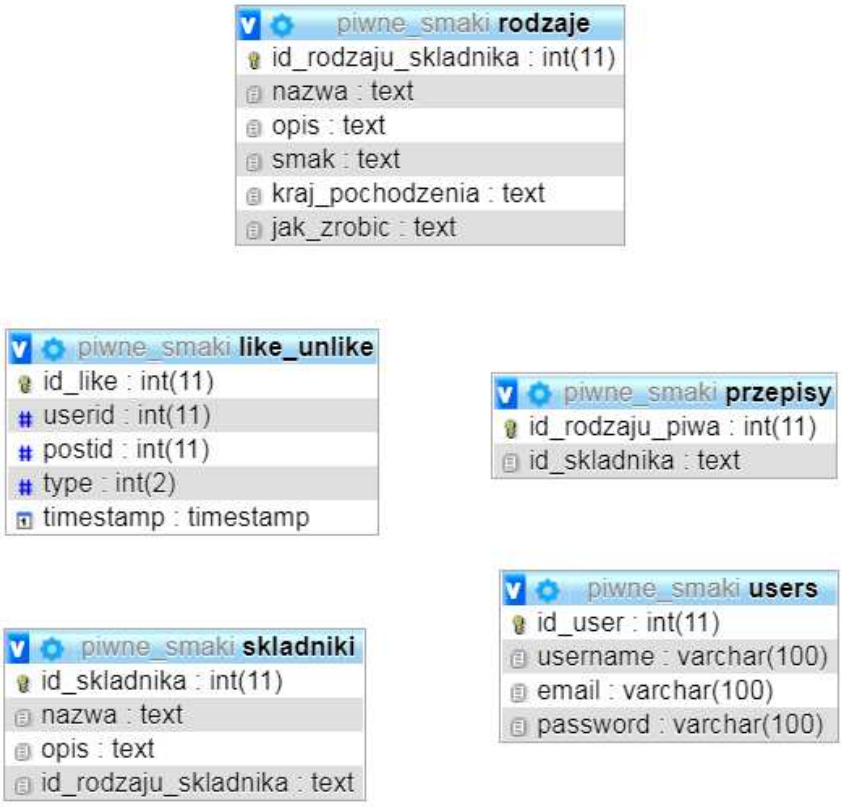

Figure 4. Table schema in the database

The mysqli_connect function is responsible for connecting the website to the database. The content on our site is in Polish, so we took care of displaying Polish characters that comply with the "utf8_polish_ci" standard.

```
$connect = mysqli_connect("localhost", "root", "", "piwne_smaki");
mysqli_query($connect, "SET CHARSET utf8");
mysqli_query($connect, "SET NAMES 'utf8' COLLATE 'utf8_polish_ci'");
```

Figure 5. Connection with database

4.3.3. Registration

A user registration system has been added to the website. We have limited this system to the minimum necessary: nickname, password, e-mail. We do not require details to be provided as this would scare off potential users. It uses a simple form " form action".

```
<form method="post" action="register.php">
    <?php include('errors.php'); ?>
    <div class="input-group">
        <label>Nick</label>
        <input type="text" name="username" value="<?php echo $username; ?>">
    </div>
    <div class="input-group">
        <label>Email</label>
        <input type="email" name="email" value="<?php echo $email; ?>">
    </div>
    <div class="input-group">
        <label>Has□o</label>
        <input type="password" name="password_1">
    </div>
    <div class="input-group">
        <label>Powtórz has□o</label>
        <input type="password" name="password_2">
    </div>
    <div class="input-group">
        <button type="submit" class="btn" name="reg_user">Zarejestruj</button>
    </div>
    <p>
        Masz ju□ konto? <a href="login.php">Zaloguj si□</a>
    </p>
</form>
```

Figure 6. Registration form

Then the data given in the form is processed by the PHP code. The following criteria are checked:

- whether all fields have been completed
- whether the given passwords are the same
- whether a user with a given nickname exists in the database

```php
if (count($errors) == 0) {
    $password = md5($password_1);
    $query = "INSERT INTO users (username, email, password)
            VALUES('$username', '$email', '$password')";
    mysqli_query($db, $query);
    $_SESSION['username'] = $username;
    $_SESSION['success'] = "zostałeś zalogowany";
    header('location: logged.php');
}
```

Figure 7. Adding user to the database

If all of the above criteria are met, the algorithm encodes the password using md5 and by using the appropriate MySQL language query, it adds the user to the database and the user is logged in to the created account. If any of the criteria is not met, the user will be informed about the errors.

Figure 8. Registration window

In the interest of user's convenience, we decided that during registration, the user will only see the registration field without unnecessary content or ads.

4.3.4. Sign in

The login system, similar to registration, is based on a simple "form action" form, in where the user's login and password are get using the "post" method. After providing this data, the correctness and compliance of the login and password in the database is checked. After logging in, the user is transferred to the main page and in the top menu the "login" and "registration" fields are replaced with "Hello nickname" and "logout". If the login details do not match, the user is notified and remains in the login tab.

A "register" button has also been placed in the login window to help unregistered users create an account.

```
if (empty($username)) {
    array_push($errors, "Nick jest wymagany");
}
if (empty($password)) {
    array_push($errors, "Has□o jest wymagane");
}
if (count($errors) == 0) {
    $password = md5($password);
    $query = "SELECT * FROM users WHERE username='$username' AND password='$password'";
    $results = mysqli_query($db, $query);

    if (mysqli_num_rows($results) == 1) {
        $_SESSION['username'] = $username;
        $_SESSION['success'] = "Zosta□e□ zalogowany";
        header('location: logged.php');
    }else {
        array_push($errors, "Z□y nick lub has□o");
```

Figure 9. PHP code to check login details

After logging in, the user gains two new functions. He can rate beers by clicking "like" or "unlike" and he can see his favourite beers in the "my beer" tab.

4.3.5. Like

Currently, many websites give us the opportunity to evaluate the content and that's why we decided to add this option to our website as well. This function has been added under every beer type. The user may vote if he is logged in to his account. If the user is not logged in, he can see the total number of votes but he can not cast his vote. The logged-in user has the option of changing the choice. Then, after the next vote, the old vote counter is decremented and the new one is incremented. After voting, the selected button changes colour.

First we have added to layout like and unlike button. Then we did function in PHP to check which user is logged in and to count post total like and unlike votes. We are using "postid" for creating like and unlike button ids because this one is use in jQuery for identifying what post button is clicked.

```
$like_query = "SELECT COUNT(*) AS cntLikes FROM like_unlike WHERE type=1 and postid=".$postid;
$like_result = mysqli_query($connect,$like_query);
$like_row = mysqli_fetch_array($like_result);
$total_likes = $like_row['cntLikes'];

$unlike_query = "SELECT COUNT(*) AS cntUnlikes FROM like_unlike WHERE type=0 and postid=".$postid;
$unlike_result = mysqli_query($connect,$unlike_query);
$unlike_row = mysqli_fetch_array($unlike_result);
$total_unlikes = $unlike_row['cntUnlikes'];
```

Figure 10. Counting post total likes and unlikes

Then we wrote the code which is responsible for sending information about the new action to the database. We took care of that the user, after casting the vote, had the opportunity to change it through the "update" query.

```
if($count == 0){
    $insertquery = "INSERT INTO like_unlike(userid,postid,type) values(".$userid.",".$postid.",".$type.")";
    mysqli_query($connect,$insertquery);
}else {
    $updatequery = "UPDATE like_unlike SET type=" . $type . " where userid=" . $userid . " and postid=" . $postid;
    mysqli_query($connect,$updatequery);
}
```

Figure 11. Update of the number of votes in the database

Everything works with Ajax, so we do not have to refresh the whole website to get the result of voting. First, we get information about the selected button. Here, if "type" value is – 0 – means user have clicked unlike button, 1 – means user have clicked like button.

```
$(".like, .unlike").click(function(){
    var id = this.id;   // Getting Button id
    var split_id = id.split("_");
    var text = split_id[0];
    var postid = split_id[1];  // postid
    // Finding click type
    var type = 0;
    if(text == "like"){
        type = 1;
    }else{
        type = 0;
    }
```

Figure 12. Ajax code to set reaction on clicking like/unlike button

Then we connect to the "likeunlike.php" file to update the sum of "like" or "unlike votes "and change the color of the button that was chosen by the user.

```
var likes = data['likes'];
var unlikes = data['unlikes'];

$("#likes_"+postid).text(likes);          // setting likes
$("#unlikes_"+postid).text(unlikes);      // setting unlikes

if(type == 1){
    $("#like_"+postid).css("color","#ffa449");
    $("#unlike_"+postid).css("color","lightseagreen");
}

if(type == 0){
    $("#unlike_"+postid).css("color","#ffa449");
    $("#like_"+postid).css("color","lightseagreen");
```

Figure 13. Updating number of votes on the website

When the like and unlike system started to work properly, we decided to use it to create a new function on the website. We have added a bookmark available only to logged-in users. There are types of beer where user clicked "like" button. After clicking on the selected type, the user will be redirected to the subpage with the beer description. To do this we used a relational databases and Ajax.

```
$query = "SELECT * FROM rodzaje, like_unlike WHERE rodzaje.id_rodzaju_skladnika = like_unlike.postid AND like_unlike.type=1 AND like_unlike.userid=$userid";
$result = mysqli_query($connect, $query);
```

Figure 14. Selection of user's favourite beers

4.3.6. Slider

The last thing what we've added to our website was slider to the sidebar. There we can placed beer ads in the future. To do it we've used JavaScript. We initialize the first image to class 'top'. The ads appear automatically after a specified period of time in the loop by function "setInterval(change, 4000);" . If the next image is not available it sets the class-top to the first image then vanish the current image to show the first one. Ad change occurs by reducing the opacity from 1 to 0 of the currently displayed image.

```
$('.slides img:first').addClass('top');
//function to alter the image index by changing the class
var change = function (){
        var curr = $('.slides img.top');
        var next = curr.next();
        if(!next.length){
            next = $('.slides img:first');
            next.addClass('top');

            curr.animate({opacity: 0.0},1000, function() {
                curr.removeClass('top');
                curr.css({opacity: 1.0});
            });

        }else{
            next.css({opacity: 0.0})
            .addClass('top')
            .animate({opacity: 1.0}, 1000, function() {
                curr.removeClass('top');
            });
```

Figure 15. Ajax code to operate the slider

4.4. Problems during application development

4.4.1. General problems

Making this website was for us the first project using HTML, CSS, MySQL, PHP, JavaScript. That was the reason for most of our problems while working on the website. The fact that we were just learning caused that we had a lot of syntax errors, which were often very difficult to locate. These errors mainly resulted from lack of experience, while over time and familiarization with technologies, the frequency of their occurrence decreased.

4.4.2. Database

The basis of our website is communication with the database. For this reason, SQL code queries appear repeatedly in the code. Often, they did not work properly or did not return the result at all.

Most problems appeared when adding the "my beer" tab. This seemingly simple modification has caused us many problems, because we do not have enough knowledge about relational databases. We wrote 50 lines of code to connect to the database and select data from one table than from the second one but it did not work.

Of course, the solution to the problem was to increase the knowledge about databases, which to some extent made our work easier. The solution that significantly accelerated our work was testing SQL queries using the phpMyAdmin tool.

4.4.3. Ajax

Ajax was a new technology for us. Initially, the implementation succeeded, it seemed that everything works properly but during the tests it turned out that there are several errors.

One of them was disappearing the text in the "content" section after pressing the "my beer" button. The solution to the problem turned out to be simple, we put a button in a separate division to which the JS script which removing the text in the "content" section was not assigned.

4.4.4. CSS

The website did not respond to changes in the CSS file. After checking whether the file names match, we started looking for a solution on the internet. It turned out that the problem probably lay in the Apache server.

The solution was simple, in the absence of a response for changes in the css file, the name of the folder in which the page files were located had to be renamed or we have to clear the cache memory.

5. Summary

During the two-month period of the project we were able to complete the work on it. Our site works correctly and has all the functionalities that we described in section 3.2. It can be generally assessed as a success.

However, the approved schedule was not done exactly point by point. Initially, we wanted to combine the Internet Technologies project and the microprocessor systems project. We quickly realized that we would not be able to make the whole brewer in such a short time, which would make communication with the microcontroller impossible. This caused that we modified our schedule. The project stages related to communication with the microcontroller have been replaced by the addition of new functions such as the registration and logging module, the ability to evaluate types of beer, the tab with favourite beers and the slider to display ads.

The change of the schedule turned out to be a good choice. Our website has gained many interesting functions. This made it more attractive to a potential user.

6. Literature

1. https://www.w3schools.com/
2. http://makitweb.com/like-unlike-using-ajax-jquery-php/
3. https://webgradients.com/
4. https://stackoverflow.com/
5. https://blog.nayma.pl/wprowadzenie-do-css-grid/
6. http://shebang.pl/kursy/podstawy-jquery/r7-ajax/
7. https://codewithawa.com/posts/complete-user-registration-system-using-php-and-mysql-database
8. http://jsfiddle.net/packetcode/0qnt7bc4/