Criterion C: Development

Complex techniques used to address the client's requirements:

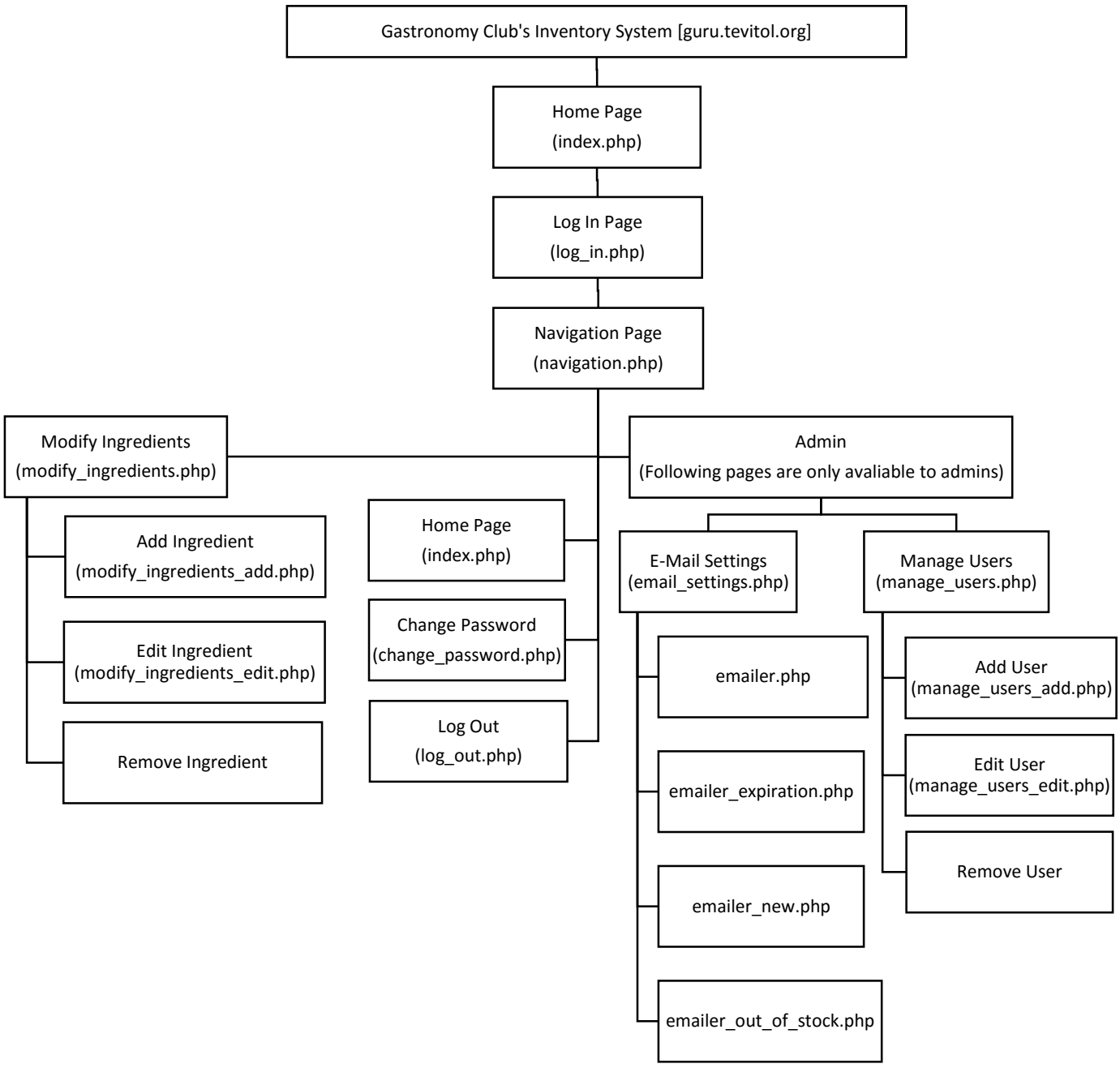
- Multimedia
    - Cascading style sheets (CSS)
    - Manipulation of codes such as HTML, PHP, JavaScript, and JQuery to customize pages and improve functionality
    - Navigation using frames and customized buttons

- Relational Databases
    - Three appropriately related tables
    - Appropriate use of data validation
    - Structured Query Language (SQL) to develop a back-end database
    - Proficient use of techniques to enable easy navigation

- Code and Programming Tools
    - Arrays
    - Loops, if–then, exit conditions
    - Parameter Passing
    - Use of MD5 hashing to protect passwords

Organization of the Product (Website)

Client's need was a browser-based interactive website that can be easily used to manage the ingredients in her activity, called Gastronomy Club. For this purpose, a website is created to meet needs of the client successfully. The structure is given below:

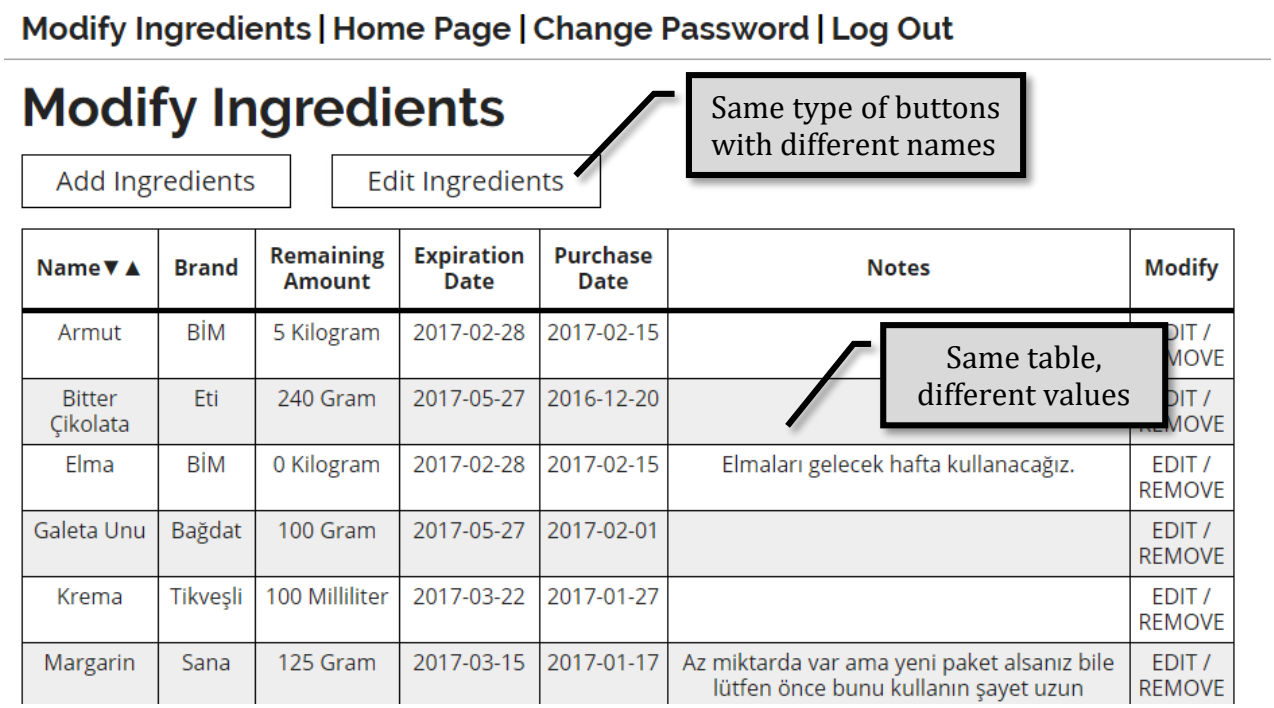


The contents are stored on the web server. The main folder contains PHP files, and the CSS folder contains corresponding CSS files. "phpmailer" folder contains the PHP Mailer library downloaded from the official website, and "db_connect.php" contains required information to connect to the database.

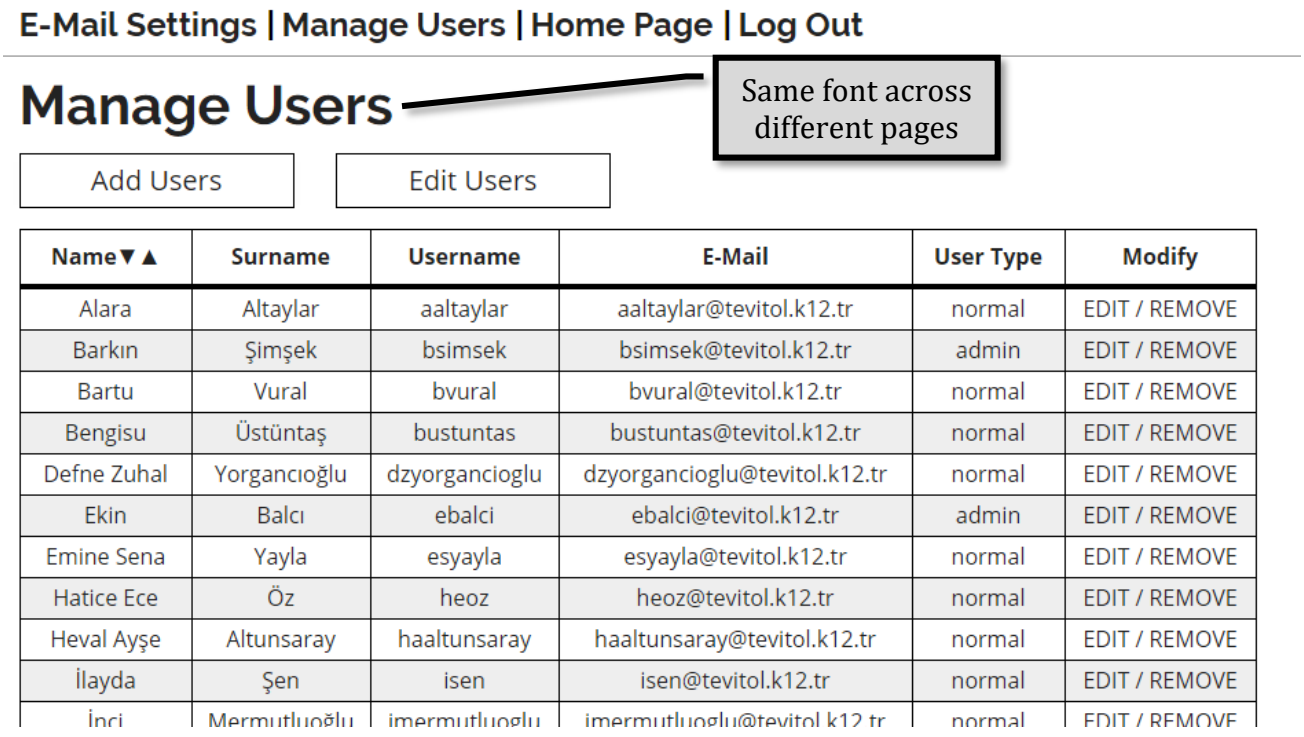
Multimedia

Cascading style sheets (CSS)

In this project, Cascading style sheet (CSS) and HTML are used together to create a consistent user interface for the Gastronomy Club's Inventory System. There are different groups of pages related to each other with the same layout, but with different content.



Name▼▲	Brand	Remaining Amount	Expiration Date	Purchase Date	Notes	Modify
Armut	BİM	5 Kilogram	2017-02-28	2017-02-15		EDIT / REMOVE
Bitter Çikolata	Eti	240 Gram	2017-05-27	2016-12-20		EDIT / REMOVE
Elma	BİM	0 Kilogram	2017-02-28	2017-02-15	Elmaları gelecek hafta kullanacağız.	EDIT / REMOVE
Galeta Unu	Bağdat	100 Gram	2017-05-27	2017-02-01		EDIT / REMOVE
Krema	Tikveşli	100 Milliliter	2017-03-22	2017-01-27		EDIT / REMOVE
Margarin	Sana	125 Gram	2017-03-15	2017-01-17	Az miktarda var ama yeni paket alsanız bile lütfen önce bunu kullanın şayet uzun	EDIT / REMOVE



Name▼▲	Surname	Username	E-Mail	User Type	Modify
Alara	Altaylar	aaltaylar	aaltaylar@tevitol.k12.tr	normal	EDIT / REMOVE
Barkın	Şimşek	bsimsek	bsimsek@tevitol.k12.tr	admin	EDIT / REMOVE
Bartu	Vural	bvural	bvural@tevitol.k12.tr	normal	EDIT / REMOVE
Bengisu	Üstüntaş	bustuntas	bustuntas@tevitol.k12.tr	normal	EDIT / REMOVE
Defne Zuhal	Yorgancıoğlu	dzyorgancioglu	dzyorgancioglu@tevitol.k12.tr	normal	EDIT / REMOVE
Ekin	Balcı	ebalci	ebalci@tevitol.k12.tr	admin	EDIT / REMOVE
Emine Sena	Yayla	esyayla	esyayla@tevitol.k12.tr	normal	EDIT / REMOVE
Hatice Ece	Öz	heoz	heoz@tevitol.k12.tr	normal	EDIT / REMOVE
Heval Ayşe	Altunsaray	haaltunsaray	haaltunsaray@tevitol.k12.tr	normal	EDIT / REMOVE
İlayda	Şen	isen	isen@tevitol.k12.tr	normal	EDIT / REMOVE
İnci	Mermutluoğlu	imermutluoglu	imermutluoglu@tevitol.k12.tr	normal	EDIT / REMOVE

Figures 1 & 2 – Same layout, different content

The integration between CSS and HTML is achieved by using division tag ( ) and HTML tags. Then assigning an id for every content will link them. Here is the tag Id's used in CSS file:

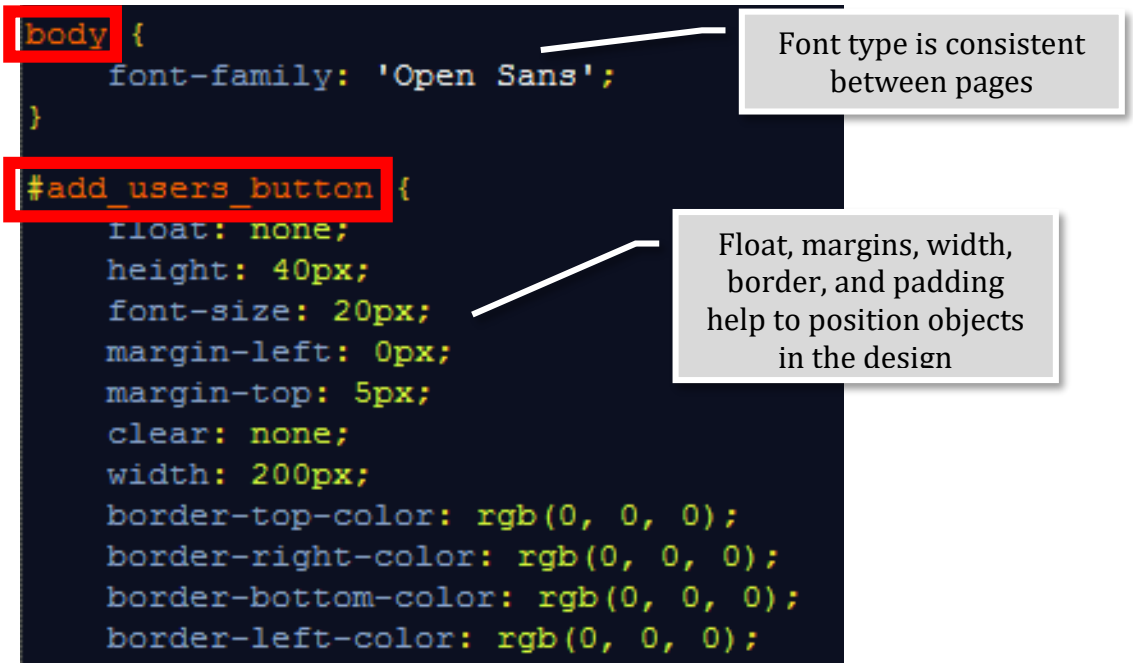


Figure 3 – Example CSS code to examine

Figure 4 – Corresponding HTML code

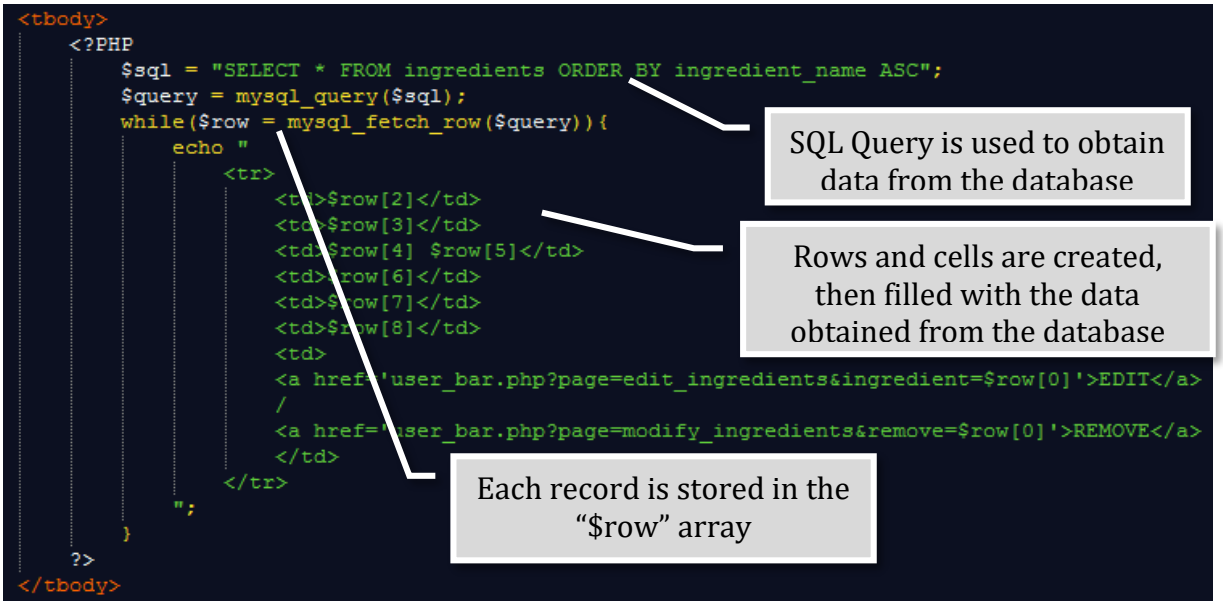
Also, the CSS file must be linked to the main page using the single line HTML code below:

```
<link rel="stylesheet" href="/css/manage_users.css">
```

Figure 5 – Example CSS linking code

Manipulation of codes such as HTML, PHP, JavaScript and JQuery to customize pages and improve functionality

The website consists of integration between HTML and PHP enhanced by JavaScript and JQuery. I used those languages because I feel confident with them, and there are many online forums to get help.
Most of the pages contain dynamically generated parts like lists and menus. The code block below is generating a table based on the information stored in the database.




```
<tbody>
    <?PHP
        $sql = "SELECT * FROM ingredients ORDER BY ingredient_name ASC";
        $query = mysql_query($sql);
        while($row = mysql_fetch_row($query)){
            echo "
                <tr>
                    <td>$row[2]</td>
                    <td>$row[3]</td>
                    <td>$row[4] $row[5]</td>
                    <td>$row[6]</td>
                    <td>$row[7]</td>
                    <td>$row[8]</td>
                    <td>
                    <a href='user_bar.php?page=edit_ingredients&ingredient=$row[0]'>EDIT</a>
                    /
                    <a href='user_bar.php?page=modify_ingredients&remove=$row[0]'>REMOVE</a>
                    </td>
                </tr>
            ";
        }
    ?>
</tbody>
```

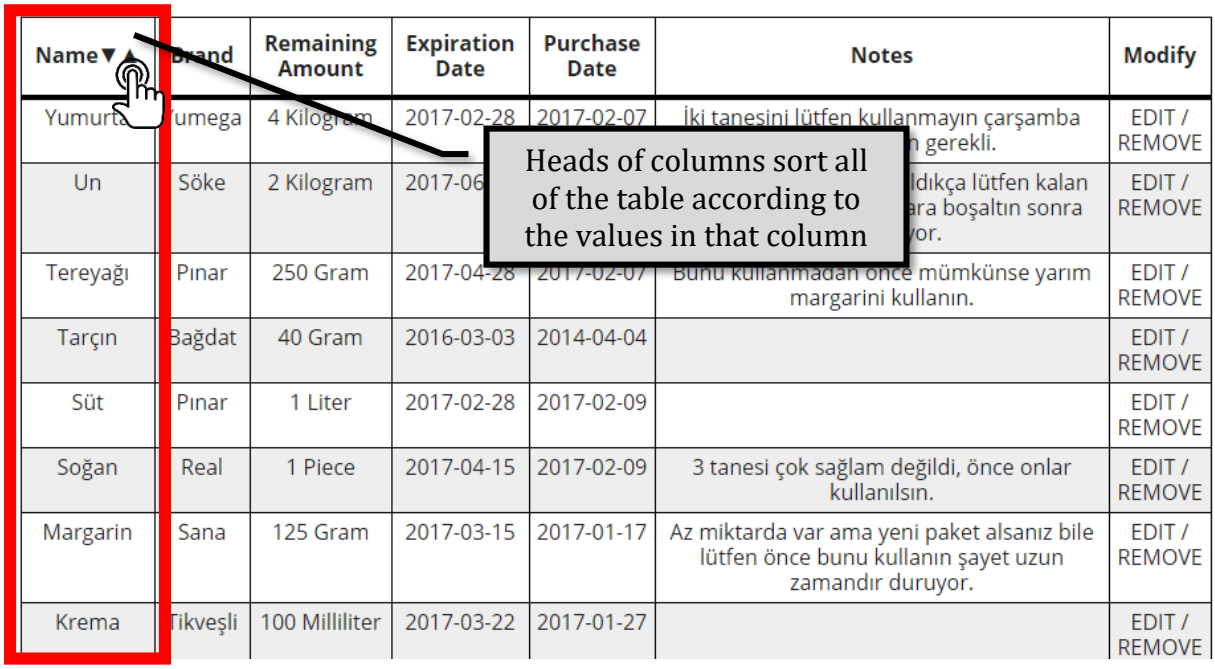
Figure 6 – Code that creates the ingredient list

Resulting ingredient list table is given below:

Name▼▲	Brand	Remaining Amount	Expiration Date	Purchase Date	Notes	Modify
Armut	BİM	5 Kilogram	2017-02-28	2017-02-15		EDIT / REMOVE
Bitter Çikolata	Eti	240 Gram	2017-05-27	2016-12-20		EDIT / REMOVE
Elma	BİM	0 Kilogram	2017-02-28	2017-02-15	Elmaları gelecek hafta kullanacağız.	EDIT / REMOVE
Galeta Unu	Bağdat	100 Gram	2017-05-27	2017-02-01		EDIT / REMOVE
Krema	Tikveşli	100 Milliliter	2017-03-22	2017-01-27		EDIT / REMOVE
Margarin	Sana	125 Gram	2017-03-15	2017-01-17	Az miktarda var ama yeni paket alsanız bile lütfen önce bunu kullanın şayet uzun zamandır duruyor.	EDIT / REMOVE
Soğan	Real	1 Piece	2017-04-15	2017-02-09	3 tanesi çok sağlam değildi, önce onlar kullanılsın.	EDIT / REMOVE
Süt	Pınar	1 Liter	2017-02-28	2017-02-09		EDIT / REMOVE

Figure 7 – Created table

The Same algorithm is used wherever dynamic lists are generated in the project. Furthermore, the functionality of the tables are improved by using a JavaScript code to sort the list alphabetically according to any column available:



Name	Brand	Remaining Amount	Expiration Date	Purchase Date	Notes	Modify
Yumurta	Yumega	4 Kilogram	2017-02-28	2017-02-07	İki tanesini lütfen kullanmayın çarşamba gerekli.	EDIT / REMOVE
Un	Söke	2 Kilogram	2017-06		ldıkça lütfen kalan ara boşaltın sonra yor.	EDIT / REMOVE
Tereyağı	Pınar	250 Gram	2017-04-28	2017-02-07	Bunu kullanmadan önce mümkünse yarım margarini kullanın.	EDIT / REMOVE
Tarçın	Bağdat	40 Gram	2016-03-03	2014-04-04		EDIT / REMOVE
Süt	Pınar	1 Liter	2017-02-28	2017-02-09		EDIT / REMOVE
Soğan	Real	1 Piece	2017-04-15	2017-02-09	3 tanesi çok sağlam değildi, önce onlar kullanılsın.	EDIT / REMOVE
Margarin	Sana	125 Gram	2017-03-15	2017-01-17	Az miktarda var ama yeni paket alsanız bile lütfen önce bunu kullanın şayet uzun zamandır duruyor.	EDIT / REMOVE
Krema	Tikveşli	100 Milliliter	2017-03-22	2017-01-27		EDIT / REMOVE

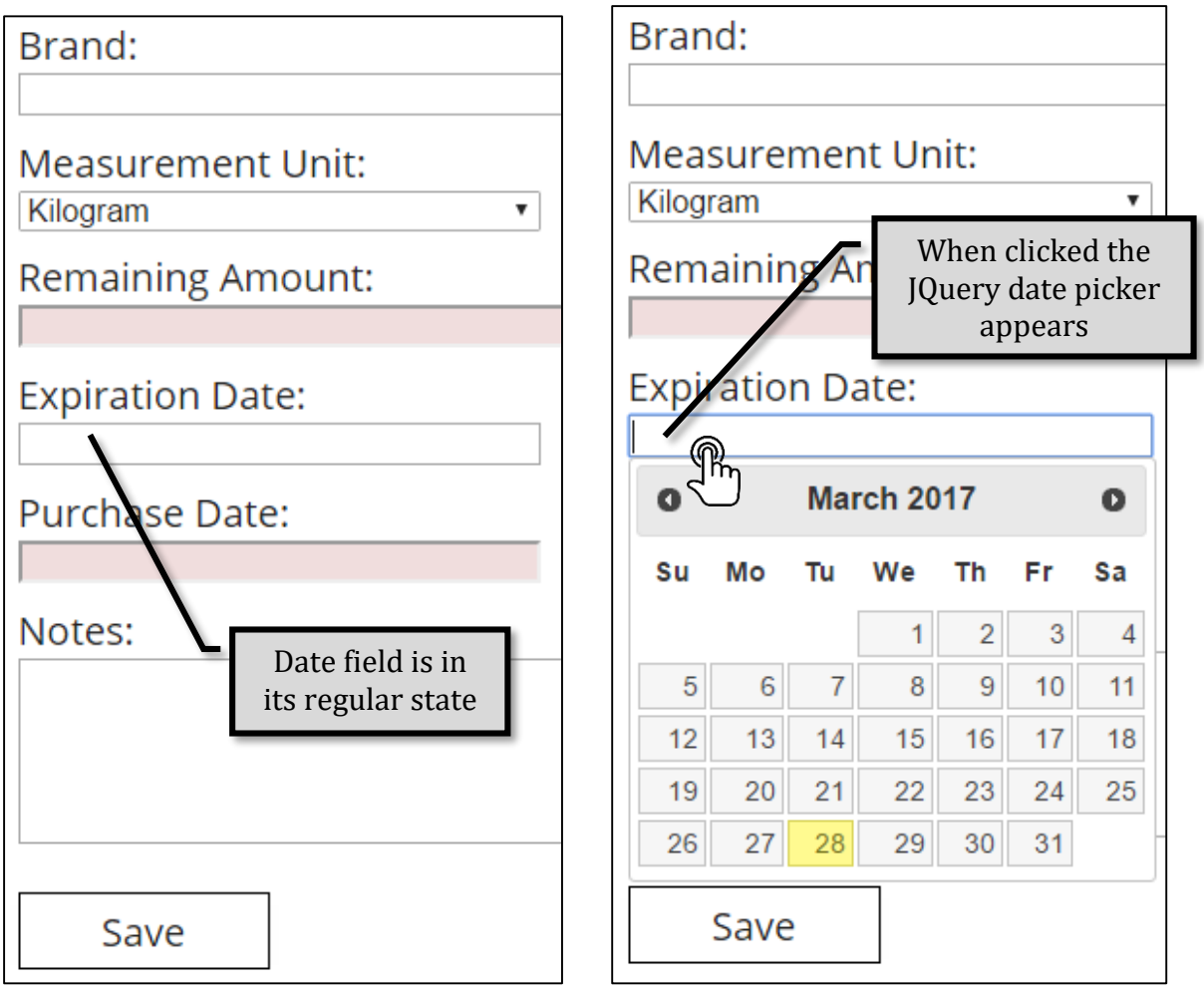
Figure 8 – Sort listed ingredient table

The JavaScript code used to sort the tables can be seen below:

```
<script type="text/javascript">
    function sortTable(table, col, reverse) {
        var tb = table.tBodies[0], // use `<tbody>` to ignore `<thead>` and `<tfoot>` rows
            tr = Array.prototype.slice.call(tb.rows, 0), // put rows into array
            i;
        reverse = -((+reverse) || -1);
        tr = tr.sort(function (a, b) { // sort rows
            return reverse // `-1 *` if want opposite order
                * (a.cells[col].textContent.trim() // using `.textContent.trim()` for test
                    .localeCompare(b.cells[col].textContent.trim())
                   );
        });
        for(i = 0; i < tr.length; ++i) tb.appendChild(tr[i]); // append each row in order
    }

    function makeSortable(table) {
        var th = table.tHead, i;
        th && (th = th.rows[0]) && (th = th.cells);
        if (th) i = th.length;
        else return; // if no `<thead>` then do nothing
        while (--i >= 0) (function (i) {
            var dir = 1;
            th[i].addEventListener('click', function () {sortTable(table, i, (dir = 1 - dir))});
        }(i));
    }

    function makeAllSortable(parent) {
        parent = parent || document.body;
        var t = parent.getElementsByTagName('table'), i = t.length;
        while (--i >= 0) makeSortable(t[i]);
    }

    window.onload = function () {makeAllSortable();};
</script>
```

Figure 9 – Corresponding JavaScript code

Another example is the date picker functionality used in the add ingredients page. There are many ways of formatting the date information. Therefore, a JQuery date picker is added to the purchase date and expiration date fields to both help users and prevent them from entering the date in the wrong format.



Figures 10 & 11 – Date picker example

The JQuery code used can be seen below:

```
<link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
<script src="https://code.jquery.com/jquery-1.12.4.js"></script>
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
<script>
    $( function() {
        $( "#ingredient_remaining_input1" ).datepicker({
            dateFormat: "yy-mm-dd"
        });
    } );
    $( function() {
        $( "#ingredient_remaining_input12" ).datepicker({
            dateFormat: "yy-mm-dd"
        });
    } );
</script>
```

Links to the JQuery library stored on the official JQuery website

JQuery date picker functions used in the page

Figure 12 – Corresponding JQuery code

Navigation using frames and customized buttons

Navigation of website entirely relies on the customized buttons. Therefore, CSS is used to make them interactive and provide easy access for users.

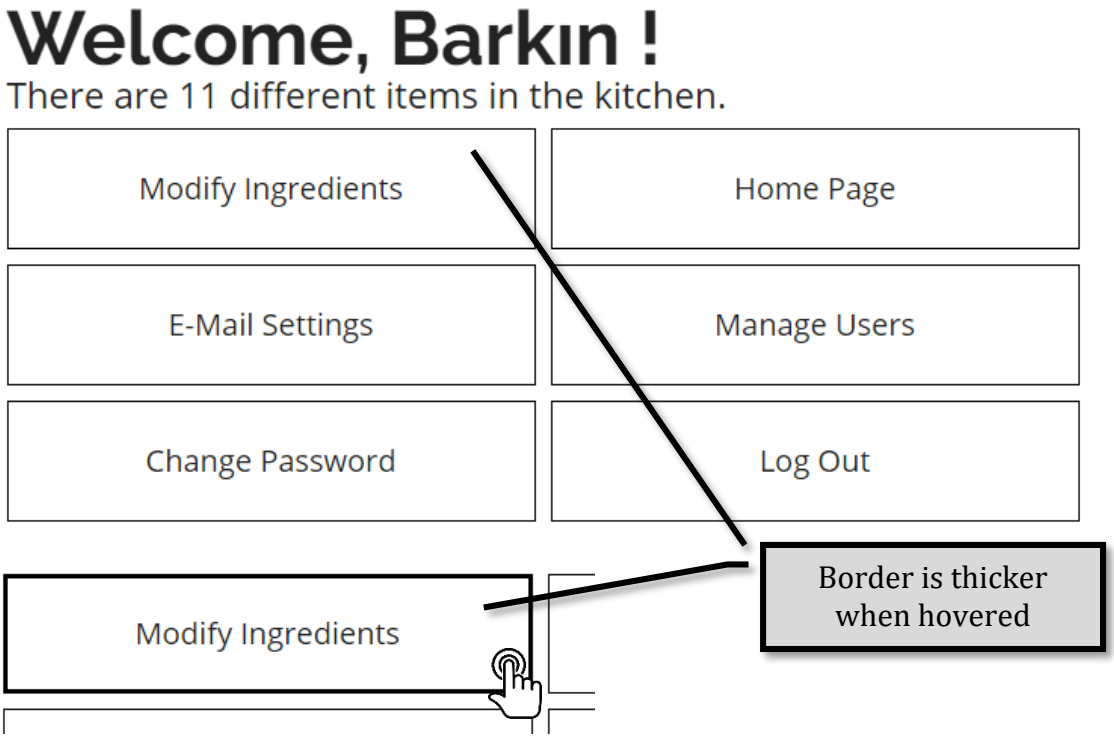


Figure 13 – CSS button example

```
<a href="user_bar.php?page=modify_ingredients">
    <input id="modify_ingredients_button" type="button" value="Modify Ingredients"></input>
</a>
```

HTML code of the "Modify Ingredient" button

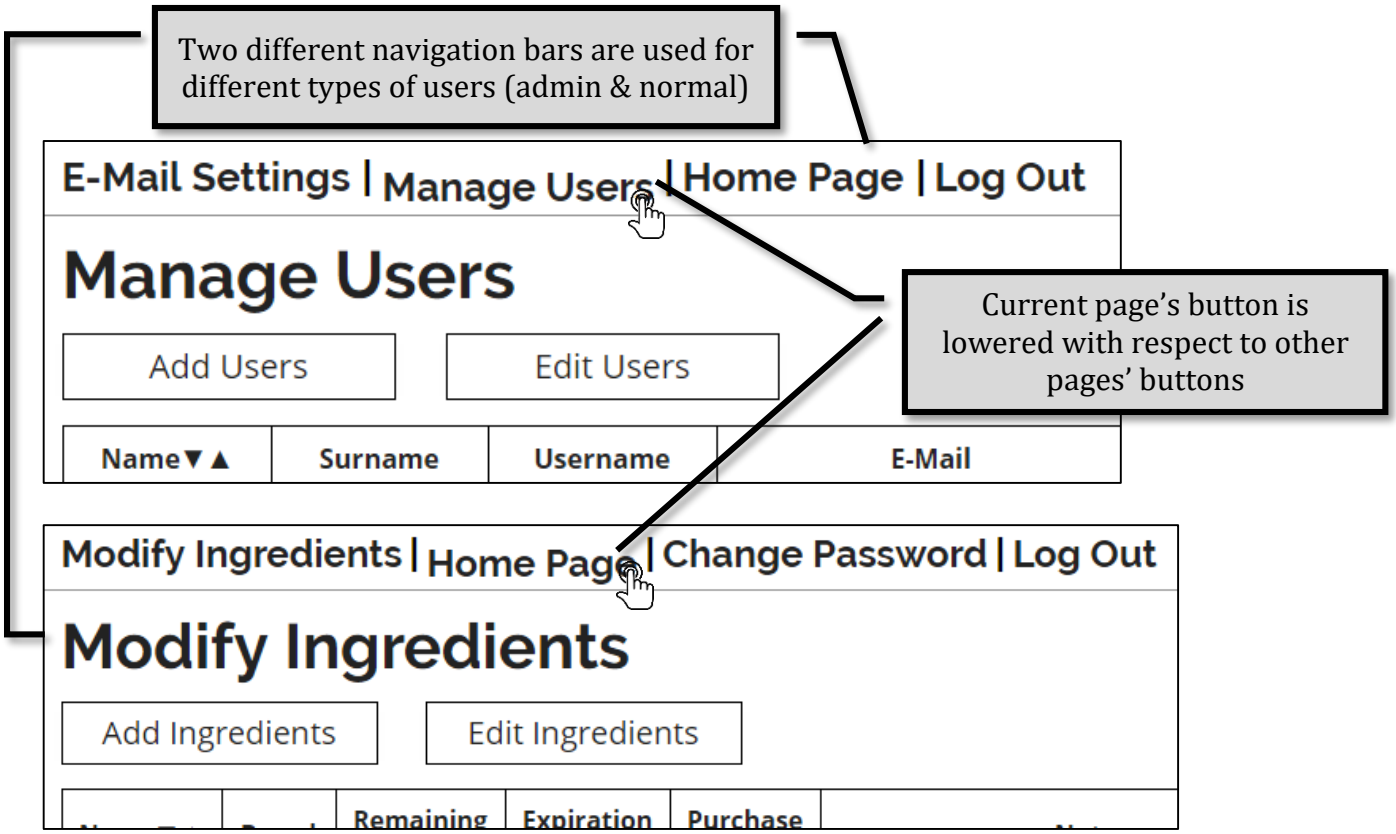
```
#modify_ingredients_button {
    float: left;
    height: 80px;
    font-size: 20px;
    margin-left: 0px;
    margin-top: 10px;
    clear: both;
    width: 350px;
    border-top-left-radius: 0px;
    border-top-right-radius: 0px;
    border-bottom-right-radius: 0px;
    border-bottom-left-radius: 0px;
    border-top-color: rgb(0, 0, 0);
    border-right-color: rgb(0, 0, 0);
    border-bottom-color: rgb(0, 0, 0);
    border-left-color: rgb(0, 0, 0);
    background-color: rgb(255, 255, 255);
    margin-right: 0px;
    display: block;
}
```

Hover and normal appearance are separated

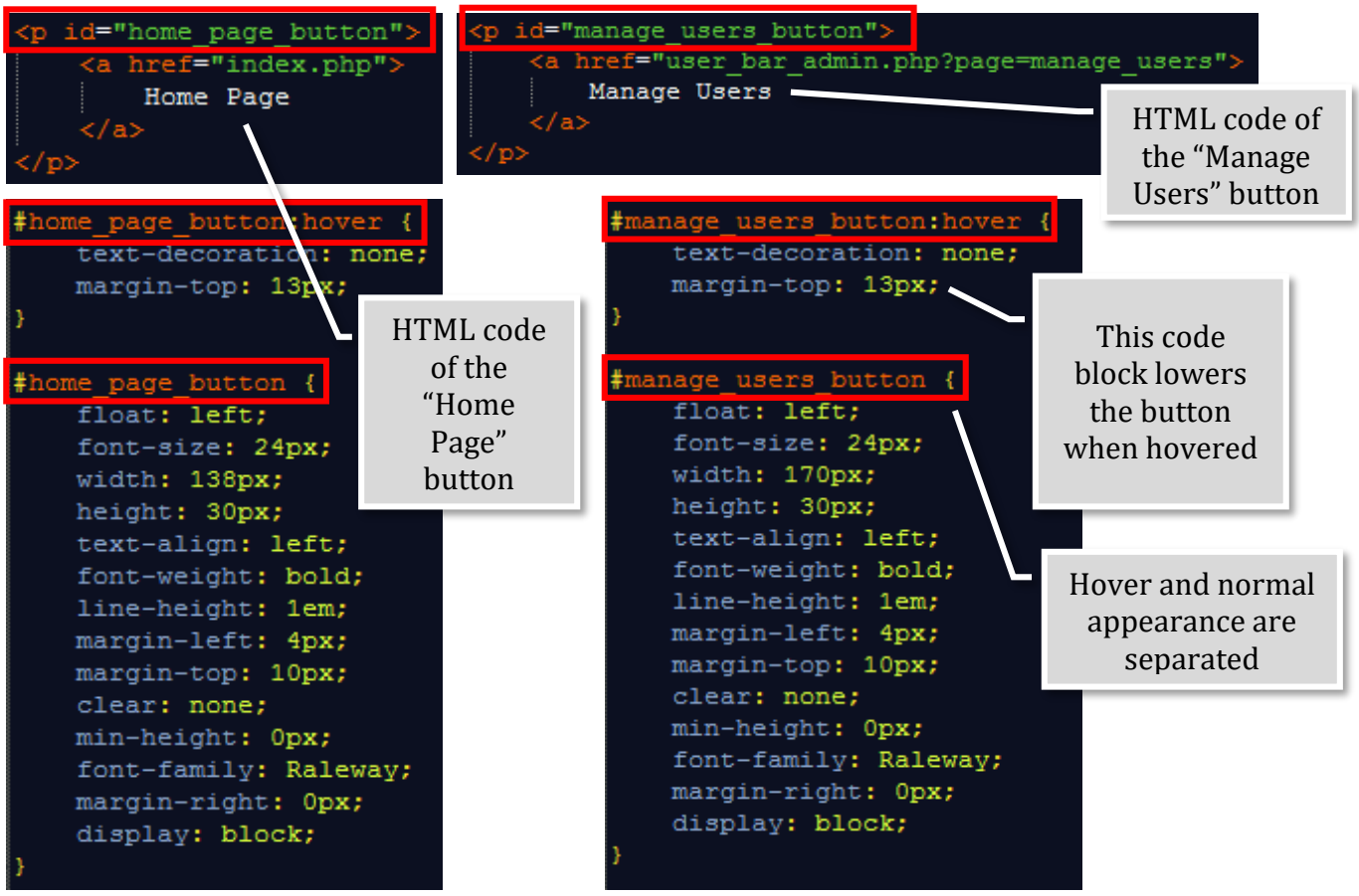
```
#modify_ingredients_button:hover {
    border-top-width: 2px;
    border-right-width: 2px;
    border-bottom-width: 2px;
    border-left-width: 2px;
    border-top-style: double;
    border-right-style: double;
    border-bottom-style: double;
    border-left-style: double;
}
```

This code block makes button's border double when hovered

Figures 14, 15, & 16 – Corresponding HTML and CSS codes



Figures 17 & 18 – Different navigation bars for different user levels



Figures 19, 20, 21 & 22 – Corresponding HTML and CSS codes

Relational Databases

Three appropriately related tables

Since Gastronomy Club's Inventory System is based on data transactions, the database is an essential part of the project. Considering the total amount of data transactions required, a relational database should be used to avoid data redundancy. The database includes three tables as can be seen in entity relationship below.

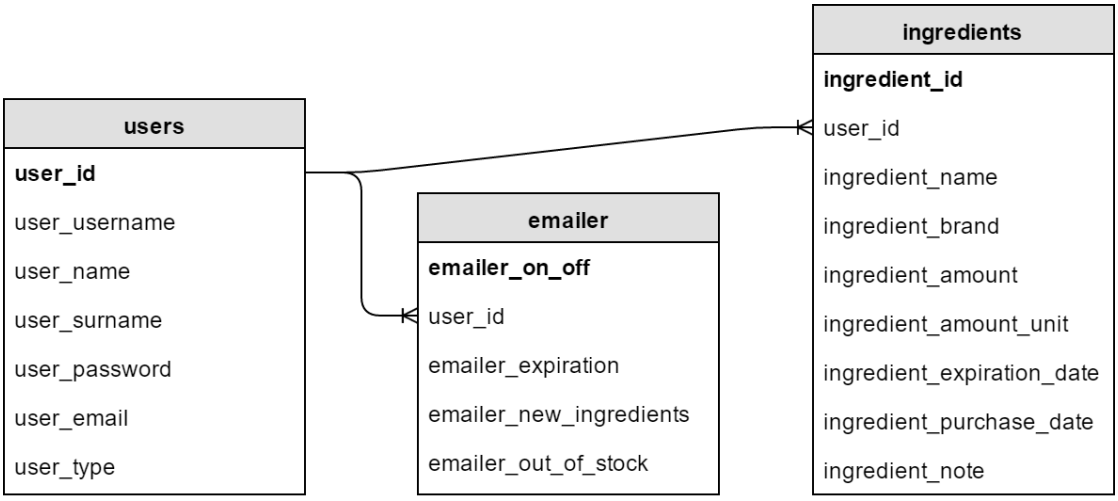


Figure 23 – Entity relationship tables

#	Column	Type	Collation	Attributes	Null	Default	Extra	Action
1	emailer_on_off	tinyint(1)			No	1		Change Drop More
2	emailer_expiration	tinyint(1)			No	1		Change Drop More
3	emailer_new_ingredients	tinyint(1)			No	1		Change Drop More
4	emailer_out_of_stock	tinyint(1)			No	1		Change Drop More
5	user_id	int(50)			No	None		Change Drop More

#	Column	Type	Collation	Attributes	Null	Default	Extra	Action
1	ingredient_id	int(100)			No	None	AUTO_INCREMENT	Change Drop More
2	user_id	int(50)			No	None		Change Drop More
3	ingredient_name	varchar(100)	utf8_turkish_ci		No	None		Change Drop More
4	ingredient_brand	varchar(100)	utf8_turkish_ci		No	None		Change Drop More
5	ingredient_amount	varchar(100)	utf8_turkish_ci		No	None		Change Drop More
6	ingredient_amount_unit	varchar(100)	utf8_turkish_ci		No	None		Change Drop More
7	ingredient_expiration_date	date			No	None		Change Drop More
8	ingredient_purchase_date	date			No	None		Change Drop More
9	ingredient_note	varchar(1000)	utf8_turkish_ci		No	None		Change Drop More

#	Column	Type	Collation	Attributes	Null	Default	Extra	Action
1	user_id	int(50)			No	None	AUTO_INCREMENT	Change Drop More
2	user_username	varchar(50)	utf8_turkish_ci		No	None		Change Drop More
3	user_name	varchar(50)	utf8_turkish_ci		No	None		Change Drop More
4	user_surname	varchar(50)	utf8_turkish_ci		No	None		Change Drop More
5	user_password	varchar(50)	utf8_turkish_ci		No	5f4dcc3b5aa765d61d8327deb882cf99		Change Drop More
6	user_email	varchar(50)	utf8_turkish_ci		No	None		Change Drop More
7	user_type	varchar(50)	utf8_turkish_ci		No	None		Change Drop More

Figures 24, 25 & 26 – Screenshots from phpMyAdmin interface

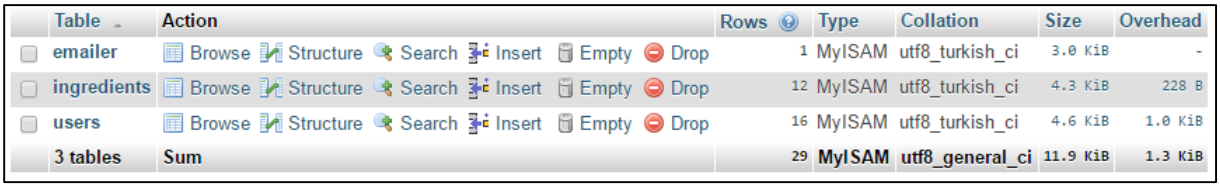


	Table	Action	Rows	Type	Collation	Size	Overhead
☐	emailer	Browse Structure Search Insert Empty Drop	1	MyISAM	utf8_turkish_ci	3.0 KiB	-
☐	ingredients	Browse Structure Search Insert Empty Drop	12	MyISAM	utf8_turkish_ci	4.3 KiB	228 B
☐	users	Browse Structure Search Insert Empty Drop	16	MyISAM	utf8_turkish_ci	4.6 KiB	1.0 KiB
	3 tables	Sum	29	MyISAM	utf8_general_ci	11.9 KiB	1.3 KiB

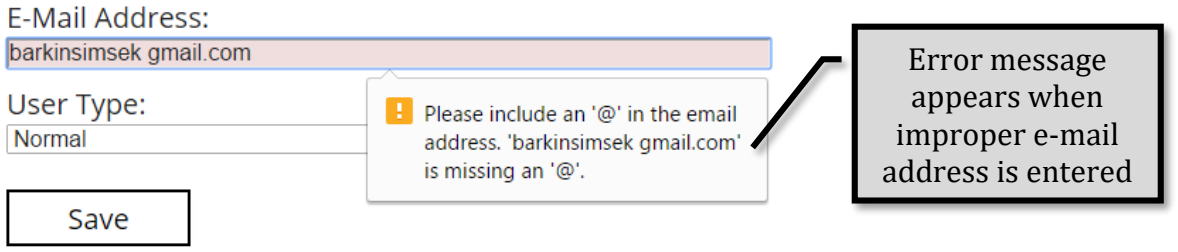
Figure 27 – General structure of the database as seen on the phpMyAdmin interface

Those three tables are emailer, ingredients, and users. PhpMyAdmin doesn't show tables in the relational view, but it can be seen from the entity relationship diagram that user_id is the primary key in users table and foreign key in ingredients and emailer tables.

Appropriate use of data validation

There are certain data types and lengths for tables in the database, which means if an improper data is available in the query, results of that query will be rejected. By that way, data validation for the database is provided, and any improper data is rejected. For example, a user_id can only be a number with maximum length of fifty characters. Anything beyond that will be rejected.

There are also other types of data validations used in the project. For example, on the add users page, email address input should be a proper e-mail address with "@" symbol. If anything improper is entered, the browser will display an error box.

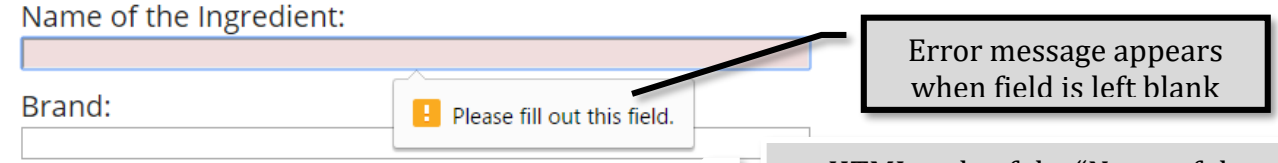



```
<label id="user_e-mail_group">
    <p id="user_e-mail_text">
    E-Mail Adress&#x3a;
    </p>
    <input id="user_e-mail_input" type="email" value="" name="user_email"></input>
</label>
```

HTML code of the "E-Mail Address" field and email validation

Figures 28 & 29 – First data validation example and corresponding HTML code

The other example is in the add ingredients page. Users cannot leave the name input blank, and the browser will display an error message if users don't fill in the name field.




```
<label id="ingredient_name_group">
    <p id="ingredient_name_text">
    Name of the Ingredient&#x3a;
    </p>
    <input id="ingredient_name_input" type="text" value="" name="ingredient_name" required="required"></input>
</label>
```

HTML code of the "Name of the Ingredient" field and data validation

Figures 30 & 31 – Second data validation example and corresponding HTML code

Structured Query Language (SQL) to develop a back-end database

SQL commands are used mostly to obtain data from the database and make changes to the database when needed. Commands such as SELECT, INSERT, UPDATE, DELETE and MySQL commands such as mysql_query(), mysql_fetch_array() are used to obtain data from the database.

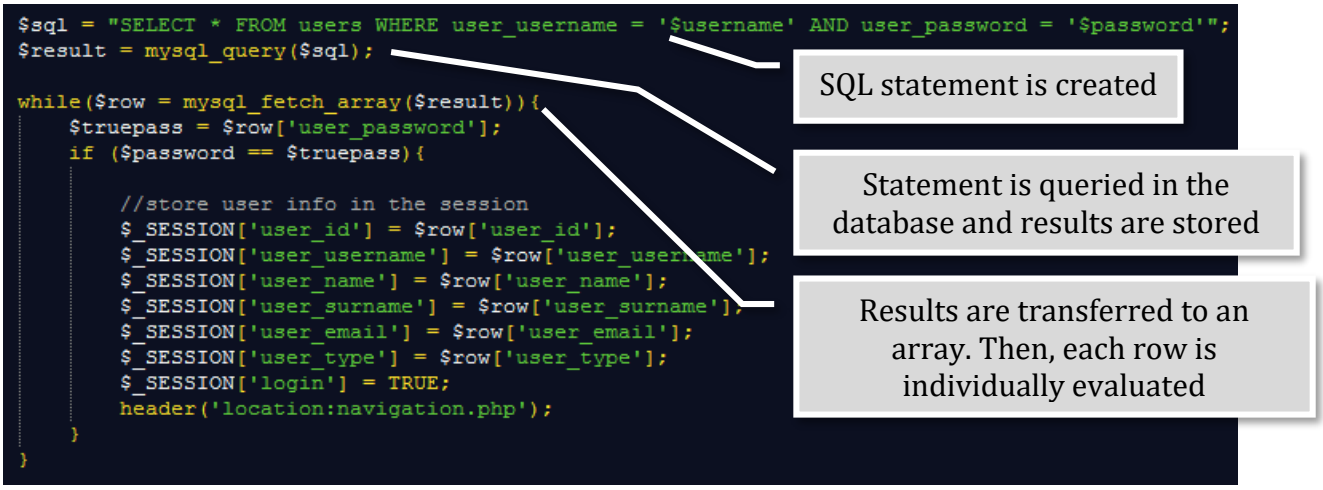
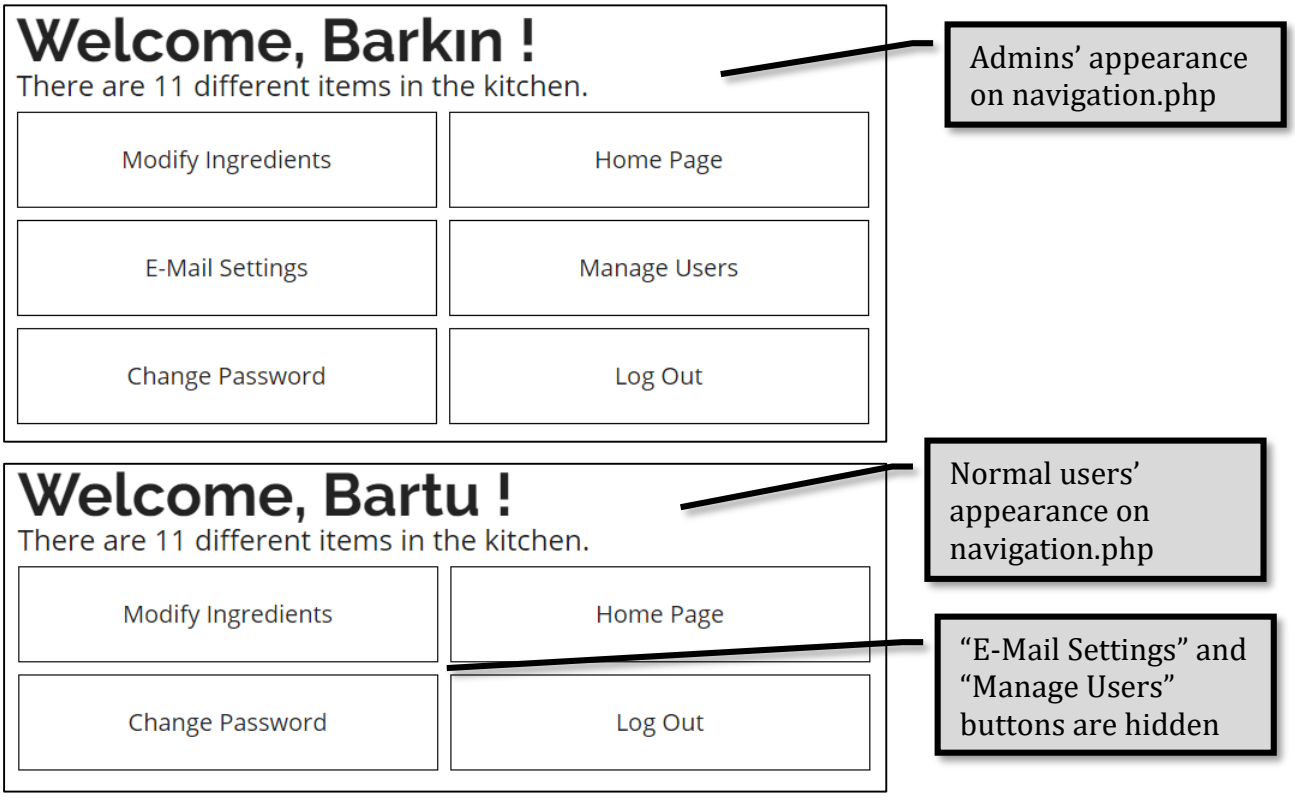


Figure 32 – SQL commands used in the log_in.php

Proficient use of techniques to enable easy navigation

The website is navigated through customized buttons and text links. Not every user has access to all pages, but only admins have permissions to access specific pages. Therefore, some of those buttons don't appear if the current user is a normal user. However, if the current user is an admin, all links and buttons appear on the web page.



Figures 33 & 34 – Example admin and normal user accounts to show hidden buttons

Figure 35 – Corresponding HTML and PHP codes

Code and programming tools

Arrays

Arrays are used very widely in the website, and it is a good way of storing a large amount of data as a list in a single variable. Here is an example:

```php
$today = date('Y-m-d');
$expired_ingredients_to_be_notified = array();

$sql = "SELECT * FROM ingredients";
$query = mysql_query($sql);
while($row = mysql_fetch_row($query)){
    //compare today's date and the expiration date on the database
    $ingredient_expiration_date = $row[6];
    $ingredient_id = $row[0];
    $date1 = date_create($today);
    $date2 = date_create($ingredient_expiration_date);
    $diff = date_diff($date1,$date2);
    $diff_variable = $diff->format("%R%a");

    //get the list of expired ingredients
    if($diff_variable < 10){
        array_push($expired_ingredients_to_be_notified, $ingredient_id);
    }
}
```

Figure 36 – Array example

In the example, data of the ingredients are gathered from the database using SQL commands. Then, mysql_fetch_row() command arranges gathered data into an array.

Also, the ingredients that are about to expire are added to the array called "$expired_ingredients_to_be_notified" by using the array_push() command.

Loops, if–then, exit conditions

Loops are used very widely in the website, and they are used to automate processes that repeat. In the example above, code block checks every column in a table in the database in a while loop:

```php
<?PHP
    $sql = "SELECT * FROM ingredients ORDER BY ingredient_name ASC";
    $query = mysql_query($sql);
    while($row = mysql_fetch_row($query)){
        echo "
            <tr>
                <td>$row[2]</td>
                <td>$row[3]</td>
                <td>$row[4] $row[5]</td>
                <td>$row[6]</td>
                <td>$row[7]</td>
                <td>$row[8]</td>
            </tr>
        ";
    }
?>
```

Figure 37 – First loops, if–then, exit conditions example

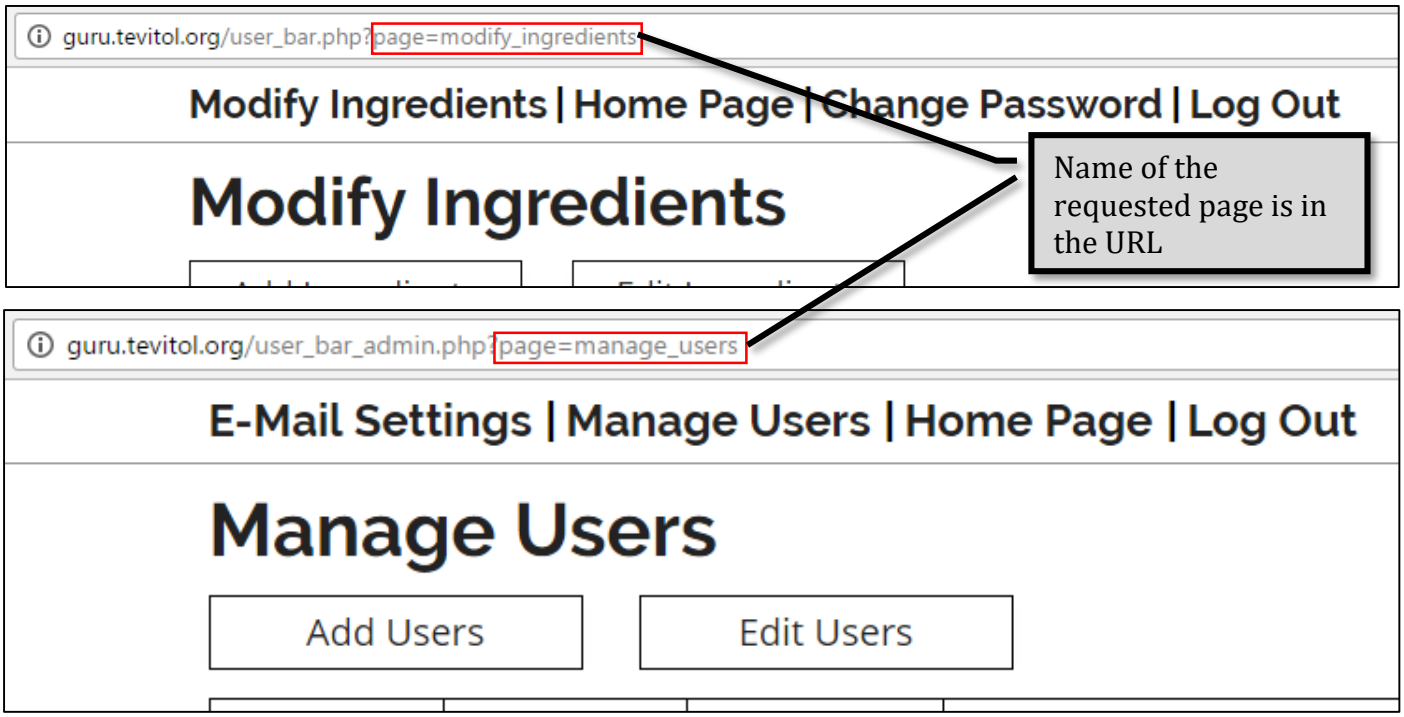
Loops are also used to count and assign elements of the arrays to variables:

```php
//Set who the message is to be sent to
$x = 0;
while($x < sizeof($recipients)){
    $adress = $recipients[$x];
    $mail->addAddress($adress);
    $x = $x + 1;
}
```

Figure 38 – Second loops, if–then, exit conditions example

Parameter Passing

The website relies on URL parameter passing for navigation. “user_bar.php” and “user_bar_admin.php” pages get the name of the pages from URL and show the corresponding page.



Figures 39 & 40 – Page names are in the URL

```php
$requested_page_id = htmlspecialchars($_GET["page"]);

if($requested_page_id == modify_ingredients){
    $page_name = "Modif Ingredients";
} else if($requested_page_id == change_password){
    $page_name = "Change Password";
} else if($requested_page_id == add_ingredients){
    $page_name = "Add Ingredients";
} else if($requested_page_id == edit_ingredients){
    $page_name = "Edit Ingredients";
} else{
    header('location:user_bar.php?page=modify_ingredients');
}
```

Command to get variables from URL

Evaluate the variable given in the URL

```php
<?PHP
    if($requested_page_id == modify_ingredients){
        require('modify_ingredients.php');
    }else if($requested_page_id == change_password){
        require('change_password.php');
    } else if($requested_page_id == add_ingredients){
        require('modify_ingredients_add.php');
    } else if($requested_page_id == edit_ingredients){
        require('modify_ingredients_edit.php');
    }
?>
```

Load the corresponding page

Figures 41 & 42 – Corresponding PHP codes

URL is also used for passing information across pages:

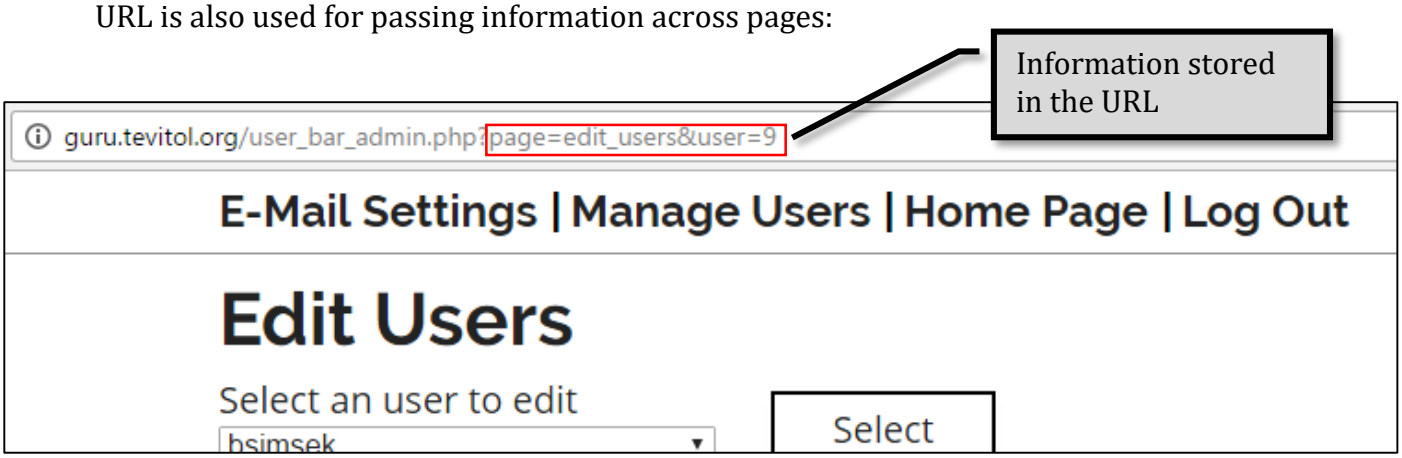


Figure 43 – URL information passing

Use of MD5 hashing to protect passwords

Since security is one of the most important issues, user passwords should be stored securely as well. With this aim, MD5 hashing is used to encrypt passwords. Since MD5 is irreversible, people with malicious intentions cannot decrypt passwords stored in the database to learn the real passwords.

user_id	user_username	user_name	user_surname	user_password	user_email	user_type
1	bsimsek	Barkın	Şimşek	5f4dcc3b5aa765d61d8327deb882cf99	bsimsek@tevitol.k12.tr	admin
2	ebalci	Ekin	Balcı	5f4dcc3b5aa765d61d8327deb882cf99	ebalci@tevitol.k12.tr	admin
6	mudursun	Mustafa Uğur	Dursun	5f4dcc3b5aa765d61d8327deb882cf99	mudursun@tevitol.k12.tr	admin
7	ouacarli	Onur Ulaş	Acarlı	5f4dcc3b5aa765d61d8327deb882cf99	ouacarli@tevitol.k12.tr	normal
8	dzyorgancioglu	Defne Zuhal	Yorgancıoğlu	5f4dcc3b5aa765d61d8327deb882cf99	dzyorgancioglu@tevitol.k12.tr	normal
9	aaltaylar	Alara	Altaylar	5f4dcc3b5aa765d61d8327deb882cf99	aaltaylar@tevitol.k12.tr	normal
10	bvural	Bartu	Vural	5f4dcc3b5aa765d61d8327deb882cf99	bvural@tevitol.k12.tr	normal

Figure 44 – Encrypted password in the database

PHP md5() command that I used to encrypt passwords entered is used in the login page and change password page.

```
$password = md5($password);
```

Figure 45 – PHP md5() command

Acknowledgment of Third-Party Code

"PHP Mailer" is gathered from https://github.com/PHPMailer/PHPMailer

"JQuery" is gathered from https://jquery.com/

Word Count: 978