

HowToOrigami

Submitted To: Dr. Raneem Qaddoura
Eng. Shatha Alhawawsheh
Date of submission: 11/6/2020
Submitted by: Zayna Amierh 18110041

1 TABLE OF CONTENTS

2	Introduction	2
3	Submitting Data Between Pages	3
3.1	GET.....	3
3.2	POST.....	6
4	Web Element Dynamic and Interactive.....	8
4.1	Show Date JavaScript	9
4.2	Read More / Read Less.....	9
4.3	HTML Content Change	12
5	Changing Web Content Without Reloading The Entire Page	13
5.1	Signup Page (Registration)	14
5.2	Contact Form	16
6	Final Result.....	18
7	References	19

2 INTRODUCTION

As a continuation to Part (1) of developing the HowToOrigami Website. We have now reached a point where all design has been completed. In addition to this, we will now move forward to adding server-side programming and making sure the website is fully functional and complete.

At first, the website is connected to a database where all data will be inserted into and selected from. The way the database is connected is through PHPAdmin WampServer and this is also done using the PHP language. Below is the code for connection:

```
<?php
$conn = mysqli_connect("localhost", "root", "", "howtoorigami");
```

To select from the database; the following code is used:

```
<?php
$conn = mysqli_connect("localhost", "root", "", "howtoorigami");
if ($conn) {
    $sql = "select * from city";
    $result = mysqli_query($conn, $sql);
    if (mysqli_num_rows($result) > 0) {
        while ($row = mysqli_fetch_assoc($result)) {
            $city = $row["city"];
            $id = $row["id"];
            $all = "<option value='$city'>$city</option>";
            echo $all;
        }
    } else {
        echo "error";
    }
    mysqli_close($conn);
}
?>
```

For insertion into the database, the following code is an example:

```
<?php
$conn = mysqli_connect("localhost", "root", "", "howtoorigami");
if ($conn) {
    if (isset($_POST['submit'])) {
        $username = $_POST['username'];
        $password = $_POST['password'];
        $email = $_POST['email'];
        $sql = "INSERT INTO user (username, password, email) VALUES ('$username', '$password', '$email')";
        if (mysqli_query($conn, $sql)) {
            $success = "<script type='text/javascript'> window.onload=function(){
            alert('Registration Success')}}";
        </script>";
            echo $success;
        } else {
        }

        mysqli_close($conn);
    }
}
?>
```

3 SUBMITTING DATA BETWEEN PAGES

There are many different methods where data is submitted between pages and it is essential to understand their importance and applicability in the website. These methods specify how to send the data (in the form for example) in which the data is sent to the page specified in the action attribute. The two methods include:

1. GET
2. POST

3.1 GET

The GET method allow the data to be sent as URL variables. The way this done is by appending the data into the URL. The GET can be used for many reasons whether it is for copying the URL to send the link to other people or to direct to other pages according to the value. It is also useful for form submissions where a user wants to bookmark the result. Yet, we should never use the GET for sending any sensitive information. (HTML form method Attribute, 2020)

Example of using GET in my website is the following:

```

<!--Side Navigation Start-->
<aside class="sidenav">
    <ul>
        <li><a href="items.php#Animals">Animals</a>
            <ul>
                <?php
                $conn = mysqli_connect("localhost", "root", "", "howtoorigami");
                if ($conn) {
                    $select = "select * from item where categoryID=1";
                    $result = mysqli_query($conn, $select);
                    if (mysqli_num_rows($result) > 0) {
                        while ($row = mysqli_fetch_assoc($result)) {
                            $id = $row["id"];
                            $alt = $row["alt"];
                            $description = $row["description"];
                            $categoryID = $row["categoryID"];
                            $li = "<li><a href='itemDetails.php?id=$id'>$alt Origami</a></li>";
                            echo $li;
                        }
                    } else {
                        echo "problem";
                    }
                }
                ?>
            </ul>
        </li>
    </ul>

```

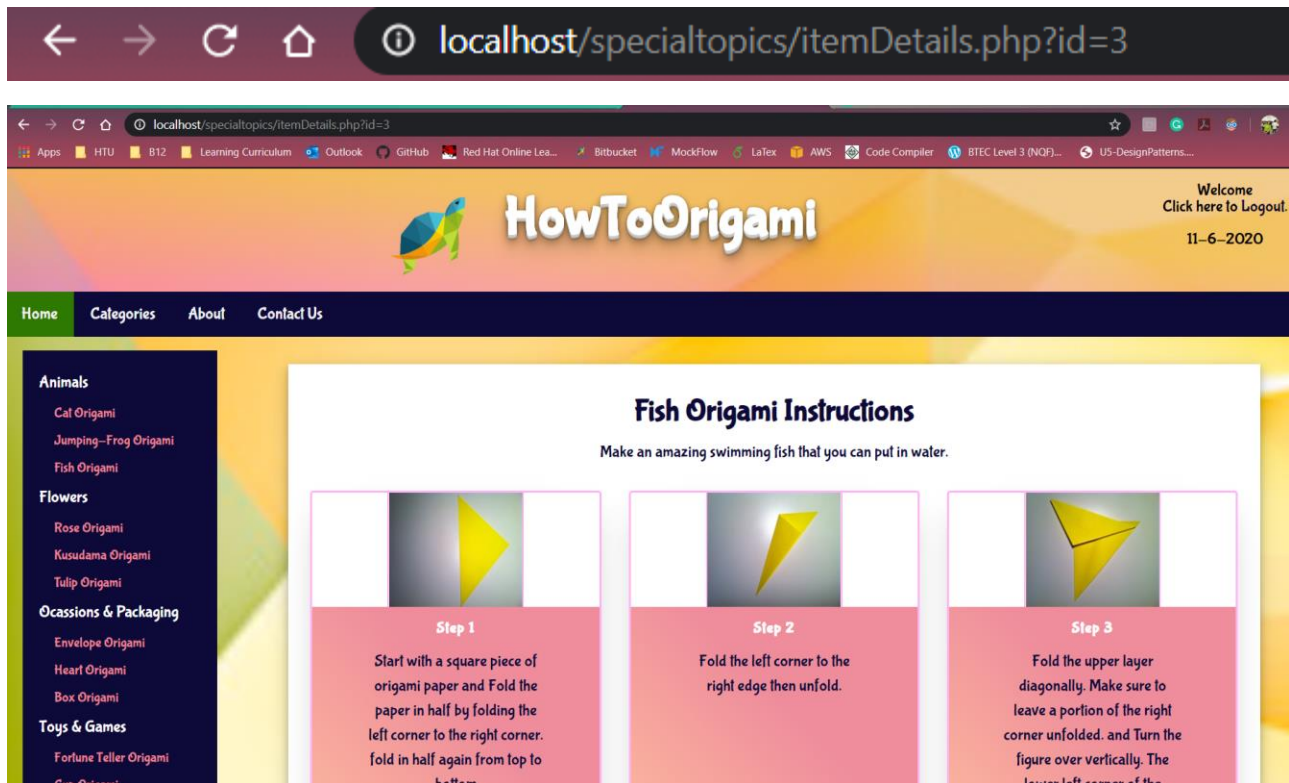
```

<section class="items">
    <?php
    if ($conn) {
        if (isset($_GET['id'])) {
            $id = mysqli_real_escape_string($conn, $_GET['id']);
            $select = "select * from item where id='$id'";
            $result = mysqli_query($conn, $select);
            if (mysqli_num_rows($result) > 0) {
                while ($row = mysqli_fetch_assoc($result)) {
                    $alt = $row["alt"];
                    $description = $row["description"];
                    $all = "<h1>$alt Origami Instructions</h1>";
                    <p>$description</p>";
                    echo $all;
                }
            } else {
                echo "IDK problem";
            }
        }
    } else {
        echo "PROBLEM";
    }
    ?>

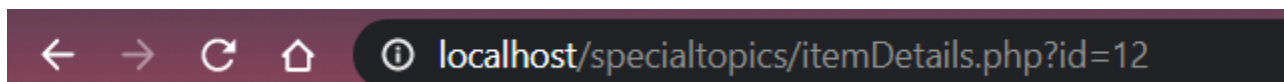
```

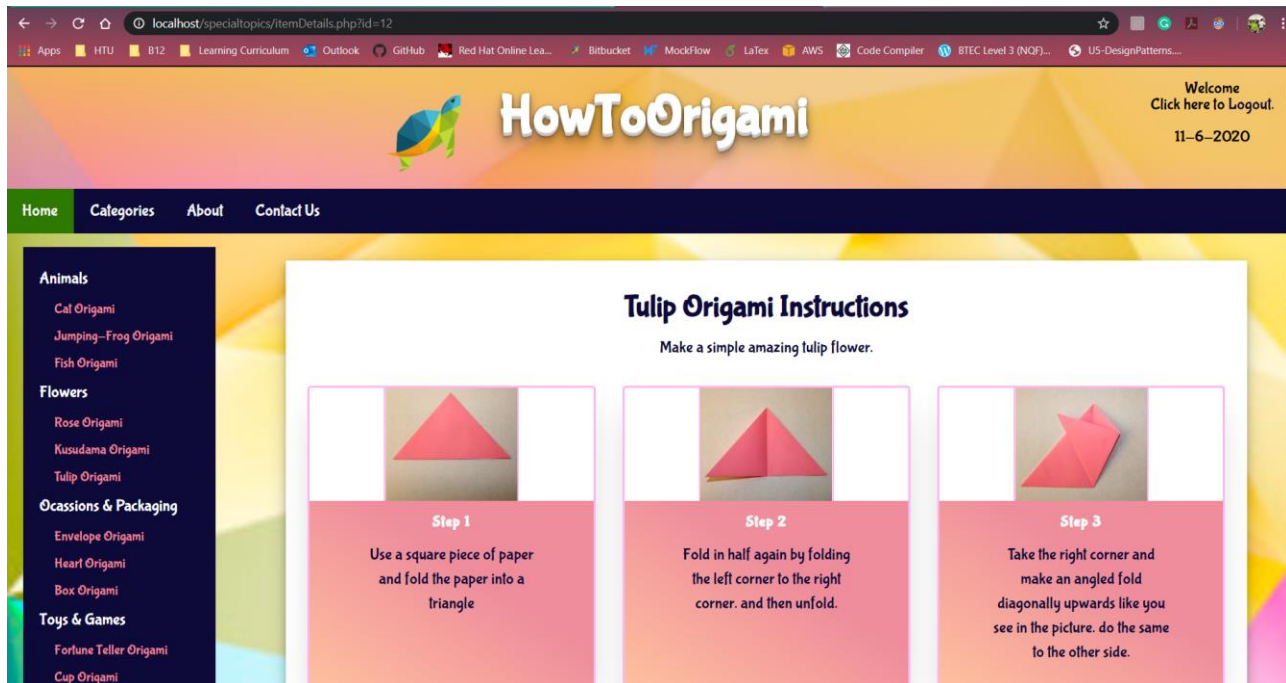
In this source code, when the user goes through the categories and wants to see the details of any item, the URL will change according to the item selected. This is all due to the GET method where the id of the item is taken and put into the URL to allow the displaying of the information depending.

The URL due to GET when selecting items is as follows:



Selecting a different item will give the following url and page





3.2 POST

The POST Method is used as HTTP post transaction where all the data is appended inside the body of the HTTP in which no data will be shown in the URL. Yet, none of the data submitted can be saved or bookmarked for example. (HTML form method Attribute, 2020)

An example of using the POST method in the website I developed is as follows:

Registration Form (Sign in):



In this source code, I have used the method used is post this is because all information is sensitive and should not be seen or saved by anyone. All information that is submitted from the registration form is sent to the database.

This registration form will also use the POST method in session where when the user logs in the website, it will allow the user to use the website. Below is an example of the source code:

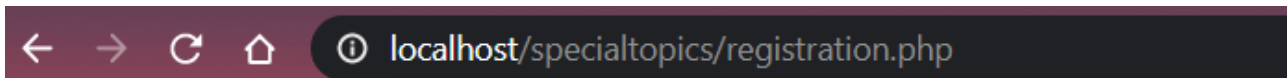
```
<?php

session_start();
if(count($_POST)>0) {
    $con = mysqli_connect('localhost','root','','howtoorigami') or die('Unable To connect');
    $result = mysqli_query($con,"SELECT * FROM user WHERE username='".$_ $_POST["user"] . "' and password = '".$_ $_POST["pass"]."'.");
    $row = mysqli_fetch_array($result);
    if(is_array($row)) {
        $_SESSION["id"] = $row['id'];
        $_SESSION["username"] = $row['username'];
    } else {

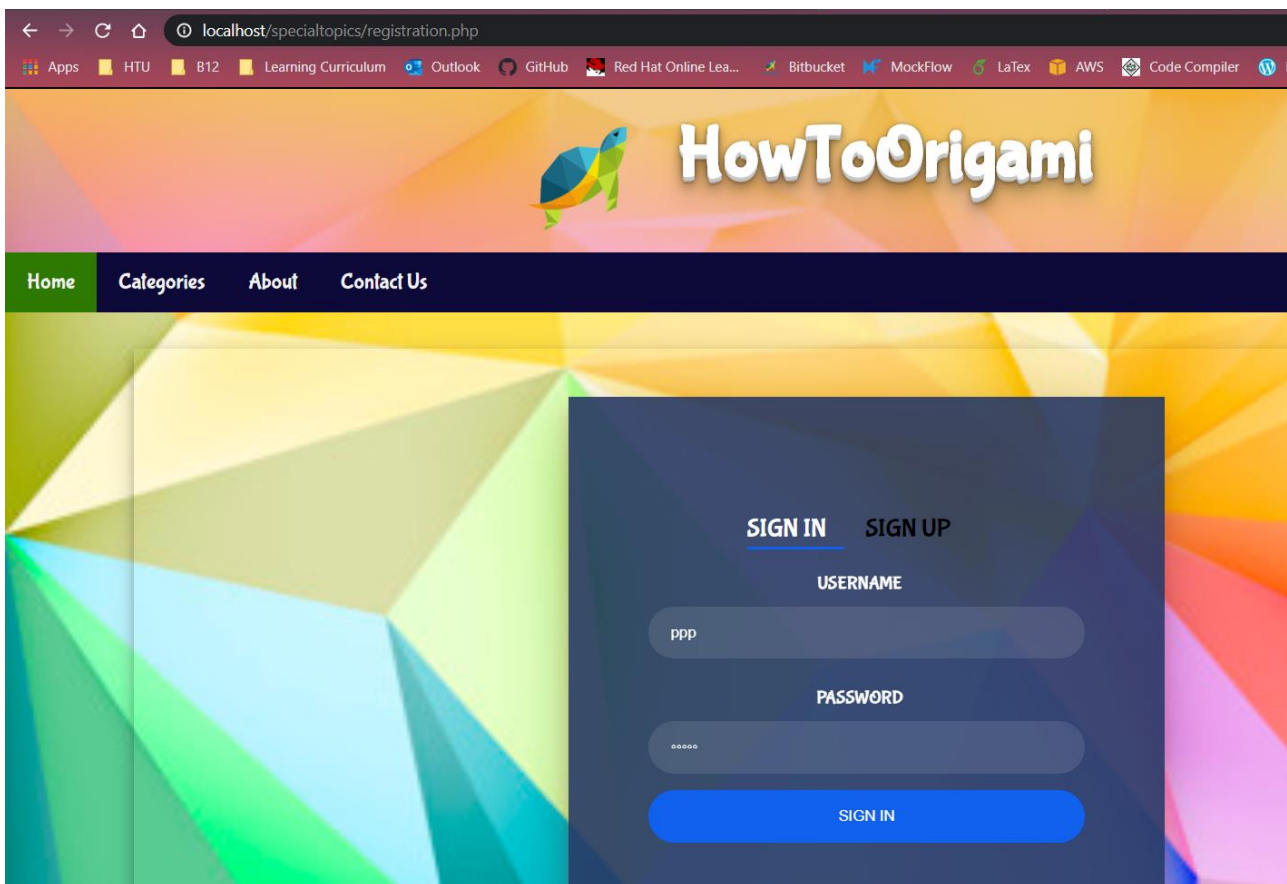
    }
}
if(isset($_SESSION["id"])) {
    header("Location:index.php");
}
?>
```

The url of the page will not change even if username and password entered is incorrect.

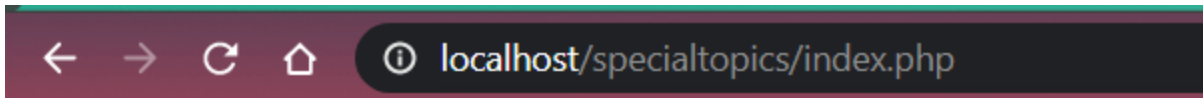
Before:



After with not successful entry:



After with successful entry:



The page is redirected to the homepage due to successful login. (as said, this is done through sessions).



4 WEB ELEMENT DYNAMIC AND INTERACTIVE

Making web elements dynamic and interactive is very essential for modern websites since having elements that are capable of changing due to actions can help us develop parts of websites that can change to our needs and specifications and help also increase user experience. The way we can make web elements dynamic and interactive in order to create more engagement of the website along with better information access and quick retrievals. This is all done by implementing JavaScript. JavaScript is used to program the behavior of the web pages whether it is to change the HTML content or styles or even elements, we can also hide elements. This is all done by adding JavaScript to anything in the website that you want to change. All JavaScript must be inserted within the script tags (script tag defines the client-side code) where we can add actions, events, change the inner HTML or add dynamically changing information (time date for example). We can also use the JavaScript to allow more interactiveness (ex. Adding alert when contacting successfully). As a whole, JavaScript is added to the website that contains HTML and CSS to bring the website alive.

(Duraj and Interactivity, 2020) (JavaScript Tutorial, 2020)

There are a few examples that I have used in the website. Below are a few examples:

4.1 Show Date JavaScript

```
document.addEventListener("DOMContentLoaded", function() {  
    var d = new Date();  
    var curr_date = d.getDate();  
    var curr_month = d.getMonth() + 1; //months are zero based  
    var curr_year = d.getFullYear();  
    document.querySelector('.date').innerHTML = curr_date + "-" + curr_month + "-" + curr_year;  
});  
</script>
```

```
<header>  
<div class="top">  
    <div id="sign">  
        <a id="sig" href="registration.php">Sign-in/Sign-up</a><br>  
        <div class="date"></div>
```

In this JavaScript code, it is used to display the date on the top screen of the website and change according to the day present. This is mainly done by getting the date and month and year and display it accordingly. The way this is done is by checking if the page is fully loaded and get the date, month and year from a predefined function and then insert into the html where the class is date and add the date accordingly. Below is the output:



4.2 Read More / Read Less

Another JavaScript example is the read more and read less that will display the information when clicking the button. Below is the code:

```

<!--History Section Start-->
<section class="history">
  <article>
    <h1>Origami History</h1>
    <h3>The life of origami...</h3>
    <p>For the initial 1500 years or so of its existence, origami was actually known as orikata, a
      Japanese word that translates to "Folded Paper". It was in 1880 that the name of the art
      form was changed to origami, derived from two Japanese words; oru, meaning to fold, and
      kami, meaning paper. The reason for the change in this term has been attributed to the
      easiness of writing the characters for this word, which made it more viable to teach to
      school children.<span id="dots">...</span></p>
    <p><span id="more">
      Paper was originally created sometime around 100 A.D. While paper is considered a common and cheap
      commodity now, at the time of its conception, the resources required to create paper were not as
      easily available. Paper was an expensive and rare commodity, so those engaging in the art of
      traditional paper folding was usually an art left for the elite rungs of society. Conversely, it was
      also used by Buddhist Monks for religious purposes. It only later became a practice open to the
      public</span></p>
  </article>
  <button onclick="buttonRead()" id="myBtn" class="btn">Read more</button>
</section>

```

```

<script>
  function buttonRead() {
    var dots = document.getElementById("dots");
    var moreText = document.getElementById("more");
    var btnText = document.getElementById("myBtn");

    if (dots.style.display === "none") {
      dots.style.display = "inline";
      btnText.innerHTML = "Read more";
      moreText.style.display = "none";
    } else {
      dots.style.display = "none";
      btnText.innerHTML = "Read less";
      moreText.style.display = "inline";
    }
  }
}

```

The below is the output:

Origami History

The life of origami..

For the initial 1500 years or so of its existence, origami was actually known as orikata, a Japanese word that translates to "Folded Paper". It was in 1880 that the name of the art form was changed to origami, derived from two Japanese words; oru, meaning to fold, and kami, meaning paper. The reason for the change in this term has been attributed to the easiness of writing the characters for this word, which made it more viable to teach to school children....

READ MORE

When clicked:

Origami History

The life of origami..

For the initial 1500 years or so of its existence, origami was actually known as orikata, a Japanese word that translates to "Folded Paper". It was in 1880 that the name of the art form was changed to origami, derived from two Japanese words; oru, meaning to fold, and kami, meaning paper. The reason for the change in this term has been attributed to the easiness of writing the characters for this word, which made it more viable to teach to school children.

Paper was originally created sometime around 100 A.D. While paper is considered a common and cheap commodity now, at the time of its conception, the resources required to create paper were not as easily available. Paper was an expensive and rare commodity, so those engaging in the art of traditional paper folding was usually an art left for the elite rungs of society. Conversely, it was also used by Buddhist Monks for religious purposes. It only later became a practice open to the public

READ LESS

4.3 HTML Content Change

Another example of using JavaScript is changing HTML content according to conditions. In my case, I said if the user logs in and signin/signup link will be changed to Welcome, click here to logout. Below is the source code showing where JavaScript is used. The changes all due to sessions when user logs in and out. When logging in, the session is set and when logging out the session is unset or destroyed.

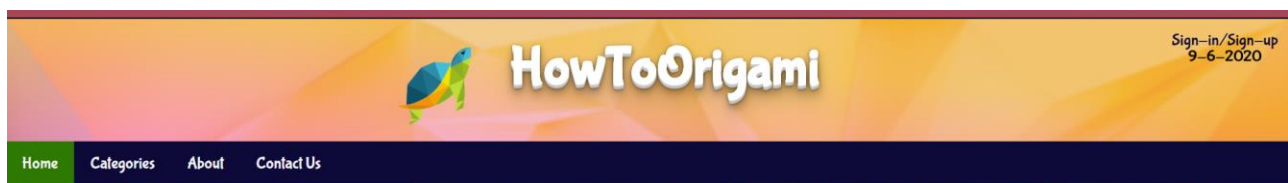
Note: JavaScript is used inside the PHP

```
<?php
session_start();
if (!isset($_SESSION['username'])) {
    header('Location:registration.php');
} else {
    $all = '<script>
    document.addEventListener("DOMContentLoaded", function() {
        var el = document.querySelector("#sig");
        var newEl = document.createElement("div");
        newEl.innerHTML ="Welcome<br>" + "Click here to <a href=logout.php>Logout.";
        el.parentNode.replaceChild(newEl, el);
    });
</script>';
    echo $all;
    $str = $_SESSION["username"];
    // echo "Welcome $str.<br> Click here to <a href='logout.php' title='Logout'>Logout.";
}
?>
```

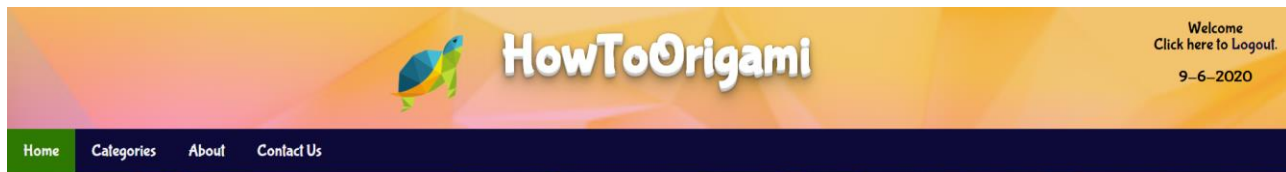
It will alter the following code where the id is “sig”:

```
headerNav.php > header > nav#myTopnav.topnav
1 <header>
2 <div class="top">
3     <div id="sign">
4         <a id="sig" href="registration.php">Sign-in/Sign-up</a><br>
5         <div class="date"></div>
6     </div>
7 </div>
```

The output when signed out is:



This is the output when signed in:



When logging out, the page is redirected to logout.php that will remove the sessions and display the content back to sign-in/signup

```
logout.php
1  <?php
2  session_start();
3  unset($_SESSION["id"]);
4  unset($_SESSION["username"]);
5  header("Location:registration.php");
6  ?>
7
```

5 CHANGING WEB CONTENT WITHOUT RELOADING THE ENTIRE PAGE

Changing web content without reloading the whole page is extremely important to understand and do in every website and is very needed. This is done by using AJAX which is used to update the web page you are in without any reloading and this technique is for creating dynamic web pages. The way this is done is by requesting and receiving data from the server after the page is loaded and this all will happen in the background when the data is sent to the server. As a result, AJAX is needed since it makes the website faster and more responsive to user actions. (AJAX Introduction, 2020). So, changing web content without reloading the entire page is needed to optimize the user experience and allow the page to be faster and more responsive so there is no delay and the user doesn't find any difficulties or issues during the use of the website. As a whole, Ajax will read data from the web server after the page is loaded and will update the webpage without reloading the page in addition it will send the data to the web server in the background.

I have added a few examples of AJAX to my website. Let's discuss them.

5.1 Signup Page (Registration)

```

<div class="sign-up-htm">
  <div class="group">
    <label for="username" class="label">Username</label>
    <input id="username" name="username" type="text" class="input">
  </div>
  <div class="group">
    <label for="password" class="label">Password</label>
    <input id="password" type="password" name="password" class="input" data-type="password">
  </div>
  <div class="group">
    <label for="email" class="label">Email Address</label>
    <input id="email" name="email" type="text" class="input">
  </div>
  <div class="group">
    <button class="button" id="si" value="Sign Up">Signup</button>
  </div>
  <div class="foot-lnk">
    <label for="tab-1">Already Member?</a>
  </div>
</div>

```

This is the code for the sign up which is used for the user to register an account on the website and all data is inserted in the database.

```

document.addEventListener('DOMContentLoaded', function() {

  document.querySelector('#si').onclick = function() {
    var request = new XMLHttpRequest();
    var username = document.querySelector('#username').value;
    var password = document.querySelector('#password').value;
    var email = document.querySelector('#email').value;
    var string = "?username=" + username + "&password=" + password + "&email=" + email;
    request.open("POST", "signup.php" + string);
    request.onreadystatechange = function() {
      if (this.readyState === 4 && this.status === 200) {
        alert("Signin Success, Please Sign In")
      }
    };
    request.send();
  }
});

```

This is the ajax code in JavaScript which is displayed in the head of the php file. The way this works is if the page is loaded, it will select the id “si” and create a request and will get the value of the username, password, and email that has been entered. The request will open in a POST method

to the signup.php file that will do specific things for insertion into the database, if it is successful, then an alert will be created and the request will be sent.

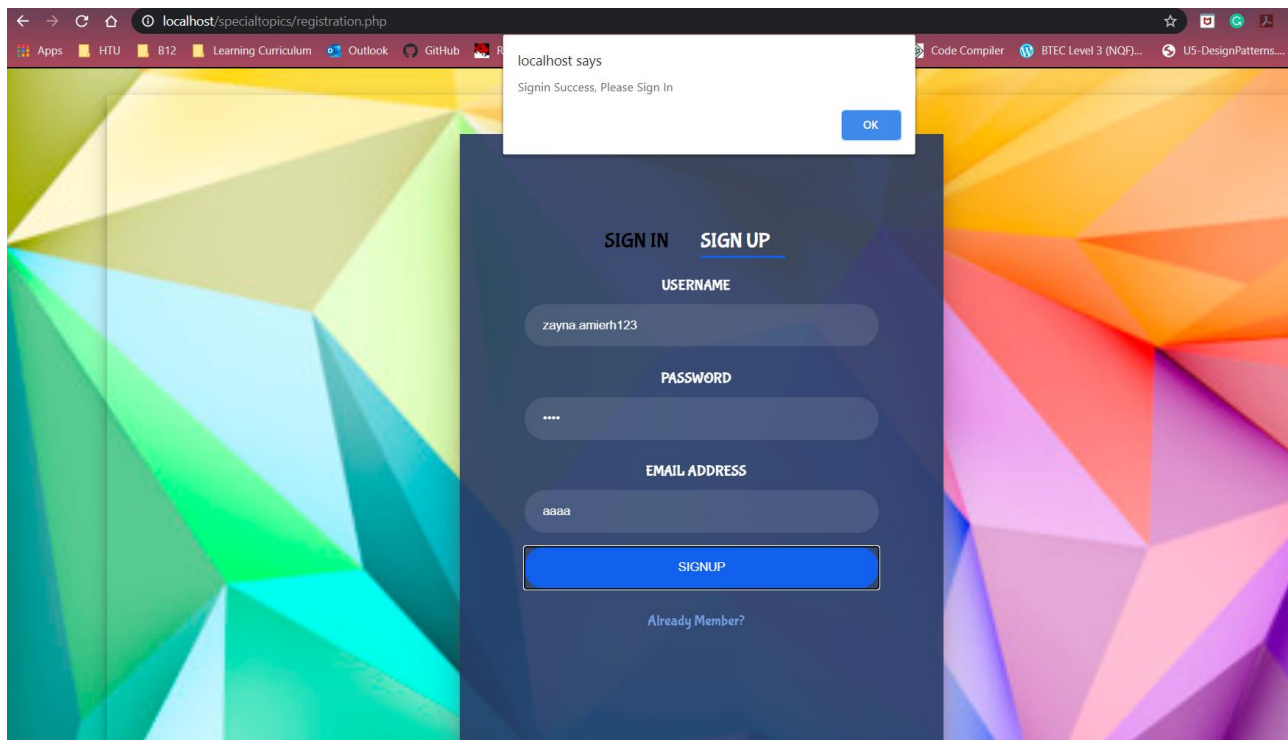
```

signup.php > ...
1  <?php
2  $conn = mysqli_connect("localhost", "root", "", "howtoorigami");
3  if ($conn) {
4  $username = $_REQUEST['username'];
5  $password = $_REQUEST['password'];
6  $email = $_REQUEST['email'];
7  $sql = "INSERT INTO user (username, password, email) VALUES ('$username', '$password', '$email')";
8  if (mysqli_query($conn, $sql)) {
9
10 } else {
11 echo "error";
12 }
13
14
15 mysqli_close($conn);
16 }
17 ?>



```

This is the PHP file that will connect to the database and take the username, password, and email and insert them into the database and close the database at the end.

The output of successful signup is as follows:



Database update:

<input type="checkbox"/>				72	zayna.amierh123	aaaa	aaaa
--------------------------	---	---	---	----	-----------------	------	------

5.2 Contact Form

Another AJAX example is the contact form:

```

113 <section class="transparent" id="contacting">
114 <h2>Contact Form</h2>
115 <p>Send Us a message:</p>
116 <div class="contactmain">
117 <div class="contact-form">
118 <div class="row">
119 <div class="col-25">
120 <label for="full_name">Full Name</label>
121 </div>
122 <div class="col-75">
123 <input type="text" id="full_name" name="full_name" required placeholder="Your full name..">
124 </div>
125 </div>
126 <div class="row">
127 <div class="col-25">
128 <label for="email">Email</label>
129 </div>
130 <div class="col-75">
131 <input type="text" id="email" name="email" required placeholder="Your email..">
132 </div>
133 </div>
134 <div class="row">
135 <div class="col-25">
136 <label for="city">City</label>
137 </div>
138 <div class="col-75">
139 <select id="city" name="city">
140 <?php
141 $conn = mysqli_connect("localhost", "root", "", "howtoorigami");
142 if ($conn) {
143 $sql = "select * from city";
144 $result = mysqli_query($conn, $sql);
145
146 if (mysqli_num_rows($result) > 0) {
147 while ($row = mysqli_fetch_assoc($result)) {
148 $city = $row["city"];
149 $id = $row["id"];
150 $all = "<option value='$city'>$city</option>";
151 echo $all;
152 }
153 } else {
154 echo "error";
155 }
156 mysqli_close($conn);
157 }
158 <?>
159 </select>
160 </div>
161 </div>
162 <div class="row">
163 <div class="col-25">
164 <label for="subject">Subject</label>
165 </div>
166 <div class="col-75">
167 <input type="text" id="subject" required name="subject" placeholder="Subject..">
168 </div>
169 </div>
170 <div class="row">
171 <div class="col-25">
172 <label for="message">Message</label>
173 </div>
174 <div class="col-75">
175 <textarea id="message" required name="message" placeholder="Write something.." style="height:200px"></textarea>
176 </div>
177 </div>

```

```

178         <div class="row">
179             <button id="sub">Submit</button>
180         </div>
181         <div id="c"></div>
182     </div>
183 </div>
184 </section>

```

This is the contact form where the user can contact easily. The user will insert their fullname, email, city, subject, and message and all this information will be inserted into the database accordingly.

Below is the AJAX code:

```

document.addEventListener('DOMContentLoaded', function() {

    document.querySelector('#sub').onclick = function() {
        var request = new XMLHttpRequest();

        var full_name = document.querySelector('#full_name').value;
        var email = document.querySelector('#email').value;
        var city = document.querySelector('#city').value;
        var subject = document.querySelector('#subject').value;
        var message = document.querySelector('#message').value;
        var string = "?full_name=" + full_name + "&email=" + email + "&city=" + city + "&subject=" + subject + "&message=" + message;
        request.open("GET", "contacting.php" + string);
        request.onreadystatechange = function() {
            if (this.readyState === 4 && this.status === 200) {
                alert("You've Contacted Us Successfully! Thank you");
                document.querySelector('#c').innerHTML = this.responseText;
            }
        };
        request.send();
    }
});

```

In the AJAX code that is in JavaScript, when the page is loaded and the submit button is clicked, it will create a request. The values of the inputs fullname, email, city, subject, and message are taken and the request opens the contacting.php file with the GET method that will do specific implementation for insertion. If successful, an alert along with a response text will be given.

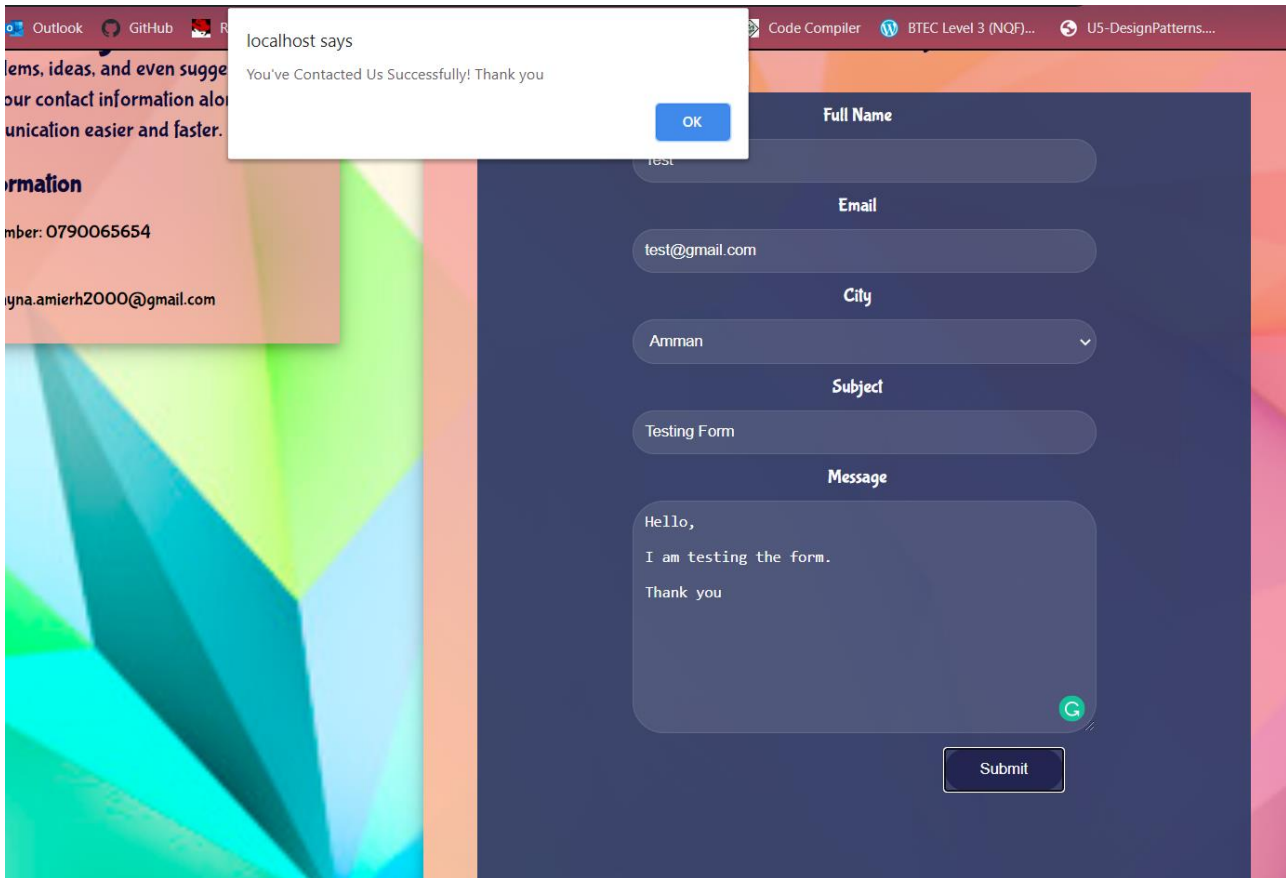
```

contacting.php > ...
1  <?php
2  $conn = mysqli_connect("localhost", "root", "", "howtoorigami");
3  if ($conn) {
4
5      $full_name = $_GET['full_name'];
6      $email = $_GET['email'];
7      $city = $_GET['city'];
8      $subject = $_GET['subject'];
9      $message = $_GET['message'];
10     $sql = "INSERT INTO contact (full_name, email, city, subject, message) VALUES ('$full_name', '$email', '$city', '$subject', '$message')";
11     if (mysqli_query($conn, $sql)) {
12         echo "Thank you!";
13     } else {
14         echo "error";
15     }
16 }
17 mysqli_close($conn);
18 }

```

This is the php file that will connect to the database and insert the inputs from the GET into the database and if successful the message will be output. The database is then closed.

The output of successfully contacting when submitted:



Database update:

<input type="checkbox"/>		Edit		Copy		Delete	44	Test	test@gmail.com	Amman	Testing Form	Hello,I am testing the form.Thank you
--------------------------	--	------	--	------	--	--------	----	------	----------------	-------	--------------	---------------------------------------

6 FINAL RESULT

As a result to developing the website from the frontend of the website to the backend of the website. The website is now said to be completely functional in all terms. The website contains different methods for submitting data between pages, interactive and dynamic web elements using JavaScript in addition to the ability of being able to change web content without reloading the entire page.

7 REFERENCES

- W3schools.com. 2020. HTML Form Method Attribute. [online] Available at: <https://www.w3schools.com/tags/att_form_method.asp>.
- Duraj, M. and Interactivity, H., 2020. How Javascript Implementation Adds Dynamic Interactivity | Pluralsight. [online] Pluralsight.com. Available at: <<https://www.pluralsight.com/guides/how-javascript-implementation-adds-dynamic-interactivity>>.
- W3schools.com. 2020. Javascript Tutorial. [online] Available at: <<https://www.w3schools.com/js/>>.
- W3schools.com. 2020. AJAX Introduction. [online] Available at: <https://www.w3schools.com/xml/ajax_intro.asp>.