ajax

```php
<?php
// Need of the ajax is getting required data .
// There will be one page to send data and another which request that
data.

//
*********************************************************
*
// SEND DATA

<?php
header("Content-Type: application/json");

$data = ["name" => "ahmo", "age" => 24];
// $data = getGames();

echo json_encode($data);
// -------------------------------------------------

//
*********************************************************
*
// GET DATA

$("button").click(function() {
    // let url = "sendData.php";
    let url = "http://localhost/AhmetOguzErgin/Web/mySolutions/11/1/
dataSend.php";

    // $.get(url, {name: "ahmet", age: 24} function(data) {
    $.get(url, function(data) {
        console.log(data);

        console.log(data[0].name);
        $("#result").html(data[0].name);
    }).then(function() {
```

```javascript
    }).then(function() {

    })
});

// or
// ----------------------------------------------------

$("button").click(function() {
    // let url = "sendData.php";
    let url = "http://localhost/AhmetOguzErgin/Web/mySolutions/11/1/
dataSend.php";

    $.ajax({
        type: "GET",
        url: url,
        async: false,
        data: {
            name: "ahmet",
            age: 24
        },
        success: function(data) {
            console.log("Data recieved.");
            console.log("Data : " + data);
            console.log(JSON.stringify(data));
        },

        cache: false,
        beforeSend: function() {
            console.log("loading...");
        },
        error: function(xhr, status, error) {
            console.log("Error in ajax request : " + error);
        }
    });
});
```

Cookie 1

```php
<?php

// Main  purpose of using cookie is storing client information for
different purposes
// First thing is authentication. ..Advertisement, development according
to user preferences
// cookies are stored in client browser
// expire date can be determined to any time
// client may change his cookie information
// lets say we have table like username password and cookie token

// Implementation
// when user get into LOGIN PAGE: CHECK
//
*********************************************************************
*****
// 1. check if has cookie and this cookie is valid for application
(compare with database) if it is true direct to main page. bypass login
page.

if (hasCookieAlready() && isCookieValid()) {
    header("Location:./main.php");
    exit;
}

function hasCookieAlready()
{
    if (!empty($_COOKIE))
        return true;
    return false;
}
```

```php
function isCookieValid()
{
    $token = $_COOKIE["token"];

    $user = getUserFromDb($token);

    if ($user == false)
        return false;
    return true;
}

function getUserFromDb($token)
{
    global $db;

    try {
        $sql = "select * from cookiesession where token = :token";
        $stmt = $db->prepare($sql);
        $stmt->bindValue(":token", $token, PDO::PARAM_STR);
        $stmt->execute();
        $row = $stmt->fetch(PDO::FETCH_ASSOC);
    } catch (Exception $ex) {
        die("Query Error : " . $ex->getMessage());
    }

    return $row;
}
//
*********************************************************************
*****
// 2. if there is no cookie and user logged on system with correct
password and username than  CREATE
// create unique token with sha1
// update database token
// send cookie to client
// and direct to authenticated page
```

Cookie2

```php
59    // and direct to authenticated page
60
61    if ($_SERVER["REQUEST_METHOD"] == "POST") {
62        // check username and password from database
63        if (checkUserPassword($username, $password)) {
64            // you are authenticated
65
66            // create unique cookie token for user
67            $token = sha1(uniqid() . "_TOKEN_" . time());
68
69            // update database token column
70            updateDatabaseToken($username, $token);
71
72            // give that token to user as cookie
73            setcookie("token", $token, time() + 60 * 60 * 1, "/", "", true,
              true);
74
75            // direct to page as authenticated with cookie
76            header("Location:./main.php");
77            exit;
78        }
79    }
80    //
      **************************************************************************
      *****
81
82    // when user get into Main Page:
83    // 1. check if user get this page by authentication or not
84    if (!hasCookieAlready() || !isCookieValid()) {
85        // if there is no cookie or cookie is not valid for us exit with
          login page
86        header("Location:./login.php");
87        exit;
88    } else {
89        // display info
90        $user =  getUserFromDb($token);
```

```php
88    } else {
89        // display info
90        $user =  getUserFromDb($token);
91        echo "<p class='center'> Hello {$user["username"]}. </br> We miss
          you! </p>";
92        echo "<p class='center'></br>Your cookie token help us to
          authenticate you ! <br> $token</p>";
93    }
94    //
      **************************************************************************
      *****
95
96    // LOGOUT
97    // basically it delete cookie by aranging time to a past
98    // delete cookie by time     as delete also you may update like that
99    if (isset($logOut)) {
100       setcookie("token", "asd", 1, "/");
101       header("Location:./login.php");
102       exit;
103   }
104
105
106
107
108   // Tricks footnotes
109   // when setcookie() is executed $_COOKIE array not refreshed so you need
      to refresh page.
110   // you cannot change cookie with array you need to use setcookie()
      expression
111   // in inspect element -> application you may see change cookies
112   // setcookie("name" : "ahmet", time()+60*60*2, "/", "ahmo.net", true,
      true)
113   //       cookiename, value  , 2 hour,  allowed folder, domain, https,
      not sending cookie with javascript
114
```

Session 1

```php
<?php
// In session, file is created by server
// File id will be given to client with cookie
// Cookie of the session automatically deleted when browser is closed or
logout
session_start(); //do not forget to put that for every page


// ----------------------------------------------------------
// 1. LOGIN CREATE
//check if there is session and direct
if (hasValidSession()) {
    header("Location:./main.php");
    exit;
}

function hasValidSession()
{
    if (isset($_SESSION["user"])) {
        return true;
    }
    return false;
}

// if there is no, form session file
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (checkUserPassword($username, $password) == true) {
        // form session file
        $_SESSION["user"] = getUserFromDb($username);

        //remember me, //set session cookie with time
        // bu aşağıdaki sayfa ilk açıldığı zaman çalışmıyor çünkü cookie
        sayfa kapanınca save ediliyor.
        // setcookie(session_name(), $_COOKIE["PHPSESSID"], time() + 60
        * 60 * 24 * 7, "/");
        // setcookie(session_name(), session_id(), time() + 60 * 60 * 24
        * 7, "/");

        // direct to main page
        header("Location:./main.php");
        exit;
    }
}

function checkUserPassword($username, $password)
{
    global $db;

    try {
        $sql = "select * from cookiesession where username = :username
        and password = :password";
        $stmt = $db->prepare($sql);
        $stmt->bindValue(":username", $username, PDO::PARAM_STR);
        $stmt->bindValue(":password", $password, PDO::PARAM_STR);
        $stmt->execute();
        $row = $stmt->fetch(PDO::FETCH_ASSOC);
    } catch (Exception $ex) {
        die("Query Error : " . $ex->getMessage());
    }

    if ($row == false) {
        echo "<h1 class='error'>Wrong Password</h1>";
        return false;
    }
    return true;
}
```

Session 2

```php
function getUserFromDb($username)
{
    global $db;

    try {
        $sql = "select * from cookiesession where username = :username";
        $stmt = $db->prepare($sql);
        $stmt->bindValue(":username", $username, PDO::PARAM_STR);
        $stmt->execute();
        $row = $stmt->fetch(PDO::FETCH_ASSOC);
    } catch (Exception $ex) {
        die("Query Error : " . $ex->getMessage());
    }

    return $row;
}

// ----------------------------------------------------------
// 2. Main Page CHECK READ
// check session to prevent unauthenticated users
if (!hasValidSession()) {
    header("Location:./login.php");
    exit;
    var_dump(hasValidSession());
}

$userData = $_SESSION["user"];

// ----------------------------------------------------------
// 3. LOG OUT DELETE  UPDATE
if (isset($logOut)) {
    //delete anything specific
    // unset($_SESSION["login"]);

    // update anything specific
    // $_SESSION["login"]["name"] = "ahmet" ;

    //clear data
    $_SESSION = [];

    // delete cookie
    setcookie(session_name(), "", 1, "/"); // delete memory cookie

    // delete session file from tmp
    session_destroy();

    header("Location: login.php");
}
```