**Medientechnik Cheatsheet improved**

**Client Server Communication (get JSON data)**

**HTML**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <script src="./script.js" defer></script>

</head>

<body>

    <input type="text" name="text" id="nameBox">

    <div id="send">send</div>

    <div id="json-data">

    </div>

</body>

</html>

**JS**

let sendBtn = document.getElementById('get');

let jsonData = document.getElementById('json-data');

sendBtn.addEventListener('click', getData);

function getData() {

    fetch(`./server.php`)

        .then((res) => {

            return res.json();

        })

        .then((data)=>{

            jsonData.innerHTML =`<pre> ${data} </pre>`;

        })

        .catch((err)=>{

            console.log(err);

        })

}

**PHP**

<?php

$data = "../data/personen.json";

if(!file\_exists($data)){

exit;

}

$data = file\_get\_contents($data);

echo $data;

**Client Server Communication (send data)**

**HTML**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <script src="./script.js" defer></script>

</head>

<body>

    <input type="text" id="name">

    <div id="get">get</div>

    <div id="returned-data"></div>

</body>

</html>

**JS**

let sendBtn = document.getElementById('get');

let returnedData = document.getElementById('returned-data');

let namBox = document.getElementById('name');

sendBtn.addEventListener('click', getData);

function getData() {

    let name = namBox.value

    fetch(`./server.php?name=${name}`)

        .then((res) => {

            return res.text();

        })

        .then((data)=>{

            returnedData.innerHTML =`<pre> ${data} </pre>`;

        })

        .catch((err)=>{

            console.log(err);

        })

}

**PHP**

<?php

if(isset($\_GET['name'])){

    echo "name";

}

**Sessions**

session\_start();

if (!isset($\_SESSION["login"])) {

    $\_SESSION["login"] = false;

}

echo $\_SESSION[login];

**Read CSV data (solution one)**

$persons\_csv = './data/personen.csv';

if (!file\_exists($persons\_csv)) {

    echo "CVS file does not exit!";

    exit;

}

$csv\_data = file\_get\_contents($persons\_csv);

$csv\_data\_array = explode("\n", $csv\_data);

$counter\_csv = 0;

foreach ($csv\_data\_array as $iterator) {

    if ($counter\_csv == 0) {

        continue;

    }

    $attributes = explode(";", $iterator);

    $name = $attributes[0];

    $lastname = $attributes[1];

    $username = $attributes[2];

    $password = $attributes[3];

    $image\_path = $attributes[4];

  $sql = "insert into v\_persons (name,lastname,username,password,image\_path)

        values('$name', '$lastname', '$username', '$password', '$image\_path')";

        if ($res = $conn->query($sql)) {

        } else {

            echo $res;

        }

}

echo "CSV-data successfully inserted! <br>";

**Solution two (CSV-Reading)**

function readCsvFileIntoObjects($filename): array

{

$lines = array\_map('str\_getcsv', file("data/$filename.csv"));

$header = array\_shift($lines);

$result = array();

foreach ($lines as $row) {

$row = array\_map('trim', $row);

try {

$result[] = array\_combine($header, $row);

} catch (Error $ignored) {}

}

return $result;

}

**Database-Connection**

$db = "verwaltung";

$user = "app";

$password = "app";

$hostname = "localhost";

$conn = new mysqli($hostname, $user, $password, $db);

//Check connection

if ($conn->connect\_error) {

    die("Connection failed: " . $conn->connect\_error);

}

$conn->close();

**Image upload**

**HTML**

<form action="./index.php" method="post" enctype="multipart/form-data">

        <br>

        <input type="text" name="name" placeholder="name">

        <br>

        <input type="text" name="lastname" placeholder="lastname">

        <br>

        <input type="text" name="username" placeholder="username">

        <br>

        <input type="password" name="password" placeholder="password">

        <br>

        <input type="password" name="re-password" placeholder="reenter password">

        <br>

        <input type="file" name="myfile">

        <br>

        <input type="submit" value="sign in" name="sign-in">

    </form>

**PHP**

if (isset($\_POST['sign-in']) && isset($\_FILES['myfile']) && !$\_SESSION["login"]) {

    $name = $\_POST['name'];

    $lastname = $\_POST['lastname'];

    $username = $\_POST['username'];

    $password = $\_POST['password'];

    $re\_password = $\_POST['re-password'];

    $img\_dir = "./img/";

    $image = $img\_dir . basename($\_FILES["myfile"]["name"]);

    $imageFileType = pathinfo($image, PATHINFO\_EXTENSION);

    $uploadOk = 1;

    $check = getimagesize($\_FILES["myfile"]["tmp\_name"]);

    if ($check !== false) {

        echo "File is an image - " . $check["mime"] . ".";

        $uploadOk = 1;

    } else {

        echo "File is not an image.";

        $uploadOk = 0;

    }

    if (file\_exists($image)) {

        echo "Sorry, file already exists.";

        $uploadOk = 0;

    }

    if ($\_FILES["myfile"]["size"] > 500000) {

        echo "Sorry, your file is too large.";

        $uploadOk = 0;

    }

    if (

        $imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"

        && $imageFileType != "gif"

    ) {

        echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";

        $uploadOk = 0;

    }

// unten gehts weiter…

        if (move\_uploaded\_file($\_FILES["myfile"]["tmp\_name"], $image)) {

            echo "The file " . basename($\_FILES["myfile"]["name"]) . " has been

            uploaded.";

            $insertStatement = "INSERT INTO v\_persons (name,lastname,username,password,image\_path) VALUES ('$name','$lastname','$username','$password','$image');";

            if ($\_res = $conn->query($insertStatement)) {

                echo "<br>Dataset successfully inserted into the database";

            } else {

                echo "<br>NO insertion into database";

            }

        }

        $\_SESSION["login"] = true;

    } else {

        $\_SESSION["login"] = false;

    }

**Thumb**

if (!isset($\_POST["submit"]) || !isset($\_FILES["fileToUpload"])) {

exit;

}

$target\_dir = "uploads/";

$target\_file = $target\_dir . basename($\_FILES["fileToUpload"]["name"]);

$uploadOk = 1;

$imageFileType = pathinfo($target\_file, PATHINFO\_EXTENSION);

if (isset($\_POST["submit"])) {

$check = getimagesize($\_FILES["fileToUpload"]["tmp\_name"]);

if ($check !== false) {

echo "<br>Files is an image - " . $check["mime"] . ".";

$uploadOk = 1;

} else {

echo "<br>File is not an image.";

$uploadOk = 0;

}

}

if (file\_exists($target\_file)) {

echo "<br>Sorry, file already exists.";

$uploadOk = 0;

}

if ($\_FILES["fileToUpload"]["size"] > 5000000) {

echo "<br>Sorry, your file is too large.";

$uploadOk = 0;

}

// Allow certain file formats

if ($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg" && $imageFileType != "gif") {

echo "<br>Sorry, only JPG, JPEG, PNG & GIF files are allowed.";

$uploadOk = 0;

}

// Check if upload ok

if ($uploadOk == 0) {

echo "<br>Sorry, your file was not uploaded.";

} else {

if (move\_uploaded\_file($\_FILES["fileToUpload"]["tmp\_name"], $target\_file)) {

echo "<br>The file " . basename($\_FILES["fileToUpload"]["name"]) . " has been uploaded.";

// Generate thumbnail

$imagefile = $target\_file;

$imagesize = getimagesize($imagefile);

$imagewidth = $imagesize[0];

$imageheight = $imagesize[1];

$imagetype = $imagesize[2];

switch ($imagetype) {

// Bedeutung von $imagetype:

// 1 = GIF, 2 = JPG, 3 = PNG, 4 = SWF, 5 = PSD, 6 = BMP, 7 = TIFF(intel byte order), 8 =

// TIFF(motorola byte order), 9 = JPC, 10 = JP2, 11 = JPX, 12 = JB2, 13 = SWC, 14 = IFF, 15

// = WBMP, 16 = XBM

case 1: // GIF

$image = imagecreatefromgif($imagefile);

break;

case 2: // JPEG

$image = imagecreatefromjpeg($imagefile);

break;

case 3: // PNG

$image = imagecreatefrompng($imagefile);

break;

default:

die('Unsupported imageformat');

}

// Maximalausmaße

$maxthumbwidth = 150;

$maxthumbheight = 100;

// Ausmaße kopieren, wir gehen zuerst davon aus, dass das Bild schon Thumbnailgröße hat

$thumbwidth = $imagewidth;

$thumbheight = $imageheight;

// Breite skalieren falls nötig

if ($thumbwidth > $maxthumbwidth) {

$factor = $maxthumbwidth / $thumbwidth;

$thumbwidth \*= $factor;

$thumbheight \*= $factor;

}

// Höhe skalieren, falls nötig

if ($thumbheight > $maxthumbheight) {

$factor = $maxthumbheight / $thumbheight;

$thumbwidth \*= $factor;

$thumbheight \*= $factor;

}

$thumbheight = floor($thumbheight);

$thumbwidth = floor($thumbwidth);

// Thumbnail erstellen

$thumb = imagecreatetruecolor($thumbwidth, $thumbheight);

imagecopyresampled(

$thumb,

$image,

0, 0, 0, 0, // Startposition des Ausschnittes

$thumbwidth, $thumbheight,

$imagewidth, $imageheight

);

// In Datei speichern

$thumbfile = 'thumbs/' . $imagefile;

imagepng($thumb, $thumbfile);

imagedestroy($thumb);

$\_db\_host = "database";

$\_db\_datenbank = "web";

$\_db\_username = "web";

$\_db\_passwort = "12345";

$conn = new mysqli($\_db\_host, $\_db\_username, $\_db\_passwort, $\_db\_datenbank);

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$insertStatement = "INSERT INTO images(path) VALUES ('$target\_file');";

if ($\_rest = $conn->query($insertStatement)) {

echo "<br>Image $target\_file has been added to the database.";

} else {

echo "<br>NO insertion into database";

}

$conn->close();

}

}

?>

**Loop through rows (table syntax html)**

    $sql = "select name, lastname, username, image\_path, password from v\_persons";

        if ($res = $conn->query($sql)) {

            echo '<table>

                <tr>

                    <th>Username</th>

                    <th>Name</th>

                    <th>Lastname</th>

                    <th>Image Path </th>

                    <th>Password </th>

                </tr>

    ';

            while ($row = $res->fetch\_assoc()) {

                echo '

                 <tr>

                    <td>' . $row['username'] . '</td>

                    <td>' . $row['name'] . '</td>

                    <td>' . $row['lastname'] . '</td>

                    <td>' . $row['image\_path'] . '</td>

                    <td>' . $row['password'] . '</td>

                </tr>

            ';

            }

            echo '

        </table>

    ';

**Date Format**

* d - The day of the month (from 01 to 31)
* D - A textual representation of a day (three letters)
* j - The day of the month without leading zeros (1 to 31)
* l (lowercase 'L') - A full textual representation of a day
* N - The ISO-8601 numeric representation of a day (1 for Monday, 7 for Sunday)
* S - The English ordinal suffix for the day of the month (2 characters st, nd, rd or th. Works well with j)
* w - A numeric representation of the day (0 for Sunday, 6 for Saturday)
* z - The day of the year (from 0 through 365)
* W - The ISO-8601 week number of year (weeks starting on Monday)
* F - A full textual representation of a month (January through December)
* m - A numeric representation of a month (from 01 to 12)
* M - A short textual representation of a month (three letters)
* n - A numeric representation of a month, without leading zeros (1 to 12)
* t - The number of days in the given month
* L - Whether it's a leap year (1 if it is a leap year, 0 otherwise)
* o - The ISO-8601 year number
* Y - A four digit representation of a year
* y - A two digit representation of a year
* a - Lowercase am or pm
* A - Uppercase AM or PM
* B - Swatch Internet time (000 to 999)
* g - 12-hour format of an hour (1 to 12)
* G - 24-hour format of an hour (0 to 23)
* h - 12-hour format of an hour (01 to 12)
* H - 24-hour format of an hour (00 to 23)
* i - Minutes with leading zeros (00 to 59)
* s - Seconds, with leading zeros (00 to 59)
* u - Microseconds (added in PHP 5.2.2)
* e - The timezone identifier (Examples: UTC, GMT, Atlantic/Azores)
* I (capital i) - Whether the date is in daylights savings time (1 if Daylight Savings Time, 0 otherwise)
* O - Difference to Greenwich time (GMT) in hours (Example: +0100)
* P - Difference to Greenwich time (GMT) in hours:minutes (added in PHP 5.1.3)
* T - Timezone abbreviations (Examples: EST, MDT)
* Z - Timezone offset in seconds. The offset for timezones west of UTC is negative (-43200 to 50400)
* c - The ISO-8601 date (e.g. 2013-05-05T16:34:42+00:00)
* r - The RFC 2822 formatted date (e.g. Fri, 12 Apr 2013 12:01:05 +0200)
* U - The seconds since the Unix Epoch (January 1 1970 00:00:00 GMT)

**Login**

$\_db\_host = "database";

$\_db\_datenbank = "web";

$\_db\_username = "web";

$\_db\_password = "12345";

SESSION\_START();

$conn = new mysqli($\_db\_host, $\_db\_username, $\_db\_password, $\_db\_datenbank);

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

if (!empty($\_POST["submit"])) {

$\_username = $conn->real\_escape\_string($\_POST["username"]);

$\_password = $conn->real\_escape\_string($\_POST["password"]);

$\_password = password\_hash($\_password, PASSWORD\_DEFAULT);

$\_sql = "SELECT \* FROM login\_username WHERE username = '$\_username' AND user\_deleted = 0 LIMIT 1";

if ($\_res = $conn->query($\_sql)) {

$result = $\_res->fetch\_assoc();

if ($\_res->num\_rows > 0 && password\_verify($\_POST["password"], $result["password"])) {

echo "Login successful<br>";

$\_SESSION["login"] = true;

$\_SESSION["user"] = $result;

$\_sql = "UPDATE login\_username SET last\_login=NOW() WHERE id=*"* . $\_SESSION["user"]["id"];

$conn->query($\_sql);

}

$\_res->close();

}

}

$conn->close();

if (!($\_SESSION["login"] ?? false)) {

include("login.html");

exit();

}

echo "User " . $\_SESSION["user"]["username"] . " is logged in since " . $\_SESSION["user"]["last\_login"] . "<br>";

strlen(trim($testtermin)) && validateDate(trim($testtermin)) && new DateTime($testtermin) > new DateTime("today")

**Pagination**

# irgendwo im head

$data = file\_get\_contents("./articles.json");

$obj = json\_decode($data);

$pagenumber = $\_GET["pagenumber"] ?? $\_SESSION["pagenumber"] ?? 1;

$from = ($pagenumber - 1) \* 3;

$to = $from + 3;

$to = min($to, sizeof($obj));

if ($from > $to || $from < 0) {

$from = 0;

$to = 3;

$pagenumber = 1;

}

$\_SESSION["pagenumber"] = $pagenumber;

for ($i = $from; $i < $to; $i++) {

$date = date("d.m.Y H:i", $obj[$i]->pubDate);

echo "<div class='box'>

<div>

<img src=\"{$obj[$i]->image}\">

</div>

<div>

<p class=\"rubric\"><small>{$obj[$i]->rubric}</small></p>

<a class=\"title\" href=\"{$obj[$i]->link}\"><h2>{$obj[$i]->title}</h2></a>

<p class=\"date\"><small>$date</small></p>

<p>{$obj[$i]->description}</p>

</div>

</div>";

}

# im body, wo die tabelle gerendert werden soll

$pages = sizeof($obj) / 3;

# ceil rounds up

$pages = ceil($pages);

if ($pagenumber - 1 < 1) {

echo "<a class=\"inactive\">&laquo;</a>";

} else {

echo "<a href=\"news-prev.php\">&laquo;</a>";

}

for ($i = 1; $i <= $pages; $i++) {

$active = $i == $pagenumber ? " class=\"active\"" : "";

echo "<a href=\"news.php?pagenumber=$i\"$active>$i</a>";

}

if ($pagenumber + 1 > $pages) {

echo "<a class=\"inactive\">&raquo;</a>";

} else {

echo "<a href=\"news-next.php\">&raquo;</a>";

}

**Prepared Statements**

<?php  
$servername = "localhost";  
$username = "username";  
$password = "password";  
$dbname = "myDB";  
  
// Create connection  
$conn = new mysqli($servername, $username, $password, $dbname);  
  
// Check connection  
if ($conn->connect\_error) {  
  die("Connection failed: " . $conn->connect\_error);  
}  
  
// prepare and bind  
$stmt = $conn->prepare("INSERT INTO MyGuests (firstname, lastname, email) VALUES (?, ?, ?)");  
$stmt->bind\_param("sss", $firstname, $lastname, $email);  
  
// set parameters and execute  
$firstname = "John";  
$lastname = "Doe";  
$email = "john@example.com";  
$stmt->execute();  
  
$firstname = "Mary";  
$lastname = "Moe";  
$email = "mary@example.com";  
$stmt->execute();  
  
$firstname = "Julie";  
$lastname = "Dooley";  
$email = "julie@example.com";  
$stmt->execute();  
  
echo "New records created successfully";  
  
$stmt->close();  
$conn->close();  
?>

**Image Upload simplified**

**PHP**

<?php  
$target\_dir = "uploads/";  
$target\_file = $target\_dir . basename($\_FILES["fileToUpload"]["name"]);  
$uploadOk = 1;  
$imageFileType = strtolower(pathinfo($target\_file,PATHINFO\_EXTENSION));  
  
// Check if image file is a actual image or fake image  
if(isset($\_POST["submit"])) {  
  $check = getimagesize($\_FILES["fileToUpload"]["tmp\_name"]);  
  if($check !== false) {  
    echo "File is an image - " . $check["mime"] . ".";  
    $uploadOk = 1;  
  } else {  
    echo "File is not an image.";  
    $uploadOk = 0;  
  }  
}  
  
// Check if file already exists  
if (file\_exists($target\_file)) {  
  echo "Sorry, file already exists.";  
  $uploadOk = 0;  
}  
  
// Check file size  
if ($\_FILES["fileToUpload"]["size"] > 500000) {  
  echo "Sorry, your file is too large.";  
  $uploadOk = 0;  
}  
  
// Allow certain file formats  
if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"  
&& $imageFileType != "gif" ) {  
  echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";  
  $uploadOk = 0;  
}  
  
// Check if $uploadOk is set to 0 by an error  
if ($uploadOk == 0) {  
  echo "Sorry, your file was not uploaded.";  
// if everything is ok, try to upload file  
} else {  
  if (move\_uploaded\_file($\_FILES["fileToUpload"]["tmp\_name"], $target\_file)) {  
    echo "The file ". htmlspecialchars( basename( $\_FILES["fileToUpload"]["name"])). " has been uploaded.";  
  } else {  
    echo "Sorry, there was an error uploading your file.";  
  }  
}  
?>

**HTML**

<!DOCTYPE html>  
<html>  
<body>  
  
<form action="upload.php" method="post" enctype="multipart/form-data">  
  Select image to upload:  
  <input type="file" name="fileToUpload" id="fileToUpload">  
  <input type="submit" value="Upload Image" name="submit">  
</form>  
  
</body>  
</html>