

Technische Projektdokumentation

Inhaltsverzeichnis

Inhaltsverzeichnis	1
Aufgabenstellung	1
Verwendete Technologien	1
Weg zum Ziel	1

Aufgabenstellung

Unsere Aufgabenstellung ist im Projektantrag dokumentiert.

Link zum Projektantrag: [📄 Projektantrag_Oberlin_Suter](#)

Verwendete Technologien

Xampp
GD-Library
Bootstrap
PHP
HTML

Weg zum Ziel

Begonnen haben wir mit der Umsetzung der Logik. Dafür haben wir uns zuerst angeschaut, wie wir ein Bild hochladen und bearbeiten können. Wir recherchierten wir auf verschiedenen Seiten, doch die beste war W3School. Wir konnten das Hochladen der eigenen Datei und das Herunterladen der veränderten Datei mit reinem PHP lösen. Das Hochladen funktioniert wie folgt:

```

<?php
$target_dir = "uploads/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
$uploadOk = 1;
$imageFileType = strtolower(pathinfo($target_file, flags: PATHINFO_EXTENSION));
// Check if image file is a actual image or fake image
if (isset($_POST["submit"])) {
    $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
    if ($check !== false) {
        switch ($imageFileType) {
            case "png":
                $im_php = imagecreatefrompng($_FILES["fileToUpload"]["tmp_name"]);
                break;
            case "jpeg":
            case "jpg":
                $im_php = imagecreatefromjpeg($_FILES["fileToUpload"]["tmp_name"]);
                break;
            case "webp":
                $im_php = imagecreatefromwebp($_FILES["fileToUpload"]["tmp_name"]);
                break;
            default:
                $im_php = imagecreatefromjpeg(filename: "teapot.jpg");
        }
    }
}

```

Wie man sieht, wird das Bild nicht effektiv gespeichert, sondern nur in einer Variable abgelegt.

Für den Download haben wir die Seite php.net verwendet.

```

$file = 'bild.png';

if (file_exists($file)) {
    header(header: 'Content-Description: File Transfer');
    header(header: 'Content-Type: application/octet-stream');
    header(header: 'Content-Disposition: attachment; filename="' . basename($file) . '"');
    header(header: 'Expires: 0');
    header(header: 'Cache-Control: must-revalidate');
    header(header: 'Pragma: public');
    header(header: 'Content-Length: ' . filesize($file));
    readfile($file);
    exit;
}

```

Und dies ist der letzte Teil unserer Logik, in welchem mir das Bild mit der GD-Library bearbeiten.

```

$im_php = imagescale($im_php, width: -1, height: 300);
$sizeX = imagesx($im_php);
$im_php = imagecrop($im_php, ['x' => ($sizeX - 300) / 2, 'y' => 0, 'width' => 300, 'height' => 300]);

imageWebP($im_php, to: "bild.png", quality: 80);
header(header: "Location: http://localhost/");
exit;

```

Und für die letzten Kompetenzen haben wir schlicht und einfach mit Bootstrap gearbeitet, so wurde es für alle Geräte anschaulich.

```
<div class="mb-3">
  <label for="fileToUpload" class="form-label">Select image to upload</label>
  <input type="file" name="fileToUpload" id="fileToUpload" class="form-control" required>

</div>
<input type="submit" value="Upload Image" name="submit" class="btn btn-primary">
```

Zudem haben wir die Rückmeldungen eingebaut, falls es einmal einen Fehler geben würde. Dies ist aber so gut wie unmöglich.