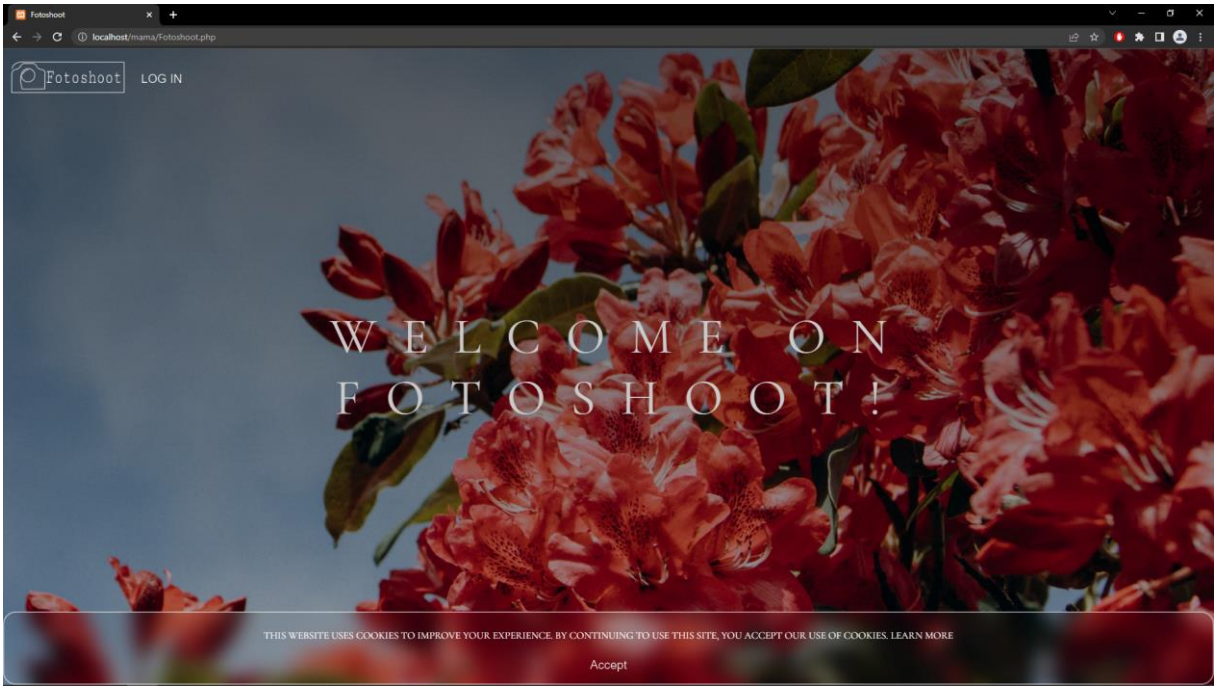
Hambuch Dominik LAJ929, Dankó Krisztián ZYEEH

2023.05.13

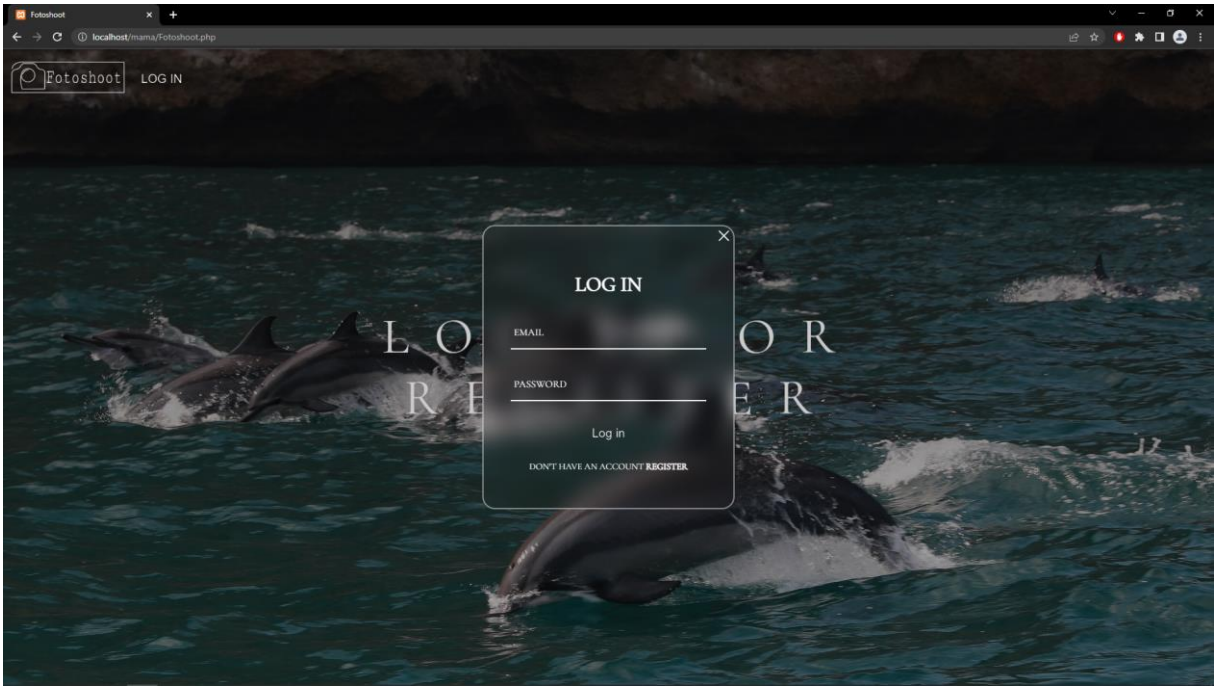
Web programming I.- Homework Fotoshoot

This website is working as a gallery. You can register using an username and an account, then you can log in. After you logged in, you can upload pictures to the gallery. The website uses animations for better experience. Here you can see the main page:

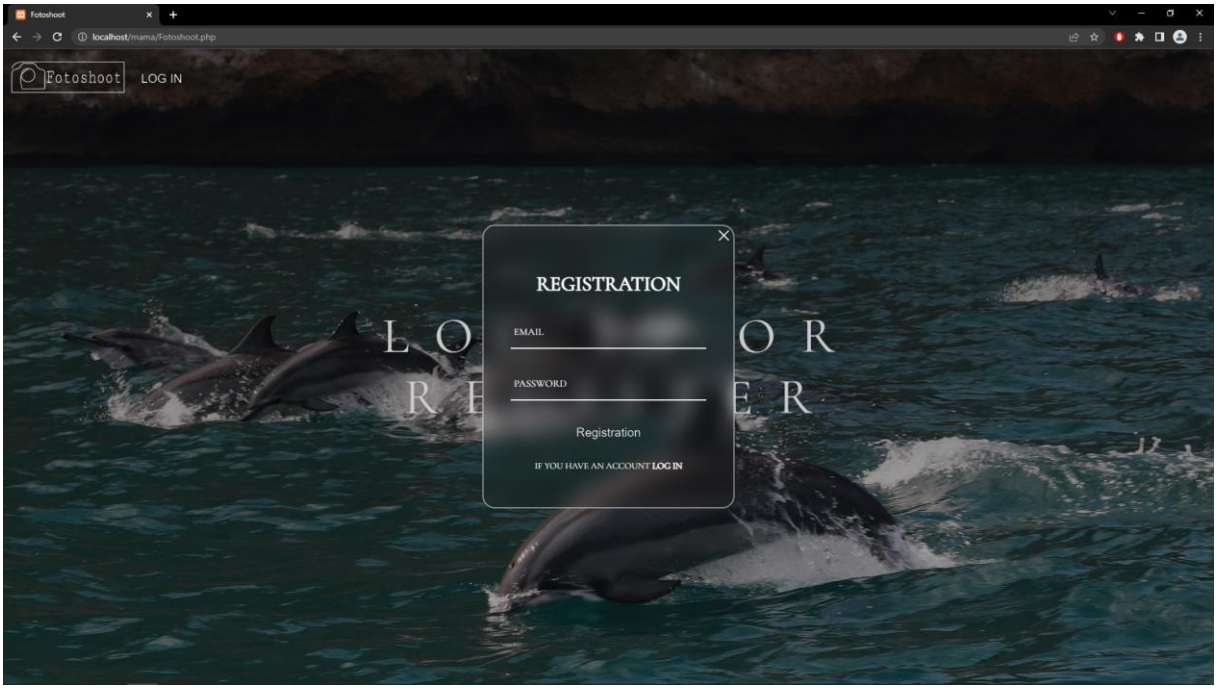
The main page contains a cookie, login/registration



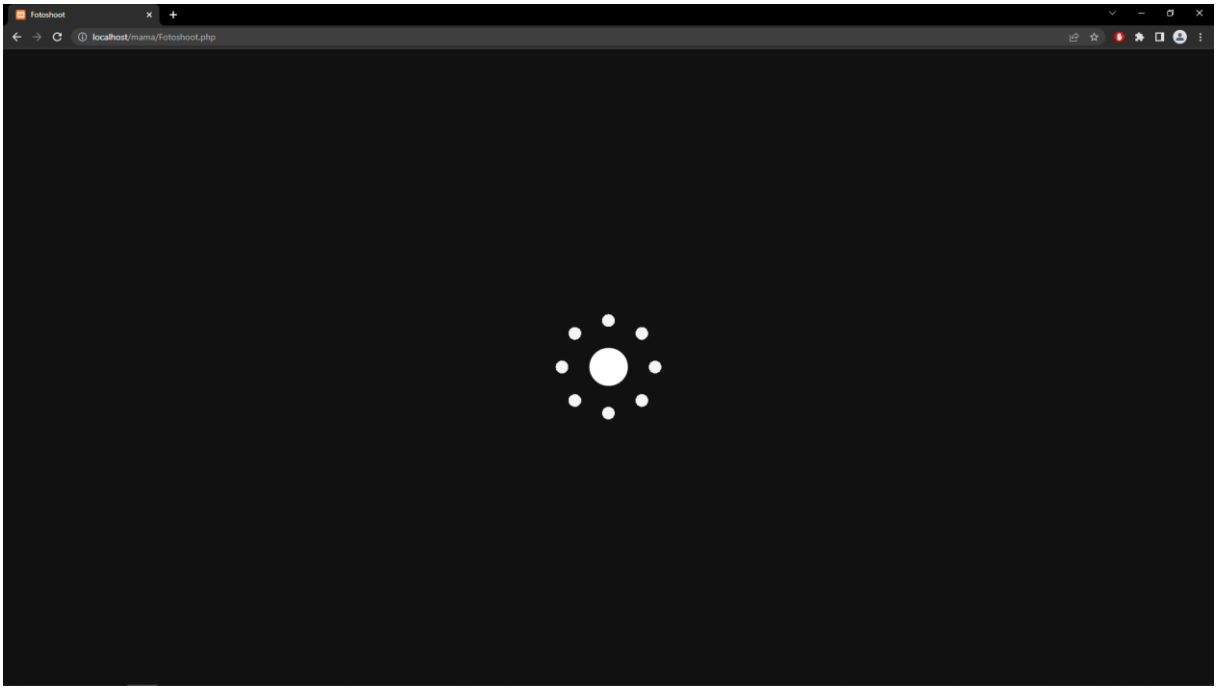
Here you can see the login panel with a registration label.



Here you can see the registration:

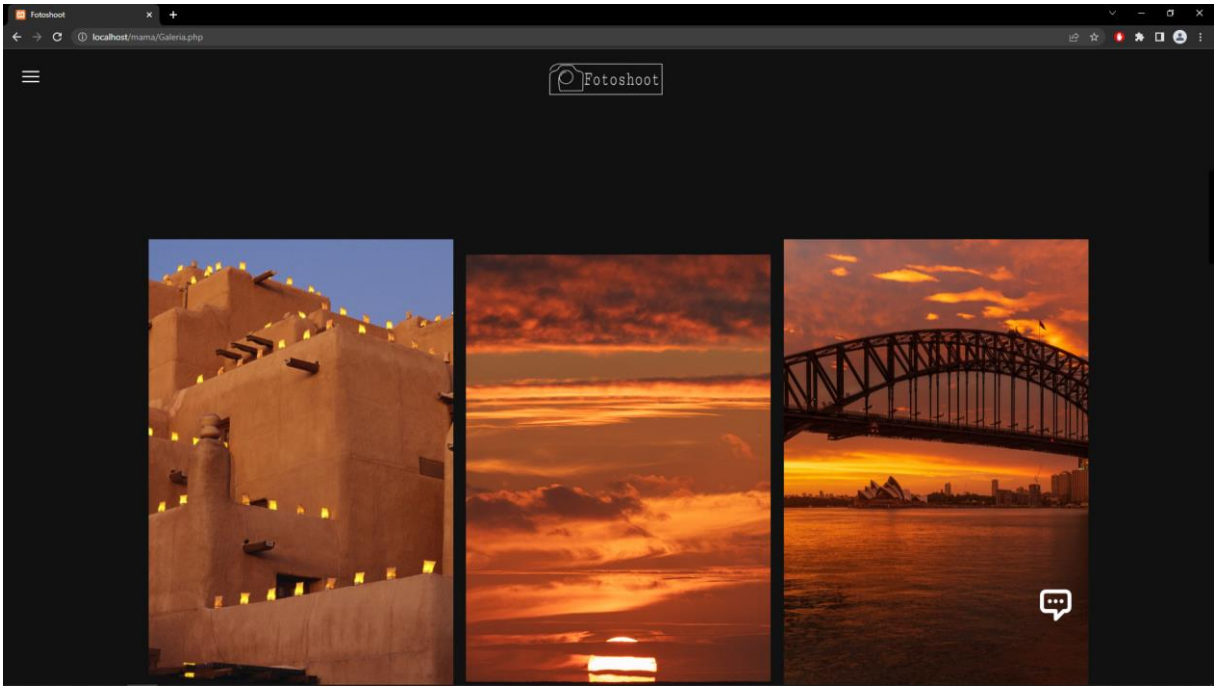


The webpage uses a loading animation, its demonstrated on the picture:



Here you can see the website when you logged in. This website also uses a nice scrolling animation. Here are all the pictures.

This is the gallery part if you scrolled down. All the base pictures are shown, and all the other pictures the users uploaded are here.



On the left side there is a dropdown menu with some additional features/options to the current website, you can see „Upload”, „About”, „Log out”

Once you click the Upload feature, a popup appears that requires you to choose a file and upload it as a picture to the gallery of the website.

Once you click the About option, you are redirected to another page that contains information about the „Fotoshoot” company. Also you can find there the Geolocation. On the „Pages” option, you can find

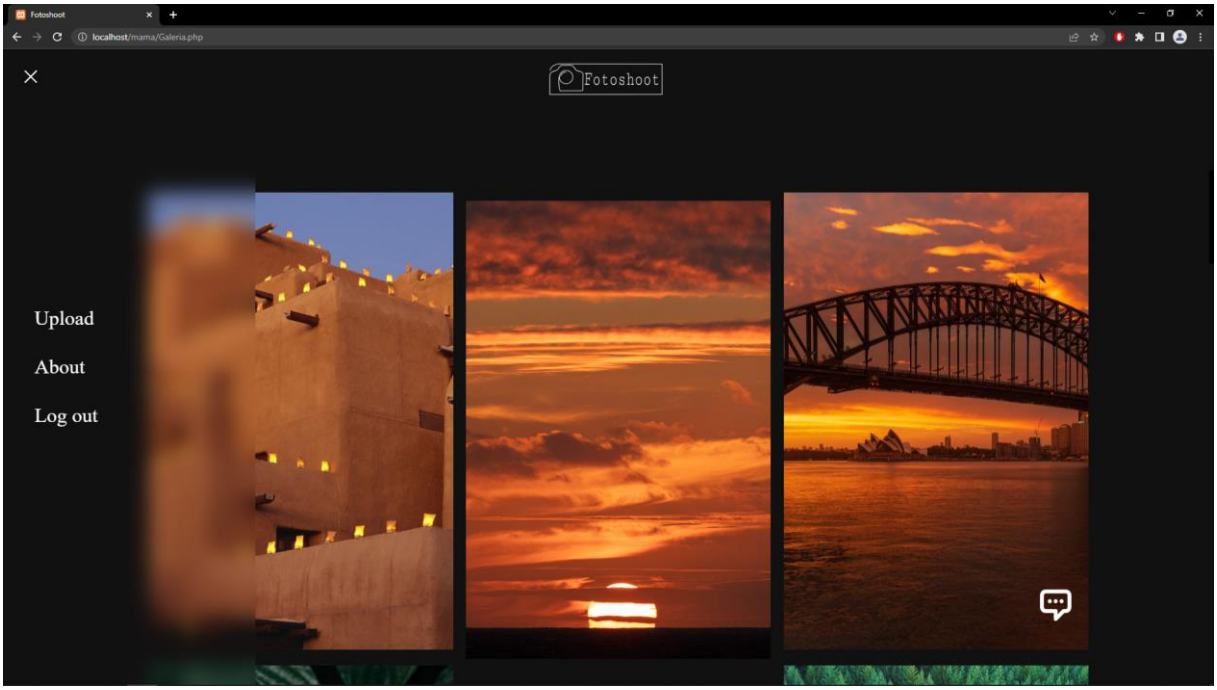
- SVG, which stands for Scalable Vector Graphics, which is a vector image format for 2D graphics.

- Web Workers, that are a JavaScript feature that allows code to run in the background of a web page, without blocking the user interface or the main thread of execution.

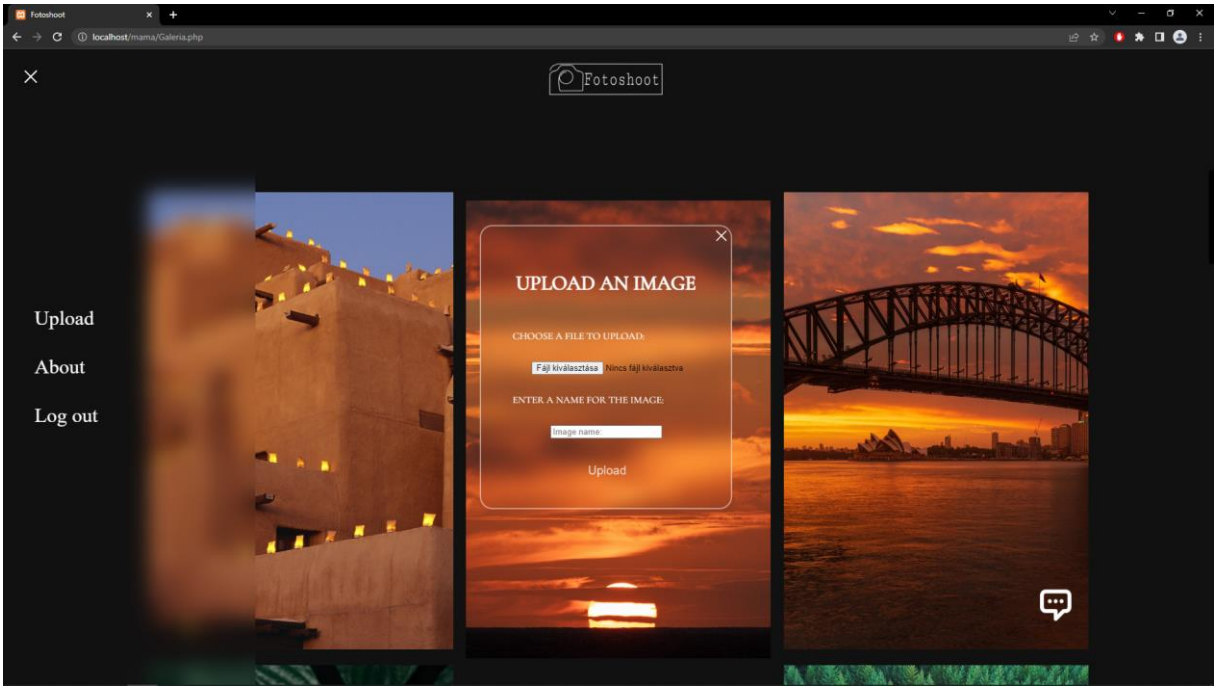
- The Drag and Drop API, this is a JavaScript interface that allows web developers to implement drag and drop functionality in web applications.

- The HTML Canvas, that is a JavaScript API that allows web developers to dynamically create and manipulate graphics and animations on a web page.

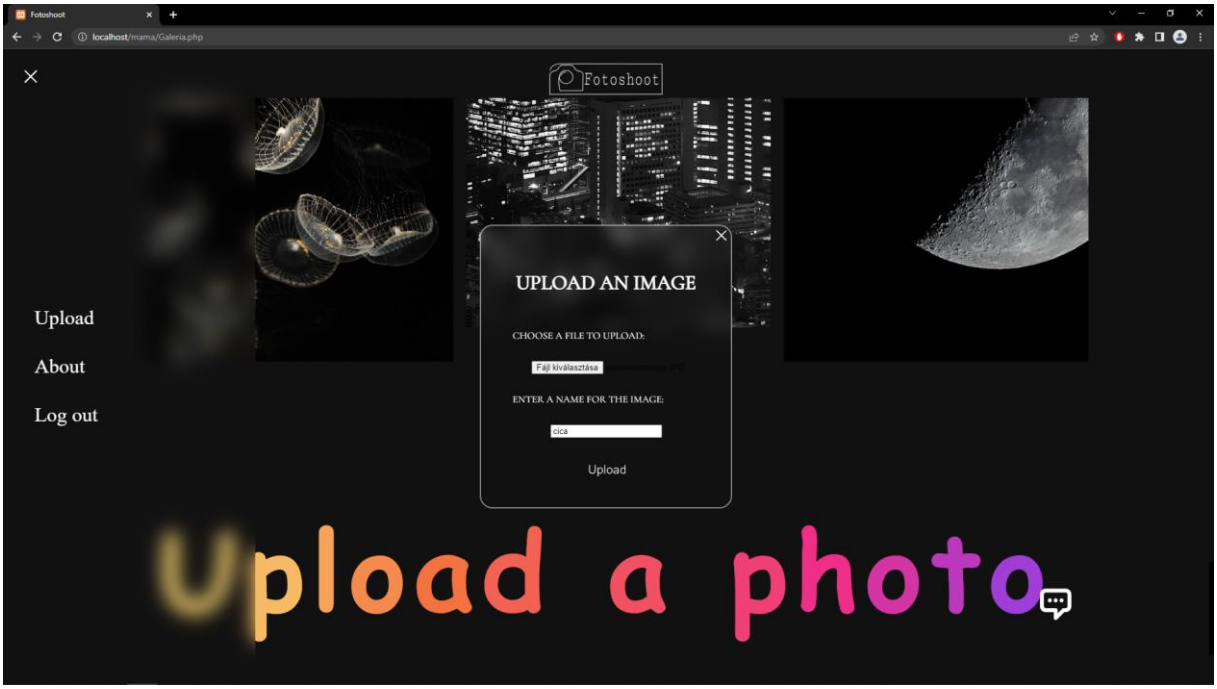
- Sass, this is a CSS preprocessor that extends the functionality of standard CSS with features such as variables, nesting, mixins, and functions.



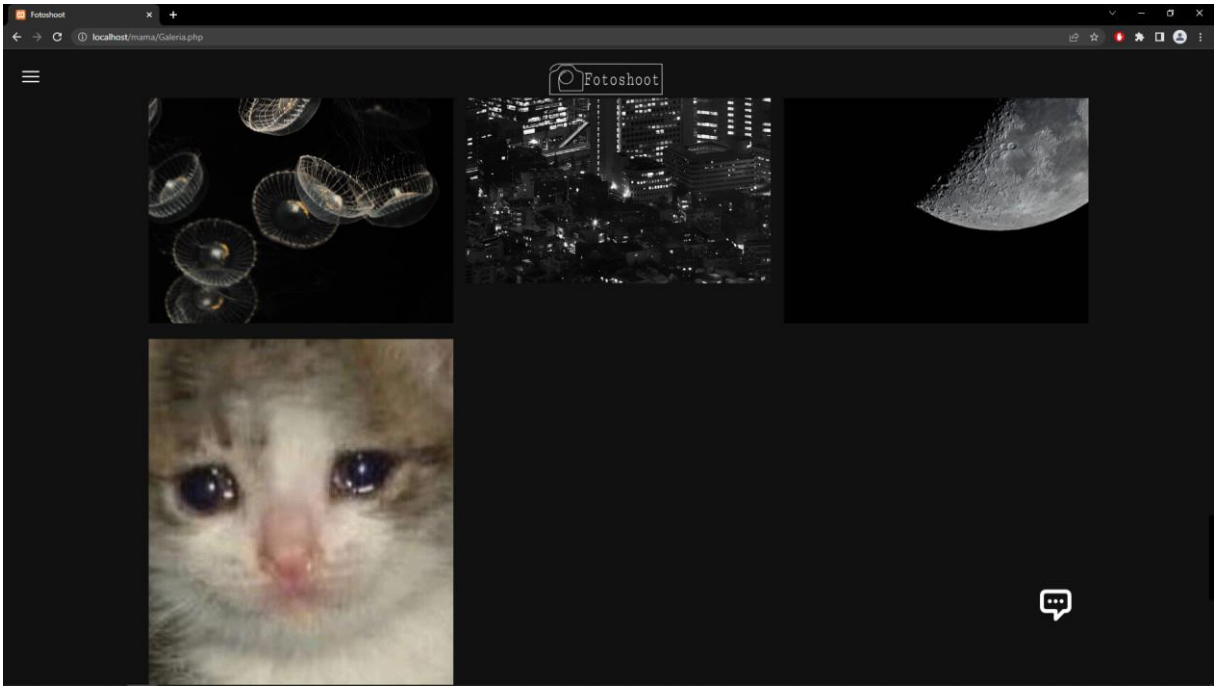
Here you can see the Upload popup:



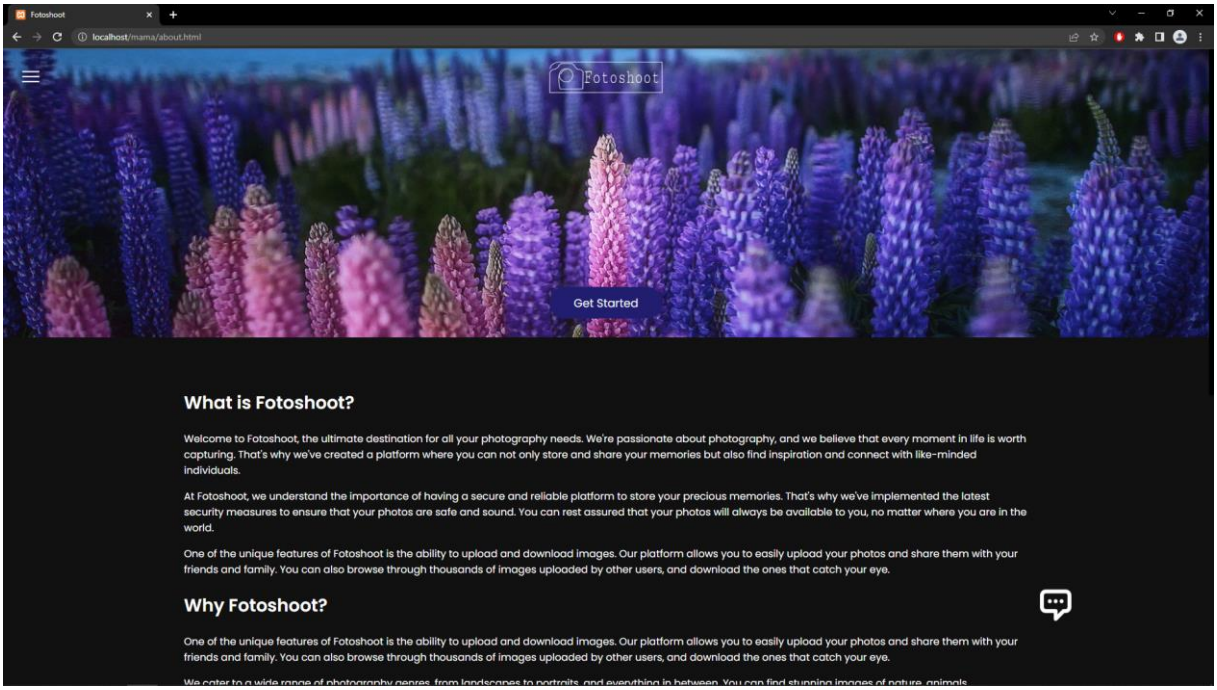
Here we are uploading a picture from our computer with the name „cica”. The picture will appear on the bottom of the gallery.



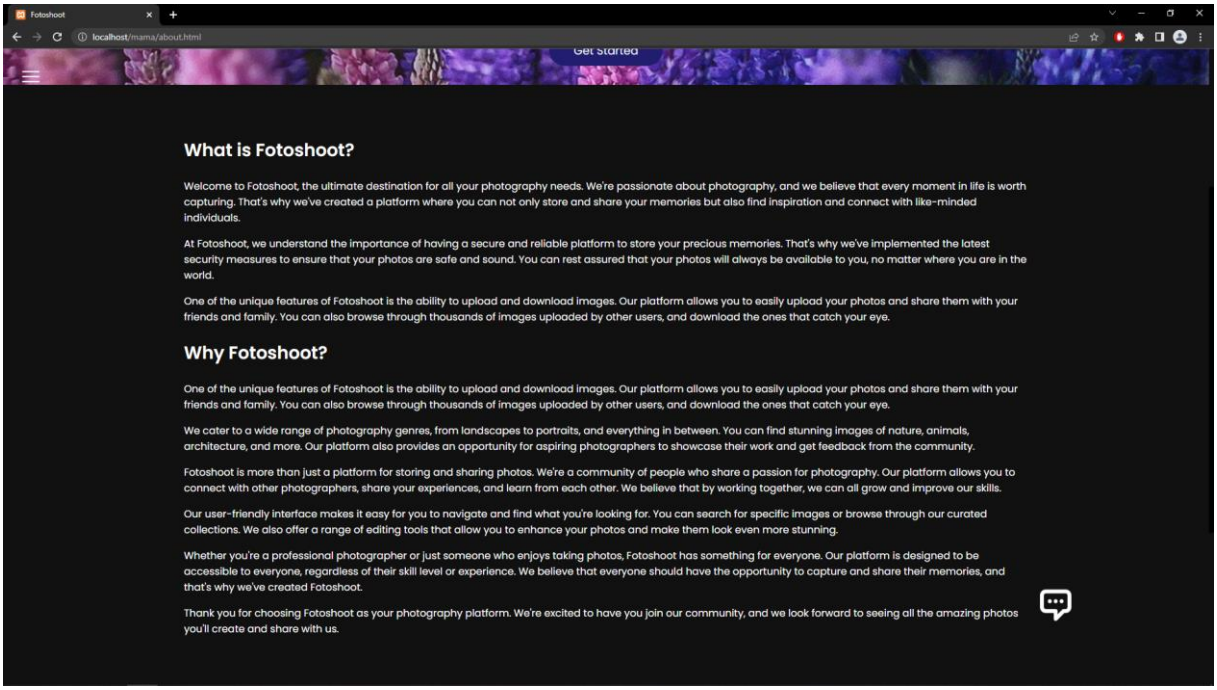
Here we can see the cat image that appeared on the bottom of the website. On this picture we can see a very sad cat.

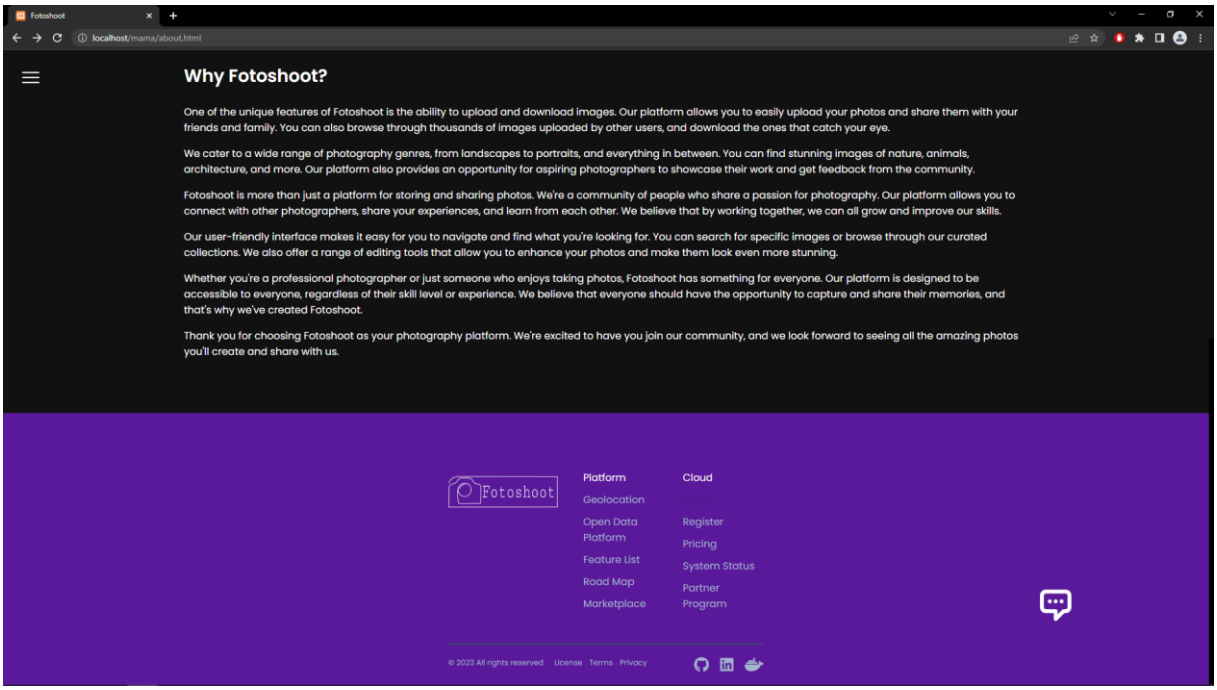


Here we can see the „About” feature, it contains a lot of information about the „Fotoshoot” company.

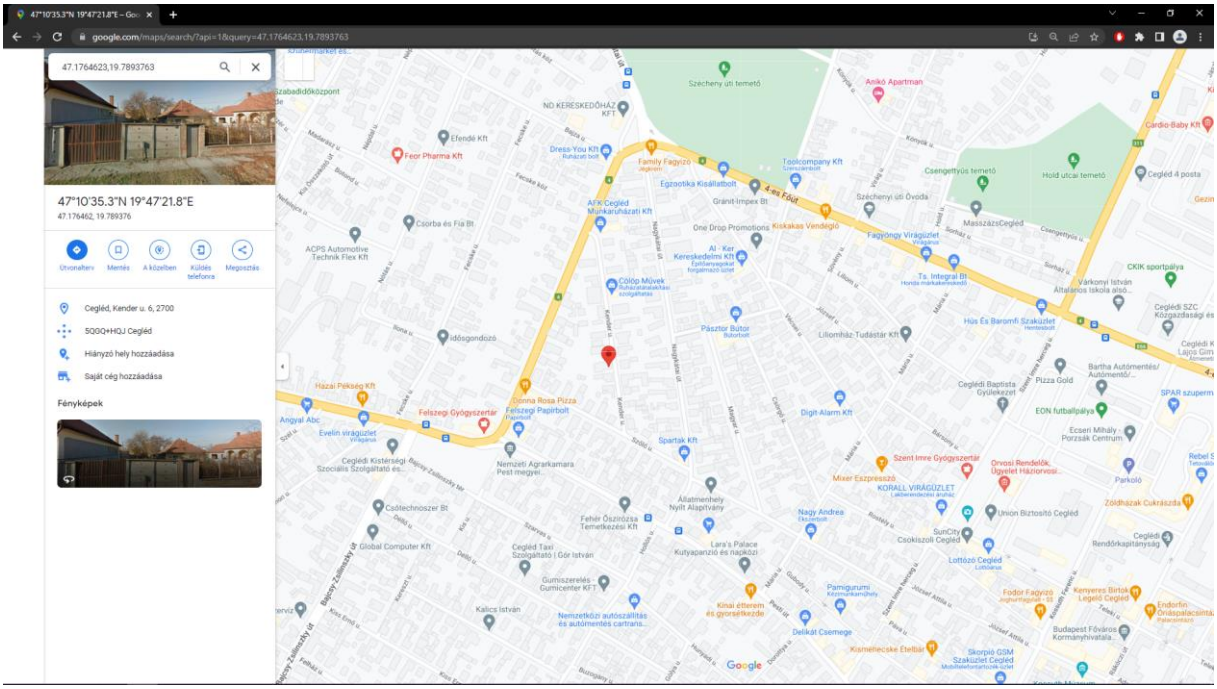


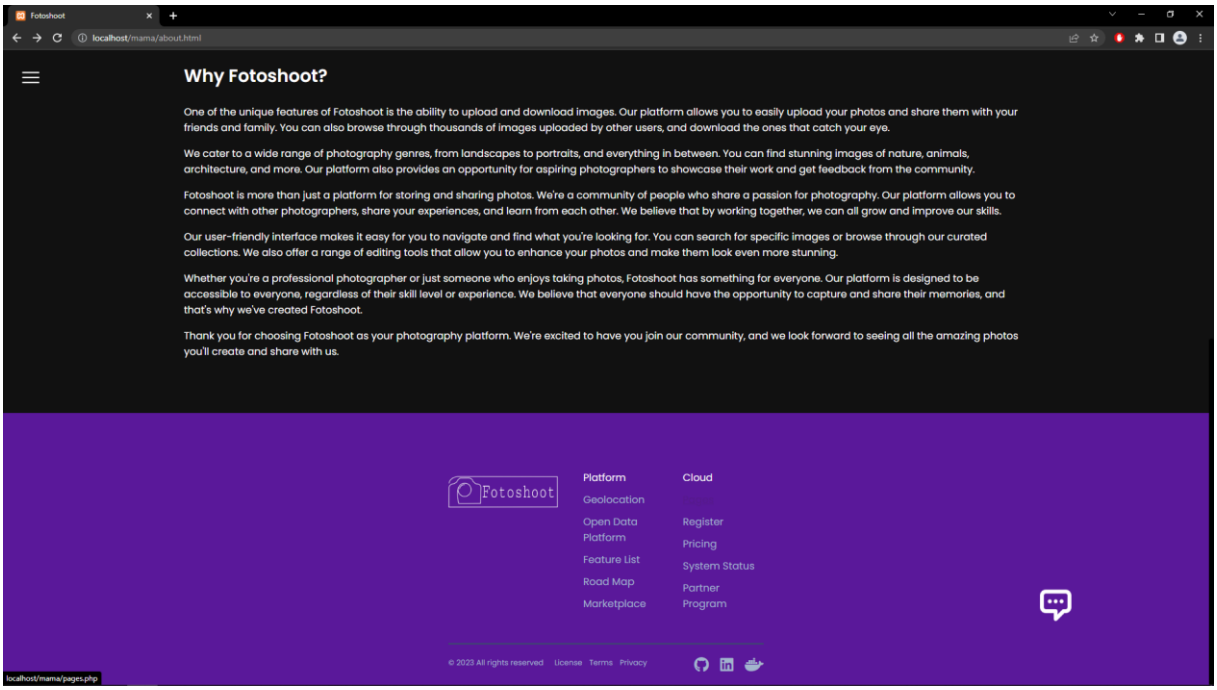
On the bottom of the website we can see the previously mentioned options. The „Pages” and the „Geolocation”.



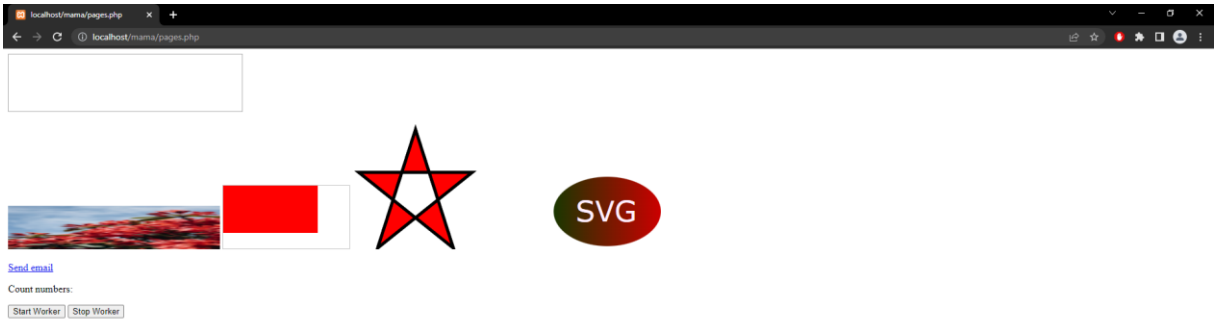


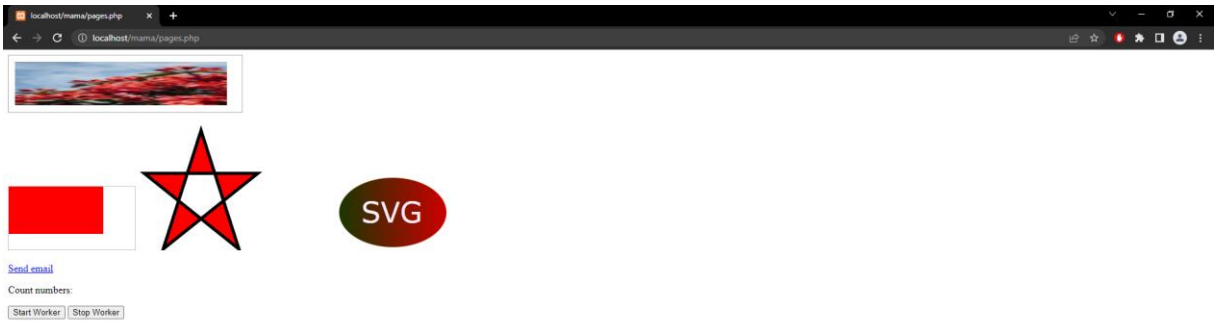
Here you can see the working Geolocation.





Here we can see the contents of the Pages website.

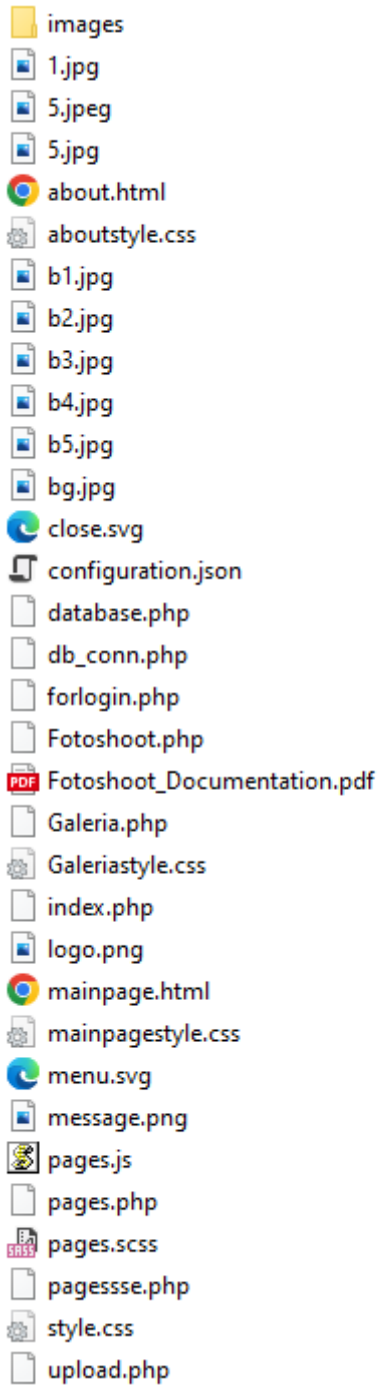




As you can see we could drag the image in the top box.


localhost/mama/pages.php
localhost/mama/pages.php
SVG
Send email
Count numbers: 4
Start Worker
Stop Worker

Codes:



This picture depicts the files used on our website.

```css
* {
    box-sizing: border-box;
}

body {
    background-color: #111;
    overscroll-behavior: none;
    margin: 0;
    padding: 0;
    overflow-x: hidden;
  display: flexbox;

}
.message{
  z-index:9999;
  height:50px;
  width:50px;
  position:fixed;
  left:86%;
  top:85%;
  cursor:pointer;
  opacity:0;
}
.box{
  height: 30px;
  display: flex;
  cursor: pointer;
  padding: 10px 20px;
  border-radius: 30px;
  align-items:center;
  background: #fff;
}
.box:hover{
  width: 400px;
}
.box input{
  outline: none;
  border: none;
}
```

Here is a code snippet of the mainpagestyle.css

It contains all the styling of the mainpage.html

```php
<?php

$sname="localhost";
$email="root";
$password="";
$db_name="test";
$conn = mysqli_connect($sname, $email, $password, $db_name);

if (!$conn) {
    echo "Connection failed!";
}
```

Here we can see the database connection with php, it echoes if the connection is failed.

```php
<?php
session_start();
include "db_conn.php";

if (isset($_POST['email']) && isset($_POST['password'])){

    function validate($data){
       $data = trim($data);
       $data = stripslashes($data);
       $data = htmlspecialchars($data);
       return $data;
    }

    $email = validate($_POST['email']);
    $pass = validate($_POST['password']);

    $user_data = 'email='. $email;


    if (empty($email)) {
        $response = array('status' => 'error', 'message' => 'Password is required');
        echo json_encode($response);
        exit();
    }else if(empty($pass)){
        $response = array('status' => 'error', 'message' => 'Password is required');
        echo json_encode($response);
        exit();
    }

    else{

        $sql = "SELECT * FROM registration WHERE email='$email' ";
        $result = mysqli_query($conn, $sql);

        if (mysqli_num_rows($result) > 0) {
            $response = array('status' => 'error', 'message' => 'The email is taken try another');
            echo json_encode($response);
            exit();
        }else {
           $sql2 = "INSERT INTO registration(email, password) VALUES('$email', '$pass')";
           $result2 = mysqli_query($conn, $sql2);
           if ($result2) {
                $response = array('status' => 'success', 'message' => 'Your account has been created');
                echo json_encode($response);
             exit();
           }else {
                $response = array('status' => 'error', 'message' => 'An unknown error occurred');
                echo json_encode($response);
                exit();
           }
        }
```

Here we can see the database.php that inserts data into our database to the registration table.

```php
<?php
session_start();
include "db_conn.php";

if (isset($_POST['email']) && isset($_POST['password'])) {

    function validate($data){
       $data = trim($data);
       $data = stripslashes($data);
       $data = htmlspecialchars($data);
       return $data;
    }

    $email = validate($_POST['email']);
    $pass = validate($_POST['password']);

    if (empty($email)) {
        $response = array('status' => 'error', 'message' => 'Email or password is incorrect');
        echo json_encode($response);
        exit();
    }else if(empty($pass)){
        $response = array('status' => 'error', 'message' => 'Password is requied');
        echo json_encode($response);
        exit();
    }else{
        $sql = "SELECT * FROM registration WHERE email='$email' AND password='$pass'";

        $result = mysqli_query($conn, $sql);

        if (mysqli_num_rows($result) === 1) {
            $row = mysqli_fetch_assoc($result);
            if ($row['email'] === $email && $row['password'] === $pass) {
                $_SESSION['email'] = $row['email'];
                $response = array('status' => 'success');
                echo json_encode($response);
            }else{
                $response = array('status' => 'error', 'message' => 'Email or password is incorrect');
                echo json_encode($response);
            }
        }else{
            $response = array('status' => 'error', 'message' => 'Email or password is incorrect');
            echo json_encode($response);
        }
    }

}else{
    header("Location: Fotoshoot.php");
    exit();
}
```

Here we can see the code used for logging in. At the end of the code it redirects us to the Fotoshoot.php.

```html
<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>
<br>
<img id="drag1" src="b1.jpg" draggable="true" ondragstart="drag(event)" width="336" height="69">
```

Here we can see the drag and drop api's html code.

```
<canvas id="myCanvas" width="200" height="100"
style="border:1px solid #c3c3c3;">
Your browser does not support the canvas element.
</canvas>
```

```
<svg width="300" height="200">
  <polygon points="100,10 40,198 190,78 10,78 160,198"
  style="fill:red;stroke:black;stroke-width:5;fill-rule:evenodd;" />
</svg>
<svg height="130" width="500">
  <defs>
    <linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">
      <stop offset="0%" style="stop-color:rgb(25,55,0);stop-opacity:1" />
      <stop offset="100%" style="stop-color:rgb(205,0,0);stop-opacity:1" />
    </linearGradient>
  </defs>
  <ellipse cx="100" cy="70" rx="85" ry="55" fill="url(#grad1)" />
  <text fill="#fff0ff" font-size="45" font-family="Verdana" x="50" y="86">SVG</text>
  Sorry, your browser does not support inline SVG.
</svg>
```

We can see the code snippets for the canvas and the svg above.

The configuration.json collects all the used urls in a menu, which can be later used in the index.php

```
{
  "menuItems": [
    {
      "name": "gallery",
      "url": "Galeria.php"
    },
    {
      "name": "about",
      "url": "about.html"
    },
    {
      "name": "logout",
      "url": "fotoshoot.php"
    }
  ]
}
```

This index.php code is for each menu item, it reads the menu names and common data from the configuration file.

```php
<?php

$configFile = 'configuration.json';
$configData = file_get_contents($configFile);
$config = json_decode($configData, true);

$menuItems = $config['menuItems'];

if (isset($_GET['page'])) {
    $page = $_GET['page'];
} else {
    $page = $menuItems[0]['name'];
}

$selectedItem = null;
foreach ($menuItems as $menuItem) {
    if ($menuItem['name'] === $page) {
        $selectedItem = $menuItem;
        break;
    }
}

if ($selectedItem !== null) {
    $url = $selectedItem['url'];
    header("Location: $url");
    exit;
} else {
    header("Location: " . $menuItems[0]['url']);
    exit;
}

?>
```

The website uses an ajax code, so when there is an error in the login or registration process, the page doesnt have to be refreshed:

```
    <script>
$(document).ready(function(){
    $("#loginForm").submit(function(event){
        event.preventDefault();
        $.ajax({
            url: "forlogin.php",
            type: "POST",
            data: $(this).serialize(),
            dataType: "json",
            success: function(response){
                if(response.status === "success"){
                    window.location.href = "mainpage.html";
                }else{
                    document.getElementById("error-message").style.display = "block";
                    $("#error-message").html(response.message);
                }
            }
        });
    });
});
</script>
<script>
$(document).ready(function(){
    $("#RegistratinForm").submit(function(event){
        event.preventDefault();
        $.ajax({
            url: "database.php",
            type: "POST",
            data: $(this).serialize(),
            dataType: "json",
            success: function(response){
                if(response.status == 'error'){
                    document.getElementById("error-message1").style.display = "block";
                    $("#error-message1").html(response.message);
                }else{
                    document.getElementById("error-message2").style.display = "block";
                    document.getElementById("error-message1").style.display = "none";
                    $("#error-message2").html(response.message);
                }
            }
        });
    });
});
</script>
```

Overall, the gallery website should provide an engaging and informative experience for users, while also showcasing the artwork in the best possible light and engagement with the gallery.

Github url:

https://github.com/hambucs/webbeadando

Websites used for the project:

https://www.w3schools.com/

https://hackr.io/tutorials/learn-php

https://www.freecodecamp.org/news/best-css-and-css3-tutorial/

https://web.dev/learn/css/

https://animate.style/

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Animations/Using_CSS_animations

The nethely link for the website:

www.fotoshoot.nhely.hu/fotoshoot.php