Hướng dẫn phát triển ứng dụng PHP tương tác với camera

# 1. Cài đặt môi trường phát triển

## 1.1 Cài đặt Laragon

* Tải và cài đặt Laragon từ [laragon.org](http://laragon.org).
* Chạy file cài đặt và làm theo các hướng dẫn để hoàn tất quá trình cài đặt.
* Khởi động Laragon và kiểm tra các dịch vụ như Apache và MySQL đã hoạt động.

## 1.2 Cài đặt PHP

* Laragon đi kèm với PHP, nhưng bạn có thể cập nhật phiên bản PHP nếu cần.
* Để cập nhật PHP, tải phiên bản PHP mong muốn từ [php.net](https://www.php.net/downloads).
* Giải nén tệp tải về và di chuyển thư mục PHP vào thư mục bin\php của Laragon.
* Trong Laragon, chọn Menu > Laragon > PHP > Version > Add Version và chọn phiên bản PHP mới.

## 1.3 Cấu hình Laragon

* Mở Laragon và vào Menu > Preferences để cấu hình các cài đặt cần thiết.
* Trong tab Services & Ports, đảm bảo rằng Apache và MySQL đang hoạt động và cổng được cấu hình đúng.
* Kiểm tra tab PHP để đảm bảo phiên bản PHP đúng đang được sử dụng.

## 1.4 Thiết lập thư mục dự án

* Tạo một thư mục mới cho dự án trong thư mục www của Laragon. Ví dụ: C:\Laragon\www\camera\_project.
* Mở Laragon, chọn Menu > Quick app > Blank và nhập tên dự án, ví dụ: camera\_project. Laragon sẽ tự động tạo thư mục dự án cho bạn.

# 2. Tạo dự án PHP mới

## 2.1 Tạo thư mục dự án

Trong Laragon, tạo một thư mục mới cho dự án của bạn trong thư mục www. Ví dụ: C:\Laragon\www\camera\_project.

Bạn cũng có thể sử dụng tính năng Quick app của Laragon để tạo một dự án trống:

* Mở Laragon.
* Chọn Menu > Quick app > Blank.
* Nhập tên dự án, ví dụ: camera\_project.
* Laragon sẽ tự động tạo thư mục dự án cho bạn trong C:\Laragon\www.

## 2.2 Thiết lập cấu trúc dự án

Trong thư mục dự án, tạo các thư mục và tệp sau để cấu trúc dự án của bạn:

* Tạo thư mục css để chứa các tệp CSS.
* Tạo thư mục js để chứa các tệp JavaScript.
* Tạo thư mục images để lưu trữ các hình ảnh chụp từ camera.
* Tạo tệp index.php trong thư mục gốc của dự án để làm tệp chính cho giao diện web của bạn.

Sau khi tạo cấu trúc, cấu trúc thư mục dự án sẽ trông giống như sau:  
Sao chép mã  
camera\_project/

├── css/

├── js/

├── images/

└── index.php

# 3. Cài đặt và cấu hình camera

## 3.1 Sử dụng thư viện JavaScript

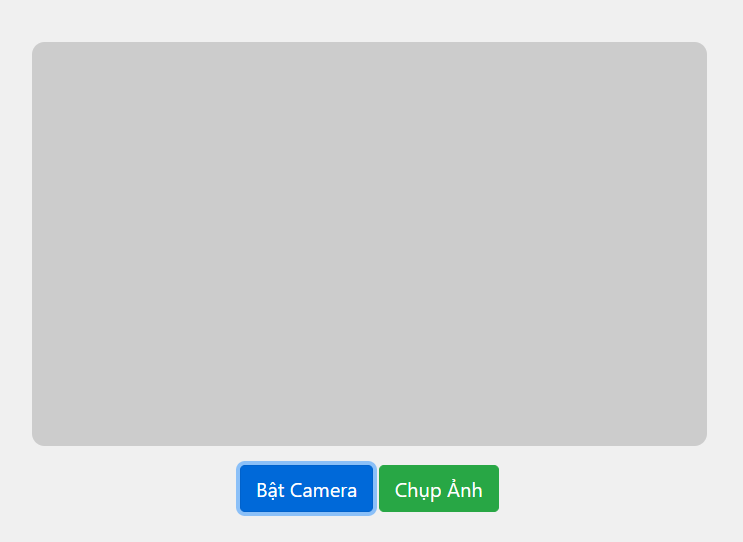
* Sử dụng thư viện getUserMedia để truy cập camera của người dùng.
* Tạo tệp camera.js trong thư mục js của dự án và chạy thử mã JavaScript sau để truy cập và hiển thị dữ liệu từ camera:

| // camera.js navigator.mediaDevices.getUserMedia({ video: true })  .then(function(stream) {  const video = document.querySelector('video');  video.srcObject = stream;  video.play();  })  .catch(function(err) {  console.log("An error occurred: " + err);  }); |
| --- |

# 4. Hướng dẫn mã nguồn chi tiết

## 4.1 Tạo tệp index.php và viết mã HTML

Trong tệp index.php này, bạn có một giao diện đơn giản để hiển thị video từ camera, các nút điều khiển để tắt/bật camera và chụp ảnh. Giao diện sử dụng Bootstrap để có giao diện thân thiện với người dùng và sẵn sàng cho việc tích hợp các tính năng bổ sung bằng JavaScript.



| **<!DOCTYPE html>** <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Camera Stream</title>  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">  <link rel="stylesheet" href="CSS/style.css"> </head>  <body>  <div class="container">  <div class="row">  <div class="col-md-6">  <video id="video" autoplay class="img-fluid"></video>  <div class="controls">  <button id="toggleCamera" class="btn btn-primary">Tắt Camera</button>  <button id="takePhoto" class="btn btn-success">Chụp Ảnh</button>  </div>  </div>  <div class="col-md-6">  <canvas id="canvas" style="display:none;"></canvas>  <img id="photo" alt="Chụp ảnh" style="display:none;" class="img-fluid" />  </div>  </div>  </div>   <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>  <script src="js/script.js"></script> </body>  </html> |
| --- |

## 4.2. Tạo tệp style.css và viết mã CSS

| **body** {  display: flex;  justify-content: center;  align-items: center;  height: 100vh;  margin: 0;  background-color: #f0f0f0; }  .container {  text-align: center; }  **#video** {  transform: scaleX(-1) ;  border-radius: 10px; }  .controls {  margin-top: 10px; } |
| --- |

## 4.2. Tạo tệp script.js

| $(document).ready(**function**() {  *// Khai báo các biến và phần tử DOM cần thiết*  **const** video = $('#video')[0];  **const** canvas = $('#canvas')[0];  **const** photo = $('#photo')[0];  **const** toggleCameraButton = $('#toggleCamera');  **const** takePhotoButton = $('#takePhoto');  **let** stream;   *// Bắt đầu camera khi tải trang*  startCamera();   *// Bắt sự kiện click vào nút Toggle Camera*  toggleCameraButton.click(**function**() {  **if** (stream) {  stopCamera();  } **else** {  startCamera();  }  });   *// Bắt sự kiện click vào nút Take Photo*  takePhotoButton.click(**function**() {  **if** (stream) {  **try** {  **const** imageData = takePhoto();  *// Gửi dữ liệu ảnh lên server bằng Ajax*  $.ajax({  type: 'POST',  url: 'Services/save\_photo.php', *// Thay đổi đường dẫn tới file PHP lưu ảnh*  data: { imageBase64: imageData }, *// Dữ liệu ảnh dưới dạng base64*  success: **function**(response) {  **const** fileName = getFileNameFromResponse(response);  displayPhoto(fileName);  },  error: **function**(xhr, status, error) {  handleAjaxError(error);  }  });  } **catch** (error) {  console.error('Lỗi khi chụp ảnh từ video:', error);  alert('Đã xảy ra lỗi khi chụp ảnh.');  }  }  });   *// Hàm bắt đầu camera*  **async** **function** **startCamera**() {  **try** {  stream = **await** navigator.mediaDevices.getUserMedia({ video: true });  video.srcObject = stream;  toggleCameraButton.text('Tắt Camera');  } **catch** (error) {  console.error('Lỗi khi truy cập camera:', error);  }  }   *// Hàm dừng camera*  **function** **stopCamera**() {  **if** (stream) {  stream.getTracks().forEach(track => track.stop());  video.srcObject = null;  stream = null;  toggleCameraButton.text('Bật Camera');  }  }   *// Hàm chụp ảnh từ video và trả về dữ liệu base64*  **function** **takePhoto**() {  **const** context = canvas.getContext('2d');  canvas.width = video.videoWidth;  canvas.height = video.videoHeight;   *// Đảo ngược hình ảnh nếu cần thiết*  context.translate(canvas.width, 0);  context.scale(-1, 1);   *// Vẽ hình ảnh từ video lên canvas*  context.drawImage(video, 0, 0, canvas.width, canvas.height);   *// Chuyển dữ liệu thành base64*  **const** data = canvas.toDataURL('image/png');   *// Đặt lại ma trận biến đổi để không ảnh hưởng đến lần chụp ảnh tiếp theo*  context.setTransform(1, 0, 0, 1, 0, 0);   **return** data;  }   *// Hàm xử lý response từ server để lấy tên file ảnh*  **function** **getFileNameFromResponse**(response) {  **const** parts = response.split(/[\\\/]/);  **return** parts[parts.length - 2] + '/' + parts[parts.length - 1];  }   *// Hàm hiển thị ảnh sau khi lưu thành công*  **function** **displayPhoto**(fileName) {  photo.src = fileName;  photo.style.display = 'block';  }   *// Hàm xử lý lỗi khi gửi Ajax*  **function** **handleAjaxError**(error) {  console.error('Lỗi khi gửi dữ liệu ảnh lên server:', error);  alert('Đã xảy ra lỗi khi lưu ảnh.');  } }); |
| --- |

## 4.3. Tạo tệp save\_image.php

| **<?php** **if** (**isset**($\_POST['imageBase64'])) {  $imageData = $\_POST['imageBase64'];   *// Chuẩn bị dữ liệu ảnh để giải mã*  $imageData = str\_replace('data:image/png;base64,', '', $imageData); *// Loại bỏ phần header của base64*  $imageData = str\_replace(' ', '+', $imageData); *// Thay thế các khoảng trắng*   *// Giải mã dữ liệu base64 thành dữ liệu nhị phân của ảnh*  $imageBinary = base64\_decode($imageData);   *// Đường dẫn lưu trữ ảnh (cùng cấp với thư mục gốc của dự án)*  $uploadPath = dirname(**\_\_DIR\_\_**) . '/Images/';   *// Tạo thư mục nếu chưa tồn tại*  **if** (!file\_exists($uploadPath)) {  mkdir($uploadPath, 0777, **true**);  }   $filename = 'photo\_' . date('Y-m-d-H-i-s') . '.png'; *// Định dạng tên file*  $filepath = $uploadPath . $filename;   *// Lưu ảnh vào thư mục*  **if** (file\_put\_contents($filepath, $imageBinary) !== **false**) {  **echo** $filepath;  } **else** {  **echo** 'Lỗi khi lưu file.';   } } **else** {  **echo** 'Không có dữ liệu ảnh được gửi lên.'; } **?>** |
| --- |

# 5. Kiểm tra và hoàn thiện

