# Xây dựng một Webform PHP để Xem Camera và Tích hợp Nhận diện Khuôn mặt

# 1. Giới thiệu

#### **Mô tả dự án**

Dự án này nhằm mục đích xây dựng một webform PHP để cho phép người dùng xem hình ảnh từ camera và tích hợp chức năng nhận diện khuôn mặt. Điều này sẽ cung cấp một giao diện đơn giản và hiệu quả cho việc quản lý và sử dụng dữ liệu hình ảnh.

# 2. Yêu cầu hệ thống

#### **Yêu cầu phần cứng**

* Máy tính với đủ tài nguyên để chạy Laravel và hỗ trợ camera nếu cần thiết.

#### **Yêu cầu phần mềm**

* PHP 8.2 trở lên
* Composer (để quản lý các dependencies của Laravel)
* Laravel Framework (phiên bản mới nhất)

# 3. Cài đặt môi trường

## 3.1. Cài đặt Composer

Composer là một công cụ quản lý gói PHP và là một yêu cầu cần thiết để cài đặt và quản lý các thư viện PHP trong dự án Laravel của bạn. Để cài đặt Composer, bạn có thể làm theo các bước sau:

1. Truy cập vào trang web chính thức của Composer để tải xuống bản cài đặt phù hợp với hệ điều hành của bạn: <https://getcomposer.org/download/>.
2. Sau khi tải xuống, bạn có thể cài đặt Composer bằng cách chạy tệp cài đặt đã tải xuống hoặc sử dụng các lệnh dòng lệnh để cài đặt toàn cầu.

Để kiểm tra xem Composer đã được cài đặt thành công hay chưa, bạn có thể mở terminal và chạy lệnh sau: composer --version

1. Nếu bạn nhận được phiên bản Composer hiện tại, điều này có nghĩa là Composer đã được cài đặt thành công trên máy tính của bạn.

## 3.2. Cài đặt Laravel

Sau khi cài đặt Composer, bạn có thể sử dụng Composer để tạo một dự án Laravel mới. Bạn có thể làm như sau:

1. Mở terminal và chuyển đến thư mục mà bạn muốn cài đặt dự án Laravel.

Sử dụng Composer để tạo một dự án Laravel mới bằng lệnh sau:  
composer create-project --prefer-dist laravel/laravel tên\_dự\_án

1. Trong đó, tên\_dự\_án là tên mà bạn muốn đặt cho dự án của mình.
2. Sau khi lệnh hoàn thành, Composer sẽ tải về và cài đặt tất cả các dependencies cần thiết cho Laravel. Quá trình này có thể mất một vài phút tùy thuộc vào tốc độ internet của bạn.

Khi quá trình tải xuống và cài đặt hoàn tất, bạn có thể di chuyển vào thư mục dự án mới bằng lệnh sau: cd tên\_dự\_án

1. Bây giờ, dự án Laravel của bạn đã sẵn sàng để phát triển. Bạn có thể tiếp tục với các bước tiếp theo để triển khai tích hợp xem camera và nhận diện khuôn mặt như bạn đã lên kế hoạch.

# 4. Thiết kế và triển khai webform

## 4.1. Tạo Controller và View cho đăng nhập

### **4.1.1. Tạo controller AuthController**

Sử dụng Artisan để tạo controller AuthController:

| php artisan make:controller AuthController |
| --- |

Lệnh này sẽ tạo một file controller có tên AuthController.php trong thư mục app/Http/Controllers.

### 4.1.2. Tạo view cho đăng nhập

Sử dụng Artisan để tạo view login.blade.php trong thư mục resources/views/auth

| php artisan make:view auth.login |
| --- |

Lệnh này sẽ tạo một file view có tên login.blade.php trong thư mục resources/views/auth.

### 4.1.3. Thiết lập route

Để sử dụng controller và view mới tạo, hãy thiết lập route trong routes/web.php:

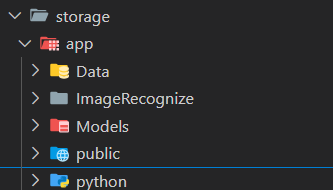
| *// routes/web.php*  **use** **App**\**Http**\**Controllers**\**AuthController**;  *// Route cho trang đăng nhập*  Route::get('/login', [AuthController::class, 'showLoginForm'])->name('login'); |
| --- |

## 4.2. Tạo thư mục và các file cần thiết cho việc nhận diện

Để triển khai tính năng nhận diện trong dự án Laravel của bạn, bạn cần tạo các thư mục sau trong thư mục storage của dự án:

* **Data**: Được sử dụng để lưu trữ tệp output.txt, nơi chứa kết quả từ quá trình nhận diện.
* **ImageRecognize**: Thư mục này sẽ lưu trữ các hình ảnh chụp từ camera để thực hiện nhận diện.
* **Model**: Nơi lưu trữ tệp encoding.txt, chứa vector trọng số khuôn mặt được huấn luyện.
* **Python**: Thư mục này dành cho các script Python liên quan đến nhận diện khuôn mặt, ví dụ như FaceRecognition.py.

Bạn cần đảm bảo các thư mục này được tạo và quản lý đúng cách để triển khai tính năng nhận diện khuôn mặt trong dự án của mình.



## 4.3. Tạo asset trong thư mục public

* Trong thư mục public/assets, bạn cần tạo các thư mục con sau:
  + image: Chứa các hình ảnh được sử dụng trong ứng dụng.
  + css: Lưu trữ các file CSS để tùy chỉnh giao diện và kiểu dáng.
  + js: Để viết các tập lệnh JavaScript để quản lý các tương tác và logic của người dùng trên giao diện.

## 4.4. Giao diện trang Login

### 4.4.1. login.blade.php

* Tạo giao diện trang đăng nhập Auth/login.blade.php để cho phép người dùng nhập thông tin đăng nhập.

| **<!DOCTYPE html>** <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <meta name="csrf-token" content="{{ csrf\_token() }}">  <title>Đăng nhập</title>  *<!-- Link CSS của Bootstrap -->*  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">  *<!-- Link thư viện jQuery -->*  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>  *<!-- CSS tùy chỉnh -->*  <link rel="stylesheet" href="{{ asset('Assets/css/login.css') }}"> </head>  <body>  <div class="container mt-5">  <div class="row justify-content-center">  <div class="col-md-6">  <div class="card">  <div class="card-header">Đăng nhập</div>  <div class="card-body">  <form method="POST" action="{{ route('login.submit') }}">  @csrf  <div class="form-group">  <label for="username">Tên đăng nhập:</label>  <input type="text" class="form-control" id="username" name="username"  value="{{ old('username') }}" required autofocus>  </div>  <div class="form-group">  <label for="password">Mật khẩu:</label>  <input type="password" class="form-control" id="password" name="password" required>  </div>  <button type="submit" class="btn btn-primary">Đăng nhập</button>   @error('loginError')  <div class="text-danger mt-2">{{ $message }}</div>  @enderror  </form>  </div>  <div class="card-footer">  <button type="button" class="btn btn-success" id="btnFaceLogin">Đăng nhập bằng khuôn  mặt</button>  </div>  </div>  </div>  <div class="col-md-6">  <div id="Recognize" class="recognizeface w-100 h-100 d-none">  <div class="wrapper">  <video class="w-100" id="video" autoplay class="img-fluid rounded"></video>  <div id="loadingIndicator" class="d-none" style="text-align: center;">  <img src="./assets/images/loading.gif" alt="Loading..." />  </div>  </div>  <div class="controls">  <button id="recognizeButton" class="btn btn-success">Nhận dạng</button>  <h1 id="personName" class="d-none"></h1>  </div>  <canvas id="canvas" class="d-none"></canvas>  </div>  </div>  </div>  </div>   <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>  <script src="{{ asset('Assets/js/login.js') }}"></script> </body>  </html> |
| --- |

Tại thư mục public/asset/image bạn có thể lưu 1 hình ảnh loading đặt tên loading.gif tại <https://i.gifer.com/ZKZg.gif>

### 4.4.2. login.css

* Tệp login.css được sử dụng để tùy chỉnh giao diện và kiểu dáng của trang đăng nhập. Nó bao gồm các quy tắc CSS để cải thiện trải nghiệm người dùng, bao gồm việc điều chỉnh màu sắc, định dạng văn bản, và bố cục của các thành phần trong trang đăng nhập.

| **body** {  display: flex;  justify-content: center;  align-items: center;  height: 100vh;  margin: 0;  background-color: #f0f0f0; }  .main {  max-height: 500px; }  **#video** {  transform: scaleX(-1);  background-color: #ccc;  border-radius: 10px;  overflow: hidden;  position: relative; }  **#Recognize** {  text-align: center; }  .controls {  margin-top: 10px; }  .wrapper {  position: relative; }  **#loadingIndicator** {  position: absolute;  left: 50%;  top: 50%;  transform: translate(-50%, -50%); }  **#loadingIndicator** **img** {  width: 50px;  height: 50px; } |
| --- |

### 4.4.3. login.js

* Tệp login.js là một tập lệnh JavaScript được sử dụng để xử lý các tương tác và logic của người dùng trên trang đăng nhập. Nó có thể bao gồm các chức năng như xác thực dữ liệu đầu vào, xử lý sự kiện nhấn nút đăng nhập, và giao tiếp với máy chủ để xác minh thông tin người dùng.

| $(document).ready(**function** () {  **let** stream;   *// Hàm bắt đầu camera*  **async** **function** **startCamera**() {  **try** {  stream = **await** navigator.mediaDevices.getUserMedia({ video: true });  **const** video = $('#video')[0];  video.srcObject = stream;  } **catch** (error) {  console.error('Lỗi khi truy cập camera:', error);  }  }   *// Hàm dừng camera*  **function** **stopCamera**() {  **if** (stream) {  stream.getTracks().forEach(track => track.stop());  **const** video = $('#video')[0];  video.srcObject = null;  stream = null;  }  }   *// Xử lý sự kiện click vào nút "Đăng nhập bằng khuôn mặt"*  $('#btnFaceLogin').click(**function** () {  $('#Recognize').toggle(); *// Hiển thị hoặc ẩn phần nhận dạng khuôn mặt*  **if** (stream) {  stopCamera();  $('#Recognize').removeClass('d-block').addClass('d-none');  } **else** {  startCamera();  $('#Recognize').removeClass('d-none').addClass('d-block');  }  });   *// Xử lý sự kiện click vào nút "Nhận dạng"*  $('#recognizeButton').click(**function** () {  takephoto();  });   **function** **takephoto**() {  *// Lấy video và canvas elements*  **const** video = $('#video')[0];  **const** canvas = $('#canvas')[0];  **const** context = canvas.getContext('2d');   *// Đặt kích thước canvas bằng với kích thước video*  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 khung hình hiện tại của video lên canvas*  context.drawImage(video, 0, 0, canvas.width, canvas.height);   *// Lấy dữ liệu ảnh từ canvas dưới dạng base64*  **const** imageBase64 = canvas.toDataURL('image/png');  $('#loadingIndicator').removeClass('d-none').addClass('d-block');   *// Gửi dữ liệu ảnh base64 đến máy chủ qua AJAX*  $.ajax({  url: '/save-photo',  method: 'POST',  headers: {  'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content'),  'Content-Type': 'application/json'  },  data: JSON.stringify({ imageBase64: imageBase64 }),  success: **function** (response) {  console.log('Ảnh đã được lưu:', response.filepath);  recognizeFace(response.filepath);  },  error: **function** (xhr, status, error) {  console.error('Lỗi:', error);  }  });  }   **function** **recognizeFace**(imagePath) {  $.ajax({  url: '/recognize-face',  method: 'POST',  headers: {  'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content'),  'Content-Type': 'application/json'  },  data: JSON.stringify({  imagePath: imagePath  }),  success: **function** (response) {  console.log('Tên người được nhận dạng:', response.recognizedName);  $('#personName').text(response.recognizedName).removeClass('d-none').addClass('d-block');  },  error: **function** (xhr, status, error) {  console.error('Lỗi:', error);  },  complete: **function** () {  $('#loadingIndicator').removeClass('d-block').addClass('d-none');  }  });  } }); |
| --- |

## 4.5. AuthController

| **<?php**  **namespace** **App**\**Http**\**Controllers**;  **use** **Illuminate**\**Http**\**Request**; **use** **Illuminate**\**Support**\**Facades**\**Storage**;  **class** **AuthController** **extends** **Controller** {  *// Hiển thị form đăng nhập*  **public** **function** **showLoginForm**()  {  **return** view('auth.login');  }   *// Xử lý đăng nhập*  **public** **function** **login**(Request $request)  {  // *Nếu đăng nhập thành công*  **return** redirect('/welcome'); *// Thay thế '/dashboard' bằng route bạn muốn chuyển hướng đến khi đăng nhập thành công*   *// Nếu đăng nhập thất bại*  *// return back()->withInput()->withErrors(['loginError' => 'Đăng nhập không thành công']);*  } } |
| --- |

### 

### 4.5.1. Phương thức lưu hình ảnh

| **public** **function** **savePhoto**(Request $request) {  **if** ($request->ajax()) {  **if** ($request->has('imageBase64')) {  $imageData = $request->input('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 tới thư mục trong storage/app*  $uploadPath = 'ImageRecognize/';   *// Tạo thư mục nếu chưa tồn tại*  **if** (!Storage::exists($uploadPath)) {  Storage::makeDirectory($uploadPath, 0777, **true**, **true**);  }   *// Tạo tên file duy nhất*  $filename = 'photo\_' . date('Y-m-d-H-i-s') . '.png';   *// Lưu ảnh vào thư mục trong storage*  Storage::put($uploadPath . $filename, $imageBinary);   *// Lấy đường dẫn tuyệt đối của file đã lưu*  $filePath = Storage::path($uploadPath . $filename);   *// Trả về đường dẫn tuyệt đối*  **return** response()->json(['filepath' => realpath($filePath)]);  } **else** {  **return** response()->json(['error' => 'Không có dữ liệu ảnh được gửi lên.'], 400);  }  } } |
| --- |

### 4.5.2. Phương thức nhận diện

| **public** **function** **recognizeFace**(Request $request) {  **if** ($request->ajax()) {  **if** ($request->has('imagePath')) {  $imagePath = $request->input('imagePath');   *// Kiểm tra và lấy đường dẫn chính xác của các file và script Python*  $pythonScriptPath = storage\_path('app/python/FaceRecognition.py');  $encodingPath = storage\_path('app/models/encodings.txt');  $outputPath = storage\_path('app/data/output.txt');   *// Tạo lệnh để gọi script Python*  $command = escapeshellcmd("python $pythonScriptPath recognize\_faces $imagePath $encodingPath $outputPath");   *// Thực thi lệnh và đọc kết quả từ file output*  exec($command, $output, $return\_var);   **if** ($return\_var !== 0) {  **return** response()->json(['error' => 'Có lỗi xảy ra khi thực thi script Python.']);  }   *// Đọc kết quả từ file output*  **if** (file\_exists($outputPath)) {  $recognizedName = file\_get\_contents($outputPath);  **return** response()->json([  'recognizedName' => $recognizedName,  'imagePath' => $imagePath,  'encodingPath' => $encodingPath,  'outputPath' => $outputPath,  'pythonScriptPath' => $pythonScriptPath,  ]);  } **else** {  **return** response()->json(['error' => 'Không tìm thấy file kết quả.'], 404);  }  } **else** {  **return** response()->json(['error' => 'Không có dữ liệu imagePath được gửi lên.'], 400);  }  } } |
| --- |

## 4.6. Script nhận dạng python

### 4.6.1. Cài đặt thư viện Python cần thiết

Đầu tiên, cài đặt các thư viện cần thiết bằng lệnh pip:

| pip install face\_recognition opencv-python numpy |
| --- |

### 4.6.2. Tạo script Python

Tạo một tệp Python tên là FaceRecognition.py trong thư mục storage/Python với nội dung sau:

| **import** face\_recognition **import** sys  **def** **load\_encoding\_file**(file\_path):  names = []  encodings = []   **with** open(file\_path, 'r') **as** file:  lines = file.readlines()   current\_name = **None**  current\_encoding = []   **for** line **in** lines:  parts = line.strip().split(': ')   **if** len(parts) == 2:  **if** current\_name **is** **not** **None**:  names.append(current\_name)  encodings.append(current\_encoding)   current\_name = parts[0]  current\_encoding = [float(value) **for** value **in** parts[1][1:-1].replace(']', '').split()]  **elif** current\_name **is** **not** **None**:  current\_encoding.extend([float(value) **for** value **in** line.strip().replace(']', '').split()])   **if** current\_name **is** **not** **None**:  names.append(current\_name)  encodings.append(current\_encoding)   **return** names, encodings   **def** **recognize\_faces**(image\_path, names, encodings, output\_file):  image = face\_recognition.load\_image\_file(image\_path)  face\_locations = face\_recognition.face\_locations(image)  face\_encodings = face\_recognition.face\_encodings(image, face\_locations)   results = []  **for** (top, right, bottom, left), face\_encoding **in** zip(face\_locations, face\_encodings):  matches = face\_recognition.compare\_faces(encodings, face\_encoding, tolerance=0.3)  name = "Unknown"   **if** **True** **in** matches:  first\_match\_index = matches.index(**True**)  name = names[first\_match\_index]   results.append({"top": top, "right": right, "bottom": bottom, "left": left, "name": name})   **if** results:  **try**:  recognized\_name = results[0]['name']  print(recognized\_name)  **with** open(output\_file, 'w') **as** file:  file.write(recognized\_name)  **except** Exception **as** e:  print(f"Lỗi khi thực thi script Python: {e}")   **if** \_\_name\_\_ == "\_\_main\_\_":  **if** len(sys.argv) < 2:  print("Usage: python main.py command [options]")  sys.exit(1)   command = sys.argv[1]   **if** command == "recognize\_faces":  image\_path = sys.argv[2]  encodings\_path = sys.argv[3]  output\_file = sys.argv[4]  names, encodings = load\_encoding\_file(encodings\_path)  recognize\_faces(image\_path, names, encodings, output\_file)  **else**:  print("Unknown command")  sys.exit(1) |
| --- |

# 5. Kiểm tra

