**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**PROJECT I**

**Hệ thống nhận diện khuôn mặt**

**Hồ Viết Huy**

[Huy.hv225200@sis.hust.edu.vn](mailto:Huy.hv225200@sis.hust.edu.vn)

**Ngành Kỹ thuật máy tính**

**Giảng viên hướng dẫn:** Phạm Mạnh Tuấn

**Hà Nội, 10/2025**

**Lời cảm ơn**

Em xin chân thành cảm ơn thầy Phạm Mạnh Tuấn đã tận tình hướng dẫn, hỗ trợ và góp ý trong suốt quá trình thực hiện đồ án.  
Em cũng xin gửi lời cảm ơn đến các thầy cô trong trường công nghệ thông tin và truyền thông đã truyền đạt những kiến thức quý báu giúp em có nền tảng để hoàn thành đề tài này.

Cuối cùng, em xin cảm ơn gia đình và bạn bè đã luôn động viên, khích lệ trong suốt thời gian học tập và thực hiện đồ án.

**Em xin chân thành cảm ơn!**

**Tóm tắt nội dung**

Đồ án trình bày quá trình xây dựng hệ thống đăng ký, đăng nhập và phân tích khuôn mặt sử dụng Flask, DeepFace và MySQL.

Hệ thống cho phép người dùng đăng ký tài khoản kèm ảnh khuôn mặt, đăng nhập bằng xác thực khuôn mặt và phân tích đặc trưng khuôn mặt như tuổi, giới tính, cảm xúc và chủng tộc.

Các API được xây dựng trên nền tảng Flask giúp việc giao tiếp giữa frontend và backend dễ dàng, trong khi DeepFace đảm nhận vai trò xử lý và so khớp khuôn mặt bằng công nghệ Deep Learning.

Kết quả thử nghiệm cho thấy hệ thống hoạt động ổn định, nhận diện khuôn mặt với độ chính xác cao, góp phần nâng cao tính bảo mật và tự động hóa trong xác thực người dùng.

**Mục lục**

[**CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI** 4](#_Toc212174854)

[1.1. Lý do chọn đề tài 4](#_Toc212174855)

[1.2. Mục tiêu của đề tài 5](#_Toc212174856)

[Xây dựng API đăng ký người dùng: 5](#_Toc212174857)

[Phân tích khuôn mặt: 5](#_Toc212174858)

[Đảm bảo tính bảo mật và ổn định: 5](#_Toc212174859)

[1.3. Ngôn ngữ và các công cụ sử dụng 5](#_Toc212174860)

[**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT** 6](#_Toc212174861)

[2.1. Giới thiệu về Flask 6](#_Toc212174862)

[Đặc điểm nổi bật của Flask: 7](#_Toc212174863)

[2.2. Giới thiệu về DeepFace 7](#_Toc212174864)

[Các chức năng chính của DeepFace: 7](#_Toc212174865)

[Một số mô hình Deep Learning được hỗ trợ: 7](#_Toc212174866)

[2.3. Nhận diện khuôn mặt (Face Recognition) 8](#_Toc212174867)

[Quy trình nhận diện khuôn mặt cơ bản gồm 3 giai đoạn chính: 8](#_Toc212174868)

[Ứng dụng của công nghệ nhận diện khuôn mặt: 8](#_Toc212174869)

[**CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG** 9](#_Toc212174870)

[3.1. Mô tả chức năng hệ thống 9](#_Toc212174871)

[1. Chức năng Đăng ký người dùng 9](#_Toc212174872)

[2. Chức năng Đăng nhập 9](#_Toc212174873)

[3. Chức năng Phân tích khuôn mặt 9](#_Toc212174874)

[3.3. Sơ đồ cơ sở dữ liệu (Database Diagram) 10](#_Toc212174875)

[**CHƯƠNG 4. KẾT QUẢ THỰC NGHIỆM** 11](#_Toc212174876)

[4.1. Giao diện 11](#_Toc212174877)

[Giao diện đăng nhập 11](#_Toc212174878)

[Đăng ký 13](#_Toc212174879)

[Trích xuất đặc điểm khuôn mặt 14](#_Toc212174880)

[4.2. Kết quả thực nghiệm 14](#_Toc212174881)

[**CHƯƠNG 6. ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN** 15](#_Toc212174882)

[6.1. Ưu điểm 15](#_Toc212174883)

[Cấu trúc đơn giản, dễ triển khai: 15](#_Toc212174884)

[Tích hợp trí tuệ nhân tạo (AI) trực tiếp vào API: 15](#_Toc212174885)

[Tính bảo mật cao: 15](#_Toc212174886)

[Dễ dàng mở rộng và tích hợp: 15](#_Toc212174887)

[6.2. Hạn chế 15](#_Toc212174888)

[Hiệu năng xử lý còn hạn chế: 16](#_Toc212174889)

[Phụ thuộc phần cứng: 16](#_Toc212174890)

[Mật khẩu chưa được mã hóa: 16](#_Toc212174891)

[6.3. Hướng phát triển 16](#_Toc212174892)

[Tối ưu hiệu năng nhận diện: 16](#_Toc212174893)

[Triển khai trên môi trường điện toán đám mây: 16](#_Toc212174894)

[Phát triển thêm các tính năng nâng cao: 16](#_Toc212174895)

[**CHƯƠNG 7. KẾT LUẬN** 16](#_Toc212174896)

[**TÀI LIỆU KHAM KHẢO** 17](#_Toc212174897)

**Danh mục bảng**

[Table 1. Ngôn ngữ và các công cụ sử dụng 6](#_Toc212175047)

[Table 2. Mô tả bảng user 10](#_Toc212175048)

[Table 3. Kết quả thực nghiệm 15](#_Toc212175049)

**Mục lục hình ảnh**

[hình 1. Sơ đồ hoạt động nhận diện khuôn mặt 10](#_Toc212175055)

[hình 2. Giao diện đăng nhập 11](#_Toc212175056)

[hình 3. Giao diện đăng nhập thành công 12](#_Toc212175057)

[hình 4. Giao diện đăng nhập thất bại 12](#_Toc212175058)

[hình 5. Giao diện đăng ký 13](#_Toc212175059)

[hình 6. Giao diện đăng ký thành công 13](#_Toc212175060)

[hình 7. Giao diện trích xuất đặc điểm khuôn mặt 14](#_Toc212175061)

[hình 8. Kết quả đặc điểm khuôn mặt 14](#_Toc212175062)

# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

## 1.1. Lý do chọn đề tài

Trong bối cảnh công nghệ số phát triển mạnh mẽ, vấn đề bảo mật thông tin cá nhân và xác thực người dùng ngày càng trở nên quan trọng. Các phương thức đăng nhập truyền thống như nhập tên đăng nhập và mật khẩu dễ bị đánh cắp, lộ thông tin hoặc bị tấn công giả mạo. Do đó, việc nghiên cứu và ứng dụng công nghệ nhận diện khuôn mặt để xác thực người dùng đang dần trở thành xu hướng tất yếu trong các hệ thống bảo mật hiện đại.

Công nghệ Trí tuệ nhân tạo (AI), đặc biệt là Deep Learning, đã mở ra khả năng vượt trội trong việc nhận dạng khuôn mặt với độ chính xác cao, xử lý nhanh chóng và phù hợp cho nhiều ứng dụng như: kiểm soát ra vào, đăng nhập tài khoản, chấm công, hay giám sát an ninh.

Bên cạnh đó, việc sử dụng Flask – một micro-framework của Python – giúp xây dựng nhanh chóng hệ thống Web API nhẹ, linh hoạt và dễ mở rộng. Kết hợp với MySQL để quản lý dữ liệu người dùng và DeepFace để phân tích, xác minh khuôn mặt, hệ thống có thể thực hiện quy trình đăng ký và đăng nhập bảo mật hơn so với phương pháp truyền thống.

Từ những lý do trên, nhóm chúng em lựa chọn đề tài:

“Xây dựng hệ thống đăng ký, đăng nhập và phân tích khuôn mặt bằng DeepFace và Flask”

nhằm tìm hiểu, ứng dụng và đánh giá hiệu quả của công nghệ nhận diện khuôn mặt trong xác thực người dùng.

## 1.2. Mục tiêu của đề tài

Mục tiêu chính của đề tài là xây dựng một hệ thống API có khả năng đăng ký, đăng nhập và phân tích khuôn mặt người dùng dựa trên công nghệ AI. Cụ thể:

### Xây dựng API đăng ký người dùng:

Người dùng nhập username, password và gửi ảnh khuôn mặt. Dữ liệu được mã hóa (Base64) và lưu vào cơ sở dữ liệu MySQL.

Xây dựng API đăng nhập:

Khi đăng nhập, hệ thống xác thực tên đăng nhập, mật khẩu và so khớp khuôn mặt bằng thư viện DeepFace, đảm bảo người dùng là thật và chính chủ.

### Phân tích khuôn mặt:

Ứng dụng DeepFace để phân tích đặc trưng khuôn mặt, nhận dạng các thông tin như độ tuổi ước tính, giới tính, cảm xúc và chủng tộc của người dùng.

### Đảm bảo tính bảo mật và ổn định:

Hệ thống lưu trữ dữ liệu người dùng trong cơ sở dữ liệu MySQL và sử dụng token để quản lý phiên đăng nhập, giúp đảm bảo an toàn và dễ mở rộng.

Thông qua đề tài, nhóm mong muốn nắm vững quy trình xây dựng một ứng dụng web sử dụng AI, đồng thời hiểu rõ cơ chế hoạt động của hệ thống xác thực bằng khuôn mặt.

## 1.3. Ngôn ngữ và các công cụ sử dụng

Table . Ngôn ngữ và các công cụ sử dụng

|  |  |  |
| --- | --- | --- |
| **Thành phần** | **Công nghệ / Công cụ sử dụng** | **Mô tả** |
| Ngôn ngữ lập trình | Python 3 | Ngôn ngữ mạnh về trí tuệ nhân tạo, có nhiều thư viện hỗ trợ xử lý ảnh và học sâu. |
| Framework Backend | Flask | Dùng để xây dựng hệ thống Web API, nhẹ, dễ triển khai và mở rộng. |
| Cơ sở dữ liệu | MySQL | Lưu trữ thông tin người dùng, bao gồm tên đăng nhập, mật khẩu và ảnh khuôn mặt (dạng Base64). |
| Thư viện AI & Xử lý ảnh | DeepFace, OpenCV, NumPy | DeepFace dùng để nhận diện và phân tích khuôn mặt; OpenCV xử lý ảnh; NumPy hỗ trợ xử lý dữ liệu ma trận. |
| Frontend (Giao diện người dùng) | HTML, CSS, JavaScript | Xây dựng giao diện web thân thiện cho phép người dùng đăng ký, đăng nhập và gửi ảnh khuôn mặt tới API. |
| Công cụ kiểm thử API | Postman | Dùng để gửi yêu cầu và kiểm tra phản hồi API trong quá trình phát triển, giúp kiểm thử backend nhanh chóng. |
| Thư viện phụ trợ | Flask-CORS, mysql-connector-python, base64, secrets | Hỗ trợ kết nối CSDL, mã hóa dữ liệu hình ảnh, tạo token bảo mật và cho phép frontend truy cập API. |
| Công cụ phát triển | Visual Studio Code, XAMPP | VS Code dùng để lập trình và quản lý mã nguồn; XAMPP hỗ trợ chạy MySQL server và quản lý cơ sở dữ liệu. |

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

## 2.1. Giới thiệu về Flask

Flask là một micro web framework được viết bằng Python, dùng để xây dựng các ứng dụng web hoặc RESTful API một cách nhanh chóng và linh hoạt.

Flask được thiết kế với triết lý tối giản, chỉ cung cấp những thành phần cốt lõi như định tuyến (routing), xử lý request/response, và hỗ trợ template. Các chức năng nâng cao có thể được mở rộng thông qua các phần mở rộng (extension).

### Đặc điểm nổi bật của Flask:

1. Gọn nhẹ, dễ học và dễ triển khai.
2. Có khả năng mở rộng cao, tương thích với nhiều thư viện Python khác.
3. Hỗ trợ xây dựng hệ thống API giao tiếp với frontend thông qua giao thức HTTP.
4. Dễ dàng tích hợp với cơ sở dữ liệu và các thư viện AI hoặc Machine Learning.

Nhờ sự linh hoạt và hiệu năng tốt, Flask được sử dụng rộng rãi trong các ứng dụng web nhỏ và trung bình, đặc biệt trong các đề tài nghiên cứu, thử nghiệm và các hệ thống AI triển khai qua API.

## 2.2. Giới thiệu về DeepFace

DeepFace là một thư viện mã nguồn mở của Python, được phát triển để hỗ trợ nhận diện, xác minh và phân tích khuôn mặt bằng các mô hình học sâu (Deep Learning Models).

DeepFace được thiết kế như một giao diện thống nhất, cho phép tích hợp nhiều mô hình nhận diện khuôn mặt nổi tiếng mà không cần huấn luyện lại từ đầu.

### Các chức năng chính của DeepFace:

1. Nhận diện khuôn mặt (Face Recognition): Xác định xem hai khuôn mặt có phải là của cùng một người hay không.
2. Phân tích khuôn mặt (Face Analysis): Dự đoán các thông tin như tuổi, giới tính, cảm xúc và chủng tộc.
3. Xác minh danh tính (Face Verification): So sánh đặc trưng khuôn mặt giữa hai hình ảnh để xác định tính trùng khớp.

### Một số mô hình Deep Learning được hỗ trợ:

1. VGG-Face: Mô hình do Đại học Oxford phát triển, hiệu quả cao trong trích xuất đặc trưng khuôn mặt.
2. Facenet: Mô hình của Google, biểu diễn khuôn mặt dưới dạng vector đặc trưng có độ chính xác cao.
3. OpenFace: Mô hình nhẹ, tối ưu cho ứng dụng thời gian thực.
4. DeepID, ArcFace, Dlib: Các mô hình khác được tích hợp sẵn, giúp tăng độ linh hoạt trong việc lựa chọn phương pháp nhận diện.

DeepFace hoạt động dựa trên việc trích xuất vector đặc trưng khuôn mặt (embedding) bằng mạng nơ-ron sâu, sau đó so sánh các vector này để xác định mức độ tương đồng giữa các khuôn mặt. Thư viện này được xem là công cụ mạnh mẽ và dễ sử dụng trong các ứng dụng liên quan đến nhận diện khuôn mặt.

## 2.3. Nhận diện khuôn mặt (Face Recognition)

Nhận diện khuôn mặt là quá trình sử dụng công nghệ máy tính để tự động xác định hoặc xác minh danh tính của một người dựa trên hình ảnh hoặc video chứa khuôn mặt.

Đây là một trong những ứng dụng tiêu biểu của thị giác máy tính (Computer Vision) và trí tuệ nhân tạo (Artificial Intelligence).

### Quy trình nhận diện khuôn mặt cơ bản gồm 3 giai đoạn chính:

1. Phát hiện khuôn mặt (Face Detection):

Xác định vị trí của khuôn mặt trong hình ảnh hoặc khung hình video. Giai đoạn này sử dụng các thuật toán học máy hoặc học sâu như Haar Cascade, MTCNN, hoặc RetinaFace.

1. Trích xuất đặc trưng khuôn mặt (Feature Extraction):

Sau khi phát hiện, khuôn mặt được chuyển thành một vector số chứa các đặc trưng đặc biệt giúp phân biệt giữa các cá nhân khác nhau. Quá trình này sử dụng các mô hình mạng nơ-ron sâu được huấn luyện sẵn.

1. So sánh và xác minh khuôn mặt (Face Matching / Verification):

Hai vector đặc trưng được so sánh bằng các phương pháp đo độ tương đồng như khoảng cách Euclidean hoặc cosine similarity. Nếu mức sai khác nhỏ hơn ngưỡng cho phép, hệ thống kết luận đó là cùng một người.

### Ứng dụng của công nghệ nhận diện khuôn mặt:

1. Xác thực đăng nhập người dùng (Face Login).
2. Kiểm soát ra vào và chấm công tự động.
3. Hệ thống giám sát an ninh.
4. Phân tích hành vi và cảm xúc trong marketing hoặc tương tác người dùng.

Nhận diện khuôn mặt hiện nay là một trong những công nghệ quan trọng của thời đại 4.0, kết hợp giữa AI, Machine Learning và xử lý ảnh, được ứng dụng rộng rãi trong các lĩnh vực an ninh, giáo dục, y tế và thương mại điện tử.

# CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG

## 3.1. Mô tả chức năng hệ thống

Hệ thống được thiết kế gồm ba chức năng chính: Đăng ký người dùng, Đăng nhập xác thực khuôn mặt và Phân tích khuôn mặt.

Các chức năng này được triển khai thông qua các API trong Flask, sử dụng cơ sở dữ liệu MySQL để lưu trữ thông tin và thư viện DeepFace để xử lý, nhận diện khuôn mặt.

### 1. Chức năng Đăng ký người dùng

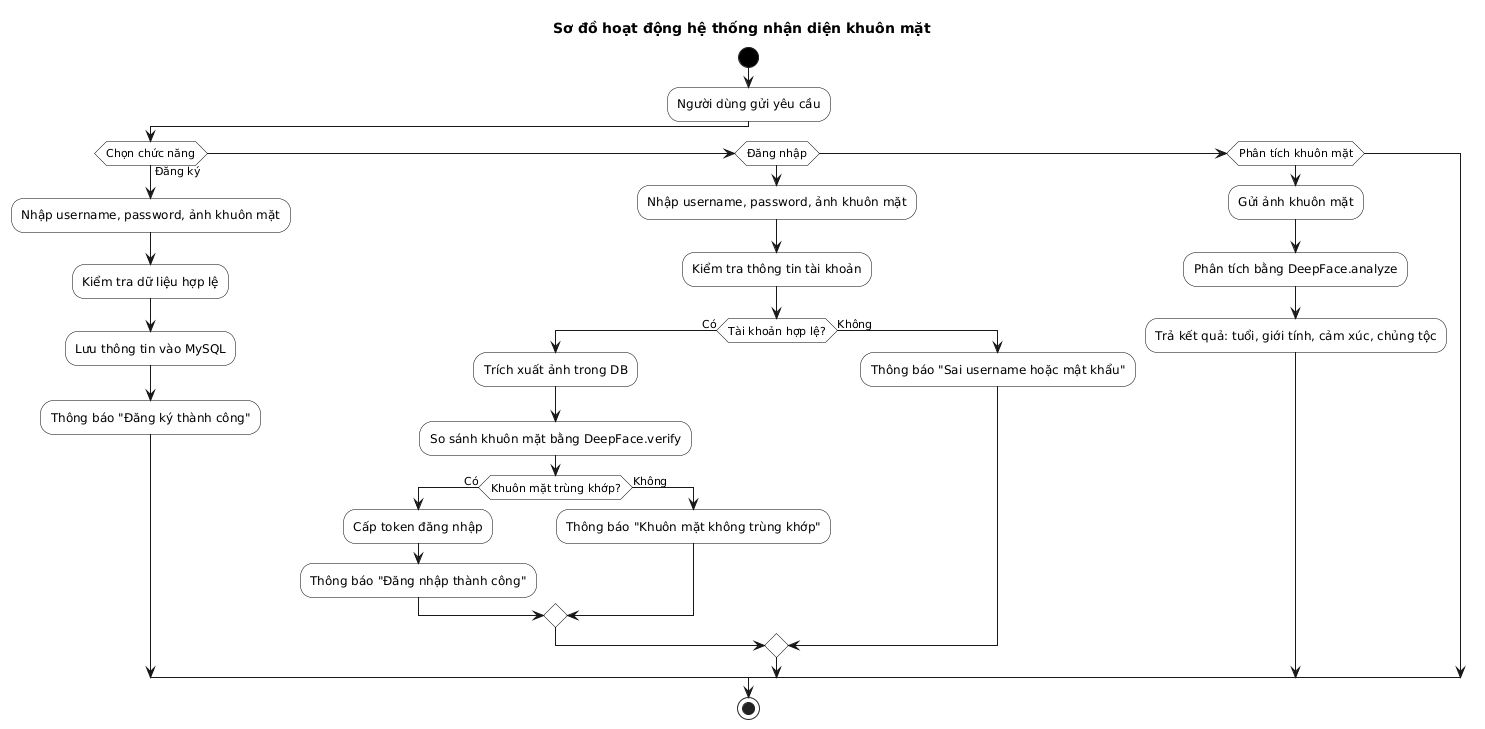
Người dùng cung cấp username, password và ảnh khuôn mặt (dạng Base64). Hệ thống kiểm tra dữ liệu đầu vào, giải mã ảnh và lưu thông tin người dùng vào CSDL MySQL. Ảnh khuôn mặt được lưu dưới dạng chuỗi Base64 để tiện truy xuất và so sánh sau này. Trường hợp thiếu thông tin hoặc ảnh không hợp lệ, hệ thống trả về lỗi tương ứng.

### 2. Chức năng Đăng nhập

Người dùng nhập username, password và gửi ảnh khuôn mặt hiện tại. Hệ thống kiểm tra thông tin đăng nhập trong CSDL: Nếu username/password sai → báo lỗi. Nếu đúng → hệ thống so sánh ảnh khuôn mặt mới với ảnh đã lưu trong CSDL bằng DeepFace.verify(). Nếu khuôn mặt trùng khớp → đăng nhập thành công, trả về token bảo mật cho người dùng. Nếu khuôn mặt không trùng → báo lỗi “Khuôn mặt không khớp”.

### 3. Chức năng Phân tích khuôn mặt

Người dùng gửi ảnh khuôn mặt qua API. Hệ thống sử dụng DeepFace.analyze() để trích xuất thông tin gồm:Tuổi (Age), Giới tính (Gender), Cảm xúc (Emotion), Chủng tộc (Race)…. Kết quả trả về dưới dạng JSON, giúp frontend hiển thị thông tin trực quan cho người dùng.

****

hình . Sơ đồ hoạt động nhận diện khuôn mặt

## ****3.3. Sơ đồ cơ sở dữ liệu (Database Diagram)****

Hệ thống chỉ sử dụng **một bảng chính user**, lưu trữ thông tin tài khoản và ảnh khuôn mặt của người dùng.  
Cấu trúc bảng được thiết kế đơn giản, dễ mở rộng trong tương lai (có thể thêm email, ngày đăng ký, hoặc vai trò người dùng).

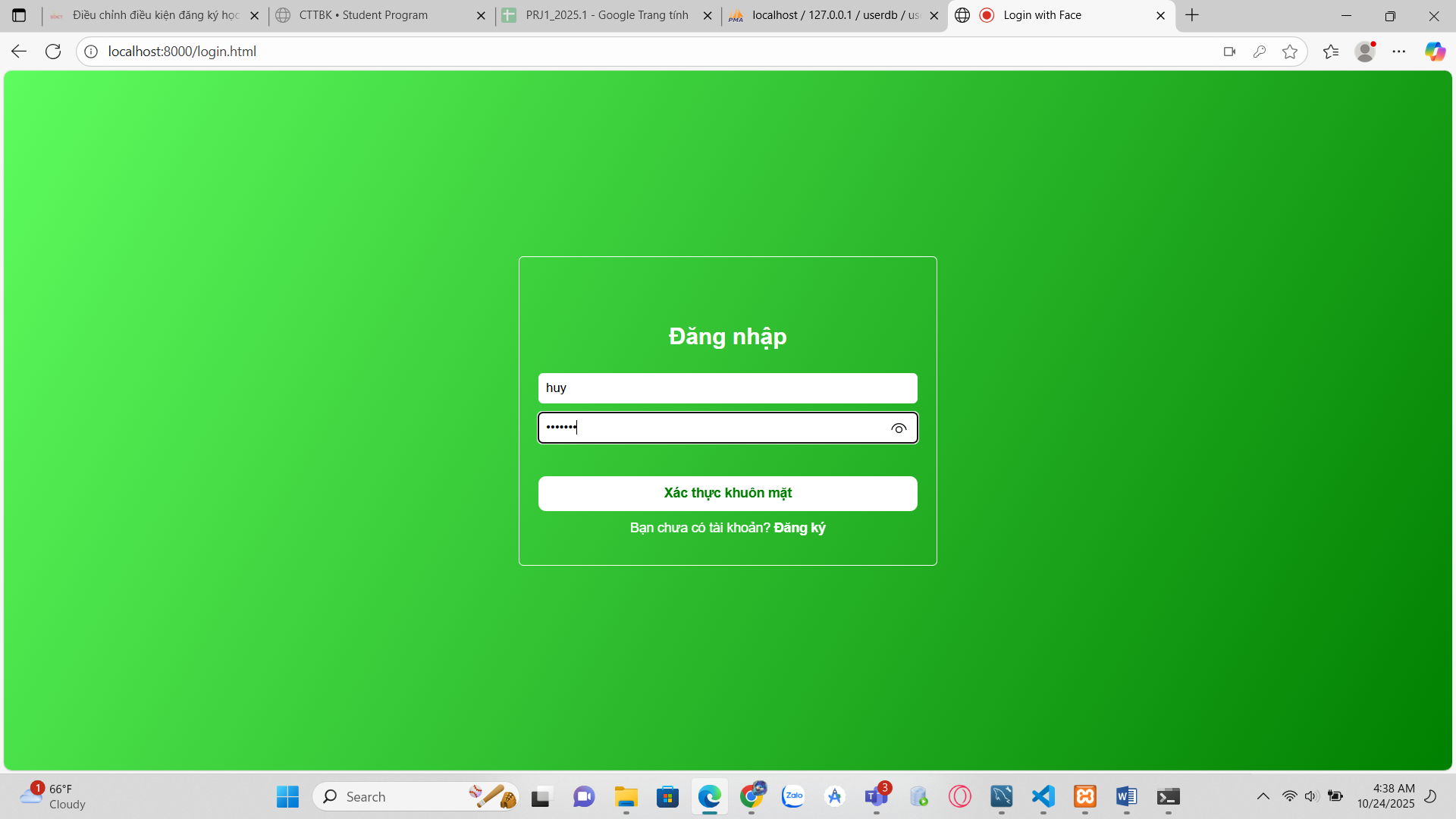
Table . Mô tả bảng user

|  |  |  |
| --- | --- | --- |
| **Cột** | **Kiểu dữ liệu** | **Mô tả** |
| ID | INT (Primary Key, Auto Increment) | Mã định danh duy nhất của người dùng. |
| userName | VARCHAR(100) | Tên đăng nhập của người dùng. |
| password | VARCHAR(255) | Mật khẩu người dùng (lưu ở dạng mã hóa). |
| img | LONGTEXT | Ảnh khuôn mặt của người dùng được mã hóa Base64. |

# CHƯƠNG 4. KẾT QUẢ THỰC NGHIỆM

## 4.1. Giao diện

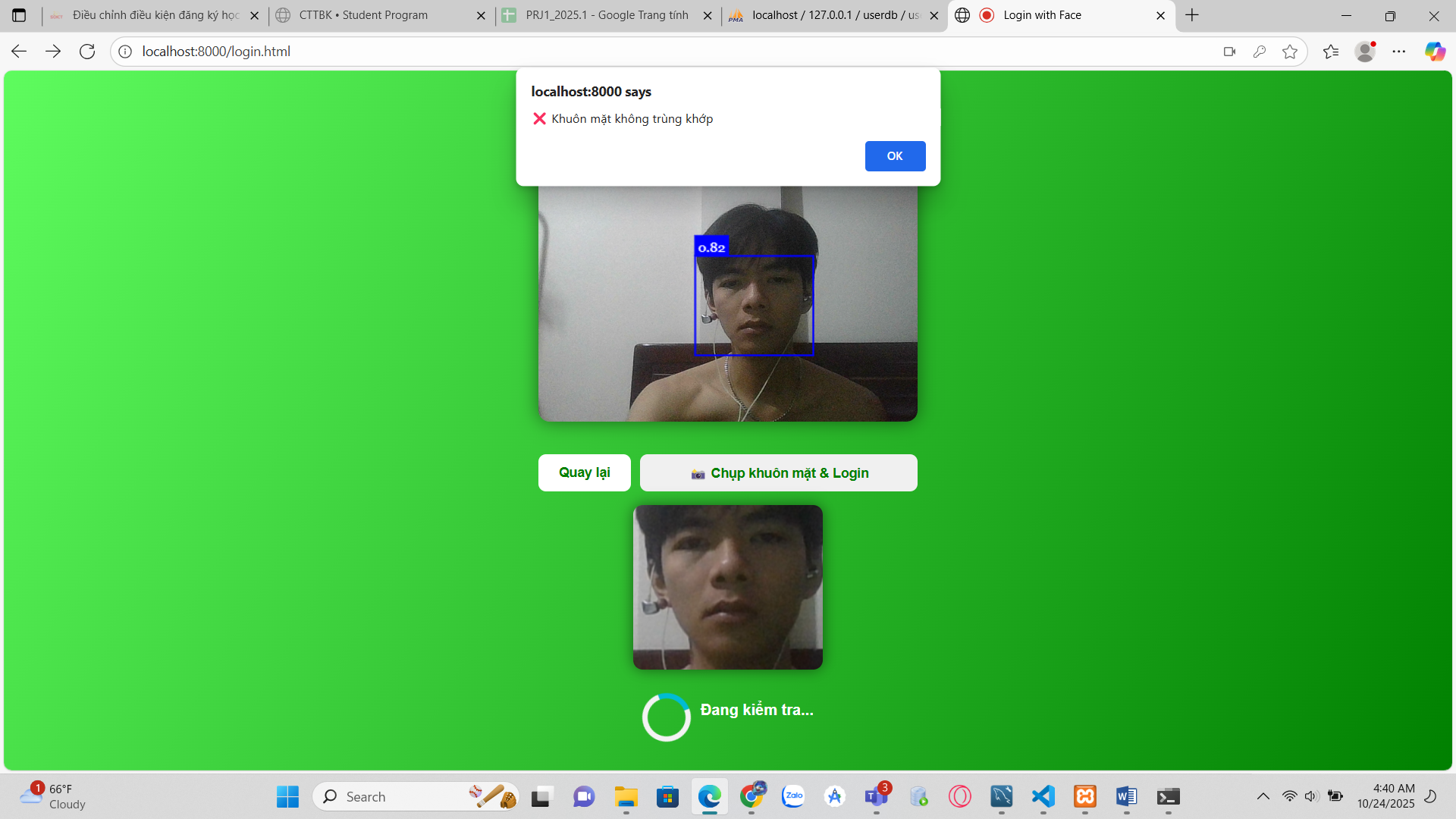
### Giao diện đăng nhập



hình . Giao diện đăng nhập

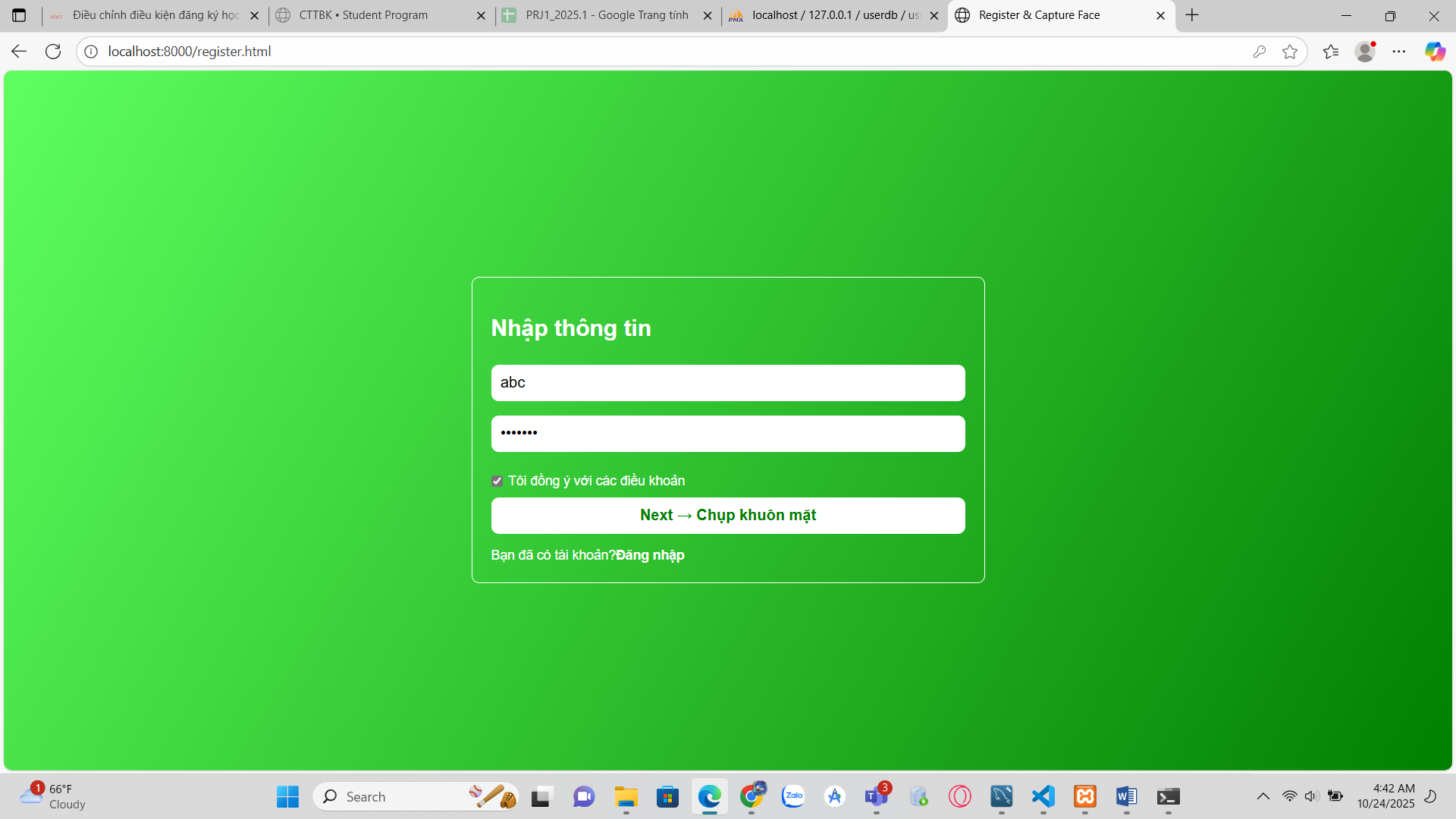


hình . Giao diện đăng nhập thành công

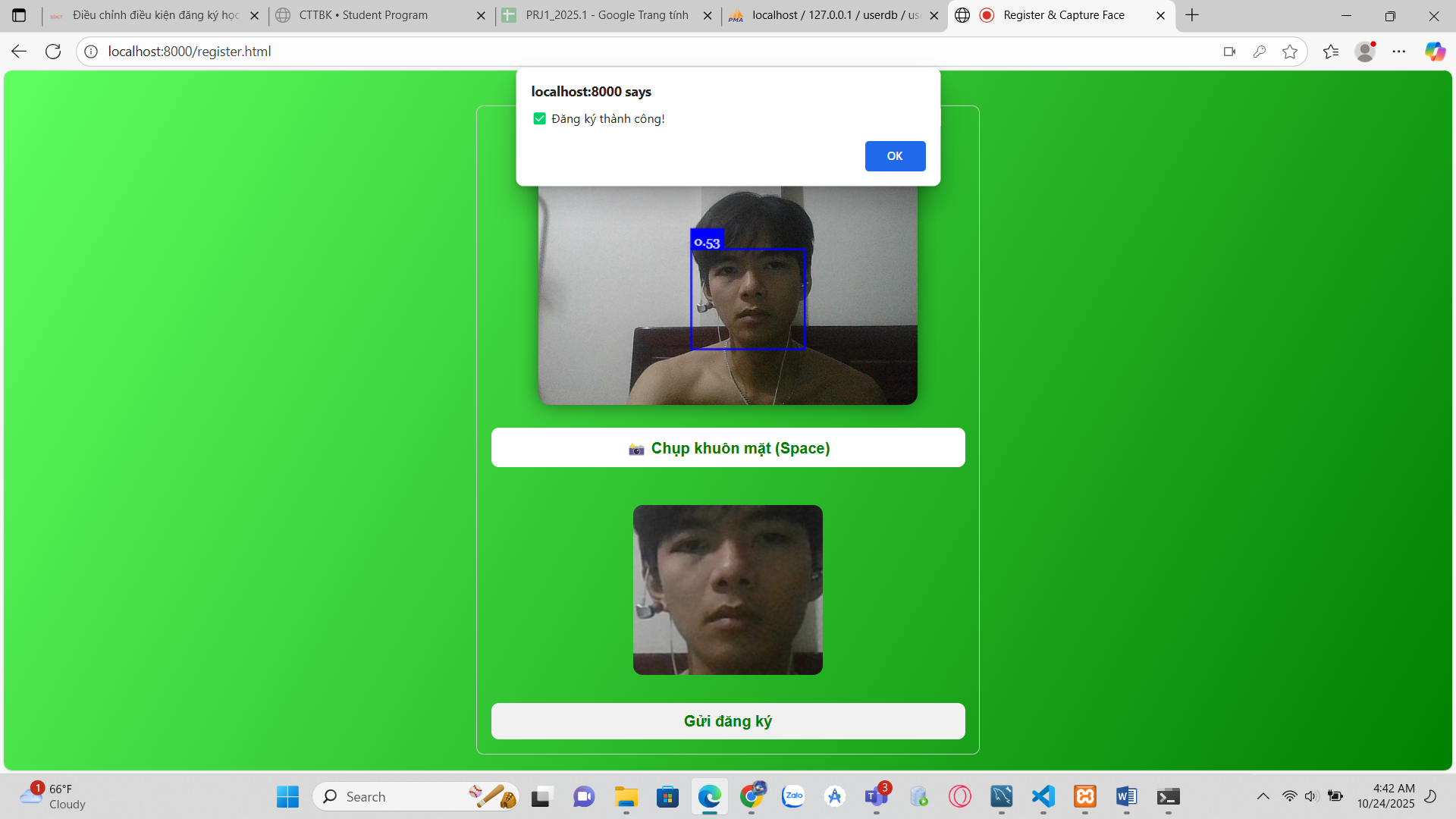


hình . Giao diện đăng nhập thất bại

### Đăng ký

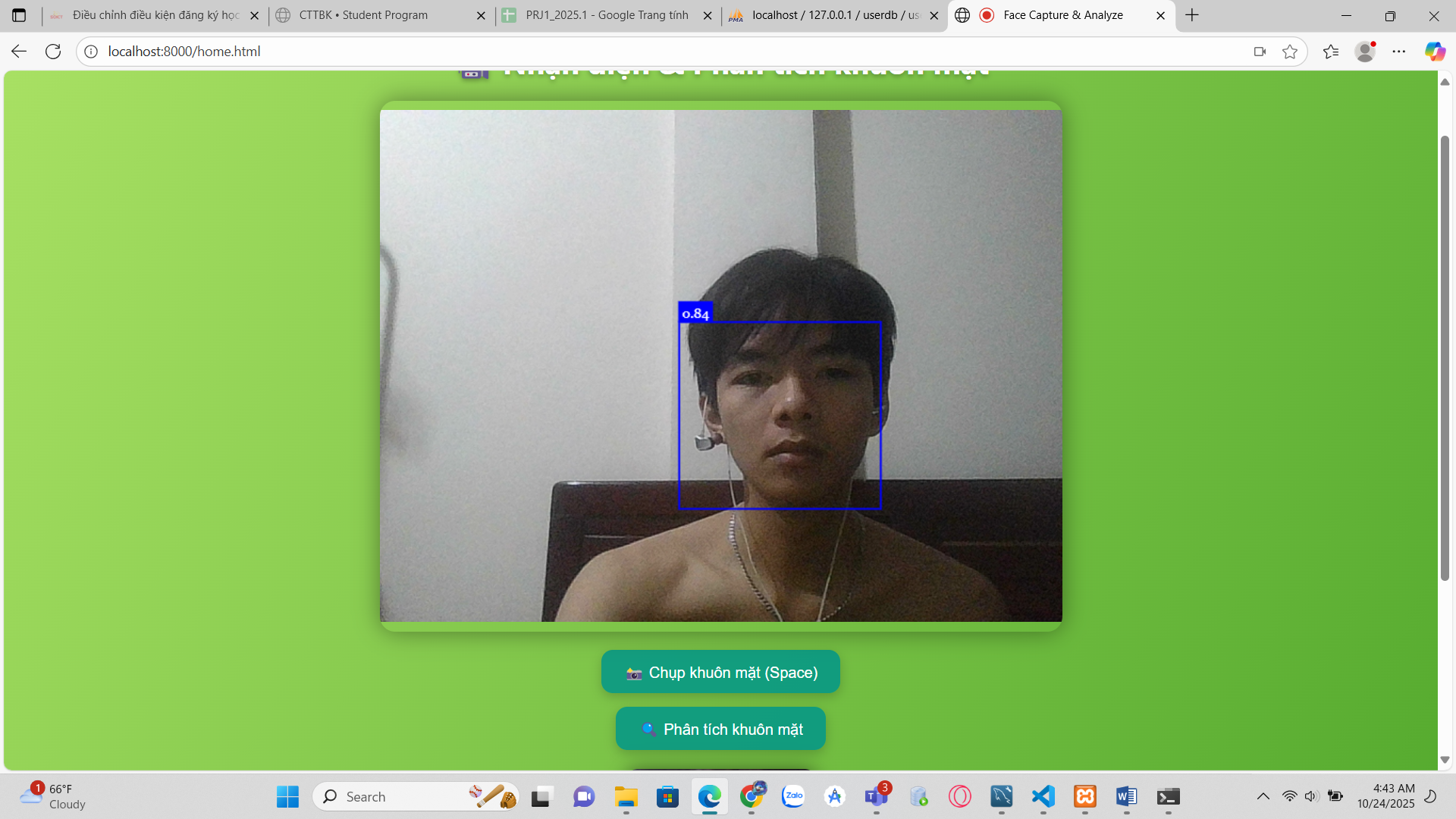


hình . Giao diện đăng ký

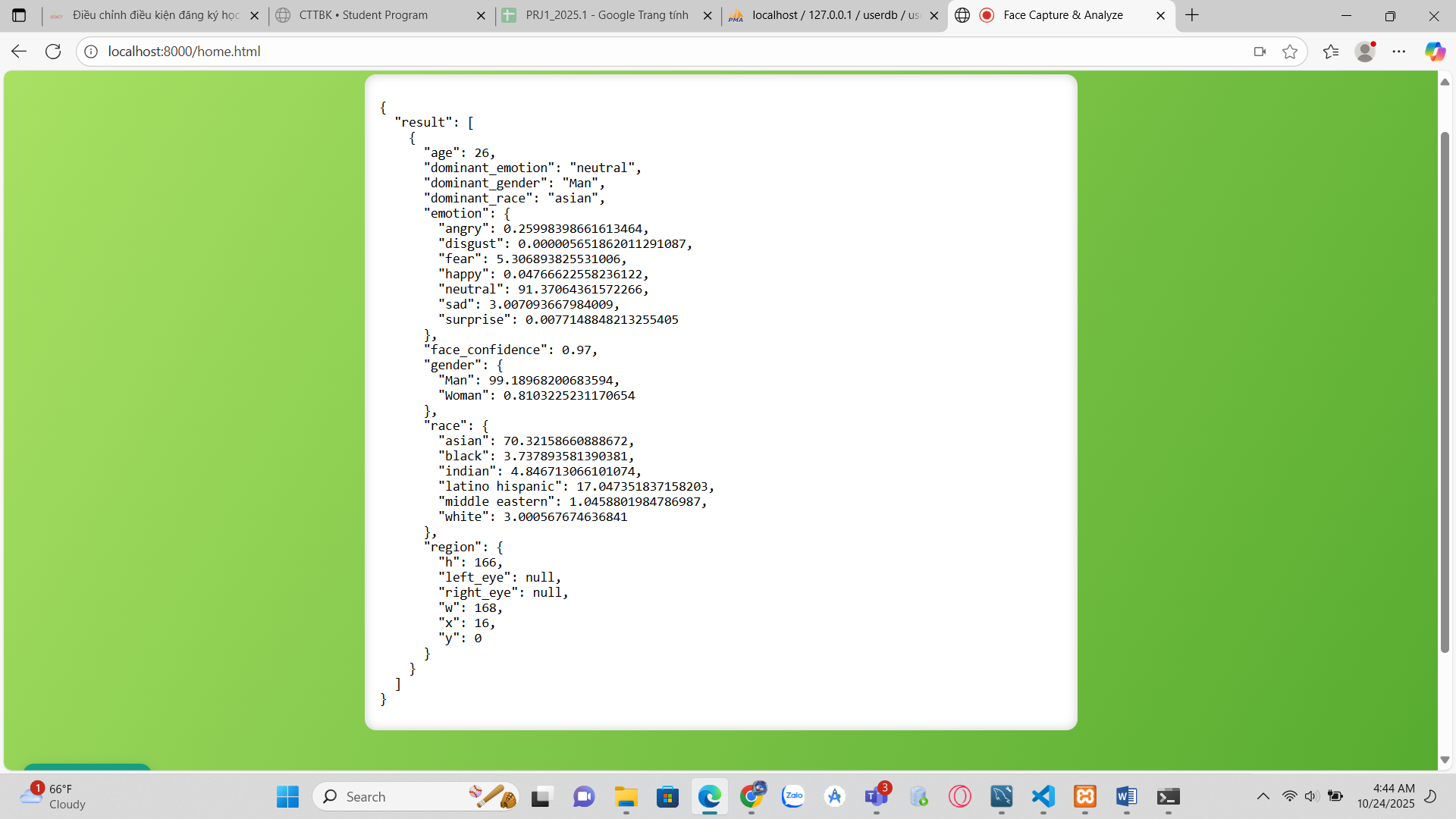


hình . Giao diện đăng ký thành công

### Trích xuất đặc điểm khuôn mặt



hình . Giao diện trích xuất đặc điểm khuôn mặt



hình . Kết quả đặc điểm khuôn mặt

## 4.2. Kết quả thực nghiệm

Table . Kết quả thực nghiệm

|  |  |  |
| --- | --- | --- |
| **Thử nghiệm** | **Mô tả** | **Kết quả** |
| Đăng ký | Lưu user mới thành công | Thành công |
| Đăng nhập đúng ảnh | Nhận diện chính xác | Thành công |
| Đăng nhập sai ảnh | Bị từ chối | Không thành công |
| Phân tích khuôn mặt | Hiển thị tuổi, giới tính, cảm xúc,… | Thành công |

# CHƯƠNG 6. ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN

## 6.1. Ưu điểm

Hệ thống nhận diện khuôn mặt sử dụng Flask và DeepFace có nhiều ưu điểm nổi bật, thể hiện qua các khía cạnh sau:

### Cấu trúc đơn giản, dễ triển khai:

Việc sử dụng Flask – một micro-framework nhẹ, giúp quá trình phát triển API nhanh chóng, dễ bảo trì và mở rộng.

### Tích hợp trí tuệ nhân tạo (AI) trực tiếp vào API:

DeepFace mang lại khả năng nhận diện và phân tích khuôn mặt chính xác, giúp hệ thống đáp ứng tốt các bài toán xác thực người dùng.

### Tính bảo mật cao:

Phương thức xác thực bằng khuôn mặt giúp giảm thiểu nguy cơ bị đánh cắp hoặc đoán mật khẩu, tăng cường an toàn cho người dùng.

### Dễ dàng mở rộng và tích hợp:

Hệ thống có thể được kết nối với các ứng dụng khác như website, phần mềm điểm danh, hệ thống quản lý nhân sự hoặc kiểm soát ra vào.

## 6.2. Hạn chế

Bên cạnh những ưu điểm, hệ thống vẫn còn tồn tại một số hạn chế cần khắc phục trong các phiên bản tiếp theo:

### Hiệu năng xử lý còn hạn chế:

Khi xử lý ảnh có kích thước lớn hoặc nhiều người dùng truy cập cùng lúc, thời gian phản hồi có thể bị chậm.

### Phụ thuộc phần cứng:

DeepFace hoạt động tối ưu trên thiết bị có GPU. Nếu chỉ sử dụng CPU, tốc độ nhận diện khuôn mặt sẽ giảm đáng kể.

### Mật khẩu chưa được mã hóa:

Hệ thống hiện lưu trữ mật khẩu dưới dạng văn bản thuần (plain text), tiềm ẩn rủi ro bảo mật nếu cơ sở dữ liệu bị xâm nhập.

## 6.3. Hướng phát triển

Để hoàn thiện và nâng cao hiệu quả sử dụng, hệ thống có thể được mở rộng và cải tiến theo các hướng sau:

### Tối ưu hiệu năng nhận diện:

Áp dụng GPU hoặc các mô hình nhẹ hơn như MobileFaceNet để cải thiện tốc độ xử lý ảnh và giảm tải cho hệ thống.

### Triển khai trên môi trường điện toán đám mây:

Đưa hệ thống lên các nền tảng như AWS, Google Cloud hoặc Heroku để đảm bảo khả năng mở rộng và tính sẵn sàng cao.

### Phát triển thêm các tính năng nâng cao:

Bổ sung khả năng nhận diện nhiều khuôn mặt cùng lúc, theo dõi khuôn mặt trong video, hoặc cải thiện độ chính xác bằng các mô hình học sâu mới.

# CHƯƠNG 7. KẾT LUẬN

Trong phạm vi đề tài này, tôi đã xây dựng thành công hệ thống đăng ký, đăng nhập và phân tích khuôn mặt dựa trên các công nghệ Flask, DeepFace và MySQL, kết hợp với giao diện người dùng (frontend) được phát triển bằng HTML, CSS và JavaScript.

Hệ thống cho phép người dùng:

Đăng ký tài khoản kèm hình ảnh khuôn mặt.

Đăng nhập bằng cách xác thực khuôn mặt kết hợp với thông tin tài khoản.

Phân tích khuôn mặt để nhận biết các thông tin như tuổi, giới tính, cảm xúc và chủng tộc.

Kết quả thử nghiệm cho thấy hệ thống hoạt động ổn định, khả năng nhận diện và xác minh khuôn mặt đạt độ chính xác cao, tốc độ xử lý nhanh trong môi trường thử nghiệm.

Việc tích hợp DeepFace với Flask giúp thực hiện việc nhận diện một cách hiệu quả, trong khi MySQL đảm bảo khả năng lưu trữ và quản lý dữ liệu người dùng.

Phần giao diện web được thiết kế đơn giản, thân thiện, giúp người dùng dễ dàng tương tác với API.

Đề tài đã thể hiện được ứng dụng thực tiễn của trí tuệ nhân tạo (AI) trong lĩnh vực bảo mật và xác thực danh tính, góp phần khẳng định tiềm năng của các mô hình học sâu trong đời sống hiện nay.

Trong tương lai, tôi dự định sẽ tiếp tục nâng cấp hệ thống theo hướng:

Cải thiện tốc độ xử lý thông qua GPU.

Áp dụng mã hóa mật khẩu nhằm tăng tính bảo mật.

Phát triển thêm ứng dụng di động để mở rộng phạm vi sử dụng.

Tổng kết lại, đề tài đã hoàn thành các mục tiêu đặt ra, góp phần củng cố kiến thức của bản thân về lập trình web, xử lý ảnh và ứng dụng AI, đồng thời tạo tiền đề cho các nghiên cứu và phát triển sâu hơn trong tương lai.

# TÀI LIỆU KHAM KHẢO