

ĐẠI HỌC BÁCH KHOA HÀ NỘI
LỚP B2CQ-CNTT03-K68

ỨNG DỤNG AI AVATAR THEO CẢM XÚC

Người thực hiện: Hoàng Kiên Quyết

Hà Nội 2024

ĐẠI HỌC BÁCH KHOA HÀ NỘI
LỚP B2CQ-CNTT03-K68

ỨNG DỤNG AI AVATAR THEO CẢM XÚC

Người thực hiện: Hoàng Kiên Quyết

NGƯỜI THỰC HIỆN

Hoàng Kiên Quyết

MỤC LỤC

NỘI DUNG BÁO CÁO	1
I. TỔNG QUAN ỨNG DỤNG	1
1.1. Giới thiệu ứng dụng	1
1.2. Mục tiêu ứng dụng	1
1.3. Phạm vi của ứng dụng.....	1
1.4. Yêu cầu đặt ra	1
II. THIẾT KẾ VÀ XÂY DỰNG ỨNG DỤNG	2
2.1. Công nghệ sử dụng	2
2.2. Cấu trúc ứng dụng.....	4
2.3. Quá trình xây dựng phần mềm.....	5
III. THỬ NGHIỆM VÀ ĐÁNH GIÁ.....	6
TÀI LIỆU THAM KHẢO	8

MỤC LỤC HÌNH ẢNH

Hình ảnh 1: Sơ đồ cấu trúc ứng dụng	4
Hình ảnh 2: Camera Component.....	5
Hình ảnh 3: Giao diện ứng dụng ở chế độ phát triển.....	6

NỘI DUNG BÁO CÁO

I. TỔNG QUAN ỨNG DỤNG

1.1. Giới thiệu ứng dụng

Tên ứng dụng: Avatar theo cảm xúc

Phiên bản: 1.0.0

Mô tả: Ứng dụng cho phép tạo avatar hoạt họa cho người dùng bằng camera trong thời gian thực.

1.2. Mục tiêu ứng dụng

Mục tiêu của ứng dụng là sử dụng AI để nhận diện và biểu đạt cảm xúc của người dùng thông qua một avatar. Ứng dụng dựa trên mô hình nhận diện cảm xúc của thư viện face-api.js, một công cụ mã nguồn mở dành cho nhận diện khuôn mặt và phân tích cảm xúc. Khi người dùng thể hiện cảm xúc trên khuôn mặt, AI sẽ phân tích biểu cảm và phản hồi bằng cách thay đổi biểu cảm của avatar sao cho phù hợp. Ứng dụng cho phép những người hướng nội, ngại giao tiếp có thể thể hiện cảm xúc của mình thông qua avatar.

1.3. Phạm vi của ứng dụng

Ứng dụng nghiên cứu trong phạm vi thử nghiệm tính năng của AI cụ thể là model nhận diện cảm xúc có sẵn của thư viện face-api.js.

Giao diện được xây dựng dựa trên thư viện ReactJs.

Triển khai: Ứng dụng có thể triển khai trên bất kỳ máy chủ web, hosting nào. Phù hợp cho hầu hết các trình duyệt như: chrome, firefox, safari,...

1.4. Yêu cầu đặt ra

Ứng dụng phải có khả năng tương tác trong thời gian thực, có độ chính xác cao trên 85%.

II. THIẾT KẾ VÀ XÂY DỰNG ỨNG DỤNG

2.1. Công nghệ sử dụng

2.1.1. Các công nghệ sử dụng

Model AI được lấy từ thư viện Face API.js. Face API.js là một thư viện JavaScript mã nguồn mở, cung cấp các công cụ để nhận diện và phân tích khuôn mặt trong trình duyệt. Được xây dựng trên nền tảng TensorFlow.js, Face API.js hỗ trợ nhận diện khuôn mặt, phát hiện đặc điểm trên khuôn mặt, ước lượng cảm xúc và nhận diện danh tính. Thư viện này phù hợp cho các ứng dụng web tương tác, vì nó hoạt động trực tiếp trên trình duyệt mà không cần gửi dữ liệu đến máy chủ, giúp bảo vệ quyền riêng tư và tăng tốc độ xử lý.

Avatar được tạo ra dựa trên Dice Avatar: Là một công cụ tạo avatar ngẫu nhiên dựa trên hình ảnh xúc xắc (dice). Nó sử dụng các hình dạng và màu sắc khác nhau để tạo ra các avatar độc đáo, thường có phong cách đơn giản và dễ nhận diện. Dice Avatar thường được dùng trong các ứng dụng, trò chơi hoặc diễn đàn để gán avatar tạm thời cho người dùng một cách ngẫu nhiên, giúp tạo ra sự đa dạng mà không cần người dùng tự thiết kế hoặc tải lên hình ảnh cá nhân.

Xây dựng giao diện dựa trên React JS: Face API.js là một thư viện JavaScript mã nguồn mở, cung cấp các công cụ để nhận diện và phân tích khuôn mặt trong trình duyệt. Được xây dựng trên nền tảng TensorFlow.js, Face API.js hỗ trợ nhận diện khuôn mặt, phát hiện đặc điểm trên khuôn mặt, ước lượng cảm xúc và nhận diện danh tính. Thư viện này phù hợp cho các ứng dụng web tương tác, vì nó hoạt động trực tiếp trên trình duyệt mà không cần gửi dữ liệu đến máy chủ, giúp bảo vệ quyền riêng tư và tăng tốc độ xử lý.

Ngôn ngữ lập trình xử dụng Typescript: TypeScript là một ngôn ngữ lập trình mã nguồn mở, được phát triển bởi Microsoft, là một superset của JavaScript. TypeScript bổ sung hệ thống kiểu tĩnh (static typing) và các tính năng hiện đại cho JavaScript, giúp phát hiện lỗi sớm và cải thiện khả năng bảo

trình mã. Các tệp TypeScript (.ts) khi biên dịch sẽ chuyển thành mã JavaScript thuần để chạy trên mọi trình duyệt hay môi trường hỗ trợ JavaScript.

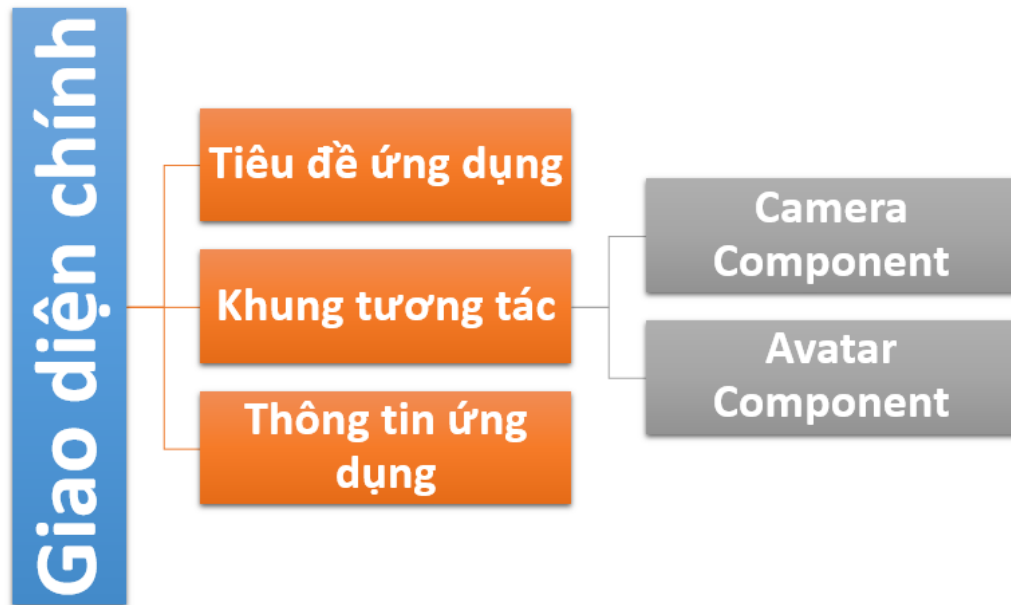
2.1.2. Quy trình phát triển và công cụ hỗ trợ

Xây dựng các component CameraComponent và AvatarComponent sau đó tích hợp vào App component để hoàn thiện ứng dụng.

Công cụ lưu trữ mã nguồn Github: Là một nền tảng lưu trữ mã nguồn và quản lý phiên bản dựa trên Git, cho phép các lập trình viên lưu trữ, chia sẻ và hợp tác phát triển phần mềm. Ngoài ra, GitHub cung cấp các công cụ để theo dõi thay đổi, quản lý dự án, và hỗ trợ làm việc nhóm, giúp dễ dàng quản lý và phát triển các dự án mã nguồn mở cũng như các dự án cá nhân.

Công cụ xây dựng mã nguồn Visual Studio Code: Là một trình soạn thảo mã nguồn mở, miễn phí, được phát triển bởi Microsoft. VS Code hỗ trợ nhiều ngôn ngữ lập trình, tích hợp Git, và có kho tiện ích mở rộng phong phú, giúp lập trình viên tùy chỉnh và mở rộng chức năng. Nhờ giao diện thân thiện và các tính năng mạnh mẽ như IntelliSense, debug và tích hợp terminal, VS Code trở thành công cụ phổ biến cho lập trình viên trên nhiều nền tảng (Windows, macOS, Linux).

2.2. Cấu trúc ứng dụng



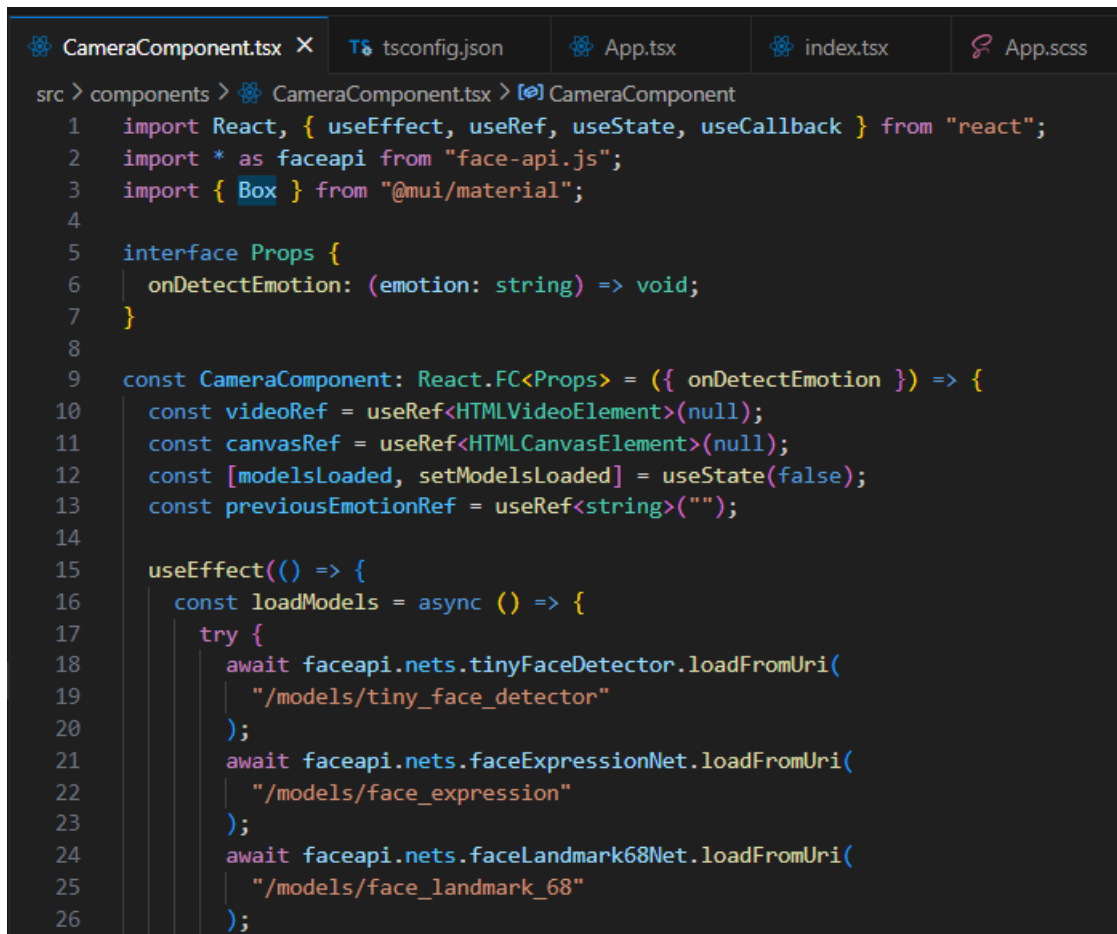
Hình ảnh 1: Sơ đồ cấu trúc ứng dụng

Ứng dụng được chia thành các thành phần chính sau:

- **Tiêu đề của ứng dụng:** Bao gồm tiêu đề và mô tả ngắn về ứng dụng.
- **Khung tương tác:** Bao gồm Camera Component và Avatar Component. Camera Component có nhiệm vụ lấy hình ảnh người dùng và nhận diện cảm xúc trong thời gian thực. Avatar Component có nhiệm vụ tạo Avatar theo cảm xúc đã nhận diện.
- **Thông tin ứng dụng:** Vị trí tại chân trang ghi thông tin của ứng dụng và người thực hiện.

2.3. Quá trình xây dựng phần mềm

2.3.1. Xây dựng Camera Component



```
src > components > CameraComponent.tsx > CameraComponent
1  import React, { useEffect, useRef, useState, useCallback } from "react";
2  import * as faceapi from "face-api.js";
3  import { Box } from "@mui/material";
4
5  interface Props {
6    onDetectEmotion: (emotion: string) => void;
7  }
8
9  const CameraComponent: React.FC<Props> = ({ onDetectEmotion }) => {
10    const videoRef = useRef<HTMLVideoElement>(null);
11    const canvasRef = useRef<HTMLCanvasElement>(null);
12    const [modelsLoaded, setModelsLoaded] = useState(false);
13    const previousEmotionRef = useRef<string>("");
14
15    useEffect(() => {
16      const loadModels = async () => {
17        try {
18          await faceapi.nets.tinyFaceDetector.loadFromUri(
19            "/models/tiny_face_detector"
20          );
21          await faceapi.nets.faceExpressionNet.loadFromUri(
22            "/models/face_expression"
23          );
24          await faceapi.nets.faceLandmark68Net.loadFromUri(
25            "/models/face_landmark_68"
26          );
```

Hình ảnh 2: Camera Component

Đầu tiên tải các model cần thiết vào thư mục public/models sau đó load các model có tên: tiny_face_detector, face_landmark_68 và face_landmark_68.

Tạo phần tử HTML tiếp nhận nguồn video đầu vào, xử lý bằng model AI để nhận diện cảm xúc và vẽ lại sơ đồ khuôn mặt theo thời gian thực bằng canvas. Đồng thời Camera Component nhận tham số đầu vào là hàm onDetectEmotion để truyền cảm xúc nhận diện được sang Avatar Component.

2.3.2. Xây dựng Avatar Component

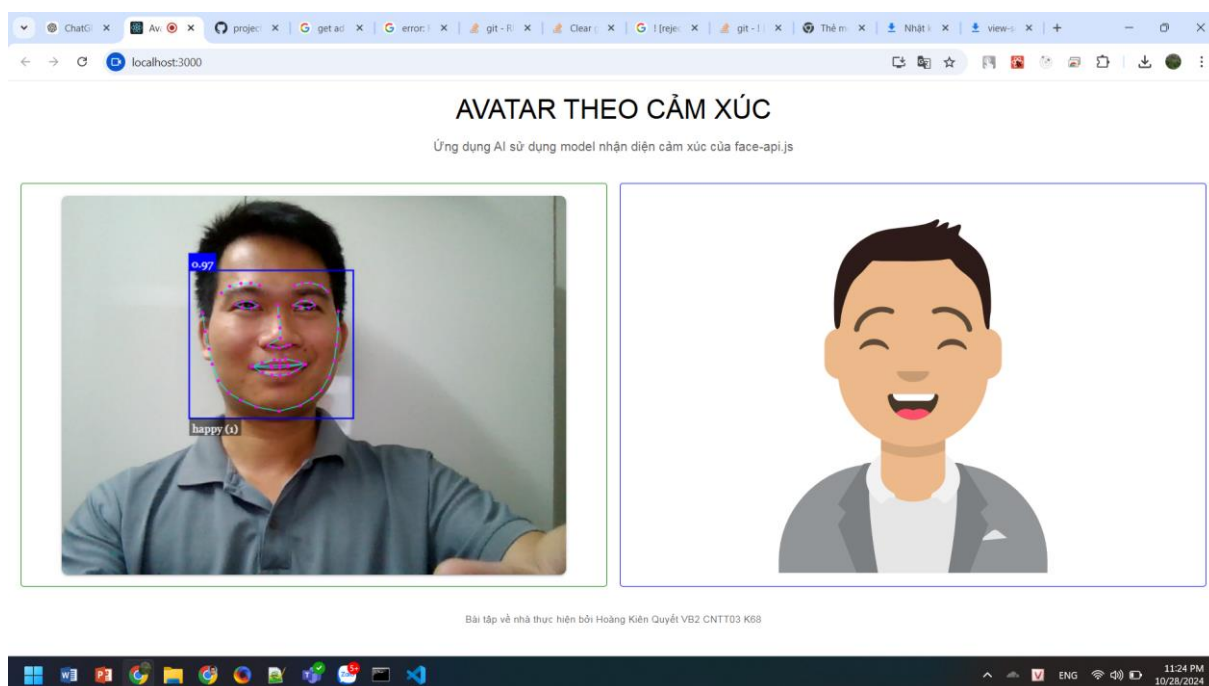
Avatar Component sẽ nhận tham số đầu vào là emotion với các giá trị như: angry, happy, sad,... tương ứng với mỗi giá trị của cảm xúc sẽ là avatar có tâm trạng tương ứng dưới dạng ảnh SVG.

2.3.3. Hoàn thiện ứng dụng

Trong App Component tiến hành tạo phần tử thông tin tiêu đề, thông tin ứng dụng (chân trang), lồng ghép Camera Component và Avatar Component trong khung tương tác, bố cục cân đối các phần tử.

Sau khi ứng dụng hoàn tất kết quả đạt được như sau:

Giao diện ứng dụng ở chế độ phát triển



Hình ảnh 3: Giao diện ứng dụng ở chế độ phát triển

Upload mã nguồn lên Github (địa chỉ mã nguồn: <https://github.com/hkienquyet/face2avatar>) để lưu trữ và chia sẻ ứng dụng. Build ứng dụng, đưa ứng dụng đã build lên địa chỉ <https://mausosanh.com/face2avatar> để tiếp cận đến người dùng.

III. THỬ NGHIỆM VÀ ĐÁNH GIÁ

Ứng dụng được triển khai tại địa chỉ <https://mausosanh.com/face2avatar> cho phép truy cập từ mọi nơi. Thử nghiệm với các trình duyệt: chrome, safari, firefox, cóc cóc, microsoft edge đều hoạt động trong thời gian thực.

Thử nghiệm kết quả nhận diện chính xác ở mức trên 87%. Ứng dụng đáp ứng được các yêu cầu đặt ra trong phạm vi nghiên cứu.

TÀI LIỆU THAM KHẢO

1. Trang <https://www.wikipedia.org/>, Bách khoa toàn thư mở đa ngôn ngữ;
2. Trang <https://nodejs.org/docs/latest/api/>, tài liệu chính thức của NodeJS;
3. Trang <https://react.dev/learn>, tài liệu chính thức của ReactJS.