

TRƯỜNG ĐẠI HỌC

**SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH**

**----🙣🕮🙡----**

HCMC University of Technology and Education

KHOA ĐÀO TẠO CHẤT LƯỢNG CAO

**NGÀNH CÔNG NGHỆ THÔNG TIN**

BÁO CÁO ĐỒ ÁN

**TRÍ TUỆ NHÂN TẠO**

**NHẬN DẠNG CHỮ VIẾT TAY VÀ NHẬN DẠNG**

**GIỌNG NÓI**

****

**GVHD: ThS. Trần Tiến Đức**

**Sinh viên thực hiện:**

Phan Thái Bảo 20110071

Nguyễn Hoàng Kha 20110500

Thành phố Hồ Chí Minh, tháng 11 năm 2022

**ĐIỂM SỐ**

|  |  |  |  |
| --- | --- | --- | --- |
| **TIÊU CHÍ** | **NỘI DUNG** | **TRÌNH BÀY** | **TỔNG** |
| **ĐIỂM** |  |  |  |

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN:**

**Giáo viên hướng dẫn:**

*(Kí và ghi rõ họ tên)*

**LỜI CẢM ƠN**

Để hoàn thành tốt đề tài và bài báo cáo này, chúng chúng em xin gửi lời cảm ơn chân thành đến giảng viên Trần Tiến Đức, các quý thầy cô trong khoa Đào tạo Chất Lượng Cao nói chung và ngành Công Nghệ Thông Tin nói riêng đã tận tình truyền đạt những kiến thức cần thiết giúp chúng chúng em có nền tảng để làm nên đề tài này, đã tạo điều kiện để chúng em có thể tìm hiểu và thực hiện tốt đề tài. Cùng với đó, chúng chúng em xin được gửi cảm ơn đến các bạn cùng khóa đã cung cấp nhiều thông tin và kiến thức hữu ích giúp chúng em có thể hoàn thiện hơn đề tài của mình.

Đề tài và bài báo cáo được chúng em thực hiện trong khoảng thời gian ngắn, với những kiến thức còn hạn chế cùng nhiều hạn chế khác về mặt kĩ thuật và kinh nghiệm trong việc thực hiện một dự án phần mềm. Do đó, trong quá trình làm nên đề tài có những thiếu sót là điều không thể tránh khỏi nên chúng em rất mong nhận được những ý kiến đóng góp quý báu của các quý thầy cô để kiến thức của chúng em được hoàn thiện hơn và chúng em có thể làm tốt hơn nữa trong những lần sau. Chúng em xin chân thành cảm ơn.

Cuối lời, chúng em kính chúc thầy luôn dồi dào sức khỏe và thành công hơn nữa trong sự nghiệp trồng người. Một lần nữa chúng em xin chân thành cảm ơn.

**TP.HCM, ngày 19 tháng 11 năm 2022**

**Chúng em xin chân thành cảm ơn**

**MỤC LỤC:**

[Chương 1: Tổng quan chương trình 5](#_Toc119763240)

[1.1. Giới thiệu về chương trình: 5](#_Toc119763241)

[1.2. Mục đích, tính năng: 5](#_Toc119763242)

[1.2.1. Mục đích: 5](#_Toc119763243)

[1.2.2. Tính năng: 5](#_Toc119763244)

[1.3. Giao diện dự kiến: 5](#_Toc119763245)

[Chương2. Thiết kế trang web và code nhận dạng 6](#_Toc119763246)

[2.1. Thiết kế giao diện: 6](#_Toc119763247)

[2.2. Code nhận dạng: 8](#_Toc119763248)

[Chương 3. Cài đặt và kiểm thử 14](#_Toc119763249)

[Chương 4. Kết Luận 15](#_Toc119763250)

[4.1. Kết Luận: 15](#_Toc119763251)

[4.2. Hướng phát triển 15](#_Toc119763252)

[Tài liệu tham khảo 16](#_Toc119763253)

# 

# Chương 1: Tổng quan chương trình

## 1.1. Giới thiệu về chương trình:

**Chế độ nhận dạng:** Trang web có hai chế độ nhận dạng là từ ảnh chữ viết tay tiếng Việt, chữ viết tay tiếng Anh, chữ tiếng Anh, tiếng Việt và chế độ còn lại là nhận dạng chữ giọng nói (chủ yếu là tiếng Anh).

**Cách dùng:** Ở phía trang chủ sẽ có hay nút là Image to Text và Audio to Text. Khi chọn nút Image to Text trang web từ trang chủ sẽ chuyển sang trang upload. Ở trang upload người dùng có thể kéo thả hoặc chọn từ máy những hình ảnh có chữ viết tay, hoặc chữ viết để nhận dạng. Sau khi upload file thành công trang web sẽ được chuyển sang trang decoded, ở đây sẽ hiển thị dữ liệu sau khi nhận dạng và có thêm chế độ chuyển ngữ và chuyển đổi thành file audio. Còn ở nút Audio to Text, khi click vào nút sẽ chuyển sang trang audio. Trong trang này sẽ có hai nút là open file và transcribe. Người dùng chọn open file để tải file audio chụ thể là file .wav (nội dung tiếng Anh) và chọn transcribe để thực hiện nhận dạng.

## 1.2. Mục đích, tính năng:

### 1.2.1. Mục đích:

Dùng trí tuệ nhân tạo để nhận dạng ảnh và tệp giọng nói sang dạng chữ.

### 1.2.2. Tính năng:

- Nhận dạng chữ viết tay tiếng Việt, tiếng Anh

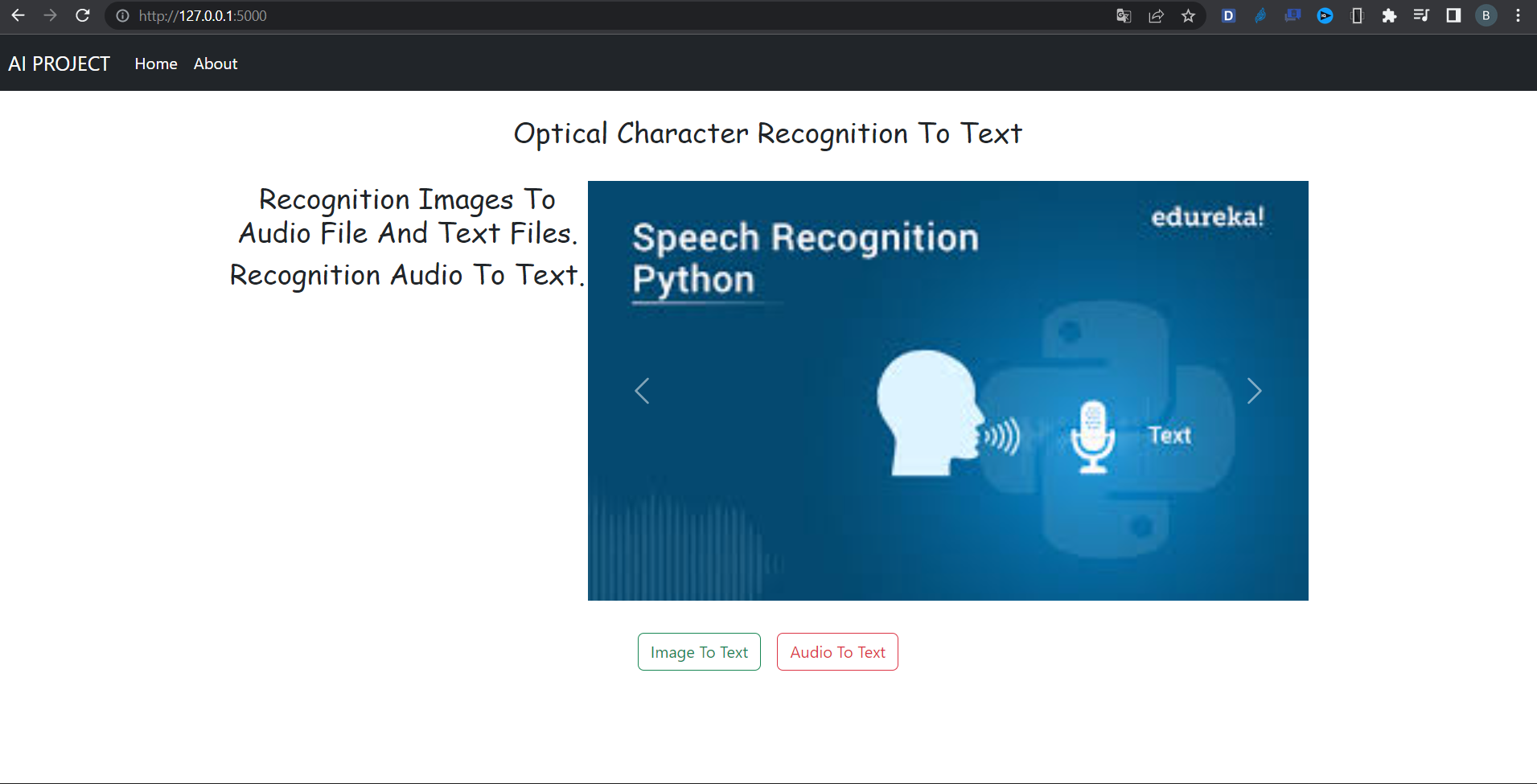
- Nhận dạng chữ viết tiếng Việt, tiếng Anh

- Chuyển đổi ngôn ngữ và tạo giọng nói từ dữ liệu được nhận dạng và chuyển đổi

- Nhận dạng giọng nói tiếng Anh

## 1.3. Giao diện dự kiến:

- Trang chủ:



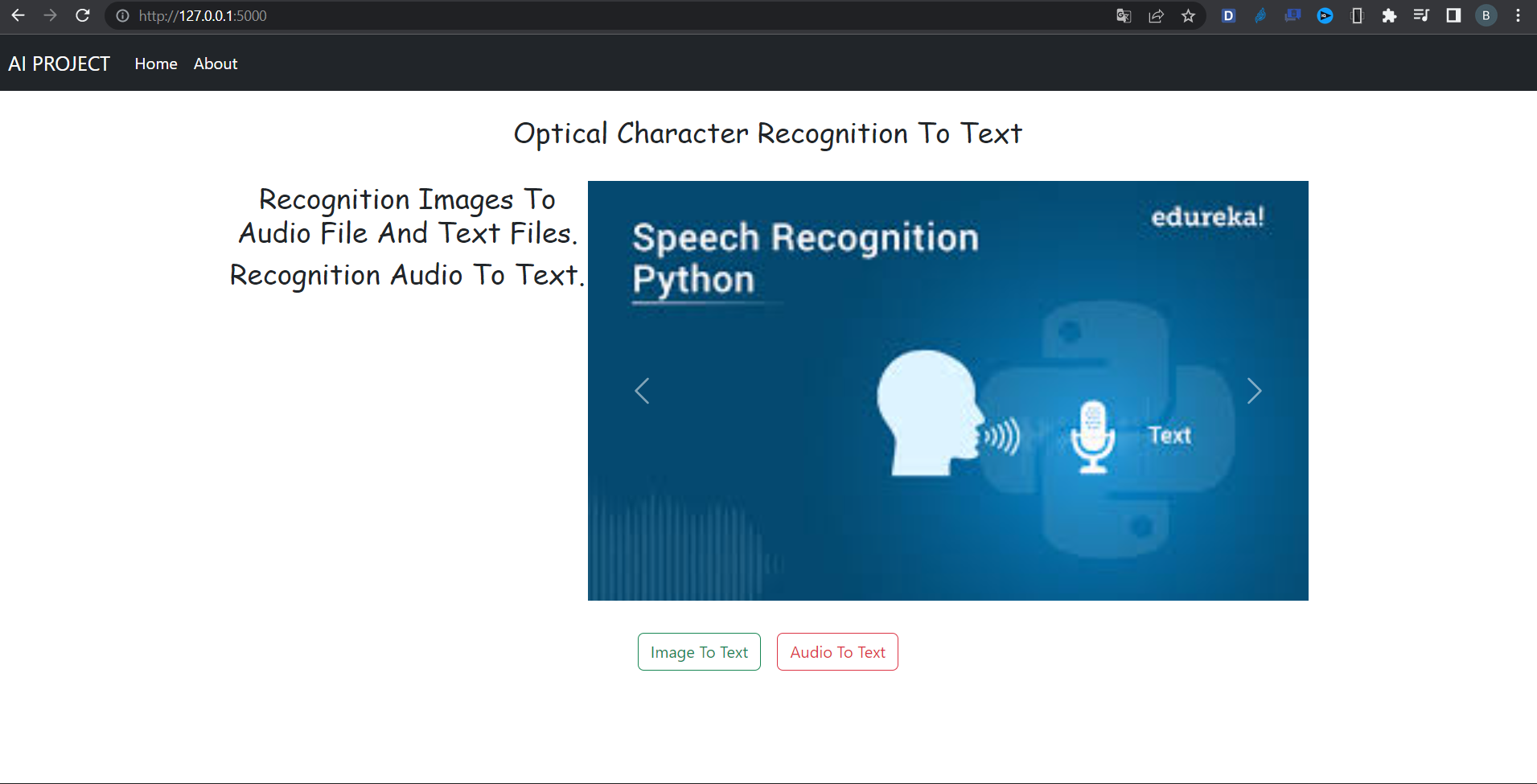
# 

# Chương2. Thiết kế trang web và code nhận dạng

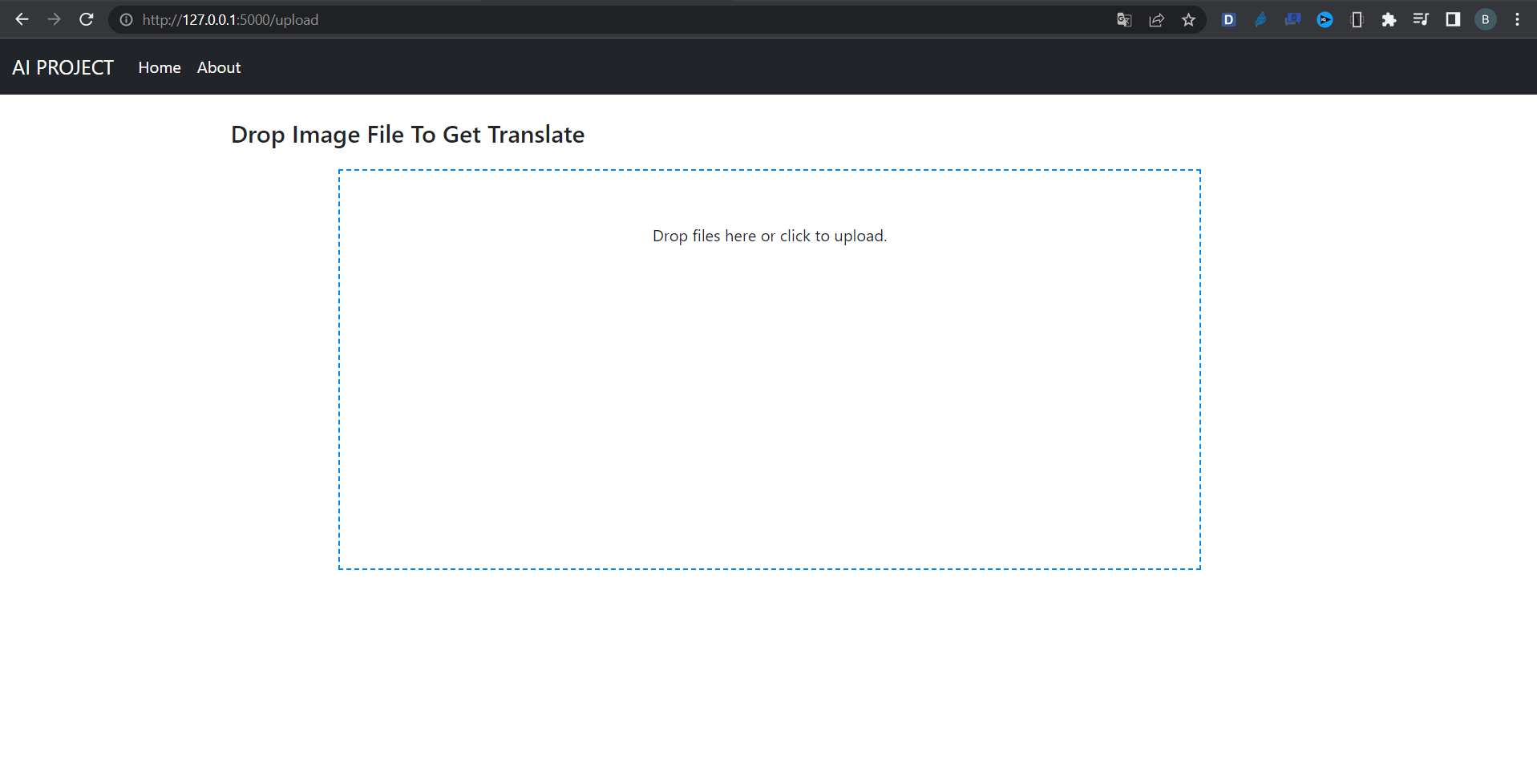
## 2.1. Thiết kế giao diện:

2.1.1. Giao diện chính:

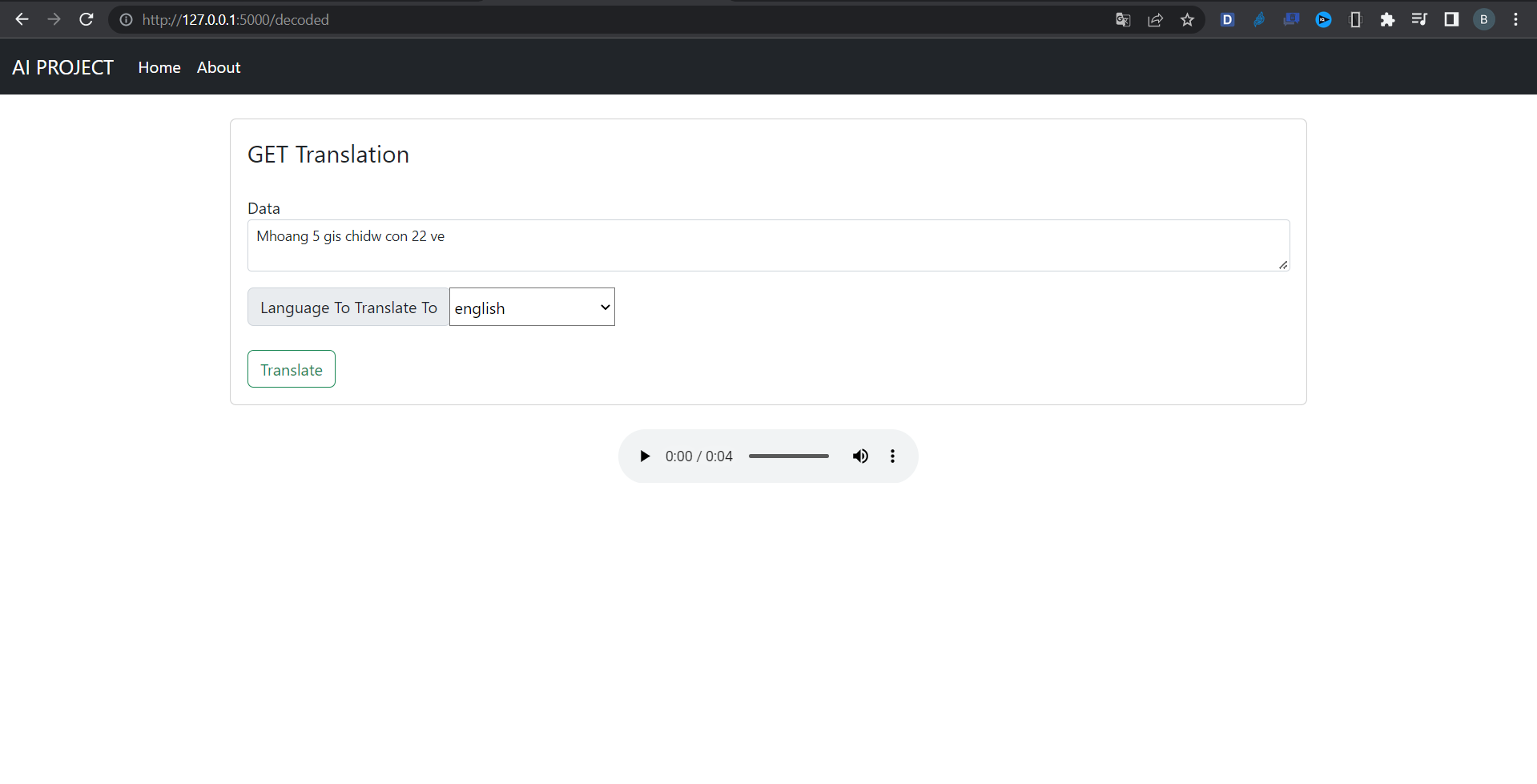
Trang chủ:



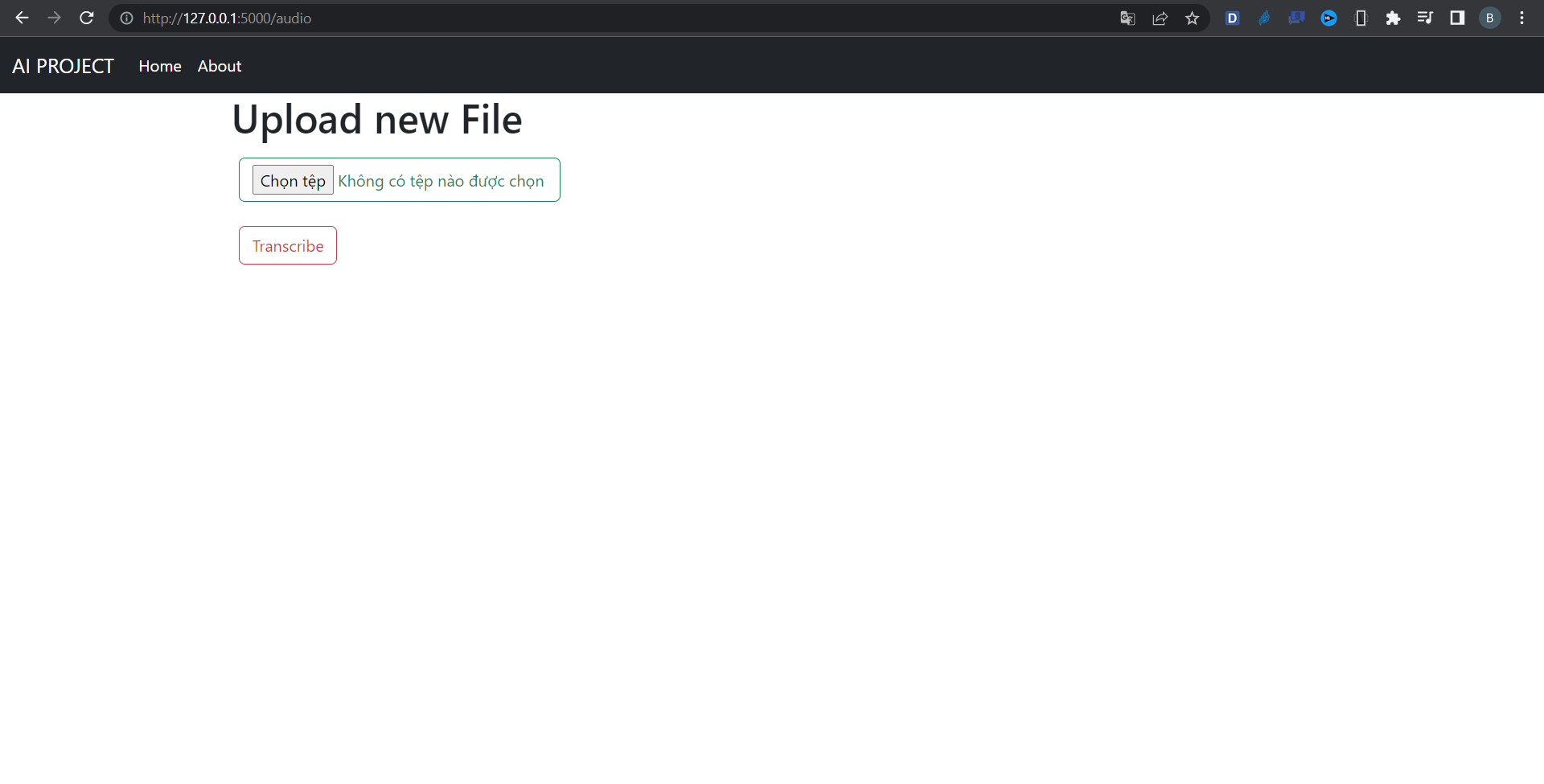
2.1.2 Giao diện trang upload



2.1.3 Giao diện trang decode



2.1.4 Giao diện trang audio



## 2.2. Code nhận dạng:

**2.2.1** **Thư viện và framework sử dụng chính:**

Để thực hiện được bài nhận dạng này chúng em đã dùng một số thư viện và framework của python để có thể hoàn thành được bài tập như là Tesseract, opencv, numpy, dùng flask framework, flask dropzone, pillow, sweetalert, speech\_recognition, pytesseract, gtts, wtforms, flask\_wtf, Flask Sessions. Dùng bootstrap để xử lí giao diện.

Như chúng ta đã biết, có rất nhiều thư viện trong python để phục vụ việc xử lí các ứng dụng liên quan đến AI. Mỗi thư viện sẽ đảm nhiệm một chức năng riêng trong ứng dụng. Trường hợp này chúng em đã sử dùng những framework và thư viện trên để có thể hoàn thành được bài.

**2.2.2 Nhận dạng chữ viết, chữ viết tay**

- Môi trường ảo: venv để có thể làm việc mà ko bị ảnh hưởng bởi những phần mềm và thư viện đã cài trong hệ thống. Được tùy biến sử dụng các thư viện cần thiết theo từng phiên bản mình muốn và có thể chạy được thoải máy trên các máy khác nhau.

- Upload file và xử lí:

@app.route("/upload", *methods*=["POST", "GET"])

def upload():

    if request.method == "POST":

        sentence = ""

        f = request.files.get("file")

        filename, extension = f.filename.split(".")

        generated\_filename = secrets.token\_hex(20) + f".{extension}"

        file\_location = os.path.join(app.config["UPLOADED\_PATH"], generated\_filename)

        f.save(file\_location)

        pytesseract.pytesseract.tesseract\_cmd = r'C:\\Program Files\\Tesseract-OCR\\tesseract.exe'

        img = cv2.imread(file\_location)

        img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

        boxes = pytesseract.image\_to\_data(img)

        for i, box in enumerate(boxes.splitlines()):

            if i == 0:

                continue

            box = box.split()

            if len(box) == 12:

                sentence += box[11] + " "

        #print(sentence)

        session["sentence"] = sentence

        os.rchúng emove(file\_location)

        return redirect("/decoded/")

    else:

        return render\_tchúng emplate("upload.html", *title*="Upload")

@app.route("/upload", methods=["POST", "GET"]) để mapping URL với phương thức POST và GET. Mở trình duyệt và truy cập 127.0.0.1:5000/upload sẽ thấy được trang upload

Trong hàm upload chúng em sẽ tiến hành upload file lên và xử lý ảnh quan học và trả về kết quả sau khi nhận dạng. Nếu yêu cầu là POST thì chúng em sẽ thực hiện lấy file và chia tệp thành danh sách bằng cách sử dụng split(.). Và để tránh trường hợp chúng ta tải các tệp trùng tên lên trang web và lưu về máy chủ nên chúng em đã dùng secrets.token\_hex(20) + f".{extension}" để tạo ra tên file ngẫu nhiên để không bị trùng tên file và sau đó sẽ lưu file vừa tải lên đó vào folder upload\_files thông qua dòng lệnh os.path.join(app.config["UPLOADED\_PATH"], generated\_filename) với UPLOADED\_PATH được chúng em tạo từ file \_\_init\_\_.py với đường dẫn vào folder uploaded\_files.

dir\_path = os.path.dirname(os.path.realpath(\_\_file\_\_))

app.config.update(

*UPLOADED\_PATH* = os.path.join(dir\_path, "static/uploaded\_files/"),

*DROPZONE\_ALLOWED\_FILE\_TYPE* = "image",

*DROPZONE\_MAX\_FILE\_SIZE* =3,

*DROPZONE\_MAX\_FILES* = 1,

*AUDIO\_FILE\_UPLOAD* = os.path.join(dir\_path, 'static/audio\_files/')

)

Sau khi upload file lên chúng em sẽ dùng tesseract để xác định tất cả các kí tự trong đó. Dùng cv2 để để đọc ảnh bằng lệnh imread() và tiến hành chuyển ảnh tạo màu kết nối bằng ctvColor và chuyển hình ảnh thành tỷ lệ RGB. Tiến hành chuyển từ ảnh sang dạng dữ liệu và sẽ tiến hành thực hiện vòng lặp để chia dữ liệu thành dòng và ghép thành chuỗi kết quả và chuyển kết quả sang trang decoded.

* Decoded nhận dạng và chuyển ngữ:
* @app.route("/decoded", *methods*=["POST", "GET"])
* def decoded():
* sentence = session.get("sentence")
* form = MyForm()
* if request.method =="POST":
* generated\_audio\_filename= secrets.token\_hex(10) + ".WAV"
* text\_data = form.text\_field.data
* translate\_to = form.language\_field.data
* print("Translate to: ", translate\_to)
* translated\_text = utils.translate\_text(text\_data, translate\_to)
* form.text\_field.data = translated\_text
* tts = gTTS(translated\_text, *lang*=translate\_to)
* file\_location = os.path.join(app.config["AUDIO\_FILE\_UPLOAD"], generated\_audio\_filename)
* tts.save(file\_location)
* return render\_tchúng emplate("decoded.html", *title*="Translations", *form* = form, *audio*=True, *file* = generated\_audio\_filename)
* else:
* form.text\_field.data = sentence
* session["sentence"] = ""
* return render\_tchúng emplate("decoded.html", *form* = form, *audio* = False)

Từ file forms.py, chúng em có tạo class MyForm và ở class này chúng em sẽ thực hiện lấy dữ liệu được nhận dạng, chuyển đổi ngôn ngữ và submit và được sử dụng để decoded.

class MyForm(FlaskForm):

    text\_field = TextAreaField("Data", *validators*=[DataRequired()])

    language\_field = SelectField("Language To Translate To", *choices*=languages\_choice)

    submit = SubmitField("Translate")

Để đảm bảo bảo mật hơn nên chúng em sẽ sử dụng thêm app.config['SECRET\_KEY'] = '82f6c92c5fa5cecb12e0d62498e35cac2c30414f0a8d78081af34a78a649' trong file \_\_init\_\_.py. Đây là khóa bí mật tự tạo bằng cách lấy random. Ở phía file routes.py trong hàm decoded chúng em sẽ lấy dữ liệu từ form và lấy ngôn ngữ dịch được chọn.

Tiếp theo, để đảm bảo an toàn hơn khi lấy dữ liệu nhận dạng chúng em sẽ dùng session và vì nó là duy nhất nên nó sẽ tránh được các cạnh tranh thương mại hay tương tự. Để tạo session chúng em dùng flask-session ở file \_\_init\_\_.py

SESSION\_TYPE = 'filesystchúng em'

app.config.from\_object(\_\_name\_\_)

Session(app)

Session này chúng em sẽ dùng để lấy dữ liệu từ trang upload và dùng hiển thị trên trang decoded

session["sentence"] = sentence

sentence = session.get("sentence")

form.text\_field.data = sentence

        session["sentence"] = ""

Sau khi đưa được dữ liệu nhận dạng lên trang web chúng em sẽ tiếp tục cho phần dịch ra các loại ngôn ngữ và chuyển thành audio. Chúng em sẽ tạo ra một file là utils.py để chứa các key và value của các ngôn ngữ đồng thời viết các hàm chuyển đổi ngôn ngữ

from googletrans import Translator

translator = Translator()

def detect\_language(*text*):

    detect\_lang\_data = translator.detect(*text*)

    lang = detect\_lang\_data.lang

    confidence = detect\_lang\_data.confidence

    return lang, confidence

def translate\_text(*text*, *dest*):

    translated\_text = translator.translate(*text*, *dest*=*dest*)

    return translated\_text.text

ở phần này chúng em dùng hàm detect\_language và translate\_text để phát hiện ngôn ngữ và thực hiện dịch sang ngôn ngữ đó và trả về trên ô data phần dữ liệu đã được chuyển đổi form.text\_field.data = translated\_text. Sau khi dịch sang ngôn ngữ khác thành công chúng em sẽ dùng thêm gtts để bắt đầu chuyển từ dạng chữ sang âm thanh. Bằng cách lấy dữ liệu trong ô data và ngôn ngữ được chọn và sử dụng hàm gTTS() để thực hiện chuyển thành tệp âm thanh.

tts = gTTS(translated\_text, *lang*=translate\_to)

Và chúng em sẽ lưu tệp âm thanh đó vào folder audio\_files

file\_location = os.path.join(app.config["AUDIO\_FILE\_UPLOAD"], generated\_audio\_filename)

với tên ngẫu nhiên và .wav để tránh sự trùng lập khi lưu file. Cứ như vậy khi ta upload file ảnh chữ viết lên thì sẽ được mã hóa nhận dạng và chuyển sang trang decoded hiển thị trên ô data và khi chọn ngôn ngữ và chọn translate thì dữ liệu sẽ được dịch và tạo ra file audio. Cả hình ảnh tải lên và audio sinh ra đều sẽ được lưu lại máy chủ.

**2.2.3 Nhận dạng giọng nói tiếng Anh**

@app.route("/audio", *methods*=["GET", "POST"])

def audio():

    transcript = ""

    if request.method == "POST":

        print("FORM DATA RECEIVED")

        if "file" not in request.files:

            return redirect(request.url)

        file = request.files["file"]

        if file.filename == "":

            return redirect(request.url)

        if file:

            recognizer = sr.Recognizer()

            audioFile = sr.AudioFile(file)

            with audioFile as source:

                data = recognizer.record(source)

            transcript = recognizer.recognize\_google(data, *key*=None)

            print(transcript)

    return render\_tchúng emplate('audio.html', *transcript*=transcript)

Cũng giống như nhận dạng chữ viết thì nhận dạng giọng nói cũng sẽ cho ta tải lên một file âm thanh. Chúng em sẽ kiểm tra xchúng em file đó có tồn tại hay ko, nếu ko thì sẽ trở về lại trang audio. Nếu có thì sẽ giữ ở lại trang audio và thực hiện nhận dạng. Để nhận dạng chúng em sử dụng thư viện speech\_recognition có sẵn của python để thực hiện.

Đầu tiên chúng em sẽ gọi trình nhận dạng và lấy tệp âm thanh đã tải lên và tạo một tệp âm thanh sau đó chuyển nó vào chức năng ghi trong module nhận dạng và bắt đầu phân tích nó, chuyển nó về thành định dạng dễ hiểu và áp dụng chức năng nhận dạng.

if file:

            recognizer = sr.Recognizer()

            audioFile = sr.AudioFile(file)

            with audioFile as source:

                data = recognizer.record(source)

            transcript = recognizer.recognize\_google(data, *key*=None)

Sau đó chúng em sẽ mở đối tượng tệp âm thanh này và đọc. Chúng em sẽ dùng các hàm có sẵn trong speech\_recognition cụ thể là recognize\_google để thực hiện chuyển thành âm thanh thành chữ và đưa lên trang web.

    {% if transcript != "" %}

        <div *id*="speechTranscriptContainer mb-3">

            <h1>Transcript</h1>

            <p *id*="speechText">{{ transcript }}</p>

        </div>

    {% endif %}

**2.2.4 Giao diện trang web**

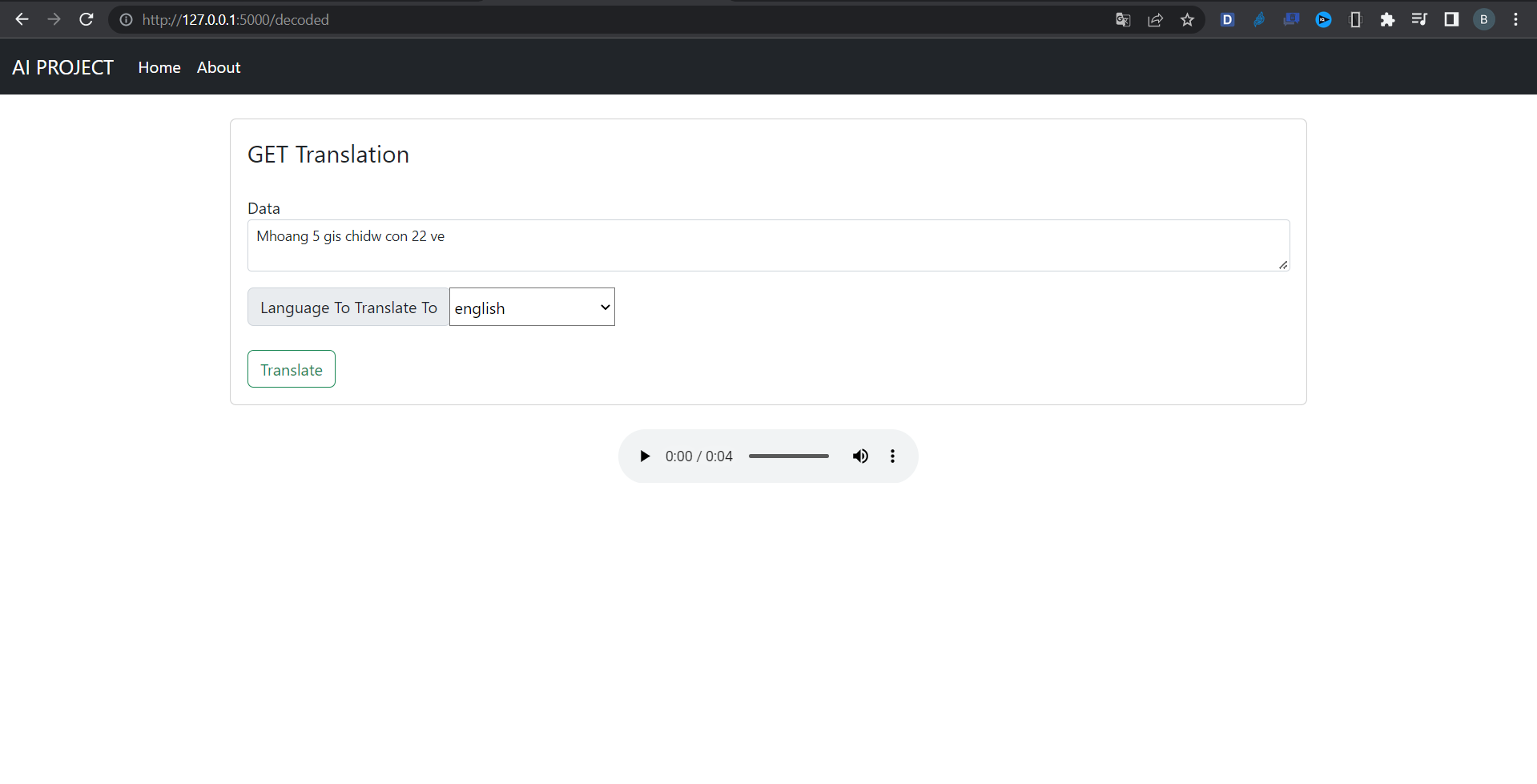
Về phía giao diện trang web thì chúng em dùng bootstrap để tạo một trang web nhanh chống với các css và js có sẵn.

Link trang bootstrap: https://getbootstrap.com/docs/5.2/getting-started/introduction/

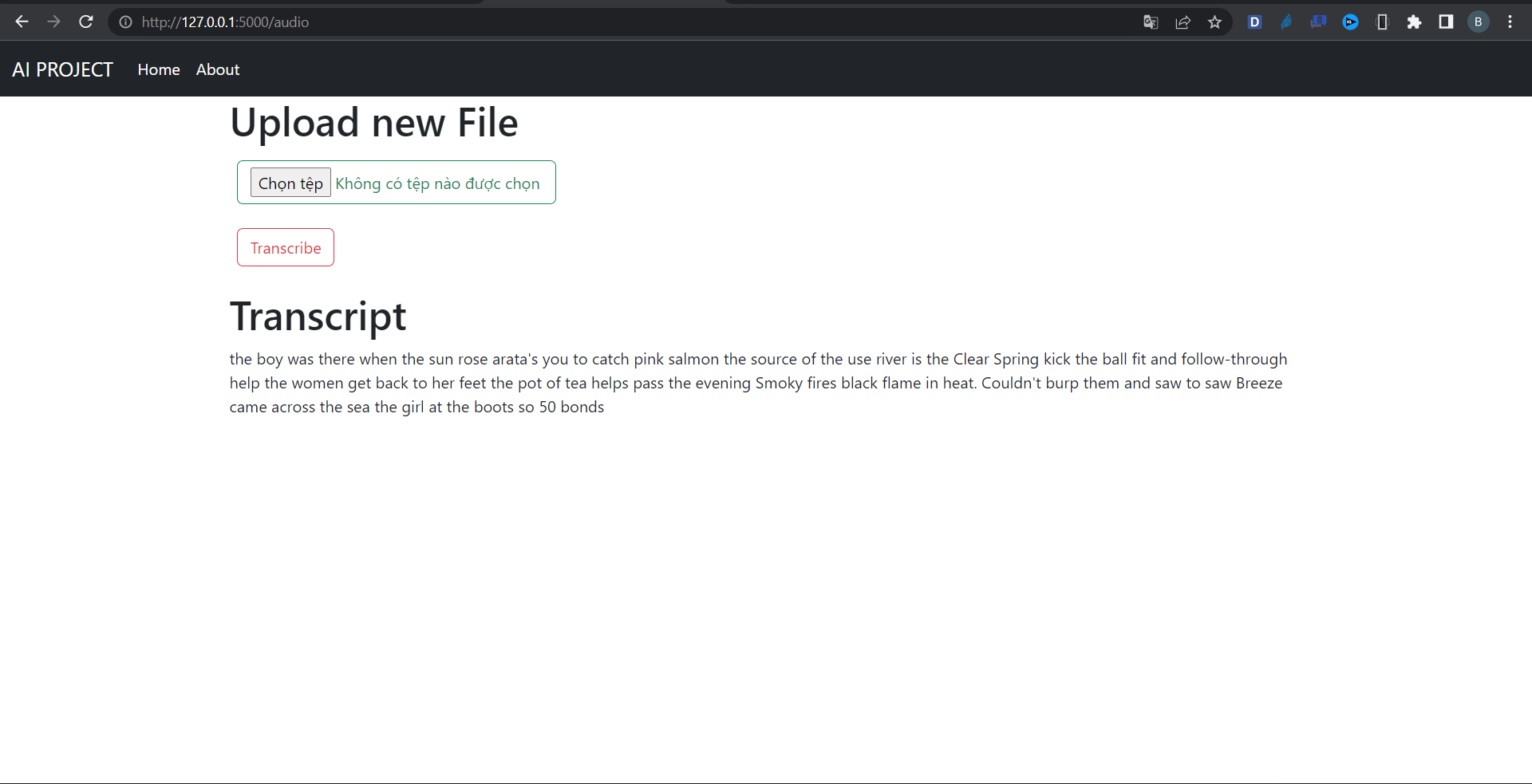
# 

# Chương 3. Cài đặt và kiểm thử

* Nhận dạng chữ viết và chuyển ngữ:



* *Nhận dạng giọng nói:*



# Chương 4. Kết Luận

## 4.1. Kết Luận:

Về cơ bản, chúng em tự nhận xét rằng chúng em đã cơ bản hoàn thành phân yêu cầu đề ra, và sau đây là một số ưu điểm và nhược điểm rút ra từ trang web nhận dạng:

* Về ưu điểm:

+ Giao diện dễ sử dụng.

+ Có 2 chế độ rõ ràng.

+ Chương trình nhẹ, hoạt động mượt mà.

+ Chế độ chuyển ngữ và chuyển thành audio.

* Về nhược điểm

+ Vẫn chưa nhận dạng được một chính xác hoàn toàn chữ viết tay tiếng Việt và tiếng Anh.

+ Vẫn chưa nhận dạng được giọng nói tiếng Việt

+ Thiết kế còn đơn giản.

Qua bài tập lớn này giúp chúng em hiểu hơn về trí tuệ nhân tạo và biết thêm được về framework flask của python.

## 4.2. Hướng phát triển

* Hoàn thiện các tính năng.
* Cải thiện việc nhận dạng.
* Thêm nhiều tính năng hơn.
* Phát triển giao diện đẹp mắt hơn.

LINK SOURCE CODE: <https://github.com/thaibao2806/Project_Recognition_AI>

# Tài liệu tham khảo

1. <https://www.youtube.com/watch?v=lTwRihIvTB4>
2. https://www.youtube.com/watch?v=vuaolF-OSGY&t=1359s
3. https://www.voiptroubleshooter.com/open\_speech/american.html