Software Development Project

Multilingual: Text to Speech
Presentation 5

Albert Millert
Shalini Priya
Wenjun Sun
Soklay Heng

Primary goal

Develop a web application that uses Grad-TTS Model for Text to Speech conversion. Languages supported by the TTS converter app are:

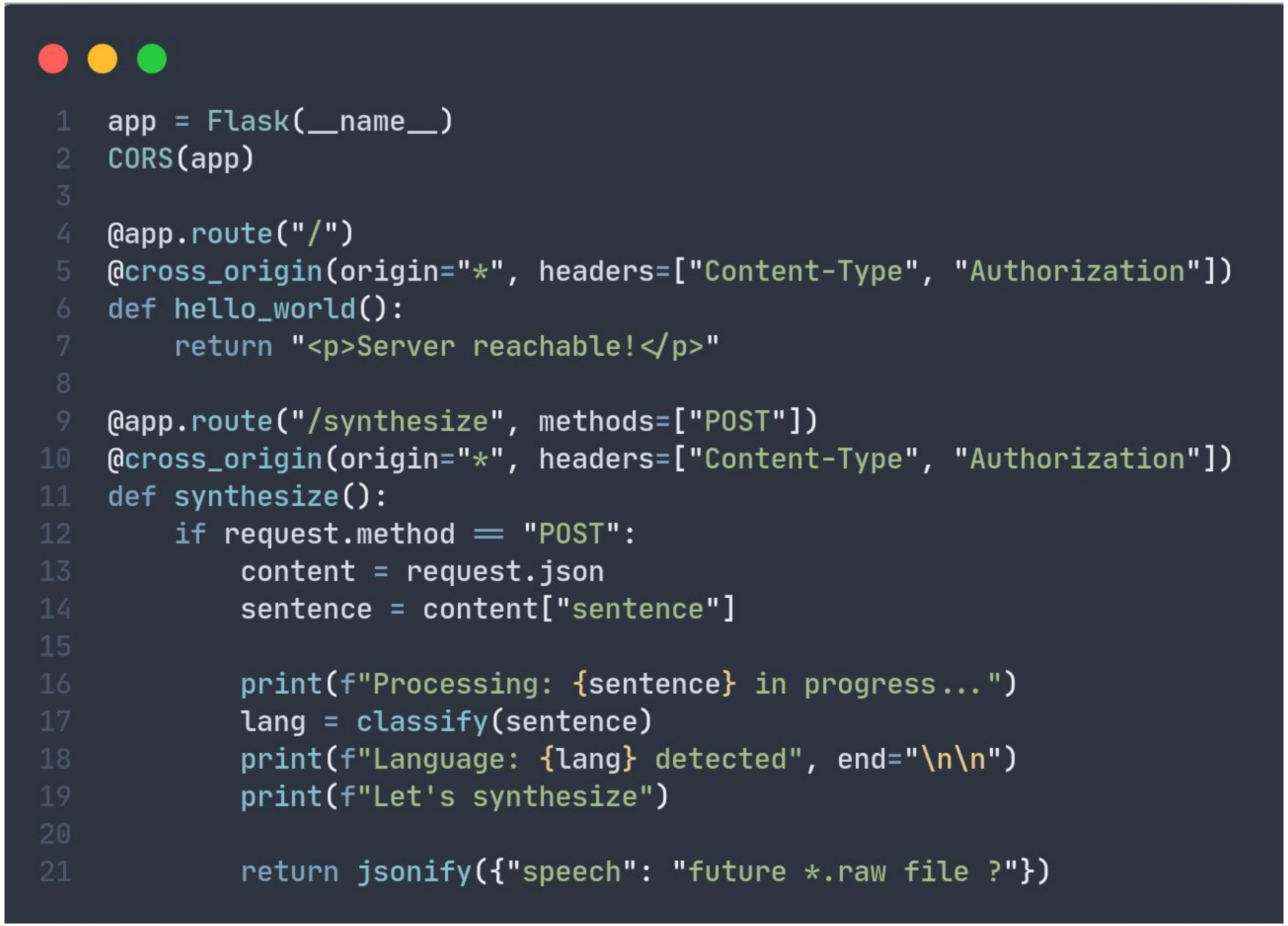
- English
- French

Completed

1. Language classifier + prediction
2. Server + (ready) modules integration
3. **Server + frontend integration**
4. English TTS

Client - server integration

- Enable Cross Origin Resource Sharing (*CORS*)
  - allows accessing resources
  - more security by specifying headers, authorization
- HTTP method update from GET to POST since React doesn’t allow to provide any payload while GETting from server



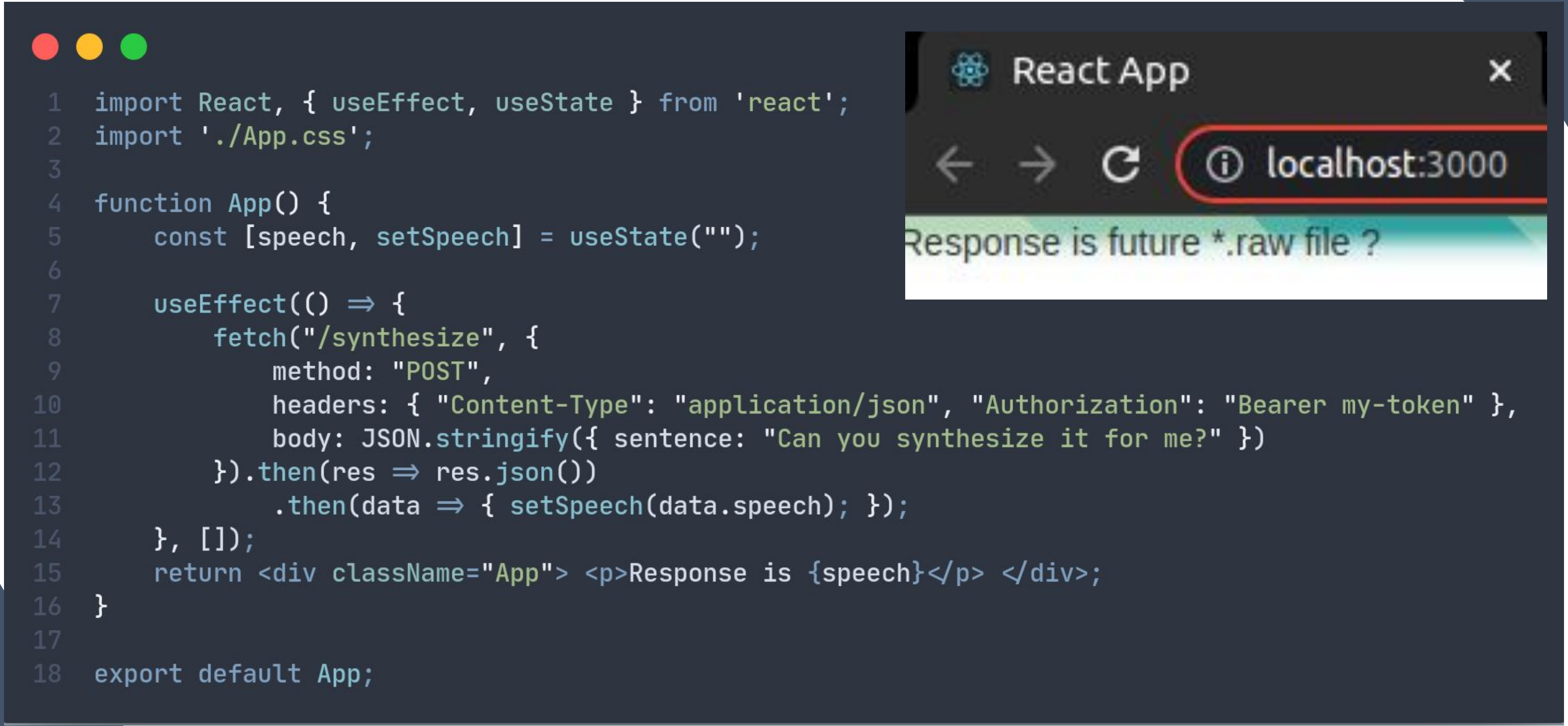
```python
app = Flask(__name__)
CORS(app)

@app.route("/")
@cross_origin(origin="*", headers=["Content-Type", "Authorization"])
def hello_world():
    return "<p>Server reachable!</p>"

@app.route("/synthesize", methods=["POST"])
@cross_origin(origin="*", headers=["Content-Type", "Authorization"])
def synthesize():
    if request.method == "POST":
        content = request.json
        sentence = content["sentence"]

        print(f"Processing: {sentence} in progress...")
        lang = classify(sentence)
        print(f"Language: {lang} detected", end="\n\n")
        print(f"Let's synthesize")

        return jsonify({"speech": "future *.raw file ?"})
```

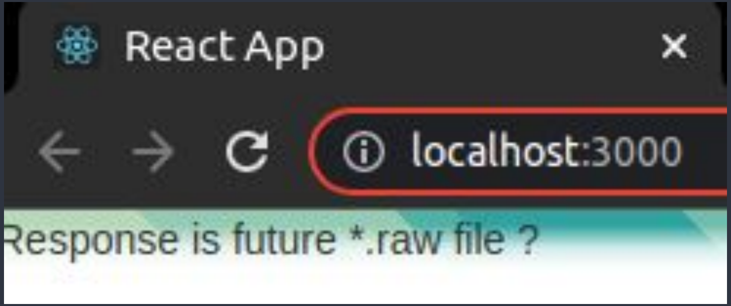


```
import React, { useEffect, useState } from 'react';
import './App.css';

function App() {
    const [speech, setSpeech] = useState("");

    useEffect(() => {
        fetch("/synthesize", {
            method: "POST",
            headers: { "Content-Type": "application/json", "Authorization": "Bearer my-token" },
            body: JSON.stringify({ sentence: "Can you synthesize it for me?" })
        }).then(res => res.json())
            .then(data => { setSpeech(data.speech); });
    }, []);
    return <div className="App"> <p>Response is {speech}</p> </div>;
}

export default App;
```

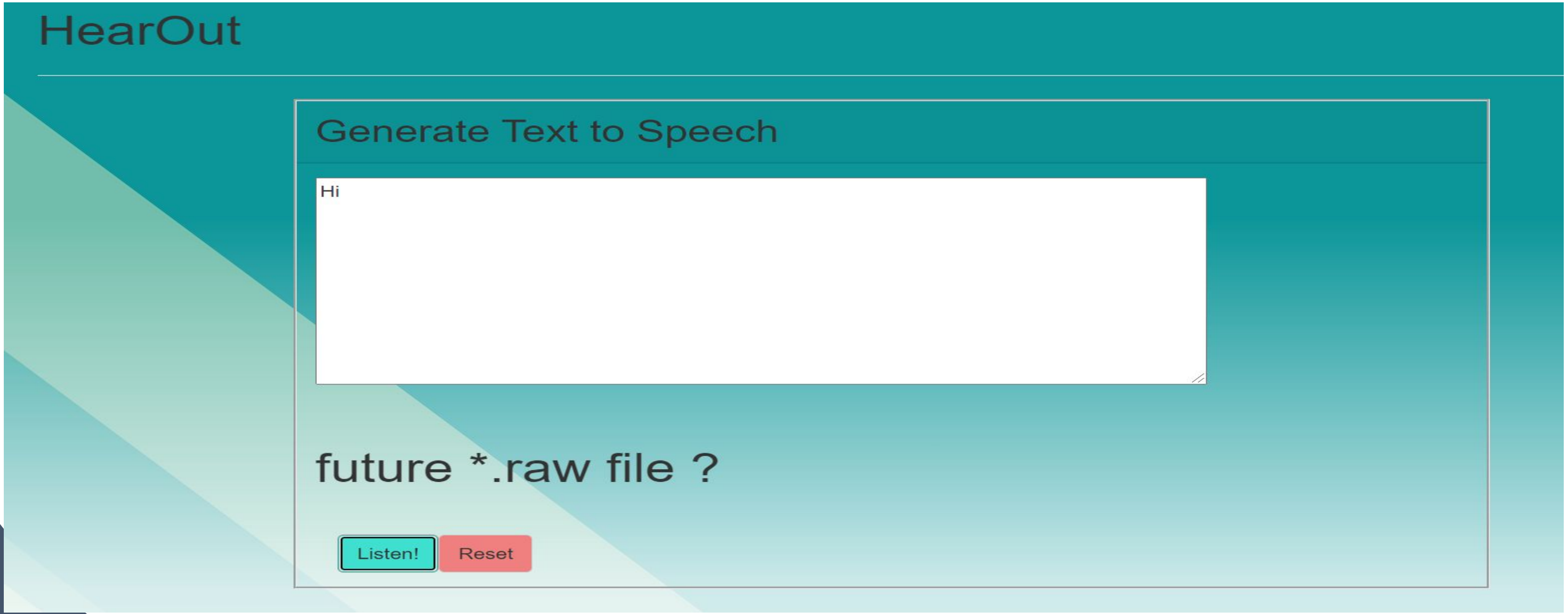


Response is future *.raw file ?

Server - English model integration

- Still waiting for the pretrained English model
  - include in the environment
  - Serve output to frontend as a somewhat quick response

Client



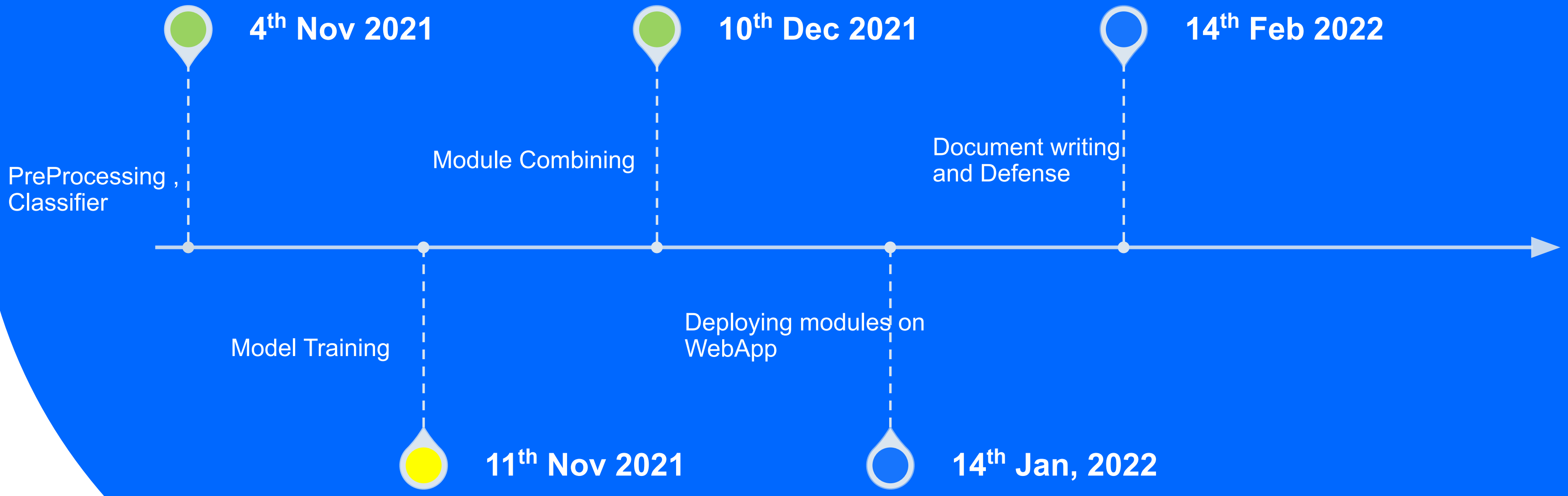
HearOut

Generate Text to Speech

Hi

future *.raw file ?

Timeline


4th Nov 2021
PreProcessing , Classifier
11th Nov 2021
Model Training
10th Dec 2021
Module Combining
14th Jan, 2022
Deploying modules on WebApp
14th Feb 2022
Document writing and Defense

New Idea

Generate emotional speech

Train the neural network to judge the sentiment of the sentence

Find a corpus and train a speech model

Process：After entering the sentence, it will automatically recognize its emotions and finally generate speech， also users can choose the sentiment of the sentence by themselves

Thank you