# 客户端app：

## 程序入口App.js

import React, { useState } from 'react';

import { StyleSheet, Text, View ,Button, Alert, TextInput} from 'react-native';

import { NavigationContainer } from '@react-navigation/native';

import { createNativeStackNavigator } from '@react-navigation/native-stack';

import AsyncStorage from '@react-native-async-storage/async-storage';

import NewHomePage from "./screen/NewHomePage"

import Profiles from './screen/Profiles';

import Live from './screen/Live'

import Weather from './screen/Weather';

import Settings from './screen/Settings'

import Help from './screen/Help'

import Info from './screen/Info'

const Stack = createNativeStackNavigator();

export default function App(){

return (

//导航堆栈

<NavigationContainer>

<Stack.Navigator initialRouteName="NewHomePage">

<Stack.Screen name = "NewHomePage" component = {NewHomePage} options={{headerShadowVisible:true,headerStyle:{backgroundColor:"skyblue"},headerShown:false,title:"MyApp"}} />

<Stack.Screen name = "Profiles" component = {Profiles} options={{animation:'slide\_from\_right',headerStyle:{backgroundColor:"#fafafa"}}}/>

<Stack.Screen name = "Live" component = {Live} options={{animation:'slide\_from\_right'}} />

<Stack.Screen name = "Weather" component = {Weather} options={{animation:'slide\_from\_right'}} />

<Stack.Screen name = "Settings" component = {Settings} options={{animation:'slide\_from\_right'}} />

<Stack.Screen name = "Help" component = {Help} options={{animation:'slide\_from\_right'}} />

<Stack.Screen name = "Info" component = {Info} />

</Stack.Navigator>

</NavigationContainer>

);

}

## 主页NewHomepage.js

import React, { useState,useEffect } from 'react';

import { StyleSheet, View, Button, Text, Alert, TouchableOpacity, TouchableHighlight } from 'react-native';

import { AntDesign } from '@expo/vector-icons';

import { MaterialIcons } from '@expo/vector-icons';

import { MaterialCommunityIcons } from '@expo/vector-icons';

import { FontAwesome5 } from '@expo/vector-icons';

import { Foundation } from '@expo/vector-icons';

export default function NewHomePage({navigation}){

return (

<View style = {styles.container}>

<View style = {{height:50,marginTop:30}}></View>

{/\*分割界线\*/}

<View style = {styles.mainTouchableArea}>

<TouchableHighlight style = {{flex:1}} onPress={() => {navigation.navigate("Profiles")}} activeOpacity={0.6} underlayColor="#DDDDDD">

<View style = {{flex:1,flexDirection:'row',justifyContent:'center'}}>

<View style = {{flex:1,justifyContent:'center',alignItems:'center'}}><AntDesign name="profile" size={32} color="blue" /></View>

<View style = {{flex:4,justifyContent:'center'}}><Text style = {styles.mainText}>Profiles</Text></View>

</View>

</TouchableHighlight>

</View>

<View style = {styles.mainTouchableArea}>

<TouchableHighlight style = {{flex:1}} onPress={() => {navigation.navigate("Live")}} activeOpacity={0.6} underlayColor="#DDDDDD">

<View style = {{flex:1,flexDirection:'row',justifyContent:'center'}}>

<View style = {{flex:1,justifyContent:'center',alignItems:'center'}}><MaterialIcons name="live-tv" size={32} color="green" /></View>

<View style = {{flex:4,justifyContent:'center'}}><Text style = {styles.mainText}>Live</Text></View>

</View>

</TouchableHighlight>

</View>

<View style = {styles.mainTouchableArea}>

<TouchableHighlight style = {{flex:1}} onPress={() => {navigation.navigate("Weather")}} activeOpacity={0.6} underlayColor="#DDDDDD">

<View style = {{flex:1,flexDirection:'row',justifyContent:'center'}}>

<View style = {{flex:1,justifyContent:'center',alignItems:'center'}}><MaterialCommunityIcons name="weather-partly-cloudy" size={32} color="orange" /></View>

<View style = {{flex:4,justifyContent:'center'}}><Text style = {styles.mainText}>Weather</Text></View>

</View>

</TouchableHighlight>

</View>

{/\*分割界线\*/}

{/\* <View style = {styles.otherTouchableArea}>

<TouchableHighlight style = {{flex:1}} onPress={() => {}} activeOpacity={0.6} underlayColor="#DDDDDD" >

<View style = {{flex:1,flexDirection:'row',justifyContent:'center'}}>

<View style = {{flex:1,justifyContent:'center',alignItems:'center'}}><FontAwesome5 name="clipboard-list" size={32} color="black" /></View>

<View style = {{flex:4,justifyContent:'center'}}><Text style = {styles.mainText}>Logs</Text></View>

</View>

</TouchableHighlight>

</View> \*/}

<View style = {styles.otherTouchableArea}>

<TouchableHighlight style = {{flex:1}} onPress={() => {navigation.navigate("Settings")}} activeOpacity={0.6} underlayColor="#DDDDDD" >

<View style = {{flex:1,flexDirection:'row',justifyContent:'center'}}>

<View style = {{flex:1,justifyContent:'center',alignItems:'center'}}><MaterialIcons name="settings" size={32} color="black" /></View>

<View style = {{flex:4,justifyContent:'center'}}><Text style = {styles.mainText}>Settings</Text></View>

</View>

</TouchableHighlight>

</View>

<View style = {styles.otherTouchableArea}>

<TouchableHighlight style = {{flex:1}} onPress={() => {navigation.navigate("Help")}} activeOpacity={0.6} underlayColor="#DDDDDD" >

<View style = {{flex:1,flexDirection:'row',justifyContent:'center'}}>

<View style = {{flex:1,justifyContent:'center',alignItems:'center'}}><MaterialIcons name="help-center" size={32} color="black" /></View>

<View style = {{flex:4,justifyContent:'center'}}><Text style = {styles.mainText}>Help</Text></View>

</View>

</TouchableHighlight>

</View>

<View style = {styles.otherTouchableArea}>

<TouchableHighlight style = {{flex:1}} onPress={() => {Alert.alert(null,"app version 0.1.0",[],{cancelable:true})}} activeOpacity={0.6} underlayColor="#DDDDDD" >

<View style = {{flex:1,flexDirection:'row',justifyContent:'center'}}>

<View style = {{flex:1,justifyContent:'center',alignItems:'center'}}><Foundation name="info" size={32} color="black" /></View>

<View style = {{flex:4,justifyContent:'center'}}><Text style = {styles.mainText}>Info</Text></View>

</View>

</TouchableHighlight>

</View>

</View>

);

}

const styles = StyleSheet.create({

container:{

flex:1,

backgroundColor:"#fafafa",

alignItems: 'center',

},

mainTouchableArea:{

height:80,

backgroundColor:"#ffffff",

width:350,

borderRadius:10,

marginBottom:12,

shadowColor:"#000",

shadowOffset:{width:0,height:1},

shadowOpacity:0.6,

shadowRadius:2,

elevation:2

},

otherTouchableArea:{

height:60,

width:350,

marginBottom:10,

},

mainText:{

textAlign:'left',

fontSize:20,

}

});

## 配置页面Profile.js

import React, { useState,useEffect } from 'react';

import { StyleSheet, Text, View , Alert, TextInput, Button, TouchableOpacity, BackHandler} from 'react-native';

import AsyncStorage from '@react-native-async-storage/async-storage';

import { FontAwesome } from '@expo/vector-icons';

const storeData = async (key,value) => {

try {

await AsyncStorage.setItem(key,value)

} catch (error) {

// Alert.alert("error",error);

}

}

const getData = async (key) => {

try {

let data = await AsyncStorage.getItem(key);

return data;

} catch(error) {

return "null";

}

}

export default function Profiles(){

const [isFocused1, setIsFocused1] = useState(false);

const [isFocused2, setIsFocused2] = useState(false);

const [VideoURL,setVideoURL] = useState("");

const [ServerURL,setServerURL] = useState("");

useEffect(() => {

async function getVideoURL(){

let videoURL = await getData("VideoURL");

setVideoURL(videoURL);

}

async function getServerURL(){

let serverURL = await getData("ServerURL");

setServerURL(serverURL);

}

getVideoURL();

getServerURL();

},[]);

return (

<View style = {styles.container} >

<View style = {{marginTop:20}}><Text style = {styles.mainText}>VideoURL</Text></View>

<TextInput

style = {[styles.textInput,isFocused1 ? styles.focused : null]}

defaultValue = {VideoURL}

placeholder = "VideoURL"

onFocus={() => setIsFocused1(true)}

onBlur={() => setIsFocused1(false)}

onChangeText={(text) => setVideoURL(text)}

/>

<View><Text style = {styles.mainText}>SeverURL</Text></View>

<TextInput

style = {[styles.textInput,isFocused2 ? styles.focused : null]}

defaultValue = {ServerURL}

placeholder = "ServerURL"

onFocus={() => setIsFocused2(true)}

onBlur={() => setIsFocused2(false)}

onChangeText={(text) => setServerURL(text)}

/>

<View style = {{height:60,alignItems:'center',marginTop: 40}}>

<TouchableOpacity

onPress={() => {

storeData("VideoURL",VideoURL);

storeData("ServerURL",ServerURL);

Alert.alert(null,"Saved");

}

}

>

<View style = {{height:60,width: 150,backgroundColor:'#fafafa',borderRadius:12,borderColor:"green",borderWidth:1,flexDirection:'row',alignItems:'center'}}>

<View style = {{flex:1,alignItems:'center'}}><FontAwesome name="save" size={24} color="black" /></View>

<View style = {{flex:2,alignItems:'center',justifyContent:'center'}}><Text style = {{color:'green',fontSize:20}}>Save</Text></View>

</View>

</TouchableOpacity>

</View>

</View>

);

}

const styles = StyleSheet.create({

container:{

flex:1,

backgroundColor:"#fafafa",

},

mainText:{

textAlign:'center',

fontSize:20

},

textInput:{

height:50,

margin:20,

backgroundColor:'#fafafa',

padding:10,

fontSize:20,

borderRadius:10,

borderWidth:1

},

focused: {

borderColor: 'green',

borderWidth: 2

},

});

## 直播页面Live.js

import React, { useState,useEffect } from 'react';

import { StyleSheet, Text, View , Alert, Button} from 'react-native';

import AsyncStorage from '@react-native-async-storage/async-storage';

import MyVideo from '../components/MyVideo';

import WebSocketBar from '../components/WebSocketBar';

import MyWebView from '../components/MyWebView'

const getData = async (key) => {

try {

let data = await AsyncStorage.getItem(key);

return data;

} catch(error) {

return "null";

}

}

export default function Live(){

const [videoURL,setVideoURL] = useState("");

const [serverURL,setServerURL] = useState("");

const [isLoading,setIsLoading] = useState(true);

useEffect(() => {

async function getVideoURL(){

let videoURI = await getData("VideoURL");

setVideoURL(videoURI);

return Promise.resolve();

}

async function getServerURL(){

let serverURL = await getData("ServerURL");

setServerURL(serverURL);

return Promise.resolve();

}

// getVideoURL();

// getServerURL();

Promise.all([getVideoURL(), getServerURL()]).then(() => {

setIsLoading(false);

}).catch((error) => {

console.log(error);

});

},[])

async function turnLeft(){

try {

let response = await fetch(serverURL + "/left",{

method:"POST",

});

let data = await response.json().then((data) => {

// console.log(data);

});

} catch (error) {

Alert.alert("ERROR",error.toString(),[{text:"OK",style: "default"}]);

}

}

async function turnRight(){

try {

let response = await fetch(serverURL + "/right",{

method:"POST",

});

let data = await response.json().then((data) => {

// console.log(data);

});

} catch (error) {

Alert.alert("ERROR",error.toString(),[{text:"OK",style: "default"}]);

}

}

return (

<View style = {styles.container}>

<WebSocketBar serverURL = {serverURL} />

<View style = {{flex:1,marginTop:10}}>

{isLoading ? (<View><Text style = {{fontSize:16}}>Loading...</Text></View>) : (

<View style = {{flex:1}}>

{/\* <MyVideo videoURI={videoURL} /> \*/}

<MyWebView videoURL={videoURL} />

<View style = {{height:60,flexDirection: 'row',marginTop:10,marginBottom:10,alignItems:'center',justifyContent:'center'}}>

<View style = {{flex:1,paddingRight:10}}><Button title='left' onPress={turnLeft}></Button></View>

<View style = {{flex:1,paddingLeft:10}}><Button title='right' onPress={turnRight}></Button></View>

</View>

</View>

)}

</View>

</View>

);

}

const styles = StyleSheet.create({

container:{

flex:1,

justifyContent:'center',

alignItems:'center',

backgroundColor:"#fafafa",

},

mainText:{

textAlign:'center',

fontSize:20

}

});

## 天气页面Weather.js

import React, { useState, useEffect } from 'react';

import { StyleSheet, Text, View , Alert, Button} from 'react-native';

import AsyncStorage from '@react-native-async-storage/async-storage';

import { FontAwesome5 } from '@expo/vector-icons';

import { Ionicons } from '@expo/vector-icons';

const getData = async (key) => {

try {

let data = await AsyncStorage.getItem(key);

return data;

} catch(error) {

return "null";

}

}

export default function Weather(){

const [serverURL, setServerURL] = useState();

const [data,setData] = useState("");

async function getServerURL(){

try {

let url = await AsyncStorage.getItem("ServerURL");

setServerURL(url);

} catch (error) {

console.log(error);

}

}

async function getWeather(){

try {

let response = await fetch(serverURL + "/weather",{

method:"GET",

});

let data = await response.json().then((data) => {

setData(JSON.stringify(data));

});

} catch (error) {

// Alert.alert("ERROR",error.toString(),[{text:"OK",style: "default"}]);

}

}

async function update(){

try {

let response = await fetch(serverURL + "/weather",{

method:"GET",

});

let data = await response.json().then((data) => {

setData(JSON.stringify(data));

});

} catch (error) {

Alert.alert("ERROR",error.toString(),[{text:"OK",style: "default"}]);

}

}

useEffect(() => {

getServerURL();

// console.log(serverURL)

// getWeather();

}, []);

return (

<View style = {styles.container}>

{/\* <Text style = {styles.mainText}>Weather</Text> \*/}

<View style = {{backgroundColor:"#ffd700",height:80,width:360,borderRadius:10,marginTop:15,flexDirection:'row',alignItems:'center'}}>

<View style = {{flex:1,justifyContent:'center',alignItems:'center'}}><FontAwesome5 name="temperature-high" size={30} color="black" /></View>

<View style = {{flex:1}}><Text style = {{color:'#1C1C1C',fontSize:20}}>Temp: </Text></View>

<View style = {{flex:2}}><Text style = {{color:'#1C1C1C',fontSize:20,marginLeft:20}}>{(data == "") ? "" :JSON.parse(data).Temp+"°C"}</Text></View>

</View>

<View style = {{backgroundColor:"#66ccff",height:80,width:360,borderRadius:10,marginTop:10,flexDirection:'row',alignItems:'center'}}>

<View style = {{flex:1,justifyContent:'center',alignItems:'center'}}><Ionicons name="water" size={30} color="black" /></View>

<View style = {{flex:1}}><Text style = {{color:'#1C1C1C',fontSize:20}}>Humidity: </Text></View>

<View style = {{flex:2}}><Text style = {{color:'#1C1C1C',fontSize:20,marginLeft:20}}>{(data == "") ? "" :JSON.parse(data).Humidity+"%"}</Text></View>

</View>

<View style = {{width:360,margin:20}}><Button title = "Refresh" color="#32CD32" onPress={update}/></View>

</View>

);

}

const styles = StyleSheet.create({

container:{

flex:1,

// justifyContent:'center',

alignItems:'center',

backgroundColor:"#fafafa",

},

mainText:{

textAlign:'center',

fontSize:20

}

});

## 设置页面Settings.js

import React, { useState } from 'react';

import { StyleSheet, Text, View , Alert} from 'react-native';

export default function Settings(){

return (

<View style = {styles.container}>

<Text style = {styles.mainText}>this page is not avaliable yet</Text>

</View>

);

}

const styles = StyleSheet.create({

container:{

flex:1,

justifyContent:'center',

alignItems:'center',

backgroundColor:"#fafafa",

},

mainText:{

textAlign:'center',

fontSize:20

}

});

## 帮助页面Help.js

import React, { useState } from 'react';

import { StyleSheet, Text, View , Alert, TouchableHighlight, Linking} from 'react-native';

export default function Help(){

open = (url) => {

Linking.openURL(url);

}

let giteeURL = "https://github.com/xiaoyuan-c/RaspberryHomework";

return (

<View style = {styles.container}>

{/\* <Text style = {styles.mainText}>Help</Text> \*/}

<View style = {{flex:1}}>

<View style = {{height:50,width:350,marginTop:10,justifyContent:'center'}}><Text style = {{fontSize:20,color:'blue'}}>Source Code</Text></View>

<View style = {{height:50,width:350,marginTop:10}}>

<TouchableHighlight onPress={() => {this.open(giteeURL)}} activeOpacity={0.6} underlayColor="#DDDDDD" >

<View>

<View></View>

<View><Text style = {styles.mainText}>{giteeURL}</Text></View>

</View>

</TouchableHighlight>

</View>

<View style = {{height:50,width:350,marginTop:10,justifyContent:'center'}}><Text style = {{fontSize:20,color:'blue'}}>Feedback</Text></View>

<View style = {{height:50,width:350,marginTop:10}}>

<TouchableHighlight onPress={() => {this.open(giteeURL+"/issues")}} activeOpacity={0.6} underlayColor="#DDDDDD" >

<View>

<View></View>

<View><Text style = {styles.mainText}>{giteeURL + "/issues"}</Text></View>

</View>

</TouchableHighlight>

</View>

</View>

</View>

);

}

const styles = StyleSheet.create({

container:{

flex:1,

// justifyContent:'center',

alignItems:'center',

backgroundColor:"#fafafa",

},

mainText:{

fontSize:20

}

});

## 产品信息页面Info.js

import React, { useState } from 'react';

import { StyleSheet, Text, View , Alert} from 'react-native';

export default function About(){

return (

<View style = {styles.container}>

<Text style = {styles.mainText}>About</Text>

</View>

);

}

const styles = StyleSheet.create({

container:{

flex:1,

justifyContent:'center',

alignItems:'center',

backgroundColor:"#fafafa",

},

mainText:{

textAlign:'center',

fontSize:20

}

});

## 组件MyTextInput.js

import React, { useState } from 'react';

import { TextInput, StyleSheet, Alert } from 'react-native';

const MyTextInput = ({defaultValue,placeholder,updateParent}) => {

const [isFocused, setIsFocused] = useState(false);

const [inputValue, setInputValue] = useState(defaultValue);

return (

<TextInput

style={[

styles.input,

isFocused ? styles.focused : null,

]}

defaultValue = {defaultValue}

placeholder = {placeholder}

onFocus={() => setIsFocused(true)}

onBlur={() => setIsFocused(false)}

onChangeText={(inputValue) => {setInputValue(inputValue);updateParent(inputValue);}}

//setInputValue是异步函数，把inputValue作为参数传递给onChangText是为了及时更新

/>

);

};

const styles = StyleSheet.create({

input: {

height: 50,

margin: 20,

marginTop: 15,

marginBottom: 35,

padding:10,

fontSize: 20,

borderWidth: 1,

borderRadius: 10,

borderColor: 'black',

},

focused: {

borderColor: 'green',

borderWidth: 2

},

});

export default MyTextInput;

## 组件MyWebView.js

import React, { useState,useEffect } from 'react';

import { StyleSheet, Text, View , Alert, FlatList} from 'react-native';

import {WebView} from 'react-native-webview';

export default function MyWebView({videoURL}){

return (

<View style = {styles.container}>

<WebView

source={{ uri: videoURL }}

/>

</View>

);

}

const styles = StyleSheet.create({

container:{

flex:1,

height:300,

width:400,

}

})

## 组件Separator.js

import React from 'react';

import { StyleSheet, View, Text } from 'react-native';

const Separator = () => {

return <View style={{height: 2,width: 1000, backgroundColor: 'black'}} />;

}

export default Separator;

## 组件WebSocketBar.js

import React, { useState,useEffect } from 'react';

import { StyleSheet, Text, View , Alert, Button, TouchableOpacity} from 'react-native';

import io from 'socket.io-client';

const WebSocketBar = ({serverURL}) => {

const [color, setColor] = useState('pink');

const [isConnected,setIsConnected] = useState(false);

const [socket, setSocket] = useState(null);

const [status,setStatus] = useState(false);

useEffect(() => {

},[]);

return (

<View style={{ height: 20, width:360, backgroundColor: color, borderRadius:16 }}>

<TouchableOpacity style = {{flex:1}} onPress={() => {

setColor('skyblue');

const newSocket = io(serverURL);

// setSocket(newSocket);

setStatus(true);

newSocket.on('connect',() => {

console.log("Connected to server");

setIsConnected(true)

});

newSocket.on('warning\_message',(msg) => {

console.log("received my response: ", msg);

setColor(msg == '1' ? 'red' : 'skyblue');

})

setTimeout(() => {

newSocket.disconnect();

console.log('disconnect');

setColor('pink');

}, 60000);

}}>

<View style = {{flex:1}}></View>

</TouchableOpacity>

</View>

);

};

export default WebSocketBar;

# 服务器端：

## app.py

from flask import Flask, render\_template

from flask\_socketio import SocketIO, emit

import Adafruit\_DHT

import time

from flask\_cors import CORS

import sys

import RPi.GPIO as GPIO

import threading

app = Flask(\_\_name\_\_)

socketio = SocketIO(app)

CORS(app)

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BOARD)

def turn(steps, clockwise):

arr = [0, 1, 2, 3];

if clockwise != 1:

arr = [3, 2, 1, 0];

ports = [40, 38, 36, 35]

for p in ports:

GPIO.setup(p, GPIO.OUT)

for step in range(0, steps):

for i in arr:

time.sleep(0.01)

for j in range(0, 4):

if j == i:

GPIO.output(ports[j], GPIO.HIGH)

else:

GPIO.output(ports[j], GPIO.LOW)

def check\_gpio():

GPIO.setup(12, GPIO.IN)

while True:

time.sleep(1)

if GPIO.input(12) == True:

current\_time = time.strftime('%Y-%m-%d %H:%M:%S', time.localtime(time.time()))

warning\_message = current\_time + " warn"

print(warning\_message)

socketio.emit('warning\_message', '1')

else:

socketio.emit('warning\_message', '0')

@app.route('/weather')

def weather():

humidity, temperature = Adafruit\_DHT.read\_retry(11, 17)

# return '{0:0.1f} {0:0.1f}'.format(temperature,humidity)

# return render\_template('weather.html', Temp=temperature, Humidity=humidity)

return {"Temp": temperature, "Humidity": humidity}

@app.route('/left', methods=["GET", "POST"])

def left():

turn(120, 0)

return '0'

@app.route('/right', methods=["GET", "POST"])

def right():

turn(120, 1)

return '1'

@app.route('/monitor', methods=['GET', 'POST'])

def monitor():

return render\_template('index.html')

@app.route('/')

def main():

return render\_template('main.html')

if \_\_name\_\_ == '\_\_main\_\_':

threading.Thread(target=check\_gpio).start()

socketio.run(app, port=5000)

# app.run(port=5000)

## new\_auto.sh

./natapp -authtoken=ac59fb7cdcb888f8 -log=stdout &

./natapp -authtoken=ceea76e0eca74641 -log=stdout &

python3 app.py &

sudo service motion start &

sudo motion &

## templates/

这个文件下为探索过程中写网页的代码，最后被app替代

### index.html

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8" />

<title></title>

</head>

<body>

<script type="text/javascript" src="https://cdn.bootcss.com/flv.js/1.4.0/flv.min.js"></script>

<video id="videoElement" style="width: 80%" controls="controls"></video>

<div id = "bar" style="height: 30px;width: 200px;"></div>

<script type="module">

import { io } from "https://cdn.socket.io/4.3.2/socket.io.esm.min.js";

if (flvjs.isSupported()) {

var videoElement = document.getElementById('videoElement')

var flvPlayer = flvjs.createPlayer({

type: 'flv',

url: 'https://nailv.mynatapp.cc/live/stream.flv'

})

flvPlayer.attachMediaElement(videoElement)

flvPlayer.load()

}

let bar = document.querySelector('#bar');

// Create WebSocket connection.

let socket = io("http://ash.nat300.top");

socket.on('connect',(msg) => {

console.log(msg);

})

socket.on("warning\_message",(msg) => {

bar.innerHTML = msg.toString()

})

</script>

</body>

</html>

### main.html

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>主页</title>

<style>

body {

display: flex;

/\* flex-wrap: wrap; \*/

flex-direction: column;

justify-content: center;

align-items: center;

height: 100vh;

margin: 20px;

padding: 0;

background-color: #fafafa;

}

.title {

height: 100px;

width: 100%;

justify-content: center;

align-items: center;

display: flex;

}

.main-area {

height: auto;

width: 100%;

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

}

.area {

display: flex;

flex-direction: row;

}

.title-text {

text-align: center;

font-size: 40px;

width: 100%;

}

.text {

text-align: center;

font-size: 24px;

}

.square {

width: 200px;

height: 200px;

margin: 20px;

border: 2px;

border-radius: 20px;

display: flex;

flex-direction: row;

justify-content: center;

align-items: center;

}

.orange {

background-color: orange;

}

.skyblue {

background-color: skyblue;

}

.lightgreen {

background-color: lightgreen;

}

.purple {

background-color: purple;

}

.pink {

background-color: pink;

}

</style>

</head>

<body>

<div class="title"><p class="title-text">树莓盘智能监控站</p></div>

<div class="main-area">

<div class="area">

<div class="square orange">

<p class="text">天气</p>

</div>

<div class="square lightgreen">

<p class="text">监控</p>

</div>

</div>

<div class="area">

<div class="square skyblue">

<p class="text">帮助</p>

</div>

<div class="square pink">

<p class="text">项目</p>

</div>

</div>

</div>

</body>

<script>

let orangeBtn = document.querySelector('.orange');

let lightgreenBtn = document.querySelector('.lightgreen');

let skyblueBtn = document.querySelector('.skyblue')

let pinkbtn = document.querySelector('.pink');

orangeBtn.addEventListener('click',() => {

window.location.href = "http://ash.nat300.top/weather"

})

lightgreenBtn.addEventListener('click',() => {

window.location.href = "http://nailv.mynatapp.cc"

})

skyblueBtn.addEventListener('click',() => {

window.location.href = "#"

})

pinkBtn.addEventListener('click',() => {

window.location.href = "#"

})

</script>

</html>

### weather.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

</head>

<body>

Temperature:{{ Temperature }}°C

Humidity: {{ Humidity }}%

</body>

</html>