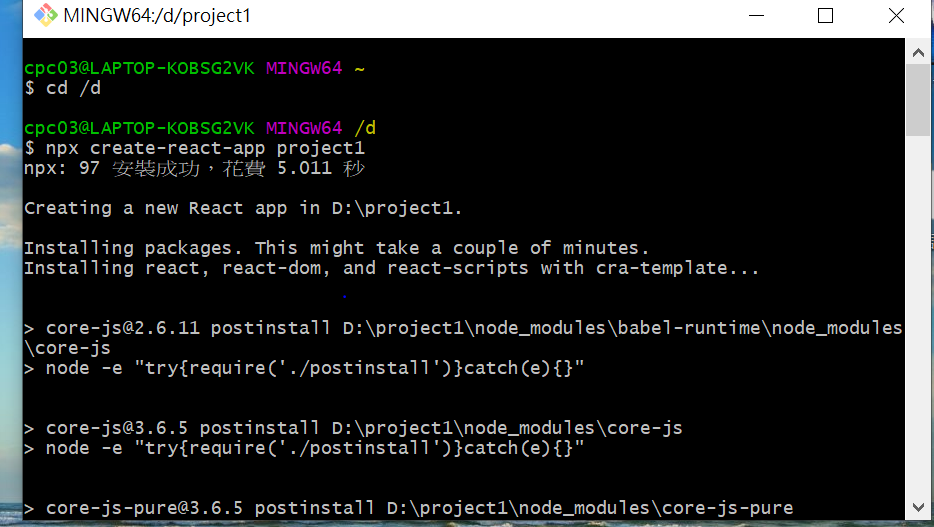
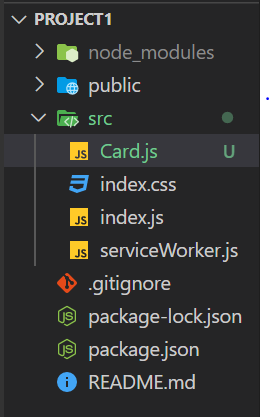
1.Create a react app for project



2.刪除不必要的檔案



3. 連接card.js顯示資料

import React from 'react'

const Card = () => {

return (

<div>

<img src='./image/intro.PNG?' width="300" alt='Intro' />

<h2>Intro</h2>

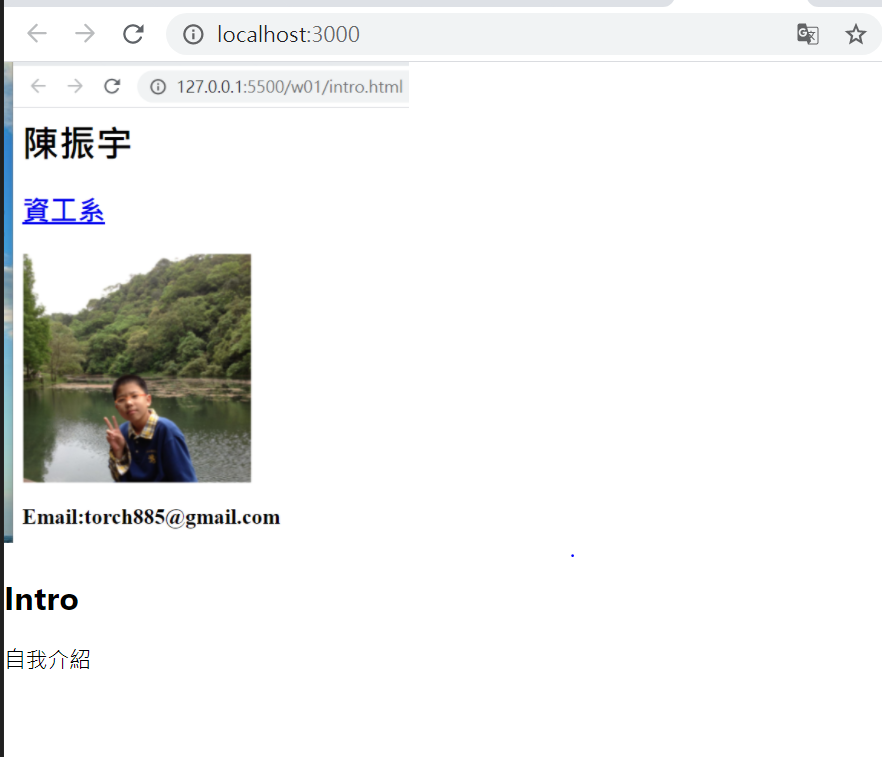
<p>自我介紹</p>

</div>

)

}

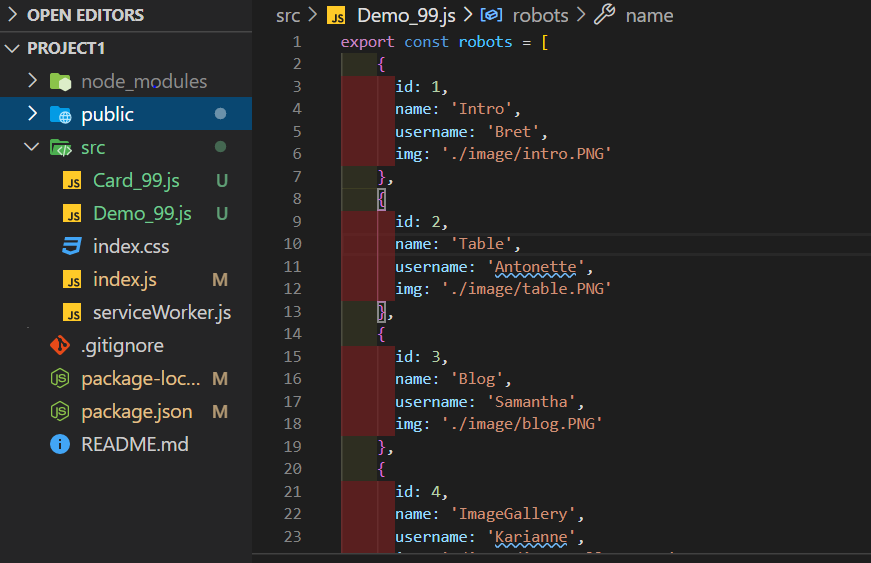
export default Card;



4.import tachyons



5.建立demo.js新增10筆資料



6.利用demo.js製作3張卡片

import React from 'react'

const Card = ({ name, username, img }) => {

return (

<div className='tc bg-light-green dib br3 pa3 ma2 grow bw2 shadow-5'>

<img src={img} width="300" alt='Intro' />

<h2>{name}</h2>

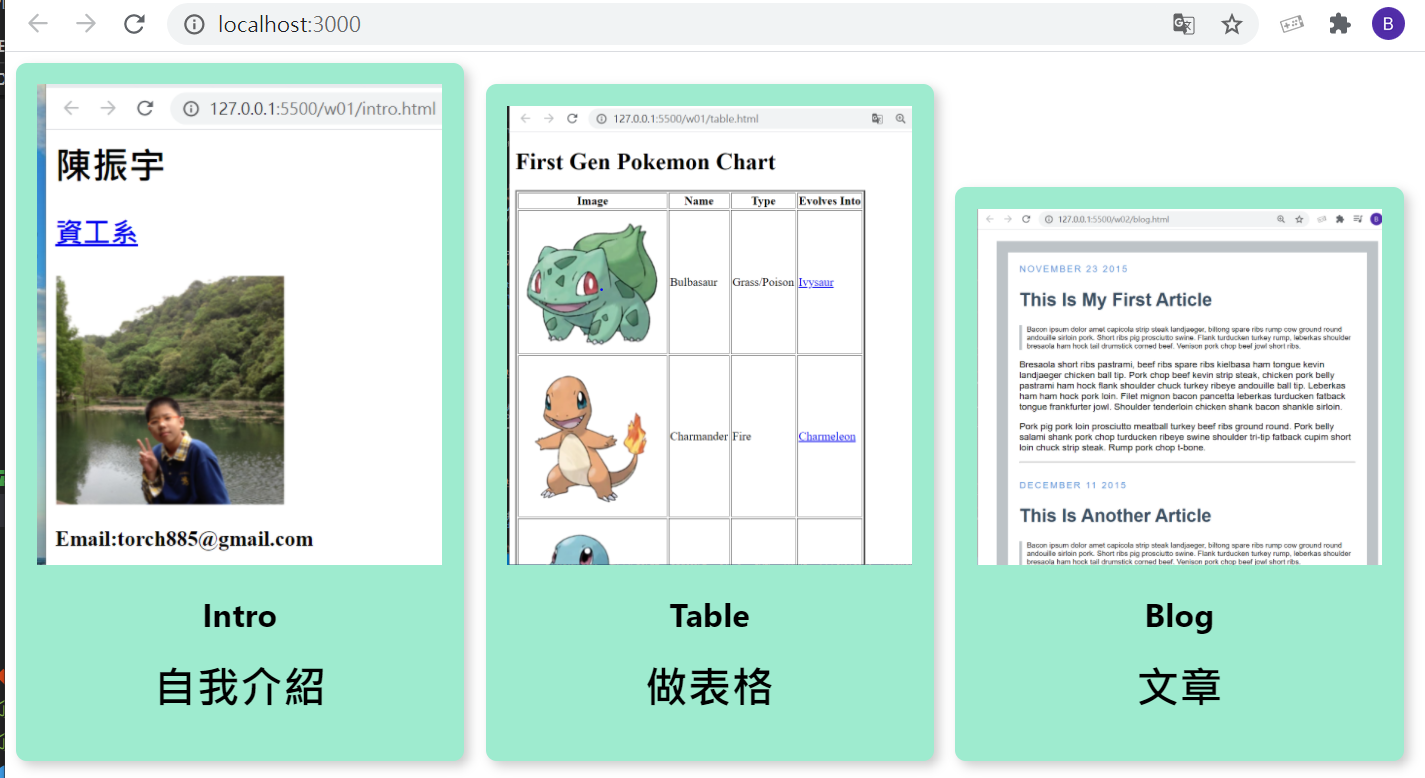
<h1>{username}</h1>

</div>

)

}

export default Card;



7.建立demoList.js並把3張卡片移到demoList

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import DemoList from './DemoList\_99';

import \* as serviceWorker from './serviceWorker';

import 'tachyons';

import { demo } from './Demo\_99';

ReactDOM.render(

<div>

<DemoList demo={demo} />

</div>,

document.getElementById('root'));

// If you want your app to work offline and load faster, you can change

// unregister() to register() below. Note this comes with some pitfalls.

// Learn more about service workers: https://bit.ly/CRA-PWA

serviceWorker.unregister();

import React from 'react'

import Card from './Card\_99';

const DemoList = ({demo}) => {

return (

<div>

<Card name={demo[0].name} username={demo[0].username} img={demo[0].img}/>

<Card name={demo[1].name} username={demo[1].username} img={demo[1].img}/>

<Card name={demo[2].name} username={demo[2].username} img={demo[2].img}/>

</div>

)

}

export default DemoList;

8.將卡片增加到10張

import React from 'react'

import Card from './Card\_99';

const DemoList = ({demo}) => {

return (

<div>

{demo.map((user,i) => {

return(

<Card key={i} name={demo[i].name} username={demo[i].username} img={demo[i].img}/>

)

})}

</div>

)

}

export default DemoList;



9.建立App.js並把CardList搬進去

import React from 'react';

import DemoList from './DemoList\_99';

import { demo } from './Demo\_99';

const App = () => {

return <DemoList demo={demo} />

}

export default App;

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import App from './App\_99';

import \* as serviceWorker from './serviceWorker';

import 'tachyons';

ReactDOM.render(

<App />

,

document.getElementById('root'));

// If you want your app to work offline and load faster, you can change

// unregister() to register() below. Note this comes with some pitfalls.

// Learn more about service workers: https://bit.ly/CRA-PWA

serviceWorker.unregister();

10.建立SearchBox.js加入標題和搜尋

import React from 'react';

import DemoList from './DemoList\_99';

import { demo } from './Demo\_99';

import SearchBox from './SearchBox\_99';

const App = () => {

return(

<div className='tc'>

<h1>Demos</h1>

<SearchBox />

<DemoList demo={demo} />

</div>

)

}

export default App;

import React from 'react'

const SearchBox = () => {

return (

<div className='pa2'>

<input

className='pa3 ba b--green bg-lightest-blue'

type='search'

placeholder='search demos'

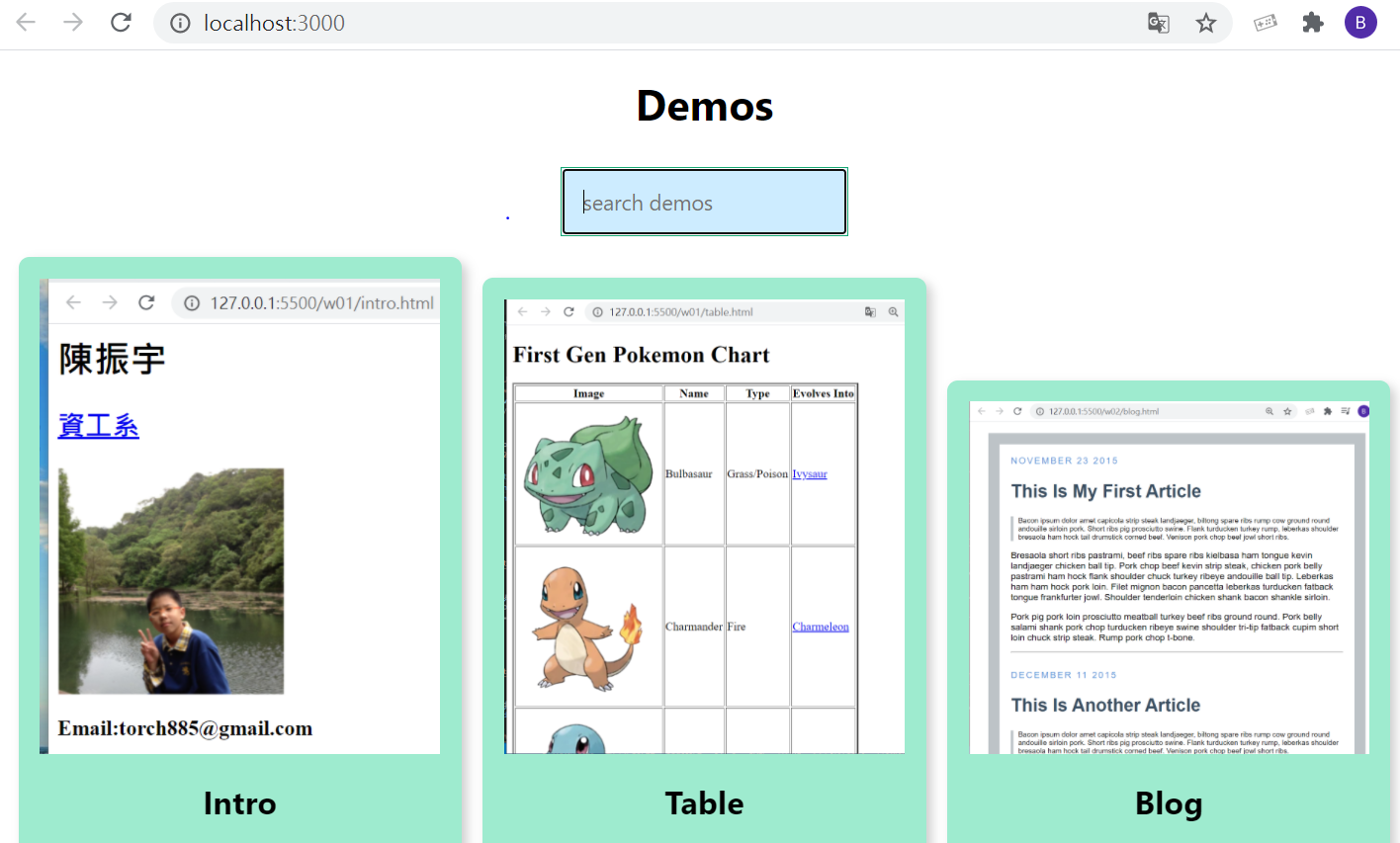
/>

</div>

)

}

export default SearchBox;



11 .add onChange

import React from 'react'

const SearchBox = ({ searchfield, searchChange }) => {

return (

<div className='pa2'>

<input

className='pa3 ba b--green bg-lightest-blue'

type='search'

placeholder='search demos'

onChange={searchChange}

/>

</div>

)

}

export default SearchBox;

12.建立關鍵字搜尋

import React, { Component } from 'react';

import DemoList from './DemoList\_99';

import { demo } from './Demo\_99';

import SearchBox from './SearchBox\_99';

class App extends Component{

constructor(){

super();

this.state={

demo:demo,

searchfield:''

}

}

onSearchChange = event =>{

this.setState({ searchfield: event.target.value })

}

render(){

const filteredDemo = this.state.demo.filter(demo=>{

return demo.name

.toLowerCase()

.includes(this.state.searchfield.toLowerCase())

})

return(

<div className='tc'>

<h1>Demos</h1>

<SearchBox searchChange={this.onSearchChange} />

<DemoList demo={filteredDemo} />

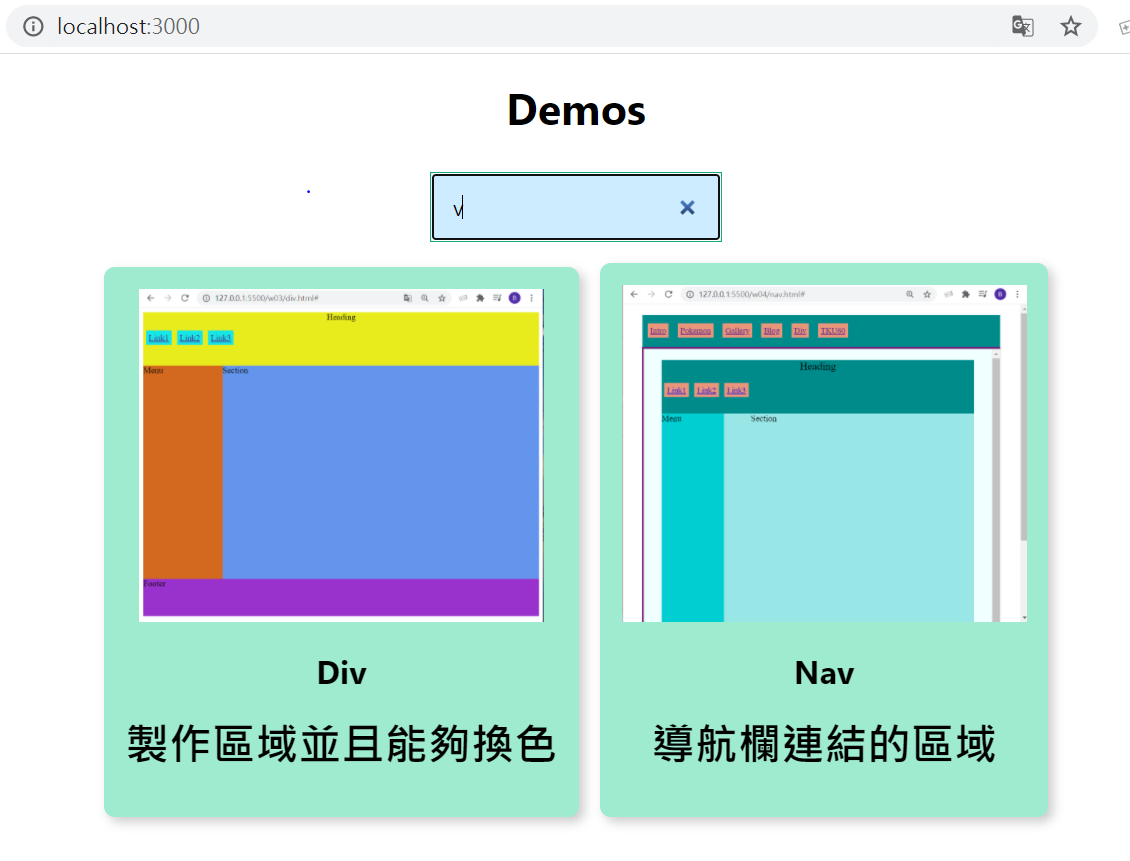
</div>

)

}

}

export default App;



13.改變背景顏色

body {

margin: 0;

padding: 0;

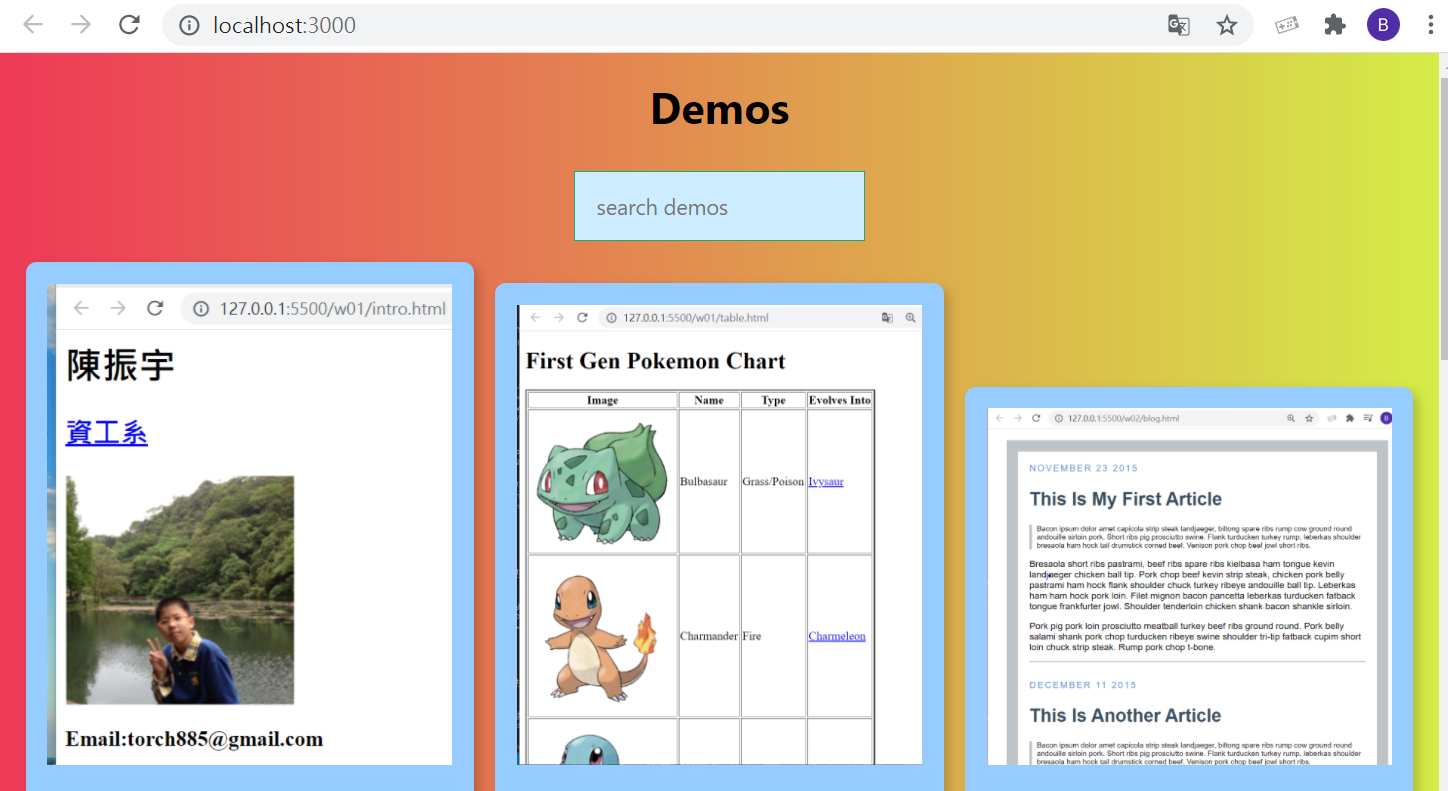
background: linear-gradient(

to left,

rgb(212, 236, 71),

rgb(238, 57, 87) 100%

);



14. 改變字體

@font-face {

font-family: "SEGA LOGO FONT";

font-style: normal;

font-weight: normal;

src: local("SEGA LOGO FONT") url("SEGA.TTF"), format("truetype");

}

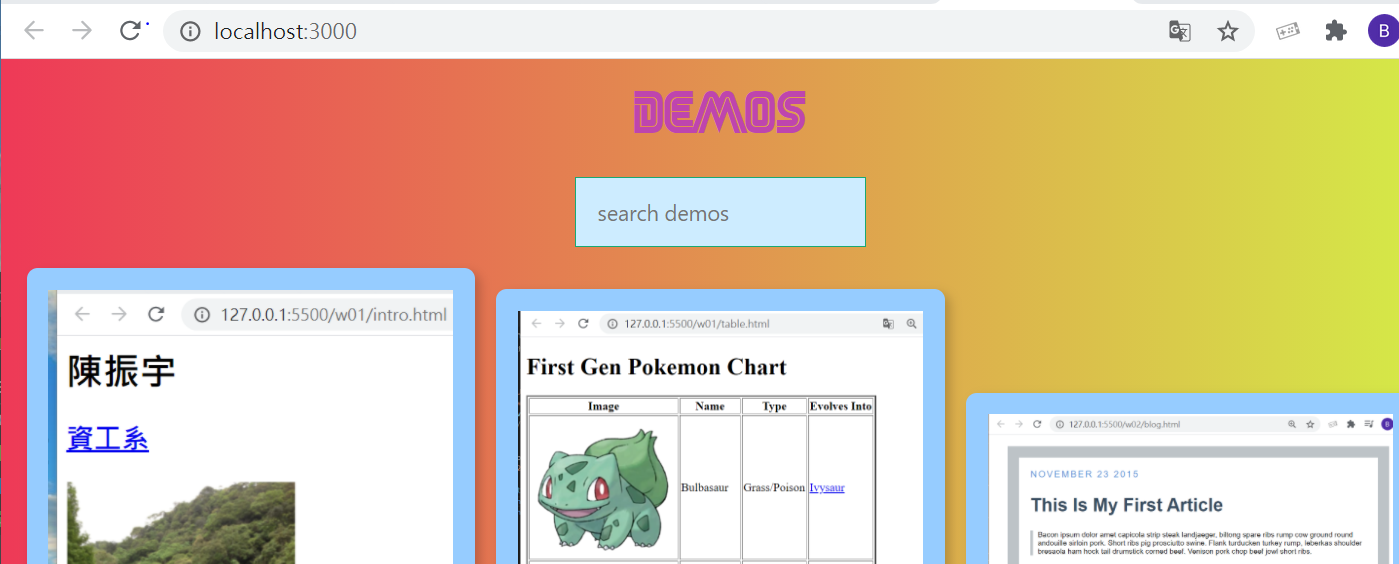
h1 {

font-family: "SEGA LOGO FONT";

font-weight: 200;

color: #be45ae;

}



15. 建立scroll.js 讓DemoListd可以滾動

import React from 'react'

const Scroll = props => {

return (

<div style={{ overflow: 'scroll', border: '1px solid black', height: '800px' }}>

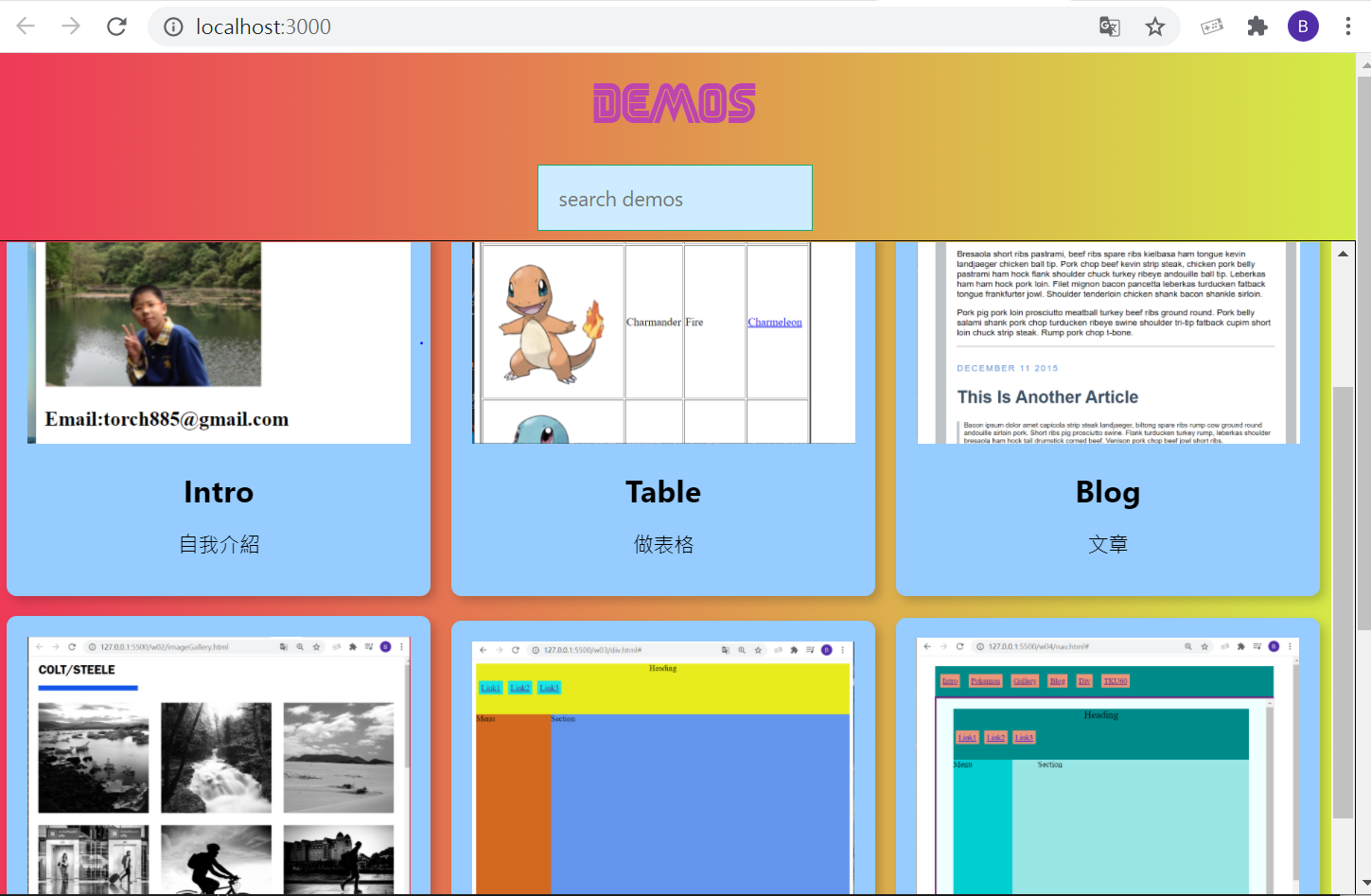
{props.children}

</div>

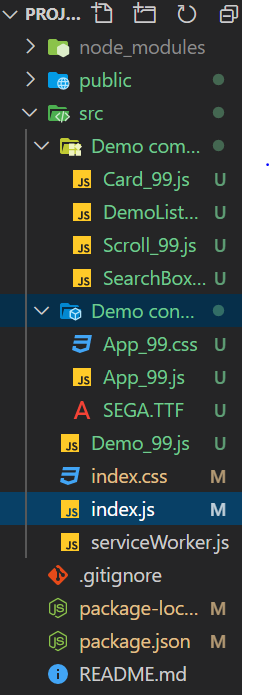
)

}

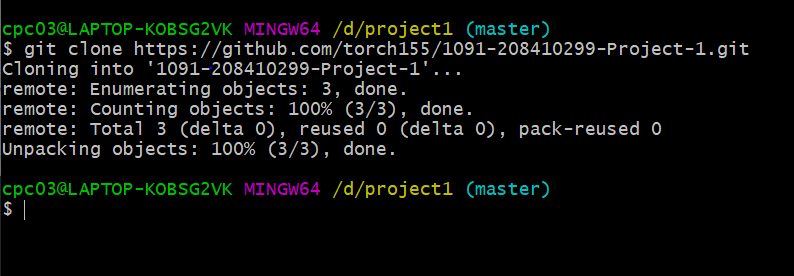
export default Scroll;



16.建立Demo components和Demo containers資料夾



17.連接到github



18. 增加homepage

{

"name": "project1",

"homepage": "https://github.com/torch155/1091-208410299-Project-1.git",

"version": "0.1.0",

"private": true,

"dependencies": {

"@testing-library/jest-dom": "^4.2.4",

"@testing-library/react": "^9.5.0",

"@testing-library/user-event": "^7.2.1",

"gh-pages": "^3.1.0",

"react": "^16.14.0",

"react-dom": "^16.14.0",

"react-scripts": "3.4.3",

"tachyons": "^4.12.0"

},

19.install gh-pages

"gh-pages": "^3.1.0",

"react": "^16.14.0",

"react-dom": "^16.14.0",

"react-scripts": "3.4.3",

"tachyons": "^4.12.0"

},

"scripts": {

"start": "react-scripts start",

"build": "react-scripts build",

"test": "react-scripts test",

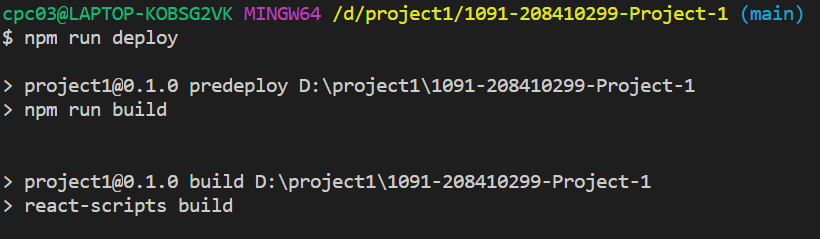
"eject": "react-scripts eject",

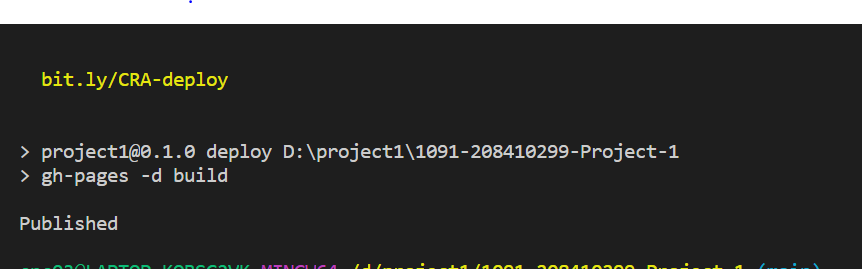
"predeploy": "npm run build",

"deploy": "gh-pages -d build"

},

20. running npm run deploy





問題

1.步驟3時,無法連接到圖片

2.持續Loading無法連到demo

3.

解決方法

1.一直重試路徑

2.沒解決

3.