TO INSTALL THE REACT SET UP -

* First install the node js application
* Got o command promt(node prompt or windows prompt)
* Run this command
* “npm install –g create-react-app “
* “create-react-app my-app”
* Now you have setup your first react application name as “my-app”
* Type cd my-app
* “npm start”
* Your application will run on the server and will open on the web browser.

NPM COMMAND -

“npm start”- to start the development server

“npm run build” – Bundless the app into static files or production.

“npm test” – start the test runner.

“npm run ejects”- it removes the tool and copies all the build dependencies, configuration files and scripts into the app directory . If you do this you cannot go back.

CREATE A COMPONENT

To create a component you need to import two basic things:

* Import react and reactDom the syntax is as below.
* import React from 'react';
* import ReactDOM from 'react-dom';
* now create a class which extends the react component. Like-
* class Hello extends React.Component{
* render(){
* return(
* <h1>this is wish</h1>
* );
* }
* }
* Now render this component in reactdom model the syntax is –
* ReactDOM.render(<component/>, mount)
* It takes two parameter first is the component name which you have created like Hello
* And another is mount place in which react-dom model you want to display.
* So complete syntax will look like-
* ReactDOM.render(<Hello/>,document.getElementById('root'));
* So the complete code of index.js file look like as :

You have created a component name as Hello.(Component name should always be first letter in Caps).

import React from 'react';

import ReactDOM from 'react-dom';

class Hello extends React.Component{

render(){

return(

<h1>this is wish</h1>

);

}

}

ReactDOM.render(<Hello/>,document.getElementById('root'));

CREATE MULTIPLE COMPONENT -

The syntax is look like as-

Date: 14/05/2018

CREATE A TODO APPLICATION IN REACT-

First create a app folder of “todo” app and navigate it into your code editor start the server and edit the code. Instruction is given below.

The folder hierarchy –

TODOAPP-

* node\_modules
* public
* src

- components

-header.js

-todoinput.js

-todoitems.js

-todoinput.css

-todoitems.css

-App.css

-App.js

-index.css

-index.js

-registerServiceWorker.js

- .gitignore

- package-lock.json

- package.json

App.js file-

import React from ‘react’;

import ‘./App.css’;

import Header from ‘./components/header’;

import TodoInput from ‘./components/todoinput’;

import TodoItems from ‘./components/todoitems’;

// create an App class which will extends the react components

Class App extends React.Component{

// define the constructor

Constructor(props){

Super(props);

This.state = {

Todos:[

{id:0, text:”wishvanath”},

{id:1, text:”anand”}

],

nextId :2

}// end of state

// bind the function

This.addTodo = this.addTodo.bind(this);

This.removeTodo = this.removeTodo(this);

}// end of constructor

// define the function

addTodo(todoText){

// check in console is method working or not

Console.log(“Text is Adding”);

}

removeTodo(id){

// check in console

Console.log(item is removing”);

}

Render(){

Return(

<div className = “App”>

<div className =”todoWrapper”>

<Header/>// include header component

// include todoInput components

<TodoInput todoText = “” addTodo = {this.addTodo}/>

// show the data of define state under ul tag

<ul>

// show the data using map function

{

This.state.todos.map( (todo) => {<TodoItems todo = {todo} key = {todo.id} id = {todo.id} removeTodo = {this.removeTodo}/>})

} </ul>

</div>

</div>

);

}// end of render method

Export default App;

Header.js Components-

Import React from ‘react’;

Import ‘./App.cs’;

Class Header extends React.Component{

Render(){

Return(

<div>

<h1>React To do Application </h1>

<hr>

</div>

); // end of return  
}// end of render

} // end of component

Export default Header;

Todoinput.js Components-

Import React from ‘react’;

Import ‘./components/todoinput.css’;

Class TodoInput extends React.Componet{

// define the constructor

Constructor(props){

Super(props);

// bind the function which you defined in the input box and on button

This.handleChange = this.handleChange.bind(this);

This.addTodo = this.addTodo.bind(this);

This.state = {

Value : “” ,// empty value

}// end of state

}// end of constructor

// define the function

handleChange(e){

// set the target value

This.setState({value:e.target.value });

}// end of handle change

addTodo(todo){

// ensure that input box is not empty

If(todo.length > 0){

This.props.addTodo(todo);

This.setState = ({value:””});

}

}//end of addTodo

Render(){

Return(

<div>

<input type = “text” value = {this.state.value} onChange = {this.handleChange} />

<button type = “submit” onClick = “{() = > this.addTodo(this.state.value)} >Submit </button>

</div>

); // end of return

}// end of render method

}// end of component

Export default TodoInput;

Todoitems.js Components-

Import React from ‘react’;

Import ‘./components/todoitems.css;

Class TodoItems extends React.Component{

// define the constructor

Constructor(props){

Super(props);

// no need to bind the function of removeTodo

} // end of constructor

// define the reomoveTodo()

removeTodo(id){

this.props.removeTodo(id);

} // end of remove todo

Render(){

Return(

<div className =”todoWrapper”>

<button className = “ remove\_btn” onClick = {(e) => this.removeTodo(this.props.id)}>Remove</button>{this.props.todo.text}

</div>

); // end of return

} // end of render

}// end of component

Export default TodoItems;

NOW YOU ARE READY TO RUN YOUR APPLICATION

TO RUN A REACT APPLICATION ON DIFFERENT PORT NO -

Edit the file –

“node\_modules”

* react\_scripts

- scripts

- start.js

// Tools like Cloud9 rely on this.

const DEFAULT\_PORT = parseInt(process.env.PORT, 10) || 3002;

const HOST = process.env.HOST || '0.0.0.0';

TO DEVELOP A SINGLE PAGE APPLICATION -

Create a basic app folder and add two more node modules-

The syntax is like as –

* “npm install –save react-bootstarp”
* “npm install – save react-router-dom”

Now copy the link and paste it into the index.html file –

The link is –

<!-- Latest compiled and minified CSS -->

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">

<!-- Optional theme -->

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.min.css" integrity="sha384-rHyoN1iRsVXV4nD0JutlnGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXwl/Sp" crossorigin="anonymous">

* Rename the file “App.js” as “App.jsx” to make it auto complete when you will write the code.
* Create assest folder inside the public folder to keep your image and any file
* Public>assets.
* Now create some components like-

Src>components>

-home.jsx

-home.css

-about.jsx

-about.css

-news.jsx

-news.css

-customnavbar.jsx

-customnavbar.css

Work on the home.jsx file

import React, { Component } from 'react';

// to import the link to navigate from ract-router components

import { Link } from 'react-router-dom';

import { Jumbotron, Image, Button, Row, Col, Grid} from 'react-bootstrap';

// import the home css file

import './home.css';

import CustomNavbar from './customnavbar';

import BodyPanel from './bodypanel';

class Home extends Component {

render() {

return (

<Grid>

<div>

<Row className ="page\_header">

<Col md = {3}>

<Link to = "/">

<Image src= "/assets/site\_logo.png" className ="site\_image" />

</Link>

</Col>

<Col md = {9} className ="site\_title">

<h2>Single Page Application</h2>

</Col>

</Row>

<Row>

<CustomNavbar />

</Row>

<Row>

<BodyPanel />

</Row>

</div>

{/\* <Link to = "/about">

<Button bsStyle = " btn btn-primary">About</Button>

</Link>

<Link to = "/news">

<Button bsStyle = "btn btn-danger">News</Button>

</Link> \*/}

</Grid>

);

}

}

export default Home;

Work on App.jsx file

import React, { Component } from 'react';

// import the browser router dependencies from react-router-dom

import { BrowserRouter as Router, Route} from 'react-router-dom';

import './App.css';

// import all the components which you have created

import Home from './components/home';

import About from './components/about';

import News from './components/news';

class App extends Component {

render() {

return (

<Router>

<div>

<Route exact path = "/" component = {Home} />

<Route path = "/about" component = {About} />

<Route path = "/news" component = {News} />

</div>

</Router>

);

}

}

export default App;

Work on CustomNavbar.jsx

import React, { Component } from 'react';

import { Navbar, Nav, Row, NavItem, MenuItem, NavDropdown} from 'react-bootstrap';

import './customnavbar.css';

class CustomNavbar extends Component {

render() {

return (

<div className = "demo\_nav">

<Navbar inverse>

<Navbar.Header>

<Navbar.Brand>

<a href="/">Home</a>

</Navbar.Brand>

</Navbar.Header>

<Nav>

<NavItem eventKey={1} href="/News">

News

</NavItem>

<NavItem eventKey={2} href="/About">

About

</NavItem>

<NavDropdown eventKey={3} title="Service" id="basic-nav-dropdown">

<MenuItem eventKey={3.1}>Web Application</MenuItem>

<MenuItem divider />

<MenuItem eventKey={3.2}>Digital Marketing</MenuItem>

<MenuItem divider />

<MenuItem eventKey={3.3}>Google Map API</MenuItem>

<MenuItem divider />

<MenuItem eventKey={3.4}>Business Solution</MenuItem>

</NavDropdown>

</Nav>

</Navbar>

</div>

);

}

}

export default CustomNavbar;

Work on bodypanel.jsx –

import React, { Component } from 'react';

import {Grid, Row, Col, Carousel} from 'react-bootstrap'

import './bodypanel.css';

class BodyPanel extends Component {

render() {

return (

<div>

<Grid>

<Row className = "slider\_section">

<Carousel>

<Carousel.Item>

<img alt = "Slider Image" src="/assets/slider\_1.jpg" />

<Carousel.Caption>

<h3>First slide label</h3>

<p>Nulla vitae elit libero, a pharetra augue mollis interdum.</p>

</Carousel.Caption>

</Carousel.Item>

<Carousel.Item>

<img alt = "Slider Image" src="/assets/slider\_2.jpg" />

<Carousel.Caption>

<h3>Second slide label</h3>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>

</Carousel.Caption>

</Carousel.Item>

<Carousel.Item>

<img alt = "Slider Image" src="/assets/slider\_1.jpg" />

<Carousel.Caption>

<h3>Third slide label</h3>

<p>Praesent commodo cursus magna, vel scelerisque nisl consectetur.</p>

</Carousel.Caption>

</Carousel.Item>

</Carousel>

</Row>

<Row className = "body\_text">

<h1>This is the Paragraph text you can include more text here...</h1>

</Row>

</Grid>

</div>

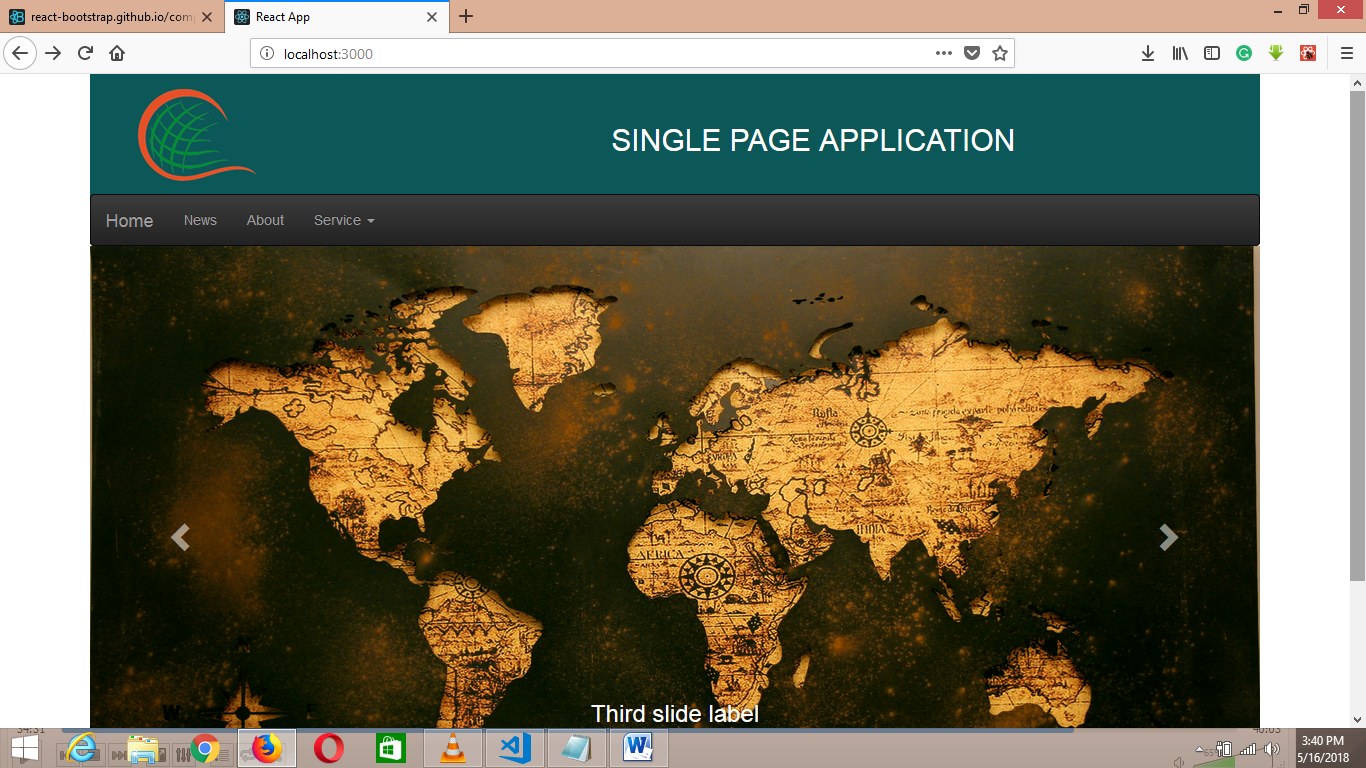
);

}

}

export default BodyPanel;

Application Design –



IMPORTANT NOTES –

* Make css code easy you have to install the react-bootstrap
* To navigate the page link you have to install the
* React router-dom which I have discuss before.
* Study the code properly you can easily develop the single page application in react.

TO DEVELOP A WEATHER APPLICATION IN REACT- (USING API)

* Create a Project folder using npm command like-
* “create-react-app weather-app”
* Now remove the content of the file-

-app.js

-app.css

-delete the logo

Create component folder inside the src.

* Src>components>

-title.js

Work on title.jsx-

import React, { Component } from 'react';

class Title extends Component {

render() {

return (

<div>

<h1>Weather Finder App</h1>

<p>Find out temprature , conditions and more...</p>

</div>

);

}

}

export default Title;

Work on form.jsx components –

import React from 'react';

class Form extends React.Component{

render(){

return(

<div>

this is my form component

</div>

); // end of return

}// end of render

}// end of class

// export it to use in another components

export default Form;

Work on weather.jsx components-

import React from 'react';

class Weather extends React.Component{

render(){

return(

<div>

this is my weather components

</div>

); // end of return function

}//end of render method

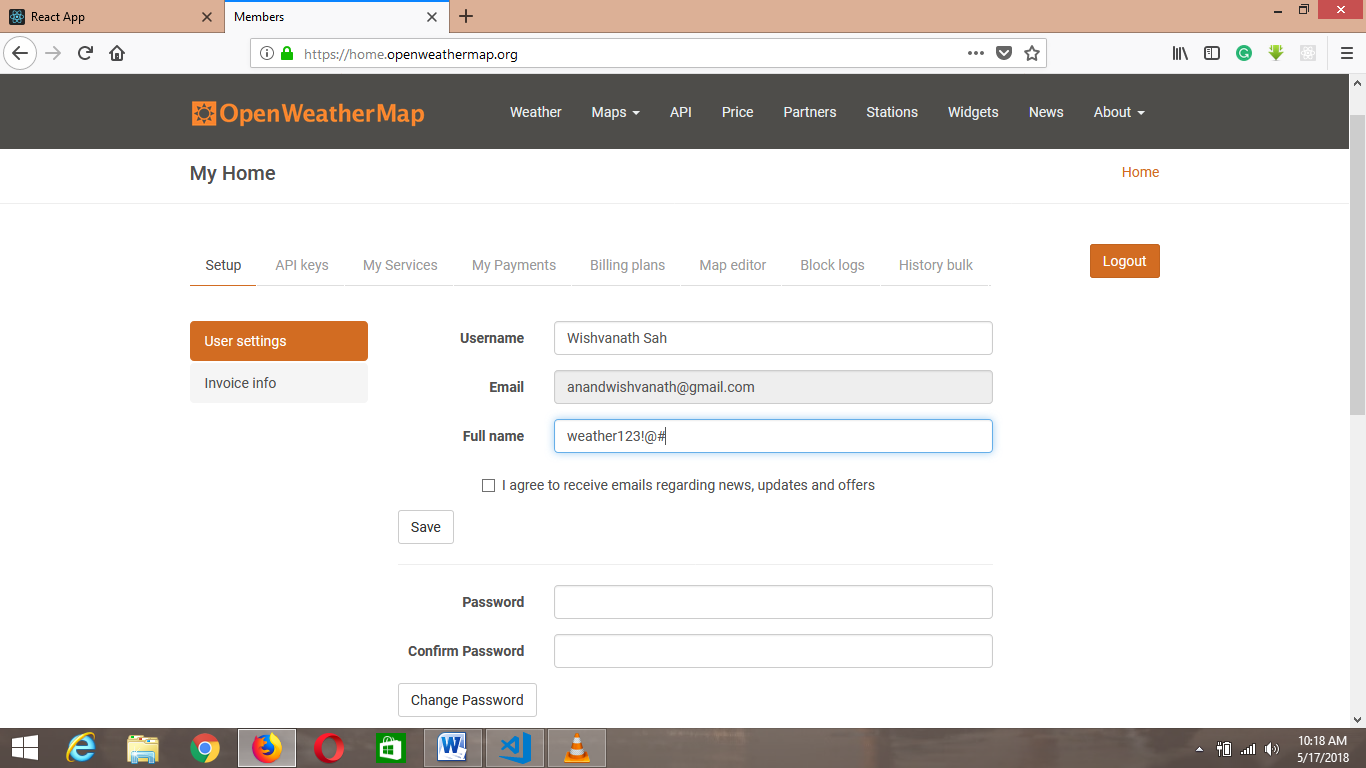
}// end of class

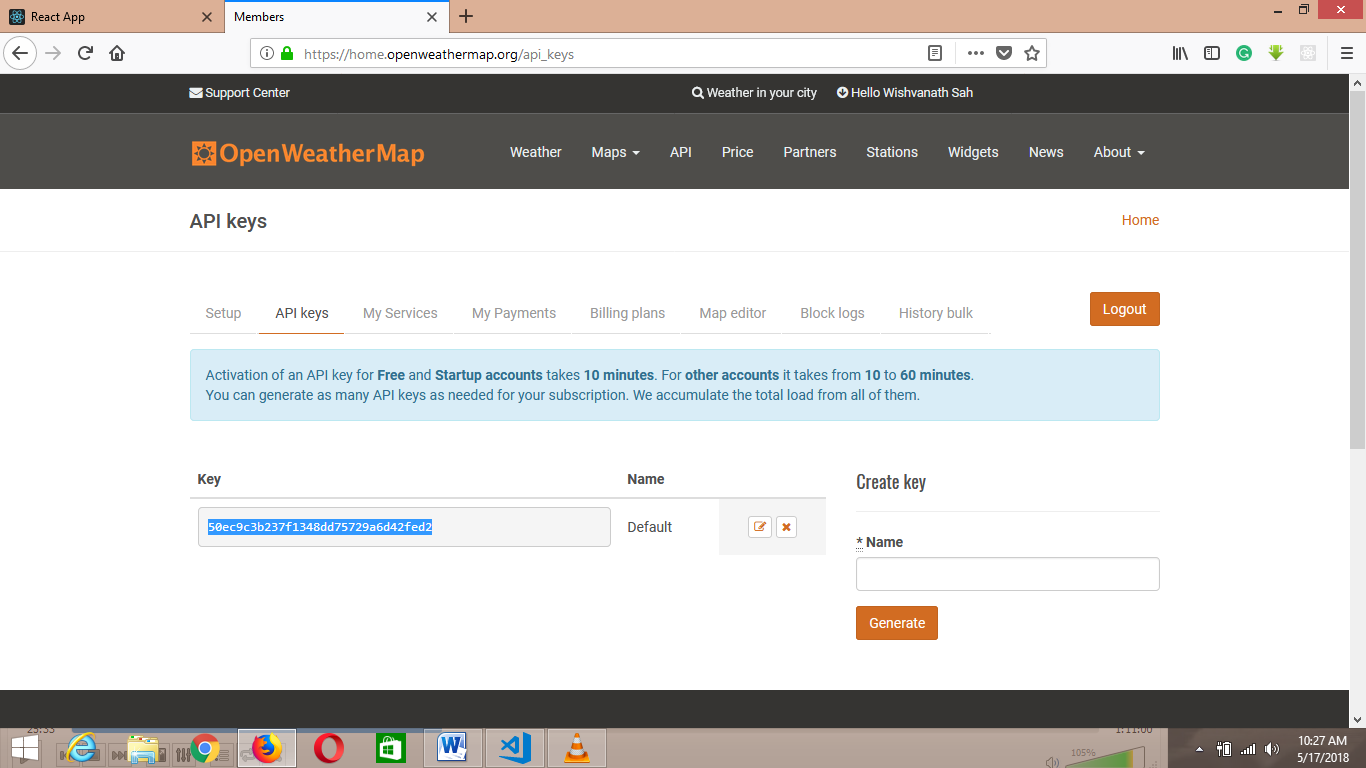
// export this component to use in other componets

export default Weather;

Now create an account on “openweathermap.org” to call an API from URL.

The screen after create Account –





Copy the API key and make it store it in const variable in App.jsx file like-

Work on App.jsx file –

import React, { Component } from 'react';

import './App.css';

import Title from './components/Title'

import Form from './components/Form';

import Weather from './components/Weather';

const API\_KEY = "50ec9c3b237f1348dd75729a6d42fed2";

class App extends Component {

render() {

return (

<div>

<Title />

<Form />

<Weather />

</div>

);

}

}

export default App;

Now search one API Url the link is –

<http://samples.openweathermap.org/data/2.5/weather?q=London,uk&appid=b6907d289e10d714a6e88b30761fae22>

Now create your own method in App.jsx file –

There is two way to create the method

Using the constructor-

constructor(props){

super(props);

// bind the method

this.getWeather = this.getWeather.bind(this);

}

getWeather(){

} // end of getWeather method

And another method is using self inner function like –

Self Inner function allow use to this keyword inside the reder function.

The syntax is like as –

// self Inner function

getWeather = () =>{

}// end of getWeather function

Now to call the Api\_call inside the function –

// self Inner function

getWeather = async () =>{

// fetch the api call

const api\_call = await fetch(`http://samples.openweathermap.org/data/2.5/weather?q=London,uk&appid=b6907d289e10d714a6e88b30761fae22`);

}// end of getWeather function

Now replace the “appid” with your Api key const variable which you have created before-

// self Inner function

getWeather = async () =>{

// fetch the api call

const api\_call = await fetch(`http://samples.openweathermap.org/data/2.5/weather?q=London,uk&appid= ${API\_KEY}`);

}// end of getWeather function

Now after fetching the data from “api\_call” we need to convert this data into the json format.

And check it in console . rember that we have not call this function in an event so –

getWeather = async () =>{

// fetch the api call

const api\_call = await fetch(`http://samples.openweathermap.org/data/2.5/weather?q=London,uk&appid= ${API\_KEY}`);

// convert the api\_call data into the json format

const data = await api\_call.json();

// check the json data into the console

console.log("data");

}// end of getWeather function

Now call the getWeather() in any event like –

Work on the <Form /> component to take the input( means the design part)

To bring the intelligence window in visual code editor press “ctrl + spacebar”

import React from 'react';

class Form extends React.Component{

render(){

return(

<div>

<form onSubmit = {this.props.getWeather}>

<input type = "text" name = "city" placeholder = "City Name" />

<input type = "text" name = "country" placeholder ="Your Country"/>

<button type = "submit">Get Weather</button>

</form>

</div>

); // end of return

}// end of render

}// end of class

// export it to use in another components

export default Form;

now run in browser

Note : you can get some error so you need to run this application in your internet explorer.

Work on App.jx file –

import React, { Component } from 'react';

import './App.css';

import Title from './components/Title'

import Form from './components/Form';

import Weather from './components/Weather';

const API\_KEY = "1ac2a9ad789f1cdd93957ef6c6c4db0b";

class App extends Component {

// self Inner function

getWeather = async (e) =>{

// to prevent the full page refresh

e.preventDefault();

const city = e.target.elements.city.value;

const country = e.target.elements.country.value;

// chek the value in console

console.log(city);

console.log(country);

// fetch the api call

const api\_call = await fetch(`http://samples.openweathermap.org/data/2.5/weather?q=${city},${country}&appid= ${API\_KEY}`);

// const api\_call = await fetch('http://samples.openweathermap.org/data/2.5/weather?q=London,uk&appid=b6907d289e10d714a6e88b30761fae22');

// convert the api\_call data into the json format

const data = await api\_call.json();

// check the json data into the console

console.log(data);

}// end of getWeather function

render() {

console.log(this.getWeather);

return (

<div>

<Title />

{/\* to call the getWeather function we need to set props so props is nothing like properties in html which gives to use in another component \*/}

<Form getWeather = {this.getWeather} />

{/\* now this getWeather method is able to use in form component using the props concept \*/}

<Weather />

</div>

);

}

}

export default App;

here we are cheking the data in console now need to display this data into the UI-

so to show the data in UI we need to define the state

there is two way to define the state-

using constructor method –

// define the state

constructor(props){

super(props);

this.state = {

}

}

Another Method –

// define the state

state = {

temperature: undefined,

city: undefined,

country: undefined,

humidity: undefined,

description: undefined,

error: undefined

}

Now set the value of this state-

import React, { Component } from 'react';

import './App.css';

import Title from './components/Title'

import Form from './components/Form';

import Weather from './components/Weather';

const API\_KEY = "1ac2a9ad789f1cdd93957ef6c6c4db0b";

class App extends Component {

// define the state

state = {

temperature: undefined,

city: undefined,

country: undefined,

humidity: undefined,

description: undefined,

error: undefined

}

// self Inner function

getWeather = async (e) =>{

// to prevent the full page refresh

e.preventDefault();

const city = e.target.elements.city.value;

const country = e.target.elements.country.value;

// chek the value in console

// console.log(city);

// console.log(country);

// fetch the api call

const api\_call = await fetch(`http://samples.openweathermap.org/data/2.5/weather?q=${city},${country}&appid= ${API\_KEY}`);

// const api\_call = await fetch('http://samples.openweathermap.org/data/2.5/weather?q=London,uk&appid=b6907d289e10d714a6e88b30761fae22');

// convert the api\_call data into the json format

const data = await api\_call.json();

// check the json data into the console

console.log(data);

// now set the state value

this.setState({

temperature: data.main.temp,

city: data.name,

country: data.sys.country,

humidity: data.main.humidity,

description: data.weather[0].description,

error: ""

});

}// end of getWeather function

render() {

console.log(this.getWeather);

return (

<div>

<Title />

{/\* to call the getWeather function we need to set props so props is nothing like properties in html which gives to use in another component \*/}

<Form getWeather = {this.getWeather} />

{/\* now this getWeather method is able to use in form component using the props concept \*/}

<Weather />

</div>

);

}

}

export default App;

now pass the value of this state in weather component to show the data using the props concept the code will look like this-

import React, { Component } from 'react';

import './App.css';

import Title from './components/Title'

import Form from './components/Form';

import Weather from './components/Weather';

const API\_KEY = "1ac2a9ad789f1cdd93957ef6c6c4db0b";

class App extends Component {

// define the state

state = {

temperature: undefined,

city: undefined,

country: undefined,

humidity: undefined,

description: undefined,

error: undefined

}

// self Inner function

getWeather = async (e) =>{

// to prevent the full page refresh

e.preventDefault();

const city = e.target.elements.city.value;

const country = e.target.elements.country.value;

// chek the value in console

// console.log(city);

// console.log(country);

// fetch the api call

const api\_call = await fetch(`http://samples.openweathermap.org/data/2.5/weather?q=${city},${country}&appid= ${API\_KEY}`);

// const api\_call = await fetch('http://samples.openweathermap.org/data/2.5/weather?q=London,uk&appid=b6907d289e10d714a6e88b30761fae22');

// convert the api\_call data into the json format

const data = await api\_call.json();

// check the json data into the console

console.log(data);

// now set the state value

this.setState({

temperature: data.main.temp,

city: data.name,

country: data.sys.country,

humidity: data.main.humidity,

description: data.weather[0].description,

error: ""

});

}// end of getWeather function

render() {

console.log(this.getWeather);

return (

<div>

<Title />

{/\* to call the getWeather function we need to set props so props is nothing like properties in html which gives to use in another component \*/}

<Form getWeather = {this.getWeather} />

{/\* now this getWeather method is able to use in form component using the props concept \*/}

<Weather

temperature= {this.state.temperature}

city = {this.state.city}

country = {this.state.country}

humidity = {this.state.humidity}

description = {this.state.description}

error = {this.state.error}

/>

</div>

);

}

}

export default App;

Now work on the <Weather />

Now we have to access the app.js props into the weather components the code will look like this –

import React from 'react';

class Weather extends React.Component{

render(){

return(

<div>

{/\* now call the state value using the concept of the props \*/}

Temperature:{this.props.temperature}

City Name:{this.props.city}

Country Name: {this.props.country}

Humidity: {this.props.humidity}

Desceription:{this.props.description}

</div>

); // end of return function

}//end of render method

}// end of class

// export this component to use in other componets

export default Weather;

Now the problem is we are getting the empty string value on the form

So omit the string value until the press getweather button.

We can not use the if else operator in react instead of that we can use the conditional operater the syntax is like ---

(condition) ? statement1:statement2;

Or we can use the and operater(&&)

import React from 'react';

class Weather extends React.Component{

render(){

return(

<div>

{/\* now call the state value using the concept of the props \*/}

{/\* <p>Temperature:{this.props.temperature} </p>

<p>City Name:{this.props.city}</p>

<p>Country Name: {this.props.country}</p>

<p> Humidity: {this.props.humidity}</p>

<p>Desceription:{this.props.description}</p> \*/}

{/\* to omit the empty sting on the screen use the concept of and operator \*/}

{this.props.temperature && <p>Temperature:{this.props.temperature} </p>}

{this.props.country && <p>Country Name: {this.props.country}</p>}

{this.props.city && <p>City Name:{this.props.city}</p> }

{this.props.humidity && <p> Humidity: {this.props.humidity}</p>}

{this.props.description && <p>Desceription:{this.props.description}</p>}

</div>

); // end of return function

}//end of render method

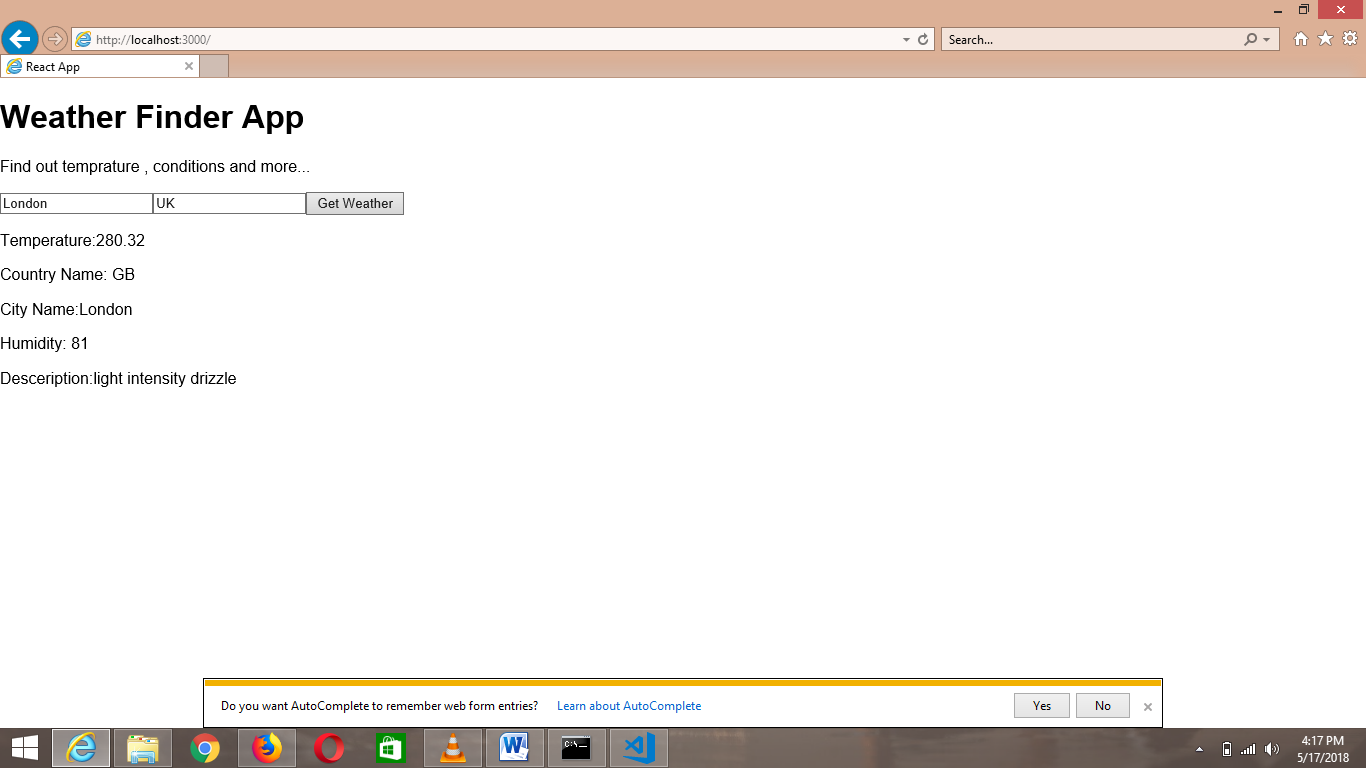
}// end of class

// export this component to use in other componets

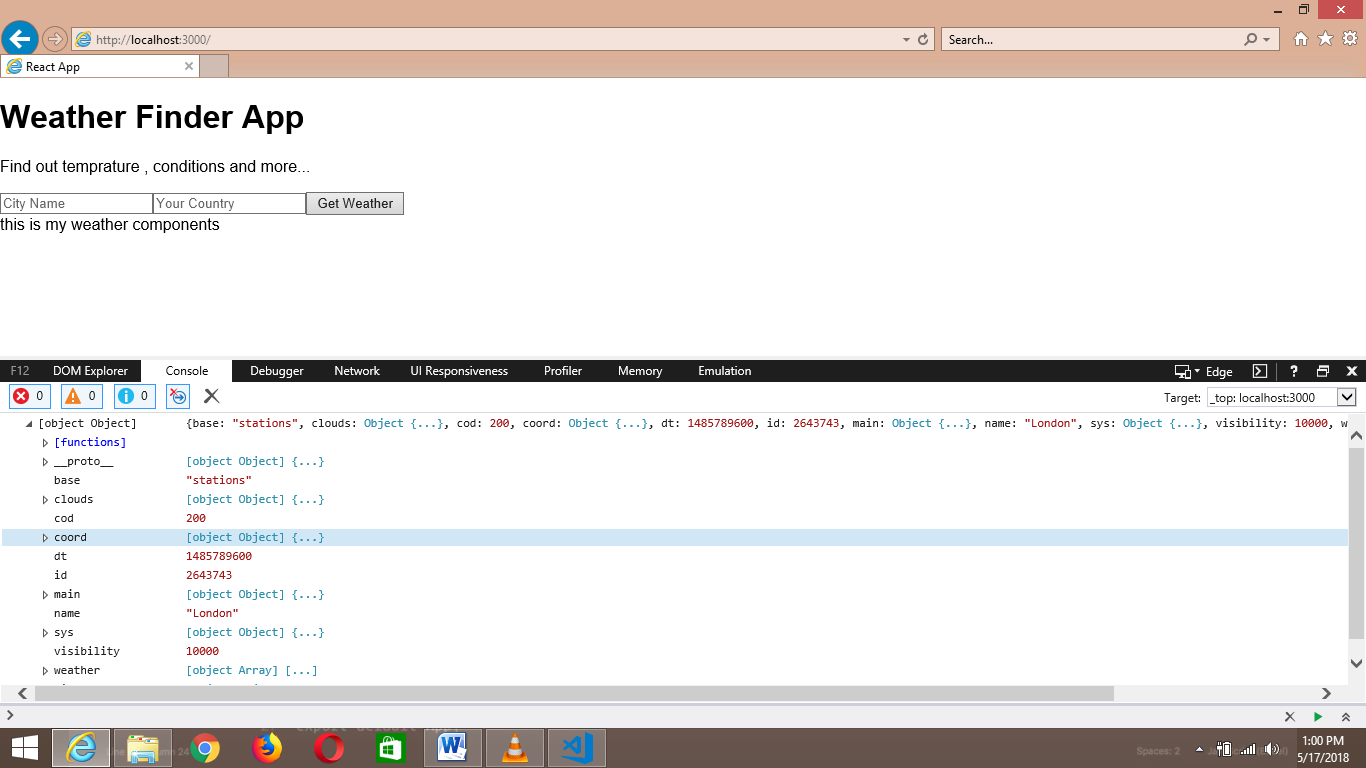
export default Weather;

Now your application is ready to go -

The screen short is –



Console screen-



So the next thing is we need these data in UI not in console