React Assignment:1

1. Install NodeJs, Visual studio code
2. create a Welcome component to greet the user using a javascript function, with a property called as name. Apply some css styling.Use the component in App component.

import React, { Component } from 'react';

export default class Greet extends Component {

constructor(props){

    super(props);

}

  render() {

    return (

      <div>

           <h1 className='name'> Hello {this.props.name}!</h1>

      </div>

    );

  }

}

1. Create a Component to add two numbers.

import React from 'react'

import './SumCss.css'

export default class Sum extends React.Component {

  constructor(props){

    super(props)

    this.state = { num1: '', num2: '', total: '' }

  }

  exe1(){

    console.log("Num1: ", this.state.num1, " - Num2: ", this.state.num2)

    this.setState({ total: parseInt(this.state.num1) + parseInt(this.state.num2) })

    console.log(this.state)

  }

  render() {

    return(

      <div className="mainContainer">

        <h1>Add Two Numbers</h1>

        <input type="text" className="inputStyle" value={this.state.num1} onChange={ (eve) => { this.setState({ num1: eve.target.value }) } }/>

        <br/><br/>

        <input type="text" className="inputStyle" value={this.state.num2} onChange={ (eve) => { this.setState({ num2: eve.target.value })} } />

        <br/><br/>

        <button className="button" onClick={()=>{this.exe1()}} >Add Numbers</button>

        <br/><br/>

        <input type="text" className="outStyle" value={this.state.total} />

        <br/><br/>

      </div>

    )

  }

}

1. Modify the card component to add a props age and perform the validation,to check it’s a number.

import React, { Component } from 'react'

export default class Card extends Component {

  constructor(props){

    super(props);

    this.state={name:'',age:0,errorName:'',errorAge:''};

  }

  handleSubmit=(e)=>{

    if(this.state.errorAge==='' && this.state.errorName===''){

      alert('Your name is: '+this.state.name +'\nYour age is: '+this.state.age);

    }

    else{

      alert('Fill the form correctly!');

    }

  }

  handleChange=(e)=>{

    const name=e.target.name;

    const value=e.target.value;

    let error='',errors='';

    if (name==="name" && value.trim()===''){

      error=<strong style={{color:'red'}}>\*{name} is required</strong>;

    }

    if(name==="age" && this.state.name.trim()===''){

      error=<strong style={{color:'red'}}>\*Name is required</strong>;

    }

    if (value<=0){

      errors=<strong style={{color:'red'}}>\*Age should be greater than 0</strong>;

    }

    this.setState({errorName:error,errorAge:errors});

    this.setState({

      [name]:value

    });

  }

  render() {

    return (

      <form onSubmit={this.handleSubmit}>

        <fieldset>

          <label for="name">Enter your name: </label>

          <input type="text" id="name"

                  name="name"

                  onChange={this.handleChange} />

           {this.state.errorName}

                  <br/><br/>

          <label for="age">Enter your age: </label>

          <input type="number" id="age"

                  name="age"

                  onChange={this.handleChange}/>

            {this.state.errorAge}

                  <br/><br/>

          <button>Submit</button>

        </fieldset>

      </form>

    )

  }

}

1. What is Virtual DOM in ReactJS

DOM stands for ‘Document Object Model’. In simple terms, it is a structured representation of the HTML elements that are present in a webpage or web-app. DOM represents the entire UI of your application. The DOM is represented as a tree data structure. It contains a node for each UI element present in the web document. It is very useful as it allows web developers to modify content through JavaScript, also it being in structured format helps a lot as we can choose specific targets and all the code becomes much easier to work with.