|  |  |  |
| --- | --- | --- |
| **Name: Sriram R** | **SRN: PES1UG20CS435** | **Section: H** |
| **Date: 11-11-2021** | **Unit 3 Assignment Exercise** |
| PROBLEM STATEMENT1 (for odd number SRNs): | | |
| Create and display an order detail list for any 4 items which will include the product name, product description, price and quantity. When an onClick event is triggered on a button the quantity has to increase and the total cost has to be calculated and displayed with the update of items quantity.  For styling: Add styling by creating a style object variable the same way as we create a JavaScript. | | |
| OBJECTIVE | | |
| The objective of this exercise is to test the student on front end frame work React JS. It evaluates the student’s knowledge React Js components, ReactJS props and state, ReactJS Lists and Keys, React JS events | | |
| PREREQUISITE | | |
| In order to complete this exercise, the student needs to understand the fundamentals of HTML,CSS, and JavaScript | | |
| SAMPLE SCREENSHOT OF OUTPUT (Just for your reference) | | |
| SAMPLE1: | | |
| PROGRAM | | |
| <html>  <head>  <title>Hello world</title>  <script crossdomain src="https://unpkg.com/react@16/umd/react.development.js"> </script>  <script crossdomain src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"> </script>  <script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"> </script>  </head>  <style>  #div1{  background-color:white;  border:0.5px solid yellow;  padding:5px;  margin:4px;  float:left;  width:30%;  }  #h4{  background-color:white;  border:0.5px solid yellow;  padding:5px;  margin:4px;  float:left;  width:30%;  position:absolute;  top:500px;  left:500px  }  </style>  <body style = "background-color:black">  <div id = "root"></div>  </body>  <script type = "text/babel">  class Mycomponent extends React.Component  {  constructor()  {  super()  this.ipref = React.createRef()  this.ipref1 = React.createRef()  this.ipref2 = React.createRef()  this.ipref3 = React.createRef()  this.state = {value:0}  }  render()  {  return (<div><h1>Order Summary</h1>  <div id = "div1">  <h3>Gersheys chocolate syrup</h3>  <p><b>Description:&nbsp;</b>A chocolate syrup with a name that taunts copyright enforcement laws  <br/><br/><b>Price:&nbsp;</b>200 ZWD (pre inflation era)  <br/><br/><b>Quantity:</b></p>  <p ref = {this.ipref}>0</p>  <button onClick = {this.fun1}>+</button>  </div>  <div id = "div1">  <h3>Dererro Mutella</h3>  <p><b>Description:&nbsp;</b>An exceptionally unhealthy chocolate spread made of 30% palm oil whose health risks are greatly outweighed by its heart stopping taste  <br/><br/><b>Price:&nbsp;</b>500 ZWD  <br/><br/><b>Quantity:</b></p>  <p ref = {this.ipref1}>0</p>  <button onClick = {this.fun2}>+</button>  </div>  <div id = "div1">  <h3>Polonium 209</h3>  <p><b>Description:&nbsp;</b>Best consumed at 2 am when trying to complete a web tech assignment due at 9 the same day thanks to the programmers next-gen time management skills  <br/><br/><b>Price:&nbsp;</b>300 ZWD (discount courtesy : Alexander Litvinenko)  <br/><br/><b>Quantity:</b></p>  <p ref = {this.ipref2}>0</p>  <button onClick = {this.fun3}>+</button>  </div>  <div id = "div1">  <h3>One pound bread</h3>  <p><b>Description:&nbsp;</b>Godhi hittu with warm water and a fungus that makes it fluffy when baked  <br/><br/><b>Price:&nbsp;</b>100000 ZWD (post inflation era lol)  <br/><br/><b>Quantity:</b></p>  <p ref = {this.ipref3}>0</p>  <button onClick = {this.fun4}>+</button>  </div>  <div id = "h4">Total: <b> {this.state.value} ZWD</b></div>  </div>)  }  fun1=()=>  {  var a = this.ipref.current.innerHTML  this.ipref.current.innerHTML = ++a  this.setState((prevState) => ({value:prevState.value+200}))  }  fun2 = () =>  {  var c = this.ipref1.current.innerHTML  this.ipref1.current.innerHTML = ++c  this.setState((prevState) => ({value:prevState.value+500}))  }  fun3 = () =>  {  var d = this.ipref2.current.innerHTML  this.ipref2.current.innerHTML = ++d  this.setState((prevState) => ({value:prevState.value+300}))  }  fun4 = () =>  {  var e = this.ipref3.current.innerHTML  this.ipref3.current.innerHTML = ++e  this.setState((prevState) => ({value:prevState.value+100000}))  }  }  ReactDOM.render(<Mycomponent/>,document.getElementById("root"))  </script>  </html> | | |
| SCREENSHOT OF YOUR OUTPUT | | |
|  | | |