|  |  |  |  |
| --- | --- | --- | --- |
| **Lab No** | 07 | **Reg. No** | 211798 |
| **Student Name** | Hassan Shahzad | **Section** | C |

|  |
| --- |
| Solution Task 1 |
| import React from 'react';  import Parent from './parent'  import './App.css';  function App() {    return (      <div>        <Parent></Parent>      </div>    );  }  export default App;  import React, {Component}  from 'react';  import ChildComponent from './child'  class ParentComponent extends Component {  constructor(props){  super(props)  this.state={ parentName:'Parent' }  this.greetParent=this.greetParent.bind(this)  }  greetParent() {  alert('Hello' +this.state.parentName)  }  render(){ return(<div><ChildComponent greetHandler={this.greetParent} /></div>  )}}  export default ParentComponent  import React from 'react';  function ChildComponent(props) {  return (  <div>  <button onClick={props.greetHandler}>greet parent</button>  </div>  )}  export default ChildComponent; |

|  |
| --- |
| Output |
|  |

|  |
| --- |
| Solution Task 3 |
| import React,{Component}  from 'react';  class UserGreeting extends Component {  constructor(props)  { super(props)  this.state={isLoggedIn:true} }  render(){  let message  if (this.state.isLoggedIn){ message=<div>Welcome Hassan</div>}      else {  message=<div>Welcome Guest</div> }      return <div>{message}</div>  }  }export default UserGreeting |

|  |
| --- |
| Solution Task 3 |
|  |

|  |
| --- |
| Solution Task 3 |
| import React from 'react';  function NameList() {  const names=['Hassan', 'Shahzad', 'Khan']  const nameList =names.map(name=><h2>{name}</h2>)  return <div> {nameList} </div>  }  export default NameList |

|  |
| --- |
| Solution Task 3 |
|  |

|  |
| --- |
| Solution Task 3 |
| import React,{Component} from 'react';  import styles from './mystyle.module.css';  class Car extends Component {    render() {      return <h1 className={styles.bigblue}>Hello Car!</h1>;    }  }  export default Car  .bigblue {    color: DodgerBlue;    padding: 40px;    font-family: Arial;    text-align: center;  } |

|  |
| --- |
| Solution Task 3 |
|  |

|  |
| --- |
| Solution Task 3 |
| import React from 'react';  import ReactDOM from 'react-dom';  class Header extends React.Component {    constructor(props) {      super(props);      this.state = {favoritecolor: "red"};    }    render() {      return (        <h1>My Favorite Color is {this.state.favoritecolor}</h1>      );    }  }  ReactDOM.render(<Header />, document.getElementById('root')); |

|  |
| --- |
| Solution Task 3 |
|  |

|  |
| --- |
| Solution Task 3 |
| import React from 'react';  import ReactDOM from 'react-dom';  class Header extends React.Component {    constructor(props) {      super(props);      this.state = {favoritecolor: "red"};    }    static getDerivedStateFromProps(props, state) {      return {favoritecolor: props.favcol };    }    render() {      return (        <h1>My Favorite Color is {this.state.favoritecolor}</h1>      );    }  }  ReactDOM.render(<Header favcol="yellow"/>, document.getElementById('root'));  export default Header |

|  |
| --- |
| Solution Task 3 |
|  |

|  |
| --- |
| Solution Task 3 |
| import React from 'react';  import ReactDOM from 'react-dom';  class Header extends React.Component {    render() {      return (        <h1>This is the content of the Header component</h1>      );    }  }  ReactDOM.render(<Header />, document.getElementById('root')); |

|  |
| --- |
| Solution Task 3 |
|  |

|  |
| --- |
| Solution Task 3 |
| import React from 'react';  import ReactDOM from 'react-dom';  class Header extends React.Component {    constructor(props) {      super(props);      this.state = {favoritecolor: "red"};    }    componentDidMount() {      setTimeout(() => {        this.setState({favoritecolor: "yellow"})      }, 1000)    }    render() {      return (        <h1>My Favorite Color is {this.state.favoritecolor}</h1>      );    }  }  ReactDOM.render(<Header />, document.getElementById('root'));  export default Header |

|  |
| --- |
| Solution Task 3 |
|  |

|  |
| --- |
| Solution Task 3 |
| import React from 'react';  import Todos from './todo'  import AddTodo from './addtodo'  class App extends React.Component {    state = {      todos: [        {id: 1, content: 'Binge-watch movies'},        {id: 2, content: 'Win against moiz in fifa'}      ]    }    deleteTodo = (id) => {      const todos = this.state.todos.filter(todo => {        return todo.id !== id      });      this.setState({        todos      });    }      addTodo = (todo)=>{      todo.id = Math.random()      let todos = [...this.state.todos, todo]      this.setState({        todos: todos      })    }    render() {      return (        <div className>          <h1 style={{color:"red", textAlign: "center"}}>Todo List</h1>          <Todos todos={this.state.todos} deleteTodo={this.deleteTodo} />      <AddTodo addTodo={this.addTodo} />          </div>      );    }  }  export default App  import React, {Component} from 'react'  class AddTodo extends Component {      state = {          content: ''      }        handleChange = (e) =>{      this.setState({          content: e.target.value      })      }        handleSubmit = (e) =>{          e.preventDefault()          this.props.addTodo(this.state)          this.setState({              content:''          })      }      render(){          return (              <div>                  <form onSubmit={this.handleSubmit}>                      <label>Add new todo: </label>                      <input type="text" onChange={this.handleChange} value={this.state.content}/>                  </form>              </div>          )      }  }  export default AddTodo  import React from 'react';  const Todos = ({todos, deleteTodo}) => {    const todoList = todos.length ? (      todos.map(todo => {        return (          <ul key={todo.id}>            <li onClick={() => {deleteTodo(todo.id)}}>{todo.content}</li>          </ul>        )      })    ) : (      <p style = {{color: "blue"}}>No work left, yay!!!</p>    );    return (      <div>        {todoList}      </div>    )  }  export default Todos; |

|  |
| --- |
| Solution Task 3 |
|  |