1. Create a sample code snippet that demonstrates passing a callback function from a parent component down to its child(ren) component.

*constructor() {*

*super();*

*this.state = {*

*};*

*this.submitForm = this.submitForm.bind(this);*

*}*

*submitForm(values) {*

*this.setState({ values })*

*}*

*<FormDemo*

*onFormSubmit={this.submitForm}*

*/>*

*import React, { Component } from "react";*

*class FormDemo extends Component {*

*constructor() {*

*super();*

*this.state = {*

*};*

*this.onInputChange = this.onInputChange.bind(this);*

*this.onSubmitForm = this.onSubmitForm.bind(this);*

*}*

*onInputChange(event) {*

*this.setState({*

*[event.target.name]: event.target.value*

*});*

*}*

*onSubmitForm() {*

*this.props.onFormSubmit(this.state)*

*}*

*render() {*

*return (*

*<div>*

*<table>*

*<tr>*

*<td>First Name :</td>*

*<td>*

*<input type="text" name="fname" onChange={this.onInputChange} />*

*</td>*

*</tr>*

*<tr>*

*<td>Last Name :</td>*

*<td>*

*<input type="text" name="lname" onChange={this.onInputChange} />*

*</td>*

*</tr>*

*<tr>*

*<td>*

*<button onClick={this.onSubmitForm}>Submit</button>*

*</td>*

*</tr>*

*</table>*

*</div>*

*);*

*}*

*}*

*export default FormDemo;*

*onSubmitForm() {*

*this.props.onFormSubmit(this.state)*

*}*

*import React, { Component } from "react";*

*import { render } from "react-dom";*

*import FormDemo from "./FormDemo";*

*class App extends Component {*

*constructor() {*

*super();*

*this.state = {*

*};*

*this.submitForm = this.submitForm.bind(this);*

*}*

*submitForm(values) {*

*this.setState({ values })*

*}*

*render() {*

*const { values } = this.state;*

*return (*

*<div>*

*<h2>Passing function to the child component</h2>*

*<hr />*

*<FormDemo*

*onFormSubmit={this.submitForm}*

*/> <hr/>*

*<div>*

*Submitted form values : <br/>*

*First name: {values && values.fname} <br/>*

*Last name: {values && values.lname}*

*</div>*

*</div>*

*);*

*}*

*}*

*render(<App />, document.getElementById("root"));*

*submitForm(values) {*

*this.setState({ values })*

*}*

*render() {*

*const { values } = this.state;*

*return (*

*<div>*

*<h2>Passing function to the child component</h2>*

*<hr />*

*<FormDemo*

*onFormSubmit={this.submitForm}*

*/> <hr/>*

*<div>*

*Submitted form values : <br/>*

*First name: {values && values.fname} <br/>*

*Last name: {values && values.lname}*

*</div>*

*</div>*

*);*

*}*

*<div>*

*Submitted form values : <br/>*

*First name: {values && values.fname} <br/>*

*Last name: {values && values.lname}*

*</div>*

*<tr>*

*<td>*

*<button onClick={() => this.props.onFormSubmit(this.state)}>Submit</button>*

*</td>*

*</tr>*

*<tr>*

*<td>*

*<button onClick={*

*() => this.props.onFormSubmit({*

*userName: 'Test123',*

*password: '123456'*

*})*

*}>*

*Submit*

*</button>*

*</td>*

*</tr>*

1. Explain props-drilling

Prop drilling (also called "threading") refers to the process you have to go through to get data to parts of the React Component tree.