**Lesson 04 Demo 01**

**Creating a React Application with Redux to Access the**

**Global Variables**

**Objective:** To create a React application with Redux, showcasing the utilization of a Redux store to manage global variables

**Tools required:** Node.js and React.js

**Prerequisites:** None

Steps to be followed:

1. Create and set up the React project
2. Configure the store in the index.js file
3. Create a user-defined component
4. Modify the App.js file

**Step 1: Create and set up the React project**

* 1. Open a terminal window and run the following command to create a React application:

**npx create-react-app** **react-redux-gs-variable**

**A screenshot of a computer program

Description automatically generated**

* 1. Open the created React application folder (**react-redux-gs-variable)** in VS Code by clicking on **File** in the top left corner and selecting **Open Folder**

**A screenshot of a computer

Description automatically generated**

* 1. Click on **Open**

**A screenshot of a computer

Description automatically generated**

The folder structure appears as follows:

**A screenshot of a computer program

Description automatically generated**

* 1. Inside the project, open the **TERMINAL** and run the following command to install the required dependencies:

**npm install**

**A screen shot of a computer

Description automatically generated**

**Note**: This command helps to install all required dependencies mentioned in the **package.json** file on the local machine in the form of a **node\_module** folder.

* 1. Open the **package.json** file and view the external dependencies

A screenshot of a computer

Description automatically generated

**Step 2: Configure the store in the index.js file**

1. Inside the **src** folder, create a **reducer.js** file and enter the below code. It contains a normal JavaScript function that takes two parameters: example, **state,** and **action**.

**const initialState = {**

**name :"Admin",**

**customer:{cid:1,name:"Raj",age:21}**

**}**

**// Reducer**

**const reducer = (state = initialState, action) => {**

**// as of now no action.**

**return state;**

**};**

**export default reducer;**

**A computer screen shot of a black screen

Description automatically generated**

1. Inside the **src** folder, create a **store.js** file and enter the below code. This store connects with the reducer to make the state variable a global variable.

**// Store**

**import { legacy\_createStore as createStore} from 'redux'**

**import reducer from './reducer';**

**export const store = createStore(reducer);**

**A screenshot of a computer

Description automatically generated**

**Step 3: Create a user-defined component**

1. Inside the **src** folder, create the **FirstComponent.js** file and enter the below code. It creates a state variable and accesses the react global variable with the help of the **useSelector** hook.

**import { useState } from "react";**

**import { useSelector } from "react-redux";**

**function FirstComponent() {**

**let [name,setState]=useState("John");**

**let globalName = useSelector(gs=>gs.name)**

**return(**

**<div>**

**<h3>First Component</h3>**

**<p>State variable value is {name}</p>**

**<p>Global varilabe value is {globalName}</p>**

**</div>**

**)**

**}**

**export default FirstComponent;**

****

1. Inside the **src** folder, create the **SecondComponent.js** file and enter the below code. It creates a state variable and accesses the react global variable with the help of the **useSelector** hook.

**import { useState } from "react";**

**import { useSelector } from "react-redux";**

**function SeconComponent() {**

**let [name,setState]=useState("Steven");**

**let globalName = useSelector(gs=>gs.name)**

**return(**

**<div>**

**<h3>Second Component</h3>**

**<p>State variable value is {name}</p>**

**<p>Global varilabe value is {globalName}</p>**

**</div>**

**)**

**}**

**export default SeconComponent;**

**A screenshot of a computer program

Description automatically generated**

1. Inside the **src** folder, create a **Customer.js** file and add the below code.It can access global state complex objects in customer components.

**import { useState } from "react";**

**import { useSelector } from "react-redux";**

**function Customer() {**

**let customer = useSelector(gs=>gs.customer)**

**return(**

**<div>**

**<h3>Customer Information</h3>**

**<p>Customer Id {customer.cid} Name is {customer.name} Age is {customer.age}</p>**

**</div>**

**)**

**}**

**export default Customer;**

**A screenshot of a computer

Description automatically generated**

**Step 4. Test the application**

1. In the **App.js** file, import **FirstComponent**, **SecondComponent,** and **CustomerComponent** by adding the following code:

**import logo from './logo.svg';**

**import './App.css';**

**import FirstComponent from './FirstComponent';**

**import SeconComponent from './SecondComponent';**

**import Customer from './Customer';**

**function App() {**

**return (**

**<div className="App">**

**<h3>Main Component</h3>**

**<hr/>**

**<FirstComponent></FirstComponent>**

**<hr/>**

**<SeconComponent></SeconComponent>**

**<hr/>**

**<Customer></Customer>**

**</div>**

**);**

**}**

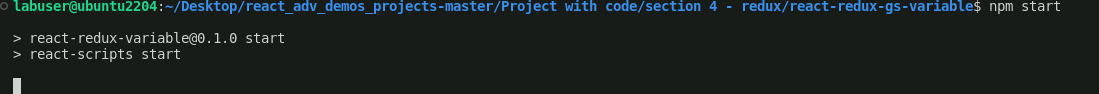
**export default App;**

A screenshot of a computer program

Description automatically generated

1. Open the terminal and run the below command to execute the application:

**npm start**



The output appears as shown below:

A screenshot of a computer

Description automatically generated

With this, you have successfully created a React application with Redux, showcasing the utilization of a Redux store to manage global variables.