**IP Exp 7**

**Shashwat Tripathi  
D15A 64**

**Batch C**

**Aim:** The Basics of React.(React features, React application building, components, props, state, etc.)

React is a widely used JavaScript library for building user interfaces in web applications. It is developed and maintained by Facebook and a community of individual developers. React is renowned for its simplicity, reusability, and efficiency in creating interactive, dynamic, and responsive UIs. Here's an overview of the key concepts and features that form the foundation of React:

**1. React Features:**

Virtual DOM (Document Object Model): React uses a virtual representation of the DOM to optimize updates and minimize re-rendering, which significantly enhances performance.

Component-Based: React is centered around the concept of reusable components, allowing developers to build complex UIs by combining and nesting these building blocks.

Unidirectional Data Flow: React follows a unidirectional data flow, making it easier to track and manage the flow of data within the application.

JSX (JavaScript XML): JSX is a syntax extension for JavaScript that allows developers to write HTML-like code within their JavaScript, making the creation of React components more intuitive.

**2. React Application Building:**

Create React App: A popular tool for creating React applications with a predefined project structure and build configurations.

Webpack and Babel: Often used in conjunction with React, Webpack handles module bundling and Babel transpiles JSX and ES6 code into browser-compatible JavaScript.

**3. Components:**

Functional Components: These are JavaScript functions that return React elements. They are simple, stateless, and used for rendering UI elements.

Class Components: Class components can hold and manage state, which makes them suitable for more complex functionality and lifecycle methods.

**4. Props (Properties):**

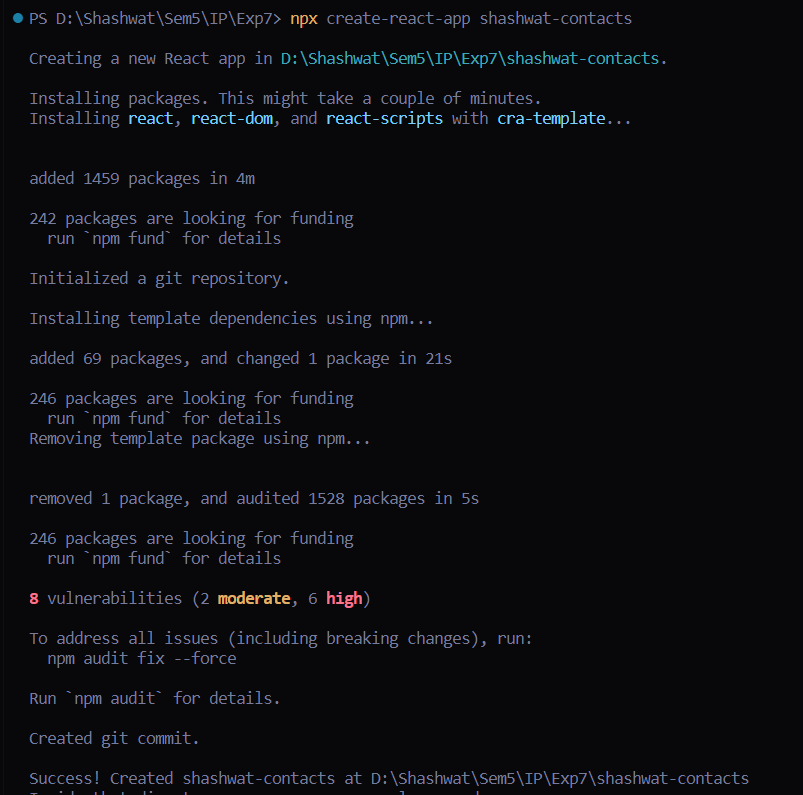
Props are used to pass data from parent components to child components. They are immutable and help create dynamic and flexible UIs.

**5. State:**

State is used to store and manage data within a component. When state changes, React automatically re-renders the component, updating the UI accordingly.

State is typically managed in class components, making them suitable for interactive elements.

**Implementation:**

****

**/src/components/AddNewContact.js**

import React from "react";

import { useState } from "react";

function AddNewContact(props) {

const [data, setData] = useState({ name: "", email: "", phone: "" });

const handleSubmit = (event) => {

event.preventDefault();

const newContact = {

id: Math.random().toString(),

name: data.name,

email: data.email,

phone: data.phone,

};

props.addContact(newContact);

};

const onChange = (event) => {

setData({ ...data, [event.target.name]: event.target.value });

};

return (

<div>

<form onSubmit={handleSubmit}>

<input

type="text"

name="name"

id="name"

value={data.name}

onChange={onChange}

placeholder="Name"

/>

<input

type="email"

placeholder="Email"

name="email"

id="email"

value={data.email}

onChange={onChange}

/>

<input

type="text"

placeholder="Phone"

name="phone"

id="phone"

value={data.phone}

onChange={onChange}

/>

<button type="submit">Add</button>

</form>

</div>

);

}

export default AddNewContact;

**/src/components/ContactList.js**

import React from "react";

class ContactList extends Component {

constructor(props){

super(props);

this.state = {

name: ‘’

phone: ‘’

email: ‘’

}

}

return (

<div>

<h2>Contact List</h2>

<ul>

{props.contacts.map((contact) => (

<li key={contact.id}>

{contact.name} - {contact.email}

<button onClick={() => props.deleteContact(contact.id)}>

Delete

</button>

</li>

))}

</ul>

</div>

);

}

export default ContactList;

**App.js**import "./App.css";

import { useState } from "react";

import AddNewContact from "./components/AddNewContact";

import ContactList from "./components/ContactList";

function App() {

const [contacts, setContacts] = useState([]);

const addContact = (newContact) => {

setContacts([...contacts, newContact]);

};

const deleteContact = (id) => {

const updatedContacts = contacts.filter((contact) => contact.id !== id);

setContacts(updatedContacts);

};

return (

<>

<div align="center">

<AddNewContact addContact={addContact} />

<ContactList contacts={contacts} deleteContact={deleteContact} />

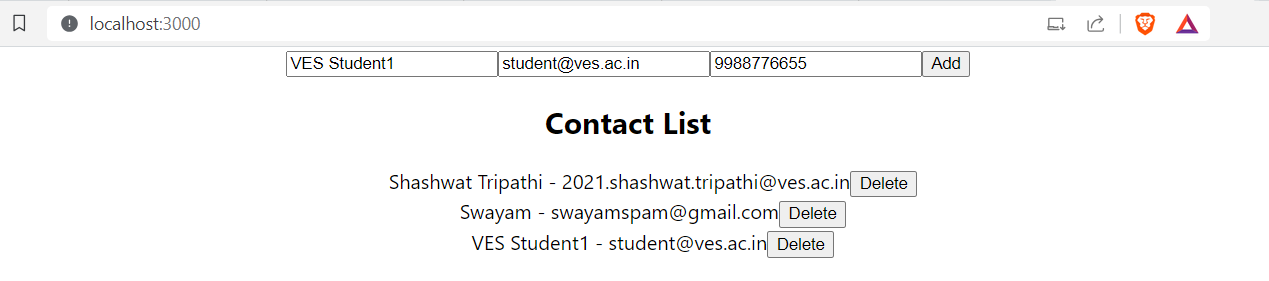
</div>

</>

);

}

export default App;

****

**Conclusion:**

Thus, we have understood the basics of React & implemented React features, React application building, components, props, state, etc.)