**Lesson 07 Demo 01**

**simple-debugger-react-app**

**Objective:** To demonstrate the react simple app which consume rest api. Debug with react devtool using browser with break point.

**Tools required:** Node JS and React JS

**Prerequisites:** HTML, CSS, JavaScript ES5/ES6, Basic React Concept

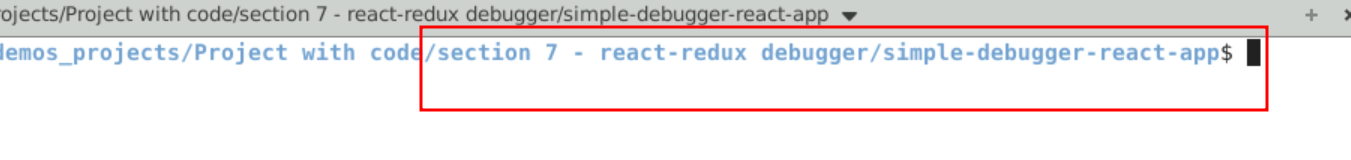
**Note** : All react js project already created with version 18.x with Sample App.js file

**Steps to be followed:**

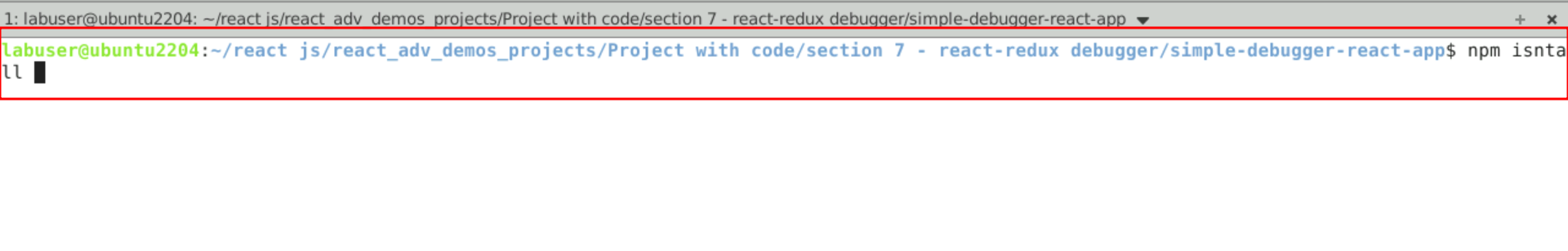
1. Set up for react js project
2. Create user defined component GitHubUser.js file.
3. Create user defined component Posts.js file
4. Now test add both component in App.js file
5. Test the application.

**Step 1: Set up for react js project**

1. Open a terminal window inside a React JS pre-created project ie **simple-debugger-react-app**

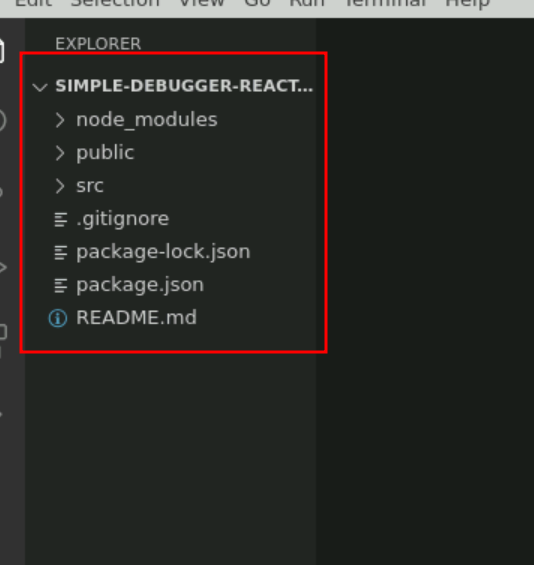


1. Now you need to run the command as **npm install.** This command helps us to installed all required dependencies mention in package.json file in local machine in the form of node\_module folder.

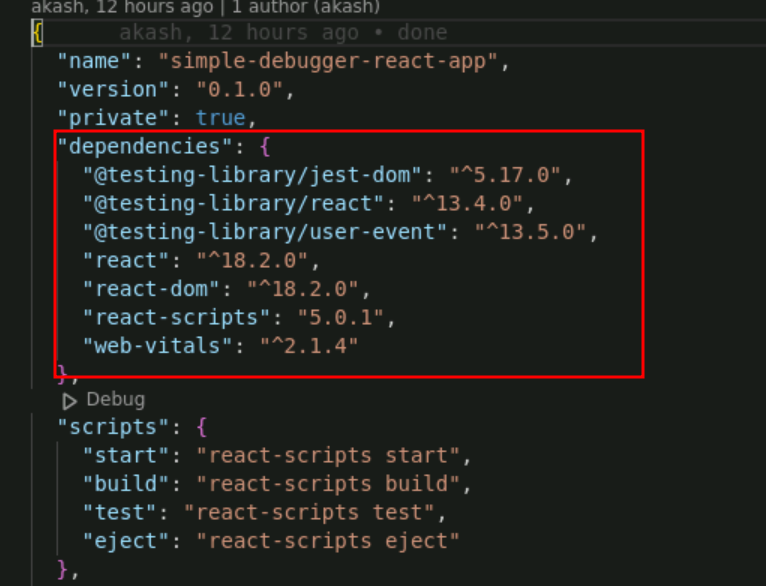


1. Now open **simple-debugger-react-app** folder in VS Code Editor

Note: short cut to open write **code .**

****

1.4 now open package.json file and view external dependencies.

****

**Step 2 : Create user defined component GitHubUser.js file.**

2.1 GitHubUser.js file is responsible to interact with default github account details in current machine with your github follower details. This api we are consuming using fetch() JavaScript function. If you want you can use axios third party modules

**GitHubUser.js**

import React, { useState, useEffect } from 'react';

const GitHubUser = () => {

const [user, setUser] = useState(null);

const [loading, setLoading] = useState(true);

useEffect(() => {

const fetchUser = async () => {

try {

const response = await fetch('https://api.github.com/users/octocat');

const userData = await response.json();

setUser(userData);

setLoading(false);

} catch (error) {

console.error('Error fetching user:', error);

setLoading(false);

}

};

fetchUser();

}, []);

if (loading) {

return <div>Loading...</div>;

}

if (!user) {

return <div>User not found</div>;

}

return (

<div>

<h1>{user.login}</h1>

<img src={user.avatar\_url} alt={user.login} style={{ width: '100px', height: '100px' }} />

<p>Followers: {user.followers}</p>

</div>

);

};

export default GitHubUser;

**Step 3: Create user defined component posts.js file.**

.

3.1 create posts.js file which is responsible to call fake rest api to consume fake post details using fetch() functions.

**Posts.js**

import React, { useState, useEffect } from 'react';

const Posts = () => {

const [posts, setPosts] = useState([]);

const [loading, setLoading] = useState(true);

const [error, setError] = useState(null);

useEffect(() => {

const fetchPosts = async () => {

try {

const response = await fetch('https://jsonplaceholder.typicode.com/posts');

//debugger;

if (!response.ok) {

throw new Error('Network response was not ok');

}

const data = await response.json();

setPosts(data);

setLoading(false);

} catch (error) {

setError(error.message);

setLoading(false);

}

};

fetchPosts();

}, []);

if (loading) {

return <div>Loading...</div>;

}

if (error) {

return <div>Error: {error}</div>;

}

return (

<div>

<h1>Posts</h1>

<ul>

{posts.map((post) => (

<li key={post.id}>

<h3>{post.title}</h3>

<p>{post.body}</p>

</li>

))}

</ul>

</div>

);

};

export default Posts;

**Step 4 :** **Now test add both component in App.js file.**

4.1 In App.js file GitHubUser.js and Posts.js component in App.js.

**App.js**

import logo from './logo.svg';

import './App.css';

import GitHubUser from './GitHubUser';

import Posts from './Posts';

function App() {

return (

<div className="App">

<GitHubUser></GitHubUser>

<hr/>

<Posts></Posts>

</div>

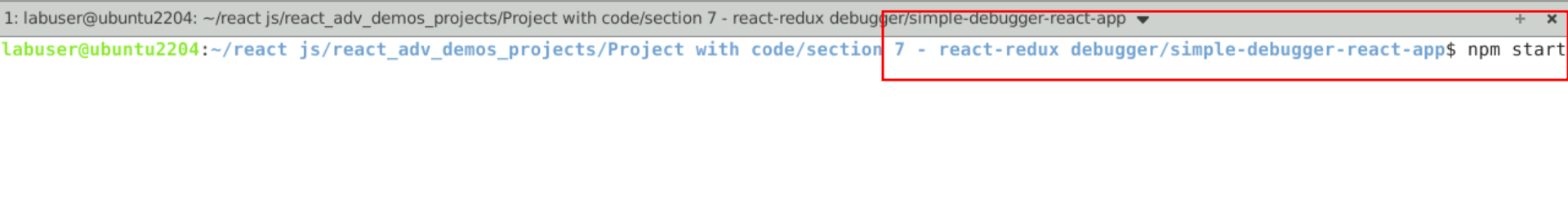
);

}

export default App;

**Step 5 Now we run the application using npm start**

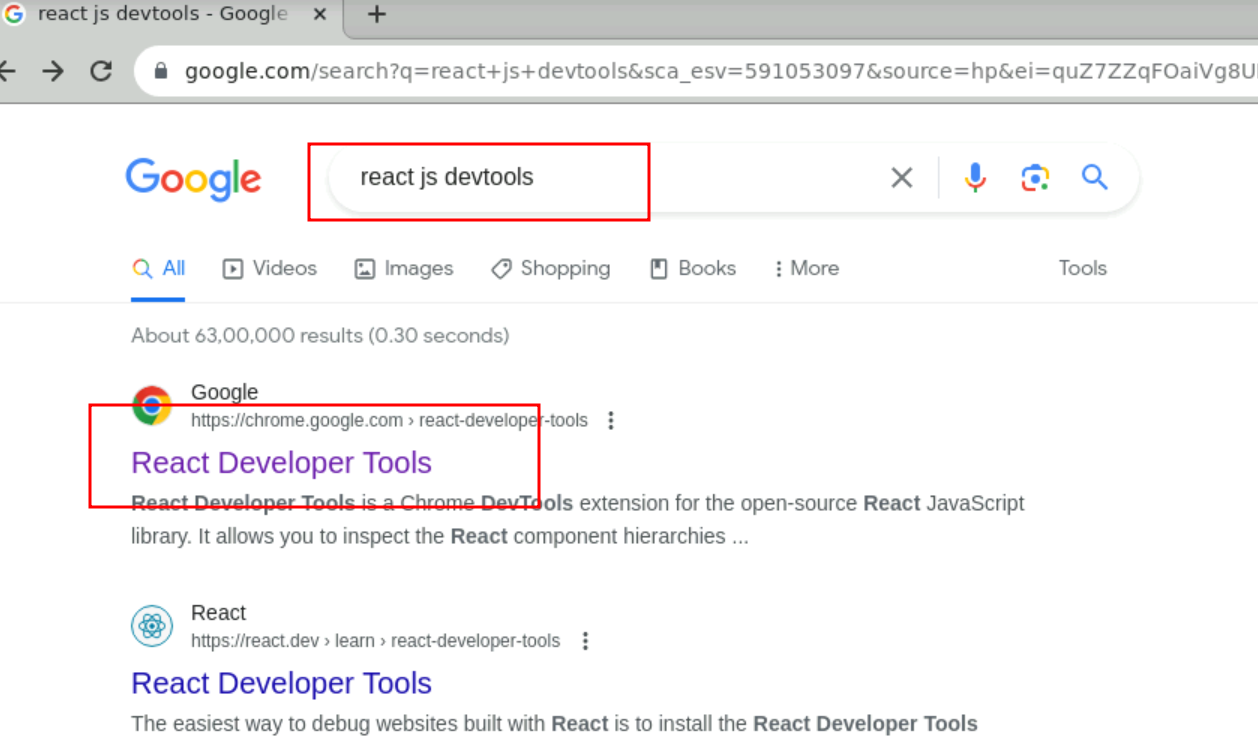
**5.1** Now run the application usiing command as **npm start**



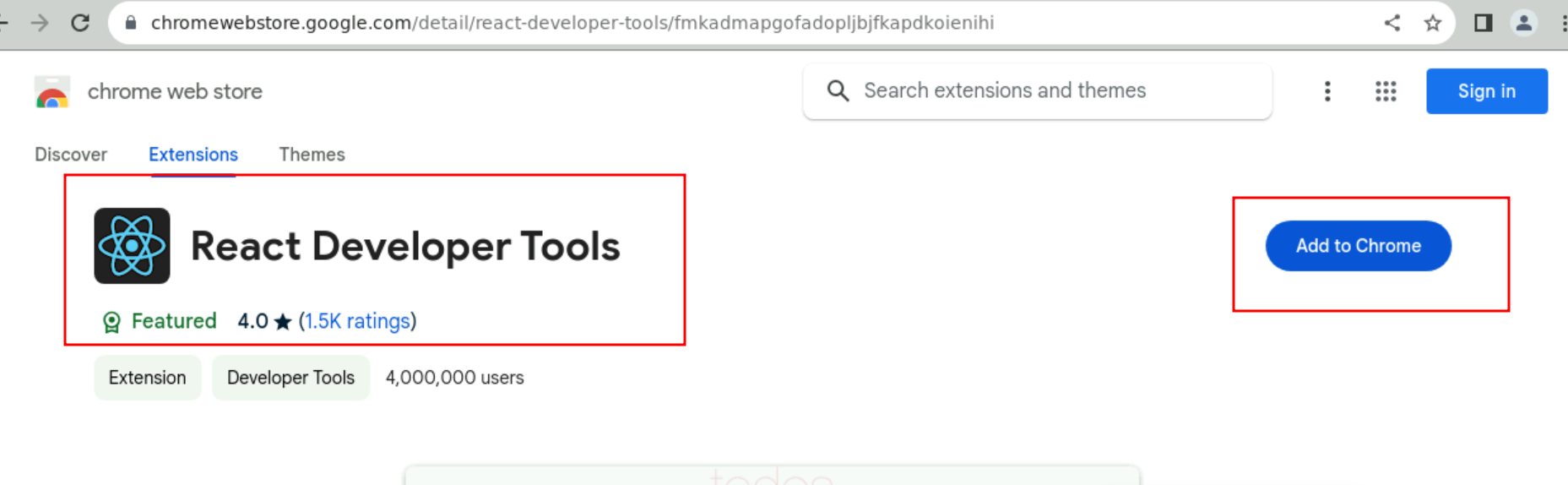
5.2 Open the browser



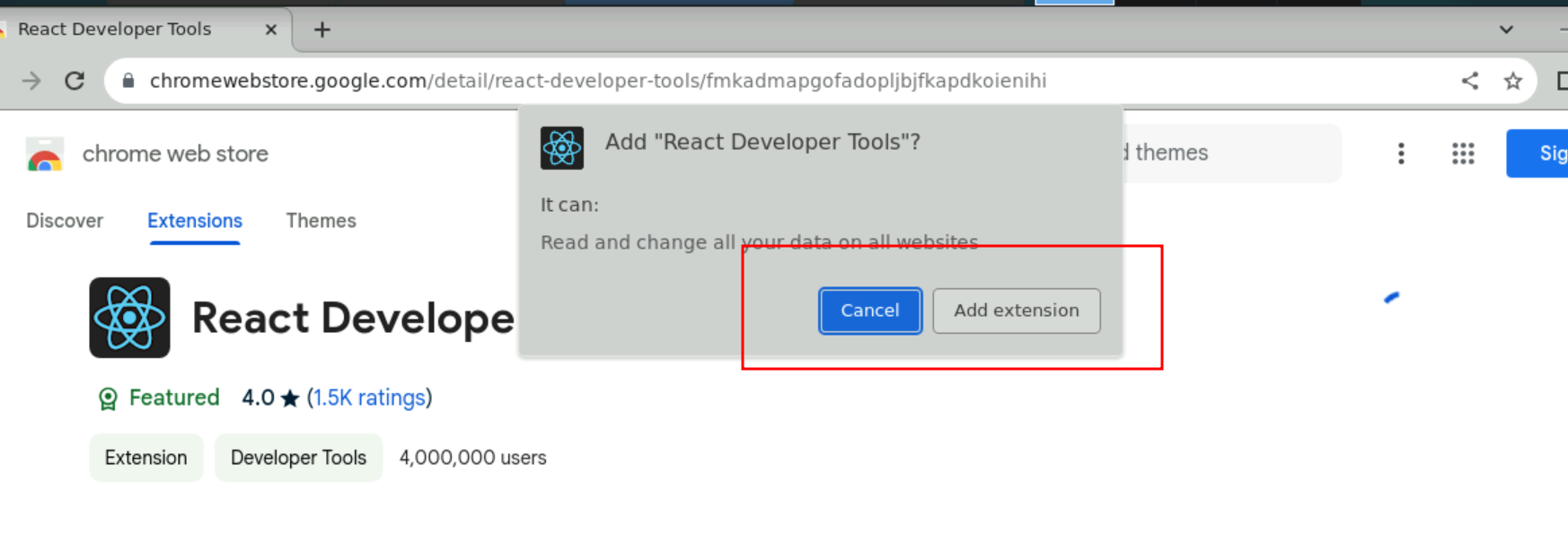
5.2 In Chrome browser. Download react devtools plugins.



**5.3** Now Click link



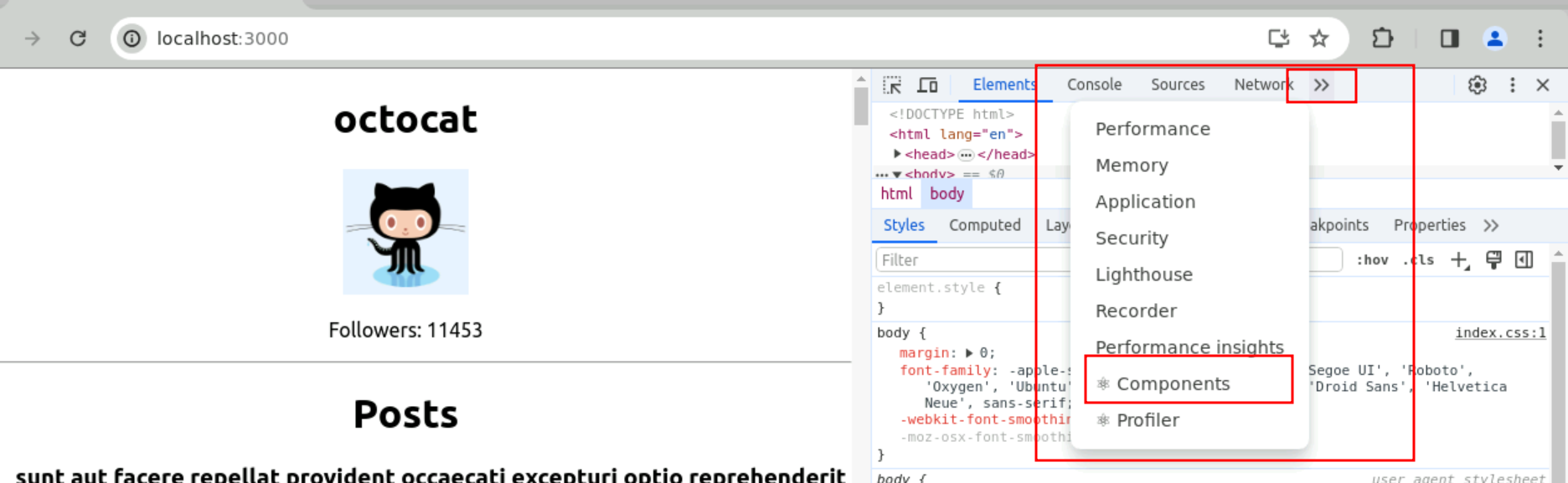
5.4 Now click on add to chrome



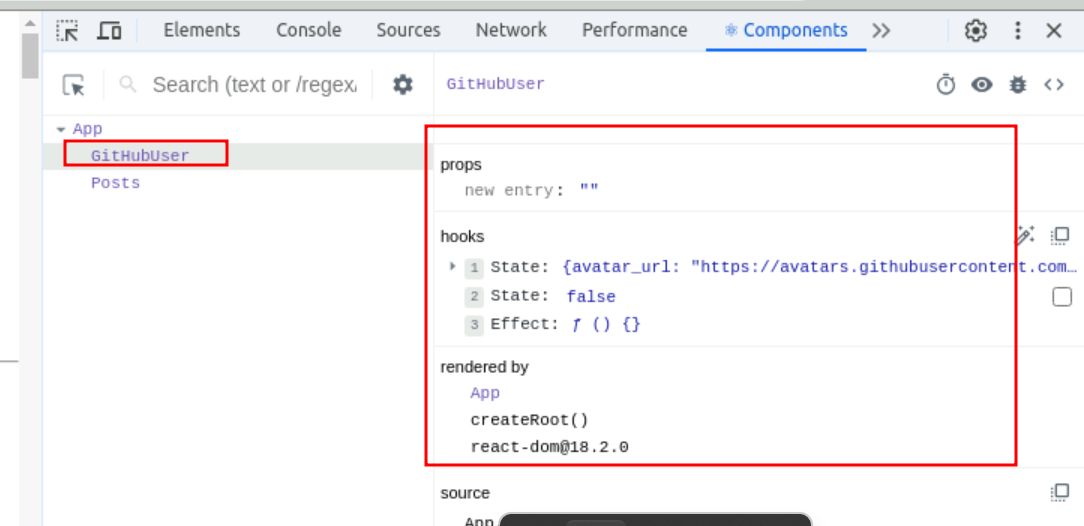
5.5 click on add extension and then restart the browser and open the application once again.



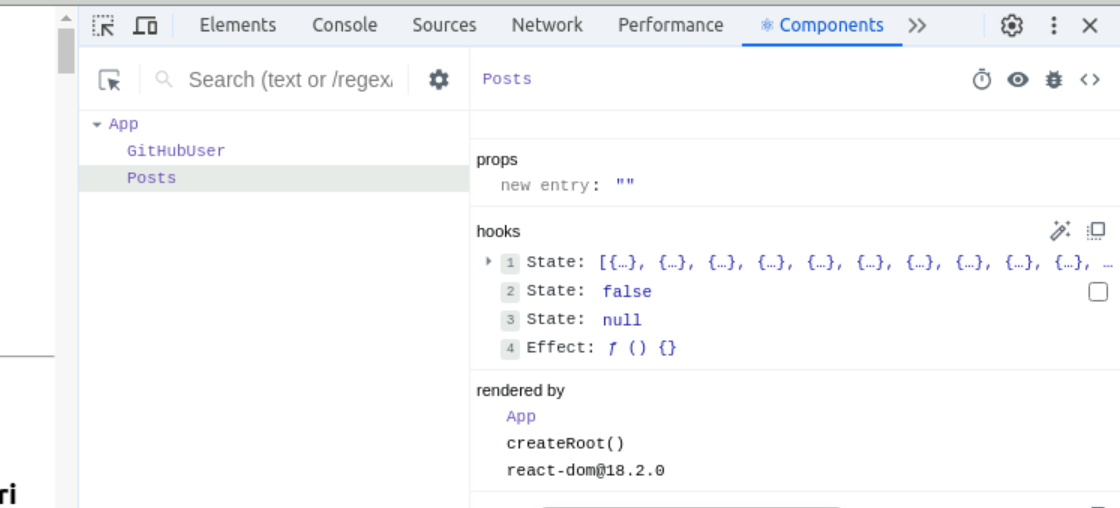
5.6 To open the debugger option in chrome browser **cntr + shift +i**



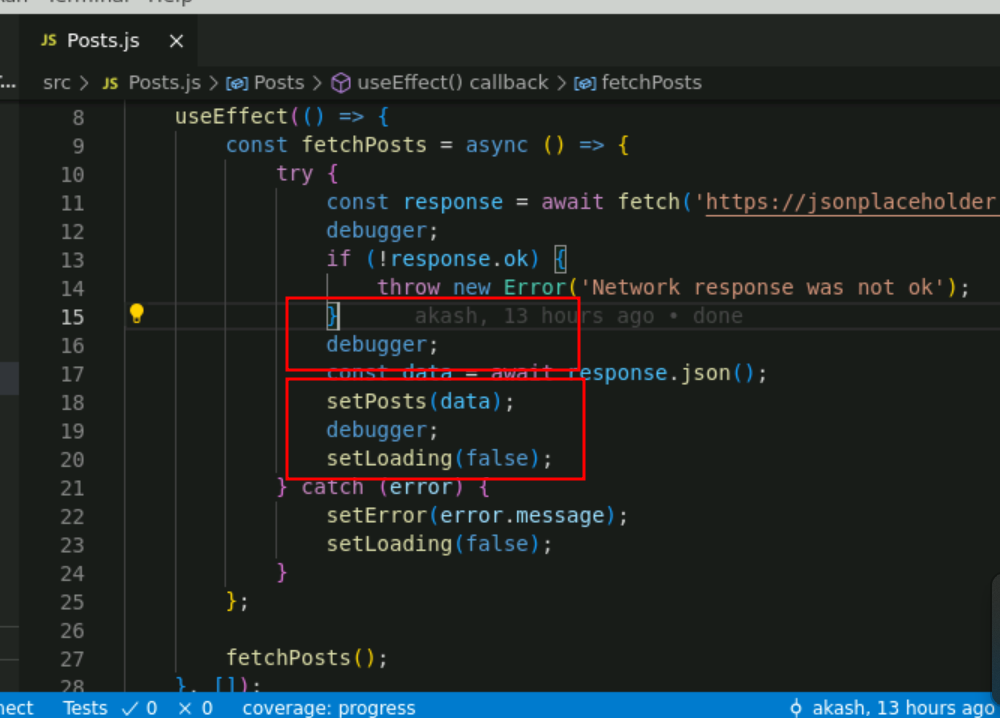
5.7 Now click on component options and select GithubUser component which will give the component details



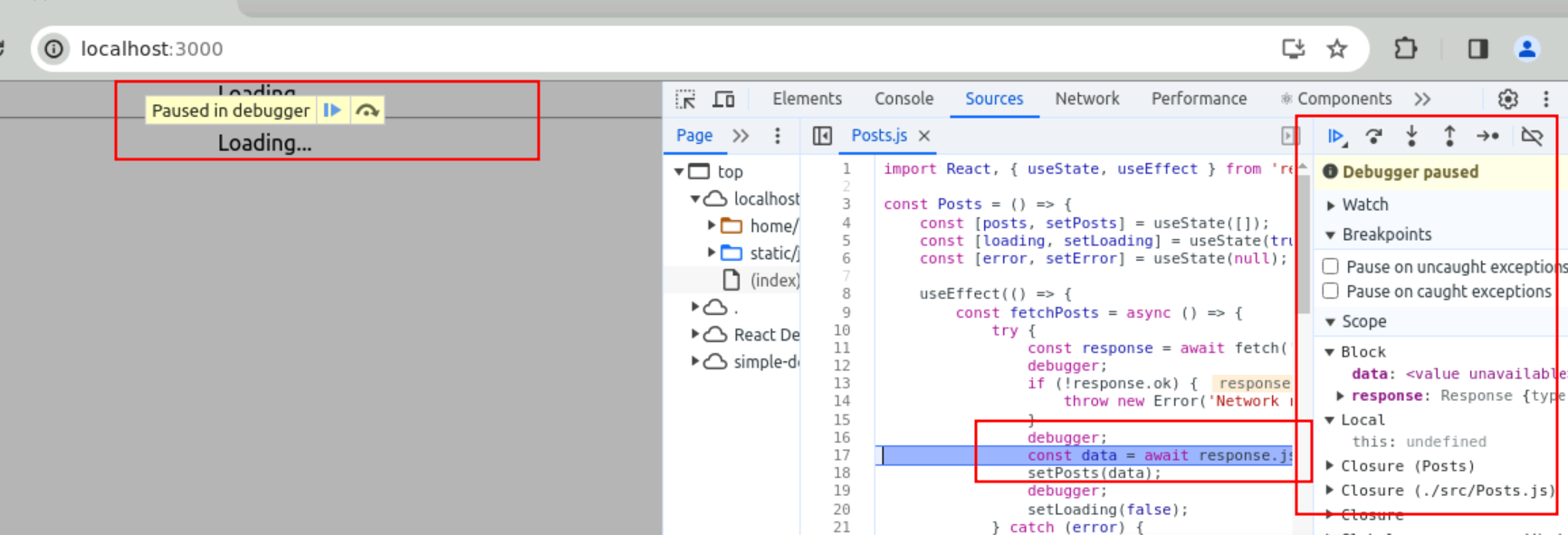
5.8 Now click on component options and select Posts component which will give the component details



5.9 : if you want to put some break point then open the source code of GitHubUser.js or Posts.js and use JavaScript debugger keyword.



5.10 check debug the application.



Now you can click on debugger option and check the code in browser console and debug the code.