**Lesson 05 Demo 01**

**Creating a React Redux Thunk API for Employee Operations**

**Objective:** To create a robust React Redux employee database that seamlessly stores and retrieves data from a JSON file, powered by Axios and Thunk middleware

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

**Prerequisites:** None

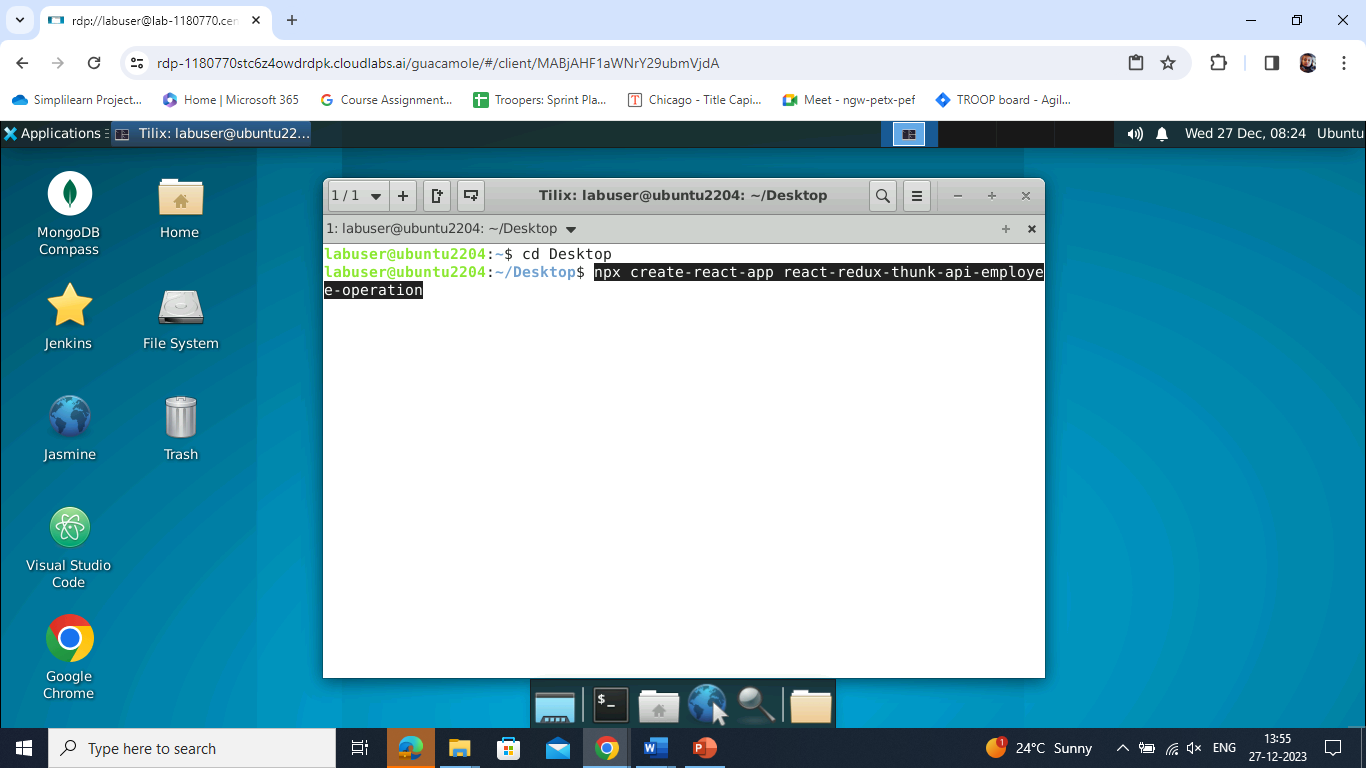
Steps to be followed:

1. Create the React project
2. Create a JSON file with a set of static employee details
3. Create an actions folder
4. Create a reducers folder
5. Create a components folder
6. Configure store and Thunk details
7. Test the application

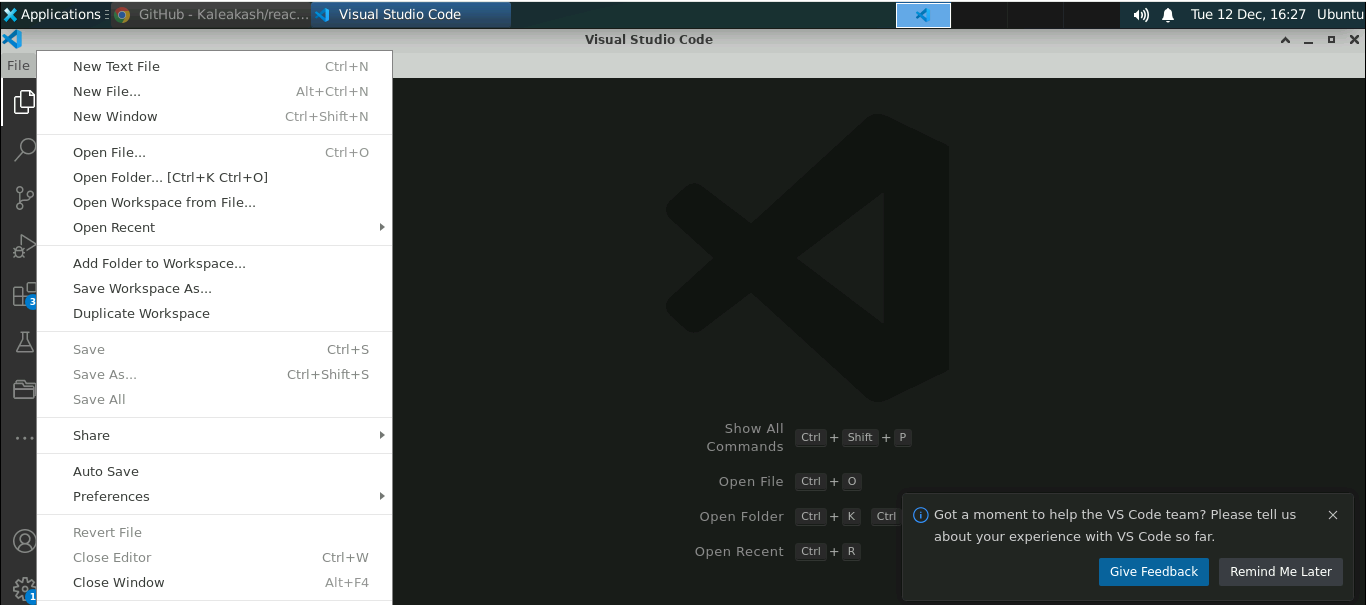
**Step 1: Create the React project**

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

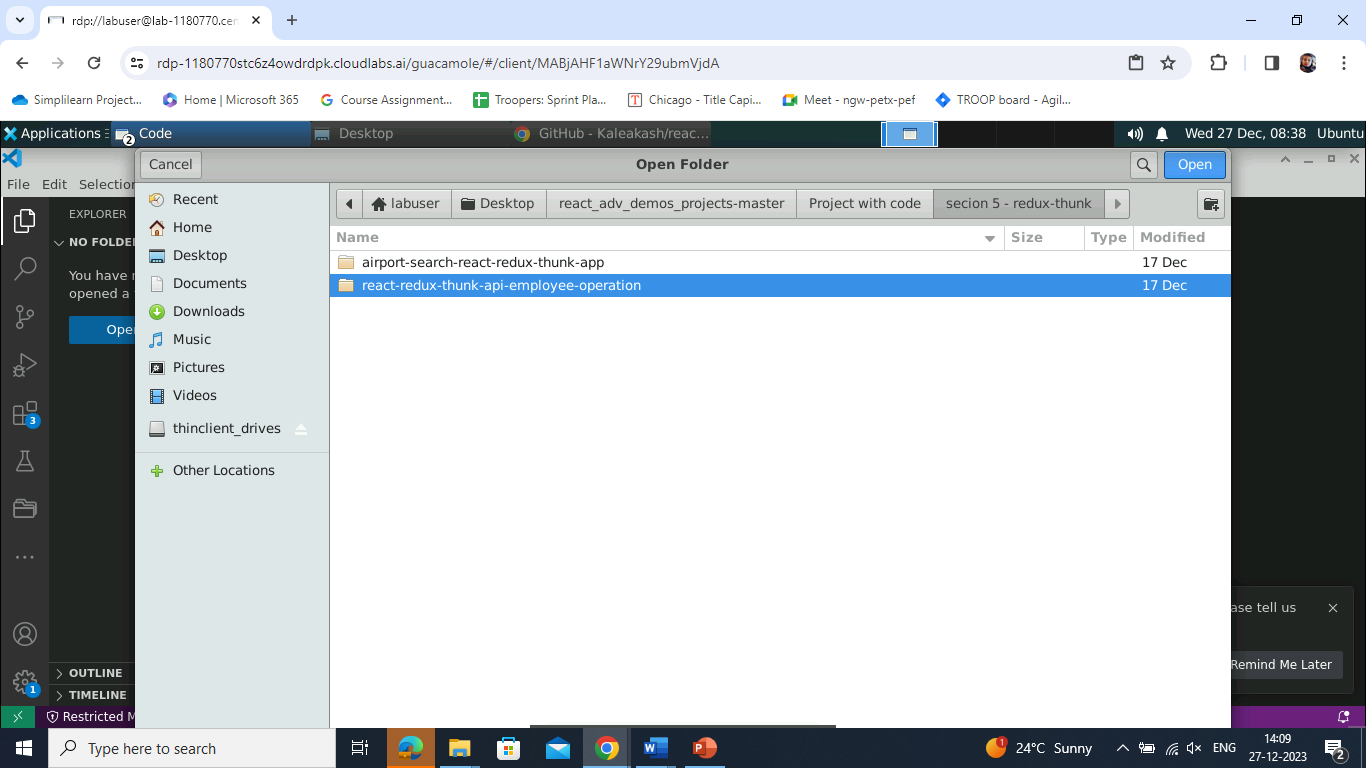
**npx create-react-app** **react-redux-thunk-api-employee-operation**



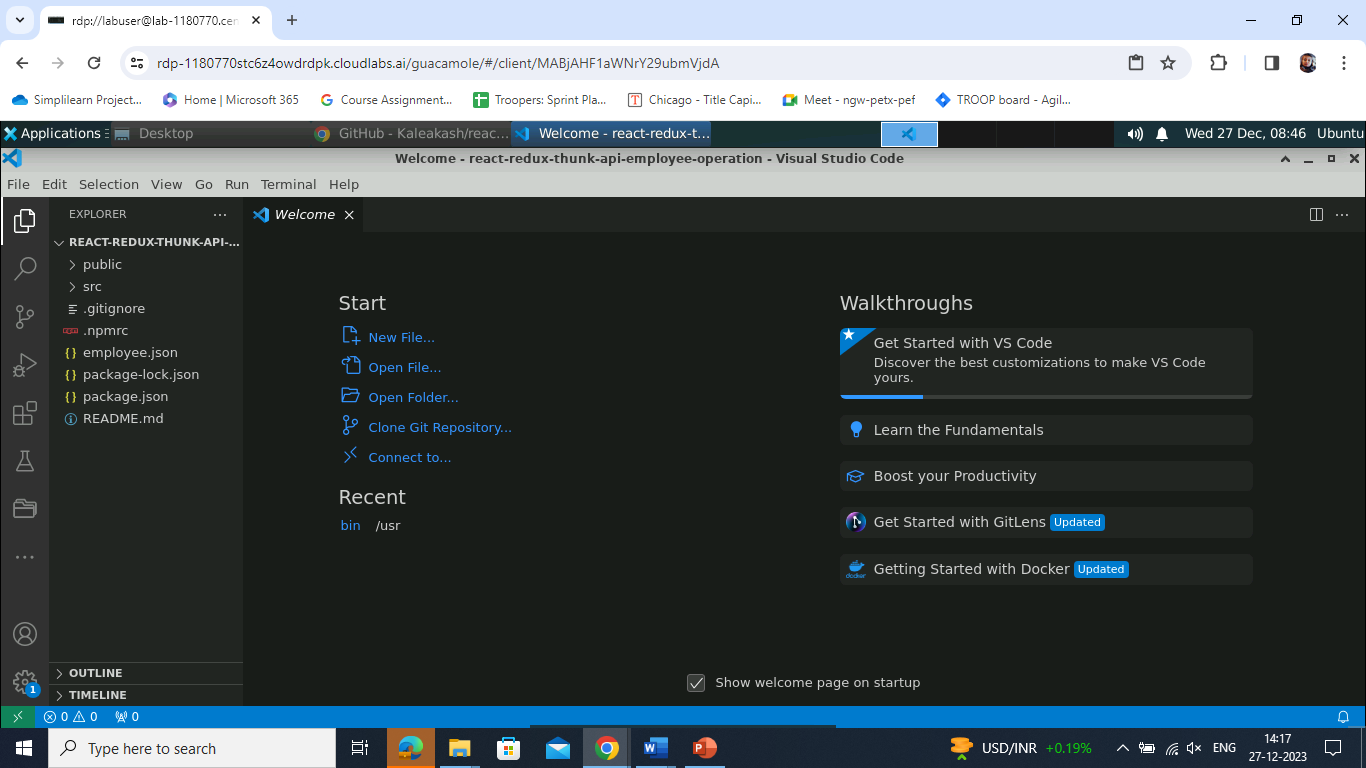
1. Open the created React application folder (**react-redux-thunk-api-employee-operation**) in VS Code by clicking on **File** in the top left corner and selecting **Open Folder**



1. Select the **react-redux-thunk-api-employee-operation** folder and click on **Open** button

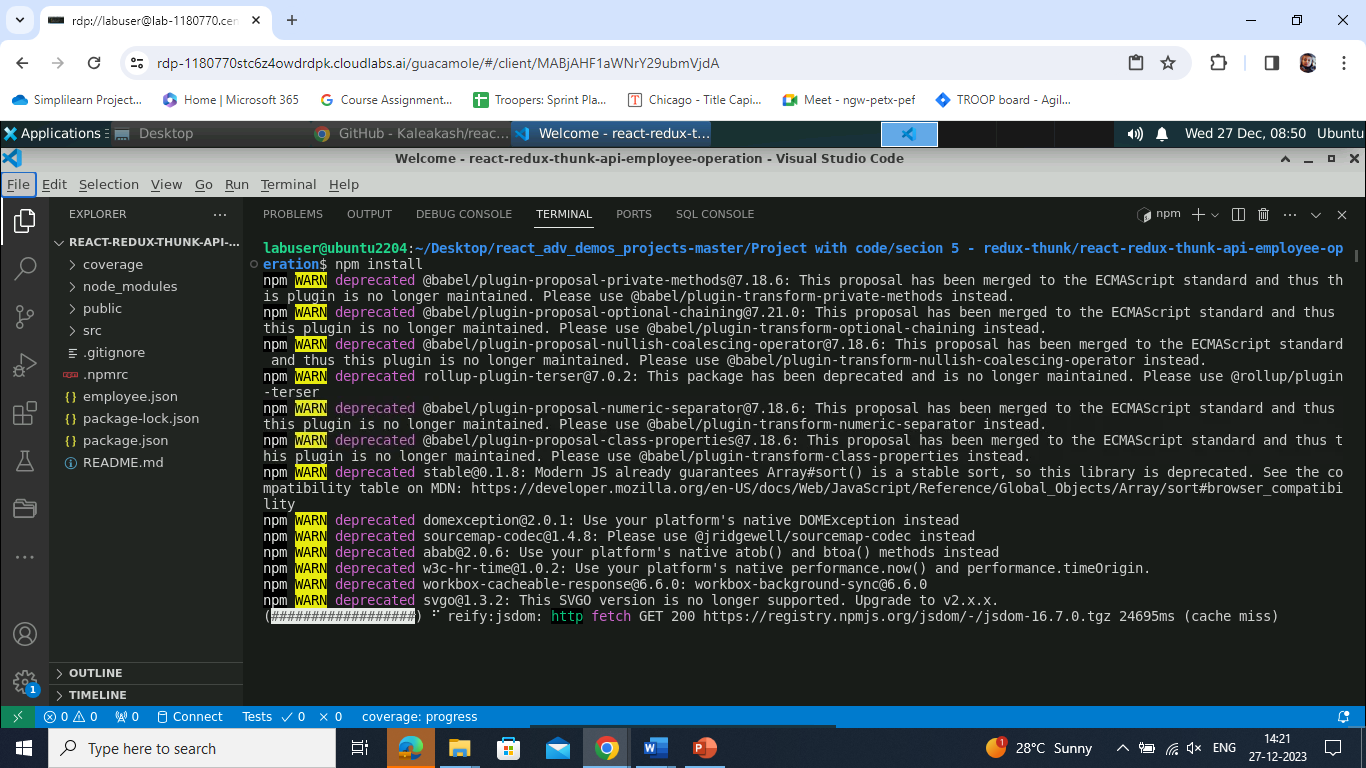


The folder structure appears as follows:

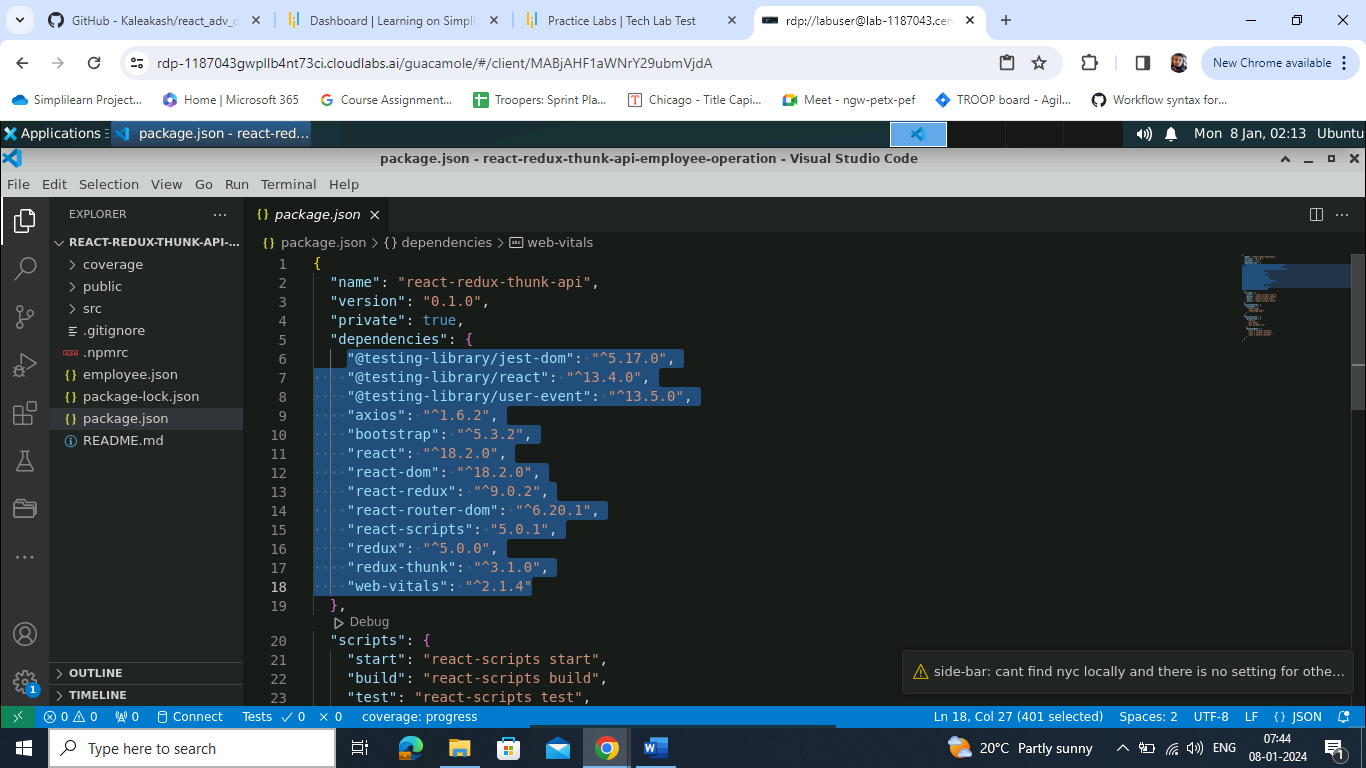


1. Open a **Terminal** inside the **react-redux-thunk-api-employee-operation** project and run the following command:

**npm install**



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



**Step 2: Create a JSON file with a set of static employee details**

1. Create an **employee.json** file and write the following code that contains static JSON data:

**{**

**"employees": [**

**{**

**"id": 100,**

**"name": "Ravi",**

**"salary": 12000**

**},**

**{**

**"id": 101,**

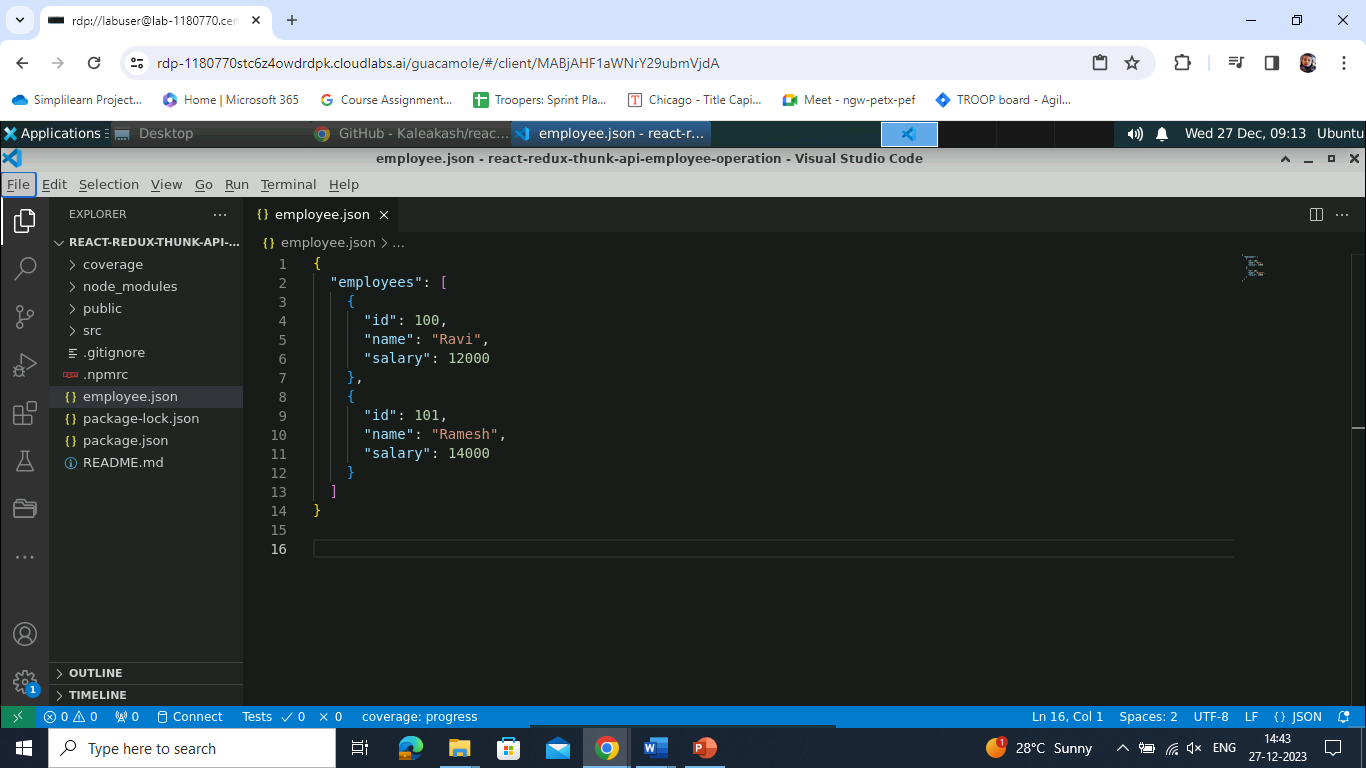
**"name": "Ramesh",**

**"salary": 14000**

**}**

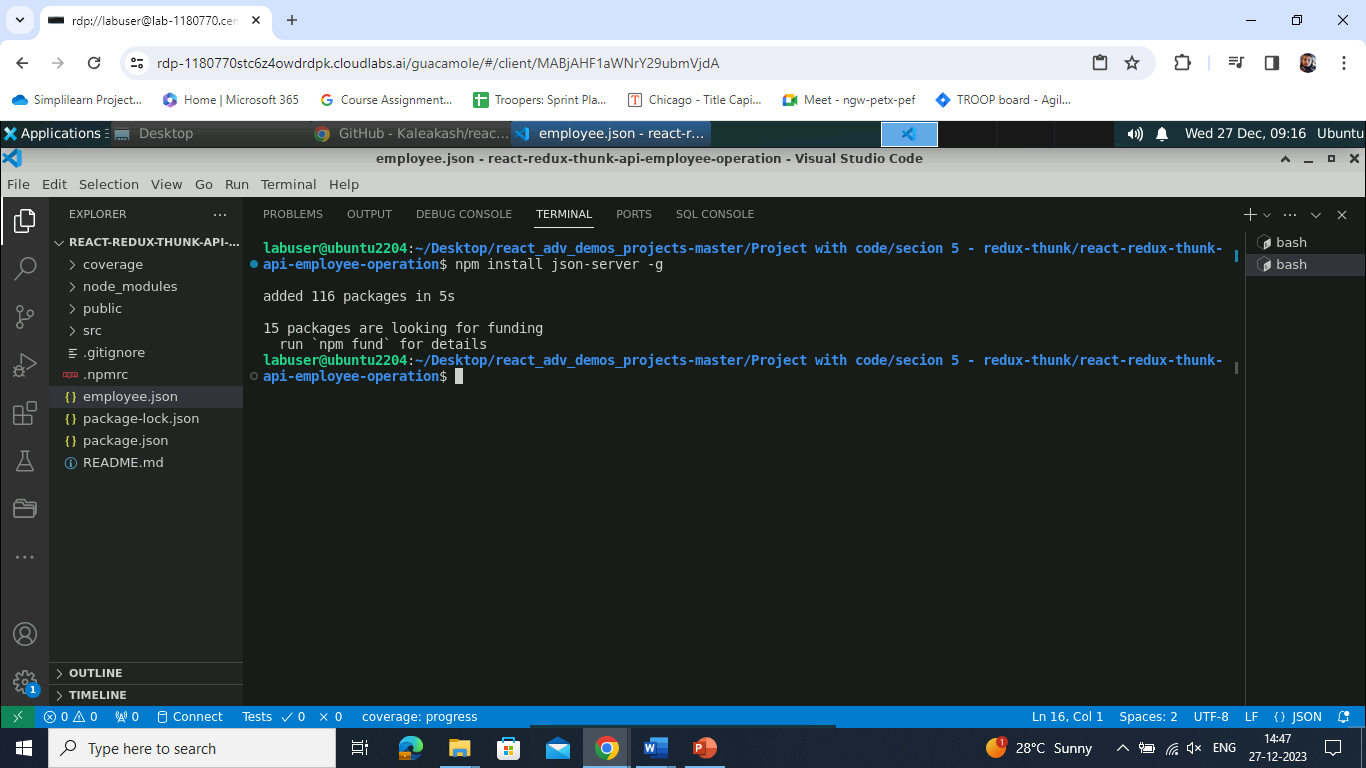
**]**

**}**

****

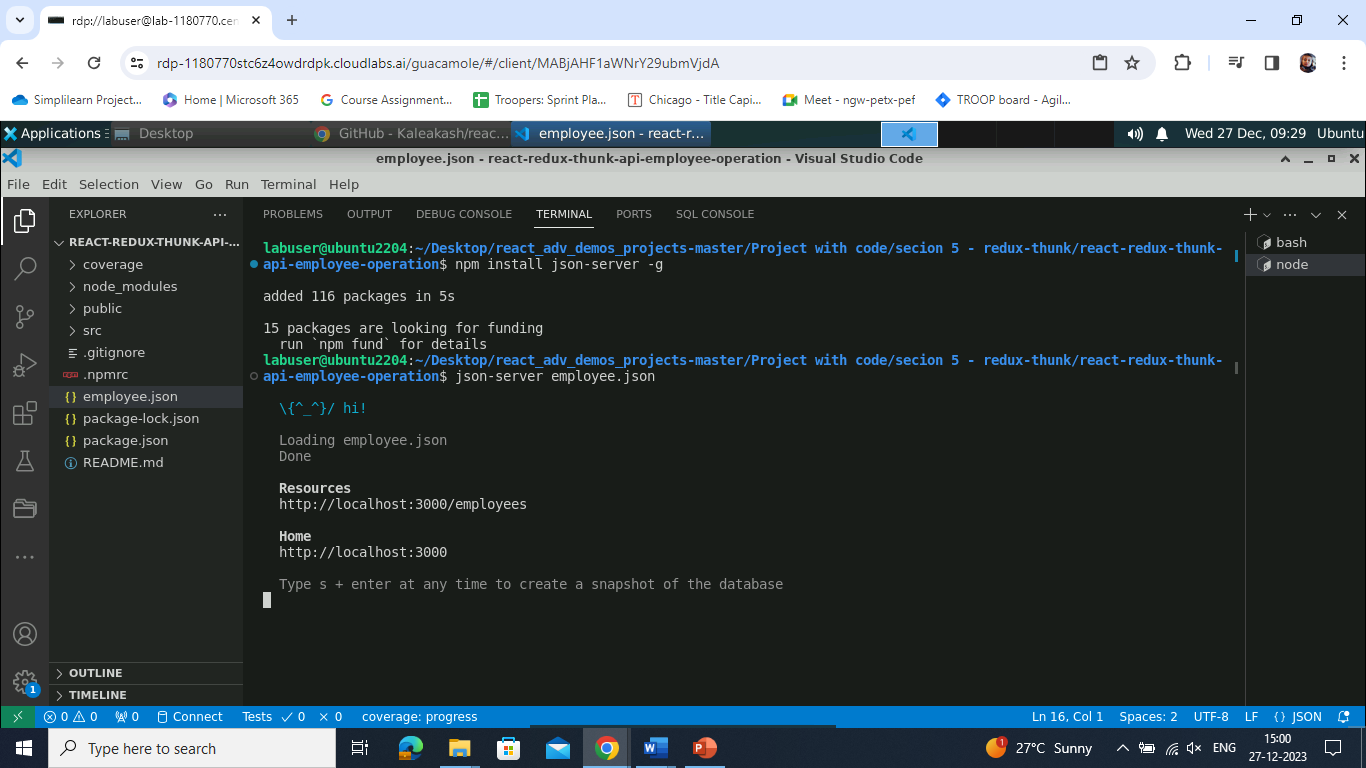
1. Open a **Terminal** inside the **react-redux-thunk-api-employee-operation** project and run the following command to install the **json-server** module:

**npm install json-server -g**

****

1. Run the following command to provide REST API:

**json-server employee.json**

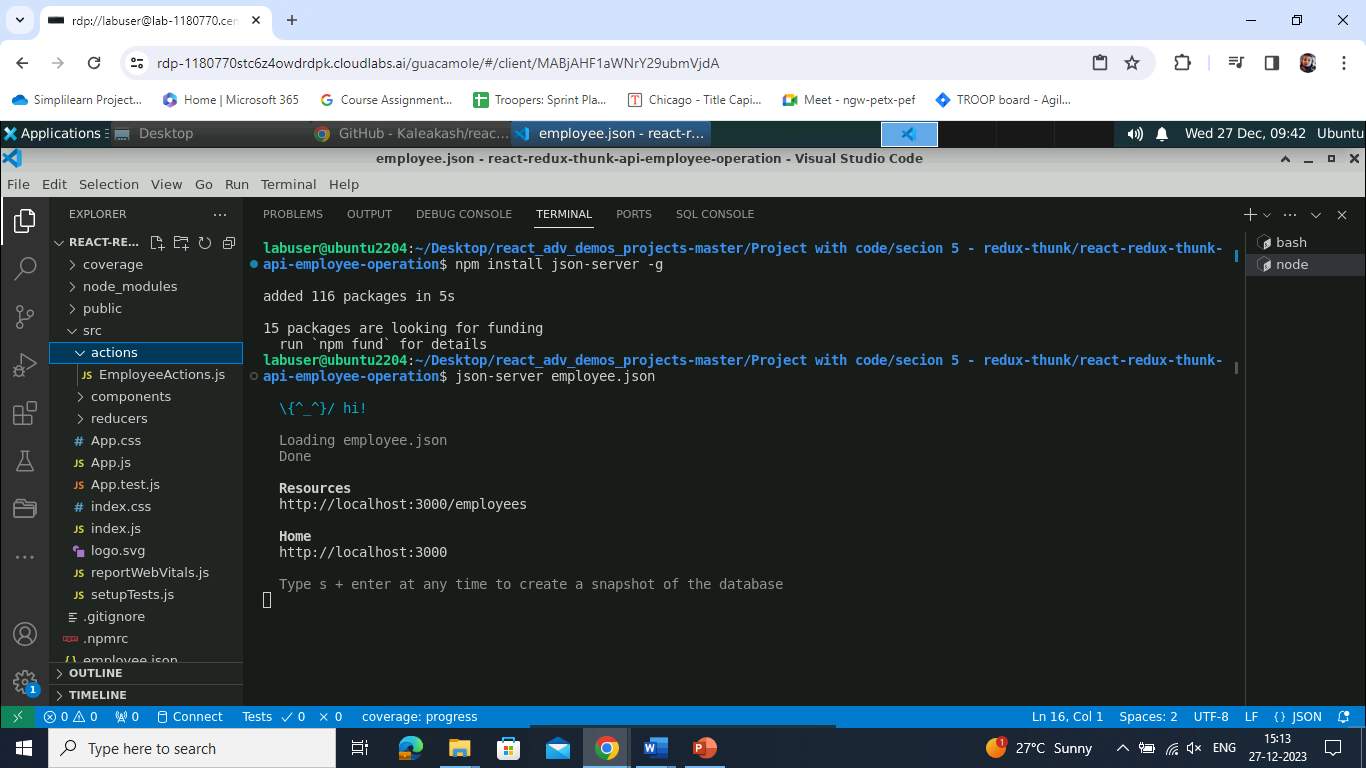


**Note:** Here, **http://localhost:3000/employees** URL provides REST API. You can add and retrieve operations through React.

**Step 3: Create an actions folder**

1. Create an **actions** folder under the **src** folder and create an **EmployeeAction.js** file

inside it



1. Write the following code inside the **EmployeeAction.js** file:

**import axios from "axios";**

**export const ADD\_EMPLOYEE="Add Employee details";**

**export const LOAD\_EMPLOYEE = "Load Employee details";**

**export const EMPLOYEE\_ERROR="Employee Error Generated"**

**export function addEmployee(data){**

**return function(dispatch){**

**axios.post("http://localhost:3000/employees", data.payload)**

**.then(function(response){**

**console.log(response)**

**dispatch({type:ADD\_EMPLOYEE, payload: response.data})**

**})**

**.catch(function(err){**

**dispatch({type:EMPLOYEE\_ERROR, payload: "There was an error."})**

**})**

**}**

**}**

**export function loadEmployee() {**

**return function(dispatch){**

**axios.get("http://localhost:3000/employees").then(function(response){**

**dispatch({type:LOAD\_EMPLOYEE, payload: response.data})**

**})**

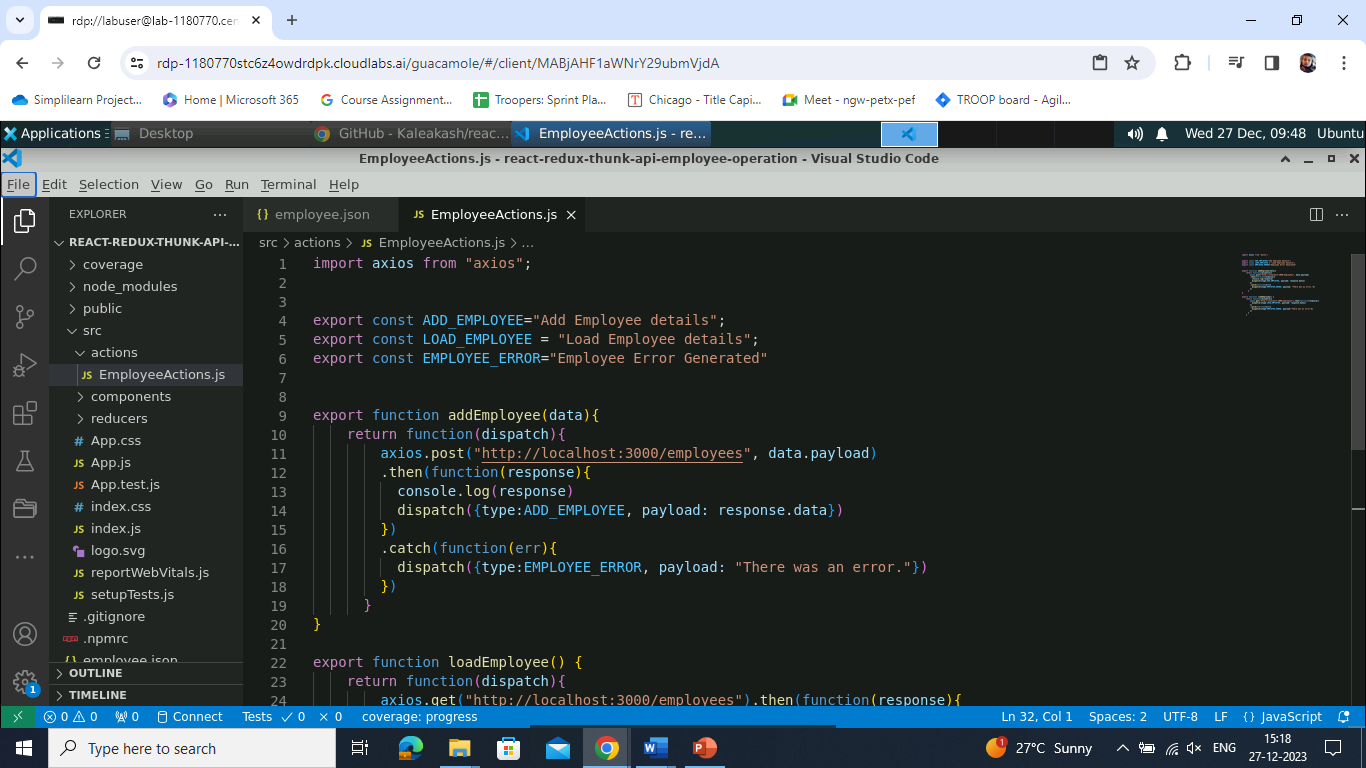
**.catch(function(err){**

**dispatch({type:EMPLOYEE\_ERROR, payload:"There was an error"})**

**})**

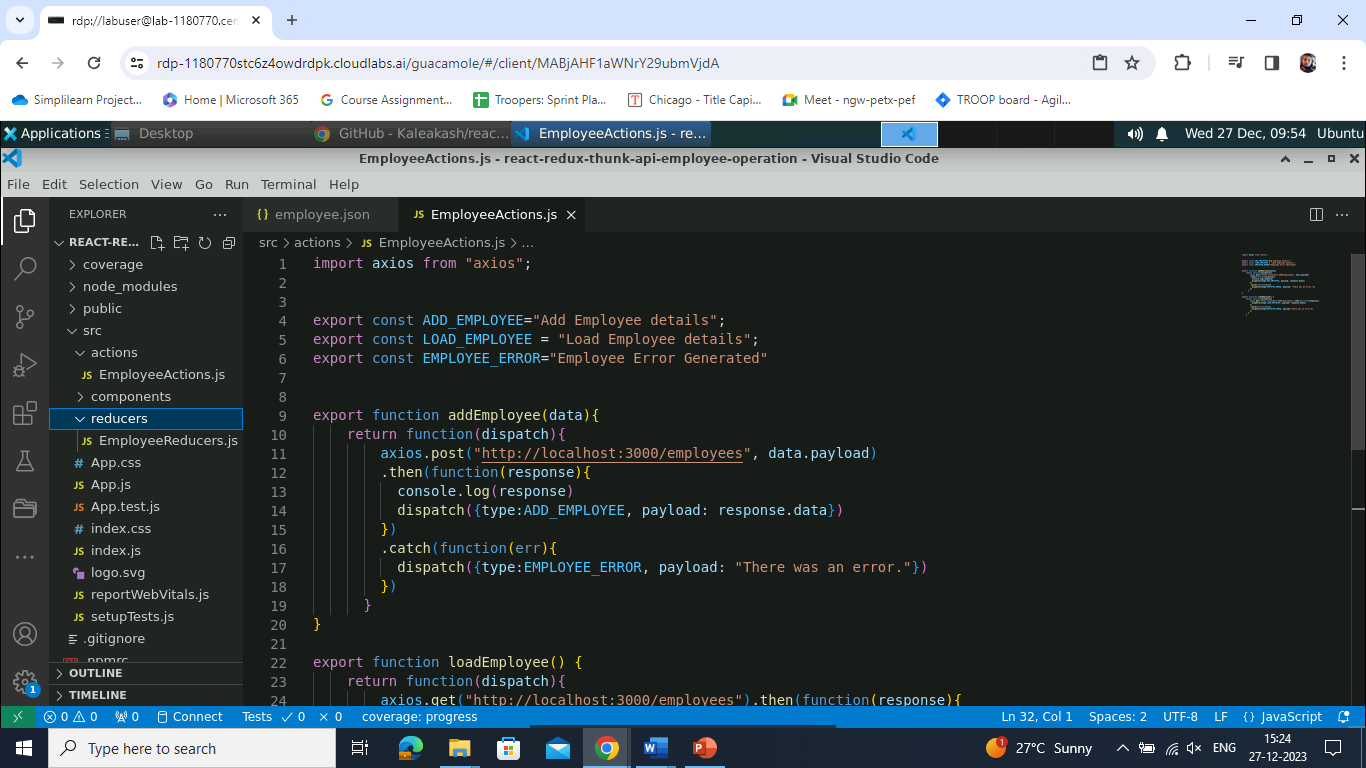
**}**

**}**

****

**Step 4: Create a reducers folder**

1. Create a **reducers** folder under the **src** folder and then create an **EmployeeReducers.js** file under it



1. Write the following code inside the **EmployeeReducers.js** file:

**import {ADD\_EMPLOYEE,LOAD\_EMPLOYEE,EMPLOYEE\_ERROR} from '../actions/EmployeeActions';**

**export default function employeeReducer(state=[],action) {**

**console.log("Action is "+action)**

**switch(action.type){**

**case LOAD\_EMPLOYEE:**

**return action.payload;**

**case ADD\_EMPLOYEE:**

**return [...state, action.payload]**

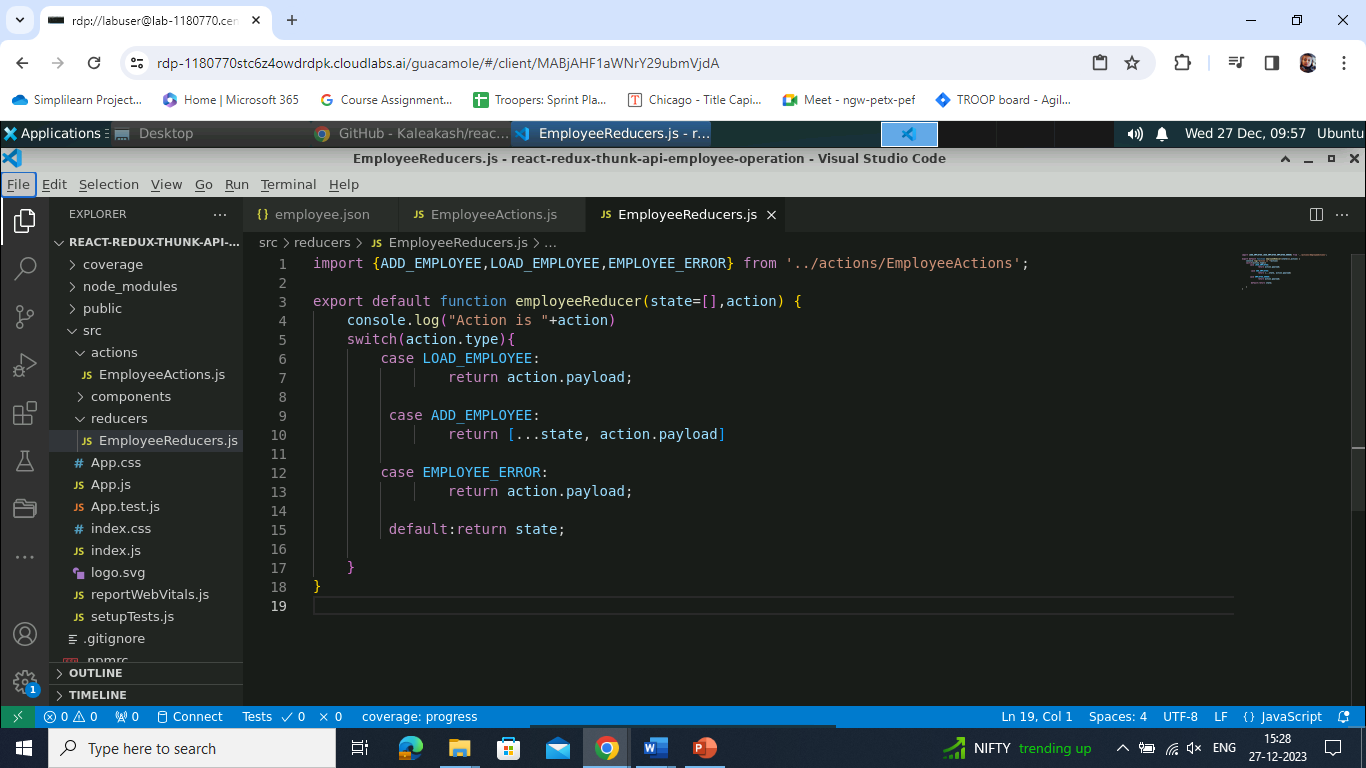
**case EMPLOYEE\_ERROR:**

**return action.payload;**

**default:return state;**

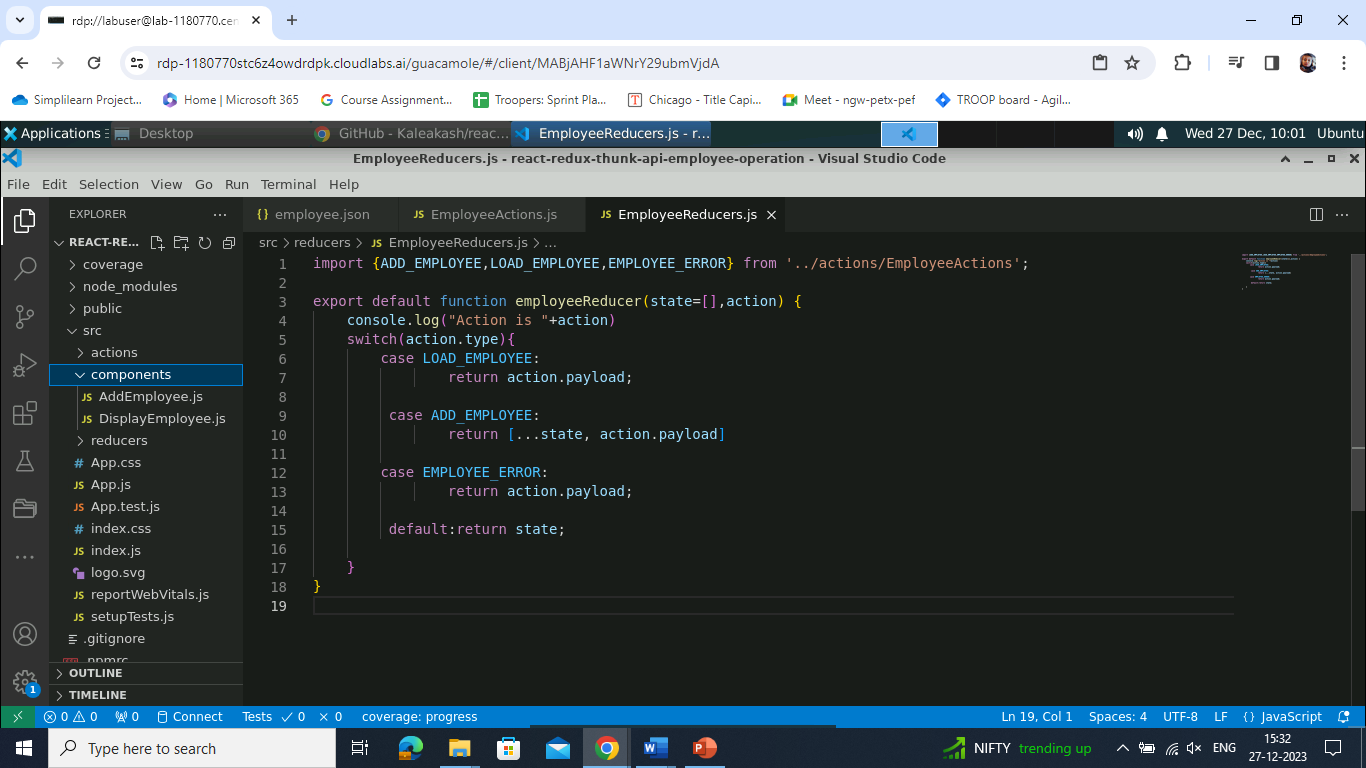
**}**

**}**



**Step 5: Create a components folder**

1. Create a **components** folder and then create **AddEmployee.js** and **DisplayEmployee.js** files inside it



1. Write the following code inside the **AddEmployee.js** file:

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

**import {connect, useDispatch} from 'react-redux';**

**import {addEmployee,ADD\_EMPLOYEE, loadEmployee} from '../actions/EmployeeActions';**

**import "../../node\_modules/bootstrap/dist/css/bootstrap.min.css"**

**function AddEmployee() {**

**let [employee,setEmployee]=useState({});**

**let dispatch = useDispatch();**

**let storeEmployeeInfo=async (event) => {**

**event.preventDefault();**

**let result = await dispatch(addEmployee({type:"ADD\_EMPLOYEE",payload:employee}));**

**console.log(result);**

**setEmployee({id:"",name:"",salary:""})**

**}**

**return(**

**<div className="container">**

**<h2>Add Employee Details</h2>**

**<form onSubmit={storeEmployeeInfo} className="form-group col-md-12">**

**Id <input type="text" name="id" value= {employee.id} onChange={**

**(event)=>setEmployee({...employee,"id":event.target.value})**

**}**

**className="form-control"**

**/><br/>**

**Name <input type="text" name="name" value= {employee.name} onChange={**

**(event)=>setEmployee({...employee,"name":event.target.value})**

**} className="form-control"/>**

**<br/>**

**Salary <input type="text" name="salary" value= {employee.salary} onChange={**

**(event)=>setEmployee({...employee,"salary":event.target.value})**

**} className="form-control"/>**

**<br/>**

**<input type="submit" value="Store Record" className="btn btn-primary"/>**

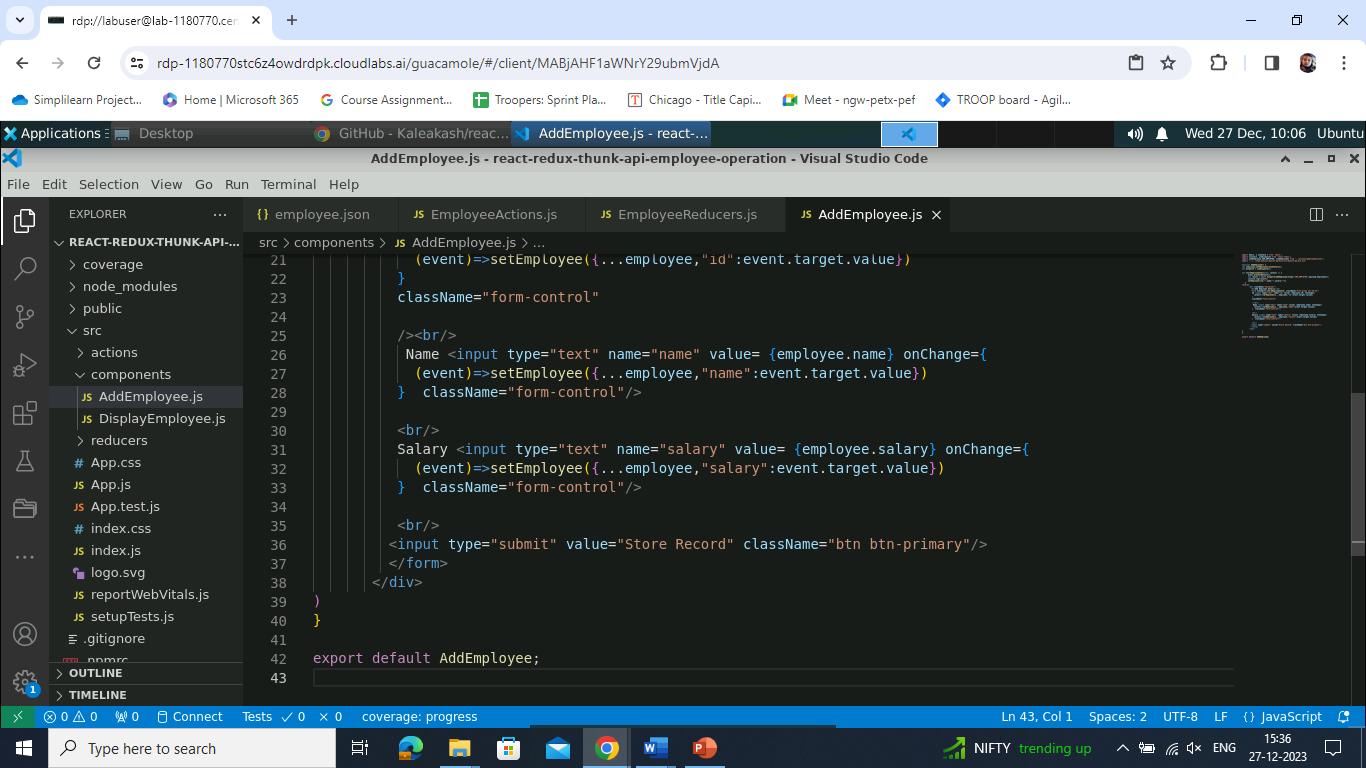
**</form>**

**</div>**

**)**

**}**

**export default AddEmployee;**



1. Write the following code inside the **DisplayEmployee.js** file:

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

**import {useDispatch, useSelector} from 'react-redux';**

**import { loadEmployee } from '../actions/EmployeeActions';**

**function DisplayEmployee() {**

**let employees = useSelector(gs=>gs.employee);**

**let dispatch = useDispatch();**

**console.log(employees);**

**useEffect(()=> {**

**const loadData=()=> {**

**dispatch(loadEmployee());**

**}**

**loadData();**

**},[])**

**return(**

**<div>**

**<h3>Employee Details are </h3>**

**<table border="1" className="table table-striped">**

**<thead>**

**<tr>**

**<th>Id</th>**

**<th>Name</th>**

**<th>Salary</th>**

**</tr>**

**</thead>**

**<tbody>**

**{employees.map((ele,i)=>**

**<tr key={i}><td>{ele.id}</td><td>{ele.name}</td><td>{ele.salary}</td></tr>)**

**}**

**</tbody>**

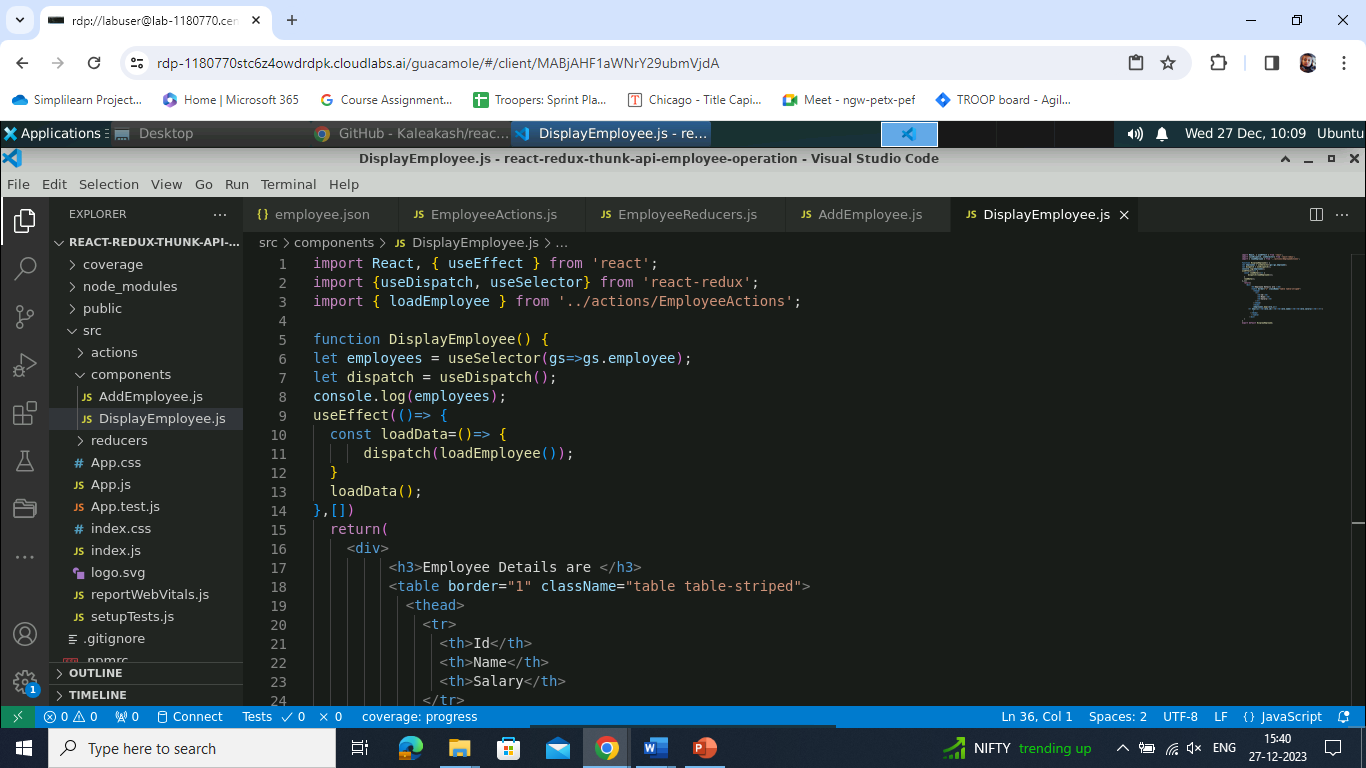
**</table>**

**</div>**

**)**

**}**

**export default DisplayEmployee;**



**Step 6: Configure store and Thunk details**

1. Write the following code inside the **index.js** file (under **components folder**) to create a store:

**import ReactDOM from 'react-dom/client';**

**import './index.css';**

**import App from './App';**

**import reportWebVitals from './reportWebVitals';**

**import React from 'react';**

**import { legacy\_createStore as createStore,combineReducers,applyMiddleware } from 'redux';**

**import {Provider} from 'react-redux'**

**import {thunk} from "redux-thunk"**

**import employeeReducer from './reducers/EmployeeReducers';**

**const allReducer = combineReducers({**

**employee:employeeReducer**

**})**

**const store = createStore(allReducer,applyMiddleware(thunk));**

**const root = ReactDOM.createRoot(document.getElementById('root'));**

**root.render(**

**<React.StrictMode>**

**<Provider store={store}><App /></Provider>**

**</React.StrictMode>**

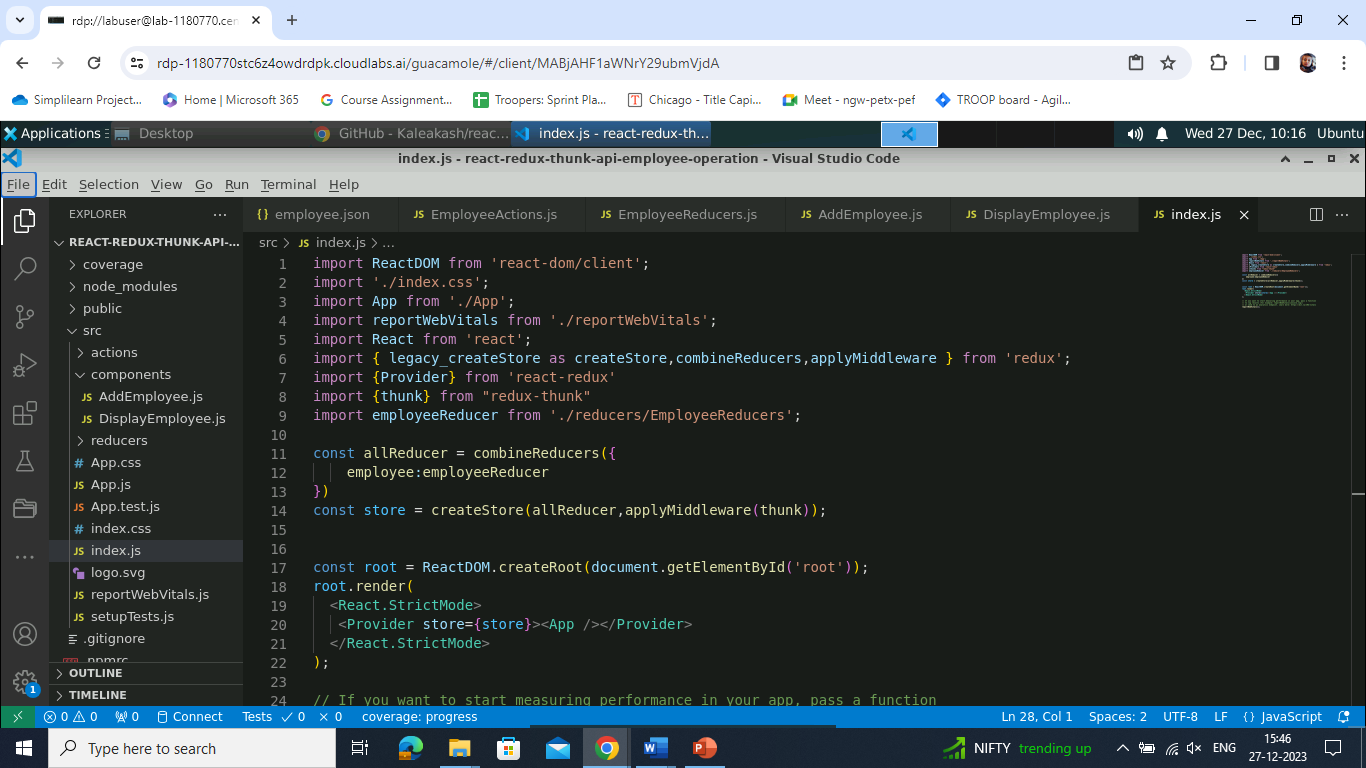
**);**

**// If you want to start measuring performance in your app, pass a function**

**// to log results (for example: reportWebVitals(console.log))**

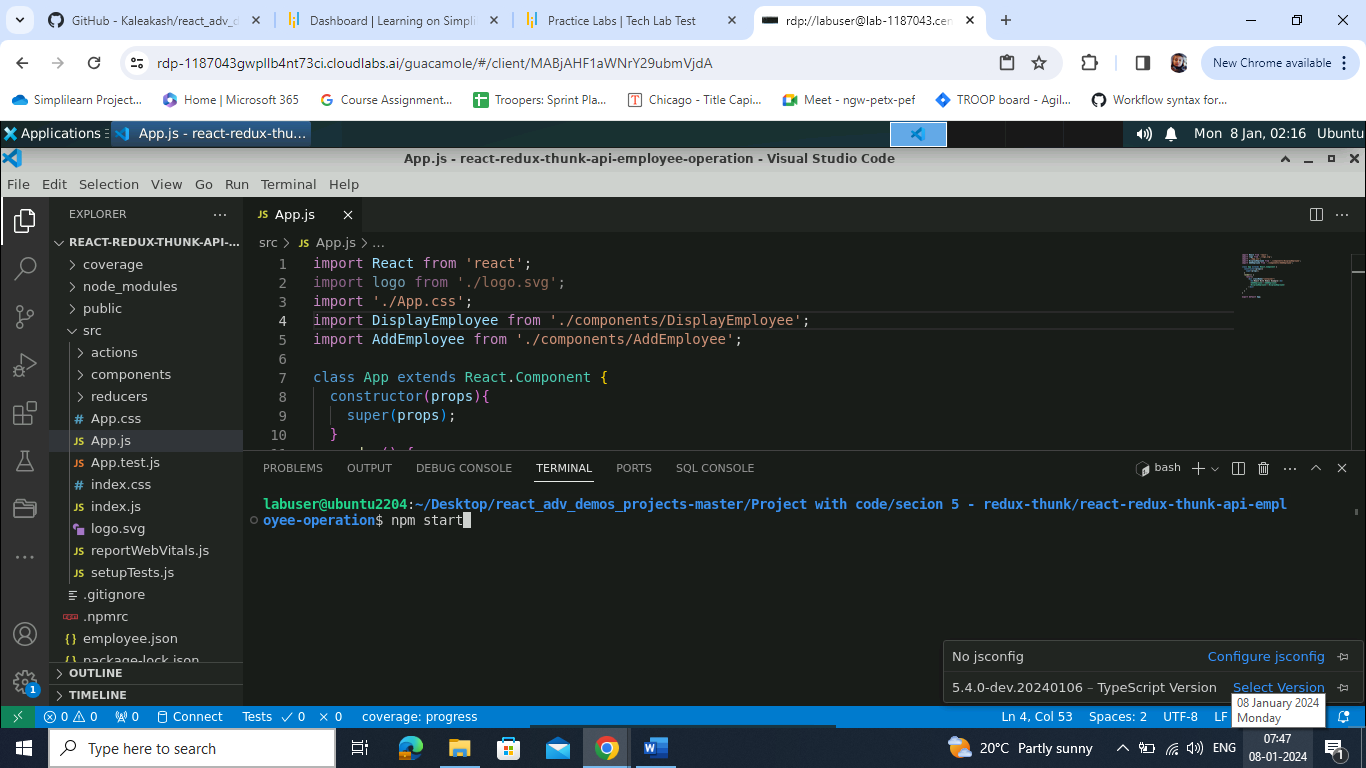
**// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals**

**reportWebVitals();**

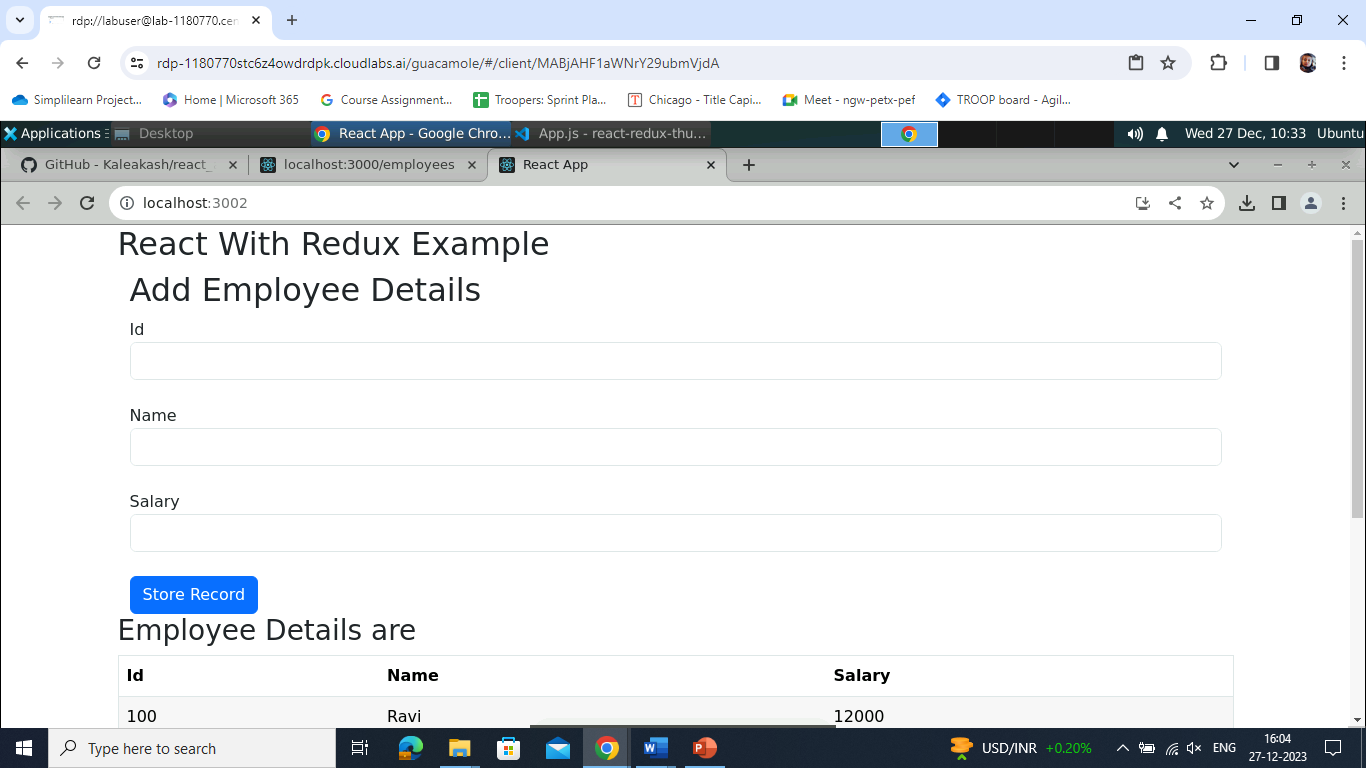


**Step 7: Run the application**

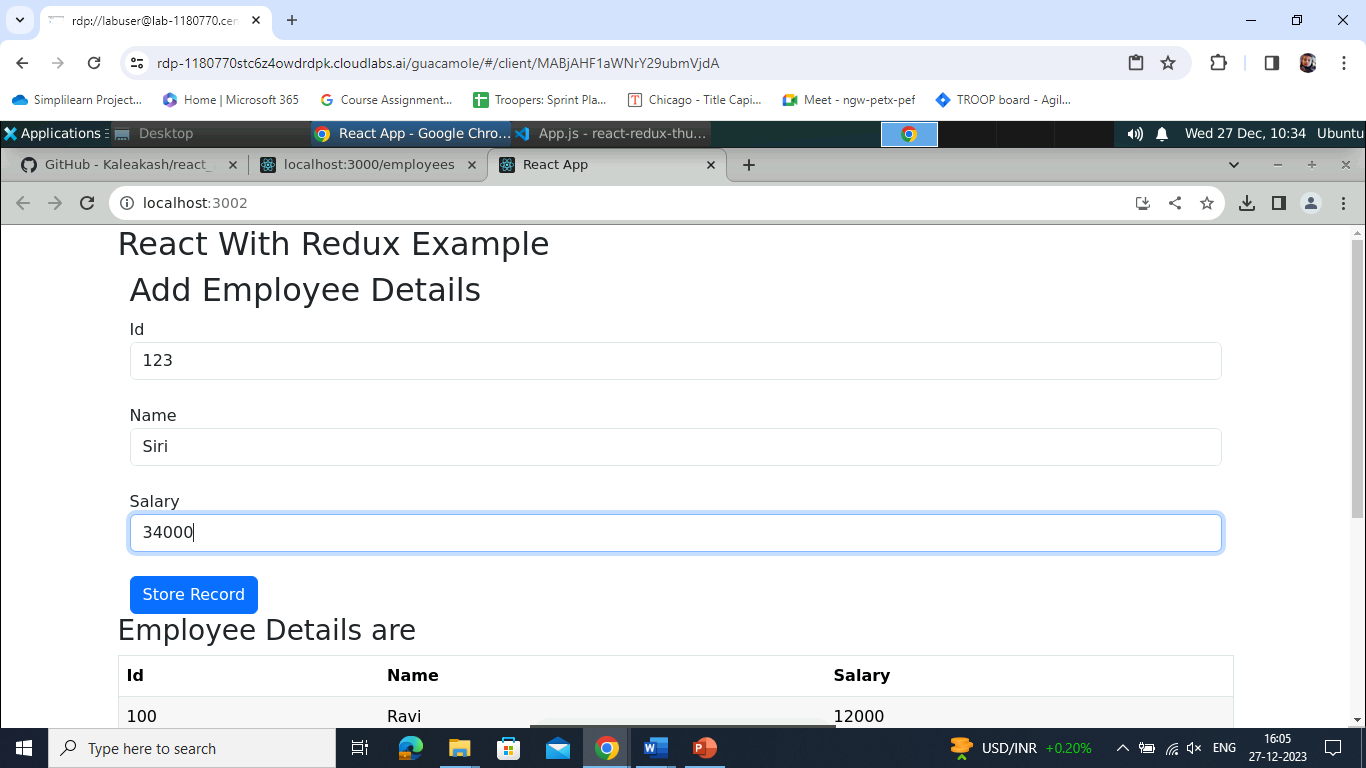
1. Run the application using the following command:  
     
   **npm start**

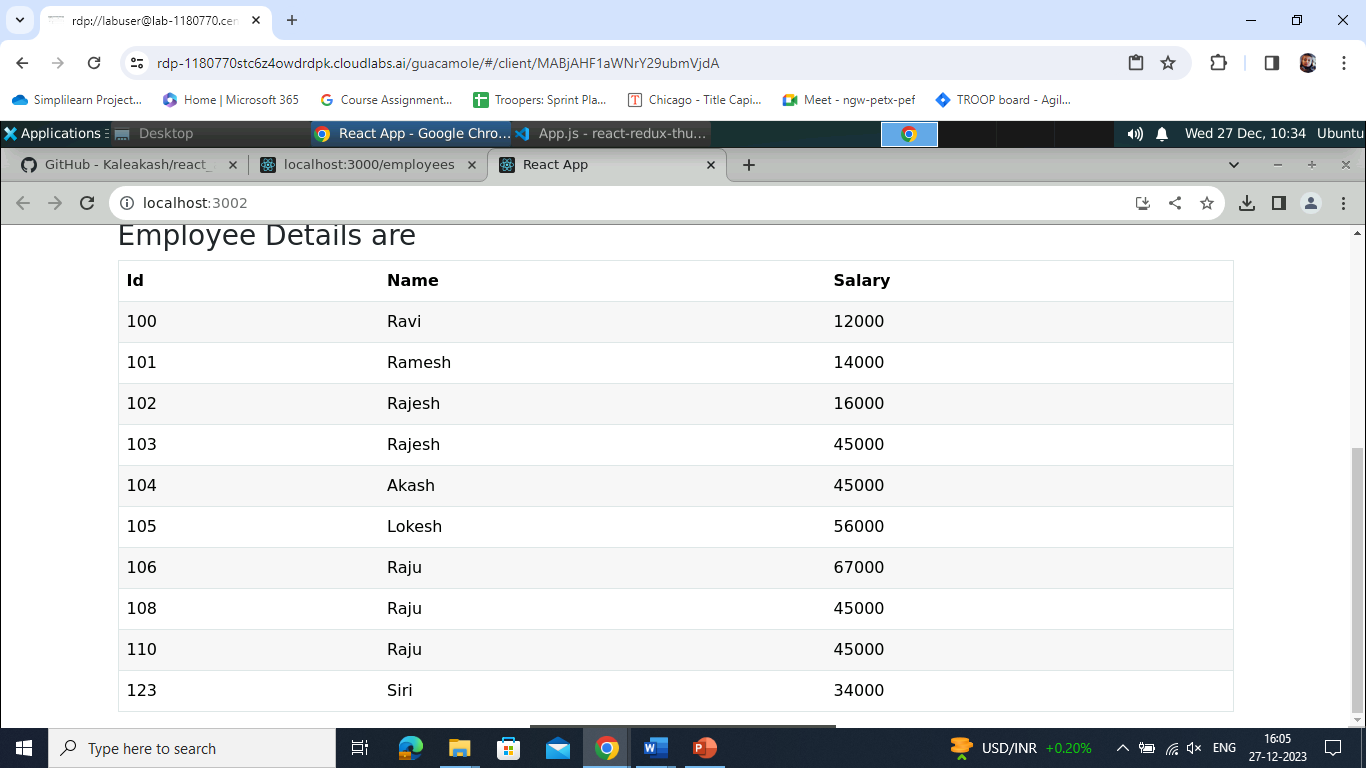
****

Output will look as follows:



1. In the **Add Employee Details** page, add **ID**, **Name**, and **Salary**, and click on the **Store Record** button as shown below:



Output will look as follows:  
  


With this, you have created a robust React Redux employee database that seamlessly stores and retrieves data from a JSON file powered by Axios and Thunk middleware.