**Lesson 09 Demo 02**

**Create a React Next.js Login Application**

**Objective:** To create a Next.js application by creating server-side components to do login with file and testing with the React component

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

**Prerequisites:** None

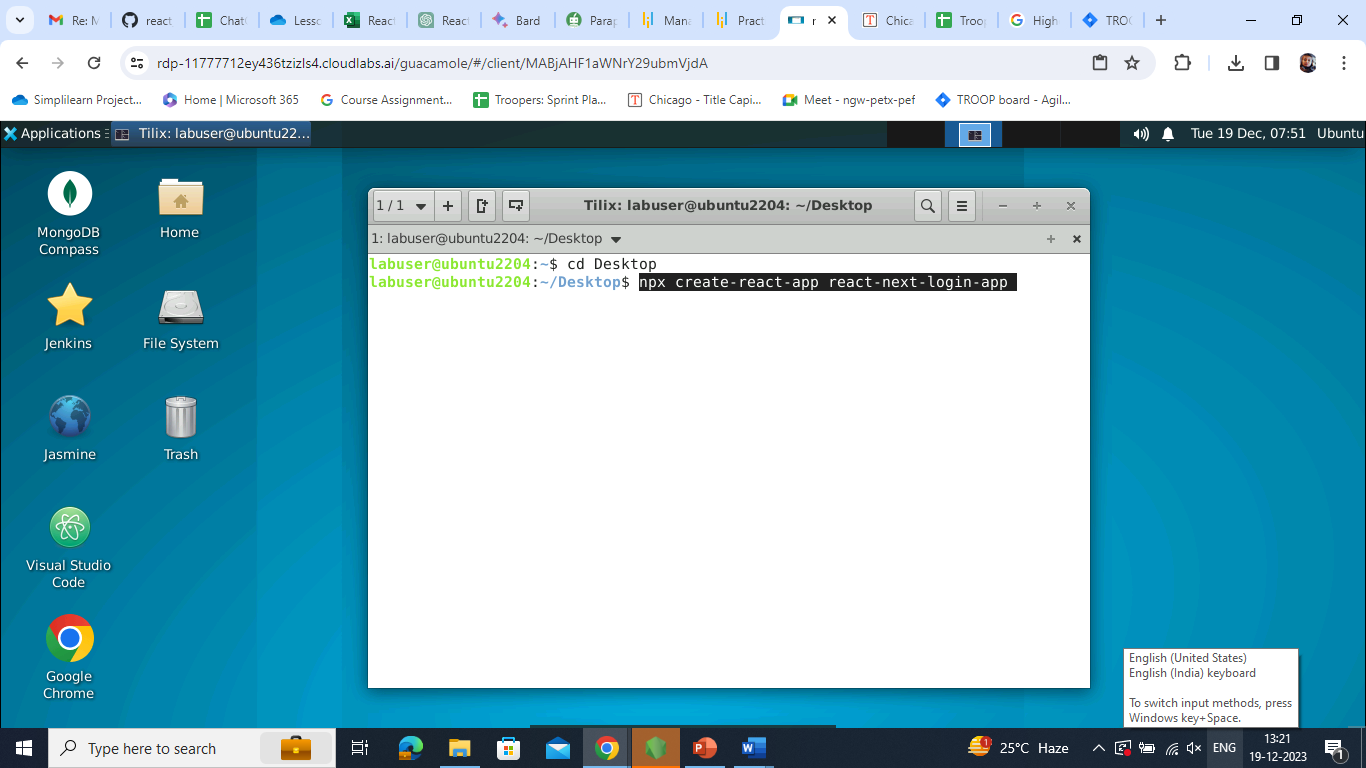
Steps to be followed:

1. Create and set up the Next.js project
2. Create a file that contains a set of login details
3. Create a server folder that holds a server-side component
4. Create a client folder
5. Run the application

**Step 1: Create and set up the Next.js project**

1. Open a terminal window to run the following command to create a Next.js project:

**npx create-react-app react-next-login-app**



1. Open the created React application folder (**react-next-login-app**) 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. Select the **react-next-login-app** folder and click on **Open** button

A computer screen with a white box

Description automatically generated

The folder structure appears as follows:

A screenshot of a computer

Description automatically generated

1. Open the **TERMINAL** inside the **react-next-login-app** folder

A computer screen shot of a black screen

Description automatically generated

1. Run the following command to download the dependencies:

**npm install**

A screenshot of a computer

Description automatically generated

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

**Step 2: Create a file that contains a set of login details**

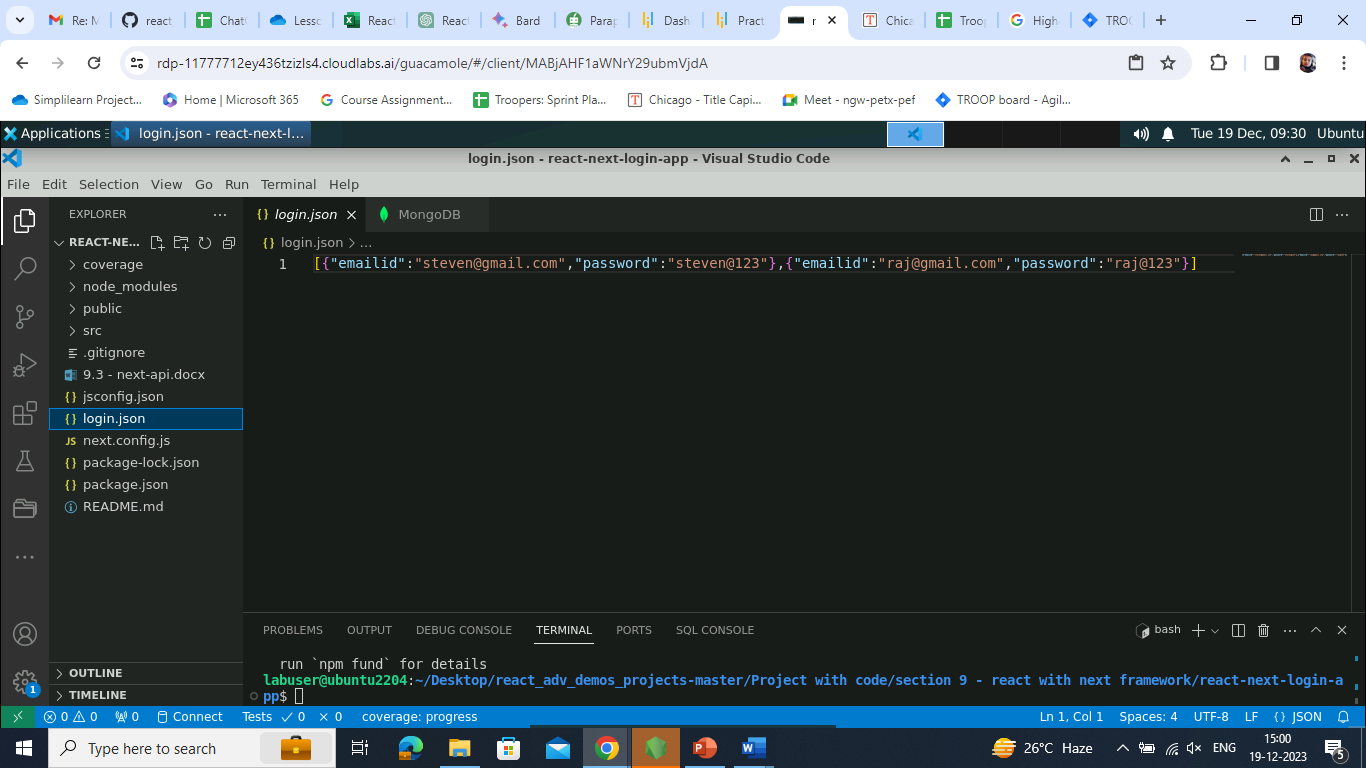
1. Create a **login.json** file under the **src** folder

**A computer screen shot of a black screen

Description automatically generated**

1. Enter the following code inside the **login.json** file:

**[{"emailid":"steven@gmail.com","password":"steven@123"},{"emailid":"raj@gmail.com","password":"raj@123"}]**



**Step 3: Create a server folder that holds a server-side component**

* 1. Create a **server/signin** folder, and inside it, a **route.js** file

A computer screen shot of a black screen

Description automatically generated

* 1. Inside the **route.js** file, enter the following code:

**import { NextResponse } from "next/server"**

**import { promises as fs } from 'fs';**

**export async function POST(request){**

**let login = await request.json();**

**const file = await fs.readFile("login.json","utf-8");**

**let loginDetails = JSON.parse(file);**

**let result = loginDetails.find(l=>l.emailid==login.emailid && l.password==login.password);**

**console.log(result);**

**if(result!=undefined){**

**return NextResponse.json({"msg":"Successfully login"})**

**}else {**

**return NextResponse.json({"msg":"In Valid emailid or password"});**

**}**

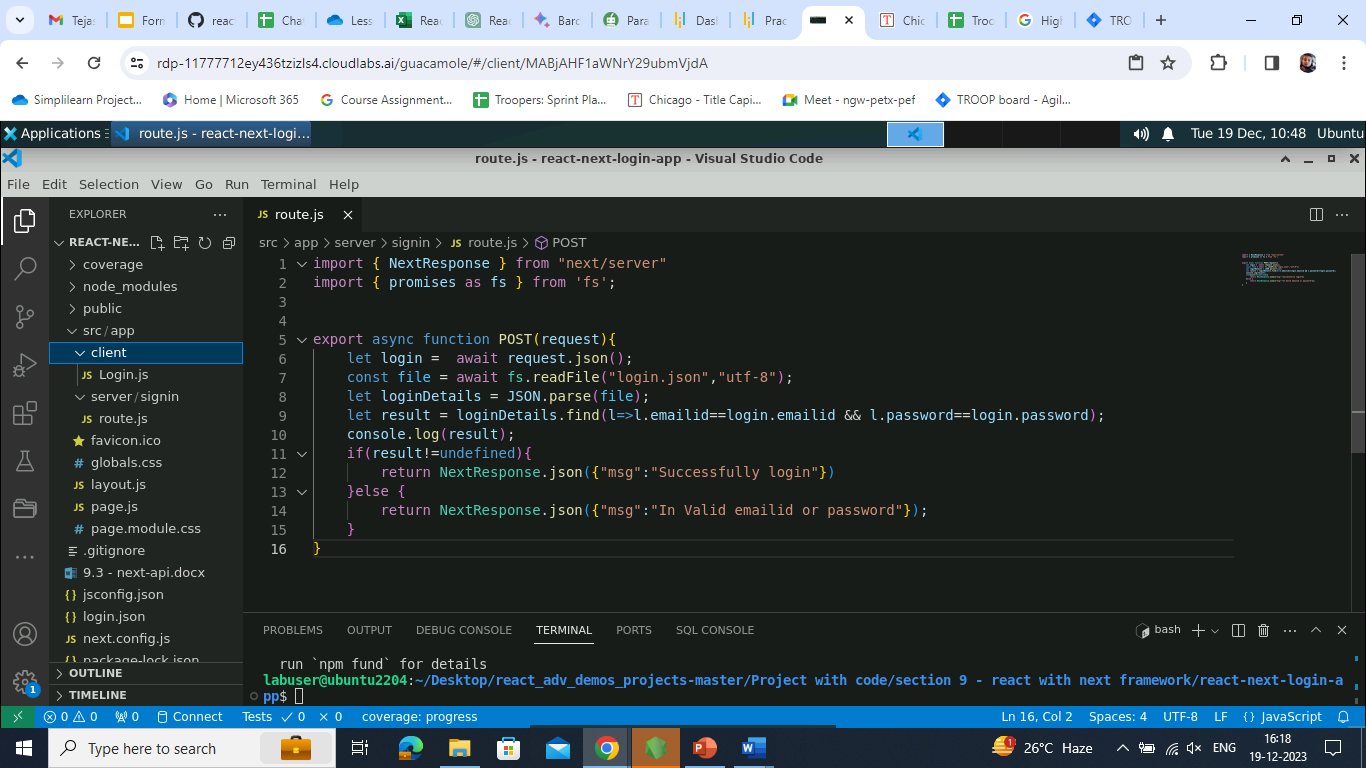
**}**

**A screenshot of a computer

Description automatically generated**

**Step 4: Create a client folder**

1. Create a **client** folder and create a component file as **Login.js** inside it

****

1. Inside the **Login.js** file, enter the following code:

**"use client"**

**import 'bootstrap/dist/css/bootstrap.css'**

**import { useState } from 'react'**

**import axios from 'axios';**

**export default function Login() {**

**let [emailid,setEmailId]=useState();**

**let [password,setPassword]=useState();**

**let [msg,setMsg]=useState("");**

**const signIn = async (event)=> {**

**event.preventDefault();**

**console.log(emailid+" "+password);**

**let login = {"emailid":emailid,"password":password};**

**let result = await axios.post(" http://localhost:3000/server/signin",login);**

**setMsg(result.data.msg);**

**setEmailId("");**

**setPassword("");**

**}**

**return(<div className='container'>**

**<form className='form-group' onSubmit={signIn}>**

**<div className='row offset-4'>**

**<div className='col-6'>**

**<h2>Login Page</h2>**

**</div>**

**</div>**

**<div className='row offset-2' >**

**<div className='col-3'>**

**<label>EmailId</label>**

**</div>**

**<div className='col-3'>**

**<input type="email" name="emailid" className="form-control"**

**onChange={(event)=>setEmailId(event.target.value)} value={emailid}**

**/>**

**</div>**

**</div>**

**<div className='row offset-2' >**

**<div className='col-3'>**

**<label>Password</label>**

**</div>**

**<div className='col-3'>**

**<input type="password" name="password" className="form-control"**

**onChange={(event)=>setPassword(event.target.value)} value={password}**

**/>**

**</div>**

**</div>**

**<div className='row offset-2' >**

**<div className='col-3'>**

**<input type="submit" value="SignIn" className='btn btn-success'/>**

**</div>**

**<div className='col-3'>**

**<input type="reset" value="reset" className='btn btn-info'/>**

**</div>**

**</div>**

**<div className='row offset-4'>**

**<div className='col-6'>**

**<span style={{"color":"red"}}>{msg}</span>**

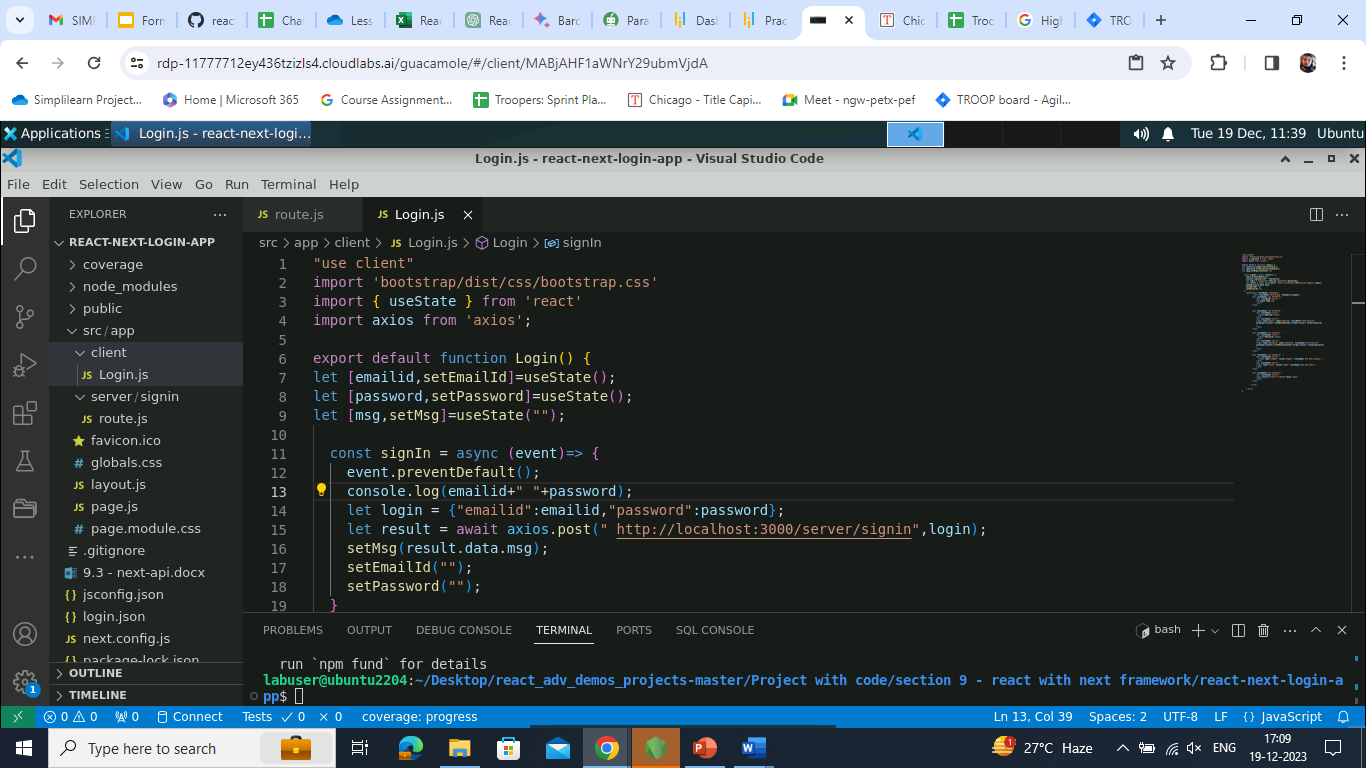
**</div>**

**</div>**

**</form>**

**</div>)**

**}**

****

**Note:** The Login.js file is responsible for interacting with the provided login page and server-side components with help of Axios third-party modules.

1. Create a **Layout.js** file and provide the client- and server-side components details using the following code:

**import { Inter } from 'next/font/google'**

**import './globals.css'**

**import Login from './client/Login'**

**const inter = Inter({ subsets: ['latin'] })**

**export const metadata = {**

**title: 'Create Next App',**

**description: 'Generated by create next app',**

**}**

**export default function RootLayout({ children }) {**

**return (**

**<html lang="en">**

**<body className={inter.className} suppressHydrationWarning={true}>**

**<Login></Login>**

**{children}**

**</body>**

**</html>**

**)**

**}**

A screenshot of a computer

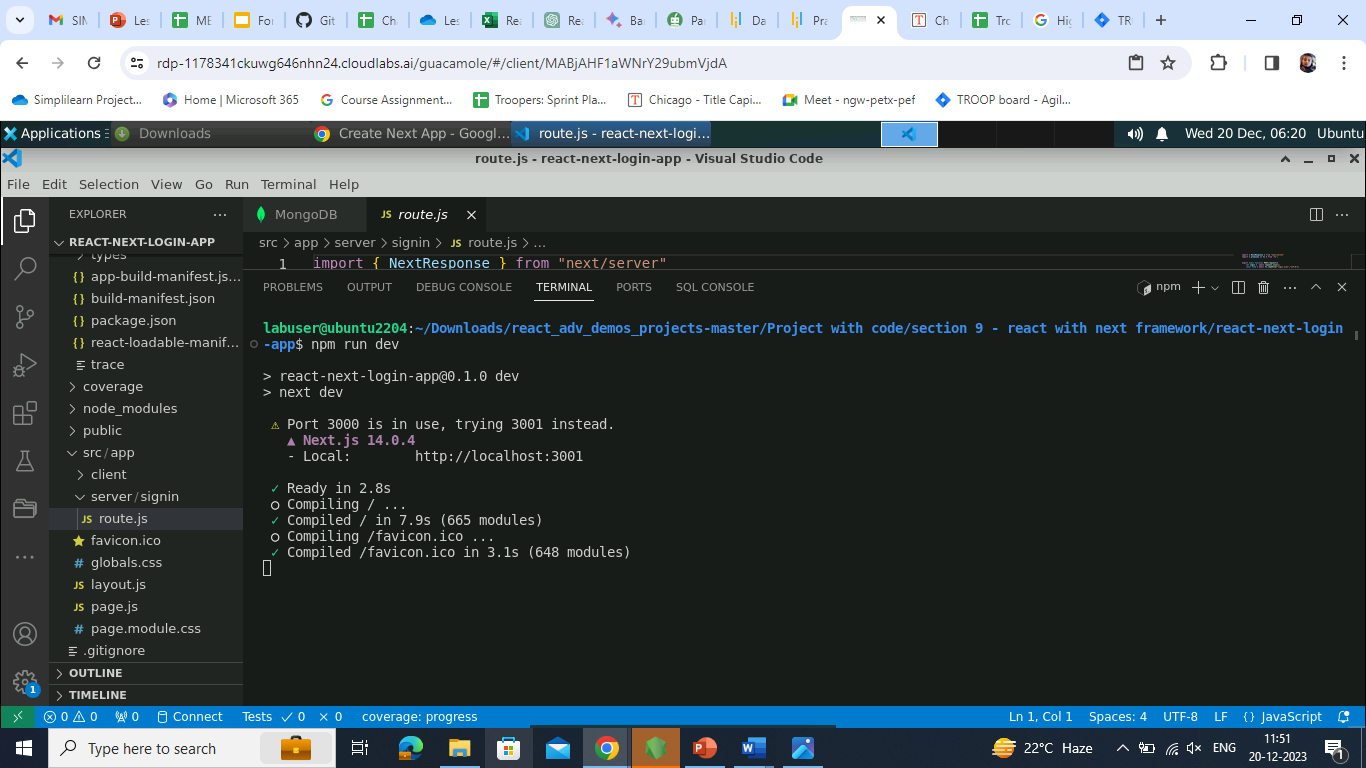
Description automatically generated

**Note:** Here, children's properties handle server-side components, while the login component manages client-side components.

**Step 5: Run the application**

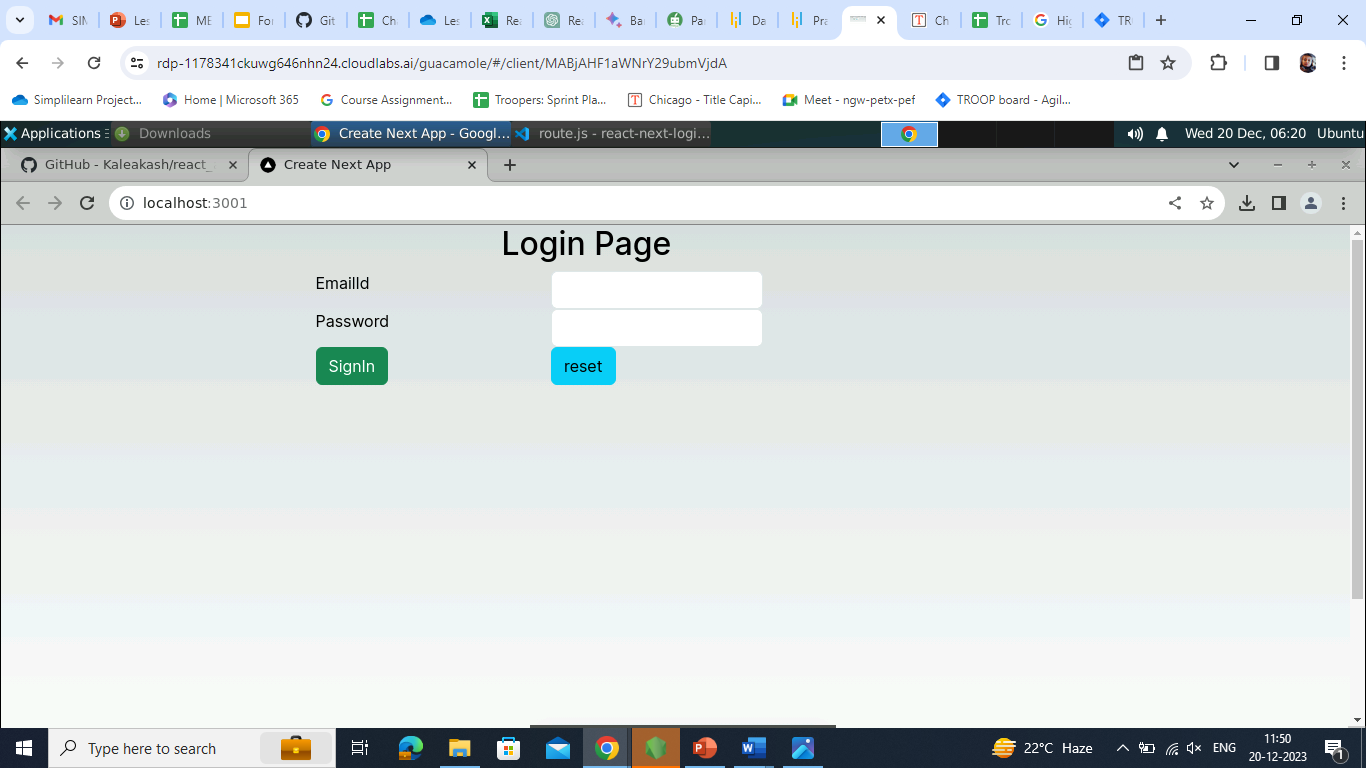
1. Run the following command in the terminal:

**npm run dev**

****

1. Open the browser and write the URL as shown:

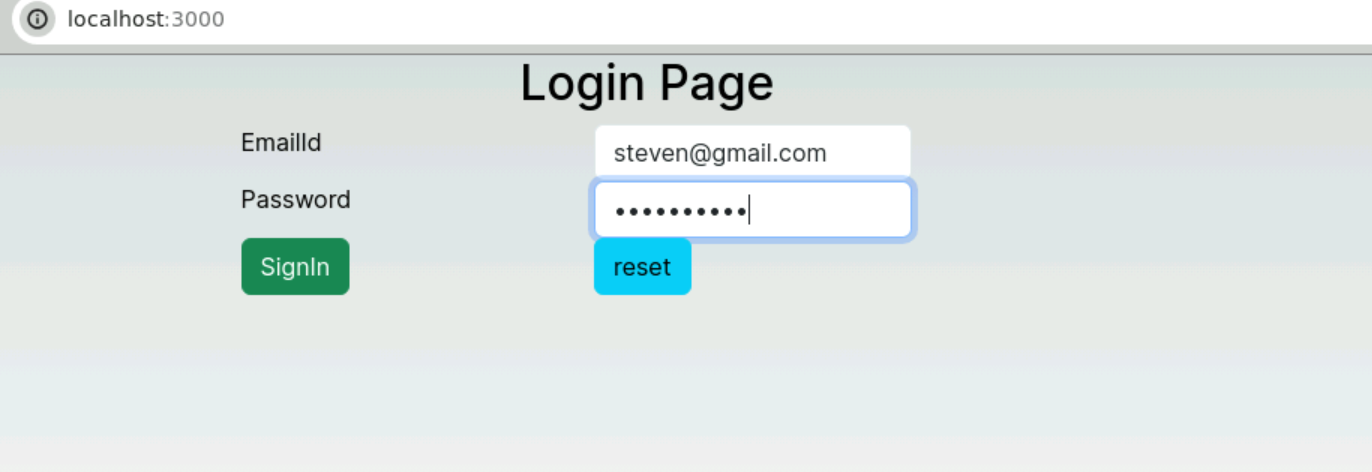
[**http://localhost:3000**](http://localhost:3000)



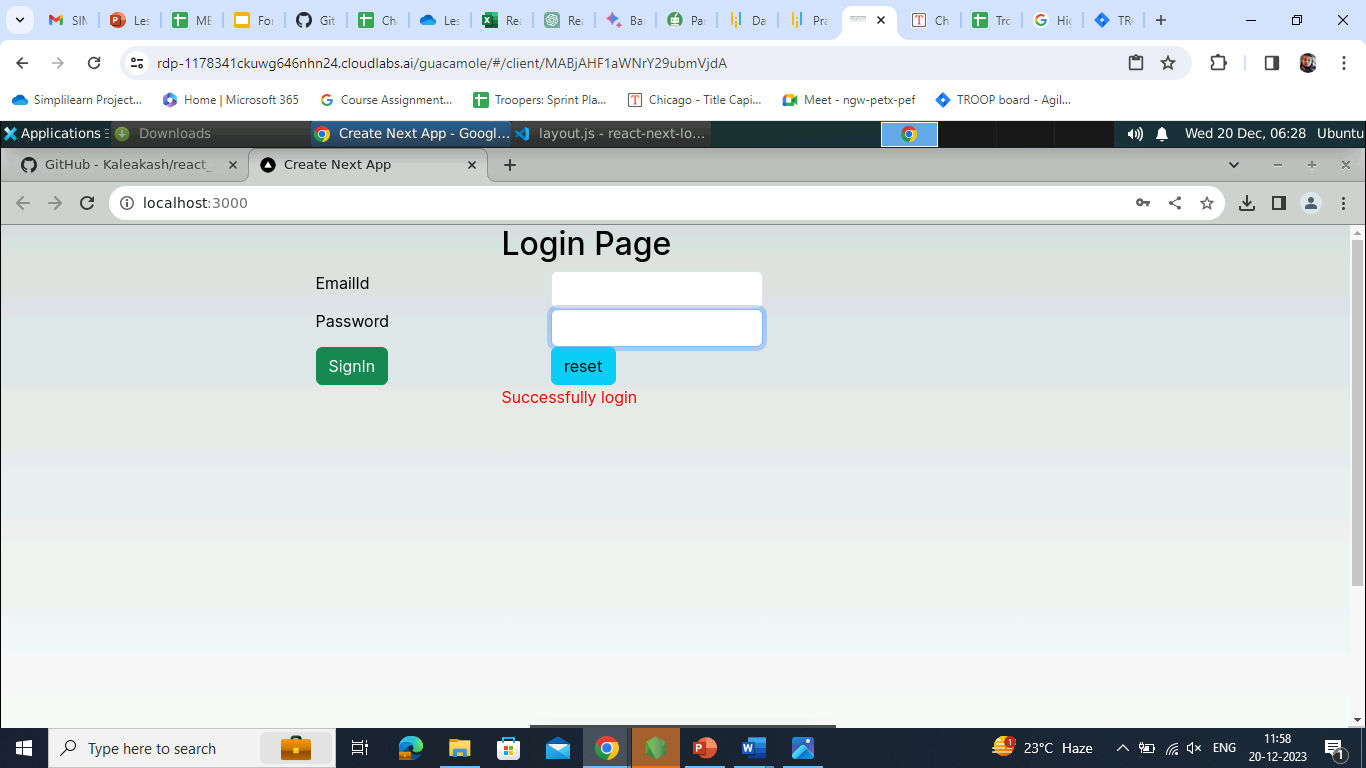
1. Check with the valid **EmailId** and **Password,** such as:

Emailld: steven@gmail.com, Password: steven@123

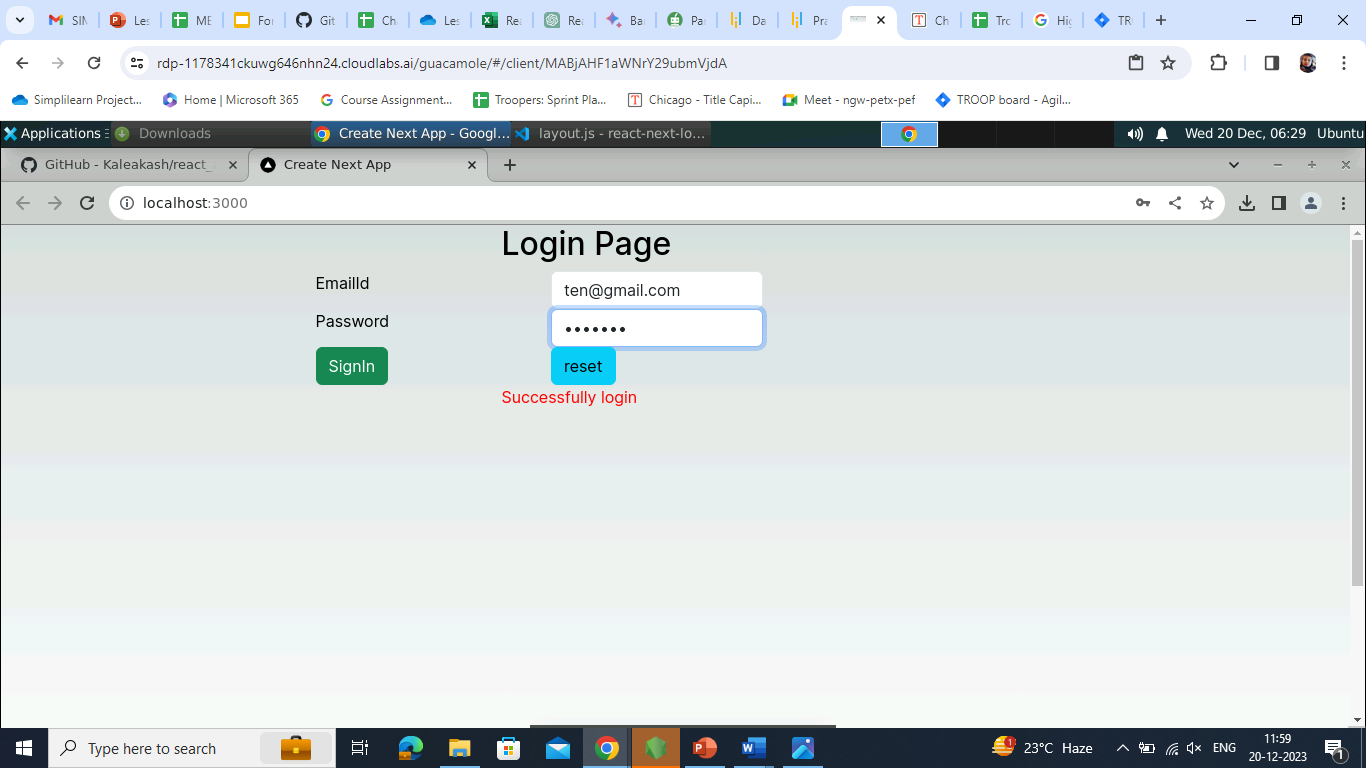
Emailld: raj@gmail.com, Password: raj@123



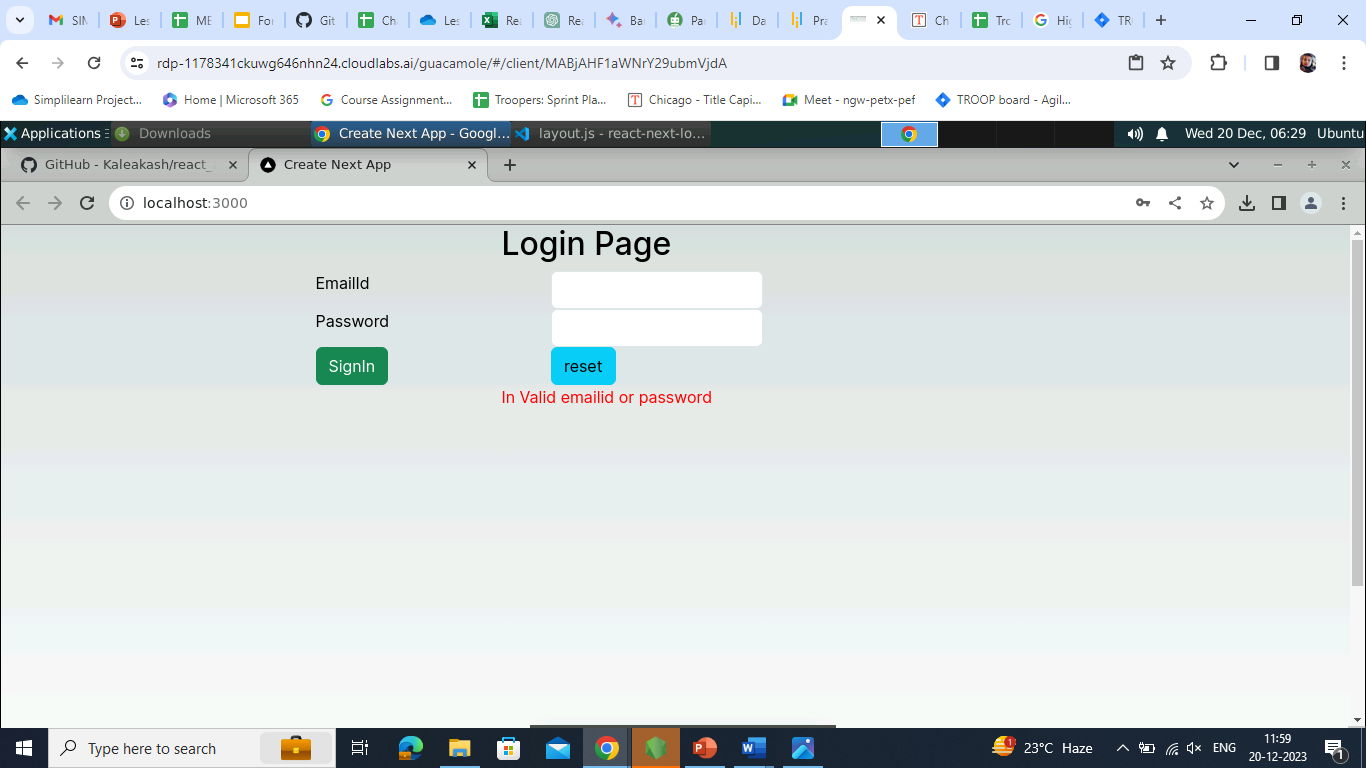
If the **EmailId** and **Password** are valid, it shows **Successfully login**



1. Check with the wrong **EmailId** or **Password**



You will get the following results:



With this, you have created a React Next.js login application, integrating server-side components for login, file handling, and testing with React components.