**CLIENT**

**1.PUBLIC**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<meta name="theme-color" content="#000000" />

<meta

name="description"

content="Web site created using create-react-app"

/>

<link rel="apple-touch-icon" href="%PUBLIC\_URL%/logo192.png" />

<!--

manifest.json provides metadata used when your web app is installed on a

user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/

-->

<link rel="manifest" href="%PUBLIC\_URL%/manifest.json" />

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<title>Freelancing App</title>

</head>

<body>

<noscript>You need to enable JavaScript to run this app.</noscript>

<div id="root"></div>

<!--

This HTML file is a template.

If you open it directly in the browser, you will see an empty page.

You can add webfonts, meta tags, or analytics to this file.

The build step will place the bundled scripts into the <body> tag.

To begin the development, run `npm start` or `yarn start`.

To create a production bundle, use `npm run build` or `yarn build`.

-->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

</body>

</html>

**Manifest.json**

{

"short\_name": "React App",

"name": "Create React App Sample",

"icons": [

{

"src": "favicon.ico",

"sizes": "64x64 32x32 24x24 16x16",

"type": "image/x-icon"

},

{

"src": "logo192.png",

"type": "image/png",

"sizes": "192x192"

},

{

"src": "logo512.png",

"type": "image/png",

"sizes": "512x512"

}

],

"start\_url": ".",

"display": "standalone",

"theme\_color": "#000000",

"background\_color": "#ffffff"

}

**Robots.txt**

# https://www.robotstxt.org/robotstxt.html

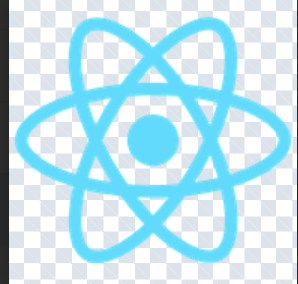
User-agent: \*

Disallow:

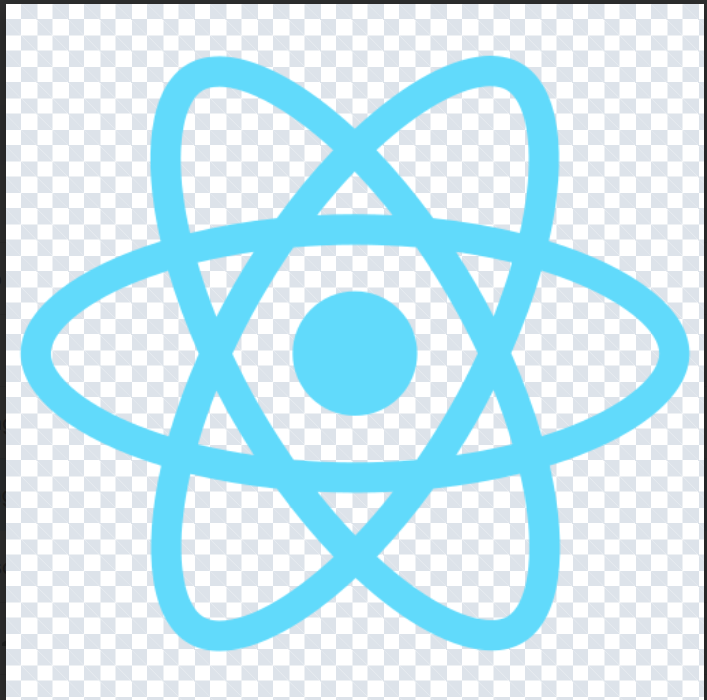
**Favicon.ico**

****

Logo192



Logo512



2.src

1. Components

* Login.jsx

1. import React, { useContext } from 'react'
2. import { GeneralContext } from '../context/GeneralContext';
3. const Login = ({setAuthType}) => {
4. const {setEmail, setPassword, login} = useContext(GeneralContext);
5. const handleLogin = async (e) =>{
6. e.preventDefault();
7. await login();
8. }
9. return (
10. <form className="authForm">
11. <h2>Login</h2>
12. <div className="form-floating mb-3 authFormInputs">
13. <input type="email" className="form-control" id="floatingInput" placeholder="name@example.com"
14. onChange={(e) => setEmail(e.target.value)} />
15. <label htmlFor="floatingInput">Email address</label>
16. </div>
17. <div className="form-floating mb-3 authFormInputs">
18. <input type="password" className="form-control" id="floatingPassword" placeholder="Password"
19. onChange={(e) => setPassword(e.target.value)} />
20. <label htmlFor="floatingPassword">Password</label>
21. </div>
22. <button type="submit" className="btn btn-primary" onClick={handleLogin}>Sign in</button>
23. <p>Not registered? <span onClick={()=> setAuthType('register')}>Register</span></p>
24. </form>
25. )
26. }
27. export default Login

* navbar.jsx

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

import '../styles/navbar.css'

import { useNavigate } from 'react-router-dom'

import { GeneralContext } from '../context/GeneralContext';

const Navbar = () => {

  const userId = localStorage.getItem('userId');

  const usertype = localStorage.getItem('usertype');

  const navigate = useNavigate();

  const {logout} = useContext(GeneralContext);

  return (

    <>

      {usertype === 'freelancer' ?

          <div className="navbar">

            <h3>SB Works</h3>

            <div className="nav-options">

                <p onClick={()=> navigate('/freelancer')} >Dashboard</p>

                <p onClick={()=> navigate('/all-projects')} >All Projects</p>

                <p onClick={()=> navigate('/my-projects')} >My Projects</p>

                <p onClick={()=> navigate('/myApplications')} >Applications</p>

                <p onClick={()=> logout()} >Logout</p>

            </div>

          </div>

      :

      ""}

      {

        usertype === 'client' ?

          <div className="navbar">

            <h3>SB Works</h3>

            <div className="nav-options">

                <p onClick={()=> navigate('/client')} >Dashboard</p>

                <p onClick={()=> navigate('/new-project')} >New Project</p>

                <p onClick={()=> navigate('/project-applications')} >Applications</p>

                <p onClick={()=> logout()}>Logout</p>

            </div>

          </div>

        :

        ""

      }

      {usertype === 'admin' ?

          <div className="navbar">

            <h3>SB Works (admin)</h3>

            <div className="nav-options">

                <p onClick={()=> navigate('/admin')} >Home</p>

                <p onClick={()=> navigate('/all-users')} >All users</p>

                <p onClick={()=> navigate('/admin-projects')} >Projects</p>

                <p onClick={()=> navigate('/admin-applications')} >Applications</p>

                <p onClick={()=> logout()}>Logout</p>

            </div>

          </div>

      : ""}

    </>

  )

}

export default Navbar

* register.jsx
* import React, { useContext } from 'react'
* import { GeneralContext } from '../context/GeneralContext';
* const Register = ({setAuthType}) => {
* const {setUsername, setEmail, setPassword, setUsertype, register} = useContext(GeneralContext);
* const handleRegister = async (e) =>{
* e.preventDefault();
* await register();
* }
* return (
* <form className="authForm">
* <h2>Register</h2>
* <div className="form-floating mb-3 authFormInputs">
* <input type="text" className="form-control" id="floatingInput" placeholder="username"
* onChange={(e)=> setUsername(e.target.value)} />
* <label htmlFor="floatingInput">Username</label>
* </div>
* <div className="form-floating mb-3 authFormInputs">
* <input type="email" className="form-control" id="floatingEmail" placeholder="name@example.com"
* onChange={(e)=> setEmail(e.target.value)} />
* <label htmlFor="floatingInput">Email address</label>
* </div>
* <div className="form-floating mb-3 authFormInputs">
* <input type="password" className="form-control" id="floatingPassword" placeholder="Password"
* onChange={(e)=> setPassword(e.target.value)} />
* <label htmlFor="floatingPassword">Password</label>
* </div>
* <select className="form-select form-select-lg mb-3" aria-label=".form-select-lg example"
* onChange={(e)=> setUsertype(e.target.value)}>
* <option value="">User type</option>
* <option value="freelancer">Freelancer</option>
* <option value="client">Client</option>
* <option value="admin">Admin</option>
* </select>
* <button className="btn btn-primary" onClick={handleRegister}>Sign up</button>
* <p>Already registered? <span onClick={()=> setAuthType('login')}>Login</span></p>
* </form>
* )
* }
* export default Register

1. context

generalcontext.jsx

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

import axios from "axios";

import { useNavigate } from "react-router-dom";

import socketIoClient from 'socket.io-client';

export const GeneralContext = createContext();

const GeneralContextProvider = ({children}) => {

  const WS = 'http://localhost:6001';

  const socket = socketIoClient(WS);

  const navigate = useNavigate();

  const [username, setUsername] = useState('');

  const [email, setEmail] = useState('');

  const [password, setPassword] = useState('');

  const [usertype, setUsertype] = useState('');

1. Images

landing-bg.jpg



1. Pages

* Admin
* Admin.jsx
* import React, { useEffect, useState } from 'react'
* import { useNavigate } from 'react-router-dom'
* import axios from 'axios';
* const Admin = () => {
* const navigate = useNavigate();
* const [projectsCount, setProjectsCount] = useState(0);
* const [completedProsCount, setCompletedProsCount] = useState(0);
* const [applicationsCount, setApplicationsCount] = useState(0);
* const [usersCount, setUsersCount] = useState(0);
* useEffect(()=>{
* fetchProjects();
* fetchApplications();
* fetchUsers();
* },[])
* const fetchProjects = async() =>{
* await axios.get("http://localhost:6001/fetch-projects").then(
* (response)=>{
* setProjectsCount(response.data.length);
* const comPros = response.data.filter((pro)=> pro.status === "Completed");
* setCompletedProsCount(comPros.length);
* }
* ).catch((err)=>{
* console.log(err);
* })
* }
* const fetchApplications = async() =>{
* await axios.get("http://localhost:6001/fetch-applications").then(
* (response)=>{
* setApplicationsCount(response.data.length);
* }
* ).catch((err)=>{
* console.log(err);
* })
* }
* const fetchUsers = async() =>{
* await axios.get("http://localhost:6001/fetch-users").then(
* (response)=>{
* setUsersCount(response.data.length);
* }
* ).catch((err)=>{
* console.log(err);
* })
* }
* return (
* <div className="admin-page">
* <div className="home-cards">
* <div className="home-card">
* <h4>All Projects</h4>
* <p>{projectsCount}</p>
* <button  onClick={()=> navigate('/admin-projects')}>View projects</button>
* </div>
* <div className="home-card">
* <h4>Completed projects</h4>
* <p>{completedProsCount}</p>
* <button  onClick={()=> navigate('/admin-projects')}>View projects</button>
* </div>
* <div className="home-card">
* <h4>Applications</h4>
* <p>{applicationsCount}</p>
* <button  onClick={()=> navigate('/admin-applications')}>View Applications</button>
* </div>
* <div className="home-card">
* <h4>Users</h4>
* <p>{usersCount}</p>
* <button  onClick={()=> navigate('/all-users')}>View Users</button>
* </div>
* </div>
* </div>
* )
* }
* export default Admin

reademe.md

**# Getting Started with Create React App**

This project was bootstrapped with [Create React App](https://github.com/facebook/create-react-app).

**## Available Scripts**

In the project directory, you can run:

**### `npm start`**

Runs the app in the development mode.\

Open [http://localhost:3000](http://localhost:3000) to view it in your browser.

The page will reload when you make changes.\

You may also see any lint errors in the console.

**### `npm test`**

Launches the test runner in the interactive watch mode.\

See the section about [running tests](https://facebook.github.io/create-react-app/docs/running-tests) for more information.

**### `npm run build`**

Builds the app for production to the `build` folder.\

It correctly bundles React in production mode and optimizes the build for the best performance.

The build is minified and the filenames include the hashes.\

Your app is ready to be deployed!

See the section about [deployment](https://facebook.github.io/create-react-app/docs/deployment) for more information.

**### `npm run eject`**

**\*\*Note: this is a one-way operation. Once you `eject`, you can't go back!\*\***

If you aren't satisfied with the build tool and configuration choices, you can `eject` at any time. This command will remove the single build dependency from your project.

Instead, it will copy all the configuration files and the transitive dependencies (webpack, Babel, ESLint, etc) right into your project so you have full control over them. All of the commands except `eject` will still work, but they will point to the copied scripts so you can tweak them. At this point you're on your own.

You don't have to ever use `eject`. The curated feature set is suitable for small and middle deployments, and you shouldn't feel obligated to use this feature. However we understand that this tool wouldn't be useful if you couldn't customize it when you are ready for it.

**## Learn More**

You can learn more in the [Create React App documentation](https://facebook.github.io/create-react-app/docs/getting-started).

To learn React, check out the [React documentation](https://reactjs.org/).

**### Code Splitting**

This section has moved here: [https://facebook.github.io/create-react-app/docs/code-splitting](https://facebook.github.io/create-react-app/docs/code-splitting)

**### Analyzing the Bundle Size**

This section has moved here: [https://facebook.github.io/create-react-app/docs/analyzing-the-bundle-size](https://facebook.github.io/create-react-app/docs/analyzing-the-bundle-size)

**### Making a Progressive Web App**

This section has moved here: [https://facebook.github.io/create-react-app/docs/making-a-progressive-web-app](https://facebook.github.io/create-react-app/docs/making-a-progressive-web-app)

**### Advanced Configuration**

This section has moved here: [https://facebook.github.io/create-react-app/docs/advanced-configuration](https://facebook.github.io/create-react-app/docs/advanced-configuration)

**### Deployment**

This section has moved here: [https://facebook.github.io/create-react-app/docs/deployment](https://facebook.github.io/create-react-app/docs/deployment)

**### `npm run build` fails to minify**

This section has moved here: [https://facebook.github.io/create-react-app/docs/troubleshooting#npm-run-build-fails-to-minify](https://facebook.github.io/create-react-app/docs/troubleshooting#npm-run-build-fails-to-minify)

PACKAGE.JSON

{

  "name": "client",

  "version": "0.1.0",

  "private": true,

  "dependencies": {

    "@testing-library/jest-dom": "^5.17.0",

    "@testing-library/react": "^13.4.0",

    "@testing-library/user-event": "^13.5.0",

    "axios": "^1.5.1",

    "react": "^18.2.0",

    "react-dom": "^18.2.0",

    "react-icons": "^4.11.0",

    "react-router-dom": "^6.19.0",

    "react-scripts": "5.0.1",

    "socket.io-client": "^4.7.2",

    "web-vitals": "^2.1.4"

  },

  "scripts": {

    "start": "react-scripts start",

    "build": "react-scripts build",

    "test": "react-scripts test",

    "eject": "react-scripts eject"

  },

  "eslintConfig": {

    "extends": [

      "react-app",

      "react-app/jest"

    ]

  },

  "browserslist": {

    "production": [

      ">0.2%",

      "not dead",

      "not op\_mini all"

    ],

    "development": [

      "last 1 chrome version",

      "last 1 firefox version",

      "last 1 safari version"

    ]

  }

}

.gitignore

**# See https://help.github.com/articles/ignoring-files/ for more about ignoring files.**

**# dependencies**

/node\_modules

/.pnp

.pnp.js

**# testing**

/coverage

**# production**

/build

**# misc**

.DS\_Store

.env.local

.env.development.local

.env.test.local

.env.production.local

npm-debug.log\*

yarn-debug.log\*

yarn-error.log\*

app.jss

/\* color: #ee4950 \*/

/\* color: #080808 \*/

body{

    /\* background-color: #141414; \*/

    /\* color: aliceblue; \*/

}

App.js

import './App.css';

import { Route, Routes } from 'react-router-dom';

import Navbar from './components/Navbar';

import Landing from './pages/Landing';

import Authenticate from './pages/Authenticate';

import Freelancer from './pages/freelancer/Freelancer'

import AllProjects from './pages/freelancer/AllProjects'

import MyProjects from './pages/freelancer/MyProjects'

import MyApplications from './pages/freelancer/MyApplications'

import ProjectData from './pages/freelancer/ProjectData'

import Client from './pages/client/Client'

import ProjectApplications from './pages/client/ProjectApplications'

import NewProject from './pages/client/NewProject'

import ProjectWorking from './pages/client/ProjectWorking'

import Admin from './pages/admin/Admin'

import AdminProjects from './pages/admin/AdminProjects'

import AllApplications from './pages/admin/AllApplications'

import AllUsers from './pages/admin/AllUsers'

function App() {

  return (

    <div className="App">

      <Navbar />

     <Routes>

        <Route exact path='/' element={<Landing />} />

        <Route path='/authenticate' element={<Authenticate />} />

        <Route path='/freelancer' element={<Freelancer />} />

        <Route path='/all-projects' element={<AllProjects />} />

        <Route path='/my-projects' element={<MyProjects />} />

        <Route path='/myApplications' element={<MyApplications />} />

        <Route path='/project/:id' element={<ProjectData />} />

        <Route path='/client' element={<Client />} />

        <Route path='/project-applications' element={<ProjectApplications />} />

        <Route path='/new-project' element={<NewProject />} />

        <Route path='/client-project/:id' element={<ProjectWorking />} />

        <Route path='/admin' element={<Admin />} />

        <Route path='/admin-projects' element={<AdminProjects />} />

        <Route path='/admin-applications' element={<AllApplications />} />

        <Route path='/all-users' element={<AllUsers />} />

     </Routes>

    </div>

  );

}

export default App;

app.text.jss

import { render, screen } from '@testing-library/react';

import App from './App';

test('renders learn react link', () => {

  render(<App />);

  const linkElement = screen.getByText(/learn react/i);

  expect(linkElement).toBeInTheDocument();

});

Index.css

body {

  margin: 0;

  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',

    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',

    sans-serif;

  -webkit-font-smoothing: antialiased;

  -moz-osx-font-smoothing: grayscale;

}

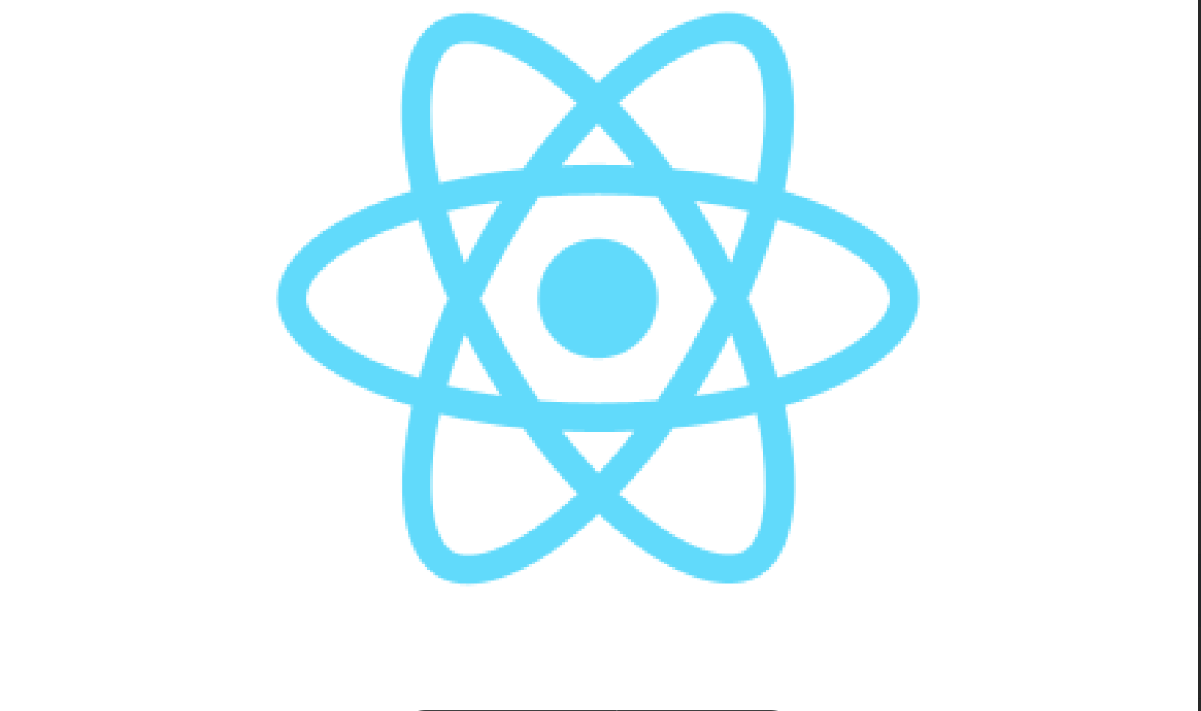
code {

  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',

    monospace;

}

Logo.svg



Reportwebvitals.js

const reportWebVitals = onPerfEntry => {

  if (onPerfEntry && onPerfEntry instanceof Function) {

    import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {

      getCLS(onPerfEntry);

      getFID(onPerfEntry);

      getFCP(onPerfEntry);

      getLCP(onPerfEntry);

      getTTFB(onPerfEntry);

    });

  }

};

export default reportWebVitals;

setuptests.js

const reportWebVitals = onPerfEntry => {

  if (onPerfEntry && onPerfEntry instanceof Function) {

    import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {

      getCLS(onPerfEntry);

      getFID(onPerfEntry);

      getFCP(onPerfEntry);

      getLCP(onPerfEntry);

      getTTFB(onPerfEntry);

    });

  }

};

export default reportWebVitals;