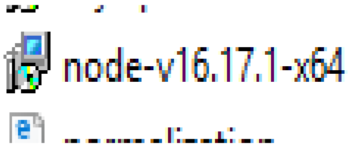
First of all download nodejs from

https://nodejs.org/en/download/



node-v16.17.1-x64

after downloading you will see setup file

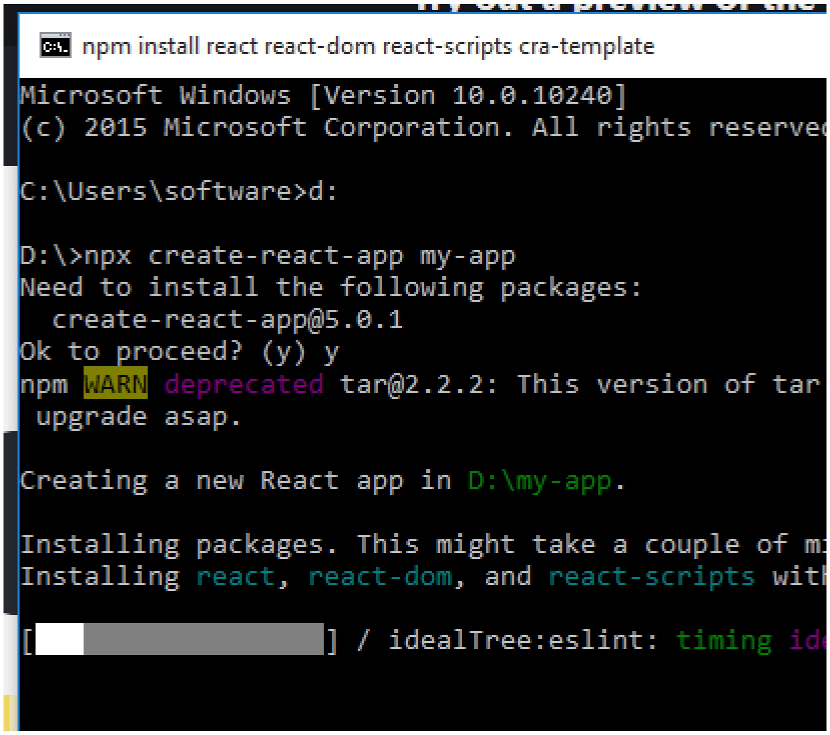
click on it will install nodejs.

After it open your command prompt

And in your command prompt type command

npx create-react-app my-app

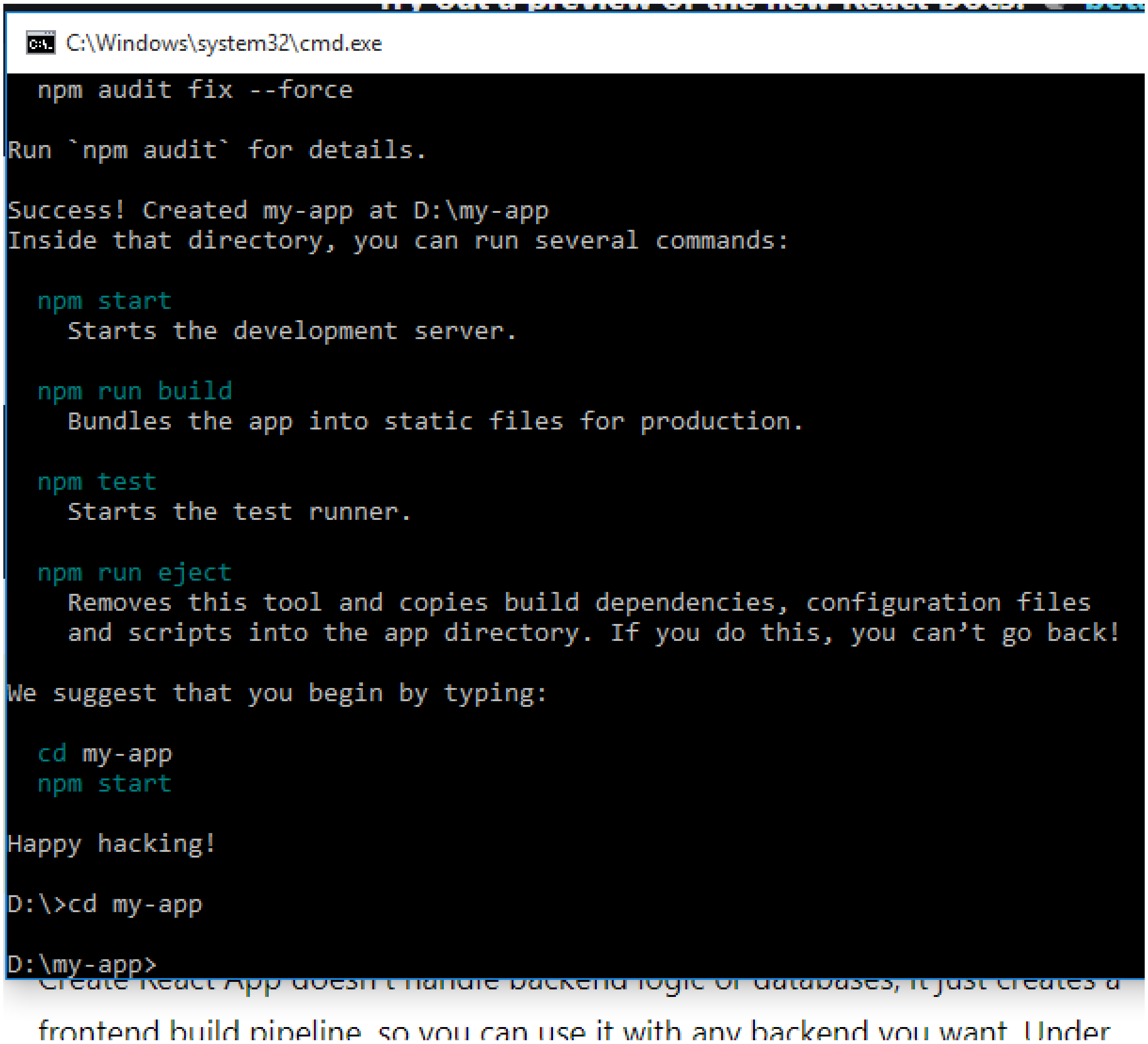
as shown below



npm install react react-dom react-scripts cra-template

```
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserve

C:\Users\software>d:

D:\>npx create-react-app my-app
Need to install the following packages:
  create-react-app@5.0.1
Ok to proceed? (y) y
npm WARN deprecated tar@2.2.2: This version of tar
 upgrade asap.

Creating a new React app in D:\my-app.

Installing packages. This might take a couple of mi
Installing react, react-dom, and react-scripts wit

[                  ] / idealTree:eslint: timing ide
```

After installation you will see following



C:\Windows\system32\cmd.exe

```
  npm audit fix --force

Run `npm audit` for details.

Success! Created my-app at D:\my-app
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can’t go back!

We suggest that you begin by typing:

  cd my-app
  npm start

Happy hacking!

D:\>cd my-app

D:\my-app>
```

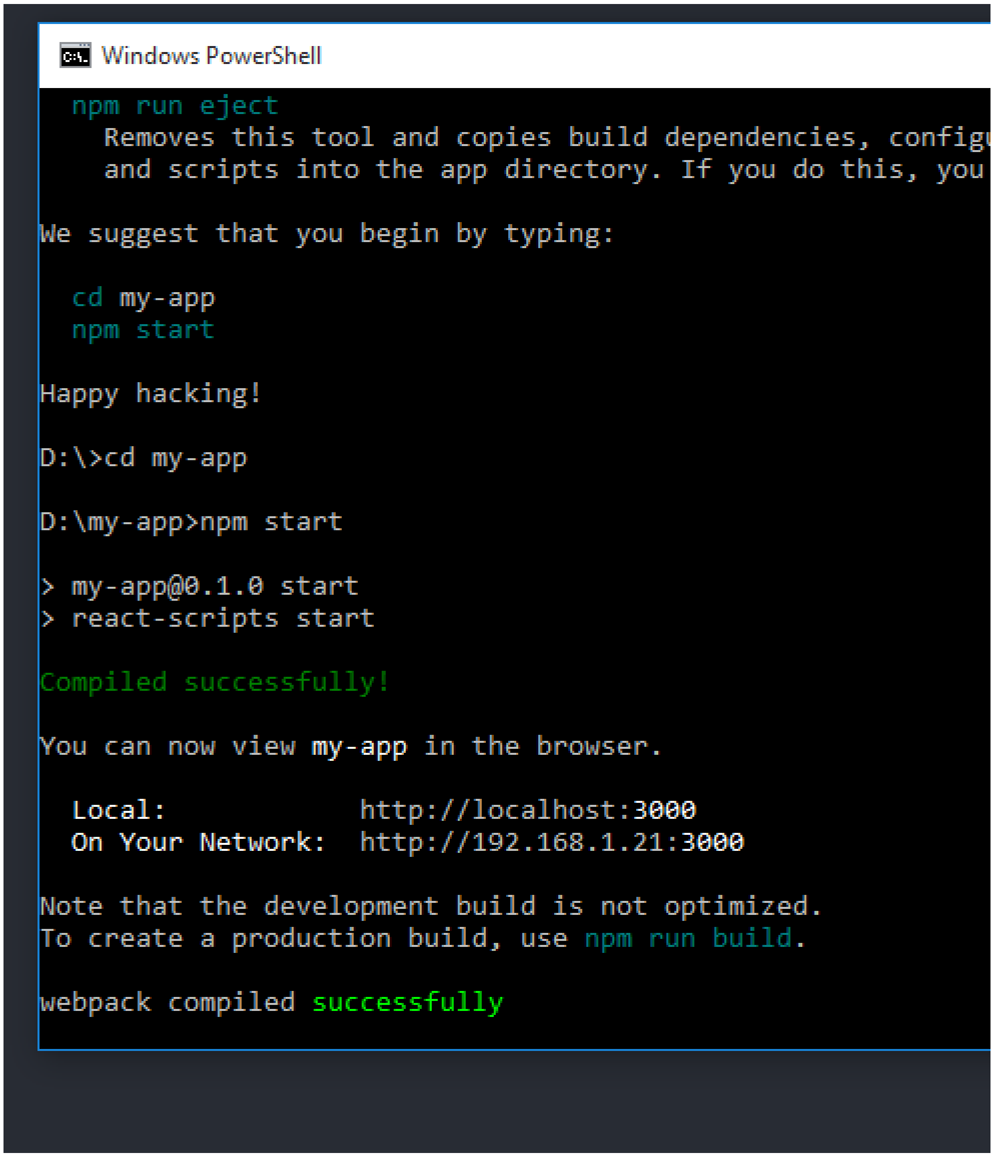
Create React App doesn't handle backend logic or databases, it just creates a frontend build pipeline, so you can use it with any backend you want. Under

Here you will go to your app folder by command Cd as shown above.

Cd my-app

After it type command

npm start as shown below :-

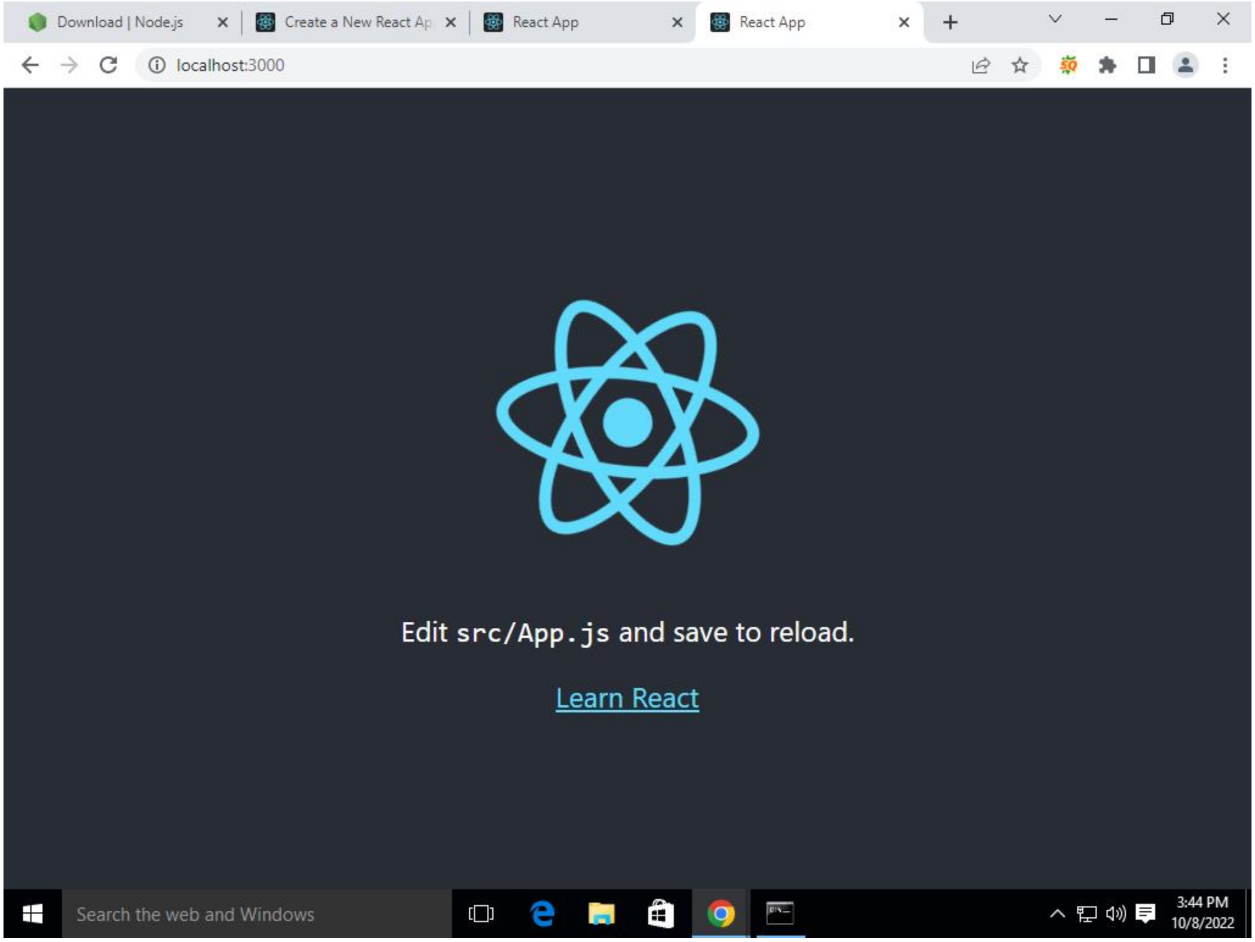


```
  npm run eject
    Removes this tool and copies build dependencies, configu
    and scripts into the app directory. If you do this, you

We suggest that you begin by typing:

  cd my-app
  npm start

Happy hacking!

D:\>cd my-app

D:\my-app>npm start

> my-app@0.1.0 start
> react-scripts start

Compiled successfully!

You can now view my-app in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.1.21:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

After it Open your browser and type

http://localhost:3000/

you will see default output :-



Edit src/App.js and save to reload.

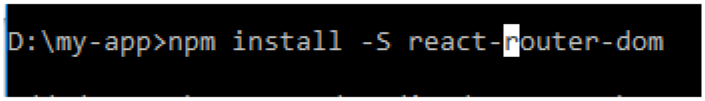
React Router:-

Create React App doesn't include page routing.

React Router is the most popular solution.

Add React Router

To add React Router in your application, run this in the terminal from the root directory of the application as shown below:-



```
D:\my-app>npm install -S react-router-dom
```

Folder Structure

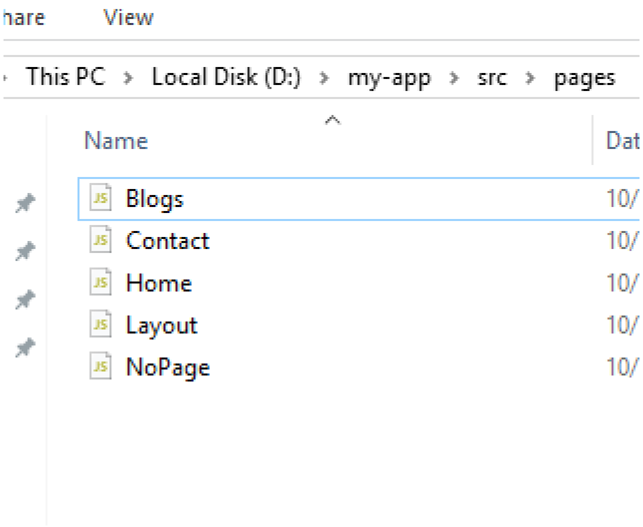
To create an application with multiple page routes, let's first start with the file structure.

Within the `src` folder of your my-app , we'll create a folder named `pages` with several files:

`src\pages\`:

- `Layout.js`
- `Home.js`
- `Blogs.js`
- `Contact.js`
- `NoPage.js`

`As shown below:-`



Each file will contain a very basic React component.

Write code for Layout.js file   in   pages folder.

```
import { Outlet, Link } from "react-router-dom";

const Layout = () => {
  return (
    <>
      <nav>
        <ul>
          <li>
            <Link to="/">Home</Link>
          </li>
          <li>
            <Link to="/blogs">Blogs</Link>
          </li>
          <li>
            <Link to="/contact">Contact</Link>
          </li>
        </ul>
      </nav>

      <Outlet />
    </>
  )
};

export default Layout;
```

Write code for Home.js file in pages folder :-

```
const Home = () => {
    return <h1>Home</h1>;
  };

  export default Home;
```

write code for Blogs.js file in pages folder :-

```
const Blogs = () => {
  return <h1>Blog Articles</h1>;
};

export default Blogs;
```

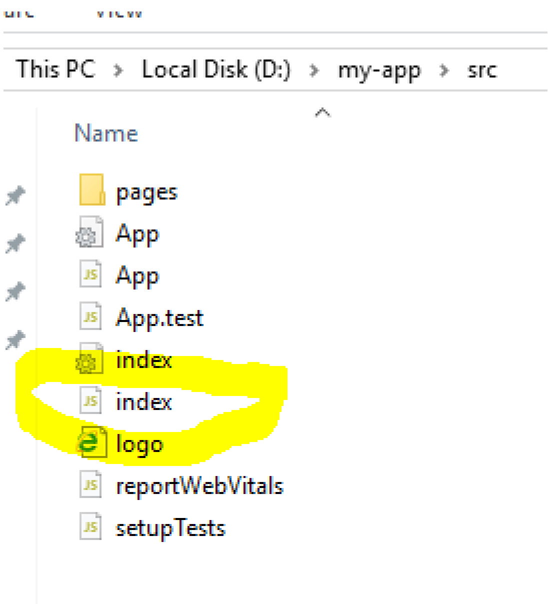
Write code for Contact.js file in pages folder :-

```
const Contact = () => {
  return <h1>Contact Me</h1>;
};

export default Contact;
```

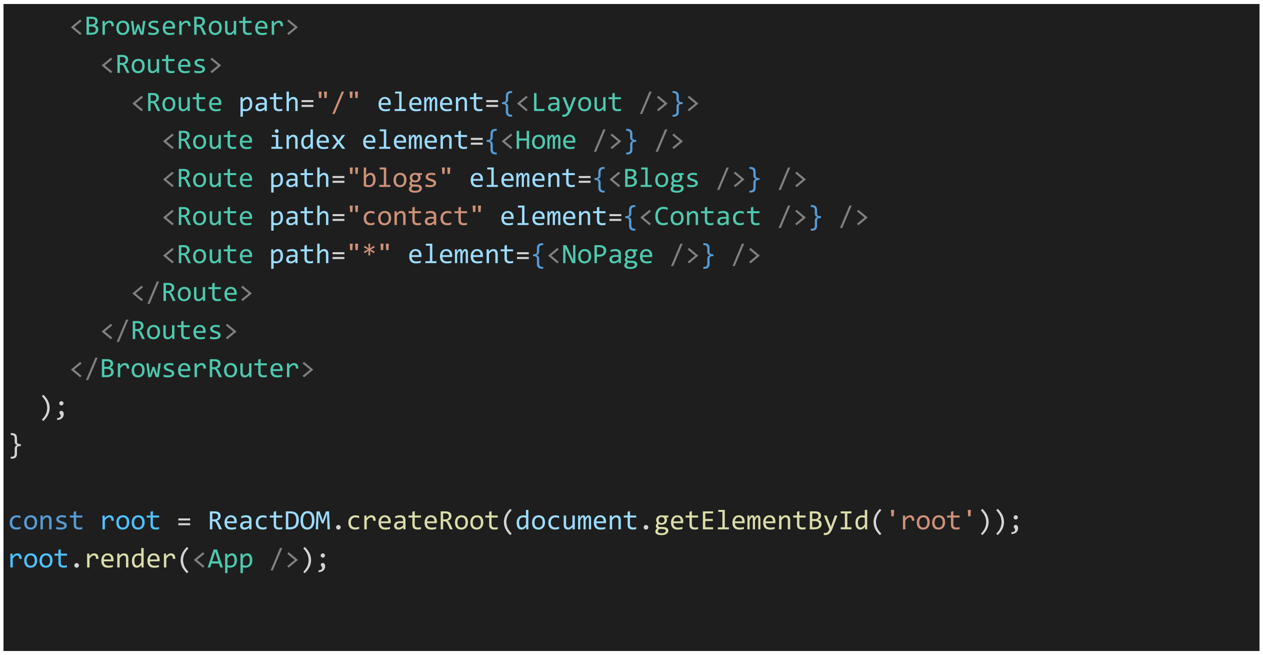
Write code for NoPage.js file in pages folder :-

```
const NoPage = () => {
  return <h1>404</h1>;
};

export default NoPage;
```

Now open your index.js file in your src folder of your my-app (which is main folder) as shown below:-

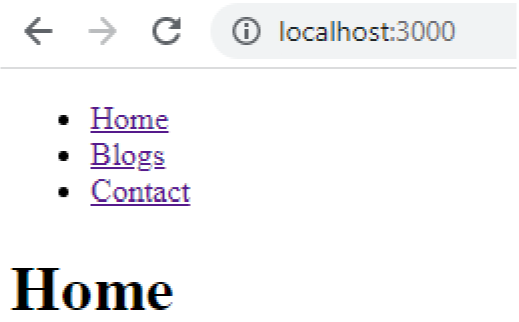


```
import ReactDOM from "react-dom/client";
import { BrowserRouter, Routes, Route } from "react-router-dom";
import Layout from "./pages/Layout";
import Home from "./pages/Home";
import Blogs from "./pages/Blogs";
import Contact from "./pages/Contact";
import NoPage from "./pages/NoPage";

export default function App() {
  return (
```



```
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<Layout />}>
          <Route index element={<Home />} />
          <Route path="blogs" element={<Blogs />} />
          <Route path="contact" element={<Contact />} />
          <Route path="*" element={<NoPage />} />
        </Route>
      </Routes>
    </BrowserRouter>
  );
}

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<App />);
```

finally go to browser and check output as shown below :-



- Home
- Blogs
- Contact

Home