

React/MongoDB

Full Stack

Project Starter Notes

Folder Structure

>  product-manager • → Create overall project folder

create/initialize front end project folder

```
TERMINAL
FullStack\product-manager> npx react-create-app client
```

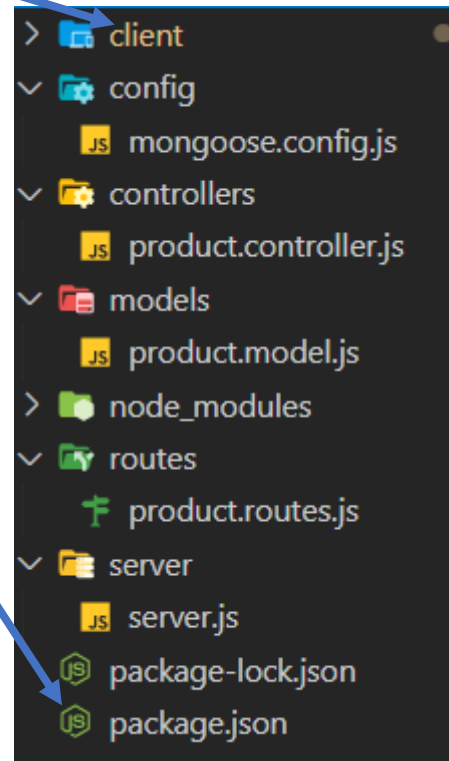
→ Inside your project folder run the following commands

→ npx react-create-app client

→ npm init -y

Initialize backend

```
TERMINAL
FullStack\product-manager> npm init -y
```



→ Also create the following folders: with the corresponding files inside.

→ config → mongoose.config.js

→ controllers → project.controller.js

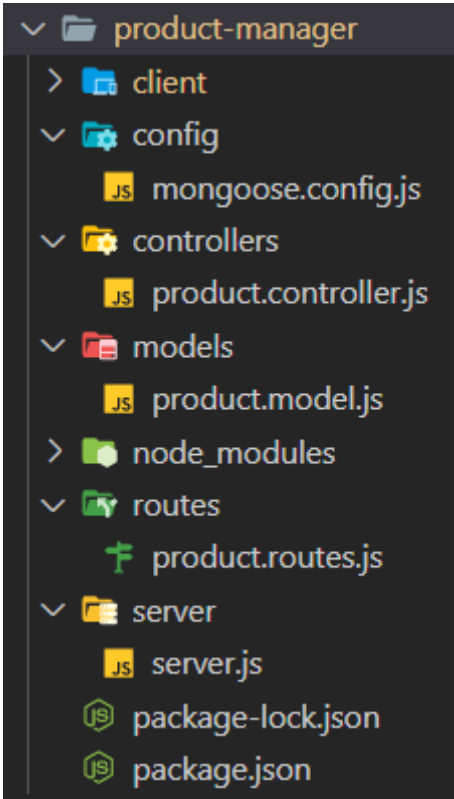
→ models → project.model.js

→ routes → project.routes.js

→ server → server.js

Folder Structure

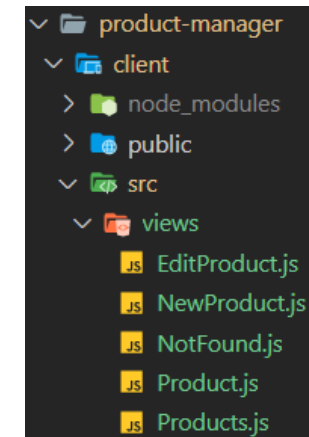
Here's a cleaner look



```
PS C:\Users\homec\Desktop\MERN\FullStack\product-manager> npx create-react-app client
```

```
PS C:\Users\homec\Desktop\MERN\FullStack\product-manager> npm init -y
```

Also in client -> src -> create views folder



Install dependencies

(make sure you're in the correct folders)

In the MAIN project folder

```
PS C:\Users\homec\Desktop\MERN\FullStack\product-manager> npm i cors express mongoose
```

Cd into client folder

```
PS C:\Users\homec\Desktop\MERN\FullStack\product-manager> cd client  
PS C:\Users\homec\Desktop\MERN\FullStack\product-manager\client> npm i @reach/router axios bootstrap
```

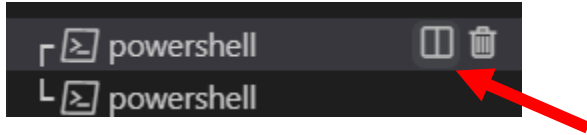
If using bootstrap you also need to into:

Client -> src -> index.js and add this line to the top

```
import "bootstrap/dist/css/bootstrap.css";
```

Start server and react

Easiest way is split terminal



One side cd into front end folder
npm start

```
PS C:\Users\homec\Desktop\MERN\FullStack\product-manager\client> npm start
```

Other side cd into server folder with server.js file
nodemon server.js

```
PS C:\Users\homec\Desktop\MERN\FullStack\product-manager\server> nodemon server.js
```