|  |
| --- |
|  |
|  |

**Project Setup and Configuration**

**8.1 Folder Structure**

To begin building the project from scratch, create two main folders:

* **client** — for the frontend (React.js)
* **server** — for the backend (Node.js, Express.js)

**8.2 Frontend Setup**

Navigate to the client folder and install the required libraries:

* React.js
* Bootstrap
* Axios

These tools help in building dynamic and responsive user interfaces. After installation, your package.json file will include dependencies like:

* "react": "^18.2.0"
* "bootstrap": "^5.3.1"
* "axios": "^1.4.0"
* "react-router-dom": "^6.14.2"
* "react-scripts": "5.0.1"

**8.3 Backend Setup**

Navigate to the server folder and install the following libraries:

* express
* mongoose
* bcrypt
* cors
* body-parser

These libraries are used for backend server management, database operations, encryption, and handling cross-origin requests. Your package.json will include entries like:

* "express": "^4.18.2"
* "mongoose": "^7.4.1"
* "bcrypt": "^5.1.0"
* "body-parser": "^1.20.2"
* "cors": "^2.8.5"

**Project setup and configuration**

**Folder setup:**

To start the project from scratch, firstly create frontend and backend folders to install essential libraries and write code.

* client
* Server

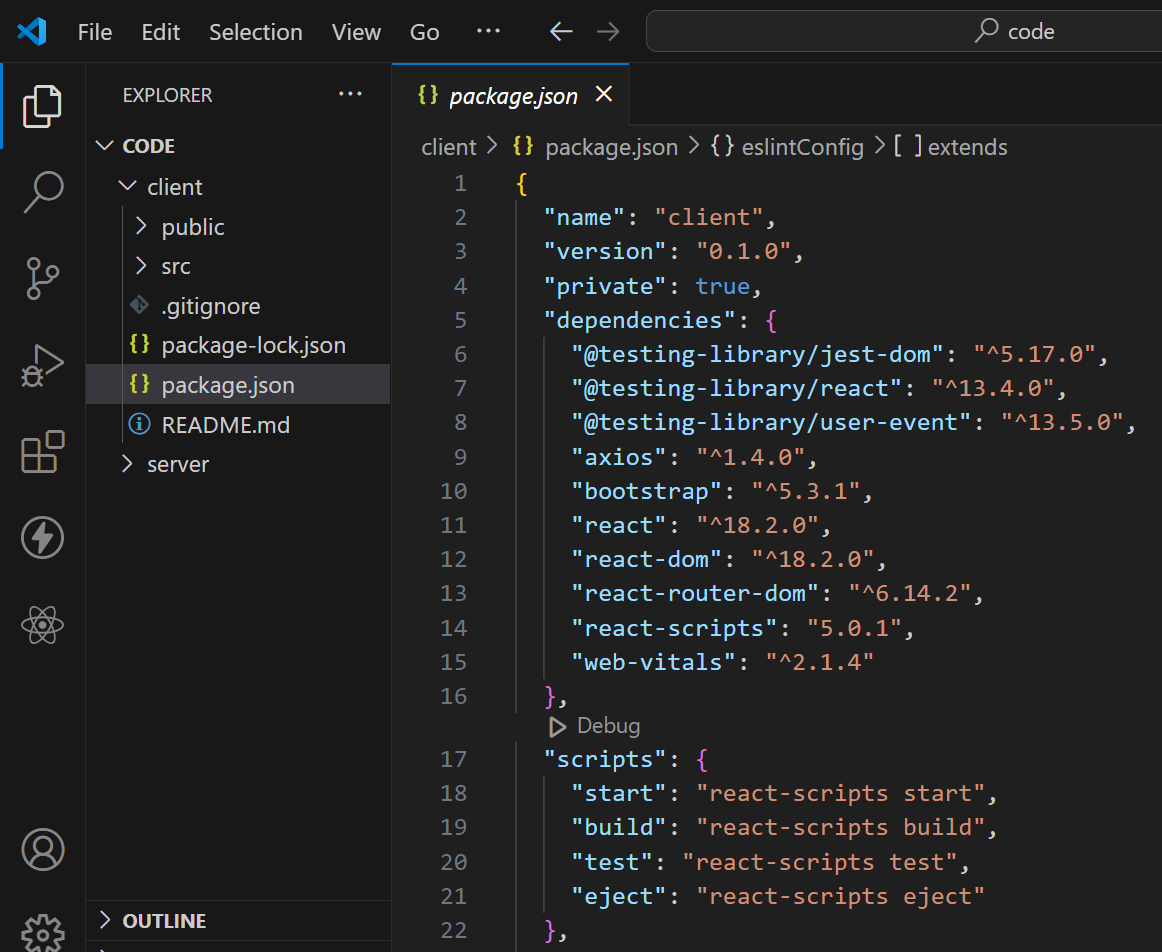
**Installation of required tools:**

Now, open the frontend folder to install all the necessary tools we use.

For frontend, we use:

* React Js
* Bootstrap
* Axios

After installing all the required libraries, we’ll be seeing the package.json file similar to the one below.

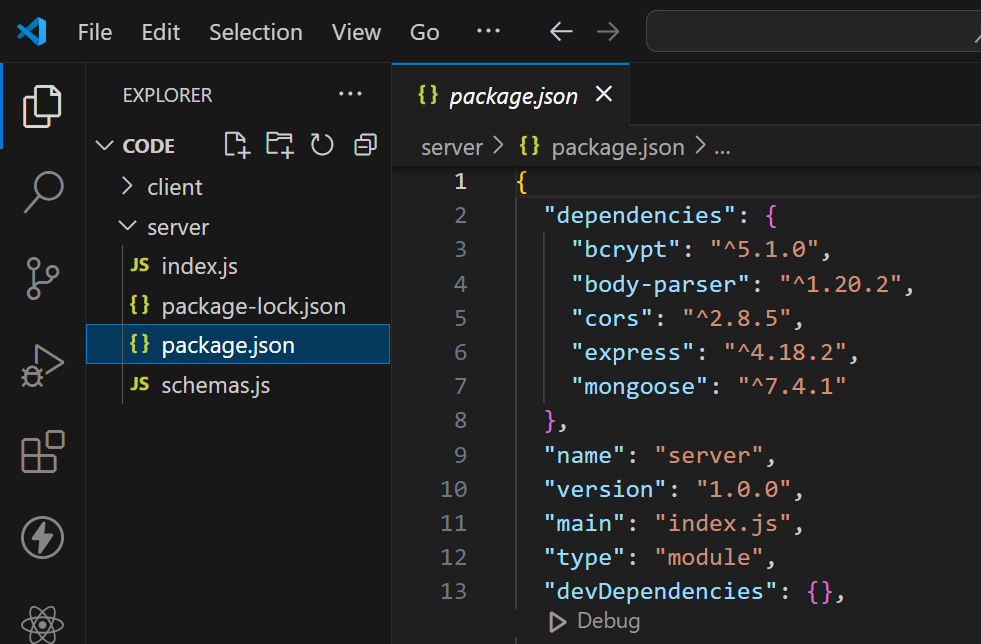


Now, open the backend folder to install all the necessary tools that we use in the backend.

For backend, we use:

* bcrypt
* body-parser
* cors
* express
* mongoose

After installing all the required libraries, we’ll be seeing the package.json file similar to the one below.

****