# **BÁO CÁO CT449**

# **ỨNG DỤNG TODO LIST**

Xây dựng ứng dụng Todo List (danh sách những việc bạn cần làm hoặc muốn làm) bạ theo mô hình Ứng dụng trang đơn (SPA) sử dụng công nghệ Node, Express, MongoDB phía backend (API server) và Vue phía frontend (GUI).

Điều kiện tiên quyết để thực hiện:

* [Nodejs v14](https://nodejs.org/en/blog/release/v14.17.3/)
* [Vue CLI v5](https://cli.vuejs.org/)
* [Vue v3](https://vuejs.org/)
* [nodemon](https://www.npmjs.com/package/nodemon)
* [concurrently](https://www.npmjs.com/package/concurrently)

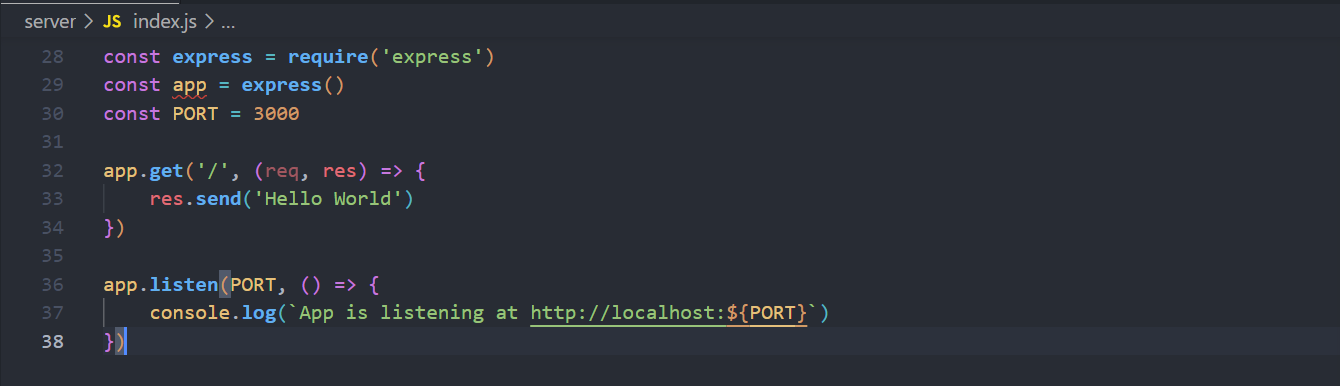
## **Backend**

* **Tạo Server với Nodejs và Express**

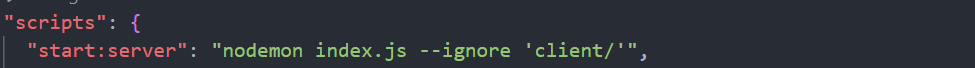
Tạo thư mục server bên trong thư mục dự án. Vào /server và khởi tạo dự án npm bằng lệnh sau: *npm init –y*

Cài đặt các gói cần thiết: *npm i express mongoose body-parser cors dotenv*

Truy cập vào thư mục index.js, thiết lập máy chủ hello world cơ bản:

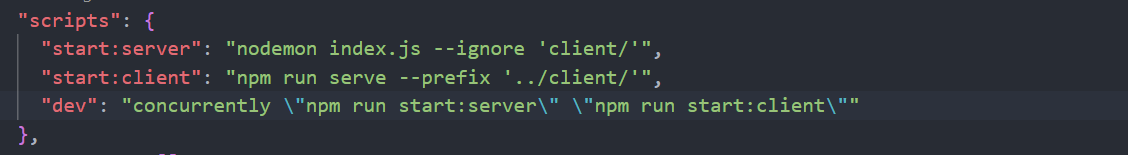


Cài đặt nodemon và concurrently và cập nhật tập lệnh scripts trong package.json: *npm i -D nodemon concurrently*



Bất cứ khi nào có những thay đổi mới trong mã ứng dụng, nó sẽ tự động làm mới ứng dụng của.

Cập nhật scripts để chạy song song cả client và server:



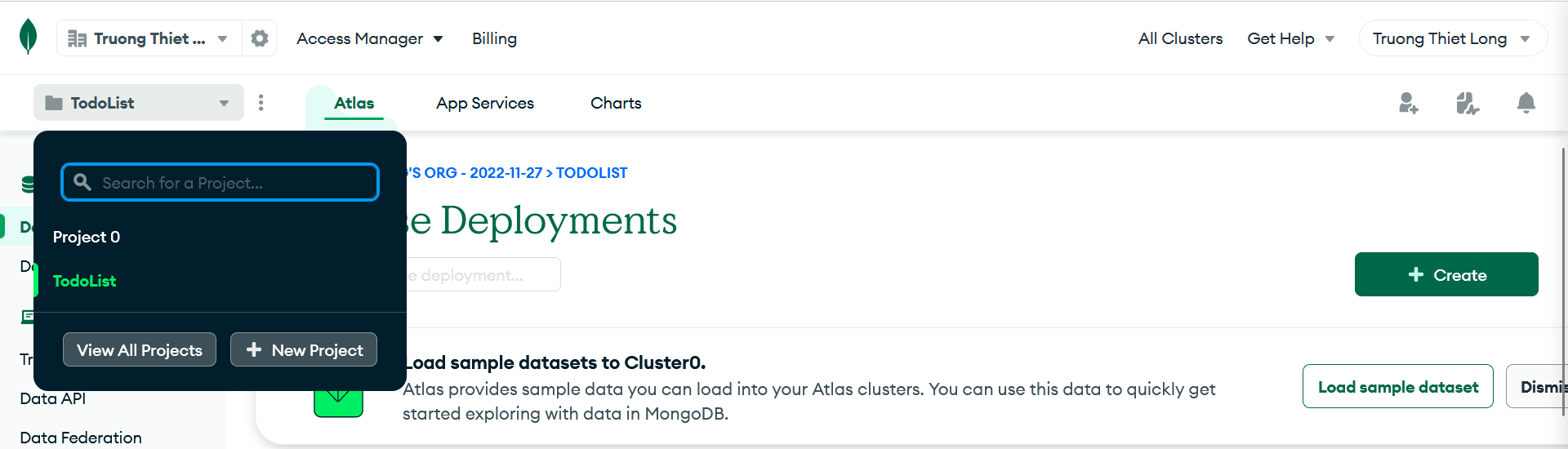
Sau đó, kết nối máy chủ với MongoDB thông qua mongoose. Thêm đoạn code sau vào file index.js



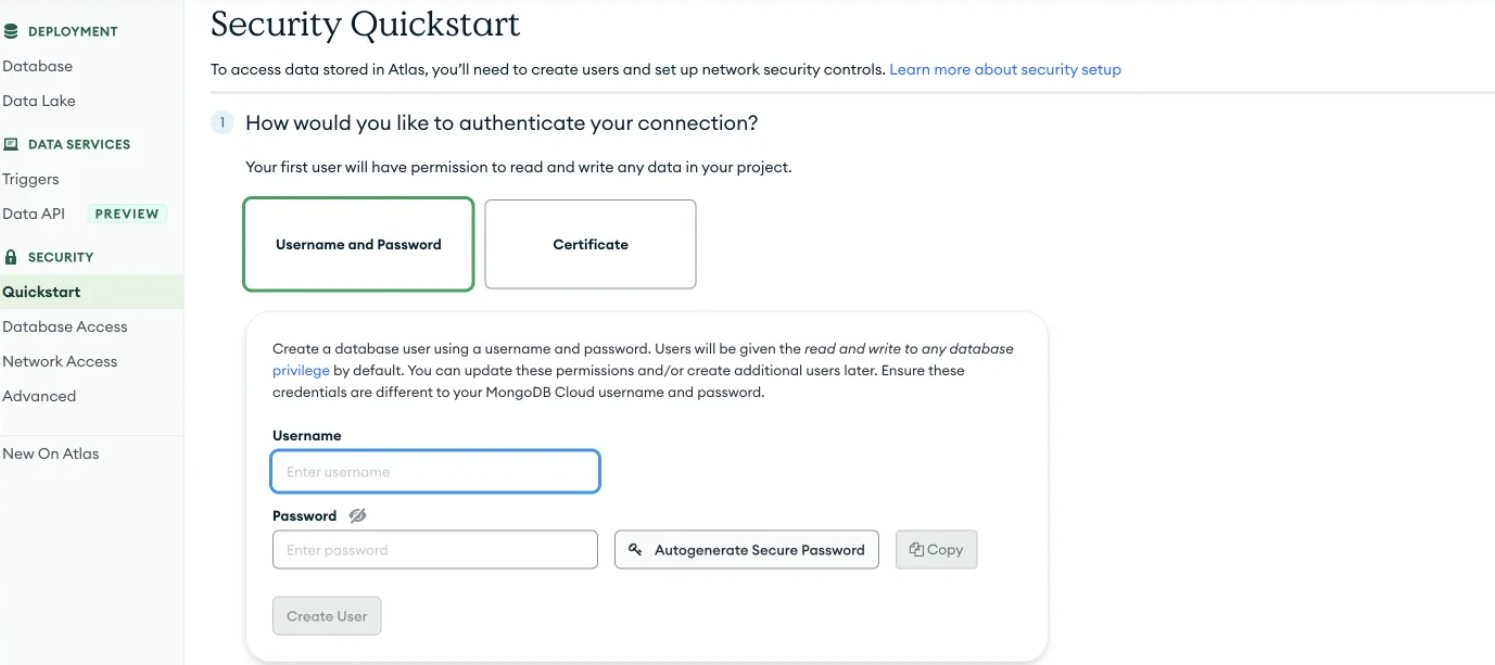
* **Tạo MongoDB User và kết nối cơ sở dữ liệu**

Truy cập <https://www.mongodb.com/atlas/database> và đăng nhập.

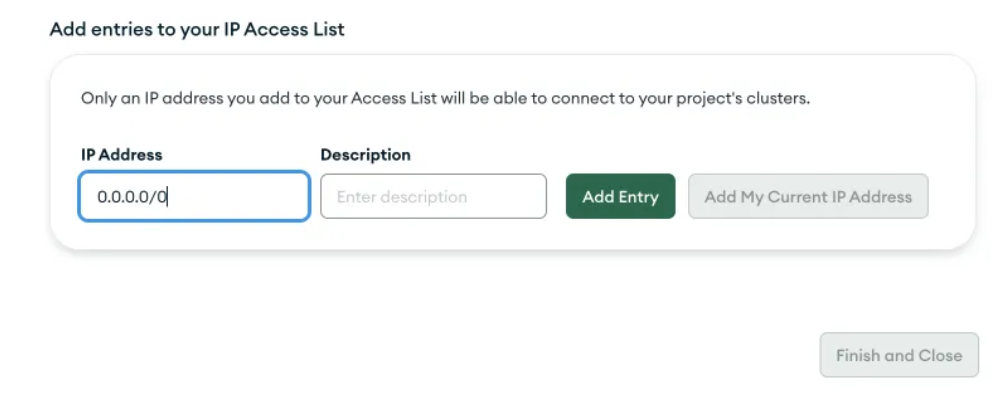
Tạo project tên TodoList:



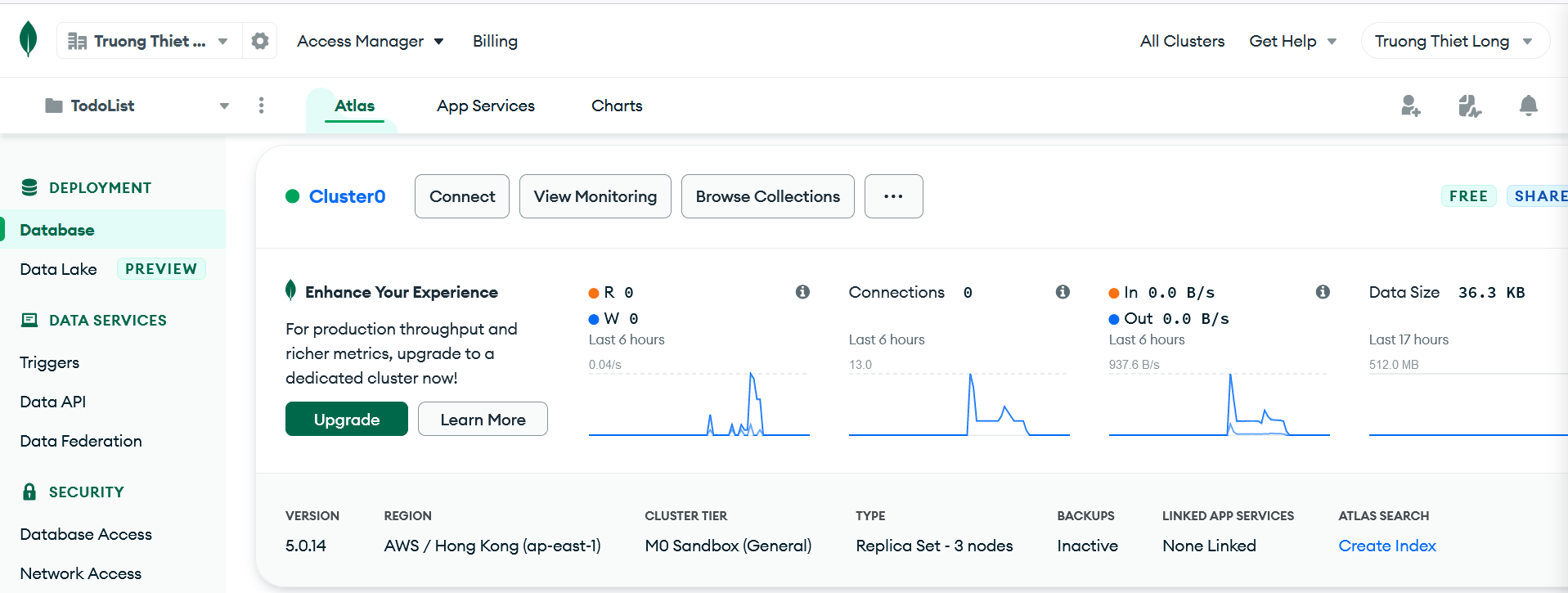
Sau đó nhấp vào Build Database và chọn Create Cluster. Cung cấp tên người dùng và mật khẩu ở đây và nhấp vào Create User.

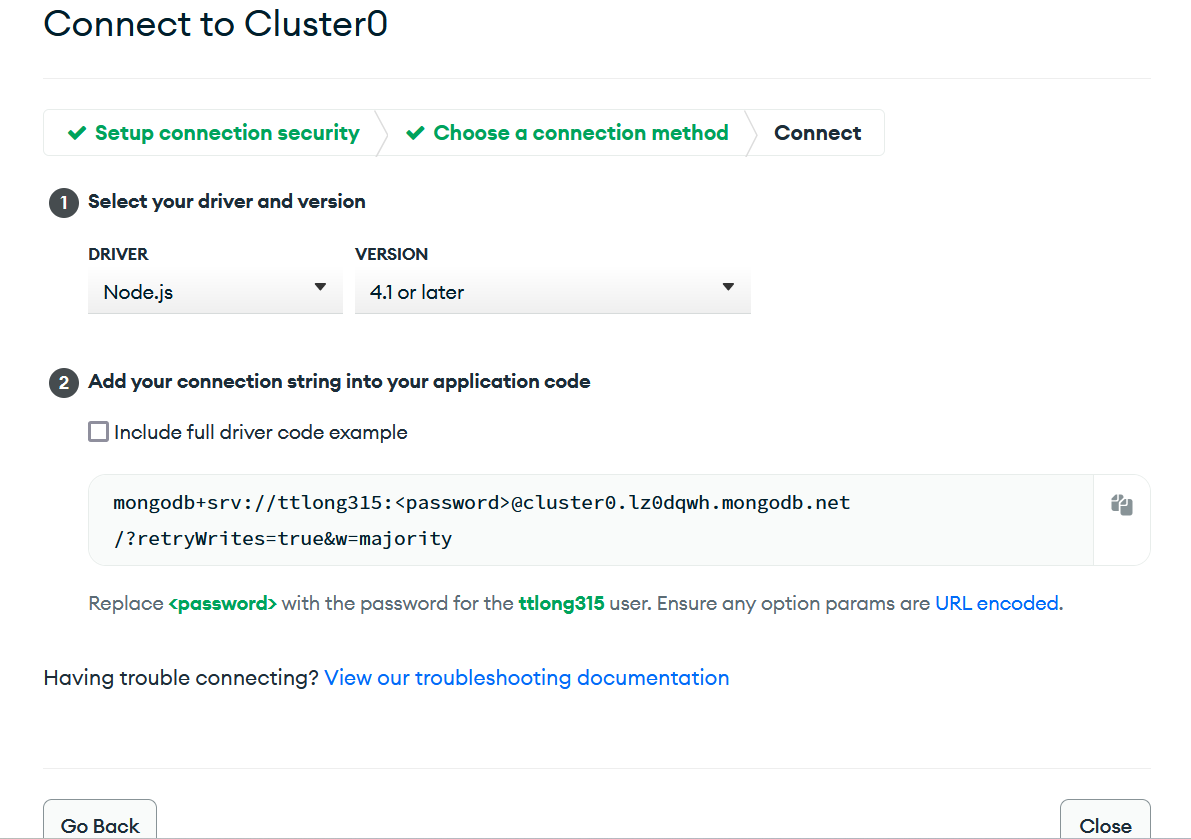


Đưa địa chỉ IP localhost vào IP Address để có thể kết nối với MongoDB Atlas. Sau đó chọn Add Entry và Finish and Close.

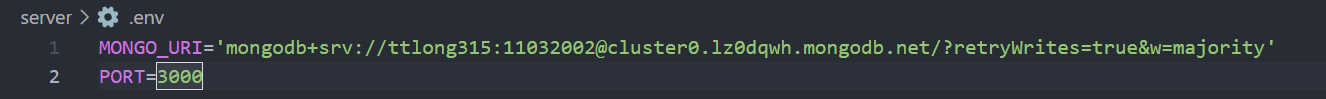


Ở giao diện MongoDB Connecting User, bấm Connect và chọn Connect Your Application sau đó copy Mongo URL:





Trong thư mục /server, tạo file .env và thêm nội dung sau:



*Với 11032002 là mật khẩu người dùng đã tạo trước đó*

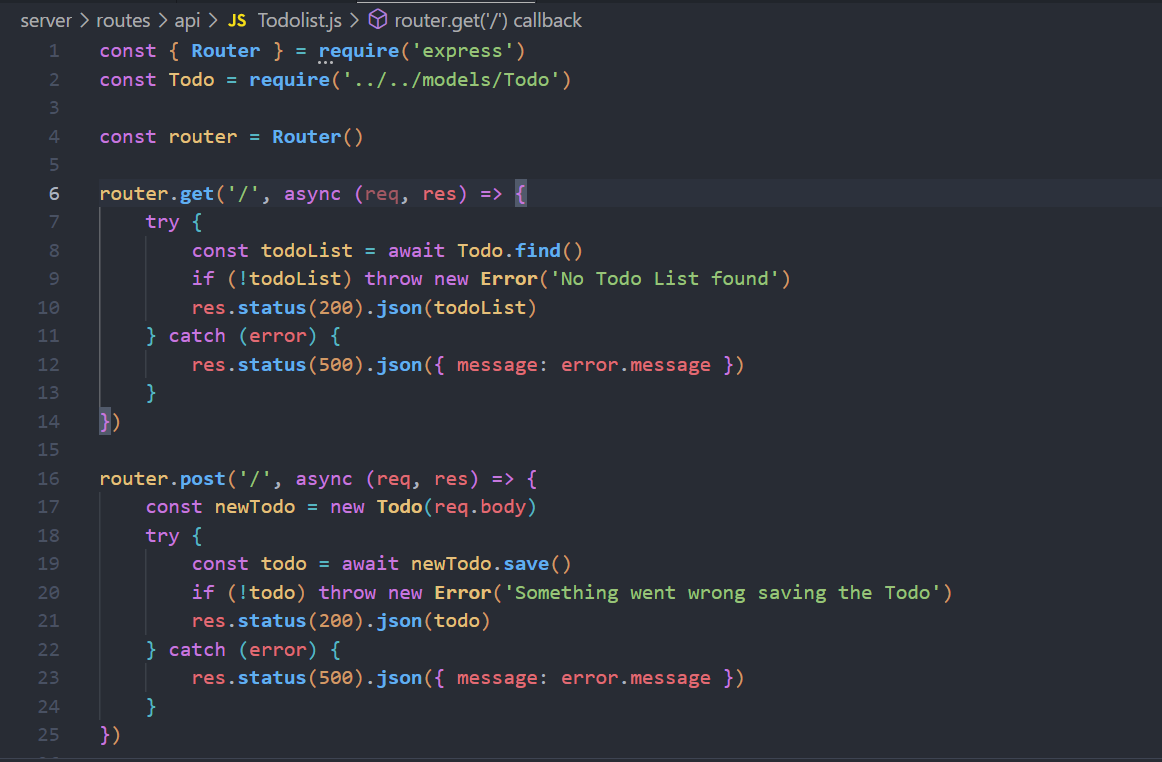
* **Tạo Todo Schema bằng Mongoose**

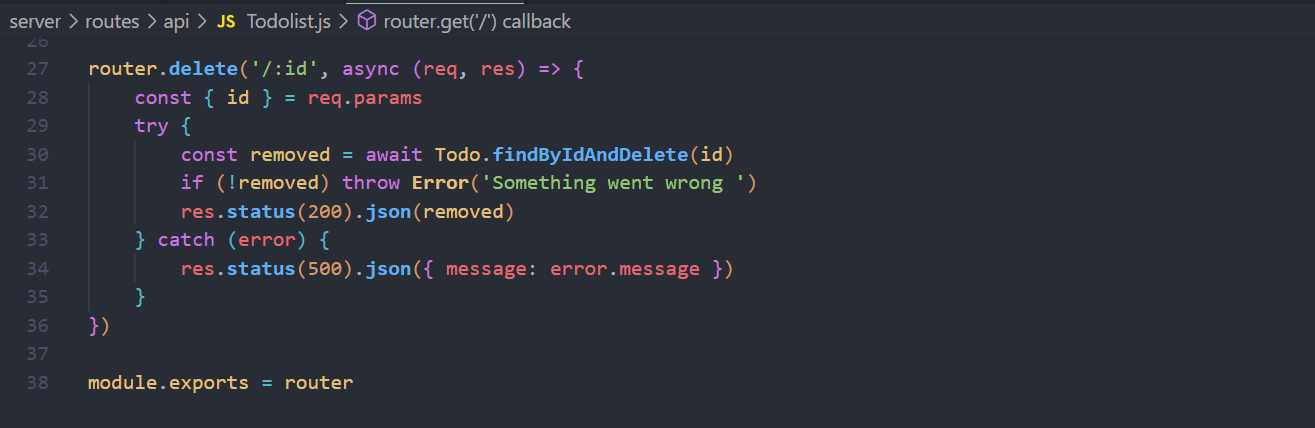
Tạo thư mục ./models trong thư mục gốc và tệp Todo.js, đồng thời thêm vào tệp Todo.js nội dung sau:



* **Tạo API Routes**

Tạo một tệp Todolist.js trong ./routes/api và sử dụng mã sau để tạo các điểm cuối /GET, /POST, /PUT & /DELETE để thực hiện liệt kê, tạo, cập nhật và xóa các tác vụ trong Todo List tương ứng.





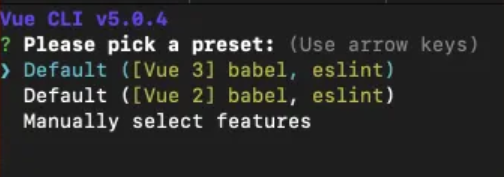
Nhập các routes vào file index.js:



## **Frontend**

* **Tạo Client Vuejs**

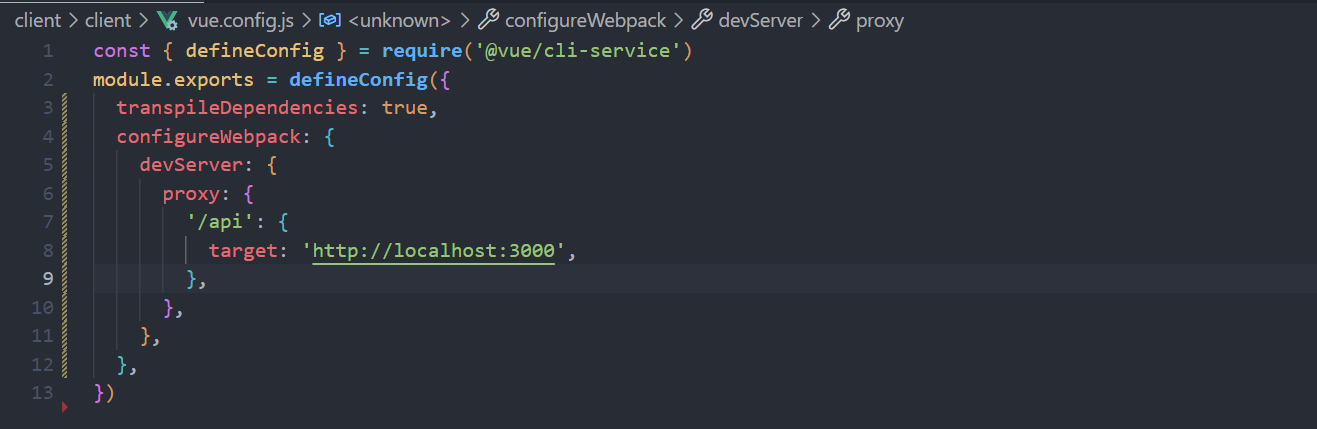
Vào thư mục /client và tạo một ứng dụng mới bằng lệnh: *vue create client.* Chọn Vue 3 và tuỳ chọn mặc định trong khi cài đặt ứng dụng.

****

* **Yêu cầu API proxy từ ứng dụng Vue**

Để gọi API, cần ủy quyền yêu cầu từ giao diện người dùng để URL yêu cầu sẽ là URL của máy chủ.

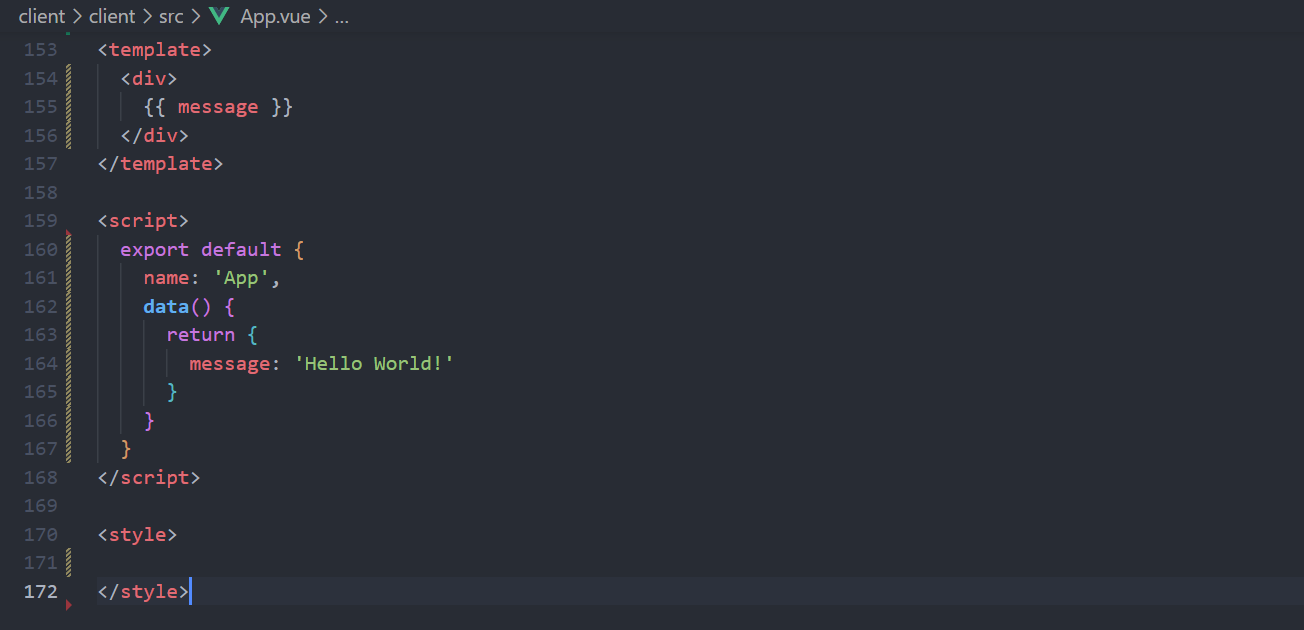
Trong vue.config.js , thực hiện thay đổi sau:

****

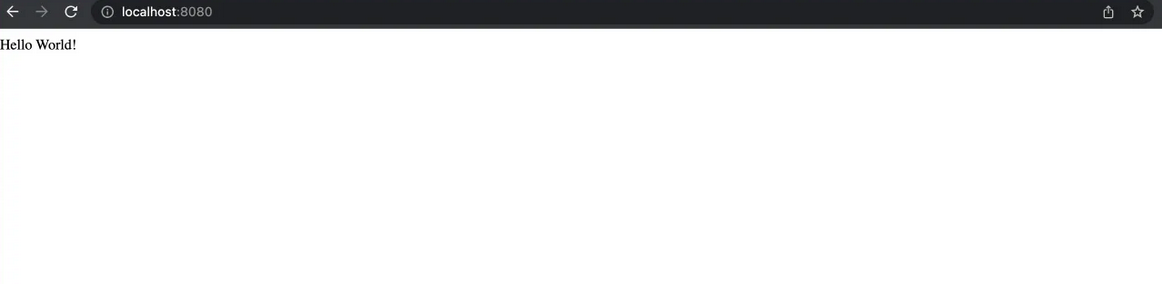
Để thực hiện các yêu cầu API, cài đặt gói axios bằng lệnh: *npm i axios.*

Xóa nội dung trong thư mục src/App.vue và /components. Đối với ứng dụng này, xây dựng đơn giản nên không cần thư mục /components.

Tạo mẫu Hello World đơn giản trên thư mục App.vue để chạy kiểm tra thử:



Vì đã thiết lập chạy đồng thời cả Client và Server, có thể trực tiếp kiểm tra điều này trên <http://localhost:8080>



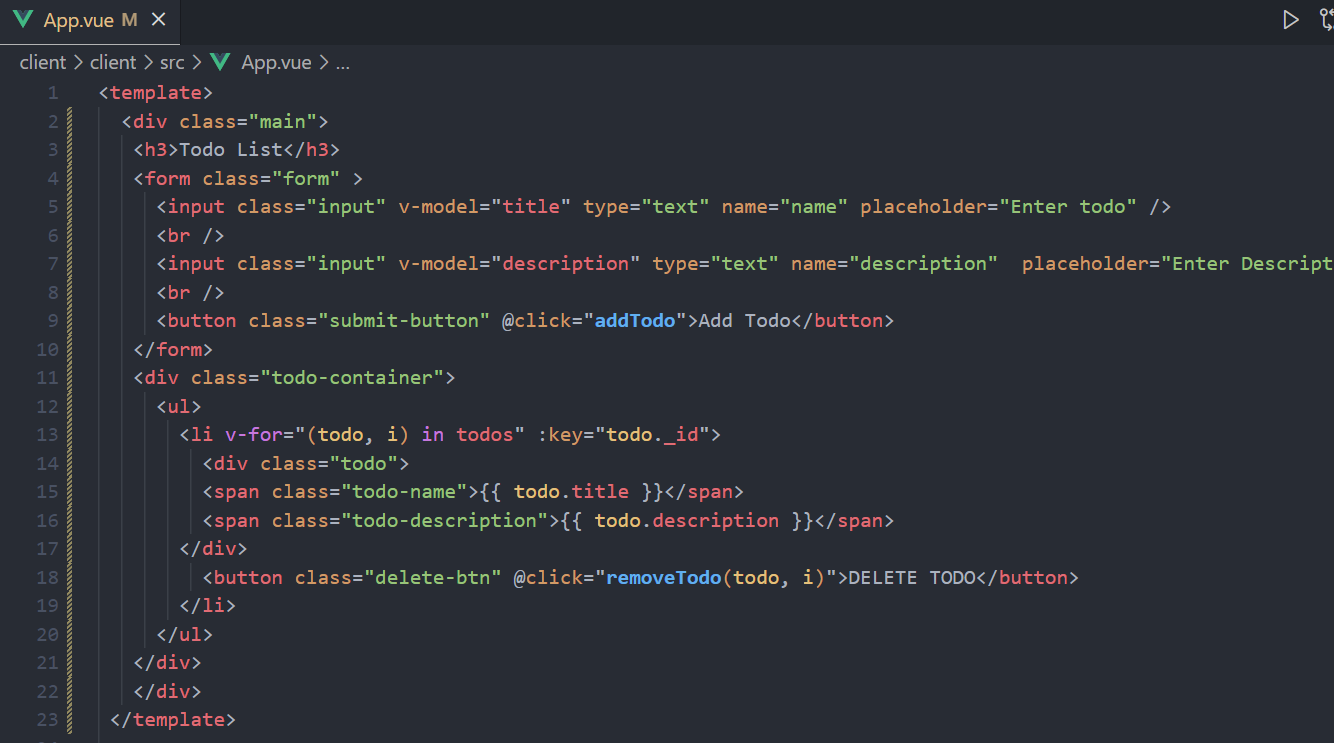
Trong file App.Vue, hiệu chỉnh phần script như sau:

* Import gói Axios
* Khởi tạo việc cần làm, title cho việc cần làm và description cho việc cần làm trên hàm data().
* Trong Methods, tạo các hàm để thêm và xóa việc cần làm.
* Hàm mounted() chạy ngay khi tải components, gọi /GET todos trong chức năng đó.

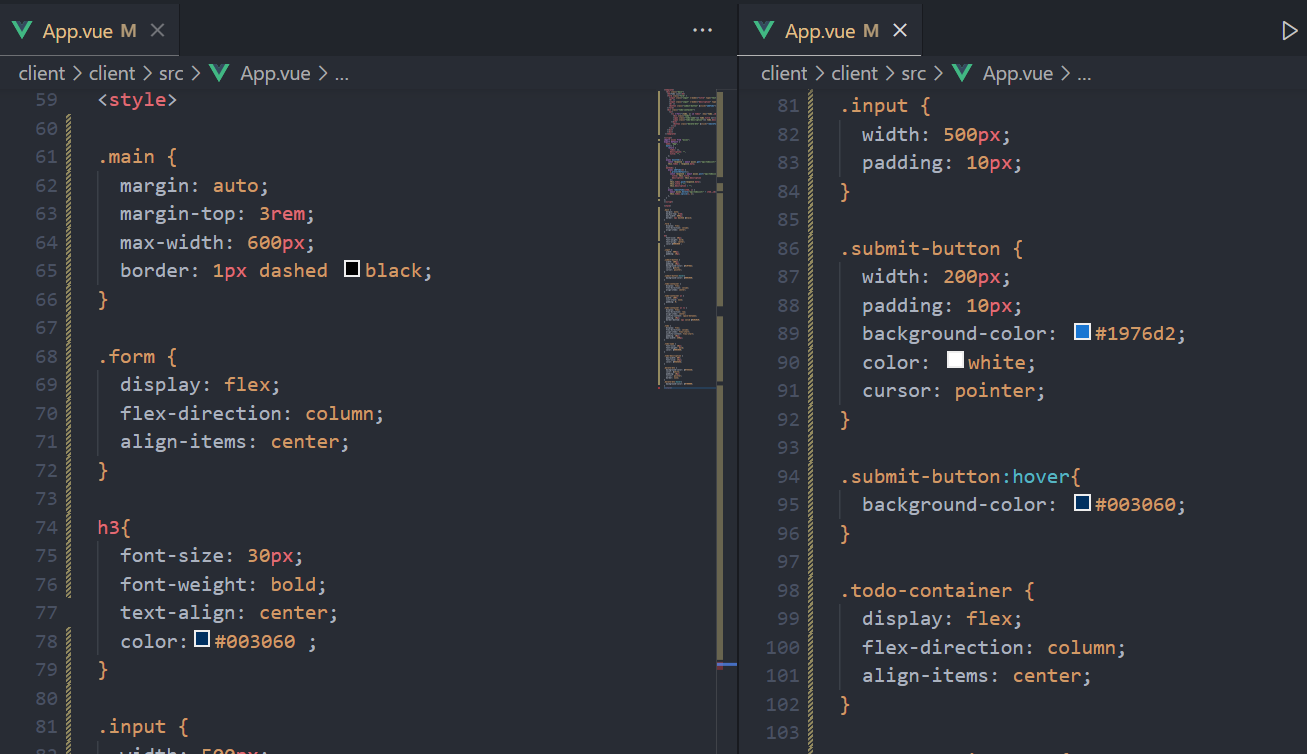


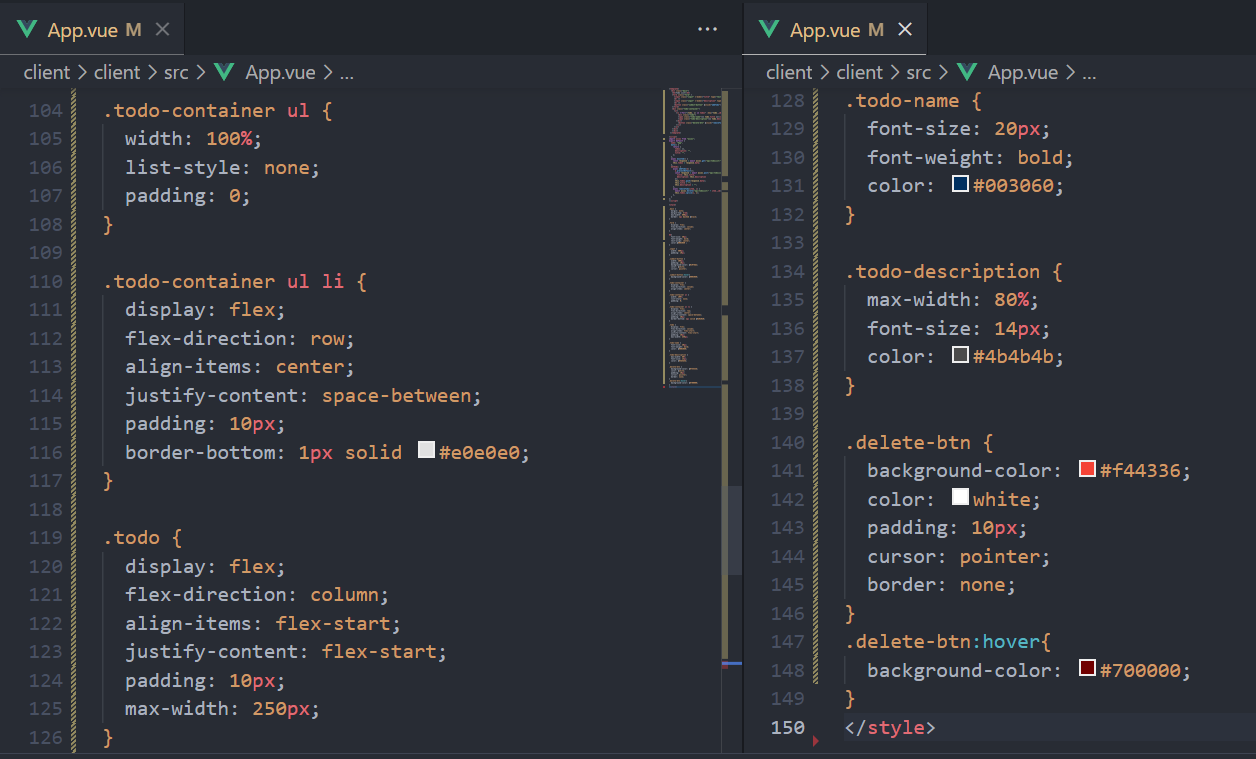


Tạo HTML cho ứng dụng dưới thẻ template trong file App.vue



Tạo CSS cho HTML bằng thẻ style trong App.vue

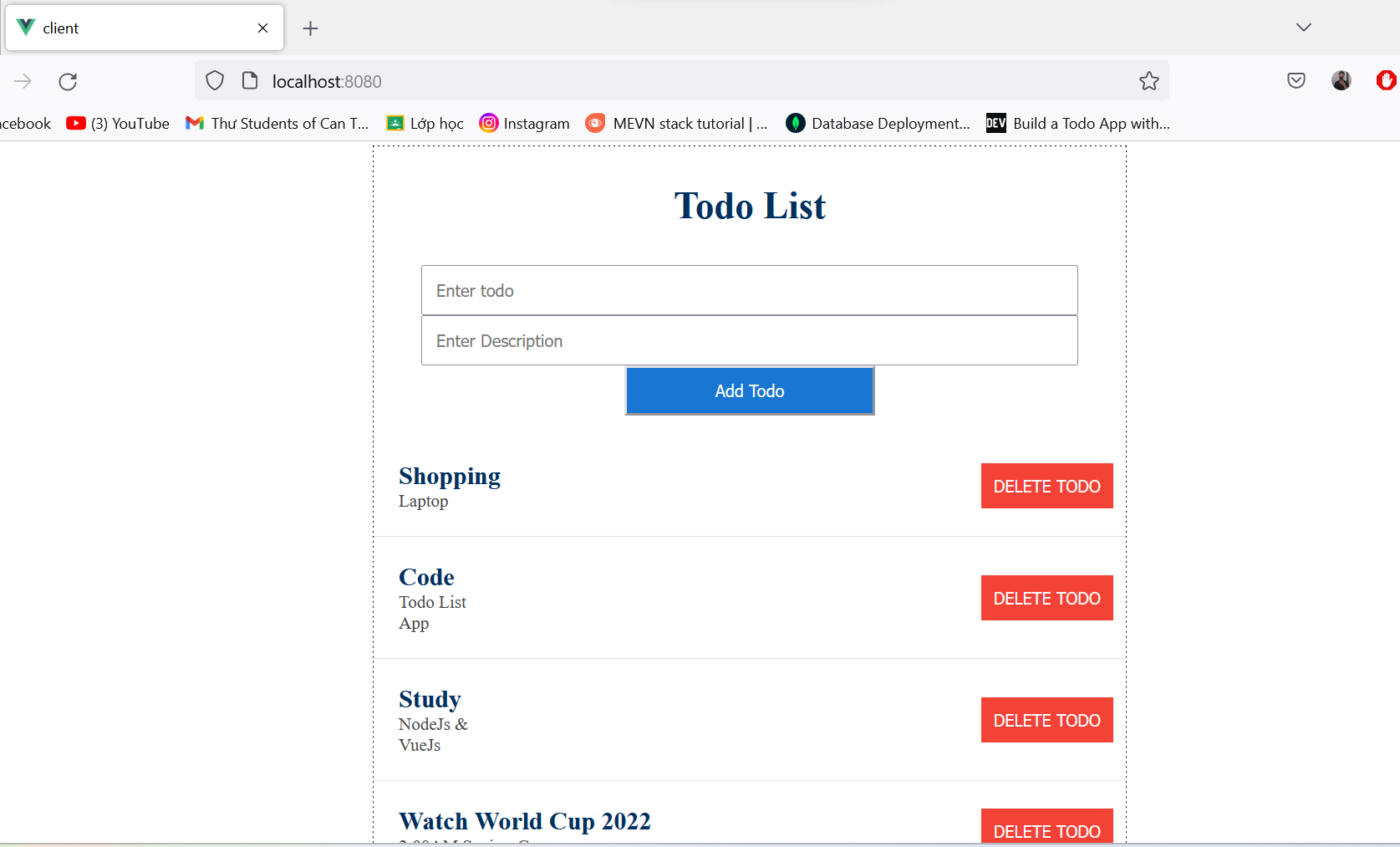




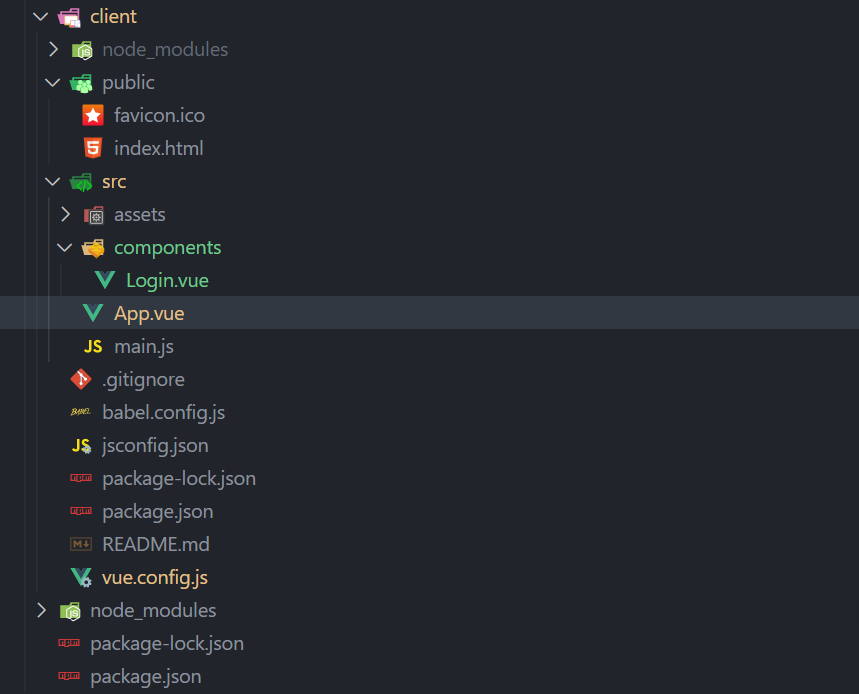
* **Chạy ứng dụng**

Chạy lệnh sau để khởi động cả máy khách và máy chủ: *npm run dev*

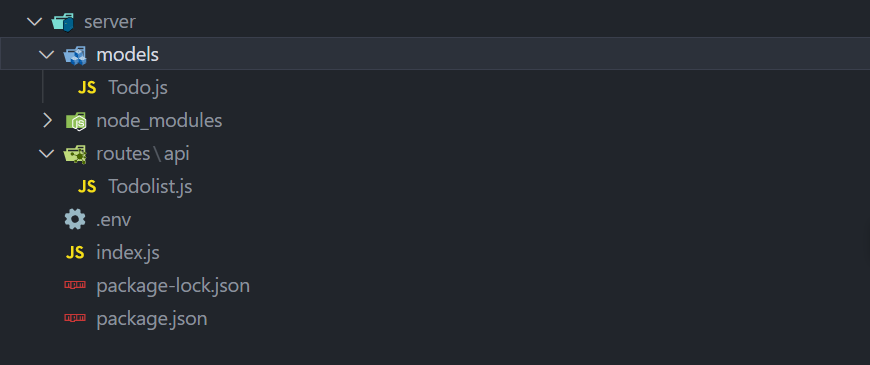
Bây giờ, Client sẽ chạy trên http://localhost:8080 trong khi server chạy trên <http://localhost:3000>

****

* **Cây thư mục ứng dụng**
  + **/Client**

****

* **/Server**

****

**- HẾT -**