

## CÀI ĐẶT MERN PROJECT

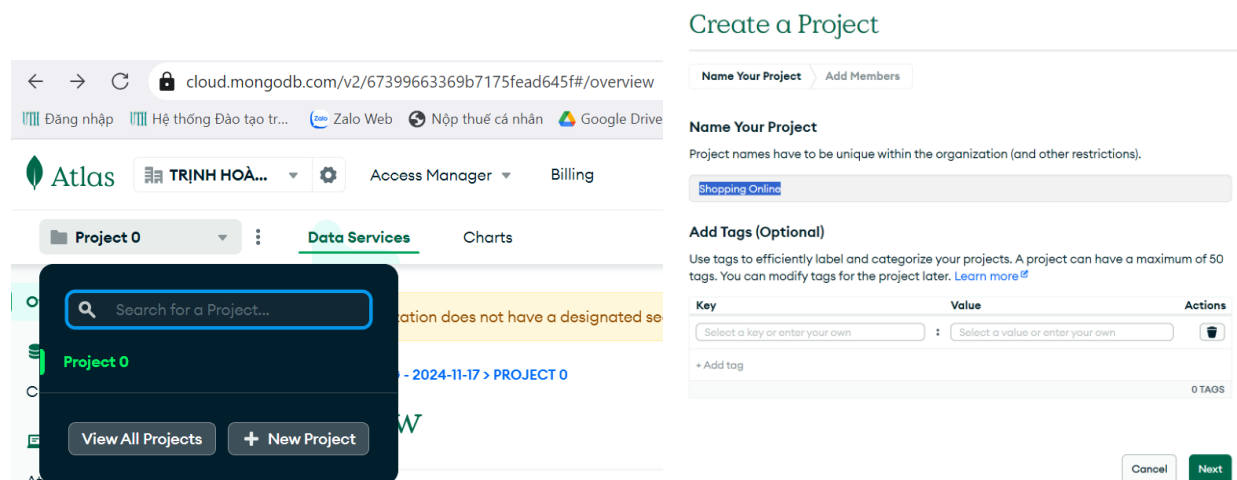
### 1. Cài đặt MongoDB

MongoDB là một nền tảng quản trị database mã nguồn mở NoSQL. Thay vì sử dụng table và row thì MongoDB sử dụng Collections và Documents. Documents được tạo ra từ một cặp key-value, Collection (giống như table trong SQL) chứa các tập hợp Documents.

Truy cập địa chỉ [www.mongodb.com](http://www.mongodb.com) sau đó đăng nhập bằng địa chỉ gmail, tiến hành cài đặt database.

### Tạo một project MongoDB mới

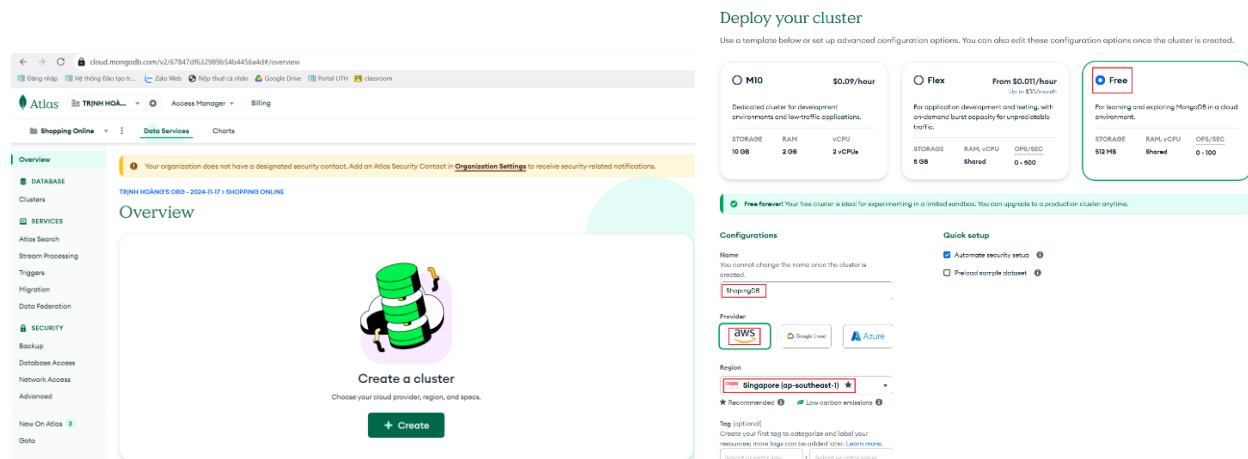
Tại mục Project nhấn mũi tên xổ xuống và click vào nút **New Project**, sau đó đặt tên cho project (*Shopping Online*), nhấn nút **Create Project** để tạo.



### Tạo Cluster

Nhấn nút **Create a cluster** để tiến hành tạo cluster.

Chọn gói miễn phí **Free**, mục Name đặt tên cluster: **ShoppingDB**, mục Provider chọn nhà cung cấp: **AWS**, mục Region: Chọn **Singapore**. Sau đó nhấn **Create Deployment** để tạo



## MERN Project: Shopping Online Tutorials

Tạo tài khoản cho database tại mục Username và Password (Ghi nhớ tài khoản để kết nối database ở bước sau) sau đó cấu hình địa chỉ IP được phép kết nối vào database: **0.0.0.0/0** (cho phép tất cả các địa chỉ IP kết nối database)

Connect to ShoppingDB

1 Set up connection security 2 Choose a connection method 3 Connect

You need to secure your MongoDB Atlas cluster before you can use it. Set which users and IP addresses can access your cluster now. [Read more](#)

1. Add a connection IP address

✓ Your current IP address (115.79.51.236) has been added to enable local connectivity. Only an IP address you add to your Access List will be able to connect to your project's clusters. Add more later in [Network Access](#).

2. Create a database user

This first user will have [atlasAdmin](#) permissions for this project.

We autogenerated a username and password. You can use this or create your own.

ⓘ You'll need your database user's credentials in the next step. Copy the database user password.

Username Password

Create Database User

Close Choose a connection method

2. Where would you like to connect from?

Enable access for any network(s) that need to read and write data to your cluster.

My Local Environment Use this to add network IP addresses to the IP Access List. This can be modified at any time.

Cloud Environment Use this to configure network access between Atlas and your cloud or on-premise environment. Specifically, set up IP Access Lists, Network Peering, and Private Endpoints.

Advanced

Add entries to your IP Access List

Only an IP address you add to your Access List will be able to connect to your project's clusters.

IP Address	Description
0.0.0.0/0	Enter description

Add Entry Add My Current IP Address

Nhấn **Choose a connection method** để cấu hình phương thức kết nối database là **Compass**. Nếu chưa cài đặt công cụ Compass thì chọn mục **I don't have MongoDB Compass installed**, chọn hệ điều hành và nhấn nút **Download Compass** để tải về và cài đặt vào máy tính, Copy chuỗi kết nối để dán vào công cụ Compass sau khi cài đặt xong. Chuỗi kết nối có cấu trúc như sau:

mongodb+srv://<db\_username>:<db\_password>@shoppingdb.9qc5h.mongodb.net/

Connect to ShoppingDB

1 Set up connection security 2 Choose a connection method 3 Connect

⚠ You can't connect yet. Set up your user security permission in the first step.

Connecting with MongoDB Compass

I don't have MongoDB Compass installed I have MongoDB Compass installed

1. Select your operating system and download MongoDB Compass

Windows 64-bit (10+)

Download Compass (1.45.0) or Copy download URL

Compass is an interactive tool for querying, optimizing, and analyzing your MongoDB data.

2. Copy the connection string, then open MongoDB Compass

Use this connection string in your application

Replace <db\_password> with the password for the <db\_username> user. Ensure any options are URL encoded. You can edit your database user password in [Database Access](#).

RESOURCES

Connect with Compass Import and Export Data

Access your Database Users Troubleshoot Connections



Go Back Close Go Back Done

Mở phần mềm Compass, chọn **Add new connection** sau đó dán chuỗi kết nối đã copy ở bước trên vào. Chú ý thay thế **username** và **password** kết nối database đã tạo ở bước trên vào đúng vị trí chuỗi kết nối. Sau đó nhấn **Save & Connet**

# MERN Project: Shopping Online Tutorials

## New Connection

Manage your connection settings

URI  Edit Connection String   
mongodb+srv://[redacted]@shoppingdb.9qc5h.mongodb.net/

Name  
shoppingdb.9qc5h.mongodb.net

Color  
No Color

☐ Favorite this connection  
Favoriting a connection will pin it to the top of your list of connections

Advanced Connection Options

Cancel

How do I find my connection string in Atlas?  
If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the cluster to which you wish to connect.  
[See example](#)

How do I format my connection string?  
[See example](#)

Save

Connect

Save & Connect

Connections Edit View Help

## Compass

My Queries

CONNECTIONS (1)

Search connections

shoppingdb.9qc5h.mongodb.net

admin

local

## Import data

Tại giao diện phần mềm Compass chọn **Create Database** để tạo database tên là **ShoppingDB** và collection tên là **admins**

## Create Database

Database Name

ShoppingDB

Collection Name

admins

☐ Time-Series

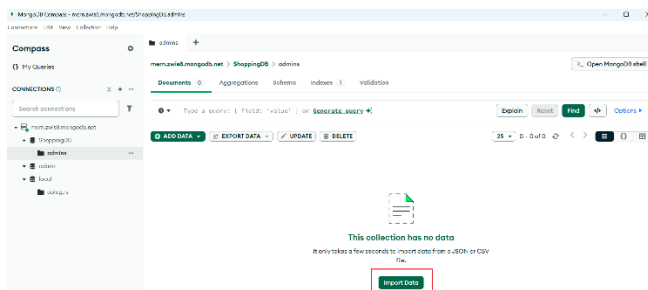
Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

Additional preferences (e.g. Custom collation, Capped, Clustered collections)

Cancel

Create Database

Vào Collection admins, nhấn nút **Import Data** để thêm dữ liệu từ file **admins.json**



## Import

To collection ShoppingDB.admins

Import file: **admins.json**

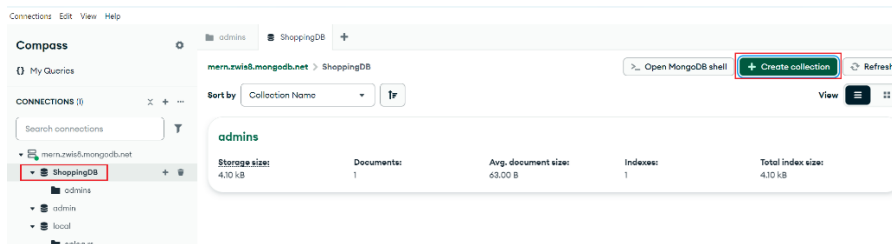
Options

☐ Stop on errors

Cancel

Import

## Tạo collection: categories



## MERN Project: Shopping Online Tutorials

Vào Collection **categories**, nhấn nút **Import Data** để thêm dữ liệu từ file **categories.json**

### Import

To collection ShoppingDB.categories

Import file: categories.json

#### Options

☐ Stop on errors

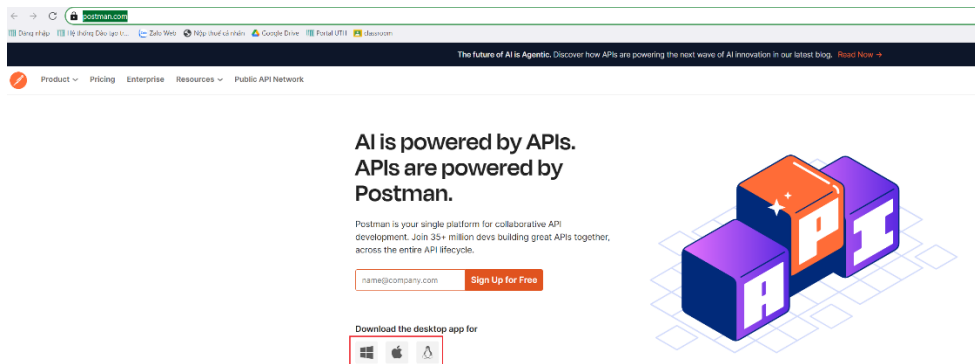
Cancel

Import

Thực hiện tương tự tạo collection: **products**. Vào Collection **products**, nhấn nút **Import Data** để thêm dữ liệu từ file **products.json**

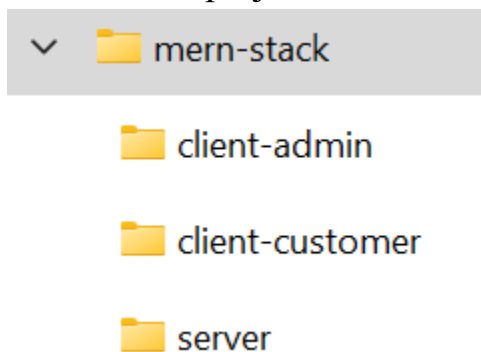
## 2. Cài đặt công cụ Postman

Truy cập địa chỉ <https://www.postman.com/> chọn phiên bản cài đặt đúng theo hệ điều hành (windows) tiến hành tải và cài đặt Postman vào máy tính.



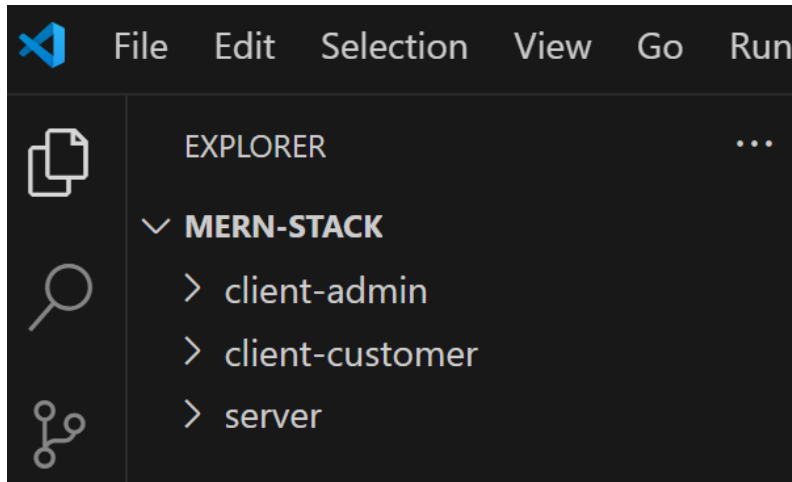
## 3. Tạo thư mục project

Tạo thư mục project trên ổ đĩa D:\ theo cấu trúc sau:



Trong đó: Thư mục **server**: chứa các tài nguyên của phần backend viết bằng NodeJS (ExpressJS); Thư mục **client-admin**: chứa tài nguyên của phần frontend nếu tài khoản là admin; Thư mục **customer-admin**: chứa tài nguyên của phần frontend nếu tài khoản là khách hàng.

Mở VSCode và Add Folder project vừa tạo vào



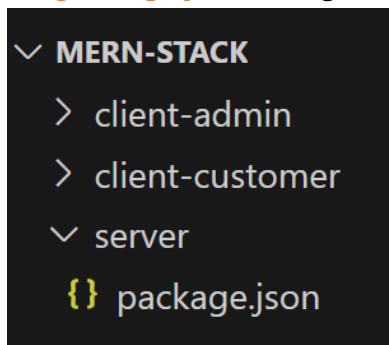
## API Server

Mở Terminal trên VSCode (phím tắt Ctrl + Shift + `) gõ lệnh sau:

```
\mern-stack> cd server
\nern-stack\server> npm init -y
```

Lệnh `cd server` để truy cập vào thư mục server

Lệnh `npm init -y` để khởi tạo project NodeJS sau khi khởi tạo sẽ tạo ra một tập tin `package.json` trong thư mục server của project



Cài đặt nodemon: Mở terminal gõ lệnh: `npm install -global nodemon`

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS F:\THINH GIANG\Gia Dinh\mern-stack\server> npm i -g nodemon
changed 29 packages in 2s
4 packages are looking for funding
run `npm fund` for details
```

Mở tập tin `/server/package.json` sửa dòng code số 6 như sau:

## MERN Project: Shopping Online Tutorials

```
{ } package.json X
server > { } package.json > { } scripts > start
1  {
2    "name": "server",
3    "version": "1.0.0",
4    "main": "index.js",
5    "scripts": {
6      "start": "nodemon index.js"
7    },
8    "keywords": [],
9    "author": "",
10   "license": "ISC",
11   "description": "",
12   "dependencies": {
13     "express": "^4.21.2"
14   }
15 }
```

Cài đặt thư viện Express: Mở terminal và gõ lệnh: `npm install express`

```
PS F:\THINH GIANG\Gia Dinh\mern-stack\server> npm install express
added 69 packages, and audited 70 packages in 3s
14 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
```

Trong thư mục **server**, tạo mới tập tin đặt tên là **index.js** sau đó viết code như sau:

```
JS index.js X
server > JS index.js > app.listen() callback
1  const express = require('express');
2  const app = express();
3  const PORT = process.env.PORT || 3000;
4
5  //Sử dụng middleware
6  app.use(express.urlencoded({extended: false}));
7
8  //Routes API
9  app.get('/', (req, res)=>{
10    res.json({message: 'Hello from Server!'})
11  })
12
13 //Tạo web server
14 app.listen(PORT, ()=>{
15   console.log(`Server is running at port ${PORT}`);
16 });
```

## MERN Project: Shopping Online Tutorials

Cài đặt thư viện cors cho phép API server chia sẻ tài nguyên với máy chủ khác nguồn gốc (origin): Mở terminal gõ lệnh **npm i cors**

Bổ sung lệnh để sử dụng thư viện cors trong tệp tin **index.js** như sau:

```
JS index.js  X
server > JS index.js > ...
1  const express = require('express');
2  const app = express();
3  const cors = require('cors');
4  const PORT = process.env.PORT || 3000;
5
6  //Sử dụng middleware
7  app.use(express.urlencoded({extended: false}));
8  app.use(cors());
9
10 //Routes API
11 app.get('/', (req, res)=>{
12   res.json({message: 'Hello from Server!'})
13 })
```

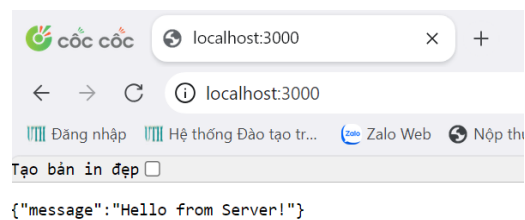
Mở terminal gõ lệnh **npm start** để start server

```
PS F:\THINH GIANG\Gia Dinh\mern-stack\server> npm start

> server@1.0.0 start
> nodemon index.js

[nodemon] 3.1.9
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
Server is running at port 3000
```

Mở trình duyệt google chrome và gõ địa chỉ localhost:3000 (hoặc 127.0.0.1:3000) để kiểm tra



The screenshot shows a Google Chrome browser window with the address bar set to 'localhost:3000'. The page content displays a JSON object: `{"message": "Hello from Server!"}`. The browser's developer tools are not open, and the page appears to be a simple text response.

Mở phần mềm Postman và thử gửi request đến server để test

### Client Admin Page

Cài đặt vite: Mở terminal gõ lệnh: **npm create vite@latest**

## MERN Project: Shopping Online Tutorials

```
CONSOLE  TERMINAL  PORTS
\nmern-stack> npm create vite@latest
```

+ project name: **client-admin**

+ framework: React

+ variant: Javascripts

Tiếp tục gõ lệnh **cd client-admin** để truy cập vào thư mục client-admin sau đó gõ lệnh **npm install** để cài đặt project React

```
\mern-stack> cd client-admin
\nmern-stack\client-admin> npm install
```

Bổ sung vào cuối tập tin **/client-admin/package.json** 2 dòng code sau:

```
{ package.json X
client-admin > {} package.json > homepage
16   "devDependencies": {
23     "eslint-plugin-react-nooks": "^5.0.0",
24     "eslint-plugin-react-refresh": "^0.4.16",
25     "globals": "^15.14.0",
26     "vite": "^6.0.5"
27   },
28   "homepage": "/admin",
29   "proxy": "http://localhost:3000"
30 }
```

Cài đặt **axios** (công cụ lấy dữ liệu từ API Server): Tại terminal gõ lệnh

**npm i --save axios**

```
MINAL  PORTS
\nclient-admin> npm i --save axios
```

Sửa nội dung code tập tin **/client-admin/App.js** như sau (xóa hết code cũ và nhập code mới như hình):



## MERN Project: Shopping Online Tutorials

```
App.jsx x
client-admin > src > App.jsx > ...
1  import axios from 'axios';
2  import React, {Component} from 'react';
3  //Cấu hình axios
4  const api = axios.create({
5    |   baseURL: `http://localhost:3000`
6  })
7  class App extends Component{
8    constructor(props){
9      |   super(props);
10     |   this.state = {
11     |     |   message: "Loading..."
12     |   }
13   }
14   //Render trang Admin
15   render(){
16     |   return(
17     |     |   <div>
18     |     |     |   <h2>Admin page</h2>
19     |     |     |   <p>{this.state.message}</p>
20     |     |   </div>
21     |   )
22   }
23   //Lấy dữ liệu từ Backend API
24   componentDidMount(){
25     |   api.get('/').then((res)=>{
26     |     |   const result = res.data;
27     |     |   console.log(result);
28     |     |   this.setState({message: result.message});
29     |   })
30   }
31 }
32 export default App;
```

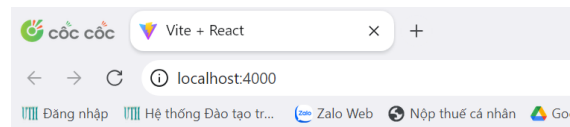
Mở terminal gõ lệnh **npm run dev** để start, sau đó mở trình duyệt gõ địa chỉ <http://localhost:4000> để xem kết quả hiển thị trang admin

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
npm help
PS F:\THINH GIANG\Gia Dinh\mern-stack\client-admin> npm run dev

> client-admin@0.0.0 dev
> vite --port 4000

VITE v6.0.7 ready in 153 ms

→ Local:   http://localhost:4000/
→ Network: use --host to expose
→ press h + enter to show help
4:16:41 PM [vite] (client) hmr update /src/App.jsx
```



### Admin page

Hello from Server!

## Client Customer Page

Tạo mới terminal gõ lệnh tại thư mục project: **npm create vite@latest**

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS F:\THINH GIANG\Gia Dinh\mern-stack> npm create vite@latest
```

+ project name: **client-customer**

+ framework: React

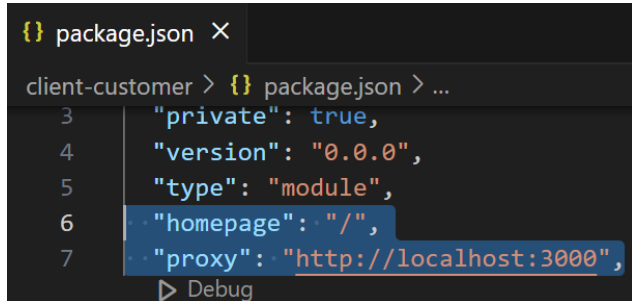
+ variant: Javascripts

## MERN Project: Shopping Online Tutorials

Tiếp tục gõ lệnh `cd client-customer` để truy cập vào thư mục client- customer sau đó gõ lệnh `npm install` để cài đặt project React

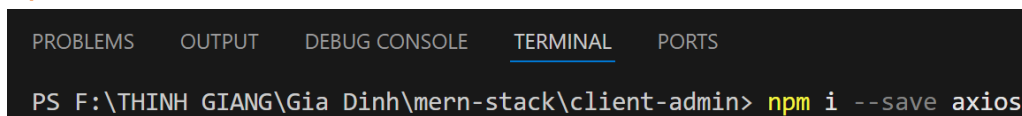
```
PS F:\THINH GIANG\Gia Dinh\mern-stack> cd client-customer
PS F:\THINH GIANG\Gia Dinh\mern-stack\client-customer> npm i
```

Bổ sung vào cuối tập tin `/client-customer/package.json` 2 dòng code sau:



```
{"private": true,
  "version": "0.0.0",
  "type": "module",
  "homepage": "/",
  "proxy": "http://localhost:3000",
}
```

Cài đặt **axios** (công cụ lấy dữ liệu từ API Server): Tại terminal gõ lệnh `npm i --save axios`



```
PS F:\THINH GIANG\Gia Dinh\mern-stack\client-admin> npm i --save axios
```

Sửa nội dung code tập tin `/client-customer/App.js` như sau:

## MERN Project: Shopping Online Tutorials

```
App.jsx X
client-customer > src > App.jsx > [default]
1  import axios from 'axios';
2  import React, {Component} from 'react';
3  //Cấu hình axios
4  const api = axios.create({
5    |   baseURL: `http://localhost:3000`
6  | })
7  class App extends Component{
8    |   constructor(props){
9    |     |   super(props);
10   |     |   this.state = {
11   |     |     |   message: "Loading..."
12   |     |   }
13   |   }
14   |   //Render trang Customer
15   |   render(){
16   |     |   return(
17   |     |     |   <div>
18   |     |     |     |   <h2>Customer page</h2>
19   |     |     |     |   <p>{this.state.message}</p>
20   |     |     |   </div>
21   |     |   )
22   |   }
23   |   //Lấy dữ liệu từ Backend API
24   |   componentDidMount(){
25   |     |   api.get('/').then((res)=>{
26   |     |     |   const result = res.data;
27   |     |     |   console.log(result);
28   |     |     |   this.setState({message: result.message});
29   |     |   })
30   |   }
31  }
32  export default App;
```

Mở terminal gõ lệnh **npm run dev** để start, sau đó mở trình duyệt gõ địa chỉ <http://localhost:5000> để xem kết quả hiển thị trang customer

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
found 0 vulnerabilities
PS F:\THINH GIANG\Gia Dinh\mern-stack\client-customer> npm run dev

> client-customer@0.0.0 dev
> vite --port 5000

VITE v6.0.7 ready in 152 ms
→ Local: http://localhost:5000/
→ Network: use --host to expose
→ press h + enter to show help
4:30:55 PM [vite] (client) hmr update /src/App.jsx
```

