First Solo FullStack Project

This is my first fullstack project and my documents of what I did so try so that I can try and remember the skills that I have picked up. This project will be a market place where users are able to post their items for sale.

**Setup**

1. First thing I did was install node but doing (brew install node).
2. Once that was one, I created two folders called frontend and backend
3. (npm install -y) to initialise the json files
4. (npn install mongoose dotenv)
5. (npm install -D nodemon)

Nodemon is really useful. Once I start running the backend, everytime I save and make changes, it will automatically restart the backend allowing me to see the changes without closing and rerunning the command.

**Initial Changes to package.Json**

1. Changed “type”: to module as it allows better importing standards.
2. Changed scripts to “dev”: nodemon backend/index.js. This allows me to run (npm run dev)

**Creating Database**

1. Make on account on mondogb
2. After save the password for later

In order to send information to the data base, I made four API functions using POST, GET, DELETE and PUT.

Since MongoDB is a NoSQL database, it uses collections and not tables. These are what the four functions will do:

POST – be able to create new items to the database

GET – be able to get all the items in the databse

DELETE – delete items in the database using the key created by MondoDB when posting

PUT – this will update the item in the database. For example, if I list an item where I want to change the name or price I can use these functions.

Initially, I put all the functions in the index.js but the code got really big and was not modular. In order to make it modular, I added 2 more folders inside the working directory.

The first folder is called route and the second folder is called controller.

The controller is where all of the code for the function will be stored. This will import from model.products where it holds the information about what fields the users needs to add to the database. This includes name, price and a picture which are all required.

The routes folder contain the requests that is made by the user and then calls the controller for the function needed.

This way, I can have a clean folder which makes it modular and easy to read.

Front End

To set up the front end, I located to the front end folder and run the command:

npm create vite@latest .

Then I selected react with javascript.