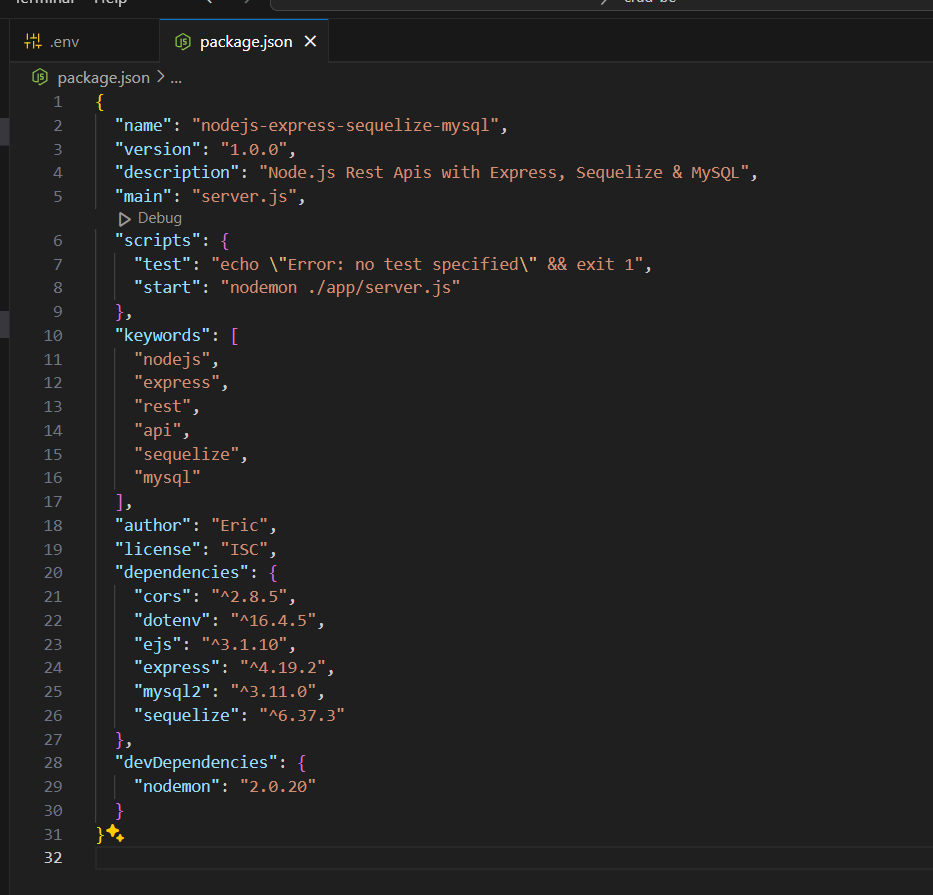
View –> route-> controller-> service -> view

* Cài đặt môi trường node js
* 
* Install express
* Install template EJS để code giao diện động
* Install dotnet ( .net) để quản lý biến cục bộ
* Nodemon để tự cập nhật lại khi file thay đổi khỏi cần chạy lại server
* Sử dụng dbeaver
* Node js và mysql , myslq2
* const mysql = require("mysql2");
* const connection = mysql.createConnection({
* host:'localhost',
* port:3307,
* user:'root',
* password:'123456',
* database:'hoidanit',
* });
* Tối ưu hóa data base

Connection pools help reduce the time spent connecting to the MySQL server by reusing a previous connection, leaving them open instead of closing when you are done with them.

import mysql from 'mysql2/promise';  
  
***// Create the connection pool. The pool-specific settings are the defaults*  
const pool = mysql.createPool({  
 host: 'localhost',  
 user: 'root',  
 database: 'test',  
 waitForConnections: true,  
 connectionLimit: 10,  
 maxIdle: 10, *// max idle connections, the default value is the same as `connectionLimit`*  
 idleTimeout: 60000, *// idle connections timeout, in milliseconds, the default value 60000*  
 queueLimit: 0,  
 enableKeepAlive: true,  
 keepAliveInitialDelay: 0,  
});**  
  
 **maxIdle: 10**: Số lượng kết nối không hoạt động tối đa mà pool có thể giữ lại. Điều này giúp quản lý tài nguyên hiệu quả hơn.

 **idleTimeout: 60000**: Nếu một kết nối không hoạt động trong thời gian này (60 giây), nó sẽ bị đóng.

 **enableKeepAlive: true**: Bật chế độ KeepAlive để duy trì kết nối mở trong thời gian dài, giúp tránh tình trạng kết nối bị đóng bởi firewall hoặc các thiết lập mạng khác.

 **keepAliveInitialDelay: 0**: Thời gian chờ trước khi KeepAlive bắt đầu hoạt động.

* Lấy dữ liệu từ ô input

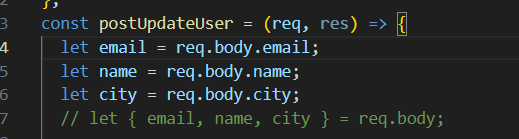
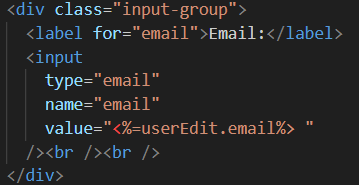
<https://stackoverflow.com/questions/9304888/how-to-get-data-passed-from-a-form-in-express-node-js>

req.body để lấy data từ form

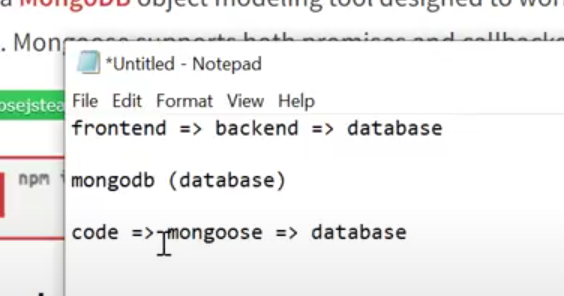
A screenshot of a computer program

Description automatically generated

Lấy dựa theo trường name ở trong ô input



* **Sequelize là một ORM (Object-Relational Mapping) dành cho Node.js, giúp tương tác với cơ sở dữ liệu dễ dàng hơn. Nó hỗ trợ nhiều loại cơ sở dữ liệu như MySQL, PostgreSQL, SQLite, và MariaDB. Với Sequelize, bạn có thể:**
* Mongoose là thư viện node js



1. RESTful api

Method :

* Get : lấy data
* Post : tạo data
* Put : tương tự post yêu cầu server sửa đổi data nhưng khác với thằng post ở chỗ nó chỉ cập nhật data, còn post nó tạo mới data
* Delete : xóa
* So sánh put và patch
* \*tham khảo : https://stackoverflow.com/questions/24241893/should-i-use-patch-or-put-in-my-rest-api
* https://stackoverflow.com/questions/28459418/use-of-put-vs-patch-methods-in-rest-api-reallife-scenarios
* https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design#define-apioperations-in-terms-of-http-methods
* - update 1 fields, hay update toàn bộ ?
* **#83. Setup lưu trữ file với Node.js**

- Với Node.js, các thư viện nổi tiếng thường dùng để upload file:

+ **multer** : https://www.npmjs.com/package/multer

+ **formidable**: https://www.npmjs.com/package/formidable

+ **busboy**: https://www.npmjs.com/package/busboy

+ **express-fileupload**: https://www.npmjs.com/package/express-fileupload

bonus multer: <https://stackoverflow.com/questions/46622473/expressjs-how-to-requiremulter-in-another-file>

1. Package them trường deleted: true/false

* npm install mongoose-delete : để khi xóa dữ liệu chỉ có chuyển thành true/ fales để khách hàng lỡ xóa nhầm thì còn có thể khôi phục được
* truyền động String URL: có thể sử dụng thư viện
* phân trang: khi lấy ví dụ 100000 user thì không thể hiển thị 1 lần hết được mình chia ra một trang 10 kết quả trả về => tối ưu hiệu năng

1. xài package JOI để validate dữ liệu