Ban đầu khởi tạo packet.json bằng npm init

Trong file .js, require(‘mongoose’)

Connect :

mongoose.connect("mongodb://localhost/my\_database", {

useNewUrlParser: true,

useUnifiedTopology: true,

useFindAndModify: false,

useCreateIndex: true,

});

Tạo biến Schema = mongoose.Schema({kiểu bảng liệu}, { collection: "…" })

Tạo biến model = mongoose.model(‘tên bảng’, biến Schema}

Từ biến model có thể sử dụng ra các phương thức có sẵn.

.create({},{}): để tạo 1 hay nhiều bảng dữ liệu. có thể tương tác trong studio 3T for mongoDB, chọn đến file cần copy value, edite document, copy value. Ngoài file chính, ấn chuột phải, chọn document 🡪 insert document 🡪 paste value(phần mềm sẽ tự động thêm \_id: và version)

.find(): trả về toàn bộ những document phù hợp nên kết quả trả về là 1 array chữa nhiều object bên trong.  
Nếu find() để rỗng sẽ show toàn bộ data, nếu trong find() có dữ liệu thì sẽ tìm ra document có dữ liệu tương ứng.  
find({'thongTin.old': {$gte : 23}}) :

gt > gte>= lt< lte<=

{$in[5,10]} bằng 10 và bằng 5 {$nin[5,10]} khác 10 và khác 5;

Các cách sử dụng Find() trên mongodb: <https://docs.mongodb.com/manual/tutorial/query-embedded-documents/>

Trong find() trả về, áp dụng .sort() để sắp xếp kiểu dữ liệu trả về ví dụ: sort({diemDanh: 1}): 1 theo chiều tăng dần, -1 theo chiều giảm dần.

.findOne(): trả về document phù hợp đầu tiên, nên kết quả trả về sẽ là 1 object

.deleteOne({}): xóa đi document mà tìm được đầu tiên thoả mãn điều kiện

.deleteMany({}): xóa đi nhiều phần tử thoả mãn điều kiện

.sort({}): sắp xếp: 1 là giảm dần, -1 tăng dần

.limit(): truyền vào số tương ứng với số lượng document muốn lấy ra.

.updateOne(): sử dụng update thông tin của document, truyền vào 2 object, object 1 là document muốn update, object 2 là cặp giá trị trong document đấy mà muốn update

.updateMany({},{}): nếu trường tìm kiếm để rỗng, sẽ tìm đến tất cả và update toàn bộ. Truyền vào trường tìm kiếm thì sẽ