

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**



**BÁO CÁO PHÁT TRIỂN ỨNG DỤNG WEB
MÃ HỌC PHẦN: CT449
NHÓM: 01**

Giảng viên hướng dẫn

Ths. Nguyễn Minh Trung

Sinh viên thực hiện

Họ tên: Trần Tấn Đạt

MSSV: B2105571

Học kỳ 2, 2024 - 2025

MỤC LỤC

ỨNG DỤNG CONTACTBOOK – BACKEND – PHẦN 1	1
Bước 0: Cài đặt node và git	1
Bước 1: Tạo ứng dụng Node.....	1
Bước 2: Quản lý mã nguồn dự án với git và GitHub.....	1
Bước 3: Cấu hình Visual Studio Code, ESLint và Prettier	3
Bước 4: Cài đặt Express	5
Bước 5: Định nghĩa Controller và các route	8
Bước 6: Cài đặt xử lý lỗi.....	12
Đường dẫn đến repo github dự án Backend 1:	14
ỨNG DỤNG CONTACTBOOK – BACKEND – PHẦN 2	15
Bước 0: Cài đặt MongoDB	15
Bước 1: Cài đặt thư viện mongodb, định nghĩa hàm trợ giúp kết nối với lớp dịch vụ truy xuất cơ sở dữ liệu (CSDL)	15
Bước 2: Cài đặt các handler.....	17
Đường dẫn đến repo github dự án Backend 2:	29

ỨNG DỤNG CONTACTBOOK – BACKEND – PHẦN 1

Bước 0: Cài đặt node và git

```
PS E:\CT449_PhatchienUngDungWeb\ThucHanh\Backend\Backend-Part1> node -v
v22.14.0
PS E:\CT449_PhatchienUngDungWeb\ThucHanh\Backend\Backend-Part1> git --version
git version 2.46.0.windows.1
PS E:\CT449_PhatchienUngDungWeb\ThucHanh\Backend\Backend-Part1> |
```

Bước 1: Tạo ứng dụng Node

- Khởi tạo ứng dụng node

```
package name: (contactbook-backend)
version: (1.0.0)
description:
entry point: (index.js) server.js
test command:
git repository:
keywords:
author: Dat Tran
license: (ISC)
About to write to E:\CT449_PhatchienUngDungWeb\ThucHanh\Backend\Backend-Part1\contactbook-backend\package.json:
{
  "name": "contactbook-backend",
  "version": "1.0.0",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Dat Tran",
  "license": "ISC",
  "description": ""
}
```

- Cài đặt các package cần thiết:

```
PS E:\CT449_PhatchienUngDungWeb\ThucHanh\Backend\Backend-Part1\contactbook-backend> npm install express cors
added 71 packages, and audited 72 packages in 3s

14 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS E:\CT449_PhatchienUngDungWeb\ThucHanh\Backend\Backend-Part1\contactbook-backend> |
```

Bước 2: Quản lý mã nguồn dự án với git và GitHub

- Tạo tập tin .gitignore:

```
PS E:\CT449_PhatchienUngDungWeb\ThucHanh\Backend\Backend-Part1\contactbook-backend> npx gitignore node
Created .gitignore file for flag type node.
PS E:\CT449_PhatchienUngDungWeb\ThucHanh\Backend\Backend-Part1\contactbook-backend> |
```

- Trong thư mục gốc dự án, chạy lệnh git init và git status:

```
PS E:\CT449_PhátTriểnỨngDungWeb\ThucHanh\Backend\Backend-Part1\contactbook-backend>
git init
Reinitialized existing Git repository in E:/CT449_PhátTriểnỨngDungWeb/ThucHanh/Backend/Backend-Part1/contactbook-backend/.git/
PS E:\CT449_PhátTriểnỨngDungWeb\ThucHanh\Backend\Backend-Part1\contactbook-backend>
git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        package-lock.json
        package.json

nothing added to commit but untracked files present (use "git add" to track)
PS E:\CT449_PhátTriểnỨngDungWeb\ThucHanh\Backend\Backend-Part1\contactbook-backend>
```

- Cho git quản lý 3 tập tin .gitignore, package-lock.json, package.json:

```
PS E:\CT449_PhátTriểnỨngDungWeb\ThucHanh\Backend\Backend-Part1\contactbook-backend>
git add .gitignore package-lock.json package.json
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'package-lock.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'package.json', LF will be replaced by CRLF the next time Git touches it
PS E:\CT449_PhátTriểnỨngDungWeb\ThucHanh\Backend\Backend-Part1\contactbook-backend>
git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   .gitignore
        new file:   package-lock.json
        new file:   package.json
```

- Lưu các thay đổi bằng lệnh git commit -m "Cai dat du an":

```
PS E:\CT449_PhátTriểnỨngDungWeb\ThucHanh\Backend\Backend-Part1\contactbook-backend>
git commit -m 'Cai dat du an'
[master (root-commit) 3e21ba8] Cai dat du an
3 files changed, 1007 insertions(+)
create mode 100644 .gitignore
create mode 100644 package-lock.json
create mode 100644 package.json
PS E:\CT449_PhátTriểnỨngDungWeb\ThucHanh\Backend\Backend-Part1\contactbook-backend>
```

- Upload dự án cục bộ lên GitHub:

```
git remote add origin <github-url>
git push origin master
```

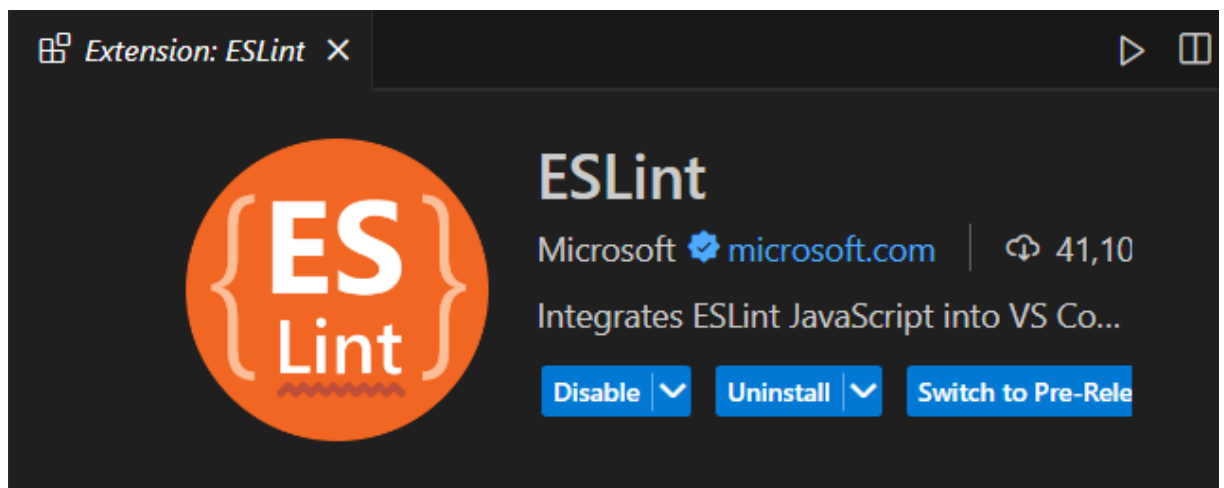
```
PS E:\CT449_PhatTrienUngDungWeb\ThucHanh\Backend\Backend-Part1\contactbook-backend>
git remote add origin https://github.com/TranTanDat2003/B2105571-TranTanDat-Backend-1.git
PS E:\CT449_PhatTrienUngDungWeb\ThucHanh\Backend\Backend-Part1\contactbook-backend>
git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 9.23 KiB | 9.23 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/TranTanDat2003/B2105571-TranTanDat-Backend-1.git
* [new branch]      master -> master
PS E:\CT449_PhatTrienUngDungWeb\ThucHanh\Backend\Backend-Part1\contactbook-backend>
```

- Kiểm tra lại các tập tin đã được tải lên GitHub:

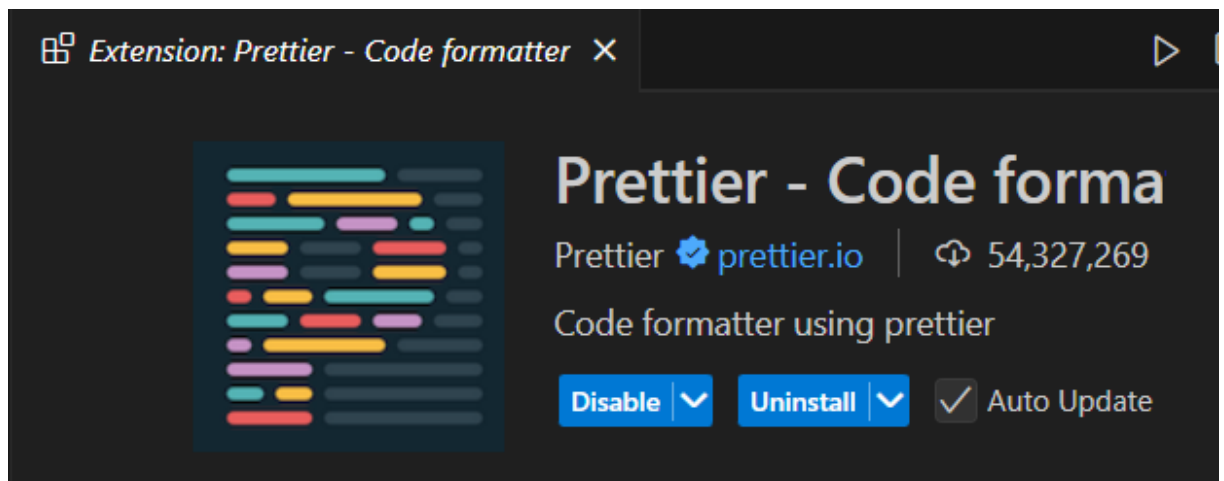


Bước 3: Cấu hình Visual Studio Code, ESLint và Prettier

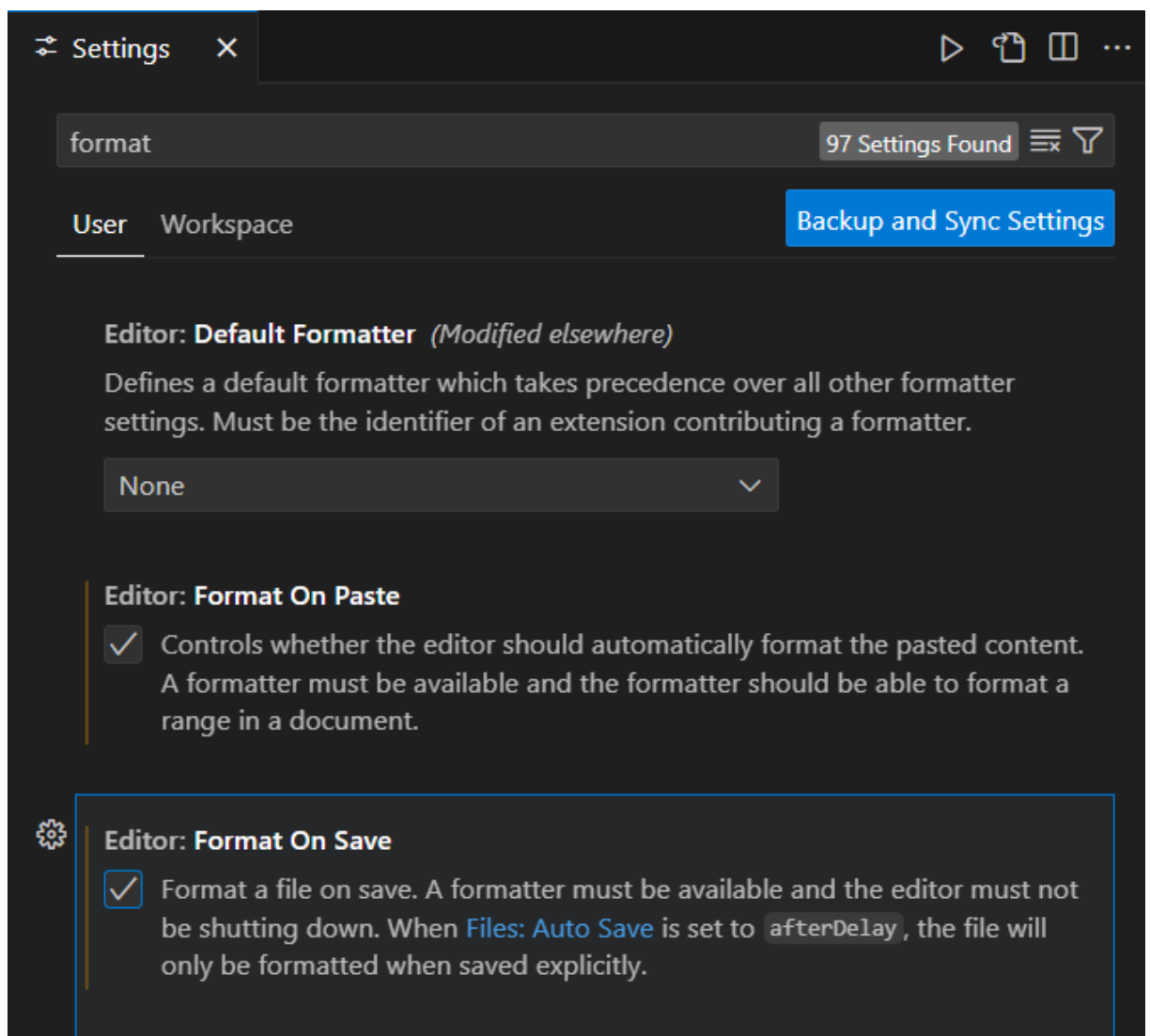
- Cài đặt ESLint extension:



- Cài đặt Prettier – Code formatter:



- Cấu hình VSCode cho phép định dạng mã lệnh khi save hoặc paste:



- Cài đặt các gói thư viện cần thiết:

```
PS E:\CT449_PhátTriểnỨngDụngWeb\ThucHanh\Backend\Backend-Part1\contactbook-backend>
npm i -D eslint prettier eslint-config-prettier

added 91 packages, and audited 163 packages in 4s

37 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS E:\CT449_PhátTriểnỨngDụngWeb\ThucHanh\Backend\Backend-Part1\contactbook-backend>
```

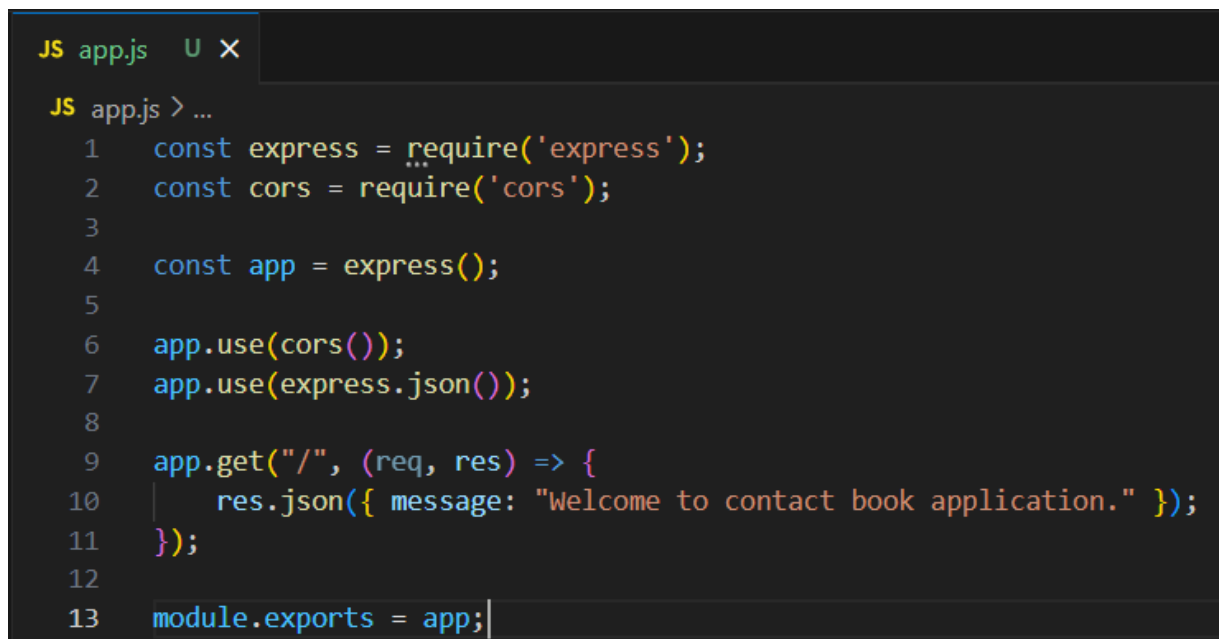
- Tạo tập tin tên .eslintrc.js tại thư mục gốc dự án:



```
.eslintrc.js > ...
1  module.exports = {
2    env: {
3      node: true,
4      commonjs: true,
5      es2021: true,
6    },
7    extends: ['eslint:recommended', 'prettier'],
8  };
```

Bước 4: Cài đặt Express

- app.js:



```
JS app.js U X
JS app.js > ...
1  const express = require('express');
2  const cors = require('cors');
3
4  const app = express();
5
6  app.use(cors());
7  app.use(express.json());
8
9  app.get("/", (req, res) => {
10    res.json({ message: "Welcome to contact book application." });
11  });
12
13  module.exports = app;
```

- server.js:

```
JS server.js U X
JS server.js > ...
1  const app = require("./app");
2  const config = require("./app/config");
3
4  const PORT = config.app.port;
5  app.listen(PORT, () => {
6    console.log(`Server is running on port ${PORT}.`);
7  });
```

- app/config/index.js:

```
JS index.js U X
app > config > JS index.js > ...
1  const config = {
2    app: {
3      port: process.env.PORT || 3000,
4    }
5  };
6
7  module.exports = config;
```

- Chạy server:

```
PS E:\CT449_PhátTriểnỨngDùngWeb\ThucHanh\Backend-Part1\contactbook-backend>
node server.js
Server is running on port 3000.
```

- Kiểm tra kết quả:

```
localhost:3000
localhost:3000
Pretty-print ☐
{"message":"Welcome to contact book application."}
```


- Cài đặt gói hỗ trợ `nodemon`:

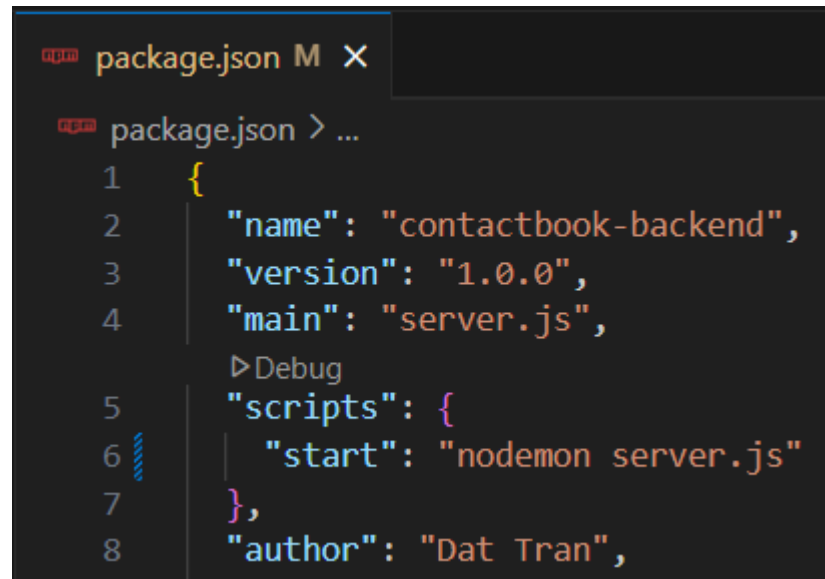
```
PS E:\CT449_PhátTriểnỨngDụngWeb\ThucHanh\Backend\Backend-Part1\contactbook-backend>
npm install nodemon --save-dev

added 23 packages, and audited 186 packages in 2s

41 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

- Thay đổi cấu hình mục “scripts” tại package.json:



```
package.json M X
package.json > ...
1  {
2    "name": "contactbook-backend",
3    "version": "1.0.0",
4    "main": "server.js",
5    "scripts": {
6      "start": "nodemon server.js"
7    },
8    "author": "Dat Tran",
```

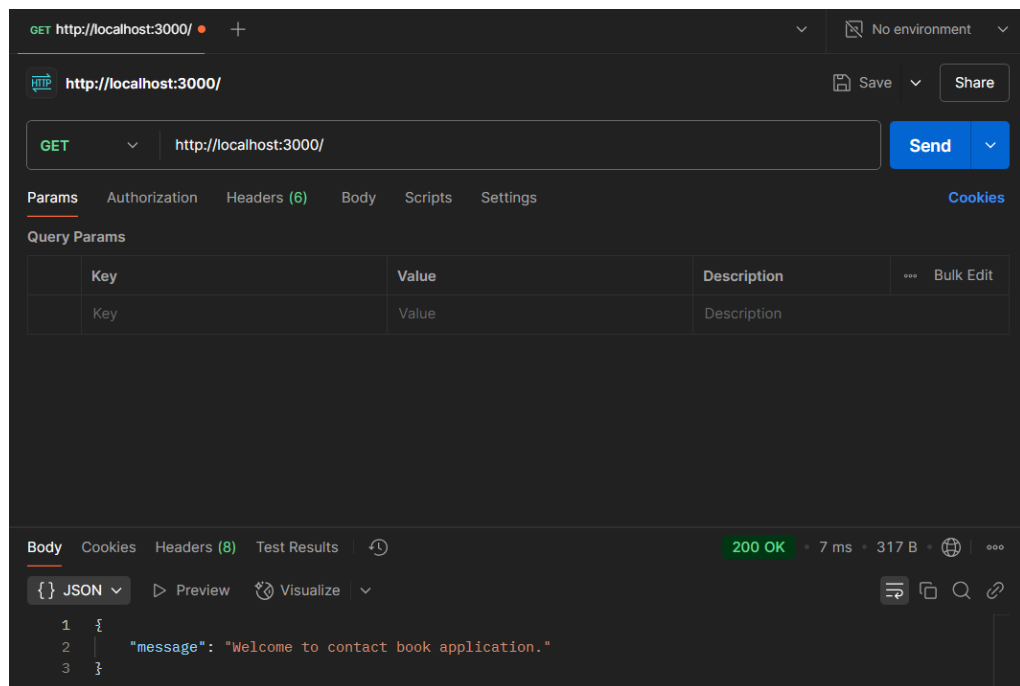
- Chạy lại server:

```
PS E:\CT449_PhátTriểnỨngDụngWeb\ThucHanh\Backend\Backend-Part1\contactbook-backend>
npm start

> contactbook-backend@1.0.0 start
> nodemon server.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 server.js`
Server is running on port 3000.
```

- Kiểm tra kết quả bằng postman:

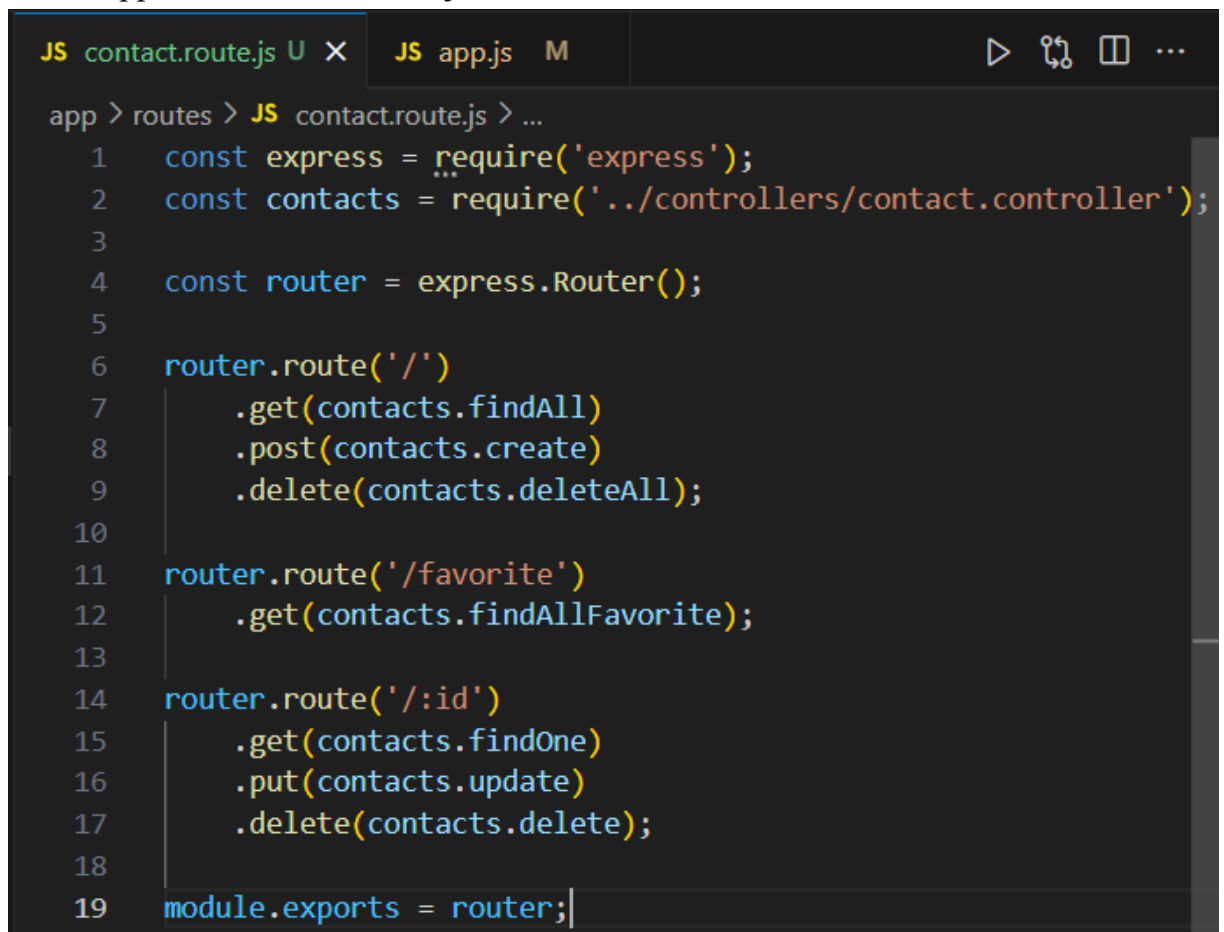


Bước 5: Định nghĩa Controller và các route

- `app/controllers/contact.controller.js`:

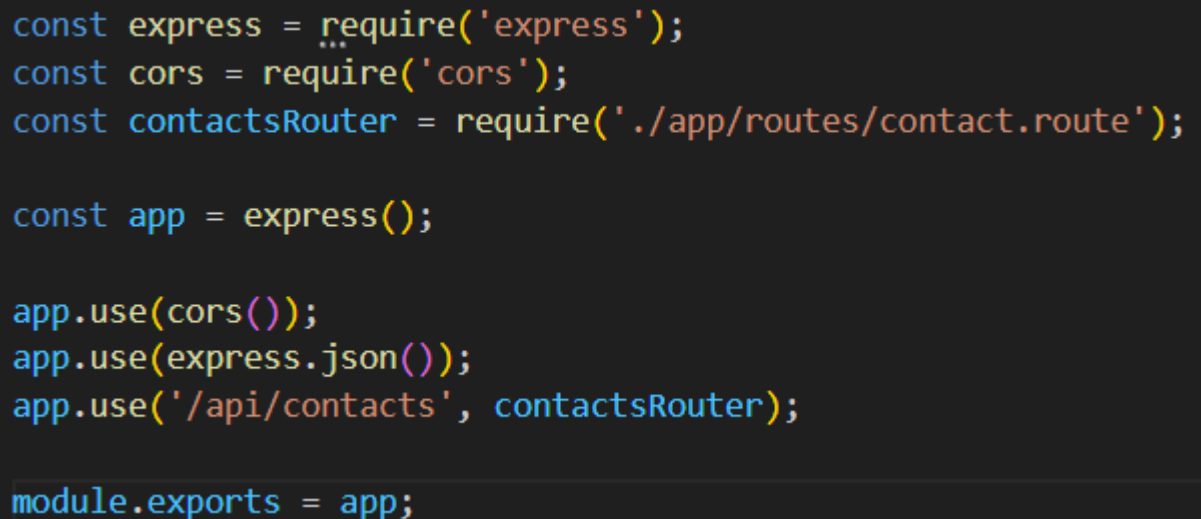
```
JS contact.controller.js U X
app > controllers > JS contact.controller.js > ...
1  exports.create = (req, res) => {
2    res.send({ message: "create handler" });
3  };
4
5  exports.findAll = (req, res) => {
6    res.send({ message: "findAll handler" });
7  };
8
9  exports.findOne = (req, res) => {
10   res.send({ message: "findOne handler" });
11 };
12
13 exports.update = (req, res) => {
14   res.send({ message: "update handler" });
15 };
16
17 exports.delete = (req, res) => {
18   res.send({ message: "delete handler" });
19 };
20
21 exports.deleteAll = (req, res) => {
22   res.send({ message: "deleteAll handler" });
23 };
24
25 exports.findAllFavorite = (req, res) => {
26   res.send({ message: "findAllFavorite handler" });
27 };
```

- app/routes/contact.route.js:



```
JS contact.route.js U X JS app.js M
app > routes > JS contact.route.js > ...
1  const express = require('express');
2  const contacts = require('../controllers/contact.controller');
3
4  const router = express.Router();
5
6  router.route('/')
7    .get(contacts.findAll)
8    .post(contacts.create)
9    .delete(contacts.deleteAll);
10
11  router.route('/favorite')
12    .get(contacts.findAllFavorite);
13
14  router.route('/:id')
15    .get(contacts.findOne)
16    .put(contacts.update)
17    .delete(contacts.delete);
18
19  module.exports = router;
```

- Hiệu chỉnh app.js:



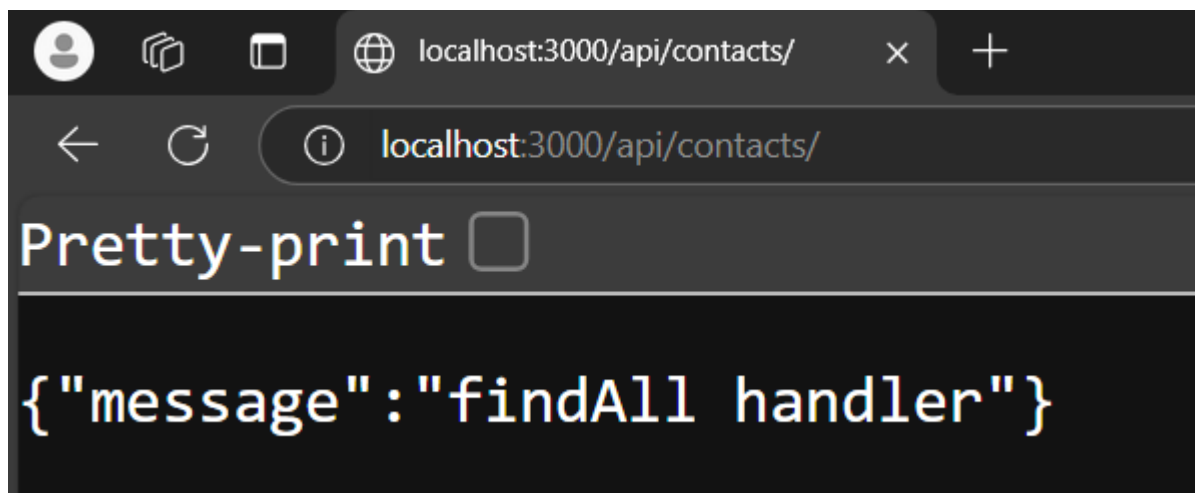
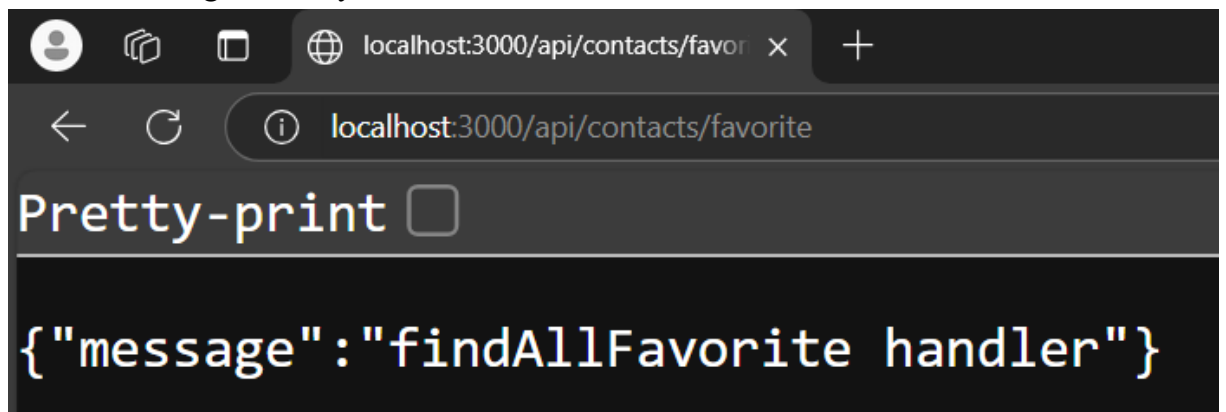
```
const express = require('express');
const cors = require('cors');
const contactsRouter = require('./app/routes/contact.route');

const app = express();

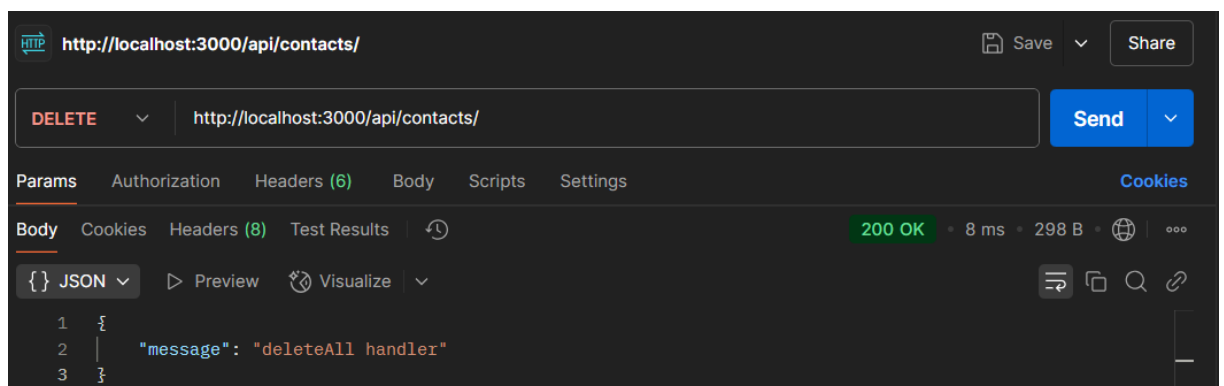
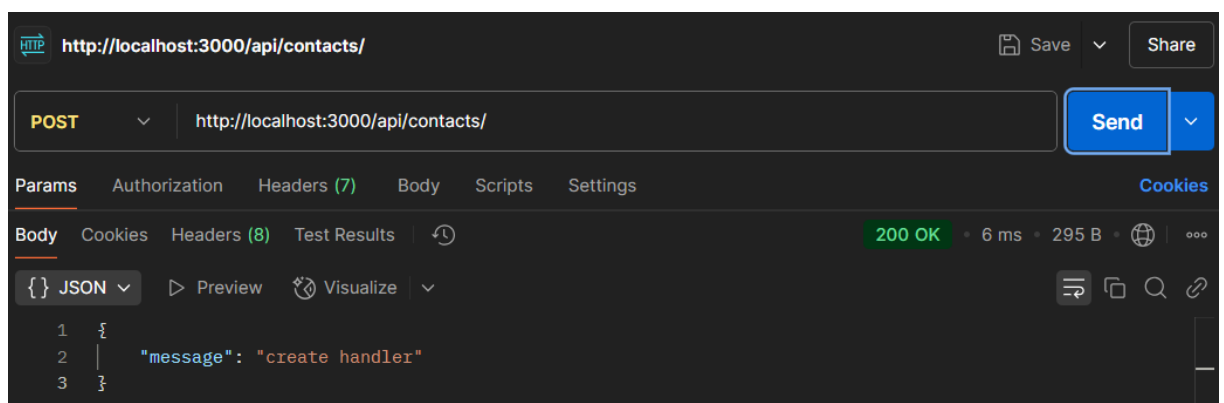
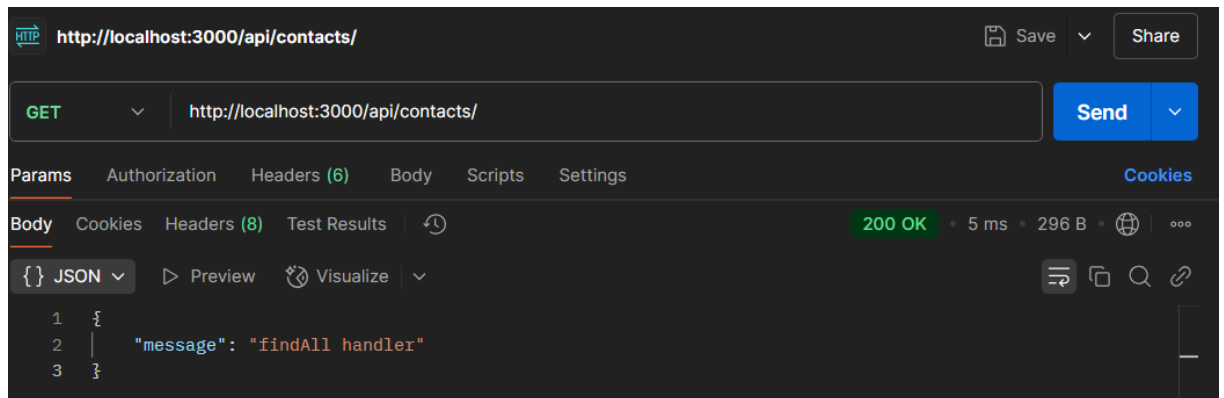
app.use(cors());
app.use(express.json());
app.use('/api/contacts', contactsRouter);

module.exports = app;
```

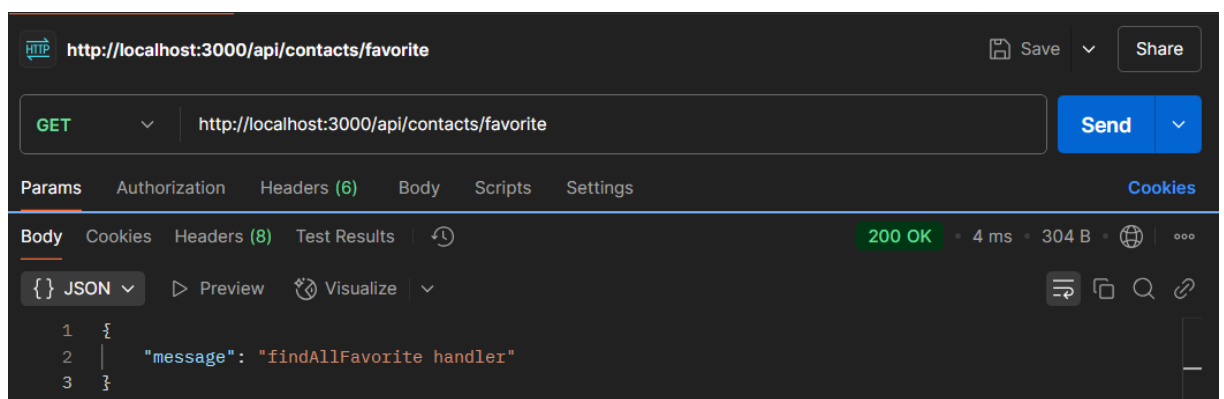
- Sử dụng trình duyệt kiểm tra các route vừa tạo:



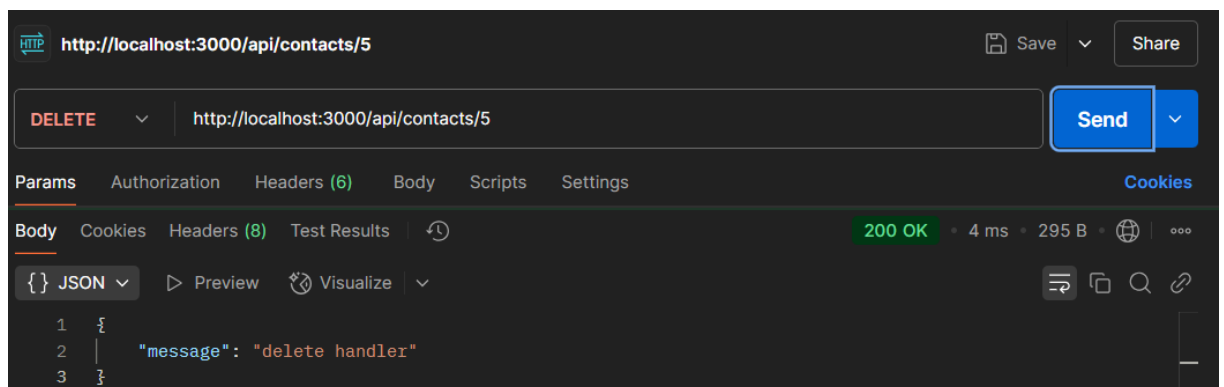
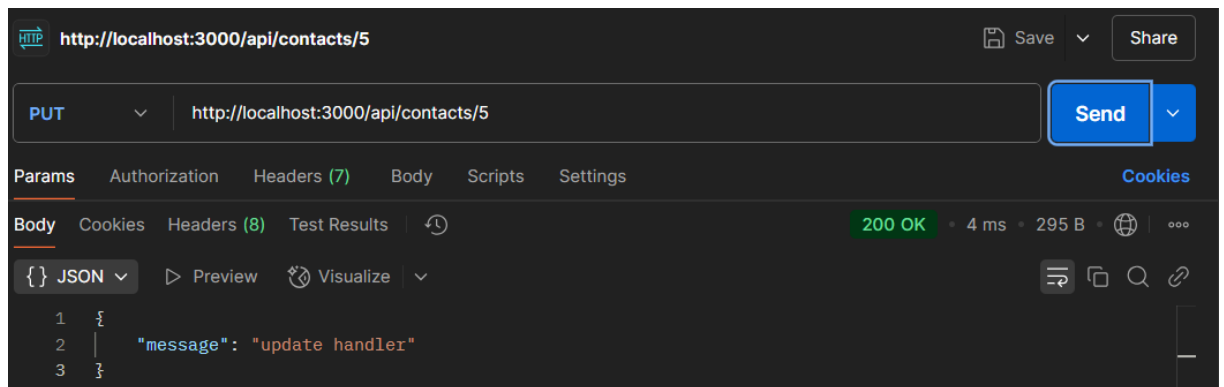
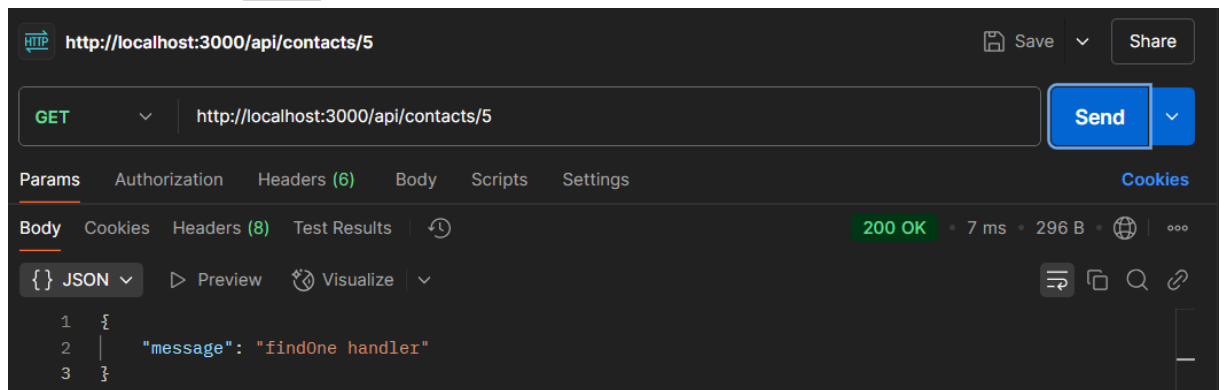
- Sử dụng Postman kiểm tra các route vừa tạo:
 - Route “/”:



- Route “/favorite”:



- Route “/:id”:



Bước 6: Cài đặt xử lý lỗi

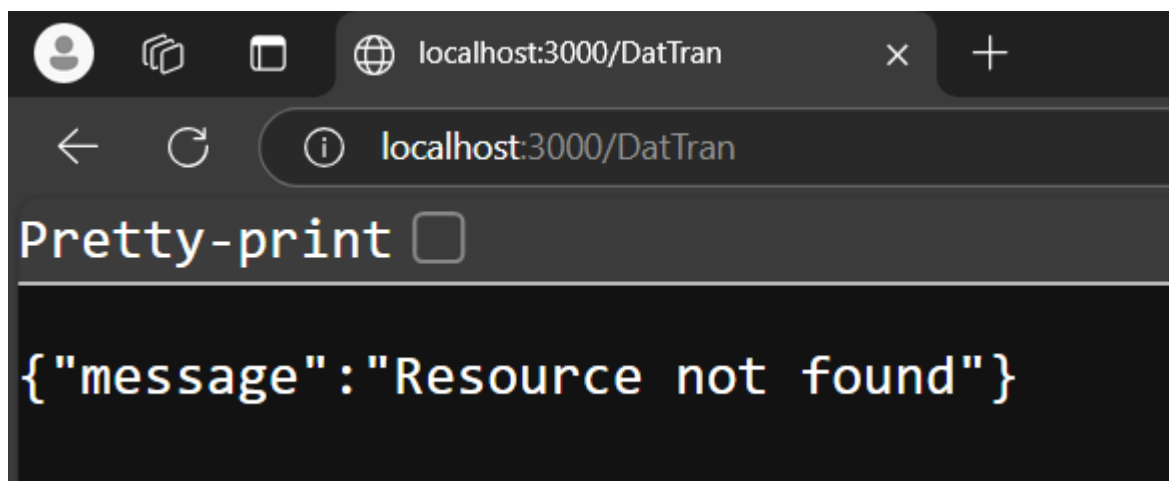
- app/api-error.js:

```
JS api-error.js U X
app > JS api-error.js > ...
1 class ApiError extends Error {
2   constructor(statusCode, message) {
3     super();
4     this.statusCode = statusCode;
5     this.message = message;
6   }
7 }
8
9 module.exports = ApiError;
```

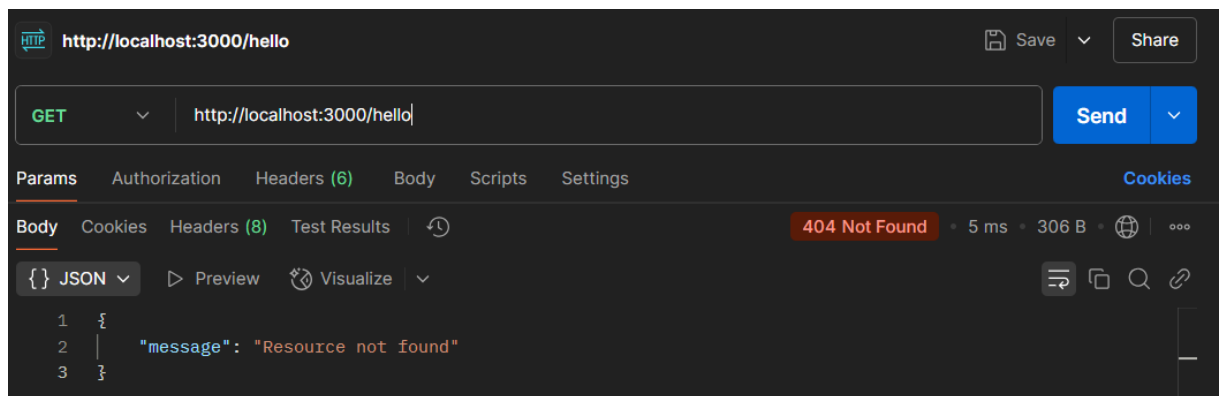
- Thêm các middleware xử lý lỗi vào tập tin app.js:

```
JS app.js M X
JS app.js > ...
1  const express = require('express');
2  const cors = require('cors');
3  const contactsRouter = require('./app/routes/contact.route');
4  const ApiError = require('./app/api-error');
5
6  const app = express();
7
8  app.use(cors());
9  app.use(express.json());
10 app.use('/api/contacts', contactsRouter);
11
12 app.use((req, res, next) => {
13   return next(new ApiError(404, "Resource not found"));
14 });
15
16 app.use((err, req, res, next) => {
17   return res.status(err.statusCode || 500).json({
18     message: err.message || "Internal Server Error",
19   });
20 });
21
22 app.get("/", (req, res) => {
23   res.json({ message: "Welcome to contact book application." });
24 });
25 app.use('/api/contacts', contactsRouter);
26
27 module.exports = app;
```

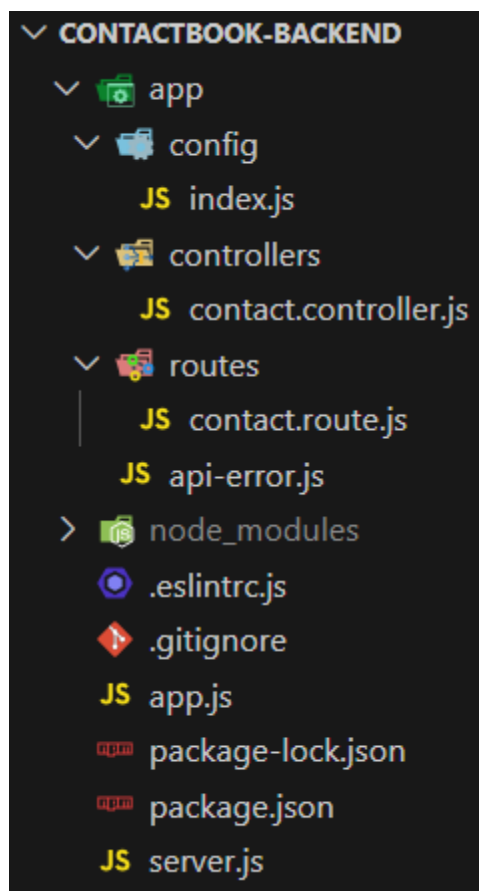
- Kiểm tra bằng trình duyệt:



- Kiểm tra bằng Postman:



- Cấu trúc cây thư mục:

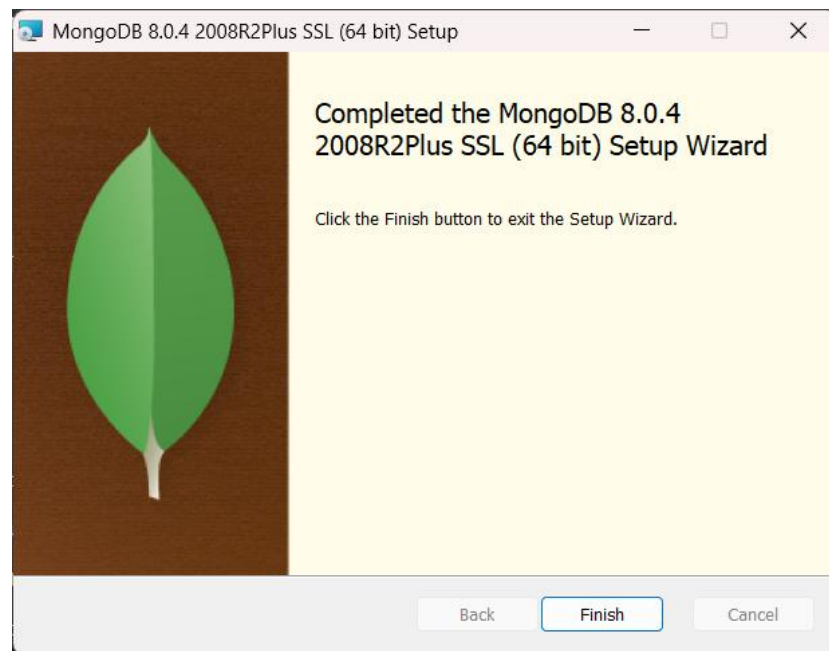


Đường dẫn đến repo github dự án Backend 1:

<https://github.com/TranTanDat2003/B2105571-TranTanDat-Backend-1.git>

ỨNG DỤNG CONTACTBOOK – BACKEND – PHẦN 2

Bước 0: Cài đặt MongoDB



Bước 1: Cài đặt thư viện mongodb, định nghĩa hàm trợ giúp kết nối với lớp dịch vụ truy xuất cơ sở dữ liệu (CSDL)

- Cài đặt thư viện mongodb vào dự án:

```
PS E:\CT449_PhatTrienUngDungWeb\ThucHanh\Backend\Backend-Part1\contactbook-backend> npm install mongodb

added 11 packages, and audited 197 packages in 3s

41 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

- app/config/index.js:

```
JS index.js M X
app > config > JS index.js > ...
1  const config = {
2    app: {
3      port: process.env.PORT || 3000,
4    },
5    db: {
6      uri: process.env.MONGODB_URI || 'mongodb://127.0.0.1:27017/contactbook',
7    }
8  };
9
10 module.exports = config;
```

- app/utils/mongodb.util.js:

```
JS mongodb.util.js U X
app > utils > JS mongodb.util.js > ...
1  const { MongoClient } = require('mongodb');
2
3  class MongoDB {
4    static connect = async (uri) => {
5      if (this.client) return this.client;
6      this.client = await MongoClient.connect(uri);
7      return this.client;
8    };
9  }
10
11  module.exports = MongoDB;
```

- server.js:

```
JS server.js M X
JS server.js > ...
1  const app = require('./app');
2  const config = require('./app/config');
3  const MongoDB = require('./app/utils/mongodb.util');
4
5  async function startServer() {
6    try {
7      await MongoDB.connect(config.db.uri);
8      console.log("Connected to the database!");
9
10     const PORT = config.app.port;
11     app.listen(PORT, () => {
12       console.log(`Server is running on port ${PORT}`);
13     });
14   } catch (error) {
15     console.log("Cannot connect to the database!", error);
16     process.exit();
17   }
18 }
19
20 startServer();
```

- app/services/contact.service.js:

```

JS contact.service.js U X
app > services > JS contact.service.js > ...
1  const { ObjectId } = require('mongodb');
2
3  class ContactService {
4      constructor(client) {
5          this.Contact = client.db().collection('contacts');
6      }
7      //
8  }
9
10 module.exports = ContactService;

```

Bước 2: Cài đặt các handler

- Cài đặt handler create:

- app/controllers/contact.controller.js:

```

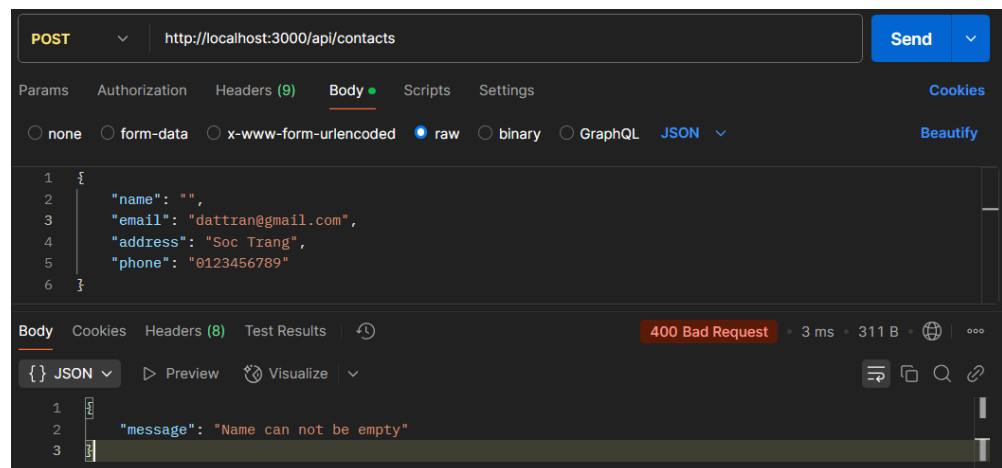
JS contact.controller.js M X
app > controllers > JS contact.controller.js > ...
1  const ContactService = require("../services/contact.service");
2  const MongoDB = require("../utils/mongodb.util");
3  const ApiError = require("../api-error");
4
5  exports.create = async (req, res, next) => {
6      if (!req.body?.name) {
7          return next(new ApiError(400, "Name can not be empty"));
8      }
9
10     try {
11         const contactService = new ContactService(MongoDB.client);
12         const document = await contactService.create(req.body);
13         return res.send(document);
14     } catch (error) {
15         return next(
16             new ApiError(500, `An error occurred while creating the contact`)
17         );
18     }
19 };

```

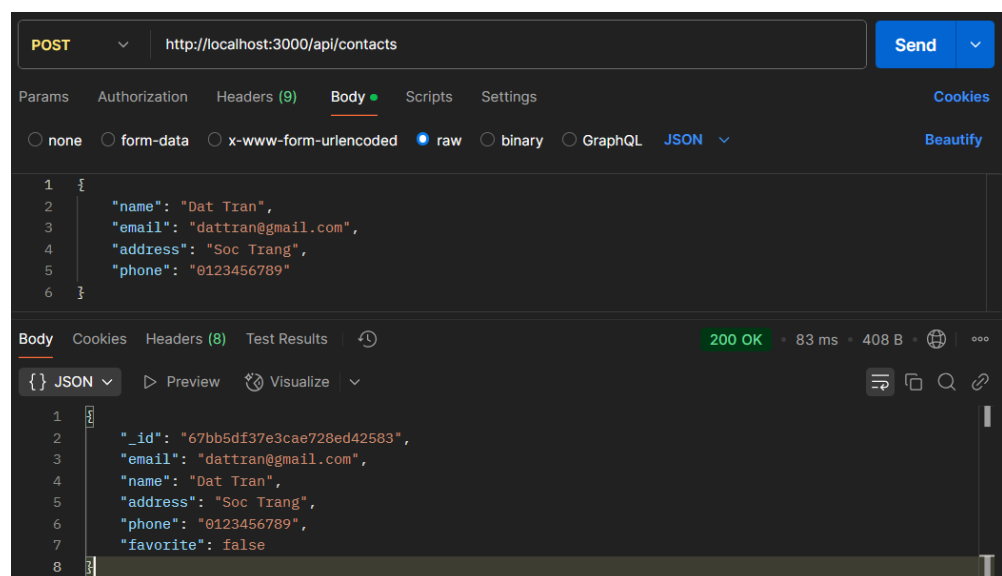
- app/services/contact.service.js:

```
JS contact.service.js U X
app > services > JS contact.service.js > ...
1  const { ObjectId } = require('mongodb');
2
3  class ContactService {
4    constructor(client) {
5      this.Contact = client.db().collection('contacts');
6    }
7    extractConactData(payload) {
8      const contact = {
9        name: payload.name,
10       email: payload.email,
11       address: payload.address,
12       phone: payload.phone,
13       favorite: payload.favorite,
14     };
15     Object.keys(contact).forEach(
16       (key) => contact[key] === undefined && delete contact[key]
17     );
18     return contact;
19   }
20
21   async create(payload) {
22     const contact = this.extractConactData(payload);
23     const result = await this.Contact.findOneAndUpdate(
24       contact,
25       { $set: { favorite: contact.favorite === true }},
26       { returnDocument: "after", upsert: true }
27     );
28     return result;
29   }
30 }
31
32 module.exports = ContactService;
```

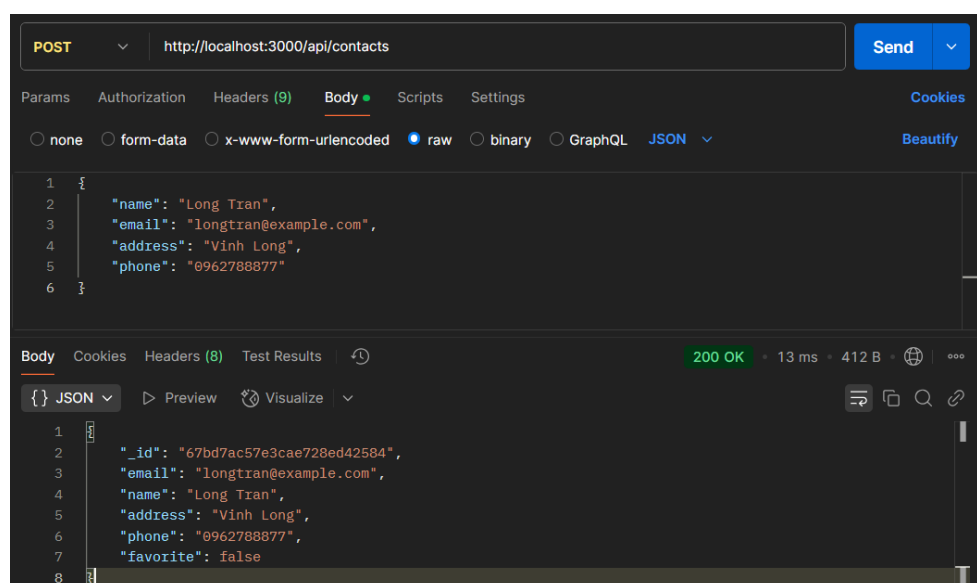
- Kiểm tra handler create bằng Postman:



- Tạo contact có name: Dat Tran



- Tạo contact có name: Long Tran



- Cài đặt handler `findAll`:

- `app/controllers/contact.controller.js`:

```

JS contact.controller.js X
app > controllers > JS contact.controller.js > findAll > findAll
20
21 exports.findAll = async (req, res, next) => {
22   let documents = [];
23
24   try {
25     const contactService = new ContactService(MongoDB.client);
26     const {name} = req.query;
27     if (name) {
28       documents = await contactService.findByName(name);
29     } else {
30       documents = await contactService.find({});
31     }
32   } catch (error) {
33     return next(
34       new ApiError(500, `An error occurred while retrieving contacts`)
35     );
36   }
37
38   return res.send(documents);
39 };
40

```

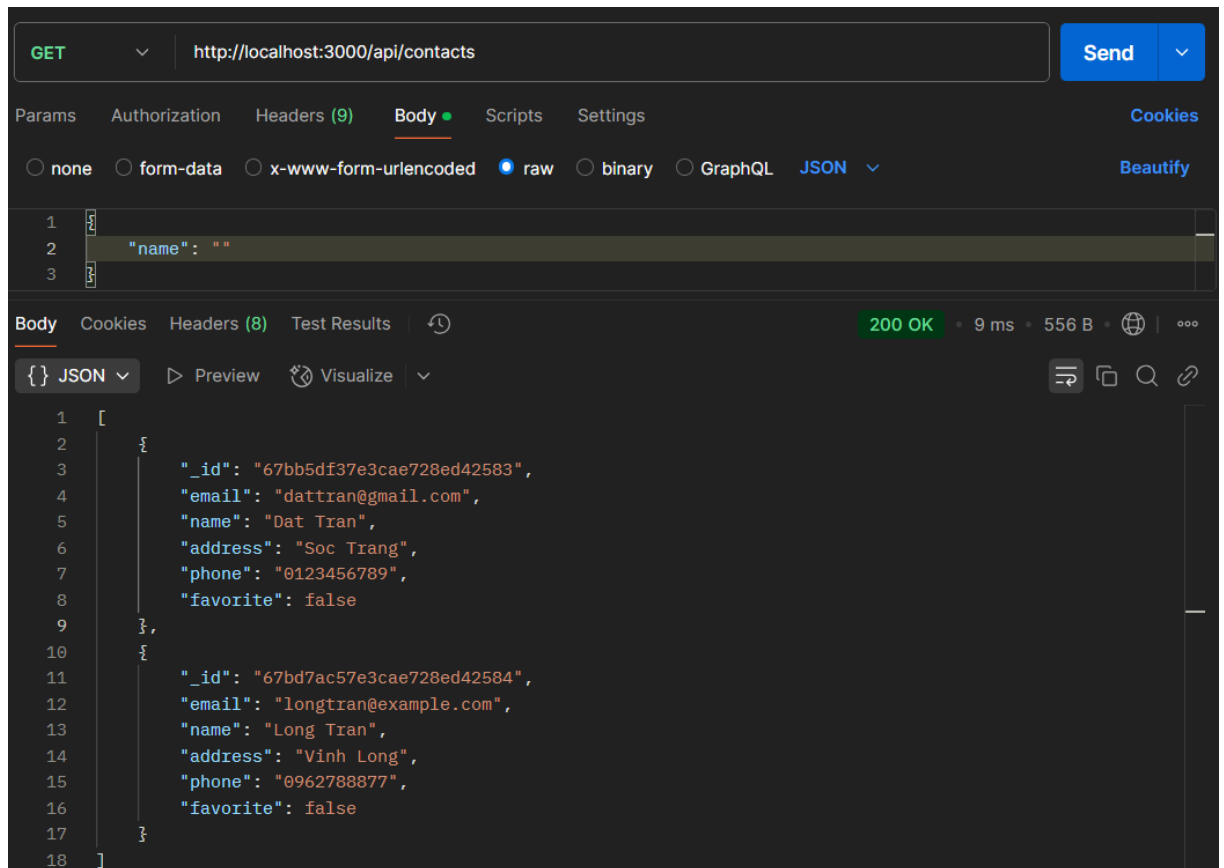
- `app/services/contact.service.js`:

```

JS contact.service.js U X
app > services > JS contact.service.js > ContactService
30
31 class ContactService {
32   async find(filter) {
33     const cursor = await this.Contact.find(filter);
34     return await cursor.toArray();
35   }
36
37   async findByName(name) {
38     return await this.find({
39       name: {$regex: new RegExp(new RegExp(name)), $options: "i"},
40     });
41   }
42
43   module.exports = ContactService;

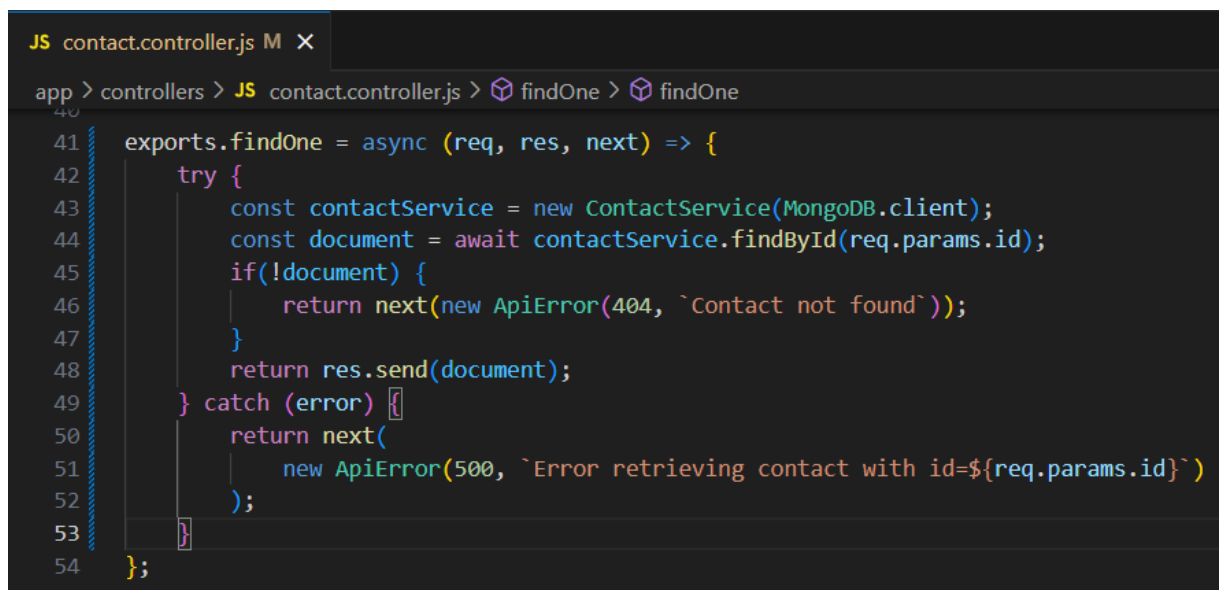
```

- Kiểm tra handler findAll bằng Postman:



- Cài đặt handler findOne:

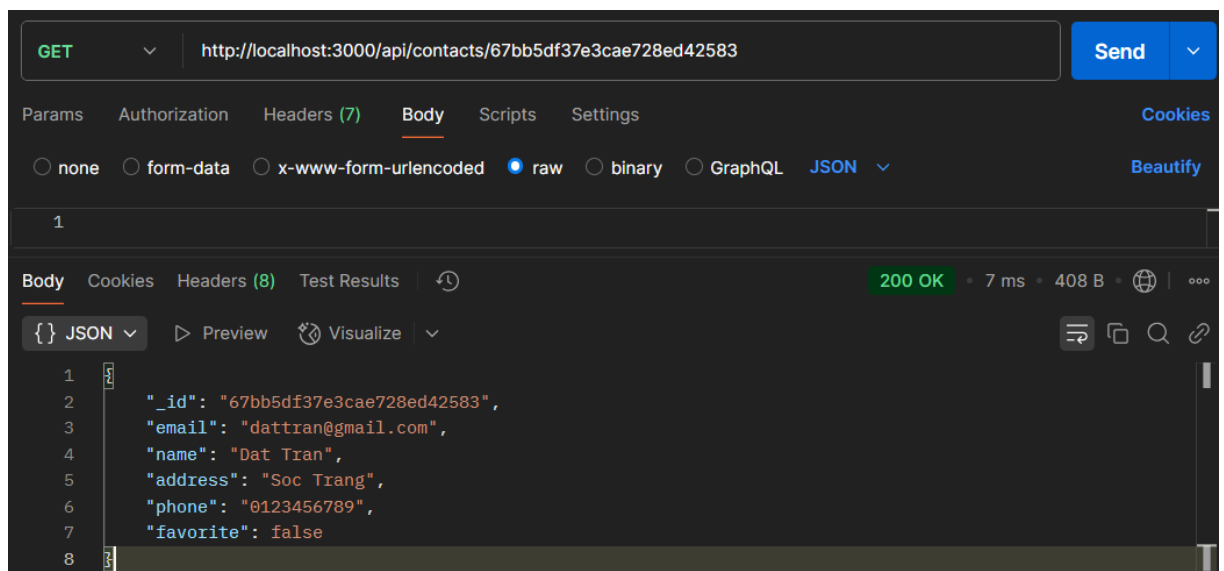
- `app/controllers/contact.controller.js`:



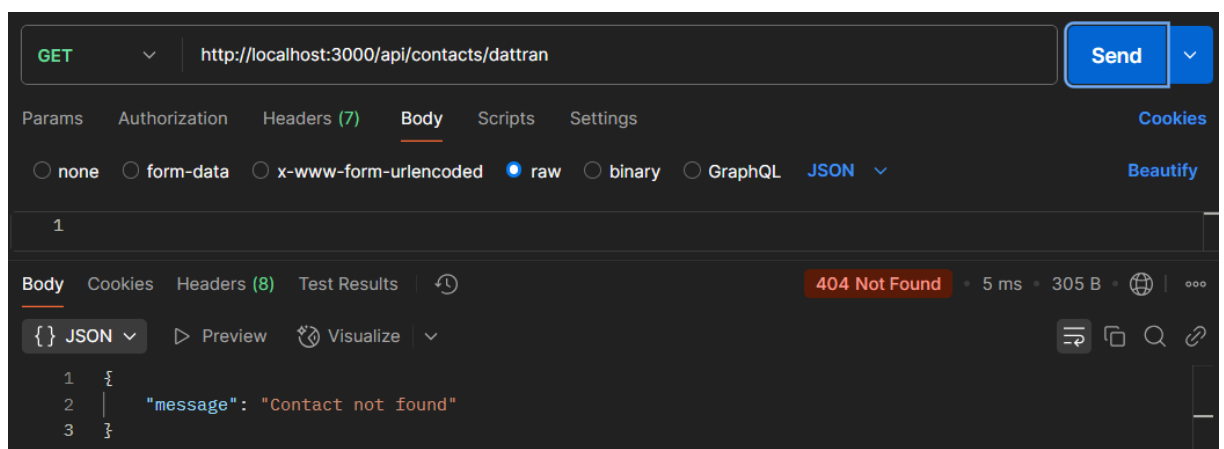
- app/services/contact.service.js:

```
JS contact.service.js U X
app > services > JS contact.service.js > ContactService
3 class ContactService {
42   async findById(id) {
43     return await this.Contact.findOne({
44       _id: ObjectId.isValid(id) ? new ObjectId(id) : null,
45     });
46   }
47 }
48
49 module.exports = ContactService;
```

- Kiểm tra handler findOne bằng Postman:



- Nếu id không tồn tại:



- Cài đặt handler update:

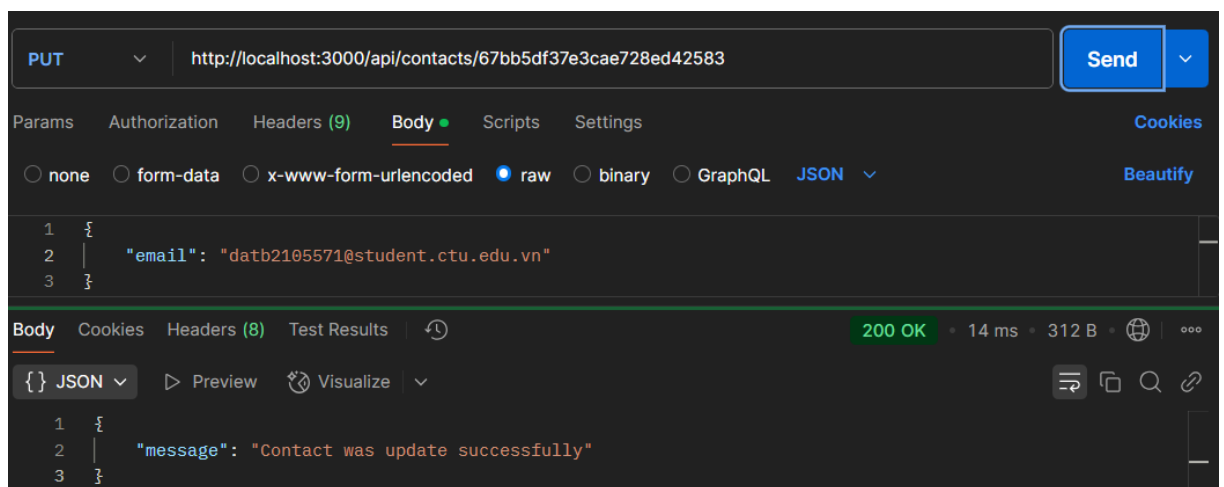
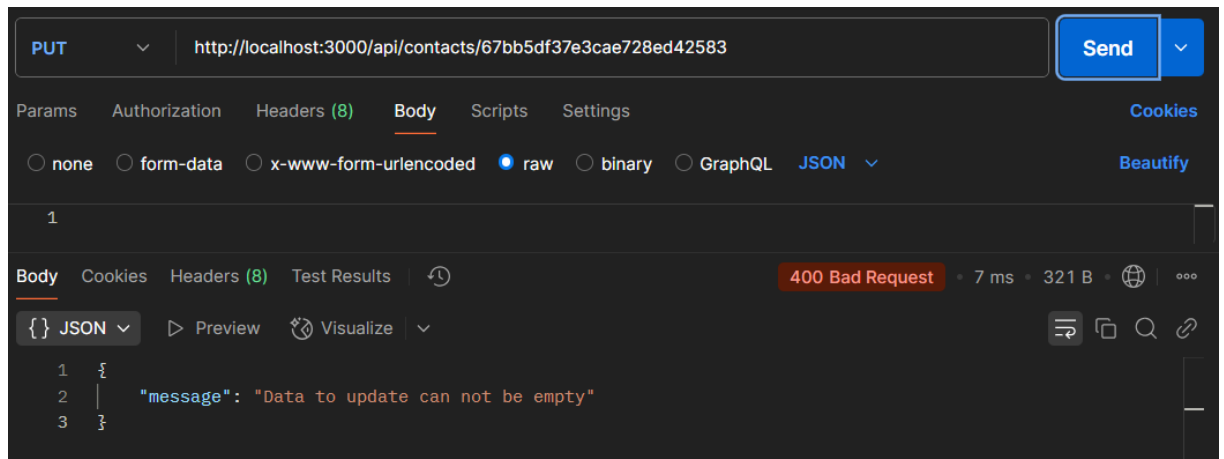
- app/controllers/contact.controller.js:

```
JS contact.controller.js M X
app > controllers > JS contact.controller.js > update > update
56 exports.update = async (req, res, next) => {
57   if (Object.keys(req.body).length === 0) {
58     return next(new ApiError(400, "Data to update can not be empty"));
59   }
60
61   try {
62     const contactService = new ContactService(MongoDB.client);
63     const document = await contactService.update(req.params.id, req.body);
64     if(!document) {
65       return next(new ApiError(404, `Contact not found`));
66     }
67     return res.send({ message: "Contact was update successfully"});
68   } catch (error) {}
69   return next(
70     new ApiError(500, `Error updating contact with id=${req.params.id}`)
71   );
72 }
73 }
```

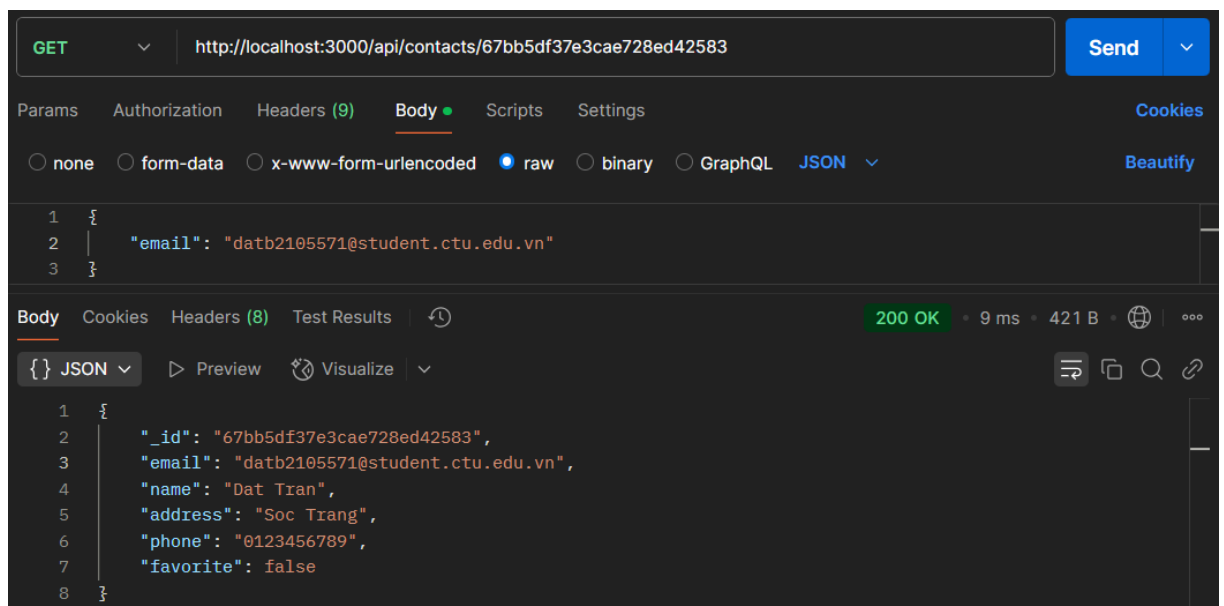
- app/services/contact.service.js:

```
JS contact.service.js U X
app > services > JS contact.service.js > ContactService > update
3 class ContactService {
48   async update(id, payload) {}
49     const filter = {
50       _id: ObjectId.isValid(id) ? new ObjectId(id) : null,
51     };
52     const update = this.extractConactData(payload);
53     const result = await this.Contact.findOneAndUpdate(
54       filter,
55       { $set: update },
56       { returnDocument: "after" }
57     );
58     return result.value;
59   }
60 }
61
62 module.exports = ContactService;
```

- Kiểm tra handler update bằng Postman:



- Sau khi update thành công thì kiểm tra id vừa update:



- **Cài đặt handler delete:**

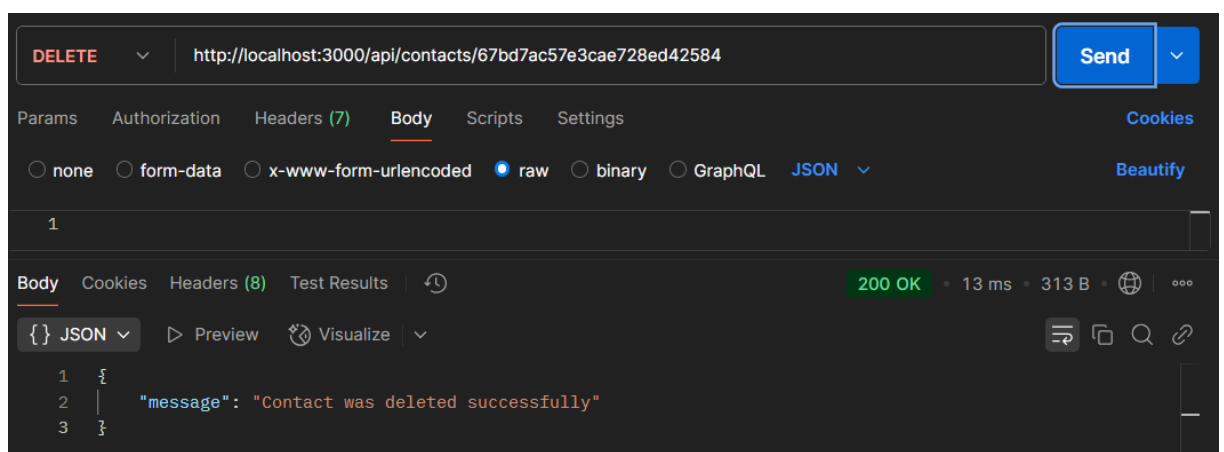
- App/controllers/contact.controller.js:

```
JS contact.controller.js X
app > controllers > JS contact.controller.js > findAll > findAll
75 exports.delete = async (req, res, next) => {
76   try {
77     const contactService = new ContactService(MongoDB.client);
78     const document = await contactService.delete(req.params.id);
79     if(!document) {
80       return next(new ApiError(404, `Contact not found`));
81     }
82     return res.send({ message: "Contact was deleted successfully"});
83   } catch (error) {
84     return next(
85       new ApiError(500, `Could not delete contact with id=${req.params.id}`)
86     );
87   }
88 };
```

- App/services/contact.service.js:

```
JS contact.service.js X
app > services > JS contact.service.js > ContactService
3 class ContactService {
61   async delete(id) {
62     const result = await this.Contact.findOneAndDelete({
63       _id: ObjectId.isValid(id) ? new ObjectId(id) : null,
64     });
65     return result;
66   }
67 }
```

- Kiểm tra handler delete bằng Postman: Xóa contact có name Long Tran.



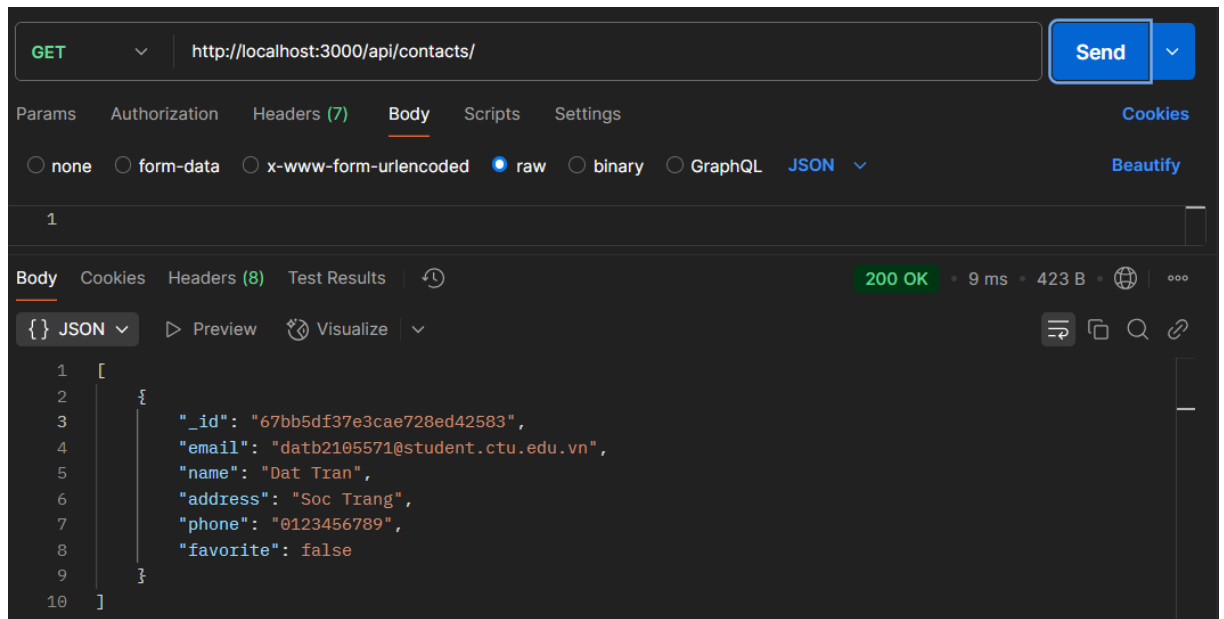
```
DELETE http://localhost:3000/api/contacts/67bd7ac57e3cae728ed42584
Send

Params Authorization Headers (7) Body Scripts Settings Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

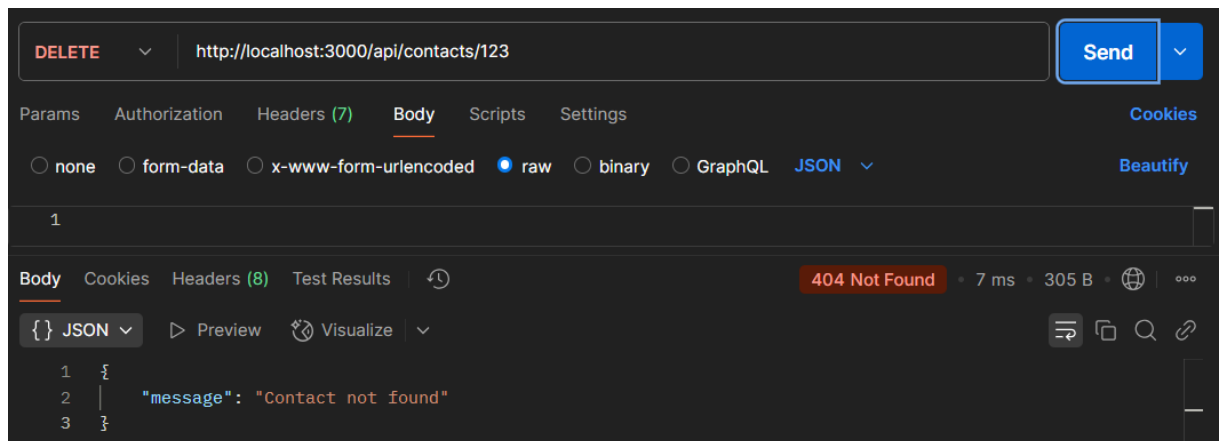
1

Body Cookies Headers (8) Test Results
200 OK 13 ms 313 B
{} JSON Preview Visualize
1 {
2   "message": "Contact was deleted successfully"
3 }
```

- Kiểm tra lại tất cả các contact đang có:

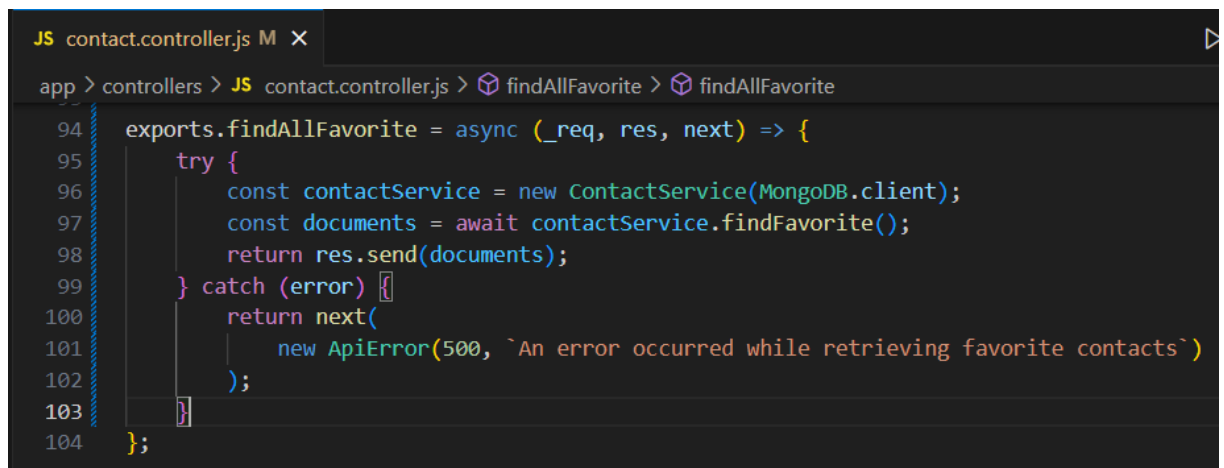


- Xóa một contact có id không tồn tại:



- Cài đặt handler findAllFavorite:

- app/controllers/contact.controller.js:

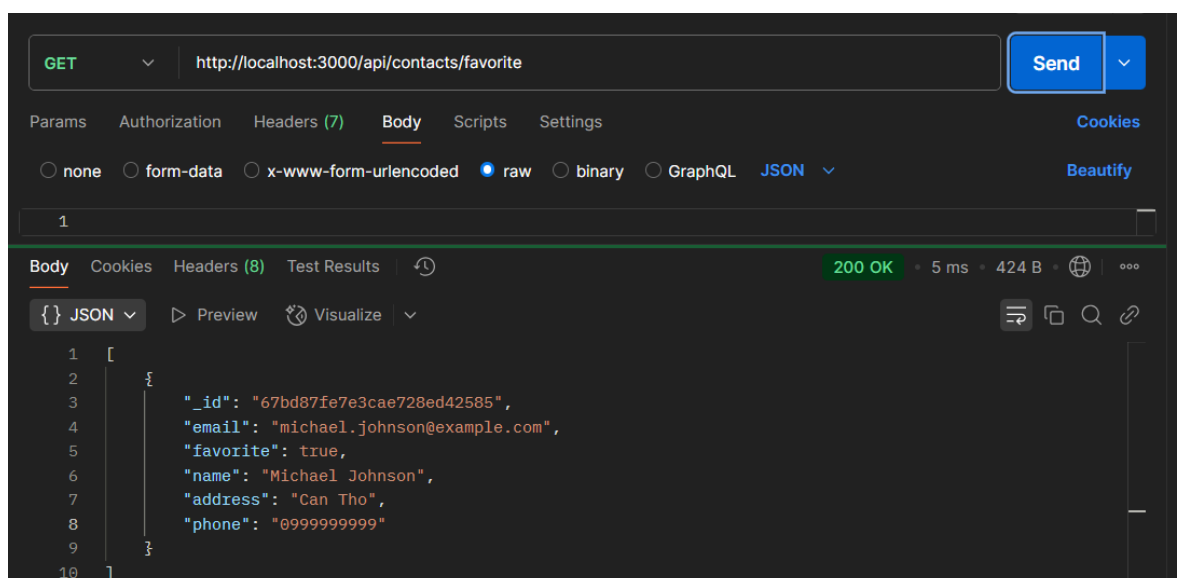
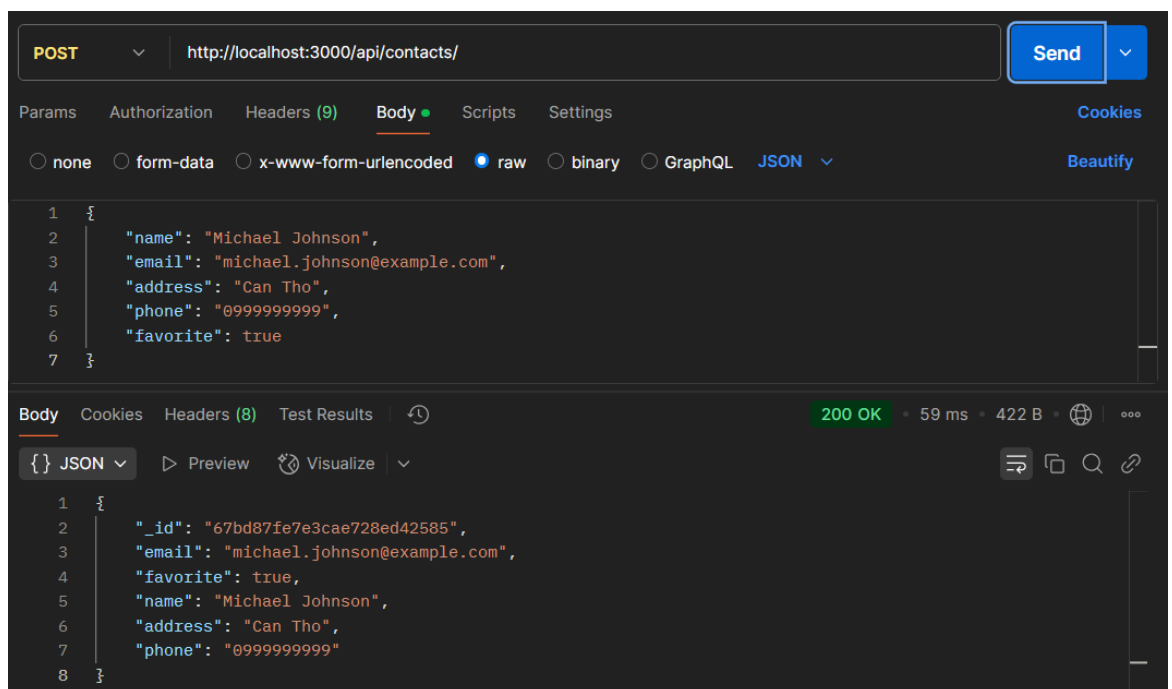


- app/services/contact.service.js:

```
JS contact.service.js X
app > services > JS contact.service.js > ContactService
3 class ContactService {
68   async findFavorite() {
69     return await this.find({ favorite: true });
70   }
71 }
```

- Kiểm tra handler findAllFavorite bằng Postman:

Tạo một contact có favorite là true để thực hiện kiểm tra



- Cài đặt handler deleteAll:
 - App/controllers/contact.controller.js:

```

JS contact.controller.js M X
app > controllers > JS contact.controller.js > deleteAll > deleteAll
90 exports.deleteAll = async (_req, res, next) => {
91   try {
92     const contactService = new ContactService(MongoDB.client);
93     const deletedCount = await contactService.deleteAll();
94     return res.send({
95       message: `${deletedCount} contacts were deleted successfully`,
96     });
97   } catch (error) {
98     return next(
99       new ApiError(500, `An error occurred while removing all contacts`)
100     );
101   }
102 };
103

```

- App/services/contact.service.js:

```

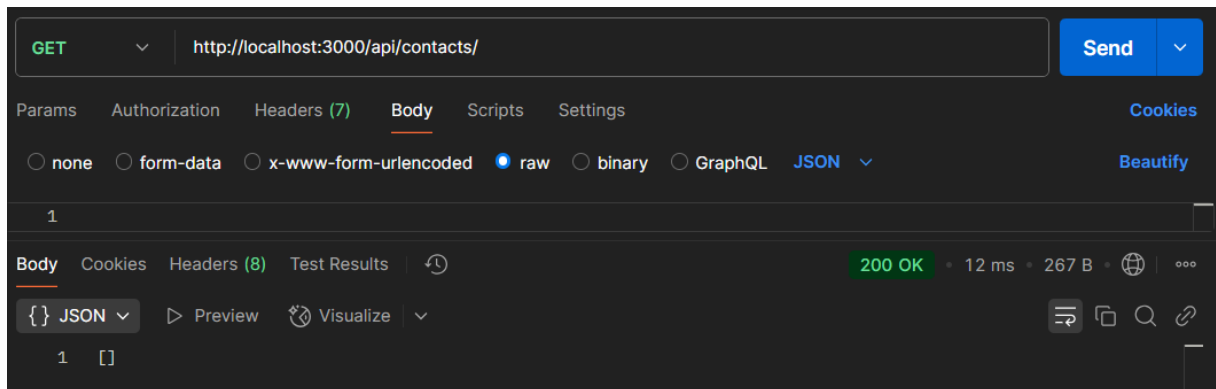
JS contact.service.js U X
app > services > JS contact.service.js > ContactService > deleteAll
3 class ContactService {
72   async deleteAll() {
73     const result = await this.Contact.deleteMany({});
74     return result.deletedCount;
75   }
76 }
77
78 module.exports = ContactService;

```

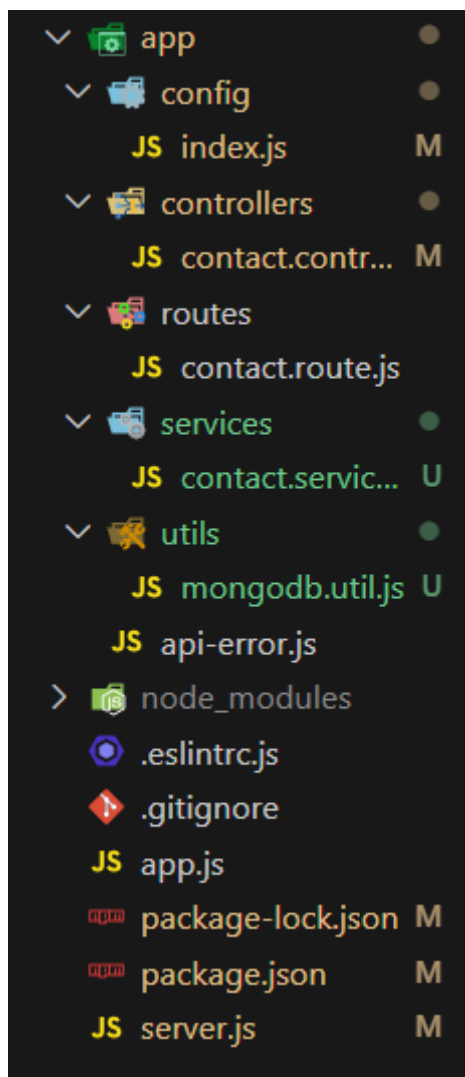
- Kiểm tra handler deleteAll bằng Postman:

The screenshot shows the Postman interface. At the top, the method is set to 'DELETE' and the URL is 'http://localhost:3000/api/contacts/'. The 'Send' button is visible. Below the URL bar, the 'Body' tab is selected, and the content type is set to 'raw'. The response is shown at the bottom, indicating a '200 OK' status with a response time of 29 ms and a body size of 317 B. The response body is a JSON object: { "message": "2 contacts were deleted successfully" }.

- Sau khi xóa thì kiểm tra lại tất cả các contact hiện có:



- Cấu trúc cây thư mục:



Đường dẫn đến repo github dự án Backend 2:

<https://github.com/TranTanDat2003/B2105571-TranTanDat-Backend-1.git>