

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



**PHÁT TRIỂN ỨNG DỤNG WEB
MÃ HỌC PHẦN: CT449**

**Đề tài:
ỨNG DỤNG CONTACTBOOK - BACKEND - PHẦN 2**

Giảng viên hướng dẫn

ThS. Lê Minh Trung

Sinh viên thực hiện

Tên: Lê Tuấn Đạt

MSSV: B2113328


Cần Thơ, ngày tháng năm 2024


MỤC LỤC

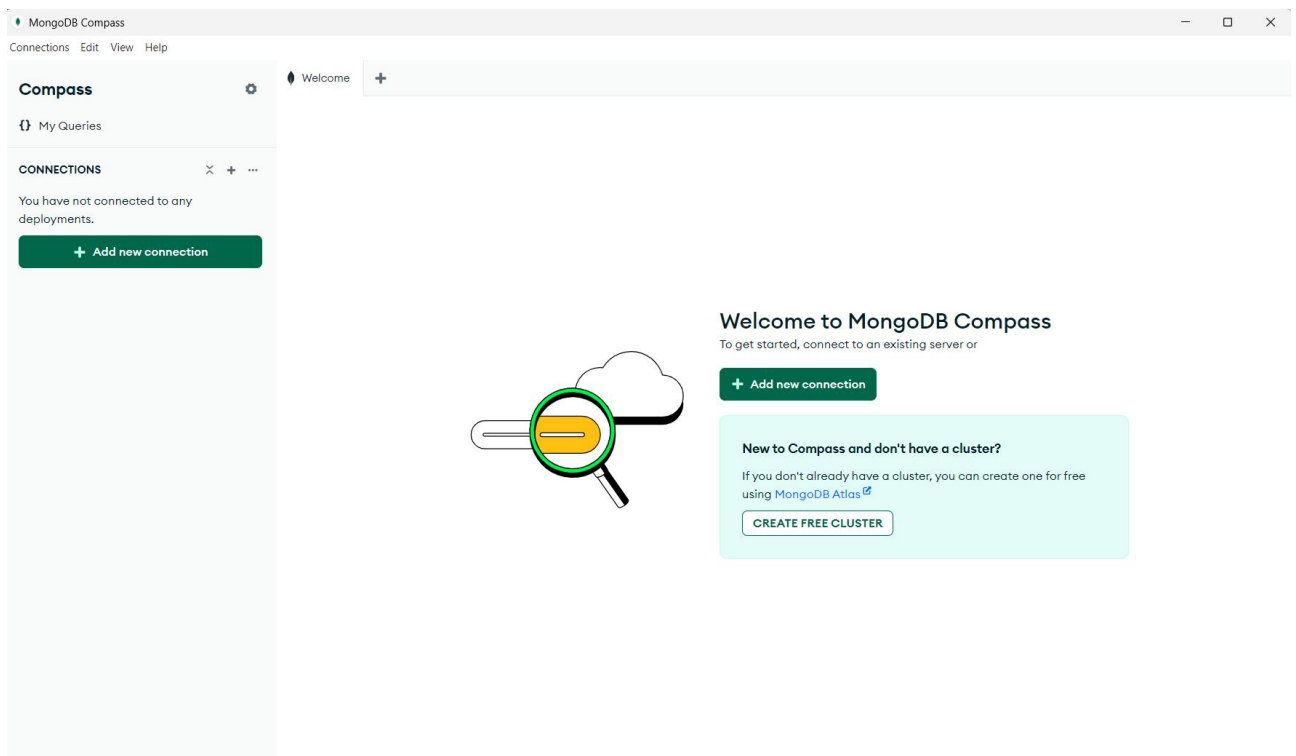
Bước 0: Cài đặt MongoDB:	1
Bước 1: Cài đặt thư viện mongo, định nghĩa hàm trợ giúp kết nối và lớp dịch vụ truy xuất cơ sở dữ liệu (CSDL)	2
Bước 2: Cài đặt các handler	4

Bước 0: Cài đặt MongoDB:

Version	7.0.14 (current)	▼
Platform	Windows x64	▼
Package	msi	▼

Download 

 Copy link



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

- Cài đặt mongodb vào dự án:

```
PS C:\Users\LE TUAN DAT\Documents\TaiLieuHocTap\CT449--PTUDW-Lab2> npm install mongodb
added 197 packages, and audited 198 packages in 17s

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

found 0 vulnerabilities
PS C:\Users\LE TUAN DAT\Documents\TaiLieuHocTap\CT449--PTUDW-Lab2> |
```

- Trong thư mục *app/config*, hiệu chỉnh tập tin *index.js*:

```
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
11
```

- Định nghĩa lớp trợ giúp kết nối đến MongoDB: *app/utils/mongodb.util.js*

```
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
```

- Thực hiện kết nối đến CSDL MongoDB khi chạy server, thay đổi toàn bộ nội dung tập tin *server.js* bằng đoạn mã sau:

```

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()

```

- Định nghĩa lớp dịch vụ *ContactService* (trong tập tin `app/services/contact.service.js`) chứa các API của thư mục `mongodb` để thực hiện thao tác với CSDL Mongo:

```

app > services > JS contact.service.js > [?] <unknown>
1  const {ObjectId} = require("mongodb")
2
3  class ContactService {
4      constructor(client){
5          this.Contact = client.db().collection("contacts")
6      }
7
8      //
9  }
10
11  module.exports = ContactService

```

Bước 2: Cài đặt các handler

- Hiệu chỉnh tập tin *app/controllers/contact.controller.js*:

```
const ApiError = require("../api-error")
const ContactService = require("../services/contact.service")
const MongoDB = require("../utils/mongodb.util")

exports.create = async (req, res, next) => {
  if (!req.body?.name) {
    return next(new ApiError(400, "Name can not be empty"))
  }

  try {
    const contactService = new ContactService(MongoDB.client)
    const document = await contactService.create(req.body)
    return res.send(document)
  } catch (error) {
    return next(
      new ApiError(500, "An error occurred while creating the contact")
    )
  }
}
```

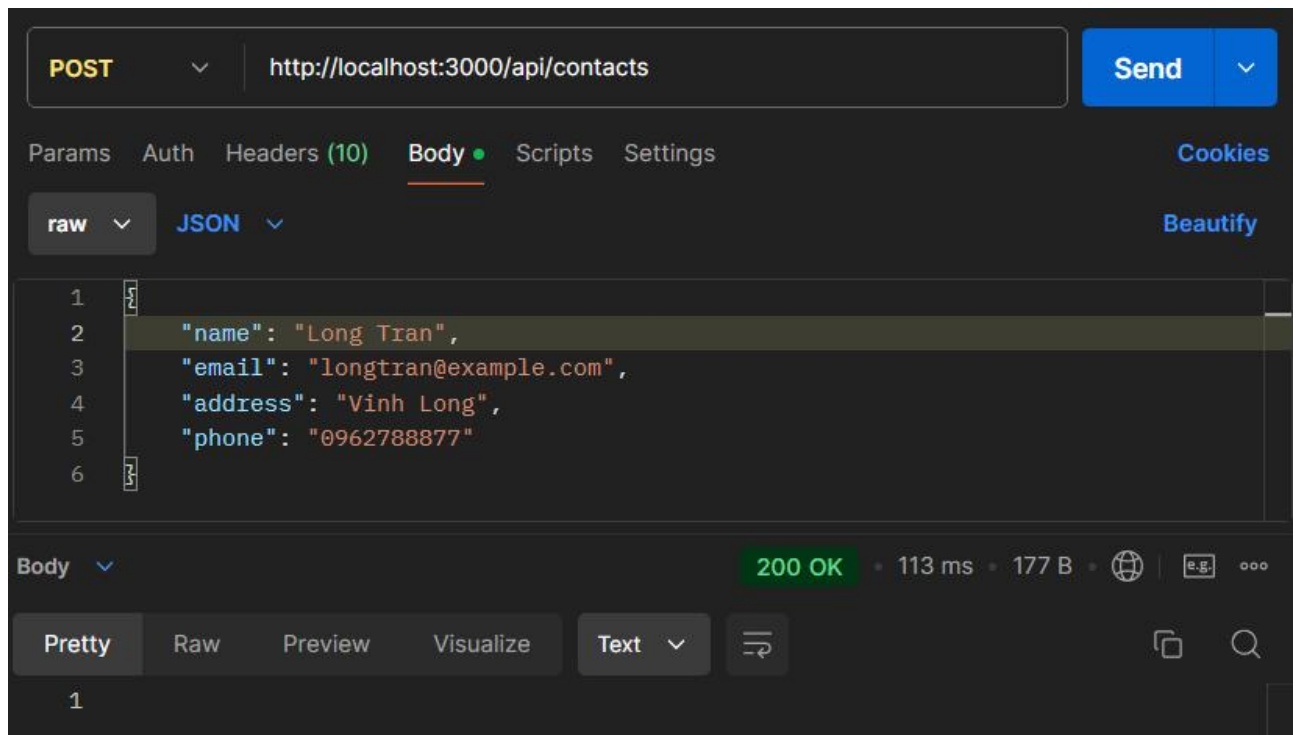
- Định nghĩa phương thức *create()* trong lớp *ContactService* ở tập tin *app/services/contact.service.js*:


```











app > services > JS contact.service.js > ContactService > extractContactData
1  const {ObjectId} = require("mongodb")
2
3  class ContactService {
4      constructor(client){
5          this.contact = client.db().collection("contacts")
6      }
7
8      extractContactData(payload) {
9          const contact = {
10             name: payload.name,
11             email: payload.email,
12             address: payload.address,
13             phone: payload.phone,
14             favorite: payload.favorite,
15         }
16
17         Object.keys(contact).forEach(
18             (key) => contact[key] === undefined && delete contact[key]
19         )
20
21         return contact
22     }
23
24     async create(payload) {
25         const contact = this.extractContactData(payload)
26         const result = await this.contact.findOneAndUpdate(
27             contact,
28             { $set: {favorite: contact.favorite === true} },
29             { returnDocument: "after", upsert: true}
30         )
31         return result.value
32     }
33 }
34
35 module.exports = ContactService

```

- Kiểm tra bằng Postman:



- Để gửi dữ liệu JSON về server với Postman, đặt “Content-Type: application/json” và đặt dữ liệu JSON trong phần Body của yêu cầu:

Params	Auth	Headers (10)	Body ●	Scripts	Settings
Headers  Hide auto-generated headers					
	Key		Value		
<input checked="" type="checkbox"/>	X-API-Key			
<input checked="" type="checkbox"/>	Postman-Token		<calculated when request is s...		
<input checked="" type="checkbox"/>	Content-Type		application/json		
<input checked="" type="checkbox"/>	Content-Length		<calculated when request is s...		
<input checked="" type="checkbox"/>	Host		<calculated when request is s...		
<input checked="" type="checkbox"/>	User-Agent		PostmanRuntime/7.42.0		
<input checked="" type="checkbox"/>	Accept		*/*		
<input checked="" type="checkbox"/>	Accept-Encoding		gzip, deflate, br		
<input checked="" type="checkbox"/>	Connection		keep-alive		
<input checked="" type="checkbox"/>	Content-Type		application/json		

- Đặt dữ liệu JSON trong phần Body:

raw  **JSON** 

```

1  {
2    "name": "Long Tran",
3    "email": "longtran@example.com",
4    "address": "Vinh Long",
5    "phone": "0962788877"
6  }
```

- Cài đặt handler `findAll` trong tập tin `app/controllers/contact.controller.js`:

```
exports.findAll = async (req, res, next) => {
  let documents = []

  try {
    const contactService = new ContactService(MongoDB.client)
    const { name } = req.query
    if (name) {
      documents = await contactService.findByName(name).toArray()
    } else {
      documents = await contactService.find({})
    }
  } catch (error) {
    return next(
      new ApiError(500, "An error occurred while retrieving contacts")
    )
  }

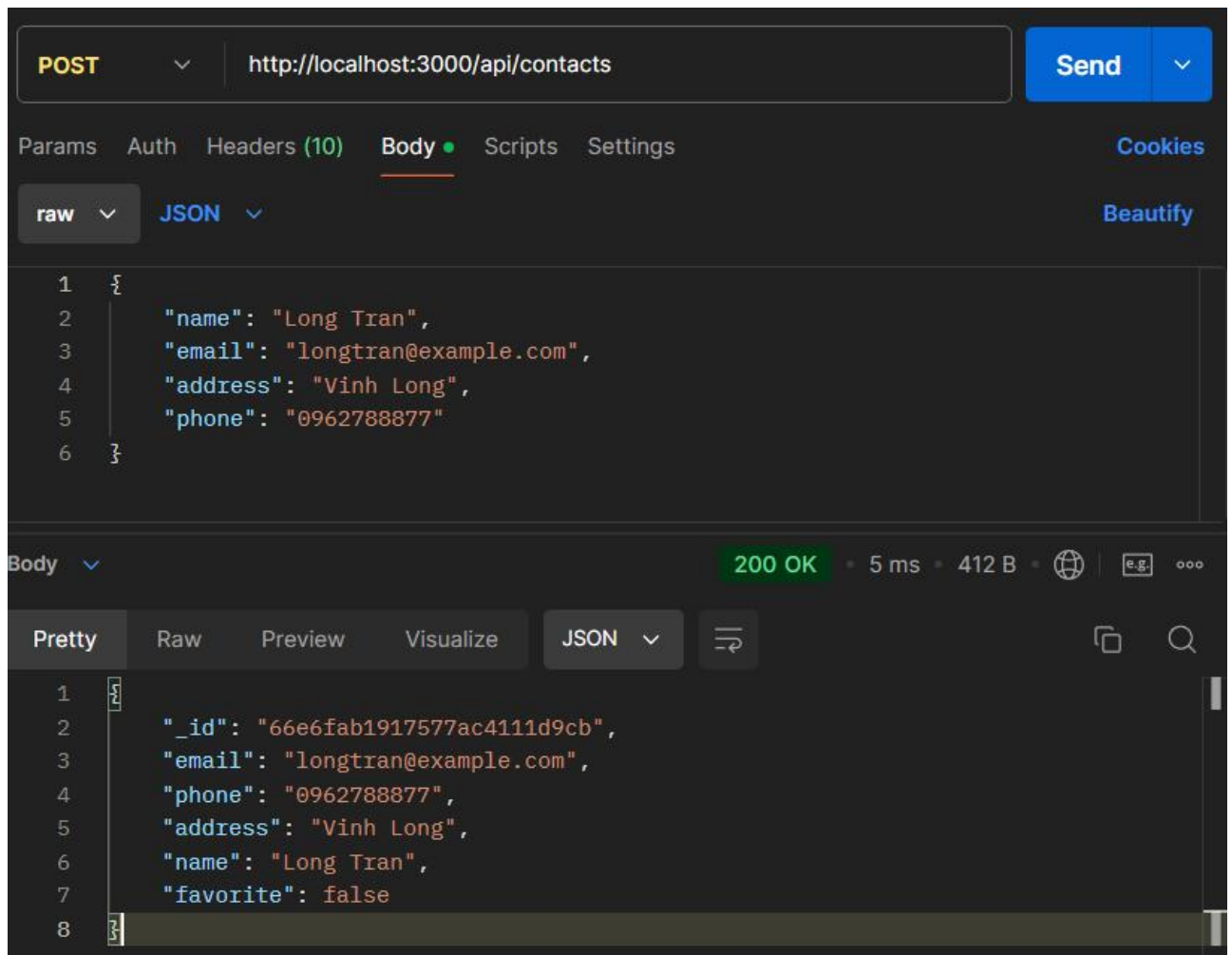
  return res.send(documents)
}
```

- Định nghĩa phương thức `find()` và `findByName()` trong tập tin `app/services/contact.service.js`:

```
async find(filter) {
  const cursor = await this.contact.find(filter)
  return await cursor.toArray()
}

async findByName(name) {
  return await this.find({
    name: { $regex: new RegExp(new RegExp(name)), $option: "i" }
  })
}
```

- Kiểm tra bằng Postman:



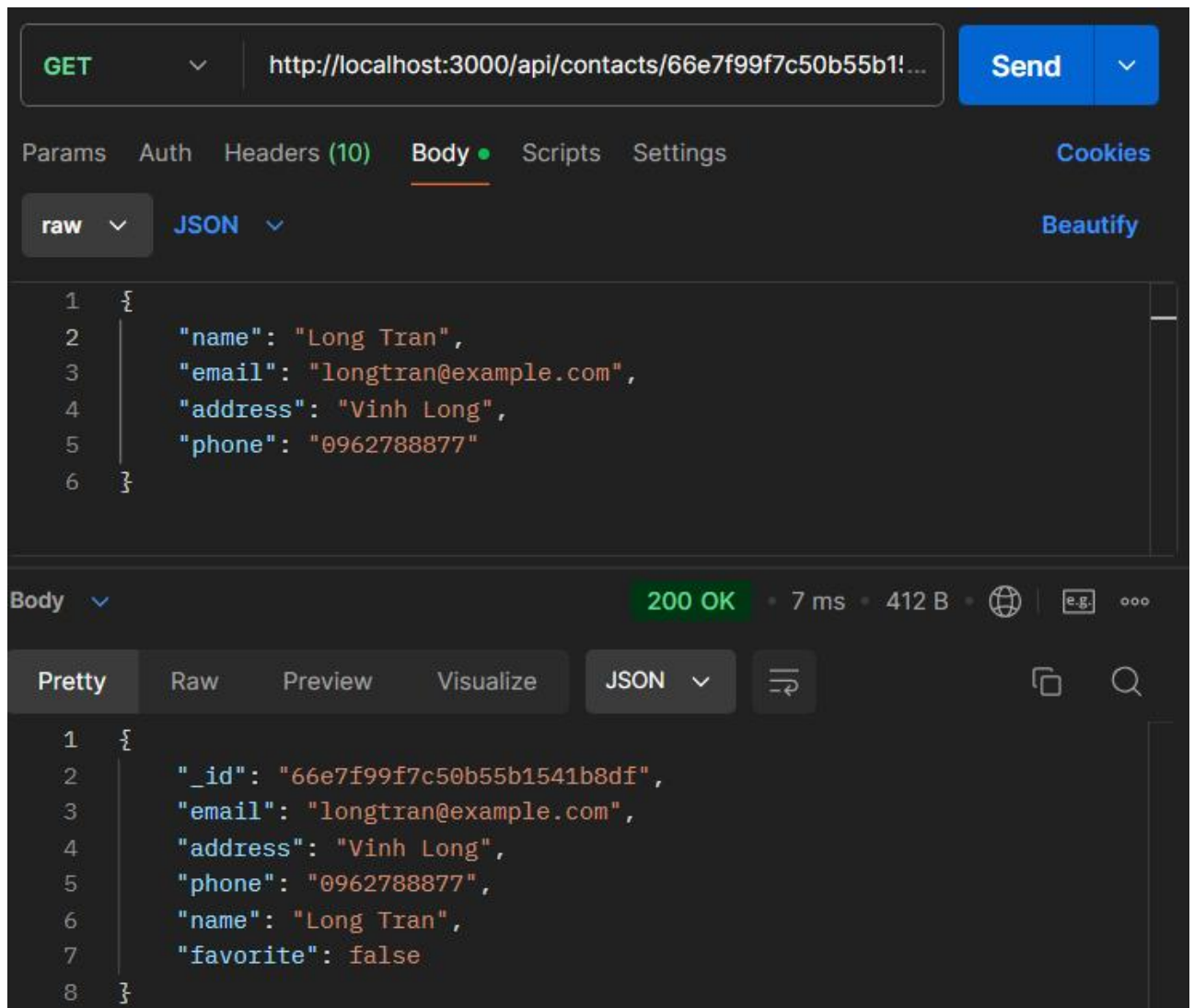
- Cài đặt handler `findOne` trong tập tin `app/controllers/contact.controller.js` như sau:

```
exports.findOne = async (req, res, next) => {
  try {
    const contactService = new ContactService(MongoDB.client)
    const document = await contactService.findById(req.params.id)
    if (!document) {
      return next(new ApiError(404, "Contact not found"))
    }
    return res.send(document)
  } catch (error) {
    return next(
      new ApiError(
        500,
        `Error retrieving contact with id=${req.params.id}`
      )
    )
  }
}
```

- Định nghĩa phương thức *findById()* trong tập tin *app/services/contact.service.js*:

```
async findById(id) {  
  return await this.contact.findOne({  
    _id: ObjectId.isValid(id) ? new ObjectId(id) : null  
  });  
}
```

- Kiểm tra trong Postman:



- Cài đặt handler *update* trong tập tin *app/controllers/contact.controller.js*:


```

exports.update = async (req, res, next) => {
  if (Object.keys(req.body).length === 0) {
    return next(new ApiError(400, 'Data to update can not be empty'))
  }
  try {
    const contactService = new ContactService(MongoDB.client)
    const document = await contactService.update(req.params.id, req.body)
    if (!document) {
      return next(new ApiError(404, "Contact not found"))
    }
    return res.send({message: "Contact was updated successfully"})
  } catch (error) {
    return next(
      new ApiError(500, `Error updating contact with id=${req.params.id}`)
    )
  }
}

```

- Định nghĩa phương thức *update()* trong tập tin *app/services/contact.service.js*:

```

async update(id, payload) {
  const filter = {
    _id: ObjectId.isValid(id) ? new ObjectId(id) : null
  }
  const update = this.extractContactData(payload)
  const result = await this.contact.findOneAndUpdate(
    filter,
    { $set: update },
    { returnDocument: "after" }
  )
  return result.value = 1
}

```

- Kiểm tra Postman:

PUT ▼ http://localhost:3000/api/contacts/66e7f99f7c50b55b1!... Send ▼

Params Auth Headers (10) **Body** ● Scripts Settings Cookies

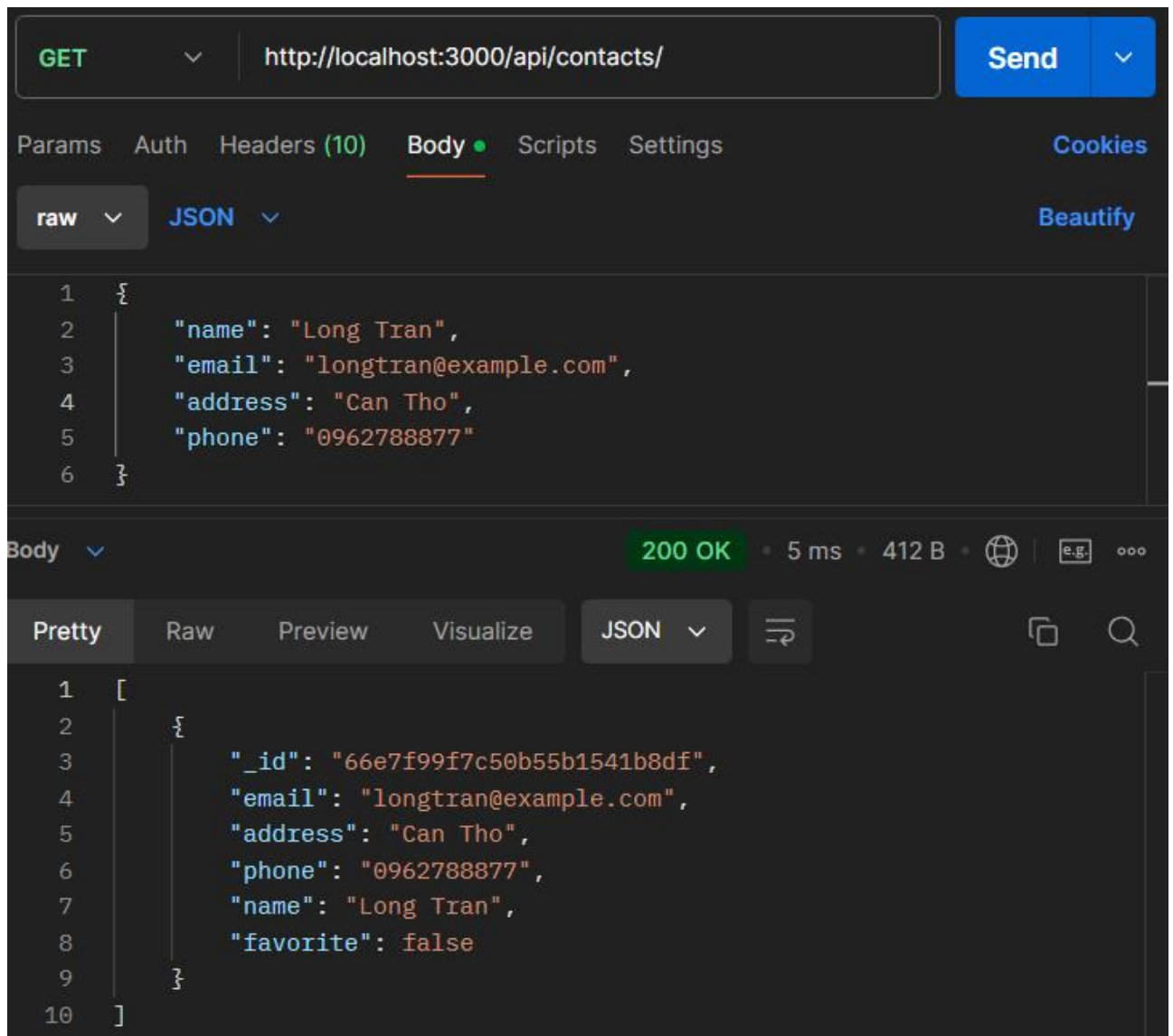
raw ▼ JSON ▼ Beautify

```
1 {  
2   "name": "Long Tran",  
3   "email": "longtran@example.com",  
4   "address": "Can Tho",  
5   "phone": "0962788877"  
6 }
```

Body ▼ 200 OK • 5 ms • 313 B • e.g. ...

Pretty Raw Preview Visualize JSON ▼ ↺ 📄 🔍

```
1 {  
2   "message": "Contact was updated successfully"  
3 }
```



- Cài đặt handler *delete* trong tập tin *app/controllers/contact.controller.js*:

```
exports.delete = async (req, res, next) =>{
  try {
    const contactService = new ContactService(MongoDB.client)
    const document = await contactService.delete(req.params.id)
    if(!document){
      return next(new ApiError(404, "Contact not found"))
    }
    return res.send({message: "Contact was deleted successfully"})
  } catch (error) {
    return next(new ApiError(500, `Could not delete contact with id=${req.params.id}`))
  }
}
```

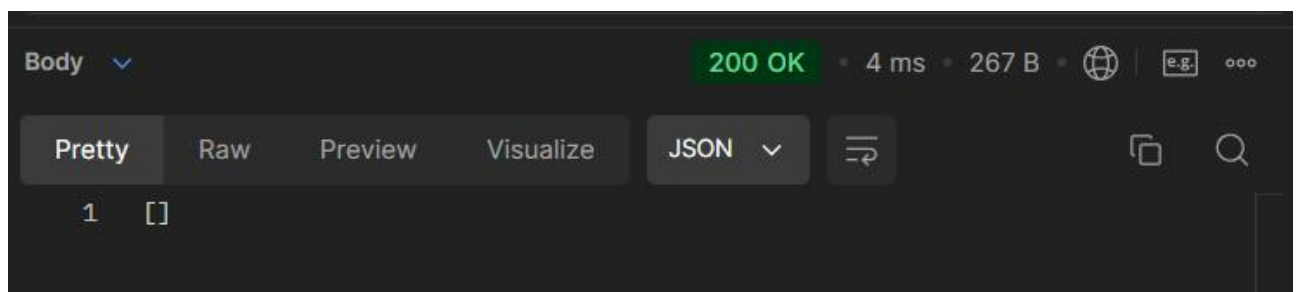
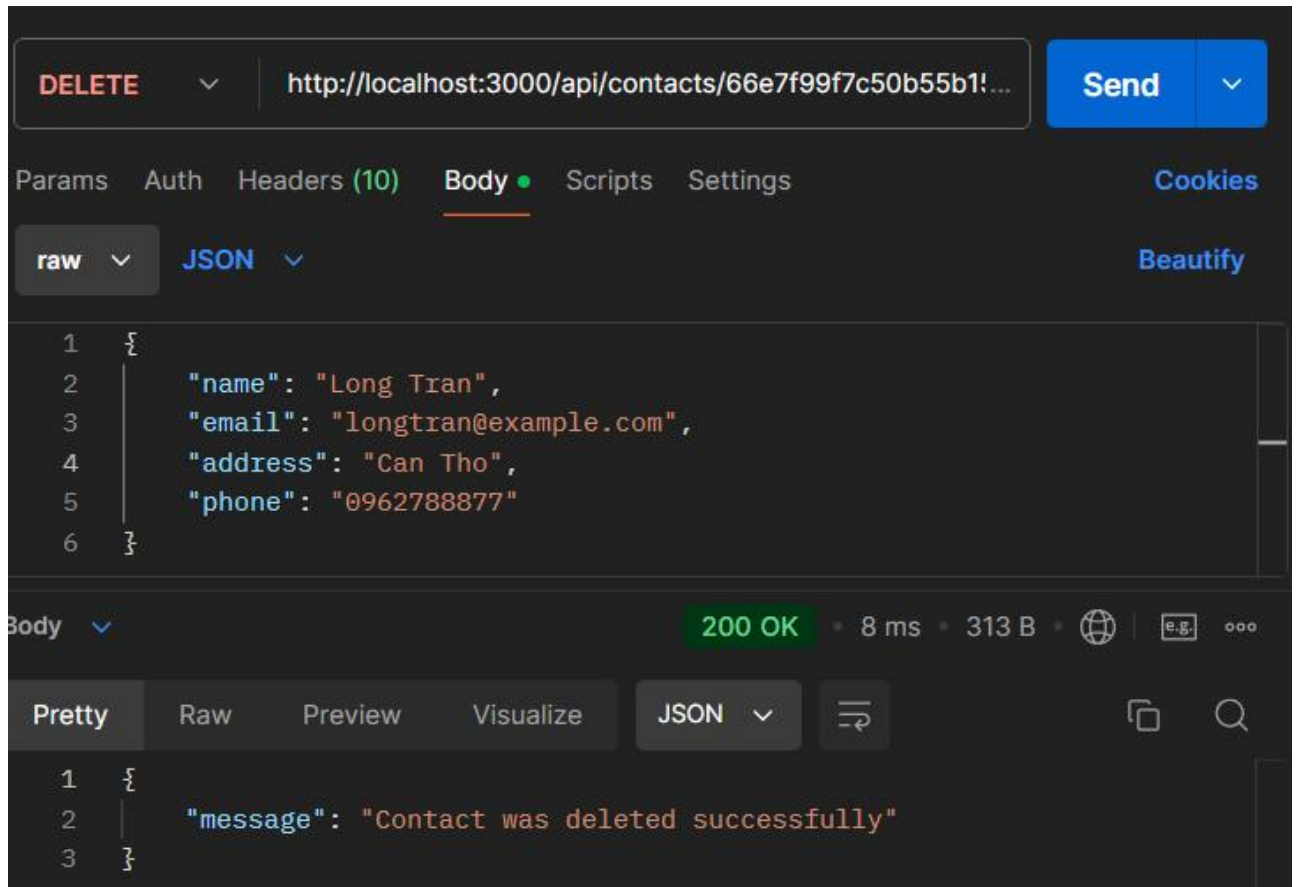
- Định nghĩa phương thức *delete(id)* trong tập tin *app/services/contact.service.js*:


```

async delete(id) {
  const result = await this.contact.findOneAndDelete({
    _id: ObjectId.isValid(id) ? new ObjectId(id) : null
  })
  return result
}

```

- Kiểm tra Postman:



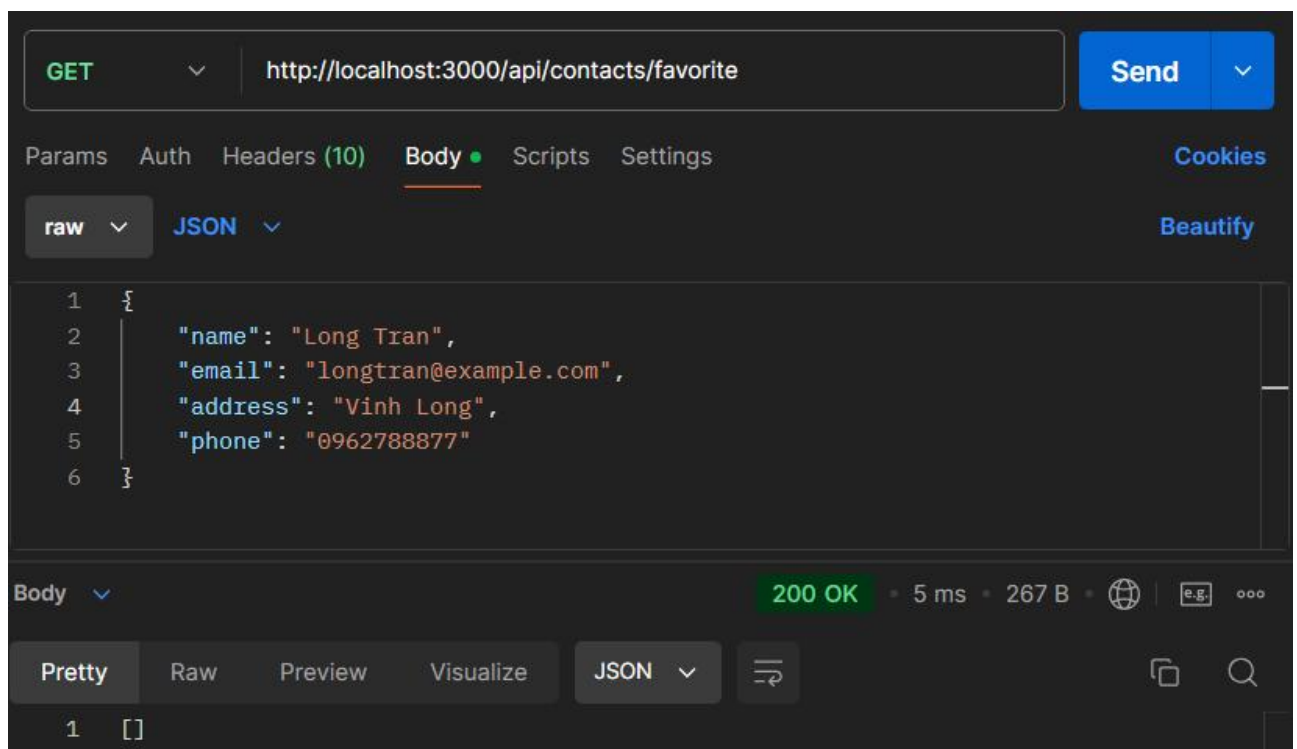
- Cài đặt handler *findAllFavorite* trong tập tin *app/controllers/contact.controller.js*:

```
exports.findAllFavorite = async (_req, res, next) => {
  try {
    const contactService = new ContactService(MongoDB.client)
    const documents = await contactService.findFavorite()
    return res.send(documents)
  } catch (error) {
    return next(
      new ApiError(
        500, "An error occurred while retrieving favorite contacts"
      )
    )
  }
}
```

- Định nghĩa phương thức *findFavorite()* trong tập tin *app/services/contact.service.js*:

```
async findFavorite() {
  return await this.find({favorite: true})
}
```

- Kiểm tra Postman:



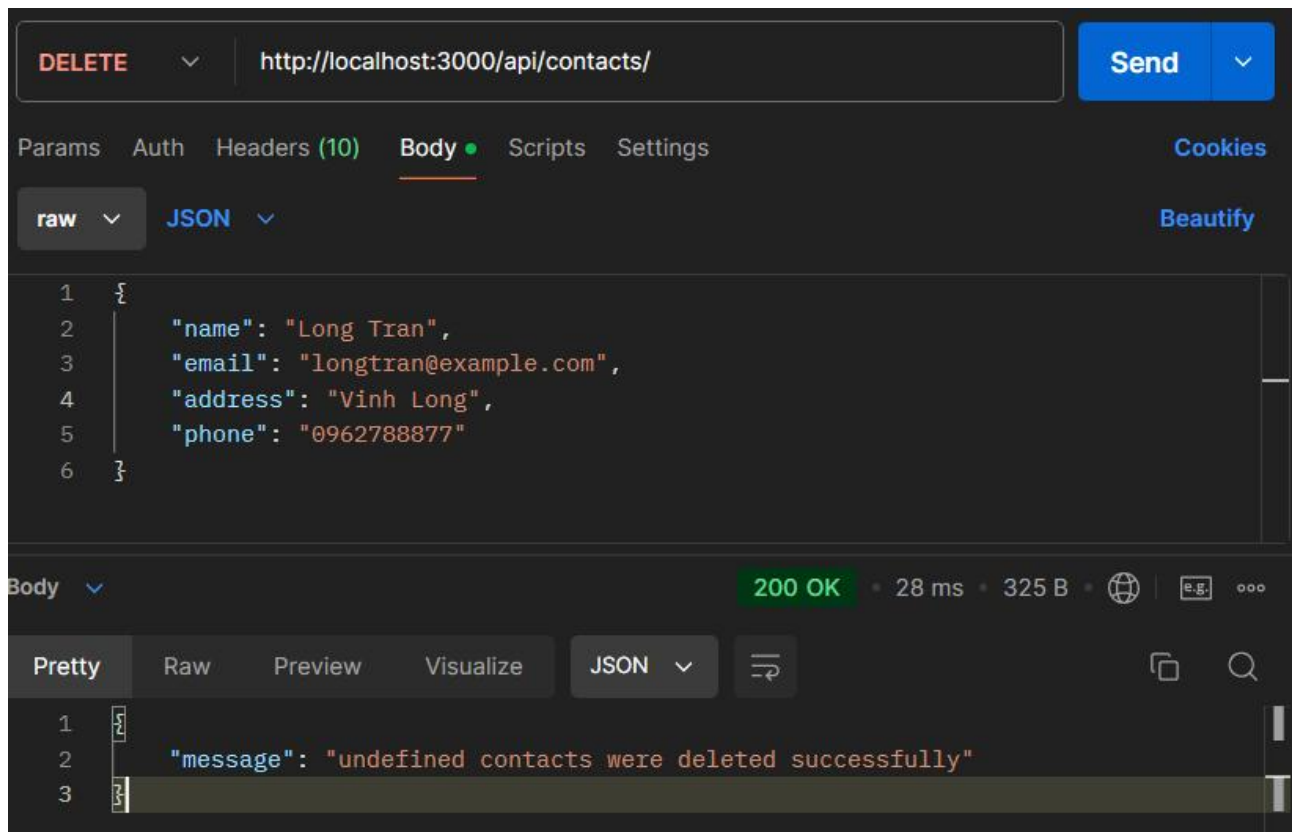
- Cài đặt handler *deleteAll*:

```
exports.deleteAll = async (req, res, next) => {
  try {
    const contactService = new ContactService(MongoDB.client)
    const deleteCount = await contactService.deleteAll()
    return res.send({
      message: `${deleteCount} contacts were deleted successfully`
    })
  } catch (error) {
    return next(
      new ApiError(500, "An error occurred while retrieving favorite contacts")
    )
  }
}
```

- Định nghĩa phương thức *deleteAll()*:

```
async deleteAll() {
  const result = await this.contact.deleteMany({})
  return result.deleteCount
}
```

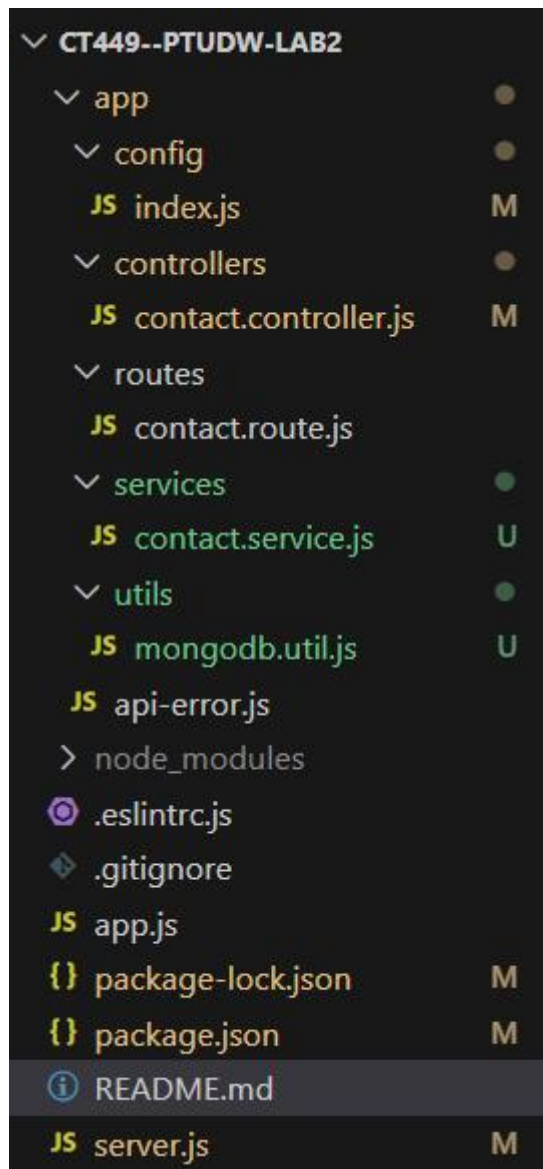
- Kiểm tra Postman:



- Lưu thay đổi và đẩy lên GitHub:

```
PS C:\Users\LE TUAN DAT\Documents\TailieuHocTap\CT449--PTUDW-Lab2> git add -u
PS C:\Users\LE TUAN DAT\Documents\TailieuHocTap\CT449--PTUDW-Lab2> git add app/utils app/services
PS C:\Users\LE TUAN DAT\Documents\TailieuHocTap\CT449--PTUDW-Lab2> git commit -m "Cai dat cac handler truy xuat CSDL"
[main 467b6a4] Cai dat cac handler truy xuat CSDL
 7 files changed, 358 insertions(+), 23 deletions(-)
 create mode 100644 app/services/contact.service.js
 create mode 100644 app/utils/mongodb.util.js
PS C:\Users\LE TUAN DAT\Documents\TailieuHocTap\CT449--PTUDW-Lab2> git push origin main
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 8 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (14/14), 4.49 KiB | 1.50 MiB/s, done.
Total 14 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/letuandatt/CT449--PTUDW-Lab2.git
 620470a..467b6a4 main -> main
PS C:\Users\LE TUAN DAT\Documents\TailieuHocTap\CT449--PTUDW-Lab2> |
```

- Cây thư mục hiện tại:



- Đường dẫn đến GitHub:

https://github.com/letuandatt/B2113328_LeTuanDat_BACKEND_2

---HẾT---