

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



Phát triển ứng dụng Web

Báo cáo thực hành

**SINH VIÊN THỰC HIỆN:
Trần Tri Min; MSSV: B2207541**

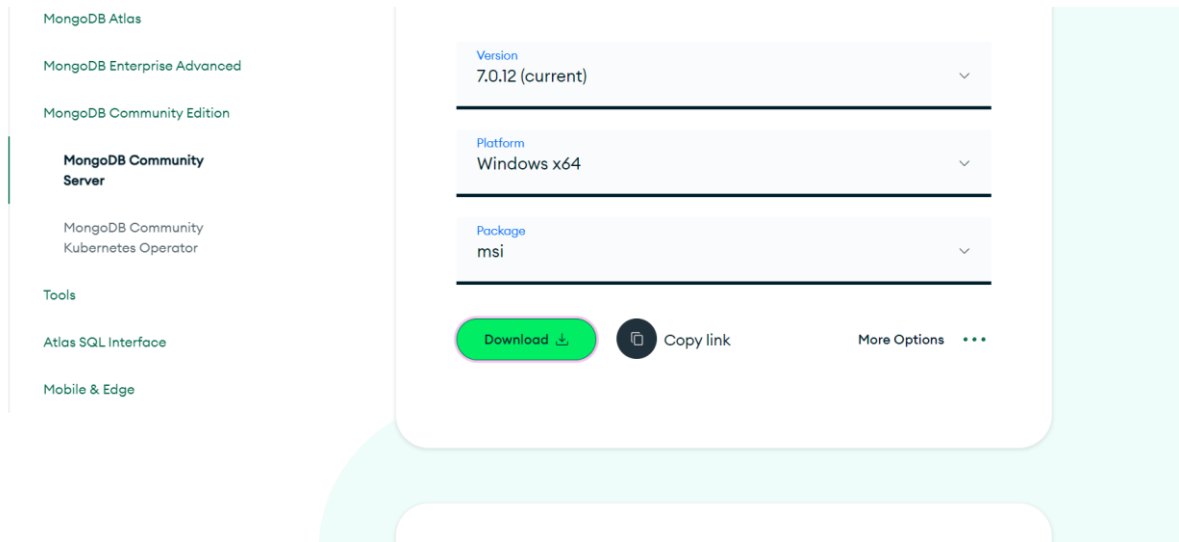
Tháng 12/2020

MỤC LỤC

MỤC LỤC.....	ii
CHƯƠNG I: ỨNG DỤNG CONTACTBOOK – BACKEND – PHẦN 2	3
1. Bước 0: Cài đặt MongoDB.....	3
2. Bước 1: Cài đặt thư viện MongoDB, định nghĩa hàm trợ giúp kết nối và lớp dịch vụ truy xuất cơ sở dữ liệu (CSDL)	3
3. Bước 2: Cài đặt các handler	5

CHƯƠNG I: Ứng dụng Contactbook – Backend – Phần 2

1. Bước 0: Cài đặt MongoDB



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

Npm install mongodb

```
● PS E:\SCHOOL WORK\PTUD Web\contactbook-backend> npm install mongodb
added 11 packages, and audited 213 packages in 27s

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

found 0 vulnerabilities
○ PS E:\SCHOOL WORK\PTUD Web\contactbook-backend> 
```

Hiệu chỉnh index.js trong app/config

```
✓ CONTACTBOOK-BACKEND
  > .idea
  > app
    > config
      JS index.js
    > routes
      JS contact.route.js
    > api-error.js
    > controllers
      JS contact.controller.js
    > node_modules
    > .eslintrc.js
    > .gitignore
    JS app.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://localhost:27017/contactbook',
7    }
8  };
9
10 module.exports = config;
11
```

Định nghĩa lớp trợ giúp kết nối

```
✓ CONTACTBOOK-BACKEND
  > .idea
  > app
    > config
      JS index.js M
    > routes
      JS contact.route.js
    > utils
      JS mongodb.util.js U
    > api-error.js
    > controllers
      JS contact.controller.js
    > node_modules
    > .eslintrc.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;
12
```

Thay toàn bộ tệp tin server.js

```
✓ CONTACTBOOK-BACKEND
  > .idea
  > app
    > config
      JS index.js M
    > routes
      JS contact.route.js
    > utils
      JS mongodb.util.js U
    > api-error.js
    > controllers
      JS contact.controller.js
    > node_modules
    > .eslintrc.js
    > .gitignore
    JS app.js
    {} package-lock.json M
    {} package.json M
    JS server.js M

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
12     app.listen(PORT, () => {
13       console.log(`Server is running on port ${PORT}`);
14     });
15   } catch (error) {
16     console.log("Cannot connect to the database!", error);
17     process.exit();
18   }
19 }
20
21 startServer();
22
```

Định nghĩa contact service:

```

CONTACTBOOK-BACKEND
├── .idea
├── app
│   ├── config
│   │   └── index.js
│   ├── routes
│   │   └── contact.route.js
│   └── services
│       └── contact.service.js
└── utils

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  module.exports = ContactService;
10

```

3. Bước 2: Cài đặt các handler

Cài đặt handler create

Hiệu chỉnh tập tin app/controllers/contact.controller.js

```

controllers
├── contact.controller.js
└── node_modules
    ├── .eslintrc.js
    ├── .gitignore
    ├── app.js
    ├── package-lock.json
    ├── package.json
    └── server.js

32
33 exports.create = async (req, res, next) => {
34     if (!req.body?.name) {
35         return next(new ApiError(400, "Name cannot be empty"));
36     }
37
38     try {
39         const contactService = new ContactService(MongoDB.client);
40         const document = await contactService.create(req.body);
41         return res.send(document);
42     } catch (error) {
43         return next(
44             new ApiError(500, "An error occurred while creating a new contact")
45         );
46     }
47 }
48

```

Tạo phương thức create() trong lớp ContactService

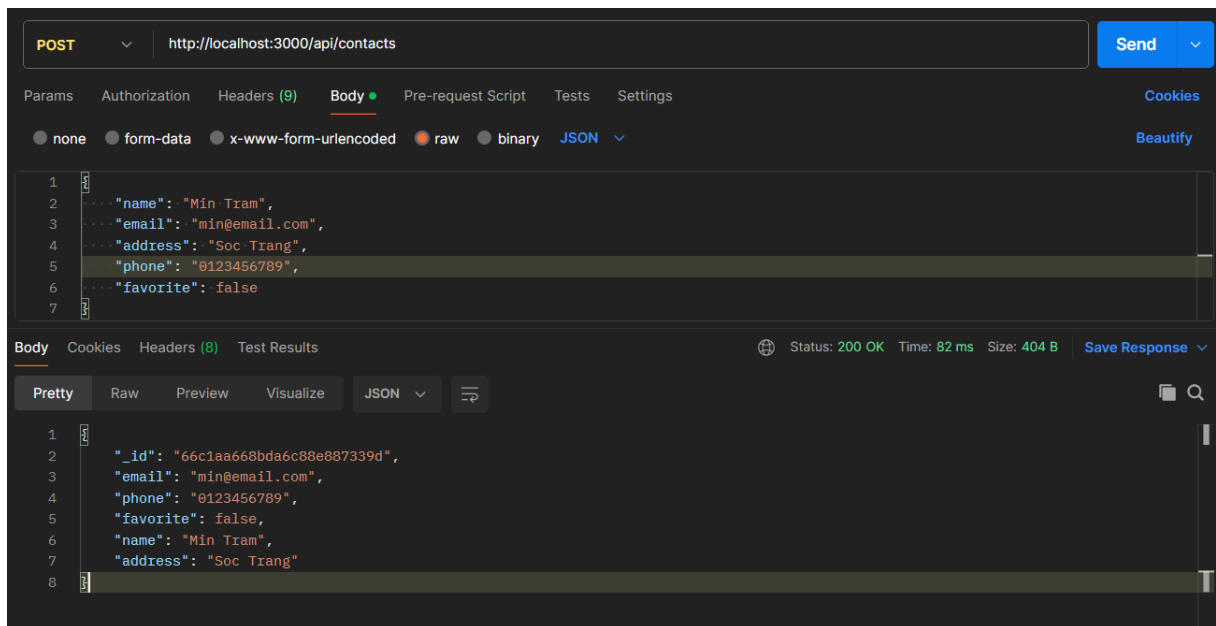
```

CONTACTBOOK-BACKEND
├── .idea
├── app
│   ├── config
│   │   └── index.js
│   ├── routes
│   │   └── contact.route.js
│   └── services
│       └── contact.service.js
└── utils
    ├── mongodb.utils.js
    ├── api-error.js
    ├── controllers
    │   └── contact.controller.js
    ├── node_modules
    │   ├── .eslintrc.js
    │   ├── .gitignore
    │   ├── app.js
    │   ├── package-lock.json
    │   ├── package.json
    │   └── server.js
    └── mongodb.utils.js

app > services > JS contact.service.js > ContactService > create
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         return contact;
21     }
22
23     async create(payload) {
24         const contact = this.extractContactData(payload);
25         const result = await this.Contact.findOneAndUpdate(
26             contact,
27             { $set: { favorite: contact.favorite === true } },
28             { returnDocument: "after", upsert: true }
29         );
30         return result;
31     }
32 }
33
34 module.exports = ContactService;
35

```

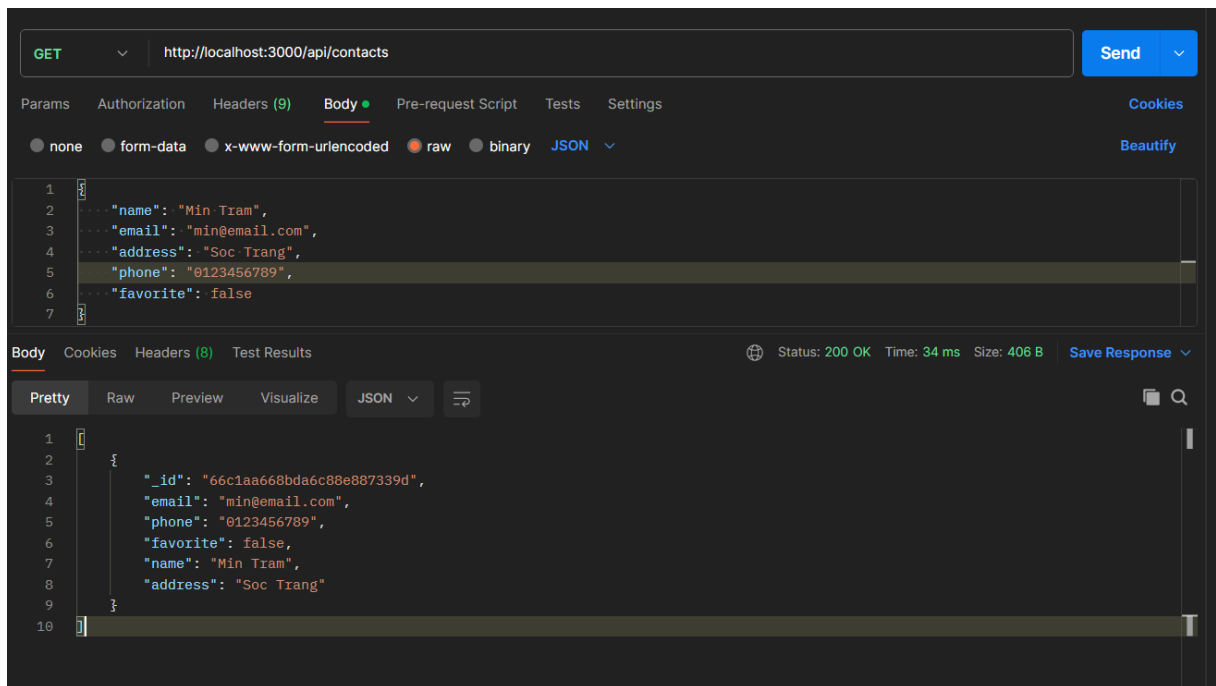
Kiểm tra với Postman:



Cài đặt handler findAll



Kiểm tra với Postman:

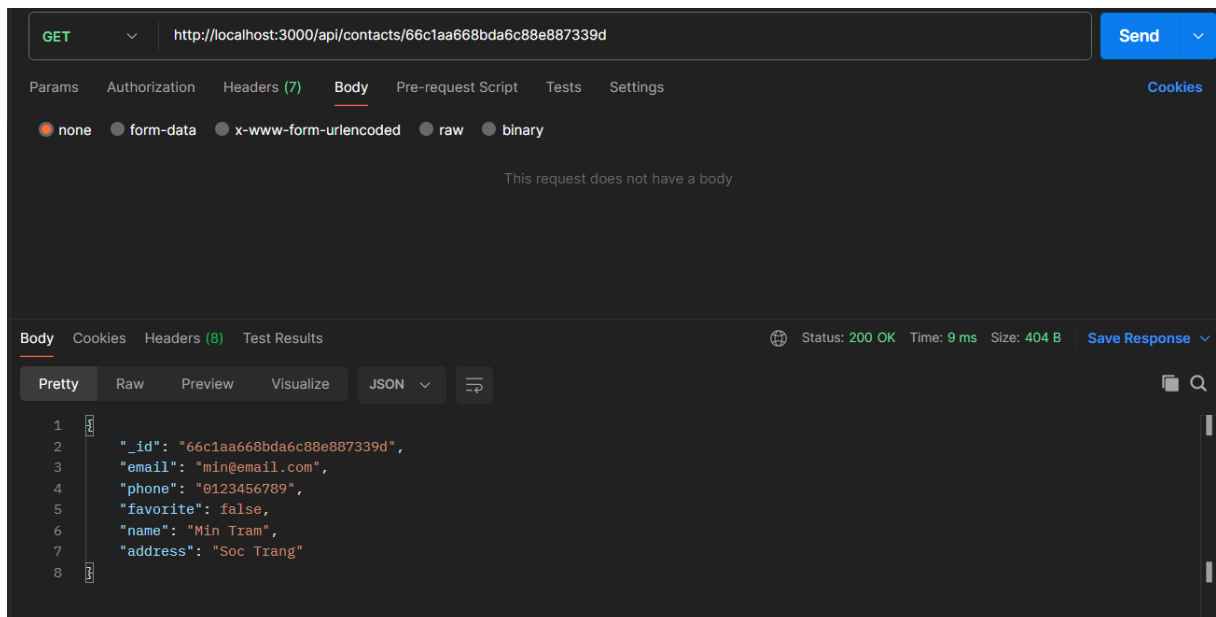


Cài đặt handler findOne:

```
29 exports.findOne = async (req, res, next) => {
30   try {
31     const contactService = new ContactService(MongoDB.client);
32     const document = await contactService.findById(req.params.id);
33     if (!document) {
34       return next(new ApiError(404, "Contact not found"));
35     }
36     return res.send(document);
37   } catch (error) {
38     return next(
39       new ApiError(500, `Error retrieving contact with id=${req.params.id}`)
40     );
41   }
42 };
43
```

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

Kiểm tra với Postman:

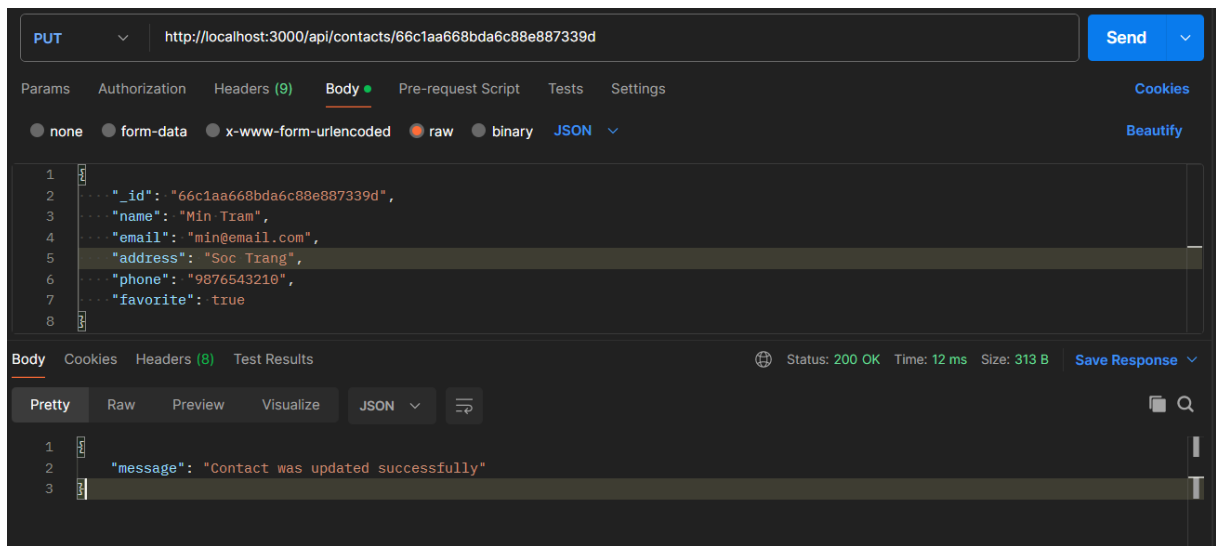


Cài đặt handler update:

```
4 exports.update = async (req, res) => {
5   if (Object.keys(req.body).length === 0) {
6     return next(new ApiError(400, "Data to update cannot be empty"));
7   }
8
9   try {
10    const contactService = new ContactService(MongoDB.client);
11    const document = await contactService.update(req.params.id, req.body);
12    if (!document) {
13      return next(new ApiError(404, "Contact not found"));
14    }
15
16    return res.send({ message: "Contact was updated successfully" });
17  } catch (error) {
18    return next(
19      new ApiError(500, `Error updating contact with id=${req.params.id}`)
20    );
21  }
22 };
23
```

```
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;
}
```


Kiểm tra với Postman:

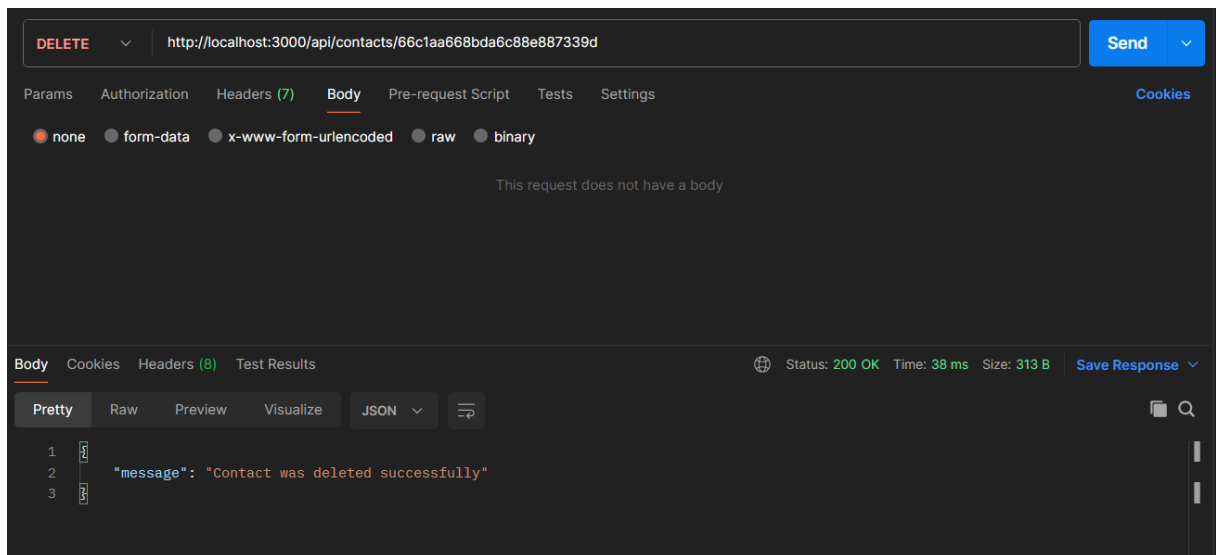


Cài đặt handler delete:

```
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}`)
    );
  }
};
```

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

Kiểm tra với Postman:

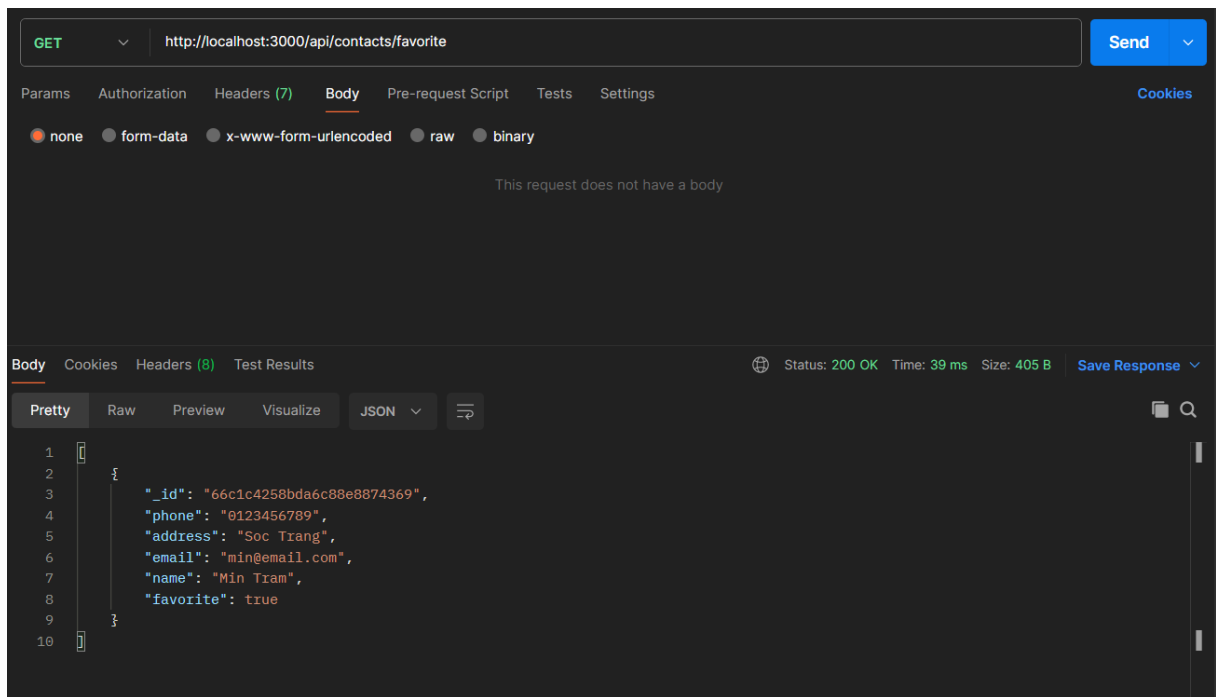


Cài đặt handler findAllFavorite

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

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

Kiểm tra với Postman:

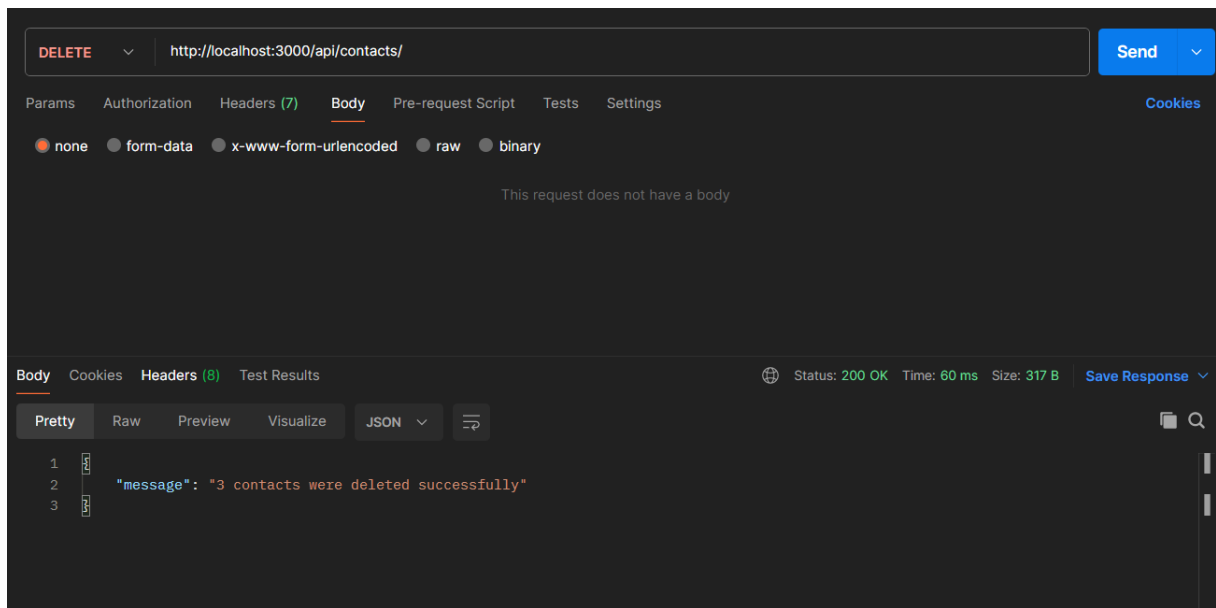


Cài đặt handler deleteAll:

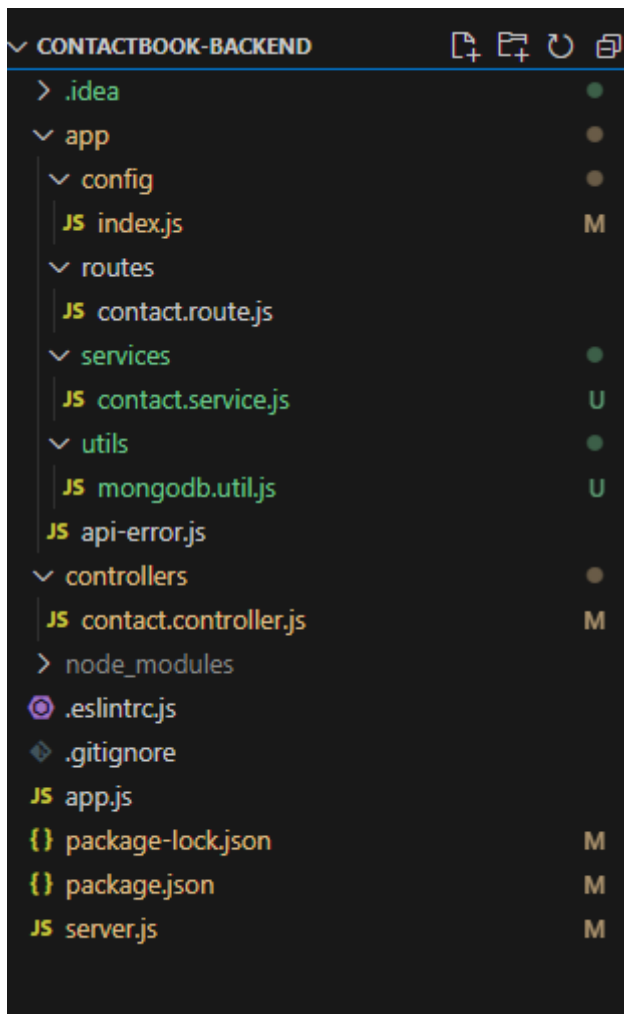
```
exports.deleteAll = async (_req, res, next) => {
  try {
    const contactService = new ContactService(MongoDB.client);
    const deletedCount = await contactService.deleteAll();
    return res.send({
      message: `${deletedCount} contacts were deleted successfully`,
    });
  } catch (error) {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

```
async deleteAll() {
  const result = await this.Contact.deleteMany({});
  return result.deletedCount;
}
```

Kiểm tra với Postman:



Cấu trúc thư mục dự án:



```
PS E:\SCHOOL WORK\PTUD Web\contactbook-backend> git add -u
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:\SCHOOL WORK\PTUD Web\contactbook-backend> git add app/Utils app/services
PS E:\SCHOOL WORK\PTUD Web\contactbook-backend> git commit -m "Cai dat cac handler truy xuat CSDL"
[master df1221d] Cai dat cac handler truy xuat CSDL
7 files changed, 351 insertions(+), 19 deletions(-)
create mode 100644 app/services/contact.service.js
create mode 100644 app/Utils/mongodb.util.js
PS E:\SCHOOL WORK\PTUD Web\contactbook-backend> git push origin master
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), 4.44 KiB | 649.00 KiB/s, done.
Total 14 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/phaspez/contactbook-backend.git
9c13555..df1221d master -> master
```

Link mã nguồn: <https://github.com/phaspez/contactbook-backend>