

# Testing Document

## Introduction of Testing Tool:

- During the system development process, API testing plays an important role in ensuring that all functions work as expected. The team used **Thunder Client**, an API testing tool integrated into Visual Studio Code, to send HTTP requests (GET, POST, DELETE, etc.) and verify the system responses.
- Thunder Client provides a simple and effective way to test APIs by displaying status codes and response data clearly, helping the team evaluate the correctness of the implemented features.

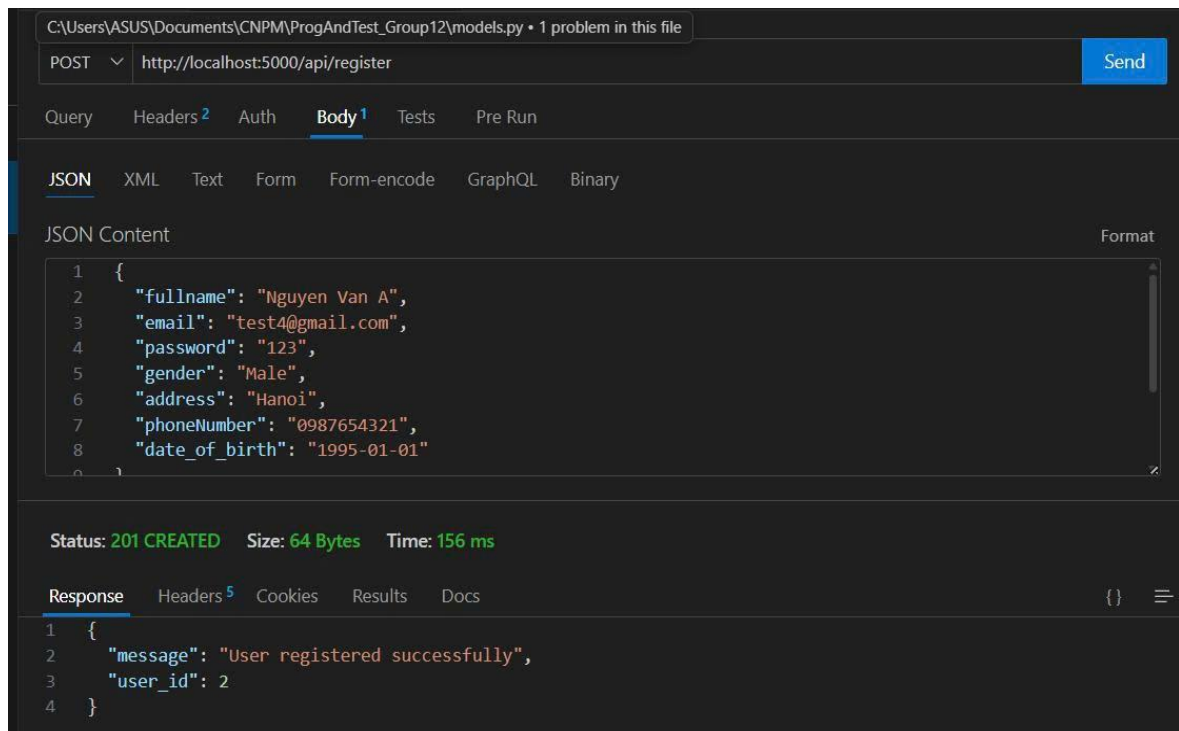
## Test case:

Test Case ID	Function	Test Steps	Input Data	Expected Result
TC01	Register	1. Go to Register page 2. Enter valid information 3. Click Register	Full Name: Nguyen Van A Email: test@gmail.com Password: 123	User is registered successfully and a default AddressBook is created
TC02	Register	1. Go to Register page 2. Enter an email that already exists 3. Click Register	Full Name: Nguyen Van A Email: test@gmail.com Password: 123	Error message " <b>Email already exists</b> " is displayed
TC03	Login	1. Go to Login page 2. Enter valid username and password 3. Click Login	Username: student01 Password: 123456	User logs in successfully
TC04	Login	1. Go to Login page 2. Enter valid username and invalid password 3. Click Login	Username: student01 Password: wrongpass	Error message " <b>Invalid username or password</b> " is displayed
TC05	Add Contact	1. Login to the system 2. Go to Add Contact page 3. Enter valid contact information 4. Click Submit	Name: Nguyễn Văn A Email: vana@gmail.com Phone: 0912345678 Group IDs: [1,2]	New contact is added successfully and appears in the contact list
TC06	Edit Contact	1. Login to the system 2. Go to Contact List 3. Select a contact and click Edit 4. Update contact information	Name: An Phone: 0987654321 Email: an@gmail.com	Contact information is updated successfully

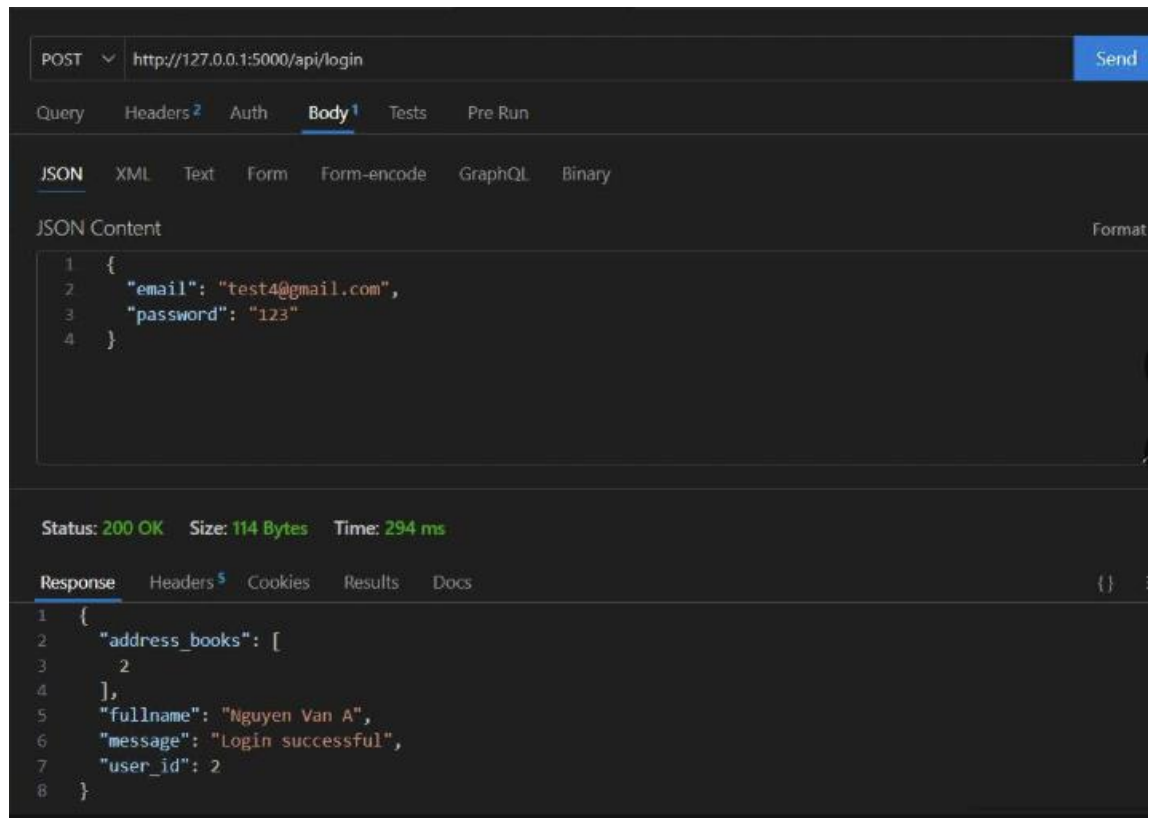
Test Case ID	Function	Test Steps	Input Data	Expected Result
		5. Click Save		
TC07	View Contact List	1. Login to the system 2. Navigate to Contact List page	AddressBook ID: 1	System returns a list of contacts successfully
TC08	View Contact List	1. Login to the system 2. Navigate to Contact List page	User account without any contacts	System displays an empty list or message " <b>No contacts found</b> "
TC09	Add Contact	1. Login to the system 2. Send POST request /api/books/2/contacts	Valid contact data	System returns error message: <i>AddressBook not found</i>
TC10	Add Contact to Group	1. Login to the system 2. Go to Add Contact page 3. Enter contact information 4. Select a group belonging to the same address book 5. Click Submit	Name: Nguyen Van A Email: vana@gmail.com Group: Friends	Contact is added successfully and assigned to the selected group
TC11	Add Contact to Group	1. Login to the system 2. Go to Add Contact page 3. Enter contact information 4. Select a group belonging to another address book 5. Click Submit	Name: David Group: Work (from another address book)	System displays an error message and contact is not added to the group

Test Case ID	Function	Test Steps	Input Data	Expected Result
TC12	Delete Contact	1. Login to the system 2. Send DELETE request /api/books/1/groups/1	Group ID: 01	Contact is deleted successfully

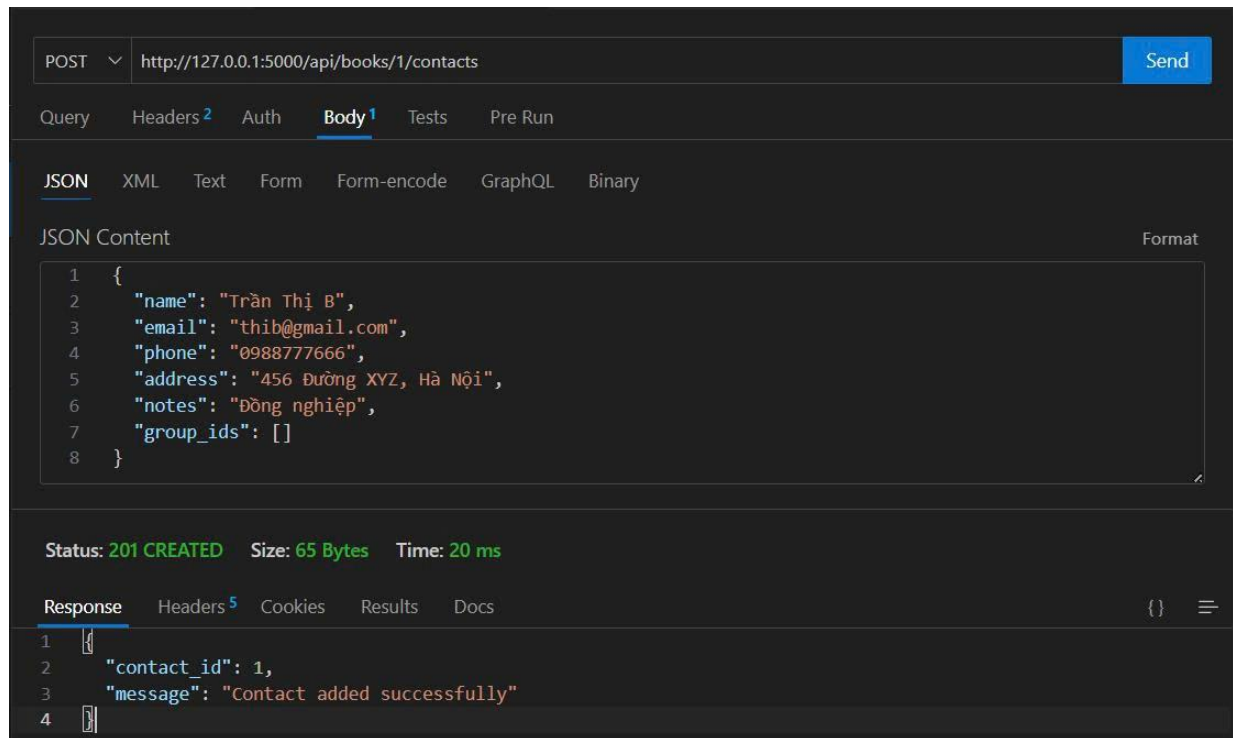
## Evidence of using Thunder Client to test the code:



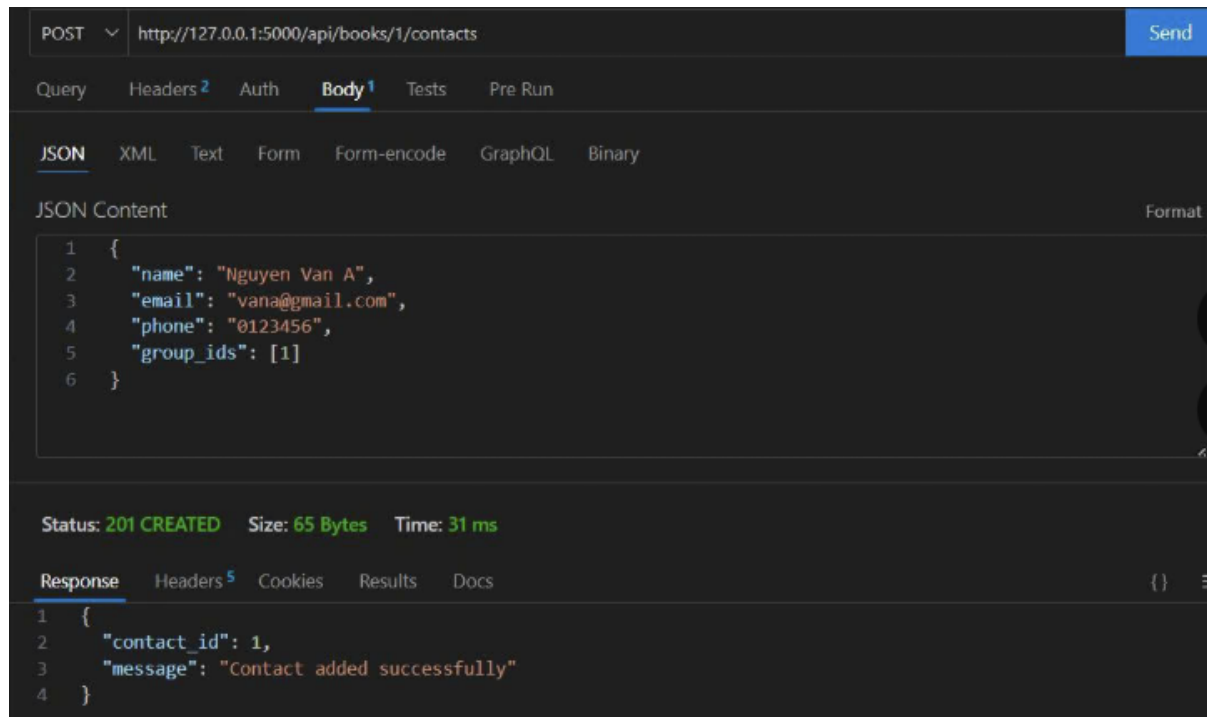
- The `/api/register` API works as expected
- Data is successfully sent and stored in the database
- The system automatically generates a `user_id` for the new user



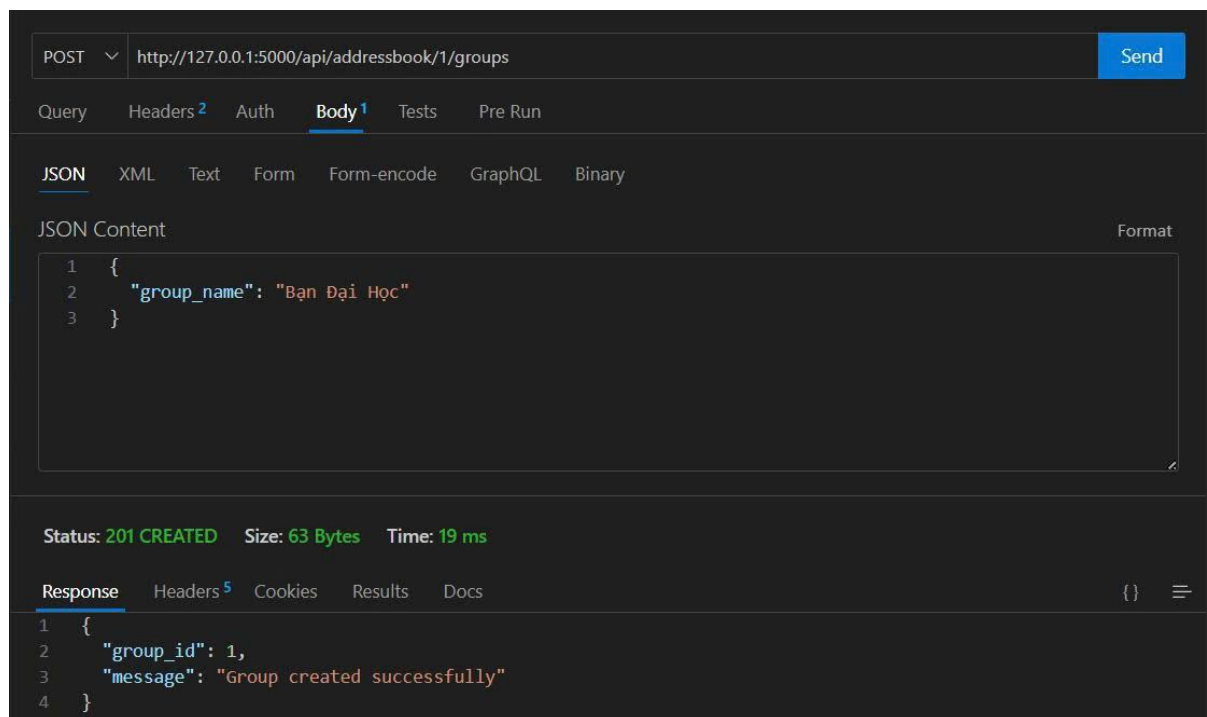
- Valid login information
- The system correctly verifies the email and password
- Returns user information and the associated AddressBook list
- The Login function works as expected\



- **API works correctly**
- **Contact is added successfully**
- **Data is stored in the database**
- **The system generates a contact\_id automatically**



- **Contact is added successfully to the specified group**
- **Contact information is processed correctly**
- **The system returns status 201 CREATED**
- **A contact\_id is automatically generated**



- **Group is created successfully**

- **Data is saved to the database**
- **The system returns status 201 CREATED**
- **A group\_id is automatically generated**

The screenshot shows a REST client interface with a POST request to `http://127.0.0.1:5000/api/books/2/contacts`. The request body is a JSON object: `{ "name": "Nguyễn Văn A", "email": "vana@gmail.com", "phone": "0912345678", "group_ids": [1, 2] }`. The response status is **404 NOT FOUND** with a size of 41 Bytes and a time of 9 ms. The response body is a JSON object: `{ "message": "AddressBook not found" }`.

POST `http://127.0.0.1:5000/api/books/2/contacts` Send

Query Headers <sup>2</sup> Auth **Body <sup>1</sup>** Tests Pre Run

**JSON** XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "name": "Nguyễn Văn A",
3   "email": "vana@gmail.com",
4   "phone": "0912345678",
5   "group_ids": [1, 2]
6 }
```

Status: **404 NOT FOUND** Size: 41 Bytes Time: 9 ms

**Response** Headers <sup>5</sup> Cookies Results Docs {} ≡

```
1 {
2   "message": "AddressBook not found"
3 }
```

Copy

- **The request failed because the AddressBook does not exist**
- **The system returns status 404 NOT FOUND**
- **Contact cannot be added to a non-existing AddressBook**

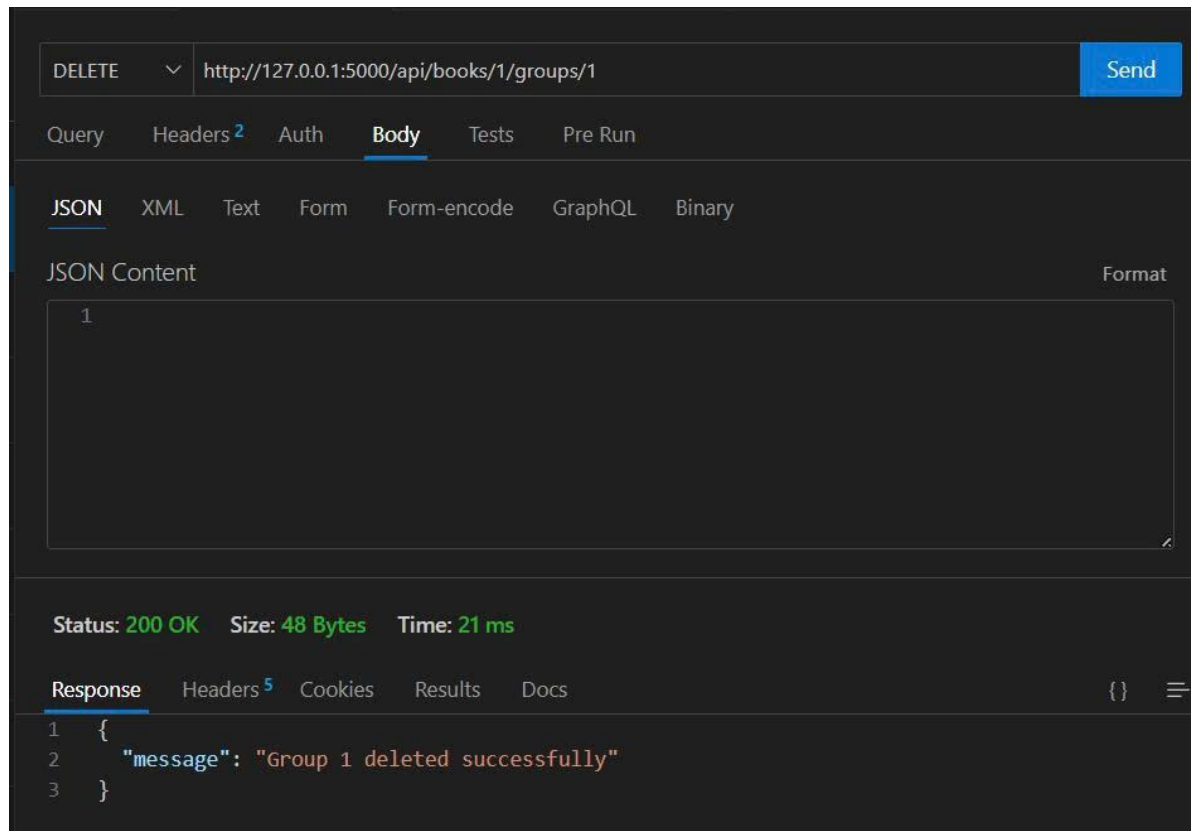
```
1  {
2    "name": "Nguyễn Văn A",
3    "email": "vana@gmail.com",
4    "phone": "0912345678",
5    "group_ids": [1, 2]
6  }
```

Status: **201 CREATED**   Size: **65 Bytes**   Time: **29 ms**

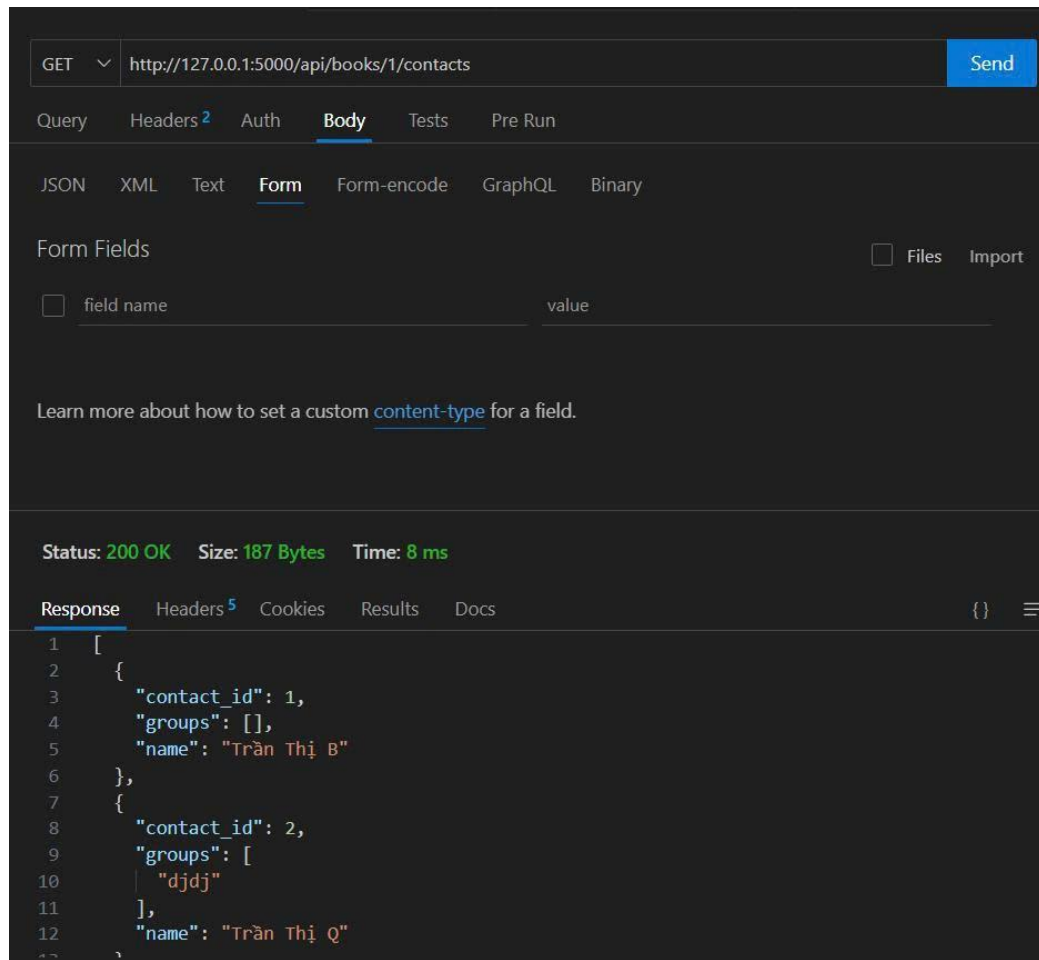
Response   Headers <sup>5</sup>   Cookies   Results   Docs

```
1  {
2    "contact_id": 1,
3    "message": "Contact added successfully"
4  }
```

- Contact information is valid
- The system successfully adds the contact to the AddressBook
- The API returns status 201 CREATED with a new contact\_id



- The system successfully deletes the selected group
- The API returns status 200 OK
- Group data is removed from the database



- **GET request to retrieve the contact list using Thunder Client, returning a successful 200 OK response with contact data.**

## Evidence of using Git for teamwork

JilueLatifah Create Dockerfile			ae6b9ee · 1 minute ago	History
Name	Last commit message	Last commit date		
.gitignore	Update and rename Bin.gitignore to .gitignore	4 minutes ago		
Dockerfile	Create Dockerfile	1 minute ago		
requirements.txt	Create requirements.txt	1 minute ago		

📄 requirements.txt	Chỉnh sửa requirements.txt	2 minutes ago
📄 gbb-bv	Chỉnh sửa gbb-bv	now
📄 Dockerfile	Chỉnh sửa Dockerfile	4 minutes ago
📄 README	Cập nhật và sửa đổi README cho README	8 minutes ago
📁 templates	Chỉnh sửa templates	1 minutes ago
📁 static	Chỉnh sửa static	now
📁 views	Thêm commit message	1 minutes ago

11:48 SA

📶 5G

🔋 59%

Code

Issues

Pull requests

More

← Files

main

...

ProgAndTest\_Group12 / models.py

📄

👤 AshesCuti

6 minutes ago

...

🕒

99 lines (75 loc) · 3.47 KB

Code

Blame

🔍

...

```

1  from sqlalchemy import create_engine, Column
2  from sqlalchemy.orm import relationship, de
3  from datetime import date
4  from flask_sqlalchemy import SQLAlchemy
5  from sqlalchemy.sql import func
6
7  db = SQLAlchemy()
8
9  contact_group = Table(
10     'contact_group',
11     db.Model.metadata,
12     Column('contact_id', Integer, ForeignKe
13     Column('group_id', Integer, ForeignKey(
14 )
15
16 class User(db.Model):
17     __tablename__ = 'users'
18
19     # Các thuộc tính
20     user_id = Column(Integer, primary_key=T
21     fullname = Column(String(100), nullable
22     email = Column(String(100), unique=True

```

4:22 CH



Public repository

main

Code



trungnhoc912-gif 3 hours ago



Add user authentication and contact management APIs  
6326d5c

Implement user registration and login APIs with contact  
management features.

static 7 hours ago

templates 7 hours ago

.gitignore 7 hours ago

Dockerfile 7 hours ago

app.py 3 hours ago

models.py 4 hours ago

requirements.txt 7 hours ago

README



## Add a README

Help people interested in this repository understand  
your project.

Add a README











main 1 Branch 0 Tags

Go to file

Add file

Code

 <b>JilueLatifah</b> Update models.py	f9fb85a · 2 hours ago	🕒 20 Commits
 static	Add files via upload	yesterday
 templates	Add files via upload	yesterday
 .gitignore	Update .gitignore	2 hours ago
 Dockerfile	Create Dockerfile	3 days ago
 app.py	Update app.py	2 hours ago
 models.py	Update models.py	2 hours ago
 requirements.txt	Create requirements.txt	3 days ago

README



## Add a README

Help people interested in this repository understand your project.