PharmaDoc Technical Report

By Ahmed Dammak

January 21, 2024

Contents

1 Introduction

PharmaDoc is a sophisticated web application developed to optimize the management of pharmaceutical products. The underlying architecture is based on a RESTful API built with the powerful and scalable ExpressJS and NodeJS frameworks. This backend structure is complemented by MongoDB, a robust database system chosen for its efficiency in handling pharmaceutical product data. A key focus of the application is to provide essential functionalities while ensuring a secure authentication mechanism for system access.

2 Technologies Used

2.1 Backend Framework

The backend relies on the robust and scalable [A] [B] ExpressJS and NodeJS framework, which ensures efficient processing and handling of requests.

ExpressJS is a minimal and flexible Node.js web application framework that provides a robust set of features to develop web and mobile applications. It acts as a middleware to help manage routes, handle HTTP requests, and define various endpoints. ExpressJS simplifies the process of building a web server, making it an ideal choice for developing RESTful APIs, which form the backbone of the Pharmacy Website Showcase.

Key features of ExpressJS include:

- **Routing:** Express provides a simple and intuitive way to define routes for different parts of the application, making it easy to organize and structure the code.
- **Middleware:** Middleware functions in Express allow for the execution of code at various points during the request-response cycle. This is invaluable for tasks such as authentication, logging, and error handling.
- **Template Engine:** While the Pharmacy Website Showcase primarily serves as an API, ExpressJS can also handle rendering dynamic content using template engines, if needed in the future.
- **Extensibility:** Express is highly extensible, allowing developers to integrate additional libraries and modules seamlessly.

ExpressJS provides a solid foundation for building scalable and maintainable server-side applications, making it a preferred choice for the backend framework.

2.1.1 NodeJS

NodeJS is the runtime that enables server-side JavaScript execution. It utilizes an event-driven, non-blocking I/O model, making it efficient for handling concurrent connections. NodeJS, coupled with ExpressJS, forms a powerful duo for building fast, scalable, and real-time applications.

Key aspects of NodeJS include:

- **Asynchronous I/O:** NodeJS is designed to efficiently handle asynchronous operations, making it well-suited for applications that require high concurrency.
- **Package Management:** Node Package Manager (NPM) is the default package manager for NodeJS, allowing easy installation and management of third-party libraries and dependencies.
- **Cross-Platform Compatibility:** NodeJS is cross-platform, ensuring that the Pharmacy Website Showcase can be deployed on various operating systems without modification.
- **Scalability:** NodeJS is known for its scalability, and its event-driven architecture enables handling a large number of simultaneous connections with low latency.

By leveraging the strengths of both ExpressJS and NodeJS, the backend of the Pharmacy Website Showcase achieves a balance between performance, flexibility, and ease of development.

2.2 Database

The Pharmacy Website Showcase relies on MongoDB, a NoSQL database, to manage pharmaceutical product data. MongoDB's NoSQL architecture offers flexibility, allowing the system to handle diverse and evolving product information efficiently.

2.2.1 NoSQL and MongoDB

MongoDB's schema-less approach is advantageous for handling dynamic pharmaceutical product data. It stores information in flexible, JSON-like documents, accommodating varied product attributes without requiring predefined schemas.

mongoDB Compass

2.2.2 Flexibility and Scalability

MongoDB's flexibility enables seamless integration of new attributes as pharmaceutical product data evolves. Its horizontal scalability ensures optimal performance, making it suitable for managing large datasets associated with a growing number of products.

2.2.3 Querying and Indexing

MongoDB provides powerful querying capabilities, allowing the application to retrieve specific sets of product data based on criteria such as searches, category-based listings, and price ranges. Efficient indexing enhances query performance, ensuring a responsive user experience.

In summary, MongoDB's NoSQL architecture, flexibility, scalability, and efficient querying make it an ideal choice for handling the diverse and dynamic data associated with pharmaceutical products in the Pharmacy Website Showcase.

2.3 Authentication

To safeguard system functionalities, secure authentication mechanisms have been implemented, leveraging technologies such as JSON Web Tokens (JWT). This ensures that only authorized users can access the system.

2.4 API Testing

POSTMAN

Postman, a comprehensive API testing tool, is employed to validate and evaluate the effectiveness of the API. This ensures the reliability and correctness of the interactions between different components of the system.

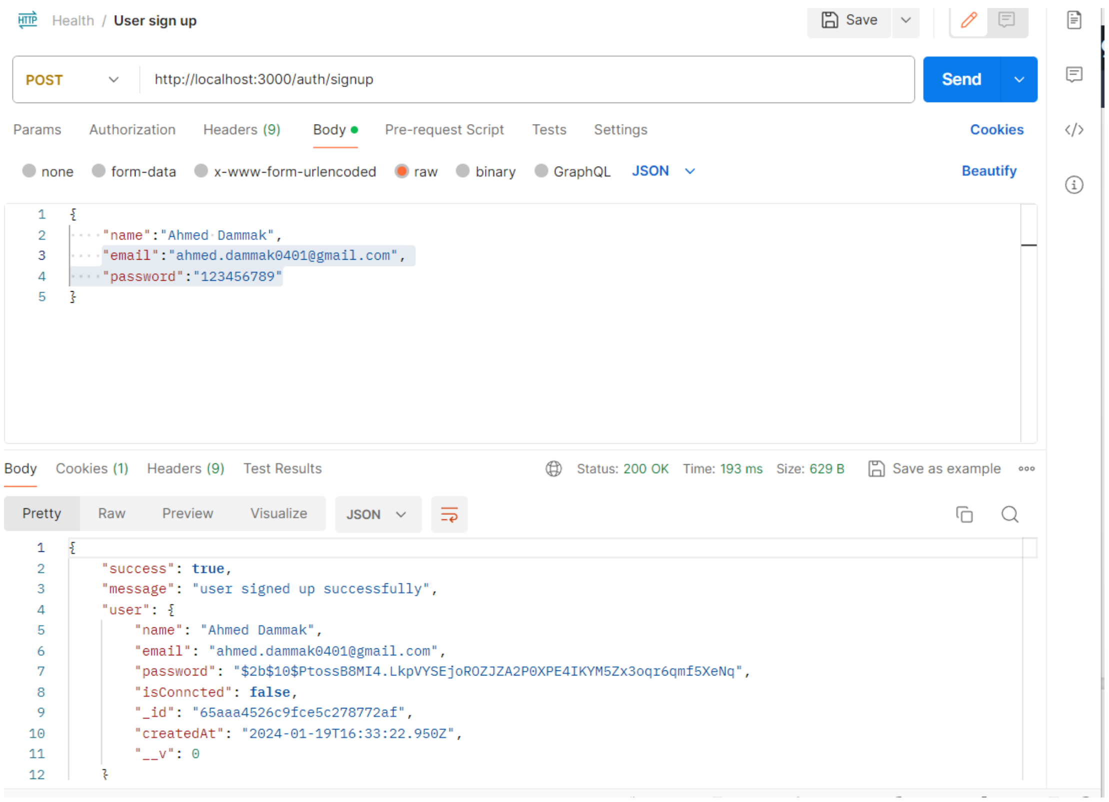
3 Functionalities

3.1 Authentication

To ensure the security of system functionalities, the Pharmacy Website Showcase employs a robust authentication mechanism, leveraging JSON Web Tokens (JWT). JWT is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. In the context of the Pharmacy Website Showcase, JWTs play a crucial role in verifying the identity of users and ensuring authorized access to the system.

3.1.1 Endpoint: `POST /auth/signup`

New users can sign up for the application through this endpoint. User registration involves submitting necessary information, and upon successful registration, the user is granted access to the system.



POST http://localhost:3000/auth/signup

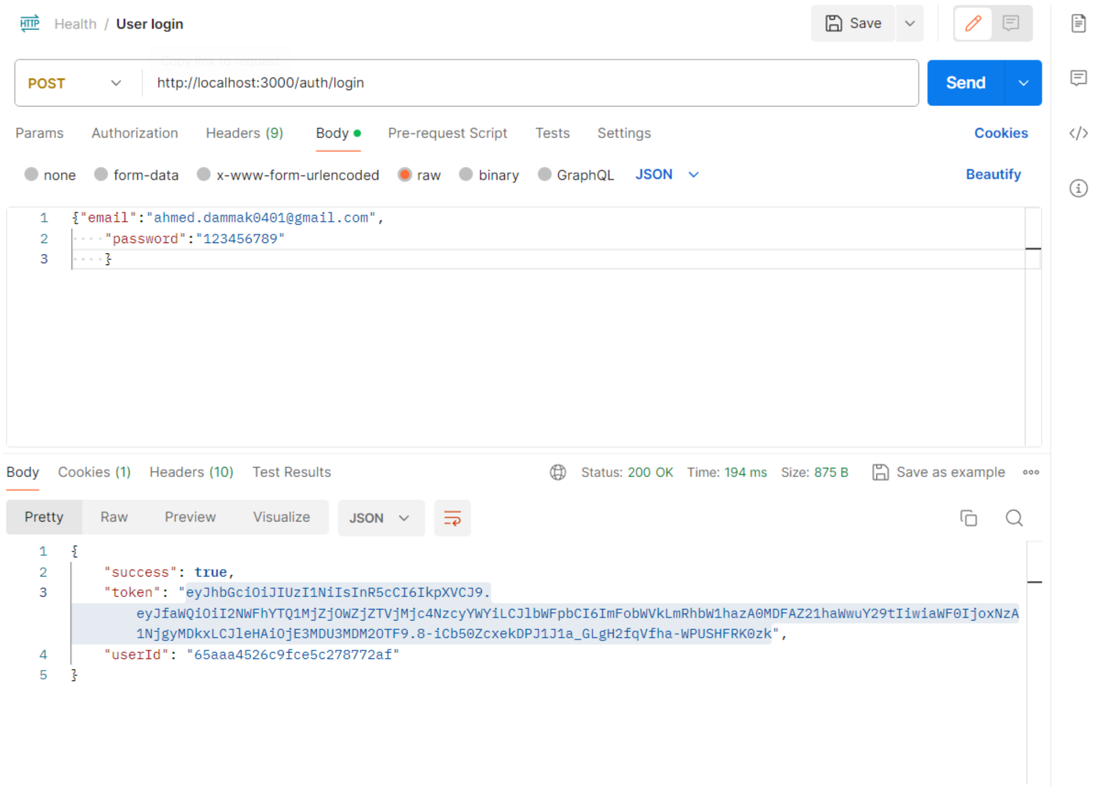
```
{
    "name":"Ahmed Dammak",
    "email":"ahmed.dammak0401@gmail.com",
    "password":"123456789"
}
```

Status: 200 OK Time: 193 ms Size: 629 B

```
{
    "success": true,
    "message": "user signed up successfully",
    "user": {
        "name": "Ahmed Dammak",
        "email": "ahmed.dammak0401@gmail.com",
        "password": "$2b$10$PtossB8MI4.LkpVYSEjoROZJZA2P0XPE4IKYM5Zx3oqr6qmf5XeNq",
        "isConncted": false,
        "_id": "65aaa4526c9fce5c278772af",
        "createdAt": "2024-01-19T16:33:22.950Z",
        "__v": 0
    }
```

3.1.2 Endpoint: `POST /auth/login`

Upon successful authentication, the system issues a JWT token. This token is used for subsequent requests, providing a secure and efficient means of user validation.



http://localhost:3000/auth/login

```
{"email":"ahmed.dammak0401@gmail.com",
    "password":"123456789"
    }
```

Status: 200 OK Time: 194 ms Size: 875 B

```
{
    "success": true,
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
        eyJfaWQiOiI2NWFhYTQ1MjZjOWZjZTVjMjc4NzcyYWYiLCJlbWFpbCI6ImFobWVkLmRhbW1hazA0MDFAZ21haWwuY29tIiwiaWF0IjoxNzA
        1NjgyMDkxLCJleHAiOjE3MDU3MDM2OTF9.8-iCb50ZcxekDPJ1J1a_GLgH2fqVfha-WPUSHFRK0zk",
    "userId": "65aaa4526c9fce5c278772af"
}
```

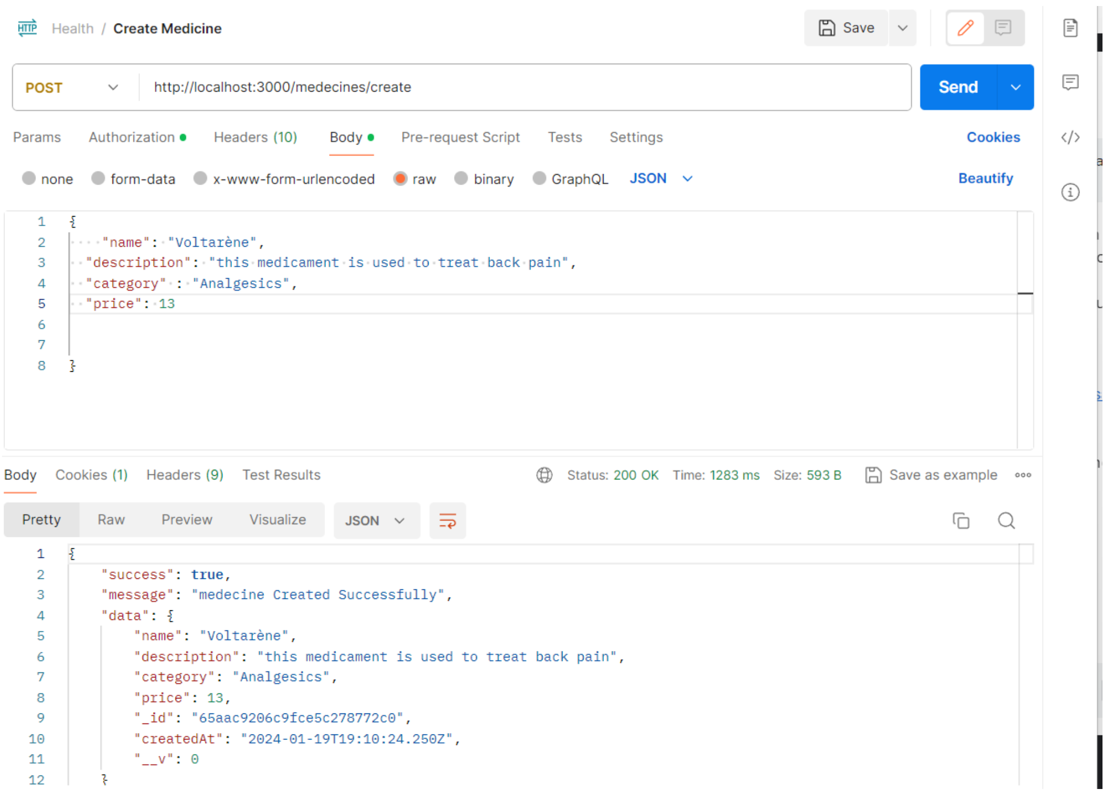
3.2 Medicine Management

3.2.1 Adding New Medicine

Users can seamlessly add new pharmaceutical products to the database through the API. This functionality includes capturing essential details such as name, description, and price.

3.2.2 Endpoint: `POST /Medecines/create`

Product details, including name, description, and price, are submitted through the API, ensuring a standardized and structured approach to data input.



Health / Create Medicine

POST http://localhost:3000/medecines/create

```
{
    "name": "Voltarène",
  "description": "this medicament is used to treat back pain",
  "category" : "Analgesics",
  "price": 13


}
```

Status: 200 OK Time: 1283 ms Size: 593 B

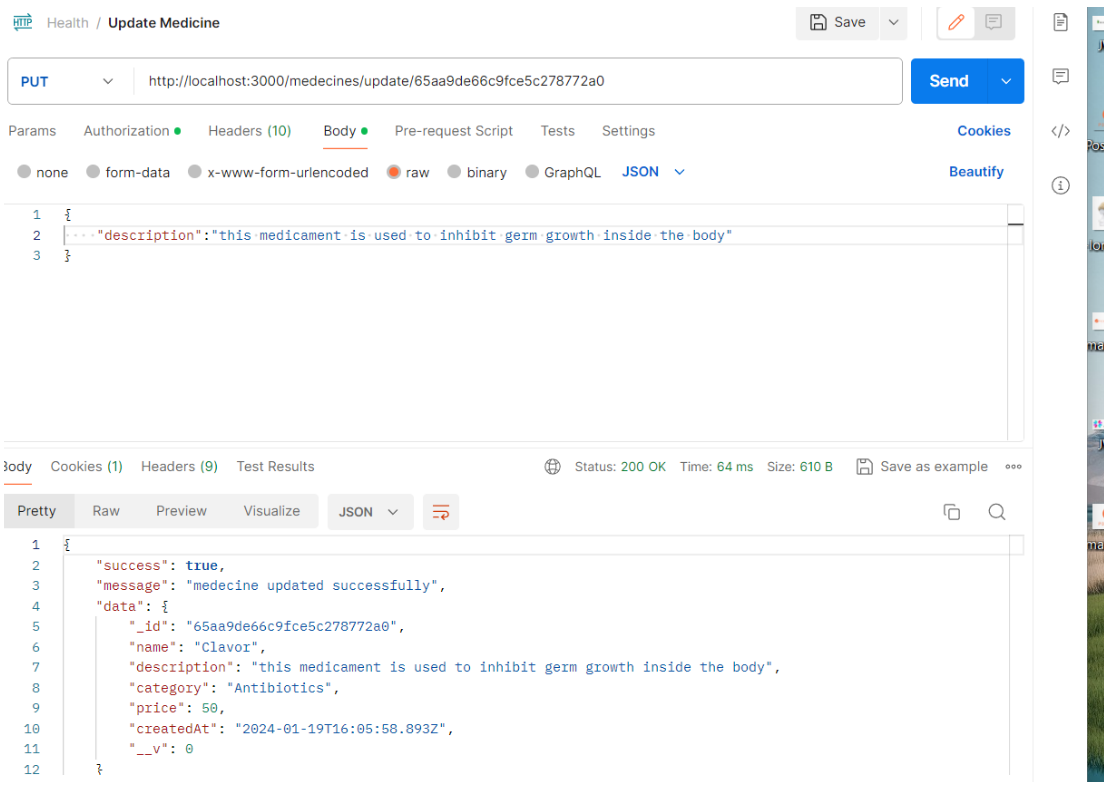
```
{
    "success": true,
    "message": "medecine Created Successfully",
    "data": {
        "name": "Voltarène",
        "description": "this medicament is used to treat back pain",
        "category": "Analgesics",
        "price": 13,
        "_id": "65aac9206c9fce5c278772c0",
        "createdAt": "2024-01-19T19:10:24.250Z",
        "__v": 0
    }
```

3.2.3 Updating Medicine

Users have the capability to update existing product information, allowing for dynamic adjustments to product details.

3.2.4 Endpoint: `PUT /Medecines/update/:productId`

Updated product details, such as price, quantity, or description, can be efficiently provided through the API. This ensures that the database reflects the most accurate and current information.



http://localhost:3000/medecines/update/65aa9de66c9fce5c278772a0

```
{
    "description":"this medicament is used to inhibit germ growth inside the body"
}
```

Status: 200 OK Time: 64 ms Size: 610 B

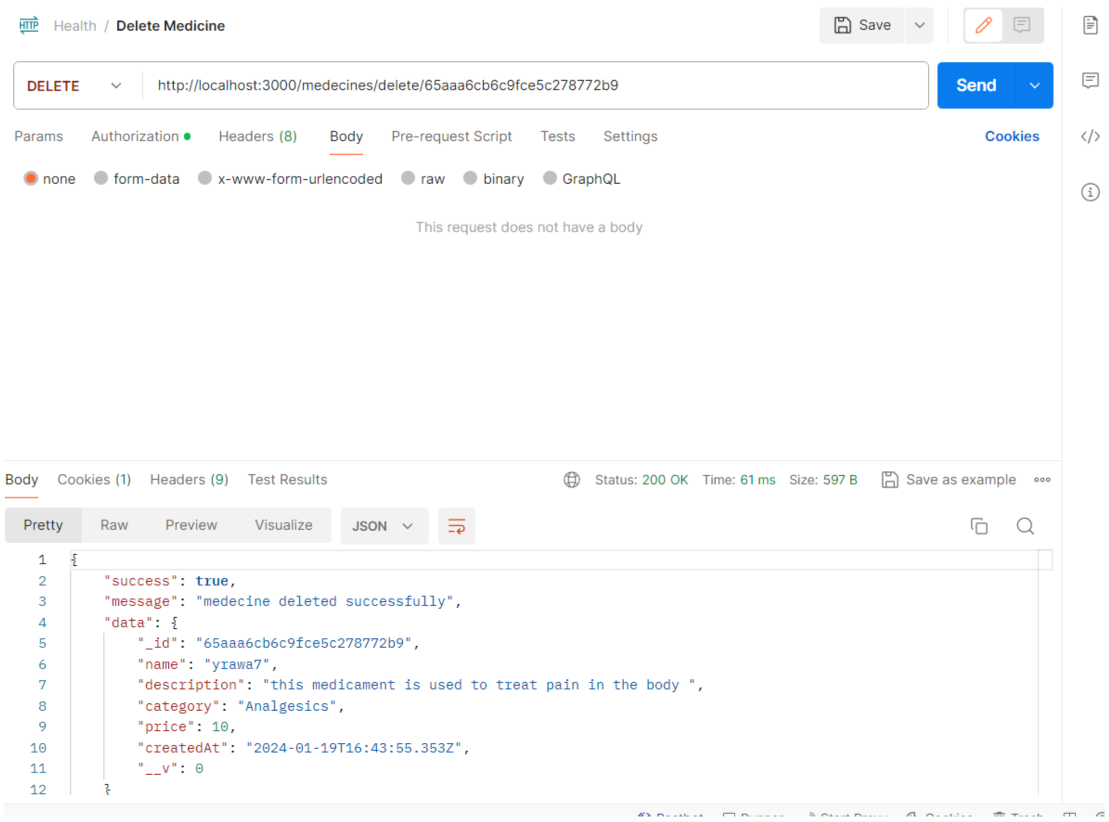
```
{
    "success": true,
    "message": "medecine updated successfully",
    "data": {
        "_id": "65aa9de66c9fce5c278772a0",
        "name": "Clavor",
        "description": "this medicament is used to inhibit germ growth inside the body",
        "category": "Antibiotics",
        "price": 50,
        "createdAt": "2024-01-19T16:05:58.893Z",
        "__v": 0
    }
```

3.2.5 Deleting Medicine

Users can remove medicines from the database using the API. This functionality is crucial for maintaining an up-to-date and relevant product inventory.

3.2.6 Endpoint: `DELETE /medecines/delete/:productId`

The specified medicine is removed from the database, ensuring the elimination of outdated or irrelevant items.



http://localhost:3000/medecines/delete/65aaa6cb6c9fce5c278772b9

This request does not have a body

Status: 200 OK Time: 61 ms Size: 597 B

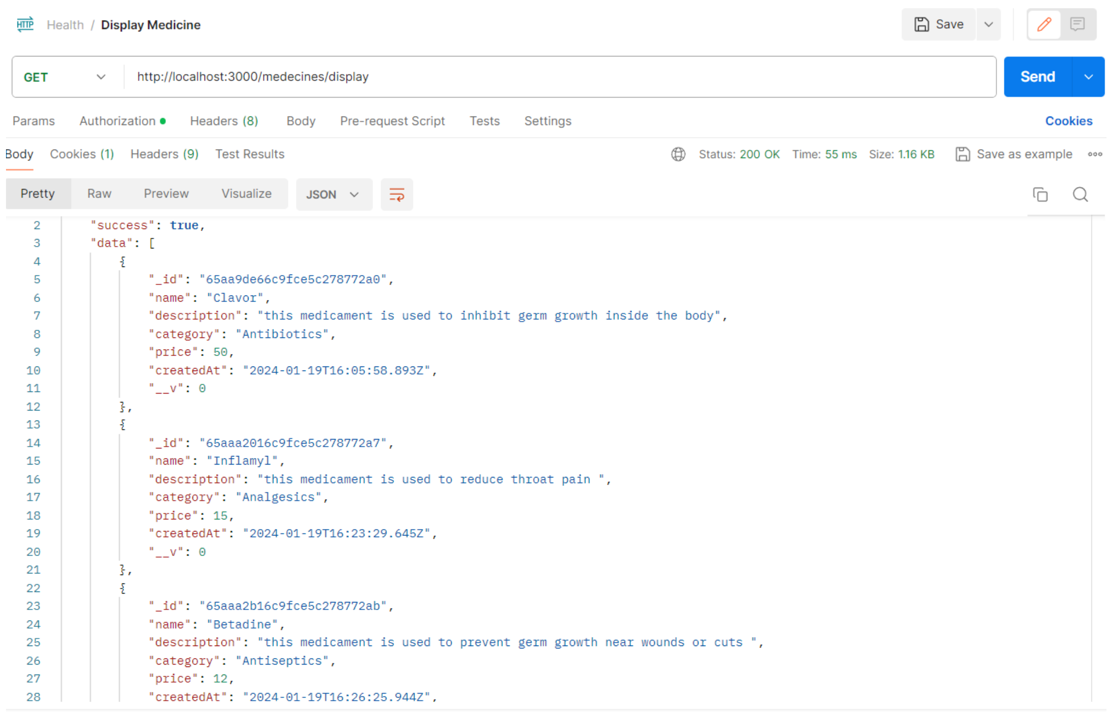
```
{
    "success": true,
    "message": "medecine deleted successfully",
    "data": {
        "_id": "65aaa6cb6c9fce5c278772b9",
        "name": "yrawa7",
        "description": "this medicament is used to treat pain in the body ",
        "category": "Analgesics",
        "price": 10,
        "createdAt": "2024-01-19T16:43:55.353Z",
        "__v": 0
    }
```

3.3 Viewing Medicines

Users can easily browse and view a list of available pharmaceutical products through dedicated API endpoints.

3.3.1 Endpoint: `GET /medecines/display`

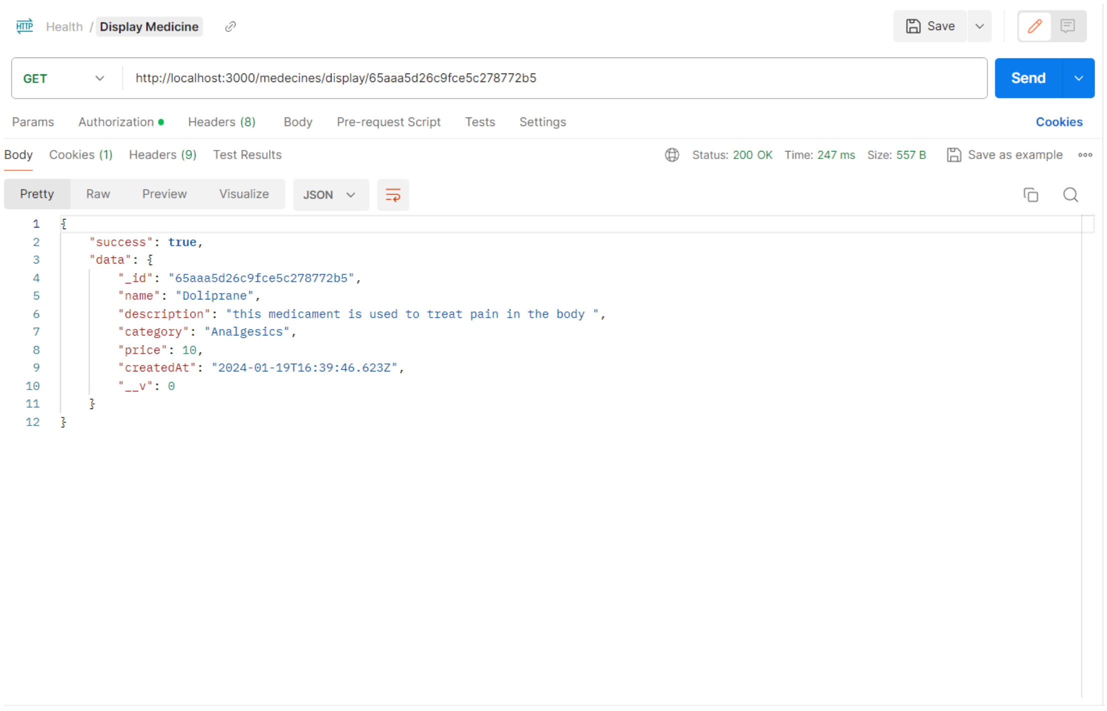
This endpoint returns a list of available medicines with basic information, providing a quick overview of the medicine catalog.


Health / Display Medicine
Save
GET
http://localhost:3000/medecines/display
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Cookies
Body Cookies (1) Headers (9) Test Results
Status: 200 OK Time: 55 ms Size: 1.16 KB
Save as example
Pretty Raw Preview Visualize JSON

```
    "success": true,
    "data": [
        {
            "_id": "65aa9de66c9fce5c278772a0",
            "name": "Clavor",
            "description": "this medicament is used to inhibit germ growth inside the body",
            "category": "Antibiotics",
            "price": 50,
            "createdAt": "2024-01-19T16:05:58.893Z",
            "__v": 0
        },
        {
            "_id": "65aaa2016c9fce5c278772a7",
            "name": "Inflamyl",
            "description": "this medicament is used to reduce throat pain ",
            "category": "Analgesics",
            "price": 15,
            "createdAt": "2024-01-19T16:23:29.645Z",
            "__v": 0
        },
        {
            "_id": "65aaa2b16c9fce5c278772ab",
            "name": "Betadine",
            "description": "this medicament is used to prevent germ growth near wounds or cuts ",
            "category": "Antiseptics",
            "price": 12,
            "createdAt": "2024-01-19T16:26:25.944Z",
```

3.3.2 Endpoint: `GET /medecines/display/:productId`

For more detailed information about a specific medicine, users can access this endpoint, obtaining comprehensive details about the specified medicine.
‘



http://localhost:3000/medecines/display/65aaa5d26c9fce5c278772b5

```
{
    "success": true,
    "data": {
        "_id": "65aaa5d26c9fce5c278772b5",
        "name": "Doliprane",
        "description": "this medicament is used to treat pain in the body ",
        "category": "Analgesics",
        "price": 10,
        "createdAt": "2024-01-19T16:39:46.623Z",
        "__v": 0
    }
}
```

4 Security Considerations

4.1 Authentication

Robust authentication mechanisms, including the use of JWT, are implemented to secure system access. This ensures that only authorized users can interact with the application.

4.2 Authorization

To control access and actions within the system, proper authorization checks are in place. This prevents unauthorized users from performing sensitive operations.

4.3 Data Encryption

Sensitive data, such as user credentials, is encrypted to enhance overall system security. This precautionary measure protects critical information from unauthorized access or potential breaches.

5 Conclusion

PharmaDoc is meticulously designed to provide a seamless and secure experience for managing pharmaceutical products. The RESTful API architecture ensures scalability, allowing the system to handle varying loads efficiently. The chosen technologies, including ExpressJS, NodeJS, and MongoDB, strike a balance between performance and security. The use of Postman as a testing tool validates the effectiveness and reliability of the API, ensuring a robust and error-free application.