Patient and Volunteer Management API Documentation

1.Introduction

Welcome to the Patient and Volunteer Management API! This API allows you to manage patients and volunteers in your healthcare system. You can view and add patients and volunteers as well as assign patients to volunteers.

Base URL

Localhost: `http://localhost:8000/api/`

2.Authentication

This API uses token-based authentication. To access the API, include the token in the ‘Authorization’ header

`Authorization`: Bearer your_access_token

3.Endpoints For Admin Dashboard

3.1 **Login admin user**

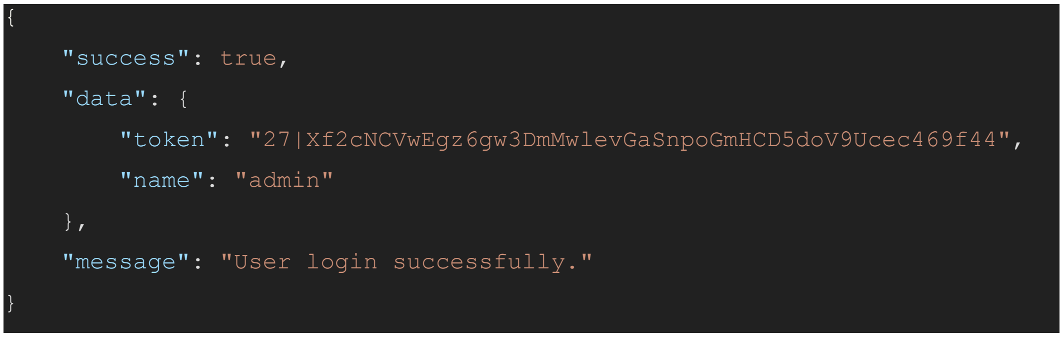
Endpoint: ‘/login’

Method: ‘POST’

Request Body:

```
{
        “email”:”admin@example.com”,
        “password”:”password”
}
```

Example Response



```
{
    "success": true,
    "data": {
        "token": "27|Xf2cNCVwEgz6gw3DmMwlevGaSnpoGmHCD5doV9Ucec469f44",
        "name": "admin"
    },
    "message": "User login successfully."
}
```

3.2 **Get Patient List**

Endpoint: “/patients”

Method: ‘GET’

Headers:

Authorization: Bearer login_user_token,

Accept : application/json

Example Response

```
{
    "success": true,
    "data": [
        {
            "id": 1,
            "name": "Patient 1",
            "code": "942642",
            "volunteer": "Linn Htet",
            "created_at": "31/08/2024",
            "updated_at": "31/08/2024"
        },
        {
            "id": 2,
            "name": "Test 2",
            "code": "942555",
            "volunteer": "Test Volunteer",
            "created_at": "31/08/2024",
            "updated_at": "31/08/2024"
        }
    ],
    "message": "Patients retrieved successfully."
}
```

3.3 **Patient Create**

Endpoint: “/patients”

Method: POST

Headers:

Authorization: Bearer login_user_token,

Accept : application/json

Request Body:

```
{
        “name”: “Example Patient”,
        “volunteer_id”: 1
}
```

Example Response

```
{
    "success": true,
    "data": {
        "id": 8,
        "name": "Patient two",
        "code": "053243",
        "volunteer": "Linn Htet",
        "created_at": "02/09/2024",
        "updated_at": "02/09/2024"
    },
    "message": "Patient created successfully."
}
```

3.4 **Volunteer List**

Endpoint: “/volunteers”
Method: ‘GET’
Headers:
Authorization: Bearer login_user_token,
Accept : application/json
Example Response

```
{
    "success": true,
    "data": [
        {
            "id": 1,
            "name": "Linn Htet",
            "code": "VOL_409578",
            "password": "291494",
            "created_at": "31/08/2024",
            "updated_at": "31/08/2024"
        },
        {
            "id": 2,
            "name": "Test Volunteer",
            "code": "VOL_310058",
            "password": "081742",
            "created_at": "31/08/2024",
            "updated_at": "31/08/2024"
        }
    ],
    "message": "Volunteer retrieved successfully."
}
```

3.5 **Volunteer Create**

Endpoint: “/patients”
Method: POST
Headers:

Authorization: Bearer login_user_token,
Accept : application/json

Request Body:

{
name: “example volunteer”
}

```
{
    "success": true,
    "data": {
        "name": "example volunteer",
        "code": "VOL_261560",
        "password": "617424"
    },
    "message": "Volunteer create successfully."
}
```

4.Endpoints for Mobile

4.1 Volunteer Login

Endpoint: “/volunteer/login”
Method: POST
Request Body:

{
“name”:”volunteer name”,
“Password”: “volunteer auto generated password”
}

Example Response

```
{
    "token": "28|6pY8qUS1zUBSBSKVKSnHzIQSfadZlr8yCuFRxFOHc49c5dda",
    "volunteer": {
        "id": 1,
        "name": "Linn Htet",
        "code": "VOL_409578",
        "created_at": "2024-08-31T03:07:16.000000Z",
        "updated_at": "2024-08-31T03:07:16.000000Z"
    }
}
```

4.2 Patient List by login volunteer

Endpoint: “/volunteer/patients”
Method: ‘GET’
Headers:

Authorization: Bearer login_user_token,

Accept : application/json

Example Response

```
{
    "success": true,
    "data": [
        {
            "id": 1,
            "name": "Patient 1",
            "code": "942642",
            "volunteer_name": "Linn Htet",
            "created_at": "2024-08-31 03:17:20",
            "updated_at": "2024-08-31 03:17:20"
        },
        {
            "id": 3,
            "name": "test 3",
            "code": "818901",
            "volunteer_name": "Linn Htet",
            "created_at": "2024-08-31 06:36:13",
            "updated_at": "2024-08-31 06:36:13"
        }
    ],
    "message": "Patient list successfully."
}
```

5.Run Application In Localhost

5.1 Use laravel for backend

RUN laravel application in localhost

1. Clone https://github.com/linndev/Volunteer-Patient-Management
2. Go to volunteer-patient-management folder and “composer install” and “php artisan serve”

RUN react application in localhost

1.go to patient-volunteer-fe and “npm install and npm run dev”