To set up and run the Patient Management System, follow these steps. This guide assumes you have Node.js, MySQL, and Postman installed.

**Step 1: Clone the Repository**

1. Clone the repository to your local machine:

git clone <repository-url>

1. Navigate to the project folder:

cd folder name

**Step 2: Install Dependencies**

1. Install required Node.js dependencies:

npm install

**Step 3: Set Up MySQL Database**

1. **Start MySQL Server**: Ensure that MySQL is running (you can use XAMPP, WAMP, or MySQL Workbench).
2. **Create a Database**:
   * Open a MySQL client (e.g., MySQL Workbench or phpMyAdmin) and run the following SQL command:

CREATE DATABASE patient\_management;

1. **Create Tables**:
   * Use the following SQL commands to set up the necessary tables:

USE patient\_management;

-- Create 'users' table

CREATE TABLE IF NOT EXISTS users (

id INT AUTO\_INCREMENT PRIMARY KEY,

email VARCHAR(255) NOT NULL UNIQUE,

password VARCHAR(255) NOT NULL,

role ENUM('Patient', 'Doctor', 'Admin') NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

-- Create 'patient\_records' table

CREATE TABLE IF NOT EXISTS patient\_records (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

age INT NOT NULL,

doctor\_id INT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

FOREIGN KEY (doctor\_id) REFERENCES users(id)

);

-- Create 'appointments' table

CREATE TABLE IF NOT EXISTS appointments (

id INT AUTO\_INCREMENT PRIMARY KEY,

doctor\_id INT NOT NULL,

patient\_id INT NOT NULL,

date DATETIME NOT NULL,

status ENUM('scheduled', 'completed', 'canceled') DEFAULT 'scheduled',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

FOREIGN KEY (doctor\_id) REFERENCES users(id)

FOREIGN KEY (patient\_id) REFERENCES users(id)

);

1. **Add Sample Data** (Optional): Insert test data into the tables if needed.

**Step 4: Configure Environment Variables**

1. **Create a .env file** in the root of the project directory:

touch .env

1. **Add Database and JWT Configuration**:
   * Open the .env file and add the following environment variables:

DB\_HOST=localhost

DB\_USER=root

DB\_PASSWORD=your\_mysql\_password

DB\_NAME=patient\_management

JWT\_SECRET=your\_jwt\_secret

PORT=5000

* + Replace your\_mysql\_password with your actual MySQL password, and your\_jwt\_secret with a secret key for JWT.

**Step 5: Run the Application**

1. Start the server:

Node server.js

1. If everything is set up correctly, you should see a message indicating the server is running:

Server running on port 5000

**Step 6: Testing the API in Postman**

1. **Open Postman**.
2. Import the provided Postman collection if available, or manually configure the requests for registration, login, and CRUD operations for patient records and appointments.
3. **Register and Login**:
   * Register users with different roles (Patient, Doctor, Admin) and log in to get JWT tokens for authorization.
4. **Authorization**:
   * Use the token from the login response in the **Authorization** header for each request as Bearer <token>.

**Example Postman Requests**

* **Register a User**: POST http://localhost:5000/api/auth/register
* **Login a User**: POST http://localhost:5000/api/auth/login
* **Create Patient Record** (Doctor/Admin only): POST http://localhost:5000/api/patient-records
* **Read Patient Records** (all roles): GET http://localhost:5000/api/patient-records
* **Create Appointment** (Patient/Admin only): POST http://localhost:5000/api/appointments

Refer to the role-based access controls to ensure users only access permitted endpoints.

**Troubleshooting**

1. **Database Connection Issues**:
   * Verify that MySQL is running and the credentials in .env are correct.
2. **JWT Issues**:
   * Ensure JWT\_SECRET in the .env file is set and matches your server configuration.
3. **Permission Errors**:
   * Ensure each endpoint in Postman includes the correct **Authorization** header with the JWT token.

Following these steps should help you successfully set up, run, and test the Patient Management System API. Let me know if you need further assistance!