Healthcare System API Design with Role-Based Authentication

Overview

This API design outlines a RESTful API for a hospital management system with role-based authentication. The system supports roles: Patient, Nurse, Doctor, and Administrator. The API uses OAuth 2.0 with JWT for authentication and authorization, ensuring compliance with HIPAA.

Authentication

- **Endpoint**: `POST /auth/login`
    - **Description**: Authenticates users and issues a JWT with role claims.

Request Body:
```
{
  "username": "string",
  "password": "string"
}
```

    -

Response:
```
{
  "access_token": "jwt_token",
  "expires_in": 3600
}
```

    -

Role Claim in JWT:
```
{
  "sub": "user_id",
  "role": "Patient|Nurse|Doctor|Administrator",
  "permissions": ["view_records", "update_status", ...]
}
```

    -

Authorization Middleware

- **Implementation**: Use middleware to verify JWT and check role-based permissions.

API Endpoints

1. View Medical Records

- **Endpoint**: `GET /records/{patient_id}`
- **Roles**: Patient (own records), Nurse, Doctor
- **Permissions**: `view_records`

Response:
```
{
  "patient_id": "string",
  "records": [...]
}
```

- **RBAC Logic**: Patients can only access their own records (`patient_id == user_id`).

2. Update Patient Status

- **Endpoint**: `PATCH /records/{patient_id}/status`
- **Roles**: Nurse, Doctor
- **Permissions**: `update_status`

Request Body:
```
{
  "status": "string"
}
```

-
- **Response**: 204 No Content

3. Prescribe Medication

- **Endpoint**: `POST /records/{patient_id}/prescriptions`

- **Roles**: Doctor
- **Permissions**: `prescribe_medication`

Request Body:
```
{
  "medication": "string",
  "dosage": "string"
}
```

-
- **Response**: 201 Created

4. Manage User Accounts

- **Endpoint**: `POST /users`
- **Roles**: Administrator
- **Permissions**: `manage_users`

Request Body:
```
{
  "username": "string",
  "role": "Patient|Nurse|Doctor|Administrator"
}
```

-
- **Response**: 201 Created

Security Measures

- **TLS**: All endpoints use HTTPS.
- **HIPAA Compliance**: Encrypt sensitive data, log access for auditing.
- **Emergency Access**: Temporary role escalation for Doctors, logged for review.

Database Schema

Table	Columns
Users	id, username, password_hash, role
Records	id, patient_id, data, created_at
Permissions	role, permission

Audit_Logs	id, user_id, action, timestamp