MediSync

- The system is a basic clinic management system

User :

- Only a registered , Authenticated user is allowed to login
- After login the Authenticated user can add patient , doctors
- User can only view patients which are added by him
- User can view all doctors even if he didn’t added them
- After creation of patient and doctors , user can perform GET, PUT ,DELETE operations on their records
- User can assign a doctor to a patient , retrive all patient doctor mappings , get all doctors assigned to a specific patient , and also can remove a doctor from a patient

s

User

User-Id , Email , password

Patient

Patient-ID , userID , patientName , Mobile-No , Disease (long sentence) , Query-DateTime , Assigned-DateTime

Doctor

Doctor-ID , Name , Email , MobileNo , Qualification

1. Authentication APIs

POST /api/auth/register - Register a new user with name, email, and password.

Request :

```
{
  "username": "test_user",
  "email": "user123@example.com",
  "password": "StrongPass@123",
  "confirmPassword": "StrongPass@123"
}
```

POST /api/auth/login - Log in a user and return a JWT token.

```
{
  "email": "user123@example.com",
  "password": "StrongPass@123",
}
```

Note : Response will include user id , keep it safely and use it for each request further

2. Patient Management APIs

POST /api/patients - Add a new patient (Authenticated users only).

Request :

```
{
  "userID": 101,
  "patientName": "Jane Doe",
  "mobileNo": "9876543210",
  "disease": "Hypertension",
  "queryDateTime": "2025-06-06T10:30:00Z",
  "assignedDateTime": "2025-06-06T12:00:00Z"
}
```

GET /api/patients - Retrieve all patients created by the authenticated user.

Request :

```
{
  "userID": 101
}
```

Note : use the user id you gotas response after login

GET /api/patients/:id - Get details of a specific patient.

Example : http://localhost:5000/patients/1

Request :

```
{
  "userID": 101


}
```

PUT /api/patients/:id - Update patient details.

Example : http://localhost:5000/patients/1

Request :

```
{
  "userID": 101,
  "patientName": " update Jane Doe",
  "mobileNo": "9876543210",
  "disease": "update Hypertension",
}
```

DELETE /api/patients/:id - Delete a patient record.

Example : http://localhost:5000/patients/1

Request :

```
{
  "userID": 101
}
```

Note: put only that userID , which you got as response after login

3. Doctor Management APIs

POST /api/doctors - Add a new doctor (Authenticated users only).

Request :

```
{
  "userID": 202,
  "name": "Test User",
  "email": "user202@example.com",
  "mobileNo": "9123456780",
  "qualification": "MBBS"
}
```

Note: put only that userID , which you got as response after login

GET /api/doctors - Retrieve all doctors.

Example : Example : http://localhost:5000/doctors/1

Request :

```
{ }
```

GET /api/doctors/:id - Get details of a specific doctor.

Request :

```
{ }
```

PUT /api/doctors/:id - Update doctor details.

Example : Example : http://localhost:5000/doctors/1

Request :

```
{
  "userID": 202,
  "name": "Test User",
  "email": "update1user202@example.com",
  "mobileNo": "9123456780",
  "qualification": "update MBBS"
}
```

Patient-Doctor Mapping

Base URL : http://localhost:5000/api/mappings

Endpoints

1. Assign Doctor to Patient

POST /api/mappings

Description:

Creates a mapping between a doctor and a patient by an admin or user.

Request Body:

```
{
  "patientId": 1,
  "doctorId": 2,
  "assignedBy": 3
}
```

Note: make sure you first create patient , doctor who has id 1 , 2 respectively and user who has id 3

Response:

- **201 Created**

```
{
  "id": 1,
  "patientId": 1,
  "doctorId": 2,
  "assignedBy": 3,
  "assignedDateTime": "2025-06-06T12:00:00.000Z"
}
```

Errors:

- **400 Bad Request** – Missing fields or foreign keys not found.
- **500 Internal Server Error** – Database or server issues.

2. Get All Mappings

GET /api/mappings

Description:

Retrieves a list of all doctor-patient mappings including related patient, doctor, and assigned user details.

Response:

- **200 OK**

```
[
  {
    "id": 1,
    "assignedDateTime": "2025-06-06T12:00:00.000Z",
    "Patient": {
      "patientId": 1,
      "patientName": "John Doe"
    },
    "Doctor": {
      "doctorId": 2,
      "doctorName": "Dr. Smith"
    },
    "AssignedByUser": {
      "userID": 3,
      "name": "Admin"
    }
  }
]
```

3. Get Mappings by Patient ID

GET /api/mappings/:patientId

Description:

Fetches all mappings for a specific patient.

Path Parameters:

- :patientId – ID of the patient.

Response:

- **200 OK**

```
[
  {
```

```
  "id": 1,
  "doctorId": 2,
  "assignedBy": 3,
  "Doctor": {
   "doctorId": 2,
   "doctorName": "Dr. Smith"
  },
  "AssignedByUser": {
   "userID": 3,
   "name": "Admin"
  }
 }
]
```

4. Delete Mapping

DELETE /api/mappings/:id

Description:

Deletes a specific doctor-patient mapping.

Path Parameters:

- :id – ID of the mapping record.

Response:

- **200 OK**

```
{
 "message": "Mapping deleted successfully"
}
```

404 Not Found – Mapping ID not found.