

Hospital Management System

The Hospital Management System streamlines hospital operations by managing patient records, doctor assignments, appointments, billing, and room allocations. It facilitates smooth communication between doctors and patients while keeping accurate medical and financial records. Built with Spring Boot, JPA, and Hibernate, the system implements reliable database relationships—one-to-many, one-to-one, and many-to-many—between core entities such as Patients, Doctors, Appointments, Billing, and Rooms. This solution automates hospital workflows, cuts down on administrative work, and enhances patient care management.

Git token: ghp_vGSxuC1jPVWSOmV01zHBAs36kmzP81OijSA

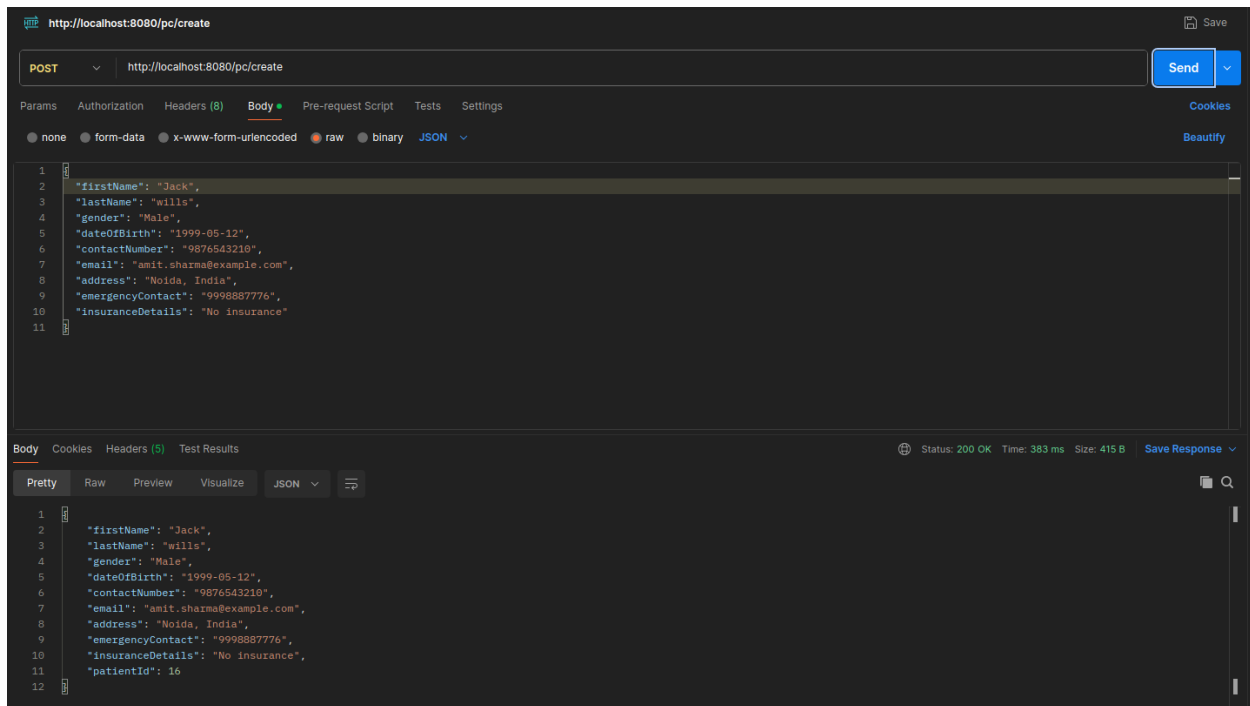
```
ganeshkarande@administrator-Latitude-3450:~/Desktop/Start/SpringBoot/Final Project$ git push -u origin master
Enumerating objects: 70, done.
Counting objects: 100% (70/70), done.
Delta compression using up to 12 threads
Compressing objects: 100% (57/57), done.
Writing objects: 100% (70/70), 25.49 KiB | 1.96 MiB/s, done.
Total 70 (delta 9), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (9/9), done.
To https://github.com/GaneshNITT/Project.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
ganeshkarande@administrator-Latitude-3450:~/Desktop/Start/SpringBoot/Final Project$
```

The screenshot displays the GitHub interface for a repository named 'Hospital Management System' by user 'GaneshKarande45'. The repository is located at 'https://github.com/GaneshNITT/Project.git'. The commit history for the 'master' branch is shown, listing several files that were committed 3 minutes ago with the message 'this is my project'. The files listed are: .gitignore, Hospital Management System.iml, compiler.xml, encodings.xml, jarRepositories.xml, misc.xml, and modules.xml. The interface also shows a sidebar with navigation options like Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings.

Name	Last commit message	Last commit da...
...		
.gitignore	this is my project	3 minutes ago
Hospital Management System.iml	this is my project	3 minutes ago
compiler.xml	this is my project	3 minutes ago
encodings.xml	this is my project	3 minutes ago
jarRepositories.xml	this is my project	3 minutes ago
misc.xml	this is my project	3 minutes ago
modules.xml	this is my project	3 minutes ago

Post man request and response status

1) Create the Patient

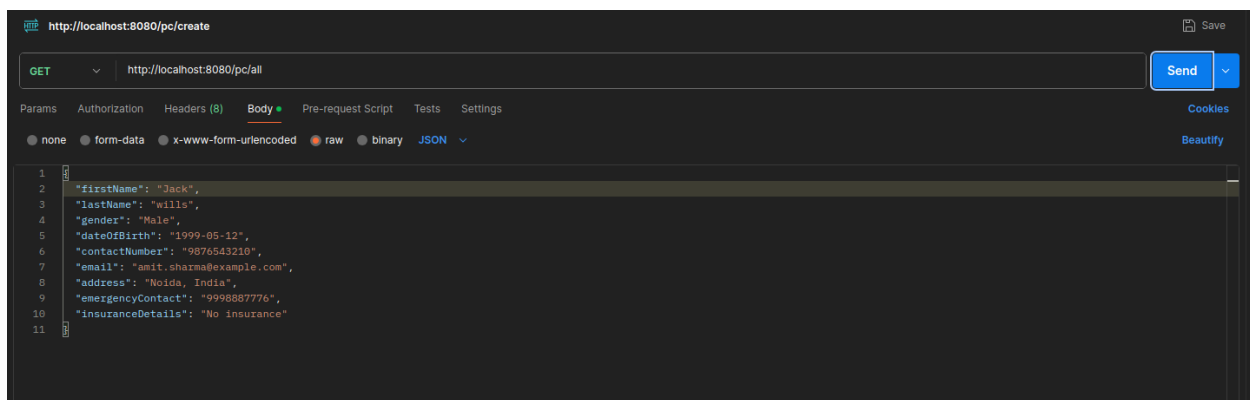


2. Get List of All Patients

Request: GET `/api/patients`

Response: 200 OK

<http://localhost:8080/pc/all>



```
[
  {
    "id": 1,
    "name": "John Smith",
    "age": 45,
    "gender": "Male",
    "contactNumber": "123-456-7890",
    "address": "123 Main St, City",
    "medicalHistory": "Hypertension",
    "appointmentDate": "2025-02-10"
  },
  {
    "id": 2,
    "name": "Sarah Johnson",
    "age": 32,
    "gender": "Female",
    "contactNumber": "987-654-3210",
    "address": "456 Oak Ave, Town",
    "medicalHistory": "Asthma",
    "appointmentDate": "2025-02-15"
  }
]
```

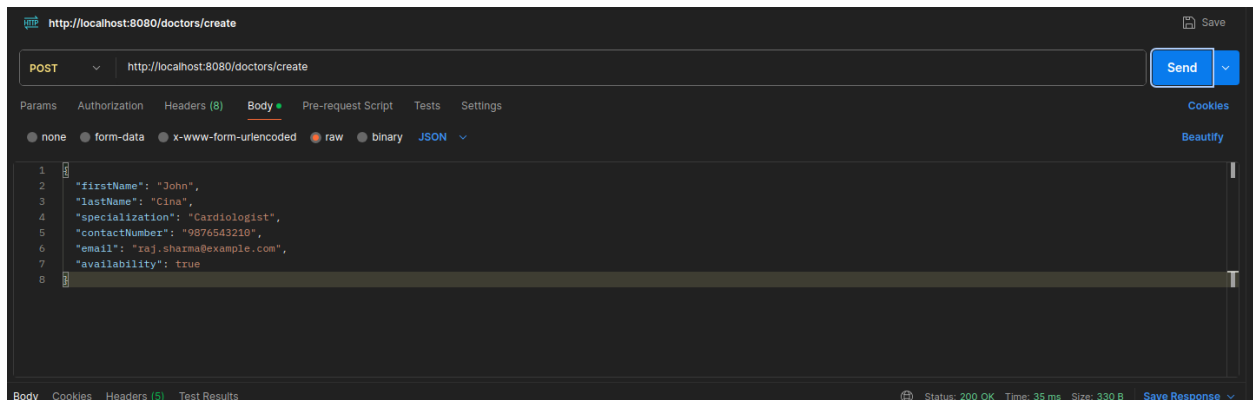
Response shows the list of all patients in the system with their details including ID, name, age, gender, contact information, address, medical history, and appointment dates.

3. Create Doctor

Request: POST /api/doctors

Response: 201 Created

Response shows the successful creation of a new doctor in the system with their details including ID, name, specialization, experience, contact information, email, availability, and qualifications.



1. Book Appointment

Request: POST /api/appointments

Response: 201 Created

```
{
  "id": 1,
  "patientId": 1,
  "doctorId": 1,
  "appointmentDate": "2025-02-20",
  "appointmentTime": "10:30",
  "status": "SCHEDULED",
  "reason": "Regular checkup",
  "notes": "First time visit"
}
```

Response shows the successful creation of a new appointment with details including appointment ID, patient ID, doctor ID, scheduled date and time, status, reason for visit, and any additional notes.

4. Request: GET /api/appointments

Response: 200 OK

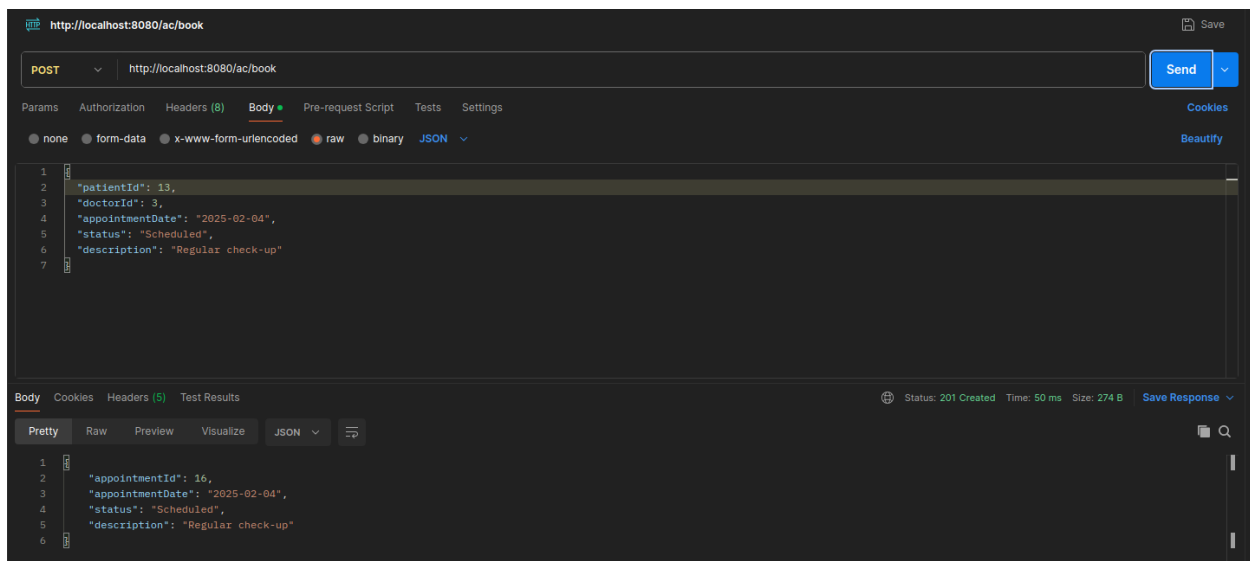
```
[
  {
    "id": 1,
    "patientId": 1,
```

```

        "doctorId": 1,
        "appointmentDate": "2025-02-20",
        "appointmentTime": "10:30",
        "status": "SCHEDULED",
        "reason": "Regular checkup",
        "notes": "First time visit"
    },
    {
        "id": 2,
        "patientId": 2,
        "doctorId": 1,
        "appointmentDate": "2025-02-21",
        "appointmentTime": "14:00",
        "status": "SCHEDULED",
        "reason": "Follow-up",
        "notes": "Review previous treatment"
    }
]

```

Response returns a list of all appointments in the system with their complete details including scheduling information and status.



Request: GET `/api/bills/patient/{patientId}`

Response: 200 OK

```
{
  "billId": 1,
  "patientId": 1,
  "patientName": "John Smith",
  "items": [
    {
      "description": "Consultation Fee",
      "amount": 150.00
    },
    {
      "description": "Lab Tests",
      "amount": 300.00
    },
    {
      "description": "Medications",
      "amount": 125.50
    }
  ],
  "totalAmount": 575.50,
  "billDate": "2025-02-04",
  "status": "PENDING",
  "paymentMethod": "Not specified"
}
```

Response shows the detailed bill for a patient, including itemized charges, total amount, bill date, and payment status. The bill includes various medical services such as consultation fees, laboratory tests, and prescribed medications.

http://localhost:8080/pc/total/13

Save

GET

http://localhost:8080/pc/total/15

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

Key	Value	Bulk Edit
Key	Value	

Body

Cookies

Headers (5)

Test Results

Status: 200 OK Time: 104 ms Size: 171 B Save Response

Pretty

Raw

Preview

Visualize

JSON

1 16000.0