

# MANUAL TESTING DOCUMENT

## Smart Doctor Appointment Booking App

### 1. Project Overview

**Project Name:** Smart Doctor Appointment Booking App

**Technology Stack:**

- Backend: Node.js, Express.js
- Database: MongoDB
- Authentication: JWT
- Geo-Location Feature: MongoDB 2dsphere
- Testing Type: Manual Testing + API Testing

**Actors:**

- Patient
- Doctor

### 2. Objective of Manual Testing

- Verify core application workflows
- Identify functional & edge-case issues
- Ensure smooth user experience
- Validate API behavior and data consistency
- Confirm role-based access control

### 3. Scope of Testing

**In Scope**

- User Registration & Login
- Role-based Authorization
- Doctor Profile Creation
- Patient Profile Creation
- Nearby Doctor Search (Geo Location)

- Doctor Availability (Slots)
- Appointment Booking
- View Appointments (Doctor & Patient)
- Appointment cancellation by patient
- Slot availability update after cancellation

#### **4. Entry & Exit Criteria**

##### **Entry Criteria**

- Backend server running
- Database connected
- APIs accessible via Postman

##### **Exit Criteria**

- All critical test cases executed
- Major bugs identified & documented
- Core workflows verified

#### **5. Test Data**

##### **Patient Sample**

```
{  
  "role": "PATIENT",  
  "name": "Anjali Gupta",  
  "email": "anjali@test.com",  
  "password": "123456",  
  "mobileNumber": "9222222222"  
}
```

##### **Doctor Sample**

```
{  
  "role": "DOCTOR",  
  "name": "Dr Neha Sharma",
```

```
"email": "neha@test.com",  
"password": "123456",  
"mobileNumber": "9999999999",  
"speciality": "cardiologist",  
"fee": 600,  
"city": "Delhi",  
"address": "Sarita Vihar"  
}
```

## 6. Manual Test Cases

### AUTHENTICATION MODULE

#### Test Case-AUTH-01

Field	Value
Test Case	TC-AUTH-01
Scenario	Register Patient
Steps	Send POST /api/auth/register
Input	Valid patient data
Expected Result	201 Created
Status	Pass

### **Test Case-AUTH-02**

<b>Field</b>	<b>Value</b>
Test Case ID	TC-AUTH-02
Scenario	Register Doctor
Steps	Send POST /api/auth/register
Input	Valid doctor data
Expected Result	Doctor registered + location stored
Status	Pass

### **Test Case-AUTH-03**

<b>Field</b>	<b>Value</b>
Test Case ID	TC-AUTH-03
Scenario	Login User
Steps	POST /api/auth/login
Expected	JWT Token returned
Status	Pass

## **NEARBY DOCTOR SEARCH**

### **Test Case -01**

<b>Field</b>	<b>Value</b>
Scenario	Find nearby doctors
API	GET /api/doctors/nearby
Params	lat, lng, distance
Expected	Doctor list returned
Status	Pass

### **Test Case -02**

Scenario - Doctor without coordinates

Expected - Doctor excluded from search

Status - Pass

### **Sorting Doctor's Explicitly**

**Test Case: Doctor list sorted by distance**

Field	Value
-------	-------

Scenario	Verify doctor list sorted by nearest distance
----------	---

Expected	Doctors returned with distance field
----------	--------------------------------------

Status	Pass
--------	------

### **DOCTOR AVAILABILITY**

#### **Test Case-01**

Scenario - Add doctor availability

API - POST /api/availability/add

Role - Doctor

Expected - Slots saved

Status - Pass

#### **Test Case-02**

Scenario - Fetch availability

API - GET /api/availability

Expected - Slots or empty object

Status - Pass

## **APPOINTMENT BOOKING**

### **Test Case -01**

Scenario - Patient books appointment

API - POST /api/appointments/book

Role - Patient

Expected - 201 Created

Status - Pass

### **Test Case-02**

Scenario - Unauthorized booking

Role - Doctor

Expected - 401 Unauthorized

Status - Pass

## **VIEW APPOINTMENTS**

### **Test Case-VIEW-01**

Scenario - Patient views bookings

API - GET /api/appointments/my

Expected - Appointment list

Status - Pass

### **Test Case-VIEW-02**

Scenario - Doctor views bookings

API - GET /api/appointments/doctor

Expected - Appointment list

Status - Pass

## **Appointment Cancellation**

### **Test Case- 01: Cancel appointment successfully**

Field	Value
Scenario	Patient cancels a booked appointment
Precondition	Patient has an existing BOOKED appointment
Steps	1. Login as patient 2. Call cancel appointment API 3. Fetch patient bookings
Expected Result	Appointment status updated to CANCELLED
Status	Pass

### **Test Case - 02: Slot becomes available after cancellation**

Field	Value
Scenario	Slot is freed after appointment cancellation
Precondition	Appointment exists and slot is booked
Steps	1. Cancel appointment 2. Fetch doctor availability
Expected Result	Slot isBooked is false
Status	Pass

### **Test Case -03: Rebooking same slot after cancellation**

Field	Value
Scenario	Another patient can book the same slot after cancellation
Precondition	Slot was previously booked and then cancelled
Steps	1. Cancel appointment 2. Book same slot again
Expected Result	Appointment booking succeeds

Field	Value
Status	Pass

#### **Test Case -04: Cancel appointment already cancelled**

Field	Value
Scenario	Prevent cancelling an already cancelled appointment
Precondition	Appointment status is CANCELLED
Steps	1. Call cancel API again
Expected Result	Error message “Appointment already cancelled”
Status	Pass

### **7. Negative Test Cases**

Case	Description	Expected
Invalid JWT	Access protected API 401	
Duplicate booking	Same slot twice	400
Missing coordinates	Geo search	Excluded
Wrong role access	Doctor books	Blocked

### **8. Bugs Identified (During Testing)**

Bug ID	Description	Status
BUG-01	Case mismatch in speciality	Fixed
BUG-02	Patients without location broke index	Fixed
BUG-03	Availability API returned array/object mismatch	Fixed



## **9. Conclusion**

- All major workflows tested manually
- System behaves as expected
- Role-based access enforced
- Geo-location search works correctly
- Appointment booking reliable
- The system supports appointment cancellation by patients.

**# Project is stable and ready for evaluation**

### **Testing Approach Used**

- Manual API Testing
- Role-based validation
- Edge-case coverage