MediTech Test Cases Summary

Test Coverage Summary

Module	Test Type	Description
User Authentication	Unit + Integration	Registration, login, JWT verification
Doctor Management	Integration	Add, update, delete, and list doctors
Appointment Booking	Unit + Integration	Book, cancel appointments
Payment Gateway	Integration	Test Razorpay flow (test mode)
Role-Based Access	Unit	Access control for Admin, Doctor, Patient
Database	Unit	MongoDB CRUD and schema validation
API Endpoints	Integration	REST API response validation
UI Tests (Optional)	UI/Functional	Form validation, navigation, and responsiveness

User Authentication Tests

Test ID	Test Case	Steps	Expected Result
UA-01	Register new user	Send POST request to `/api/user/register` with valid data	Status 201 — “User registered successfully”
UA-02	Duplicate registration	Register same email twice	Status 400 — “User already exists”
UA-03	Login valid user	POST `/api/user/login` with correct credentials	Returns valid JWT token
UA-04	Login invalid password	POST `/api/user/login` with wrong password	Status 401 — “Invalid credentials”
UA-05	Token verification	Access protected route with token	Returns decoded user data

Doctor Management Tests

Test ID	Test Case	Steps	Expected Result
DM-01	Add doctor	Admin sends POST `/api/admin/add-doctor`	Status 201 — “Doctor added successfully”
DM-02	Update doctor profile	PUT `/api/doctor/update-profile`	Status 200 — Updated doctor data returned
DM-03	Get all doctors	GET `/api/admin/all-doctors`	Returns list of all doctors
DM-04	Unauthorized update	Doctor tries to update another doctor’s profile	Status 403 — “Access denied”

Appointment Booking Tests

Test ID	Test Case	Steps	Expected Result
AP-01	Book appointment	POST `/api/user/book-appointment` with valid data	Status 201 — Appointment created successfully
AP-02	Book overlapping slot	POST with same date/time slot	Status 409 — “Slot already booked”
AP-03	Cancel appointment	DELETE `/api/user/cancel-appointment`	Status 200 — “Appointment cancelled successfully”
AP-04	Fetch patient appointments	GET `/api/admin/appointments`	Returns list of patient appointments
AP-05	Unauthorized booking	No token provided in header	Status 401 — “Unauthorized”

Role-Based Access Tests

Test ID	Test Case	Steps	Expected Result
RB-01	Patient access admin route	Access `/api/admin/dashboard` as patient	Status 403 — “Access denied”
RB-02	Doctor access patient data	Access `/api/user/get-profile` as doctor	Status 403 — “Access denied”
RB-03	Admin full access	Access all routes as admin	Status 200 — Access granted successfully

Database Tests

Test ID	Test Case	Steps	Expected Result
DB-01	Create user document	Save user model to database	Data saved successfully
DB-02	Invalid schema	Save data with missing required fields	Validation error occurs
DB-03	Delete doctor record	Remove document by ID	Record deleted successfully

Testing Approach (Manual)

All functionalities were manually tested using **Postman** for backend APIs and **browser testing** for frontend.

Tools Used:

- Postman – For API validation
- MongoDB Compass – For verifying data storage
- Browser DevTools – For checking UI and API responses

Validation Checklist:

- API responses and HTTP codes verified
- Proper CRUD operations confirmed
- Role-based access tested
- Payment gateway tested in test mode
- UI forms validated for errors and success messages