📄 PERFORMANCE TESTING – DocSpot

👥 Team Details

Field	Details
Team ID	LTVIP2025TMID30950
Team Size	2
Team Leader	Lakshmi Sravya Savaram
Team Member	Pathela Praveen Chakravarthi

1️⃣ Objective of Performance Testing

The goal of performance testing in the **DocSpot** project is to ensure:

- Fast response times
- Stable and scalable behavior under real-world conditions
- Efficient resource usage across API endpoints

2️⃣ Tools Used

Tool	Purpose
Postman / Thunder Client	API testing and load simulation
Browser DevTools	Network and performance analysis
Lighthouse	Frontend speed & optimization check
MongoDB Atlas Monitor	DB performance and query time tracking

3️⃣ Test Scenarios Conducted

Test Case	Method	Expected Result	Status
User registration/login	POST /api/auth	<2s response, JWT token issued	✅ Pass
Fetching doctor list	GET /api/doctors	<1.5s for 10+ records	✅ Pass
Booking appointment	POST /api/appointments	Slot reserved, confirmation returned	✅ Pass
Handling multiple users booking	Simulated with Thunder	No collision, DB handles simultaneous requests	✅ Pass
Load test: 50 parallel requests	Scripted in Postman	System remains stable, <3s per call	✅ Pass
Page load time	Lighthouse / Chrome DevTools	Score > 90% for performance	✅ Pass

---

4️⃣ Observations

- ⚡ **API Responses** were consistently under 2 seconds under moderate load.
- 🧠 **MongoDB Indexing** helped fetch doctor availability and booking records fast.
- 🧱 **Frontend Optimization**: Code-splitting with React reduced first-load delays.
- 🔐 **JWT Verification** remained efficient even with 50+ simultaneous token validations.

5️⃣ Improvements Suggested

Area	Recommendation
Rate Limiting	Add middleware like express-rate-limit to prevent API abuse
Caching	Use Redis or in-memory caching for doctor availability
Logging	Implement centralized logging for performance errors
Async Queuing	Introduce job queue for email notifications