PERFORMANCE TESTING

6.1Performance Testing

Performance testing is a critical aspect of evaluating the LearnHub platform to ensure that it can handle real-world usage scenarios effectively. This section outlines the methods and observations from testing the performance of core functionalities such as user login, course browsing, enrollment, and video playback.

Performance Testing Objectives:

- To ensure that the system can handle concurrent users logging in and browsing courses without latency.
- To verify the response time of major functionalities such as course creation, section upload, and assignment to students.
- To assess the stability of video streaming and course playback across devices.
- To evaluate how the system performs under load and during peak activity.
-

Tools Used:

- Chrome DevTools for analyzing network performance.
- Postman for API response testing.
- Browser-based Lighthouse audit for performance score.
- MongoDB Atlas monitoring dashboard.

Test Scenarios and Results:

Test Case	**Scenario**	**Expected Result**	**Actual Result**	**Status**
TC-PT-01	Login with valid credentials under 20 users	Login in < 1.5s	Avg. Login: 1.2s	✅ Pass
TC-PT-02	Browse course categories	Load within 2s	Avg. Load: 1.6s	✅ Pass
TC-PT-03	Stream embedded YouTube video	Video buffers within 3s	Avg. Buffer: 1.8s	✅ Pass
TC-PT-04	Assign course to 50 students	Server responds within 2s	Response: 1.7s	✅ Pass
TC-PT-05	Simulate 100 concurrent API calls to /courses	Maintain server stability	Minor lag after 80 users	⚠️ Partial

Observations:

- The platform performs well for typical user volumes (under 50 concurrent users).
- MongoDB Atlas handled the read/write load without timeouts.
- Course video streaming using iframe embeds maintained consistent

performance.
- Under high load (100 concurrent requests), slight delays were observed but did not crash the server.

Recommendations:

- Implement server-side caching using Redis for frequent API calls like /courses and /browse.
- Optimize image loading using lazy loading techniques for thumbnails and banner images.
- Deploy a CDN for static assets to enhance response time.
- Consider horizontal scaling or using a cloud-based load balancer for production deployment.