

NIRMA UNIVERSITY

Institute:	Institute of Technology, School of Technology
Name of Programme:	BTech CSE
Course Code:	4CS208ME25
Course Title:	Web Performance Optimization
Course Type:	Department Elective-IV
Year of Introduction:	2025-26

L	T	Practical Component				C
		LPW	PW	W	S	
3	-	2	-	-	-	4

Course Learning Outcomes (CLO):

At the end of the course, the students will be able to –

1. summarise the importance of Web Performance (BL2)
2. apply Optimization Techniques to improve the web performance (BL3)
3. analyse the web performance data using tools and metrics (BL4)
4. create a web performance improvement plan, implementing various optimization techniques. (BL6)

Unit	Contents	Teaching Hours (Total 45)
Unit-I	Introduction to Web Performance Optimization: Importance of web performance, Key performance metrics, Understanding and interpreting performance metrics, Tools for measuring performance	05
Unit-II	Optimizing Front-End Code: Reducing and optimizing HTML, CSS, and JavaScript Minification and compression techniques	07
Unit-III	Images and Multimedia Optimization: Lazy loading images and videos, implementing image formats (WebP, AVIF), Network and Server Optimization: Reducing server response times, Implementing HTTP/2 and HTTP/3, Content delivery networks (CDNs)	10
Unit-IV	Responsive Design and Mobile Optimization: Strategies for responsive web design, Mobile-first development, Touch and gesture optimizations	08
Unit-V	Web Performance Testing and Monitoring: Automated testing tools and continuous integration, Real user monitoring (RUM) and synthetic monitoring	08
Unit-VI	Web Performance in Modern Web Development: Progressive Web Apps (PWAs), Serverless architecture and JAMstack, Optimizing single-page applications (SPAs), Introduction to Web 3.0	07

Self-Study:

The self-study contents will be declared at the commencement of the semester. Around 10% of the questions will be asked from self-study contents

Suggested Readings/ References:

1. Sufyan bin Uzayr, *Web Performance Optimization: A Practical Approach*, CRC Press
2. Steve Souders, *Even Faster Web Sites: Performance Best Practices for Web Developers*, O'Reilly
3. Nicholas C. Zakas, *High Performance JavaScript – Build Faster Web Application Interfaces*, O'Reilly
4. Steve Souders, *High Performance Web Sites: Essential Knowledge for Front-End Engineers*, O'Reilly
5. Tom Barker, *High Performance Responsive Design: Building Faster Sites Across Devices*, O'Reilly
6. Shailesh Kumar and Shivakumar, *Modern Web Performance Optimization: Methods, Tools, and Patterns to Speed Up Digital Platforms*, Apress

Suggested List of Experiments:

Sr. No.	Name of Experiments/Exercises	Hours
1	a. Analyze the performance of a website using tools like Google PageSpeed Insights and WebPageTest. b. Select two or more websites with different performance levels and compare their loading times, user experience, and business impact.	04
2	a. Take a sample HTML, CSS, or JavaScript code and apply minification techniques to reduce file sizes. Measure the impact on loading times. b. Combine multiple CSS and JavaScript files into single files. Assess the effects on performance and loading times.	04
3	Implement lazy loading for images and videos on a webpage. Measure the reduction in initial page load time.	02
4	Convert images to modern formats like WebP or AVIF. Analyze the benefits in terms of improved performance.	02
5	a. Optimize server response times by identifying and addressing server bottlenecks or code inefficiencies. b. Configure a server to support HTTP/2 and HTTP/3. Measure the impact on loading times and security.	04
6	Set up a Content Delivery Network (CDN) for a website and compare loading times before and after CDN implementation	04
7	Develop a mobile-first version of a website or webpage. Compare performance metrics with the desktop version.	02
8	Implement touch and gesture optimizations for a website. Test the improvements on mobile devices.	02
9	Set up automated performance tests using tools like Jenkins or Travis CI. Monitor performance changes with each code update.	04
10	Implement prefetching of resources like fonts and critical images to speed up page loading.	02