Please go through the link for html report for performance testing
Performance testing

1. Improve First Contentful Paint (FCP) and Largest Contentful Paint (LCP)

- **Optimize Images**: Ensure images are appropriately sized and compressed. Use modern image formats like WebP.
- **Preload Critical Resources**: Use `<link rel="preload">` to prioritize fetching critical resources.
- **Inline Critical CSS**: Inline the CSS required for above-the-fold content and defer the rest.
- **Minimize Render-Blocking Resources**: Defer non-critical CSS and JavaScript files to improve loading times.

2. Reduce JavaScript Execution Time

- **Code Splitting**: Split your JavaScript code into smaller chunks to avoid loading unnecessary code on the initial page load.
- **Minify JavaScript**: Minify and compress your JavaScript files to reduce their size.
- **Remove Unused JavaScript**: Identify and eliminate unused JavaScript to reduce the amount of code the browser needs to execute.

3. Improve Time to Interactive (TTI)

- **Lazy Load Resources**: Use lazy loading for non-critical resources such as images and videos.
- **Optimize Third-Party Scripts**: Limit the impact of third-party scripts by deferring their loading or using async.
- **Reduce Main-Thread Work**: Optimize your JavaScript to reduce the amount of work done on the main thread.

4. Reduce Total Blocking Time (TBT)

- **Minimize Long Tasks**: Break up long-running tasks to improve responsiveness.
- **Use Web Workers**: Offload intensive computations to web workers to free up the main thread.

5. Minimize Cumulative Layout Shift (CLS)

- **Set Size Attributes for Media**: Define width and height for all images and videos to prevent layout shifts.
- **Use CSS Aspect Ratio Boxes**: For responsive images and videos, use aspect ratio boxes.

- **Avoid Inserting Content Above Existing Content**: Ensure new content added to the page does not push existing content down.

6. Optimize Server Response Time

- **Enable Compression**: Use GZIP or Brotli compression for text-based resources.
- **Optimize Server Performance**: Improve your server's response time by optimizing database queries, using faster server hardware, or leveraging CDNs.
- **Use a Content Delivery Network (CDN)**: Distribute content closer to users to reduce latency.

7. Optimize Resource Loading

- **Use HTTP/2**: Take advantage of HTTP/2 for multiplexing requests and reducing latency.
- **Cache Assets**: Implement proper caching strategies to ensure that repeated visits are faster.
- **Prefetch Resources**: Use `<link rel="prefetch">` to prefetch resources that might be needed soon.

8. Improve Accessibility

- **Ensure Color Contrast**: Make sure text has enough contrast against its background.
- **Label All Inputs**: Ensure that all form inputs have associated labels for accessibility.
- **Use ARIA Roles**: Appropriately use ARIA roles and properties to enhance the accessibility of interactive elements.