Software Requirement Specification

Version 1.0 September 28, 2025

Success Ladder Technologies (SLT) Pvt. Ltd.

48, Moti Doongri, Alwar, Rajasthan 301001
01442980098

Software Requirements Specification

Success Ladder Technologies (SLT) Pvt. Ltd.

Audit and Page Performance Analyzer System.

Version 1.0.

SR. No.	Name	Role
1	Piyush	Full Stack
2	Dheeraj	Full Stack
3	Mayank	Full Stack

Supervisor : Aditya Soni
Organization:- Success Leader Technologies

1. Introduction

Purpose of the Project:

The primary purpose of the Site Audit and Page Performance Analyzer System is to provide a comprehensive tool that evaluates websites across multiple dimensions, including performance, SEO, accessibility, security, and user experience. In today's digital landscape, slow or poorly optimized websites can negatively affect user engagement, search engine rankings, and conversion rates. This system aims to bridge that gap by offering actionable insights in a user-friendly interface, enabling developers, SEO specialists, and business owners to quickly identify and resolve issues. The tool also establishes a foundation for future enhancements such as AI-driven recommendations, exportable reports, and multilanguage support.

Scope:

The system is a web-based platform allowing users to input a website URL, which is then validated and analyzed using APIs like Google PageSpeed Insights. Core functionalities include performance metrics, SEO checks, accessibility verification, and conversion insights, displayed in categorized, interactive reports. Features not included in the initial release are user authentication, historical data tracking, advanced security scanning, and export options (PDF/CSV).

Definitions & Acronyms:

Key terms include LCP (Largest Contentful Paint), CLS (Cumulative Layout Shift), SEO (Search Engine Optimization), API (Application Programming Interface), UX/UI (User Experience/User Interface), and WCAG (Web Content Accessibility Guidelines). These definitions ensure clarity and consistent understanding among stakeholders.

2. Overall Description

Product Perspective:

The Site Audit and Page Performance Analyzer System is a **new, standalone webbased application** designed to consolidate multiple website evaluation functions into a single platform. Unlike existing tools that focus on individual aspects such as SEO, performance, or accessibility, this system offers a holistic approach, combining technical audits, SEO analysis, accessibility compliance, UX evaluation, and conversion insights. The modular client-server architecture, with a React-based frontend and Node.js backend, ensures scalability and future extensibility. The system interacts with external APIs like Google PageSpeed Insights for performance metrics, and its design allows for easy integration of additional analytics or AI-driven modules in subsequent versions.

User Needs and Characteristics:

The system is intended for a diverse range of users. Developers require precise performance metrics and technical recommendations. SEO specialists need insights into metadata, headings, and crawlability. Accessibility consultants focus on WCAG compliance, contrast ratios, and semantic HTML usage. Business owners and marketing teams prefer simplified dashboards and actionable insights to improve conversion rates. QA testers and project managers need standardized reports to verify site quality and track improvements. By addressing these varied needs, the system provides both technical depth and userfriendly summaries.

Assumptions and Dependencies:

The successful operation of the system assumes that users provide valid, publicly accessible URLs and have stable internet connectivity. Core dependencies include the Google PageSpeed Insights API for performance data, Node.js and Express for backend processing, React and TailwindCSS for frontend rendering, and cloud hosting services (AWS, Vercel, or Netlify) for deployment. The system also relies on modern browsers supporting ES6 and standard web protocols. Future enhancements may depend on additional APIs or database services for storing audit history.

3. Functional Requirements

Feature 1: URL Input and Validation Description:
This feature serves as the entry point for the system. Users provide a website URL that will be analyzed. The system validates the URL format to ensure it is properly structured and publicly accessible. Without valid input, no audit can proceed. This step prevents errors in downstream processes and improves user experience by providing immediate feedback on incorrect inputs.

Inputs:

- Website URL entered by the user (HTTP/HTTPS).
- Optional parameters such as audit type (Performance, SEO, Accessibility).

Outputs: • Confirmation of valid URL submission.

- Error messages for invalid or unreachable URLs, such as “Invalid URL format” or “Website not accessible.”
- Trigger signal to backend to initiate audits for valid URLs.

Feature 2: Performance Analysis Description:
The Performance Analysis feature evaluates the technical performance of a website using metrics such as Largest Contentful Paint (LCP), First Contentful Paint (FCP), Cumulative Layout Shift (CLS), and Total Blocking Time (TBT). Data is retrieved from Google PageSpeed Insights API and presented to users in an easy-to-understand format. Color-coded indicators (green, yellow, red) highlight the quality of each metric, while textual explanations provide actionable recommendations, such as image optimization or script deferment. This feature helps developers and site owners identify bottlenecks and improve user experience and SEO rankings.

Inputs:

- Validated website URL from Feature 1.
- API key for Google PageSpeed Insights (server-side).

Outputs:

- Structured performance metrics including LCP, FCP, CLS, TBT, and overall performance score.
- Visual representations such as progress bars or charts.
- Recommendations for optimization based on metric thresholds.
- Categorized audit card displaying metrics, explanations, and improvement suggestions.

Together, these features ensure that users can easily provide a website for analysis and immediately receive meaningful insights into its technical performance, forming the foundation for other audit modules such as SEO, accessibility, and conversion evaluation.

Frontend (React):

1. Display a form with:
    - A single input for the URL ○ A "Analyze" button
2. Validate the input (e.g., must be a valid URL format).
3. On form submission:
    - Send a POST request to the backend endpoint

4. Display:
    - Loading indicator while waiting for the response ○ On success: show performance score, load time, and other metrics ○ On error: show a meaningful error message (e.g., invalid URL, server error)

Backend (Node.js + Express):

1. Define a route: POST /data
2. Extract the URL from the query parameter.
3. Validate the URL.
4. Call an external page speed analysis API (e.g., Google PageSpeed Insights).
    - Endpoint: https://www.googleapis.com/pagespeedonline/v5/runPagespeed?url=<URL>&key=<API_KE Y>
5. Parse and return a simplified JSON response to the frontend (e.g., performance score, first contentful paint, time to interactive, etc.)
6. Handle errors appropriately (invalid URLs, external API issues, etc.)

4. Non-Functional Requirements

Performance:

The system must deliver audit results quickly and efficiently to ensure a positive user experience. For a single website audit, results should appear within 5–7 seconds, depending on API response times and network latency. The backend must be capable of handling at least 100 concurrent requests per minute during the MVP phase without degradation in performance. The system should optimize resource utilization to prevent excessive CPU or memory consumption on both client and server sides. Future versions should support horizontal scaling using cloud services to manage thousands of simultaneous requests, ensuring smooth operation under high traffic conditions. Visual feedback, such as real-time loaders and progress indicators, will help maintain perceived performance, letting users know the audit is actively running.

Security:

Since the system interacts with external networks and APIs, strong security measures are essential. All communications between client, server, and APIs must be encrypted using HTTPS (TLS 1.2 or higher). API keys and sensitive credentials must be securely stored on the server and never exposed to the client. Input validation must prevent malicious entries, including XSS (Cross-Site Scripting) and injection attacks. Rolebased access control may be implemented in future versions when user authentication is added, ensuring only authorized users can access sensitive functions. Although the MVP does not store personal data, future iterations must comply with GDPR and other privacy regulations if user information is introduced.

Reliability & Availability:

The system must provide consistent and dependable results. Audit reports should accurately reflect data retrieved from Google PageSpeed Insights and internal checks. Partial failures, such as unavailable metrics, should be clearly indicated to the user, while the rest of the report remains accessible. The system should aim for 99.5% uptime during the MVP phase. In case of server crashes, automated recovery mechanisms like container orchestration or server monitoring should restart services promptly, minimizing downtime. Future versions may include redundancy and load balancing to ensure continuous availability.

Usability:

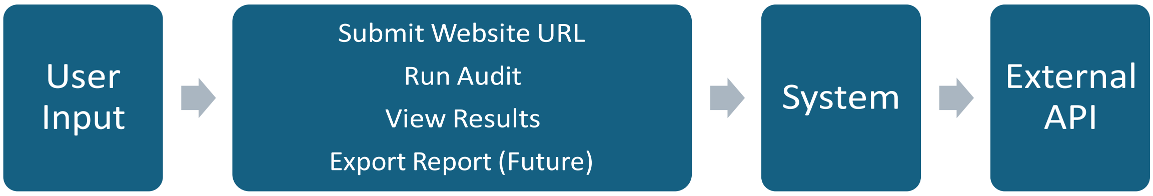
The system must be intuitive and accessible to both technical and non-technical users. A clean, darkthemed interface with responsive design ensures usability across devices, including desktops, tablets, and smartphones. Input forms, dashboards, and audit cards should be easy to navigate, with color-coded indicators and clear textual explanations. Accessibility standards (WCAG 2.1) must be followed, including keyboard navigation, proper contrast ratios, and screen reader compatibility. User feedback mechanisms, such as error messages, loaders, and notifications, will improve overall interaction and satisfaction.

5. System Models

Use Case Diagram:

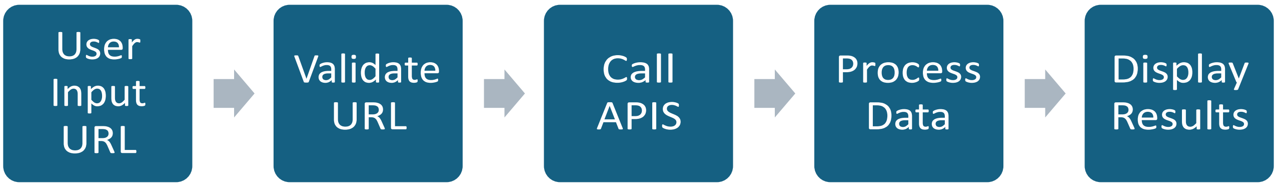
The use case diagram shows interactions between users and the system. Primary actors include **User**, **System**, and **External APIs** (e.g., Google PageSpeed Insights). The main use cases are: submit URL, run audit, view results, and export report (future enhancement). **Entity Relationship Diagram (ERD):**

The ERD illustrates data storage and relationships. Key entities include **User**, **Website Audit**, **Metrics**, and **Reports**. Each user can request multiple audits, each audit generates multiple metrics, and reports are linked to audits.



Flowchart / Data Flow:

The flow of data starts when the user submits a URL. The system validates it, fetches metrics from APIs, processes results, and displays categorized audit reports.



Wireframes:

The frontend interface includes:

- Fixed sidebar for navigation (Performance, SEO, Accessibility, UX, etc.)
- Main dashboard with categorized audit cards
- Loader for real-time feedback
- Dark-themed UI with color-coded indicators

Description:

These system models provide a visual understanding of system behavior and interactions. The use case diagram clarifies user-system interactions. The ERD outlines how data is structured and linked. Flowcharts depict step-by-step processing, while wireframes guide the UI/UX design. Together, these models ensure developers, testers, and stakeholders have a clear, shared understanding of the system's architecture and functionality.

6. Other Requirements

Legal and Compliance Requirements:

The Site Audit and Page Performance Analyzer System must operate within the boundaries of applicable laws and industry standards. Since it processes website information, it should comply with **data protection regulations** such as GDPR when user data or audit histories are stored in future versions. The system must respect **robots.txt guidelines** to avoid unauthorized crawling or accessing restricted content. Integration with third-party APIs, such as Google PageSpeed Insights, must follow the provider’s licensing terms, usage quotas, and data privacy policies. Accessibility compliance is also a legal and ethical requirement; the system adheres to **WCAG 2.1 guidelines**, ensuring inclusivity for users with disabilities. Any future expansion involving user accounts or personal data must include secure consent collection, encryption, and data retention policies consistent with local regulations.