Software Requirements Specification (SRS) for Web Accessibility Checker and Improver Tool

Name of Product : **AccessiScan**

1. Introduction

1.1 Purpose

The purpose of this document is to define the requirements for the development of a Web Accessibility Checker and Improver Tool. This tool aims to assist web developers in ensuring their websites comply with accessibility standards, specifically the Web Content Accessibility Guidelines (WCAG), thereby making the internet more inclusive and accessible to individuals with disabilities.

1.2 Scope

The software will provide automated checks, suggestions for improvements, educational resources, and real-time assistance to help developers create accessible websites. It will include features for checking compliance, generating reports, suggesting fixes, and monitoring accessibility over time.

1.3 Definitions, Acronyms, and Abbreviations

- **WCAG**: Web Content Accessibility Guidelines
- **API**: Application Programming Interface
- **IDE**: Integrated Development Environment
- **CI/CD**: Continuous Integration/Continuous Deployment
- **ARIA**: Accessible Rich Internet Applications
- **UI**: User Interface
- **UX**: User Experience

2. Overall Description

2.1 Product Perspective

The Web Accessibility Checker and Improver Tool will be a standalone web application that integrates with existing web development environments and tools. It will provide APIs for integration into CI/CD pipelines and support multiple browsers for accessibility testing.

2.2 Product Functions

- Automated scanning of web pages to identify accessibility issues.
- Evaluation of websites against WCAG standards (2.0, 2.1, and future versions).
- Generation of detailed accessibility reports categorized by severity.
- Provision of actionable suggestions and code snippets to fix accessibility issues.
- Real-time feedback and guidance during the development process.
- Educational resources including tutorials, articles, and videos on accessibility best practices.
- Compliance dashboard for monitoring accessibility status and improvements.

2.3 User Classes and Characteristics

- **Developers**: Users who build and maintain websites and web applications.
- **Accessibility Specialists**: Users who specialize in accessibility testing and compliance.
- **Administrators**: Users who manage settings, users, and API integrations within the tool.

2.4 Operating Environment

- The tool will operate as a web application accessible via modern web browsers (Chrome, Firefox, Safari, Edge).
- Backend services will be hosted on a scalable cloud platform (e.g., AWS, Azure) to ensure reliability and performance.

2.5 Design and Implementation Constraints

- The tool must comply with WCAG standards itself to ensure accessibility for users with disabilities.
- It should be scalable to handle large volumes of website scans and user interactions concurrently.
- Integration APIs should be well-documented and easy to implement for developers.

2.6 Assumptions and Dependencies

- Assumption: Users have basic knowledge of web development principles and WCAG guidelines.
- Dependency: Availability of third-party accessibility testing libraries and APIs for automated scanning.

3. Specific Requirements

3.1 Functional Requirements

1. **Accessibility Checking**
    - The tool shall scan web pages for accessibility issues automatically.
    - It shall support scanning of single web pages via URL entry or browser extension.
    - It shall support batch scanning of multiple pages for compliance checking.
2. **WCAG Compliance**
    - The tool shall evaluate websites against WCAG 2.0, 2.1, and upcoming versions.
    - It shall categorize accessibility issues by severity (e.g., critical, major, minor).
3. **Reporting and Suggestions**
    - It shall generate detailed accessibility reports with references to relevant WCAG guidelines.
    - It shall provide actionable suggestions and code snippets to fix identified accessibility issues.
4. **Real-time Assistance**
    - The tool shall provide real-time feedback on accessibility issues during code development.
    - It shall simulate screen reader interactions and keyboard navigation for dynamic elements.
5. **Educational Resources**

- It shall offer tutorials, articles, and videos on accessibility best practices.
- It shall integrate educational content within the tool interface for easy access.

6. **Compliance Dashboard**

   - The tool shall provide a dashboard to monitor accessibility scores and trends over time.
   - It shall allow customization of accessibility reports and dashboard views.

7. **API Integration**

   - It shall provide APIs for integration into CI/CD pipelines and development environments.
   - APIs shall allow access to scanning results, reports, and compliance data programmatically.

3.2 Non-Functional Requirements

1. **Performance**

   - The tool shall handle concurrent requests for scanning and reporting without significant delays.
   - Response times for scanning and generating reports shall be optimized for user experience.

2. **Security**

   - Data transmission and storage shall comply with industry standards for encryption and security.
   - User authentication and authorization mechanisms shall be implemented securely.

3. **Accessibility**

   - The tool's UI and features shall be accessible to users with disabilities, complying with WCAG 2.1 AA standards.

4. **Usability**

   - The UI shall be intuitive and user-friendly, with clear navigation and informative feedback.
   - Help documentation and tooltips shall be provided where necessary to aid user understanding.

5. **Scalability**

   - The tool shall be scalable to accommodate an increasing number of users and websites over time.
   - Backend services shall scale dynamically based on demand and usage patterns.

6. **Compatibility**

   - The tool shall be compatible with modern web browsers and accessible across different devices (desktop, tablet, mobile).

4. Technical Requirements

4.1 Hardware Requirements

- The tool will be accessible via standard web browsers and will not have specific hardware requirements beyond those necessary for typical web browsing.

4.2 Software Requirements

- **Backend**: Node.js, Express.js, MongoDB for database storage.
- **Frontend**: React.js for the user interface, Redux for state management, D3.js for data visualization in the compliance dashboard.
- **Accessibility Libraries**: Integration with axe-core for automated accessibility testing, Pa11y for continuous accessibility monitoring.
- **Development Tools**: ESLint for code quality, Jest and Enzyme for unit testing.

4.3 System Requirements

- The tool shall be compatible with the latest versions of major web browsers (Chrome, Firefox, Safari, Edge).
- Responsive design for usability across different devices (desktop, tablet, mobile).

4.4 Security Requirements

- Data transmission and storage shall be encrypted using industry-standard protocols (e.g., HTTPS, SSL/TLS).
- User authentication and authorization mechanisms shall be implemented to control access to sensitive information and administrative functions.

4.5 Performance Requirements

- The tool shall handle concurrent user sessions and scanning requests without significant performance degradation.
- Response times for scanning, reporting, and real-time feedback shall be optimized for user experience.

4.6 Compliance Requirements

- The tool's user interface and features shall comply with WCAG 2.1 AA standards to ensure accessibility for users with disabilities.
- Regular updates and compliance checks shall be conducted to align with evolving WCAG guidelines and industry best practices.

5. Conclusion

The Web Accessibility Checker and Improver Tool aims to address the critical need for accessible web content by providing developers with a comprehensive solution to evaluate, improve, and maintain accessibility standards on their websites. By adhering to the Web Content Accessibility Guidelines (WCAG) and integrating automated checks, real-time feedback, educational resources, and compliance monitoring, the tool aims to enhance inclusivity and usability for all users, regardless of their abilities. This document outlines the functional and non-functional requirements necessary for the successful development and deployment of the tool, ensuring it meets industry standards and user expectations.

6. Appendices

6.1 Glossary

- **WCAG**: Web Content Accessibility Guidelines, a set of guidelines developed by the World Wide Web Consortium (W3C) to ensure websites are accessible to people with disabilities.
- **API**: Application Programming Interface, a set of rules and protocols that allow different software applications to communicate with each other.
- **IDE**: Integrated Development Environment, a software application that provides comprehensive facilities to programmers for software development.
- **CI/CD**: Continuous Integration/Continuous Deployment, practices used by software development teams to automate the process of testing and deploying code.
- **ARIA**: Accessible Rich Internet Applications, a set of attributes that define ways to make web content and applications more accessible to people with disabilities.

6.2 References

- Web Content Accessibility Guidelines (WCAG) 2.0 and 2.1 - W3C Recommendation
- Section 508 of the Rehabilitation Act (United States)
- Americans with Disabilities Act (ADA) - Department of Justice (United States)
- Pa11y Accessibility Testing Tool
- axe-core Accessibility Testing Library

Factors impact the accessibility of a website.

These factors influence how easily people with disabilities can perceive, navigate, and interact with web content. Here are key elements that significantly affect accessibility:

1. **Text Alternatives (Alt Text)**:
    - Images, graphs, and charts should have descriptive alternative text (alt text) that conveys the same meaning or function as the visual content. This helps people who are blind or have low vision understand the content through screen readers.
2. **Semantic HTML**:
    - Proper use of HTML elements (like headings, lists, and links) ensures the structure of the content is clear and consistent. Screen readers rely on these elements to navigate and present content accurately.
3. **Keyboard Accessibility**:
    - All functionality and interactive elements should be operable using a keyboard alone, without requiring a mouse. This includes navigation menus, form controls, and media players.
4. **Color Contrast**:
    - Text and interactive elements should have sufficient color contrast against their background to ensure readability for users with low vision or color blindness.

5. **Video and Audio Accessibility**:
    - Videos should include captions or transcripts to make the content accessible to individuals who are deaf or hard of hearing. Audio content should have text alternatives or transcripts.
6. **Navigational Clarity**:
    - Navigation menus and links should be clear and consistent throughout the website. Logical navigation helps all users, especially those using screen readers or navigating with keyboard shortcuts.
7. **Forms and Interactive Elements**:
    - Forms should be accessible with clear labels, error messages, and instructions. Interactive elements like buttons and dropdowns should be accessible via keyboard and provide clear feedback.
8. **Readable Text and Fonts**:
    - Text should be resizable without losing functionality or readability. Fonts should be legible, and the layout should not require users to zoom in excessively to read content.
9. **Adaptive and Responsive Design**:
    - Websites should be responsive, adapting to different screen sizes and devices. Adaptive design ensures content remains accessible and usable across various platforms and contexts.
10. **Accessibility of Third-Party Content and Plugins**:
    - Content embedded from third-party sources (like social media widgets or plugins) should also be accessible. Developers should ensure these components comply with accessibility standards.
11. **Consistent Page Structure**:
    - Consistent page layouts and structures help users understand and navigate content more easily. This includes predictable placement of navigation bars, headers, and content sections.
12. **Accessibility Testing and Compliance**:
    - Regular accessibility audits and testing are essential to identify and address accessibility barriers. Compliance with standards like WCAG (Web Content Accessibility Guidelines) ensures websites meet recognized accessibility criteria.