UNIVERSITY OF ENGINEERING AND TECHNOLOGY, PESHAWAR PAKISTAN

Main Campus

Software Engineering Lab

Assignment 5

Name:	**Muhammad Mohsin**
Registration No.	**23PWBCS0973**
Semester:	**BS CS 5th**
Section:	**A**

Submitted To : Miss Kanwal Aneeq

DEPARTMENT OF COMPUTER SCIENCE & IT

UNIVERSITY OF ENGINEERING AND TECHNOLOGY, PESHAWAR, PAKISTAN

Software Requirements Specification (SRS)

Project Title: Website Accessibility Simulator
Team Members / Roll Numbers:

- Abdul Baseer,
- Muhammad Mohsin,
- Hooriya Altaf,
- Saad Abdullah

Instructor: Miss Kanwal Aneeq
Date: 10/25/2025

1. Introduction

1.1 Purpose

The purpose of this document is to define in detail the requirement specifications for the "Website Accessibility Simulator" project. This document will ensure that the entire team is on the same page and any ambiguity in specifications during the requirement elicitation process is removed. Thus the document will help the developers in maintaining a smooth development workflow and avoiding errors due to ambiguities encountered in the requirement elicitation process.

1.2 Scope

The project proposes a tool that scans a given website to check for accessibility requirements in accordance with the **Web Content Accessibility Guidelines (WCAG 2.1 AA)**.
It scans the website for accessibility issues, simulates different impairments (such as color blindness or low vision), and generates detailed reports highlighting violations.
The systems goal is to help testers in their audits to find accessibility issues, it is not meant to replace manual audits, thus the reason for inclusion of simulated disabilities for manual audits.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
UI	User Interface
UX	User Experience
WCAG	Web Content Accessibility Guidelines
API	Application Programming Interface
DBMS	Database Management System

Term	Definition
PDF	Portable Document Format

1.4 References

- World Wide Web Consortium (W3C) – *Web Content Accessibility Guidelines (WCAG) 2.1*
- Mozilla Developer Network (MDN) – Accessibility Documentation
- IEEE Std 830-1998 – *IEEE Recommended Practice for Software Requirements Specifications*
- Accessibility Developer Tools and Chrome Lighthouse documentation

1.5 Overview

This document is organized into several sections:

- **Section 2** provides an overview of the system, including its context, main functions, constraints, and assumptions.
- **Section 3** specifies detailed functional and non-functional requirements.
- **Section 4** presents system models such as use case and data flow diagrams.
- **Section 5** includes other requirements such as backup, security, and compliance.
- **Section 6** lists appendices and supporting materials.

2. Overall Description

2.1 Product Perspective

The "Website accessibility simulator" is an independent website accessibility checker and disability simulator. The system will also provide an API to allow experienced developers to add their own scripts for testing. It uses a backend engine for scanning pages and a frontend interface for displaying simulation previews and reports.

2.2 Product Functions

- Scan a given website and detect accessibility violations.
- Simulate visual impairments (color blindness, low vision, grayscale, high contrast) and other impairments (if time permits).
- Allow navigation via both mouse and keyboard.
- Generate downloadable reports (PDF) with issue details.
- Provide role-based access for testers, and Admins.
- Store project history, previous scans, and results.
- Offer an API for programmatic scan triggering and report retrieval.

2.3 User Characteristics

User Type	Description
Tester	Runs accessibility scans, analyzes results, and reports issues. Moderate technical knowledge.
Developer	Fixes accessibility issues based on reports and simulations. High technical expertise.
Admin	Oversees projects, reviews reports, and manages user roles. Basic to moderate technical knowledge.

2.4 Constraints

- Requires a stable internet connection.
- Must comply with WCAG 2.1 AA guidelines.
- Dependent on modern web browsers (Chrome, Edge, Firefox).
- Server processing time may limit scan size (e.g., max 100 pages per project).
- Authentication must use secure protocols (HTTPS, JWT).
- Not Limited to analyzing publicly accessible pages (can scan local/internal URLs).

2.5 Assumptions and Dependencies

- Users have valid credentials and necessary permissions to run scans.
- The system depends on external APIs or libraries for accessibility rule checking (e.g., axe-core).
- Cloud storage or database services are available for report archiving.
- Continuous network access is required for remote scans.

3. Specific Requirements

3.1 Functional Requirements

ID	Requirement Description	Priority
FR1	The system shall allow users to input a website URL and initiate an accessibility scan.	High
FR2	The system shall detect and categorize accessibility violations according to WCAG 2.1 AA.	High
FR3	The system shall simulate impairments such as color blindness, grayscale, and high contrast.	High
FR4	The system shall allow users to export reports in PDF format.	High
FR5	The system shall store project scan history and allow users to re-run previous scans.	Medium
FR6	The system shall provide role-based access (Tester, Developer, Admin).	High

ID	Requirement Description	Priority
FR7	The system shall allow API access for triggering scans and fetching results.	Medium
FR8	The system shall send email notifications when scans are completed.	Low

3.2 Non-Functional Requirements

Category	Description
Performance	Each scan should complete within 5 minutes for 20 pages.
Security	Use HTTPS, JWT authentication, and encrypted storage.
Reliability/Availability	System uptime of at least 98%.
Usability	User interface should be responsive and accessible (WCAG AA).
Scalability	Support up to 1000 concurrent scans (overall).
Maintainability	Modular codebase for easy updates and rule additions.

3.3 External Interface Requirements

- **User Interface:**
  Web dashboard with options for inputting URLs, viewing reports, and simulating impairments along with options for specifying impairment type and severity.
- **Hardware Interface:**
  Computer or mobile device with a web browser.
- **Software Interface:**
  Interfaces with accessibility rule engines (axe-core API), databases, and PDF generators.
- **Communication Interface:**
  HTTPS for all network communication, REST API endpoint for integration.

3.4 System Features

Feature 1: Accessibility Scan

- **Description:** Scans website for accessibility issues.
- **Inputs:** Website URL or Local URL.
- **Processing:** Runs automated rule checks.
- **Outputs:** List of detected violations with severity levels (WCAG AA).

Feature 2: Simulation

- **Description:** Simulates the website as experienced by users with specific impairments.
- **Inputs:** Select impairment type.
- **Processing:** Applies changes for impairments and re-renders sites.

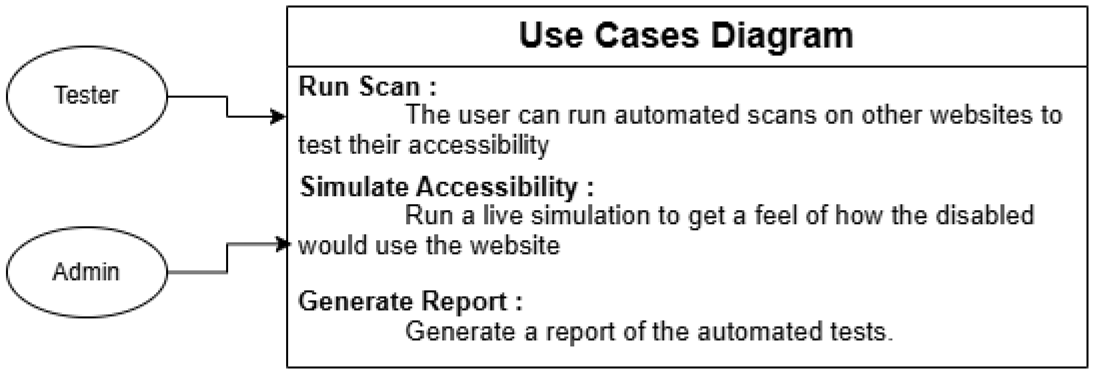
- **Outputs:** Simulated website.

Feature 3: Report Generation

- **Description:** Generates detailed reports of accessibility results.
- **Inputs:** Scan results.
- **Processing:** Formats results into PDF (with remediation tips (if required by client)).
- **Outputs:** Downloadable report files.

4. System Models

4.1 Use Case Diagram


Tester
Admin
Use Cases Diagram
Run Scan :
The user can run automated scans on other websites to test their accessibility
Simulate Accessibility :
Run a live simulation to get a feel of how the disabled would use the website
Generate Report :
Generate a report of the automated tests.

5. Other Requirements

- **Backup and Recovery:**
  Automatic daily backups of scan data and reports to cloud storage.
- **Security:**
  Role-based authentication, encryption for sensitive data, secure token-based API access.
- **Compliance:**
  Must follow WCAG 2.1 AA and GDPR data protection standards.
- **Error Handling:**
  Graceful recovery and user-friendly error messages.

6. Appendices

- **Glossary:**
  - *Accessibility:* Designing websites usable by people with disabilities.
  - *Simulation:* Reproducing visual experiences of different impairments.
  - *Violation:* Accessibility rule not satisfied by the web content.
- **Mock-ups:**
  Don’t have any demo yet.

- **Data Dictionary (Example):**

Field	**Type**	**Description**
project_id	VARCHAR	Unique project identifier
user_id	VARCHAR	Owner of the scan
url	VARCHAR	Website URL scanned
result_json	TEXT	JSON-formatted scan results