**Software Requirements Specification (SRS)**

**Project Name:** CharityLens  
**Author:** Kbrom Mehari  
**Date:** July 2025  
**Version:** 1.0

### 1. ****Introduction****

#### 1.1 Purpose

This document outlines the software requirements for **CharityLens**, a web-based platform that enhances transparency in charity operations by allowing donors and the public to trace how donated funds are used.

#### 1.2 Intended Audience

* Developers and designers building the system
* Stakeholders such as charity organizations and donors
* Future contributors (open-source community)

#### 1.3 Scope

CharityLens provides tools for:

* Tracking donation flow from input to impact
* Displaying financial reports and transactions
* Managing donor accounts and dashboards
* Providing public transparency via a live project activity feed

### 2. ****Overall Description****

#### 2.1 Product Perspective

CharityLens is a standalone system, with potential integration in the future with payment gateways or accounting systems. The project follows a modular architecture to separate frontend, backend, and data services.

#### 2.2 Product Functions

* Donors can register and view their donation history.
* Charities can log income and expenses related to projects.
* Admins can generate summary reports.
* The public can view anonymized financial trails and updates.

#### 2.3 User Classes and Characteristics

| **User Class** | **Description** |  |
| --- | --- | --- |
| **Admin** | Manages platform settings and user access |  |
| **Donor** | Views donation history and project impact |  |
| **Charity Manager** | Adds and manages charity financial data |  |
| **Public** | Views general activity and transparency data |  |

#### 2.4 Operating Environment

* **Backend**: Java, Spring Boot
* **Frontend**: HTML/CSS/JavaScript (or React)
* **Database**: MySQL/PostgreSQL
* **Platform**: Web (Desktop & Mobile responsive)

#### 2.5 Constraints

* Data must be stored securely and comply with privacy laws.
* Must be built within 2–4 months as MVP.
* Hosted on cloud services (AWS, Render, etc.)

### 3. ****Functional Requirements****

#### 3.1 User Registration & Login

* Users can register and log in securely.
* OAuth support for social login (optional).

#### 3.2 Donor Dashboard

* View donation history
* View impact metrics for contributed projects

#### 3.3 Charity Project Management

* Add new charity projects
* Log income and expenses per project
* Upload supporting documents

#### 3.4 Transparency Feed

* Public view of anonymized donation flow
* Timeline of updates on projects

#### 3.5 Reporting Module

* Admins and charities can generate monthly/annual reports
* Export as PDF/CSV

### 4. ****Non-Functional Requirements****

#### 4.1 Performance

* Should support 100+ concurrent users in MVP stage.

#### 4.2 Security

* Password hashing using BCrypt.
* Role-based access control (RBAC).
* HTTPS for data transmission.

#### 4.3 Usability

* Clean, intuitive UI for all user types.
* Mobile-responsive design.

#### 4.4 Scalability

* Modular design to allow easy scaling as user base grows.

#### 4.5 Maintainability

* Clean, well-documented codebase
* Follows standard naming and coding conventions

### 5. ****Appendices****

* Diagrams (to be added in Technical Design)
* User stories or use cases (can be added later)

### Summary

This SRS defines the key system behavior and expectations of the CharityLens platform. It will evolve during the development process as user feedback and technical insights are collected.