Software Requirements Specification (SRS) for Certificate Automation System

# Introduction

## Purpose

The Certificate Automation System is designed to streamline the creation, storage, and

verification of digital certificates. The system will collect user data, generate certificates in PDF/IMG formats with QR codes, store them in a database, and provide a frontend for

verification.

## Scope

 Collect user data via Google Forms.

 Store responses in Google Sheets and transfer them to a database (PostgreSQL/MySQL/MongoDB).

 Automatically generate certificates with pre-stored templates, credentials, and QR codes.  Provide a frontend (React/HTML+CSS+JS) for certificate verification via QR scanning.

# Overall Description

## System Architecture

The system consists of four main components:

* + 1. Data Collection Layer (Google Forms + Sheets)
    2. Database Layer (PostgreSQL/MySQL/MongoDB)
    3. Backend Layer (Certificate Generation + Storage)
    4. Frontend Layer (Verification UI)

## Functional Requirements

ID Requirement Description

FR1 Data Collection Google Form for user input, stored in Google

Sheets.

FR2 Database Integration Transfer data from Sheets to SQL/NoSQL

databases.

|  |  |  |
| --- | --- | --- |
| FR3 | Certificate Generation | Auto-generate PDF/IMG certificates with QR codes. |
| FR4 | Certificate Storage | Save certificates in structured folders (PDF/IMG). |
| FR5 | Verification Frontend | Display certificate details when QR is scanned. |
| FR6 | Unique URL Redirect | Each QR links to a unique verification page. |
| FR7 | Credential Validation | Show "Verified" status on successful validation. |

FR8 Manual Check Option Allow users to manually enter credentials for

verification.

## Non-Functional Requirements

ID Requirement Description

NFR1 Performance Fast certificate generation (<5 sec per

certificate).

NFR2 Security Encrypted QR codes to prevent forgery.

|  |  |  |
| --- | --- | --- |
| NFR3 | Scalability | Support 10,000+ certificates without degradation. |
| NFR4 | Compatibility | Cross-platform support (Web, Mobile). |
| NFR5 | Usability | Intuitive UI for verification. |

# System Features

## Data Collection & Storage

 Google Form → Collects user data (Name, Course, Email, etc.).

 Google Sheet → Stores responses in JSON format via API.

 Database Schema → Structured tables in PostgreSQL/MySQL/MongoDB.

## Backend Processing

 Certificate Generation

 Uses pre-stored JPG templates for different courses.  Embeds user data + credentials + QR code.

 Saves certificates in /PDF and /IMG folders.

 API Integration

 RESTful endpoints for automation (e.g., /generate-certificate ).

## Frontend Verification

 React App (Primary) & HTML/CSS/JS (Fallback).

 QR Scanning → Redirects to baseURL/credential-id .

 Verification Page → Displays:  Certificate in IMG format.  "Verified" badge (if valid).

 Option to manually check another certificate.

# Technical Specifications

## Database Schema (PostgreSQL Example)

sql

CREATE TABLE certificates ( id SERIAL PRIMARY KEY,

user\_name VARCHAR(100),

course\_name VARCHAR(100), email VARCHAR(100),

credential\_id UUID UNIQUE, qr\_code\_url VARCHAR(255), pdf\_path VARCHAR(255),

img\_path VARCHAR(255),

created\_at TIMESTAMP DEFAULT NOW()

);

## File Structure

text

Certificate-Automation/

├── Database/

│ ├── Schema.SQL

│ ├── PostgreSQL/

│ ├── MySQL/

│ └── MongoDB/

├── Google/

│ ├── Form-link.md

│ ├── Sheet-API.JSON

│ └── sheet\_link.md

├── Frontend/

│ ├── static/

│ │ ├── Index.html

│ │ ├── Style.css

│ │ └── Script.JS

│ └── React/

│ └── React-flow/

├── Backend/

│ ├── Generated-Certificates/

│ │ ├── PDF/

│ │ └── IMG/

│ ├── Certificate\_Templates/

│ │ └── (JPG templates)

│ ├── .env

│ └── API/

└── README.md

## Technologies

 Backend: Node.js (Express) / Python (Flask/Django)

 Database: PostgreSQL / MySQL / MongoDB

 Frontend: React.js + HTML/CSS/JS

 QR Generation: qrcode npm library / Python qrcode module

 PDF Generation: pdfkit / reportlab