Software Requirements Specification (SRS)

Project: Samuel Afon Photography Website

---

1. Introduction

1.1 Purpose

This document defines the functional and non-functional requirements for the **Samuel Afon Photography Website**, which serves as an online platform to showcase work, manage bookings, and streamline communication between photographer and clients.
It also outlines backend integration, hosting, and future system enhancements for scalability.

1.2 Scope

The platform will allow users to:

- View categorized photography galleries (portraits, events, automotive, real estate, etc.)
- Learn about the photographer's background and services
- Submit bookings with detailed session preferences and images
- Receive email confirmation upon submission
- Review transparent pricing and avoid bargaining

The system will include an **admin dashboard** for managing:

- Bookings (view/edit/delete)
- Portfolio updates
- Text reminders
- Future integration for payments

Deployment is via **GitHub Pages** for the frontend, with **Flask backend** integration to handle form submissions, authentication, and admin management.

1.3 Definitions and Acronyms

Term	Definition
SRS	Software Requirements Specification
UI / UX	User Interface / User Experience
CRUD	Create, Read, Update, Delete
DB	Database

Term	Definition
SES	Simple Email Service
SMS	Short Message Service
API	Application Programming Interface

---

2. Overall Description

2.1 Product Perspective

The system consists of:

- **Frontend (static)** hosted on GitHub Pages
- **Backend (Flask)** connected via API to manage bookings, data storage, authentication, and automated communication
- **Database (SQLite or AWS RDS)** for persistent storage

External APIs:

- **ClickSend API** for text message reminders
- **Email service (Flask-Mail / AWS SES)** for confirmation emails

---

2.2 Product Features

1. **Home Page:** Hero banner, call-to-action (“Book a Session”), and portfolio preview.
2. **About Page:** Background, photography philosophy, and service overview.
3. **Portfolio Page:** Organized gallery with filtering (portraits, events, automotive, real estate).
4. **Pricing Page:** Transparent pricing breakdown (no bargaining).
5. **Booking Page:** Interactive form with:
    - Contact details
    - Session type
    - Outfit count and creative direction
    - File upload (max 4 images)
    - Studio or travel add-ons
6. **Contact Page:** Direct message form.
7. **Admin Dashboard (Flask):**
    - Authentication required
    - Manage portfolio and booking entries
    - View messages
    - Send confirmations or reminders
    - Control pricing data

2.3 User Classes and Characteristics

User Type	Description	Access Level
Client	Books sessions, views portfolio	Public
Admin (Samuel)	Manages content, bookings, text reminders	Authenticated only

---

2.4 Operating Environment

- **Frontend:** HTML5, CSS3, JavaScript, Bootstrap 5
- **Backend:** Python (Flask Framework)
- **Database:** SQLite or AWS RDS (for scalability)
- **Hosting:** GitHub Pages (Frontend), AWS Amplify or Flask Server (Backend)
- **Third-party APIs:** ClickSend (SMS), AWS SES or Flask-Mail (Email)
- **Browsers:** Chrome, Safari, Firefox, Edge

2.5 Constraints

- Responsive layout (mobile-first)
- HTTPS enforced
- Authentication required for admin panel
- File uploads limited to 4 images per booking
- Uploaded images automatically deleted after a set retention period (e.g., 7 days)
- Lightweight performance to ensure fast load times

2.6 Assumptions and Dependencies

- Users have internet access
- GitHub Pages and backend hosting remain operational
- ClickSend API key and email service credentials are active
- Manual payment remains default until online payment integration is added

---

3. Specific Requirements

3.1 Functional Requirements

FR1 – Portfolio Display

- Display categorized galleries dynamically.
- Allow admin to add, update, or remove photos.

FR2 – Booking Form

- Collect name, email, phone, session type, date/time, outfits, creative direction, and up to 4 images.

- Require email confirmation for clients after submission.
- Send a copy to the admin dashboard.
- Optionally trigger ClickSend SMS reminder before the session.

FR3 – Pricing Information

- Clearly display pricing per session type, add-ons, and conditions.
- Update pricing data via admin dashboard.

FR4 – About Page

- Editable bio, philosophy, and mission sections.

FR5 – Contact Page

- Simple contact form with validation and backend email relay.

FR6 – Admin Dashboard

- Secure login authentication (username/password).
- CRUD operations for bookings and portfolio items.
- Control for pricing and service details.
- Optional "Send Reminder" feature using ClickSend API.

FR7 – File Handling

- Accept image uploads (max 4 files, ≤5MB each).
- Delete stored images after predefined retention period.

---

4. Non-Functional Requirements

ID	Category	Requirement
NFR1	**Performance**	Pages must load in ≤3 seconds on standard broadband.
NFR2	**Security**	HTTPS required; sanitize all inputs to prevent XSS/SQL injection.
NFR3	**Usability**	Fully responsive design; simple navigation and CTA prominence.
NFR4	**Reliability**	Expected uptime 99% under hosting services.
NFR5	**Maintainability**	Use modular file structure and version control (GitHub).
NFR6	**Scalability**	Backend should allow migration to cloud DB or API scaling.
NFR7	**Design Consistency**	Theme colors: **#3A8963 (green)**, **white**, and **black** across all pages.

---

5. Future Enhancements

- **Online payments:** Integrate Stripe or PayPal.
- **Automated scheduling:** Calendar sync via Google/Apple APIs.
- **Client accounts:** Access past bookings and galleries.
- **Analytics dashboard:** Monitor visitor engagement and conversion rates.
- **Cloud image storage:** Migrate uploads to AWS S3 or Google Cloud Storage.

---

6. References

- ClickSend API Documentation
- Flask Framework Documentation
- Bootstrap 5 Docs
- GitHub Pages Deployment Guide
- AWS SES Email Setup