Software Requirements Specifications

for

Incognito-Vault

Version 1.0

Prepared by
Name: Biswadeb Mukherjee.
Phone: +91 9836763466.

ParseSphere Innovations.

Date Created: 14th December, 2024

This Software Requirements Specification (SRS) document is the original and sole official documentation for Incognito-Vault, developed by Biswadeb Mukherjee (ParseSphere Innovations). Unauthorized reproduction, duplication, or distribution of this document, in whole or in part, is strictly prohibited without prior written consent from the developer.

Sincerly,
Biswadeb Mukherjee.
Ethical Hacker.
ParseSphere Innovation.

Table of Contents

1. Introduction

1.1 Purpose

The *Incognito-Vault* platform is designed to provide a secure, user-friendly, and feature-rich system for managing databases, logs, and forms. It is tailored for use in environments where data integrity, security, and ease of administration are crucial. This document outlines the detailed software requirements to guide the development, deployment, and maintenance of the *Incognito-Vault* system.

1.2 Intended Audience

- **System Administrators:** Configure and manage the platform.
- **End-Users**: Regular users who interact with forms and databases.
- **Developers**: Contribute to or maintain the system.
- **Security Experts**: Review and enhance security measures.

1.3 Scope

The *Incognito-Vault* system allows administrators to securely manage databases, track logs, and handle forms within an intuitive interface. It provides robust security features to ensure data protection against malicious activity, while also supporting scalability for handling growth.

ID	Stakeholder	Description
S-1	Individual	Can purchase our the software for their own use.
S-2	Students & Educational Institutions	Can use this platform for academic purposes, such as managing students records, forms, & communications logs
S-3	IT professional	Provide technical support during deployment & operation.
S-4	Company	Can purchase our software for managing their company database, logs, etc ,etc.
S-5	Researcher	Can use it for their research purpose.

1.4 References

1. ISO/IEC 27001:2013 - Information technology -- Security techniques -- Information security management systems -- Requirements
2. NIST Special Publication 800-53: Security and Privacy Controls for Federal Information Systems and Organizations
3. OWASP Secure Coding Practices: Quick Reference Guide
4. SDLC Lifecycle: Prototype Model in Software Development

2. System Overview

Incognito-Vault is a secure, scalable platform designed for handling database management, logging, and form submission. It includes security features such as **CAPTCHA** for login, session management, rate limiting, and encryption & hashing. The system is built using Python (Flask) for backend services and MySQL for database storage, while Redis is employed for caching, server-side storage and session management.

The platform consists of the following components:

- **Frontend**: User interface for both regular users and admins, implemented using modern HTML, CSS, Bootstrap and JavaScript.
- **Backend**: Handles business logic and data processing through Flask, ensuring secure interactions.
    - **Database**: MySQL, storing user data, logs, and system configurations.
    - **Cache**: Redis, used for session management, server-side storage and fast data retrieval.

2.1 Product Perspective:

Incognito-Vault is a web-based application, accessible through a secure web interface. The system is designed to be scalable, flexible, and adaptable to meet the evolving needs of individuals, business, reaseachers, etc, etc. The system is designed to integrate with other systems and applications, as required.

2.2 Product Functions:

The Incognito-Vault system is provided with the following functions:

- User registration and login.
- Fast data storage.
- Data encryption and decryption.
- Access control and permission management.
- Data backup and recovery.
- User interface for data management and monitoring.
- System administration and maintenance.

3. System Features

3.1. Secure User Authentication

- **Description**: A login system that uses rate-limiting, CAPTCHA, user lockout, and encrypted passwords for user verification.
- **Functional Requirements**:
    - Users can log in using a unique username and password.
    - After multiple failed login ISO Standard attempts, the system will temporarily lock the account for 15 mins.
    - CAPTCHA has been integrated to prevent automated attacks.

Use Case:

- **Actor**: Regular User
- **Description**: A user attempts to log in with incorrect credentials. After 5 failed attempts, the system locks the account for 15 minutes.

3.2. Super Admin Dashboard

- **Description**: A comprehensive dashboard that allows super admins to manage user accounts, monitor logs, and configure system settings.
- **Functional Requirements**:
    - Super Admin can add, modify, or delete user accounts.
    - Super Admin can access logs to monitor user activities and detect potential security issues.
    - Provides a mechanism to manage system settings like database configurations.

Use Case:

- **Actor**: Super admin User
- **Description**: The super admin logs into the dashboard, accesses the "User Management" section, and disables an inactive user account.

3.3. Database Management

- **Description**: Super admins can view, add, modify, or delete records in the system's databases.
- **Functional Requirements**:
    - Admins can perform CRUD operations on the database.
    - The system offer templates for structured data management to avoid errors.

Use Case:

- **Actor**: Super admin User
- **Description**: Admin accesses the database management module and updates an entry for a record.

3.4. Error Handling and Logging

- **Description**: Error pages and logging mechanisms to ensure proper error reporting and tracking.
- **Functional Requirements**:
  - The system will display informative error pages to users.
  - Server-side logs are created for each error, which are accessible by the Super admin.

Use Case:

- **Actor**: Super admin User
- **Description**: Super admin reviews error logs to diagnose a failed login attempt and resolves the issue by resetting the user's password.

4. Security Features

4.1. Rate Limiting:

- Protects against brute-force attacks by limiting login attempts to a specific number per minute. If the limit is exceeded, the user is temporarily blocked.

4.2. Lockout Mechanism:

- After several failed login attempts, accounts will be locked for a configurable period. This prevents attackers from continuously guessing passwords.

4.3. Session Management

- Ensures that user sessions are securely managed, including session timeouts and the prevention of session hijacking.

4.4. HTTPS Support:

- In production environments, the platform supports **HTTPS** for encrypted communication, ensuring data integrity during transmission.

4.5. CAPTCHA Integration:

- **CAPTCHA** verification is used during login to prevent bots from attempting unauthorized access.

5. System Architecture and Design:

6. Deployment Flow:

Deployment Steps:

1. **Environment Setup**:
   - Install Python 3.8+, MySQL, Redis, and a WSGI server (uWSGI or Gunicorn).
2. **Web Server Configuration**:
   - Configure a reverse proxy (Nginx or Apache) to handle incoming requests and forward them to the application server.
3. **HTTPS Setup**:
   - Use Certbot to obtain an SSL certificate for HTTPS support, ensuring secure communication.
4. **Data Caching**:
   - Redis is configured for caching session data and frequently accessed records to enhance system performance.
5. **System Monitoring**:
   - Use tools like Prometheus and Grafana for monitoring server performance, resource usage, and error logs.

ER Diagram:

7. Data Structure and Database Design

8. Use Case Diagrams:

9. Maintenance and Support

Logging and Monitoring:

- Logs should capture error details, user activity, and system performance metrics.
- Admins should have access to these logs for troubleshooting.

Security Audits:

- The platform will undergo regular security audits to identify potential vulnerabilities and implement fixes.

10. System Requirements

To ensure the proper functioning of *Incognito-Vault*, the following system specifications are recommended for both development and production environments:

- **Operating Systems**:
  - **Production**: Linux (Ubuntu 20.04+ or CentOS 7+)
  - **Development**: macOS, Windows 10/11 (WSL2 recommended for Windows)
- **Memory**:
  - **Minimum**: 2 GB RAM
  - **Recommended**: 4 GB RAM or more
- **Disk Space**:
  - **Minimum**: 20 GB free
  - **Recommended**: 50 GB free
- **CPU**:
  - 64-bit processor
  - Multi-core recommended for better performance

11. Server Setup Configurations

For the system to be deployed in a production environment, the following configuration steps and server requirements must be met:

1. Python Environment:

- Ensure **Python 3.8+** is installed.
- Set up a **Python Virtual Environment** to isolate dependencies:

```
python -m venv .venv
source .venv/bin/activate
```

2. Server Requirements:

- Install a **production-ready WSGI server uWSGI** to serve the application.
- Install and configure **Redis** for session management and caching.
- Install **MySQL** for the backend database.

3. Web Server Setup:

- **Reverse Proxy**: Use **Nginx** or **Apache** (optional but recommended) for better performance and security.
- Configure a reverse proxy to forward requests to the WSGI server (e.g., Gunicorn or uWSGI).

4. Nginx Example Configuration:

```
server {
    listen 80;
    server_name yourdomain.com;

    location / {
        proxy_pass http://127.0.0.1:8800;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root /usr/share/nginx/html;
    }
}
```

5. HTTPS Configuration:

- Use **Let's Encrypt** or another SSL provider for HTTPS support:

```
sudo apt install certbot python3-certbot-nginx
sudo certbot --nginx -d yourdomain.com
```

6. Redis Setup:

- Ensure Redis is installed and running:

```
sudo systemctl start redis
sudo systemctl enable redis
```

-
-
-
-
-

-
-
-
-

For enhanced security, bind Redis to localhost and require a password for access:

- **Redis Config**:

```
/etc/redis/redis.conf:
bind 127.0.0.1
requirepass your-strong-password
```

These configurations will allow the platform to run smoothly and securely in a production environment, ensuring scalability, performance, and secure communication.

CONCLUSION

The development of the Incognito-Vault system aims to provide a secure, user-friendly, and efficient platform for managing sensitive information while maintaining user privacy and data integrity. Through the requirements outlined in this document, we have established a clear framework for the functional, non-functional, and design aspects of the system.

By prioritizing robust encryption, seamless user experience, and scalability, Incognito-Vault is designed to meet the needs of individuals and organizations requiring reliable and confidential data storage. The comprehensive requirements defined in this document will guide the development process to ensure that the final product aligns with stakeholder expectations and industry standards. Moving forward, it is critical to validate these requirements through continuous stakeholder collaboration, rigorous testing, and iterative improvements. The successful implementation of the Incognito-Vault system will empower users with the confidence to manage and protect their data securely in an increasingly digital and interconnected world.

End Note

This Software Requirements Specification (SRS) document for Incognito-Vault has been created with diligence and accuracy, reflecting the best understanding of the system's requirements and goals as of the date of its preparation.

We acknowledge that software development is a dynamic process, and new insights or changes may arise during the implementation phase. Any such changes will be documented and incorporated into updated versions of this SRS.

This document is intended to serve as the definitive reference for stakeholders, ensuring clarity and alignment in the design, development, and deployment of the Incognito-Vault system.