**Software Requirements Specification (SRS) for Password Manager**

**1. Introduction**

**1.1 Purpose:** The purpose of this document is to provide a detailed description of the requirements for the password manager application. This includes functional and non-functional requirements, user interface requirements, and external interface requirements. The document is intended for developers, testers, and stakeholders involved in the project.

**1.2 Scope**: This password manager application is designed to securely store and manage user passwords and sensitive information. It includes features like password generation, storage, retrieval, and auto-fill capabilities. The application is intended for individual users and provides a user-friendly interface while maintaining robust security protocols.

**1.3 Definitions, Acronyms, and Abbreviations**

**AES:** Advanced Encryption Standard

**UI:** User Interface

**UX**: User Experience

**2FA:** Two-Factor Authentication

**2. Overall Description**

**2.1 Product Perspective:** The password manager is a standalone application that operates independently and integrates with web browsers for auto-fill functionalities.

**2.2 Product Functions**

* **User Authentication:** Secure user login and registration.
* **Password Storage:** Secure storage and management of passwords.
* **Password Generation:** Tool for generating strong passwords.
* **Auto-fill:** Automatically filling login information for users.
* **Password Sharing:** Secure sharing of passwords between users.
* **Audit Logs:** Tracking user activities within the application.

**2.3 User Classes and Characteristics**

* **End-Users:** Individuals who use the application to store and manage passwords.
* **Administrators:** Individuals who manage the application and provide support to end-users.

**2.4 Operating Environment**

**Desktop Application:** Windows, macOS, Linux.

**Mobile Application:** iOS, Android.

**2.5 Constraints**

* The application must comply with data protection regulations such as GDPR.
* The application should work seamlessly across various devices and operating systems.

**2.6 Assumptions and Dependencie**s

* Users have a basic understanding of using software applications.
* The application relies on secure internet connectivity for certain functionalities.

**3. Specific Requirements**

**3.1 Functional Requirements**

* **User Authentication**
* Users must be able to register and log in securely.
* Implement two-factor authentication for added security.
* **Password Storage**
* Store passwords securely using AES-256 encryption.
* Provide options to categorize and label stored passwords.
* **Password Generation**
* Generate strong, random passwords.
* Allow users to customize password criteria (length, complexity).
* **Auto-fill**
* Integrate with web browsers to auto-fill login forms.
* Securely store and auto-fill login credentials.
* **Password Sharing**
* Enable secure sharing of passwords between users.
* Implement access controls for shared passwords.
* **Audit Logs**
* Maintain logs of user activities (login, password changes, etc.).
* Allow administrators to review audit logs.

**3.2 User Interface Requirements**

* **Desktop UI**
  + Intuitive and clean design.
  + hrough stored passwords.
* **Mobile UI**
  + Responsive design suitable for mobile devices.
  + Access to all functionalities available on the desktop version.

**3.3 External Interface Requirements**

* Hardware Interfaces
* Compatible with standard hardware configurations.
* **Software Interfaces**
* Integration with popular web browsers (Chrome, Firefox, Safari).

**3.4 Performance Requirements**

* Quick response time for accessing and retrieving passwords.
* Efficient memory usage to ensure smooth performance.

**3.5 Security Requirements**

* Implement AES-256 encryption for data storage.
* Ensure secure transmission of data over the internet.
* Regular security updates and patches to address vulnerabilities.

**3.6 Non-Functional Requirements**

* Security: Maintain the highest level of encryption and data protection.
* Performance: Ensure responsive and fast operations, even with large volumes of data.
* Usability: Provide a user-friendly interface that is easy to learn and use.
* Reliability: Ensure the application is reliable and minimizes downtime.
* Maintainability: Ensure the application can be easily maintained and updated.

**4. System Features**

**4.1 User Authentication**

* **Description:** Secure user login and registration with two-factor authentication.
* **Functional Requirements:** Encrypt user credentials, implement secure login processes.
  1. **Password Storage**
* **Description:** Secure storage and management of passwords.
* **Functional Requirements:** Use AES-256 encryption, provide easy access to stored passwords, allow password categorization.

**4.3 Password Generation**

* **Description**: Tool for generating strong, random passwords.
* **Functional Requirements:** Customizable password criteria (length, complexity).

**4.4 Auto-fill**

* **Description:** Automatically fills login forms for users.
* **Functional Requirements:** Integrate with web browsers, securely store and auto-fill login credentials.

**4.5 Password Sharing**

* **Description:** Secure sharing of passwords between users.
* **Functional Requirements:** Implement access controls, encrypt shared passwords.

**4.6 Audit Logs**

* **Description:** Track user activities within the application.
* **Functional Requirements:** Maintain activity logs, provide access to administrators for reviewing logs.

**5. Other Requirements**

**5.1 Legal and Regulatory Requirements**

* Compliance with data protection regulations (e.g., GDPR).
* Ensure user consent for data collection and usage.

**5.2 Documentation Requirements**

* User manuals and help guides.
* Technical documentation for developers and administrators.

**6. Technologies Used**

**6.1 Front-End**

* **HTML5:** For structuring the web pages and content.
* **CSS3:** For styling and layout of the application.
* **JavaScript:** For dynamic interactions and functionalities.
* **PHP:** For server-side scripting and integration with the back-end.

**6.2 Back-End**

* **Apache Server:** For handling HTTP requests and serving web pages.
* **PHP:** For server-side logic and integration with the database.

**6.3 Database**

* **MySQL:** For storing user credentials and other data.

**6.4 Security**

* **AES-256 Encryption:** For securing stored passwords and sensitive data.
* **OAuth 2.0:** For secure user authentication and authorization.
* **Two-Factor Authentication (2FA):** For enhanced security during user login.
* **HTTPS:** For secure communication over the internet.

**6.5 DevOps and Deployment**

* **XAMPP:** For local development and testing environment.
* **AWS/Azure:** For cloud hosting, storage, and other services.

**6.6 Version Control**

* **Git:** For version control and code collaboration.
* **GitHub:** For code repository hosting and project management.

**7. Conclusion**

This Software Requirements Specification (SRS) document outlines the complete requirements for the development, implementation, and maintenance of the password manager application. By specifying both functional and non-functional requirements, along with user interface and security specifications, this SRS ensures a comprehensive guide for delivering a secure, efficient, and user-friendly password manager application. The application leverages XAMPP-based technologies to meet high standards of performance and security. With a clear and detailed blueprint, the project is set for successful development and deployment.