**SDA Assignment 1**

**K226007**

**BCS-5L**

**Yousha Mehdi**

**Phase 1:**

Task:

1: Security Vulnerabilities: Many systems are susceptible to breaches, exposing confidential data. Inadequate encryption or insecure authentication mechanisms can compromise document integrity.

Poor User Experience: Some systems may be cumbersome, with overly complex workflows for uploading, sharing, and accessing documents.

Access Control Issues: Insufficient granular access control leads to unauthorized access or accidental sharing of sensitive data.

2: End-to-End Encryption: Ensure that documents are encrypted during both storage and transit, making unauthorized access impossible.

User Authentication and Multi-Factor Authentication (MFA): Strengthen login security by requiring users to verify their identity through multiple authentication methods.

Granular Access Control: Allow document owners to specify which users or groups can view, edit, or share the document. Include expiration dates for access.

Real-Time Activity Logging: Provide clear logs showing when documents are accessed, by whom, and what actions were taken (view, edit, share).

User-Friendly Interface: Ensure the system is easy to navigate, allowing non-technical users to securely share documents without difficulty.

3: Corporate Clients:

Need secure document sharing for internal teams and external partners.

Require detailed audit logs and compliance with industry regulations.

Law Firms:

Demand airtight confidentiality and legal compliance for client data.

Require role-based access control (only authorized lawyers or staff can access certain documents).

Government Agencies:

Require top-level encryption and strict access controls to prevent unauthorized access to sensitive data.

Demand compliance with national security and data protection regulations.

**Phase 2:**

Task:

1. Technically feasible with available resources and skilled developers. However, careful design will be needed to avoid vulnerabilities in encryption and key management.

* Development Team
* Tools and Infrastructure
* Testing and Compliance
* Maintenance Costs

The high security and compliance features will justify a premium pricing model, ensuring a good return on investment. While initial costs may be high, the long-term benefits, particularly for high-end clients, outweigh the investment.

3. AlphaSecure's existing infrastructure includes secure servers and networks, scaling may be required depending on user growth, which cloud infrastructure can easily provide. Compatibility with current systems, such as authentication mechanisms and databases, will be essential, though minimal upgrades may be needed for handling increased traffic and data processing

**Phase 3:**

Task:

1.

* End-to-end encryption for all documents
* User authentication with multi-factor support.
* Granular access control for documents (view, edit, share permissions).
* Document upload, download, and share functionality.
* Secure document storage.

2.

* Security
* Performance
* Scalability
* Usability
* Reliability
* Data Integrity

3.

* User login
* Document upload
* encryption
* Admin sets access
* Document sharing
* Activity logging

**Phase 4:**

Task:

1.

Microservices architecture will be more suitable as it allows modular development, where different components (e.g., user authentication, document management, access control) are independently developed and scaled. This ensures scalability, better fault isolation, and ease of future upgrades for SecureShare.

2.

DB schema will include,

a) A Users table stores user details (user ID, username, password hashed, email, role, created timestamp).

b) A Documents table stores document metadata (document ID, owner ID, document name, file path, encrypted key, created timestamp).

c) An Access Logs table records user actions on documents (log ID, user ID, document ID, action type, timestamp).

3.

To ensure a user-friendly interface, SecureShare will feature intuitive navigation with clear menus and icons for core functions like upload, share, and view logs. A consistent design language across colors, fonts, and layout will make actions predictable. Tooltips and pop-ups will guide users through complex processes, while responsive design ensures functionality on both desktop and mobile. Immediate feedback will confirm user actions with success or error messages.

**Phase 5:**

Tasks:

1. Front-end:

Html, CSS, Tailwind, JS, React, TypeScript

Back-end:

Node.js, Express.js

Database:

MySQL

2.

**Modularization:**

Break down the application into smaller, reusable components to improve code organization and maintainability.

**Security Best Practices:**

Input validation: Sanitize user input to prevent injection attacks.

Secure communication: Use HTTPS to encrypt data in transit.

Regular updates: Keep dependencies and libraries up to date to address security vulnerabilities.

Code reviews: Conduct regular code reviews to identify and address potential issues.

**Phase 6:**

Task:

1. Use unit testing, functional, integrational and User acceptance testing.

2. Use automated tools and allow ethical hackers to test the security.

3. Response time must be good, Scalability and data throughput.

**Phase 7:**

Task:

1. Prepare servers, install application, migrate data securely, test thoroughly, onboard users.
2. Encrypt data, validate data, control access, test, backup.
3. Manuals, tutorials, sessions, support channels.

**Phase 8:**

1. Performance tools, error tracking, security monitoring.
2. Issue tracking, prioritization, version control.
3. Internal audits, external audits, penetration testing, compliance.

**Part 3**

Task1:

The development environment serves as a sandbox for developers to experiment, write, and integrate code. It's a flexible space where changes can be made frequently without affecting other environments. Developers work collaboratively under the guidance of the Development Manager to ensure code meets initial requirements. The isolation of this environment prevents unstable or incomplete code from impacting other stages.

Task2:

The test environment differs from the development environment in its focus on stability and control. While both environments serve to test code, the test environment aims to replicate a more production-like setting to ensure reliable testing. Testers, working under the Test Manager, execute various test cases to verify functionality and detect bugs. Stability is crucial in the test environment to accurately assess the software's performance before moving to later stages.

Task3:

The staging environment is a near-replica of the production environment, designed to validate the final product before it goes live. The Project Manager ensures that the environment closely mirrors production to enable the team to finalize configurations, perform load testing, and verify integrations. This step helps identify potential issues before the software is released to end-users.  
  
Task4:

The staging environment is a near-replica of the production environment, designed to validate the final product before it goes live. The Project Manager ensures that the environment closely mirrors production to enable the team to finalize configurations, perform load testing, and verify integrations. This step helps identify potential issues before the software is released to end-users.

Task5:

The production environment is the live environment where the software is made available to end-users. It handles real user data and transactions, making changes tightly controlled to avoid disruptions. The Project Manager and Configuration Manager oversee this environment, while the Deployment Team executes updates or fixes. Reliability, security, and performance are paramount in the production environment.

Task6:

The mirror environment is a replica of the production environment used for disaster recovery and large-scale testing. It ensures that the system can be quickly restored in case of a failure in production. The Deployment Team and Configuration Manager maintain the mirror environment in sync with production to allow for real-time testing of updates or backups.