

# **GuardFile Web Application Proposal:**

## **GuardFile: Cloud-Based Secure File Storage System**

By: Dylan Miller & Mina Astafanous



CPSC 490

Professor: Dr. Lidia Morrison

Department of Computer Science

California State University, Fullerton

Fall 2025

## Table of Contents

|  |    |
|--|----|
| <b>Table of Figures</b>                                  | 3  |
| <b>Abstract</b>  | 4  |
| 1.0 Introduction   | 5  |
| 1.1 Background   | 5  |
| 1.2 Motivation   | 6  |
| <b>1.3 Related Work</b>                                  | 7  |
| 1.3.1 Multi-Factor Authentication for Data Protection    | 7  |
| 1.3.2 Cloud Storage Security & Encryption Practices      | 8  |
| 1.3.3 User Security Education & Awareness Tools          | 9  |
| <b>2.0 Problem Statement</b>                             | 9  |
| <b>3.0 Proposed Project &amp; Significance</b>           | 10 |
| <b>4.0 Objectives (Step-by-step Requirements and UX)</b> | 11 |
| <b>Overall Project Objective</b>                         | 11 |
| <b>4.1 UX Design and Functions</b>                       | 11 |
| 4.1.1 Getting Started with GuardFile                     | 11 |
| 4.1.2 Creating User Security Profile                     | 13 |
| 4.1.3 Uploading and Managing Files                       | 14 |
| <b>5.0 Activities</b>                                    | 16 |
| 5.1 Functionality  | 16 |
| 5.2 User-friendly Design                                 | 16 |
| 5.3 Security   | 17 |
| <b>6.0 Development Environment</b>                       | 18 |
| 6.1 Software Requirements                                | 18 |
| 6.2 Hardware Requirements                                | 18 |
| <b>7.0 Reports and Products</b>                          | 19 |
| <b>8.0 Schedule</b>                                      | 19 |
| <b>9.0 References</b>                                    | 20 |

## Table of Figures

|  |    |
|--|----|
| Figure 1 GuardFile Web Login Page                            | 11 |
| Figure 2 GuardFile Dashboard                                 | 12 |
| Figure 3 GuardFile Account Settings                          | 13 |
| Figure 4 GuardFile File Upload Page                          | 14 |
| Figure 5 GuardFile File Management Interface                 | 15 |
|  |    |
| Table 1: Software Requirements for GuardFile Web Application | 18 |
| Table 2: Hardware Requirements for GuardFile Web Application | 18 |

## **Abstract**

The GuardFile system is a secure web based cloud application designed to help users safely store, organize, and manage personal files while maintaining the core cybersecurity principles of confidentiality, integrity, and availability. GuardFile applies strong security protections including asymmetric encryption, multi factor authentication, audit logs, malware scanning, and adaptive access control. Research in encryption and secure cloud storage shows that strong cryptographic methods are essential for protecting sensitive data in cloud environments (Al Saeed 2019; Vijayarangan and Florence 2023). To verify user identity, GuardFile integrates password authentication, face recognition, and voice matching, which aligns with modern research that highlights the importance of multi factor authentication for preventing unauthorized access (Zhou, Chekole, and Ang 2025).

GuardFile also provides real time email alerts and detailed audit logs that record sign in time, device, IP address, and general location to help users identify unusual activity. In addition to securing data, GuardFile features a Security Score system that educates users on safe digital practices and encourages stronger cybersecurity habits. By combining encryption, authentication, access control, and user education. GuardFile creates a practical and secure environment for individuals who want both protection and awareness while managing their data in the cloud.

## **1.0 Introduction**

As digital storage becomes a major part of everyday life, users rely heavily on cloud services to manage important documents, photos, school assignments, and other personal files. Despite the convenience, many individuals remain unaware of how vulnerable their data can be without strong security measures. Cyberattacks, identity theft, man in the middle attacks, and weak authentication practices leave users exposed to dangerous threats. Research shows that cloud systems can be compromised when data is transmitted without strong cryptographic protection or when access control is mismanaged (Reece et al. 2024).

GuardFile was created to address these concerns by offering a secure and user-friendly cloud platform. It not only protects sensitive files through encryption and multi factor authentication but also informs users about safe cybersecurity practices. Instead of relying only on automated security features, GuardFile encourages users to understand and improve their protective habits. This combination of security and education supports long term digital safety.

### ***1.1 Background***

Many people assume their information is safe simply because it is stored in the cloud, but research shows that this is not always the case. Cloud environments are vulnerable to data leakage, insecure storage, weak keys, and unauthorized access when proper protections are not applied (Vijayarangan and Florence 2023). Users also frequently rely on password only authentication, which is no longer sufficient. Single factor authentication cannot reliably protect accounts, especially when passwords are weak or reused (Zhou et al. 2025).

Additionally, system design confidentiality is often overlooked. Sensitive information can be exposed during the development phase if proper modeling protections are not applied. Research on secure modeling shows that applying confidentiality rules and security policies early in the design process helps prevent unauthorized access to system details (Bourdellès, El Hachem, and Sadou 2024).

GuardFile responds to these challenges by using encryption, multi factor authentication, adaptive access control, malware scanning, and secure design principles. These methods ensure that user data remains protected at all stages, from login to storage.

## ***1.2 Motivation***

People store more digital information than ever before. Schoolwork, legal documents, personal photos, financial records, and identity information are frequently uploaded to cloud platforms. Unfortunately, many individuals do not understand how easily this information can be stolen, intercepted, or exposed if strong security measures are not in place. Research shows that encryption and access control are essential for protecting data stored in multi cloud systems (Reece et al. 2024). GuardFile is motivated by the need to give users a safe and simple way to protect their data without requiring deep technical knowledge.

Research also shows that multi factor authentication significantly reduces unauthorized access by adding additional layers of identity verification (Zhou et al. 2025). Combining knowledge factors with biometrics helps ensure that only verified users can access sensitive files. GuardFile implements these protections along with educational tools to encourage users to build safe habits.

Another motivation behind GuardFile is that many users are not aware of how to protect themselves from attacks. A system that both protects and teaches users creates a stronger defense against cyber threats. GuardFile's Security Score helps users understand their vulnerabilities and learn how to stay safe online.

### ***1.3 Related Work***

#### *1.3.1 Multi Factor Authentication and Identity Verification*

Multi factor authentication plays a major role in preventing unauthorized account access. Zhou, Chekole, and Ang (2025) explain that single factor authentication is no longer effective against modern cyber threats. Their study identifies eight important criteria used to evaluate multi factor authentication protocols and emphasizes the need for using more than one verification method. This research supports GuardFile's use of three layered authentication which includes password entry, face recognition, and voice verification.

#### *1.3.2 Cloud Storage and Data Protection*

Vijayarangan and Florence (2023) introduced a secure cloud storage mechanism that uses encryption, data dispersion, and distributed storage to prevent data leaks. They highlight how encryption protects files during transmission and at rest, which aligns with GuardFile's design to keep data unreadable to unauthorized individuals.

Research on multi cloud attacks also shows that web applications can be targeted through man in the middle attacks if data is not encrypted properly during communication (Reece et al. 2024).

GuardFile applies Transport Layer Security to prevent interception and ensure safe data exchange.

### *1.3.3 Access Control and Requirements Confidentiality*

Modern cloud systems must ensure that users only have access to the data they are authorized to view. Fernandes and Martins (2024) propose an adaptive role based access control model that adjusts permissions based on context such as user location, device, and session behavior. This research supports GuardFile's plan to offer flexible file permissions and safe account management tools.

### *1.3.4 Confidentiality as a Core Design Principle*

Confidentiality in system design is also important. Bourdellès, El Hachem, and Sadou (2024) highlight methods for applying the Bell La Padula model to prevent unauthorized access to system information during the design process. These concepts guide GuardFile's development strategy by emphasizing confidentiality at every stage.

### *1.3.5 Integrity Mechanisms in System Development*

Maintaining data integrity is another foundational security objective in system design. The Biba model (Biba, 1977) introduces a formal method for preventing unauthorized or accidental modification of sensitive information. By enforcing “no write-up” and “no read-down” rules, Biba establishes a structured approach to protecting the accuracy and reliability of data. These integrity-focused concepts inform GuardFile's approach to safeguarding stored files, ensuring that data remains unaltered and trustworthy throughout system operation.

### *1.3.6 Threat Modeling Approaches in Modern Systems*

Threat modeling is widely used to anticipate potential attack vectors before system deployment. Shostack (2014) formalized the STRIDE methodology, which categorizes threats into spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege. Using such structured frameworks allows designers to visualize how attackers might exploit system components. GuardFile's design leverages these principles to identify weaknesses early and ensure defenses address realistic, model-driven threat scenarios.

## **2.0 Problem Statement**

As cloud storage usage increases, so do security risks. Many users do not understand how vulnerable their data can be when stored online without strong protections. Weak authentication, poor encryption, insecure communication channels, and improper access control can all lead to data breaches. Research shows that man in the middle attacks, cloud misconfigurations, and system design errors can expose sensitive information (Reece et al. 2024; Bourdellès et al. 2024).

Another problem is the lack of user education. Cloud platforms often provide storage but do not teach users how to recognize unsafe habits such as using weak passwords or ignoring suspicious activity. This leads to higher levels of risk and greater chances of data loss. GuardFile aims to address these problems by implementing strong technical protections and providing user-friendly educational tools.

### **3.0 Proposed Project and Significance**

GuardFile is a cloud based web application designed to provide secure file storage by using encryption, multi factor authentication, adaptive access control, and detailed audit logging. The system prevents unauthorized access while helping users understand safe cybersecurity practices. What makes GuardFile significant is its combination of protection and education.

Encryption protects files during storage and transfer, which directly supports research on cloud confidentiality and secure communication (Al Saeed 2019; Vijayarangan and Florence 2023).

Multi factor authentication follows modern recommendations for reducing account compromises (Zhou et al. 2025). GuardFile also incorporates secure access control strategies, including adaptive permissions and confidentiality modeling, as recommended by Fernandes and Martins (2024) and Bourdellès et al. (2024).

Additionally, GuardFile includes a Security Score that helps users measure their safety level and improve their habits over time. This creates a complete solution that protects data and builds long term awareness, making GuardFile both effective and educational.

## 4.0 Objectives (Step by Step Requirements and UX)

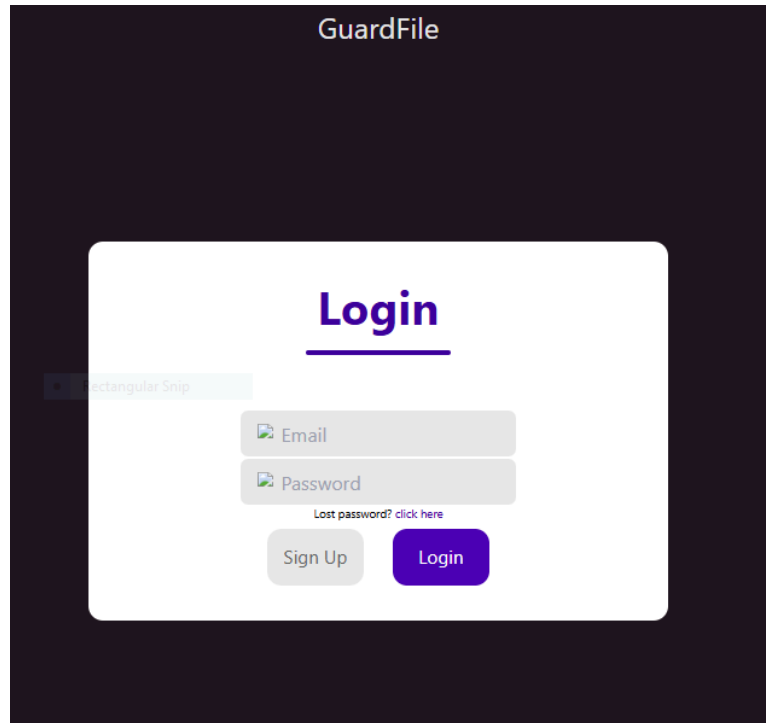
### *Overall Project Objective*

The main objective of GuardFile is to create a secure and user friendly cloud system that protects user data while teaching safe cybersecurity practices. GuardFile must apply encryption, authentication, access control, malware scanning, and education features in a way that is simple and accessible to users.

### *4.1 UX Design and Functions*

#### *4.1.1 Getting Started with GuardFile*

Users access GuardFile through a web browser. The homepage allows them to sign up or log in.

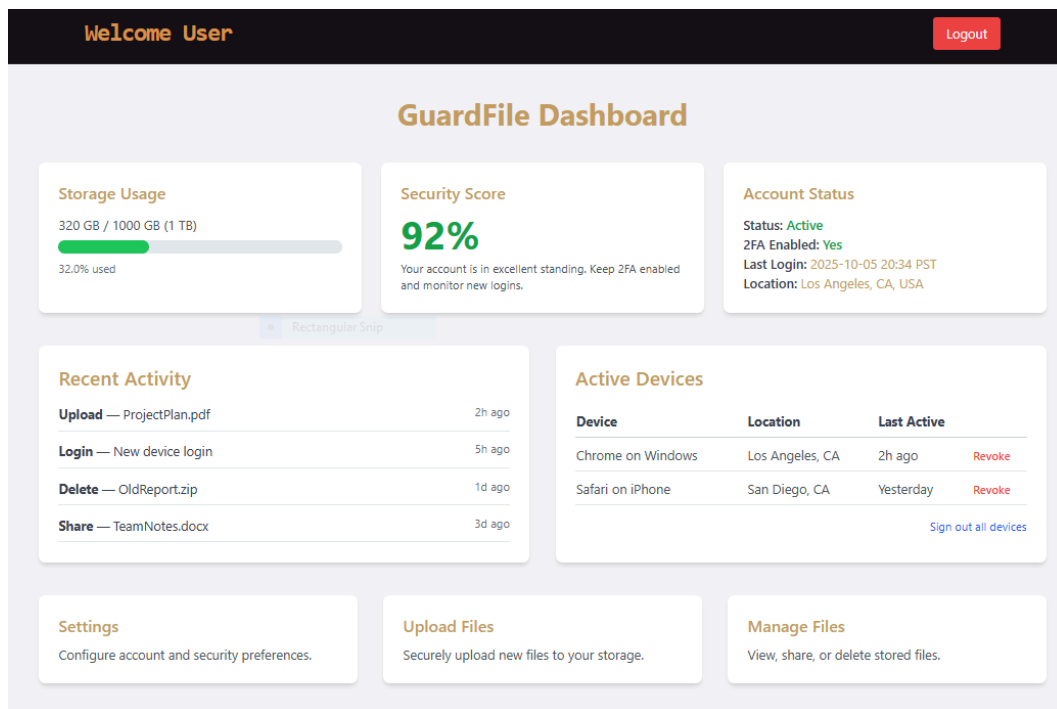


*Figure 1 GuardFile Web Login Page*

New users create an account by entering their first name, last name, and email address. Returning users sign in using their credentials and then complete multi factor authentication steps which include:

1. Password verification
2. Facial recognition using a device camera
3. Voice authentication through the user's microphone

These authentication methods follow strong security guidelines supported by MFA research (Zhou et al. 2025). After logging in, the user receives an email that includes sign in time, IP address, and location to help detect unusual activity. The user is then taken to the dashboard.



*Figure 2 GuardFile Dashboard*

#### 4.1.2 Creating the User Security Profile

By navigating to the account settings page shown in figure 3, users register their face data and voice sample. This information strengthens account protection by combining multiple identity factors. Users are also introduced to the Security Score system, which evaluates their account safety level. The system may suggest improvements such as updating passwords, enabling file level protection, or reviewing login history.

These recommendations are based on modern research in access control and user behavior adaptation (Fernandes and Martins 2024).

**Account Settings**

**Personal Details**  
Name: User  
Email: User@example.com  
Phone: 714-999-9999

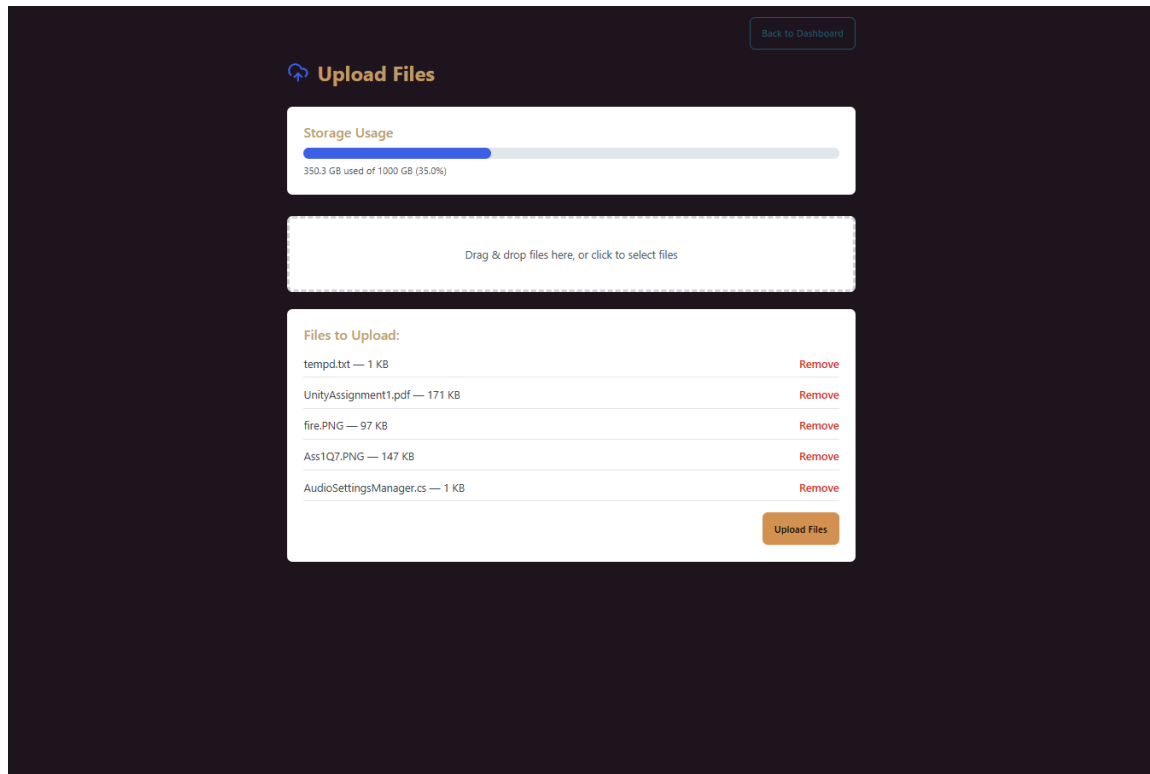
**Storage Information**  
Storage Used: 320GB / 1000GB

**Change Password**

**Update Biometrics**  
Upload Voice Sample  
 No file chosen  
Upload Facial Scan  
 No file chosen

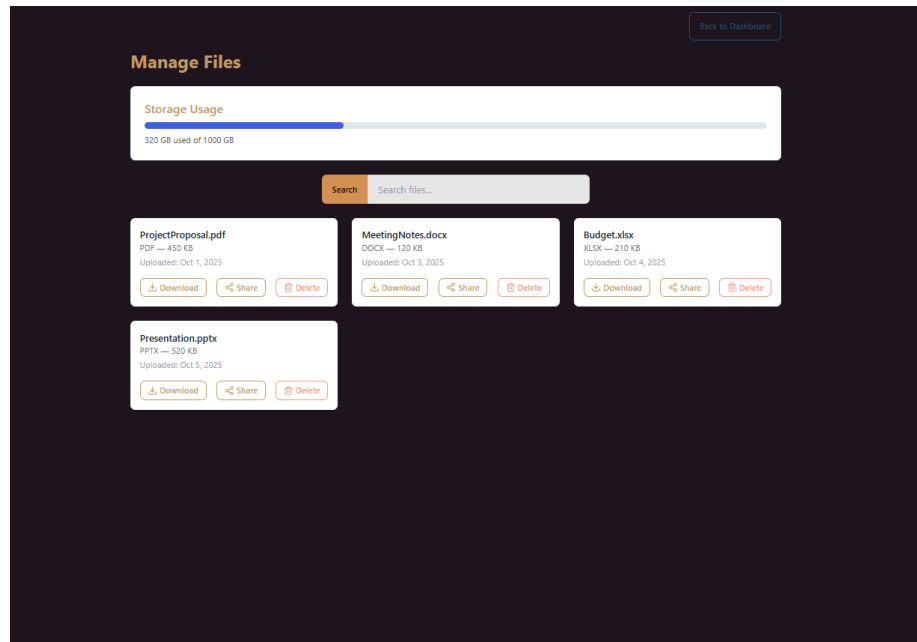
*Figure 3 GuardFile Settings Configuration*

### 4.1.3 Uploading and Managing Files



*Figure 4 GuardFile File Upload Page*

As seen in figure 4, files can be added using the browse button or by dragging and dropping them into the upload area. Below the upload box there is a preview of the type of file along with the size and name. Users may rename files before uploading and add optional password protection to sensitive files. Progress bars show upload status, and the system sends an email after each completed upload.



*Figure 5 GuardFile File Management Interface*

Shown in figure 5, all current files appear in a grid layout. Each file shows metadata including upload date and file size. The buttons on the individual files allow the users to accomplish various objectives including:

- Search for files
- Download files
- Delete files
- Rename files
- Change file permissions
- View detailed information

Before downloading, all files are scanned for malware. This feature follows research on preventing man in the middle attacks and ensuring safe communication across cloud environments (Reece et al. 2024).

## **5.0 Activities**

### ***5.1 Functionality***

GuardFile must support core features including user registration, secure login, and multi-factor authentication to ensure only authorized users can access the system. The platform will allow users to upload and manage files while providing robust permissions control to regulate access based on user roles and privileges.

All system functions must operate reliably and efficiently while maintaining data consistency and integrity. Additional security measures such as audit logging and secure logout will be implemented to track user activity and prevent unauthorized access, ensuring the system meets both functional and security requirements.

### ***5.2 User Friendly Interface***

The system will feature a clean, organized, and intuitive user interface designed to minimize complexity and improve overall usability. Visual elements such as consistent layouts, clear labeling, and logical navigation will be used to help users easily understand and interact with the platform. Special consideration will be given to users who may not have prior experience with cybersecurity concepts, ensuring that essential functions are accessible without requiring technical expertise. Usability testing will be conducted throughout development to gather user feedback, identify potential points of confusion, and refine the interface accordingly. This

iterative approach will help ensure the platform remains efficient, approachable, and easy to use for a wide range of users.

### ***5.3 Security***

Security is the most important part of GuardFile. The system must apply encryption in transit and at rest, follow secure communication standards, protect confidentiality during the design and development process, and enforce strict access control based on user roles and context. These ideas directly follow established research in encryption, cloud security, and modeling confidentiality (Al Saeed 2019; Vijayarangan and Florence 2023; Bourdellès et al. 2024; Fernandes and Martins 2024).

## 6.0 Development Environment

### 6.1 Software Requirements

| Type                  | Software                                |
|-----------------------|---|
| Programming Languages | JavaScript, HTML, CSS                   |
| IDE                   | Visual Studio Code                      |
| Operating System      | Windows 10 or Windows 11                |
| Frameworks            | React for frontend, Node.js for backend |
| Security Tools        | TLS, bcrypt, JSON Web Tokens            |

*Table 1: Software Requirements for GuardFile Web Application*

### 6.2 Hardware Requirements

| Type            | Hardware           |
|-----------------|--------------------|
| Processor       | AMD64 or Intel x64 |
| Processor Speed | 3.5 GHz            |
| RAM             | 8 GB minimum       |
| Internet        | 25 Mbps or higher  |

*Table 2: Hardware Requirements for GuardFile Web Application*

## 7.0 Reports and Products

The final product for this project will be a cloud-based web application designed to support the system's intended functionality. This application will be demonstrated and delivered to the Computer Science Department and the project advisor at the conclusion of the semester. In addition to the working application, supporting materials will include the complete frontend and backend source code, UML and system architecture diagrams, audit log examples, testing documentation, a step-by-step user guide, a comprehensive final report, and an end-of-semester presentation.

## 8.0 Schedule

Below is the planned schedule for completion of this project within the timeframe of the Fall 2025 semester at California State University, Fullerton.

| 2025          | Aug |    | Sept |    |    |    | Oct |    |    |    | Nov |    |    |    |    | Dec |    |   | Summary |         |
|---------------|-----|----|------|----|----|----|-----|----|----|----|-----|----|----|----|----|-----|----|---|---------|---------|
| Tasks:        | 1   | 2  | 1    | 2  | 3  | 4  | 1   | 2  | 3  | 4  | 1   | 2  | 3  | 4  | 5  | 1   | 2  | 3 | Hours   | Percent |
| Research      | 9   | 11 | 13   | 9  |    |    |     |    |    |    |     |    |    |    |    |     |    |   | 42      | 19%     |
| Design        |     | 9  | 6    | 5  | 7  | 9  | 5   |    |    |    |     |    |    |    |    |     |    |   | 41      | 18%     |
| Development   |     |    |      |    | 5  | 5  | 11  | 13 | 11 | 9  |     | 11 | 9  | 9  |    |     |    |   | 83      | 37%     |
| Testing       |     |    |      |    |    |    |     |    | 3  | 4  |     | 5  | 3  | 2  |    |     |    |   | 17      | 8%      |
| Modification  |     |    |      |    |    |    |     |    |    |    |     |    |    |    | 3  | 9   | 5  |   | 17      | 8%      |
| Final Report  |     |    |      |    |    |    |     |    |    |    |     |    |    |    |    | 4   | 11 | 9 | 24      | 11%     |
| Demonstration |     |    |      |    |    |    |     |    |    |    |     |    |    |    |    |     | 7  | 5 | 12      | 5%      |
| Hours:        | 9   | 20 | 19   | 14 | 12 | 14 | 16  | 13 | 14 | 13 |     | 16 | 12 | 11 | 14 | 13  | 11 | 7 | 236     | 100%    |

## 9.0 References

Al Saeed, Imad. “Applying Asymmetric Encryption Algorithms Using Kryptos.” *Journal of Computing Sciences in Colleges*, vol. 34, no. 4, 2019, pp. 110–113.

Bourdellès, M. El Hachem, J., and Sadou, S. “Ensuring Requirements Confidentiality During System Design Modelling.” *Proceedings of the 39th ACM SIGAPP Symposium on Applied Computing*, 2024.

Fernandes, J., and Martins, C. “Adaptive Role Based Access Control for Cloud Applications Using Context Aware Policies.” *Proceedings of the 39th ACM SIGAPP Symposium on Applied Computing*, 2024.

Reece, M., Lander, T., Mittal, S., Rastogi, N., Dykstra, J., and Sampson, A. “Defending Multi Cloud Applications Against Man in the Middle Attacks.” *Proceedings of the 29th ACM Symposium on Access Control Models and Technologies*, 2024.

Vijayarangan, E. V., and Florence, S. M. “Cloud Secure Storage Mechanism Based on Data Dispersion and Encryption.” *Proceedings of the 4th International Conference on Information Management and Machine Intelligence*, 2023.

Zhou, Jianying, Eyasu Getahun Chekole, and Kok Wee Ang. “Unveiling the Covert Vulnerabilities in Multi Factor Authentication Protocols.” *ACM Computing Surveys*, vol. 57, no. 11, 2025, pp. 1–37.