PROGRESS PATH

Christian Jay F. Dimas, Juliah Jane B. Gealon, Xander Jave P. Jamio
College of Computing and Information Sciences
Cor Jesu College, Inc.
Digos City, Philippines

Executive Summary

ProgressPath is a web-based Student Performance Tracking System designed to support Outcome-Based Education (OBE). Its main purpose is to help schools monitor, record, and evaluate student performance in a more organized and efficient way. The system provides a centralized digital platform where academic records, grades, learning activities, and program outcomes can be managed in one place. ProgressPath aims to replace traditional paper-based methods that are time-consuming and prone to errors. By using a digital system, the process of tracking student progress becomes faster, more accurate, and more transparent, which helps improve decision-making and learning outcomes.

The intended users of the system are:

- Students – Students will use the system to view their enrolled courses, check grades, and monitor their academic progress in real time. This helps them understand how their activities and assessments contribute to their learning outcomes.
- Faculty Members – Faculty will use the system to manage subjects, record and submit grades, assess student performance, and map activities to program outcomes. This reduces paperwork, saves time, and minimizes human error.
- Administrators – Administrators will use the system to manage users, departments, and courses, as well as monitor overall academic data. This allows better control and smoother operation of the academic system.

Security is important because the system handles sensitive academic and personal data, such as student records, grades, and learning progress. If this information is not protected, it could be accessed, changed, or misused by unauthorized users.

- Data privacy for students and faculty Accuracy and integrity of academic records
- Trust in the system by users and the institution
- Compliance with laws such as the Data Privacy Act of 2012

Without proper security, the system could face data breaches, loss of information, or unauthorized changes to academic records.

To protect the system and its data, ProgressPath implements several security controls:

- User Authentication – Users are required to log in using a valid username and password to ensure only authorized users can access the system.
- Role-Based Access Control – Access to features is based on user roles (Student, Faculty, Administrator). Each user can only view or modify data that is related to their role.
- Data Encryption – Password to prevent unauthorized access.
- Input Validation – The system checks user inputs to reduce errors and prevent invalid or harmful data from being saved.
- Regular Backups – System data is backed up to prevent data loss in case of system failure or accidental deletion.

This project focuses on the security aspects of the system called **ProgressPath: A Student Performance Tracking System with Outcome-Based Education Standards**, which is designed to monitor, record, and evaluate student academic performance in alignment with outcome-based education standards. The system is primarily used by students, faculty members, and administrators and handles sensitive academic data such as grades, performance records, program outcomes, and user information. From a security perspective, the most critical concerns addressed in this project include unauthorized access to academic records. To mitigate these risks, the project adopts a security strategy based on a role-based access approach, ensuring that users can only access features and data related to their assigned roles. Several security controls were implemented, including user authentication, role-based access control, input validation, and activity logging, which help protect the system's confidentiality, integrity, and availability. Despite these measures, some residual risks remain due to time constraints and limited implementation scope. These risks are documented and can be addressed through future security enhancements and system updates.

Keywords: Outcome-Based Education, Web Application Security, Student Performance System

I. Security Overview

A. System Context

What does the system do?

The system helps schools monitor, record, and evaluate student performance in a more organized and efficient way. The system provides a centralized digital platform where academic records, grades, learning activities, and program outcomes can be managed in one place.

Who uses the system?

Students – Students will use the system to view their enrolled courses, check grades, and monitor their academic progress in real time.

Faculty Members – Faculty will use the system to manage subjects, record and submit grades, assess student performance, and map activities to program outcomes.

Administrators – Administrators will use the system to manage users, departments, and courses, as well as monitor overall academic data.

Why is security important for this system?

Security is important because the system stores sensitive academic information, such as student grades, performance records, and personal details. If this data is accessed or modified without permission, it may lead to incorrect evaluations, privacy violations, and loss of trust in the system. Proper security also ensures that the system remains reliable, accurate, and protected from misuse or data loss. ProgressPath is a web-based Student Performance Tracking System used by students, faculty members, and administrators. Each user has a specific role with different access permissions. From a security perspective, the system must protect academic records by ensuring that only authorized users can view or modify data. Security controls are essential to maintain data integrity, protect user privacy, and ensure the system supports outcome-based education processes in a safe and dependable manner.

What are the most security-critical assets?

The most security-critical assets of the ProgressPath system include user credentials, student academic records, program outcome data, and administrator and faculty privileges.

- User credentials (usernames and passwords) are critical because they control access to the system.
- Student academic records, such as grades, course enrollments, and progress reports, are sensitive personal and academic data.
- Program outcome mapping and assessment data are important because they are used to evaluate compliance with Outcome-Based Education standards.

- Administrator and faculty privileges are critical because these accounts can modify system data, manage users, and configure system settings.

If these assets are compromised, the system may suffer serious issues related to confidentiality, integrity, and availability.

- Confidentiality: Unauthorized access to user credentials or academic records may expose private student information. This can lead to data privacy violations and loss of trust from students and faculty.
- Integrity: If attackers alter grades or program outcome data, the accuracy of academic records will be affected. This may result in incorrect evaluations of student performance and program compliance.
- Availability: If administrator accounts are misused or the system is attacked, system services may become unavailable. This can prevent users from accessing academic data, submitting requirements, or performing important tasks.

The system under study is ProgressPath: A Student Performance Tracking System with Outcome-Based Education Standards, which functions as a web-based application for monitoring, recording, and evaluating student academic performance. It centralizes student records and program outcome data to support Outcome-Based Education processes. It is used by students, faculty members, and administrators to view academic progress, manage courses and subjects, record grades, map program outcomes, and generate academic reports.

Security is important in this system because it handles sensitive academic and personal data, such as student grades and performance records. Unauthorized access or misuse of the system may affect academic integrity, data accuracy, and the reliability of institutional records.

The most security-critical assets of the system include:

- User credentials (usernames and passwords)
- Student academic records and grades
- Faculty and administrator access privileges

If these assets are compromised, the impact would affect confidentiality, integrity, and availability of the system. Sensitive student information may be exposed, leading to privacy violations and loss of trust. Data such as grades and academic records could be altered, which would affect the integrity of academic results. In addition, unauthorized access or system attacks may cause service disruption, preventing users from accessing the system when needed and resulting in possible data loss or system misuse.

B. Threat Landscape

The identified threats are realistic for the ProgressPath system because it is a web-based application that uses PHP, MySQL, and a web server and will be accessed by users over a network. Since the system handles student records, grades, and program outcome data, it is exposed to common web threats such as unauthorized access, weak passwords, SQL injection, and data leakage. The use of role-based access (students, faculty, administrators) also makes the system vulnerable if access controls are not properly enforced. Among the identified threats, unauthorized access to user accounts poses the highest risk. This threat is highly likely because users may use weak passwords or share login credentials. Its impact is also severe, as attackers could view or modify sensitive academic records, change grades, or misuse administrator privileges. Such incidents would affect data confidentiality, integrity, and system reliability, making this threat the most critical for the system.

The ProgressPath system faces several cybersecurity threats common to web-based academic systems. Unauthorized access is a major risk, where attackers may gain entry through weak passwords or stolen credentials and access sensitive student data. Injection attacks, such as SQL injection, can occur if user inputs are not properly validated, allowing attackers to manipulate or damage the database. Data leakage is also a concern if academic records are not properly protected through encryption and access controls. Server or system misconfiguration can expose the system to vulnerabilities if updates, permissions, or security settings are poorly managed. Other possible threats include cross-site scripting (XSS), which can be used to steal user session data, session hijacking, where active user sessions are taken over, and denial-of-service (DoS) attacks, which can disrupt system availability. These threats highlight the need for strong authentication, secure coding practices, proper access control, and regular security maintenance.

These cybersecurity threats are realistic for ProgressPath because of its web-based architecture, technology stack, and deployment environment. The system is built using PHP, MySQL, HTML, CSS, and JavaScript, which are commonly targeted by attackers if input validation, session management, or access controls are not properly implemented. For example, SQL injection attacks are possible because MySQL databases rely on queries that can be manipulated if user input is not sanitized. Similarly, cross-site scripting (XSS) and session hijacking are risks due to the use of client-side JavaScript and web sessions. The system is deployed on a local server using Laragon, which, while controlled, can still be vulnerable if server settings are misconfigured, software updates are missed, or permissions are too permissive. Unauthorized access and misuse of privileges are realistic because the system has multiple user roles (students, faculty, administrators), and incorrect role assignments or weak passwords could allow users to perform actions beyond their intended permissions. Finally, data leakage is possible because the system stores sensitive student and academic records, and without proper encryption and secure backup procedures, this data could be exposed or accessed by attackers. These threats are therefore directly linked to the system’s technologies, user management design, and deployment setup, making them practical concerns that must be addressed.

The threat that poses the highest risk to ProgressPath is unauthorized access. This is because it has both a high likelihood and a high impact. With multiple user roles (students, faculty members, administrators) and sensitive academic data stored in the system, weak passwords, stolen credentials, or misconfigured access controls could allow attackers or unauthorized users to access, modify, or delete important records. The impact of such a breach is severe: it could compromise student grades, learning outcomes, and institutional records, potentially affecting academic decisions and institutional accreditation. Compared to other threats like SQL injection or XSS, which require specific technical skills, unauthorized access can occur more easily through social engineering or weak password practices, making it the most critical threat to address.

The most relevant cybersecurity threats applicable to this system are:

- Unauthorized Access
- SQL Injection
- Data leakage and misuse of user privileges

These threats are realistic because ProgressPath is a web-based system built with PHP, MySQL, HTML, CSS, and JavaScript, which can be targeted if user inputs are not properly validated or if access controls are weak. The system has multiple user roles (students, faculty, administrators), making unauthorized access and misuse of privileges possible if passwords are weak or roles are misconfigured. It is deployed on a local server using Laragon, so improper server settings or missed updates could also lead to data leakage or security breaches.

Among these threats, unauthorized access poses the highest risk because it is both likely to occur and can have severe consequences. With multiple user roles and sensitive academic data stored in the system, weak passwords or misconfigured access permissions could allow attackers or unauthorized users to view, modify, or delete important student records. The impact of such a breach is significant, potentially affecting academic decisions, student grades, and institutional records, making it the most critical threat to address.

C. Overall Security Strategy

The overall security strategy used in ProgressPath is a secure-by-design and defense-in-depth approach. The system was designed from the start with security in mind, ensuring that user authentication, role-based access control, and data validation are built into the core functionality. Multiple layers of protection are implemented, including input validation to prevent injection attacks, secure handling of user credentials through hashed password storage, and proper configuration of the local server environment. This layered strategy helps reduce the risk of unauthorized access, data leakage, and other cyber threats, while providing a secure and reliable platform for students, faculty, and administrators.

The project adopted a risk-based security strategy combined with a defense-in-depth approach. Security risks were first identified based on how the system is used, who accesses it, and what type of data it handles. The highest priority was given to risks that could affect student records, grades, and personal information, since these have the greatest impact if compromised. User authentication and role-based access control were prioritized to prevent unauthorized access and misuse of privileges. Next, input validation and secure database handling were applied to reduce the risk of injection attacks and data manipulation. Basic server security measures, such as proper configuration and access restrictions, were also implemented early to protect the system during deployment on the local server.

Which security principles guided your decisions?

The security decisions in the project were guided by several important security principles. Confidentiality was applied to protect student records, grades, and personal information so that only authorized users can access them. Integrity was ensured by making sure that academic data, such as grades and assessments, cannot be changed without proper permission. Availability was also considered so the system remains accessible, especially during important periods like grading and evaluations. The principle of least privilege was followed by giving users only the access they need based on their role, such as student, faculty, or administrator, which helps reduce misuse. Lastly, defense in depth was used by applying multiple security layers, including login controls, role-based access, and input validation, so that if one security layer fails, others can still protect the system.

The security decisions for the system were guided by a risk-based approach, defense in depth, and secure-by-design principles. A risk-based approach was used to focus first on the most likely and harmful threats, such as unauthorized access to student records and misuse of user privileges. Because the system handles sensitive academic data, controls like user authentication and role-based access were prioritized early in development. Defense in depth was applied by using multiple layers of security, including login verification, access controls, input validation, and database protections, so that even if one layer fails, others can still protect the system. Secure-by-design was followed by considering security during the system design phase rather than adding it later, ensuring that features like data validation, secure sessions, and proper user roles were built into the system from the start.

Which planned controls were not implemented and why?

Some planned security controls were not implemented due to time, complexity, and scope limitations. Features such as multi-factor authentication (MFA) were not added because they require extra setup, user training, and more development time. Advanced intrusion detection and monitoring tools were also not implemented since they are complex and need specialized resources to configure and maintain. In addition, full data encryption at rest and automated security testing tools were planned but not included because the system was deployed on a local server with limited configuration options and a fixed project timeline. Instead, the project focused on core security measures like user authentication, role-based access control, and input validation to ensure basic system security within the available time and scope.

Some security controls were planned but not implemented due to time, complexity, and project scope limitations. These include multi-factor authentication (MFA), and full encryption of data at rest. These controls were deferred because they require additional infrastructure, longer development time, and more technical expertise to properly implement and maintain. Instead of full implementation, these controls were documented in the system's security plan and future improvement section of the project documentation. The documentation explains how these controls can be added in later versions, including their purpose, expected benefits, and suggested implementation approaches. This ensures that future developers or the institution's IT staff can clearly understand the security gaps and prioritize these controls when the system is expanded or deployed on a larger scale.

The overall security strategy adopted in this project is a risk-based and secure-by-design approach. Risks were prioritized based on likelihood and potential impact within the academic scope, which explains why user authentication, role-based access control, and input validation were implemented first.

The security decisions were guided by the following principles:

- Defense in depth
- Risk-based security
- Least privilege

Some security controls were planned but not implemented, such as multi-factor authentication and full data encryption at rest, due to time, complexity, and project scope limitations. These controls are instead documented in the security roadmap.

II. Residual Risks & Limitations

What security risks remain in the system?

The security risks that remain in the system include monitoring gaps, as the system does not have automated intrusion detection or continuous security monitoring. Limited testing is also a risk since full penetration testing and vulnerability scans were not performed. Additionally, human factors such as weak passwords, accidental sharing of login credentials, or misuse of access privileges could compromise system security. These risks highlight areas where future improvements and additional controls are needed.

The remaining security risks and limitations of the system include several areas that could affect its overall safety. Monitoring gaps exist because the system does not have automated intrusion detection or continuous activity tracking, which makes it harder to detect unauthorized access quickly. Some planned security controls, such as multi-factor authentication and full data encryption, were not implemented, leaving parts of the system less protected. Limited testing is also a factor, as full penetration testing and vulnerability scans were not conducted, which may allow unnoticed security weaknesses. Human factors, such as weak passwords, accidental sharing of login credentials, or improper use of access privileges, could lead to security breaches.

Why are these risks acceptable at this stage?
These risks are acceptable at this stage because the system is still a prototype intended for academic purposes and limited deployment within the school's local network. The academic scope of the project means it is primarily used for testing, learning, and demonstrating functionality, rather than handling high-stakes, large-scale operations. Since the system is not yet deployed widely or exposed to the public internet, the likelihood and impact of security breaches are relatively low. This allows the development team to focus on core functionality, usability, and essential security measures, while leaving advanced controls and full-scale protections for future iterations.

These risks are acceptable at the current stage because the system is still in a prototype phase and primarily serves an academic purpose. Its deployment is limited to the school's local network, which reduces exposure to external threats and lowers the likelihood of serious security incidents. Additionally, the system is not yet fully mature, so the focus has been on implementing core functionalities and essential security measures. Given these factors, the remaining risks are manageable and can be addressed in future versions as the system matures and is deployed on a larger scale.

How will these risks be addressed in the future?
These risks will be addressed in the future by following a security improvement roadmap. Planned actions include implementing multi-factor authentication, enabling full data encryption, adding automated monitoring and intrusion detection, and conducting regular penetration testing. User training and stricter access policies will also help reduce human-factor risks. These improvements will strengthen the system's security as it moves from a prototype to a fully deployed solution.

These risks will be addressed in the future according to the Security Roadmap, which includes implementing multi-factor authentication, full data encryption, automated monitoring, and regular security testing, as well as improving user access controls and providing training for safe system use.

Despite the implemented controls, the system still has remaining risks such as limited real-time monitoring for unauthorized access, incomplete encryption of stored data, reliance on users to maintain strong passwords, potential vulnerabilities in third-party libraries, and the absence of full-scale penetration testing.

These risks are considered acceptable at the current stage because ProgressPath is still a prototype used for academic purposes and deployed only within the school's local network. Its limited deployment reduces exposure to external threats, and the system's primary goal at this stage is to test core functionality and usability rather than handle high-stakes, real-world operations. This allows the development team to focus on essential features and basic security measures while planning more advanced protections for future versions.

In future versions, these risks will be addressed by implementing multi-factor authentication, full data encryption, automated monitoring and intrusion detection, regular security testing, and stricter user access controls, as outlined in the system's Security Roadmap.

III. Secure System Architecture

A. Applied Secure Design Principles

In ProgressPath, secure design principles were applied to ensure the system is protected from potential threats. The least privilege principle was used by giving users only the access they need based on their roles, so students, faculty, and administrators cannot perform unauthorized actions. Defense in depth was implemented by adding multiple layers of protection, including user authentication, role-based access control, input validation, and basic logging, so that if one layer fails, others still provide security. Secure defaults were applied by configuring the system to use safe settings out of the box, such as requiring strong passwords, restricting access to sensitive data, and disabling unnecessary features. These principles collectively help make the system more resilient against common security risks.

In ProgressPath, these secure design principles shaped key architectural decisions. Least privilege guides how the system assigns permissions, students can only view their own progress, faculty can manage course tasks and grades, and administrators handle system-wide settings, ensuring no user can access data beyond their role. Defense in depth influenced the layering of security measures, such as validating user input, implementing role-based access at both the front-end and database, and keeping logs of critical actions. Secure defaults affect initial configurations: new accounts are created with strong password requirements, sensitive modules are hidden until access is granted, and unnecessary services are disabled by default. Together, these principles led to a system structure where sensitive data is protected, modules are isolated to limit risks, and access is carefully controlled, all while keeping the system usable for its intended academic purposes.

The system applies secure design principles such as:

- Least Privilege: Access is limited based on user roles. Students can only view their own grades and progress, faculty can manage course tasks and assessments, and administrators have control over system-wide settings. This ensures that users cannot access information or perform actions outside their responsibilities.

- Defense in Depth: Multiple layers of protection are used throughout the system. These include user authentication, role-based access control at both the application and database levels, input validation to prevent malicious data, and logging of important actions. If one layer fails, the others continue to provide protection.

- Secure Defaults: The system is configured with safe settings from the start. New accounts require strong passwords, sensitive modules are hidden until proper access is granted, unnecessary features are disabled, and default user roles restrict access to critical data. This reduces the risk of accidental or unauthorized access.

How did these principles influence system design decisions?
These principles directly shaped how ProgressPath was designed. Least Privilege influenced the creation of separate user roles and permissions, ensuring that students, faculty, and administrators could only access the functions and data they need. Defense in Depth guided the placement of security measures at multiple levels, such as adding authentication, input validation, and logging both in the application and database, so that a breach in one layer would not compromise the entire system. Secure Defaults affected how the system was initially configured, with strong password requirements, hidden sensitive modules, and disabled unnecessary features, ensuring that the system is safe to use from the moment it is deployed. These decisions helped balance security with usability while protecting academic data and system integrity.

B. Security Architecture Overview

The security system architecture of ProgressPath is designed to protect sensitive academic data while supporting smooth system operations. The architecture uses a layered approach, starting with user authentication and role-based access control to ensure that students, faculty, and administrators can only access data and features relevant to their roles. Input validation and logging mechanisms are implemented at both the application layer and the database layer to prevent malicious activity and track important actions. The system modules, such as academic records, task management, and consultation tracking, are isolated, reducing the risk that a compromise in one module affects others. Secure defaults, like strong password enforcement and disabled unnecessary services, are applied at deployment. Overall, the architecture combines access control, layered protections, and module isolation to create a secure, manageable environment for academic data management.

Security-relevant components in ProgressPath include the authentication module, academic records database, and administrator interfaces, which are separated by role-based access controls and module isolation. These boundaries ensure that students can only access their own progress and submissions, faculty can manage courses and grades, and administrators can control system-wide settings, preventing unauthorized access and protecting sensitive academic data.

C. Security Controls Mapped to Components

System Component	Security Control	Security Objective
User Authentication & Access Control Module	Role-Based Access Control (RBAC)	Ensure users only access features allowed for their role (Least Privilege)
Enrollment & Class Management Module	Activity logging (who changed what and when)	Provide accountability and traceability for changes (Integrity)
Program Outcome Mapping Module	Permission restriction (Admin only)	Prevent students from editing outcome mapping (Integrity)
Submission & Grading Module	Access restriction (only assigned faculty can grade)	Prevent unauthorized grade edits (Integrity)

D. Design Limitations & Trade-Offs

In ProgressPath, one security-related design trade-off was not implementing multi-factor authentication (MFA) for user logins during the initial release. While MFA would strengthen account security, it was deferred due to time constraints of ProgressPath, as well as concerns that requiring students and faculty to use additional authentication steps might slow down access to tasks, grades, and consultation logs. To maintain security in the meantime, ProgressPath relies on strong passwords, role-based access controls, and secure default settings, with MFA planned for future versions as part of the system's Security Roadmap.

What risk does this trade-off introduce?

This trade-off introduces the risk of unauthorized access to user accounts in ProgressPath. Without multi-factor authentication, if a password is guessed, stolen, or reused from another system, an attacker could potentially log in and view or modify sensitive academic data, such as student grades, course tasks, and consultation records. While role-based access limits the scope of damage, the absence of MFA makes the system more vulnerable to account compromise compared to a fully secured environment.

The missing multi-factor authentication (MFA) in ProgressPath increases the risk that unauthorized users could gain access to accounts if passwords are weak, stolen, or reused. This could allow attackers to view or modify sensitive academic data, such as student grades, course tasks, and consultation records. Even though role-based access limits what a user can do, the lack of MFA makes it easier for attackers to bypass the first layer of security, potentially compromising the confidentiality, integrity, and trustworthiness of the system's data.

How will this risk be mitigated in the future?
This risk will be mitigated in future versions of ProgressPath by implementing multi-factor authentication (MFA) for all user accounts, requiring an additional verification step during login. Additional measures, such as regular password policies, automated account lockouts for multiple failed login attempts, and user education on strong password practices, will also be applied. These improvements are planned in the system's Security Roadmap to enhance account security and reduce the likelihood of unauthorized access.

In future iterations of ProgressPath, this risk will be mitigated by implementing multi-factor authentication (MFA), which will require users to provide a second form of verification in addition to their password. Additional compensating measures will include enforcing stronger password rules, automatic account lockouts after multiple failed login attempts, and periodic user training on secure password practices. These planned controls are documented in the Security Roadmap and aim to reduce the likelihood of unauthorized access while maintaining the system's usability.

One security-related trade-off in the system is not implementing multi-factor authentication (MFA) during the initial release.

This introduces the risk of unauthorized access to user accounts, which could lead to exposure or modification of sensitive academic data such as student grades, course tasks, and consultation records.

In future iterations, this risk will be mitigated by implementing multi-factor authentication (MFA), enforcing strong password policies, and adding account lockouts after multiple failed login attempts.

IV. Privacy Impact Assessment (PIA)

A. Personal Data Inventory

Personal Data	Purpose of Processing	Storage Location
Full name	Used to identify students, faculty, and administrators within the system.	MySQL Database
Username	Used for user authentication and system login.	MySQL Database
Password	Used to verify user identity during login	MySQL Database
Grade and Academic Performance	Used to evaluate student progress and program outcomes.	MySQL Database
Program outcome data	Used for Outcome-Based Education assessment.	MySQL Database
Submission Records	Used to track student activities and outputs.	MySQL Database

B. Data Flow Description

In ProgressPath, personal data will flow through the system in this order: collection → storage/transfer → access/use → retention → disposal. Personal data will be collected when users log in and when faculty/admin encode or update academic records (e.g., profiles, enrollments, grades, and program outcome data). The data will be stored in the MySQL database hosted on the school server. It will be accessed only by authorized users through role-based access control (Student, Faculty, Administrator). Data will be retained for academic and reporting purposes based on institutional needs, and it will be disposed of securely when it is no longer required.

Who can access personal data and under what conditions?

- Students can access their own profile, classes, grades, and submissions progress only after successful login.
- Faculty can access student records only for the classes/subjects assigned to them, and only after login. They can encode grades and evaluate submissions within their assigned scope.
- Administrators can access broader records (user accounts, departments, courses, and system data) only after login and for official school management purposes.
- Access is allowed only when:
  - the user is authenticated (valid login/session),
  - the user has the correct role permission,
  - and the data is within the user's authorized scope (e.g., "my account," "my class").

Data Collection	**Storage & Transfer**	**Use**	**Retention**	**Disposal & Destruction**
Users will enter personal data through login forms and system inputs (e.g., user profiles, enrollment info, grade encoding, submissions, and outcome mapping).	Data will be stored in the MySQL database. Data will be transferred between the browser and server through web requests.	Data will be used to authenticate users, show dashboards, manage classes, record grades ,and map program outcomes.	Data will be kept as long as it is needed for academic records, reporting, and institutional requirements (e.g., current semester).	When data is no longer needed, accounts/records will be deactivated or deleted by an administrator, and database records will be securely removed. Old backups containing deleted data will be rotated/expired based on the backup schedule.

C. Privacy Risks and Mitigation

Privacy Risk	***Likelihood***	***Impact***	***Mitigation***
Unauthorized access to student records	Medium	High – may expose grades, personal data, and academic history	Enforce strong login authentication, role-based access control, and session timeout
Weak or shared passwords	High	High – attackers may take over user accounts	Apply password strength rules, login attempt limits, and user education
Data leakage through improper access permissions	Medium	High – students or staff may see data not meant for them	Implement strict role-based access and authorization checks on every page
SQL injection or malicious input	Medium	High – attackers may read or modify database records	Use input validation, prepared statements, and server-side filtering
Accidental data exposure by faculty or admin	Low	Medium – privacy breach due to human error	Add confirmation prompts, access logs, and activity tracking
Data loss due to system failure	Low	High – loss of academic records and reports	Perform regular database backups and recovery testing

D. Legal Compliance Considerations

How does the system align with Philippine cybersecurity and data protection laws? Explain how the system aligns the legal framework on cybersecurity in the Philippines.

The system aligns with Philippine cybersecurity and data protection laws, particularly the Data Privacy Act of 2012 (RA 10173), by applying basic safeguards to protect user data. Access control is implemented through role-based accounts, ensuring that only authorized users such as administrators, teachers, and students can access specific information. Data minimization is practiced by collecting only necessary student and academic data required for monitoring performance. Security measures such as user authentication, controlled database access, and secure handling of records help prevent unauthorized access and data misuse. These practices support the legal framework on cybersecurity in the Philippines by promoting data confidentiality, integrity, and responsible data management within the system.

What compliance gaps exist and what is planned? Identify any gaps in compliance.

The system aligns with Philippine cybersecurity and data protection laws by implementing safeguards such as secure login authentication, role-based access control, encrypted data storage, and regular database backups.

These measures support the requirements of the Data Privacy Act of 2012 (RA10173) by protecting personal and academic information, and the Cybercrime Prevention Act of 2012 (RA 10175) by preventing unauthorized access and system misuse.

However, some compliance gaps still exist. These include the absence of a formal Privacy Impact Assessment (PIA), a documented incident response plan, and regular vulnerability assessments or penetration testing. In addition, advanced security features such as multi-factor authentication (MFA) and real-time security monitoring are not yet fully implemented.

These gaps are planned for future implementation. The development team intends to conduct a Privacy Impact Assessment, establish a formal incident response procedure, perform regular security testing, and introduce stronger authentication and monitoring tools to further strengthen system compliance and security.

V. Security Technologies, Policies, and SETA

A. Security Technologies

- **User Authentication (Username and Password Login)**
  This ensures that only registered students, faculty, and administrators can access the system.

- **Role-Based Access Control (RBAC)**
  This limits access based on user roles (Student, Faculty, Administrator). Each user can only access features related to their responsibilities.

- **Password Hashing**
  Passwords are stored in encrypted form to protect user credentials if the database is compromised.

- **Input Validation and Server-Side Checks**
  This prevents invalid or malicious data from being saved in the database.

- **Database Backups**
  Regular backups protect against data loss caused by system failure or accidental deletion.

These technologies are appropriate because the system handles sensitive academic data such as grades, performance records, and user information. They reduce risks such as unauthorized access, data leakage, and system misuse.

B. Security Policies

The following security policies apply to the system:

- **Password Policy**
  Requires users to create strong passwords and keep them confidential.

- **Access Control Policy**
  Ensures users can only access features and data based on their assigned role.

Data Protection Policy
Protects personal and academic data from unauthorized access or misuse.

These policies support secure behavior by guiding users and administrators on how to properly protect accounts, handle data responsibly, and follow security best practices.

C. Security Education, Training, and Awareness (SETA)

SETA activities focus on administrators, faculty, and users of the system.

SETA activities include:

- Basic security orientation for administrators and faculty
- Guidelines on proper password use and account protection
- Awareness reminders about data privacy and responsible system use.

These activities help reduce human-related security risks by teaching users how to avoid common mistakes such as sharing passwords, leaving sessions open, or entering incorrect data.

VI. Final Security Assessment Summary

Overall security posture: Moderate

Main security strengths:

- Role-based access control
- Secure login authentication
- Input validation
- Regular database backups

Main security weaknesses:

- No multi-factor authentication (MFA) yet.
- No formal incident response plan
- No regular vulnerability scanning

Remaining risks and justification for acceptance:
Some risks remain due to time and scope limitations of the capstone project. These risks are acceptable at this stage because the system is still in development and will be improved before full deployment.

Overall, the system's security posture can be described as improving. The main strengths are strong access control and data protection, while the main weaknesses involve the need for more advanced monitoring and response mechanisms. The remaining risks are acceptable for now because future security improvements are already planned.

B. INFORMATION SECURITY ROADMAP

I. Current Security Baseline

Currently, the system has the following security controls:

- User login authentication
- Role-based access control
- Input validation
- Database backups

The following controls are assumed or planned but not yet fully implemented:

- Multi-factor authentication (MFA)
- Real-time security monitoring
- Incident response plan
- Regular vulnerability scanning

These planned improvements will be implemented in future versions to further strengthen system security.

II. Security Integration Model

Security Layer	**Current State**	**Identified Risk**	**Planned Improvement**	**Priority**
Secure Design	The system follows role-based access and basic authentication during login.	Weak passwords or shared accounts may allow unauthorized access.	Add password strength rules, login attempt limits, and optional multi-factor authentication.	**H**
Architecture	The system uses a PHP backend, MySQL database, and a web-based frontend.	Single-server setup may cause downtime if the server fails.	Implement backup servers and database replication for high availability.	**M**
Privacy & PIA	User data is stored in the database and protected through login access.	Risk of data exposure if access controls fail or data is leaked.	Conduct a Privacy Impact Assessment (PIA) and improve data encryption policies.	**H**
Vulnerability Assessment	Basic testing is performed during development.	Security flaws may remain undetected.	Conduct regular vulnerability scanning and penetration testing.	**H**
Incident Response	No formal incident response plan is currently documented.	Delayed response during a security breach may increase damage.	Create an incident response plan and define reporting procedures.	**M**
Monitoring & IMprovement	System logs user actions and login activity.	Attacks or misuse may go unnoticed without real-time monitoring.	Add real-time monitoring and alert systems.	**M**

Governance (Tech, Policy, SETA)	Basic security policies are followed during development.	Users may not be aware of security best practices.	Conduct security awareness training and define IT security policies.	**L**

III. Individual Contribution

Each student must complete this section individually.

	Christian Jay F. Dimas
	User Authentication and Access Control layer. This layer is important because it is the first line of defense of the system. It ensures that only authorized students, faculty, and administrators can access ProgressPath. Without proper login and access control, anyone could enter the system and view or modify academic records.
	One key risk I discovered is the possibility of weak or shared passwords, which could allow unauthorized users to access student grades and records.
	I proposed adding password strength rules and login attempt limits to prevent brute-force attacks and reduce the risk of account compromise.
	I learned that security is not only about technology but also about how users interact with the system. Even a well-designed system can be compromised if users are careless with their passwords. Information security must protect data, users, and system reliability at the same time.
	Juliah Jane B. Gealon
	Infrastructure and Monitoring layer. This layer is important because it ensures that the system remains available and operational. If the server fails or is attacked, users will not be able to access their academic data.
	One key risk is server downtime or data loss due to system failure or cyberattack.
	I proposed implementing regular backups, server monitoring, and an incident response plan to quickly recover from failures or security incidents.
	I learned that security is not just about preventing attacks, but also about being prepared when something goes wrong. A secure system must be able to recover quickly and continue serving users without losing important data.

	Xander Jave P. Jamio
	Application and Database Security layer. This layer is important because it protects the system’s core data such as grades, program outcomes, and user records. If this layer is weak, attackers could modify or steal academic data.
	One key risk is SQL injection, which could allow attackers to access or manipulate database records.
	I proposed using prepared statements, input validation, and parameterized queries to prevent malicious inputs from reaching the database.
	I learned that building secure systems requires thinking like an attacker. Developers must always assume that users may input harmful data, and the system must be designed to block these attacks before they reach the database.