Student Assignment Brief

CONFIDENTIAL DOCUMENT

This document is intended solely for Softwarica College of IT & E-Commerce students for their own use in completing their assessed work for this module. It must not be passed to third parties or posted on any website.

Contents

Assignment Information

Module Name:	Security
Module Code:	ST6005CEM
Assignment Title:	Coursework: Individual Report
Assignment Due:	1 August 2025 (11:55 PM)

Assignment Credit:	20 credits
Word Count:	2000 words
Assignment Type:	Coursework
Grading:	50% (Core Assessment)

Assessment Overview

You will be provided with an overall grade between 0% and 100%. You have one opportunity to pass the assignment at or above 40%.

Important Notice

The work you submit for this assignment must be your **own independent work**. More information is available in the 'Assignment Task' section of this assignment brief.

Assessed Module Learning Outcomes

The Learning Outcomes for this module align with the marking criteria which can be found at the end of this brief. Ensure you understand the marking criteria to ensure successful achievement of the assessment task.

1. Critically evaluate a range of encryption and authentication methods for a given set of requirements.
2. Utilize systematic knowledge to create secure environments at the host or network level.
3. Develop and evaluate software that addresses the most common and most severe security concerns.

Assignment Task

Task and Mark distribution: Total Marks 100

Web Application:

Your task is to design and develop a unique and secure web application that caters to a specific user need. The application should be original in concept and implementation, avoiding the use of premade templates or AI-generated content to prevent the introduction of business logic flaws.

Core Features:

1. **User-Centric Design:**
   Design an intuitive interface that enhances the user experience, ensuring accessibility and ease of navigation for all users.
2. **User Registration and Authentication:**
   Implement a secure user registration process with a focus on robust authentication methods, including Multi-Factor Authentication (MFA) and mechanisms to prevent brute-force attacks, such as account lockout after a specified number of failed login attempts.
3. **Customizable User Profiles:**
   Allow users to personalize their profiles in alignment with the application's purpose, ensuring that these profiles are secure and meet user needs.
4. **Secure Transaction Processing:**
   Ensure secure and encrypted transaction handling through either third-party services like Stripe or a custom-built solution, maintaining data integrity and security during all transactions.
5. **Activity Logging:**
   Implement comprehensive activity logging to support auditing, troubleshooting, and security reviews by maintaining detailed records of all user actions within the application.

Security Features:

1. **Password Security:**
   Enforce a robust password policy that includes the following:
   a. **Password Length and Complexity:** Define minimum and maximum length requirements, and require a mix of uppercase letters, lowercase letters, numbers, and special characters.
   b. **Password Reuse and Expiry:** Prevent users from reusing recent passwords and enforce password expiry to enhance security.
   c. **Real-Time Strength Assessment:** Provide users with feedback on password strength during registration or password changes.
2. **Brute-Force Prevention:**
   Throughout the system, implement mechanisms to prevent brute-force attacks, such as limiting the number of requests a user can make within a specific time frame to protect against repeated unauthorized access attempts.
3. **Access Control (RBAC):**
   Implement Role-Based Access Control to manage user permissions effectively, ensuring that users have access only to the resources they need.
4. **Session Management:**
   Ensure secure session creation, handling, and expiration policies, including the use of secure session headers and automatic session expiration to protect user sessions from hijacking.
5. **Encryption:**
   Store all critical user information, such as passwords and sensitive data, in an encrypted form within the database to safeguard against unauthorized access.
6. **Audit and Internal Penetration Testing:**
   Students must commit their code to GitHub and perform an internal audit, including penetration testing. They must submit a video (any platform) demonstrating the security

measures implemented, along with a Proof of Concept that explains potential vulnerabilities within the application stack.

Additional Security Features:
Students are required to research and implement additional security features beyond those listed above according to their findings. This approach encourages independent exploration and the application of advanced security measures based on current best practices.

Report Requirements

1. **Cover Page**
2. **Acknowledgment**
3. **Table of Contents**
4. **Table of Figures** – Includes a list of all figures, diagrams, and images used in the report.
5. **Table of Abbreviations** – A comprehensive list of abbreviations and their meanings used throughout the report.
6. **Abstract** – A brief summary of the entire project, including its purpose, methods, results, and conclusions.
7. **Introduction** – Provides an overview of the project and its objectives, explaining the significance of developing a secure web application. It introduces the chosen concept and addresses the specific user need.
8. **Software Details** – Covers the frameworks and programming languages used (e.g., Node.js, Express, JavaScript, HTML, CSS), details about the database management system (e.g., MongoDB, MySQL), and the steps taken for secure deployment of the web application.
9. **Design and Implementation** – This section starts with a system design overview, describing the web application's architecture and key component interactions. It discusses security considerations in the design process, explaining how the principles of "security by design" are integrated. A security analysis of key elements, including potential security issues, is provided, along with code examples. Recommendations for addressing security and data protection issues are offered, explaining how these mitigate risks. The security implementation is demonstrated through specific code examples, focusing on session management, password management, and user access control. This section also includes a detailed account of GitHub commits, with at least 30 commits showcasing security-related code, and an explanation of how each addresses particular security concerns.
10. **Proof of Concept** – This section includes a video demonstration of the security measures implemented, along with a detailed explanation of potential vulnerabilities within the application stack. It justifies why these vulnerabilities could exist and how they were addressed.
11. **Conclusion** – Summarizes the project's outcomes, reflects on the importance of security in web application development, and suggests future improvements and additional security features.
12. **References** – A list of all sources cited in the report, with a minimum of 15 references from diverse sources, including websites, journals, books, and research papers.

Note: *Your **proof of concept** (PoC) video must display your face and have clear audio and the screenshots in report must be clear, readable, and deterministic.*

Submission Instructions

Requirement	Details
File Naming	NAME_studentID
File Format	.docx/.pdf format
Submission Method	Campus 4.0 platform (submission link provided 2 weeks before deadline)

Marking and Feedback

How will my assignment be marked?

Your assignment will be marked by the Module Team using standardized criteria.

How will I receive grades and feedback?

Provisional marks will be released once internally moderated. Feedback will be provided alongside grades release within 2 weeks (10 working days).

What will I be marked against?

Details of the marking criteria for this task can be found in the Assessment Marking Criteria section at the end of this brief.

Grade Requirements

You must achieve 40% or above to pass this assessment. Ensure you understand the marking criteria for successful completion.

Assignment Support and Academic Integrity

Getting Help

If you have any questions about this assignment, please meet your respective module leader or teacher for more information.

Language Standards

You are expected to use effective, accurate, and appropriate language within this assessment task.

Academic Integrity

The work you submit must be your own. All sources of information need to be acknowledged and attributed; therefore, you must provide references for all sources of information and acknowledge any tools used in the production of your work, **excluding Artificial Intelligence (AI)**.

We use detection software and make routine checks for evidence of academic misconduct. Definitions of academic misconduct, including plagiarism, self-plagiarism, and collusion can be found in Student handbook in Campus 4.0.

All cases of suspected academic misconduct are referred to for investigation, the outcomes of which can have profound consequences to your studies.

Support for Students with Disabilities

If you have a disability, long-term health condition, specific learning difference, mental health diagnosis or symptoms, contact the Student Support Office for assistance.

Unable to Submit on Time?

If events prevent you from submitting on time, guidance on extenuating circumstances is available in the Student Handbook or from the Student Support Office.

Administration of Assessment

Module Leader Name:	Arya Pokharel

Module Leader Email:	stw00105@softwarica.edu.np
Assignment Category:	Written
Attempt Type:	Standard
Component Code:	CW2

Marking Criteria

- **Report Structure and Document Formatting: 25**
- **Security Controls and Implementation: 20**
- **Audit and Security Testing: 20**
- **Advanced Security Measures: 20**
- **Proof of Concept: 15**

Assessment Criteria

0-39	40-49	50-59	60-69	70-100
The application does not meet the basic objectives of the task, showing major omissions in required features such as secure user registration, brute-force prevention, or privacy controls. The work is mostly descriptive with no critical analysis, and the argument lacks logical structure or coherence. Core security features like MFA, RBAC, encryption, or session management are either completely missing or improperly implemented.	The application addresses some parts of the task, but there are significant gaps in essential security measures like access control or encryption. The structure of the report is weak, relying heavily on descriptions without meaningful analysis. The argument lacks clarity and fails to clearly link security features with their purpose. Implemented security measures may be incomplete or non-functional. Evidence of testing or internal review is minimal or superficial.	The application meets most task objectives, with core functionality included, but some features such as brute-force prevention or proper session handling is underdeveloped. The report presents a generally clear argument but may be overly descriptive and not consistently analytical. Security features like MFA, RBAC, and encryption are implemented at a basic level, though testing or auditing may be incomplete or lack depth.	The application effectively meets the task's objectives, incorporating key features such as secure registration, privacy controls, and secure transaction processing. The report presents a well-structured argument with evaluative discussion, although some aspects of the implementation may lack depth. Core security measures including MFA, encryption, RBAC, and session management are present and mostly well-implemented. There is good evidence of testing, auditing, and attention to	The web application is highly aligned with the task objectives and demonstrates attention to complex security mechanisms. All major/minor security features are implemented and extended with user-centric design, robust brute-force prevention, and secure workflows. The report is logically structured and demonstrates in-depth critical analysis with original insights and well-articulated connections between security features. The implementation includes strong measures like MFA, RBAC, encryption,

There is no evidence of testing, auditing, or understanding of system vulnerabilities. Minimal or no research is present, and referencing is either missing or incorrect. The submission reflects a poor grasp of fundamental security concepts.	Research is sparse, and few (if any) additional security features are implemented. Referencing is inconsistent, and the report demonstrates a limited understanding of security principles.	Research is evident but limited, and the inclusion of additional features is minimal. Referencing is present but may contain errors. The overall understanding of security concepts is basic but functional.	vulnerabilities, even if not exhaustive. Research is relevant and leads to some additional security feature implementation. Referencing is mostly correct with minor issues..	secure sessions, and password policies, supported by extensive testing, internal audits, and penetration testing. Research goes beyond the baseline and leads to well-justified additional security implementations. All references are accurate, and the work reflects a sophisticated understanding of modern application security practices.