NATIONAL UNIVERSITY
COLLEGE OF COMPUTING AND INFORMATION TECHNOLOGY

Information Assurance and Security
(Learning Activities Scoresheet)

Student Name: 1. **Asuncion, Beatriz Uy** 2. **Esquejo, Sherdon Rappah** 3. **Galvez, Aldrin** 4. **Pagulayan, Kamira Allison F.** 5. **Rico, Ronaldo Jr. D.**	**Date: February 11, 2025**

Learning Activity Requirement(s)

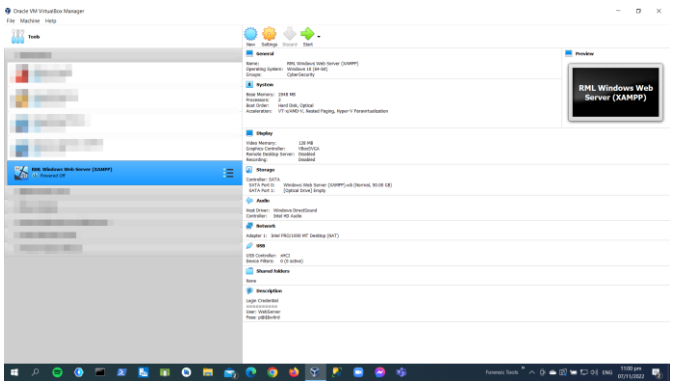
Instructions/Directions:

*Be able to implement the necessary "**Server Security Hardening**" techniques on your web application server.*

Procedures:

1. Based on the web vulnerability assessment report of your web application, be able to implement all the necessary security hardening techniques on your server, database, and network (optional).
2. Provide a checklist of the vulnerability assessment and the recommended security hardening technique.
3. Provide a screenshot of the necessary configuration of a completed task from the checklist.
4. Upload the configuration file (.conf or .ini) of your server.

*Show proof of your activity completion by providing necessary screenshot of your entire workstation (refer to the sample image below). Use the **Output Presentation and Discussion** section of your activity template for the needed screenshot(s).*



Submission Note:

1. *Document filename LASTNAME_Activity5.pdf*
2. *Configuration file (httpd.conf) must be archived as Zip file with filename LASTNAME_ZipActivity5.zip*

SECURITY HARDENING

Based on the web vulnerability assessment report of your web application, be able to implement all the necessary security hardening techniques on your server, database, and network (optional).

System Vulnerability Checklist

Provide a checklist of system-related vulnerability based on OWASP web vulnerability results and the recommended security hardening technique.

Vulnerability	Severity	Recommended Fix
Vulnerable JS Library	High	Phase: Architecture and Design Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid. For example, use anti-CSRF packages such as the OWASP CSRFGuard.
Absence of Anti-CSRF Tokens	Medium	Phase: Architecture and Design Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid. For example, use anti-CSRF packages such as the OWASP CSRFGuard. Phase: Implementation Ensure that your application is free of cross-site scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script. Phase: Architecture and Design Generate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the

		form. Be sure that the nonce is not predictable (CWE-330). Note that this can be bypassed using XSS. Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation. Note that this can be bypassed using XSS. Use the ESAPI Session Management control. This control includes a component for CSRF. Do not use the GET method for any request that triggers a state change. Phase: Implementation Check the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may have disabled sending the Referer for privacy reasons.
Application Error Disclosure	Medium	Review the source code of this page. Implement custom error pages. Consider implementing a mechanism to provide a unique error reference/identifier to the client (browser) while logging the details on the server side and not exposing them to the user.
CSP: Wildcard Directive	Medium	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.

CSP: script-src unsafe-eval	Medium	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.
CSP: script-src unsafe-inline	Medium	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.
CSP: style-src unsafe-inline	Medium	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.
Content Security Policy (CSP) Header Not Set	Medium	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.
Hidden File Found	Medium	Consider whether or not the component is actually required in production, if it isn't then disable it. If it is then ensure access to it requires appropriate authentication and authorization, or limit exposure to internal systems or specific source IPs, etc.
Missing Anti-clickjacking Header	Medium	Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app. If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.
Vulnerable JS Library	Medium	Please upgrade to the latest version of jquery-validation.

Big Redirect Detected (Potential Sensitive Information Leak)	Low	Ensure that no sensitive information is leaked via redirect responses. Redirect responses should have almost no content.
Cookie No HttpOnly Flag	Low	Ensure that the HttpOnly flag is set for all cookies.
Cookie without SameSite Attribute	Low	Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.
Cross-Domain JavaScript Source File Inclusion	Low	Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.
Information Disclosure - Debug Error Messages	Low	Disable debugging messages before pushing to production.
Private IP Disclosure	Low	Remove the private IP address from the HTTP response body. For comments, use JSP/ASP/PHP comment instead of HTML/JavaScript comment which can be seen by client browsers.
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low	Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.
Server Leaks Version Information via "Server" HTTP Response Header Field	Low	Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.
Strict-Transport-Security Header Not Set	Low	Ensure that your web server, application server, load balancer, etc. is configured to enforce Strict-Transport-Security.
X-Content-Type-Options Header Missing	Low	Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages. If possible, ensure that the end user uses a standards-compliant and modern web

		browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.

Security Hardening Implementation

Provide a screenshot of the necessary configuration of a completed task from the checklist.

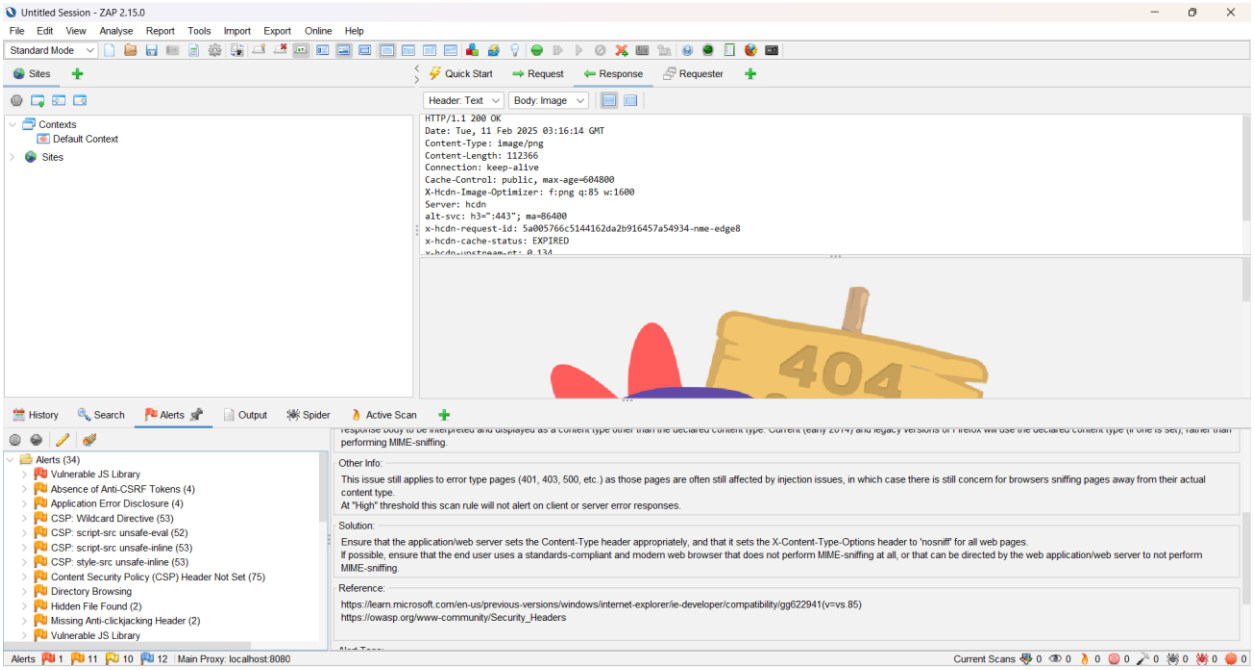
1. **Modified Apache and MySQL configuration files.**

```
    #
    # Options Indexes FollowSymLinks Includes ExecCGI
```

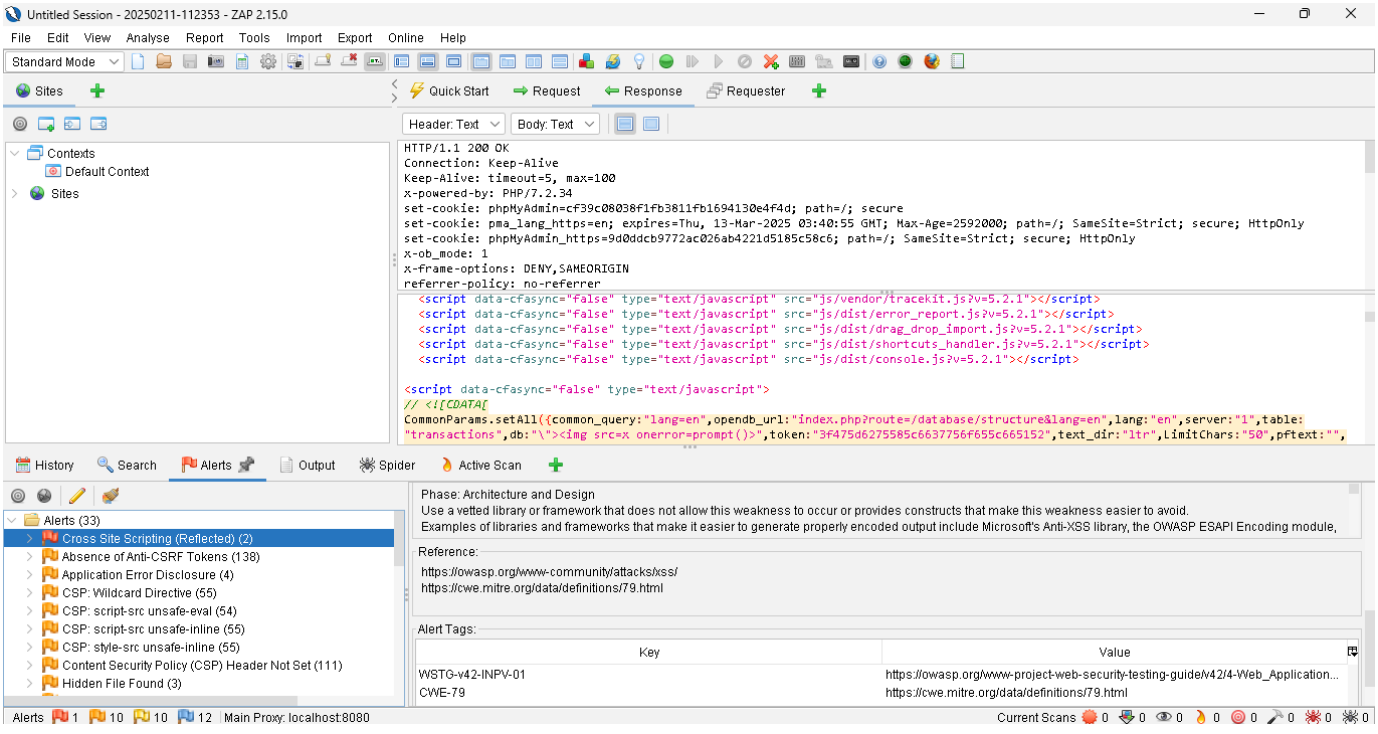
```
#   Server Certificate:
#   Point SSLCertificateFile "conf/ssl.crt/server.crt"
#   the certificate is encrypted, then you will be prompted for a
#   pass phrase.  Note that a kill -HUP will prompt again.  Keep
#   in mind that if you have both an RSA and a DSA certificate you
#   can configure both in parallel (to also allow the use of DSA
#   ciphers, etc.)
#   Some ECC cipher suites (http://www.ietf.org/rfc/rfc4492.txt)
#   require an ECC certificate which can also be configured in
#   parallel.
SSLCertificateFile "conf/ssl.crt/server.crt"
#SSLCertificateFile "conf/ssl.crt/server.crt"
#SSLCertificateFile "conf/ssl.crt/server.crt"

#   Server Private Key:
#   If the key is not combined with the certificate, use this
#   directive to point at the key file.  Keep in mind that if
#   you've both a RSA and a DSA private key you can configure
#   both in parallel (to also allow the use of DSA ciphers, etc.)
#   ECC keys, when in use, can also be configured in parallel
SSLCertificateKeyFile "conf/ssl.key/server.key"
#SSLCertificateKeyFile "conf/ssl.key/server.key"
#SSLCertificateKeyFile "conf/ssl.key/server.key"

#   Server Certificate Chain:
#   Point SSLCertificateChainFile at a file containing the
#   concatenation of PEM encoded CA certificates which form the
#   certificate chain for the server certificate. Alternatively
#   the referenced file can be the same as SSLCertificateFile "conf/ssl.crt/server.crt"
```

2. OWASP vulnerability scan (before and after).

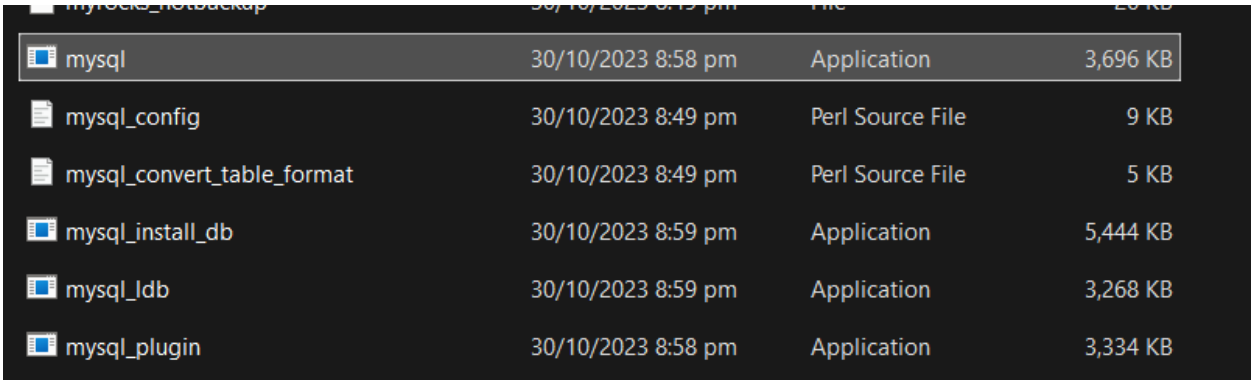
Before



After



3. **Windows Firewall rule configuration.**

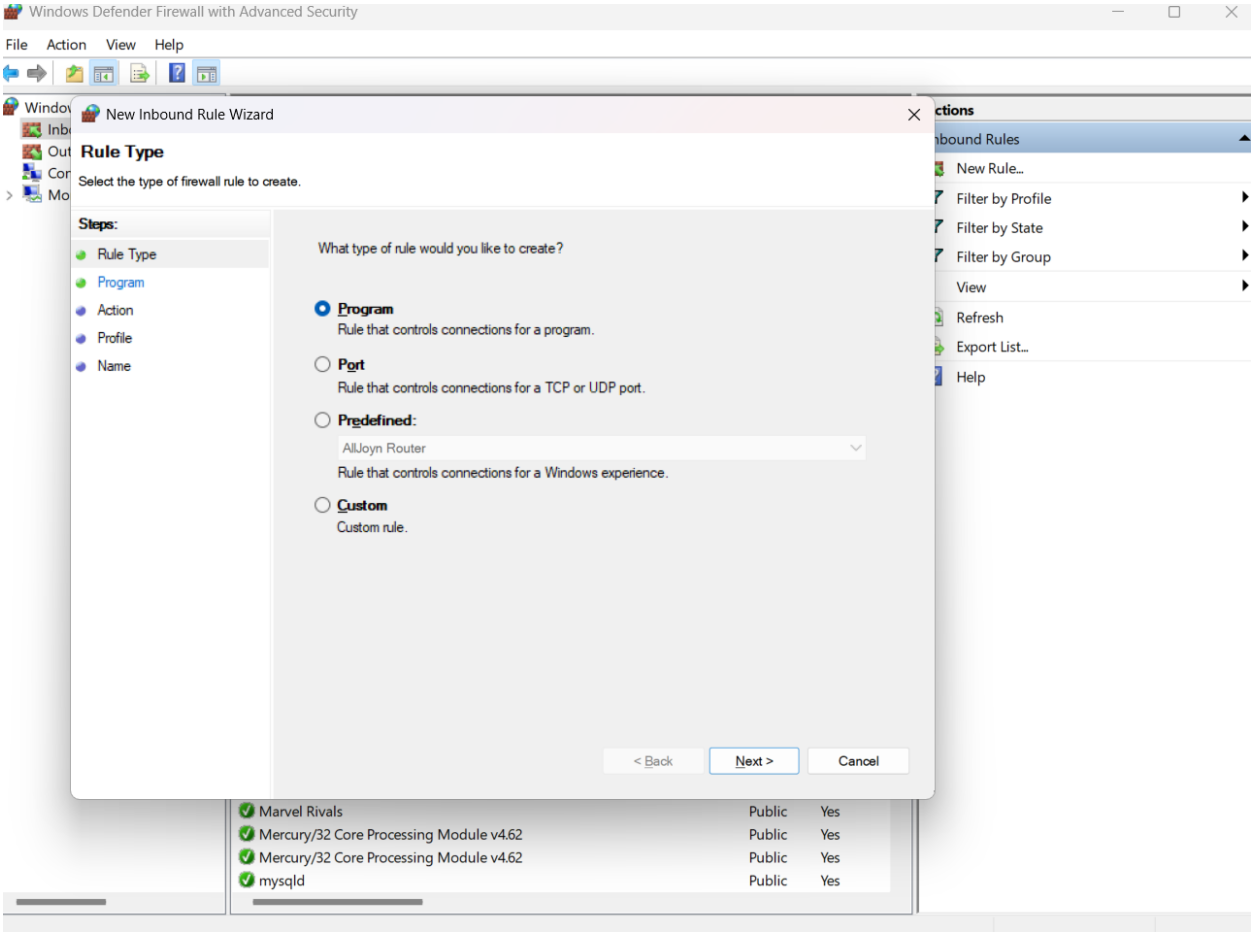


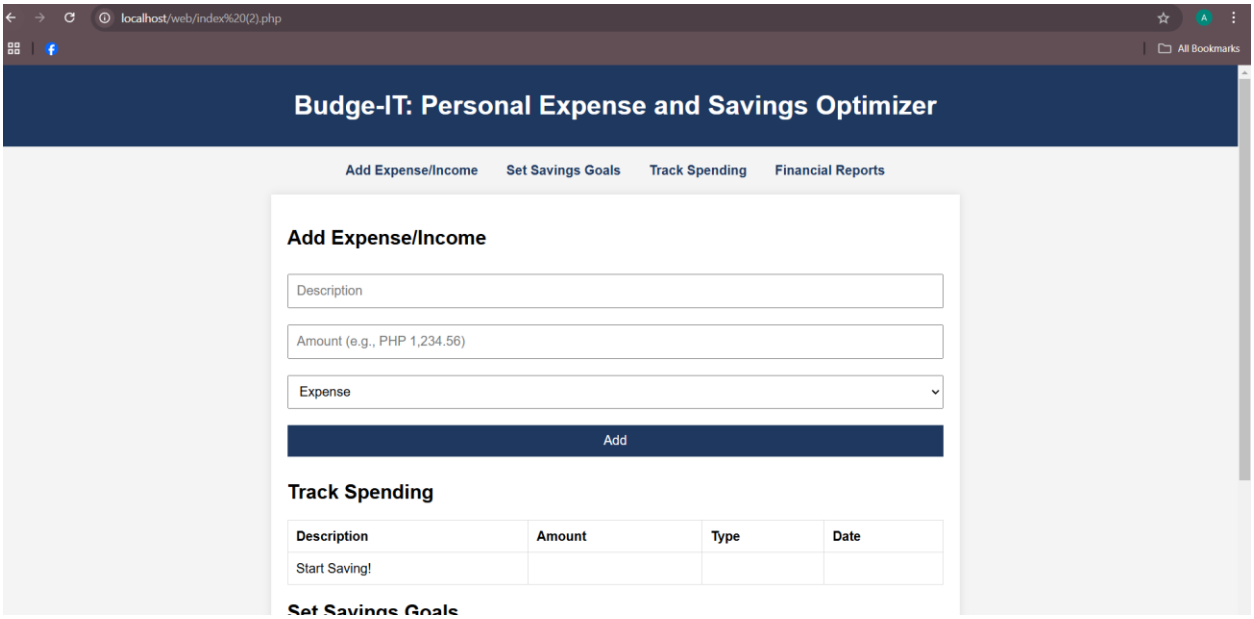
mysql	30/10/2023 8:58 pm	Application	3,696 KB
mysql_config	30/10/2023 8:49 pm	Perl Source File	9 KB
mysql_convert_table_format	30/10/2023 8:49 pm	Perl Source File	5 KB
mysql_install_db	30/10/2023 8:59 pm	Application	5,444 KB
mysql_ldb	30/10/2023 8:59 pm	Application	3,268 KB
mysql_plugin	30/10/2023 8:58 pm	Application	3,334 KB

```
# The MySQL server
default-character-set=utf8mb4
[mysqld]
bind-address = 127.0.0.1
skip-networking
port=3306
socket="C:/xampp/mysql/mysql.sock"
basedir="C:/xampp/mysql"
tmpdir="C:/xampp/tmp"
datadir="C:/xampp/mysql/data"
pid_file="mysql.pid"
# enable-named-pipe
key_buffer=16M
max_allowed_packet=1M
```

```
; Whether to allow the treatment of URLs (like http:// or ftp://) as files.
; https://php.net/allow-url-fopen
allow_url_fopen = Off display_errors = Off log_errors = On
```

4. HTTPS-enabled web application.





Criteria	Scoring Rubric				Score
	Not Meeting Expectations	Difficulty Meeting Expectations	Progressing towards Meeting Expectations	Meeting Expectations	
	(1 point)	(2 point)	(3 points)	(4 points)	
Principle and Techniques	No server hardening techniques mentioned	1 server hardening technique mentioned and no additional non-class technique/area of study included	included mention of at least 2 server hardening techniques in addition to 1 technique/area of study we did NOT cover in class	included mention of at least 3 server hardening techniques AND at least 2 techniques/areas of study we did NOT cover in class	
Evidence	No evidence or artifacts included	1 piece of evidence and analysis are revealed.	2 pieces of evidence and evidence analysis were revealed.	3 or more pieces of evidence and evidence analysis are revealed	
Functionality	The code is not functional, meeting no significant design specification, or was not attempted	The program is producing incorrect result, or the program contains syntax errors	The program produces correct results but does not display them correctly	The program works and produces the correct results and displays them correctly. Other specifications are also meet.	
Report Documentation	No Report Documentation provided	Provided less report documentation with limited supporting documents	Report documentation provided is limited with minimal supporting documents	Report documentation provided is complete with supporting documents	
Submission	No Submission with no supporting documents	Late Submission (more than a day) with limited supporting documents	Late Submission (within the day) with minimal to complete supporting documents	On-Time Submission with complete documents	
				Total Score	