Unit 5: Hosting, Deployment, and Security of Educational Web Applications

Unit 5 — Moodle Introduction

Unit 5 brings the module to its culmination. You will deploy your eLearning web application to a live hosting environment, configure a production-ready server using cPanel, and conduct a final security audit of your application. The end product of this module—a functional, hosted, accessible, and secure eLearning web application—demonstrates your readiness to contribute to real-world educational technology development.

REFLECT

Critical Reflection (Schön, 1983): Unit 5 includes a mandatory 300-word Deployment Log in which students document every step taken, every error encountered, and how each error was resolved. This structured reflection activity develops the metacognitive awareness that distinguishes expert practitioners from novices—the ability to analyse one's own problem-solving process and extract transferable lessons.

Unit Aim

The aim of Unit 5 is to enable students to deploy web-based eLearning applications to production hosting environments using both local and online hosting with cPanel, configure production server settings, and apply web application security principles to protect deployed educational systems.

Intended Learning Outcomes

- Deploy a web-based eLearning application to a local development server (XAMPP/Laragon) and a live production server using cPanel (Bloom's Level 3 — Apply).
- Configure a cPanel hosting environment, including domain/subdomain setup, MySQL database creation, file management via File Manager and FTP, and SSL certificate installation (Bloom's Level 3 — Apply).
- Conduct a web application security audit identifying and resolving vulnerabilities including SQL injection, XSS, CSRF, insecure file uploads, and missing HTTPS (Bloom's Level 4 — Analyse).
- Estimate the total cost of hosting and running a web-based eLearning application over a twelve-month period, comparing at least three hosting providers (Bloom's Level 4 — Analyse).

5.1 Local Development with XAMPP and Laragon

Tool	Description and Recommendation
XAMPP	Cross-platform (Windows, macOS, Linux) local server stack. Includes Apache, MariaDB/MySQL, PHP, and phpMyAdmin. Free and open-source. Available at https://www.apachefriends.org.
Laragon	Windows-only local development environment. Faster and lighter than XAMPP. Supports multiple PHP versions simultaneously.

	Automatic virtual host creation (e.g., myapp.test). Recommended for Windows development.
Docker (Advanced)	Containerised development environment. Ensures the development environment is identical to the production server. Recommended for team projects to eliminate 'it works on my machine' problems.

5.2 Online Hosting with cPanel

cPanel is the most widely used web hosting control panel. It provides a graphical interface for managing all aspects of a shared or VPS hosting account, including file management, database administration, domain configuration, email accounts, and SSL certificates.

5.2.1 Deploying an eLearning Application via cPanel

7. Purchase a hosting plan from a provider that supports PHP 8.x and MySQL 8.x (e.g., Hostinger, SiteGround, Namecheap, or a Rwandan provider).
8. Access your cPanel dashboard via https://yourdomain.com/cpanel.
9. Create a MySQL database and database user via the MySQL Databases section. Grant the user ALL PRIVILEGES on the new database.
10. Upload your application files using File Manager (for small projects) or an FTP client such as FileZilla (for larger projects) to the public_html directory.
11. Import your database SQL file using phpMyAdmin, available from cPanel.
12. Edit your application's database configuration file to use the cPanel database name, username, and password.
13. Install an SSL certificate using the SSL/TLS section of cPanel. Most modern hosts provide Let's Encrypt certificates for free, enabling HTTPS.
14. Test the live application thoroughly, paying particular attention to file upload paths, email configuration, and HTTPS redirection.

MULTIMEDIA *Screen-Capture Video: A 12-minute narrated OBS Studio recording demonstrating the complete cPanel deployment workflow (steps 1–8 above) is available in the Unit 5 Moodle section as an MP4 file. The video is captioned and has an accompanying text-based step-by-step guide (the same content in two formats—UDL Principle 1: Multiple Means of Representation) to support learners in bandwidth-constrained environments.*

5.3 Web Application Security

Vulnerability	Definition and Mitigation
SQL Injection (A03:2021)	ALREADY COVERED — Use PDO prepared statements. Never concatenate user input into SQL queries.
Cross-Site Scripting (XSS) — A03:2021	Never output user-supplied data to HTML without escaping. In PHP: echo htmlspecialchars($userInput, ENT_QUOTES, 'UTF-8'). In Node.js: use the DOMPurify or sanitize-html library.

Broken Access Control (A01:2021)	Verify user roles on every server-side request. Never rely on client-side hiding of sensitive functions. Check that the authenticated user owns the resource they are trying to access.
Cryptographic Failures (A02:2021)	Use HTTPS everywhere. Never store passwords in plaintext—use bcrypt. Encrypt sensitive student data at rest if required by law or policy.
CSRF (Cross-Site Request Forgery)	Generate a unique, session-tied CSRF token for every form. Validate the token on every POST/PUT/DELETE request.
Security Misconfiguration (A05:2021)	Remove default accounts and passwords. Disable directory listing. Hide error messages from users (log them server-side). Remove development files from production servers.

5.4 Hosting Cost Analysis

Professional educational web developers must be able to estimate and justify the financial cost of hosting a web application. This skill is essential for project proposals, grant applications, and self-employment. Costs include: domain registration, hosting plan, SSL certificate (often free with Let's Encrypt), email hosting, backup storage, and CDN services.

When analysing hosting providers, compare the following criteria: PHP and MySQL version support; monthly traffic allowance (bandwidth); disk storage capacity; number of databases and email accounts; automated backups; 24/7 support quality; geographic server location (closer to Rwanda = lower latency); and whether the provider offers a Rwanda or East Africa point of presence.

Unit 5 Assessments — ABC Learning Design Framework

Activity	ABC Type	Description	Bloom's Level	Weight
Local Deployment Walkthrough — E-tivity 1	Practice / Acquisition	Record a 5-minute OBS Studio screen-capture demonstrating the complete local deployment process: XAMPP/Laragon setup, database import, configuration file editing, and application running at localhost. Upload to Moodle as an MP4.	Apply (L3)	5% (ICA)
Live Hosting Deployment — E-tivity 2	Production	Deploy your complete eLearning application to a live cPanel hosting account. Submit: the live URL, screenshots of cPanel, evidence of HTTPS, and a 300-word deployment log documenting each step taken and any problems resolved.	Apply (L3)	15% (ICA)

Security Audit Report — E-tivity 3	Investigation / Analysis	Conduct a security audit of your deployed application. Use OWASP ZAP for automated scanning. Write a 600-word report: for each vulnerability found, state the risk level, explain the technical cause, and provide the corrected code or configuration. Minimum: address SQL injection protection, XSS, CSRF, and HTTPS configuration.	Analyse (L4)	10% (ICA)
Hosting Cost Analysis	Investigation / Discussion	Compare three hosting providers appropriate for the UR eLearning application. Present a cost breakdown table covering 12 months and write a 250-word recommendation justifying the best choice for a Rwandan university context.	Analyse (L4)	5% (ICA)
Final Project Presentation — E-assessment 2	Production / Collaboration / Discussion	Present your complete, deployed eLearning web application to the class. Demonstrate: the live application running on the internet; CRUD operations; file upload functionality; and HTTPS. Duration: 15 minutes + 10 minutes Q&A. Peer and lecturer feedback collected via Moodle Workshop rubric.	Create / Evaluate (L6 / L5)	Part of 40% Practical total

End-Unit Assessment 1: Moodle Quiz — Hosting, Deployment & Security

Ready-to-Use Moodle Quiz — Copy & Paste Questions

10 questions. Set: Attempts = 2, Grading = Highest, Shuffle = Yes, Feedback = Immediate.

Question & Answer	Notes
Q1. Which cPanel tool should a developer use to upload large application files (>10 MB) to the public_html directory? ✓ A) FileZilla FTP client B) cPanel File Manager C) phpMyAdmin D) cPanel Terminal Feedback: FileZilla (or any FTP client) is more reliable than cPanel's browser-based File Manager for uploading large files, as browser uploads are susceptible to timeout failures on slow connections.	

Q2. After deploying a PHP eLearning application to a shared cPanel host, the application's database configuration file still contains the local credentials (localhost, root, no password). What is the consequence? ✓ A) The application will fail to connect to the production database, causing a fatal error B) The application will automatically detect the new environment C) Moodle will override the configuration file with cPanel credentials D) The application will use a read-only connection to the production database Feedback: Production cPanel databases use a different host, database name, username, and password from local environments. The configuration file must be updated before uploading to the live server.	
Q3. What does the Apache directive Options -Indexes in an .htaccess file do? ✓ A) Prevents directory listing, hiding the file structure of the application from unauthorised visitors B) Enables Apache to serve PHP files as HTML C) Restricts the application to HTTPS connections only D) Removes the server version header from HTTP responses Feedback: Without Options -Indexes, Apache displays a list of all files in a directory if there is no index.html or index.php file. This can expose sensitive files (configuration, uploaded student documents) to public access.	
Q4. A security audit identifies that the error details of a database connection failure are printed directly to the browser. Which OWASP Top Ten category does this represent and how should it be remediated? ✓ A) A05:2021 Security Misconfiguration — log errors server-side and show only a generic message to users B) A03:2021 Injection — use prepared statements C) A02:2021 Cryptographic Failures — encrypt the error message D) A01:2021 Broken Access Control — require admin login to view errors Feedback: Exposing internal error details (database structure, file paths, credentials) is a security misconfiguration. Use PHP's error_reporting(0) in production and log errors with error_log() to a server-side log file.	
Q5. Which command in an .htaccess file redirects all HTTP traffic to HTTPS? RewriteEngine On RewriteCond %{HTTPS} off ✓ A) RewriteRule ^ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301] B) RewriteRule ^ http://%{HTTP_HOST}%{REQUEST_URI} [L,R=302] C) RewriteCond %{HTTPS} on D) Header always set Strict-Transport-Security Feedback: The RewriteCond checks whether the request is NOT already HTTPS. If it is not, the RewriteRule permanently redirects (301) to the HTTPS version of the same URL. R=302 is a temporary redirect and should not be used for HTTPS enforcement.	
Q6. A developer uses OWASP ZAP to scan their deployed eLearning application and finds a 'Reflected XSS' alert. What does this mean and how is it fixed in PHP? ✓ A) User-supplied input is reflected in the HTML response without escaping; fix with echo htmlspecialchars($userInput, ENT_QUOTES, 'UTF-8') B) SQL is injected into a reflected query; fix with PDO prepared statements C) The CSRF token is reflected in the URL; remove it from the query string D) Session data is reflected in cookies; set HttpOnly flag Feedback: Reflected XSS occurs when user input (e.g., from a GET parameter) is included in the HTML response without sanitisation, allowing an attacker to inject malicious JavaScript via a crafted URL. htmlspecialchars() converts characters like < > " ' to HTML entities, neutralising the injection.	
Q7. Why is Docker recommended over XAMPP for team-based eLearning application development? ✓ A) Docker containers ensure all developers run identical server environments, eliminating 'it works on my machine' inconsistencies B) Docker is faster than XAMPP for processing PHP requests C) Docker is required by cPanel for deployment D) XAMPP does not support MySQL databases Feedback: Environment inconsistency between developers is a major source of	

bugs in collaborative projects. Docker containers package the exact PHP version, MySQL version, and configuration, ensuring every developer's environment is identical to the production server.	
Q8. Let's Encrypt is recommended for SSL certificates on eLearning hosting. What is the main advantage? ✓ A) Let's Encrypt certificates are free, auto-renewing, and recognised by all major browsers B) Let's Encrypt provides Extended Validation (EV) certificates at no cost C) Let's Encrypt is required by Rwanda's data protection law D) Let's Encrypt certificates never expire Feedback: Let's Encrypt provides Domain Validation (DV) certificates at no charge with 90-day validity and automatic renewal via the Certbot client. Most modern cPanel hosts integrate Let's Encrypt directly, making HTTPS configuration a one-click operation.	
Q9. A UR developer needs to choose between hosting in a data centre in Kigali versus a European data centre. Which consideration is most relevant for compliance with Rwandan data protection law? ✓ A) Rwanda's Law No. 058/2021 may impose data localisation requirements for personal data of Rwandan citizens; a Kigali data centre ensures compliance B) European data centres provide better uptime guarantees C) Kigali data centres are cheaper than European ones D) European data centres comply with GDPR, which supersedes Rwandan law Feedback: Cross-border data transfers under Law No. 058/2021 require appropriate safeguards. Hosting personal data of Rwandan students in Rwanda minimises compliance risk and reduces latency for students in Rwanda.	
Q10. [Short Answer] Describe the purpose of the X-Content-Type-Options: nosniff HTTP security header and explain why it is important for an eLearning application that allows file uploads. Model Answer: The X-Content-Type-Options: nosniff header instructs browsers not to 'sniff' (guess) the MIME type of a response and instead respect the Content-Type header declared by the server. Without this header, some browsers may attempt to interpret a file as a different type than declared—for example, treating a malicious HTML file uploaded as an image as executable HTML. In an eLearning application that serves student-uploaded files, an attacker could upload a file containing HTML/JavaScript and, if the browser sniffs its type as text/html, execute the script in the context of the application, leading to stored XSS. Setting nosniff prevents this browser-side type confusion.	

End-Unit Assessment 2: Final Project Presentation Rubric (Moodle Workshop)

Criterion	Distinction (75–100%)	Merit (60–74%)	Pass (50–59%)	Fail (<50%)
Functional Completeness	Application is fully operational on a live URL with all CRUD operations, file upload, and HTTPS functional and demonstrable.	Application live and mostly functional; one minor feature non-operational.	Application partially functional; 2–3 features non-operational.	Application not accessible on live URL or fundamental features absent.
Security Implementation	All OWASP Top 5 vulnerabilities addressed; HTTPS enforced; CSRF tokens; prepared	Most security controls present; one control missing or incorrectly implemented.	Some security controls present; critical vulnerability (e.g., no prepared	Security controls largely absent; application has critical

	statements; password hashing demonstrated.		statements) unaddressed.	unaddressed vulnerabilities.
Presentation Quality	Presentation is structured, confident, and within time. Live demo proceeds smoothly with clear narration.	Mostly structured and clear; minor pacing or technical issues.	Partially clear; significant technical difficulties or time management issues.	Presentation is unclear, unstructured, or significantly over/under time.
Q&A Performance	Accurate, confident responses to all questions; demonstrates understanding beyond the presented application.	Answers most questions correctly; struggles with one or two deeper questions.	Answers some questions; significant difficulty with technical follow-up.	Unable to address most questions about own application.

Module References

All references below follow the APA 7th Edition citation format.

- Anderson, L. W., & Krathwohl, D. R. (Eds.). (2001). A taxonomy for learning, teaching, and assessing: A revision of Bloom's educational objectives. Longman.
- Ausubel, D. P. (1968). Educational psychology: A cognitive view. Holt, Rinehart and Winston.
- Biggs, J., & Tang, C. (2011). Teaching for quality learning at university (4th ed.). Open University Press.
- Black, P., & Wiliam, D. (1998). Assessment and classroom learning. Assessment in Education: Principles, Policy & Practice, 5(1), 7–74.
- Branch, R. M. (2009). Instructional design: The ADDIE approach. Springer.
- CAST. (2018). Universal design for learning guidelines version 2.2. http://udlguidelines.cast.org
- Cook, D. A., & Dupras, D. M. (2004). A practical guide to developing effective web-based learning. Journal of General Internal Medicine, 19(6), 698–707.
- Coronel, C., & Morris, S. (2019). Database design process. Cengage Learning.
- Dean, J. (2019). Web programming with HTML5, CSS, and JavaScript. Jones & Bartlett Learning.
- Dick, W., Carey, L., & Carey, J. O. (2015). The systematic design of instruction (8th ed.). Pearson.
- Elmasri, R., & Navathe, S. B. (2016). Fundamentals of database systems (7th ed.). Pearson.
- Gagné, R. M. (1985). The conditions of learning and theory of instruction (4th ed.). Holt, Rinehart and Winston.
- Kirkpatrick, D. L. (1994). Evaluating training programs: The four levels. Berrett-Koehler.
- Lave, J., & Wenger, E. (1991). Situated learning: Legitimate peripheral participation. Cambridge University Press.
- Mayer, R. E. (2009). Multimedia learning (2nd ed.). Cambridge University Press.
- Merrill, M. D. (2002). First principles of instruction. Educational Technology Research and Development, 50(3), 43–59.
- Nash, S. S., & Rice, W. (2018). Moodle 3 e-learning course development (4th ed.). Packt Publishing.
- OWASP. (2021). OWASP Top Ten 2021. https://owasp.org/www-project-top-ten/
- Paivio, A. (1970). On the functional significance of imagery. Psychological Bulletin, 73(6), 385–392.
- Republic of Rwanda. (2021). Law No. 058/2021 relating to the protection of personal data and privacy. Official Gazette.
- Roediger, H. L., & Butler, A. C. (2011). The critical role of retrieval practice in long-term retention. Trends in Cognitive Sciences, 15(1), 20–27.
- Salmon, G. (2002). E-tivities: The key to active online learning. Kogan Page.
- Schön, D. A. (1983). The reflective practitioner: How professionals think in action. Basic Books.
- Smaldino, S. E., Lowther, D. L., Mims, C., & Russell, J. D. (2019). Instructional technology and media for learning (12th ed.). Pearson.