Exam: WordPress Certified Developer

Exam Link: https://knowledge-pillars.com/wp-certified-developer/

Course: Web Design

TEKS Code & Standard	TEKS Description	Corresponding WordPress Certified Developer Objective(s)	Alignment Description
§127.685(c)(1)(A)	Identify and describe the function of a web server and its role in web hosting.	Unit 1 – PHP for WordPress: Creating a Theme, The Loop	Developer setup includes configuring hosting, themes, and understanding server behavior.
§127.685(c)(2)(A)	Apply design and web programming terminology appropriately.	Unit 4 – Template Hierarchy: Working with style.css, functions.php	Uses technical terminology such as hooks, loops, and functions consistent with TEKS.
§127.685(c)(2)(B)	Demonstrate knowledge of HTML and CSS to create web pages.	Unit 2 – Pages: Menus, Page Templates	Students use page templates to implement structured HTML/CSS across content types.
§127.685(c)(2)(C)	Use HTML and CSS to create semantic structure and layout.	Unit 4 – Template Hierarchy: Custom Templates, Search Templates	Custom templates require semantic markup, echoing TEKS layout expectations.
§127.685(c)(3)(A)	Apply web design best practices including navigation, hierarchy, and readability.	Unit 4 – Template Hierarchy: Sidebars, Archives, Markup	Advanced styling and markup promote best practices in readability and structure.
§127.685(c)(4)(A)	Develop and implement a website using web authoring tools.	Unit 6 – REST API: Load Content, Create JSON, Add Fields	API endpoints allow dynamic page generation and simulate complex maintenance updates.
§127.685(c)(5)(B)	Demonstrate knowledge of website maintenance and updating content.	Unit 7 – Hooks: Action and Filter Hooks in WordPress	Hooks let developers customize site behavior, enhancing long-term site updates.
§127.685(c)(6)(A)	Understand the importance of website security and implement basic security measures.	Unit 8 – Plugin Dev: Plugin Paths, Settings, Options API	Security is handled at code level, directly linking to TEKS expectations on security.
§127.685(c)(8)(A)	Analyze ethical and legal issues related to web publishing.	Unit 8 – Plugin Dev: Security Best Practices in Plugins	Students engage with plugin development that respects licensing and legal practices.
§127.685(c)(8)(B)	Demonstrate ethical use of digital content and respect for intellectual property.	Unit 8 – Plugin Dev: Licensing, Attribution, Copyright Handling	Plugin development reinforces proper digital content attribution and compliance.

Course: Advanced UX Design

TEKS Code & Standard	TEKS Description	Corresponding WordPress Certified Developer Objective(s)	Alignment Description
§127.792(c)(1)(C)	Achieve at least one UX professional certification and build resumes and portfolios.	Unit 2 – Pages: Templates, Parent/Child Structures, Navigation Menus	Students design and build full site layouts using templated systems, matching UX architecture.
§127.792(c)(2)(A)	Identify intellectual property violations within given scenarios.	Unit 5 – Template Tags: Comment Tags, Author Tags, Conditional Logic	Developer tools include enforcement of proper authorship, content display logic, and metadata.
§127.792(c)(2)(B)	Summarize the ramifications and consequences of plagiarism and copyright infringement within a business context.	Unit 8 – Plugin Dev: Licensing, Security, Attribution	Plugin dev process reinforces ethical use, digital rights, and open-source attribution.
§127.792(c)(3)(A)	Apply UX design conceptual foundations to real-world scenarios.	Unit 4 – Template Hierarchy: Custom Templates, Index Files, Sidebars	Students adapt templates to reflect UX decisions based on hierarchy and user flow.
§127.792(c)(4)(B)	Analyze testing data to refine UX designs.	Unit 6 – REST API: Create JSON Data, Dynamic Output	REST API allows real-time output testing and refinement for improved UX.
§127.792(c)(5)(A)	Create visual designs that effectively communicate ideas.	Unit 1 – PHP for WordPress: Create a Theme, Apply Layout Markup	HTML/PHP structures allow for precise layout construction to communicate ideas.
§127.792(c)(6)(A)	Develop and deliver presentations to convey UX design concepts.	Unit 7 – Hooks: Actions and Filters to Control Presentation	Hooks offer UX-level customization for specific roles, interactions, or outputs.

Course: Web Game Development

TEKS Code & Standard	TEKS Description	Corresponding WordPress Certified Developer Objective(s)	Alignment Description
§127.767(c)(2)(A)	Demonstrate knowledge of HTML and CSS for structuring and designing webpages.	Unit 1 – PHP for WordPress: Theme Creation, Layout Markup	HTML/CSS template structure matches web interface designs for games.
§127.767(c)(2)(C)	Utilize reusable templates and styling frameworks to ensure consistent design.	Unit 2 – Pages: Navigation Menus, Template Hierarchies	Template reuse supports efficient design and code logic across game pages.
§127.767(c)(3)(B)	Apply best practices in visual layout and design.	Unit 4 – Template Hierarchy: Working with Custom Templates and Sidebars	Design strategies improve layout flow and UI logic across game screens.
§127.767(c)(4)(A)	Construct interactive elements using forms and multimedia.	Unit 6 – REST API: Output Dynamic Content, Add JSON Fields	Dynamic content hooks (REST API) align with real-time game event rendering.
§127.767(c)(5)(C)	Incorporate testing and debugging techniques for user feedback.	Unit 7 – Hooks: Action and Filter Hooks in WordPress Core	Hooks simulate dynamic game triggers or scene transitions.
§127.767(c)(6)(A)	Develop responsive web content that adapts across devices.	Unit 5 – Template Tags: Comment Tags, Author Tags, Display Logic	Template tags drive conditional output in layouts like player stats or levels.
§127.767(c)(7)(A)	Demonstrate knowledge of digital citizenship including legal and ethical behaviors.	Unit 8 – Plugin Dev: Licensing, Security, Attribution	Plugin architecture enforces proper reuse of assets and licensing best practices.