Open Awards

Quality Endorsed Unit

1 Unit Details

Unit Title:	Advanced JavaScript Programming
Unit Code:	CK3/4/WR/002
Level:	4
Credit Value:	9

2 Learning Outcomes and Criteria

Learning Outcome (The Learner will):	Assessment Criterion (The Learner can):
1. Create and run more advanced JavaScript programs.	1.1 Use advanced scope, including parent scope and block scope.
	1.2 Use arrow functions, including multi-line arrow functions.
	1.3 Use template literals, including tagged template literals.
	1.4 Implement advanced object properties, including computed properties.
	1.5 Implement array and object destructuring, including nested array and object destructuring.
	1.6 Implement advanced classes and objects, including creating a class with functions, and the import and export of modules.
	1.7 Use transpilation, iterators, and generators.
2. Implement advanced asynchronous programming techniques	2.1 Implement asynchronous programming, including event loops and the run-to-completion model.
	2.2 Implement call backs, including call back nesting.
	2.3 Implement promises, including promise chaining and associated functions such as resolve() and reject().

		2.4	Use the async and wait and keywords to implement error handling
3.	Manipulate JavaScript's Document Object Model (DOM)	3.1	Implement DOM chaining and navigation, including building an HTML document from the DOM tree.
		3.2	Manage DOM manipulation, including removing or replacing a node in the DOM and using the innerHTML() property.
		3.3	Implement DOM events and event objects, including adding and removing event listeners.
		3.4	Implement jQuery selectors, including traversing the DOM and jQuery method chaining.
4.	Implement advanced JavaScript testing techniques	4.1	Implement a test-driven development (TDD) environment and implementing the TDD cycle.
		4.2	Implement a variety of testing techniques, including black box testing, white box testing, unit and functional testing, and write a test-tracker.
		4.3	Use a variety of testing tools and environments, including the Mocha framework.
5.	Deploy functional programming techniques.	5.1	Build a range of imperative and declarative functions.
		5.2	Build a range of pure functions with no side-effects and including referential transparency.
		5.3	Build higher-order functions, including immutability and recursive immutability.
6.	Code inside a JavaScript environment.	6.1	Work inside a JS environment, such as Node.js, to produce code for mobile devices.
		6.2	Develop software using Node.js and the Node Package Manager and create a basic HTTP server.
		6.3	Implement streams and pipes, including readable and writable streams and piping streams.
		6.4	Create a basic Express server, with advanced routing and error handling.
		6.5	Implement a React server using the ReactDOM library