

## Module 21 Web Technologies

<b>Module title</b>	Web Technologies
<b>Module NFQ level (only if an NFQ level can be demonstrated)</b>	7
<b>Module number/reference</b>	BSCO-WT
<b>Parent programme(s)</b>	Bachelor of Science in Computing Science
<b>Stage of parent programme</b>	Award stage
<b>Semester (semester1/semester2 if applicable)</b>	Semester 2
<b>Module credit units (FET/HET/ECTS)</b>	ECTS
<b>Module credit number of units</b>	10
<b>List the teaching and learning modes</b>	Direct, Blended
<b>Entry requirements (statement of knowledge, skill and competence)</b>	Learners must have achieved programme entry requirements.
<b>Pre-requisite module titles</b>	BSCH-CSWD, BSCH-SSWD
<b>Co-requisite module titles</b>	None
<b>Is this a capstone module? (Yes or No)</b>	No
<b>Specification of the qualifications (academic, pedagogical and professional/occupational) and experience required of staff (staff includes workplace personnel who are responsible for learners such as apprentices, trainees and learners in clinical placements)</b>	Qualified to as least a Bachelor of Science (Honours) level in Computer Science or equivalent and with a Certificate in Training and Education (30 ECTS at level 9 on the NFQ) or equivalent.
<b>Maximum number of learners per centre (or instance of the module)</b>	60
<b>Duration of the module</b>	One Academic Semester, 12 weeks teaching
<b>Average (over the duration of the module) of the contact hours per week</b>	6
<b>Module-specific physical resources and support required per centre (or instance of the module)</b>	One class room with capacity for 60 learners along with one computer lab with capacity for 25 learners for each group of 25 learners

Analysis of required learning effort		
	Minimum ratio teacher / learner	Hours
<b>Effort while in contact with staff</b>		
Classroom and demonstrations	1:60	36
Monitoring and small-group teaching	1:25	36
Other (specify)		
<b>Independent Learning</b>		
Directed e-learning		
Independent Learning		78
Other hours (worksheets and assignments)		100
Work-based learning – learning effort		
<b>Total Effort</b>		250

Allocation of marks (within the module)					
	Continuous assessment	Supervised project	Proctored practical examination	Proctored written examination	Total
<b>Percentage contribution</b>	100%				100%

### Module aims and objectives

This module provides learners with the knowledge and skills needed to design and deploy web applications using a range of web application frameworks and content management systems. Learners study general concepts behind web application frameworks and CMSs, including how they work, design patterns, installation, and practical use. Learners get hands on experience deploying single-page applications, server-side web applications, and content management systems.

### Minimum intended module learning outcomes

On successful completion of this module, the learner will be able to:

1. Discuss the features of web application frameworks
2. Compare and contrast single-page applications with other formats
3. Evaluate the architecture and features of content management systems and discuss the design and architecture of web application frameworks and content management systems
4. Install and configure web application frameworks, single-page applications and content management systems
5. Plan, design and build web applications using both client-side web application frameworks and content management systems

## **Rationale for inclusion of the module in the programme and its contribution to the overall MIPLOs**

This module extends learners abilities in web development, building upon their core knowledge of client-side and server-side development with the latest techniques and technologies used in the deployment of modern web-based applications. Appendix 1 of the programme document maps MIPLOs to the modules through which they are delivered.

## **Information provided to learners about the module**

Learners receive a programme handbook to include module descriptor, module learning outcomes (MIMLO), class plan, assignment briefs, assessment strategy and reading materials.

## **Module content, organisation and structure**

### **Content Management Systems**

- Types of CMS
- Design
- Features

### **Working with Content Management Systmes (e.g. Drupal)**

- Installation
- Configuration
- User Management
- Creating Content
- Blocks
- Themes
- Modules
- Views

### **Web Application Frameworks**

- Types of web frameworks
- Design and architectures
- Features
- Client Side and Server Side JavaScript

### **Node.js**

- Installation
- Configuration
- Asynchornous I/O
- Events
- Modules
- NPM
- Express.js

### **Single-page applications**

- Types of single-page applications
- Design and architecture
- Features

### **Angular.js**

- Installation
- Configuration
- Data Binding
- Controllers
- Scope

### **Module teaching and learning (including formative assessment) strategy**

The module is taught as a combination of lectures and lab sessions. The lectures discuss and explain to learners the various concepts that underpin client-side and server-side web application development, beginning with server-side content management systems, before covering server-side run time environments, web application frameworks and single-page web application development.

In tutorials and practical lab sessions, learners practically apply these concepts by installing a CMS and building a simple data-driven website, before learning how to install and configure node.js, use npm and install and use frameworks to build a series of web applications.

Assessment is a series of practical assignments that cover the core topics, and a written report. The first is to build a website using a content management system. The second is to build a web application with a user interface using node.js and express.js. The third is a written report analysing and comparing web application frameworks covered so far. Finally, in groups, learners will plan, design and develop a complex full stack web application.

### **Timetabling, learner effort and credit**

The module is timetabled as two 1.5-hour lectures and two 1.5-hour labs per week.

Continuous assessment spreads the learner effort to focus on small steps before integrating all steps into the overall process of a development project.

There are 72 contact hours made up of 24 lectures delivered over 12 weeks with classes taking place in a classroom. There are also 24 lab sessions delivered over 12 weeks taking place in a fully equipped computer lab. The learner will need 78 hours of independent effort to further develop the skills and knowledge gained through the

contact hours. An additional 100 hours are set aside for learners to work on worksheets and assignments that must be completed for the module.

The team believes that 250 hours of learner effort are required by learners to achieve the MIMLOs and justify the award of 10 ECTS credits at this stage of the programme.

### **Work-based learning and practice-placement**

There is no work based learning or practice placement involved in the module.

### **E-learning**

The college VLE is used to disseminate notes, advice, and online resources to support the learners. The learners are also given access to Lynda.com as a resource for reference.

### **Module physical resource requirements**

Requirements are for a classroom for 60 learners equipped with a projector, and a 25 seater computer lab for practical sessions with access to SQLite and MySQL, a development environment (such as Notepad++) and a selection of web browsers (Chrome, Firefox, Edge) (this may change should better technologies arise).

### **Reading lists and other information resources**

#### **Recommended Text**

Abbott, N. and Jones, R. (2016) *Learning Drupal 8*. Birmingham, UK: Packt Publishing.

Powers, S. (2016) *Learning node: moving to the server side*. Sebastopol: O'Reilly.

#### **Secondary Reading:**

Haverbeke, M. (2018) *Eloquent JavaScript*. Available at: <https://eloquentjavascript.net/><sup>2</sup>

### **Specifications for module staffing requirements**

For each instance of the module, one lecturer qualified to at least Bachelor of Science (Honours) in Computer Science or equivalent, and with a Certificate in Training and Education (30 ECTS at level 9 on the NFQ) or equivalent.. Industry experience would be a benefit but is not a requirement.

Learners also benefit from the support of the programme director, programme administrator, learner representative and the Student Union and Counselling Service.

---

<sup>2</sup> Last accessed 26/07/2018

### Module Assessment Strategy

The assignments constitute the overall grade achieved, and are based on each individual learner's work. The continuous assessments provide for ongoing feedback to the learner and relates to the module curriculum.

No.	Description	MIMLOs	Weighting
1	Install, configure and build a web site using a server-side content management system (Drupal)	3,4	25%
2	Design and build a simple web application using node.js and express.js	3,4	20%
3	Write a report discussing and comparing web application frameworks	1,2	15%
4	Research, plan, design and build a single-page web application using angular.js and node.	1,2,3,4	40%

All repeat work is capped at 40%.

### Sample assessment materials

Note: All assignment briefs are subject to change in order to maintain current content.

## **Griffith College Dublin**

### **Assignment 1**

For this assignment you are to install Drupal and use it to build a small website for a company who run a gym.

The site must have a public facing part, and an administration section for maintaining the content.

The company requires:

#### A home page

This page should feature an introduction, links to the class timetable page as well as a preview of the 3 latest news stories.

#### A blog / latest news section

#### A class timetable

This page should display a list of classes. Each class should have the following information: Class name, the day of the week it is on, the time, a description, the instructor name.

(Examples of class types – Yoga, Spin, Step Aerobics, Zumba)

#### Photo gallery

All pages should have a consistent, persistent header, with the site title, description and navigation system.

All pages should be editable by the owner. They should also be able to create/delete and add pages.

The owner should also be able to create and modify an image gallery per page. Each image should have: an id, filename, alt text and caption. There can be a variable number of images per gallery. An image can be used on multiple pages.

#### **Marking:**

- Installation 20%
- Features 50%
- Presentation 10%
- Innovation 10%

#### **Submission Requirements:**

A zip file containing the following:

Your Drupal Project and MySQL Database dump files

A text document with your name, student number and any passwords needed to get the website running

Note:

You can install any modules from Drupal that you may need.

## **Assignment 2**

### **Build a Blog With Node.js and Express.js**

For this assignment you are to install node.js and express.js and use it to build a simple blog.

The blog should users to create/read/update/delete blog posts.

The site should display the last 10 blog posts, along with links to post a new entry.

Clicking on the heading of any blog post should display a detail page for that blog entry, along with links to delete and edit the post.

The site should have some CSS applied to make it usable.

#### Basic functionality:

Read, write, update, delete blog posts

A home page with chronological listing of past 10 posts

Detail pages for each post

#### For extra marks:

The ability to log in.

Only logged in users can edit etc.

#### **Marking:**

- Installation 10%
- Features 30%
- Coding 40%
- Presentation 10%
- Innovation 10%

#### **Submission Requirements:**

A zip file containing all files needed.

A text file with any installation instructions.



### **Assignment 3**

#### **Report – Comparison of Web Frameworks**

For this assignment you are to research and write a report comparing 2 or more web frameworks. Your report should discuss: what a web-framework is, key features, different types of architectures and compare and contrast at least 2 web frameworks.

#### Your report should cover:

- Server-side vs Client-side Frameworks
- Opinionated vs Unopinionated
- MVC
- Push vs Pull

#### Marking:

- Research 30%
- Content 50%
- Writing 10%
- Structure 5%
- Presentation 5%

#### Submission Requirements:

One PDF, named yourname\_studentnumber\_assignment\_03.pdf uploaded to Moodle.

### **Assignment 4**

#### **Research, Plan, Design and Build a Single-Page Web Application**

For this assignment you are to form groups of 2 and research, plan, design and build a single-page web application using node.js and angular.js.

The app should be a productivity/to do app that allows users to:

- Sign up
- Log in
- Create tasks (With a date for completion, level of priority etc.)
- Set tasks as complete
- Assign others to a task
- Get alerts about incomplete tasks

As well as these features you should come up with your own features.

The app should be built using HTML, CSS, JavaScript, node.js, angular.js and any other plugins or libraries you deem necessary.

In addition to the main app you should supply documentation outlining:

- The features
- The architecture of the app
- The database structure
- The project timeline
- Reflection on how the project went

**Marking:**

- App design/architecture 20%
- Coding 30%
- Features 20%
- Innovation 10%
- Usability/Design 5%
- File structure 5%
- Documentation 10%

**Submission Requirements:**

All files, database dump, and package information in one folder called groupNumber\_assignment\_04. Zipped and uploaded. Also include in this the documentation, in PDF format, named groupnumber\_assignment\_04\_documentation.

Groups will be assigned a number.