Module: Web Programming 381

Module name:	Web Programming 381
Code:	WPR381
NQF level:	7
Type:	Fundamental – Bachelor of Computing (Software Engineering stream)
Contact time:	38 hours
Structure time:	6 hours
Self-directed time	46 hours
Notional hours:	90 hours
Credits:	9
Prerequisites:	WPR281, PRG282

Purpose

The purpose of this course is to teach students how to use a framework when building a web application. The application will make use of a data repository to persist its state. Concepts of APIs, Application Security and Asynchronous Programming will also be explored.

Outcomes

Upon successful completion of this module, the student will be able to:

- Demonstrate integrated knowledge of the central areas of dynamic website programming, including an understanding of and the ability to apply and evaluate the key terms, concepts, facts, principles, rules and theories of dynamic web programming; and detailed knowledge of web development frameworks, web systems and web standards.
- Demonstrate an understanding of a range of methods of enquiry in dynamic web programming, and their suitability to specific investigations; and the ability to select and use appropriate website development techniques in particular to use the features of some framework for the purpose of designing and deploying a dynamic website.
- Identify, analyse, evaluate and critically reflect on strengths and weaknesses of web design and implementation by means of a framework, applying evidence-based solutions and theory-driven arguments.
- Communicate effectively with a variety of audiences through a range of modes and media, in particular to present a clear, coherent and independent exposition of functional websites to IT and/or non-IT personnel via reports or presentations and using appropriate academic discourse.
- Identify, evaluate and address his or her learning needs in a self-directed manner, and to facilitate collaborative learning processes by consulting various sources of information and peer networks.

Assessment

Assessment is performed using a variety of instruments:

- Continuous evaluation of theoretical work through two projects, one formative assessment, and a summative test.
- Continuous evaluation of project work, whereby the student must create and deploy a website.
- Final assessment through a written examination.
- The assignments or projects collectively will count 30% of your class mark.
- All tests will collectively account for 70% of your class mark.
- Your class mark contributes 30% towards your final mark for the subject, while the final assessment accounts for 70% of your final mark.

Teaching and Learning

Learning materials

Prescribed books (EBSCO)

- 🕮 Mithun Satheesh, Bruno Joseph D'mello and Jason Krol (2015) Web Development with MongoDB and NodeJS - Second Edition : Build an Interactive and Full-featured Web Application From Scratch Using Node.js and MongoDB. Birmingham, UK: Packt Publishing (Community Experience Distilled).
- 🕮 Yaapa, H. (2013) Express Web Application Development. Birmingham: Packt Publishing.
- 🕮 David Herron (2018) Node.js Web Development : Server-side Development with Node 10 Made Easy, 4th Edition. [S.l.]: Packt Publishing.

Additional Material

- 🕮 Wilken, J. (2018). *Angular in Action.* Manning. ISBN 9781617293313.
- 🕮 Karpov, V., Netto, D. (2015). *Professional AngularJS*. Manning. ISBN: 978-1-118-83207-3

Learning activities

The teaching is a combination of the presentation of practical and theoretical concepts, and exercises and discussions. It is practice-oriented, with a mandatory assignment and project which must be completed during the course. The course also includes a component of research, and the research will need to be presented during class in a formal session.

Notional learning hours

Activity	Units	Contact Time	Structured Time	Self-Directed Time
Lecture		27.0		14.0
Formative feedback		6.0		
Project	2	5.0		12.0
Assignment	1			3.0
Test	2		4.0	8.0
Exam	1		2.0	9.0
		38.0	**6.0**	**46.0**

Syllabus

- An exploration of the architecture of a web framework.
- Comparison of frameworks, and their suitability for some business problem.
- Setting up a project using a framework, for example Angular, Node and Express.
- Installing and configuring dependencies.
- Concepts of directives and data binding within a framework.
- Fundamentals of routing and navigation within a framework.
- An overview of web services and how web applications use data.
- Integrating APIs with web applications.
- Integrating external libraries.
- A consideration of security aspects in a web application.
- Deploying a web application.