Unit 35: Web Applications Development

Unit code: K/601/1510

QCF Level 5: BTEC Higher National

Credit value: 15

Aim

To enable learners to understand the concepts of web applications and apply the skills to develop and test web applications using server-side technologies.

Unit abstract

The internet has emerged as a dominating area of IT development. The ever-expanding applications within the global community that communicates, trades and exchanges information (seamlessly) has meant that the Internet and its associated technologies is a rapidly growing and changing area that requires in-depth knowledge as well as a wide range of skills. These web technologies have also been used to develop e-Commerce, intranet, extranet and social networking systems to meet the needs of modern businesses and associations.

Learners should already be familiar with website design and management, and will be able to apply their own web development skills to this unit. Learners will enjoy extending their web-development skills by understanding the concepts of web-based applications using server-side technologies. This unit takes client-side web-development one step further by introducing server-side application development. This unit focuses on server-side technologies and how server-side scripting can be used to create sophisticated web-based applications.

Learners will understand the concepts of web-based applications using one or more different server-side scripting languages, such as ASP, JSP and PHP. Many web-based systems also include database systems, which enables data to be processed dynamically. Learners will gain experience of developing web-enabled database systems, using SQL statements combined with server-side scripts to manage the process of information.

Learners will develop skills in specific techniques and also able to select when and where they are most appropriate, basing this decision on client and user needs. They will also ensure that their applications comply with the relevant legislation and guidelines.

Learning outcomes

On successful completion of this unit a learner will:

1 Understand the concepts of web application development

2 Be able to design web applications

3 Be able to implement web applications

4 Be able to test web applications.

Unit content

1 Understand the concepts of web application development

Users: types eg expert, regular, occasional, novice, special needs; requirements, eg psychological, cultural, social and environmental, health and safety, education and work

Site analysis: purpose eg communication, real-time information, commercial, government, education, business, entertainment, downloading/uploading, web storage; fit for purpose, eg meets organisational and site objectives; planning, eg storyboarding, structure, hypermedia linkage, search engine key words, graphical design, user interface, audio/video sources, animation, text design; maintenance, eg plans, logs, disaster recovery, testing

Accessibility: features eg alternative text, resizable fonts, support for screen readers, adjustable fonts

Legislation: laws, guidelines and standards, eg Disability Discrimination Act, Data Protection Act, E-Commerce Regulations Act, W3C validation, copyright and intellectual property rights

Functionality: functions, eg shopping cart, reserve order, manage user profile, web content management, upload files

Scripting languages: server-side eg ASP (Active Server Pages), ASP.NET, PHP (Hypertext Preprocessor), JSP (Java Server Pages), Cold Fusion, Perl, Java Applet, Flash; advantages eg faster processing time, data processing, data storage; client-side eg JavaScript, VBScript

Security: security requirements, eg user accounts, account restrictions, procedures for granting and revoking access, terms of use, system monitoring

2 Be able to design web applications

Identification of need: nature of interactivity eg online transactions, static versus dynamic; client needs and user needs, eg image; level of security, eg user/administrator access; development timescales, support and maintenance contracts, costs, visibility on search engines; end user need eg appropriateness of graphics, complexity of site, delivery of content

Design tools: concept designing, eg mood boards, storyboarding; layout techniques eg frames, tables, block level containers (DIV), inline containers (SPAN); templates; colour schemes; screen designs, use case diagrams, pseudo code; other eg outline of content; database design, eg data flow diagrams, entity relationship diagrams

Database design: documenting the design; back end design, eg defining relationships, normalisation, naming conventions; front end design, eg user interface, security measures

3 Be able to implement web applications

Structure: layout of pages; navigation; format of content and cascading style sheets (CSS); page elements, eg headings, rules, frames, buttons, text and list boxes, hyperlinks/anchors, graphical images, clickable images/maps; interactive features, eg catalogue of products, shopping cart; images and animation

Content: proofed, correct and appropriate; reliability of information source; structured for purpose, eg prose, bullets, tables

Development: mark-up languages eg HTML (Hypertext Mark-up Language), DHTML (Dynamic Hypertext Mark-up Language); client-side scripting languages eg JavaScript, VBScript; features and advantages of software languages; web authoring software tools

Tools and techniques: navigation diagram eg linear, hierarchy or matrix; building interactivity tools, eg pseudo-code for client-server scripting; adding animation and audio/visual elements; ensuring compliance with W3C; meta-tagging; cascading style sheets

Server-side interaction: manage and process data, eg client, server; action events, action responses, login/logout

Server-side scripting languages: ASP, JSP, PHP, Cold Fusion, Perl

Database connectivity: common methods of using/accessing databases on a web server eg SQL (Structured Query Language), MySQL, ODBC (Open Database Connectivity), JDBC (Java Database Connectivity), ADODB (ActiveX Data Objects).

Web-programming concepts: objects, eg response, request, application, session, server, error, file system, text stream, drive, file, folder, dictionary, ADO; components, eg email, file, file uploads, date/time; syntax, variables, procedures, forms, cookies, sessions, applications.

4 Be able to test web applications

Review: functionality testing (user environments, links and navigation); content; check against user requirements; user acceptance; audit trail of changes.

Mechanisms: browser compatibility testing, platform testing, script-language testing; valid HTML, server-script and database-script code; checking functionality against requirements, check internal and external hyperlinks to other web pages and media content (web files, web documents, images), error detection, error messages, dry running.

Feedback: record feedback, eg surveys, questionnaire, interviews; analyze feedback; present results

Supportive documentation: test plan (test data, expected results, actual results); test results; programmer guidance; user guidance (instructions)

Testing by: types eg administrator, user, automated scripts.

Learning outcomes and assessment criteria

Learning outcomes On successful completion of this unit a learner will:	Assessment criteria for pass The learner can:
LO1 Understand the concepts of web application development	1.1 critically evaluate the functions and advantages of web applications 1.2 critically compare different types of server-side and client-side scripting languages 1.3 examine web security concerns and make recommendations for security improvements
LO2 Be able to design web applications	2.1 design a web application to meet a given requirement 2.2 synthesise client-side and server-side functionality in a web application 2.3 apply a database design for use in a web application 2.4 evaluate alternative designs and solutions to meet a given requirement
LO3 Be able to implement web applications	3.1 implement a web application to a prepared design using client-side and server-side scripting languages 3.2 implement a web-enabled database management system to store, retrieve and manipulate data in a web application 3.3 identify and implement opportunities for error handling and reporting
LO4 Be able to test web applications	4.1 critically review and test a web application using a web-enabled database management system 4.2 analyse actual test results against expected results to identify discrepancies 4.3 critically evaluate independent feedback on a developed web application and make recommendations for improvements 4.4 create user documentation for a developed web application.

Guidance

Links to National Occupational Standards, other BTEC units, other BTEC qualifications and other relevant units and qualifications

The learning outcomes associated with this unit are closely linked with:

Level 3	Level 4	Level 5
	Unit 14: Website Design	Unit 39: Computer Games Design and Development
	Unit 15: Website Management	Unit 40: Distributed Software Applications
	Unit 18: Procedural Programming	Unit 41: Programming in Java
	Unit 19: Object Oriented Programming	Unit 42: Programming in .NET
	Unit 20: Event Driven Programming Solutions	
	Unit 21: Software Applications Testing	
	Unit 22: Office Solutions Development	
	Unit 23: Mathematics for Software Development	

This unit has links to the Level 4 and Level 5 National Occupational Standards for IT and Telecoms Professionals, particularly the areas of competence of:

- Human Computer Interaction
- IT/Technology Infrastructure Design and Planning
- Software Development.