# Next Generation Higher National Unit Specification

## Application Development for Web (SCQF level 8)

**Unit code:** J7E1 48

**SCQF level:** 8 (24 SCQF credit points)

**Valid from:** session 2023–24

## Prototype unit specification for use in pilot delivery only (version 1.0) June 2023

This unit specification provides detailed information about the unit to ensure consistent and transparent assessment year on year.

This unit specification is for teachers and lecturers and contains all the mandatory information required to deliver and assess the unit.



This edition: June 2023 (version 1.0)

# Unit purpose

This unit covers a range of competencies, including knowledge and understanding of modern front-end and back-end web markup and programming languages, frameworks, and tools and technologies. It enables learners to build interactive, dynamic web applications (apps) deployed to the cloud.

This is a specialist unit, suitable for learners with an interest in software development, particularly full-stack web app development.

Learners analyse a client requirement and design and build a secure full-stack interactive web app that meets the requirement. They implement their web app using selected front-end and back-end technologies and deploy it to the cloud.

Learners use a version control system to manage and keep track of their source code history and as a tool for sharing, collaborating, and repositing code.

Before starting the unit, learners should have some experience in front-end web development, including knowledge of hypertext markup language (HTML), cascading style sheets (CSS) and JavaScript.

On completion of this unit, learners can progress to other units in software development and web development at SCQF level 8 and higher.

# Unit outcomes

Learners who complete this unit can:

1 use a version control system to manage and track source code history
2 use a software tool for sharing, collaborating, and repositing code
3 design an interactive front-end for a web app
4 develop an interactive front-end for a web app to a design
5 build a secure back-end for a web app using an appropriate programming language and technology stack
6 perform and document stringent testing of the front-end and back-end of a web app
7 deploy a full-stack web app to a cloud platform

## Evidence requirements

Learners must provide product evidence. Knowledge is inferred from the product evidence.

Learners' product evidence is the design and build of a secure full-stack interactive web app, which they must deploy to the cloud and implement using current front-end and back-end technologies.

Learners must demonstrate use of:

- a version control system to manage and keep track of their source code history
- a tool for sharing, collaborating, and repositing code

Provide learners with a brief, its associated requirements, and a design specification. They then build and test a secure full-stack web app to match the client and design requirements.

The brief must provide sufficient assessment evidence to demonstrate the use of both front-end and back-end development, and current technologies. Learners should produce evidence for all skills covered by the unit outcomes.

Learners' completed web apps should demonstrate:

- clean, standards-compliant, and fully documented code
- full functionality and responsiveness across a range of modern browsers and mobile devices

Learners should document the results of stringent testing and include functionality, performance, compatibility and security testing.

Learners can produce evidence over an extended period and under lightly-controlled conditions. This assessment is open book and learners may work in their chosen location with access to suitable resources, such as code libraries. Evidence produced in lightly-controlled conditions must be authenticated. The Guide to Assessment provides further advice on methods of authentication.

The standard of evidence should be consistent with the SCQF level of the unit.

# Knowledge and skills

The following table shows the knowledge and skills covered by the unit outcomes:

| Knowledge | Skills |
|---|---|
| Learners should understand:<br><br>♦ the difference between, and uses of, websites, web apps, progressive web apps (PWA) and mobile apps<br>♦ trends and factors to consider in converting web apps to mobile apps<br>♦ the importance of version control during web development and current, popular version control systems<br>♦ the current tools they can use for version control and to manage and track source code history<br>♦ the current tools they can use for sharing, collaborating, and repositing code<br>♦ databases and their use in full-stack web app development<br>♦ the uses of structured query language (SQL) and not only structured query language (NoSQL)<br>♦ the differences between SQL and NoSQL databases<br>♦ the client-server model<br>♦ the Document Object Model (DOM)<br>♦ JavaScript libraries and frameworks<br>♦ programming languages suited to web app development<br>♦ CSS frameworks<br>♦ essential security considerations related to web app development<br>♦ the purpose and role of stringent testing of web apps before public release<br>♦ cloud deployment for web apps | Learners can:<br><br>♦ use a version control tool during the build of a web app<br>♦ use a tool for sharing, collaborating and repositing code<br>♦ code using front-end markup and programming languages<br>♦ design and build an interactive front-end web app<br>♦ code using a back-end programming language<br>♦ use a framework to support back-end development<br>♦ build a full-stack web app<br>♦ deploy a full-stack web app to the cloud<br>♦ carry out stringent testing<br>♦ implement a cloud-based web app |

# Meta-skills

Throughout this unit, learners develop meta-skills to enhance their employability in the computing sector.

## Self-management

This meta-skill includes:

- focusing: sorting and maintaining documentation for software apps; attention to detail
- adapting: critically reflecting on own skills; self-learning to develop wider skills and extend development beyond requirements and taught content

## Social intelligence

This meta-skill includes:

- communicating: giving information; producing user documentation; commenting on code during development; writing reports on outcomes of testing processes
- collaborating: social perceptiveness; listening and conveying information

## Innovation

This meta-skill includes:

- curiosity: questioning constructively to identify requirements; problem recognition
- creativity: using imagination to provide a solution that meets the needs of both the app and the user; visualising to create an overall impression of the completed solution throughout the process
- sense-making: analysis; seeing the bigger picture
- critical thinking: logical thinking to ensure a coherent approach and to meet requirements appropriately

# Delivery of unit

You can deliver this unit on its own or as part of a group award. If you deliver it as part of a group award, you can combine assessment with other units in the award.

The time required varies depending on the previous experience of individual learners and if you are delivering the unit in conjunction with other related units. While the exact time allocated is at your centre's discretion, the notional design length is 120 hours. Suggested guidelines are:

**Outcome 1** — Use a version control system to manage and track source code history
(5 hours)

**Outcome 2** — Use a software tool for sharing, collaborating, and repositing code
(5 hours)

**Outcome 3** — Design an interactive front-end for a web app
(15 hours)

**Outcome 4** — Develop an interactive front-end for a web app to a design
(25 hours)

**Outcome 5** — Build a secure back-end for a web app using an appropriate programming language and technology stack
(55 hours)

**Outcome 6** — Perform and document stringent testing of the front-end and back-end of a web app
(10 hours)

**Outcome 7** — Deploy a full-stack web app to a cloud platform
(5 hours)

# Additional guidance

The guidance in this section is not mandatory.

## Content and context for this unit

It is beneficial if learners have previous experience of coding in HTML and CSS for designing a front-end web interface. Experience of using another programming language, especially JavaScript, is also helpful.

When you deliver the unit, you should make sure learners understand the difference between, the uses of, and the benefits and disadvantages of:

- websites
- web apps
- progressive web apps (PWAs)
- native mobile apps

You should always refer to current trends and technologies. Learners must have access to cloud services.

### Use a version control system to manage and track source code history, and use a software tool for sharing, collaborating, and repositing code (outcomes 1 and 2)

You can deliver these outcomes together. This should include the importance of version control during web app development and current, popular version control systems that can manage and track source code history, and be used for sharing, collaborating, and repositing code.

You should expose learners to a variety of options, where possible, but they should become familiar in the use of at least one version control system that they can use throughout the unit.

We suggest the following options:

- Git and GitHub: GitHub is a well-known open-source project management tool. Git is a version control system. GitHub is a platform where people share Git repositories online.
- Amazon Web Services (AWS) CodeCommit: Amazon’s Git-based version control service is an option if learners use other AWS services.
- Azure DevOps Server: this includes the functionality of GitHub with some additional unique features.

### Design and develop an interactive front-end for a web app (outcome 3)

To build a front-end web app, learners should use a combination of:

- HTML for content structure
- CSS to manage formatting, responsiveness and layout
- JavaScript for interactivity and to be able to use a JavaScript front-end framework

How you deliver this outcome depends on the level of HTML, CSS and JavaScript experience learners already have. You can deliver the unit in conjunction with other unit(s) that also cover these front-end languages.

The unit does not cover UX and user interface (UI) design, but you should make learners aware of their importance in the development of any front-end interface. You can give learners designs in the form of wireframes, style guides and digital mock-ups to replicate by implementing code.

Learners may use a front-end framework and library. Widely used frameworks include Angular and Vue which are both written in JavaScript. React is a popular and powerful front-end library that is well suited for large-scale web projects. Vue is known as 'the progressive JavaScript framework'. Vue is smaller in size and easier to learn than Angular, and is good for most project sizes.

Learners can be supplied with design documentation that exemplifies how to provide a good user experience (UX).

### Build a secure back-end for a web app using an appropriate programming language and tech stack (outcome 4)

Learners can use various options to develop their full-stack web app. The choice is at your centre's discretion, but it should be a current technology.

The back-end framework selected is based upon programming language and tech stack.

We suggest the following options:

- MongoDB, Express, React and NodeJS. Together these are known as full-stack MERN.
- Django is a popular Python framework that focuses on code reusing. It is a secure framework that is user-friendly and easy to learn.
- Laravel, the PHP framework based on MVC architecture, can be used to build modern secure web apps.

### Perform and document stringent testing of the front-end and back-end of a web app (outcome 5)

Debugging and testing of the front-end and back-end is ongoing throughout the build. You should emphasise the importance of stringent testing for functionality, performance, compatibility and security before public release. This should include the essential security considerations related to web app development.

Learners should test their web app for:

- functionality
- interface
- compatibility
- performance
- security
- usability

### Deploy a full-stack web app to a cloud platform (outcome 6)

How you deliver this outcome depends on learners' chosen programming languages and the tech stack, version control and collaboration used for the build.

Options could include:

- Netlify
- web hosting services from Google Cloud
- AWS: web app hosting services
- Azure

## Approaches to assessment

We recommend that you assess the unit through a single project assignment that provides learners with the opportunity to demonstrate knowledge and skills in all of the unit outcomes.

This assignment should take the form of a full-stack web app for which you have provided a brief. The design should require consideration of UX as well as full documentation of the project.

Learners should demonstrate their final product as a working web app.

# Equality and inclusion

This unit is designed to be as fair and as accessible as possible with no unnecessary barriers to learning or assessment.

You should take into account the needs of individual learners when planning learning experiences, selecting assessment methods or considering alternative evidence.

Guidance on assessment arrangements for disabled learners and/or those with additional support needs is available on the assessment arrangements web page: www.sqa.org.uk/assessmentarrangements.

# Information for learners

## Application Development for Web (SCQF level 8)

This information explains:

- what the unit is about
- what you should know or be able to do before you start
- what you need to do during the unit
- opportunities for further learning and employment

## Unit information

This unit covers current front-end and back-end web technologies, along with markup and programming languages, frameworks and tools that enable you to build interactive, dynamic web applications (apps) that you can deploy in the cloud.

This is a specialist unit, designed to introduce you to full-stack web app development.

Before starting the unit, it is beneficial for you to have experience in front-end web development using hypertext markup language (HTML), cascading style sheets (CSS) and JavaScript.

You analyse a client requirement and design, and then build and test a secure full-stack interactive web app. You implement it using current front-end and back-end languages and frameworks and deploy it to the cloud.

You learn how to use a version control system to manage and track your source code history and as a tool for sharing, collaborating, and repositing code.

You are assessed on product evidence in the form of a secure full-stack interactive web app that you build and test to match the client and design requirements in a brief.

Throughout the unit, you develop meta-skills covering self-management, social intelligence, and innovation.

On completion of this unit, you can progress to other units in software development and web development at SCQF level 8 and higher.

# Administrative information

**Published:** June 2023 (version 1.0)

**Superclass:** CB

## History of changes

| Version | Description of change | Date |
|---|---|---|
| | | |
| | | |
| | | |
| | | |

Note: please check SQA's website to ensure you are using the most up-to-date version of this document.