ICAWEB503A Create web-based programs

introduction to web-based programs

This topic covers the performance outcomes, skills and knowledge required to develop web applications.

learning outcomes

On successful completion of this topic you will be able to:

* obtain requirements
* develop web applications that keep track of user data between browser requests
* document web applications

To achieve the learning outcomes for this topic you will need to set aside approximately **60 hours** to complete the learning activities and the final project. Refer to the study schedule to see the expected timeframes for completing this topic.

pre-requisite requirements

There are no pre-requisite requirements for completing this topic. It is expected that you have a basic understanding of web design principles.

resources

The following text is a recommended text for students studying these units:

*Title:* PHP & MYSQL: The Missing Manual  
*Author:* Brett McLaughlin  
*Publisher:* O'Reilly Media / Pogue Press  
*Formats:* Print, eBook, Safari Books Online  
*Published:* November 2011  
*Print ISBN:* 978-0-596-51586-7 | ISBN 10: 0-596-51586-3  
*Ebook ISBN:* 978-1-4493-1887-1 | ISBN 10: 1-4493-1887-8

This book is available in both print and ebook versions from a number of bookshops such as Amazon, Fishpond and O'Reilly

assessment

See the project introduction page.

units of competency

The is topic covers the requirements for the following units of competency:

* ICAWEB503A Create web-based programs

key to symbols

These symbols are located throughout the documentation. They an indicate an action to be taken at a particular stage of the learning resource.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Activity symbol | **Activity** | Highlights an activity to enable you to practice the skills you have just learnt. | Reading symbol | **Reading** | Indicates additional reading on the subject matter. |
| Audio symbol | **Audio** | Indicates an audio or voice recording. | Website symbol | **Website** | Indicates a link to an external website that contains additional information. |
| Chat symbol | **Chat** | Indicates a web conference or chat session. | Video symbol | **Video** | Indicates a video to watch |
| Important symbol | **Important** | Indicates an important concept or idea to take note of. |  |  |  |

# introduction to web-based programs

# Key conceptskey concepts

* What is a web application?
* Hypertext Transfer Protocol (HTTP)
* Stateless programming
* Session Management
* Creating web applications
* Development Frameworks

In addition to the material provided you will need to spend time in further research of these and more concepts.

# what is a web application?

The days of the web consisting of static pages and content are moving further and further behind us. If you have ever used one of the common webmail applications such as Hotmail or Gmail, or regularly do your banking online then you are already a converted user of web-based applications.

Essentially a web application is "an application that is usable only with an active Internet connection and that uses HTTP as its primary communications protocol".

*Source: http://www.businessdictionary.com/definition/web-based-application.html, accessed September 2014*

Generally these applications use a web browser as the client making them readily accessible to many internet connected devices. In a previous stage of this course you developed a simple website that integrated with a database for viewing and updating a used car yards stock information. This was an example of a simple web application.

In the early days web application where commonly referred to as 'software as a service', or SaaS but this has been, in many cases, replaced by the current buzzword 'cloud computing'. The basic idea is still the same however of delivering a device independent, globally available software application.

Two big players in this field currently are Microsoft and Adobe. Microsoft officially released Office 365 in June 2011, that same year, in October, Adobe announced the release of Creative Cloud as a subscription based service for their suite of products. According to Adobe, the CS6 version of it's software will be the last locally installed version available.

But cloud based applications aren't new. Google Docs offers web users access to a web-based suite of products that includes a word processor, spreadsheet and presentation package and has been available to users since 2007. Even prior to that however web users have been making use of webmail applications. Some may remember a web-based email application known as hotmail which was launched backed in 1996. Hotmail gave users access to email from any internet connected computer, anywhere in the world and was not reliant on a users ISP. Hotmail was later purchased by Microsoft and has since being updated and re-badged as Outlook.com.

These web-based applications enable users to access and work on their files from any internet connected, web enabled device and often offering some amount of storage for files as well as the ability to share and collaborate on files.

## Shopping cartsome common examples:

* content management systems
* webmail applications
* shopping carts

## web applications advantages

Some of the advantages of web applications include:

* they function independently of the operating system
* they can be deployed almost anywhere
* they, usually, require no installation
* they can be accessed from any device with a web browser

## web applications disadvantages

Some of the disadvantages, or considerations that need to be made, of web applications include:

* Security - with your data and information stored on a remote server you need to be confident that there is adequate security and backups procedures in place
* Copyright and ownership - who actually owns the content on these remote servers. It is often prudent to read the fine print.

### **Further researchfurther research**

Web Applications: What are They? What of Them?: <http://www.acunetix.com/websitesecurity/web-applications/>

What is a Web App? Here's our definition: <http://web.appstorm.net/general/opinion/what-is-a-web-app-heres-our-definition/>

Web Application: <http://en.wikipedia.org/wiki/Web_application>

## Importanttextbook reading:

The following readings from the textbook looks at the practical application of developing web applications:

Part 3: From Web Pages to Web Applications

# hypertext transfer protocol

The Hypertext Transfer Protocol (HTTP) is the communication protocol that formats and sends requests from the client's browser to the web server and then formats and sends the web server's response back to the client. HTTP defines the standards responsible for the exchange of information, such as files and other data, across the World Wide Web. This communication protocol is therefore responsible for the transfer of data within a web application and so it is important that you have a basic understanding of how it works.

Hypertext Transfer Protocol over Secure (HTTPS) is a secure version of HTTP where the data is encrypted before being sent over the network.

It is important also to understand the following two aspects of HTTP in terms of web application development. (The issues and there affect on web application will be discussed in the next sections)

* HTTP is connectionless which means that the connection between the client and the web server is disconnected after each request / response.
* HTTP is stateless meaning that there is no record of previous requests. See the next section for more detailed information

HTTP is also media independent which means that any type of data can be transmitted providing both the client and the web server understand it. This means that, as mentioned previously, web applications can run across multiple devices and operating systems without some of the shortcomings of locally installed software applications.

### **Further researchfurther research**

Hypertext Transfer Protocol: <http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol>

# stateless programming

Stateless programming means that there is no record or information about what has previously occurred in an application. This means that each request must be processed without any knowledge of what has previously been requested and is processed entirely on the information supplied with the request. Every HTTP request is theoretically independent of previous requests and therefore HTTP is an example of stateless programming.

In practical terms however there is often a need for web application developers to maintain state. For example when dealing with shopping carts it is important to be able to maintain the state of the shopping cart such as when the user wants to add or delete items from the cart. Even though HTTP is stateless web developers make use of cookies or API's to maintain state.

### **Further researchfurther research**

Stateless: <http://whatis.techtarget.com/definition/stateless>

# session management

To understand session management we firstly need to understand the term session in the context of web applications. A session can be defined as exchange of information between a unique user and the web site / application. This means that each user must be assigned a unique session ID that identifies the user with the web application. Session management therefore, in the context of web applications, refers to the web application creating a session token for the user that identifies them and retains their information.

For example when you login to a website you will often want the site to retain your login information for that session so that you do not need to enter it every time you make a request of the site. This is important when developing web applications and the implications to the application user.

### **Further researchfurther research**

Session Management Cheat Sheet: <https://www.owasp.org/index.php/Session_Management_Cheat_Sheet>

Web Based Session Management: <http://www.technicalinfo.net/papers/WebBasedSessionManagement.html>

## Importanttextbook reading:

The following readings from the textbook also cover the practical application of stateless programming and session management in PHP:

Chapter 13: Cookies, Sign-Ins, and Ditching Crummy Pop-Ups

Chapter 14: Authorization and Sessions

# creating web applications

Like any other development task including creating websites, the development of web applications should follow a structured process. Essentially the four basic steps are:

* Plan
* Design
* Develop
* Test

Throughout this course you have looked at and implemented these different areas. For example you have planned and developed database structures prior to creating and implementing the database. You have been involved in the development of tests and the testing of your completed sites.

### **Further researchrecommended reading**

Tips for building your first web app: <http://sixrevisions.com/web-development/tips-first-web-app/>

Web Application Development Guidelines: <http://www.cygnismedia.com/web-application/web-app-development.html>

How to build your first web application - Tutorial series: <http://bigemployee.com/how-to-build-your-first-web-application-tutorial-series/>

Web applications can be developed in quite a number of modern languages such as:

* Active Server Pages (ASP & ASP.NET)
* Perl Hypertext Preprocessor (PHP)
* Java Server Pages
* jQuery / AJAX
* Python
* Ruby

### **Further researchfurther research**

The following articles refer to different tools / frameworks for developing web applications. Whilst the particular code may not be your chosen preference, the principal behind each of these tutorials still stands:

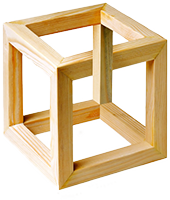
Creating a PHP Application: <http://www.phpro.org/tutorials/Creating-A-PHP-Application.html>

Creating your first PHP Application: <http://buildinternet.com/2009/12/creating-your-first-php-application-part-1/>

How to build a web application from scratch with no experience: <http://lifehacker.com/5336113/how-to-build-a-web-application-from-scratch-with-no-experience> created using Ruby on Rails

Creating a Web App from Scratch: <http://css-tricks.com/app-from-scratch-1-design/> an older article this app is built using AJAX

# development frameworks

Web application frameworks are software frameworks that assist in the rapid development of web applications, once you are familiar with using the framework. You do not have to use a framework to develop your application, as mentioned previously you have developed a basic application using PHP, MySQL, HTML and CSS. There are quite a number of frameworks available for developing web applications for the different languages that are available.

### **Further researchfurther research**

Web application framework: <http://en.wikipedia.org/wiki/Web_application_framework>

Comparison of web application frameworks: <http://en.wikipedia.org/wiki/Comparison_of_web_application_frameworks>

### **Further researchframeworks**

The following frameworks are provided as a starting list of possible frameworks for developing web-based applications. As a developer you should evaluate each as to it's suitability or otherwise to your development needs:

#### **php frameworks**

Symfony: <http://symfony.com/>

Yii Framework: <http://www.yiiframework.com/>

#### **other frameworks**

Ruby on Rails: <http://rubyonrails.org/> (uses Ruby language)

project part 4: web application development & project monitoring

For this project you are required to develop your web application and monitor the execution of the project with reference to your project plan.

You should save all project files in your previously created '**web\_app**' folder, in sub folders as necessary.

task 1 - web application development

For this task you are required to develop your web application based on your project plan. You should:

* Design your web application
* You should ensure that you properly create your design documentation that may include some or all of the following:
  + database design documents: (conceptual model, logical data, entity relationship diagram)
  + wireframes and/or mockups
  + interface screens mockups
  + color schemes
  + navigation maps / site maps
  + technical specifications and requirements
* Develop your web application
* Test your web application. You should ensure that you:
  + test against your quality management processes developed for your project plan
  + carry out usability and user testing with a minimum of one user

task 2 - project monitoring

NoteThis task should be completed in conjunction with Task 1.

For this task you are required to monitor and manage the implementation of the project. You should:

* You will need to document a minimum of one (1) change to the project.
* For each change to your project you should complete a change request form
* You should also update your project plan where necessary including gantt chart
* You should complete two (2) project performance reports according to the study schedule

project part 5: project finalisation

For this project you are required to complete the closure stage of your project including the delivery of a finalisation report and your completed web application.

You should save all project files in your previously created '**web\_app**' folder, in sub folders as necessary.

task 1 - finalisation report

For this task you are required to create a finalisation report for your project (you are welcome to use the template provided in the learning resource as the basis for your finalisation report). You should:

* Ensure you complete the information where necessary for each of the nine project management key areas in the finalisation report
* Ensure that you complete the lessons learned section for each area identifying 1 - 2 lessons learned for each area.
* Your finalisation report should include both:
  + things that worked well and how these could be implemented in future projects
  + things that did not work well and how these can be avoided in future projects
* Note: You may find that some lessons learned covered several of the key areas so you are welcome to carry these across and identify the application to the particular area.

task 2 - web application documentation

For this task you are required to document your web application. You should:

* Create a brief user guide / manual for your application. This may be included as a simple help page in your site

you will need to submit

The following should be included in your uploaded zip file (please ensure that there are no additional files):

* your finalisation report (Task 1)
* your web application documentation (Task 2)