

2024/25 Module Handbook

Web Application Technologies

School of Built Environment,
Engineering and Computing

Level 5

Semester A

(20 Credits)

CRN 14104



LEEDS BECKETT UNIVERSITY

Contents

1	What this Module is About.....	3
1.1	Introduction from the Module Leader	3
1.2	Module Aims	4
1.3	Module Learning Outcomes	4
1.4	Module Learning Activities.....	4
1.5	Indicative Module Content.....	5
1.6	Graduate Attributes Developed and Assessed	6
1.7	Communication	6
2	Weekly Schedule	7
3	Key Resources to Support Learning.....	8
4	Assessment	9
4.1	Assessment Summary	9
4.1.1	Assessment 1: Phase Test.....	9
4.1.2	Assessment 2: Portfolio	9
4.1.3	Reassessment	9
4.2	Assessment Details.....	9
4.3	Feedback on Your Assessments	10
5	Understanding Your Assessment Responsibilities	10
5.1	Extenuating Circumstances and Mitigation	10
5.2	Late Submission.....	11
5.3	Academic Misconduct	11
6	The End	11

Student Name _____

Email Address _____

Course _____

Group _____

Module Tutor _____

Tutor's Email Address _____

Communication Protocol: *module staff will reply to student questions within a reasonable time but this will normally be within office hours only. Students are advised to check this handbook and also to see if there are any online/noticeboard announcements or FAQ answers that deal with their enquiry before contacting staff.*

1 What this Module is About

1.1 Introduction from the Module Leaders

Welcome to *Web Application Technologies*, or as we will usually call it “WAT”.

This module aims to provide you with an introduction to some of the technologies that are used in developing modern web applications. This is a very broad topic! We have designed the module to give a broad coverage, including both front-end and back-end technologies. The assessment should allow you to explore whichever aspects you find the most interesting.

The module will cover front-end with HTML and CSS. In line with the current trends in industry we will use a CSS toolkit to provide some “ready made” components, and to ease the process of developing modern-looking sites. There are many toolkits we could have chosen; we have selected based on current use and ease of learning in, what is after all, a very short course.

For back-end we will build on your knowledge of Python to use the Django web development framework. Again, this emphasis fits with current industry practice. We will look briefly at the “old days” of web development so that we can see how, and why, things have changed.

Underpinning all this we will try to follow good practice in management of source code, and in managing multiple development projects. We will again use GitHub and hopefully you will be able to continue building up a portfolio of work there. With Python we will make use of *virtual environments*, which will allow us to introduce some of the issues surrounding the management of dependencies in complex projects.

At the end of the module we hope you will have an understanding of the tools and processes that are involved in creating a modern web app.

This is the first year that this module has run, following a major updating of the content. This means that we are very open to your feedback! We also hope that, in return, you will be tolerant if things seems a little less than organised.

The rest of this handbook contains useful information on how the module will run. Have a read and bring any questions to your first practical session.

Cheers!

Patrick and Tony

Module Leaders: Patrick Ingham and Tony Jenkins

Email: p.ingham@leedsbeckett.ac.uk a.m.jenkins@leedsbeckett.ac.uk

1.2 Module Aims

The ability for Internet-based services to interact with data stores and exchange data is essential for developing many web-based application systems. This module aims to equip students with the basic grounding to design and build such systems.

1.3 Module Learning Outcomes

By studying this module, the learners are expected to:

- LO1: Apply appropriate principles, knowledge, and techniques to the design of a web-based system.
- LO2: Identify and apply appropriate algorithms and data structures within a web scripting language.
- LO3: Develop secure, data-oriented, web application systems.

The assessments provide the opportunity for students to demonstrate that they have attained these outcomes.

1.4 Module Learning Activities

The topics in this module form two broad groups. First, there are the concepts and knowledge that underpin modern web development; these can be learned and appreciated by studying articles and even books. Second, the technologies and their use, which must be learned by doing. There is therefore a range of learning activities in the module:

Lectures [Roughly 1 Hour per Week]

Lectures will be recorded and made available via MyBeckett. The length will vary depending on the topic to be covered, and there may be more than one lecture in any given week. Lectures will be made available at the start of each week, and should be viewed before the practical sessions at the end of the week.

Practical [2 Hours per Week]

The weekly practicals will provide an opportunity to explore the weekly topics as a group and independently. They also provide a time to ask questions, or for more general support. Most practicals will start with some demonstration, but all will assume that the lecture videos, and reading, are familiar.

Self-Study [Minimum 4 Hours per Week]

There are many resources available to support learning these topics. A set of links will be built and maintained on MyBeckett. Sources include, but are not limited to:

- Real Python (<https://realpython.com/>) for tutorials on advanced topics in Python.
- The official Git documentation at <https://git-scm.com/doc>.
- The Mozilla Web Docs (for example <https://developer.mozilla.org/en-US/docs/Web/HTML>) for HTML and CSS.

- The docs for the Bulma CSS framework at <https://bulma.io/>.
- The official Django documentation (<https://docs.djangoproject.com/en/5.1/>), and especially the associated tutorials.

In addition, we have created a module “book” which can be found at:

<https://www.tony-jenkins.org.uk/watbook/>

This contains some activities that will be used in the practicals.

Note: This module book is “under development”, so at the start of the module some of the later chapters of the book may be incomplete. Be sure to check back or reload to make sure you have the current version.

As always, there are many tutorials available as web pages and videos on YouTube. You will be directed to key resources and/or recommendations, but you should not feel limited by this. Part of the skillset of a web developer is the ability to seek out this sort of material, and to learn independently.

1.5 Indicative Module Content

The aim in the module is to introduce some of the tools and techniques used in the development of a modern web application. Where more than one tool exists, exemplars based on current use in industry have been chosen. Coverage is broad, so that by the end of the module students will have had experience of both front-end and back-end technologies.

The module will cover HTML and CSS for static front-end development. It will emphasise the use of a utility CSS framework rather than “straight” CSS. This is partly to allow more complete results to be achieved in the short time available, but also reflects trends in industry.

For back-end, an MVC framework will be used. Again, this reflects the move away from simply embedding scripts in HTML towards a toolkit-based approach. It will also allow more complete applications to be developed, without the need to reinvent components that are now usually provided.

The specific frameworks used in the current version of the module will be the Bulma CSS Framework, and the Django Web Framework. All students have previous experience of Python, which drives the framework choice.

Software engineering principles, including source code management, will also be developed further. This will include the management of dependencies.

No specific data store technology will be introduced, as this is usually irrelevant with a modern framework. SQLite will be used with Django in demonstrations, but students may prefer to use MySQL or similar.

All the technologies are examples of current industry use. It is expected that students would be able to transfer their knowledge to other, similar, technologies.

1.6 Graduate Attributes Developed and Assessed

Graduate attributes provide a language for students to articulate their skills and strengths, and a basis to self-assess their personal development. The graduate attributes in this module are developed and assessed:

<i>Attribute</i>	<i>How Developed</i>	<i>How Assessed</i>
Enterprise	Consideration of security in the development of a website.	Justification of decisions.
Digital Literacy	Application of a range of digital tools in the development of a website.	Quality of website developed, and organisation of underpinning code.
Global Outlook	Students are developing skills that are globally recognised and which meet a significant skills shortage in industry.	Awareness of the bigger picture in systems development.

1.7 Communication

The main formal means of communication will be via MyBeckett and your University student email account. You should ensure that you check your University email account every other day. Staff will not respond to emails sent from any other email account. Not having read an email will not be an excuse for anything.

A weekly “bulletin” email will be sent on Mondays with pointers for the week’s work and any important announcements.

For short questions you are encouraged to use the “Chat” feature of Microsoft Teams rather than email. This is much more efficient, and you will get an answer sooner. If asking about a program error be sure to provide the offending code as well as the error message. Remember that debugging program code that uses a framework is complex, and may have to wait until a practical session.

The Module Team aim to reply to emails within **48 working hours**. Response to queries sent via Teams is likely to be faster.

To avoid duplication, we ask that you send ALL queries to BOTH Module Leaders.

If you feel you are falling behind it is important to contact the Module Team as soon as possible. In this module in particular it is very difficult to catch up if you do fall behind. You must notify your Course Administrator if you are absent for more than one day (for example, for an interview, because of emergency unforeseen circumstances, or for compassionate leave). If you are going to apply for mitigation you will need to provide written evidence of the reason for your absence.

2 “Weekly” Schedule

The general plan is shown below. Like all good plans it is subject to change. This plan is indicative, and does not necessarily correspond to weekly events. It is in the nature of this module that some students will progress faster than others; this is fine as long as the end point is the same for all.

	Topic	Overview	Activities
1	<i>Introduction and Refresher</i>	Web protocols; Python packages; Managing dependencies.	Using Postman for HTTP; Creating Python environments; Managing source code.
2	<i>Web Basics</i>	APIs; Programming with an API; HTTP verbs; JSON.	Using APIs in Python; Sending queries; Receiving results; Packing and unpacking JSON.
3	<i>HTML</i>	HTML concepts; Page structure; The DOM; HTML tags.	Creating basic HTML pages; Building a static site; Using tags; Encoding document structure.
4	<i>CSS</i>	CSS concepts; Adding CSS to a web page; Selectors; Page layouts; Responsive design.	Adding static CSS to HTML pages; Ensuring consistency; Design elements.
5	<i>CSS Toolkits</i>	Adding a toolkit; Motivation; Benefits; Utility toolkits; Examples.	Migrating CSS to Bulma toolkit; Using CSS for page layout; Consistency; Customising Bulma.
6	<i>Consolidation (Reading) Week</i>	Web design; Good UX and UI; Accessibility guidelines and issues.	Reading (to be provided).
7	<i>Web Frameworks and MVC</i>	MVC concepts; History; Overview of web frameworks.	Creating a Django project; Django admin; Managing a Django project code.
8	<i>Django Basics</i>	Django concepts; Project structure; MVT.	Adding endpoints to a Django project; displaying results; Data stores.
9	<i>Django Models</i>	Modifying Django admin; Model features; Relationships; ORM.	Defining a complex model; Managing admin; Seeding a database; Changing data store.
10	<i>A Django Site</i>	Templates; Inheritance within Templates; Errors.	Displaying data via endpoints; Using Bulma in templates; Adding 404 pages.
11	<i>Summary and Review</i>	A complete Django site.	Assignment work.

3 Key Resources to Support Learning

Resources for the module will be linked from MyBeckett. As noted above, students are encouraged to research and select from the plethora of resources available online.

There are books available relating to Django but they suffer from the fast-moving nature of the subject. By the time a book has been written and published, frameworks like Django will have moved on several versions! For this reason, it is better to rely on online material.

An electronic book is maintained specifically for the module here:

<http://www.tony-jenkins.org.uk/watbook/>

This “book” follows the pattern of the module reasonably closely. A PDF version of the book can be found linked from the first section of the web version.

Students are encouraged to work on their own machines if possible, so that they can also learn about setting up projects. Some aspects of project development are not possible on lab machines where the user does not have appropriate access permissions.

Those wishing to work on their own machine will need to download and install Python. This is free and can be found via www.python.org. Users of Windows can also install Python via the “Microsoft Store”. Users of Linux will probably find that it just works, and is already installed. The module will use Python 3 (at the time of writing the current version is 3.12, but anything after 3.10 will be fine). Students should be very sure they have installed Python 3, as some of the module content will not work under Python 2.

Django is an easy install in a Python virtual environment. Instructions for installing other packages can be found in their own documentation.

Students are free to use any IDE they want. We recommend JetBrains PyCharm, which is available free of charge under a student licence. See:

<https://www.jetbrains.com/community/education/#students>

When suitably configured, PyCharm offers several features that ease navigation around the code that makes up a Django project.

JetBrains also offer WebStorm, intended for JavaScript, but also useful for developing with HTML and CSS. It offers a slightly less complex interface than PyCharm, which some might find preferable. It is available free via a student licence.

Many alternatives exist. Microsoft Visual Studio Code is a popular alternative and is equally free. It also has support for Django.

All the code developed should run on Mac, Linux, and even Windows.

Students are strongly encouraged to keep their own code under version control, using Git with a remote repository. This will make moving work between home and campus seamless. GitHub is recommended, but GitLab or BitBucket are also free to some extent. The University also provides a Git facility, but this is available only on campus, or via a VPN. GitHub will be used for the assignment work.

Any students requiring additional support or alternative arrangements must declare this and provide evidence of their disability to the Disability Advice Team as early as possible. If there are any delays with the Disability Advice Team, talk to the Module Leaders.

4 Assessment

4.1 Assessment Summary

There are two assessments. An online test carries 40% of the marks, with the remaining 60% coming from a web development project. The project requires the submission of source code via GitHub, and a short video of a demonstration of the final site.

Detailed specifications are in the Assessment folder on MyBeckett. Basic details are below.

4.1.1 Assessment 1: Phase Test

Assessment Method:	Online Test	Re-assessment Method:	As original.
Length	75 Minutes	Length	As original.
Assessment Date	During Week 9	Re-assessment Date and Time:	April 2025, precise date to be confirmed.
Feedback Method:	Immediate, via VLE.	Feedback Method:	As original.
Feedback Date:	Immediate.	Feedback Date:	Immediate.

4.1.2 Assessment 2: Programming Project

Assessment Method:	Repository and Video	Re-assessment Method:	As original.
Assessment Date	Submission January 2025.	Re-assessment Date and Time:	April 2025, precise date to be confirmed.
Feedback Method:	Feedback and marking grid via VLE.	Feedback Method:	Feedback and marking grid via VLE.
Feedback Date:	3 working weeks	Feedback Date:	Results release.

4.1.3 Reassessment

The usual rules for reassessment apply. Details are in the assignment specs. You will be given feedback on the original submission, along with a marking grid that clearly shows where improvements can be made.

4.2 Assessment Details

The assessments are provided in the “Assessment” folder on the VLE. The version found there should be considered definitive. To avoid confusion, it is not reproduced here.

4.3 Feedback on Your Assessments

Feedback forms a large part of your learning experience and is vital to your personal and professional development.

You will get written feedback on the development project, which will be sent to you via MyBeckett. Formative feedback will also be available before submission, when you will be able to ask for informal feedback on your work during practical sessions.

For obvious reasons you will not be able to see the correct answers to the test, but if any questions (or answers) trouble you, please ask your tutor.

Feedback on how well you are doing in your studies is important to your learning. The methods of feedback used in this module are:

- **Formative feedback** during online sessions, by email and by Teams: when you ask a question and receive an answer or when your tutor comments upon your work or drafts of submissions. Obviously we have to limit the number of times we can offer feedback outside of timetabled sessions.
- **Summative feedback**: written feedback with the returned assessments and verbal feedback providing general comments about your work or the work of the class.

You are encouraged to provide feedback about the module to the module and course team. This is especially important as this is the first time this module has been offered in its new form.

Regular reviews will be included in the module. These are opportunities to resolve module issues promptly before they become too significant. In addition, you will have the opportunity to complete a module evaluation at the end of your module. These comments will be reviewed by the whole course team and may be considered at course planning meetings.

5 Understanding Your Assessment Responsibilities

5.1 Extenuating Circumstances and Mitigation

If you are experiencing problems which are adversely affecting your ability to study (called “extenuating circumstances”), then you can apply for mitigation. You can find full details of how to apply for mitigation at: www.leedsbeckett.ac.uk/studenthub/mitigation.htm.

This is the only way to get an extension. If you email the Module Leaders to ask for an extension you will just be told to apply as above, and this will obviously also introduce a delay.

The University operates a fit to sit/fit to submit approach to extenuating circumstances which means students who take their assessment are declaring themselves fit to do so. More information is available at the above link and here: <http://www.leedsbeckett.ac.uk/studenthub/examinations/>

5.2 Late Submission

Without any form of extenuating circumstances, standard penalties apply for late submission of assessed work. Full details of the penalties for late submission of coursework are available at www.leedsbeckett.ac.uk/public-information/academic-regulations.

If you anticipate that you will submit work late, it is *always* best to discuss this with the module leader beforehand.

5.3 Academic Misconduct

Academic misconduct occurs when you yourself have not done the work that you submit. It may include cheating, plagiarism, self-plagiarism, collusion and other forms of unfair practice. What is and what is not permitted is clearly explained in *Factsheet for Students* which is available to view at: www.leedsbeckett.ac.uk/studenthub/academic-integrity.

There are a range of resources available to help you understand what is and what is not permitted and how to use other people's ideas in your assessed work. These include the Skills for Learning website which can be found at <http://skillsforlearning.leedsbeckett.ac.uk>

The serious consequences of plagiarism and other types of unfair practice are detailed in the Academic Regulations at www.leedsbeckett.ac.uk/public-information/academic-regulations.

If you find that you are tempted to cheat, for example by submitting programs written by another, you are strongly encouraged not to, and to instead discuss how you might produce your own work. This tends to lead to a better outcome for everyone.

AI tools are very useful in developing programs, and it is important to gain experience in their correct use. But submitting complete programs generated by AI will be considered to be serious academic misconduct. If you are tempted to use AI in this way, please seek advice and support instead.

6 The End

Hopefully, this document has explained all you need to know about the module.

If not, do not be afraid to get in touch.

Enjoy the module!

This version: 23/08/2024 03:48