

Advanced Web Development

Module description

This module covers server-side web development and the related internet protocols and standards; HTTP, REST and web sockets. The course focus on applied server-side programming in Python and makes extensive use of the Django web framework. Students will develop a range of skills for building and understanding web server programming. The module covers HTTP and HTML, CSS and JavaScript programming, the Django web framework, security and common issues in deploying web sites. This will allow students to develop skills across a range of web programming topics.

Module goals and outcomes

The module introduces a variety of topics around contemporary web server programming. Each topic explores both the theory and practice behind building different parts of a web server. As part of the work, you will produce two coursework assignments, which focus on using the skills that have been learnt to develop web servers.

Upon successful completion of this module, you will be able to explain the HTTP protocol, and how to build a web server using Python. You will be familiar with, and able to explain, protocols on the web, such as web sockets and REST, and how to use a variety of web APIs. You will be able to explain and implement security features in a modern web framework.

Textbook and readings

Specific readings for each topic from the following list are included in the Reading pages in each topic.

[Django](#)
[Django documentation](#)
[Django REST framework](#)
[Django Channels](#)
[IETF Documents](#)
[Apache HTTP Server Project](#)

Hall, Eric [*Internet Core Protocols: the Definitive Guide*](#), O'Reilly Media, Incorporated, 2000.

Harrington, J.L. [Relational Database Design and Implementation](#), Elsevier Science & Technology, 2016.

Kumar, Akshi [Web Technology : Theory and Practice](#), CRC Press LLC, 2018.

Module outline

The module consists of ten topics that focus on key areas of the fundamentals of computer science.

Topic 1. Learning outcomes:

After completing the study of this topic you should be able to:

- write a basic Django application that displays a simple message in a web browser
- install and configure Django
- explain the main components of a full stack webserver.

Key concepts:

- web stacks
- Model View Controller application design.

Topic 2. Learning outcomes:

After completing the study of this topic you should be able to:

- create and alter databases using migrations
- describe how the Django ORM maps function calls to SQL queries
- create simple relational databases to model a variety of data

Key concepts:

- SQL and relational databases
- ORM.

Topic 3. Learning outcomes:

After completing the study of this topic you should be able to:

- lay out simple webpages
- use Django forms and validators to accept user input
- use a templating language to generate dynamic web pages.

Key concepts:

- web framework user interaction and HTML templating languages.

Topic 4. Learning outcomes:

After completing the study of this topic you should be able to:

- use Django REST framework to build a simple REST API
- explain how CRUD relates to REST
- explain REST
- explain CRUD.

Key concepts:

- REST, CRUD and Django REST framework (DSF)
- unit testing.

Topic 5. Learning outcomes:

After completing the study of this topic you should be able to:

- build a REST API from a specification document
- build a single page application
- use JavaScript to use a server side API
- describe AJAX.

Key concepts:

- AJAX and single page applications.

Topic 6. Learning outcomes:

After completing the study of this topic you should be able to:

- describe the web sockets standard
- implement websockets using Django channels
- write code that interacts with a Websockets server
- use a JavaScript framework to build a websockets frontend.

Key concepts:

- Celery and adding Celery to Django
- Django channels.

Topic 7. Learning outcomes:

After completing the study of this topic you should be able to:

- describe different types of API on the internet
- use the command line and JavaScript to interact with different APIs
- describe OpenAPI and how it is used.

Key concepts:

- web APIs
- OpenAPIs.

Topic 8. Learning outcomes:

After completing the study of this topic you should be able to:

- describe a user authentication model for web applications
- implement and use Django User Authentication
- describe a number of common security vulnerabilities in web apps and their solution.

Key concepts:

- web APIs
- Django Authentication.

Topic 9. Learning outcomes:

After completing the study of this topic you should be able to:

- configure a production webserver
- configure a [Django] web application for production deployment
- describe some common deployment automation tools.

Key concepts:

- production deployment
- deployment automation.

Topic 10. Learning outcomes:

After completing the study of this topic you should be able to

- explain how to profile a web application
- explain how to configure a production deployment to be load balanced
- describe common approaches to scalable web application design.

Key concepts:

- profiling a web application.

Activities of this module

The module comprises the following elements:

- Lecture videos will include overviews and discussions of topics as well as practical examples of working with different data types, formats, tools and techniques.
- Practice quizzes will be used to reinforce your learning and understanding.
- Activities drive the work that you do for each topic, where you are asked to solve challenges of different types.
- Graded assignments include a small practical coursework assignment, a large practical coursework assignment and a written exam.
- Discussions with your peers will help to guide your work and encourage you to explore different types of solutions to problems.
- Readings will help to reinforce your learning of concepts. The course has one main text and is supplemented by documentation to help with practical elements.

How to pass this module

The module has two major assessments, each worth a percentage of your grade as follows:

- First coursework: In this project you will be tasked with implementing a small RESTful web server to a given specification. This will make use of skills and knowledge you have learnt in Topics 1 to 5. The submission will comprise the code for a working Django implementation of the specification. This will be worth 50% of your overall grade.
- Second coursework: This will be a self-driven web application project. You will be tasked with building a small social network web site that can host many users, using all the skills you have learnt from Topics 1 to 8. The submission will comprise the code for a working web application that implements a social network. You will also be required to submit a short report explaining your web application. This will be worth 50% of your overall grade.

Activity	Required?	Deadline week	Estimated time per module	% of final grade
Written, staff graded coursework	Yes	11	Approximately 10 hours	50%
Written, staff graded coursework	Yes	20	Approximately 20 hours	50%