CM3035 Advanced Web Development

Course Description

This module covers server-side web development and the related internet protocols and standards; HTTP, REST and web sockets. The course focuses 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.

Course Goals and Objectives

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 course, you will be able to:

1. Explain the HTTP protocol and how to build a web server using Python
2. Understand and explain protocols on the web such as web sockets and REST
3. Use a variety of web APIs
4. 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
- Apache HTTP Server Project
- Internet Core Protocols: the Definitive Guide by Hall, E. (2000): O'Reilly Media, Incorporated
- Relational Database Design and Implementation by Harrington, J.L. (2016): Elsevier Science & Technology
- Web Technology: Theory and Practice by Kumar, A. (2018): Chapman and Hall/CRC

Course Outline

The course consists of ten topics divided into 20 weeks that focus on key concepts.

Topic 1: The Web Stack	**Key concepts:** You will learn about web stacks and the MVC application design. **Learning outcomes:** • 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
Topic 2: Database schemas and ORMs	**Key concepts:** You will learn about SQL and relational databases. You will also learn about ORM. **Learning outcomes:** • 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
Topic 3: Interaction through serving HTML, CSS and JavaScript	**Key concepts:** You will learn about web framework user interaction and HTML templating languages. **Learning outcomes:** • Lay out simple webpages • Use Django forms and validators to accept user input • Use a templating language to generate dynamic web pages

Topic 4: Build a CRUD and RESTful API (part 1)	**Key concepts:** You will learn about REST, CRUD and Django REST framework. You will also learn about unit testing. **Learning outcomes:** • Use Django REST framework to build a simple REST API • Explain how CRUD relates to REST • Explain REST • Explain CRUD
Topic 5: Build a CRUD and RESTful API (part 2)	**Key concepts:** You will learn about AJAX and single page applications. **Learning outcomes:** • Build a REST API from a specification document • Build a single page application • Use JavaScript to use a server-side API • Describe AJAX
Topic 6: Asynchronous Web Services	**Key concepts:** You will learn about Celery and adding Celery to Django. You will also learn about Django channels. **Learning outcomes:** • Describe the web sockets standard • Implement web sockets using Django channels • Write code that interacts with a WebSocket server • Use a JavaScript framework to build a web sockets frontend

Topic 7: Working with external APIs	**Key concepts:** You will learn about web APIs and OpenAPIs. **Learning outcomes:** • 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
Topic 8: User authentication and security	**Key concepts:** You will learn about web APIs and Django Authentication. **Learning outcomes:** • Describe a user authentication model for web applications • Implement and use Django User Authentication • Describe several common security vulnerabilities in web apps and their solution
Topic 9: Deploying a website	**Key concepts:** You will learn about production deployment and deployment automation. **Learning outcomes:** • Configure a production webserver • Configure a [Django] web application for production deployment • Describe some common deployment automation tools

Topic 10: Load balancing, scalability	**Key concepts:** You will learn about profiling a web application. **Learning outcomes:** • 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

Learning Activities of This Course

The course 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 and a large practical coursework assignment.

- **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 Course

The module will contain a range of summative and formative assessments. Summative assessments are assessments which contribute directly towards your final grade. Formative assessments do not count directly towards your final grade. Instead, they provide you with opportunities for low stakes practice and will often provide some sort of feedback about your progress. For example, a practice quiz might provide you with feedback about why a particular answer was wrong.

The course has two major assessments. Both assessments are worth 50% of your grade:

- Coursework 1: this consists of a project which will be assessed halfway through course (approximate deadline in week 13)
- Coursework 2: this also consists of a project which will be assed at the end of the course (approximate deadline in week 22).

This is a detailed breakdown of all the marks:

Activity	**Required? (Summative)**	**Deadline week**	**Estimated time per course**	**% of final grade**
Written, staff graded coursework 1	Yes	Approximately week 13	10 hours	50%
Written, staff graded coursework 2	Yes	Approximately week 22	20 hours	50%