

• [Unit 11](#)

## Unit 11: Web Development in Python

Welcome to Week 11 this unit will tie together several of the subjects covered to date such as database development and Python programming. This unit focuses on developing web applications using a Python web framework called Flask.

### In this unit we shall:

- Introduce the concept of a Model-View-Controller.
- Discuss the key elements of a web application.
- Discuss some of the security considerations of a web development.

### On completion of this unit you will be able to:

- Implement a simple web server.
- Create dynamic templates using Jinja.
- Connect a database to a Flask application.
- Understand some of the security considerations around web applications.

The unit introduces the core concepts of web application development with examples using the Python Flask framework.

**Please note, the topics covered in units 9, 10 & 11 are to expand your knowledge of OOS and are not specifically required for the System Implementation assignment. However, if you would like to demonstrate any of these elements in the assignment you are welcome to do so.**

Your progress

• **Lecturecast** | 1 hrs

- [Web Development and PythonURL](#)

• **Reading** | 6 hrs

- [Unit 11 ReadingPage](#)

• **Formative activities** | 7 hrs

**Codio.** Work through the Codio module called '[Introduction to Flask](#)'.

**Remember** to record your results, ideas and feedback in your e-portfolio.

- **Seminar 6** | 2 hr

**Title: e-Portfolio Submission**

Join the [Seminar Session](#) this week. Remember to bring the seminar questions and your attempted answers.

Visit the 'Seminar 6 preparation' below to complete the activities prior to the seminar session. As we will discuss these concepts in the seminar session, ensure you have completed the activities before this week's Seminar session.

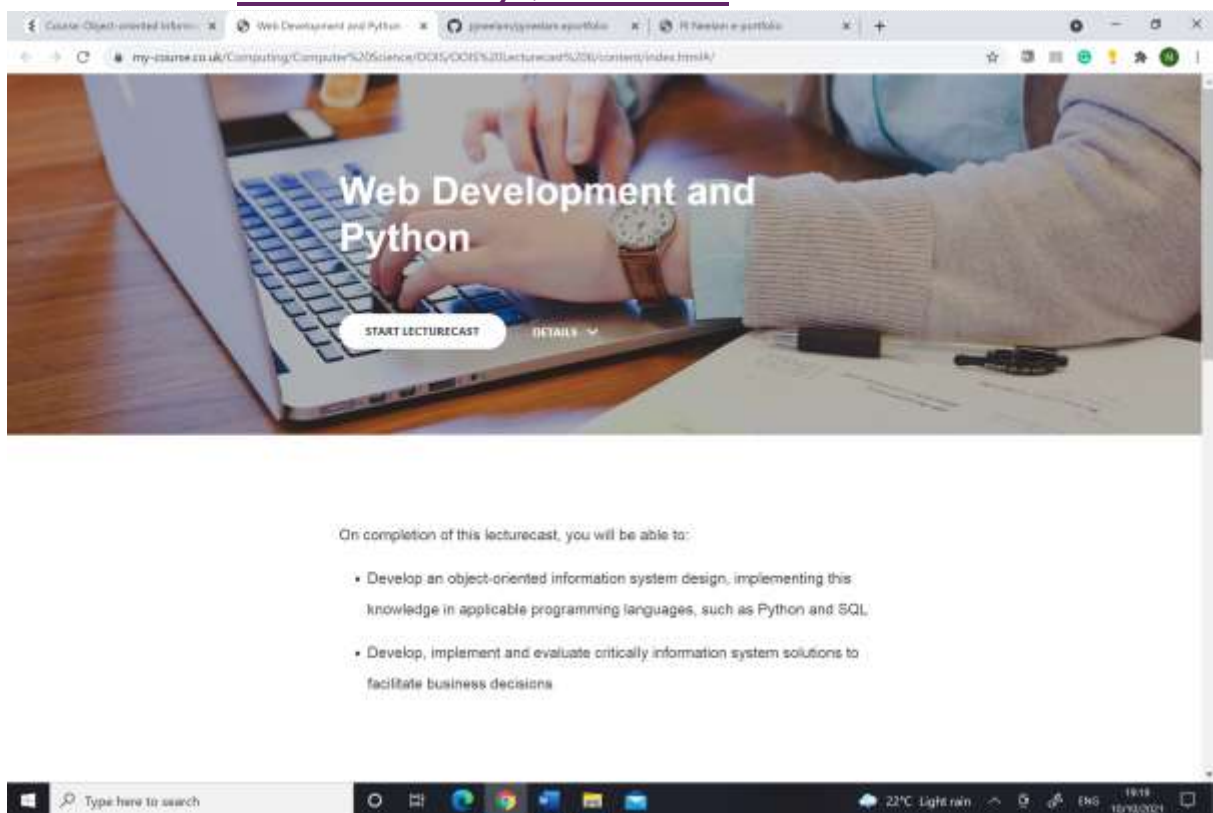
- [Seminar 6 preparationPage](#)

- **Summative assessment** | 8 hrs

- [System ImplementationAssignment](#)

- **Research Study Survey** | 10 mins

- [e-Portfolio SurveyQuestionnaire](#)



The screenshot shows a web browser window with multiple tabs. The active tab is titled 'my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lectur...'. The main content area features a large image of hands typing on a laptop keyboard, with the text 'Web Development and Python' overlaid. Below the image are two buttons: 'START LECTURECAST' and 'DETAILS'. Further down, there is a section titled 'On completion of this lecturecast, you will be able to:' followed by a bulleted list of learning outcomes.

On completion of this lecturecast, you will be able to:

- Develop an object-oriented information system design, implementing this knowledge in applicable programming languages, such as Python and SQL.
- Develop, implement and evaluate critically information system solutions to facilitate business decisions



Course Object-oriented Inform... Web Development and Python... my-course.co.uk/Computing/Computer%20Science/OOS/OOS%20Lectures/2020/contents/index.html#sessions/4QwN67A/R8tjwV0\_UCGa18FGWcOL...

## Web Development and Python

11% COMPLETE

- What is Web Development?
- Web Development
- Model-View-Controller (MVC)**
- Flask
- Routes
- A Simple Flask Application
- Templates
- Malicious Code Injection

### Lesson 2 of 18

## Model-View-Controller (MVC)

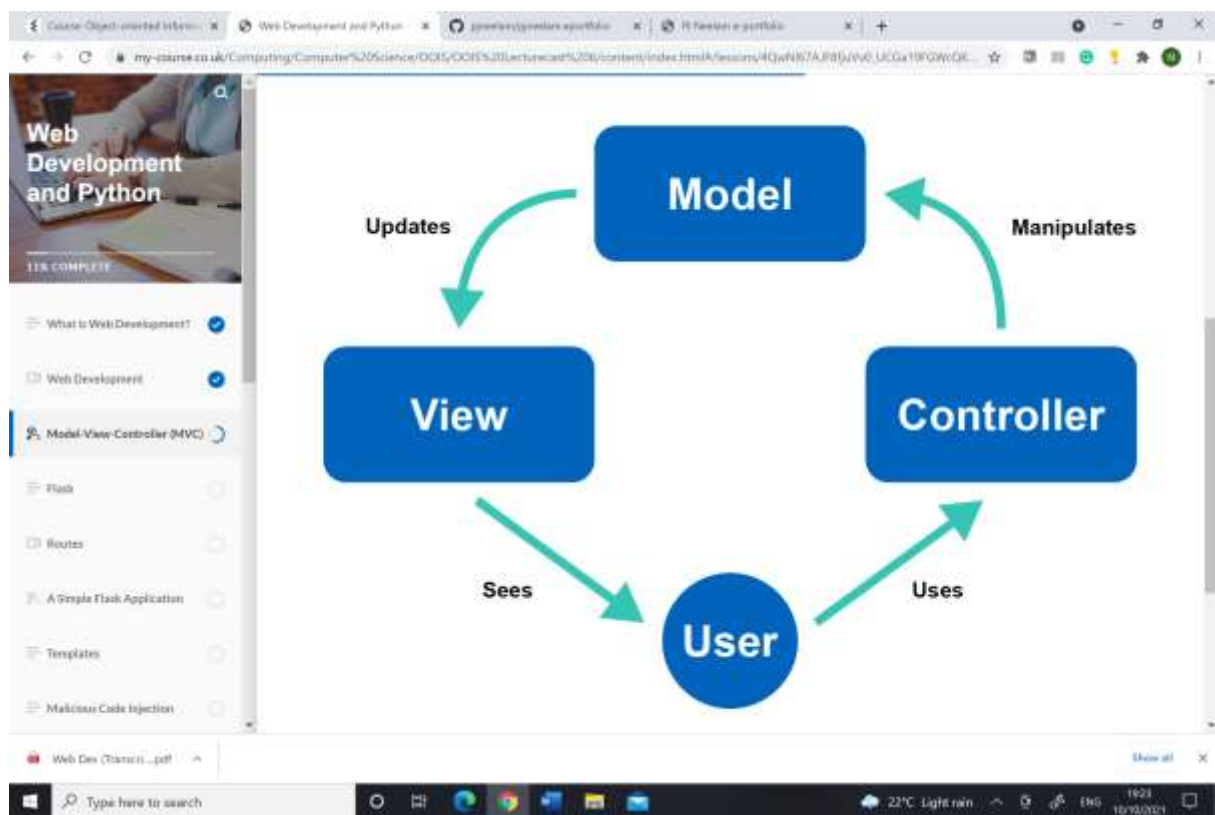
Before we move onto specifics and the implementation side of things, we are going to look at a software design pattern that is commonly used for developing user interfaces but has become increasingly popular for web applications.

It splits out the related software logic into interconnected elements. This is done to separate the internal representation from the way it is presented to the user, it essentially provides independence between the way our information is stored and the way it is displayed.

Web Dev (Transcript) ...pdf

Type here to search

22°C Light rain 16:23 10/10/2021



Course Object-oriented Information Systems

Web Development and Python

python/python-exercises

Python exercises

my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lectures/003/contents/index.html#section/4QwN67AJR8tj/vv0\_UCGa19FGWwOE...

Web Development and Python

11% COMPLETE

What is Web Development?

Web Development

Model-View-Controller (MVC)

Flask

Routes

A Simple Flask Application

Templates

Malicious Code Injection

View

Controller

Sees

Uses

User

This is known as the model-view-controller or MVC and many popular languages such as JavaScript, Python, Java, PHP and C# have MVC frameworks right out of the box.

CONTINUE

Web Dev (Transcript) ...pdf

Type here to search

22°C Light rain

ENG

19:21

19/10/2021

Course Object-oriented Information Systems

Web Development and Python

python/python-exercises

Python exercises

my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lectures/003/contents/index.html#section/4QwN67AJR8tj/vv0\_UCGa19FGWwOE...

Web Development and Python

17% COMPLETE

What is Web Development?

Web Development

Model-View-Controller (MVC)

Flask

Routes

A Simple Flask Application

Templates

Malicious Code Injection

Let's take a look at the different elements of the MVC and see what they do and how they interact.

Click the 'start' button below, then click the arrow buttons on the left and right to navigate.

START >

https://www.my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lectures/003/contents/index.html#section/4QwN67AJR8tj/vv0\_UCGa19FGWwOE...

Web Dev (Transcript) ...pdf

Type here to search

22°C Light rain

ENG

19:21

19/10/2021



Course Object-oriented Information Systems - Web Development and Python - my-course.co.uk/Computing/Computer%20Science/OOS/OOS%20Lectures/OOS%20Lectures/index.html#lessons/4QwN67AJR8tj/vU0\_UCGx19FGWvOE...

## Web Development and Python

17% COMPLETE

- What is Web Development?
- Web Development
- Model-View-Controller (MVC)**
- Flask
- Routes
- A Simple Flask Application
- Templates
- Malicious Code Injection

```
graph TD; Model[Model] -- Updates --> View[View]; Controller[Controller] -- Manipulates --> Model; View -- Sees --> User((User)); User -- Uses --> Controller; Controller -- Uses --> Model;
```

First of all, we have the model, which can be thought of as the central component of the pattern. The data model represents the core information that your application is being used to access and manipulate. We use the model to directly manage the data itself, any associated logic and the rules of the application. The key thing to remember at this point is that the model is independent and is not reliant on anything else within the system.

The golden rule of an MVC is that the model represents the data and does nothing else, it is entirely independent of the controller or the view.

Web Dev (Transcript) ...pdf

Type here to search

22°C Light rain 19:24 18/10/2021

Course Object-oriented Information Systems - Web Development and Python - my-course.co.uk/Computing/Computer%20Science/OOS/OOS%20Lectures/OOS%20Lectures/index.html#lessons/4QwN67AJR8tj/vU0\_UCGx19FGWvOE...

## Web Development and Python

17% COMPLETE

- What is Web Development?
- Web Development
- Model-View-Controller (MVC)**
- Flask
- Routes
- A Simple Flask Application
- Templates
- Malicious Code Injection

```
graph TD; Model[Model] -- Updates --> View[View]; Controller[Controller] -- Manipulates --> Model; View -- Sees --> User((User)); User -- Uses --> Controller; Controller -- Uses --> Model;
```

The view displays the data contained within the model, this is essentially any representation of the information like a chart or a table. This can be used to provide multiple views of the same information. For example, you might wish to display accounts information as a table or a chart depending on the intended end user.

Web Dev (Transcript) ...pdf

Type here to search

22°C Light rain 19:25 18/10/2021



Course Object-oriented Inform... Web Development and Python gprelans/gprelans portfolio R Notebook portfolio

my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lecturecast%206/content/index.html#lessons/13h-WW4hu7pHobmydL5y372RUPB8

## Web Development and Python

22% COMPLETE

- What is Web Development? ✓
- Web Development ✓
- Model-View-Controller (MVC) ✓
- Flask** ✓
- Routes
- A Simple Flask Application
- Templates
- Malicious Code Injection

Web Dev (Transcript).pdf

Type here to search

22°C Light rain 19:27 10/10/2021

The remainder of this lecturecast is going to focus on working with Flask. Flask is a micro web framework written in Python.

### What do we mean by a micro web framework?

This means that Flask does not require particular tools or libraries, it has no database abstraction layer, form validation or any other components where pre-existing third-party libraries provide common functions.

We are going to look at common web development concepts in practice with Flask, how Flask works, and how to implement a web application using an MVC-style approach.

Lesson 5 - Routes

Course Object-oriented Inform... Web Development and Python gprelans/gprelans portfolio R Notebook portfolio

my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lecturecast%206/content/index.html#lessons/13h-WW4hu7pHobmydL5y372RUPB8

## Web Development and Python

22% COMPLETE

- What is Web Development? ✓
- Web Development ✓
- Model-View-Controller (MVC) ✓
- Flask ✓
- Routes**
- A Simple Flask Application
- Templates
- Malicious Code Injection

Web Dev (Transcript).pdf

Type here to search

22°C Light rain 19:27 10/10/2021

## Routes

Lesson 5 of 18

Routes



Course Object-oriented Intro Web Development and Python Routes (Transcript.pdf) jperkins@perkinspython.com R Notebook - portfolio

my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lecturecast%206/content/index.html#lessons/Wf5bHfCsGwJ\_uPhelDG1WIC7hooxig

# Web Development and Python

28% COMPLETE

- What is Web Development?
- Web Development
- Model-View-Controller (MVC)
- Flask
- Routes
- A Simple Flask Application**
- Templates
- Malicious Code Injection

## Lesson 6 of 18

# A Simple Flask Application

Let's take a look at a really simple example of a basic Flask example to help underline some of the key concepts we are going to be working with throughout the rest of the lecturecast.

Click the buttons on the image below to learn more.

Routes (Transcript.pdf) Web Dev (Transcript.pdf) Show all

Type here to search

22°C Light rain 19:21 10/10/2021

Course Object-oriented Intro Web Development and Python Routes (Transcript.pdf) jperkins@perkinspython.com R Notebook - portfolio

my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lecturecast%206/content/index.html#lessons/Wf5bHfCsGwJ\_uPhelDG1WIC7hooxig

# Web Development and Python

28% COMPLETE

- What is Web Development?
- Web Development
- Model-View-Controller (MVC)
- Flask
- Routes
- A Simple Flask Application**
- Templates
- Malicious Code Injection

Click the buttons on the image below to learn more.

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
@app.route('/index')

def index():
    return "Hello, World!"

if __name__ == '__main__':
    app.run()
```

1

The first thing to note is the `__name__` value that is passed to the Flask class, note that name is preceded and followed by TWO underscores. This is a predefined variable in Python that is used to set the name of the module in which it is being used. This is the first place that Flask will look when it needs to load any of the associated resources such as template files, which we will cover in more detail later.

Routes (Transcript.pdf) Web Dev (Transcript.pdf) Show all

Type here to search

22°C Light rain 19:22 10/10/2021

my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lectures/2020/contents/index.html#lessons/W15aHfCsGwJ\_uJhbf0C-1WIC9soxig

Web Development and Python

28% COMPLETE

- What is Web Development?
- Web Development
- Model-View-Controller (MVC)
- Flask
- Routes
- A Simple Flask Application
- Templates
- Malicious Code Injection

Click the buttons on the image below to learn more.

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
@app.route('/index')

def index():
    return "Hello, World!"

if __name__ == '__main__':
    app.run()
```

2

The application then imports the routes module, and as seen previously routes are URLs that the application can implement. Flask uses Python functions for request handlers, which are called view functions. A view function can be mapped to one or more route URLs so that Flask knows what logic to execute whenever the client requests a specific URL.

The example here defines the index function and associates it with two routes, the root directory or slash index.

Routes (Transcript).pdf Web Dev (Transcript).pdf

Type here to search

22°C Light rain 16:32 18/03/2021

my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lectures/2020/contents/index.html#lessons/W15aHfCsGwJ\_uJhbf0C-1WIC9soxig

Web Development and Python

28% COMPLETE

- What is Web Development?
- Web Development
- Model-View-Controller (MVC)
- Flask
- Routes
- A Simple Flask Application
- Templates
- Malicious Code Injection

```
@app.route('/index')

def index():
    return "Hello, World!"

if __name__ == '__main__':
    app.run()
```

This is a predefined variable in Python that is used to set the name of the module in which it is being used. This is the first place that Flask will look when it needs to load any of the associated resources such as template files, which we will cover in more detail later.

The example function is again a really simple function that gives a familiar string as a greeting. The definition of the function uses decorators above it, which are used to define that association between the URL given as an argument and the function itself.

CONTINUE

Routes (Transcript).pdf Web Dev (Transcript).pdf

Type here to search

22°C Light rain 16:33 18/03/2021

my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lectures/2020/contents/index.html/lessons/WL5aHCSaGaj\_uJbhefDCr1WICbsoxig

## Web Development and Python

22% COMPLETE

- What is Web Development?
- Web Development
- Model-View-Controller (MVC)
- Flask
- Routes
- A Simple Flask Application
- Templates
- Malicious Code Injection

The example function is again a really simple function that gives a familiar string as a greeting. The definition of the function uses decorators above it, which are used to define that association between the URL given as an argument and the function itself.

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
@app.route('/hello')
def index():
    return "Hello, World!"

if __name__ == '__main__':
    app.run()
```

127.0.0.1:5000

Routes (Transcript).pdf Web Dev (Transcript).pdf

Type here to search

22°C Light rain 16:34 18/03/2021

my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lectures/2020/contents/index.html/lessons/WL5aHCSaGaj\_uJbhefDCr1WICbsoxig

## Web Development and Python

22% COMPLETE

- What is Web Development?
- Web Development
- Model-View-Controller (MVC)
- Flask
- Routes
- A Simple Flask Application
- Templates
- Malicious Code Injection

```
@app.route('/')
@app.route('/hello')
def index():
    return "Hello, World!"

if __name__ == '__main__':
    app.run()
```

127.0.0.1:5000

If you were to create a Flask project and enter the previous code to create your server, you have a functioning Flask web application. All you need to do is run the file and use a web browser to navigate to 127.0.0.1:5000, which is the default address that Flask associates with its applications.

Lesson 7 - Templates

Routes (Transcript).pdf Web Dev (Transcript).pdf

Type here to search

22°C Light rain 16:34 18/03/2021

Course Object-oriented Intro

Web Development and Python

Routes (Transcript).pdf

python/flask/pythonapp/Portfolio

Flask in a portfolio

my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lecturecard%206/content/index.html#lessons/15/2x81ydxw665WUjxTogymf3x4GdaD

Web Development and Python

22% COMPLETE

What is Web Development?

Web Development

Model-View-Controller (MVC)

Flask

Routes

A Simple Flask Application

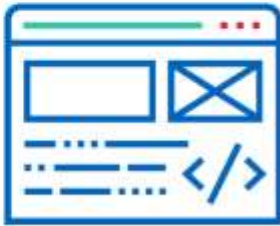
Templates

Malicious Code Injection

Lesson 7 of 18

Templates

When we have talked about the MVC approach to systems development, we have discussed the separation of the underlying data (the model) and the way it is presented to the user (the view).



Templates in Python Flask offer a way of separating the business logic from the way that it is presented on a web page – they are the means used to provide different views on the data.

Routes (Transcript).pdf

Web Dev (Transcript).pdf

Show all

Type here to search

22°C Light rain 19:33 18/10/2021

Course Object-oriented Intro

Web Development and Python

Routes (Transcript).pdf

python/flask/pythonapp/Portfolio

Flask in a portfolio

my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lecturecard%206/content/index.html#lessons/15/2x81ydxw665WUjxTogymf3x4GdaD

Web Development and Python

22% COMPLETE

What is Web Development?

Web Development

Model-View-Controller (MVC)

Flask

Routes

A Simple Flask Application

Templates

Malicious Code Injection

While it might not always be necessary to use a template, for a very small application it is good practice to begin to use them where possible as it is good development practice.

CONTINUE

Routes (Transcript).pdf

Web Dev (Transcript).pdf

Show all

Type here to search

22°C Light rain 19:33 18/10/2021



Course Object-oriented Intro Web Development and Python Routes (Transcript).pdf jperkins@perkinspython.com R Notebook - perkins

my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lectures/005/005%20Lecture%2005/contents/index.html#section/1523d81ydcw665WUjeTggym613d4GdaD


## Web Development and Python

99% COMPLETE

- What is Web Development? ✓
- Web Development ✓
- Model-View-Controller (MVC) ✓
- Flask ✓
- Routes ✓
- A Simple Flask Application ✓
- Templates** ✓
- Malicious Code Injection

Templates are stored in a specific templates directory in a Flask application.

A template contains both static data and placeholders which act as wildcards for dynamic data, we say that a template is rendered with specific data to produce a final document. Flask makes use of the Jinja template library to render its templates into dynamic web content.



In a web application, you make use of templates to render HTML, to display in a user's browser.

Routes (Transcript).pdf Web Dev (Transcript).pdf Show all

Type here to search

22°C Light rain 19:31 10/10/2021


Course Object-oriented Intro Web Development and Python Routes (Transcript).pdf jperkins@perkinspython.com R Notebook - perkins

my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lectures/005/005%20Lecture%2005/contents/index.html#section/1523d81ydcw665WUjeTggym613d4GdaD

## Web Development and Python


99% COMPLETE

- What is Web Development? ✓
- Web Development ✓
- Model-View-Controller (MVC) ✓
- Flask ✓
- Routes ✓
- A Simple Flask Application ✓
- Templates** ✓
- Malicious Code Injection



In a web application, you make use of templates to render HTML, to display in a user's browser.

Before we dive into a template file, we are going to take a quick detour to discuss the importance of user input and why you should never be too trusting of it by looking at code injection vulnerabilities.



Routes (Transcript).pdf Web Dev (Transcript).pdf Show all

Type here to search

22°C Light rain 19:38 10/10/2021



Web Development and Python

4/5 COMPLETE

- What is Web Development?
- Web Development
- Model-View-Controller (MVC)
- Flask
- Routes
- A Simple Flask Application
- Templates
- Malicious Code Injection

Lesson 8 of 18

## Malicious Code Injection

Malicious code injection is the generic name for any flaw created when an application sends untrusted data to the interpreter. So in our case, when looking at web development, this typically means sending untrusted code to the web server, with unexpected results.

The most common forms of this in web applications is SQL injection or cross-site scripting – while we don't need to go into each of these in great detail, we will just cover some high-level concepts.

Routes (Transcript).pdf

Web Dev (Transcript).pdf

Type here to search

22°C Light rain 19:38 18/10/2021

Web Development and Python

4/5 COMPLETE

- What is Web Development?
- Web Development
- Model-View-Controller (MVC)
- Flask
- Routes
- A Simple Flask Application
- Templates
- Malicious Code Injection

The most common forms of this in web applications is SQL injection or cross-site scripting – while we don't need to go into each of these in great detail, we will just cover some high-level concepts.

**The key thing to remember is that these vulnerabilities are only really an issue when dealing with user-provided content and implicitly trusting that our users only have the best intentions!**

Lesson 9 - SQL Injection

Routes (Transcript).pdf

Web Dev (Transcript).pdf

Type here to search

22°C Light rain 19:38 18/10/2021

Course Object-oriented Intro

Web Development and Python

Routes (Transcript).pdf

python3/python3.py

Flask in a nutshell

my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lecturecard%206/contents/index.html/lessons/PyjtdQdMLoXUHT-55WkV7G8gDm5DX

Web Development and Python

4/4 COMPLETE

What is Web Development?

Web Development

Model-View-Controller (MVC)

Flask

Routes

A Simple Flask Application

Templates

Malicious Code Injection

Lesson 8 of 18

SQL Injection

So, what's happening when you actually submit a form on a webpage?

**http://127.0.0.1/basicForm.php?username=username**

Routes (Transcript).pdf

Web Dev (Transcript).pdf

Type here to search

22°C Light rain 16:28 10/10/2021

Course Object-oriented Intro

Web Development and Python

Routes (Transcript).pdf

python3/python3.py

Flask in a nutshell

my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lecturecard%206/contents/index.html/lessons/PyjtdQdMLoXUHT-55WkV7G8gDm5DX

Web Development and Python

4/4 COMPLETE

What is Web Development?

Web Development

Model-View-Controller (MVC)

Flask

Routes

A Simple Flask Application

Templates

Malicious Code Injection

A BASIC HTML FORM - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://127.0.0.1/basicForm

username

username

Login

**http://127.0.0.1/basicForm.php?username=username**

Routes (Transcript).pdf

Web Dev (Transcript).pdf

Type here to search

22°C Light rain 16:28 10/10/2021

The web client will bundle together all of the elements from a web form and send them as part of a query string, which is the bit after your URL, usually following a question mark. In the above example, we have a really simple web form that has one element, called username.

Our web application simply takes the parameters from the web form and uses them to form an SQL query or a command for the database. The database will go away, do some work and then execute the commands or queries as necessary. Finally, the values are returned with the results being displayed on a screen and the database is modified as necessary.

Course Object-oriented Info Web Development and Python Routes (Transcript).pdf jupyter.org/ipython/notebook/ R Notebook - portfolio my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lecturecard%20U/content/index.html/lessons/Pyj4QdMLcLhT-55WkV7G8gDn5DX

## Web Development and Python

4/5 COMPLETE

- What is Web Development? ✓
- Web Development ✓
- Model-View-Controller (MVC) ✓
- Flask ✓
- Routes ✓
- A Simple Flask Application ✓
- Templates ✓
- Malicious Code Injection ✓

The web client will bundle together all of the elements from a web form and send them as part of a query string, which is the bit after your URL, usually following a question mark. In the above example, we have a really simple web form that has one element, called username.

Our web application simply takes the parameters from the web form and uses them to form an SQL query or a command for the database. The database will go away, do some work and then execute the commands or queries as necessary. Finally, the values are returned with the results being displayed as a view and the database is modified in some way.

This in itself is fine but what happens if one of our users tries to enter some SQL code instead of a username?

CONTINUE

Routes (Transcript).pdf Web Dev (Transcript).pdf Show all

Type here to search 22°C Light rain 16:28 10/10/2021

Course Object-oriented Info Web Development and Python Routes (Transcript).pdf jupyter.org/ipython/notebook/ R Notebook - portfolio my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lecturecard%20U/content/index.html/lessons/Pyj4QdMLcLhT-55WkV7G8gDn5DX

## Web Development and Python

4/5 COMPLETE

- What is Web Development? ✓
- Web Development ✓
- Model-View-Controller (MVC) ✓
- Flask ✓
- Routes ✓
- A Simple Flask Application ✓
- Templates ✓
- Malicious Code Injection ✓

**A BASIC HTML FORM - Mozilla Firefox**

File Edit View Go Bookmarks Tools Help

http://127.0.0.1/basicForm

username  
username Login

**SELECT \*  
FROM Users  
WHERE Username = *textbox***

If we think about the previous example, then it is easy to imagine that, behind the scenes, we might have a simple query that looks something like this. We are doing a username search and the value that is entered into the web form is directly included in our SQL query.

CONTINUE

Routes (Transcript).pdf Web Dev (Transcript).pdf Show all

Type here to search 22°C Light rain 16:42 10/10/2021

Course Object-oriented Intro Web Development and Python Routes (Transcript).pdf jperkins@perkinspython.co.uk IT Network e-portfolios


my-course.co.uk/Computing/Computer%20Science/OOS/OOS%20Lecturesad%20U/content/index.html/lessons/PyjdQdMLokUH7-SSWkV7GgDmSDX

# Web Development and Python

100% COMPLETE

- What is Web Development?
- Web Development
- Model-View-Controller (MVC)
- Flask
- Routes
- A Simple Flask Application
- Templates
- Malicious Code Injection

What happens if a user enters this into the box?



**100 or 1=1**  
**FROM users**  
**WHERE Username = *textbox***

If we take this as a statement in SQL or indeed any programming language, then we know it will always return true because 1 should always equal 1. So our query becomes "return all of the usernames where the username is true". This would mean that every record in our database would get returned - something that we don't want to happen!

Lesson 10 - Check Your Knowledge

Routes (Transcript).pdf Web Dev (Transcript).pdf

Type here to search

22°C Light rain 16:42 10/10/2021

Course Object-oriented Intro Web Development and Python Routes (Transcript).pdf jperkins@perkinspython.co.uk IT Network e-portfolios

my-course.co.uk/Computing/Computer%20Science/OOS/OOS%20Lecturesad%20U/content/index.html/lessons/ml\_h2Bg\_NBZ7gD3cg97Ce8AKQ/WH4...

# Web Development and Python

100% COMPLETE

- What is Web Development?
- Web Development
- Model-View-Controller (MVC)
- Flask
- Routes
- A Simple Flask Application
- Templates
- Malicious Code Injection

Lesson 10 of 28

## Check Your Knowledge

START QUIZ >

Routes (Transcript).pdf Web Dev (Transcript).pdf

Type here to search

22°C Light rain 16:42 10/10/2021



Web Development and Python

14% COMPLETE

What is Web Development? ✓

Web Development ✓

Model-View-Controller (MVC) ✓

Flask ✓

Routes ✓

A Simple Flask Application ✓

Templates ✓

Malicious Code Injection ✓

Question

02/02

Which of the following is not a benefit of web development?

☐ Components can be reused

☒ Software runs more quickly

☐ Software can be delivered incrementally

Correct

Web development helps ensure that software runs more quickly

NEXT



Course Object-oriented JS... Web Development and Python... Routes (Transcript).pdf... jprelars/jprelars-portfolio... IT Network e-portfolio... my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lecturecard%206/content/index.html#lessons/1gCAuWt-BUub-TW5q-EEWC4/7Em...


# Web Development and Python

14% COMPLETE

- What is Web Development?
- Web Development
- Model-View-Controller (MVC)
- Flask
- Routes
- A Simple Flask Application
- Templates
- Malicious Code Injection

## Lesson 11 of 18

# Cross-Site Scripting



Routes (Transcript).pdf Web Dev (Transcript).pdf

Type here to search

22°C Light rain 1942 19/10/2021


OOIS\_F00K071 August 2021... Web Development and Python... jprelars/jprelars-portfolio... IT Network e-portfolio... my-course.co.uk/Computing/Computer%20Science/OOIS/OOIS%20Lecturecard%206/content/index.html#lessons/1gCAuWt-BUub-TW5q-EEWC4/7Em...


# Web Development and Python

61% COMPLETE

- Check Your Knowledge
- Cross-Site Scripting
- Mitigations
- Templates
- It's an Example
- Models and Controllers
- Check Your Knowledge
- REFERENCES
- References

Click below to download the transcript for the video.

 Cross-Site Scripting (Transcript).pdf  
135.9 KB

 You can achieve some fairly damaging results with cross-site scripting. You can find out more about this in this week's reading.

Lesson 12 - Mitigations

Cross-Site Scripting...pdf Routes (Transcript).pdf Web Dev (Transcript).pdf

Type here to search

22°C Light rain 2008 19/10/2021



Web Development and Python

67% COMPLETE

- Check Your Knowledge
- Cross-Site Scripting
- Mitigations**
- Templates
- Jinja Example
- Models and Controllers
- Check Your Knowledge
- REFERENCES
- References

## Blacklisting

You can develop your own blacklist, where you maintain a list of disallowed text or patterns.

## Whitelisting

Whitelisting is the inverse of blacklisting, where you create a list of all the acceptable inputs. This is marginally more preferable but far from ideal.

## Escaping Input

Escaping the input is a preferable solution, which means that you essentially strip out any unwanted data, meaning that input is simply characters and not code.

Most modern frameworks, like Flask, will automatically escape input.

Task 1: Cross-Site Scripting (Transtorpt).pdf

Task 2: Routes (Transtorpt).pdf

Task 3: Web Dev (Transtorpt).pdf

22°C Light rain

2007 10/10/2021


Web Development and Python

72% COMPLETE

- Check Your Knowledge
- Cross-Site Scripting
- Mitigations
- Templates**
- Jinja Example
- Models and Controllers
- Check Your Knowledge
- REFERENCES
- References

## Lesson 12 of 18

# Templates



As previously mentioned, Flask uses the Jinja library, which is configured to auto-escape any that is rendered into HTML templates. As we have just covered when talking about malicious code injection this means that it is safe to render user input.

Any characters that a user has entered that could be used to mess with the HTML such as a greater or less than symbol will be escaped. This means they are essentially replaced with safe

Task 1: Cross-Site Scripting (Transtorpt).pdf

Task 2: Routes (Transtorpt).pdf

Task 3: Web Dev (Transtorpt).pdf

22°C Light rain

2007 10/10/2021

Web Development and Python

72% COMPLETE

- Check Your Knowledge
- Cross-Site Scripting
- Mitigations
- Templates
- Jinja Example
- Models and Controllers
- Check Your Knowledge

REFERENCES

References

As previously mentioned, Flask uses the Jinja library, which is configured to auto-escape any that is rendered into HTML templates. As we have just covered when talking about malicious code injection this means that it is safe to render user input.

Any characters that a user has entered that could be used to mess with the HTML such as a greater or less than symbol will be escaped. This means they are essentially replaced with safe values that look the same in the browser without any unwanted effects.

Lesson 14 - Jinja Example

22°C Light rain 10/10/2021

Web Development and Python

72% COMPLETE

- Check Your Knowledge
- Cross-Site Scripting
- Mitigations
- Templates
- Jinja Example
- Models and Controllers
- Check Your Knowledge

REFERENCES

References

Lesson 14 of 18

## Jinja Example

Click the 'start' button below, then click the arrow buttons on the left and right to navigate.

START

22°C Light rain 10/10/2021

Web Development and Python

79% COMPLETE

- Check Your Knowledge
- Cross-Site Scripting
- Mitigations
- Templates
- Jinja Example**
- Models and Controllers
- Check Your Knowledge

REFERENCES

References

# 1

```
<html>
<head>
  {% if title %}
  <title>{{ title }} Page</title>
  {% else %}
  <title>Welcome to the Page</title>
  {% endif %}
</head>
<body>

<nav>
<form action="/search/" method="GET"><input type="text" name="s" />
<input type="submit" value="Search" /></form>
</nav>
<hr>
</body>
</html>
```

The code shown above provides a simple example of a Jinja template.

If you have familiarity with HTML, this should look relatively familiar to you. However, Jinja also looks and behaves like Python with the addition of special delimiters to highlight the jinja syntax over the static data in the template.

1 2 3

Cross-Site Scripting... pdf Routes (Transcript) pdf Web Dev (Transcript) pdf

Type here to search

22°C Light rain 10/10/2021

Web Development and Python

79% COMPLETE

- Check Your Knowledge
- Cross-Site Scripting
- Mitigations
- Templates
- Jinja Example**
- Models and Controllers
- Check Your Knowledge

REFERENCES

References

# 2

```
<html>
<head>
  {% if title %}
  <title>{{ title }} Page</title>
  {% else %}
  <title>Welcome to the Page</title>
  {% endif %}
</head>
<body>

<nav>
<form action="/search/" method="GET"><input type="text" name="s" />
<input type="submit" value="Search" /></form>
</nav>
<hr>
</body>
</html>
```

Anything that is in between double curly braces is an expression that will be output to the final document. In our example, you can see that the title is one such example, that will be assigned dynamically by the template.

1 2 3

Cross-Site Scripting... pdf Routes (Transcript) pdf Web Dev (Transcript) pdf

Type here to search

22°C Light rain 10/10/2021



Web Development and Python

78% COMPLETE

Check Your Knowledge

Cross-Site Scripting

Mitigations

Templates

**Jinja Example**

Models and Controllers

Check Your Knowledge

REFERENCES

References

3

```

<html>
<head>
  {% if title %}
  <title>{{ title }} Page</title>
  {% else %}
  <title>Welcome to the Page</title>
  {% endif %}
</head>
<body>

<nav>
<form action="/search/" method="GET"><input type="text" name="s" />
<input type="submit" value="Search" /></form>
</nav>
<hr>
</body>
</html>

```

Jinja also offers control flow statements, like if or for and are denoted by the use of a percent sign. As you can see in the example, we use conditional statements to determine the title of the page. Unlike in Python, whitespace is not important here as blocks are denoted by start and end tags, and static text within a block could change the level of indentation and thus the meaning of the code.

Cross-Site Scripting...pdf

Routes (Transcript).pdf

Web Dev (Transcript)...pdf

Show all

Type here to search

22°C Light rain

ENG

2020 10/10/2021

Web Development and Python

78% COMPLETE

Check Your Knowledge

Cross-Site Scripting

Mitigations

Templates

Jinja Example

**Models and Controllers**

Check Your Knowledge

REFERENCES


References

Lesson 15 of 18

# Models and Controllers

---

Remember when we discussed routes? These are functions associated with certain URLs within your web application.



One way to think of routes is as a controller action – when the route is visited and the function is called, some logic is usually used to retrieve information from the model, typically a database. This data will then be passed to a view, which renders the requested page.

Cross-Site Scripting...pdf

Routes (Transcript).pdf

Web Dev (Transcript)...pdf

Show all

Type here to search


22°C Light rain

ENG

2020 10/10/2021

Web Development and Python

Remember when we discussed routes? These are functions associated with certain URLs within your web application.



One way to think of routes is as a controller action – when the route is visited and the function is called, some logic is usually used to retrieve information from the model, typically a database. This data will then be passed to a view, which renders the requested page.

When we retrieve data from a model, it is usually added to a Python data structure, like a dictionary or list, and the structure is the thing that we send to the view using the render template function.

CONTINUE

Task-Size Scripts...pdf Routes (Transcript).pdf Web Dev (Transcript)...pdf

Type here to search 22°C Light rain 2020 18/10/2021

Web Development and Python

```
@app.route("/")
def landing_page():
    database = get_db()
    database_cursor = database.execute("select * from entries")
    entries = database_cursor.fetchall()
    return render_template("index.html", entries=entries)
```

Within the example view function, we are taking data from the database and then we perform some very basic logic. This returns a list, which we have called entries that we pass to the template called index.html.

Lesson 16 - Check Your Knowledge

Task-Size Scripts...pdf Routes (Transcript).pdf Web Dev (Transcript)...pdf

Type here to search 22°C Light rain 2020 18/10/2021

Web Development and Python

100% COMPLETE

- Mitigations
- Templates
- Jinja Example
- Models and Controllers
- Check Your Knowledge**
- REFERENCES
- References
- FEEDBACK
- Feedback

Lesson 16 of 18

## Check Your Knowledge

[START QUIZ >](#)

Drop-Size Scripts...pdf Routes (Transcript).pdf Web Dev (Transcript)...pdf

Type here to search

22°C Light rain 18/10/2021

Web Development and Python

100% COMPLETE

- Mitigations
- Templates
- Jinja Example
- Models and Controllers
- Check Your Knowledge**
- REFERENCES
- References
- FEEDBACK
- Feedback

Question 01/02

Pick the odd one out:

- ☒ Deleting
- ☐ Whitelisting
- ☐ Escaping

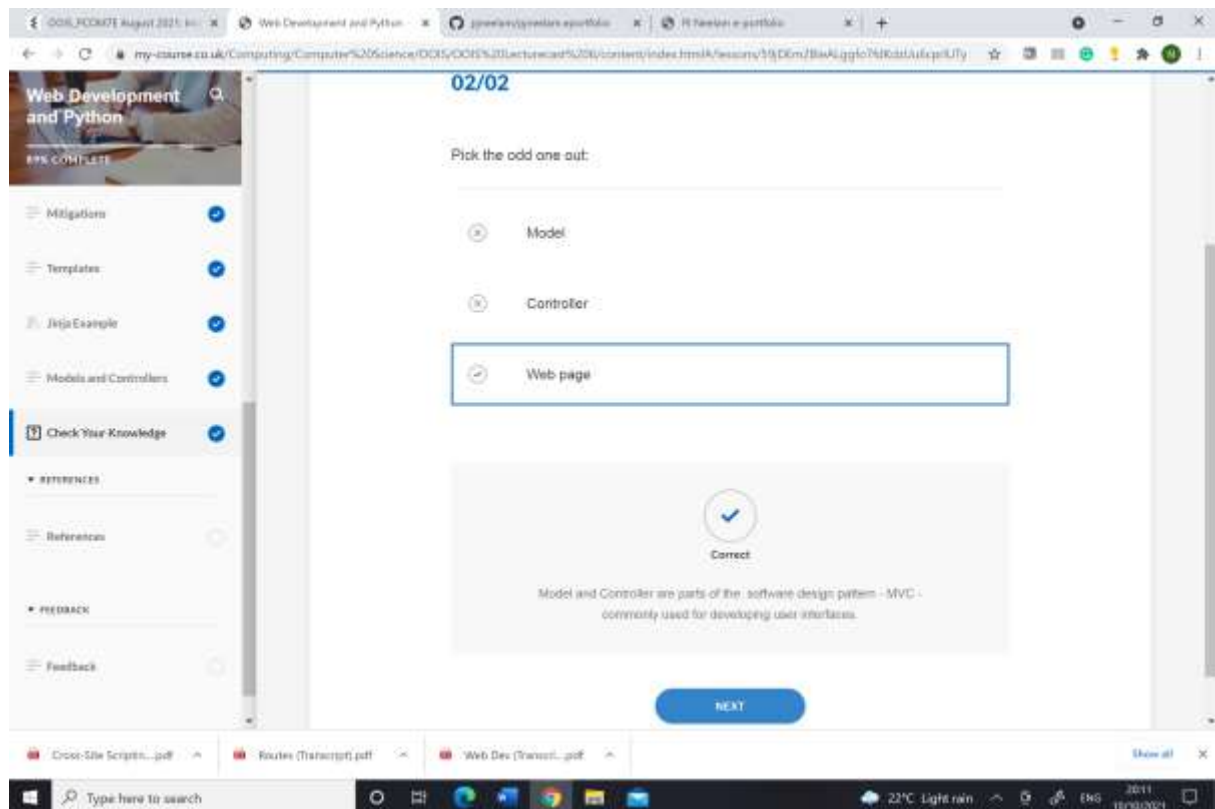
☒ Correct

Escaping and Whitelisting are ways to mitigate malicious code injections.

Drop-Size Scripts...pdf Routes (Transcript).pdf Web Dev (Transcript)...pdf

Type here to search

22°C Light rain 18/10/2021



## Unit 11 Reading

The reading this week focuses on further information on web application development and specifically working with Python and Flask.

### Required Reading

Ginberg, M. (2017) The Flask Mega Tutorial

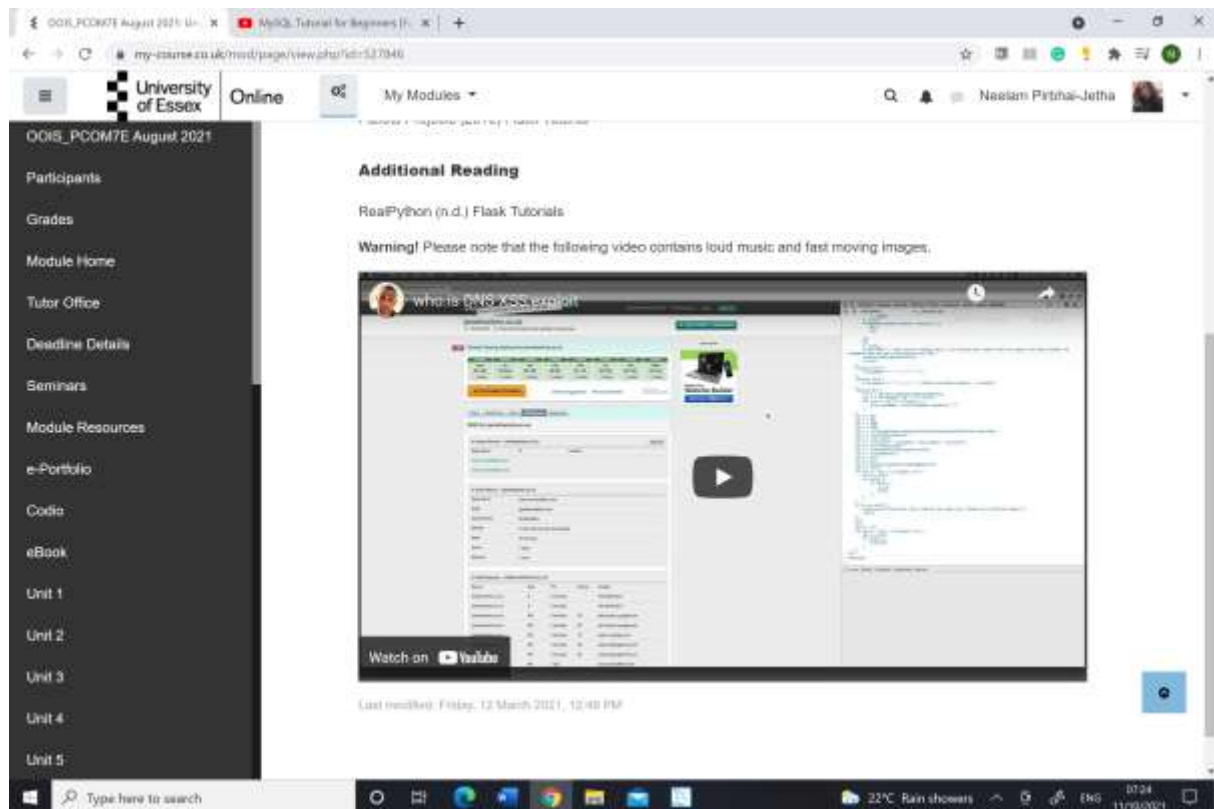
Pallets Projects (2010) Flask Tutorial

### Additional Reading

RealPython (n.d.) Flask Tutorials

**Warning!** Please note that the following video contains loud music and fast moving images.

Last modified: Friday, 12 March 2021, 12:48 PM



## Unit 11: Web Development in Python

Welcome to Week 11 this unit will tie together several of the subjects covered to date such as database development and Python programming. This unit focuses on developing web applications using a Python web framework called Flask.

### In this unit we shall:

- Introduce the concept of a Model-View-Controller.
- Discuss the key elements of a web application.
- Discuss some of the security considerations of a web development.

### On completion of this unit you will be able to:

- Implement a simple web server.
- Create dynamic templates using Jinja.
- Connect a database to a Flask application.
- Understand some of the security considerations around web applications.

The unit introduces the core concepts of web application development with examples using the Python Flask framework.

**Please note, the topics covered in units 9, 10 & 11 are to expand your knowledge of OOS and are not specifically required for the System Implementation assignment.**



However, if you would like to demonstrate any of these elements in the assignment you are welcome to do so.

Your progress

- **Lecturecast** | 1 hrs

- [Web Development and PythonURL](#)

- **Reading** | 6 hrs

- [Unit 11 ReadingPage](#)

- **Formative activities** | 7 hrs

**Codio.** Work through the Codio module called '[Introduction to Flask](#)'.

**Remember** to record your results, ideas and feedback in your e-portfolio.

- **Seminar 6** | 2 hr

### Title: e-Portfolio Submission

Join the [Seminar Session](#) this week. Remember to bring the seminar questions and your attempted answers.

Visit the 'Seminar 6 preparation' below to complete the activities prior to the seminar session. As we will discuss these concepts in the seminar session, ensure you have completed the activities before this week's Seminar session.

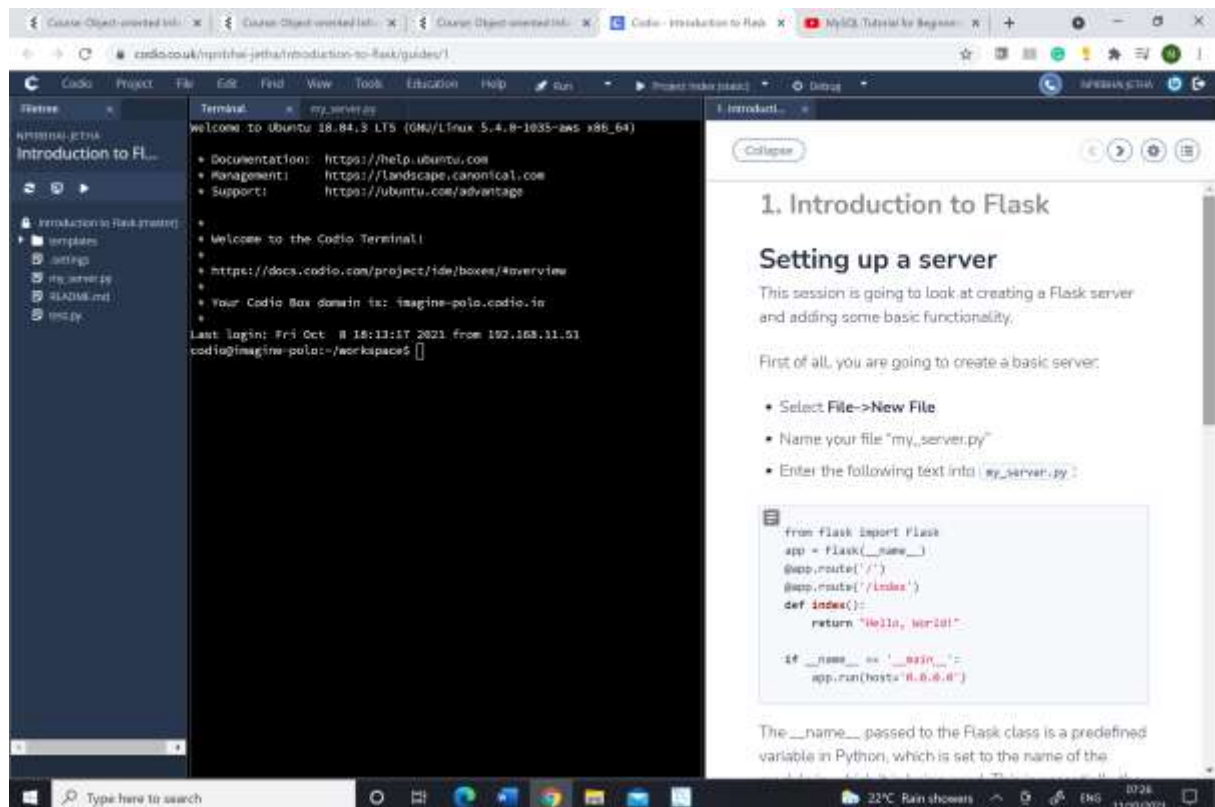
- [Seminar 6 preparationPage](#)

- **Summative assessment** | 8 hrs

- [System ImplementationAssignment](#)

- **Research Study Survey** | 10 mins

- [e-Portfolio SurveyQuestionnaire](#)



## 1. Introduction to Flask

### Setting up a server

This session is going to look at creating a Flask server and adding some basic functionality.

First of all, you are going to create a basic server:

- Select **File->New File**
- Name your file “my\_server.py”
- Enter the following text into `my_server.py`:

```
from flask import Flask
```

```
app = Flask(__name__)
```

```
@app.route('/')
```

```
@app.route('/index')
```

```
def index():
```

```
    return "Hello, World!"
```

```
if __name__ == '__main__':
```

```
    app.run(host='0.0.0.0')
```

The `__name__` passed to the Flask class is a predefined variable in Python, which is set to the name of the module in which it is being used. This is essentially the first place that Flask will look when it needs to load any associated resources such as template files. Typically, passing `__name__` will be the correct way to configure Flask correctly.

The application will then import the routes module. Routes are the different URLs that the application can implement. Flask uses Python functions for the request handlers, which are called view functions. A view function is mapped to one or more route URLs so that Flask knows what logic to execute whenever the client requests a specific URL.

The first view function defined in the code that you have just entered, `index`, is a really simple function that will just return a greeting as a string. The definition of the function uses decorators above it, which you may have come across before in Java. The decorator here creates an association between the URL given as an argument and the function itself. This means that whenever the URL is visited Flask will invoke the associated function.

Try running the application in the terminal with:

```
python my_server.py
```

You can view the web page you have created by using the 'Box URL' option in the Preview menu or navigate to the URL at <https://<name of your box>-5000.codio-box.uk/> (for example, <https://liter-shine-5000.codio-box.uk/>).

