E-Portfolio – Final Submission

Course: MSc Computer Science

Module: Object-Oriented Information Systems

Assignment: ePortfolio

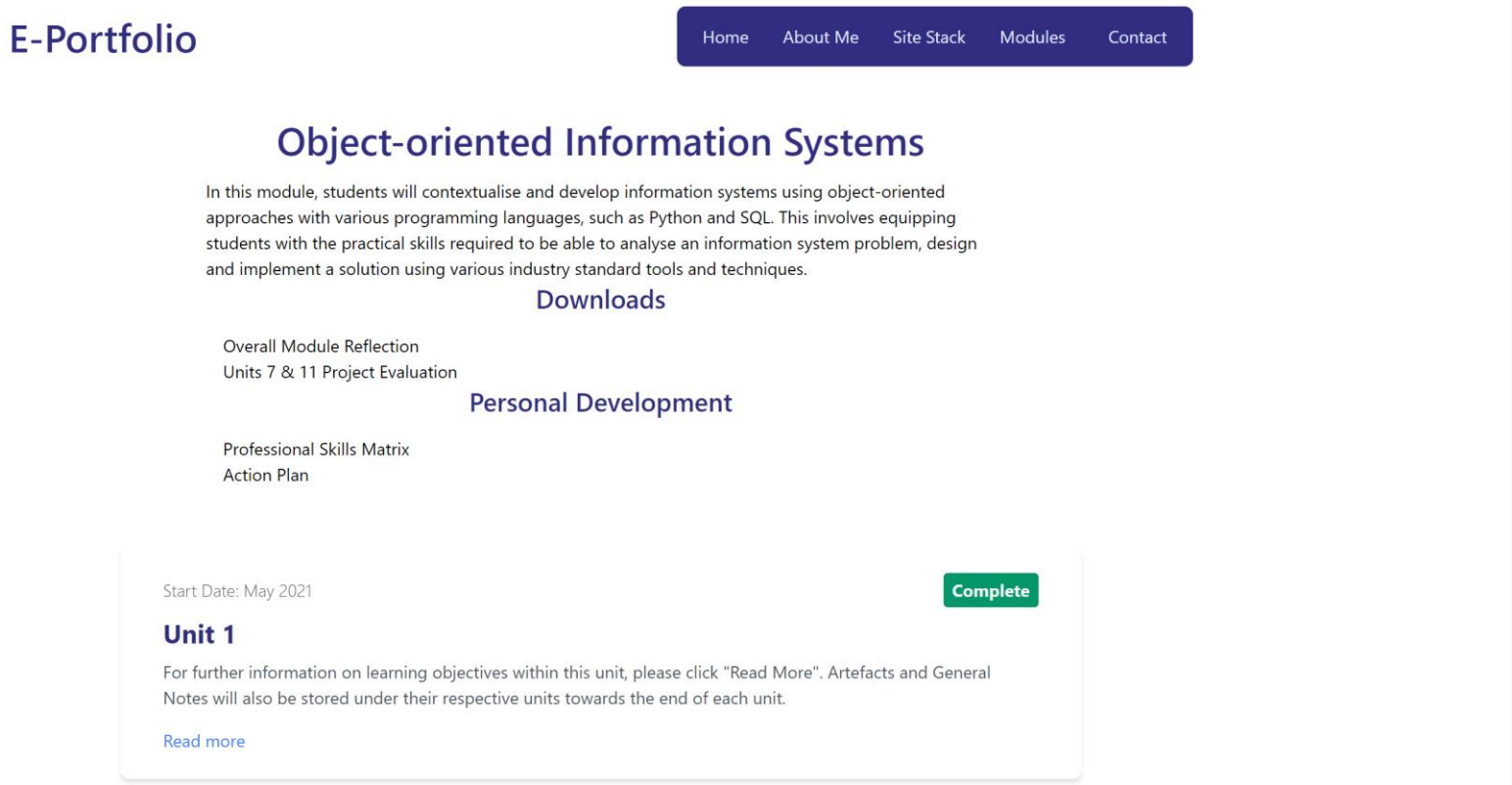
Date: Sunday 25th July 2021

Student ID: 126853

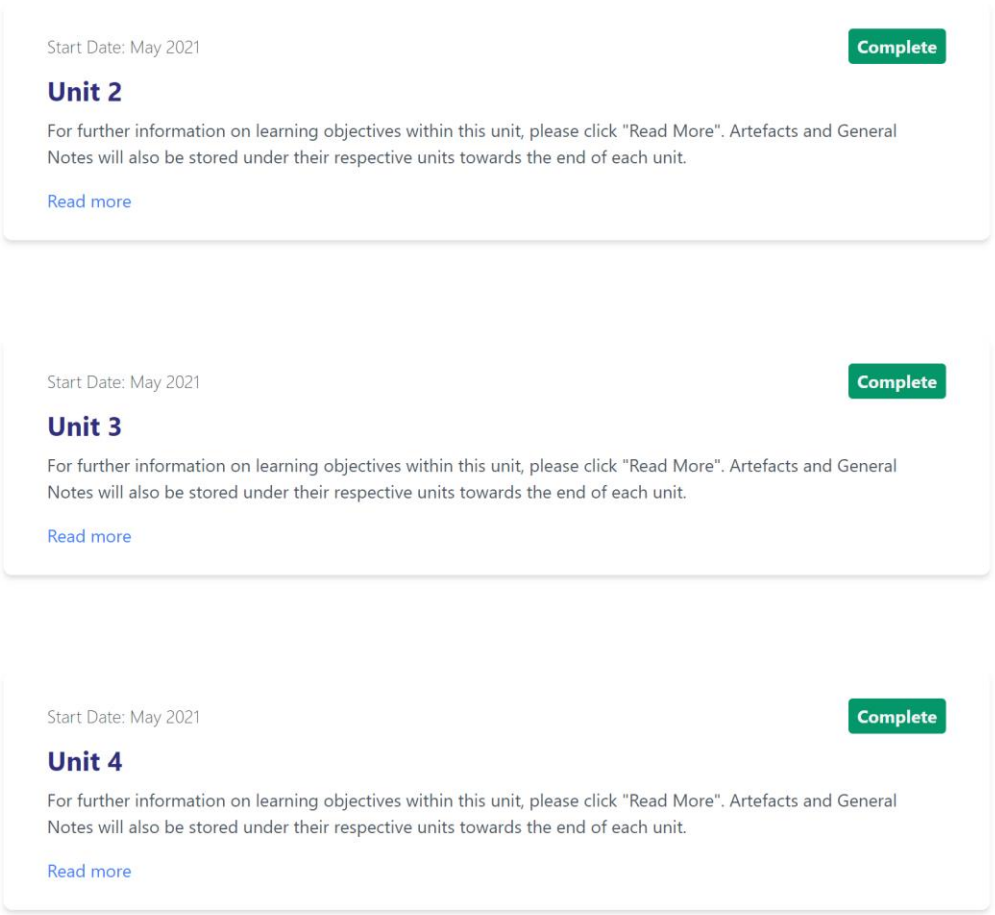
Contents

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

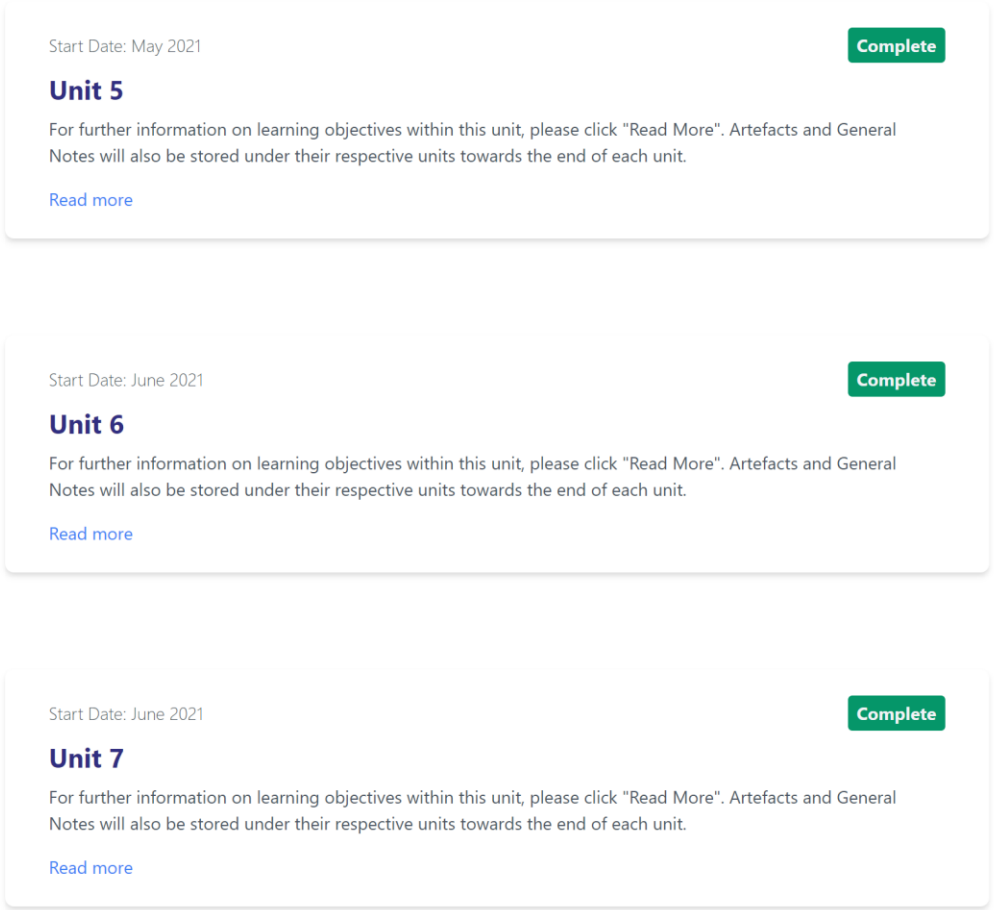
Main Module Home:



https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

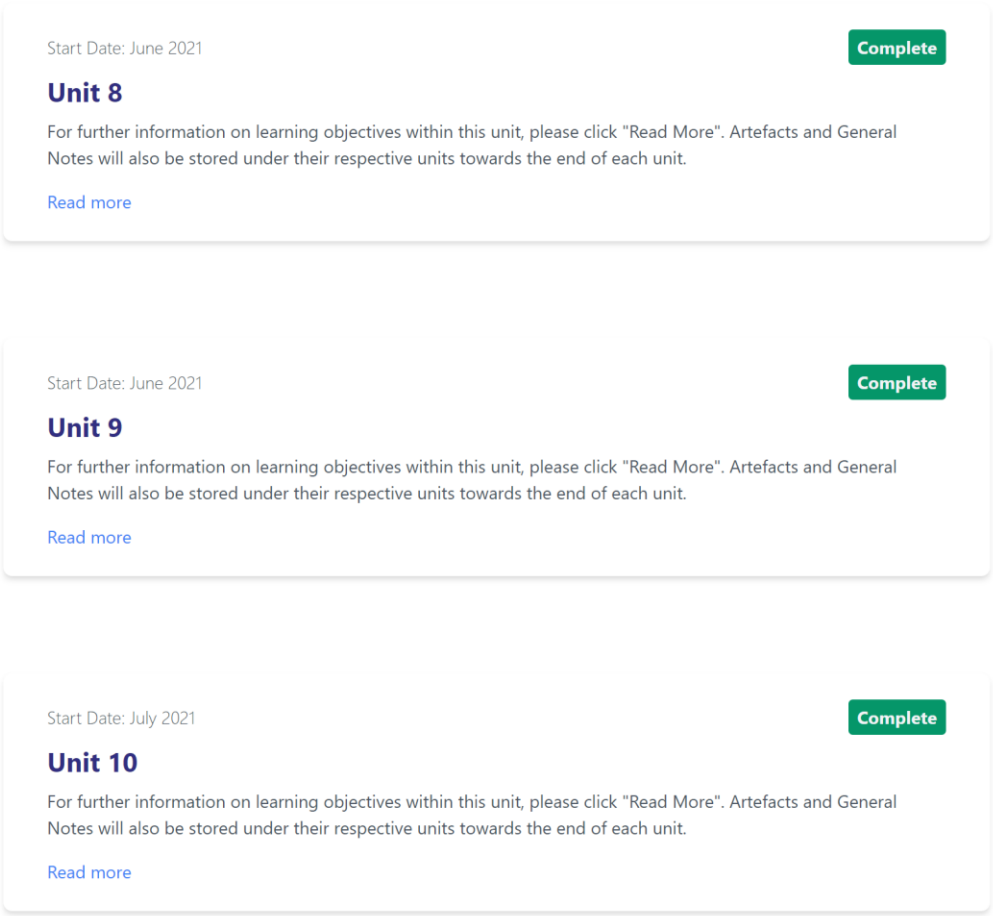


https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

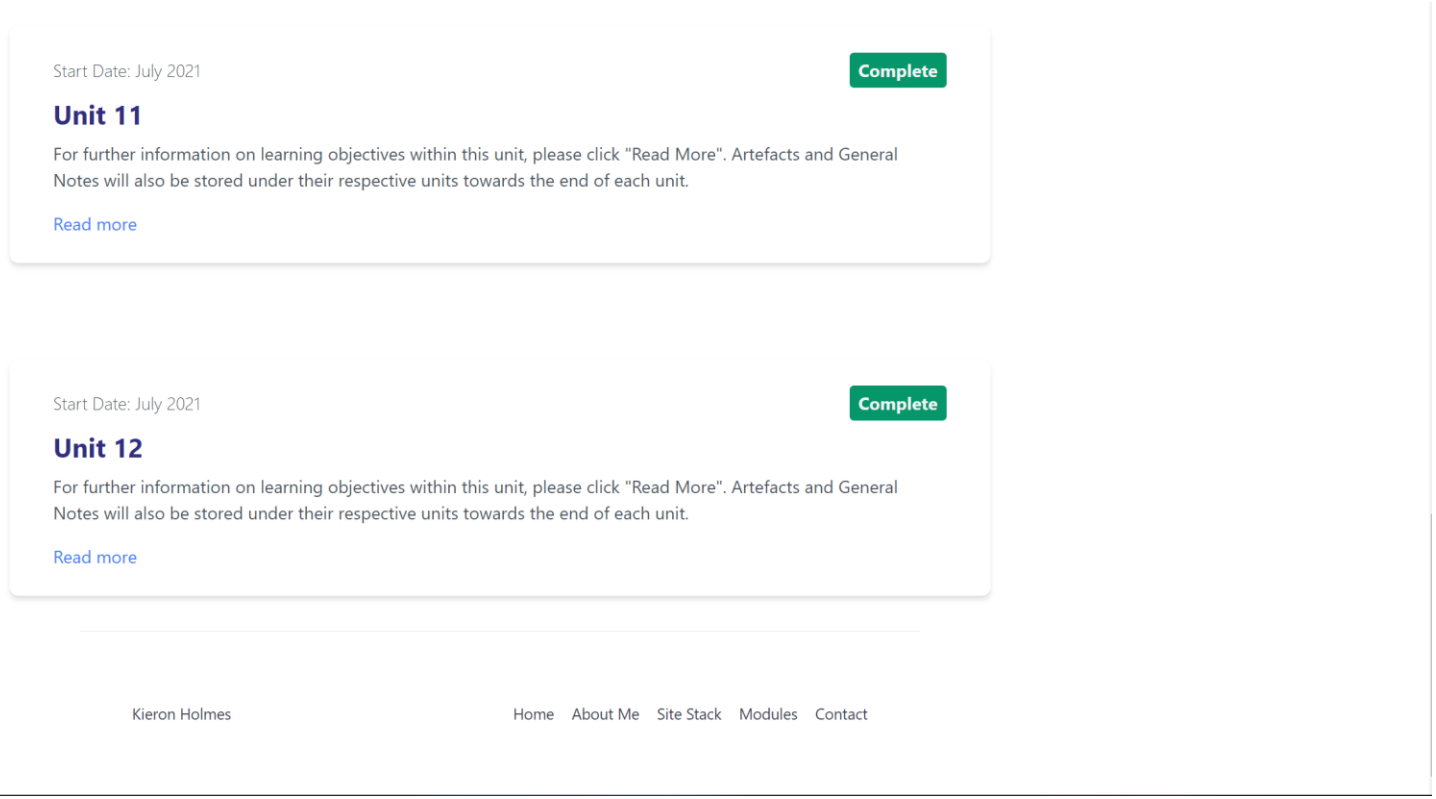


https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

---



https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems



Object-oriented Information Systems

In this module, students will contextualise and develop information systems using object-oriented approaches with various programming languages, such as Python and SQL. This involves equipping students with the practical skills required to be able to analyse an information system problem, design and implement a solution using various industry standard tools and techniques.

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Downloads

Overall Module Reflection

Units 7 & 11 Project Evaluation

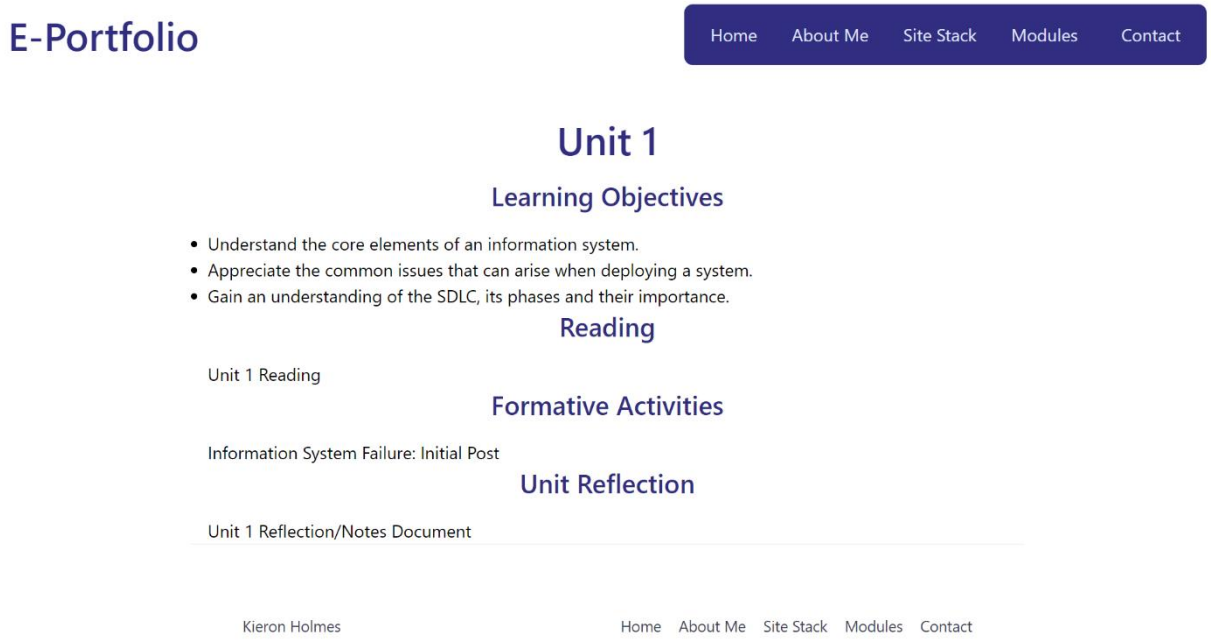
Personal Development

Professional Skills Matrix

Action Plan

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Unit 1:



Learning Objectives

- Understand the core elements of an information system.
- Appreciate the common issues that can arise when deploying a system.
- Gain an understanding of the SDLC, its phases and their importance.

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Reading

Unit 1 Reading

Formative Activities

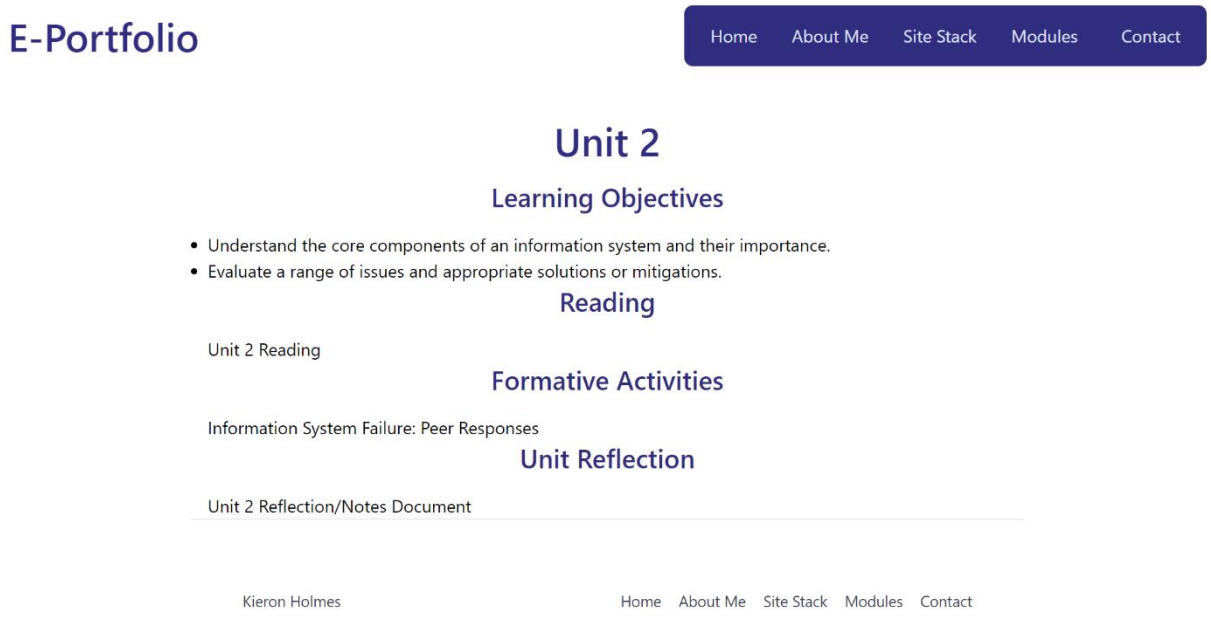
Information System Failure: Initial Post

Unit Reflection

Unit 1 Reflection/Notes Document

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Unit 2:



Learning Objectives

- Understand the core components of an information system and their importance.
- Evaluate a range of issues and appropriate solutions or mitigations.

Reading

Unit 2 Reading

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Formative Activities

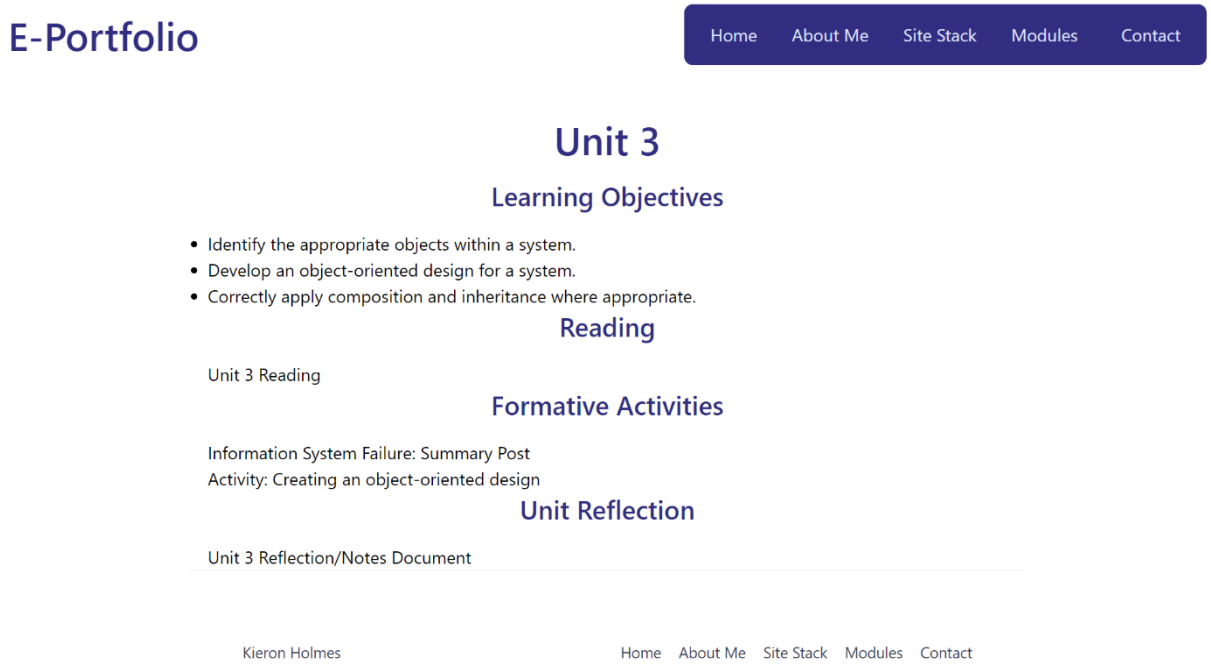
Information System Failure: Peer Responses

Unit Reflection

Unit 2 Reflection/Notes Document

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Unit 3:



Learning Objectives

- Identify the appropriate objects within a system.
- Develop an object-oriented design for a system.
- Correctly apply composition and inheritance where appropriate.

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Reading

Unit 3 Reading

Formative Activities

Information System Failure: Summary Post

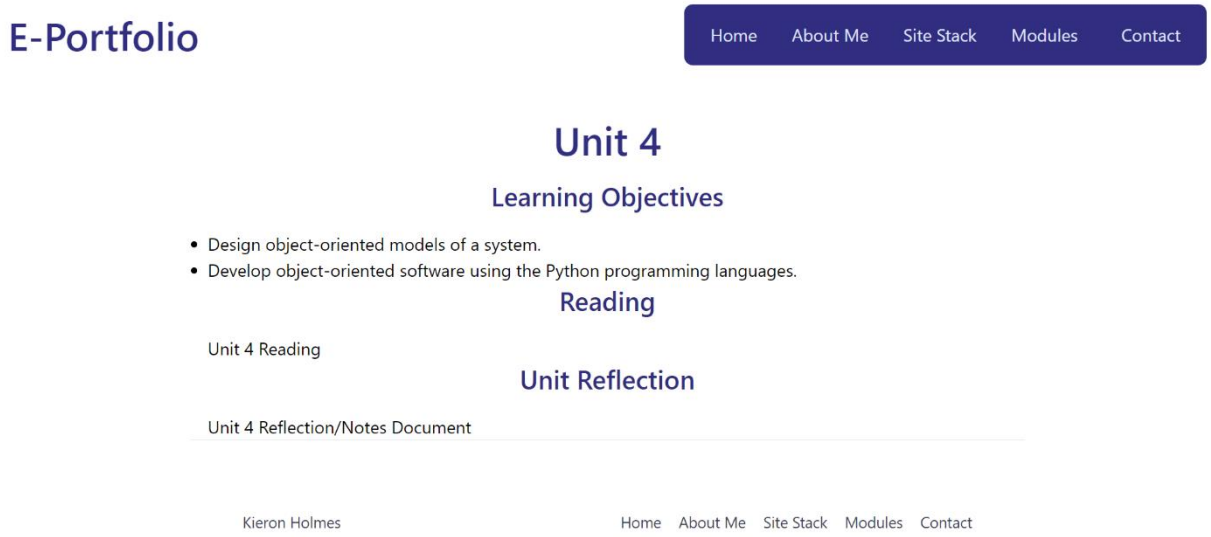
Activity: Creating an object-oriented design

Unit Reflection

Unit 3 Reflection/Notes Document

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Unit 4:



Learning Objectives

- Design object-oriented models of a system.
- Develop object-oriented software using the Python programming languages.

Reading

Unit 4 Reading

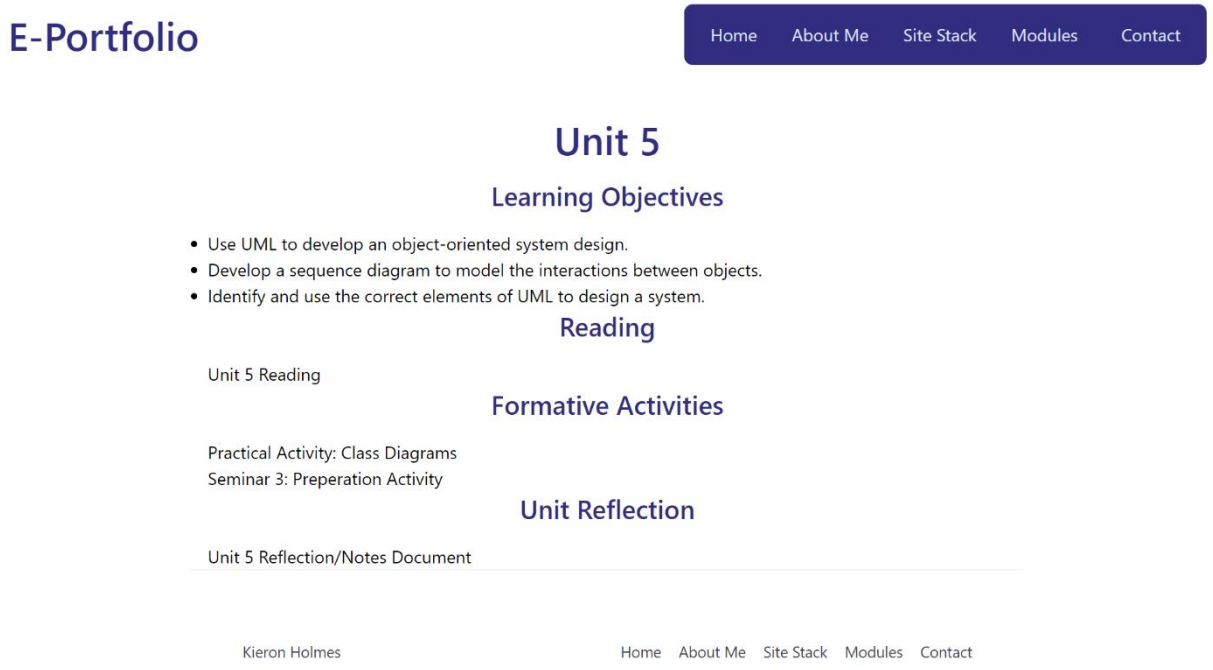
Unit Reflection

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Unit 4 Reflection/Notes Document

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Unit 5:



https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Learning Objectives

- Use UML to develop an object-oriented system design.
- Develop a sequence diagram to model the interactions between objects.
- Identify and use the correct elements of UML to design a system.

Reading

Unit 5 Reading

Formative Activities

Practical Activity: Class Diagrams

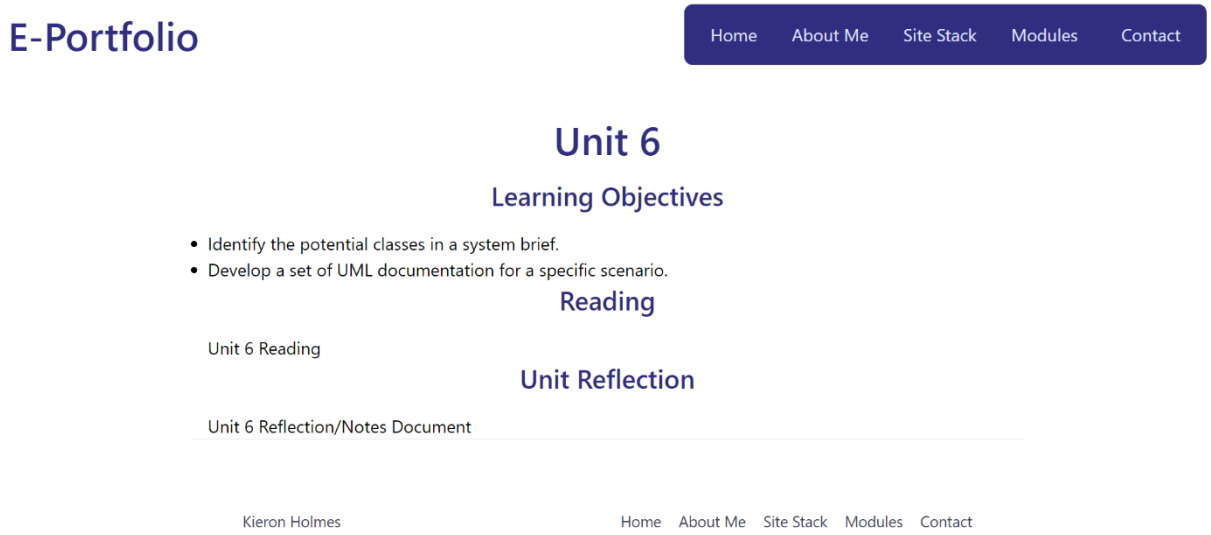
Seminar 3: Preperation Activity

Unit Reflection

Unit 5 Reflection/Notes Document

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Unit 6:



Learning Objectives

Identify the potential classes in a system brief.

Develop a set of UML documentation for a specific scenario.

Reading

Unit 6 Reading

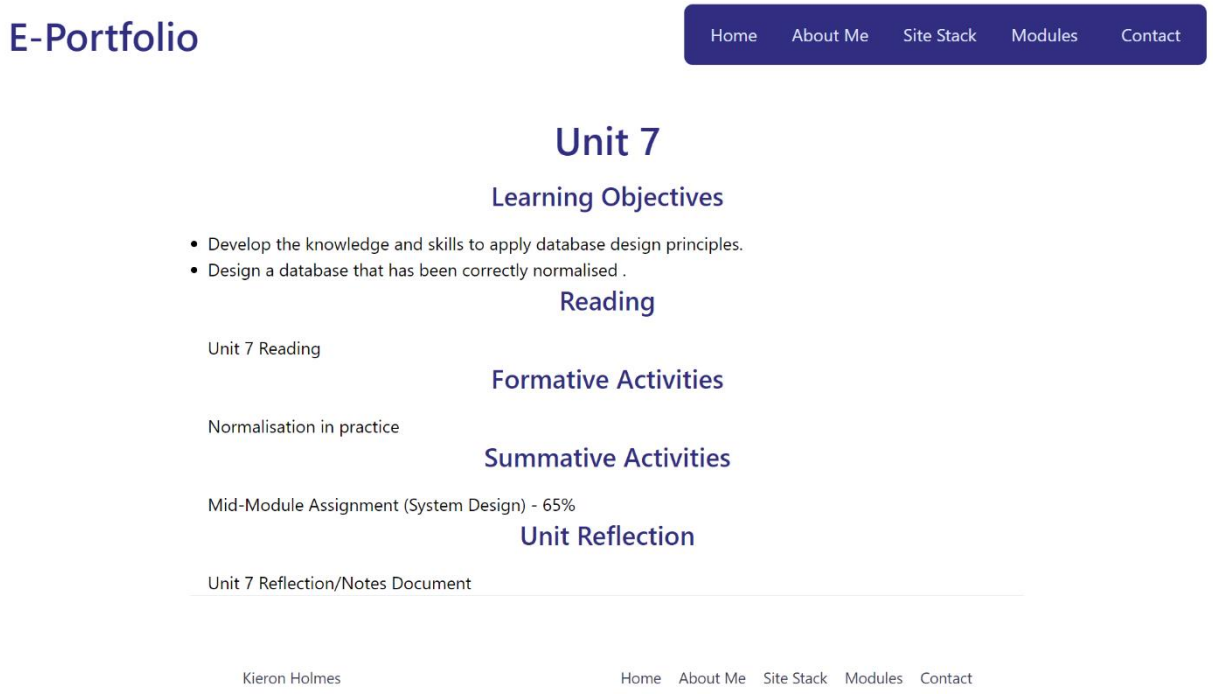
Unit Reflection

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Unit 6 Reflection/Notes Document

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Unit 7:



https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Learning Objectives

- Develop the knowledge and skills to apply database design principles.
- Design a database that has been correctly normalised

Reading

Unit 7 Reading

Formative Activities

Normalisation in practice

Summative Activities

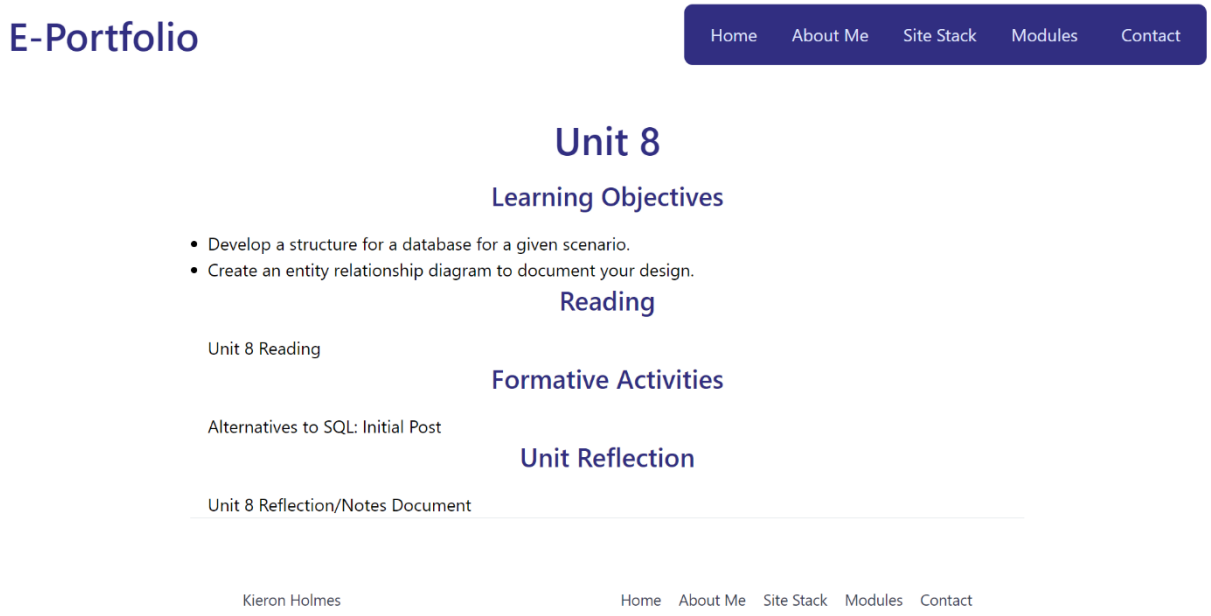
Mid-Module Assignment (System Design) - 65%

Unit Reflection

Unit 7 Reflection/Notes Document

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Unit 8:



Learning Objectives

- Develop a structure for a database for a given scenario.
- Create an entity relationship diagram to document your design.

Reading

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Unit 8 Reading

Formative Activities

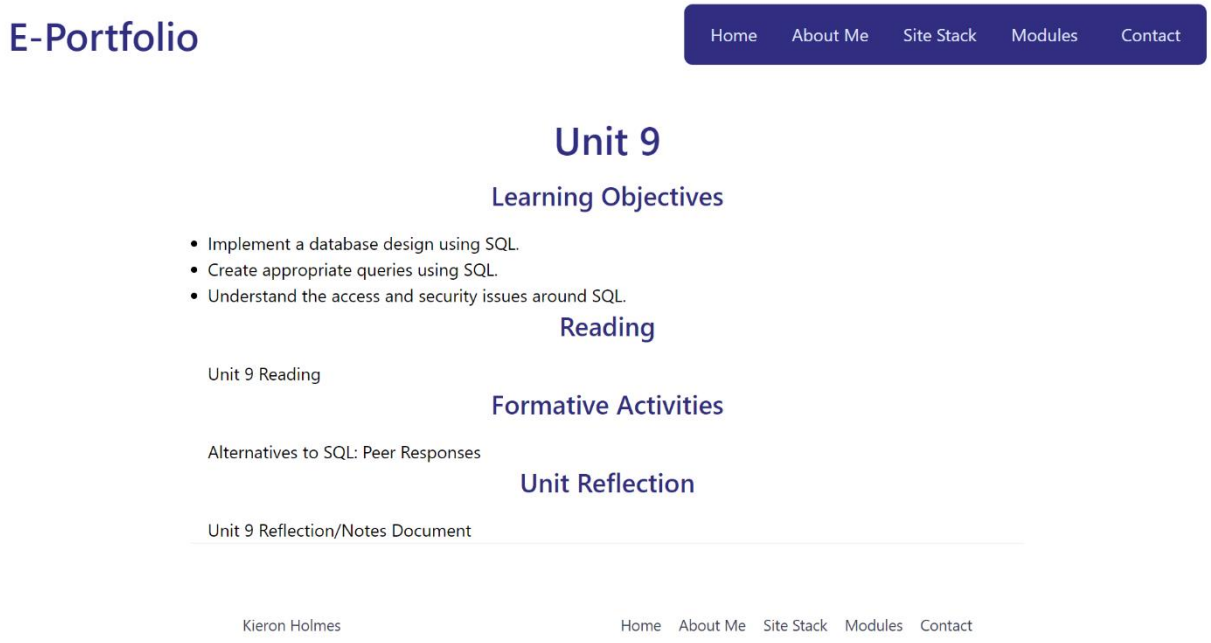
Alternatives to SQL: Initial Post

Unit Reflection

Unit 8 Reflection/Notes Document

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Unit 9:



Learning Objectives

- Implement a database design using SQL.
- Create appropriate queries using SQL.
- Understand the access and security issues around SQL.

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Reading

Unit 9 Reading

Formative Activities

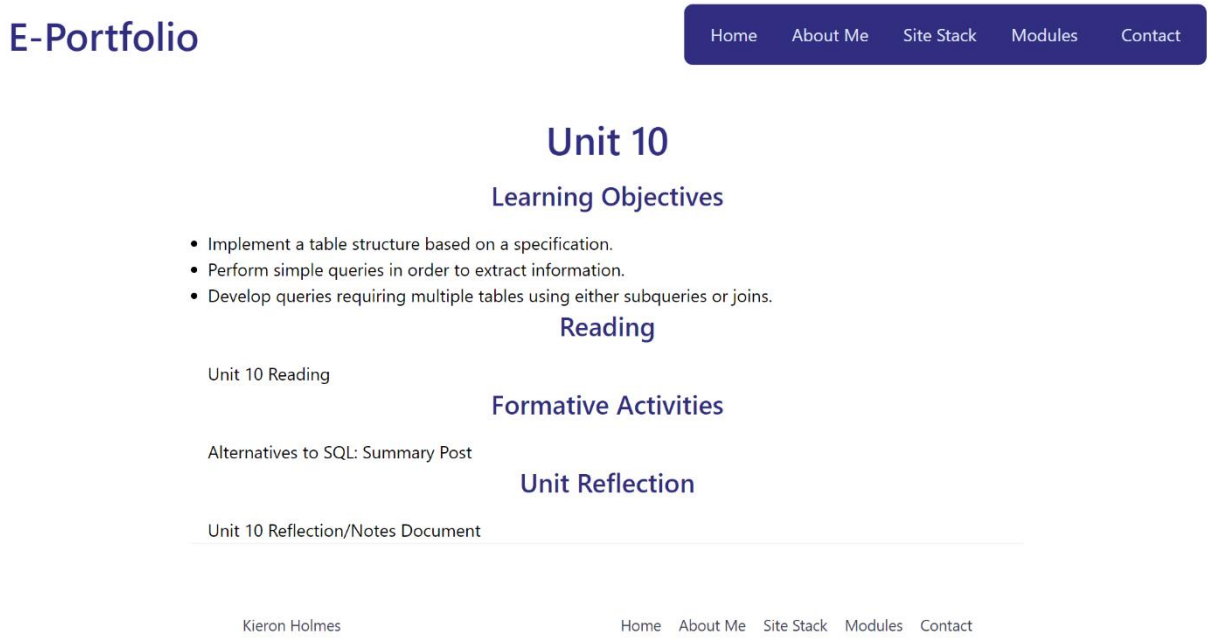
Alternatives to SQL: Peer Responses

Unit Reflection

Unit 9 Reflection/Notes Document

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Unit 10:



Learning Objectives

- Implement a table structure based on a specification.
- Perform simple queries in order to extract information.
- Develop queries requiring multiple tables using either subqueries or joins.

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Reading

Unit 10 Reading

Formative Activities

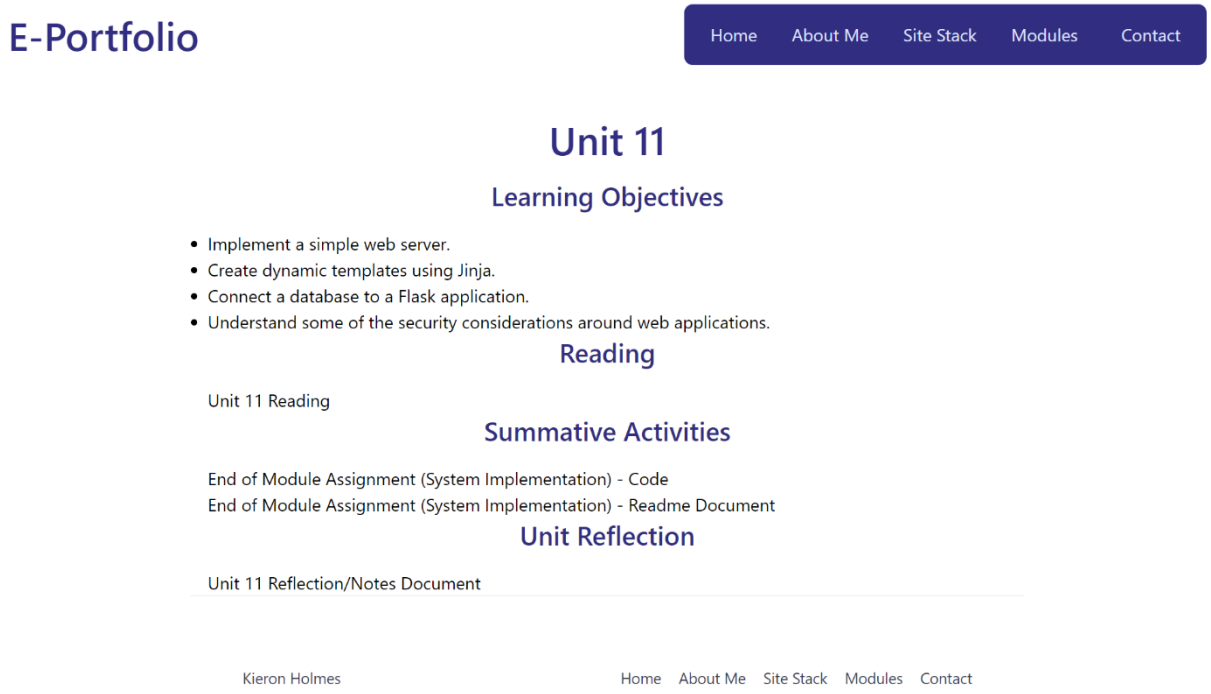
Alternatives to SQL: Summary Post

Unit Reflection

Unit 10 Reflection/Notes Document

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Unit 11:



https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Learning Objectives

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.

Reading

Unit 11 Reading

Summative Activities

End of Module Assignment (System Implementation) - Code

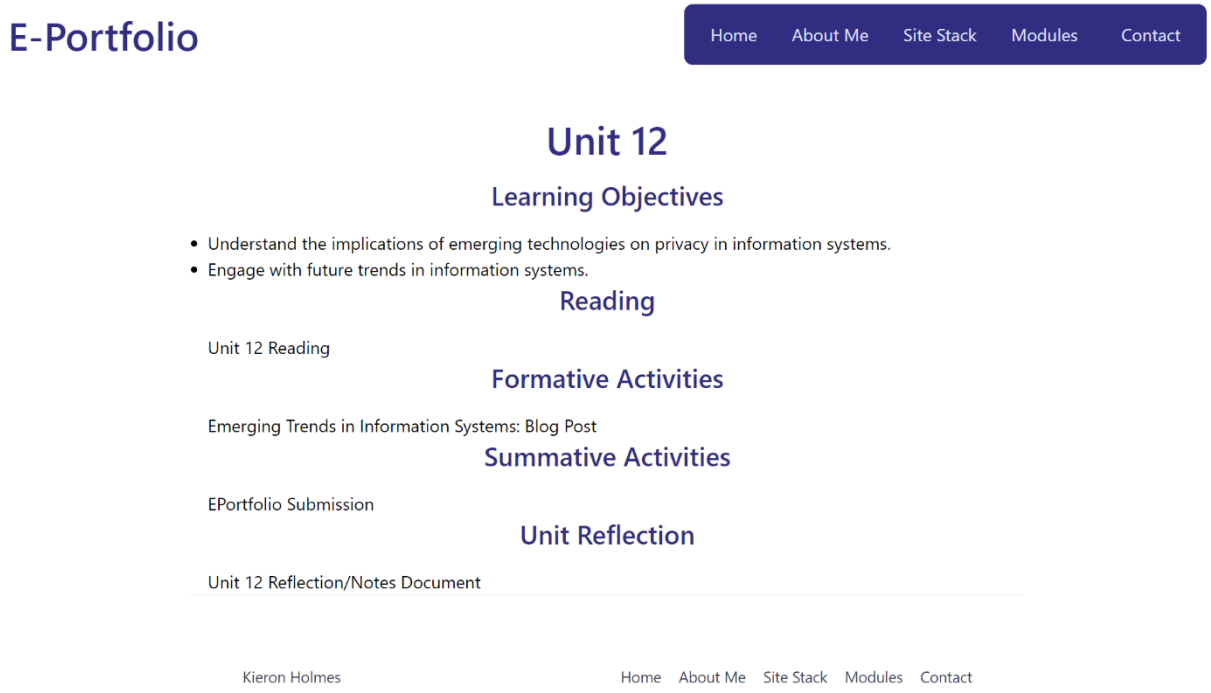
End of Module Assignment (System Implementation) - Readme Document

Unit Reflection

Unit 11 Reflection/Notes Document

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Unit 12:



Learning Objectives

- Understand the implications of emerging technologies on privacy in information systems.

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

- Engage with future trends in information systems.

Reading

Unit 12 Reading

Formative Activities

Emerging Trends in Information Systems: Blog Post

Summative Activities

EPortfolio Submission

Unit Reflection

Unit 12 Reflection/Notes Document

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Module Reflection:

The second module of the MSc Computer Science course provided us with an in-depth introduction to object-oriented systems development/design. Throughout this module, we have been introduced to various techniques, including UML diagrams, database design (Including Normalisation) and object-oriented Python development.

One of the main aims of this module was to appraise and evaluate the concepts and principles of an information system in relation to object-oriented systems. In addition, we focused on identifying whether best practices were used in specific situations and how this factor can affect the overall functionality of the Information System.

In particular, we discussed the overall risks of Information System failure and how these can have adverse effects not only on other businesses but the day-to-day life of an individual. From this aim, I have understood the overall concepts of Information Systems and considering the importance of comprehensive system planning to avoid potentially catastrophic consequences.

The next aim of this module was to design and document an object-oriented information system. We were introduced to the Unified Modelling Language to complete this aim, creating a series of diagrams including State Diagrams, Activity Diagrams and Class Diagrams. The Unit Reading list included the UML Distilled (Fowler, 2013) and Elements of UML Style (Ambler, 2003) books, both of which were valuable resources when learning the core concepts and limitations of the UML standard.

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Through this module, I have learned the importance of performing adequate system planning at the early stages of the Software Development Life Cycle (SDLC) and how adequate system designs can help prevent information system failure later down the line. Within my professional career, I will aim to use techniques learnt within this module, such as UML, to identify distinct resources with a given brief, and plan the user journey using State Transition diagrams where appropriate.

The third aim of this module was to develop an object-oriented system design. Firstly, we were introduced to the core principles of object-oriented design, which includes Inheritance, Composition, Encapsulation and Polymorphism. These resources were available in the Codio online learning platform, which provided me with real-time feedback confirming my understanding of those topics and their implementation within Python.

To test our understanding of these techniques and the correct application, we were required to create a basic application for an ‘online store’. In addition, we would provide separate facilities for Customers, Staff Members and Sellers, which should demonstrate our practical understanding of object-oriented techniques, including Inheritance, Composition, Encapsulation and Polymorphism. This aim has helped me further understand the correct application of object-oriented techniques within a mainstream programming language such as Python.

The final aim of this module was to “Develop, implement and evaluate critically information system solutions to facilitate business decisions.”. This aim has required

us to consider the specific technologies used when implementing a final solution based on its particular merits in that situation. To demonstrate this, we were introduced to the concept of NoSQL databases and considering cases where its implementation could be would be preferred over conventional relational databases.

We learnt that, unlike conventional SQL databases, NoSQL databases share no common query language, and they don’t require the user to pre-define a schema (MongoDB, n.d.), making it ideal for big data applications such as Analytics and Financial. A series of discussions occurred between myself and my peers within the Discussion forum. We considered the positives and negatives of implementing this technology based on various contributory factors, including business size, the volume of data, and overall structure. From this aim, I have understood that various factors can influence technologies used within Information Systems development, which need to be appraised on a case-by-case basis.

Throughout this unit, I have completed various tasks to demonstrate my understanding of Object-Oriented systems. Examples of these include Collaborative Learning (In the form of a Discussion Forum), creating UML diagrams, and implementing an object-oriented design within Python. These have enhanced my understanding of object-oriented systems, particularly with regard to the implementation within the Python language, as well as the creation of in-depth design documentation. This will be used to ensure I continue to produce high-standard applications, following industry-standard design methodologies.

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

References:

Fowler, M. (2013) UML Distilled: a brief guide to the standard object modeling language. 3rd ed. Boston, MA: Addison Wesley

Ambler, S. (2003) Elements of UML Style. Cambridge: Cambridge University Press

MongoDB. (n.d.) NoSQL vs Relational Databases. Available From: https://www.mongodb.com/scale/nosql-vs-relational-databases [Accessed 24th July 2021].

https://eportfolio.kieronholmes.me/modules/object-oriented-information-systems

Project Evaluation:

Within the Object-Oriented Information Systems module, we were provided with an ‘online store’ scenario. This was required to have a Customer, Seller and Warehouse portal, providing various functions to the user. Within Unit 7, we were required to produce a series of UML diagrams to base our final implementation, consisting of a Class Diagram, an Activity Diagram, and a State Diagram (Highlighting system transitions). Finally, within Unit 11, we were required to implement the designs from Unit 7 in an object-oriented fashion using Python, highlighting cases where a SQL database could be used.

In the system planning stage (Unit 7), a collection of three UML diagrams was completed, highlighting the key elements of each class and documenting the transitions/events that were to occur at a specific point within the program. The first diagram produced was a Class Diagram, which was created based upon key items identified as ‘objects’ within the assignment brief, focusing on the type of relation and method/function visibility. The second diagram produced was an Activity Diagram, which highlights actions that the users can take during the checkout process. The final diagram built was the State Diagram, highlighting the various stages of the order process (Such as In Progress, Awaiting Picking etc.). These diagrams have been handy at the later stages of the development process, as they provide a crucial reference point when creating classes and functions.

When creating the code deliverables for this project, I had ensured that each class was separated into a standalone file, which the requesting class would import. This

ensured that resources remained easily readable and distinct and provided benefits such as code reusability and optimised compilation (Mefford, 2019).

All code produced was intended to conform to the PEP8 style guidelines (Van Rossum, 2001), which is a set of style guidelines produced by a series of developers from the Python Software Foundation. All code was run through the AutoPEP8 (PyPi, n.d.) module, which is available through the Python repository to perform a ‘first fix’ of the code, ensuring it is compliant with the style guidelines. A final manual check was run through the style checker within the PyCharm IDE to ensure it was fully compliant. Implementing this stage into my development process ensures that code is in a standardised format that is easily readable by other Python developers.

When producing the Readme document to accompany the code, I ensured that detailed platform-agnostic documentation on how to execute the code was included. For example, my project makes use of standard Python Virtual Environments, which can ensure that regardless of packages installed on the host machine, you can set up an identical environment to that of my development setup. In addition, this ensures that code is less likely to contain versioning errors, especially when using external modules that could have changed, such as the PrettyTables module.

As testing is an essential part of the development process, each class and method were tested individually during the development process, as well as all together at the end of the project. I compiled a thorough list of items/functions that needed to be tested, which was used to produce a comprehensive test plan. This test plan contained a list of the expected outputs of the system, which was based upon the

system brief and was then compared to the actual outputs of the system, highlighting any inconsistencies. This stage was completed to ensure that all code was functioning as expected, following the provided assignment brief.

In conclusion, I believe that I have thoroughly met all aims and objectives of this module whilst demonstrating my knowledge and understanding of object-oriented techniques, the Python programming language, and the production of UML diagrams. Although the overall object-oriented techniques are the same, this project has highlighted the vast differences in how they are implemented across programming languages I am more familiar with, such as PHP.

References:

Mefford, A. (2019) Should I keep each Python class in a separate file?. Available From: https://www.quora.com/Should-I-keep-each-Python-class-in-a-separate-file [Accessed 24th July 2021].