Nicolas Pallant 28785959

INDIVIDUAL LEARNING SUMMARY

INTRODUCTION

This report summarises what I learnt in FIT5032 – Internet Applications Development. It includes a self-assessment against the criteria described in the unit outline, a justification of the pieces included, details of the coverage of the unit's learning outcomes, and a reflection on my learning.

OVERVIEW OF PIECES INCLUDED

This section outlines the pieces that I have included in my portfolio…

1. Task 1.1 ASP.NET LocalHost first time setup of web application and screenshot
2. Task 1.2 Discussion of different IDEs and the features of Visual Studio as an IDE
3. Task 2.1 Initial setup of CSS and Bootstrap responsiveness
4. Task 2.2 First-time development of an ASP.NET MVC application using scaffolding
5. Task 3.1 Hello World C# introduction to Views
6. Task 3.2 Initial setup of GitHub repository and URL
7. Task 4.1 Setup of ASP.NET MVC Web Application using Model First Approach
8. Task 4.2 Setup of ASP.NET MVC Web Application using Code First Approach
9. Task 5.1 Introduction to Javascript and the difference between defer and async
10. Task 5.2 Introduction to using the Bootstrap datepicker
11. Task 6.1 Implementation of Validation in C# Models
12. Task 6.2 Explanation of jQuery unobtrusive validation and its use in ViewModels
13. Task 7.1 Group Presentation on MD5 and SHA1, and its weaknesses/alternative
14. Task 7.2 Using ASP.NET Identity and Google as an additional login service
15. Task 8.1 Setting up Email API with SendGrid as a 3rd party emailing tool
16. Task 8.2 The advantages and disadvantages of using a 3rd party emailing tool
17. Task 9.1 Using responsive images and the difference between .woff and .woff2
18. Task 9.2 The new features of .NET Core and difference with ASP.NET Framework
19. Task 10.1 Setting up first Web API using .NET Core
20. Task 10.2 First time set up of AngularJS application
21. Task 11.1 Hosting a web app using Azure Cloud and the differences between CI and CD
22. Task 11.2 Different software testing approaches and Unit Testing using ASP.NET MVC

COVERAGE OF THE LEARNING OUTCOMES

This section outlines how the pieces I have included demonstrate the depth of my understanding in relation to each of the unit's learning outcomes.
[For each Learning Outcome, describe the work you have included in your portfolio that demonstrates your ability in relation to the required outcomes.]

LO 1: DEMONSTRATE THE IMPACT OF THE HISTORY OF WEB APPLICATIONS DEVELOPMENT ON CURRENT WEB-TECHNOLOGY

The following pieces demonstrate my ability in relation to this LO:

- Task 1.2 shows how different IDEs, old and new can be used to develop web applications, and the IDEs used in the past for developing Web Applications
- Task 5.1 shows the differences between async and defer and how it impacts the user, and reflects the impact that previous web applications have on today's.

- Task 6.2 shows how unobtrusive validation has become the normal nowadays, which shows it was a problem in the past
- Task 7.1 showed a lot about the past in terms of technology such as encryption methods and why they have needed to evolve for modern web development
- Task 8.2 showed why the need for 3rd party emailing tools are there, and how people used to manage sending emails automatically in the past
- Task 9.1 showed exactly how .woff had evolved into .woff2 to represent the time's need to advance and become quicker
- Task 9.2 showed how ASP.NET Framework had become outdated for the needs of modern web development when compared to ASP.NET Core
- Task 11.1 showed exactly how development methods such as CI and CD have changed over the years
- Task 11.2 showed the different types of software testing approaches and how they came to be, and why they are needed in modern web application development

LO 2: DESIGN, CONSTRUCT AND PUBLISH WEB-DATABASE APPLICATIONS

The following pieces demonstrate my ability in relation to this LO:

- Task 1.1 Developing an ASP.NET LocalHost
- Task 2.1 Setting up CSS and Bootstrap responsiveness
- Task 2.2 ASP.NET MVC Web Application development using scaffolding
- Task 3.1 C# Introduction to Website Views
- Task 4.1 Setup of ASP.NET MVC Web Application using Model First Approach
- Task 4.2 Setup of ASP.NET MVC Web Application using Code First Approach
- Task 5.1 Introduction to Javascript and simple scripts for websites
- Task 5.2 Introduction to using Bootstrap datepicker to make development simpler and more versatile
- Task 6.1 Implementation of validation to reduce input error
- Task 7.2 Using ASP.NET Identity to manage users
- Task 10.1 Developing a Web API using .NET Core
- Task 10.2 First time setup of AngularJS application
- Task 11.1 Hosting and publishing a web-database application using Azure Cloud
- Task 11.2 Unit testing using ASP.NET MVC and its importance to development

LO 3: ANALYSE AND CRITIQUE THE KEY TECHNOLOGICAL ISSUES CONFRONTING DEVELOPERS BUILDING WEB-DATABASE APPLICATIONS

The following pieces demonstrate my ability in relation to this LO:

- Task 1.2 Discussing the advantages and disadvantages of Visual Studio when it comes to building Web apps within it
- Task 3.2 GitHub and its importance for version control
- Task 5.1 The differences between async and defer and the issues between them
- Task 6.2 Why jQuery unobtrusive validation is needed in a modern world and offloading from the server
- Task 7.1 The rapid betterment of computing hardware and hacking abilities leading to encryption methods going out of date, and the need to adapt
- Task 8.2 How 3rd party apps like Sendgrid could be an issue for developers
- Task 9.1 The importance of responsive images and the need for compression to get better with .woff and .woff2
- Task 11.2 The different types of software testing approaches and the issues of each one, need more multiple

Nicolas Pallant 28785959

LO 4: TEST THE KEY FEATURES OF PROGRAMMING LANGUAGES WHICH ARE COMMONLY USED FOR DEVELOPING WEB-DATABASE APPLICATION

The following pieces demonstrate my ability in relation to this LO:

- Task 2.1 CSS/Bootstrap/HTML
- Task 2.2 SQL
- Task 3.1 C# Views
- Task 4.1 SQL Models and C# Views
- Task 4.2 C# Models and C# Views
- Task 5.1 Javascript

LO 5: ASSESS THE MVC DESIGN PATTERN AND CONSTRUCT A WEB-DATABASE APPLICATION USING THE MVC DESIGN PATTERN

The following pieces demonstrate my ability in relation to this LO:

- Task 2.2 First time development of a MVC application
- Task 3.1 Using C# in Views within a MVC application
- Task 4.1 Using the Model First approach in a MVC application
- Task 4.2 Using the Code First approach in a MVC application
- Task 5.1 Using Javascript within Views in a MVC application
- Task 5.2 Using Bootstrap Datepickers within Views in a MVC application
- Task 6.1 Validation in Models in a MVC application
- Task 6.2 Need for jQuery unobtrusive validation in ViewModels of a MVC application
- Task 8.2 Using ASP.NET Identity within Views of a MVC application
- Task 11.1 Hosting a MVC application using Azure Cloud
- Task 11.2 Unit testing of ASP.NET MVC Applications

LO 6: APPLY, ANALYSE AND CRITIQUE A PROFESSIONAL APPROACH TOWARDS THE DEVELOPMENT OF WEB-DATABASE APPLICATIONS

The following pieces demonstrate my ability in relation to this LO:

- Task 1.2 Analyzing of professional software such as Visual Studio and other IDEs
- Task 2.1 The need for Bootstrap as professional applications within Web Applications
- Task 3.2 The importance of version control within web app development
- Task 5.1 The differences between async and defer and when to use them
- Task 6.1 The importance of validation within web applications
- Task 6.2 The need for jQuery unobtrusive validation within web applications
- Task 8.2 The benefits and detriments to using a third-party emailing software such as SendGrid
- Task 9.1 The need for responsive images within web applications
- Task 9.2 The new features of ASP.NET Core and the benefits it has over ASP.NET Framework when it comes to profession web applications
- Task 11.1 Professional practices with Continuous integration and Continuous deployment and the need for both
- Task 11.2 Profession software testing approaches and their importance in development of web-applications

Nicolas Pallant 28785959

REFLECTION

THE MOST IMPORTANT THINGS I LEARNT:

The most important things I learnt along the way of this unit is the need for back end developers within the workplace and the complexity that comes with it. While I have learned a lot about web application development, I feel like I have a lot to learn and adapt to within the industry.

Another important thing I learnt is the need and importance of reducing the amount of time a client needs to wait within web applications and the needs for things such as .woff/.woff2 and jQuery unobtrusive validation.

Before this unit, I didn't know anything at all when it came to web development, no HTML, CSS, Javascript, and now I feel like I can and have developed functional web applications which is a fantastic feeling, even if it takes me a super long time.

THE THINGS THAT HELPED ME MOST WERE:

- My Teaching Associates: Kevin and Roshan were immensely helpful in my learning journey for this unit, and whenever I didn't understand how to do something, or was hitting my head against the keyboard on something, they'd show me exactly where to go to find out.
- Studio Tasks: The studio tasks were instrumental in my learning for this unit, as actually applying the theory we learnt in the lectures and pre-class is my best way to learn
- Studio Extension Tasks: The studio extension tasks helped me achieve many of the learning objectives in this unit as well as many of the business requirements in the assignment.
- Google: While all of the above was a great starting point, whenever I was unsure about how to approach something, google was a fantastic research with websites like StackExchange. However this was a double-edged sword, as a lot of the solutions were above my skill level.

I FOUND THE FOLLOWING TOPICS PARTICULARLY CHALLENGING:

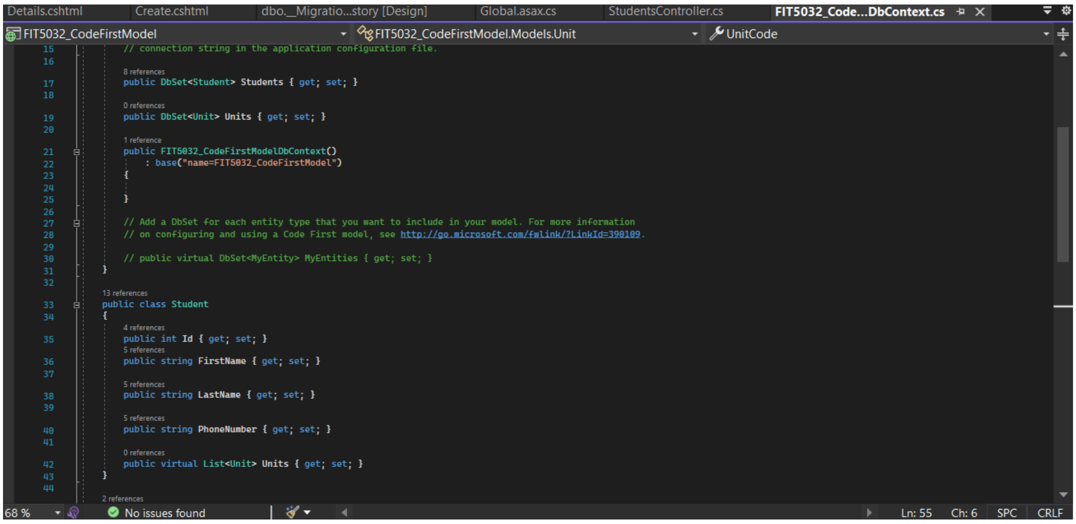
- Lambda Expressions: I find Lambda expressions super hard to approach when compared to SQL statements, and even then I'm not the best with SQL.
- HTML: Before this unit, I had done no HTML whatsoever, so being thrown in the deep end and having to learn it with the rest of everything was a very hard challenge.
- CSS: Again, as with HTML, I hadn't touched any CSS in my life, so picking it up and knowing when it use it was challenging to me.
- Design: I am not the most creative person and coding is my forte, so anything design related and making stuff look nice was very hard for me.

I FOUND THE FOLLOWING TOPICS PARTICULARLY INTERESTING:

- Code First Approach to MVC: As a game developer and teaching associate, I felt very at home with C# and using code to link my databases, so I clung to anything C# I could use which made it one of the most interesting topics to me.
- Validation: I've always known validation was important, but I had no idea the impact it could have on applications if done wrongly, which made it much more interesting.

I FEEL I LEARNT THESE TOPICS, CONCEPTS, AND/OR TOOLS REALLY WELL:

- C#: As I mentioned before, as a Games Developer and TA, I found C# very familiar to me and easy to use, and hence allowed me to create much more rebust applications when it came to using it, such as in my Models and Controllers.



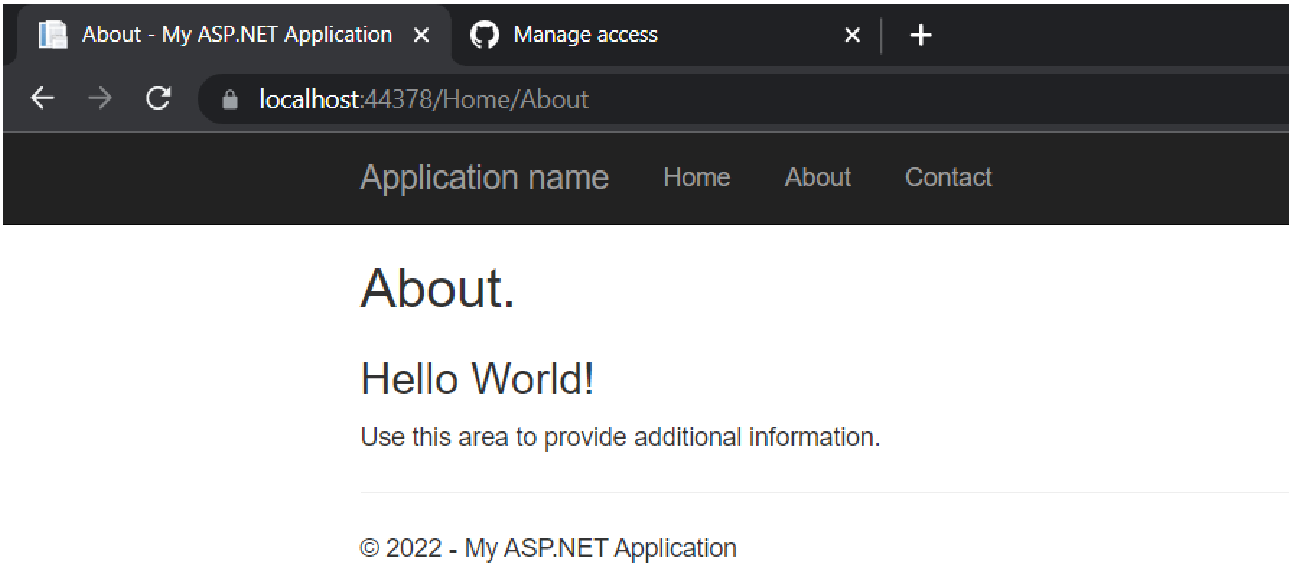
```
        // connection string in the application configuration file.

        8 references
        public DbSet<Student> Students { get; set; }

        0 references
        public DbSet<Unit> Units { get; set; }

        1 reference
        public FIT5032_CodeFirstModelDbContext()
            : base("name=FIT5032_CodeFirstModel")
        {

        }

        // Add a DbSet for each entity type that you want to include in your model. For more information
        // on configuring and using a Code First model, see http://go.microsoft.com/fwlink/?LinkId=390109.

        // public virtual DbSet<MyEntity> MyEntities { get; set; }
    }

    13 references
    public class Student
    {
        4 references
        public int Id { get; set; }
        5 references
        public string FirstName { get; set; }

        5 references
        public string LastName { get; set; }

        5 references
        public string PhoneNumber { get; set; }

        0 references
        public virtual List<Unit> Units { get; set; }
    }

    2 references
```

(Task 4.2 from eFolio)

The image above shows one of the only screenshots from my eFolio work that had any C# in it. And while the C# above isn't anything impressive, I feel like I am much better at it than anything else in the unit

I STILL NEED TO WORK ON THE FOLLOWING AREAS:

- HTML: As said before, I am still very new to HTML, so I need to learn a lot more to become proficient in it, as I still have to refer to notes on what things like src and div do.



About.

Hello World!

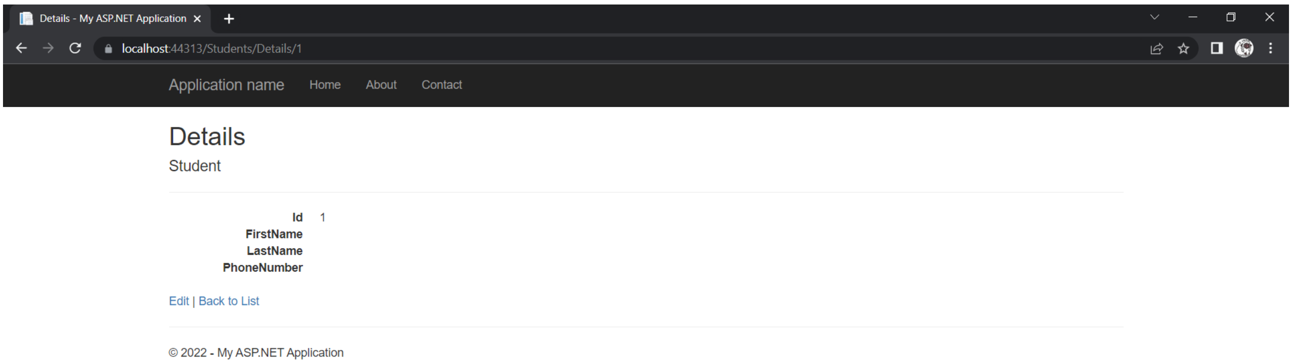
Use this area to provide additional information.

© 2022 - My ASP.NET Application

(Task 3.1 from eFolio)

As shown by the image above, I really haven't done anything to the default application because I really have no idea how to make it look any nicer, so I need to do a lot of work with HTML.

- CSS: Again as before, I am very new to CSS, so much more work is needed to become proficient, and I'm really bad at making things look nice.


Details - My ASP.NET Application
localhost:44313/Students/Details/1
Application name
Home
About
Contact

Details

Student

Id	1
FirstName	
LastName	
PhoneNumber	

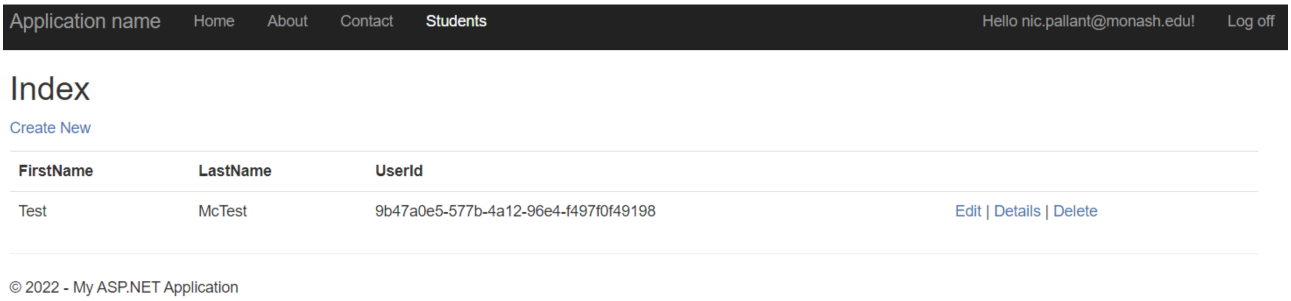
Edit | Back to List

© 2022 - My ASP.NET Application

(Task 4.2 from eFolio)

As shown above, again I have left everything basic, and not applied any kind of additional CSS to the elements of the page.

- Design: Like I said before, again… I am not the most creative person in the world, and am solidly a developer through and through, so I really need to work on designing things, which is very hard for me.



Index

Create New

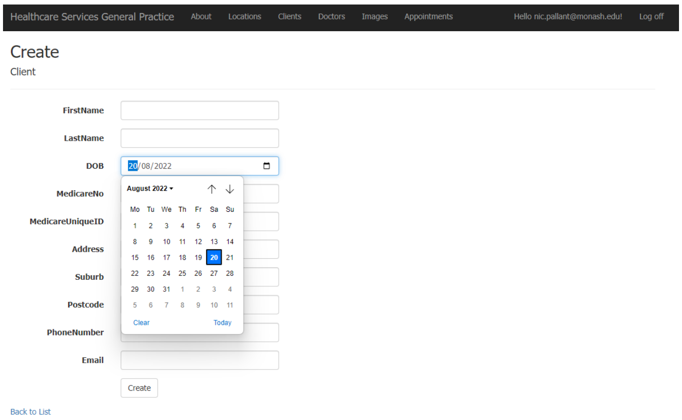
FirstName	LastName	UserId	
Test	McTest	9b47a0e5-577b-4a12-96e4-f497f0f49198	Edit | Details | Delete

© 2022 - My ASP.NET Application

(Task 7.2 from eFolio)

As shown, this table has no additional formatting to make it look any nicer, and things like the First Name and Last name are still joined, which I could have separated.

- Javascript: While I am better at Javascript than HTML and CSS, I still struggle with it because the debugger never seems to function with it, making it hard to tell what's going on.


Healthcare Services General Practice
About
Locations
Clients
Doctors
Images
Appointments
Hello nic.pallant@monash.edu!
Log off

Create

Client

FirstName	
LastName	
DOB	20/08/2022
MedicareNo	
MedicareUniqueID	
Address	
Suburb	
Postcode	
PhoneNumber	
Email	

August 2022

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

Clear Today

Create

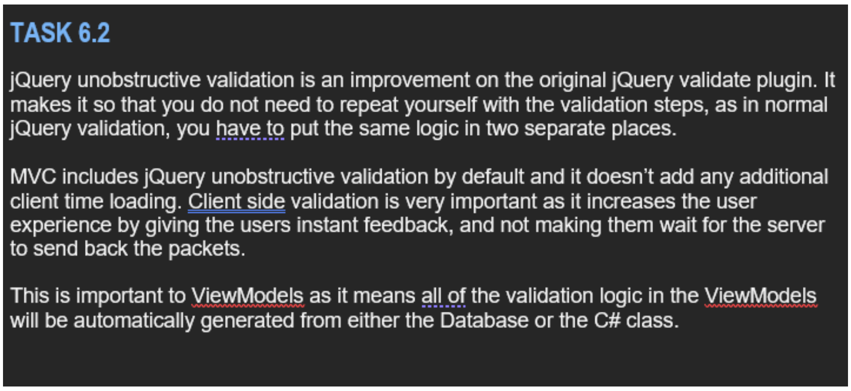
Back to List

(Task 5.2 from eFolio)

Getting the Bootstrap Datepicker to work for me took hours and hours of trying different things until I got it to work, when it really should have taken 5 minutes.

Nicolas Pallant 28785959

MY PROGRESS IN THIS UNIT WAS …:



TASK 6.2

jQuery unobstructive validation is an improvement on the original jQuery validate plugin. It makes it so that you do not need to repeat yourself with the validation steps, as in normal jQuery validation, you have to put the same logic in two separate places.

MVC includes jQuery unobstructive validation by default and it doesn't add any additional client time loading. Client side validation is very important as it increases the user experience by giving the users instant feedback, and not making them wait for the server to send back the packets.

This is important to ViewModels as it means all of the validation logic in the ViewModels will be automatically generated from either the Database or the C# class.

(Task 6.2 from eFolio)

My progress in this unit was very slow when it came to learning anything new in web development, but I feel as if my written content on theory is much better as I can understand the content based on my previous experience and other knowledge, whereas all of the actual development is almost entirely new to me.

THIS UNIT WILL HELP ME IN THE FUTURE:

The content I learnt about things like Validation is super useful when it comes to taking user input, as it allows for significantly less errors.

On top of that, I feel like all the new content I learnt about accessibility can easily be applied in developing and teaching games, so I feel like that is very valuable for me in the future

IF I DID THIS UNIT AGAIN I WOULD DO THE FOLLOWING THINGS DIFFERENTLY:

- Give it more time: I feel as if I didn't give this unit the time it deserved as I was super busy with being a Teaching Associate for 3 different units and being the Admin TA for one of them, while developing lectures and labs for another one. I feel like if I spent more time on this unit, and the assignment, I could have done a lot better in my development.
- Attend lectures: As I said above, I really didn't have a lot of time, and the lectures overlapped with one of the classes I had to teach, and catching up on them just never became an option because I was so busy. So I really would have liked to attend in person to the lectures and actively participate.

OTHER…:

When I initially started this unit, I had no clue whatsoever of the first thing when it came to website development, let alone integrating it with databases. While I definitely could have done some things better, and struggled in making it look nice, I am very proud of what I have accomplished in the creation of my final website. I feel as if I had to do it again, I would be much more proficient, and maybe it wouldn't take me 60 hours this time. Thanks for the unit everyone, Russell, Kevin, Roshan, you're all great.

Declaration

I declare that this learning summary, eFolio tasks and the linked code are my individual work except group submission. I have not copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text and code, nor has any part of this submission been written for me by another person.

Signature: N Pallant