FIT5032 Learning Summary
Vincent: 3005286

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
    a. Run Visual Studio Community Edition IDE.
2. Task 1.2
    a. Research and list five different IDEs currently used in the Agile software development industry.
    b. Briefly discuss five core features of Visual Studio Community Edition as an IDE.
3. Task 2.1
    a. CSS example.
    b. Demonstrate responsiveness using Bootstrap.
4. Task 2.2
    a. Database First Development of ASP.NET MVC Web Application using Visual Studio Scaffolding.
5. Task 3.1
    a. Introduction to C#.
6. Task 3.2
    a. Version control.
7. Task 4.1
    a. Model First Development of ASP.NET MVC Web Application using Visual Studio Scaffolding.
8. Task 4.2
    a. Code First Development of ASP.NET MVC Web Application.
9. Task 5.1
    a. JavaScript example – Notify.js.
10. Task 5.2
    a. Use of Bootstrap datepicker.
11. Task 6.1
    a. Application of View Model.
    b. Using Data Annotation for view model validation.
12. Task 6.2
    a. Understand jQuery unobtrusive validation.
13. Task 7.1
    a. Security Issues and Countermeasures.
    b. Using ASP.NET Identity.
14. Task 7.2
    a. Setup MS Identity to sign in with Google Account.
15. Task 8.1
    a. Obtain API key from SendGrid.
    b. Integrate SendGrid into ASP.NET application for sending emails.
16. Task 8.2
    a. Research and understand the key features of 3rd party emailing tools.
    b. Briefly discuss the advantages and disadvantages of using a 3rd party emailing tool.

17. Task 9.1
    a. Optimize my project.
18. Task 9.2
    a. Introduction to .NET Core.
19. Task 10.1
    a. Creating First Web API using .NET Core.
20. Task 10.2
    a. Compare and contrast CSR and SSR.

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.

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 1.1 Run Visual Studio Community Edition IDE can be used as an introductory step to familiarize myself with one of the current widely used IDE (i.e., Visual Studio). Although ASP.NET Framework has been replaced by .NET 6 and eventually .NET 8 on November 2023, its application is the best starting point in web applications development because other IDEs or current web-technology has almost similar patterns and structures to ASP.NET Framework in developing a web application.

- Task 1.2 Research and list five different IDEs currently used in the Agile software development industry. This task directly addresses the historical and current landscape of web development tools. Researching and listing various IDEs used in Agile development gives myself a sense of how the field has evolved and adapted to changing development methodologies. It helps me understand the historical context of web development tools and their relevance in the current Agile development industry as well as advantages and disadvantages for each of them.

LO 2: Design, construct and publish web-database applications.

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

- Task 2.1 CSS example and Demonstrate responsiveness using Bootstrap. CSS and responsiveness are essential for designing the appearance and layout of web applications as well as maintaining the design in different screen sizes. This task helps me to learn how to style and design web interfaces, which is a fundamental aspect of web development. Although there are a lot of CDNs nowadays that can instantly design web applications, understanding CSS and responsiveness have helped me to create my own desired design for my final project.
- Task 2.2 Database First Development of ASP.NET MVC Web Application using Visual Studio Scaffolding. This task directly addresses the construction of a web-database application using the Database First approach in ASP.NET MVC which I later used as my main approach when constructing my final project.
- Task 3.1 Introduction to C#. Learning C# is important as it's commonly used in web application development and a prerequisite for constructing web-database applications.
- Task 3.2. Version control. Version control is crucial for collaborative development, making it an important part of the application construction process.
- Task 4.1 Model First Development of ASP.NET MVC Web Application using Visual Studio Scaffolding. This task focuses on constructing a web application with an emphasis on the model layer, which is another key aspect of web-database application development after Task 2.2 above.
- Task 4.2 Code First Development of ASP.NET MVC Web Application. This task also addresses web application construction, but it focuses on the Code First approach, providing an alternative perspective on application development.

- Task 5.1 JavaScript example – Notify.js. Understanding and using JavaScript is crucial for enhancing the functionality of web applications especially in giving feedback (Norman principle) after a user action within my web application, making this a relevant component of application design and construction.
- Task 5.2 Use of Bootstrap datepicker: Bootstrap is a popular framework for web design, and using its datepicker component demonstrates how to leverage tools and libraries in web application development. By doing this task, I managed to implement it in the booking constraint business requirement of my final project.
- Task 6.1 Application of View Model and Using Data Annotation for View Model Validation. This task demonstrates how to construct a view model in the situation where I need to display information from multiple tables in my database and only need to aggregate the necessary data. This allows me to retrieve and present the data efficiently without directly exposing the complexity of the underlying database structure to the view. For the data annotation for view model validation, it ensures that the data submitted by users is accurate, safe, and conforms to the requirements of the application. This task is also directly relevant to constructing web-database applications as it involves validating data within the view model. Task 6.2. Understand JQuery unobtrusive validation. Understanding this concept gives me an option in validating users' input that align to a principle of creating websites or web applications that do not disrupt upon the user's experience.

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 7.1 Security Issues and Countermeasures. This task demonstrates the critical issue of security in web-database applications such as XSS, SQL injection, and other malicious attacks which demonstrates the key technological issues nowadays that my web applications must be secured all the time.
- Task 7.2 Setup MS Identity to sign in with Google Account. This task directly engages with the challenges of user authentication and authorization. When setting up MS Identity to sign in with Google Account, I need to complete a tedious process such as creating credentials for OAuth Client ID before successfully implementing it which demonstrates the key technological issues confronting developers building web-database applications.
- Task 8.1 Obtain API key from SendGrid. Analyzing this process demonstrates the security and privacy considerations when connecting to external services and handling sensitive data such as email content. It also emphasizes the importance of securing API keys.

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 9.1 Optimize my project. By analyzing and improving the project's performance, this task demonstrates that I will likely delve into Language-specific optimizations. This task encourages developers to explore the language's capabilities for optimization.
- Task 9.2 Introduction to .NET Core. Learning about .NET Core, a cross-platform framework for building web applications, involves understanding its programming languages such as C#. Analyzing .NET Core's features and how they relate to web-database application development aligns with the objective of testing programming language features.
- Task 10.1 Creating First Web API using .NET Core. By creating a web API, I can assess how the chosen language and framework handle web-database interactions, data serialization, routing, and other critical aspects of web development.
- Task 10.2 Compare and contrast CSR and SSR. This task is essential for testing and evaluating programming languages because it involves comparing client-side rendering (CSR) and server-side rendering (SSR) techniques.

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 Database First Development of ASP.NET MVC Web Application using Visual Studio Scaffolding. This task involves constructing a web application using the ASP.NET MVC framework. It focuses on a database-first development approach, which is one of the techniques used to implement the MVC pattern.
- Task 4.1 Model First Development of ASP.NET MVC Web Application using Visual Studio Scaffolding. Similar to Task 2.2, this task is about constructing a web application using the ASP.NET MVC framework but emphasizes a model-first development approach. It demonstrates a different way of applying the MVC pattern, where you start with designing the models before creating views and controllers.
- Task 4.2 Code First Development of ASP.NET MVC Web Application. This task introduces another development approach, Code First, which is closely related to the MVC pattern. It focuses on creating models in code and then generating the database schema from those models.

LO 6: Apply, analyse, and critique a professional approach towards the development of web-database applications, considering data privacy and cybersecurity aspects.

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

- Task 7.1 Security Issues and Countermeasures. This task demonstrates the critical issue of security in web-database applications such as XSS, SQL injection, and other malicious attacks.
- Task 8.1 Obtain API key from SendGrid. Analyzing this process demonstrates the security and privacy considerations when connecting to external services and handling sensitive data such as email content. It also emphasizes the importance of securing API keys.

Reflection

Reflect on your learning and discuss these areas. Read the suggestions in [ ] for each question. Write your reflections then delete the text in the [ ] and delete this box

The most important things I learnt:

Learning and creating web applications using ASP.NET Framework has been a transformative journey for me. This unit is very different from my Java or Python unit before because it helps me to understand HTTP and URLs to mastering server-side programming in C#. The first most important thing that I learnt is security practices, including user authentication and authorization. Learning to protect against common web vulnerabilities like SQL injection and XSS has been an eye-opener in terms of safeguarding user data. I did not expect that we will learn about security practices, but this experience has taught me that, I will need to be good at this topic in the future. Secondly, due to my lack of expertise in MVC development methodology and as I encountered challenges and learned to troubleshoot issues, my problem-solving and critical-thinking skills sharpened. The concept of the Model-View-Controller (MVC) architecture has transformed the way I structure my code, promoting organization, and maintainability. Finally, I also learned a lot about the use of APIs and CDNs in assisting our web applications development.

The things that helped me most were:

1. **Security Awareness**: Understanding web security and applying practices like user authentication, authorization, and protection against common vulnerabilities like SQL injection and XSS has been a vital skill, making my applications safer for users.

2. **Performance Optimization**: The knowledge gained in optimizing application performance through techniques such as minify, uglify, caching, using CDNs and reducing HTTP connection has led to faster and more efficient web applications in my final project.

I found the following topics particularly challenging:

1. **Security**: Understanding and implementing robust security measures, including user authentication, authorization, and protection against common vulnerabilities like cross-site scripting (XSS) and SQL injection, was complex. Although most of the security measures are automatically implemented by ASP.NET Framework through scaffolding feature, implementing it

by myself in the newly created view, controller, or model is very challenging to me.

I found the following topics particularly interesting:

1. **MVC Architecture**: The Model-View-Controller (MVC) architectural pattern has been a captivating concept. Understanding how it promotes code organization, separation of concerns, and maintainability has been a great experience. It's not just a practical approach but a fascinating one in terms of software design.

I feel I learnt these topics, concepts, and/or tools really well:

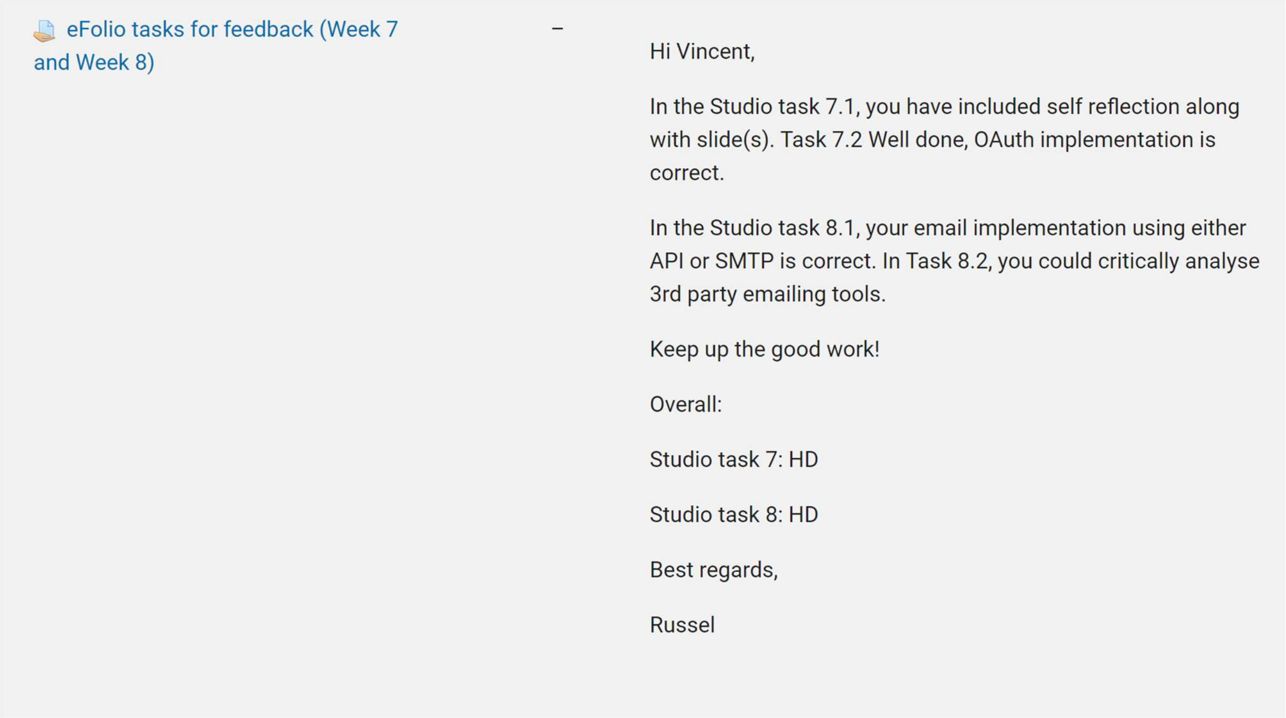
I have learnt most of the topics, concepts, and tools in this unit well. This can be proof from all the studio tasks completed as well as I managed to complete up to BR (F.1). One example is database interaction (Entity Framework). Since I choose Database First approach, I have acquired a deep understanding of how to interact with databases, particularly using Entity Framework. Data modelling, querying, and managing database connections have become my areas of expertise.

I still need to work on the following areas:

1. **DevOps and Cloud Deployment**: I had like to enhance my skills in DevOps practices, including infrastructure as code (IaC), and become more proficient in deploying applications to various cloud platforms like Azure, AWS, and Google Cloud.
2. **Security Practices** since I am not from cybersecurity background. I want my web application to be secured for the users.

My progress in this unit was …:

My progress in this unit was great and I learnt a lot of new things that I did not get from my previous coding unit. Furthermore, I also managed to complete the final project up to BR (F.1) and get HD to all my studio tasks except Task 6.2.



eFolio tasks for feedback (Week 7 and Week 8)

–

Hi Vincent,

In the Studio task 7.1, you have included self reflection along with slide(s). Task 7.2 Well done, OAuth implementation is correct.

In the Studio task 8.1, your email implementation using either API or SMTP is correct. In Task 8.2, you could critically analyse 3rd party emailing tools.

Keep up the good work!

Overall:

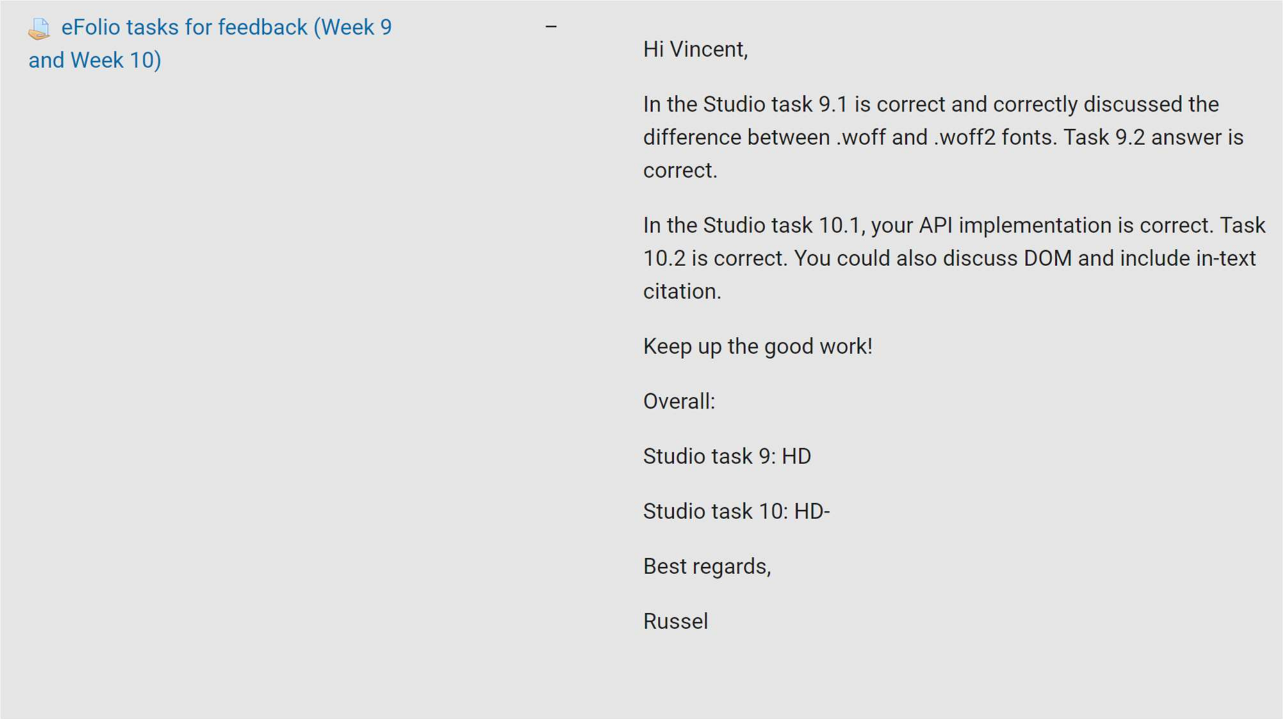
Studio task 7: HD

Studio task 8: HD

Best regards,

Russel

Figure 1. Tasks 7 and 8 Screenshot



eFolio tasks for feedback (Week 9 and Week 10)

–

Hi Vincent,

In the Studio task 9.1 is correct and correctly discussed the difference between .woff and .woff2 fonts. Task 9.2 answer is correct.

In the Studio task 10.1, your API implementation is correct. Task 10.2 is correct. You could also discuss DOM and include in-text citation.

Keep up the good work!

Overall:

Studio task 9: HD

Studio task 10: HD-

Best regards,

Russel

Figure 2. Tasks 9 and 10 Screenshot

This unit will help me in the future:

I think this unit will help me a lot in finding job especially my goal is to become a software engineer after graduation. One reason is that ASP.NET is widely used in the industry, particularly in enterprise-level applications. Consequently, there is a strong demand for developers with ASP.NET skills. Proficiency in ASP.NET can open doors to job opportunities and career growth. Moreover, ASP.NET can be used to build a wide range of applications, from web apps and APIs to mobile backends and more. This versatility means I can adapt to different roles and projects, making myself a valuable asset in the job market.

If I did this unit again I would do the following things differently:

If I did this unit again, I would finish my final project immediately after Russel posting week 5 post activities materials. I would like to improve my time management and task allocation planning skills as well as actively seek feedback through consultations.

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:______________________________________