**Portfolio**

Introduction:

My previous acquired knowledge regarding the subject of full stack app building is with C# and windows forms, as well as my experience with PHP and CSS.

The learning style I prefer is demand based because it allows me to choose my own path while still having some sort of structure that provides me with a sort of check point system to make sure I’m going on the right path.

I took upon myself to choose C# for the reasons listed in one of my research papers comparing asp net to other available options.

This portfolio aims to show how I successfully achieved the required learning outcomes needed.

Project Description:

My project is simply based off of me wanting to build something that meets the requirements and quality standards of a commercial level software product, that is also why I have chosen to have a stake holder with an aspiring business in jewellery to be.

Learning Outcomes:

LO 01: Web-Application:

What did you do to complete the task?

My Web Application is built using the React javascript framework and it is linked to the C# Asp net rest API,the web application communicates with the controllers of the API to retrieve the information (GET functions) or send information needed using the POST functions.

The Web Application homepage:

Graphical user interface

Description automatically generated

The Web Applications Register Page:

Graphical user interface, text, application, email

Description automatically generated

The WebApplications Login page:

Graphical user interface, text, application

Description automatically generated

The WebApplications Products Page:

Graphical user interface, application

Description automatically generated

The web applications Product page:

Graphical user interface

Description automatically generated

Example of error handling:

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

The use of a mediator to minify the direct interaction between different objects.

Graphical user interface

Description automatically generated with medium confidence

I have ensured the usage of persistence of data (by usage of ORM) through the usage of the entity framework which is essentially an ORM framework. Below is an example of the entity framework composition.

Graphical user interface, application

Description automatically generated

I have also used Migrations to update the database schema while preserving existing data in the database:

Text, letter

Description automatically generated

The use of asynchronous functions is clear in the controllers as an example the products controller following functions:

Text

Description automatically generated

What have you learned?

* I have learned how to Implement a C# API from scratch.
* I have learned how to create a mediator.
* I have learned how to make an SQL server interact with an API.
* I have learned how to use entity framework.
* I have learned how to create async functions for controllers
* I have learned how to communicate React javascript with a C# API
* I have learned the basics of React
* I have learned how to implement other APIs into my front end

What are you proud of?

LO 03: CI/CD:

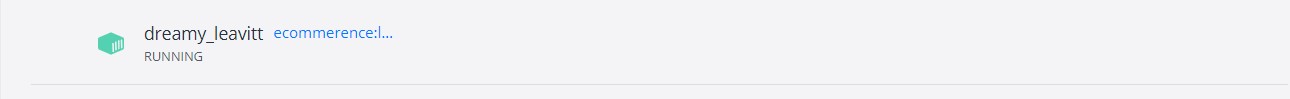
Docker:

As shown below in the containers tab in visual studio and after building the App through docker:

Graphical user interface, text, application, email

Description automatically generated

As shown below in the docker desktop app showing the running containers:



Github CI:

Below if proof of build through continuous integration through GITHUB CI and docker creating a CI/CD pipeline with the docker container, thus decreasing build time and optimize workflow.

Text

Description automatically generated

Further proof while pushing changes to github

Graphical user interface, text, application, email

Description automatically generated

LO 05: Professional:

What did you do to complete the task?

During the semester we set up ways in our group to schedule our tasks and organise them, the way I have seen it done in my own company where I work. These tasks where separated by a program called shortcut which allowed us to use agile sprints to create a short, timed schedule that provides the stakeholder with continuous updates and for group members to provide updates on their progress on each task.

Through this in my group project and personal project through my own stakeholder I have proved the professional learning object as can be seen with some of the shortcut tasks

Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated

I have also communicated with my colleagues by creating a whatsapp group that allows easier and more urgent contact between the different team members, replying to any help they might need and asking for help as well from them.

Graphical user interface, text, application, chat or text message

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

I also used the concept of user stories in my own personal project, using the git issues tab:

Graphical user interface, text, application, email, Teams

Description automatically generated