#### Weekly Report

#### To: Kevin Comba From: Brian Ndickers Date: 23-05-2024

#### RE: Concept Covered This The Week

## Introduction

This week task was to enhance my knowledge in typescript. The goal was to understand the core concepts of typescript to enhance code quality and maintainability in our projects. We covered these topics throughout the week, introduction to typescript, variable and special types, object, array, and tuples, union and intersection types, interfaces, type aliases, functions and callbacks, promises, type casting and omit utility type. This report is to provide a summary of the topics learned over the week.

## Discussion

I learned various types in typescript, primitive types. Additionally, I explored special types which help in defining more precise function signatures and handling exceptional cases.

We also covered how objects and array can be defined with specific properties and their types. How arrays can be typed to hold specific types of elements and tuples provide a way to represent an array with a fixed number of elements.

How union types provide flexibility and precision in type definitions by allowing a variable to hold values of different types, while intersection types enable combining multiple types into one.

We learned how interfaces define the shape of an object, describing what properties it should have and their types. They are useful for ensuring that different parts of an application adhere to specific contracts.

How type aliases allow creating a new name for a type which is useful for simplifying complex type definitions and improving code readability.

Also, how typescript provides a way to define function signatures, including the types of parameters and return values and how typescript ensures that these callbacks adhere to specified types.

Typescript enhances the use of promises by enabling type definitions for the values a promise can resolve or reject and how type casting is used to override the inferred type of a variable.

We end the week with learning, omit utility type which allows creating a new type by excluding one or more properties from an existing type.

## Conclusion

This week, I gained a comprehensive understanding of typescript, including its type system and various advanced features. By learning about variables, special types, objects, arrays, tuples, unions, intersections, interfaces, type aliases, functions, callbacks, promises, type casting, and the omit utility type, I am now better equipped to write more robust and maintainable code.