**Node.js**

Task2

**TypeScript Generics**

Generics are a built-in feature in type script that helps in identifying the type of the object, since TypeScript is a strongly typed language. This feature helps in case an input was typed generically yet still contains a type, for example:

**Let test = ‘’testing’’**

In this case the variable test was not written with a type, however it will be identified with type string, since the variable hold a string.

As mentioned that generics can identify types, this provides flexibility to reuse the code through setting a variable as a type.

Function **input <** type **> ( x :** type **) :** type **{**

Return **x**

in this case the generics feature will automatically identify the type of variable, as a result if a string is set to the variable, it will be set with type string and if a number was set, will be of type number. This function however will not respond to arrays, so if the function was set to receive arrays, then the “type” should be written as “type[]” and it will identify the type of the array based on its inputs.

**References**

Documentation - generics (no date) TypeScript. Microsoft. Available at: <https://www.typescriptlang.org/docs/handbook/2/generics.html>.(Accessed: February 9, 2023).

TypeScript Basic Generics (no date) Typescript basic generics. W3Schools. Available at: <https://www.w3schools.com/typescript/typescript_basic_generics.php>.(Accessed: February 9, 2023).