

## Diploma Advanced Database Task 2

### Set Up:

This task is dependent on the prior completion of Diploma Advanced Database Task 2 (TSQL)

### Background:

You have just completed the TSQL work allocated by your 'Dod & Gy Technology Wholesalers' team lead in 'Diploma Advanced Database Task 1'

Now you have been allocated the task of building a .net core api (REST API using JSON) to call the stored procedures you previously completed. Some endpoints are required for which there are no stored procedures, and in these cases you can use direct database access / queries to service the endpoint.

The API endpoints required by the front-end developers are detailed in the table below:

| Endpoint          | Http Method | Stored Procedure                       |
|-------------------|-------------|--|
| /location         | POST        | ADD_LOCATION                           |
| /location/id      | GET         | GET_LOCATION_BY_ID                     |
| /location         | GET         | NA – Lists all Locations (all columns) |
| /product          | POST        | ADD_PRODUCT                            |
| /product/id       | GET         | GET_PRODUCT_BY_ID                      |
| /product          | GET         | NA – Lists all Products                |
| /purchase         | POST        | PURCHASE_STOCK                         |
| /account          | POST        | ADD_CLIENT_ACCOUNT                     |
| /account/id       | GET         | GET_CLIENT_ACCOUNT_BY_ID               |
| /account          | GET         | NA – Lists all client accounts         |
| /authorisedperson | POST        | ADD_AUTHORISED_PERSON                  |
| /payment          | POST        | MAKE_ACCOUNT_PAYMENT                   |
| /order            | POST        | CREATE_ORDER                           |
| /order            | GET         | GET_OPEN_ORDERS                        |
| /order/id         | GET         | GET_ORDER_BY_ID                        |
| /order            | PUT         | FULLFILL_ORDER                         |
| /orderline        | POST        | ADD_PPRODUCT_TO_ORDER                  |
| /orderline        | DELETE      | REMOVE_PRODUCT_FROM_ORDER              |

The only requirements are that you use .net core and that you provide the specified endpoints, within that scope you can proceed however you see fit.

If you need to retrospectively 'tweak' a stored procedure to make it work, you can do that.

You can use entity framework (or not) or use a mix of EF and other methods.

Deliver the required functionality in .net core, the rest is up to you.

HINT: Keep in mind that you MAY need to **deploy** both the TSQL and API in the future.