# **FortisureIT Pre-Employment Project**

# Introduction

Completion of this project is a requirement to seek full-time employment at FortisureIT. The goal of this project is to evaluate aptitude in the types of work that you as an employee will regularly be a part of. This project contains focuses on programming skills related primarily to python. For this project, the dataset will be provided to you along with directions for installing any necessary tools to complete the project.

# Prerequisites

The tools needed to complete the project are as follows:

* FortisureIT Pre-Employment Sales Data.xlsx
* Python – Which can be downloaded [here.](https://www.python.org/downloads/)

# Source Data

Source data will come in the form of the FortisureIT Pre-Employment Sales Data.xlsx workbook. This is a structured dataset that consists of the following 5 tabs:

* Sales Order Detail – This tab contains an itemized list of each line item that was involved in a purchase. There may be multiple line items per sale depending on the number of different products purchased.
* Sales Order Header – This tab contains a summarized list of all purchases made by customers. There will be one line item per sale.
* Sales Reason – A list of the 10 reasons that a sale was made.
* Sales Order Header w Reason – Lists the reason a sale was made associated with a particular sales order. There may be more than 1 sales reason per sales order.
* Sales Territory – Lists the 10 different sales territories.

# Task #1 – Data Cleansing

There have been 3 major issues identified in the Excel data. Each of these issues should be addressed and fixed programmatically using Python logic.

1. The values in the Unit Price column on the Sales Order Detail tab have been mistakenly doubled for product id 709. The unit price for each line item for product 709 should be halved in your python script.
2. 500 duplicate records have been added to the Sales Order Header tab. Use python to remove the duplicate records so that there is a unique Sales Order ID per each line item.
3. All dollar amounts must be rounded to the nearest hundredth decimal place in the final output.

# Task #2 - Validation

The successful resolution of these data issues can be validated with the following measures below. Please add these validation measures in your python script.

1. On Sales Order Detail, the Unit Price multiplied by the Order Qty should equal the Line Total.
2. The grand total of the Line Total column from Sales Order Detail should match with the grand total of the Sub Total column from Sales Order Header when rounded to the nearest dollar.

# Task #3 – Parameterization

This process will be run each week, and there may be different processes that need to be run and different products that need their Unit Price adjusted. In order to account for this, please include the following in your python script.

1. User input to determine which process should run (updating unit price, de-duplicating Sales Order Header records, rounding dollar amounts).
2. User input to determine the product id for updating Unit Price.
3. A date parameter should be added that will be appended to the end of the output file name to show when the file was created.

# Task #4 – Output

Once the changes to the data have been made and validated, output the results from Python to a new Excel workbook title *Your Name –* FIT Sales Data *Date*.xlsx. Each tab should be named the same as it were in the original file. There should also be a new tab added that shows the functions that were run in the python script to produce this output as well as any input parameters or variables.