**Users Guide: the datarepresentation Project, 2019**

**Francis Adepoju (G00364694)**

**Project FOLDER in GitHub**:

* **drProjects –** All works related to the Data Representation module are stored in this folder inside the datarepresentation repository, named as: <https://github.com/dewaledr/dataRepresentation>
* This repository hosts all files and documentation produced for the datarepresentation Project.
* In addition, The project is hosted on the free (pythonanywhere.com) cloud service which can be accessed <http://dewaledr.pythonanywhere.com/customerViewer.html>

**Relevant files:**

* customerDAO.py Data Access Object file
* customerServer.py Flask Server for this app
* customerViewer.html Single page WEB interface to customer records
* doCreateTable.py Script to create database table (customers)
* doInsertToTable.py Script to populate database table with sample records
* dbCONFIG.py Database Configuration file
* requirements.txt Requirements Python 3.7.4, Flask 1.1.1, etc…
* runit Script to run the Flask server locally in DEBUG mode
* README.md GitHub readme file

**To run the WEB app, please:**

* Ensure that a modern browser like Chrome or Firefox is available on your machine. [Python/Visual Studio Code] might also be needed at some point to make changes if needed.

**Please NOTE:**

* Amongst other validations, the database/server will reject the JSON object during customer creation via the Web interface if the Total Amount Spent is not double.
* Update on any customer record will not happen if any of the input fields are empty
* Flask Server and MySQL database needs to be configured on your machine in order to run this application locally.
* The Configuration file also needs to be modified to suit local preferences.- Ability to interact with MySQL is very important for all CRUD operations in this project.
* It is also very important to have a good understanding of HTML, Javascript, AJAX, JSON Objects
* A basic understanding of RESTful API and Server are needed.
* **Aside:**
  1. A small shell script named ***runit*** is included to start the flask server and enable error logging
  2. A Python configuration file template is also included. This can be modified as a **config.py** file to hide the database login details
  3. All relevant files needed to run the project are stored in the same location inside the **drProjects** folder
  4. Ability to use **curl** for testing the Server interaction, **vi** for basic file editing and **GitHub** for versioning and file transfer are also very essential for this project