**Lend Invest Coding challenge – Bhuvaneshwari Ramakrishnan**

**Requirement 1:**

* Total Customer Spend: Created Roll up summary field to calculate the sum of Closed won opportunities.
* Customer classification – Created Formula field based on total Customer spend.

Created both the fields on account and chosen the standard salesforce functionalities to achieve better performance and efficiency.

**Requirement 2:**

* Created Trigger to check whenever account record gets updated and old record did not have GOLD in the customer classification field, and then the Twilio API is called.
* The CEO number information has been stored in Custom Settings for better handling by Admins. They would not have to change in the User object and verify the information every time the Number changes.
* The Authentication for the Twilio has been achieved through Named Credentials.
* The Named credentials have been created with the thought that the entire org would use a single credential or “to” number. Hence “Named Principle” Identity Type.
* The “from” number has been stored in another custom settings –twilioCreds, so that each time the phone number changes the admins can change the value from custom settings instead of changing in code.
* Along with that the AccountSID (Username) is also stored in the same custom setting – twilioCreds. This is useful when coding the end point in the callout as the sms endpoint required the AccountSID in the URL. In this way the code need not be modified and only the custom settings values need to be updated.
* The API is been called in future method since it gets invoked by trigger.

**Requirement 3:**

* The code to access the API and send SMS is the same as the one in Requirement 2.
* In the LWC, a text area has been created for the internal users to type in any message.
* The phone number of the customer is fetched by the current account record Phone field, by leveraging the LDS mechanism in LWC. This helps to avoid apex calls to fetch single record info.
* A toast message is displayed for success or error during the API callout and sending the message.
* The component has been added to the record page and visibility has been set in accordance with the Customer classification field.
* The LWC becomes visible only when the customer classification changes to GOLD.

**Note:**

* All of these functionalities have been achieved by creating a Twilio trial account.
* Test class has been created with mock API response. Code coverage achieved – 95%.
* Trigger has been handled using the Interface functionality. This ensures one trigger per object mechanism.

**Components and Classes:**

* LWC Component – goldCustomerSMS
* Classes:
  + Account trigger class – AccountTriggerClass
  + API Callout apex – TwilioAPIApex
  + API Utility – APICalloutsUtility
  + Test class - Test\_TwilioAPIApex
  + Mock HTTP Class- MockTwilioAPIResponse

**Things I would do differently / Future works:**

* Use flows instead of apex trigger to call the API. Better when processing large batches of data. This could be possible by making the API callout method in apex as Invocable. But for requirement 3, when using it with Aura, the aura enabled method must call the Invocable method. Since Invocable methods use List as parameters, Wrapper class can be used if other information needs to be added to the sms body.
* Use a separate test data utility class, which would be common for the org. In this way Test setup need not be one for each and every test class.