**Cognitica AI – Machine Learning Software Developer**

**Problem Statement:**

Build a CNN-LSTM model for Activity Recognition on the UCF-101 dataset. The dataset can be downloaded from the following link: <https://www.crcv.ucf.edu/data/UCF101.php> . The model should be built as follows:

1. Encoder is a CNN. The input image is given to CNN to extract the features. The last hidden state of the CNN is connected to the Decoder.
2. Decoder is LSTM. The first time step receives the encoded output from the encoder.
3. Add other layers as needed.

**The following steps and rubrics will be considered for evaluation:**

1. Data Processing:
   1. The candidate can **choose a subset** of the entire dataset for this task. For Example, **40 classes can be chosen for model training**.
   2. Convert the data into the correct format which could be used for the DL model.
   3. Plot at least two samples and their captions (use matplotlib/seaborn/any other library).
   4. Load the data into train and test data in the required format.
2. Model Building:
   1. Use any pretrained model or a custom-built model as CNN encoder for feature extraction.
   2. Create k-layered LSTM model and other relevant layers.
   3. Add one layer of dropout at the appropriate position and give reasons
   4. Choose the appropriate activation function for all the layers.
   5. Print the model summary
   6. Justify the choice of number of layers, activation function and any other hyper parameters used
3. Model Compilation:
   1. Compile the model with the appropriate loss function
   2. Use an appropriate optimizer
   3. Justify the choice of learning rate, optimizer, loss function and any other hyper parameter used.
4. Model Training:
   1. Train the model for an appropriate number of epochs.
   2. Print the train and validation loss for each epoch. Use the appropriate batch size.
   3. Plot the loss and accuracy history graphs for both train and validation set.
   4. Print the total time taken for training.
5. Model Evaluation:
   1. Take 5 random data from the test set and perform Expression recognition.
   2. Print confusion metrics and classification report for the test data.

**Submission Files:**

1. Candidates instructed use Google Colab or Jupyter notebook.
2. No need to upload data.
3. Candidates are expected to submit the Python notebook file (ipynb), the pdf of the notebook showing the outputs clearly.