Problem Statement

The volume of unstructured data (Text data, log lines, images, binary files) in existence is growing dramatically, and
MR, Spark are excellent framework for analyzing this type of data.

You will implement a MR application that calculates the most common words from Complete Works of William Shakespeare
Please refer a file Complete_Shakespeare.txt

Here are the brief steps for writing the word counting program:

1) Create a MR application which is going to calculates the most common words from Complete Works of William Shakespeare -
use 'Complete_Shakespeare.txt' file residing in your home on HDFS.
2) The most common words will be decided based on the stop words
3) Stop words are common words that are often uninteresting. For example "I", "the", "a" etc., are stop words.
You can remove many obvious stop words with a list of your own. But for this exercise, you will just remove
the stop words from a curated list `stop_words` provided to you in your environment (Refere a file stop_words.txt)
4) You will create following classes
    1. Outer job class acting as a Tool,
    having name as WordCountCompleteShakespere
    2. StopWordMapper class representing map task
    as inner class inside WordCountCompleteShakespere class
    3. StopWordReducer class representing reduce task
    as inner class inside WordCountCompleteShakespere class
5) Business demands that output should be generated in a folder
Shakespere_work on HDFS into 3 files
6) Business want you to run the application using following command only
yarn jar shakesperework.jar <Input path> <Output path>
Input path - Represents a path of a file/dataset on which the
Job/application is suppose to run
Output path - Represents a path of a output folder on HDFS
where the Job/application is suppose to put the final output
7) Automate the entire build process by writing down a shell script (This step is optional)

Timeline - 27 Oct 2025, EOD