Step 1

Acquire Data: I have downloaded the book form the link

https://ztcprep.com/library/story/Harry_Potter/Harry_Potter_(www.ztcprep.com).pdf

Step 2

Extract Data

1. **Select the Book:** My birth month is June (6), and thus according to the instructions, the book to be utilized is Book 6 (Harry Potter and the Half-Blood Prince).
2. **file1.txt:** my birthday is June 17, so file1.txt start from page 17 of Book 6 and pull 10 pages.
3. **file2.txt:** my birth year is 2001, so the page number is 101.Extract 10 pages from Book 6 from page 101

import the libraries

```
!pip install pyspellchecker
!pip install PyPDF2
!pip install fpdf
```

```
Requirement already satisfied: pyspellchecker in /usr/local/lib/python3.11/dist-packages (0.8.2)
Requirement already satisfied: PyPDF2 in /usr/local/lib/python3.11/dist-packages (3.0.1)
Requirement already satisfied: fpdf in /usr/local/lib/python3.11/dist-packages (1.7.2)
```

```
from PyPDF2 import PdfReader  # Extract text from PDF
import re  # Regular expressions for text processing
import pandas as pd  # Data manipulation and storage
from collections import Counter  # Count occurrences of words
from spellchecker import SpellChecker  # Identify non-English words
from fpdf import FPDF  # Generate PDF report
import matplotlib.pyplot as plt  # Generate graphs
```

```
PDF_PATH = "/content/Harry_Potter_(www.ztcprep.com).pdf"
FILE1_PATH = "file1.txt"
FILE2_PATH = "file2.txt"

# Define the book and pages based on birthdate 17/06/2001
BIRTH_MONTH = 6  # June
BIRTH_DATE = 17  # Day
BIRTH_YEAR = 2001  # Year

BOOK_NUMBER = 6  # Using Half-Blood Prince
PAGE1_START = BIRTH_DATE  # Extract pages 17-26
PAGE2_START = 101  # Extract pages 101-110

# Function to extract text
def extract_text_from_pdf(pdf_path, pages):
    reader = PdfReader(pdf_path)
    extracted_text = []
    for p in pages:
        if p <= len(reader.pages):
            text = reader.pages[p - 1].extract_text()
            if text:
                extracted_text.append(text)
    return "\n".join(extracted_text)

# Define pages
pages_file1 = list(range(PAGE1_START, PAGE1_START + 10))
pages_file2 = list(range(PAGE2_START, PAGE2_START + 10))

# Extract text
```

```
text_file1 = extract_text_from_pdf(PDF_PATH, pages_file1)
text_file2 = extract_text_from_pdf(PDF_PATH, pages_file2)

# Save text to files
with open(FILE1_PATH, "w", encoding="utf-8") as f:
    f.write(text_file1)
with open(FILE2_PATH, "w", encoding="utf-8") as f:
    f.write(text_file2)

print(f"Extracted text saved to {FILE1_PATH} and {FILE2_PATH}")
```

```
Extracted text saved to file1.txt and file2.txt
```

Step 3

1. Write Python code and use MapReduct to count occurrences of each word in the first text file (file.txt). How many times each word is repeated?

```
FILE1_PATH = "file1.txt"

# Function to tokenize text
def tokenize(text):
    text = text.lower()
    words = re.findall(r'\b\w+\b', text)
    return words

# Load text file
with open(FILE1_PATH, "r", encoding="utf-8") as f:
    text_file1 = f.read()

# Count word occurrences
words_file1 = tokenize(text_file1)
word_counts_file1 = Counter(words_file1)

# Convert to DataFrame
df_file1 = pd.DataFrame(word_counts_file1.items(), columns=["Word", "Count"]).sort_values(by="Count", ascending=False)

# Save word count to CSV
df_file1.to_csv("word_count.csv", index=False)

print("\nWord Count from file1.txt (All words):")
print(df_file1.to_string(index=False))
```

```
Word Count from file1.txt (All words):
        Word  Count
         the     64
          he     43
           a     40
          to     36
         and     34
          it     34
         was     30
           i     26
           s     25
           t     23
  dumbledore     23
         you     19
         his     19
   professor     18
          of     18
          on     17
         all     17
  mcgonagall     16
         she     16
        that     16
        said     15
         had     14
          be     14
          in     14
        know     13
          at     13
           y     13
```

```
       harry     13
          re     12
          ou     12
         for     12
         but     12
           e     11
        have     11
        they     11
         him     11
          as     10
         cat     10
     ztcprep     10
      potter     10
         www     10
         com     10
         can      9
         who      9
         her      9
        been      8
        name      7
         out      7
      saying      7
           d      7
    oldemort      6
      street      6
           j      6
           k      6
      people      6
```

Step 4

2. From the second text file (file2.txt), write Python code and use MapReduct to count how many times non-English words (names, places, spells etc.) were used. List those words and how many times each was repeated.

```
FILE2_PATH = "file2.txt"
OUTPUT_CSV = "non_english_words.csv"

# Initialize SpellChecker
spell = SpellChecker()

# Function to tokenize text
def tokenize(text):
    text = text.lower()
    words = re.findall(r'\b\w+\b', text)
    return words

# Load text from file2.txt
with open(FILE2_PATH, "r", encoding="utf-8") as f:
    text_file2 = f.read()

# Tokenize words
words_file2 = tokenize(text_file2)

# Identify non-English words using SpellChecker
non_english_words = [word for word in words_file2 if word not in spell]

# Count occurrences of non-English words
non_english_word_counts = Counter(non_english_words)

# Convert to DataFrame
df_file2 = pd.DataFrame(non_english_word_counts.items(), columns=["Non-English Word", "Count"]).sort_values(by="Count", ascending=False)

# Save results
df_file2.to_csv(OUTPUT_CSV, index=False)

print("\nNon-English Words from file2.txt (All words):")
print(df_file2.to_string(index=False))
```

```

Non-English Words from file2.txt (All words):
Non-English Word  Count
          hagrid     27
             ter     19
             www     10
         ztcprep     10
             yeh     10
```

```
                   ll      7
                ernon      6
                 didn      6
            gringotts      5
                   ap      3
                 stuf      3
                   ve      3
                 hadn      3
                albus      2
                  eah      2
               gettin      2
               izards      2
                 wasn      2
                knuts      2
            deliverin      1
                   69      1
                payin      1
                  teh      1
                   mm      1
               wouldn      1
                   70      1
                 cept      1
              fetchin      1
            everythin      1
                   68      1
               meself      1
                 ther      1
               diagon      1
                   67      1
                   ou      1
                   66      1
                 aren      1
              speakin      1
              shouldn      1
                   65      1
                insul      1
                 ying      1
              dumbled      1
                 goin      1
               muggle      1
                   64      1
               pposed      1
```

Step 5: PDF Extraction

```
# File paths
OUTPUT_WORD_COUNT = "word_count.csv"
OUTPUT_NON_ENGLISH = "non_english_words.csv"
OUTPUT_PDF = "MapReduce_Report.pdf"

# Load data
df_file1 = pd.read_csv(OUTPUT_WORD_COUNT)
df_file2 = pd.read_csv(OUTPUT_NON_ENGLISH)

# Initialize PDF
pdf = FPDF()
pdf.set_auto_page_break(auto=True, margin=15)
pdf.add_page()

pdf.set_font("Arial", size=14)
pdf.cell(200, 10, "MapReduce Word Analysis Report", ln=True, align="C")
pdf.ln(10)

# Word Count Section
pdf.set_font("Arial", size=12)
pdf.cell(200, 10, "Word Count Analysis from file1.txt", ln=True, align="L")
pdf.ln(5)

for index, row in df_file1.iterrows():  # Include all words
    pdf.cell(200, 10, f"{row['Word']} - {row['Count']}", ln=True)

pdf.ln(10)

# Non-English Words Section
pdf.cell(200, 10, "Non-English Words from file2.txt", ln=True, align="L")
pdf.ln(5)

for index, row in df_file2.iterrows():  # Include all non-English words
    pdf.cell(200, 10, f"{row['Non-English Word']} - {row['Count']}", ln=True)
```

```
        pdf.cell(200, 10, f"{row['Non English Word']}: {row['Count']}", ln=True)

    # Save PDF
    pdf.output(OUTPUT_PDF)
    print(f"\nPDF Report saved as: {OUTPUT_PDF}")
```

```
PDF Report saved as: MapReduce_Report.pdf
```

Start coding or generate with AI.