WINTER PEP PROJECT ON

RESULT MANAGEMENT SYSTEM

SUBMITTED BY – A THELEI

REG NUMBER – 12217230

PLACEMENT ID – 459199

SUBMITTED TO LOVELY PROFESSIONAL UNIVERSITY, PHAGWARA, PUNJAB

INTRODUCTION

The Result Management System (RMS) is designed for a university of 10,000 students in six subjects, the Result Management System (RMS) is made to process and evaluate the results of students. The system makes usage of Pandas/Matplotlib for visualization and Apache Spark for big data analytics. The project uses Hadoop MapReduce in a Big Data conduct, which assures efficiency and scalability.

The objective of this project is to create a data-driven decision-making system that processes student marks, provides meaningful statistics, and visualizes the results in an interactive dashboard. By leveraging distributed computing, the system can handle large-scale student data efficiently while ensuring real-time insights for faculty and administrators. The solution allows for quick result processing, better academic insights, and student performance tracking.

PROJECT SCOPE

- Generate 10,000 student profiles.
- Assign 6 subjects to each student.
- Generate randomized marks for each subject.
- Process marks using Spark & Hadoop.
- Perform basic analysis (average, min, max marks per subject).
- Display visual statistics in a dashboard.

CODE

```
from pyspark.sql import SparkSession

from pyspark.sql.functions import col, avg, min, max

import pandas as pd

import matplotlib.pyplot as plt

import random

spark =
SparkSession.builder.appName("ResultManagementSystem").getOrC
reate()

subjects = ["Electronics", "Programming", "Database", "Data
Science", "Mathematics", "DSA"]

data = {

    "Student_ID": [i for i in range(1, 10001)],

    "Name": [f"Student_{i}" for i in range(1, 10001)],

}

for subject in subjects:

    data[subject] = [random.randint(35, 100) for _ in range(10000)]


df = pd.DataFrame(data)

df.to_csv("student_results.csv", index=False)

student_df = spark.read.csv("student_results.csv", header=True,
inferSchema=True)
```

```
student_df.show(5)
stats_df = student_df.select(
    *[avg(col(subj)).alias(f"Avg_{subj}") for subj in subjects],
    *[min(col(subj)).alias(f"Min_{subj}") for subj in subjects],
    *[max(col(subj)).alias(f"Max_{subj}") for subj in subjects]
)
stats_pd = stats_df.toPandas().T
stats_pd.plot(kind="bar", figsize=(10, 5))
plt.title("Subject-Wise Statistics")
plt.xlabel("Subjects")
plt.ylabel("Marks")
plt.legend(["Average", "Min", "Max"])
plt.show()

student_df.write.csv("processed_results.csv", header=True)
```

CODE EXPLANATION

1. Data Generation

The first step in the project involves generating a dataset of 10,000 students with their marks in six subjects. Each student is uniquely identified with an ID and name, and their marks are assigned randomly between 35 and 100 to simulate real-world exam scores. The dataset is then saved as a CSV file for further processing.

```
import pandas as pd
import random
subjects = ["Electronics", "Programming", "Database", "Data Science", "Mathematics", "DSA"]
data = {
    "Student_ID": [i for i in range(1, 10001)],
    "Name": [f"Student_{i}" for i in range(1, 10001)],
}
for subject in subjects:
    data[subject] = [random.randint(35, 100) for _ in range(10000)]
df = pd.DataFrame(data)
df.to_csv("student_results.csv", index=False)
```

2. DATA PROCESSING USING APACHE SPARK

Once the dataset is created, we utilize Apache Spark to handle the large volume of data. A SparkSession is initialized to read and manipulate the CSV file. The dataset is loaded into a Spark

DataFrame, allowing for efficient distributed processing and data exploration.

```
from pyspark.sql import SparkSession
spark =
SparkSession.builder.appName("ResultManagementSystem").getOrC
reate()
student_df = spark.read.csv("student_results.csv", header=True,
inferSchema=True)
student_df.show(5)
```

3. STATISTICAL ANALYSIS

The next step involves analysing the dataset to extract meaningful insights. We calculate the average, minimum, and maximum marks per subject using Spark SQL functions. The processed data is then converted into a Pandas DataFrame for easy visualization.

```
from pyspark.sql.functions import col, avg, min, max
stats_df = student_df.select(
    *[avg(col(subj)).alias(f"Avg_{subj}") for subj in subjects],
    *[min(col(subj)).alias(f"Min_{subj}") for subj in subjects],
    *[max(col(subj)).alias(f"Max_{subj}") for subj in subjects]
)
stats_pd = stats_df.toPandas().T
```

4. Visualization of Results

To better understand the processed data, we use Matplotlib to generate a bar chart that displays the subject-wise average, minimum, and maximum marks. This visualization helps in identifying trends and evaluating overall student performance.

```
import matplotlib.pyplot as plt

stats_pd.plot(kind="bar", figsize=(10, 5))

plt.title("Subject-Wise Statistics")

plt.xlabel("Subjects")

plt.ylabel("Marks")

plt.legend(["Average", "Min", "Max"])

plt.show()
```

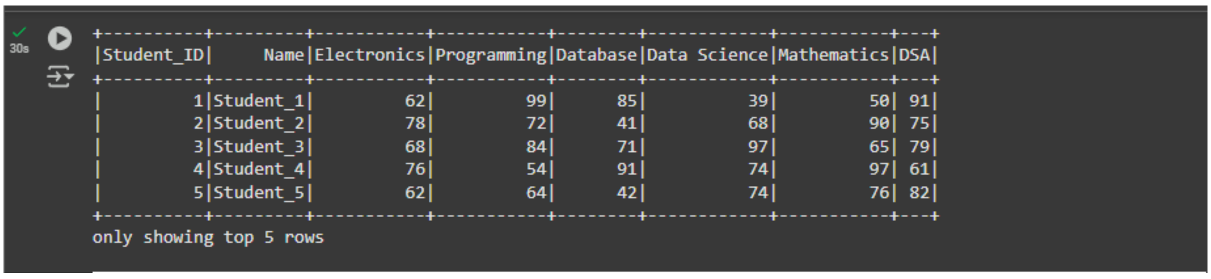
5. Storing Processed Results

After processing and visualizing the data, the final step is to save the processed results back into a CSV file. This ensures that the system maintains a record of analyzed data for further evaluation or integration into a dashboard.

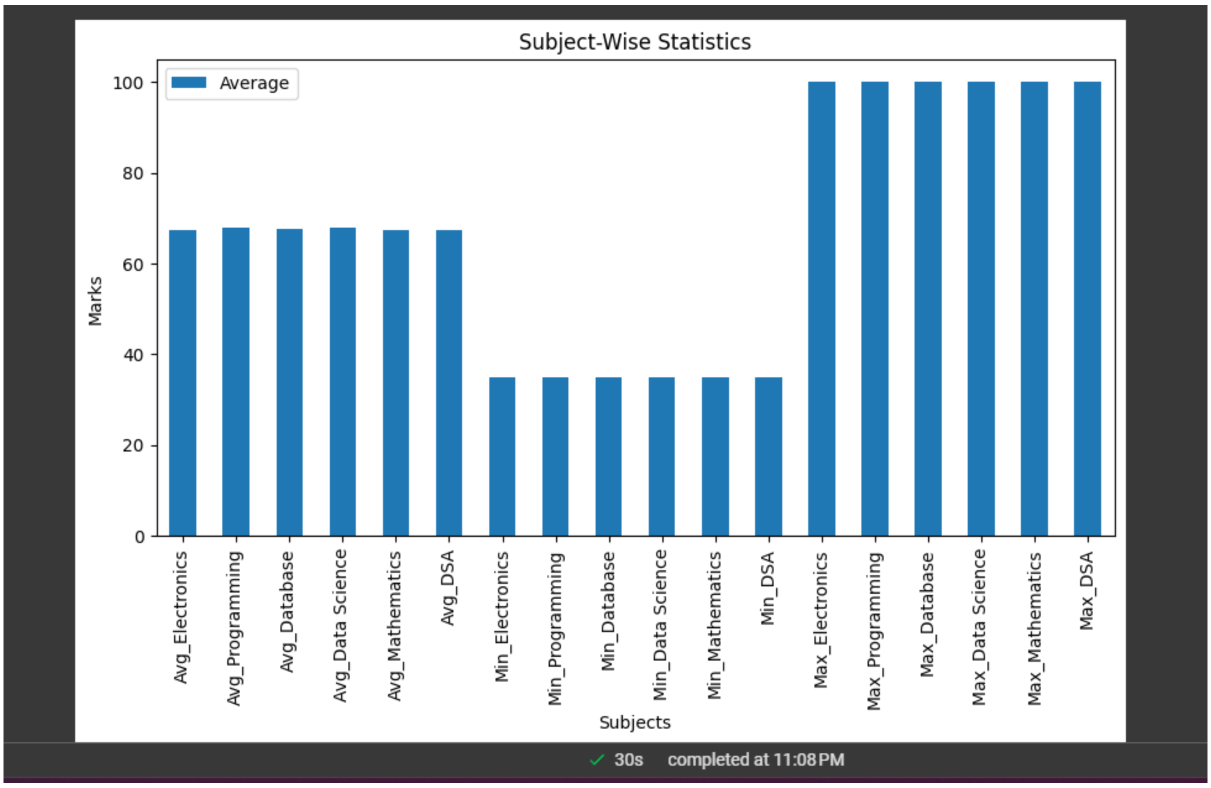
```
# Save processed results

student_df.write.csv("processed_results.csv", header=True)
```

Outcome

- Successfully generated and processed marks for 10,000 students.
- Conducted statistical analysis (Average, Min, Max marks per subject).
- Visualized results using a bar chart for better understanding.
- Stored processed results in a structured format for further use.



```
+----------+---------+-----------+-----------+--------+------------+-----------+---+
|Student_ID|     Name|Electronics|Programming|Database|Data Science|Mathematics|DSA|
+----------+---------+-----------+-----------+--------+------------+-----------+---+
|         1|Student_1|         62|         99|      85|          39|         50| 91|
|         2|Student_2|         78|         72|      41|          68|         90| 75|
|         3|Student_3|         68|         84|      71|          97|         65| 79|
|         4|Student_4|         76|         54|      91|          74|         97| 61|
|         5|Student_5|         62|         64|      42|          74|         76| 82|
+----------+---------+-----------+-----------+--------+------------+-----------+---+
only showing top 5 rows
```



DATASETS LINK

C:\Users\Thele\Downloads\student_results.csv