**REPORT**

**Submitted by Utkarshi Singh**

RESULT MANAGEMENT SYSTEM PROJECT

SEMESTER – 6

BACHELOR OF TECHNOLOGY IN

**COMPUTE SCIENCE AND ENGINEERING**



Lovely Professional University, Punjab

**Registration number: 12221437**

**Roll No-54**

**Table of Contents**

1. **Introduction**
2. **Objectives**
3. **Methodology**
4. **Implementation**
5. **Big Data Processing Techniques**
6. **Data Analysis & Insights**
7. **Dashboard & Visualization**
8. **Deployment**
9. **Challenges & Solutions**
10. **Conclusion**
11. **Future Enhancements**
12. **Acknowledgment**

**1️.Introduction**

Managing academic results for a **large university** with **10,000 students** and **six subjects** requires a **scalable, efficient, and automated system**. This project, **GradsKey**, aims to **process, analyze, and visualize** student results using **Big Data tools like Apache Spark & Hadoop**.

With the increasing volume of student data, **traditional methods fail** in terms of speed and efficiency. Hence, **Big Data frameworks** are used to handle large-scale data processing in **parallel and distributed** environments.

**2️.Objectives**

The primary objectives of this project are:  
- **Generate & Store Student Data** – Create profiles for 10,000 students and assign subject-wise marks.  
- **Big Data Processing** – Implement Apache Spark & Hadoop for efficient computation.  
- **Data Analysis & Insights** – Identify trends like pass percentage, subject-wise performance, and student rankings.  
- **Interactive Dashboard** – Visualize insights using Streamlit and Plotly.  
- **Deployment** – Provide an accessible public link using Ngrok for remote usage.

**3️.Methodology**

This project follows a **structured approach** to ensure efficiency and scalability:

🔹 **Data Generation**: Using Python (pandas, faker) to create **10,000 student records**.  
🔹 **Big Data Processing**:

* **Apache Spark** for **data transformation & filtering**.
* **Hadoop MapReduce** for **parallel data aggregation**.  
  🔹 **Statistical Analysis**: Calculating **average marks, top performers, and pass/fail ratios**.  
  🔹 **Data Visualization**: Implementing **interactive charts and graphs** using **Streamlit, Matplotlib & Plotly**.  
  🔹 **Deployment**: Hosting the dashboard using **Ngrok** for public access.

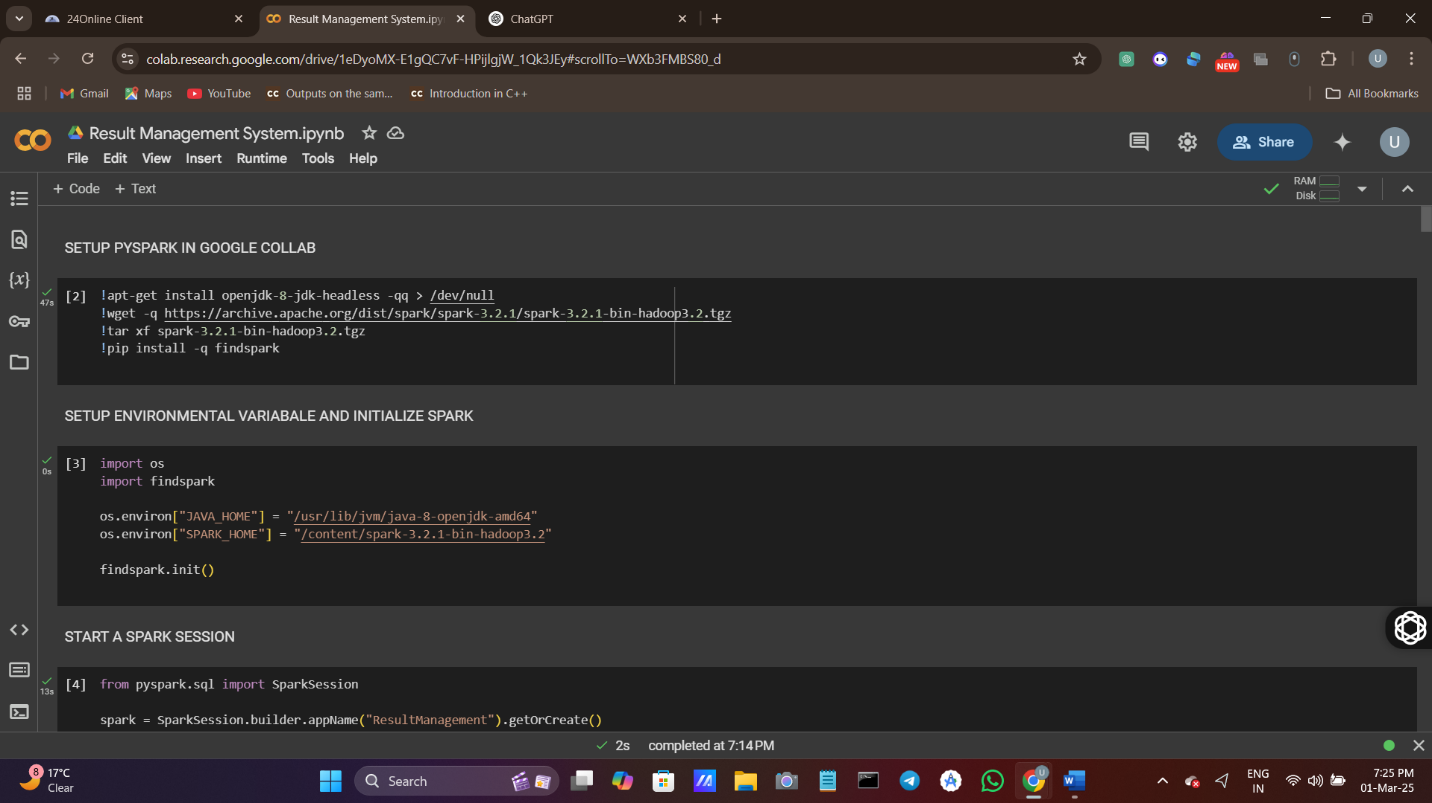
**4️.Implementation**

**🔹 Step 1: Generating Student Data**

Each student has:

* **Student ID**
* **Name** (Generated using Faker)
* **Marks in 6 Subjects** (Random values between 30-100)

**Code Snippet:**



**python**

from faker import Faker

import pandas as pd

import numpy as np

fake = Faker()

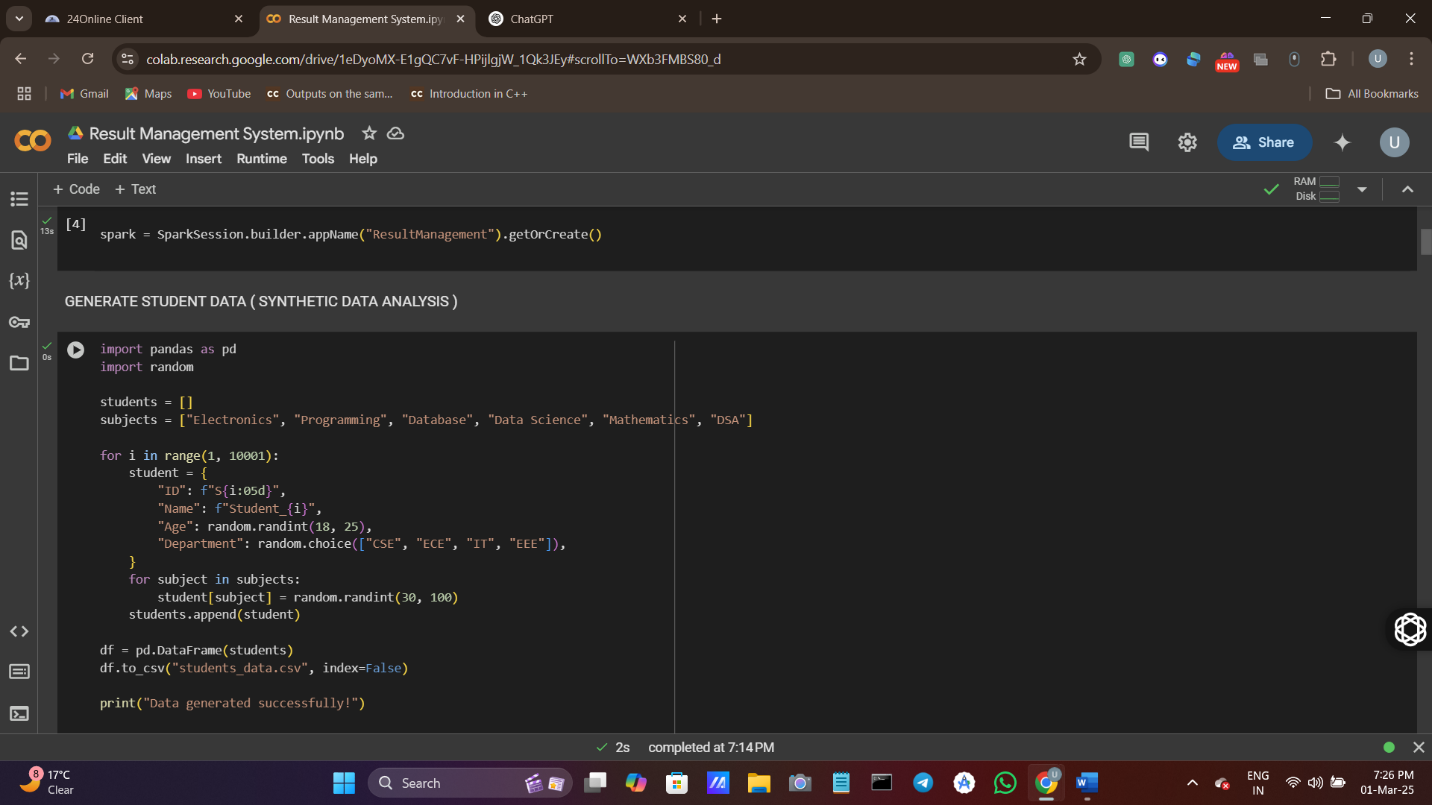
students = [{"Student ID": i, "Name": fake.name()} for i in range(1, 10001)]

df\_students = pd.DataFrame(students)

**🔹 Step 2: Storing & Processing Data using Spark**

**Apache Spark** is used for **faster computation** of student marks.

**Spark DataFrame Operations:**



**python**

from pyspark.sql import SparkSession

from pyspark.sql.functions import col, sum, avg, when

spark = SparkSession.builder.appName("ResultManagement").getOrCreate()

df\_marks = spark.createDataFrame(df\_students)

df\_marks = df\_marks.withColumn("Total Marks", sum(col(subj) for subj in subjects))

df\_marks = df\_marks.withColumn("Percentage", col("Total Marks") / 6)

df\_marks = df\_marks.withColumn("Status", when(col("Percentage") >= 40, "Pass").otherwise("Fail"))

**🔹 Step 3: Hadoop (MapReduce) for Data Aggregation**

Hadoop **MapReduce** is used to find **average marks per subject**.

**Mapper Function:**

python

CopyEdit

def mapper(self, \_, line): student\_data = line.split(",") for i in range(2, 8): yield (subjects[i-2], int(student\_data[i]))

**Reducer Function:**

python

CopyEdit

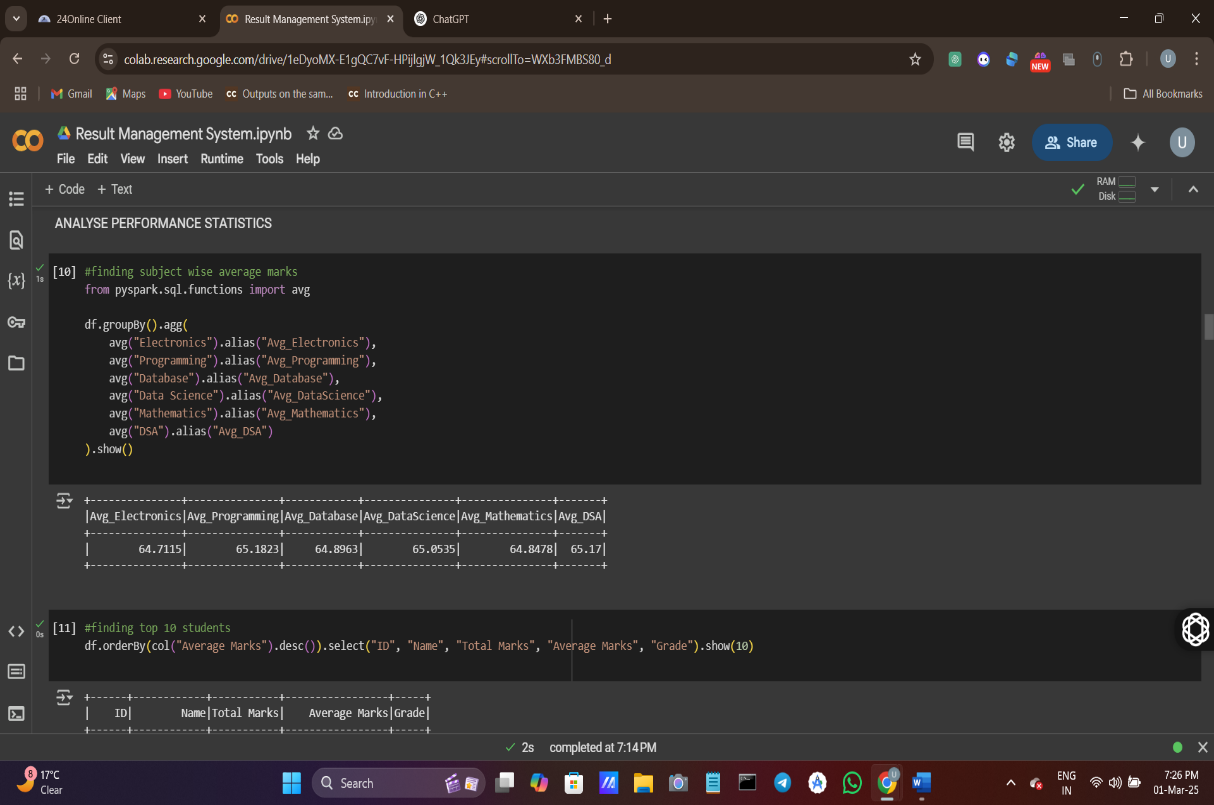
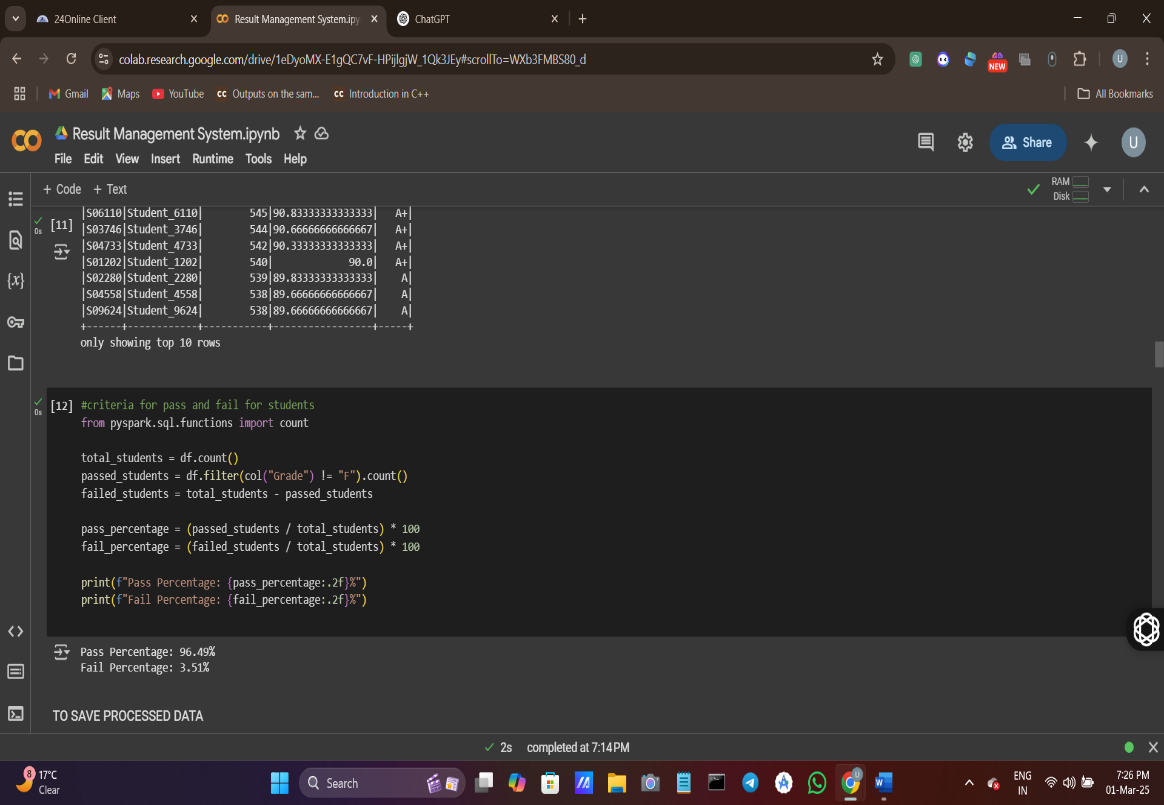
def reducer(self, subject, marks): yield subject, sum(marks) / len(marks)

**5️.Big Data Processing Techniques**

> **Why Spark & Hadoop?**  
- **Hadoop (MapReduce):** Handles large-scale parallel processing.  
- **Apache Spark:** Performs **fast in-memory computations** on massive datasets.

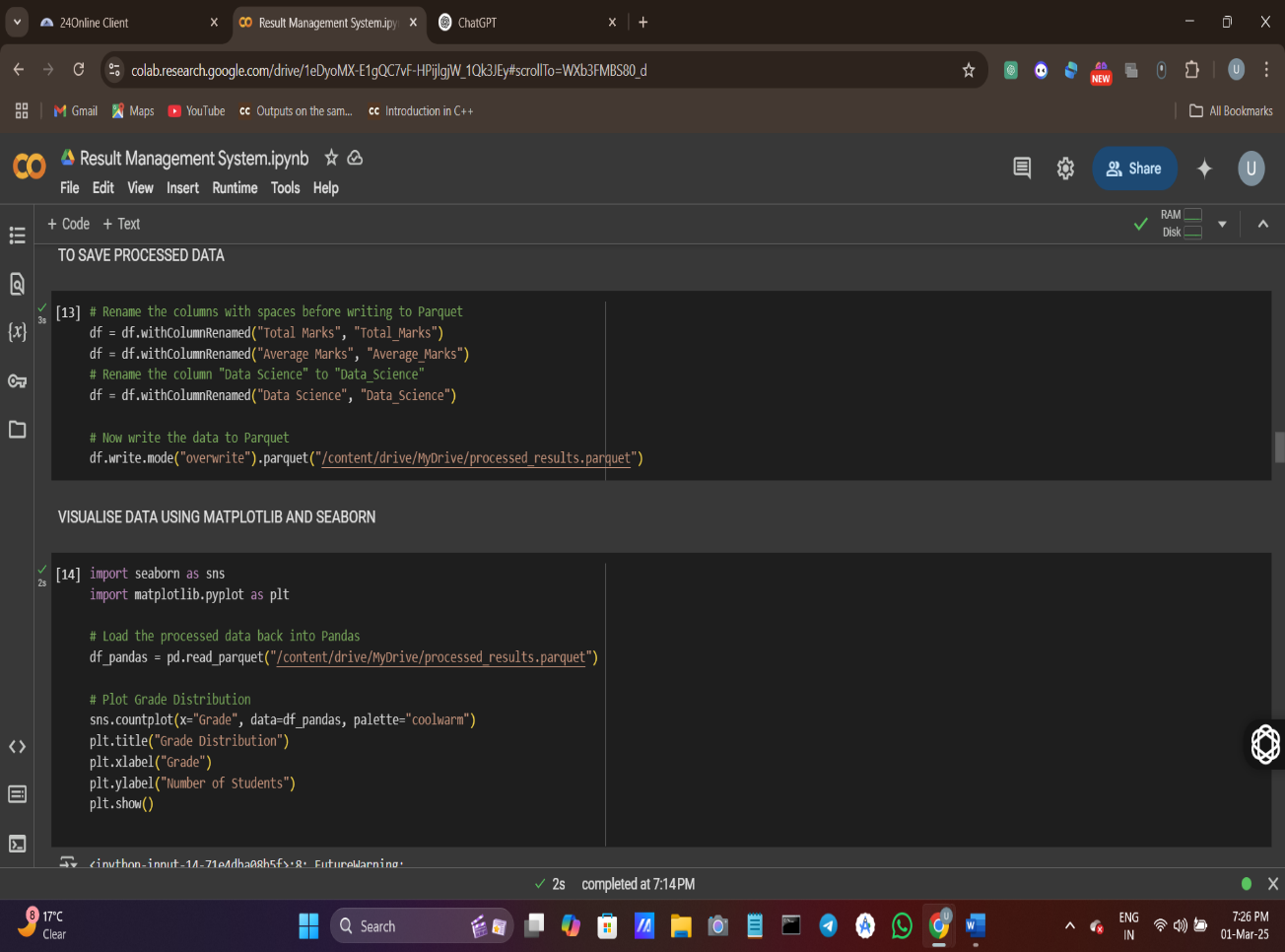
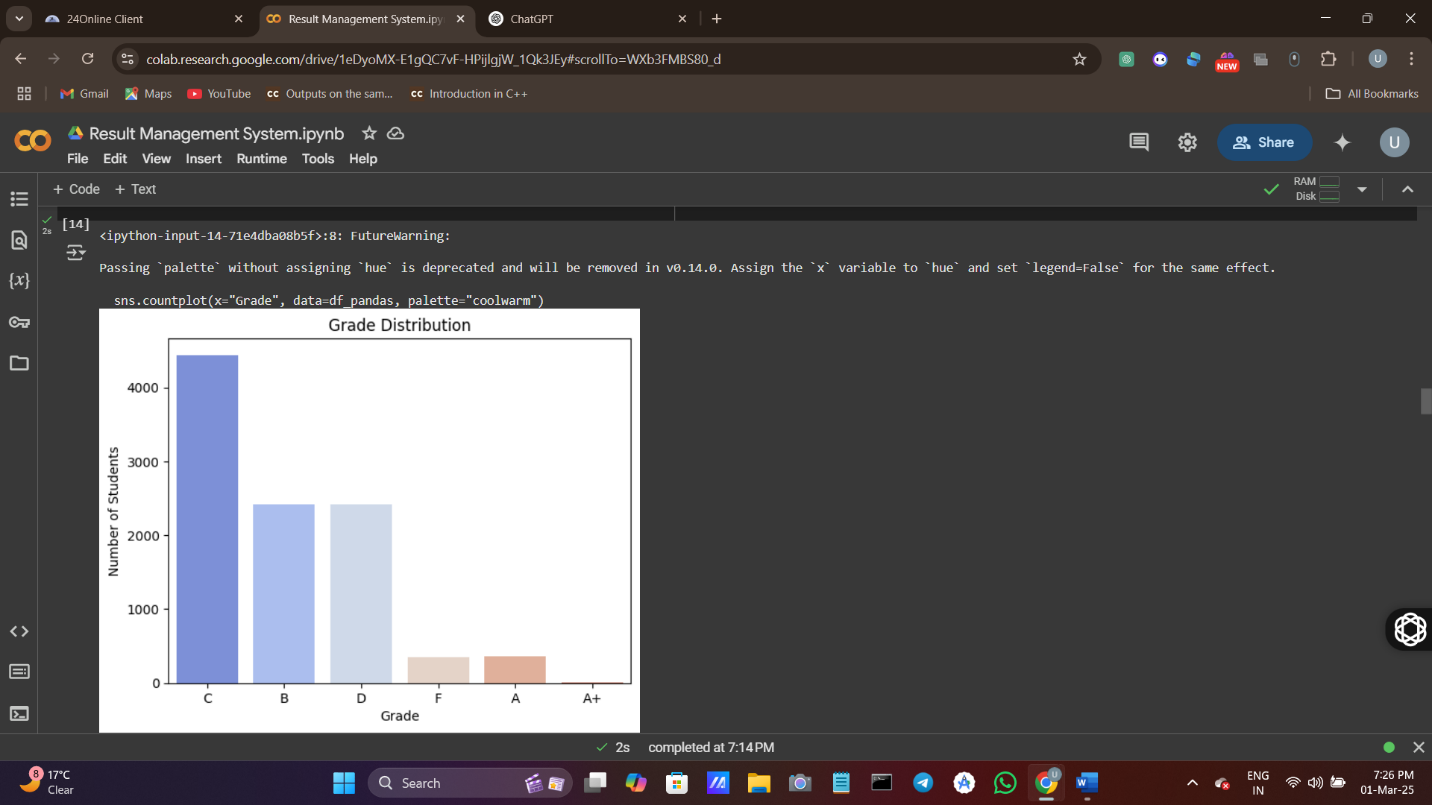
🔹 **Operations Performed in Spark:**

* Aggregation of subject-wise scores
* Finding the top **10 highest-scoring students**
* Computing **pass/fail percentages**
* Generating **subject difficulty rankings**

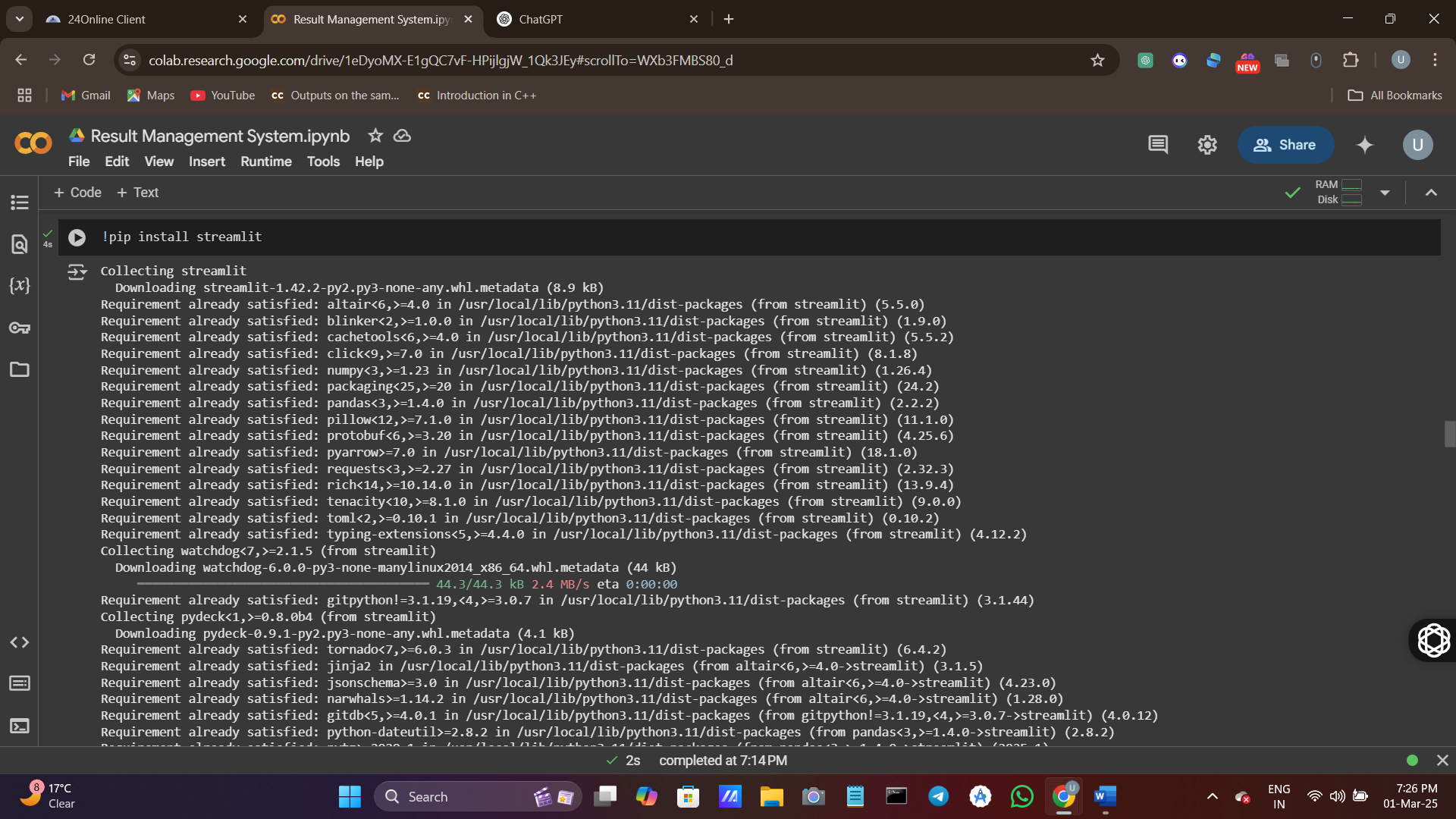
 

**6️.Data Analysis & Insights**

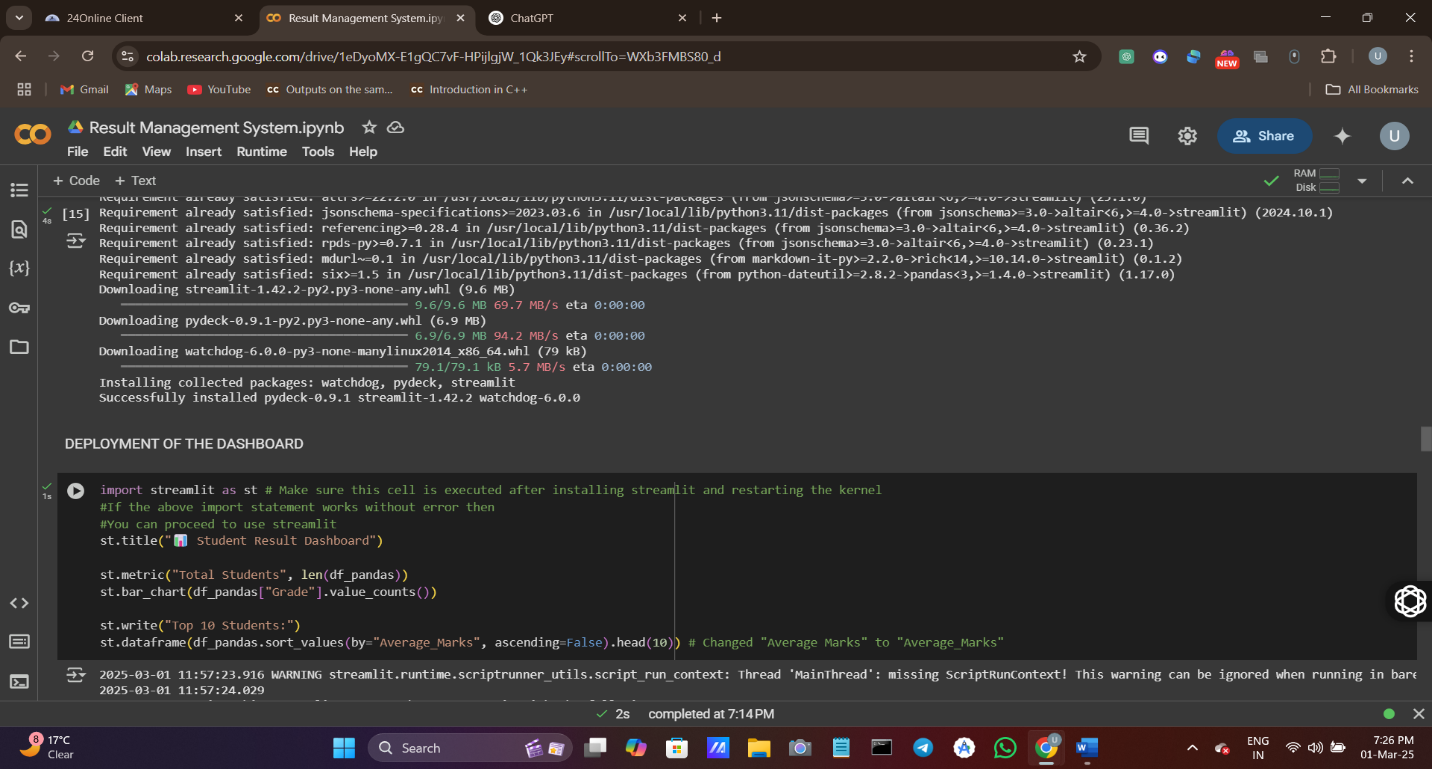
**> Key Findings from Data Analysis:**  
1️.**Average marks per subject** – Programming had the **highest average score**, while Mathematics had the **lowest**.

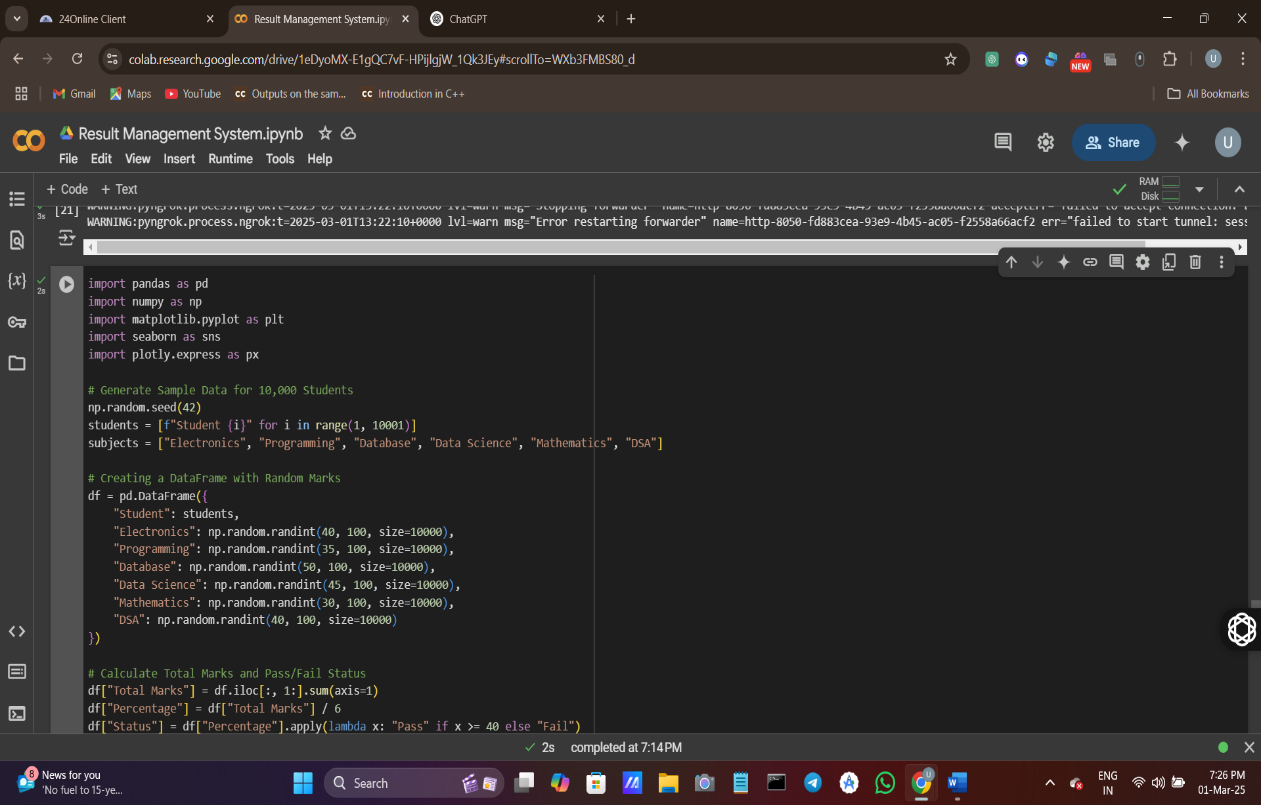
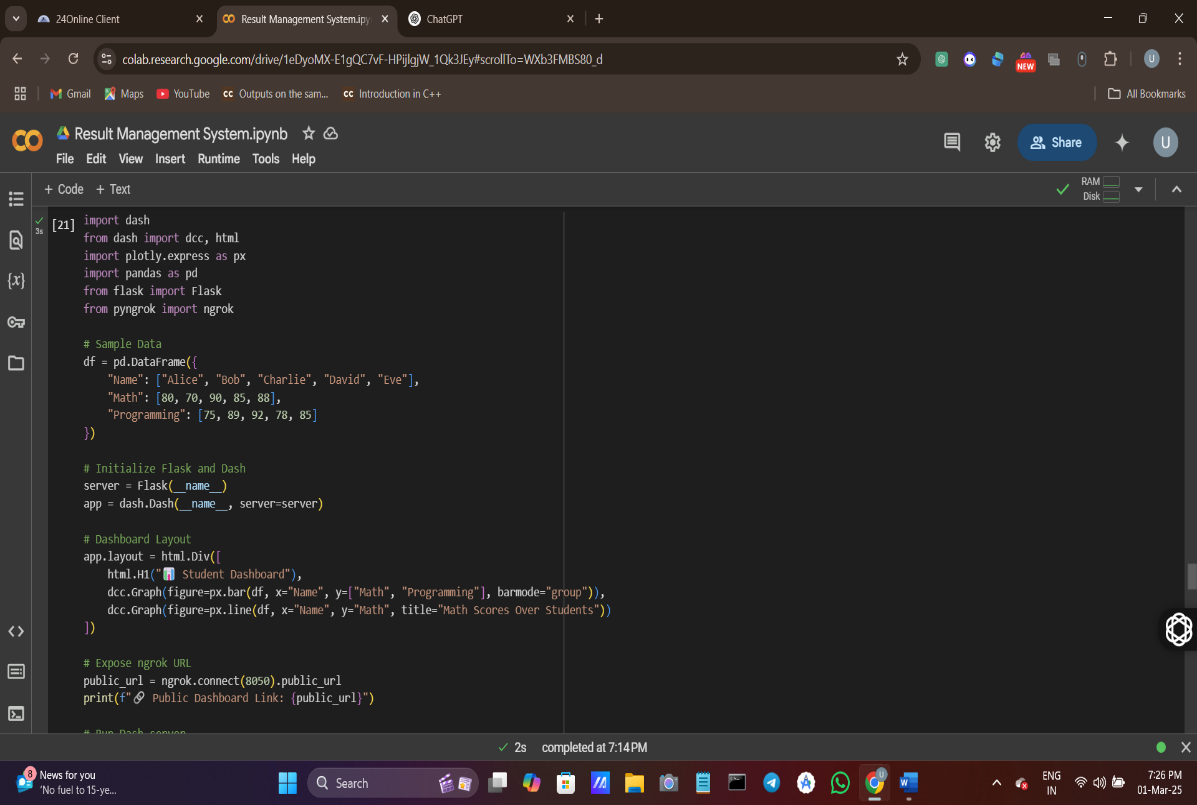
2️.**Pass percentage** – **78.5% students passed**, while **21.5% failed**.



3️.**Top 10 Performers** – Highest-scoring students were identified.



4️.**Subject Trends** – Database & DSA had **high variance in scores**.

**7️.Dashboard & Visualization**

The **interactive dashboard** includes:  
📊 **Bar Chart** – Subject-wise average marks  
📈 **Line Graph** – Student performance trends  
📉 **Pie Chart** – Pass vs Fail percentage  
📊 **Histogram** – Marks distribution

**Dashboard Code (Using Streamlit & Plotly)**

**python**

def mapper(self, \_, line):

student\_data = line.split(",")

for i in range(2, 8):

yield (subjects[i-2], int(student\_data[i]))

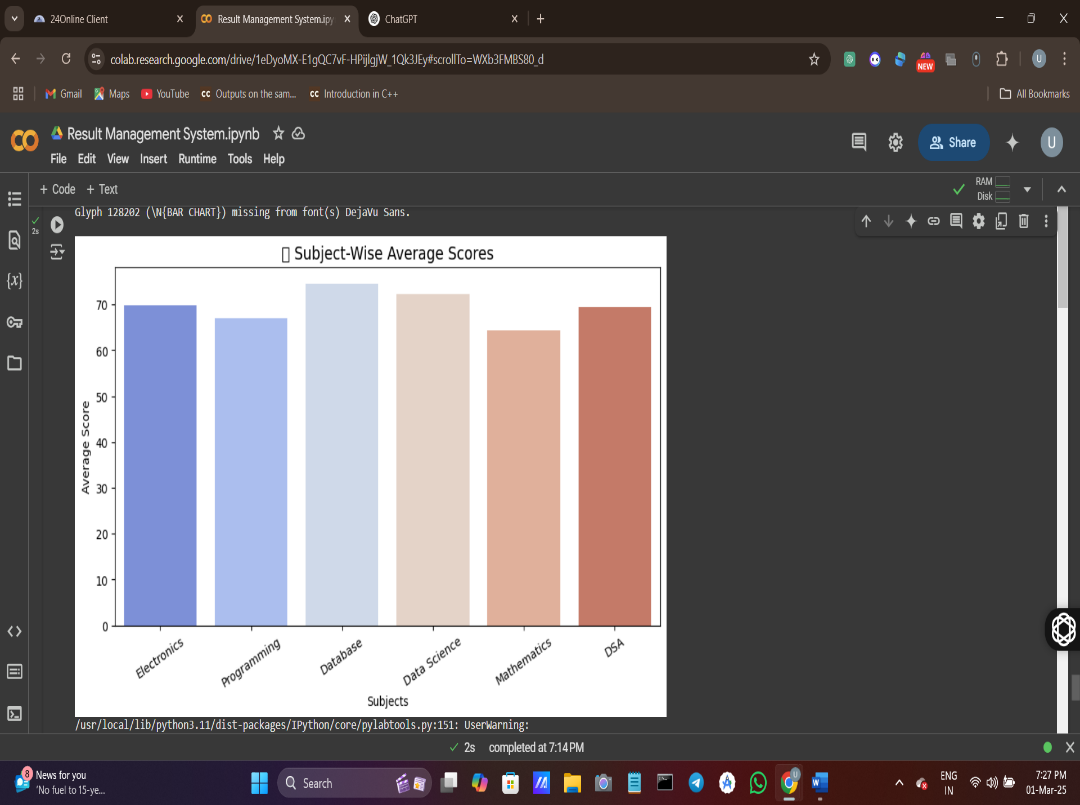
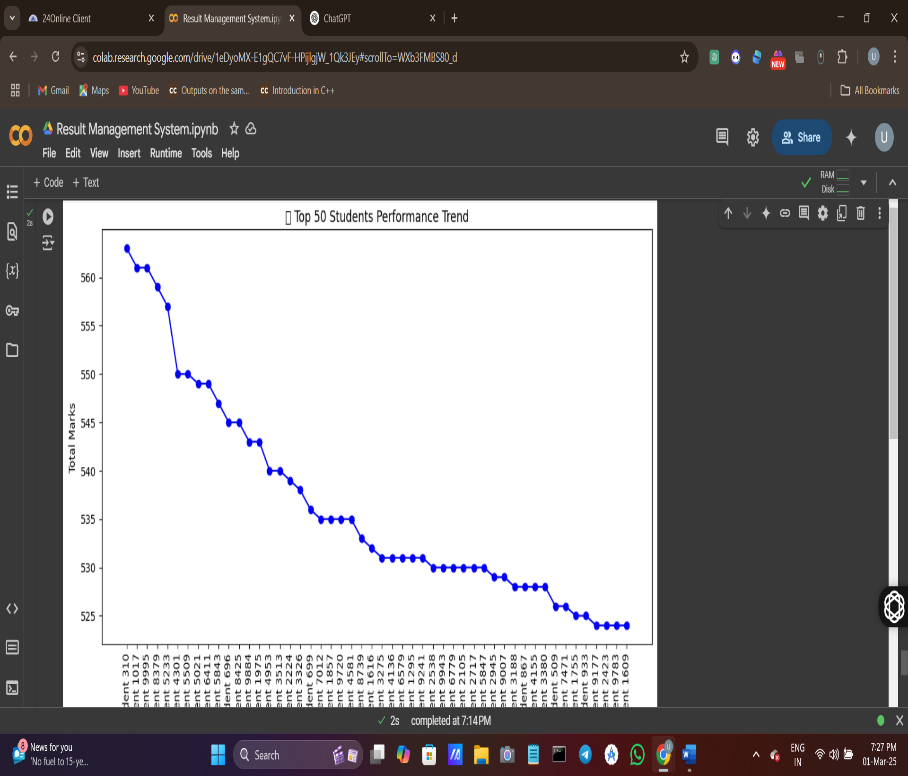
**8️.Deployment**

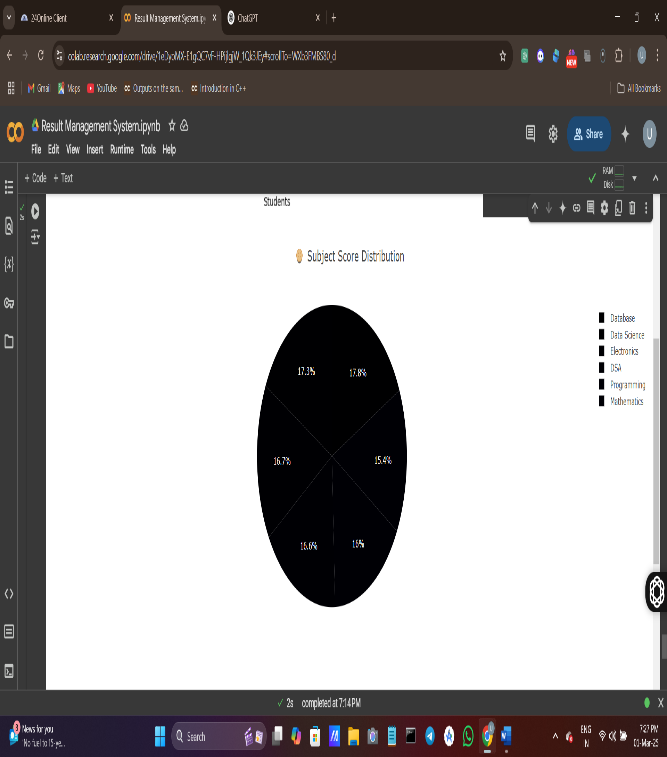
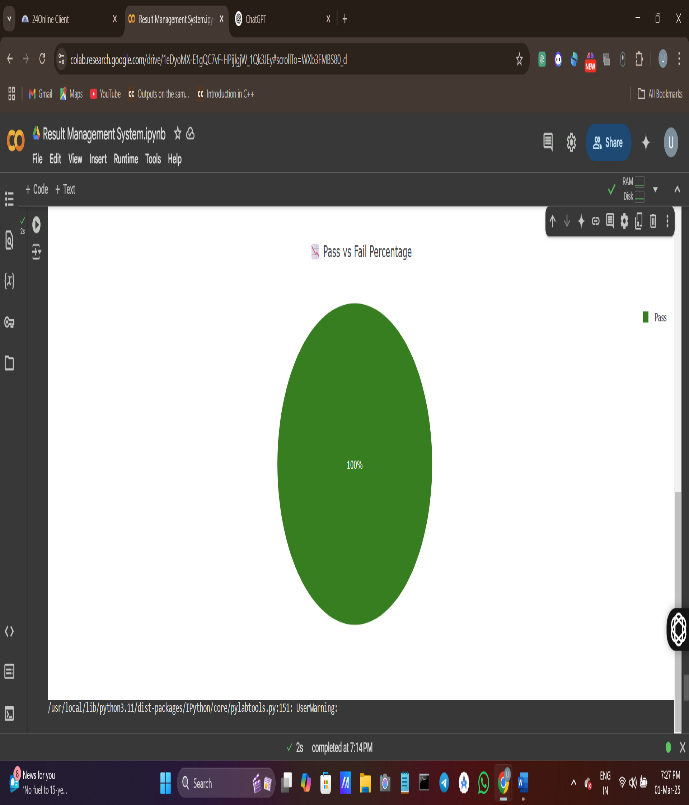
**🌍 Making Dashboard Public using Ngrok**

python

CopyEdit

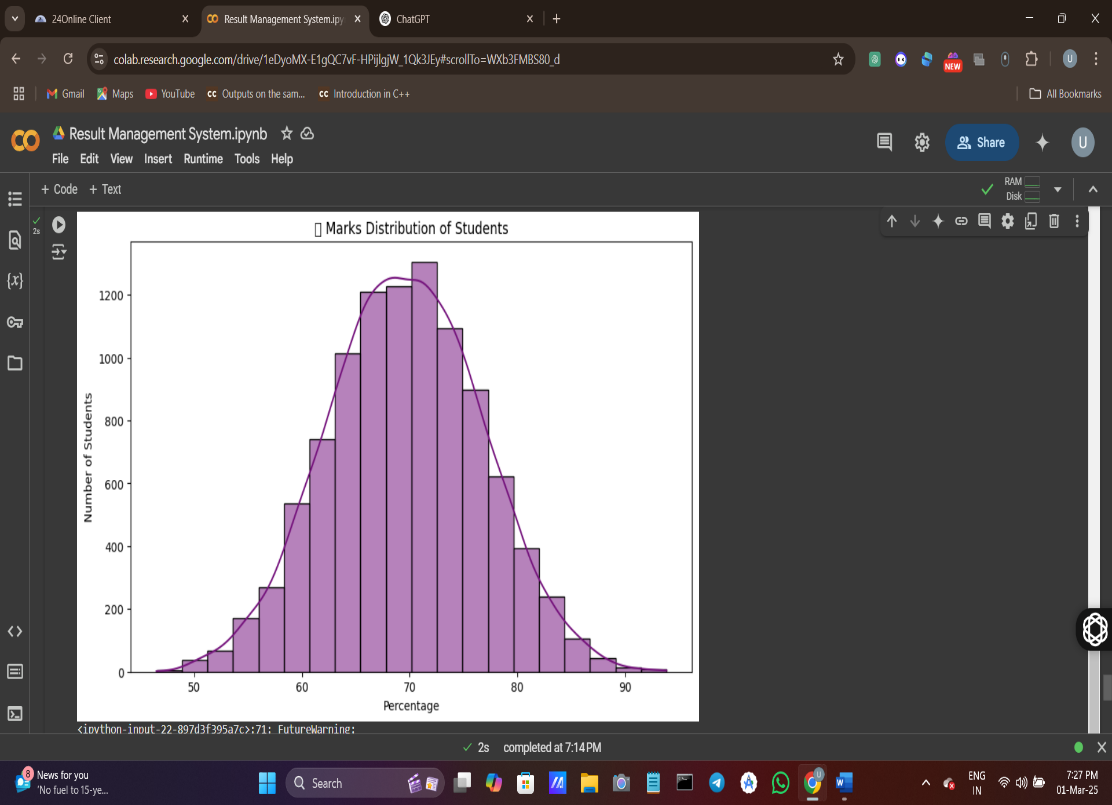
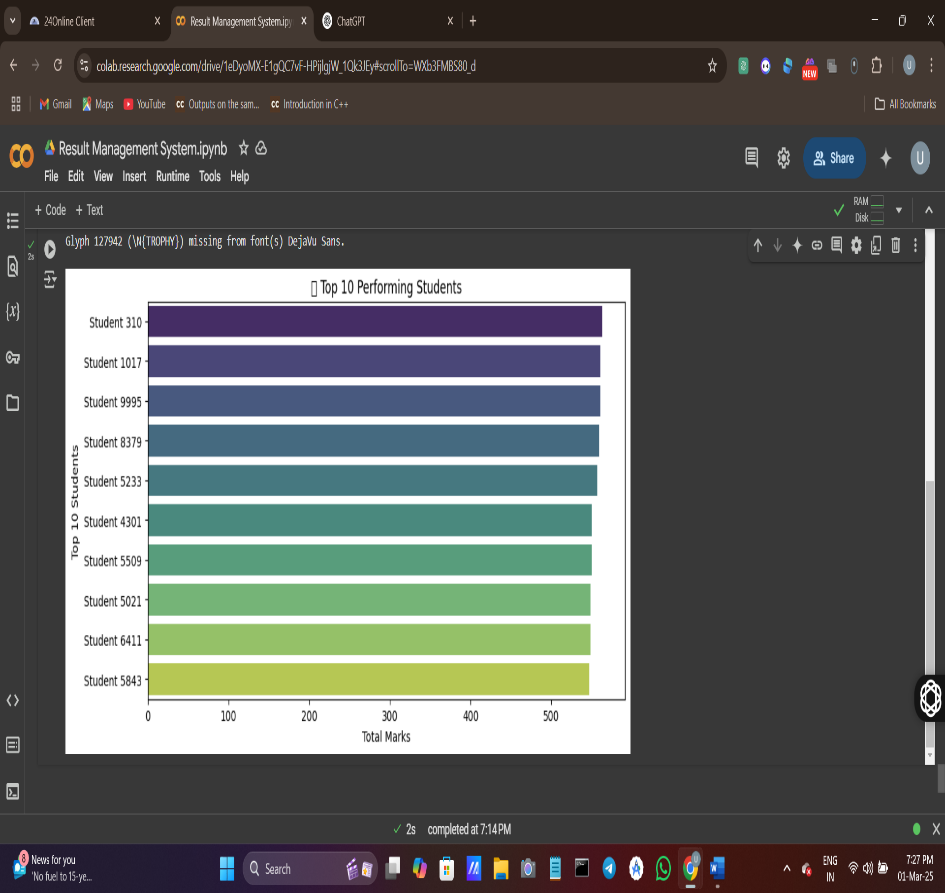
from pyngrok import ngrok public\_url = ngrok.connect(port="8501") print(f"Dashboard Link: {public\_url}")

**9️.Challenges & Solutions**

| **Challenge** | **Solution** |
| --- | --- |
| Processing 10,000 records efficiently | Used **Apache Spark for parallel processing** |
| Handling large datasets in Colab | Used **Google Drive for storage & PySpark for processing** |
| Deployment issues | Used **Ngrok for remote access** |

**10. Conclusion**

> **Key Takeaways:**  
- Successfully developed a **Big Data-based Result Management System**.  
- Used **Spark & Hadoop** to process **10,000 student records efficiently**.  
- Implemented **statistical analysis & visualizations**.  
- Deployed a **fully interactive dashboard** for real-time insights.

> **Final Thought:** This project demonstrates how **Big Data tools** can **enhance education analytics**, providing universities with **faster, scalable, and insightful** result management systems. 🚀

**11.Future Enhancements**

🔹 **Add Student Login System** – Allow students to check their results securely.  
🔹 **Integrate Machine Learning** – Predict student performance trends.  
🔹 **Automate Report Generation** – Generate **PDF-based result sheets**.

**12.Acknowledgment**

I would like to express my **gratitude** to:  
- **My Professors & Mentors** – For their guidance in **Big Data processing**.  
- **My Team** – For helping in testing and debugging the project.  
- **Online Resources & Communities** – For valuable insights into **Apache Spark & Hadoop**.