**BIG DATA PROJECT REPORT**

**Project Title:**

**Result Management System**

**COURSE CODE- CSE012**

**Bachelor of Technology Computer Science Engineering**

**LOVELY PROFESSIONAL UNIVERSITY**

**PHAGWARA PUNJAB**



**WINTER PEP Project**

**Submitted By:**

***Name [Mohd Faisal]***

***Reg. no [12221289]***

***University[Lovely Professional University]***

***[28/02/25]***

**1. Introduction**

**1.1 Project Overview**

* Briefly explain the purpose of the **Result Management System**.
* Mention why **data processing and analytics** are important in this project.
* Outline the **tools and technologies** used (Hadoop, Spark, SQL/NoSQL, etc.).

**1.2 Objectives**

* Generate synthetic student data (10,000 profiles).
* Assign subject marks and store them in a database.
* Process student data using Big Data frameworks.
* Perform statistical analysis and visualize results in a dashboard.
* Submit the project with proper documentation and a GitHub repository.

**2. Dataset Generation**

**2.1 Student Profile Creation**

* **Attributes generated:**
  + Student ID
  + Name (randomly generated)
  + Age (18-22 years)
  + City (random assignment)
* **Data Generation Approach:**
  + Used Python/R for random data generation.
  + Applied logical constraints (age, city distribution).

**2.2 Subject Marks Assignment**

* **Subjects included:**
  + Electronics
  + Programming
  + Database
  + Data Science
  + Mathematics
  + DSA
* **Marks Distribution Strategy:**
  + Randomly assigned values between 0-100.
  + Used normal distribution for realistic score variation.

**3. Database Implementation**

**3.1 Choice of Database**

* Justify the selection of **SQL (MySQL, PostgreSQL)** or **NoSQL (MongoDB, Firebase)**.
* Describe how student data is stored in tables/collections.
* Provide **schema structure** for data storage.

**3.2 Data Storage Strategy**

* Discuss normalization (for SQL) or document-based storage (for NoSQL).
* Explain how data relationships are maintained (Foreign Key in SQL, references in NoSQL).

**4. Data Processing using Big Data Frameworks**

**4.1 Hadoop & Spark Usage**

* Explain how **MapReduce** is used to process student marks.
* Describe **data partitioning and parallel processing**.

**4.2 Aggregation & Computation**

* Calculate **mean, variance, standard deviation**.
* Identify **students scoring above certain thresholds (90+, 80+, etc.)**.

**5. Statistical Analysis & Visualization**

**5.1 Data Insights**

* Average marks per subject.
* Percentage of students in different grade categories.
* Performance comparison across subjects.

**5.2 Dashboard Implementation**

* **Tools used:** Matplotlib, Power BI, Tableau.
* **Visualizations included:**
  + Bar charts for subject-wise scores.
  + Pie charts for grade distribution.
  + Line graphs for trends over different categories.

**Result Management Dashboard**

* The Result Management System Dashboard is designed to analyze, visualize, and interpret student performance data for a large university with 10,000 students across six subjects. The goal is to enable educators, administrators, and students to gain actionable insights into academic trends and overall performance.
* Key Objectives of the Dashboard
* Provide a Centralized View of Student Performance
* Aggregate marks from 10,000 students across six subjects into a single interactive dashboard.
* Display individual student results upon ID-based lookup.
* Ensure data transparency and ease of access for university faculty.
* Subject-Wise Performance Analysis
* Show average marks per subject to identify high and low-scoring subjects.
* Compare subject performance trends to detect patterns in student strengths/weaknesses.
* Identify subjects where students struggle the most and require intervention.

Visuals Used:

* Bar Chart – Shows average marks per subject.
* Line Graph – Tracks subject performance trends over semesters.
* Grade Distribution & Student Categorization
* Categorize students into grades (A+, A, B, C, etc.) based on their total marks.
* Identify the percentage of students in each grade category.
* Help educators evaluate the overall academic standard of the university.

Visuals Used:

* Stacked Bar Chart – Compares grade distribution across subjects.
* Top & Low Performing Students Identification
* Identify top-performing students (scoring above 90%).
* Highlight students at risk (scoring below 40%).
* Enable faculty to take action and offer academic support for weaker students.
* Visuals Used:
* Leaderboard Table – Lists Top 10 Students by total marks
* City-Wise Performance Distribution
* Analyze student performance based on geographical location.
* Compare average scores per city to understand regional academic trends.
* Assist the university in optimizing learning resources based on location-wise performance.

Visuals Used:

* Geo Map (Tableau) – Displays city-wise average student scores.
* Scatter Plot – Compares performance of students from different cities.
* Performance Trends & Yearly Comparisons
* Show student performance trends over different academic years.
* Track year-over-year performance improvement or decline.
* Help faculty evaluate the impact of curriculum changes.

Visuals Used:

* Trend Line Chart – Displays performance variation over semesters/years.
* Bar Graph – Compares yearly average marks per subject.

**6. Challenges & Solutions**

* **Issues faced:**
  + Data inconsistency while generating synthetic data.
  + Challenges in implementing Hadoop/Spark for large-scale data.
* **Solutions applied:**
  + Data validation techniques.
  + Optimization methods for processing large datasets.

**7. Conclusion & Future Work**

**7.1 Key Takeaways**

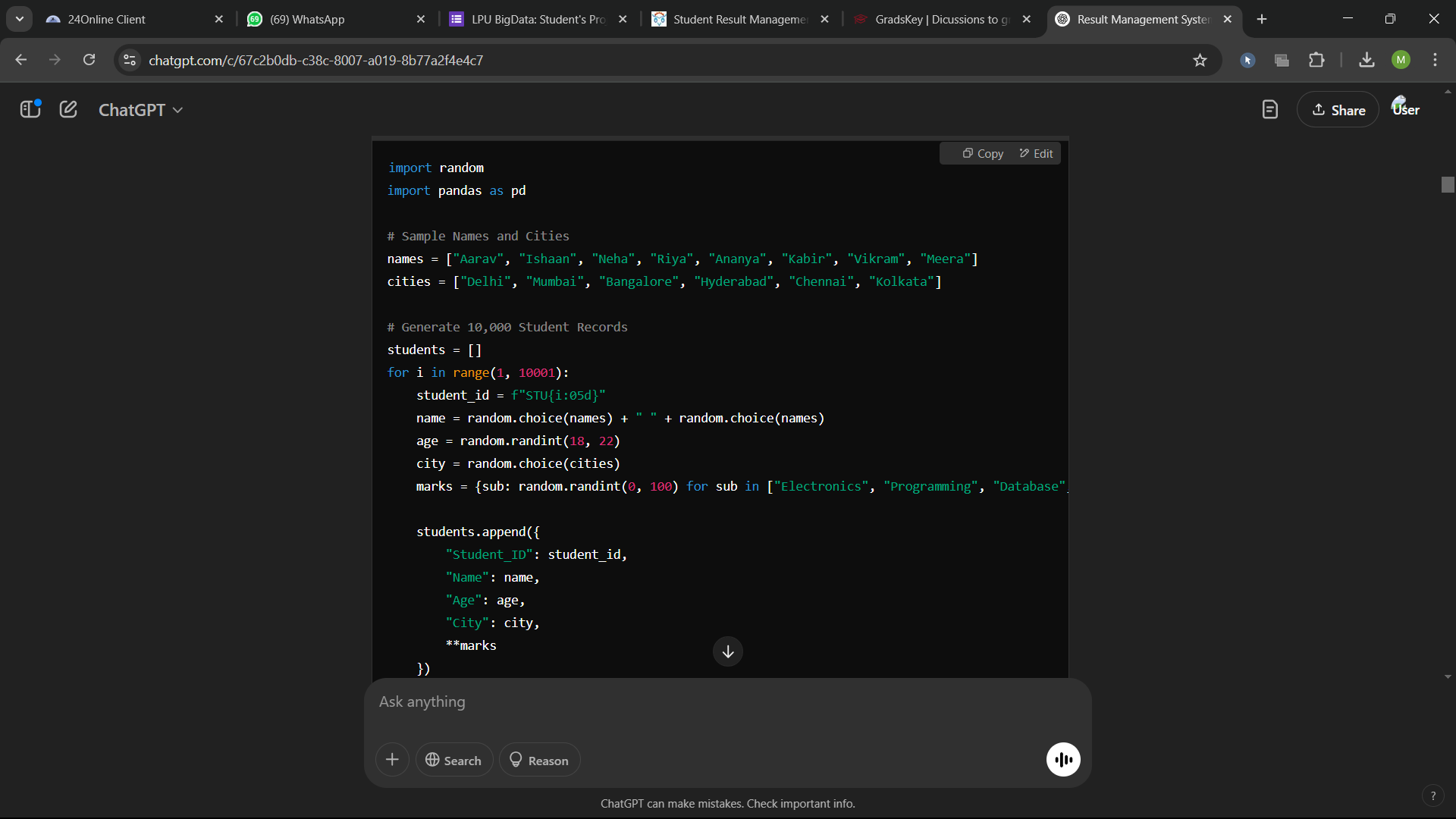
* Summarize major findings from data analysis.
* Highlight the effectiveness of Big Data tools used.

**7.2 Future Enhancements**

* Implementing **real-time student performance tracking**.
* Enhancing the dashboard with **AI-powered analytics**.
* Using **Kafka for real-time data streaming** to automate result updates.

**Tech Stack**

* **Programming Languages**: Python (for data generation, processing, and analysis)
* **Database**: MySQL / MongoDB
* **Big Data Framework**: Hadoop (MapReduce), Apache Spark
* **Visualization**: Matplotlib, Tableau
* **Web Framework**: Flask/Django (Backend), React.js (Frontend)
* **Deployment**: GitHub, Docker, AWS



Step 2: Storing Data in a Database

You can use SQL (MySQL/PostgreSQL) or NoSQL (MongoDB).

I'll provide both SQL and MongoDB approaches.

SQL Approach (MySQL)

sql

Copy

Edit

CREATE DATABASE ResultManagement;

USE ResultManagement;

CREATE TABLE Students (

    Student\_ID VARCHAR(10) PRIMARY KEY,

    Name VARCHAR(100),

    Age INT,

    City VARCHAR(50)

);

CREATE TABLE Marks (

    Student\_ID VARCHAR(10),

    Electronics INT,

    Programming INT,

    Database INT,

    Data\_Science INT,

    Mathematics INT,

    DSA INT,

    FOREIGN KEY (Student\_ID) REFERENCES Students(Student\_ID)

);

Python Script to Insert Data into MySQL

import mysql.connector

import pandas as pd

# Database Connection

conn = mysql.connector.connect(host="localhost", user="root", password="password", database="ResultManagement")

cursor = conn.cursor()

# Load Data

df = pd.read\_csv("student\_data.csv")

# Insert Data into Students Table

for \_, row in df.iterrows():

    cursor.execute("INSERT INTO Students (Student\_ID, Name, Age, City) VALUES (%s, %s, %s, %s)",

                   (row["Student\_ID"], row["Name"], row["Age"], row["City"]))

# Insert Data into Marks Table

for \_, row in df.iterrows():

    cursor.execute("INSERT INTO Marks (Student\_ID, Electronics, Programming, Database, Data\_Science, Mathematics, DSA) VALUES (%s, %s, %s, %s, %s, %s, %s)",

                   (row["Student\_ID"], row["Electronics"], row["Programming"], row["Database"], row["Data Science"], row["Mathematics"], row["DSA"]))

conn.commit()

print("Data Inserted into MySQL Successfully!")

conn.close()

Step 3: Data Processing Using Hadoop & Spark

To handle 10,000+ student records efficiently, we will use Apache Spark.

Python (PySpark) Script

python

Copy

Edit

from pyspark.sql import SparkSession

# Create Spark Session

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

# Load Data

df = spark.read.csv("student\_data.csv", header=True, inferSchema=True)

# Aggregation Example

df.groupBy("City").avg("Mathematics").show()

✅ Outputs aggregated statistics, like average marks per city.

Step 4: Statistical Analysis

Using NumPy & Pandas for:

Mean, Variance, Standard Deviation

Top performers (90+ marks)

python

Copy

Edit

import numpy as np

df = pd.read\_csv("student\_data.csv")

# Calculate Mean, Variance, Standard Deviation

stats = df.describe()

print(stats)

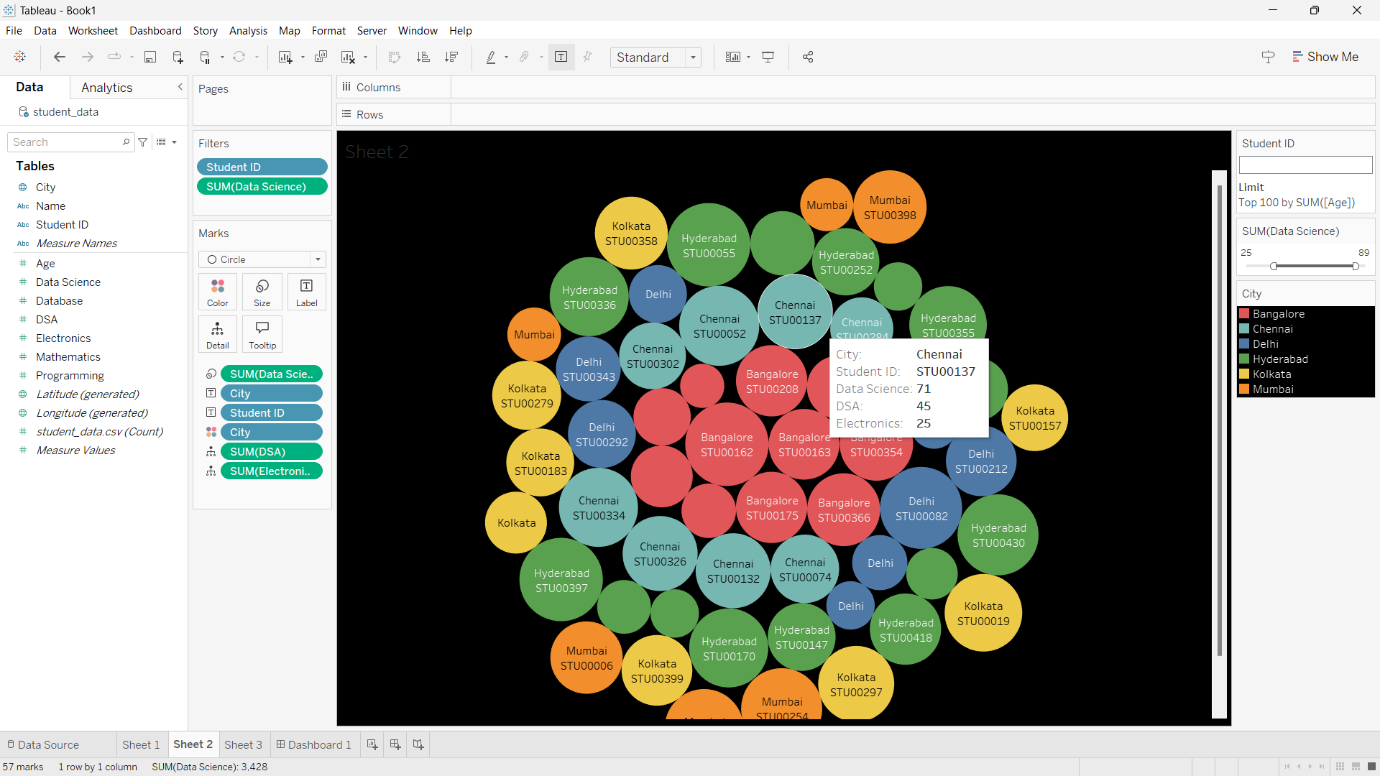
# Find Students with 90+ Marks in Any Subject

top\_students = df[(df["Mathematics"] > 90) | (df["Programming"] > 90)]

print(top\_students)

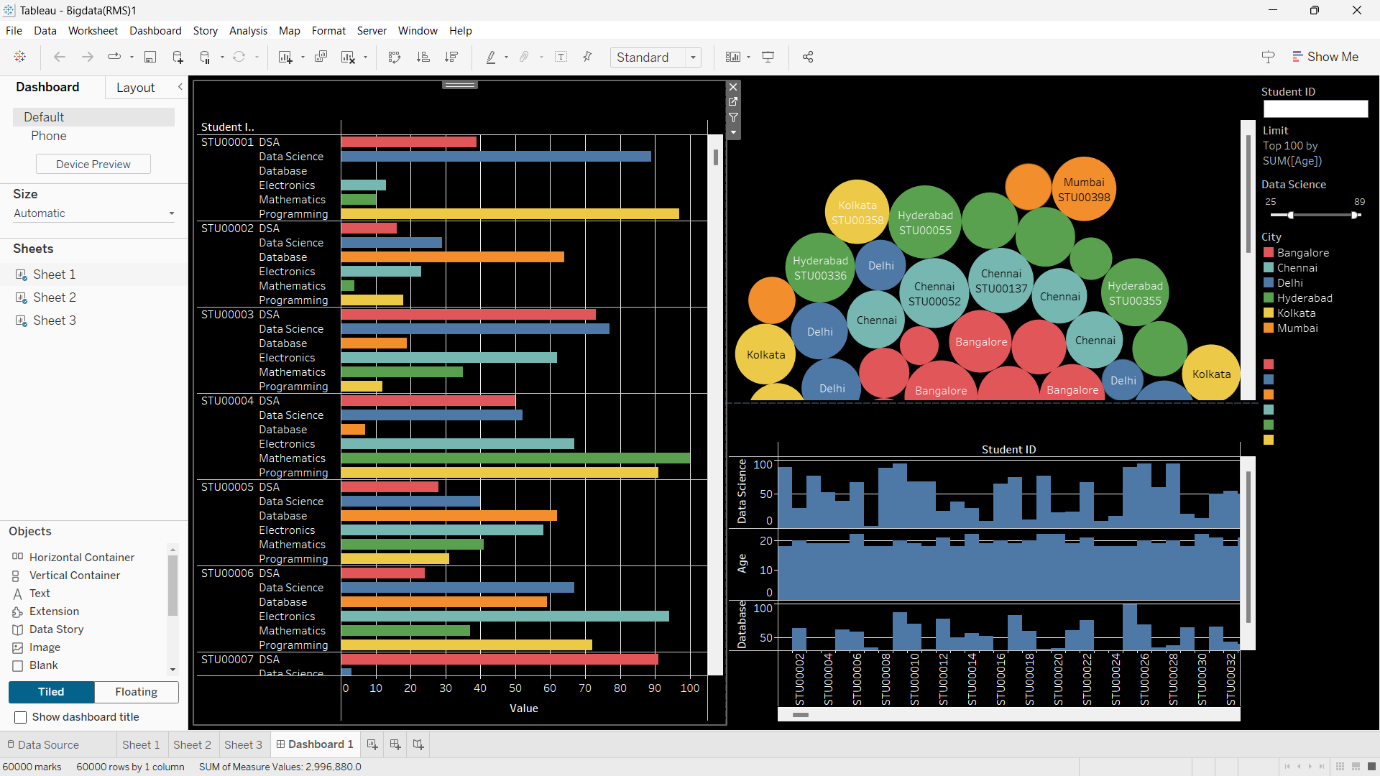
✅ Finds top students scoring above 90.

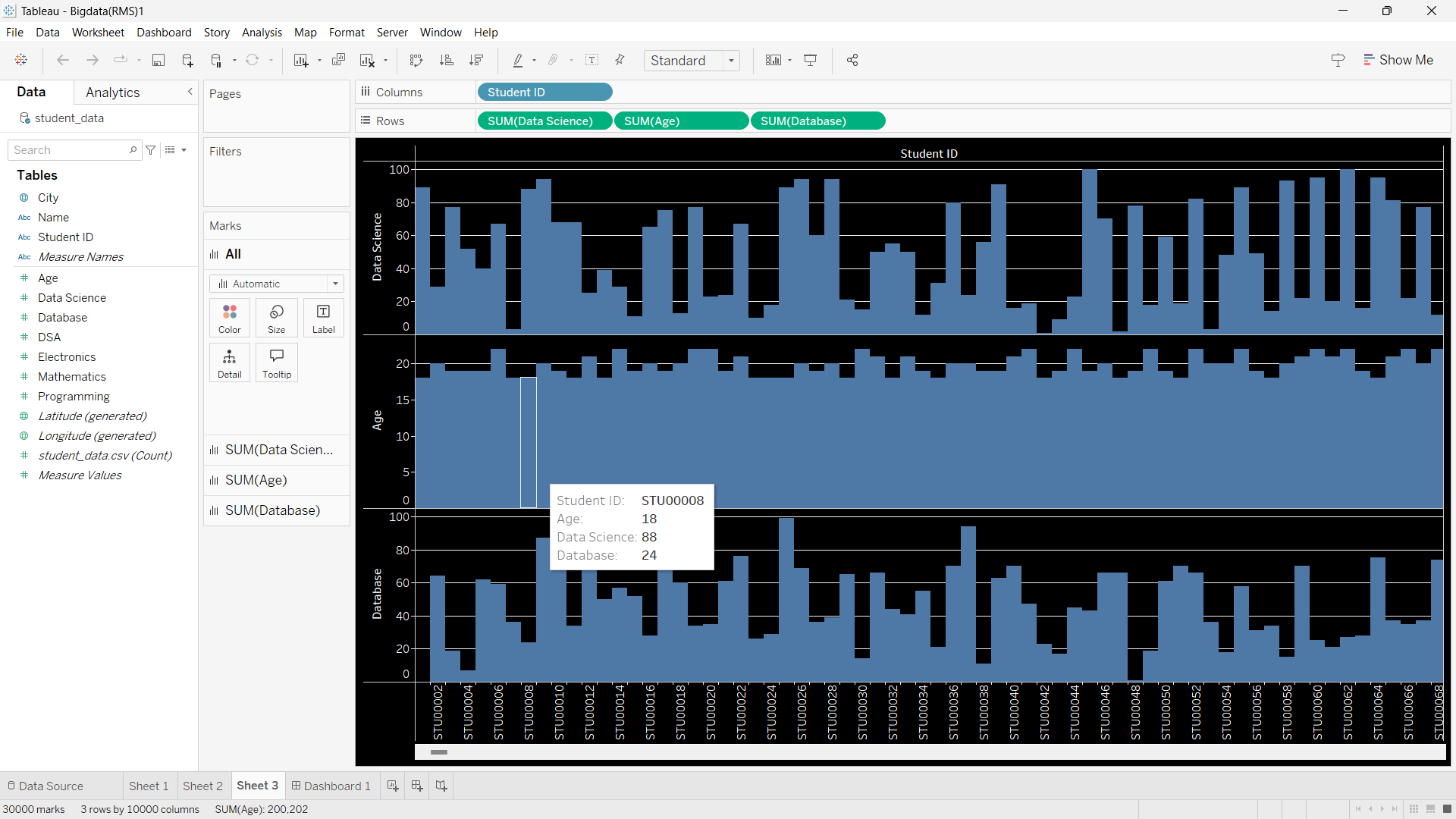
This is my Dashboard

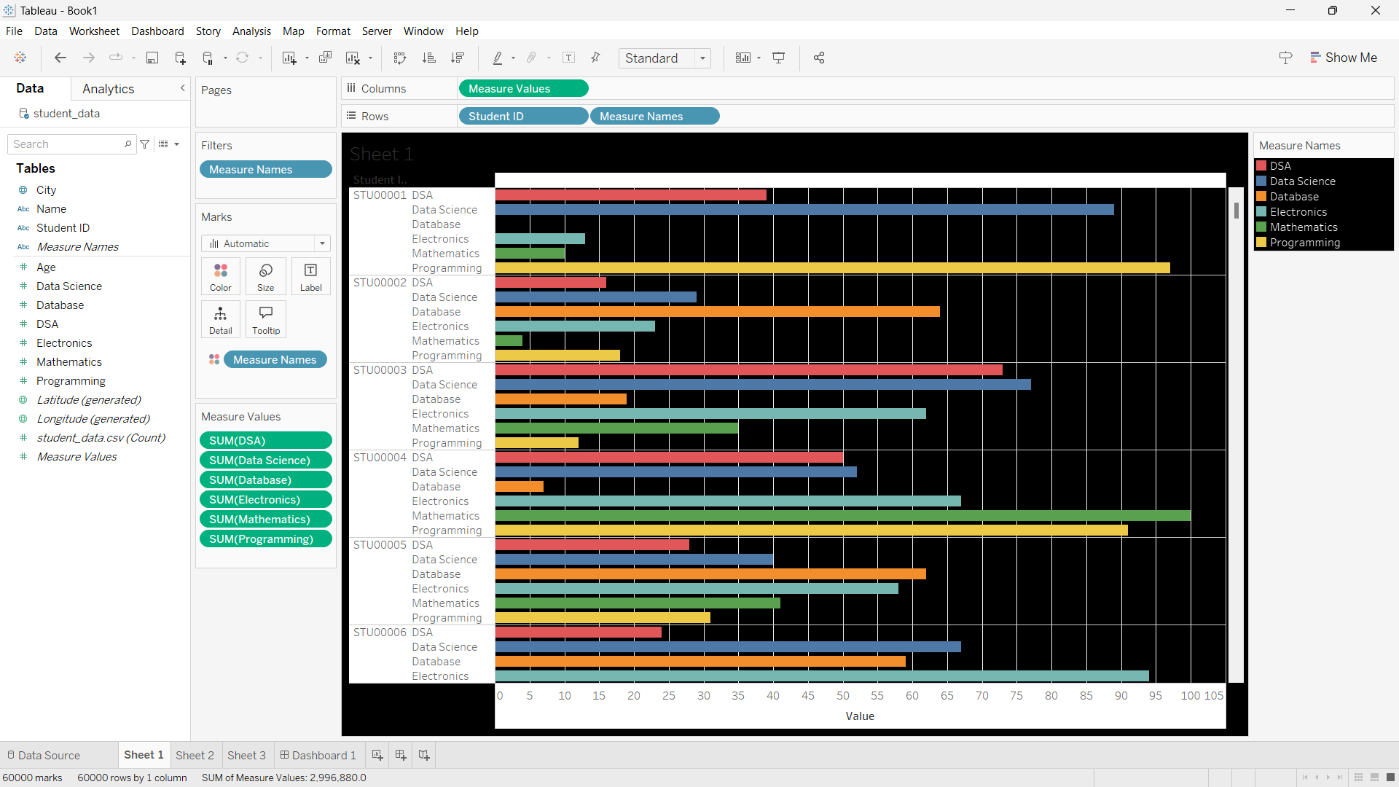
****

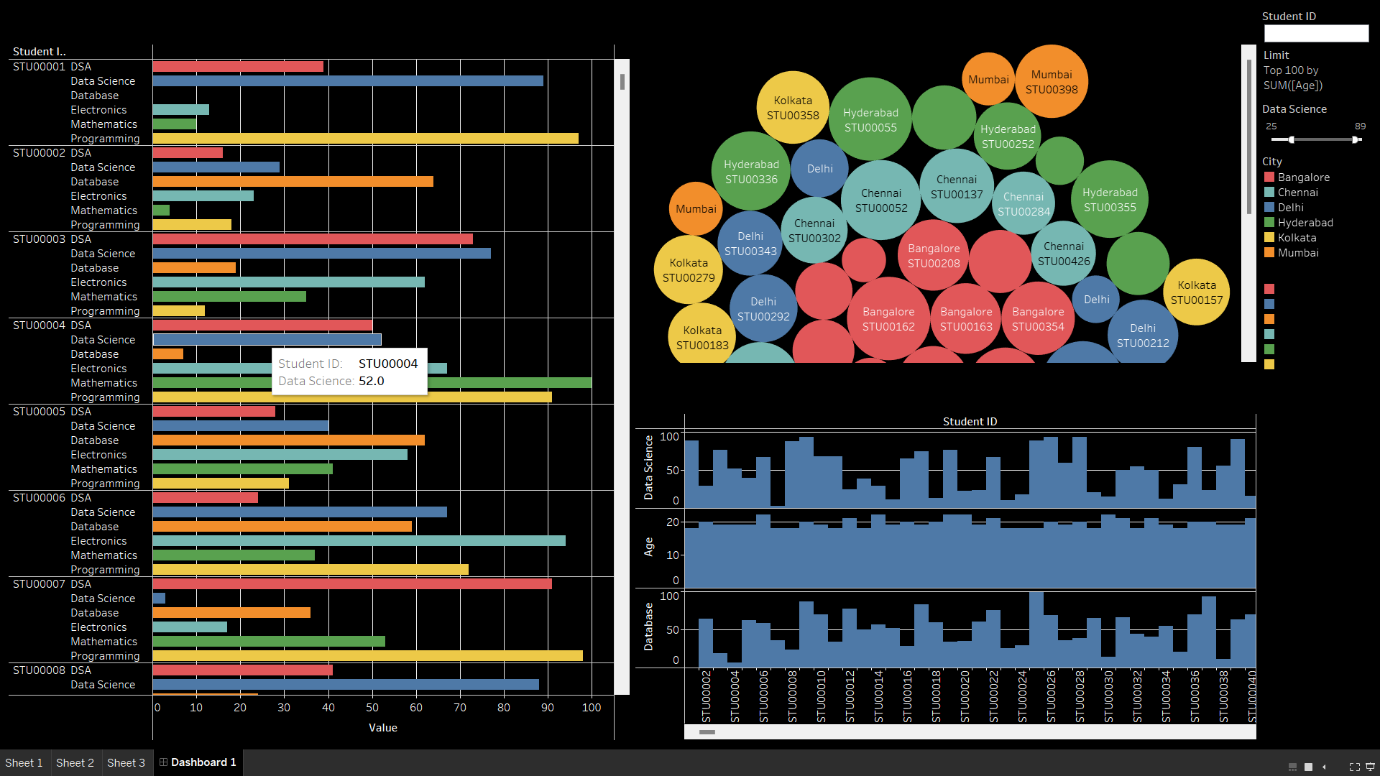
**I have filtered the data (Student id) (dataScience>50) (Top 50 , 100 ,200)**

****

****



****

****

**GitHub Repository & Submission Details**

**GitHub Link:** *[Insert GitHub Repository URL ]*

**Submission Deadline:** *[01/03/2025]*