**Project Report**

**Result Management System**

**About Project:**

The Result Management System is designed to efficiently process, analyse, and manage student academic records using Apache Spark, MongoDB, and Kafka. This system handles large datasets, performs statistical analysis, and provides insights into student performance.

Name: Vedant Gurav

Reg.no: 12200809

Batch: 9W084

GitHub Link: <https://github.com/vedantgurav9/Result-Management-System>

1. **Introduction**

The **Result Management System** is designed to streamline the processing and analysis of student performance data using **big data technologies** like Apache Spark and Hadoop. With increasing student populations and the need for real-time result processing, traditional methods often fail to provide efficient solutions. This project leverages **distributed computing** to handle large datasets, perform advanced analytics, and generate insightful visualizations. The system not only computes **subject-wise averages, highest and lowest marks,** but also provides **real-time feedback streaming** using Kafka. Additionally, the integration of **machine learning techniques** allows for **predictive performance analysis,** making the system a comprehensive solution for educational institutions.

1. **Objectives**

* Automate student result processing.
* Compute average, highest, and lowest marks for each subject.
* Store and retrieve large datasets efficiently using Hadoop Distributed File System (HDFS).
* Utilize Kafka for real-time data streaming.
* Implement visualizations to present performance trends.
* Provide feedback analysis using machine learning techniques.

1. **Technologies Used**

* **Apache Spark – Data processing and analytics**
* **Hadoop & HDFS – Data storage and retrieval**
* **MapReduce – Parallel computing for large datasets**
* **Kafka – Real-time data streaming**
* **PySpark – Interface for working with Spark using Python**
* **Matplotlib & Seaborn – Data visualization**

1. **Data Processing Workflow**
   1. **Data Collection:**

**Student data including marks in multiple subjects is generated using Faker Library.**

* 1. **Data Storage:**

**Data is stored in HDFS for distributed processing.**

* 1. **Data Cleaning:**

**Handling missing values, duplicates, and formatting issues.**

* 1. **Data Transformation:**

**Converting raw data into structured format using PySpark.**

* 1. **Computing Statistics:**

**Average Marks per subject.**

**Highest & Lowest Marks per subject.**

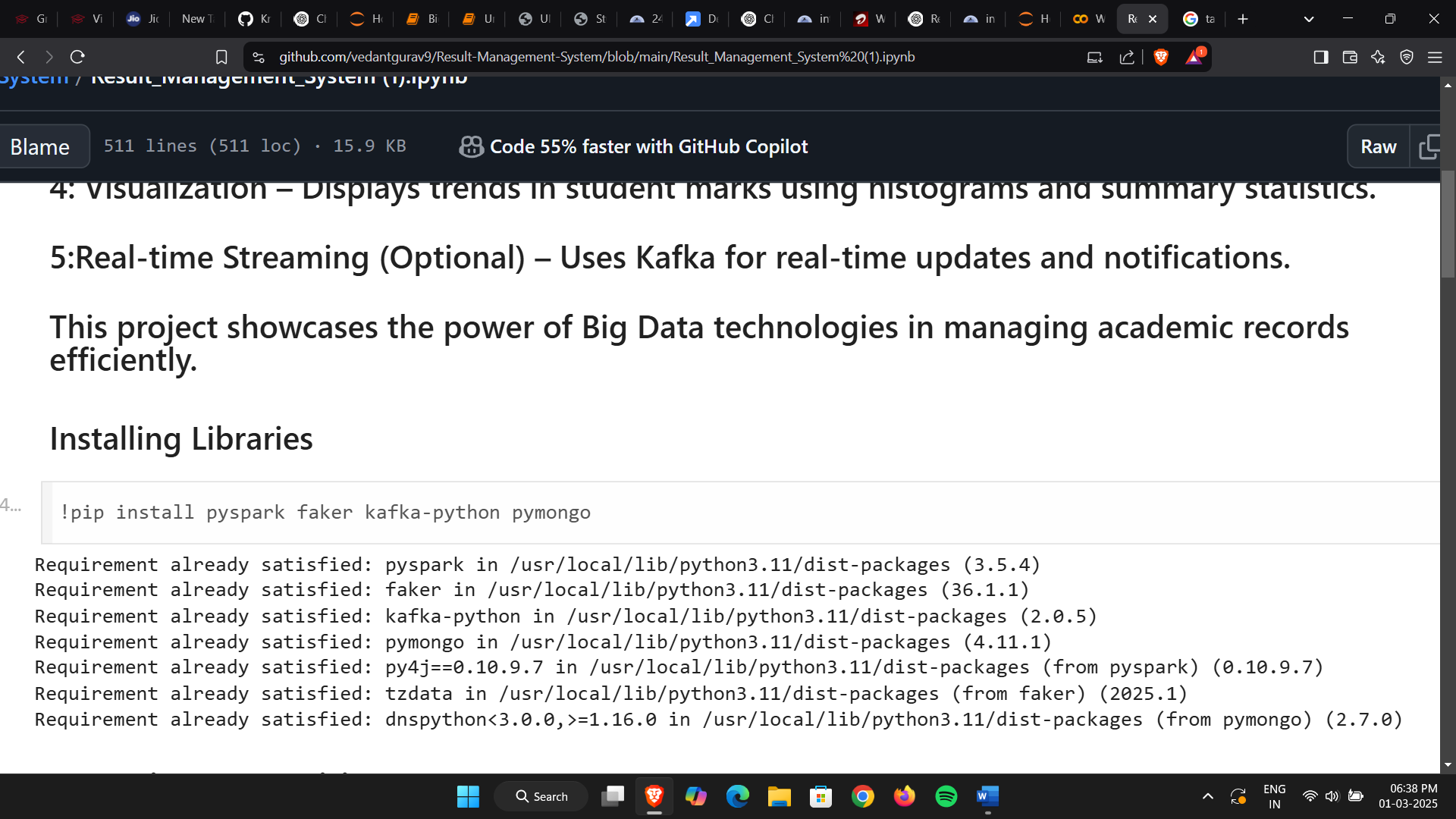
* 1. **Overall Student Performance trends.**
  2. **Data Streaming:**

**Kafka sends processed statistics for visualization.**

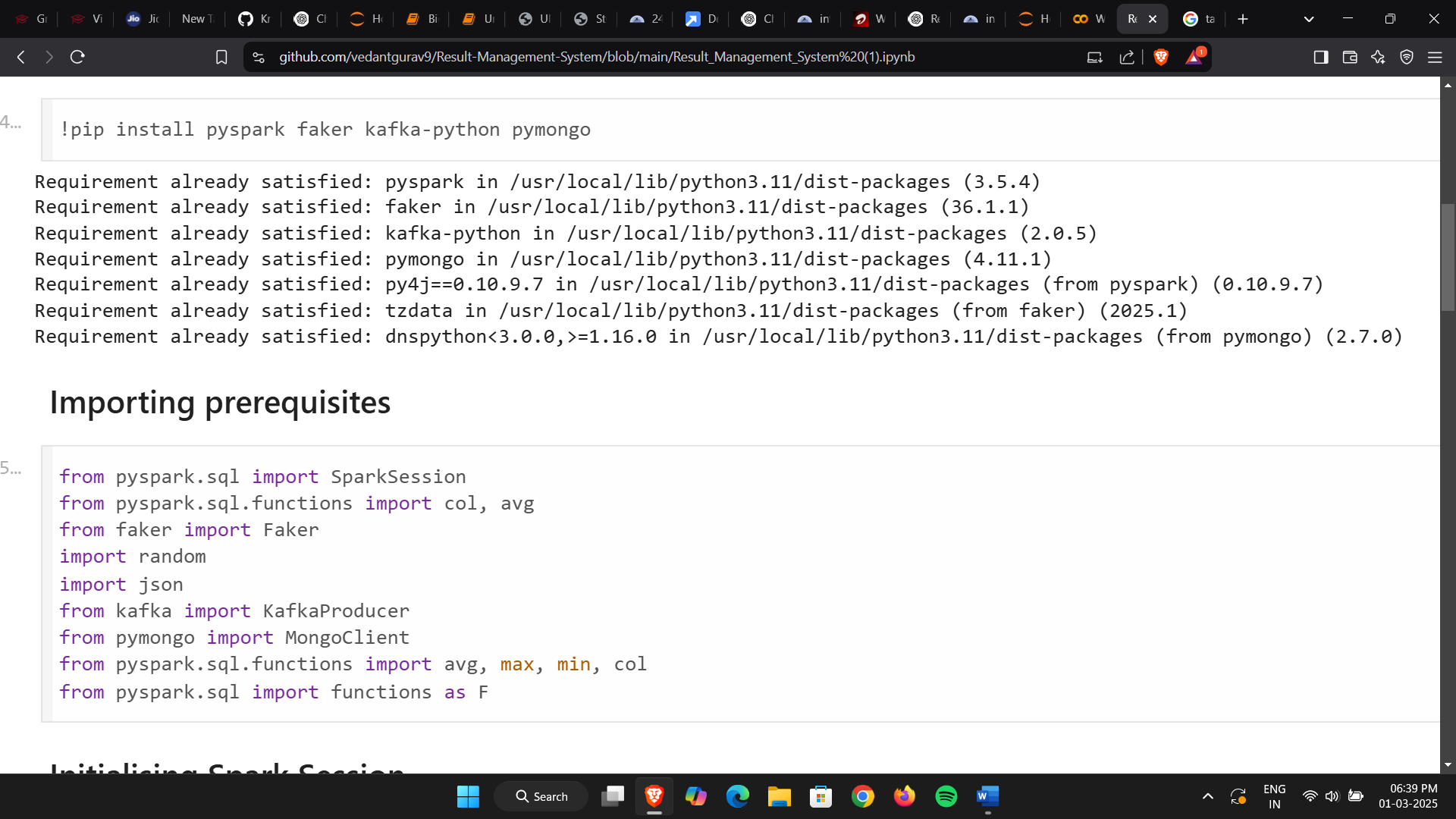
* 1. **Visualization & Reporting:**

**Data is analyzed and visualized using Matplotlib and Seaborn.**

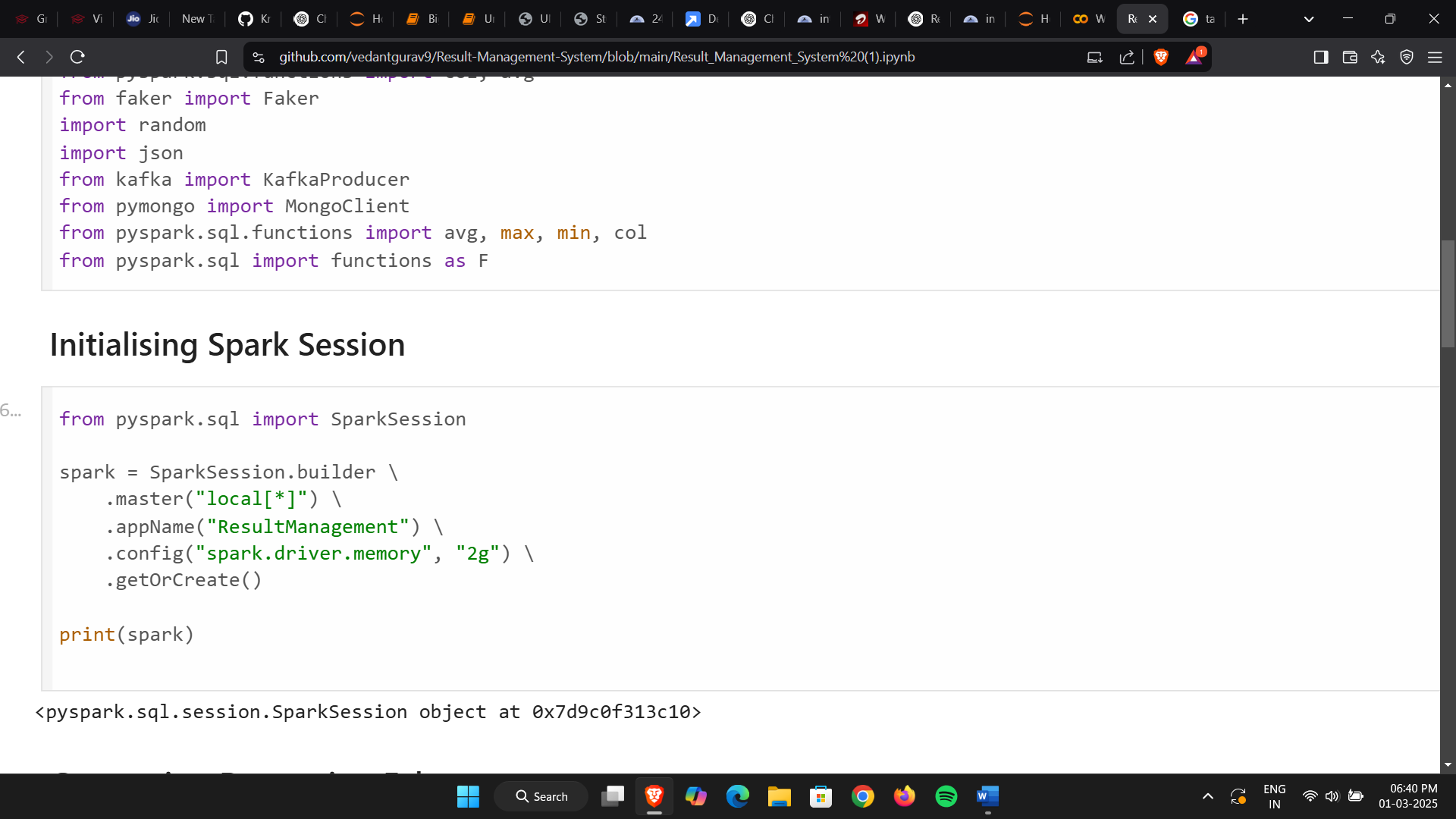
1. **Steps**
   1. **Installing Libraries**



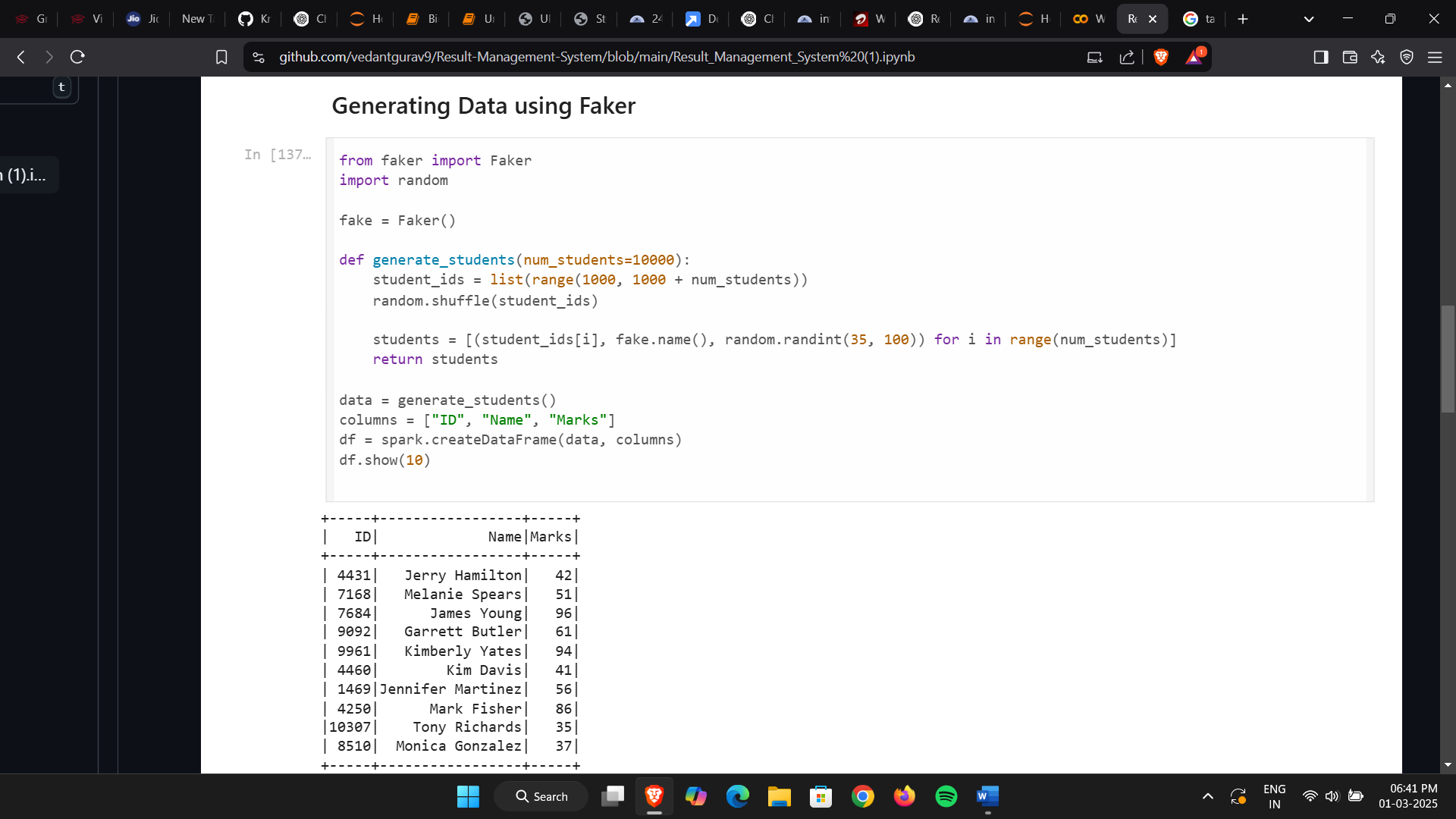
* 1. **Importing Pre requisites**



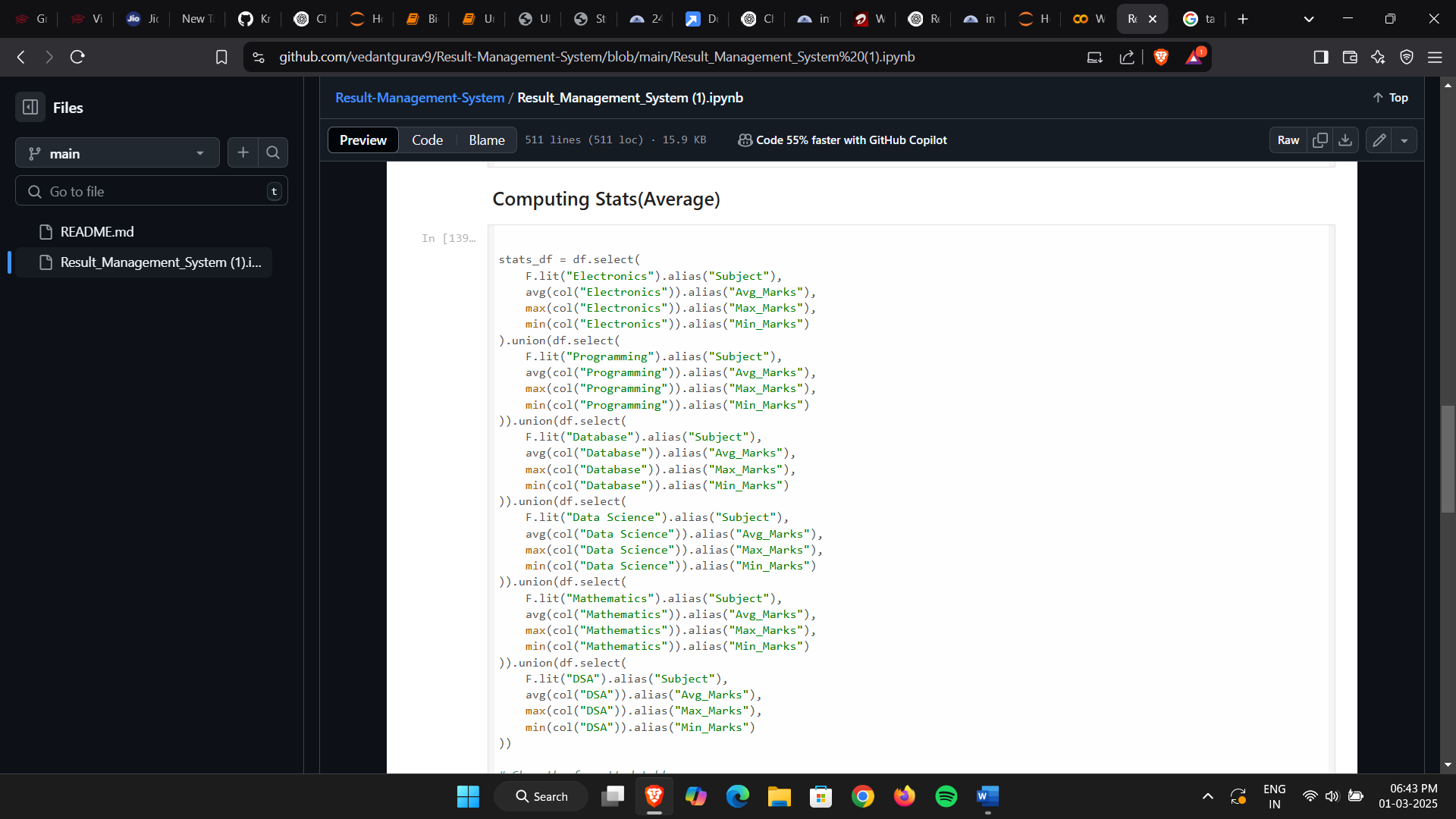
* 1. **Initialising Spark Session**



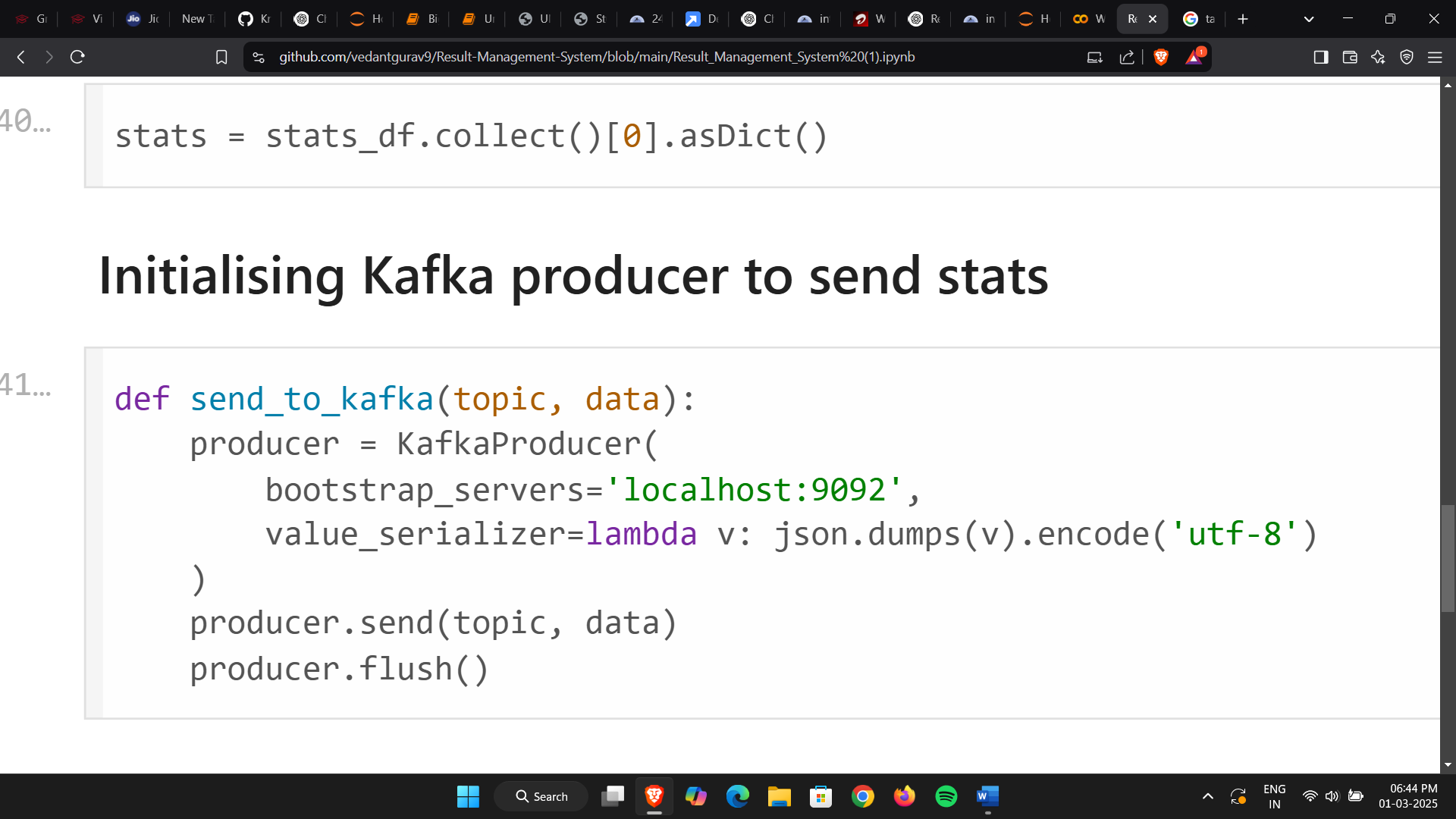
* 1. Generating Data



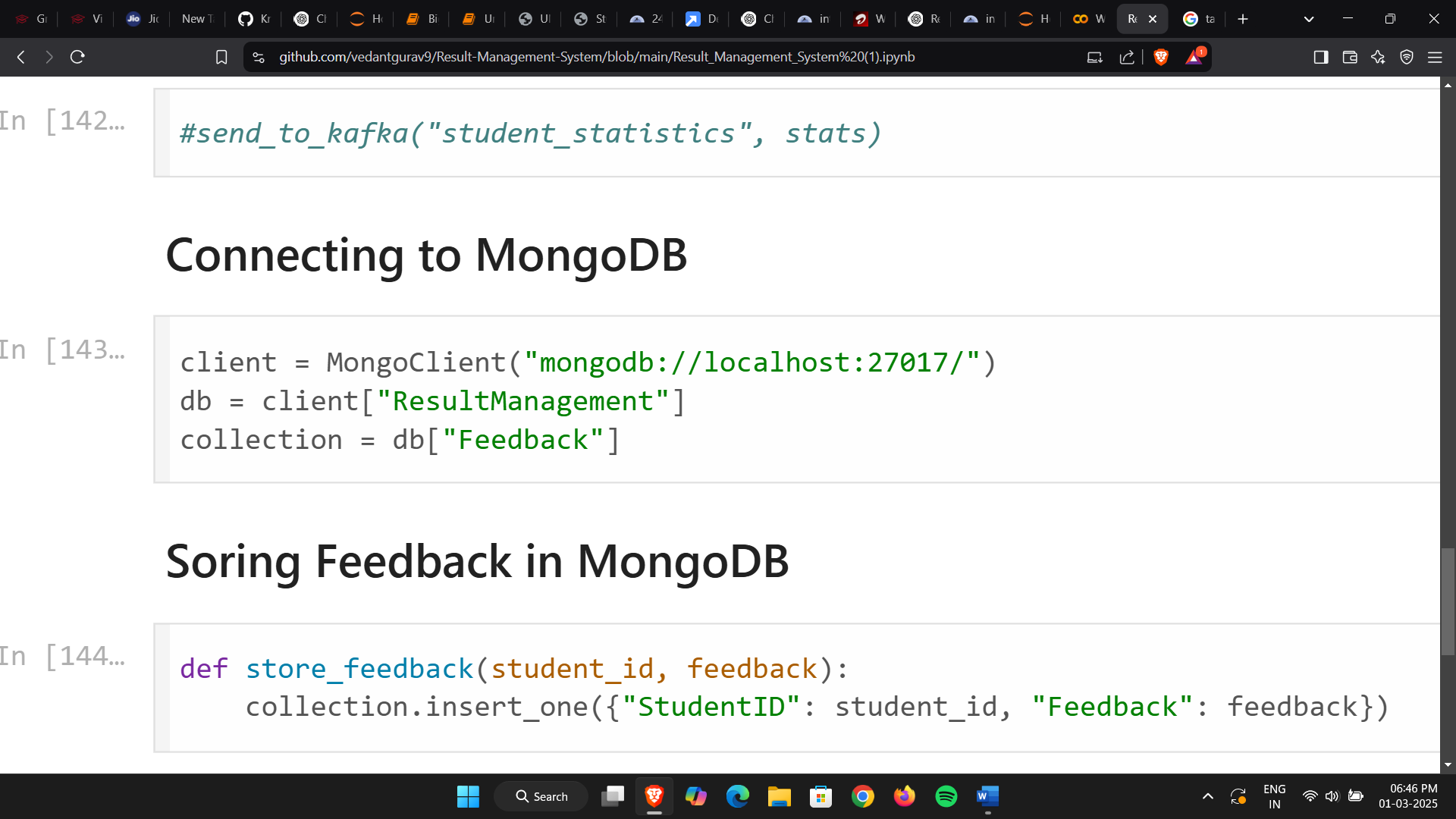
* 1. Computing Stats



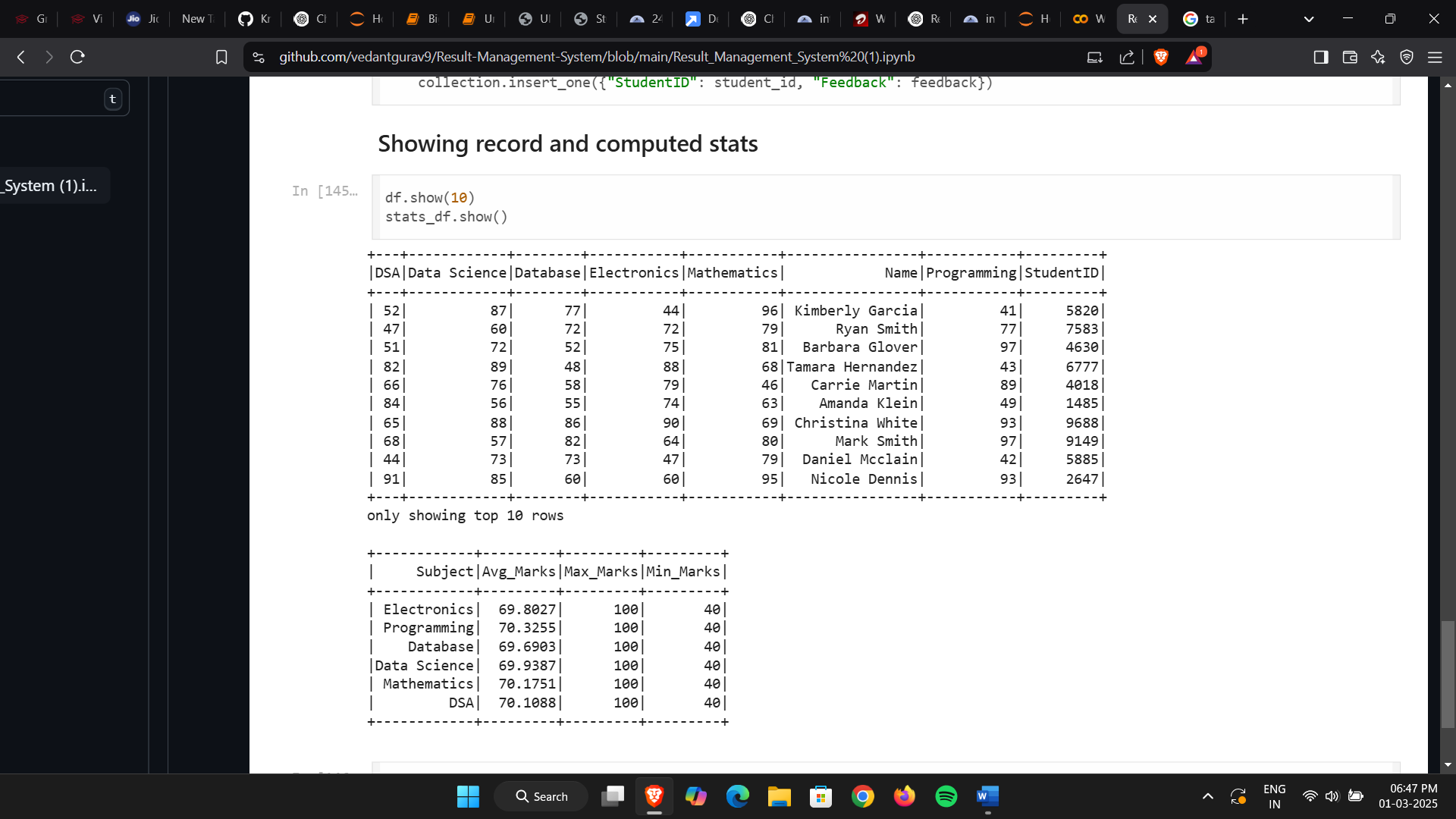
* 1. Initialising Kafka Producer to send stats



* 1. Connecting and Storing Feedback in MongoDB



* 1. Showing Record and Computed Stats



1. **Future Enhancements**

* Machine Learning-based Student Performance Prediction
* Database Sharding for efficient retrieval and storage
* Real-time Dashboards using Flask/Django
* Integration with LMS (Learning Management Systems)

1. **Conclusion**

* The Result Management System provides an efficient and scalable way to process and analyze student performance. The use of Spark, Hadoop,andMapReduce enables handling large datasets, while Kafka ensures real-time updates. The integration of data visualization enhances the interpretability of results, making it valuable for educational institutions.
* This project lays the foundation for advanced analytics, including predictive modeling and AI-driven insights to improve student performance tracking.