**Madiha Aimon Tappal**

[madihaaimon@gmail.com](mailto:madihaaimon@gmail.com)

**Data Engineering Batch – 1**

**Day – 12 Assignment**

**PySpark**

**Apache spark: -**

Apache Spark is an open-source distributed computing system that is designed for big data processing and analytics. It provides a fast and general-purpose cluster computing framework for large-scale data processing. Spark was developed to address the limitations of the MapReduce programming model, offering improved performance and ease of use.

Key features of Apache Spark include:

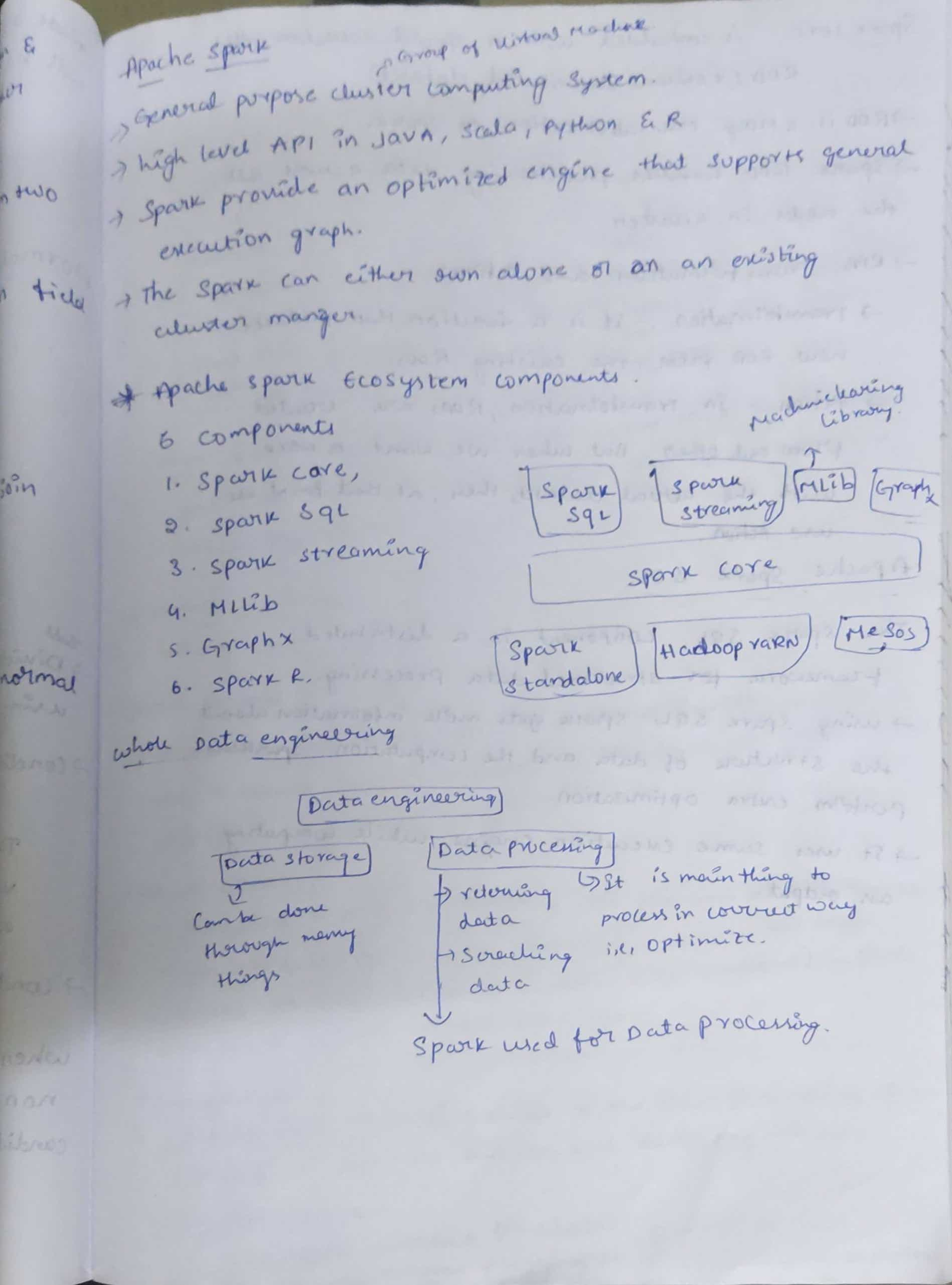
1. **Speed:** Spark is known for its speed, primarily due to its ability to perform in-memory processing. It can cache intermediate data in memory, reducing the need to read and write to disk.
2. **Ease of Use:** Spark provides high-level APIs in Java, Scala, Python, and R, making it accessible to a wide range of developers. It also includes a built-in set of high-level libraries for tasks such as SQL queries, machine learning, graph processing, and stream processing.
3. **Generality:** Spark is designed to handle a wide range of data processing workloads, including batch processing, iterative algorithms, interactive queries, and streaming.
4. **Fault Tolerance:** Spark provides fault tolerance through resilient distributed datasets (RDDs). RDDs are immutable distributed collections of objects, and they automatically recover from node failures.
5. **Compatibility:** Spark is compatible with Hadoop Distributed File System (HDFS) and can run on existing Hadoop clusters. It can also be integrated with other data sources and storage systems.
6. **Spark SQL:** Spark includes a module called Spark SQL that allows developers to query structured data using SQL as well as Data Frame APIs. This facilitates seamless integration with existing SQL-based tools and databases.
7. **Machine Learning Library (MLlib):** Spark includes a machine learning library for scalable and distributed machine learning tasks. It supports various algorithms for classification, regression, clustering, and collaborative filtering.
8. **GraphX:** Spark's GraphX library provides a distributed graph processing framework for graph analytics.

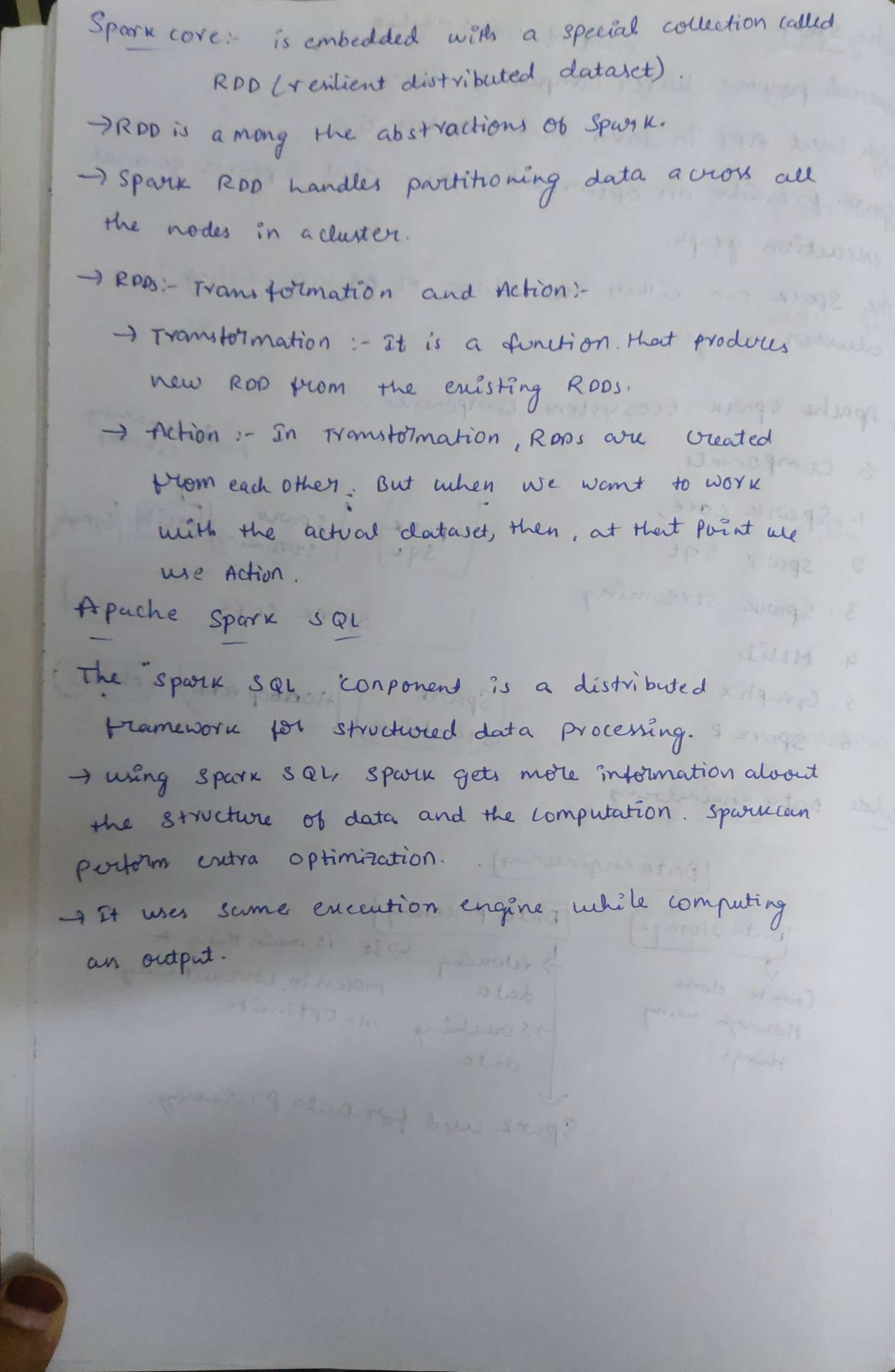
**Apache ecosystem components: -**

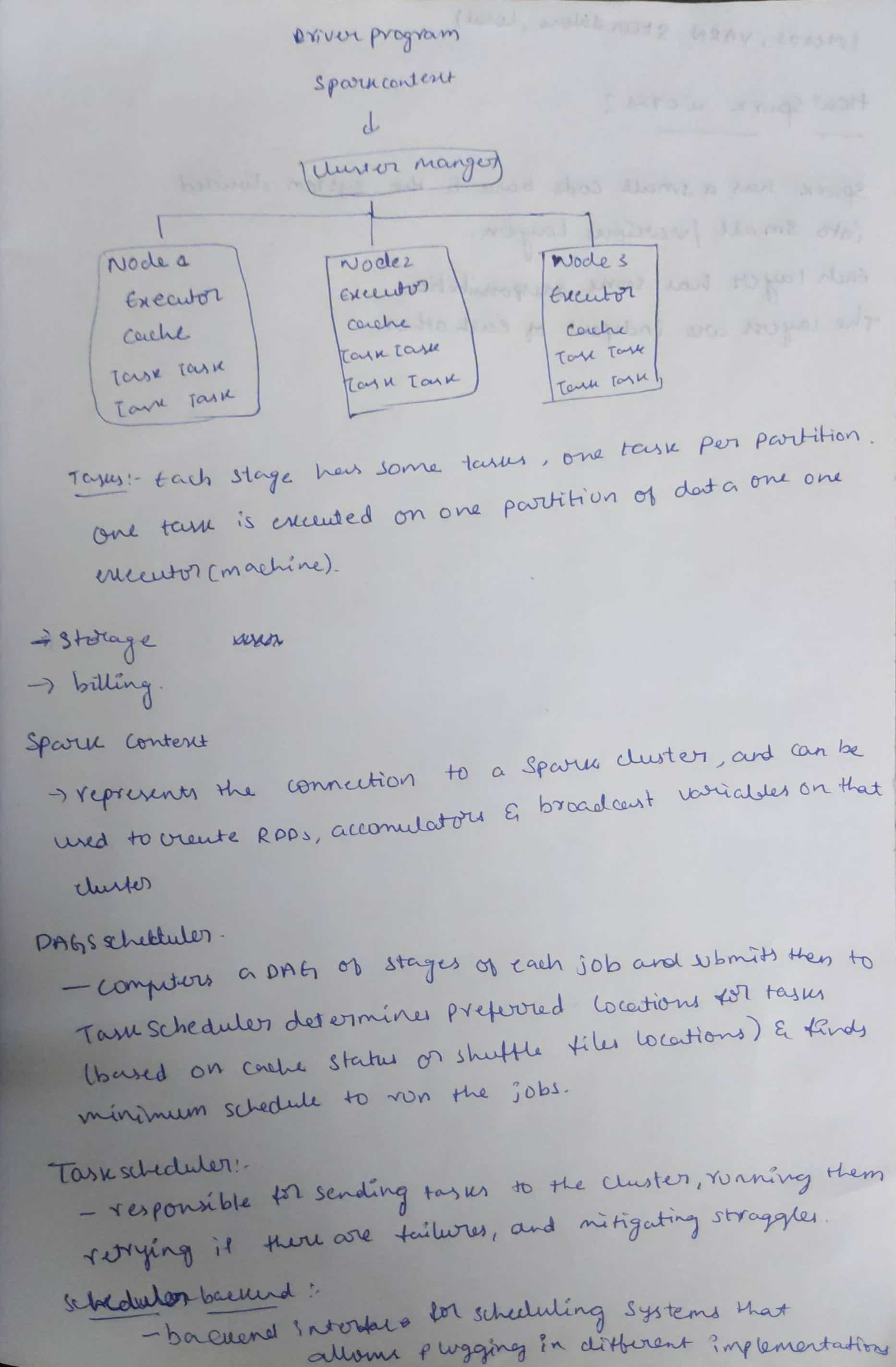
Apache Spark consists of various components that work together to provide a unified and comprehensive platform for distributed data processing. Here are brief descriptions of the six components you mentioned:

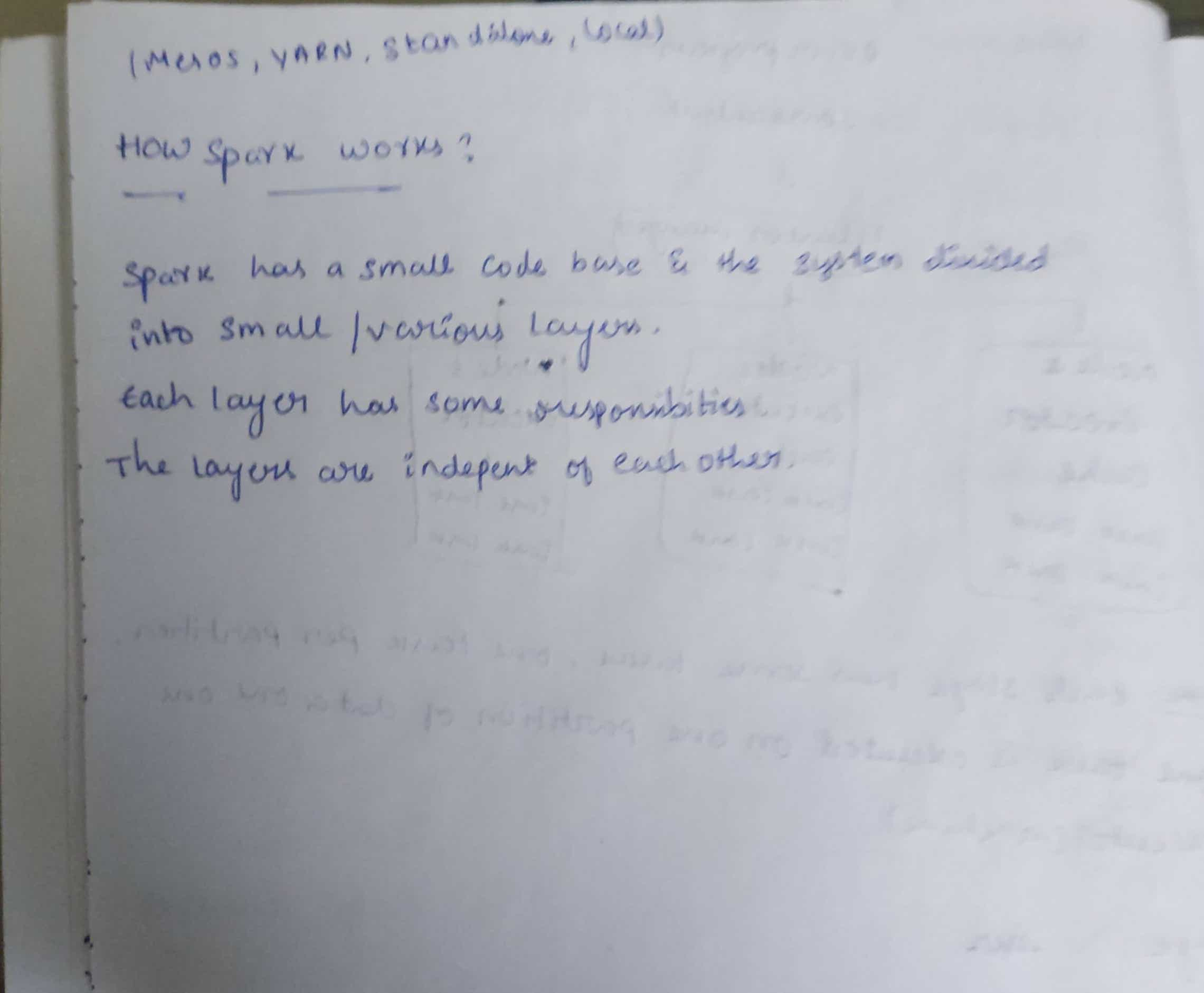
1. **Spark Core:**
   * **Description:** The foundational and distributed data processing engine of Apache Spark. It provides the basic functionality for distributed task scheduling, fault recovery, and communication between nodes in a Spark cluster.
   * **Key Features:**
     + Resilient Distributed Datasets (RDDs): Immutable distributed collections of data that can be processed in parallel.
     + Task scheduling and execution.
     + Fault tolerance through lineage information.
2. **Spark SQL:**
   * **Description:** A Spark module for structured data processing that enables the execution of SQL queries on Spark data. It provides a programming interface for data manipulation using SQL-like syntax as well as DataFrame APIs.
   * **Key Features:**
     + Supports reading data from various sources like Parquet, Avro, JSON, and Hive.
     + Integrates seamlessly with Hive.
     + Allows users to execute SQL queries alongside Spark programs.
3. **Spark Streaming:**
   * **Description:** An extension of the Spark core that enables scalable, high-throughput, fault-tolerant stream processing of live data streams. It processes data in small, micro-batch intervals.
   * **Key Features:**
     + Input sources include Kafka, Flume, HDFS, and more.
     + Windowed operations for time-based processing.
     + Enables near-real-time analytics on streaming data.
4. **MLlib (Machine Learning Library):**
   * **Description:** A scalable machine learning library for Spark that provides a variety of algorithms and tools for machine learning tasks, including classification, regression, clustering, and collaborative filtering.
   * **Key Features:**
     + Distributed implementations of common machine learning algorithms.
     + Integration with Spark's Data Frame API for seamless data processing.
     + Support for model persistence and tuning.
5. **GraphX:**
   * **Description:** A distributed graph processing framework built on top of Spark. It provides an abstraction for expressing graph computation and includes a set of graph algorithms.
   * **Key Features:**
     + Graph computation using vertex and edge RDDs.
     + Pregel API for expressing iterative graph computations.
     + Integration with Spark's ecosystem for seamless data processing.
6. **Spark R:**
   * **Description:** An R package that allows R users to interact with Spark, leveraging Spark's distributed data processing capabilities directly from the R programming language.
   * **Key Features:**
     + R Data Frame support for distributed data processing.
     + Interoperability between Spark and R.
     + Access to Spark's machine learning algorithms from R.

These components work together to enable a unified platform for batch and stream processing, machine learning, graph analytics, and interactive querying on large-scale distributed datasets.









**Installation of Apache Spark and Setup**

**Apache PySpark**

Apache Spark is an open-source distributed computing system that provides a fast and

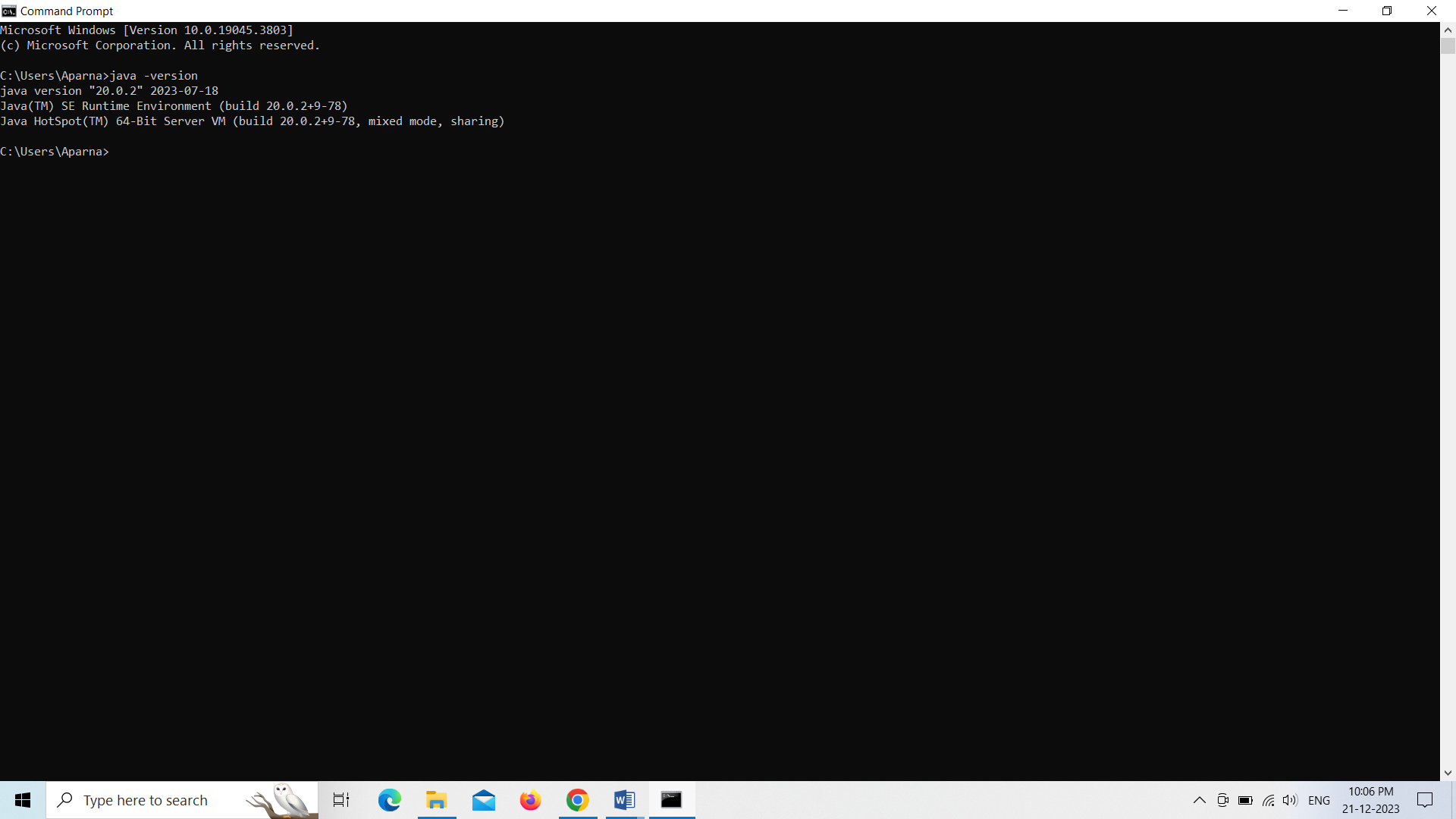
general-purpose cluster-computing framework for big data processing. PySpark is the Python

API for Apache Spark, allowing you to write Spark applications using Python.

**Python+Spark=PySpark**

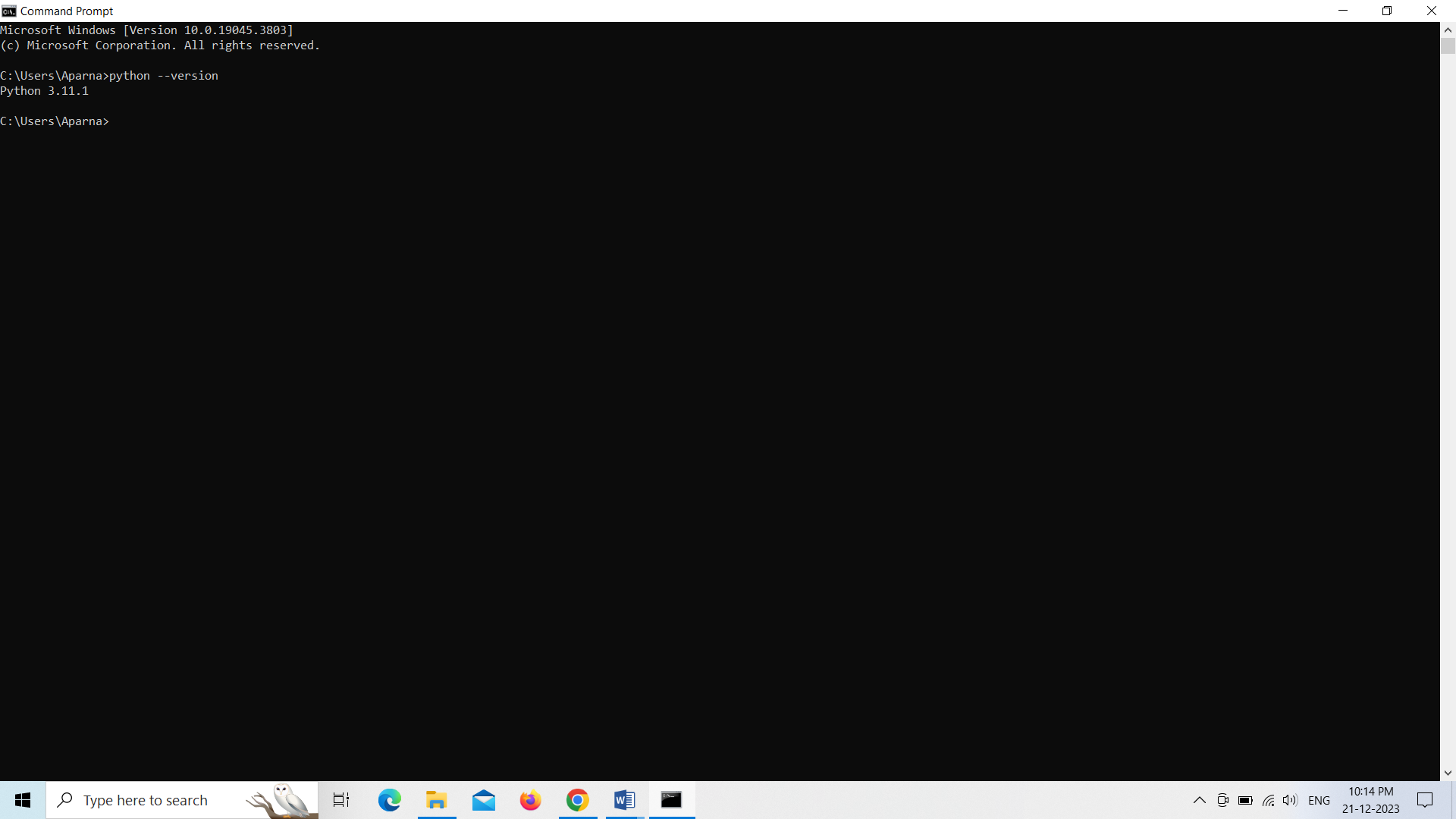
**Install Java**

* Firstly, we have to install java jdk version which is compatable to your system. It’s good to download and install latest standard version of java**.**
* After that we have to check in command prompt by typing **java –version** which displays the version of java which you downloaded.
* In means that you have successfully downloaded and installed in your system.



**Install Python**

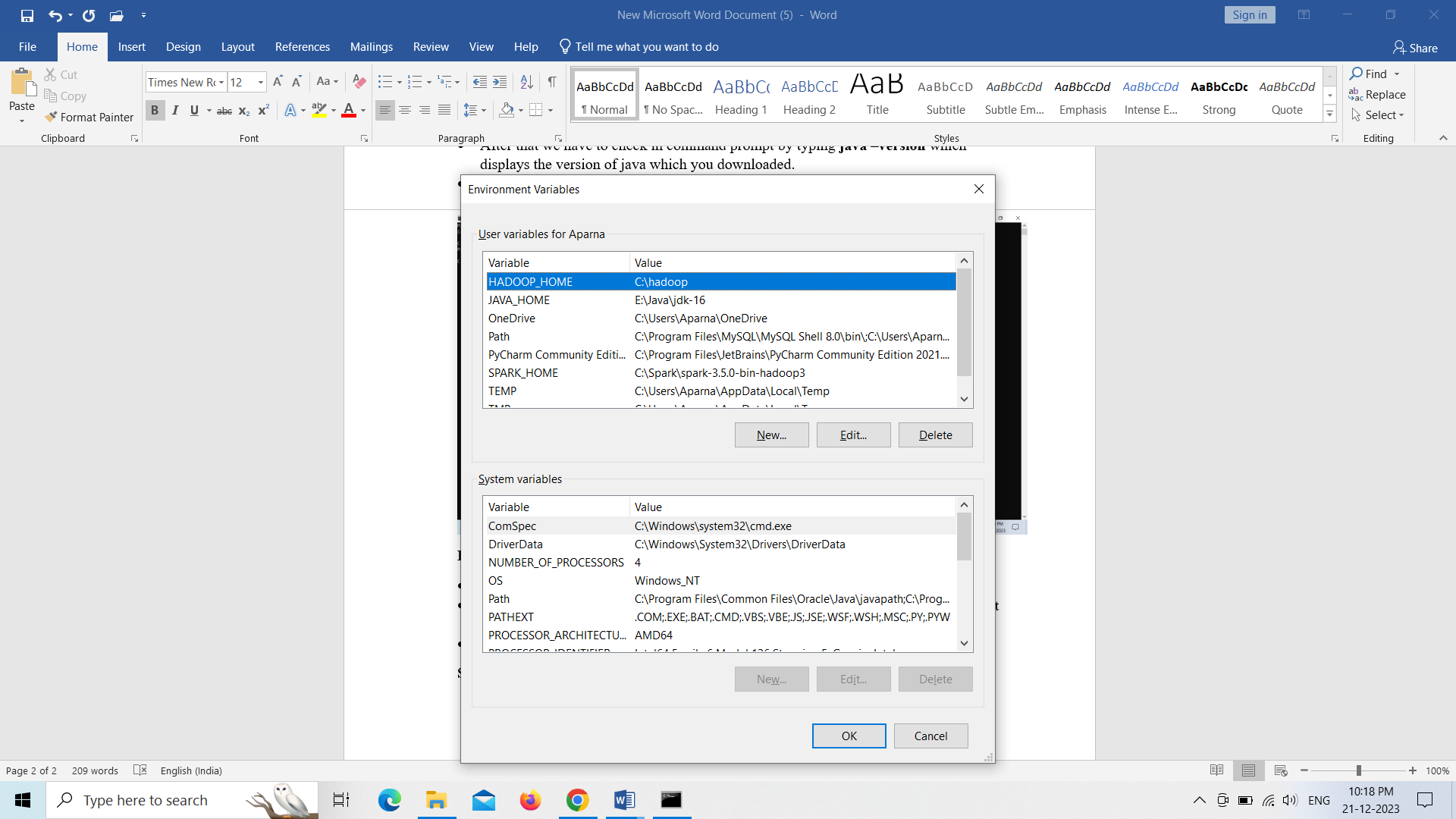
* Along with java, we have to install python environment into our system. Its good to download and install latest and standard version of python.
* After that we have to check in command prompt by typing **java –version** which displays the version of java which you downloaded.
* In means that you have successfully downloaded and installed in your system.

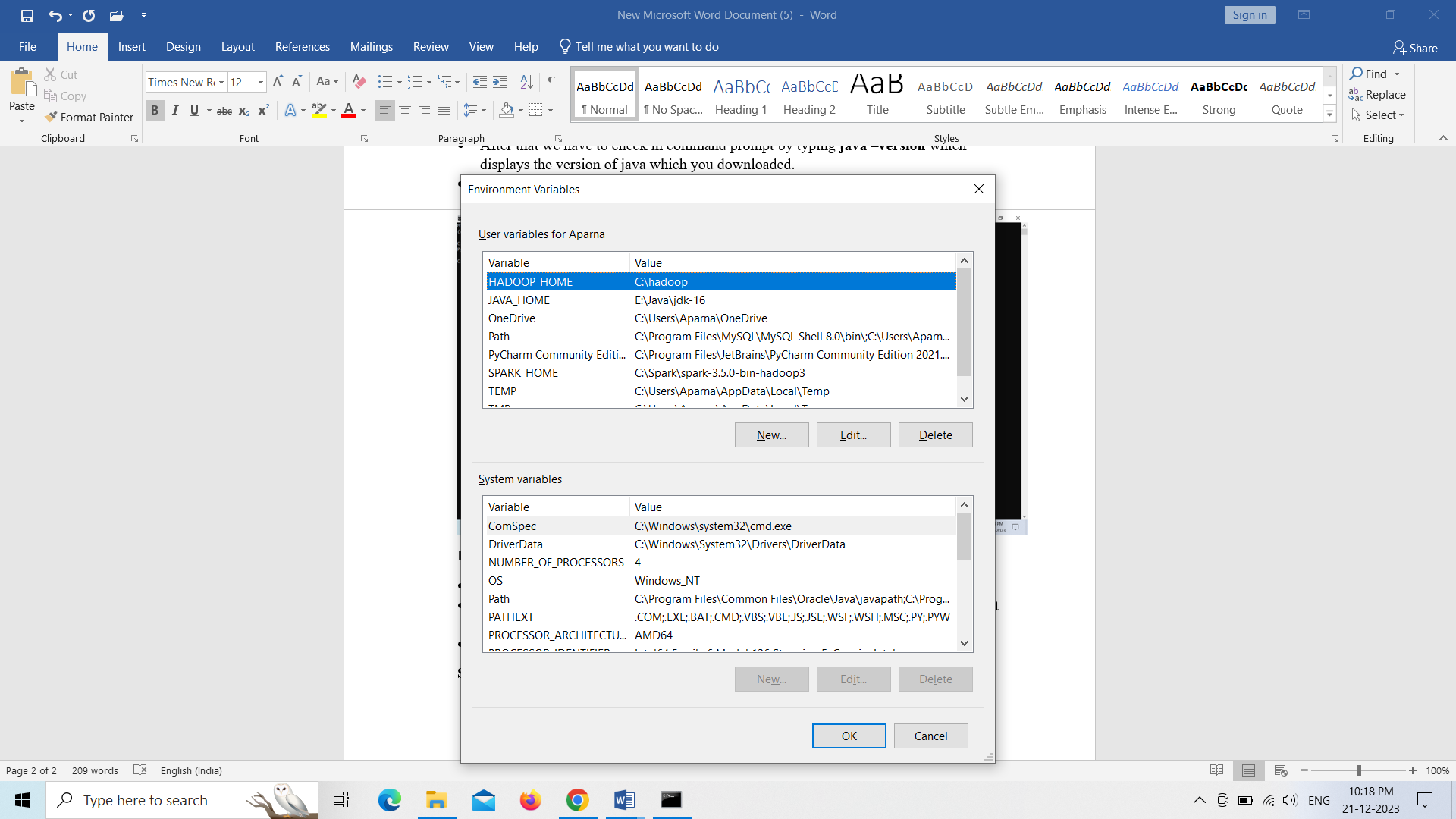


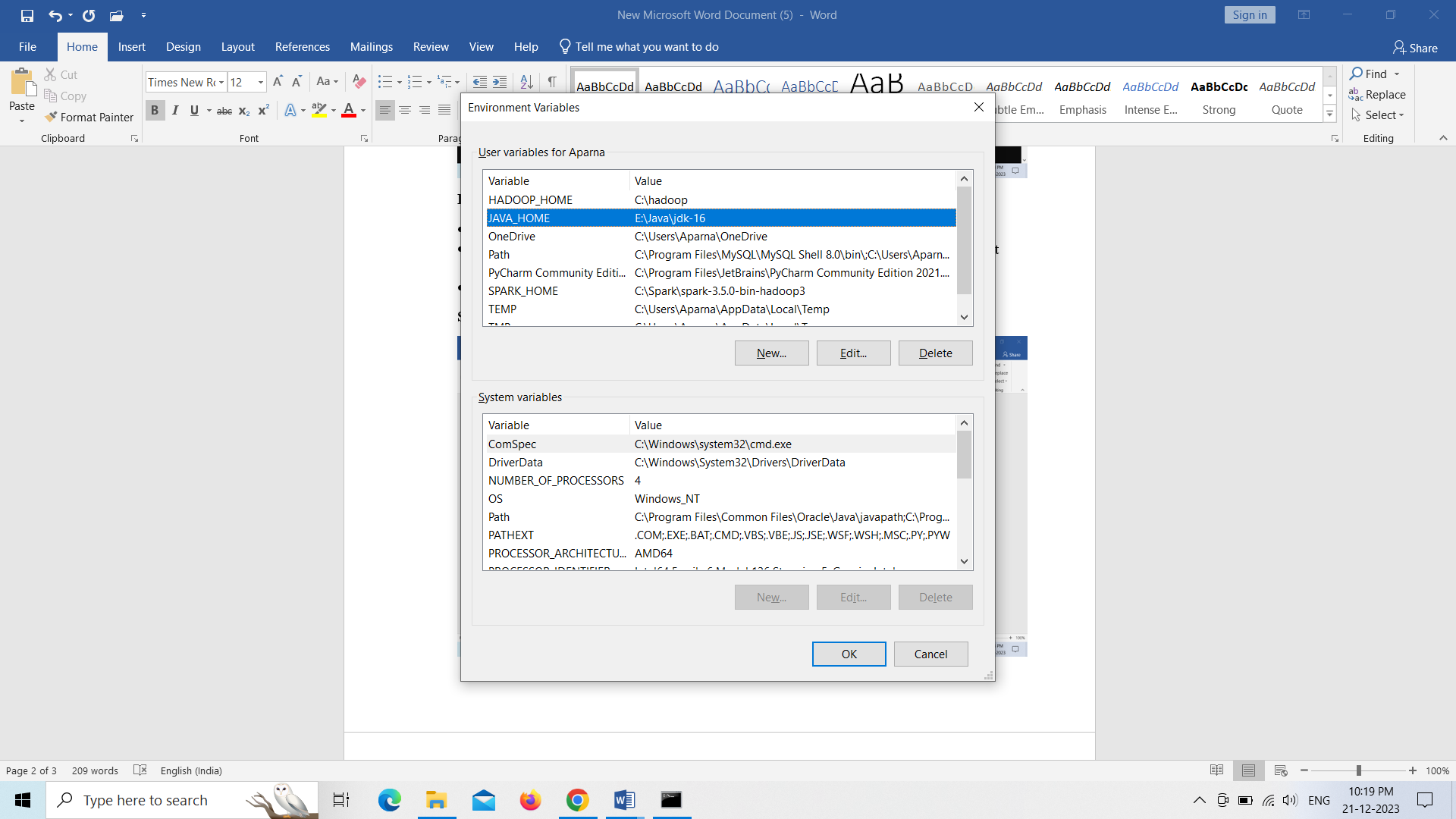
**Install Apache Spark**

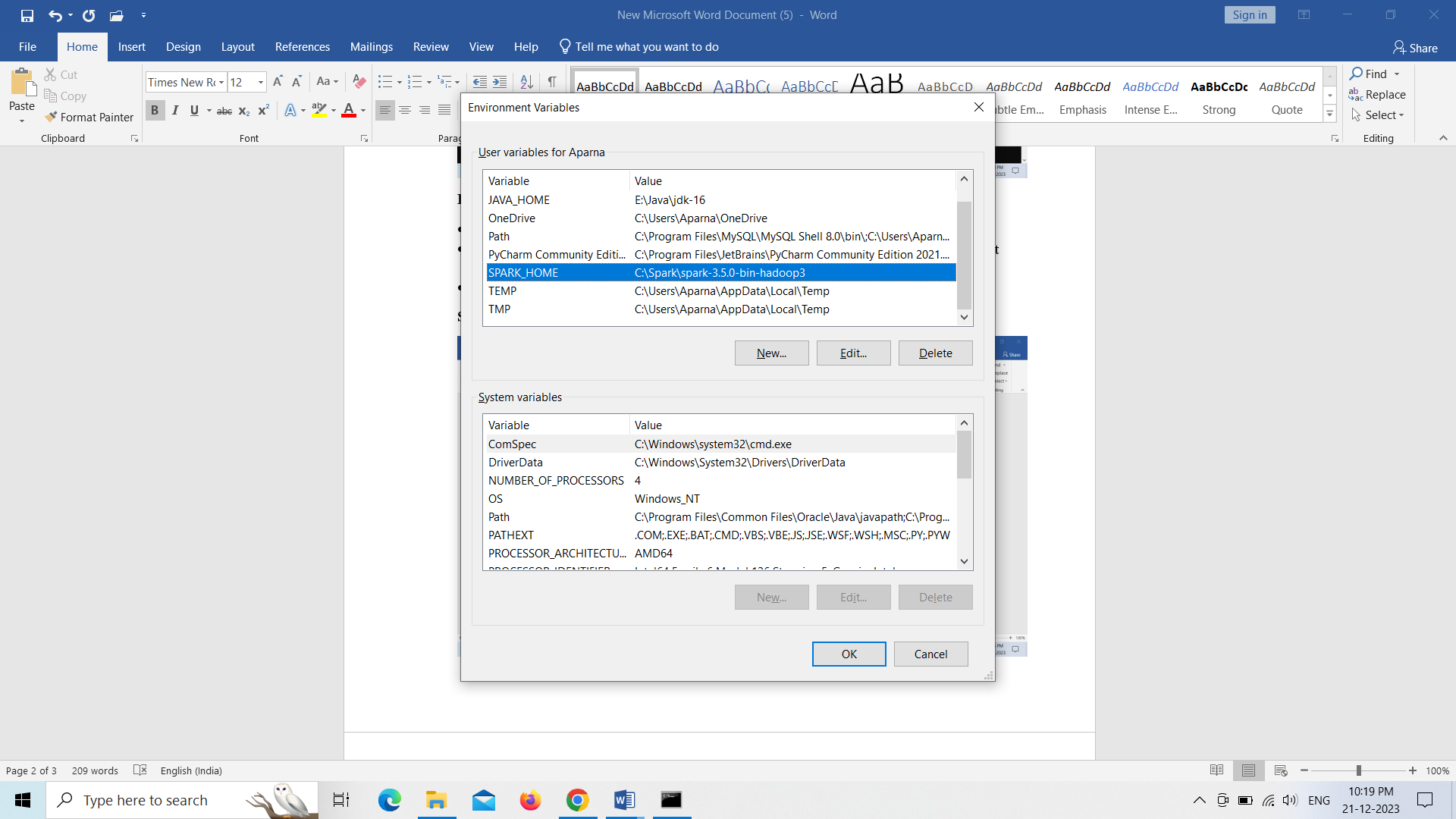
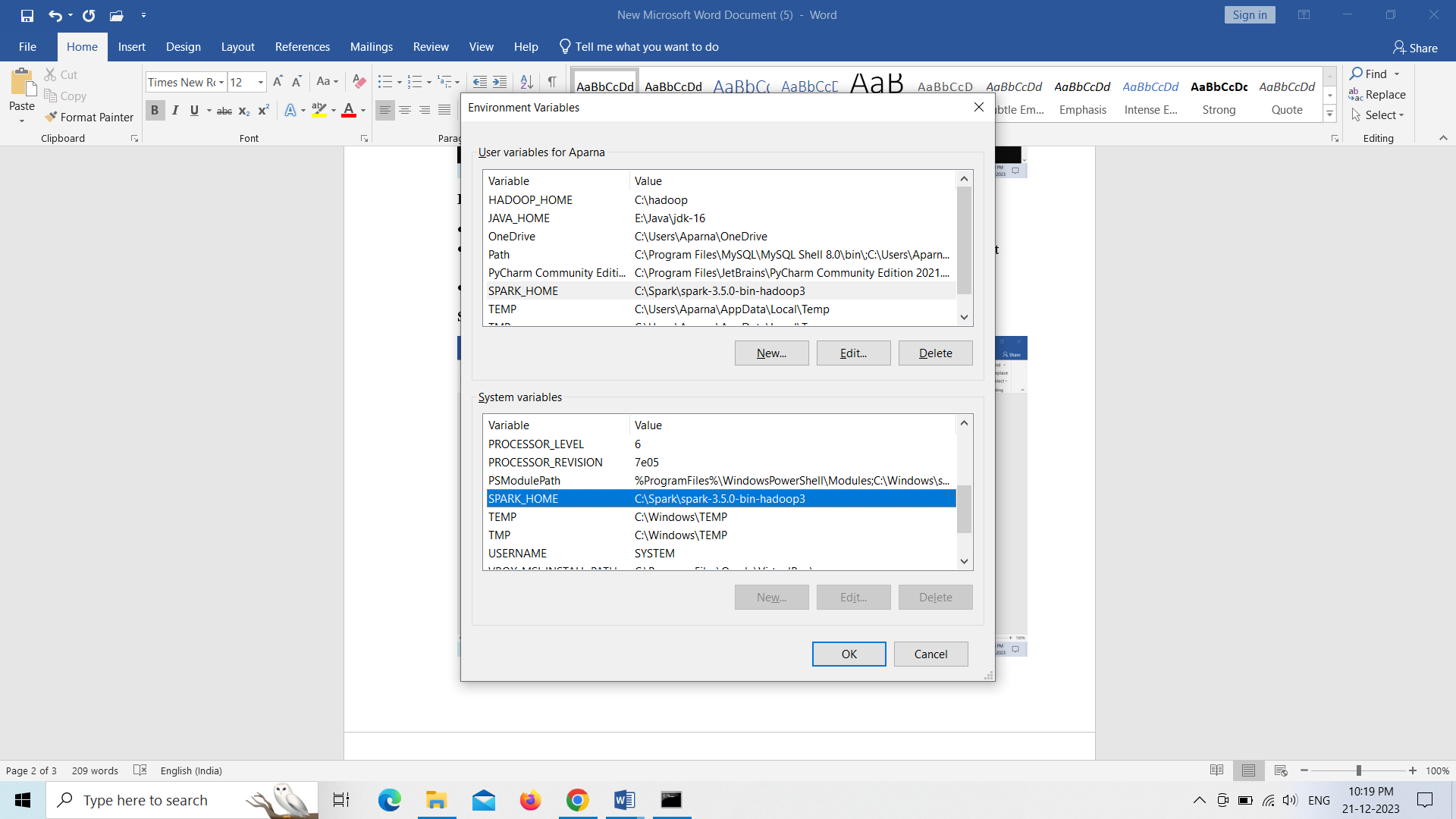
* Visit the Apache Spark download page.
* Choose the latest version of Spark and download the pre-built package for Hadoop. It will be a tarball (.tgz) file.
* Extract the downloaded tarball to a location on your machine.

**Set Environment Variables**







After setting up the environment variables, we need to save all of them and have to go to the command prompt and type **pyspark** as below.

* If it shows like these, you have successfully set up the Apache Spark setup.
* If you want to see pyspark web UI, you have to type in chrome as like
* [**http://localhost:4040/jobs/**](http://localhost:4040/jobs/)you will get the web page as follows which indicates your pyspark is installed as follows

