**PART 1. Reading Assignment [MongoDB in Action - eBook Links Provided]**

Chapter 6. Aggregation

https://learning.oreilly.com/library/view/mongodb-in-action/9781617291609/kindle_split_016.html

Chapter 7. Updates, atomic operations, and deletes

https://learning.oreilly.com/library/view/mongodb-in-action/9781617291609/kindle_split_017.html

Chapter 8. Indexing and query optimization

https://learning.oreilly.com/library/view/mongodb-in-action/9781617291609/kindle_split_019.html

Chapter 9. Text search

https://learning.oreilly.com/library/view/mongodb-in-action/9781617291609/kindle_split_020.html

**PART 2 - MongoDB indexing**

Most of the time, you'll want to declare your indexes before putting your application into production. This allows indexes to be built incrementally, as the data is inserted. But there are two cases where you might choose to build an index after the fact. The first case occurs when you need to import a lot of data before switching into production. For instance, you might be migrating an application to MongoDB and need to seed the database with user information from a data warehouse. You could create the indexes on your user data in advance, but doing so after you have imported the data will ensure an ideally balanced and compacted index from the start. This will also minimize the net time to build the index. Use the NYSE dataset to declare your indexes before putting your application into production.

**PART 3 - MongoDB Indexing**

Insert the NYSE dataset into a new database. You may use the existing NYSE database created before.

Now, create indexes on existing data sets.

**PART 4 – Programming Assignment**

All hadoop commands are invoked by the bin/hadoop script. Running the hadoop script without any arguments prints the description for all commands.

Usage: hadoop [--config confdir] [--loglevel loglevel] [COMMAND] [GENERIC_OPTIONS] [COMMAND_OPTIONS]

Execute each hadoop command once, and place the screenshots into a word file. If a command cannot be executed for any reason (such as, a distributed environment is needed), you may write the definition of the command, and skip execution.

http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html

**PART 5 – Programming Assignment**

Copy the attached 'http_access.log' file into HDFS under /logs directory.

Using the access.log file stored in HDFS, implement MapReduce in Hadoop to find the number of times each IP accessed the website.

**PART 6 – Programming Assignment**

Download and Copy all the files (http://msis.neu.edu/nyse/) (DailyPrices_A to DailyPrices_Z) to a folder in HDFS.

Write a MapReduce to find the Max price of stock_price_high for each stock. Capture the running time programmatically (or manually using a wristwatch or smartphone).

**PART 7 – Programming Assignment**

Write one MapReduce program using each of the classes that extend FileInputFormat<k,v>
(CombineFileInputFormat, FixedLengthInputFormat, KeyValueTextInputFormat, NLineInputFormat, SequenceFileInputFormat, TextInputFormat)

http://hadoop.apache.org/docs/current/api/org/apache/hadoop/mapreduce/lib/input/FileInputFormat.html

You could use any input file of your choice. The size of the input files is not important. The MR programs could simply do counting, or any other analysis you choose.