PART 1. Reading Assignment (Chapters Attached)
Chapter 5 - Queries and aggregation
Chapter 6 - Updates, atomic operations, and deletes
Chapter 7 - Indexing and query optimization

PART 2. Write a .bat/.sh to import the entire NYSE dataset (stocks A to Z) into MongoDB.
 NYSE Dataset Link: http://msis.neu.edu/nyse/

PART 3.1. Use the NYSE database to find the average price of stock_price_high values for each stock using MapReduce.

PART 3.2. Part 3.1 result will not be correct as AVERAGE is a commutative operation but nor associative. Use a FINALIZER to find the correct average. (Hint: pass sum and count from the reducer) (https://docs.mongodb.com/manual/reference/method/db.collection.mapReduce/index.html)

PART 4. Modify PART 3 by adding a finalizer to find out the average stock price of each price of all stocks in the finalizer.

PART 5 - 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 6 - 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 7 - Programming Assignment
MapReduce to find each of the followings:
    Task 1. MapReduce to find the top 25 rated movies in the movieLens dataset
    Task 2. MapReduce to find the number of males and females in the movielens dataset
    Task 3. MapReduce to find the number of movies rated by different users