**Big Data Assignment 3**

1.)

Since it is a price comparison website, the computational power needed is not that large. Especially since the number of web-shops/shops selling a certain item is limited which results in approximately 30-50 rows per item where each row represents a seller and its price. Additionally, further information is being provided about the product once a user clicks on it. The data for the prices must be updated once a day, whereas the technical details remain the same.

Therefore, a relational database system is being chosen in this case. More specifically, an SQL Aurora database on AWS is being chosen. If one wants to ensure that the requirements for performance, scalability and reliability are satisfied, one must consider several aspects regarding a proper implementation. **Scalability**: A scaling-out solution running on a distributed database is being chosen which mitigates the risk of default (**reliability**). Moreover, during primetimes such as in the evening or at the weekend, more nodes can easily be added which addresses the aspect of **performance**. Finally, it is also more affordable.

The architecture used will be a “Shared nothing” which is typically to be found in web-scale systems. Therefore, this seems to be a perfect fit.

Taking everything into consideration, distributed key-value stores are being used which is characterized by the following points:

* Provide transactional access to key-value pairs
* Focus on fault-tolerance and transaction speed
* Typical use-cases are backends for web applications.

The database is being linked to the website with an API. Another solution could be that simple SQL queries can be executed on the server whose results will be displayed on the website. Both options seem to be plausible, and it can be decided in the future, which approach is the final one.

2.)

A particular advantage of this approach is that regular SQL can be used. Therefore, it should be rather simple to build the backend properly compared to the other systems. Most of the functions of the website can be implemented in native SQL and run directly on the database without using any other tool. Additionally, there are quite a lot of options available to choose from regarding the provider.

3.)

Since it is the least advanced approach compared to MapReduce and Spark, performance could be a problem in comparison to the other two solutions. But as mentioned before, the computations involved are not huge since most prices are mostly just updated once a day. Therefore, it should be fine.