Q1) **LocalStorage:** it stores the data and the data remains persistent even after the window is closed. Here the data retrieval is quick and has good performance. It basically caches the data in local storage which is why it has good performance.

**Session Storage:** Both local storage and session storage are similar except for some areas, here the data doesn’t persists. Once the browser is closed then the session also gets removed.

**Cookies:** they are used for storing small amount of data which is sent to the server when it receives the request from the client.

Q2) output we get is: 5,5,5,5,5, Since setTimeout is an asynchronous function it will execute after the loop is completed. By the time async function executes the value of I becomes 5, therefore we get 5 as output.

Q3) Sharding allows MongoDB to scale multiple servers horizontally also providing improved performance and high availability .

Q4)Promise chaining is used to handle the chaining of asynchronous operations which takes place one after the other. It is used to simplify the data flow.

Q5) Higher order component is a component which takes another component as an input and returns a new component.

Q6) Callback hell is referred to the multiple nested callbacks which makes the code very complex and it reduces the readability of the code.

We solve this problem using Promises:

Async/ await:

Async function fetchData(){

Try{

Const data = await fetchDataAPI();

Console.log(‘Data Received’, data);

}catch(error){

Console.log(“Error while fetching the data”, error);

}

}

Q8) output: 1, 4, 3, 2, the delay argument in the setTimeOut function takes the number in miliseconds, so among 2 and 3, 3 will be executed first. Even though 3 has 0 miliseconds still it will let the synchronous code to complete first then only async function will execute.

Q9) array 1: length=5 last=j,o,n,e,s

array 2: length=5 last=j,o,n,e,s

Q10) Output is: 1, 4, 2, 3 , because the setTimeout functions are asynchronous operations and they take time to execute. It takes two arguments i.e. callback and delay.