Assignment Questions

1. Synchronous and Asynchronous have a very basic common difference and that lies in there waiting time which they take to complete a task for instance in case of Synchronous Programming Languages they wait until the previous process is executed whereas in the case of Asynchronous Programming languages if the time taken by a process is large then the control would shift to the next line and it will get executed and once it is done the line which had been taking time then that is going to be executed.
2. Web API’s (Application Programming Interfaces) are interfaces provided by web browsers that allow developers to interact with various web technologies and access features like manipulating the DOM, making HTTP requests, handling timers, working with local storage.
3. setTimeout and setInterval.
   1. setTimeOut is a function that schedules the execution of a callback function after a specified delay in milliseconds.
   2. setInterval is similar to ‘setTimeout’, but it repeatedly executes the callback function at a specified interval until cleared.
4. Handling Async Code is JavaScript function like the following:-
   1. Async code in JavaScript can be handled using callbacks, promises, and async/await.
   2. Callbacks are functions passed as arguments to other functions, allowing them to be executed asynchronously.
   3. Promises provide a more structured way of handling async operations and allow chaining multiple operations together.
   4. async and await keywords provide a syntactic sugar on top of promises, making async code look more like synchronous code.
5. The major difference between **CallBacks and CallBack hell** are:-
   1. Callbacks are functions passed as arguments to other functions to be executed later, often used in async programming.
   2. Callback Hell refers to the situation where multiple nested callbacks are used, resulting in unreadable and difficult-to-maintain code.
6. Promises and Promise Methods could be defined as the following :
   1. Promises are objects that represent the eventual completion (or failure) of an asynchronous operation. They provide a more structured way of handling async code.
   2. Some common Promise methods include ‘then’ for chaining callbacks, ‘catch’ for error handling, and ‘finally’ for performing cleanup tasks.
7. Both ‘async’and ‘await’ are one of the most crucial elements of development through javaScript as:
   1. async and await are keywords introduced in ES2017 (ES8) to simplify asynchronous programming.
   2. async is used to define an asynchronous function, which automatically returns a Promise.
   3. await is used inside an async function to pause the execution and wait for a Promise to resolve. It can only be used inside async functions
8. The Try and Catch blocks are used for error handling in JavaScript. Code inside the **try** block is executed, and if an error occurs, it is caught by the **catch** block. The **catch** block contains the error handling logic, allowing you to handle the error gracefully.

1. In ES6 a new keyword **fetch** was introduced. **fetch** is a modern JavaScript function used for making HTTP requests. It returns a Promise that resolves to the response of the request. It provides a more flexible and powerful alternative to traditional XMLHttpRequest.
2. To define an asynchronous function, prefix the function declaration or expression with the **async** keyword. Inside the async function, you can use the **await** keyword to pause the execution and wait for a Promise to resolve before continuing.