Assignment No.5

Q.1 What's the difference between Synchronous and Asynchronous?

Ans :- Synchronous and asynchronous are two different approaches to handling tasks and managing the flow of execution in programming. Here's the difference between the two:

Synchronous:
- In synchronous programming, tasks are executed one after the other, in a sequential manner.
- Each task must complete before the program can move on to the next task.
- Synchronous tasks are blocking, meaning that they can cause the program to pause and wait until the task is finished.
- Synchronous programming is straightforward and easier to understand, as the execution flow follows a predictable order.
- However, it can lead to inefficiency if tasks have to wait for I/O operations or other long-running tasks to complete.

Asynchronous:
- In asynchronous programming, tasks can start and run independently of each other.
- Tasks can be initiated and allowed to run in the background while the program continues executing other tasks.
- Asynchronous tasks are non-blocking, meaning that they don't cause the program to wait for their completion.
- Asynchronous programming is useful when dealing with I/O operations, network requests, or any tasks that involve waiting for external resources.
- It can improve performance and efficiency by allowing multiple tasks to run concurrently.
- However, it can be more complex to understand and requires the use of callbacks, promises, or async/await syntax to handle the results of asynchronous task.

Q.2 What are Web Apis ?

Ans :- Web APIs (Application Programming Interfaces) are sets of rules and protocols that allow different software applications to communicate and interact with each other over the internet. They define a collection of methods, data formats, and conventions that enable the exchange of data and functionality between different systems.
Web APIs are widely used in web development to enable the integration of different services, platforms, and applications. They provide a standardized way for developers to access and manipulate resources and data on remote servers.

Q.3 Explain SetTimeOut and setInterval ?

Ans :- setTimeout and setInterval are JavaScript functions that allow you to schedule the execution of code at specified time intervals. They are commonly used in web development for implementing timers, animations, and periodic tasks.

Q.4 How can you handle Async code in JavaScript ?

Ans :- In JavaScript, there are two common ways to work with asynchronous operations: then/catch method chaining and async/await . Both methods can be used to handle promises, which are objects that represent the eventual completion (or failure) of an asynchronous operation.

Q.5 What are Callbacks & Callback Hell ?

Ans :- Callbacks are functions that are passed as arguments to other functions and are executed at a certain point in the program or when a particular event occurs. They allow you to specify the behavior that should happen after an asynchronous operation completes.

Callback Hell, also known as the Pyramid of Doom, is a situation that arises when working with multiple asynchronous operations in JavaScript, especially when they depend on each other. In such cases, callbacks are nested inside each other, resulting in deeply nested and indented code that can become difficult to read, understand, and maintain.

Q.6 What are Promises & Explain Some Three Methods of Promise?

Ans :- Promises are useful when you have to handle more than one asynchronous task, one after another. For that, we use promise chaining. You can perform an operation after a promise is resolved using methods then() , catch() and finally() .

Q.7 What’s async & await Keyword in JavaScript?

Ans :- The purpose of async / await is to simplify the syntax necessary to consume promise-based APIs. The behavior of async / await is similar to combining generators and promises. Async functions always return a promise.

Q.8 Explain Purpose of Try and Catch Block & Why do we need it?

Ans :- A try block is the block of code (contains a set of statements) in which exceptions can occur; it's used to enclose the code that might throw an exception. The try block is always followed by a catch block, which handles the exception that occurs in the associated try block.

Q.9 Explain fetch?

Ans :- fetch() is a built-in JavaScript function used to make HTTP requests to a server and retrieve resources, such as JSON data, HTML content, or binary files. It provides a modern and

more flexible alternative to the older XMLHttpRequest (XHR) object for making network requests.

Q.10 How do you define an asynchronous function in JavaScript using async/await?

Ans :- To define an asynchronous function using async/await in JavaScript, you can prefix the function declaration with the async keyword. This indicates that the function will contain asynchronous code and allows you to use the await keyword inside the function to pause the execution and wait for a Promise to resolve.