Q2: promises, Async Await:

1. To handle many callback function as asynchronous, promises are used as it uses callbacks as resolve and reject.
promise actually starts with pending status and stays there until gets resolved or rejected.

it has methods as .then, .catch and finally.

· then is used to catch the status if it is resolved.
· catch is used to catch the status if it is rejected.

· finally is to do next of .then or .catch.

2. Async / Await:

Async function is a advanced function model that directly gives back a promise of status resolve if the desired object is found or reject only if desired object is not found.

For example:

```
async function fetchData(){
 try {
  let response = await fetch('.....');
  let data = await response.json();
   console.log(data)
   }
Catch (error) {
    console.error ("Error fetching data:", error);
    }
    }.

    fetchData();
```

Here, like promises try, catch is used to output or execute the resolved part and rejected part.

3, Error handling :.

```
1. Let promise1 = new promise ((resol, rej ){
        reject ("Error occured");
        };

    promise 1. catch((err) => console.log (err));

2. Async function fetchData () {
      try {
        Let response = await fetch ("......")}
        }
        catch (err){
          console.log (err );
          }
        }
    fetch Data ();
```

+ In first promise, the promise gets rejected and error handled.
As same in Async await no data fetched so, its error also handled in catch block.

ну