Full stack Questions

Q1. Differnce Between setTimeout,promises and async/await?

Ans1. setTimeout -> ”means do this later” .It Puts your task on a big waiting list(macrotask) to run after everything else is done.

* Promise.then()->”means do this as soon as the current work is done”

goes in microtask Queue executes before setTimeout().

* Async/await -> is just a cleaner way to write promises. Await pauses you function until the promise finishes but behind the scenes.

Q2.Differnce between let,var and const?

Ans2

* Var->old way ,it’s function scoped and can be redeclared and hoisted(i.e. you can use it before declaring gives undefined).
* Let->modern and faster ,it is block scoped and can change value but cannot be redeclared.
* Const-> fixed , it is block scoped and cannot be changed or redeclared.

Q3.Differnce between Prototype and \_\_proto\_\_ ?

Ans3

* Prototype ->Object is attached to function and is used when creating new instances.
* \_\_proto\_\_ ->It’s a link that points to it’s parent prototype.

Q4. Differnce between call,apply and bind ?

Ans4

* Call -> calls functions immediatelty and arguments are seprated by comma.
* Apply -> calls functions immediately and argument are seprated by array.
* Bind -> returnsnew function call ,can call later.

Q4. Difference Between Shallow copy and Deep copy?

Ans4

* Shallow copy -> makes the copy of only the top level of the object and It has another object inside it and that inner one is still shared.
* Deep copy ->makes a complete copy including all inner objects. And changes anywhere doesn’t affect the original one.

Q5. Why does changing a nested object in J.S affects the original one?

Ans5 Because in JS objects are copied by reference not by values.that means when we assign or shallow copy an object we are not copying the actual data,we are just copying the memory address.

Q6. How does Js manage memory?

Ans6 J.S has automatic garbage collection,you don’t need to manually free ethe memory.When an object is no longer reachable Js automatically removes it from the memory.

Q7.When can Closure cause memory Leaks?

Ans7 -> Closure remembers variables even after the outer function ends.  
If they keep large data that’s never used or released the memory never gets freed and it causes memory leaks.

Q8.What’s the role of mark and sweep algorithm?

Ans8-> It’s the garbage collector cleaning method.In this Method J.S marks all values that are still in use. And then it sweeps away everything else.

Q9.Explain Debounce vs Throttle in JS?

Ans9 -> Debounce ->Waits until you stop doing something before running the function.

* EX. Typing in a search bar it waits till we stop typing to send the api request.

Throttle.-> Runs the function at regular intervals no matter how many times you trigger it.

* Ex. Scroll event -even if we scroll like crazy it only fires ones every X(ms).

Q10.What is Event Delegation in JS?

Ans10 ->Instead of adding event listeners to many child elements.We can add the listener to their parent and let events bubble up from child to parent.

Q11.What is Currying in JS?

Ans 11->currying means breaking a function that takes multiple arguments into a chain of functions each taking one argument at a time.

Q12.Explain event bubbling vs capturing?

Ans12->

* Capturing Phase->Event goes from top to down (from parent to object)
* Target Phase->Event reaches the element you clicked.
* Bubbling Phase->Event goes from bottom to up.

Down first ->Hit target->Bubble uo

Q13. Diff between == and ===?

* == ->checks value only and also does type coercion(JS Convert one type of value to other for performing operations) and is also loose.
* ===-> It checks value and type both and donot do coercion and is also strict.

Q14. Why does [] == ![] is true?

Ans14 [] is truth is JS. And

* ![] =false so therefore [] == false
* Js == operation does type coercion
* [] is object false is Boolean Js converts Boolean to number for comparision. False->0
* []==0(now)
* [] -> “” ->0
* So in last 0 == 0 true

Q15.What is Closure and when would you use it?

Ans15->A Closure is basically a function that remembers the variables around it. Even after the function has finished running.

* Function carries its scope everywhere it goes.
* It is used for Private Variables ,memorization or callbacks