Question 1: What is error handling in JavaScript? Explain the try, catch, and finally blocks with an example.

Answer:
Error handling means **managing mistakes in code** so that the program does not crash.

JavaScript uses **try–catch–finally** to handle errors safely.

1. try block

- Code that *might* cause an error is written here.
- JavaScript "tries" to run it.

```
try {
  let num = 10 / 0;
}
```

2. catch block

- This block runs **only when an error happens**.
- It "catches" the error safely.

```
catch(error) {
  console.log("An error occurred!", error);
}
```

3. finally block

- This block runs **every time**, whether there's an error or not.
- Usually used for cleanup work.

```
finally {
  console.log("Execution finished");
}
```

Example:

```
try {
   let result = x + 5; // x is not defined → error
}
catch (err) {
   console.log("Something went wrong!");
}
finally {
   console.log("This will always run.");
}
```

catch handles the error
finally always runs

Question 2: Why is error handling important in JavaScript applications?

Answer:
Error handling is important because:

- It **prevents the program from crashing**
- It shows **user-friendly error messages**
- Makes websites more **stable**
- Helps developers find and fix bugs
- Keeps the app running even when something goes wrong