Q1. Explain the .map() function in JavaScript and provide three examples with detailed explanations.

- The .map()function is a method in JavaScript, commonly used to transform data in arrays.
- It is particularly useful when you need to perform the same operation on each element of an array and get a new array with the results, all while keeping the original array unchanged.

Examples:

- const numbers = [1, 2, 3, 4, 5];
  const squares = numbers.map(number => number * number);
  console.log(squares);
  Output: [1,4,9 16,25]

- const employees = [
  { name: 'Esha', age: 23 },
  { name: 'Vaishu', age: 28 },
  { name: 'Jin', age: 32 }
  ];
  const names = employees.map(employee => employee.name);
  console.log(names);
  Output: [‘Esha’, ‘Vaishu’, ‘Jin’]

- const numbers = [1, 2, 3, 4, 5];
  const incrementedNumbers = numbers.map(number => number + 1);

  console.log(incrementedNumbers);
  Output: [2, 3, 4, 5, 6]

2. Explain the .reduce() function in JavaScript and provide three examples with detailed explanations.

- The .reduce() function in JavaScript is an array method that applies a function against an accumulator and each element in the array (from left to right) to reduce it to a single value.
- **Accumulator**: The value that accumulates the result of the reduction across the array.
- **Callback Function**: The function executed on each element, which receives the accumulator, the current element, the current index, and the array itself.
- **Initial Value**: The optional initial value for the accumulator. If not provided, the first element of the array is used as the initial value, and the reduction starts from the second element.

Examples:

- const numbers = [1, 2, 3, 4, 5];

```
const sum = numbers.reduce((accumulator, currentValue) => accumulator +
currentValue, 0);

console.log(sum);
```

Output: 15

- const arrays = [[1, 2], [3, 4], [5, 6]];

```
const flattenedArray = arrays.reduce((accumulator, currentValue) =>
accumulator.concat(currentValue), []);

console.log(flattenedArray);
```

Output: [1,2,3,4,5,6]

Q3. Explain the .filter() function in JavaScript and provide three examples with detailed explanations.

- The .filter() function in JavaScript is an array method that creates a new array with all elements that pass a condition.
- If the callback function returns true, the element is included in the new array; if false, it is excluded.
- This method returns a new array

Examples:

- const numbers = [1, 2, 3, 4, 5, 6];

```
const evenNumbers = numbers.filter(number => number % 2 === 0);

console.log(evenNumbers);
```

Output: [2, 4, 6]

- const words = ['apple', 'banana', 'kiwi', 'grape', 'cherry'];

```
const longWords = words.filter(word => word.length >= 5);

console.log(longWords);
```

Output: ['apple', 'banana', 'cherry']

- const users = [

```
 { name: 'Alice', age: 22 },
 { name: 'Bob', age: 17 },
 { name: 'Charlie', age: 30 },
 { name: 'David', age: 15 },
```

```
  { name: 'Eve', age: 19 }
];

const adultUsers = users.filter(user => user.age >= 18);

console.log(adultUsers);
 Output: [
   { name: 'Alice', age: 22 },
   { name: 'Charlie', age: 30 },
   { name: 'Eve', age: 19 }
]
```