### 1. Reverse String with Delay

```
// Function to reverse a string after a delay of 2 seconds
function reverseStringAfterDelay(input) {
    setTimeout(() => {
        const reversed = input.split('').reverse().join('');
        console.log('Reversed String:', reversed);
    }, 2000); // 2 seconds delay
}

// Example usage
const input = "Hello, World!";
reverseStringAfterDelay(input);
```

### 2. Random Number Generator with Delay and Progress Indication

```
// Function to generate a random number after a delay with progress indication
function randomNumberWithDelay(delay) {
    let remainingTime = delay / 1000; // Convert milliseconds to seconds

    const interval = setInterval(() => {
        if (remainingTime > 0) {
            console.log(`Time remaining: ${remainingTime} seconds`);
            remainingTime--;
        } else {
            clearInterval(interval);
            const randomNumber = Math.floor(Math.random() * 100) + 1;
            console.log('Generated Random Number:', randomNumber);
        }
    }, 1000); // 1 second interval
}

// Example usage
const delay = 3000; // 3 seconds delay
randomNumberWithDelay(delay);
```

### 3. Store's Inventory Conversion to INR

```
// Function to convert USD prices to INR using map
function convertPricesToINR(inventory) {
    const exchangeRate = 80; // 1 USD to 80 INR
    const convertedInventory = Object.fromEntries(
        Object.entries(inventory).map(([item, priceUSD]) => [item, priceUSD * exchangeRate])
```

```
    );
    return convertedInventory;
}

// Example usage
const storeInventory = {
    'Apple': 1.5,
    'Banana': 1.2,
    'Orange': 2.0
};

const inventoryInINR = convertPricesToINR(storeInventory);
console.log('Inventory in INR:', inventoryInINR);
```

**### 4. Filtering and Capitalizing Books Published After 2010**

```
// Function to filter books published after 2010 and capitalize author names
function filterAndCapitalizeBooks(books) {
    return books
        .filter(book => book.year > 2010)
        .map(book => ({
            title: book.title,
            author: book.author.toUpperCase(),
            year: book.year
        }));
}

// Example usage
const books = [
    { title: 'Book A', author: 'Author A', year: 2012 },
    { title: 'Book B', author: 'Author B', year: 2008 },
    { title: 'Book C', author: 'Author C', year: 2015 }
];

const filteredBooks = filterAndCapitalizeBooks(books);
console.log('Filtered and Capitalized Books:', filteredBooks);
```

**### 5. URL Validation**

```
// Function to validate URLs using regex
function validateURL(url) {
    const urlRegex = /^(http:\/\/|https:\/\/)[a-zA-Z0-9._-]+\.[a-zA-Z]{2,}$/;
```

```
    if (urlRegex.test(url)) {
        console.log('Valid URL');
    } else {
        console.log('Invalid URL');
    }
}

// Example usage
const url1 = 'https://example.com';
const url2 = 'ftp://example.com';

validateURL(url1); // Valid URL
validateURL(url2); // Invalid URL
```

**### 6. LinkedIn Profile URL Validator**

```
// Function to validate LinkedIn profile URLs using regex
function validateLinkedInURL(url) {
    const linkedinRegex = /^https:\/\/www\.linkedin\.com\/in\/[a-zA-Z0-9_-]{5,30}[a-zA-Z0-9]$/;
    if (linkedinRegex.test(url)) {
        console.log('Valid LinkedIn Profile URL');
    } else {
        console.log('Invalid LinkedIn Profile URL');
    }
}

// Example usage
const profileURL1 = 'https://www.linkedin.com/in/john-doe_123';
const profileURL2 = 'https://www.linkedin.com/in/johndoe@123';

validateLinkedInURL(profileURL1); // Valid LinkedIn Profile URL
validateLinkedInURL(profileURL2); // Invalid LinkedIn Profile URL
```