### Assignment 1: Using Callback for Asynchronous Sum

javascript
CopyEdit

```
function sumAsync(a, b, callback) {
    setTimeout(() => {
        callback(a + b);
    }, 1000);
}

sumAsync(3, 4, (result) => {
    console.log("Sum:", result);
});
```

---

### Assignment 2: Returning a Promise

javascript
CopyEdit

```
function getData() {
    return new Promise((resolve) => {
        setTimeout(() => {
            resolve("Data fetched successfully.");
        }, 2000);
    });
}

getData().then((message) => console.log(message));
```

---

### Assignment 3: Fetch API to Retrieve Data

javascript
CopyEdit

```
async function fetchData(url) {
    try {
        const response = await fetch(url);
        const data = await response.json();
        return data;
    } catch (error) {
        console.error("Error fetching data:", error);
    }
}

fetchData("https://jsonplaceholder.typicode.com/todos/1").then(console.log);
```

---

### Assignment 4: Fetch API with URL

javascript
CopyEdit

```
async function fetchData(url) {
    try {
        const response = await fetch(url);
        const data = await response.json();
        return data;
    } catch (error) {
        console.error("Error fetching data:", error);
    }
}

fetchData("https://jsonplaceholder.typicode.com/todos/1").then(console.log);
```

---

### Assignment 5: Multiply Array Elements Using Callback

javascript
CopyEdit

```
function multiplyWithCallback(arr, callback) {
    const result = arr.map(num => num * 2);
    callback(result);
}

multiplyWithCallback([1, 2, 3, 4], (output) => {
    console.log("Doubled Array:", output);
});
```

---

### Assignment 6: Fetch User Details and Posts Using Promise Chaining

javascript
CopyEdit

```
function fetchUserDataAndPosts(userId) {
    const userUrl =
`https://jsonplaceholder.typicode.com/users/${userId}`;
    const postsUrl =
`https://jsonplaceholder.typicode.com/posts?userId=${userId}`;

    return fetch(userUrl)
        .then(response => response.json())
        .then(user => {
            return fetch(postsUrl)
                .then(response => response.json())
                .then(posts => ({ user, posts }));
        });
}

fetchUserDataAndPosts(1).then(console.log);
```

---

### Assignment 7: Fetch Multiple URLs Using Promise.all()

javascript
CopyEdit

```
function fetchMultipleData(urls) {
    const fetchPromises = urls.map(url => fetch(url).then(response =>
response.json()));
    return Promise.all(fetchPromises);
}

const urls = [
    "https://jsonplaceholder.typicode.com/todos/1",
    "https://jsonplaceholder.typicode.com/todos/2",
    "https://jsonplaceholder.typicode.com/todos/3"
];
```

```
fetchMultipleData(urls).then(console.log);
```

---

### Assignment 8: Fetch First Resolved Promise Using Promise.race()

javascript
CopyEdit

```
function racePromises(promises) {
    return Promise.race(promises);
}

const promise1 =
fetch("https://jsonplaceholder.typicode.com/todos/1").then(res =>
res.json());
const promise2 =
fetch("https://jsonplaceholder.typicode.com/todos/2").then(res =>
res.json());

racePromises([promise1,
promise2]).then(console.log).catch(console.error);
```