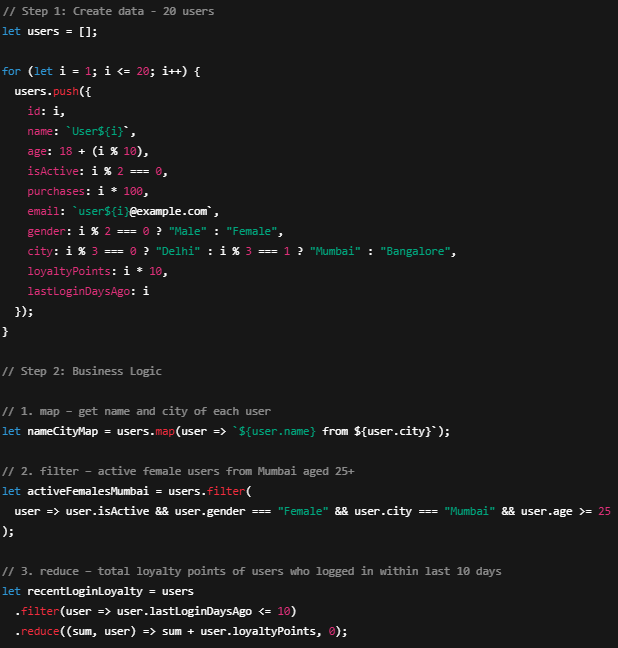
**Assignment – 3**

**Question 1: Build business logic, object of data=20 (use map, filter and reduce)**

**Code:**

****

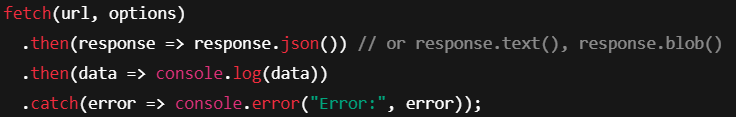
**Output:**

****

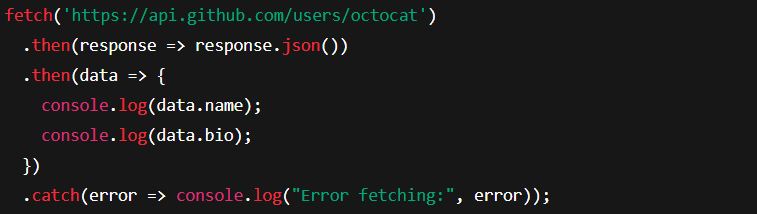
**Question 3: Fetch API**

The **Fetch API** is a modern way to **request data from servers** (APIs, files, etc.) using JavaScript.

**Basic Syntax :**

****

**Example : Get Data from API**

****

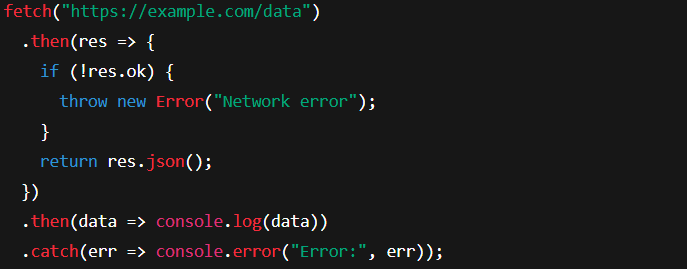
**Common Response Methods**

| **Method** | **Description** |
| --- | --- |
| res.text() | Returns response as plain text |
| res.json() | Converts JSON to JavaScript object |
| res.blob() | Used for files/images (binary data) |
| res.status | Status code like 200, 404 |
| res.ok | Boolean if response status is OK |

**Get vs Post in Fetch**

| **Feature** | **GET** | **POST** |
| --- | --- | --- |
| Use | To **fetch data** | To **send data** |
| Body | ❌ Not allowed | ✅ Allowed |
| Cacheable | ✅ Often cached | ❌ Usually not cached |
| Syntax | fetch(url) | fetch(url, { method: 'POST', ... }) |

**Handling Errors in Fetch**

****

**Fetch Options Object**

****

**Async/Await Version**

****

**Real-World Use Cases**

* Fetching product lists from a backend
* Submitting forms without refreshing the page
* Fetching weather/news data
* Working with REST APIs (like GitHub, OpenWeather, etc.)