

## Module 36.5-Have a World tour with API

আজকে একটু বাড়তি কষ্ট করবে সারা জীবন এইটার সুবিধা পাবে। তাই আজকে ফাটিয়ে দাও। জাভাস্ক্রিপ্ট এর ES6 এর কয়েকটা জিনিস বিশেষ করে let, const তারপর arrow function, এছাড়াও হালকা করে map, forEach একটু দেখে রাখো। এছাড়াও ধীরে ধীরে === এর জগতে প্রবেশ করতে থাকো। খুব বিশেষ দরকার না হলে == ইউজ করবে না। আর API জিনিসটা তো তোমার ফ্রেন্ড হয়েই থাকবে। মোটামুটি লাগবে। তাই সেটাও একটু ভালোমতো প্রাকটিস করো। দরকার হলে মডিউল এর ভিডিওগুলো যেখানে যেখানে ক্লিয়ার মনে হচ্ছে না সেগুলো ফটাফট আরেকবার দেখে ফেলো। এইটাই গুছিয়ে নেয়ার চান্স। কোথাও বুঝতে না পারলে গুগল মামাকে জিজ্ঞেস করো। গুগল মামা ঠিকমতো কিছু না বললে, ইউটিউব আন্টি এর কাছে যাও। সেও কিছু না বললে গুছিয়ে এই গরুপে প্রশ্ন করে ফেলো। সুযোগ থাকলে গুগল মিট এ জয়েন করো।

### মাইলস্টোন ৬ চেকলিস্ট

১. একটা অবজেক্ট ডিক্লেয়ার করো। সেই অবজেক্ট এর মিনিমাম পাঁচটা প্রপার্টি থাকবে। তারপর একটা প্রপার্টি এর মান হবে অন্য আরেকটা অবজেক্ট, আরেকটা প্রপার্টি এর মান হবে একটা array। একটা প্রপার্টি এর মান হবে একটা ফাংশন(মেথড)। এই মেথড কে কল করলে সে মেথড এর ভিতর থেকে অবজেক্ট এর যেকোন একটা প্রপার্টি রিটার্ন করবে। আর বাকি দুইটা তোমার ইচ্ছা মতো।

২. একটা টেমপ্লেট স্ট্রিং দিয়ে একটা স্ট্রিং ভেরিয়েবল ডিক্লেয়ার করো। সেটার মধ্যে উপরের অবজেক্ট এর একটা তিনটা প্রপার্টি এর মান তোমার টেমপ্লেট স্ট্রিং এর মধ্যে বসে। এর মধ্যে একটা প্রপার্টি সেট করবে--যেটা নেস্টেড অবজেক্ট আছে সেটার প্রপার্টি থেকে। আরেকটা প্রপার্টি হবে যে প্রপার্টি এর মান একটা array সেটার সেকেন্ড পজিশন এর উপাদান দিয়ে। আর বাকি একটা তোমার ইচ্ছা মতো বসাতে পারো।

৩.১. শূন্য প্যারামিটার ওয়ালা একটা অ্যারো ফাংশন লিখবে যেটা ৮৯ রিটার্ন করবে।

৩.২. এক প্যারামিটার ওয়ালা একটা অ্যারো ফাংশন ডিক্লেয়ার করবে। এই ফাংশনের কাজ হবে যে প্যারামিটার নিবে সেটাকে ৭ দিয়ে ভাগ করে ভাগফল রিটার্ন করবে।

৩.৩ দুই, প্যারামিটার ওয়ালা একটা অ্যারো ফাংশন লিখবে। এই ফাংশনের কাজ হবে যে দুইটা প্যারামিটার ইনপুট নিবে। সেই দুইটা প্যারামিটারকে যোগ করে যোগফলকে দুই দিয়ে ভাগ করে ভাগফল রিটার্ন করে দাও।

৩.৪ একাধিক লাইন ওয়ালা অ্যারো ফাংশন লিখো। সেটাতে দুইটা প্যারামিটার নিবে। ওই arrow ফাংশনটা হবে অনেকগুলো লাইনের। সেখানে প্রত্যেকটা ইনপুট প্যারামিটার এর সাথে ৭ যোগ করবে তারপর যোগফল দুইটাকে আবার যোগ করবে। ক্যামনে করবে সেটা চিন্তা করে বের করার চেষ্টা করো।

৪. অনেকগুলো সংখ্যার একটা array হবে। তারপর তোমার কাজ হবে array এর উপরে map ইউজ করে। প্রত্যেকটা উপাদানকে ৭ দিয়ে ভাগ করে ভাগফল আরেকটা array হিসেবে রাখবে। পুরা কাজটা এক লাইনে হবে।

৫. জাভাস্ক্রিপ্ট এ array এর map, forEach, filter, find কোনটা দিয়ে কি হয়। সেটার একটা সামারি লিখে ফেলো।

৬. সিম্পল একটা জাভাস্ক্রিপ্ট অবজেক্ট এর কোন একটা প্রোপার্টিকে ভেরিয়েবল হিসেবে ডিক্লেয়ার করার জন্য destructuring ইউজ করো। array এর destructuring করবে আর সেটা করার জন্য তুমি এরে এর সেকেন্ড পজিশন এর উপাদান কে destructuring করে 'balance' নামক একটা ভেরিয়েবল এ রাখবে।

৭. একদম প্রাথমিক স্টেপ হিসেবে jsonplaceholder এর ওয়েবসাইট থেকে ডাটা fetch করে সেটাকে কনসোল এ দেখাতে হবে। ধরো তুমি তাদের ওয়েবসাইট এ photos এর API এর লিংক কোনটা সেটা জাভাস্ক্রিপ্ট দিয়ে কোড করে সেই ডাটা কনসোল এ দেখতে পারতেছো কিনা। তারপর কয়েকটা কার্ড বানাবে (হতে পারে বুটস্ট্রাপ এর কার্ড) সেগুলো আবার তুমি html দিয়ে ওয়েবসাইট এ ছবি এবং ছবির নিচে ছবির টাইটেল দেখাবে।

৮. প্রত্যেকটা ফটোতে ক্লিক করলে উপরে আরেকটা div থাকবে। ক্লিক করার পর ডাইনামিকভাবে ওই ফটো এর ডাটা লোড করবে। সেটা করার জন্য আগের ডাটা লোড করার API এর লিংক এর পরে photos এর পরে ডাইনামিকভাবে \${id} বসিয়ে দিবে। এরপর ডিটেইল অংশে photo এর url দিয়ে মেইন ইমেজ এবং thumbnail ইমেজ দুইটাই দেখাবে। সাথে সাথে title প্রপার্টিও দেখাবে।

৯. তুমি যে ডাটা লোড করেছো। বা ডাটা ওয়েবসাইট এ দেখাচ্ছ। সেই কোড এর মধ্যে arrow ফাংশন ইউজ করতে পারতেছো কিনা। যখন লুপ চালাচ্ছ সেখানে forEach ইউজ করতে পারতেছো কিনা।

১০. সিরিয়াস প্রাকটিস

হয় মডিউল ৩৩ ভালো করে দেখে ফেলো। বিশেষ করে the meal db রিলেটেড ৩৩-৫ থেকে ৩৩-৮ পর্যন্ত। তারপরে আরো সময় থাকলে এর আরেকটা খালতো ভাই the cocktaildb থেকে কিছু জিনিস এনে দেখাবে। একজাঙ্কলি কি দেখাতে হবে। সেটা আমি বলে দিবো না। তুমি ওদের ওয়েবসাইট এ যাও। সেখানে কি কি লেখা আছে সেগুলো পড়ো। api গুলো এর ছোট করে কি কি করে বলা আছে। সেগুলো দেখো। তারপর কিছু ডাটা লোড করো। সেই ডাটাগুলো দেখাও। এইখানে সার্চ ফাংশনালিটি ইমপ্লিমেন্ট করো। অনেকটা mealdb এর মতো। আবার কোন একটাতে ক্লিক করলে সেটার ডিটেইল দেখাবে।

এসাইনমেন্ট ৬ প্রিপারেশন:

২৮ ফেব্রুয়ারি এসাইনমেন্ট ৬ আসবে। এ এসাইনমেন্ট এ ভালো করার জন্য API রিলেটেড জিনিসগুলো একটু বুঝে বুঝে করলেই বাকি কাজ হয়ে যাবে। অর্থাৎ কিভাবে API থেকে ডাটা নিবে। কিভাবে সেই ডাটা ওয়েবসাইট এ দেখাবে। কিভাবে একের বেশি API কল করবে। এমন জিনিসপাতি।

### 36\_5-1 API ES6 and concept milestone practice

```
/*
1. Object with property (string, boolean, number, object, array,
method which will return a property value)
2. template string. minimum 3 property will be dynamic (nested object
property, array second element, another property)
3.1: no parameter arrow function. return 89
3.2: single parameter arrow function. will divide input by 7
3.3: arrow function with two parameters.
3.4 multi-line arrow function. two parameter.
.
4. map divide each element by 7
5. map, forEach, filter, find
6. destructuring on object and array
.
*/
```

### 36\_5-2 Common API Data access related issues

```
const loadSingleUser = () => {
  fetch("https://randomuser.me/api/")
    .then((res) => res.json())
    .then((data) => displaySingleUser(data.results[0]));
};

loadSingleUser();

const displaySingleUser = (user) => {
  // console.log(user.name);
};

const loadMeals = (searchText) => {
```

```

const url =
`https://www.themealdb.com/api/json/v1/1/search.php?s=${searchText}`;

fetch(url)
  .then((res) => res.json())
  .then((data) => displayMeals(data.meals));
};

const displayMeals = (meals) => {
  const container = document.getElementById("meals");

  meals.forEach((element) => {
    const div = document.createElement("div");
    div.innerHTML = `
      <h1 style="color:red">${element.strMeal}</h1>
      <button
onclick="loadMealDetail('${element.strMeal}')">${element.strMeal}</but
ton>
    `;
    container.appendChild(div);
  });
};

loadMeals("fish");

const loadMealDetail=(mealName)=>{
  console.log(mealName);
}

```

### 36\_5-3 Add spinner and toggle content while loading data

```

const loadSingleUser = () => {
  fetch("https://randomuser.me/api/")
    .then((res) => res.json())
    .then((data) => displaySingleUser(data.results[0]));
};

loadSingleUser();

const displaySingleUser = (user) => {
  // console.log(user.name);
};

```

```

// meal db
const toggleSpinner=displayStyle=>{
  document.getElementById('spinner').style.display=displayStyle;
}

const toggleSearchResult=displayStyle=>{
  document.getElementById('meals').style.display=displayStyle;
}

function searchMeal() {
  const searchText = document.getElementById("search-field").value;

  // display spinner
  toggleSpinner('block');
  toggleSearchResult('none');

  if (searchText == "") {
    alert("empty");
    toggleSpinner("none");
  } else {
    loadMeals(searchText);

    document.getElementById("search-field").value = "";
  }
}

const loadMeals = (searchText) => {
  const url =
`https://www.themealdb.com/api/json/v1/1/search.php?s=${searchText}`;

  fetch(url)
    .then((res) => res.json())
    .then((data) => displayMeals(data.meals));
};

const displayMeals = (meals) => {
  const container = document.getElementById("meals");
  container.textContent = "";

  meals?.forEach((element) => {
    const div = document.createElement("div");
    div.innerHTML = `
      <h1 style="color:red">${element.strMeal}</h1>
      <P>${element.strIngredient18?element.strIngredient18:""}</p>
    `
  })
}

```

```
        <button
onclick="loadMealDetail('${element.strMeal}')">${element.strMeal}</button>
    `;
    container.appendChild(div);
  });
  toggleSpinner("none");
  toggleSearchResult('block');
};

// loadMeals("");

const loadMealDetail = (mealName) => {
  console.log(mealName);
};
```