

Module 45: React Core Concepts, JSX, props, state

45-1: Module Introduction, Create React App, 6 core concepts

ইন্সটল: নিচের তিন টি কমান্ড এর মাধ্যমে একটি React APP তৈরি করা যাই:

```
npx create-react-app my-app
cd my-app
npm start
```

এখানে প্রথম লাইনের my-app টি হল App এর নাম, এখানে অন্য যেকোনো নাম ও দেওয়া যেতে পারে। দ্বিতীয় লাইন দ্বারা directory পরিবর্তন করে app এর ঐ ফোল্ডার এ জাওয়া হয়েছে। এবং শেষ লাইন দ্বারা live server কে open করার মাধ্যমে Browser এ ঐ App টি Run করানো হয়েছে।

React হল JavaScript এর একটি Library (Framework না)

যেকোনো একটা Modern JavaScript ফ্রেমওয়ার্ক শিখলেই হবে তবে নতুনদের জন্য React দিয়ে শুরু করলে ভালো হবে, VueJS ও শেখা যেতে পারে (এটি কাস্টমাইজ করা যাই কিন্তু React এর তুলনায় কম) তবে নতুন হিসেবে শুরুতে AngularJS না করায় ভালো, এটিতে কাস্টমাইজ করা যায়না বলেই চলে, তাদের ডেভেলপাররা যেগুলো তৈরি করে রেখেছেন সেগুলো নিয়েই কাজ করতে হয়। এজন্য নতুনদের জন্য React সর্বোত্তম পছন্দ হতে পারে তাছাড়া এটাতে চাইলে নিজের ইচ্ছামত কাস্টমাইজ করা যায়।

www.reactjs.org সাইটে গিয়ে Documentation টি পড়তে হবে, না বুঝলেও সমস্যা নাই তবে পড়তে হবে। উপরের কমান্ড গুলো দিয়ে React APP ইন্সটল হয়ে গেলে আমরা সেটিকে vs code দিয়ে open করে src ফোল্ডারের মধ্যে App.js নামক ফাইলের মধ্যে আমাদের প্রয়োজনীয় কোড গুলো লিখব।

45-2 JSX, Dynamic content, Dynamic Style in React

JSX: JavaScript XML কে সংক্ষেপে JSX বলে। JSX এর উদ্দেশ্য- JavaScript File এর মধ্যে HTML লেখা। কিন্তু JS ফাইল এর মধ্যে আমরা যে HTML Tag গুলো লিখি সেগুলো দেখতে সাধারণ HTML হলেও JavaScript কিন্তু সেগুলো কে চিনে না, এই ট্যাগের মত দেখতে জিনিষ গুলোকে পরবর্তীতে JavaScript Object এর মত করে রূপান্তর করে ফেলে React এর একটা Element তৈরি করে, যেটা HTML Tag এর সাথে অনেকটাই মিল আছে এজন্য আমরা বলতে পারি JS File এর মধ্যে HTML লিখছি, কিন্তু আসল কাজ যেটি করছি সেটা হল- সহজ ভাবে JavaScript এর Object গুলোকে লিখছি।

Shortcut: HTML File এ কোড লেখার সময় আমরা Shortcut ব্যবহার করে দ্রুত কোড লিখে ফেলতে পারতাম কিন্তু JS এর মধ্যে যখন আমরা কোড লিখতে যাই তখন আগের মত সেই Shortcut গুলো ব্যবহার করা যাই না, এবার চেষ্টা করব এটির একটা সমাধান বাহির করার:

VS Code এর Setting এ গিয়ে Extension সেকশন গিয়ে দেখতে পাব Emmet (যদি না থাকে তবে আগে ইন্সটল করে নিতে হবে, কিভাবে VS Code এ কোন Extension ইন্সটল করতে হয় সেটি আমরা আগে থেকেই জানি) এবার এই Emmet এ ক্লিক করলে ডানে Include Language নামে একটি Option পাব যেখানে Add Item বাটন এ ক্লিক করে Item হিসেবে: javascript এবং Value হিসেবে: javascriptreact দিয়ে OK বাটনে ক্লিক করলেই JavaScript File এর মধ্যে Emmet টি যুক্ত হয়ে যাবে, তখন এর Shortcut গুলোও সহজে ব্যবহার করতে পারব।

JS File এ HTML Code: এখন আমরা (.my-class) Shortcut ব্যবহার করে একটি class তৈরি করলে দেখতে পাব Attribute টি class="my-class" এভাবে না এসে className="my-class" লেখা আসে কারন, আগেই বলেছি- এটি আমাদের পুরপুরি HTML লেখা না এটা JavaScript যেভাবে বুঝে সেভাবে লেখা, এটি সম্পর্কে আমরা ইতিমধ্যে JSX এর বর্ণনায় কিছুটা জেনেছি।

এখন প্রশ্ন আসতে পারে, HTML এর সকল Tag গুলো কে ঠিক মত লিখতে পারলে এই class="" Attribute টি কেন আগের মত করে লিখতে পারব না? আমরা জানি, JavaScript হল একটা OOP (Object Oriented Programming) Language আর OOP এর মূল কনসেপ্ট এর মধ্যে অন্যতম একটি কনসেপ্ট হল class। আমরা যেমন কোন ফাংশন লিখতে গেলে function Keyword টি ব্যবহার করি তেমনি OOP ব্যবহার করে কোন প্রোগ্রাম লিখতে হলে class Keyword টি লিখে তার একটি নাম দিয়ে {সেকেন্ড ব্রাকেট} এর মধ্যে সকল function, variable etc. লিখতে হয়। যেহেতু class নামে একটি Keyword JavaScript এ আছে তাই আমরা Attribute এর নাম হিসেবে যখন class লিখি JavaScript সেটিকে তার Keyword OOP এর class ভেবে কাজ করতে থাকে কিন্তু OOP এর class লেখার পর class এর একটি নাম সেকেন্ড ব্রাকেট { } ইত্যাদি অংশ থাকে, যেগুলো কে Attribute হিসেবে লেখার ফলে, এর class এর পরে নাম { } না পেয়ে Error দেখায়। আর এই সমস্যার থেকে মুক্তি পাওয়ার জন্য Attribute হিসেবে আমরা যে class কথাটি লিখি সেটিকে পরিবর্তন করে অন্য যেকোনো একটি নামে: className হিসেবে লেখা হয়। কিন্তু এভাবে লিখলেও Final Output হিসেবে যখন Browser এ কোড গুলো দেখানো হয় তখন React যেখানে যেখানে className কথাটি আছে সেখানে সেখানে HTML এর উপযোগী class কথাটি লিখে দেয়।

একই ভাবে আমরা এমন অনেক কিছুই যেগুলো JavaScript এর Keyword সেগুলো কে সরাসরি HTML আকারে এখানে লিখতে পারব না, ধীরেধীরে আমরা সেগুলো সম্পর্কে আরও জানবো।

Custom CSS লেখা: এখন আমরা উপরের .my-class টি ব্যবহার করে তার জন্য CSS লিখতে পারি App.css নামক ফাইলের মধ্যে। যেহেতু এটা .css ফাইল তাই সাধারণ CSS যেভাবে লিখি সেভাবে লিখতে হবে কিন্তু যদি আমরা Inline বা Internal CSS লিখতে চাই তখন আমাদের বাধ্য হয়ে আমাদের যে, App.js ফাইল আছে সেখানেই লিখতে হবে আর সেখানে লিখতে হলে আমাদের কিছু নিয়ম জানতে হবে। কেননা একটা JS ফাইলের মধ্যে আমরা CSS লিখতে চাচ্ছি সেটা যেন আগের মত (HTML এর class আর JS OOP এর class) না হয়ে যাই তার জন্য কিছু নিয়ম মেনে CSS লিখতে হবে।

JS File এর মধ্যে CSS লেখা: App.js এর মধ্যে আমরা Internal CSS, Inline CSS লিখতে হবে Object আকারে।

Internal CSS লেখার নমুনা:

```
const myStyle={
  color : "red",
  border : "1px solid black",
  backgroundColor : "green"
}
```

তবে লেখার সময় খেয়াল রাখতে হবে CSS প্রপার্টি লেখার সময় ভেতরে কোন - থাকতে পারবে না সেক্ষেত্রে Camel Case আকারে লিখতে হবে যেমন: background-color না লিখে backgroundColor এভাবে লিখতে হবে। এবং এভাবে Dynamic ভাবে style Attribute এর ভেলু টি লিখতে হবে-

```
<p style={myStyle}> My Inline CSS Style </p>
```

Dynamic ভাবে কোন কিছু লেখার নিয়ম নিয়ে পরবর্তীতে বিস্তারিত আলোচনা করা হবে।

Inline CSS লেখার নমুনাঃ এটিও পূর্বের ন্যায় লিখতে হয়, একই সুধু পার্থক্য টা হল আগের টা তে আমরা Object একটা Variable এ লিখে সেটিকে Dynamic ভাবে Attribute এর value হিসেবে যুক্ত করেছি আর এখানে সরাসরি { } সেকেন্ড ব্রাকেট এর মধ্যে Object টি লিখব। উদাহরণঃ

```
<p style={{
  color : "red",
  border : "1px solid black",
  backgroundColor : "green"
}}> My Inline CSS Style </p>
```

Dynamic ভাবে Data লেখা বা দেখানোঃ আমরা চাইলে আমাদের এই JSX এর মধ্যে Dynamic ভাবে কোন variable বা অন্য কিছুর মান দেখাতে পারি সেক্ষেত্রে সেকেন্ড ব্রাকেট এর মধ্যে Variable টির নাম দিয়ে দিলেই হয়ে যাবে। যেমনঃ

```
let age = 24;
```

এই Variable যেটিকে নিচের p Tag এর মধ্যে Dynamic ভাবে দেখাতে চাই-

```
<p> My Age is {age} </p>
```

Output: My Age is 24

এভাবে সুধু Variable কে ছাড়া ও যতভাবে Dynamic Data দেখানো যায় সবগুলোই ব্যবহার করতে পারব। সুধু তাই না এই { সেকেন্ড ব্রাকেট } এর মধ্যে আমরা JavaScript এর যেকোনো কোড ই লিখতে পারি।