# Explanation of React

## What is React?

1. React is a JavaScript library used for building user interfaces, especially single-page applications.
2. It allows developers to create reusable UI components, which makes development faster and more efficient.
3. React uses a virtual DOM to update only the parts of the web page that change, improving performance.
4. It follows a component-based architecture, making the code more organized and easier to maintain.
5. React is widely used in modern web development by companies like Facebook, Instagram, and Netflix.

### Real Time Example

Suppose you are using an E-commerce Website (like Amazon or Flipkart).  
- When you add a product to the cart, the cart count at the top updates instantly.  
- But the entire page does not reload.  
- This smooth and fast update happens because the website is built using React.

## React என்றால் என்ன?

1. React என்பது JavaScript library ஆகும், இது இணையதளத்தின் User Interface (UI) உருவாக்க பயன்படுத்தப்படுகிறது.
2. React மூலம் மீண்டும் பயன்படுத்தக்கூடிய UI components உருவாக்க முடியும், இதனால் வேகமாகவும் எளிதாகவும் development செய்யலாம்.
3. இது Virtual DOM பயன்படுத்தி, மாற்றம் ஏற்பட்ட பகுதியை மட்டும் update செய்கிறது, இதனால் வேகமான செயல்திறன் கிடைக்கும்.
4. Component-based architecture பின்பற்றுவதால், code சீராகவும் பராமரிக்க எளிதாகவும் இருக்கும்.
5. Facebook, Instagram, Netflix போன்ற பெரிய நிறுவனங்கள் React-ஐ அதிகமாக பயன்படுத்துகின்றன.

### உண்மையான உதாரணம்

நீங்கள் ஒரு மின்னணு வணிக வலைத்தளத்தை (அமேசான் அல்லது பிளிப்கார்ட் போன்றவை) பயன்படுத்துகிறீர்கள் என்று வைத்துக்கொள்வோம்.  
- நீங்க ஒரு product-ஐ cart-க்கு add பண்ணும்போது, மேல இருக்குற cart count உடனடியாக update ஆகும்.  
- ஆனால் முழுப் பக்கமும் மீண்டும் reload ஆகாது.  
- வலைத்தளம் ரியாக்ட் ஐப் பயன்படுத்தி உருவாக்கப்பட்டுள்ளதால் இந்த மென்மையான மற்றும் வேகமான புதுப்பிப்பு நிகழ்கிறது.

## Why React is Important?

1. Faster Performance – React uses Virtual DOM to update only the required parts of a page.
2. Reusable Components – Same UI components can be reused, saving time and effort.
3. Easy to Maintain – Component-based architecture makes applications more organized.
4. Large Community Support – Millions of developers contribute, ensuring continuous growth.
5. High Industry Demand – React developers are in high demand in web development jobs.

## React ஏன் முக்கியமானது?

1. வேகமான செயல்திறன் – React Virtual DOM-ஐ பயன்படுத்தி தேவையான பகுதியை மட்டும் update செய்கிறது.
2. மீண்டும் பயன்படுத்தக்கூடிய Components – ஒரே UI components-ஐ பலமுறை பயன்படுத்தலாம்.
3. பராமரிக்க எளிது – Component-based architecture காரணமாக application சீராக இருக்கும்.
4. பெரிய Community Support – லட்சக்கணக்கான developers React-க்கு support தருகிறார்கள்.
5. வேலை வாய்ப்பு அதிகம் – React developer-க்களுக்கு IT துறையில் அதிக வேலை வாய்ப்பு உள்ளது.

## Features of React

1. JSX (JavaScript XML) – Allows writing HTML-like code inside JavaScript.
2. Virtual DOM – Improves performance by updating only changed elements.
3. Component-Based – Applications are divided into reusable small components.
4. One-Way Data Binding – Data flows in one direction, making debugging easier.
5. Rich Ecosystem – Compatible with many tools, libraries, and extensions.

## React-ன் சிறப்பம்சங்கள்

1. JSX (JavaScript XML) – JavaScript-க்குள் HTML போல code எழுத அனுமதிக்கிறது.
2. Virtual DOM – மாற்றம் ஏற்பட்ட பகுதிகளை மட்டும் update செய்வதால் வேகம் அதிகரிக்கும்.
3. Component-Based – Application சிறிய reusable components-ஆக பிரிக்கப்பட்டிருக்கும்.
4. One-Way Data Binding – Data ஒரே திசையில் செல்லும், debugging எளிதாகும்.
5. பெரிய Ecosystem – பல tools, libraries, extensions-க்கு React இணக்கமானது.

## Library vs Framework

|  |  |  |
| --- | --- | --- |
| Aspect | Library (React) | Framework (Angular) |
| Definition | Collection of reusable code that developers call when needed. | Complete structure that controls the flow of the application. |
| Control | Developer has control – you decide when and how to use it. | Framework controls the flow – you follow its rules. |
| Flexibility | More flexible, can integrate with other libraries easily. | Less flexible, follows strict patterns. |
| Learning Curve | Easier to learn and start quickly. | Steeper learning curve due to many built-in features. |
| Example | React, Lodash | Angular, Django, Spring |

## Library vs Framework (நூலகம் vs கட்டமைப்பு)

|  |  |  |
| --- | --- | --- |
| விளக்கம் | Library (React) | Framework (Angular) |
| அர்த்தம் | தேவையானபோது பயன்படுத்தக்கூடிய reusable code களின் தொகுப்பு. | Application-ன் முழு structure-ஐ நிர்வகிக்கும் system. |
| கட்டுப்பாடு | Developer-க்கு கட்டுப்பாடு இருக்கும் – எப்போது, எப்படி பயன்படுத்தணும் நீங்கள்தான் முடிவு. | Framework-க்கு கட்டுப்பாடு இருக்கும் – அது சொன்ன விதிகளை நீங்க பின்பற்றணும். |
| நெகிழ்வு | அதிக நெகிழ்வு, மற்ற Libraries-ஐ எளிதில் சேர்க்கலாம். | குறைந்த நெகிழ்வு, strict patterns பின்பற்றணும். |
| கற்றல் சிரமம் | எளிதில் கற்றுக்கொள்ளலாம், விரைவாக தொடங்கலாம். | அதிக சிரமம், ஏனெனில் பல built-in features இருக்கும். |
| உதாரணம் | React, Lodash | Angular, Django, Spring |