

# React Native

## ১. React Native কী?

React Native হলো একটি ফ্রেমওয়ার্ক যা দিয়ে আপনি একবার কোড লিখে Android এবং iOS উভয় মোবাইল অ্যাপে ব্যবহার করতে পারেন। এটি JavaScript এবং React ব্যবহার করে তৈরি করা হয়।

সরাসরি বলতে গেলে:

- React Native হলো **Cross-Platform Mobile App Development Tool**।
- এর মাধ্যমে আপনি একই কোডবেস দিয়ে Android ও iOS অ্যাপ বানাতে পারেন।
- এটি **নেটিভ কম্পোনেন্ট** ব্যবহার করে, তাই পারফরম্যান্স ভালো।

React Native মূলত তৈরি করেছে **Facebook** (এখন Meta) ২০১৫ সালে।

---

## ২. React Native এর মূল বৈশিষ্ট্য (Features)

### 1. Cross-Platform Development

একবার কোড লিখলেই, iOS ও Android দুটোতেই চলে।

### 2. Hot Reloading / Fast Refresh

কোড পরিবর্তন করলে অ্যাপটি তাত্ক্ষণিক আপডেট হয়।

### 3. Native Components

React Native সরাসরি **নেটিভ UI কম্পোনেন্ট** ব্যবহার করে, ফলে পারফরম্যান্স ভালো হয়।

### 4. JavaScript + React

যদের React বা JavaScript জানা আছে, তারা সহজেই React Native শিখতে পারে।

### 5. Community Support

React Native এর বড় একটি কমিউনিটি আছে, তাই লাইব্রেরি ও সমাধান সহজে পাওয়া যায়।

---

### ৩. React Native কেন ব্যবহার করা হয়? (Why React Native)

#### ১. দ্রুত ডেভেলপমেন্ট

Native Android/iOS থেকে দ্রুত অ্যাপ বানানো যায়।

#### ২. কস্ট কমানো

দুই প্ল্যাটফর্মের জন্য আলাদা কোড লিখতে হয় না, তাই ডেভেলপমেন্ট খরচ কমে।

#### ৩. নেটিভ পারফরম্যান্স

React Native সরাসরি নেটিভ কম্পোনেন্ট ব্যবহার করে, তাই পারফরম্যান্স অনেকটা Native অ্যাপের মত।

#### ৪. UI উন্নতমানের

iOS এবং Android এর look & feel বজায় থাকে।

#### ৫. উন্নত সমর্থন ও লাইব্রেরি

Facebook ও কমিউনিটির সমর্থন থাকায় সমস্যা সমাধান সহজ।

---

### ৪. React Native এর প্রকার (Types)

React Native সাধারণত প্রকারভেদে ভাগ করা যায় না, কারণ এটি নিজেই একটি ফ্রেমওয়ার্ক। তবে, উপায় বা স্টাইল অনুযায়ী ব্যবহার ভাগ করা যায়:

#### ১. Pure React Native

- শুধুমাত্র React Native এর নিজস্ব কম্পোনেন্ট ও লাইব্রেরি ব্যবহার করে অ্যাপ তৈরি করা।
- উদাহরণ: View, Text, ScrollView, Button ইত্যাদি।

#### ২. React Native with Expo

- Expo হলো React Native এর উপরে তৈরি একটি প্ল্যাটফর্ম।
- এতে কোড রান করা সহজ, লাইব্রেরি ইন্সটল সহজ, আর Android/iOS এ তাড়াতাড়ি দেখা যায়।
- ছোট প্রজেক্ট বা প্রোটোটাইপের জন্য ভালো।

### ৩. React Native with Native Modules

- যদি কোনো নির্দিষ্ট নেটিভ ফিচার দরকার হয় যা React Native সরাসরি সমর্থন করে না, তখন **নেটিভ কোড (Java/Kotlin বা Swift/Objective-C)** লিখে ব্যবহার করা হয়।
- 

### ৪. React Native এ কি কি ব্যবহার করা হয়?

- Language:** JavaScript (বা TypeScript)
  - Framework:** React
  - UI Components:** Native Components (View, Text, Image)
  - Navigation:** React Navigation বা React Native Navigation
  - State Management:** Redux, Context API
  - API Calls:** Fetch, Axios
  - Styling:** CSS-এর মতো স্টাইলিং, কিন্তু JavaScript object আকারে
- 

### ৫. React Native vs Native Development

Feature	Native (Android/iOS)	React Native
<b>Language</b>	Java/Kotlin, Swift/Objective-C	JavaScript + React
<b>Code Reuse</b>	কম	অনেক
<b>Development Speed</b>	ধীর	দ্রুত
<b>UI Performance</b>	খুব ভালো	প্রায় Native এর মতো
<b>Community Support</b>	সীমিত	বিশাল
<b>Cost</b>	বেশি	কম

---

## ৭. React Native কেন জনপ্রিয়?

- একবার কোড লিখলেই Android ও iOS কাজ করে।
- Facebook, Instagram, Tesla এর মতো বড় কোম্পানিরা ব্যবহার করছে।
- নতুনদের জন্য শেখা সহজ, কারণ React ও JavaScript ব্যবহার করা হয়।
- বড় লাইব্রেরি ও কমিউনিটি।

### 💡 সারসংক্ষেপে:

React Native হলো **একটি Cross-Platform Mobile Development ফ্রেমওয়ার্ক**, যা **JavaScript + React** ব্যবহার করে দুই প্ল্যাটফর্মে Native অ্যাপ বানাতে দেয়। এটি দ্রুত ডেভেলপমেন্ট, কম খরচ, এবং ভালো পারফরম্যান্স দেয়।

# React VS Next.js VS React Native

## ১. React কী?

React হলো একটি JavaScript লাইব্রেরি যা দিয়ে আমরা UI (User Interface) তৈরি করি।

### মূল বৈশিষ্ট্য:

#### 1. Component-Based

UI ছোট ছোট কম্পোনেন্টে ভাগ করা যায়।

উদাহরণ: Button, Header, Footer ইত্যাদি।

#### 2. Declarative

UI কেমন হবে তা ডিক্লেয়ার করতে হয়, React নিজেই DOM আপডেট করে।

### 3. Virtual DOM

পারফরম্যান্স ভালো রাখতে React একটি ভার্চুয়াল DOM ব্যবহার করে।

ব্যবহার:

- ওয়েব অ্যাপ ডেভেলপমেন্টে
- ইন্টারেক্টিভ UI তৈরি করতে

উদাহরণ:

```
function App() {
  return <h1>হ্যালো, React!</h1>;
}
```

## ২. Next.js কী?

Next.js হলো React এর উপরে তৈরি একটি Full-Stack Framework।

মূল বৈশিষ্ট্য:

### 1. Server-Side Rendering (SSR)

পেজ সার্ভার থেকে রেন্ডার হয় → SEO এবং লোড টাইম ভালো হয়।

### 2. Static Site Generation (SSG)

ডেটা নিয়ে Static HTML তৈরি করা যায়।

### 3. API Routes

ব্যাকএন্ড ফাংশনও একই প্রজেক্টে লেখা যায়।

### 4. File-based Routing

পেজ তৈরি করতে শুধু pages ফোল্ডারে ফাইল বানাতে হয়।

ব্যবহার:

- SEO-friendly ওয়েবসাইট

- ব্লগ, ই-কমার্স, ড্যাশবোর্ড অ্যাপ
- Full-Stack React অ্যাপ

**উদাহরণ:**

```
export default function Home() {
  return <h1>হ্যালো, Next.js!</h1>;
}
```

### ৩. React Native কী?

React Native হলো **Mobile App Development Framework** যা React ব্যবহার করে iOS এবং Android অ্যাপ বানায়।

**মূল বৈশিষ্ট্য:**

#### 1. Cross-Platform

একবার কোড লিখলেই iOS ও Android দুটোতেই চলে।

#### 2. Native Components

পারফরম্যান্স ভালো থাকে, Native look & feel বজায় থাকে।

#### 3. Hot Reloading

কোড পরিবর্তন করলে তাত্ক্ষণিক রিফ্রেশ হয়।

**ব্যবহার:**

- মোবাইল অ্যাপ ডেভেলপমেন্ট
- যাদের React জানা আছে, তারা সহজেই শিখতে পারে

**উদাহরণ:**

```
import { Text, View } from 'react-native';

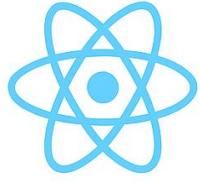
export default function App() {
  return (
    <View>
      <Text>হ্যালো, React Native!</Text>
    </View>
  );
}
```

## সংক্ষেপে পার্থক্য:

বিষয়	React	Next.js	React Native
প্রকার	JavaScript লাইব্রেরি	React এর ফ্রেমওয়ার্ক	Mobile App ফ্রেমওয়ার্ক
ব্যবহার	ওয়েব UI	ওয়েবসাইট ও ওয়েব অ্যাপ	মোবাইল অ্যাপ
Rendering	Client-side	SSR + SSG + CSR	Native UI
SEO Support	সীমিত	ভালো	না
কোড ব্যবহার	Web only	Web only	iOS & Android

### 💡 মন্তব্য:

- React হলো UI লাইব্রেরি,
- Next.js হলো React + Server-side rendering ফ্রেমওয়ার্ক,
- React Native হলো React ব্যবহার করে মোবাইল অ্যাপ বানানোর  
ফ্রেমওয়ার্ক।



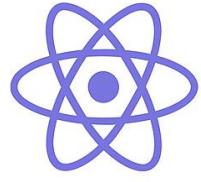
## React

JavaScript library  
for building  
user interfaces



## Next.js

React framework  
with server-side  
rendering



## React Native

Mobile app  
framework for  
iOS and Android