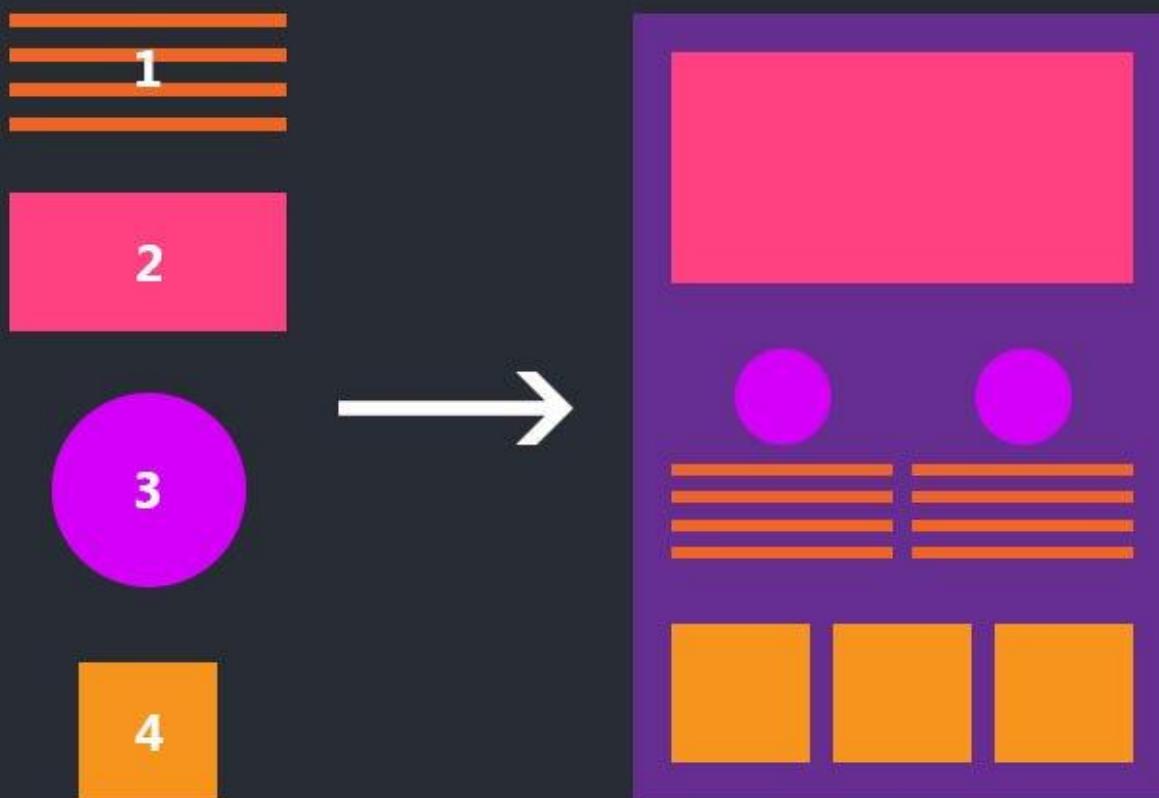


Components

- A component is a reusable piece of UI.
- Building blocks of every React app.
- Can be combined to form complex interfaces.

React Components



CATEGORIES

- Tshirts (3301)
- Tops (2647)
- Kurtas (866)
- Earrings (760)
- Flip Flops (384)
- Dresses (303)
- Bracelet (282)
- Sandals (249)

[+ 65 more](#)

BRAND

- SZN (1183)
- glitchez (740)
- DressBerry (390)
- TRENDSSATTIRE (329)
- Kasendra (126)
- LIMEROAD (101)
- BASE 41 (65)
- Kook N Keech (44)

[+ 23 more](#)

PRICE

[₹100 - ₹300](#)

COLOR

- Black (496)
- White (375)
- Grey (367)
- Blue (337)
- Navy Blue (230)

**Stylecast X Kotty**

Printed Oversized T-shirt

Rs. 197 Rs. 1799 (89% OFF)**DIMPY GARMENTS**

Women Raw Edge T-shirt

Rs. 293 Rs. 1799 (Rs. 1506 OFF)**SZN**

Graphic Printed T-shirt

Rs. 232 Rs. 999 (Rs. 767 OFF)**SZN**

Women Printed Applique T-shirt

Rs. 232 Rs. 999 (Rs. 767 OFF)**Stylecast X Kotty**

Typography Printed T-shirt

Rs. 197 Rs. 1799 (89% OFF)**SZN**

Printed Oversized T-shirt

Rs. 232 Rs. 999 (Rs. 767 OFF)**glitchez**

Printed Oversized T-shirt

Rs. 276 Rs. 799 (Rs. 523 OFF)**Stylecast X Kotty**

Women Printed T-shirt

Rs. 197 Rs. 1799 (89% OFF)**SZN**

Graphic Printed T-shirt

Rs. 232 Rs. 999 (Rs. 767 OFF)**SZN**

Women Printed Cotton T-shirt

Rs. 233 Rs. 899 (Rs. 666 OFF)**NEW**

Filters

CATEGORIES

< Mobiles & Accessories

Mobiles

PRICE

Min to

BRAND

 Search Brand Apple Google MOTOROLA vivo OPPO Infinix

91 MORE

 Assured

CUSTOMER RATINGS

 4★ & above 3★ & above

GST INVOICE AVAILABLE

RAM

Home > Mobiles & A... > Mobiles

Showing 1–24 of 9,200 results for "smart phones"

Sort By **Relevance** Popularity Price -- Low to High Price -- High to Low Newest First**Samsung Galaxy F05 (Twilight Blue, 64 GB)**

4.2 ★ 55,419 Ratings & 3,257 Reviews

- 4 GB RAM | 64 GB ROM | Expandable Upto 1 TB
- 17.12 cm (6.74 inch) HD+ Display
- 50MP + 2MP | 8MP Front Camera
- 5000 mAh Battery
- Helio G85 Processor
- 1 Year Manufacturer Warranty for Device and 6 Months for In-Box Accessories

₹6,499

₹9,999 35% off

Upto ₹5,250 Off on Exchange

Bank Offer

**MOTOROLA g05 (Forest Green, 64 GB)**

4.2 ★ 36,685 Ratings & 2,753 Reviews

- 4 GB RAM | 64 GB ROM | Expandable Upto 1 TB
- 16.94 cm (6.67 inch) HD+ Display
- 50MP Rear Camera | 8MP Front Camera
- 5200 mAh Battery
- Helio G81 Processor
- 1 Year Warranty on Handset and 6 Months Warranty on Accessories

₹7,299

₹9,999 27% off

Upto ₹5,950 Off on Exchange

Bank Offer

**POCO C71 (Cool Blue, 128 GB)**

4.1 ★ 5,334 Ratings & 324 Reviews

- 6 GB RAM | 128 GB ROM | Expandable Upto 2 TB
- 17.48 cm (6.88 inch) HD+ Display
- 32MP Rear Camera | 8MP Front Camera
- 5200 mAh Battery
- Unisoc T7250 Max clock speed: 2 x A75@1.8GHz 6 x A55@1.6GHz Processor
- 1 Year Manufacturer Warranty for Phone and 6 Months Warranty for In the Box Accessories

₹6,999

₹9,999 30% off

Upto ₹5,700 Off on Exchange

Bank Offer

Types of Components

1. Functional Components (Preferred, modern way)
2. Class Components (Older, still important historically)

Functional Component

```
function Card() {  
  return <h1>This is my card!</h1>;  
}  
export default Card;
```

- A function that returns JSX.
- No need for this, simpler & cleaner.
- With Hooks, functional components replaced most class components.

Class Component

```
import { Component } from "react";

class Card extends Component {
  render() {
    return <h1>Hello, React!</h1>;
  }
}
export default Card;
```

- ❑ Older style, before Hooks.
- ❑ Uses render() method.
- ❑ Still seen in legacy codebases.

Props (Component Inputs)

```
function WelcomeCard(props) {  
  return <h1>Hello, {props.name}</h1>;  
}  
  
// Usage  
<WelcomeCard name="Manas" />
```

Props = Properties

Passed from parent to child

Read-only (cannot be changed inside child)

Props make components dynamic & reusable.

State (Component Memory)

State = Data that changes inside component

Managed with useState in functional components

Props = external input, State = internal memory.

```
import { useState } from "react";

function Counter() {
  const [count, setCount] = useState(0);

  return (
    <>
      <p>{count}</p>
      <button onClick={() => setCount(count + 1)}>Increment</button>
    </>
  );
}
```

Component Composition

- Nesting components inside others
- Makes UI modular
- Small components combined → complex UIs.

Component Lifecycle (Class vs Hooks)

```
import { useEffect } from "react";

useEffect(() => {
  console.log("Component Mounted");
  return () => console.log("Component Unmounted");
}, []);
```

- Mounting → Updating → Unmounting
- Class: componentDidMount, componentDidUpdate, componentWillUnmount
- Function: useEffect hook

Advanced Component Patterns

- Controlled vs Uncontrolled Components
- Higher-Order Components (HOC)
- Render Props
- Compound Components
- Custom Hooks
- Notes: ↗ Useful when scaling apps and managing complex UIs.

Best Practices

- Keep components small & focused
- Use PascalCase for component names
- Reuse logic with custom hooks instead of duplicating
- Separate UI (presentational) from business logic (container components)

Component v/s Functions

Aspect	JavaScript Function	React Component
Definition	A block of code that performs a specific task and returns a value.	A special function (or class) that returns JSX to describe UI.
Return Value	Returns primitive values, objects, arrays, or other functions.	Must return JSX (React elements) or null.
Naming Convention	Can be lowercase or uppercase (sum, addNumbers).	Must be PascalCase (Greeting, UserCard).
Input	Takes parameters/arguments.	Takes props as input.
Usage	Called directly like sum(2,3).	Used like an HTML tag : <Greeting name="Manas" />.
State Handling	Functions don't have built-in memory/state.	Components can manage state with useState, useReducer, etc.
Reusability	Reusable logic, but not tied to UI.	Reusable UI + logic as building blocks.
Lifecycle	Functions don't have lifecycle.	Components have lifecycle (mount, update, unmount).

Why Components?

- Reusability → Write once, use anywhere
- Maintainability → Code is easier to manage
- Readability → UI is split into logical parts
- Scalability → Large apps made manageable

Summary

Components = building blocks of React

Functional components are standard

Props = input, State = memory

Composition makes apps scalable

Lifecycle managed with Hooks

Advanced patterns → reusable, maintainable apps