Set 4 : JSX and HTML

---

1. Create a component that returns a `div` with your name and city inside `h1` and `p`.
2. Build a React component that shows a heading and 3 paragraphs using JSX.
3. Create a component that uses 3 different heading tags (`h1`, `h2`, `h3`) to display information.
4. Write a component that returns an unordered list with 5 favorite movies.
5. Display a static image using the `img` tag and `src` from the internet inside a React component.
6. Create a `<Greeting />` component that returns "Hello, welcome to React world!"
7. Create a React component with a table showing 3 students' names and marks.
8. Write a JSX snippet that includes `b`, `i`, `u`, and `mark` tags inside a paragraph.
9. Build a component that uses 3 `span` tags inside a `div` to separate words.
10. Create a component that returns 2 `<div>` blocks one after another using React fragments.

---

1. Create a `UserInfo` component that accepts `name`, `email`, and `age` as props and displays them.
2. Make a `ProductCard` component that accepts `title` and `price` as props. Render them in JSX.
3. Pass `imgURL`, `caption`, and `alt` as props to an `ImageBox` component and render an `img` element.
4. Build a `Greeting` component that takes a `name` prop and displays "Good Morning, {name}!"

5. Create a component called `Book` that takes `title`, `author`, and `year` as props.

6. Create a parent component `App` that renders 3 `ProductCard` components with different props.

7. Create a `WelcomeMessage` component that accepts a `language` prop and shows "Welcome" in that language.

8. Pass a long paragraph as a prop to a `TextBlock` component and display it in a `p` tag.

9. Pass an array of skills as a prop to a `SkillsList` component and render them using `map()` and `<li>`.

10. Make a `ProfileBadge` component with props: name, avatar (img src), and location. Render nicely.

---

1. Create a `UserCard` component that uses a nested `Avatar` and `Bio` component inside it. Pass props to each.

2. Create a `Person` component that accepts an `age` prop and shows “Adult” if age ≥ 18 else “Minor”.

3. Create a `FoodItem` component that takes `price` and `name` props and renders a sentence like “Pizza costs \$10”.

4. Make a component `ColorInfo` that takes a color prop and displays “The color is {color}”.

5. Pass 2 numbers as props to `Add` component and display their sum using `{}`.

6. Create a `Rating` component that takes a number between 1–5 as prop and renders stars (★) that many times.

7. Make a `YesNo` component that takes a `flag` prop (true/false) and shows “Yes” or “No”.

8. Pass an array of fruits as prop to `FruitList` and display it inside a `<ul>`.

9. Pass two strings to a component and show them combined using template literals in JSX.

10. Create a `TimeMessage` component that takes an `hour` prop (e.g., 14) and returns “Good Afternoon” if hour is between 12–17.

📌 Bonus: Structure Example

```
function Greeting(props) {
  return (
    <div>
      <h1>Hello, {props.name}</h1>
    </div>
  );
}
```