**Explanation of React Code:**

* **Setting Up the SVG Canvas** – The useRef hook is used to reference the SVG element, and d3.select initializes it with a width and height of 400x400 pixels.
* **Drawing the Cable Structure** – A large black circle represents the outer boundary of the cable, and inside it, a smaller white circle with a gray border acts as the insulation layer.
* **Adding Conductors** – Four conductors are placed inside the insulation, each with an outer black border and a colored core (orange, blue, and light pink).
* **Enhancing with Details** – Small brown strands are positioned inside each conductor to represent wire bundles, and gray ellipses are drawn around the conductors to create a structured layout.
* **Final Touches** – An extra set of strands is added to the top-right for additional detailing, along with a small gray arc to complete the visualization.

**Explanation of Next.js Code:**

* **Reused React Component** – Maintained modularity by integrating the previously created component. This ensures consistency across implementations.
* **Set Up SVG Canvas** – Used useRef and d3.select to initialize a 400x400 pixel canvas. This allows dynamic rendering of the cable structure.
* **Created Cable Structure** – Designed outer insulation, conductors, and strands for an accurate visualization. Different colors and shapes represent distinct cable elements.
* **Enhanced with Extra Details** – Added additional strands and a gray arc for improved realism. These elements refine the visual representation of the cable.
* **Implemented SSR as Props** – Enabled server-side rendering to dynamically pass cable structure data. This enhances performance and ensures up-to-date visualization.

**Explanation of Vue.js Code:**

* **Reused Previously Created Vue Component** – Ensured consistency and modularity by leveraging an existing component.
* **Implemented SSR as Props** – Added support for server-side rendering by passing necessary props.
* **Created Cable Structure** – Designed the outer insulation, conductors, and strands using D3.js for visualization.
* **Added Extra Strands & Arc** – Enhanced visual detailing and realism with additional elements.
* **Used Ref & Lifecycle Hooks** – Utilized ref and onMounted for dynamic SVG rendering.  
    
    
  ***Note: For all these I have used the same css code.***