💻 Video Streaming Dashboard Assignment:

Technical Implementation (React Edition)
This assignment requires the creation of a synchronized multi-stream dashboard using React and HTTP Live Streaming (HLS).

📝 Task Details: Technical Implementation
1. HLS Generation Pipeline
- RTSP Source:
  - Use the provided RTSP link as the source for your streams: rtsp://13.60.76.79:8554/live
- HLS Generation:
  - Identify and use a suitable tool (e.g., FFmpeg, MediaMTX, or similar media server software) to continuously convert the RTSP stream into an accessible HLS URL (.m3u8 playlist).
- Crucial Step: You must implement a method to simulate or create 5 to 6 distinct HLS URLs from the single provided RTSP source.

2. React Dashboard Development
- Framework: Develop the entire dashboard using the React framework (functional components with Hooks).
- Player Library: Integrate an appropriate HLS Player library that works well within a React environment (e.g., a React wrapper for hls.js, or Video.js).
- Display Layout:
  - Create a dashboard that displays all 5 to 6 video players simultaneously.
  - The layout should be a responsive grid or matrix (e.g., 2x3 or 3x2).

- Synchronization (Primary Focus):
  - The most critical requirement is to ensure the 5-6 streams play as close to "in sync" as possible.
  - Implement specific React logic, player configuration, or control methods to achieve and maintain this tight synchronization across all video elements.

3. Reference
- The website https://monitor.theun1t.com/ serves as a visual reference for the general multi-view monitoring dashboard layout and feel.

📦 Deliverables
- Dashboard Code (GitHub Repository Link):
  - A link to the repository containing the complete React application code.
- The repository README.md must contain all technical documentation, including:
  - A detailed explanation of the RTSP \rightarrow HLS conversion process and the tools used.
  - An explanation of how the 5-6 distinct HLS streams were created/simulated.
  - A description of the React component logic and player methods implemented to achieve stream synchronization.

- The README.md must also explain how to set up the environment and run the application (npm install and npm start).
- Live Workable Link (Vercel/Netlify/Similar):
    - A link to the live, deployed version of the React dashboard (e.g., hosted on Vercel, Netlify, or similar platforms). This link must be functional and demonstrate the synchronized stream playback.

⏱️ Deadline:

- 5pm, Saturday, 21st November 2025

Please Submit the Assignment to **careers@protomedialive.com**