User Documentation — Livestream Overlay App

Overview

The **Livestream Overlay App** allows users to play livestreams (RTSP/HLS) and manage custom overlays (text or image). These overlays can be **dragged**, **resized**, and automatically saved in MongoDB.

Prerequisites

Before starting, ensure the following are installed on your system:

- **Python 3.10+**
- **Node.js 18+**
- **React (via Vite)**
- **MongoDB** (local or MongoDB Atlas)
- **FFmpeg** (installed and added to PATH)
- **npm** (Node Package Manager, usually comes with Node.js)

Setup Instructions

1. Backend Setup (Flask)

1. Open Command Prompt and navigate to your project backend folder:

   ```
   cd backend
   ```

2. Create and activate a virtual environment:

   ```
   python -m venv venv
   venv\Scripts\activate
   ```

3. Install all required dependencies:

   ```
   pip install -r requirements.txt
   ```

4. Create a `.env` file in your backend directory:

```
MONGO_URI=mongodb://localhost:27017/overlaydb
PORT=5001
```

5. Start the Flask backend:

   ```
   python app.py
   ```

   The backend will run at **http://127.0.0.1:5001**

---

2. Frontend Setup (React + Vite)

1. Open a new terminal window and navigate to your frontend folder:

   ```
   cd frontend
   ```

2. Install dependencies:

   ```
   npm install
   ```

3. Start the React development server:

   ```
   npm run dev
   ```

4. Open the app in your browser at **http://localhost:5173**

Using the Application

Step 1 — Input RTSP / HLS Stream URL

The app uses a default HLS test stream:

```
https://test-streams.mux.dev/x36xhzz/x36xhzz.m3u8
```

If you want to use your own RTSP stream: 1. Open `App.jsx`. 2. Locate this line: `jsx const streamUrl = "https://test-streams.mux.dev/x36xhzz/x36xhzz.m3u8";` 3. Replace it with your RTSP URL, for example: `jsx const streamUrl = "rtsp://192.168.1.2:554/live";` 4. Save the file — the video player will automatically load your new stream.

Supported formats: `.m3u8`, `.mp4`, `.rtsp`, `.flv`

Step 2 — Create Overlays

1. On the dashboard, find the **Create Overlay** form.
2. Fill in details:

- **Name**: Overlay name (e.g., *LogoOverlay*)
- **Type**: Choose between `text` or `image`
- **Content**:
  - For text overlays → write text to display
  - For image overlays → paste image URL
- Adjust `x`, `y`, `width`, `height`, and `opacity` values.

3. Click **Add Overlay**.

The overlay will appear over the video instantly and be saved in MongoDB.

Step 3 — Move and Resize Overlays

1. Toggle the **Editing Mode** switch (top-right of video player).
2. Drag overlays anywhere on the video.
3. Resize overlays by dragging their corners.
4. When you release the overlay:
   - The new position (x, y) and size (`width`, `height`) are auto-saved to the backend.
   - You'll see confirmation logs in the console.

These updated values are stored in MongoDB — refreshing the page will show the new overlay positions.

Step 4 — Delete Overlays

1. Scroll to the **Available Overlays** list.
2. Click **Delete** next to any overlay you want to remove.
3. The overlay disappears from both the video and MongoDB instantly.

Example Usage Workflow

1. Start both backend and frontend.
2. Enter your RTSP or HLS video URL.
3. Create a new overlay named *Live Update* with red text.
4. Add your channel logo using image overlay.
5. Drag both to preferred positions.
6. Resize if needed.
7. Refresh the page — all overlays persist!
