Real-Time Programme Display System: Technical Manual

Overview

This project is a real-time schedule display and admin control system, designed for professional programme/event management. It enables an admin to control, edit, and synchronize a schedule across multiple display windows in real time, with a polished, user-friendly interface.

Project Structure

- `src/app/admin/` — Admin UI (Angular component, HTML, CSS)
- `src/app/display/` — Display UI (Angular component, HTML, CSS)
- `src/app/shared-schedule.service.ts` — Shared schedule logic and localStorage sync
- `public/` — Static assets (favicon, etc.)
- `angular.json`, `package.json`, `tsconfig.json` — Angular project config

Core Features

- **Real-time sync**: Admin and display windows stay in sync using `localStorage` events and `postMessage`.
- **Admin UI**: Add, edit, skip, and end programmes; open display in new window; control fullscreen remotely; download/upload schedule.
- **Display UI**: Professional, fullscreen-friendly, auto-updating, with large time and subtle title.
- **Persistent display position**: Remembers last display window position.
- **Modern, responsive design**: Clean, accessible, and mobile-friendly.

Real-Time Sync Architecture

- **localStorage events**: Schedule changes are written to localStorage; all open windows listen for `storage` events to update.
- **postMessage**: Used for fullscreen control and cross-window communication.
- **Display position**: Saved to localStorage on move/close, reused when opening display.

Admin UI Details

- **Tabs**: Input, Schedule, Current Programme
- **Input**: Add via form, paste, or file upload
- **Schedule**: View, download as text, status tracking
- **Current**: See current item, edit inline, skip, end now, preview display, open/remote fullscreen

---

Display UI Details

- **Large, bold time**: Fills 3/4 of screen
- **Small, subtle title**: For context
- **Fullscreen icon**: Top-right, for manual fullscreen
- **No controls visible to audience**

How to Build & Run

1. Install dependencies:

   ```
   npm install
   ```

2. Start the dev server:

   ```
   npm start
   ```

3. Open the admin UI in your browser (usually at `http://localhost:4200`).

4. Use "Open Display in New Window" to launch the display.

Deployment

- Build for production:

  ```
  npm run build
  ```

- Deploy the contents of `dist/` to your web server.

Extensibility & Customization

- **Schedule format**: Each line is `Title;Start Time;Duration` (e.g., `Show 1;09:00;30`)
- **Add more fields**: Extend the `Programme` interface and update forms/UI as needed.
- **Styling**: Tweak CSS in `admin.component.css` and `display.component.css`.

- **Sync**: For multi-user or server-based sync, replace localStorage with a backend API/websocket.

Troubleshooting

- If real-time sync fails, ensure all windows are on the same origin and localStorage is enabled.
- Fullscreen may require user interaction in some browsers.
- For best results, use Chrome or Edge.

Authors & License

- Developed by [Your Name/Team]
- MIT License (customize as needed)