🌐 Global Schedule Database Deployment Guide

Your schedule system now uses a **server-side database** that's accessible globally! Here's how to deploy it:

📋 What You Need:

Files to Upload:

1. `api.php` - Server API for database operations
2. `viewer.html` - Updated viewer (uses server API)
3. `admin.html` - Updated admin panel (uses server API)

Server Requirements:

- ✅ **PHP 7.0+** (most web hosts support this)
- ✅ **File write permissions** (for JSON database file)
- ✅ **CORS enabled** (for API access)

🚀 Deployment Steps:

Step 1: Upload Files

```
your-website.com/
├── schedule/
│   ├── api.php          ← Server API
│   ├── viewer.html      ← Updated viewer
│   ├── admin.html       ← Updated admin
│   └── schedule_data.json ← Auto-created database
```

Step 2: Update API URLs

In both `viewer.html` and `admin.html`, change this line:

```
javascript

const API_URL = 'api.php';
```

To your server path:

```
javascript
```

```
const API_URL = 'https://yourserver.com/schedule/api.php';
```

Step 3: Set Permissions

Make sure your server can write files:

```
bash

chmod 755 /path/to/schedule/
chmod 644 /path/to/schedule/api.php
```

Step 4: Test the Setup

1. **Open admin panel** → login and add events
2. **Open viewer** → should show the same events
3. **Check from different devices** → all should sync!

🔧 API Endpoints:

GET / - Get all schedule data

```
http

GET /api.php
Response: {"2024-01-15": [events...], "2024-01-16": [events...]}
```

GET /?date=YYYY-MM-DD - Get specific date

```
http

GET /api.php?date=2024-01-15
Response: {"date": "2024-01-15", "events": [events...]}
```

POST / - Save events for a date

```
http

POST /api.php
Body: {"date": "2024-01-15", "events": [events...]}
Response: {"success": true, "message": "Schedule updated"}
```

DELETE /?date=YYYY-MM-DD - Delete events for a date

```
http

DELETE /api.php?date=2024-01-15
Response: {"success": true, "message": "Schedule deleted"}
```

🎯 Benefits of Global Database:

✅ Universal Access:

- **Same data everywhere** - all devices show identical schedules
- **Real-time updates** - changes appear on all screens within 30 seconds
- **No setup required** - viewers just need the URL

✅ Easy Management:

- **Centralized control** - admin manages from anywhere
- **Backup friendly** - single JSON file contains everything
- **Portable** - easy to move between servers

✅ Scalable:

- **Multiple locations** - each can have their own API
- **Multiple admins** - all can edit the same schedule
- **Multiple viewers** - unlimited displays can connect

🛡️ Security Features:

- **Admin authentication** remains client-side for simplicity
- **API is read-heavy** - viewers only read data
- **CORS protection** - can restrict which domains can access
- **No database credentials** - uses simple file storage

🔄 Fallback Protection:

If server is unavailable:

- **Viewer shows** "Offline mode" with cached data
- **Admin shows** "Server error" and uses local mode
- **Automatic retry** every 30 seconds until reconnected

📱 Usage Examples:

Hotel Setup:

- Upload to: `hotel-website.com/schedule/`
- Viewers in rooms: `hotel-website.com/schedule/viewer.html`
- Admin access: `hotel-website.com/schedule/admin.html`

Office Setup:

- Upload to: `company.com/events/`
- Conference room displays: `company.com/events/viewer.html`
- Event managers: `company.com/events/admin.html`

🔧 Troubleshooting:

"Server Error" in Admin:

- Check API URL in admin.html
- Verify PHP is working
- Check file permissions

"Offline Mode" in Viewer:

- Check API URL in viewer.html
- Test API directly: `yourserver.com/schedule/api.php`
- Check CORS settings

Changes Not Syncing:

- Verify both viewer and admin use same API URL
- Check browser console for errors
- Test API endpoints manually

🎉 You're Done!

Your schedule system is now **globally accessible**! Changes made on the admin panel will appear on all viewer displays worldwide within 30 seconds.

Perfect for hotels, offices, conference centers, or any multi-location setup! 🌟