



Deployment Summary - PR #149

Critical Issues Fixed



Issue 1: Wolfpack Labels Lost After Reboot (CRITICAL - FIXED)

Problem: User's Wolfpack input/output labels were being lost after rebooting the system.

Root Cause: Multiple database files existed in different locations:

- `./data/sports_bar.db` (GitHub repo)
- `./prisma/data/sports_bar.db` (User's installation)

The app was switching between these databases, making labels appear "lost."

Solution:

- Standardized database location to `./prisma/data/sports_bar.db`
 - Created automatic migration script
 - Updated deployment process to handle migration
-



Issue 2: Image Validation Errors (FIXED)

Problem: Next.js Image component throwing validation errors for uploaded layout images.

Solution: Replaced Next.js `Image` component with regular `` tag in `LayoutConfiguration` component.



Issue 3: Wolfpack Rows of 4 (VERIFIED WORKING)

Status: Already working correctly from PR #148. No changes needed.



Pull Request Created

PR #149: <https://github.com/dfultonthebar/Sports-Bar-TV-Controller/pull/149>

Branch: `fix/wolfpack-persistence-and-images`

Files Changed:

1. `.env.example` - Updated `DATABASE_URL`
 2. `src/components/LayoutConfiguration.tsx` - Fixed image validation
 3. `update_from_github.sh` - Added automatic migration
 4. `scripts/migrate-database-location.sh` - New migration script
 5. `CRITICAL_FIXES_PR149.md` - Comprehensive documentation
 6. `CRITICAL_FIXES_PR149.pdf` - PDF documentation
-

Quick Deployment Guide

For the User's Installation

```
# Navigate to installation directory
cd ~/Sports-Bar-TV-Controller

# Backup current data (IMPORTANT!)
cp prisma/data/sports_bar.db prisma/data/sports_bar.db.backup.$(date +%Y%m%d_%H%M%S)
cp .env .env.backup.$(date +%Y%m%d_%H%M%S)

# Pull the latest changes
git pull origin main

# Run the migration script (handles database consolidation)
./scripts/migrate-database-location.sh

# Install dependencies
npm install

# Regenerate Prisma client
npx prisma generate

# Build the application
npm run build

# Restart the application
pm2 restart sports-bar-tv-controller

# Check status
pm2 status
pm2 logs sports-bar-tv-controller --lines 50
```


Verification Steps

1. Verify Database Location

```
cat ~/Sports-Bar-TV-Controller/.env | grep DATABASE_URL
# Expected: DATABASE_URL="file:./prisma/data/sports_bar.db"

ls -lh ~/Sports-Bar-TV-Controller/prisma/data/sports_bar.db
# Should show the database file exists
```

2. Verify Wolfpack Labels Persist

1. Open Wolfpack configuration page
2. Check all input/output labels are present
3. Make a test change to a label
4. Save the configuration
5. Run: `pm2 restart sports-bar-tv-controller`
6. Verify the label change persisted 

3. Verify Image Display

1. Go to Layout Configuration page
2. Upload a test image or PDF
3. Verify it displays without errors ✓
4. Check browser console (no validation errors) ✓

4. Verify Rows of 4 Display

1. Open Wolfpack/Matrix control page
2. Verify inputs display in rows of 4 ✓
3. Verify outputs display in rows of 4 ✓



What the Migration Script Does

The `scripts/migrate-database-location.sh` script:

1. **Finds all database files** in both locations
2. **Selects the most recent one** (contains latest data)
3. **Copies it to standardized location** (`prisma/data/sports_bar.db`)
4. **Updates .env file** with correct DATABASE_URL
5. **Creates backups** of existing files
6. **Verifies database integrity**

Safe to run multiple times - it's idempotent and won't cause data loss.



Safety Features

1. **Automatic Backups:** Migration script creates backups before making changes
2. **No Data Loss:** Selects most recent database (contains latest data)
3. **Idempotent:** Safe to run multiple times
4. **Integrated into Update Script:** Runs automatically during updates
5. **Rollback Plan:** Simple rollback if issues occur



Technical Details

Database Path Resolution

- **Before:** `DATABASE_URL="file:../data/sports_bar.db"`
- **After:** `DATABASE_URL="file:../prisma/data/sports_bar.db"`

Prisma resolves paths relative to project root (where `package.json` is).

Why This Location?

- Conventional location for Prisma databases
- Keeps database with Prisma schema files
- Prevents confusion with multiple locations

- `.env` is in `.gitignore` - persists across updates

Image Component Change

- **Before:** Next.js `Image` component (strict validation)
- **After:** Regular `` tag (graceful fallback)
- **Result:** No validation errors for dynamic uploads

Troubleshooting

If Labels Still Missing After Deployment

```
# Check which database file is being used
cat ~/Sports-Bar-TV-Controller/.env | grep DATABASE_URL

# Check if database file exists
ls -lh ~/Sports-Bar-TV-Controller/prisma/data/sports_bar.db

# Run migration script again
cd ~/Sports-Bar-TV-Controller
./scripts/migrate-database-location.sh

# Restart application
pm2 restart sports-bar-tv-controller
```

If Images Still Show Validation Errors

```
# Check if the fix was applied
grep -n "img src" ~/Sports-Bar-TV-Controller/src/components/LayoutConfiguration.tsx

# Should show regular img tag, not Image component

# Rebuild application
cd ~/Sports-Bar-TV-Controller
npm run build
pm2 restart sports-bar-tv-controller
```

Check Application Logs

```
# View recent logs
pm2 logs sports-bar-tv-controller --lines 100

# View errors only
pm2 logs sports-bar-tv-controller --err --lines 50

# Monitor in real-time
pm2 logs sports-bar-tv-controller
```

Rollback Plan

If issues occur after deployment:

```
cd ~/Sports-Bar-TV-Controller

# Restore previous .env
cp .env.backup.YYYYMMDD_HHMMSS .env








# Restore previous database
cp prisma/data/sports_bar.db.backup.YYYYMMDD_HHMMSS prisma/data/sports_bar.db

# Rebuild and restart
npm run build
pm2 restart sports-bar-tv-controller
```

Documentation Files

1. **CRITICAL_FIXES_PR149.md** - Comprehensive technical documentation
2. **CRITICAL_FIXES_PR149.pdf** - PDF version for offline reference
3. **DEPLOYMENT_SUMMARY_PR149.md** - This quick reference guide
4. **scripts/migrate-database-location.sh** - Executable migration script

Expected Results After Deployment

1.  Wofpack labels persist after application restart
2.  Wofpack labels persist after system reboot
3.  Layout images display without validation errors
4.  Wofpack inputs/outputs display in rows of 4
5.  No console errors related to images or database
6.  Application runs stably on port 3001
7.  All existing functionality continues to work

Support

GitHub PR: <https://github.com/dfultonthebar/Sports-Bar-TV-Controller/pull/149>

Documentation: See CRITICAL_FIXES_PR149.md for detailed information

Logs Location: `pm2 logs sports-bar-tv-controller`

Database Location: `~/Sports-Bar-TV-Controller/prisma/data/sports_bar.db`

Important Reminders

1. **Always backup before deploying** - The migration script does this automatically
2. **Run migration script** - It's now integrated into the update process
3. **Verify after deployment** - Follow the verification checklist

4. **Check logs** - Monitor for any errors after restart
 5. **Test Wolfpack labels** - Make a test change and verify persistence
-

Ready to Deploy!

The PR is ready for review and deployment. All critical issues have been addressed with comprehensive solutions, automatic migration, and safety features.

Next Steps:

1. Review the PR on GitHub
 2. Follow the deployment guide above
 3. Verify all fixes are working
 4. Confirm Wolfpack labels persist across reboots
-

Created: October 8, 2025

PR Number: #149

Status: Ready for Deployment 