

Deployment Fix Status - Script Execution Error

Issue Analysis

- **Error:** "Exited with status 127 while running your pre-deploy script"
- **Cause:** Bash script execution failed on Render
- **Status Code:** 127 = command not found
- **Failed Commit:** fb75e1e

Root Cause

The `migrate:deploy` script was attempting to execute an external bash script:

```
"migrate:deploy":  
"bash scripts/resolve-failed-migration-20251218.sh && prisma migrate deploy"
```

Why it failed:

- External script dependencies are unreliable in Render's build environment
- Path resolution issues
- Potential shell interpreter unavailability
- Script permissions might not be preserved during deployment

Solution Implemented

Changed Approach

Inlined the migration resolution command directly in `package.json`:

Before:

```
"migrate:deploy":  
"bash scripts/resolve-failed-migration-20251218.sh && prisma migrate deploy"
```

After:

```
"migrate:deploy": "npx prisma migrate resolve --rolled-back  
20251218162945_update_homes_to_active || true && npx prisma migrate deploy"
```

How It Works

1. `npx prisma migrate resolve --rolled-back 20251218162945_update_homes_to_active`
 - Resolves the specific failed migration
 - Marks it as rolled back in the database

2. `|| true`

- Ensures the command continues even if migration is already resolved
- Makes the command idempotent (safe to run multiple times)

3. `&& npx prisma migrate deploy`

- Proceeds to deploy all pending migrations
- Only runs if the previous command succeeds

Benefits

- No external script dependencies
- More reliable in CI/CD environments
- Idempotent - safe to run multiple times
- Better error visibility
- Portable across different deployment platforms

Deployment Timeline

Step	Status	Time
Fix developed	<input checked="" type="checkbox"/> Complete	Now
Changes committed	<input checked="" type="checkbox"/> Complete	Commit 938b333
Pushed to GitHub	<input checked="" type="checkbox"/> Complete	Now
Render auto-deploy triggered	 In Progress	~1-2 min
Build phase	 Pending	~3-5 min
Pre-deploy script	 Pending	~10-15 sec
Migration deployment	 Pending	~5-10 sec
Health checks	 Pending	~1-2 min
Total Est. Time		~6-10 minutes

Expected Behavior

Pre-Deploy Script Execution

```
# What Render will execute:  
npm run migrate:deploy  
  
# Which translates to:  
npx prisma migrate resolve --rolled-back 20251218162945_update_homes_to_active ||  
true && npx prisma migrate deploy
```

Expected Output

```
✓ Migration 20251218162945_update_homes_to_active marked as rolled back (or already resolved)
✓ Running prisma migrate deploy...
✓ Applied 0 migrations (all up-to-date)
✓ Database schema is current
```

Verification Steps

Once deployment completes:

1. Check Render Dashboard

- Navigate to: <https://dashboard.render.com/web/srv-d3isol3uibrs73d5m1g/events>
- Look for: “Deploy succeeded”
- Verify: No status 127 errors

2. Check Application Health

```
# Test homepage
curl https://carelinkai.onrender.com/

# Should return 200 OK
```

3. Test Tour Submission

- Navigate to: <https://carelinkai.onrender.com/find-care>
- Submit a test tour request
- Verify: No “internal server error”
- Check: Database for new tour entry

4. Check Database Migration Status

```
# In Render Shell or locally with production DB:
npx prisma migrate status

# Expected output:
# Database schema is up to date!
# No pending migrations.
```

Rollback Plan

If this deployment still fails:

Option 1: Remove Migration Resolution

Update package.json to skip the resolution:

```
"migrate:deploy": "npx prisma migrate deploy"
```

Option 2: Manual Database Fix

Connect to production database and run:

```
UPDATE _prisma_migrations
SET finished_at = now(),
    migration_name = '20251218162945_update_homes_to_active',
    logs = 'Manually marked as rolled back'
WHERE migration_name = '20251218162945_update_homes_to_active'
AND finished_at IS NULL;
```

Option 3: Fresh Migration

If all else fails, create a new idempotent migration that handles the schema changes.

Files Modified

File	Change	Purpose
package.json	Updated <code>migrate:deploy</code> script	Inline migration resolution

Commits

Commit	Message	Files
938b333	fix: Inline migration resolution	package.json
fb75e1e	Previous attempt (failed)	Multiple

Monitoring

Key Metrics to Watch

1. **Deployment Duration:** Should complete in 6-10 minutes
2. **Build Success:** No compilation errors
3. **Migration Success:** Pre-deploy script exits with 0
4. **Health Checks:** All pass within 2 minutes
5. **Tour Submissions:** Working without errors

Alert Conditions

⚠ Red Flags:

- Deployment takes > 15 minutes
- Status 127 errors persist
- Health checks fail
- Tour submissions still fail

Next Steps

1. Monitor Render dashboard for deployment progress
2. Wait for “Deploy succeeded” message (~6-10 min)
3. Test tour submission functionality
4. Verify database migration status
5. Confirm all features working

Technical Details

Migration Being Resolved

- **Name:** 20251218162945_update_homes_to_active
- **Purpose:** Update AssistedLivingHome status to ACTIVE
- **Issue:** Failed during initial deployment
- **Resolution:** Marking as rolled back, then reapplying via deploy

Database Schema Change

```
-- Migration adds/updates:
UPDATE "AssistedLivingHome"
SET status = 'ACTIVE'
WHERE status IS NULL OR status != 'ACTIVE';
```

Success Criteria

- Deployment completes without errors
- Pre-deploy script exits with status 0
- Database migrations applied successfully
- Application health checks pass
- Tour submission form works
- No console errors in browser
- No internal server errors in logs

Conclusion

This fix addresses the root cause of the deployment failure by:

- Eliminating external script dependencies
- Using native Prisma CLI commands
- Making the migration resolution idempotent
- Improving reliability across deployment environments

Status: Fix deployed - awaiting Render build completion

Next Update: Check deployment status in 10 minutes

Generated: December 18, 2025

Commit: 938b333

Deployment: Auto-triggered on push