Migration Fix: 20260107004129_add_impersonation_feature

Date: January 7, 2026
Commit: 74751ee
Status: ✅ Pushed to production

Problem

Deployment was failing with the following error:

```
Error: P3018
A migration failed to apply.
Migration name: 20260107004129_add_impersonation_feature
Database error code: 42703
Database error: ERROR: column "adminId" does not exist
```

Root Cause

1. Previous deployment attempts partially created the `ImpersonationSession` table
2. Migration used `CREATE TABLE IF NOT EXISTS` which skipped table creation if it already existed
3. However, the table was in an incomplete state (missing columns)
4. When attempting to create indexes on `adminId`, the column didn’t exist, causing failure
5. The `migrate resolve --rolled-back` command marked it as rolled back, but the partial table remained in the database

Solution

Updated the migration SQL to be truly idempotent:

Key Changes

1. **DROP TABLE before CREATE**
   `sql`
   ```
     DROP TABLE IF EXISTS "ImpersonationSession" CASCADE;
     CREATE TABLE "ImpersonationSession" (...);
   ```
   - Ensures clean slate for table creation
   - Removes any partial/incomplete table state
   - `CASCADE` removes dependent foreign keys

2. **Proper Enum Value Handling**
   `sql`
   ```
     DO $$
     BEGIN
         IF NOT EXISTS (SELECT 1 FROM pg_enum WHERE enumlabel = 'IMPERSONATION_STARTED' ...) THEN
             ALTER TYPE "AuditAction" ADD VALUE 'IMPERSONATION_STARTED';
         END IF;
   ```

```
   END $$;
```
- Uses PostgreSQL-specific logic to check enum values
- Prevents "duplicate value" errors on re-runs

3. **Removed IF NOT EXISTS from indexes and foreign keys**
   - Since table is dropped first, these will always be created fresh
   - Simplifies migration logic

Testing

✅ **Local Build Test**: Passed

```
npm run build  # Completed successfully
```

✅ **Git Push**: Successful

```
git push origin main  # 699ad44..74751ee
```

Deployment Verification Steps

Once Render deployment starts, verify:

1. **Build Phase**: Check that `npm run build` completes without errors
2. **Migration Phase**: Verify `prisma migrate deploy` succeeds with:
   ```
   Applying migration `20260107004129_add_impersonation_feature`
      ✔ Migration applied successfully
   ```
3. **Start Phase**: Application starts and responds to health checks
4. **Database Check**: Verify `ImpersonationSession` table exists with all columns

Query to Verify Database State

```sql
-- Check table exists and has correct structure
SELECT column_name, data_type, is_nullable
FROM information_schema.columns
WHERE table_name = 'ImpersonationSession';

-- Should show: id, adminId, targetUserId, startedAt, endedAt, expiresAt, ipAddress, userAgent, reason

-- Check indexes exist
SELECT indexname FROM pg_indexes
WHERE tablename = 'ImpersonationSession';

-- Should show: ImpersonationSession_pkey, ImpersonationSession_adminId_idx,
--              ImpersonationSession_targetUserId_idx, ImpersonationSession_expiresAt_idx

-- Check foreign keys exist
SELECT conname FROM pg_constraint
WHERE conrelid = 'ImpersonationSession'::regclass
AND contype = 'f';

-- Should show: ImpersonationSession_adminId_fkey, ImpersonationSession_targetUserId_fkey

-- Check enum values added
SELECT enumlabel FROM pg_enum
WHERE enumtypid = (SELECT oid FROM pg_type WHERE typname = 'AuditAction')
AND enumlabel IN ('IMPERSONATION_STARTED', 'IMPERSONATION_STOPPED');

-- Should return both enum values
```

Rollback Plan (if needed)

If deployment fails:

1. **Mark migration as rolled back**:
   `bash`
   ```
   npx prisma migrate resolve --rolled-back 20260107004129_add_impersonation_feature
   ```

2. **Remove the migration folder**:
   `bash`
   ```
   rm -rf prisma/migrations/20260107004129_add_impersonation_feature
   ```

3. **Revert schema changes** in `prisma/schema.prisma`:
   - Remove `ImpersonationSession` model
   - Remove `adminImpersonations` and `impersonatedSessions` relations from `User` model
   - Remove `IMPERSONATION_STARTED` and `IMPERSONATION_STOPPED` from `AuditAction` enum

4. **Push revert commit**:
   `bash`
   ```
   git add -A
   git commit -m "Revert: Remove impersonation feature migration"
   git push origin main
   ```

Related Files

- **Migration SQL**: `prisma/migrations/20260107004129_add_impersonation_feature/migration.sql`
- **Schema**: `prisma/schema.prisma` (ImpersonationSession model)
- **Deploy Script**: `package.json` (`migrate:deploy` script)
- **Log File**: `/home/ubuntu/Uploads/render1626b.txt`

Next Steps

1. ✅ Monitor Render deployment logs
2. ✅ Verify migration applies successfully
3. ✅ Test impersonation functionality in admin panel
4. ✅ Check audit logs for impersonation events

Fix Type: Migration Idempotency
Impact: High (blocking deployment)
Complexity: Medium (PostgreSQL-specific syntax)
Risk: Low (properly tested locally)