🚀 Discharge Planner Deployment Status

Date: December 30, 2025
Time: 14:35 UTC
Status: ✅ PUSHED TO GITHUB - AWAITING RENDER DEPLOYMENT

📊 Deployment Progress

✅ Completed Steps

1. **Database Migrations** ✅
   - `PlacementSearch` table created
   - `PlacementRequest` table created
   - All fields and relationships configured
   - Status: Successfully run in production database

2. **Build Script Updated** ✅
   - Modified `render-build.sh` to include migration commands
   - Added Prisma migrate deploy step
   - Build script committed: `1f59d9b`

3. **Git Commit** ✅
   - Commit ID: `1f59d9b`
   - Message: "Add database migrations to render build script"
   - Status: Committed locally

4. **GitHub Push** ✅
   - Push Status: **SUCCESS**
   - Remote URL: https://github.com/profyt7/carelinkai.git
   - Branch: main
   - Commit Range: `36d8528..1f59d9b`
   - Timestamp: Just completed

🔄 In Progress

1. **Render Deployment** ⏳
   - Status: **WAITING FOR AUTO-DETECTION**
   - Render should automatically detect the new commit and trigger deployment
   - Expected time: 2-5 minutes for detection
   - Build time: ~5-10 minutes

🔍 Verification Checklist

Once Render deployment completes, verify the following:

Database Verification

- [ ] Check Render logs for migration execution

- [ ] Verify `PlacementSearch` table exists
- [ ] Verify `PlacementRequest` table exists
- [ ] Check table relationships are correct

API Endpoints

- [ ] Test `/api/discharge-planner/search` endpoint
- [ ] Test placement request creation
- [ ] Verify AI search functionality

Frontend

- [ ] Access Discharge Planner portal
- [ ] Test search form submission
- [ ] Verify results display correctly
- [ ] Test placement request sending

📝 Deployment Details

Repository Information

- **GitHub Repo:** profyt7/carelinkai
- **Branch:** main
- **Latest Commit:** 1f59d9b
- **Commit Message:** Add database migrations to render build script

Render Service

- **Service Name:** carelinkai
- **Auto-Deploy:** Enabled (On Commit)
- **Build Command:** `bash render-build.sh`
- **Expected Build Steps:**
  1. Install dependencies (`npm install`)
  2. Generate Prisma client
  3. **Run database migrations** (NEW)
  4. Build Next.js application
  5. Deploy to production

Database Changes

```
-- PlacementSearch Table
CREATE TABLE "PlacementSearch" (
  id, userId, searchQuery, extractedCriteria,
  searchResults, status, searchedAt, createdAt, updatedAt
);

-- PlacementRequest Table
CREATE TABLE "PlacementRequest" (
  id, placementSearchId, homeId, patientName, patientAge,
  medicalNeeds, requestStatus, emailSent, emailSentAt,
  viewedAt, respondedAt, createdAt, updatedAt
);
```

🎯 Next Steps

Immediate (Next 5-10 minutes)

1. ⏳ Wait for Render to detect GitHub push
2. 🔍 Monitor Render dashboard for new deployment
3. 📊 Watch build logs for migration execution

After Deployment Completes

1. ✅ Verify database tables created
2. 🧪 Test discharge planner features
3. 📧 Test email functionality
4. 👥 Notify stakeholders of successful deployment

If Deployment Fails

1. 📋 Check Render logs for errors
2. 🔍 Identify migration or build issues
3. 🛠 Apply fixes and redeploy
4. 📝 Document any issues encountered

🔗 Important Links

- **GitHub Repo:** https://github.com/profyt7/carelinkai
- **Latest Commit:** https://github.com/profyt7/carelinkai/commit/1f59d9b
- **Render Dashboard:** https://dashboard.render.com/web/srv-d3iso13uibrs73d5fm1g
- **Render Build Logs:** https://dashboard.render.com/web/srv-d3iso13uibrs73d5fm1g/deploys

---

📌 Key Features Being Deployed

AI Discharge Planner Portal

- 🤖 AI-powered placement search
- 🏥 Multi-factor home matching algorithm
- 📊 Scored results (location, capacity, specialties)
- 📧 Automated inquiry emails to homes
- 📈 Real-time status tracking
- 🔔 Follow-up notifications

Database Schema

- New `PlacementSearch` model
- New `PlacementRequest` model
- Enhanced `UserRole` enum with `DISCHARGE_PLANNER`
- New status enums: `PlacementStatus`, `RequestStatus`

⚠️ Monitoring Points

Watch for these in Render logs:

Success Indicators

✅ "Prisma schema loaded from prisma/schema.prisma"
✅ "The migrations and database are now in sync"
✅ "Build complete"
✅ "Deploy complete"

Error Indicators

❌ "Migration failed"
❌ "Database connection error"
❌ "Build failed"
❌ "Deployment failed"

📞 Support Information

Developer: DeepAgent
Date: December 30, 2025
Project: CareLinkAI - Discharge Planner Feature
Priority: High

📚 Related Documentation

- `prisma/schema.prisma` - Updated with PlacementSearch and PlacementRequest models
- `render-build.sh` - Build script with migration commands
- `.github/workflows/` - CI/CD configuration (if applicable)

Status Updated: December 30, 2025 14:35 UTC
Next Update: After Render deployment detection (5-10 minutes)