

Discharge Planner Demo Account - Implementation Summary

✓ **Task Completed: December 30, 2025**

Objective

Add a discharge planner demo account to the seed data, following the same pattern as other demo accounts (admin, operator, family, provider, caregiver).

What Was Done

1. Updated Seed Script (`prisma/seed.ts`)

- Added `upsertDischargePlannerUser()` function
- Follows same pattern as `upsertAdminUser()`
- Integrated into `main()` function
- Uses environment variables with sensible defaults

2. Account Details

- **Email:** `demo.discharge@carelinkai.com`
- **Password:** `Demo123!`
- **Role:** `DISCHARGE_PLANNER`
- **Status:** `ACTIVE`
- **Email Verified:** ✓ Yes (auto-verified)
- **User ID:** `cmjtcrcz9t000kxqkthjz4nted`

3. Database Seeding

- Executed seed script against production database
- Successfully created account
- Verified account details in database

4. Verification

- Created verification script (`verify-discharge-planner.ts`)
- Confirmed account exists with correct role
- Validated email verification status
- Tested account structure

5. Documentation

- Created credentials document: `/home/ubuntu/DISCHARGE_PLANNER_DEMO_ACCOUNT.txt`
- Includes login instructions
- Lists all account capabilities

- Documents seed script usage

Technical Implementation

Code Changes

prisma/seed.ts - New Function

```
/**
 * Ensure a development DISCHARGE_PLANNER user exists
 */
async function upsertDischargePlannerUser() {
  const email = process.env.DISCHARGE_PLANNER_EMAIL ??
'demo.discharge@carelinkai.com';
  const rawPassword = process.env.DISCHARGE_PLANNER_PASSWORD ?? 'Demo123!';

  // Hash password (bcrypt, 10 rounds)
  const passwordHash = await bcrypt.hash(rawPassword, 10);

  const dischargePlanner = await prisma.user.upsert({
    where: { email },
    update: {
      passwordHash,
      status: UserStatus.ACTIVE,
      emailVerified: new Date(),
    },
    create: {
      email,
      firstName: 'Demo',
      lastName: 'Discharge Planner',
      passwordHash,
      role: UserRole.DISCHARGE_PLANNER,
      status: UserStatus.ACTIVE,
      emailVerified: new Date(),
    },
  });

  console.log(`Discharge Planner user ready: ${dischargePlanner.email} (${discharge-
Planner.id})`);
}
```

prisma/seed.ts - Main Function Update







```
async function main() {
  console.log('Starting database seed process...');
  await seedMarketplaceTaxonomy();
  await upsertAdminUser();
  await upsertDischargePlannerUser(); // ← NEW
  await seedMockCaregivers(12);
  await seedMockFamilyJobs(15);
  await seedMockCaregiverReviews();
  console.log('Seed complete');
}
```

How to Use

Login to CareLinkAI

1. Navigate to: <https://carelinkai.com/login>
2. Enter email: `demo.discharge@carelinkai.com`
3. Enter password: `Demo123!`
4. Click "Sign In"

Available Features

-  AI-powered placement searches
-  Patient discharge planning
-  Matched home recommendations with AI scoring
-  Placement request management
-  Request tracking and status updates
-  Facility communication

Re-running the Seed

If you need to re-seed the database:

```
cd /home/ubuntu/carelinkai-project

# Using production database
DATABASE_URL="postgresql://..." npx tsx prisma/seed.ts

# Or using local database (from .env)
npx tsx prisma/seed.ts
```

Verification Results

Seed Script Output

```
Starting database seed process...
Seeding marketplace taxonomy...
Marketplace taxonomy seeded
Admin user ready: admin@carelinkai.com (cmgjlilex0000ew52dc4kliq4)
Discharge Planner user ready: demo.discharge@carelinkai.com (cmjtcruz9t000kxqkthjz4n-
ted)
...
Seed complete
```

Database Verification

```
{
  "id": "cmjtcz9t000kxqkthjz4nted",
  "email": "demo.discharge@carelinkai.com",
  "firstName": "Demo",
  "lastName": "Discharge Planner",
  "role": "DISCHARGE_PLANNER",
  "status": "ACTIVE",
  "emailVerified": "2025-12-31T01:44:14.416Z",
  "createdAt": "2025-12-31T01:44:14.417Z"
}
```

Git Commits

Commit 1: Seed Script Update

```
commit 2c635e2
Add discharge planner demo account to seed data

- Added upsertDischargePlannerUser() function to seed script
- Creates demo.discharge@carelinkai.com with role DISCHARGE_PLANNER
- Password: Demo123!
- Email pre-verified and account active
- Follows same pattern as other demo accounts
- Account successfully seeded to production database
```

Commit 2: Verification Script




```
commit 2ba0f32
Add discharge planner account verification script
```

Files Modified/Created



Modified




-  prisma/seed.ts - Added discharge planner seeding function

Created

-  verify-discharge-planner.ts - Verification script
-  /home/ubuntu/DISCHARGE_PLANNER_DEMO_ACCOUNT.txt - Credentials document
-  DISCHARGE_PLANNER_DEMO_ACCOUNT_SUMMARY.md - This summary






Security Notes

-  Password is hashed using bcrypt (10 rounds)
-  Email is pre-verified (no verification email needed)

-  Account is ACTIVE and ready to use
 -  Follows same security pattern as other demo accounts
 -  This is a DEMO account - use for testing only
-

Summary

The discharge planner demo account has been successfully:

-  Added to the seed script
-  Created in the production database
-  Verified with correct role and status
-  Documented with login credentials
-  Committed to version control

You can now log in and test the AI Discharge Planner Portal!

Next Steps

1. **Test Login:** Try logging in with the demo credentials
 2. **Test Features:** Perform a placement search
 3. **Verify Dashboard:** Check that discharge planner UI loads correctly
 4. **Test Workflow:** Send a placement request to a facility
-

Related Documentation

- Database Connection: See project context for connection string
 - Seed Script: `/home/ubuntu/carelinkai-project/prisma/seed.ts`
 - Credentials: `/home/ubuntu/DISCHARGE_PLANNER_DEMO_ACCOUNT.txt`
 - Verification: `verify-discharge-planner.ts`
-

Implementation Date: December 30, 2025

Status:  Complete and Verified