🚨 URGENT: Fix Demo Accounts on Production

Critical Issue

Status: 🔴 **BLOCKING ALL TESTING**
Priority: P0 - Immediate Action Required
Impact: Cannot verify any recent fixes or perform any testing

All demo accounts are failing authentication on production with "Invalid email or password" error.

Quick Fix Steps

Option 1: Re-run Seed Script (Recommended) ⭐

Time: ~5 minutes

1. **Access Render Shell**:
   - Go to https://dashboard.render.com
   - Select the `carelinkai` service
   - Click "Shell" tab
   - Wait for shell to connect

2. **Run Seed Script**:
   ```bash
   cd /opt/render/project/src
   npm run seed:demo
   ```

3. **Verify Accounts Created**:
   ```bash
   node -e "const { PrismaClient } = require('@prisma/client'); const prisma = new PrismaClient(); (async () => { const users = await prisma.user.findMany({ where: { email: { contains: 'demo' } }, select: { email: true, role: true, status: true } }); console.log('Demo accounts found:', users.length); users.forEach(u => console.log('  -', u.email, '/', u.role, '/', u.status)); await prisma.\$disconnect(); })();"
   ```

Expected Output:
```
Demo accounts found: 5
   - demo.admin@carelinkai.test / ADMIN / ACTIVE
   - demo.operator@carelinkai.test / OPERATOR / ACTIVE
   - demo.aide@carelinkai.test / AIDE / ACTIVE
   - demo.family@carelinkai.test / FAMILY / ACTIVE
   - demo.provider@carelinkai.test / PROVIDER / ACTIVE
```

1. **Test Login**:
   - Go to https://carelinkai.onrender.com/auth/login
   - Try: `demo.admin@carelinkai.test` / `DemoUser123!`
   - Should redirect to admin dashboard

Option 2: Manual Account Creation

Use if: Seed script fails or is not available

Time: ~2 minutes per account

```
# Create admin account
node -e "
const { PrismaClient } = require('@prisma/client');
const bcrypt = require('bcryptjs');
const prisma = new PrismaClient();

(async () => {
  try {
    const hash = await bcrypt.hash('DemoUser123!', 10);
    const user = await prisma.user.upsert({
      where: { email: 'demo.admin@carelinkai.test' },
      update: { passwordHash: hash, status: 'ACTIVE' },
      create: {
        email: 'demo.admin@carelinkai.test',
        passwordHash: hash,
        firstName: 'Admin',
        lastName: 'User',
        phone: '(555) 000-0000',
        role: 'ADMIN',
        status: 'ACTIVE',
        emailVerified: new Date()
      }
    });
    console.log('✅ Created/Updated:', user.email, '/', user.role);
  } catch (error) {
    console.error('❌ Error:', error.message);
  } finally {
    await prisma.\$disconnect();
  }
})();
"
```

Repeat for other roles:
- Replace `ADMIN` with `OPERATOR`, `AIDE`, `FAMILY`, `PROVIDER`
- Replace email accordingly

Option 3: Diagnose Existing Accounts

Use if: You want to check if accounts exist but have wrong passwords

Time: ~1 minute

```
node -e "
const { PrismaClient } = require('@prisma/client');
const bcrypt = require('bcryptjs');
const prisma = new PrismaClient();

(async () => {
  const users = await prisma.user.findMany({
    where: { email: { contains: 'demo' } },
    select: {
      email: true,
      role: true,
      status: true,
      passwordHash: true
    }
  });

  console.log('\\n📊 Demo Accounts Status\\n');
  console.log('Found', users.length, 'demo accounts:\\n');

  if (users.length === 0) {
    console.log('❌ NO DEMO ACCOUNTS FOUND - Need to run seed script\\n');
  } else {
    for (const user of users) {
      const match = await bcrypt.compare('DemoUser123!', user.passwordHash);
      console.log('Email:', user.email);
      console.log('Role:', user.role);
      console.log('Status:', user.status);
      console.log('Password Valid:', match ? '✅ YES' : '❌ NO');
      console.log('');
    }
  }

  await prisma.\$disconnect();
})();
"
```

Possible Outcomes:
- **0 accounts found**: Run seed script (Option 1)
- **Accounts found, password invalid**: Update passwords (Option 2)
- **Accounts found, password valid**: Check authentication logic

---

Verification Steps

After running any fix option:

1. Test Each Demo Account

Account	Email	Password	Expected Role
Admin	demo.admin@carelinkai.test	DemoUser123!	ADMIN
Operator	demo.operator@carelinkai.test	DemoUser123!	OPERATOR
Aide	demo.aide@carelinkai.test	DemoUser123!	AIDE
Family	demo.family@carelinkai.test	DemoUser123!	FAMILY
Provider	demo.provider@carelinkai.test	DemoUser123!	PROVIDER

Test Process:

1. Go to https://carelinkai.onrender.com/auth/login
2. Enter email and password
3. Click "Sign in"
4. ✅ Should redirect to role-appropriate dashboard
5. ❌ If fails, check error message and logs

2. Run Automated Tests

Once at least one account works:

```
# From local machine
cd /home/ubuntu/carelinkai-project
BASE_URL=https://carelinkai.onrender.com npx playwright test tests/auth.spec.ts
```

Expected: All auth tests should pass

3. Test Gallery Upload

Once family account works:

1. Login as `demo.family@carelinkai.test`
2. Navigate to Gallery tab
3. Upload a test image
4. ✅ Should succeed and show in gallery
5. ✅ Should appear in activity feed

Prevention for Future

Add Seed to Build Process

Option A: Update `package.json`

```json
{
  "scripts": {
    "build": "next build",
    "postbuild": "npm run seed:demo"
  }
}
```

Option B: Update Render Build Command

In Render dashboard:

```
npm install && npx prisma generate && npm run build && npm run seed:demo
```

Add Health Check for Demo Accounts

Create `/api/health/demo-accounts`:

```typescript
// src/app/api/health/demo-accounts/route.ts
import { NextResponse } from 'next/server';
import { PrismaClient } from '@prisma/client';

const prisma = new PrismaClient();

export async function GET() {
  try {
    const demoAccounts = await prisma.user.findMany({
      where: { email: { contains: 'demo' } },
      select: { email: true, role: true, status: true }
    });

    return NextResponse.json({
      ok: demoAccounts.length === 5,
      count: demoAccounts.length,
      accounts: demoAccounts
    });
  } catch (error) {
    return NextResponse.json({ ok: false, error: error.message }, { status: 500 });
  }
}
```

Then monitor: https://carelinkai.onrender.com/api/health/demo-accounts

---

Troubleshooting

Issue: Seed script not found

Solution: Check `package.json` for seed scripts:

```
{
  "scripts": {
    "seed:demo": "ts-node --compiler-options {\"module\":\"CommonJS\"} prisma/seed-
demo.ts"
  }
}
```

If missing, run directly:

```
npx ts-node --compiler-options '{"module":"CommonJS"}' prisma/seed-demo.ts
```

Issue: DATABASE_URL not set

Solution: Check environment variables in Render:
- Go to Render dashboard
- Select service
- Click "Environment" tab
- Verify `DATABASE_URL` is set

Issue: Prisma Client not generated

Solution: Regenerate Prisma Client:

```
npx prisma generate
```

Issue: bcrypt version mismatch

Solution: Reinstall dependencies:

```
npm install bcryptjs
```

Issue: Accounts exist but password wrong

Solution: Reset password for one account:

```
node -e "
const { PrismaClient } = require('@prisma/client');
const bcrypt = require('bcryptjs');
const prisma = new PrismaClient();

(async () => {
  const hash = await bcrypt.hash('DemoUser123!', 10);
  await prisma.user.update({
    where: { email: 'demo.admin@carelinkai.test' },
    data: { passwordHash: hash }
  });
  console.log('✅ Password reset for demo.admin@carelinkai.test');
  await prisma.\$disconnect();
})();
"
```

After Fix Checklist

- [ ] All 5 demo accounts can login
- [ ] Each account redirects to correct dashboard
- [ ] Gallery upload works (family account)
- [ ] Document upload works (family account)
- [ ] Activity feed shows uploads
- [ ] Playwright tests pass
- [ ] Comprehensive test report updated

Next Steps After Fix

1. **Immediate** (5 min):
   - Test all 5 demo accounts manually
   - Verify basic functionality

2. **Short-term** (1 hour):
   - Run full Playwright test suite
   - Test gallery and document uploads
   - Verify activity feed

3. **Medium-term** (2-3 hours):
   - Complete comprehensive testing
   - Update test report
   - Document any new issues found

4. **Long-term** (1 day):
   - Add demo account health check
   - Add seed to build process
   - Set up monitoring/alerts

Contact / Escalation

If you cannot fix this issue:

1. **Check Logs**:
   - Render deployment logs
   - Application logs
   - Database logs

2. **Verify Environment**:
   - DATABASE_URL correct
   - Node.js version: 18+
   - Prisma version: 6.7.0

3. **Get Help**:
   - Check Render documentation

- Review Prisma documentation
- Contact DevOps team

Summary

Problem: Demo accounts not working on production
Impact: Blocks all testing
Solution: Re-run seed script or create accounts manually
Time: 5-10 minutes
Priority: URGENT - Do this first before any other work

Quick Command (if you have Render shell access):

```
cd /opt/render/project/src && npm run seed:demo
```

Then test: https://carelinkai.onrender.com/auth/login with `demo.admin@carelinkai.test` / `DemoUser-123!`

Document Created: December 14, 2025
Status: URGENT - ACTION REQUIRED
Next Update: After demo accounts are fixed