

# Prisma Error Fix: primaryContactEmail Field

## Issue Summary

**Error:** Unknown argument 'primaryContactEmail' in prisma.family.findFirst()

**Timestamp:** 2025-12-17T06:23:58Z

**Context:** Error occurred during user login (demo.family@carelinkai.test)

**Impact:** Prevented Family users from viewing their inquiries

## Root Cause

The code in `/src/app/api/operator/inquiries/route.ts` was attempting to query the `Family` model using a non-existent field `primaryContactEmail`.

**Incorrect Code (Line 66):**

```
const family = await prisma.family.findFirst({
  where: {
    OR: [
      { primaryContactEmail: user.email },
      { secondaryContactEmail: user.email },
    ],
  },
});
```

## Schema Analysis

Looking at `prisma/schema.prisma`, the `Family` model (lines 242-272) does NOT have `primaryContactEmail` or `secondaryContactEmail` fields. Instead, it has:

- `userId` (String @unique) - Foreign key to User model
- `primaryContactName` (String?) - Name, not email
- `phone` (String?)
- `relationshipToRecipient` (String?)

The email is stored in the related `User` model, accessible via the `user` relation.

## Solution

Changed the query to use the correct `userId` field to match the Family record with the authenticated user:

**Correct Code:**

```
const family = await prisma.family.findFirst({
  where: {
    userId: user.id,
  },
});
```

## Files Modified

---

1. `/src/app/api/operator/inquiries/route.ts`
  - Line 66: Fixed Prisma query to use `userId` instead of `primaryContactEmail`
  - Simplified the query since Family-User relationship is 1:1 via `userId`
2. `/ROLE_BASED_INQUIRIES_IMPLEMENTATION.md`
  - Updated documentation to reflect correct implementation
  - Fixed code examples in Section 1

## Verification

---

✅ **Build Status:** Compiled successfully with no errors

```
npm run build
# Result: Build completed with only pre-existing warnings
```

✅ **Git Status:**

- Commit: `1fb7c8a`
- Pushed to: `origin/main`
- Repository: `https://github.com/profyt7/carelinkai`

## Impact Assessment

---

- **Severity:** HIGH - Blocked Family user functionality
- **Users Affected:** All Family role users attempting to view inquiries
- **Fix Type:** Code correction (no schema changes required)
- **Deployment:** Automatic via Render on push to main

## Testing Recommendations

---

After deployment, verify:

1. ✅ Family users can log in successfully
2. ✅ Family users can view their inquiries at `/operator/inquiries`
3. ✅ Operator/Admin users can still view all inquiries
4. ✅ No Prisma errors in production logs

## Related Schema Reference

```
model Family {
  id          String @id @default(cuid())
  userId      String @unique           // ← Correct field to use
  emergencyContact String?
  emergencyPhone String?

  primaryContactName String?           // ← Name, not email
  phone              String?
  relationshipToRecipient String?






  // Relationships
  user User @relation(fields: [userId], references: [id], onDelete: Cascade)
  inquiries Inquiry[]
  // ... other relations
}
```

## Prevention

To prevent similar issues:

1. Always reference Prisma schema before writing queries
2. Use TypeScript types generated by Prisma for autocomplete
3. Test with actual data after schema changes
4. Review error logs regularly for Prisma-related issues

## Deployment Status

-  Code fixed
-  Build verified
-  Committed to Git
-  Pushed to GitHub (main branch)
-  Render auto-deployment in progress

**Fixed By:** AI Assistant

**Date:** December 17, 2025

**Priority:** HIGH

**Status:**  RESOLVED