Gallery Upload Fix Summary

Issue

Gallery photos uploaded successfully to Cloudinary, but users saw an error message. The backend failed with:

```
Invalid `prisma.galleryPhoto.create()` invocation:
Argument `fileUrl` is missing.
```

Root Cause

The API code was using incorrect field names that didn't match the Prisma schema:

Code was using:
- `familyId` → ❌ GalleryPhoto doesn't have this field
- `uploadedById` → ❌ Should be `uploaderId`
- `cloudinaryUrl` → ❌ Should be `fileUrl`
- `albumId` → ❌ Should be `galleryId`

Schema expects:

```
model GalleryPhoto {
  galleryId    String  // ✅ Required
  uploaderId   String  // ✅ Required
  fileUrl      String  // ✅ Required (was missing!)
  thumbnailUrl String  // ✅ Was correct
  ...
}
```

Architecture Understanding

The correct data model is:

```
Family → SharedGallery → GalleryPhoto
```

Not directly: ~~Family → GalleryPhoto~~

Solution Implemented

1. Get or Create SharedGallery

```
// Get or create default "Family Photos" gallery
let gallery = await prisma.sharedGallery.findFirst({
  where: { familyId, title: 'Family Photos' }
});

if (!gallery) {
  gallery = await prisma.sharedGallery.create({
    data: {
      familyId,
      creatorId: session.user.id,
      title: 'Family Photos',
      description: 'Default photo gallery for family',
    },
  });
}
```

2. Use Correct Field Names

```
const photo = await prisma.galleryPhoto.create({
  data: {
    galleryId: gallery.id,        // ✅ Correct
    uploaderId: session.user.id,  // ✅ Correct
    fileUrl: uploadResult.secure_url, // ✅ Correct
    thumbnailUrl,
    caption: caption || file.name,
    metadata: {
      cloudinaryPublicId: uploadResult.public_id,
      fileType: file.type,
      fileSize: file.size,
      originalFilename: file.name,
    },
  },
  include: {
    uploader: { ... },  // ✅ Correct relation name
    gallery: { ... },   // ✅ Correct relation name
  },
});
```

3. Fixed References

- Activity feed now uses `photo.gallery.familyId`
- SSE event publishes to `family:${photo.gallery.familyId}`
- Metadata includes Cloudinary info in JSON field

Files Modified

- `src/app/api/family/gallery/upload/route.ts`

Testing Steps

1. Access Family Gallery

```
https://carelinkai.onrender.com/dashboard/gallery
```

2. Upload a Photo

- Click “Upload Photos” button
- Select an image file
- Add a caption (optional)
- Click “Upload”

3. Expected Result

✅ Upload succeeds without error
✅ Photo appears in gallery immediately
✅ Activity feed shows “uploaded a photo”
✅ Photo is stored in Cloudinary
✅ Database has correct GalleryPhoto record

4. Verify Database

```sql
SELECT
  gp.id,
  gp.fileUrl,
  gp.galleryId,
  gp.uploaderId,
  sg.title as gallery_title,
  sg.familyId
FROM "GalleryPhoto" gp
JOIN "SharedGallery" sg ON gp."galleryId" = sg.id
ORDER BY gp."createdAt" DESC
LIMIT 5;
```

Should show:
- ✅ `fileUrl` populated with Cloudinary URL
- ✅ `galleryId` links to SharedGallery
- ✅ `uploaderId` links to User
- ✅ Gallery belongs to Family

Deployment

- **Commit**: `4404093`
- **Branch**: `main`
- **Status**: Pushed to GitHub ✅
- **Auto-Deploy**: Render will deploy automatically

Success Criteria

- [x] Logs analyzed and root cause identified
- [x] Fix implemented with correct schema field names

- [x] Build succeeds locally
- [x] Changes committed and pushed to GitHub
- [ ] Render deployment completes successfully
- [ ] Gallery upload works without errors in production
- [ ] Photos appear in gallery after upload

Technical Details

Schema Relationships

```
Family (id) → SharedGallery (familyId)
SharedGallery (id) → GalleryPhoto (galleryId)
User (id) → GalleryPhoto (uploaderId)
```

Metadata Storage

File metadata is now stored in the JSON `metadata` field:

```json
{
  "cloudinaryPublicId": "carelinkai/family/.../...",
  "fileType": "image/jpeg",
  "fileSize": 306414,
  "originalFilename": "IMG_0866.jpg"
}
```

Default Gallery

When no album/gallery is specified, photos go to a default “Family Photos” SharedGallery that’s automatically created per family.

Rollback Instructions

If issues occur, revert commit:

```
git revert 4404093
git push origin main
```

Next Steps

1. Monitor Render deployment logs
2. Test gallery upload in production
3. Verify photos display correctly
4. Check activity feed updates
5. Confirm no new errors in logs

Fixed: December 14, 2025
Engineer: DeepAgent via Abacus.AI