

Family Portal Mock Mode Fix - Summary

Issue

Documents, Activity Feed, and Gallery tabs were showing errors in production because they were potentially using mock mode with `'demo-family'` as the familyId, while Notes and Emergency tabs were working correctly with the real familyId.

Root Cause

The issue was an inconsistency in how tabs received the `showMock` prop from the parent `page.tsx`:

Working Tabs (Notes & Emergency):

- **Did NOT** receive `showMock` prop from parent
- Defaulted to `showMock = false` in component props
- Always used real `familyId` from API

Failing Tabs (Documents, Timeline, Gallery, Billing, Members):

- **Received** `showMock={showMock}` prop from parent
- If `showMock` was somehow `true`, they would use mock data with `'demo-family'`
- This caused API errors: [NOTES] Family `demo-family` not found

Solution

Removed the `showMock` prop from ALL tabs to ensure consistency:

```

// BEFORE (inconsistent):
{activeTab === 'documents' && (
  <DocumentsTab
    familyId={familyId}
    showMock={showMock} // ← Removed this
    onUploadClick={...}
  />
)}

{activeTab === 'notes' && (
  <NotesTab
    familyId={familyId}
    // No showMock prop
  />
)}

// AFTER (consistent):
{activeTab === 'documents' && (
  <DocumentsTab
    familyId={familyId}
    onUploadClick={...}
  />
)}

{activeTab === 'notes' && (
  <NotesTab
    familyId={familyId}
  />
)}

```

Changes Made

Modified Files:

- `/src/app/family/page.tsx`

Specific Changes:

1. Removed `showMock={showMock}` from `DocumentsTab`
2. Removed `showMock={showMock}` from `TimelineTab`
3. Removed `showMock={showMock}` from `GalleryTab`
4. Removed `showMock={showMock}` from `BillingTab`
5. Removed `showMock={showMock}` from `MembersTab`

Verification

Build Status:

`npm run build` completed successfully

Expected Behavior After Fix:

- All tabs now default to `showMock = false`
- All tabs use the real `familyId` fetched from `/api/family/membership`
- No more `demo-family` errors in production logs
- Documents tab loads real documents

- Activity Feed tab loads real activity
- Gallery tab loads real photos

Testing Checklist

After deployment to Render:

- [] Navigate to Family Portal (<https://carelinkai.onrender.com/family>)
- [] Click on **Documents** tab → Should show real documents (not error)
- [] Click on **Activity** tab → Should show real activity feed (not error)
- [] Click on **Gallery** tab → Should show real photos (not error)
- [] Click on **Notes** tab → Should still work (no regression)
- [] Click on **Emergency** tab → Should still work (no regression)
- [] Check browser console → No errors about `demo-family` not found

Deployment Info

Commit: 6977a61

Branch: main

Status: Pushed to GitHub

Render: Auto-deploy triggered

Technical Notes

Why This Fix Works:

1. All tab components have `showMock` parameter with default value `false` :


```
typescript
interface DocumentsTabProps {
  familyId: string | null;
  showMock?: boolean; // Defaults to false
}
```
2. When `showMock` is not passed, it defaults to `false`
3. When `showMock` is `false`, tabs always use the real `familyId` prop
4. The real `familyId` is fetched from `/api/family/membership` and passed to all tabs

Mock Mode Control:

Mock mode is still controlled in the parent `page.tsx` via:

- Environment variable check (disabled in production)
- `/api/runtime/mock` endpoint (respects production flag)

However, since NO tabs receive the `showMock` prop anymore, even if `showMock` state becomes `true` in development, tabs will ignore it and use real data.

Related Files

- `/src/app/family/page.tsx` - Main Family Portal page
- `/src/components/family/DocumentsTab.tsx`
- `/src/components/family/TimelineTab.tsx`

- `/src/components/family/GalleryTab.tsx`
- `/src/components/family/NotesTab.tsx`
- `/src/components/family/EmergencyTab.tsx`
- `/src/components/family/BillingTab.tsx`
- `/src/components/family/MembersTab.tsx`

Success Criteria

- All tabs use consistent approach (no `showMock` prop)
- Build succeeds without TypeScript errors
- Changes committed and pushed to GitHub
- Render deployment triggered

Next Steps

1. Monitor Render deployment logs
2. Test all tabs in production after deployment
3. Verify no `demo-family` errors in Render logs
4. Confirm all tabs load real data successfully