

Reports Module - Fix Summary

Date: December 12, 2025

Commit: 6a89201

Status:  Ready for Deployment

Issues Fixed

Issue 1: Missing Sidebar Navigation

Problem: Reports pages were not wrapped in DashboardLayout component, causing them to render without the sidebar navigation.

Files Modified:

- Created `/src/app/reports/layout.tsx` - Layout wrapper for all reports pages
- Modified `/src/app/reports/page.tsx` - Removed standalone wrapper
- Modified `/src/app/reports/history/page.tsx` - Removed standalone wrapper
- Modified `/src/app/reports/scheduled/page.tsx` - Removed standalone wrapper

Changes:

1. Created `ReportsLayout` component that wraps all reports pages in `DashboardLayout`
2. Removed `min-h-screen bg-gradient-to-br from-slate-50 via-blue-50 to-indigo-50 p-4 md:p-8` wrapper divs
3. Removed `max-w-7xl mx-auto` wrapper divs
4. Simplified page structure to work with layout system

Result: Reports pages now show the standard sidebar navigation and match the layout of other dashboard pages.

Issue 2: Database Migration for Report Tables

Problem: Report and ScheduledReport tables don't exist in the database, causing report generation API to fail.

Files Created:

- `/prisma/migrations/20251212000000_add_reports_and_scheduled_reports/migration.sql`

Migration Contents:

1. New Enums:

- `ReportType` : OCCUPANCY, FINANCIAL, INCIDENT, CAREGIVER, COMPLIANCE, INQUIRY, RESIDENT, FACILITY_COMPARISON, CUSTOM
- `ReportFormat` : PDF, EXCEL, CSV
- `ReportSchedule` : DAILY, WEEKLY, MONTHLY

1. AuditAction Enum Updates:

- Added `REPORT_GENERATED`

- Added `REPORT_SCHEDULED`
- Added `REPORT_DOWNLOADED`

2. New Tables:

- `Report` : Stores generated report metadata
 - Fields: id, title, type, format, fileUrl, config (JSON), generatedBy, createdAt, updatedAt
 - Indexes: generatedBy, type, createdAt
 - Foreign key to User table
- `ScheduledReport` : Stores scheduled report configurations
 - Fields: id, title, type, format, schedule, dayOfWeek, dayOfMonth, time, recipients (array), config (JSON), enabled, lastRun, nextRun, createdBy, createdAt, updatedAt
 - Indexes: createdBy, enabled, nextRun, type
 - Foreign key to User table

Migration Safety:

- Uses idempotent SQL with `IF NOT EXISTS` checks
 - Can be run multiple times without errors
 - Handles partial application scenarios gracefully
-

Deployment Steps

1. Automatic Deployment (Render)

When pushed to GitHub, Render will automatically:

1. Pull the latest code
2. Run `npm install`
3. Run `npm run build`
4. **Run database migrations** (via build command: `prisma migrate deploy`)

2. Verify Deployment

After deployment completes on Render:

1. Check Build Logs:

- ✓ Prisma migrations applied successfully
- ✓ Prisma Client generated
- ✓ Next.js build completed

2. Test Reports Pages:

- Visit <https://carelinkai.onrender.com/reports>
- Verify sidebar navigation is visible
- Verify page layout matches other dashboard pages

3. Test Report Generation:

- Click “Generate Report” button
- Fill in the form (Occupancy Report, Last 30 Days, PDF)
- Click “Generate Report”
- Should see success message “Report generated successfully”
- Should see the new report in “Recent Reports” section

4. Check Database:

```
```sql
- Verify Report table exists
SELECT * FROM "Report" LIMIT 1;

- Verify ScheduledReport table exists
SELECT * FROM "ScheduledReport" LIMIT 1;

- Verify enums exist
SELECT enumlabel FROM pg_enum WHERE enumtypid =
(SELECT oid FROM pg_type WHERE typname = 'ReportType');
```

```

Verification Checklist

Layout Fixes

- [] Reports page (/reports) shows sidebar
- [] Reports History page (/reports/history) shows sidebar
- [] Scheduled Reports page (/reports/scheduled) shows sidebar
- [] Page styling is consistent with other dashboard pages
- [] Navigation links work correctly

Database & API

- [] Migration applied successfully (check Render logs)
- [] Report table exists in database
- [] ScheduledReport table exists in database
- [] Enums created successfully
- [] Foreign key constraints in place
- [] Report generation API works (POST /api/reports/generate)
- [] Reports list API works (GET /api/reports)
- [] Scheduled reports API works (GET /api/reports/scheduled)

Functionality

- [] Can generate Occupancy Report
- [] Can generate Financial Report
- [] Can generate Incident Report
- [] Generated reports appear in Recent Reports
- [] Can view Report History
- [] Can filter reports by type and format
- [] Can delete reports
- [] Scheduled reports can be toggled on/off

Rollback Instructions

If issues occur, rollback to previous commit:

```
# On Render (via dashboard or manual deployment):
git revert 6a89201
git push origin main

# Or deploy specific commit:
# In Render dashboard: Manual Deploy → Select commit ca49a9b
```

Note: Rolling back will:

1. Restore standalone page layouts (no sidebar)
 2. Migration will remain in database (safe - tables won't be dropped)
 3. Report generation will still fail until migration is applied
-

Technical Details

Migration Strategy

- **Idempotent Design:** Uses `IF NOT EXISTS` and exception handling
- **Safe Enum Updates:** Checks for existing values before adding
- **Foreign Key Constraints:** Ensures data integrity with User table
- **Indexes:** Optimized for common queries (generatedBy, type, createdAt, etc.)

Layout System

- Uses Next.js App Router layout system
- `layout.tsx` wraps all child pages
- Inherits from `DashboardLayout` component
- Provides consistent navigation across reports section

API Endpoints

- `POST /api/reports/generate` - Generate new report
 - `GET /api/reports` - List reports (with pagination)
 - `GET /api/reports/scheduled` - List scheduled reports
 - `DELETE /api/reports/{id}` - Delete report
 - `PUT /api/reports/scheduled/{id}` - Update scheduled report
-

Known Limitations

1. **File Storage:** Report `fileUrl` is currently optional - actual file generation not implemented yet
2. **Export Formats:** Excel and CSV export not implemented yet (only PDF placeholders)
3. **Email Delivery:** Scheduled report email delivery not implemented yet
4. **Report Preview:** Preview functionality shows placeholder message

These are expected and will be implemented in future phases.

Success Criteria

- All reports pages show sidebar navigation
 - Database migration runs successfully on deployment
 - Report generation API works without errors
 - Reports can be viewed in history
 - No TypeScript or build errors
 - No console errors on reports pages
-

Next Steps (Future Enhancements)

1. Implement actual report file generation (PDF export)
 2. Add Excel and CSV export functionality
 3. Implement scheduled report email delivery
 4. Add report preview functionality
 5. Add report download functionality
 6. Implement report sharing features
-

Support

Repository: <https://github.com/profy7/carelinkai>

Deployment: <https://carelinkai.onrender.com>

Commit: 6a89201

If deployment issues occur:

1. Check Render build logs
2. Verify database migration applied
3. Check for any database connection issues
4. Verify Prisma Client generation
5. Contact support with commit hash and error logs