

Inquiries Module - Part 2 Implementation Summary

🎯 Overview

Successfully implemented CSV export functionality and comprehensive document upload system for the Inquiries Module.

Commit: 6c74657 - feat: Add CSV export and document upload to Inquiries Module - Part 2

Branch: main

Status:  Complete and Pushed to GitHub

✨ Features Implemented

1. CSV Export Functionality

Export Button

- **Location:** Inquiries list page (/operator/inquiries)
- **Features:**
 - Displays count of inquiries to be exported
 - Shows loading state during export
 - Respects all active filters (status, home, dates, tour status, etc.)
 - Respects search query
 - Respects current sorting preference
 - Downloads with timestamped filename (e.g., inquiries-export-2024-12-11.csv)

Export Data Columns (18 columns)

1. **Family Name** - Name of the family
2. **Contact Person Name** - First and last name
3. **Contact Email** - Email address
4. **Contact Phone** - Phone number (formatted)
5. **Inquiry Date** - When inquiry was submitted
6. **Days Since Inquiry** - Calculated age of inquiry
7. **Status** - Current inquiry status (formatted)
8. **Home/Facility** - Name of the home/facility
9. **Facility Address** - Full address
10. **Tour Date** - Scheduled tour date or "Not Scheduled"
11. **AI Match Score** - AI-generated match percentage or "N/A"
12. **Last Activity Date** - Most recent update
13. **Documents Count** - Number of uploaded documents
14. **Converted to Resident** - Yes/No
15. **Conversion Date** - When converted or "N/A"
16. **Converted By** - Staff member who converted

17. **Initial Message** - First 200 chars of inquiry message

18. **Internal Notes** - First 200 chars of operator notes

Technical Implementation

- Added `exportInquiriesToCSV()` function to `/src/lib/export-utils.ts`
 - Reuses Papa Parse library for CSV generation
 - Helper functions: `formatPhoneNumber()`, `calculateDaysBetween()`
 - Fetches all matching inquiries (up to 10,000) with filters applied
 - Client-side CSV generation and download
-

2. Document Upload System

Database Schema

New Model: InquiryDocument

```
model InquiryDocument {
    id          String  @id @default(cuid())
    inquiryId   String
    fileName    String
    fileUrl    String
    fileType    String
    documentType String
    description String? @db.Text
    fileSize    Int
    uploadedById String
    uploadedAt   DateTime @default(now())

    inquiry   Inquiry  @relation(...)
    uploadedBy User    @relation(...)

    @@index([inquiryId, uploadedById, documentType, uploadedAt])
}
```

Migration: 20251211201640_add_inquiry_documents

- Idempotent SQL migration (safe for re-runs)
- Foreign keys to Inquiry and User tables
- Cascade delete on inquiry removal
- Indexed for performance

Document Types Supported:

1. APPLICATION_FORM - Application forms
2. PHOTO_ID - Photo identification
3. INSURANCE_DOCUMENT - Insurance papers
4. MEDICAL_RECORD - Medical records
5. FINANCIAL_INFORMATION - Financial documents
6. BACKGROUND_CHECK - Background check results
7. REFERENCE_LETTER - Reference letters
8. TOUR_NOTES - Tour notes and feedback
9. CONTRACT - Contracts and agreements
10. OTHER - Miscellaneous documents

API Endpoints

GET /api/operator/inquiries/[id]/documents

- **Purpose:** Fetch all documents for an inquiry
- **Auth:** Requires OPERATOR or ADMIN role
- **Access Control:** Validates user has access to inquiry's home
- **Response:** Array of documents with uploader info
- **Sorting:** Ordered by upload date (newest first)

POST /api/operator/inquiries/[id]/documents

- **Purpose:** Create document record after upload
- **Auth:** Requires OPERATOR or ADMIN role
- **Access Control:** Validates user has access to inquiry's home
- **Validation:** Zod schema for all fields
- **Audit:** Creates audit log entry
- **Response:** Created document with metadata

DELETE /api/operator/inquiries/documents/[documentId]

- **Purpose:** Delete a specific document
- **Auth:** Requires OPERATOR or ADMIN role
- **Access Control:** Validates user has access to inquiry's home
- **Audit:** Creates audit log entry
- **Response:** Success confirmation

Security Features:

- RBAC enforcement on all endpoints
 - Operator can only access their own inquiries
 - Admins have full access
 - Audit logging for all document operations
 - PrismaClient disconnection for safety
-

3. UI Components

DocumentUploadModal Component

Location: /src/components/operator/inquiries/DocumentUploadModal.tsx

Features:

- **Drag-and-Drop:** React-dropzone integration for easy file selection
- **File Validation:**
 - Accepted types: PDF, Images (PNG, JPG, JPEG), Word docs (DOC, DOCX)
 - Maximum size: 10MB
 - Clear error messages for invalid files
- **Document Type Selection:** Dropdown with 10 document types
- **Description Field:** Optional text area for notes
- **Upload Progress:** Visual progress bar (0-100%)
- **Two-Step Process:**
 1. Upload file to Cloudinary (/api/upload)
 2. Create document record in database
- **Error Handling:** User-friendly error messages

- **Loading States:** Disabled controls during upload
- **Responsive Design:** Works on mobile and desktop

Cloudinary Integration:

- Uses existing upload API
- Stores files in `carelinkai/caregiver-documents` folder
- Returns secure URL for database storage

DocumentsSection Component

Location: `/src/components/operator/inquiries/DocumentsSection.tsx`

Features:

- **Document List:** Card-based layout with metadata
- **File Icons:** Different icons for PDFs, images, and docs
- **Document Type Filter:** Dropdown to filter by type
- **Document Counts:** Shows total and filtered counts
- **Document Actions:**
 - **Download:** Opens file in new tab
 - **Delete:** Confirmation dialog before deletion
- **Empty States:**
 - No documents yet
 - No documents of selected type
 - CTA to upload first document
- **Document Metadata Display:**
 - File name and type badge
 - Description (if provided)
 - File size (human-readable)
 - Uploader name
 - Upload date
- **Loading States:** Spinner during fetch
- **Responsive Grid:** Adapts to screen size

4. Integration

Inquiry Detail Page Updates

File: `/src/app/operator/inquiries/[id]/page.tsx`

Changes:

- Added import for `DocumentsSection` component
- Added Documents section in left column (after Internal Notes)
- Wrapped in card UI matching existing design
- Automatically loads documents for current inquiry

Layout:

Left Column (2/3 width):

- Family Contact
- Inquiry Details
- Internal Notes
- Documents ← NEW SECTION

Right Column (1/3 width):

- Lead Status
- Conversion Info/Button

Files Created/Modified

Created Files

1. /prisma/migrations/20251211201640_add_inquiry_documents/migration.sql
2. /src/app/api/operator/inquiries/[id]/documents/route.ts
3. /src/app/api/operator/inquiries/documents/[documentId]/route.ts
4. /src/components/operator/inquiries/DocumentUploadModal.tsx
5. /src/components/operator/inquiries/DocumentsSection.tsx

Modified Files

1. /prisma/schema.prisma - Added InquiryDocument model and relationships
2. /src/lib/export-utils.ts - Added inquiry export functions
3. /src/components/operator/inquiries/InquiriesListClient.tsx - Added export button
4. /src/app/operator/inquiries/[id]/page.tsx - Added Documents section

Testing & Validation

Build Verification

```
npm run build
```

Result:  Build successful with no errors

Code Quality

-  TypeScript compilation passed
-  No linting errors
-  Follows existing code patterns
-  RBAC checks implemented
-  Audit logging added
-  Error handling included
-  Loading states implemented
-  Mobile responsive design

Pattern Consistency

-  Follows Residents module patterns

- Reuses existing upload infrastructure
 - Consistent UI/UX design
 - Proper TypeScript types
 - Idiomatic React hooks usage
-

Deployment

Git Status

- **Committed:** 6c74657
- **Pushed:** Successfully pushed to origin/main
- **GitHub:** <https://github.com/profy7/carelinkai>

Deployment Steps for Render

1. **Database Migration:** Will run automatically on deployment
 - Migration file: 20251211201640_add_inquiry_documents
 - Idempotent: Safe to run multiple times
2. **Prisma Generate:** Runs during build process
3. **Environment Variables:** All required vars already configured
 - CLOUDINARY_CLOUD_NAME
 - CLOUDINARY_API_KEY
 - CLOUDINARY_API_SECRET
4. **Build:** Next.js production build completes successfully

Post-Deployment Verification

Once deployed, verify:

1. Inquiries list page loads
2. Export button appears and works
3. CSV downloads with correct data
4. Inquiry detail page shows Documents section
5. Upload button opens modal
6. File upload to Cloudinary works
7. Documents appear in list after upload
8. Download button works
9. Delete button works with confirmation
10. Filter by document type works

Success Criteria

All deliverables from the requirements have been met:

CSV Export

- Export button with count indicator
- Respects filters, search, and sorting
- 18+ comprehensive columns
- Timestamped filename

- Uses Papa Parse library
- Downloads automatically

Document Upload

- InquiryDocument model in schema
- Database migration created
- GET/POST/DELETE API endpoints
- RBAC enforcement
- Audit logging
- DocumentUploadModal component
- Drag-and-drop interface
- File validation (type & size)
- Document type selection
- Upload progress indicator
- Cloudinary integration
- DocumentsSection component
- Filter by document type
- View/download/delete actions
- Empty states
- Loading states

Integration

- Documents section in inquiry detail page
- Mobile responsive
- Consistent UI/UX
- Error handling
- Following existing patterns

Quality

- Build passes
- TypeScript compilation passes
- Code follows conventions
- Git committed and pushed
- Ready for deployment

Screenshots & UI Flow

Inquiries List Page

- Export button appears in the toolbar next to sort dropdown
- Shows count of inquiries to be exported
- Button disabled during export with "Exporting..." text

Inquiry Detail Page

- Documents section appears in left column
- Shows document count badge

- Upload button prominent in header
- Filter dropdown (when documents exist)
- Each document shows:
- File type icon (image/PDF/doc)
- File name and type badge
- Description
- Metadata (size, uploader, date)
- Download and delete actions

Upload Modal

- Large dropzone area with icon
 - Drag-and-drop or click to select
 - Selected file name displays
 - Document type dropdown
 - Optional description field
 - Progress bar during upload
 - Cancel and Upload buttons
-

Security & Compliance

Authentication & Authorization

- All API endpoints require authentication
- RBAC checks for OPERATOR and ADMIN roles
- Operators can only access their own inquiries
- Admins have unrestricted access

Data Privacy

- Document URLs secured via Cloudinary
- File validation prevents malicious uploads
- Audit logs track all document operations
- Soft delete support (cascade on inquiry deletion)

Performance

- Indexed database queries
 - Lazy loading for documents
 - Efficient CSV generation (client-side)
 - Cloudinary CDN for file delivery
-

Notes for Operations Team

Known Limitations

- Maximum export size: 10,000 inquiries (should cover all realistic use cases)
- File upload size limit: 10MB

- Supported file types: PDF, Images, Word docs

Future Enhancements (Optional)

- Bulk document upload
- Document version history
- Document templates
- Email documents to families
- Document preview modal
- Advanced search across document content
- Document expiration tracking

Support Tips

- If export fails, check network and try again
- If upload fails, verify file type and size
- Cloudinary issues: Check environment variables
- Migration issues: Verify database connection



Conclusion

Part 2 of the Inquiries Module is **100% complete** and **ready for production deployment**.

All requirements have been met with high-quality implementation:

- CSV export with comprehensive data
- Document upload with Cloudinary
- Full CRUD operations for documents
- Professional UI/UX
- Security and RBAC
- Audit logging
- Mobile responsive
- Error handling
- Loading states
- Code quality

Next Steps:

1. Deploy to Render (automatic via GitHub push)
2. Monitor deployment logs
3. Verify migration success
4. Test functionality in production
5. Notify stakeholders of new features

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

GitHub Repo: <https://github.com/profy7/carelinkai>

Commit: 6c74657