

Audit Logs Viewer Implementation - Complete

Overview

Successfully implemented the **Audit Logs Viewer** feature (Critical Priority #1) for HIPAA compliance and security auditing. This feature was cleanly extracted from the backup branch without any problematic Bugsnag code.

Implementation Summary

Feature Branch

- **Branch:** feature/audit-logs-viewer
- **Merged to:** main
- **Commit:** b36fbdc
- **Status:**  Pushed to GitHub, Render deployment triggered

Files Added (4 files, 819 lines)

1. Frontend Page

File: src/app/admin/audit-logs/page.tsx (509 lines)

- **Purpose:** Admin interface for viewing and managing audit logs
- **Features:**
 - Real-time log viewing with pagination (50 logs per page)
 - Advanced filtering by:
 - Action type (LOGIN, LOGOUT, CREATE, UPDATE, DELETE, etc.)
 - User ID
 - Resource type
 - Date range
 - Free-text search
 - Statistics dashboard showing action counts
 - Individual log detail modal with full metadata
 - CSV export functionality
 - Responsive design with Tailwind CSS
 - Admin-only access (automatic redirect for non-admins)

2. Main API Route

File: src/app/api/admin/audit-logs/route.ts (125 lines)

- **Endpoint:** GET /api/admin/audit-logs

- **Features:**

- Paginated log retrieval
- Query parameters:
 - `page` : Page number (default: 1)
 - `limit` : Items per page (default: 50)
 - `action` : Filter by action type
 - `userId` : Filter by user ID

- `resourceType` : Filter by resource type
- `startDate` : Filter by start date
- `endDate` : Filter by end date
- `search` : Free-text search
- Returns: logs, pagination metadata, action statistics
- Authorization: Admin-only (403 for non-admins)

3. Individual Log API Route

File: `src/app/api/admin/audit-logs/[id]/route.ts` (59 lines)

- **Endpoint:** `GET /api/admin/audit-logs/[id]`

- **Features:**

- Fetches individual log details
- Includes user and `actionedBy` user information
- Returns 404 for non-existent logs
- Authorization: Admin-only

4. Export API Route

File: `src/app/api/admin/audit-logs/export/route.ts` (126 lines)

- **Endpoint:** `GET /api/admin/audit-logs/export`

- **Features:**

- Exports audit logs to CSV format
- Supports same filtering as main API route
- Creates audit log entry for export action
- Returns downloadable CSV file
- Filename format: `audit_logs_YYYY-MM-DD.csv`
- Authorization: Admin-only



Security & Authorization

All endpoints enforce **admin-only access**:

```
if (!session?.user || session.user.role !== 'ADMIN') {
  return NextResponse.json({ error: 'Unauthorized' }, { status: 403 });
}
```



Database Schema

Uses existing `AuditLog` model from `prisma/schema.prisma` :

```

model AuditLog {
    id          String      @id @default(cuid())
    userId      String
    actionedBy  String?
    action     AuditAction
    resourceType String
    resourceId   String?
    description  String
    ipAddress   String?
    userAgent   String?
    metadata    Json?
    createdAt   DateTime  @default(now())
    actionedByUser User?  @relation("ActionedByUser", fields: [actionedBy], refer-
ences: [id])
    user       User      @relation("UserAuditLogs", fields: [userId], references:
[id])
    @@index([userId])
    @@index([actionedBy])
    @@index([action])
    @@index([resourceType, resourceId])
    @@index([createdAt])
}

```

No database migrations needed - the AuditLog model already exists in production.

Build Verification

 **Build Status:** Success (no errors)

Route (app)	Size	First Load
JS		
...		
 λ /admin/audit-logs	24.1 kB	209 k
B		
 λ /api/admin/audit-logs	0 B	0
B		
 λ /api/admin/audit-logs/[id]	0 B	0
B		
 λ /api/admin/audit-logs/export	0 B	0
B		
...		

How to Access

Production URL

URL: <https://getcarelinkai.com/admin/audit-logs>

Access Requirements

- Must be logged in as an **ADMIN** user
- Non-admin users will be automatically redirected

Admin Test Account

You can use your admin account credentials to access the feature.

Verification Checklist

Render Deployment

1.  GitHub push successful (b36fbdc)
2.  Render auto-deploy triggered (monitor at dashboard)
3.  Build should complete successfully
4.  Service should restart automatically

Post-Deployment Testing

Once Render deployment completes, verify the following:

1. Page Access

- [] Navigate to `https://getcarelinkai.com/admin/audit-logs`
- [] Page loads without errors
- [] Sidebar navigation shows “Audit Logs” link
- [] Non-admin users are redirected

2. Log Viewing

- [] Audit logs are displayed in a table
- [] Pagination controls work
- [] Statistics dashboard shows action counts
- [] Individual log details can be viewed

3. Filtering

- [] Filter by action type works
- [] Filter by user works
- [] Filter by resource type works
- [] Date range filtering works
- [] Search functionality works
- [] Multiple filters can be combined

4. Export

- [] Export button is visible
- [] CSV export downloads successfully
- [] CSV contains correct data
- [] Export action is logged in audit logs

5. API Endpoints

- [] GET `/api/admin/audit-logs` returns data
- [] GET `/api/admin/audit-logs/[id]` returns individual log
- [] GET `/api/admin/audit-logs/export` downloads CSV
- [] Non-admin requests return 403

Monitoring

Render Logs

Monitor for any errors related to:

- `/admin/audit-logs` page rendering

- `/api/admin/audit-logs` API calls
- Database queries on `AuditLog` table

Expected Behavior

- Fast page load (<2s)
- Smooth filtering and pagination
- CSV export completes in <5s for typical datasets

Troubleshooting

Issue: Page Not Loading

Solution: Check Render deployment status and logs

Issue: No Logs Displayed

Solution:

- Verify database has audit log entries
- Check API response in browser dev tools
- Verify admin authentication

Issue: Export Not Working

Solution:

- Check browser console for errors
- Verify API endpoint returns 200 status
- Check Content-Type is `text/csv`

Issue: Unauthorized Errors

Solution:

- Verify user is logged in as ADMIN
- Check session cookie is valid
- Verify NextAuth configuration

Feature Capabilities

Supported Actions

The viewer tracks all audit actions from the `AuditAction` enum:

- `LOGIN` / `LOGOUT`
- `CREATE` / `READ` / `UPDATE` / `DELETE`
- `EXPORT` / `IMPORT`
- `SETTINGS_UPDATED`
- `COMMUNICATION_SENT`
- `IMPERSONATION_STARTED` / `IMPERSONATION_STOPPED`
- And more...

Metadata Support

Each log can contain arbitrary JSON metadata for additional context:

```
{
  "changes": { "field": "oldValue -> newValue" },
  "ip": "192.168.1.1",
  "timestamp": "2025-01-05T04:20:00Z"
}
```



Next Steps

Priority Order

1. **Audit Logs Viewer** (COMPLETE)
2. **System Health Monitoring** (Next)
3. **User Impersonation** (After that)

Current Status

- Audit Logs Viewer is ready for production use
- Monitor Render deployment completion
- Test thoroughly before implementing next feature



Related Files

Core Dependencies

- `@/lib/auth` - NextAuth configuration
- `@/lib/prisma` - Database client
- `@/lib/audit` - Audit logging utilities
- `@prisma/client` - Prisma types

No Dependencies On

- Bugsnag (removed)
- Sentry (removed)
- Custom instrumentation (avoided)



Code Quality

Standards Met

- TypeScript strict mode
- Proper error handling
- RBAC enforcement
- Pagination for large datasets
- Input validation
- Responsive UI design
- Clean separation of concerns

Testing Approach

- Build verification completed
- Runtime testing pending deployment
- Manual QA checklist provided above

Usage Examples

API Query Examples

Get Recent Logs

```
GET /api/admin/audit-logs?page=1&limit=50
```

Filter by Action

```
GET /api/admin/audit-logs?action=LOGIN&page=1
```

Date Range Query

```
GET /api/admin/audit-logs?startDate=2025-01-01&endDate=2025-01-31
```

Search Logs

```
GET /api/admin/audit-logs?search=user@example.com
```

Export Filtered Logs

```
GET /api/admin/audit-logs/export?action=LOGIN&startDate=2025-01-01
```



Summary

The Audit Logs Viewer is now:

- Implemented and committed
- Pushed to GitHub
- Build verified (no errors)
- Deployed to Render (auto-deploy triggered)
- Ready for production testing

Critical for: HIPAA compliance, security auditing, incident investigation

Status:  Ready for Production Use

Implementation completed: January 5, 2025

Commit: b36fbdc

Branch: main