Phase 2 Testing Guide: Assessments & Incidents

This guide will help you test the new Assessments and Incidents functionality for the Residents module.

Prerequisites

1. Apply Database Migration

```
cd /home/ubuntu/carelinkai-project

# Generate Prisma Client with new schema
npx prisma generate

# Apply the migration (if database is available)
npx prisma migrate deploy
```

Note: If you don't have a database running, the migration SQL file is already created at: `prisma/migrations/20251208170953_add_assessments_incidents_fields/migration.sql`

2. Seed Demo Data (Optional)

```
# Run the enhanced residents demo seed script
npm run seed:residents-demo
```

This will create 6 demo residents, each with:
- 3-5 comprehensive assessments (ADL, Cognitive, Nutritional, Fall Risk, Pain)
- 1-3 detailed incidents (Falls, Medication Errors, Behavioral incidents)

Quick Start Testing

1. Access the Application

Navigate to: `http://localhost:3000` (or your deployment URL)

2. Login as Operator

Use the operator credentials from your seed data (e.g., `operator@carelinkai.com`)

3. Navigate to Residents

1. Go to **Operator** → **Residents**
2. Click on any resident from the list
3. You should see the resident detail page

Testing Assessments Tab

Test 1: View Assessments List

1. Click on the **"Assessments"** tab (clipboard icon)
2. **Expected:** See a grid of assessment cards with:
   - Assessment type (e.g., "ADL", "Cognitive Assessment")
   - Status badge (color-coded)
   - Score (if applicable)
   - Conducted date and staff member
   - Notes preview
   - Action buttons (View, Edit, Delete)

Test 2: Create New Assessment

1. Click the **"New Assessment"** button (top-right)
2. **Expected:** Modal opens with comprehensive form
3. Fill in the form:
   - **Assessment Type:** Select "ADL (Activities of Daily Living)"
   - **Score:** Enter `22`
   - **Status:** Select "COMPLETED"
   - **Conducted By:** Enter `John Doe, RN`
   - **Conducted At:** Select current date/time
   - **Notes:** Enter sample observation text
   - **Recommendations:** Enter sample recommendations
4. Click **"Create Assessment"**
5. **Expected:**
   - Toast notification: "Assessment created"
   - Modal closes
   - New assessment appears in the grid
   - Page refreshes with updated data

Test 3: View Assessment Details

1. Click the **eye icon** on any assessment card
2. **Expected:** View modal opens showing:
   - Full assessment details
   - All fields displayed in readable format
   - Status badge with color
   - Created/Updated timestamps at bottom
   - "Close" and "Edit Assessment" buttons

Test 4: Edit Assessment

1. Click the **pencil icon** on any assessment card (OR click "Edit Assessment" in view modal)
2. **Expected:** Edit modal opens with form pre-filled with current values
3. Modify some fields (e.g., change status to "IN_PROGRESS", update notes)
4. Click **"Update Assessment"**
5. **Expected:**
   - Toast notification: "Assessment updated"
   - Modal closes
   - Assessment card updates with new information

Test 5: Delete Assessment

1. Click the **trash icon** on any assessment card
2. **Expected:** Confirmation dialog: “Are you sure you want to delete this assessment?”
3. Click **OK/Confirm**
4. **Expected:**
   - Toast notification: “Assessment deleted”
   - Assessment card removed from grid

Test 6: Empty State

1. Delete all assessments for a resident (or use a resident with no assessments)
2. Navigate to Assessments tab
3. **Expected:** Empty state with:
   - Clipboard icon
   - “No assessments yet” heading
   - Descriptive text
   - “Create Assessment” button

Testing Incidents Tab

Test 1: View Incidents List

1. Click on the **“Incidents”** tab (alert triangle icon)
2. **Expected:** See a list of incident cards with:
   - Incident type (e.g., “Fall (without injury)”)
   - Severity badge (Minor/Moderate/Severe/Critical - color-coded)
   - Status badge (Reported/Under Review/Resolved/Follow-up Required)
   - Description preview
   - Occurrence details grid (Occurred date, Location, Reported by, Follow-up flag)
   - Action buttons (View, Edit, Delete)

Test 2: Report New Incident

1. Click the **“Report Incident”** button (top-right)
2. **Expected:** Modal opens with comprehensive multi-section form
3. Fill in the form:

Incident Information:
- **Incident Type:** Select “Fall (without injury)”
- **Severity:** Auto-selected to “MODERATE” (based on type)
- **Status:** Select “REPORTED”
- **Follow-up Required:** Check the box
- **Description:** Enter incident description

Occurrence Details:
- **Occurred At:** Select date/time
- **Location:** Enter “Room 101”
- **Witnessed By:** Enter “Jane Smith, CNA”

Reporting Details:
- **Reported By:** Enter “John Doe, RN”

- **Reported At:** Select current date/time
- **Actions Taken:** Enter detailed actions

1. Click **“Report Incident”**
2. **Expected:**
   - Toast notification: “Incident reported”
   - Modal closes
   - New incident appears in the list

Test 3: View Incident Details

1. Click the **eye icon** on any incident card
2. **Expected:** View modal opens showing:
   - Full incident details
   - All sections with complete information
   - Status and severity badges
   - Follow-up requirement flag (if applicable)
   - Resolution details (if status is RESOLVED)
   - Created/Updated timestamps
   - “Close” and “Edit Incident” buttons

Test 4: Update Incident Status to Resolved

1. Click the **pencil icon** on any incident card
2. **Expected:** Edit modal opens
3. Change **Status** to “RESOLVED”
4. **Expected:** Resolution Details section appears
5. Fill in resolution fields:
   - **Resolved By:** Enter staff name
   - **Resolved At:** Select date/time
   - **Resolution Notes:** Enter how it was resolved
6. Click **“Update Incident”**
7. **Expected:**
   - Toast notification: “Incident updated”
   - Modal closes
   - Incident card updates with RESOLVED status badge (green)

Test 5: Delete Incident

1. Click the **trash icon** on any incident card
2. **Expected:** Confirmation dialog: “Are you sure you want to delete this incident report?”
3. Click **OK/Confirm**
4. **Expected:**
   - Toast notification: “Incident deleted”
   - Incident card removed from list

Test 6: Empty State

1. Use a resident with no incidents (or delete all incidents)
2. Navigate to Incidents tab
3. **Expected:** Empty state with:
   - Alert triangle icon
   - “No incidents reported” heading

- Descriptive text: “This is a good sign!”
- “Report Incident” button

Testing Responsive Design

Mobile Testing

1. Open browser DevTools (F12)
2. Toggle device toolbar (Ctrl+Shift+M or Cmd+Shift+M)
3. Select a mobile device (e.g., iPhone 12)
4. Test both Assessments and Incidents tabs:
   - Grid should switch to single column
   - Modals should be scrollable
   - Forms should be touch-friendly
   - Buttons should be easily tappable

Tablet Testing

1. Select a tablet device (e.g., iPad)
2. Test both tabs:
   - Grid should show 2 columns
   - Modals should fit well
   - Forms should be well-spaced

Testing API Endpoints (Optional)

Using cURL or Postman

List Assessments

```
curl http://localhost:3000/api/residents/[RESIDENT_ID]/assessments \
  -H "Cookie: your-session-cookie"
```

Create Assessment

```
curl -X POST http://localhost:3000/api/residents/[RESIDENT_ID]/assessments \
  -H "Content-Type: application/json" \
  -H "Cookie: your-session-cookie" \
  -d '{
    "type": "ADL",
    "score": 24,
    "status": "COMPLETED",
    "conductedBy": "Test Staff, RN",
    "conductedAt": "2024-12-08T10:00:00Z",
    "notes": "Test notes",
    "recommendations": "Test recommendations"
  }'
```

List Incidents

```
curl http://localhost:3000/api/residents/[RESIDENT_ID]/incidents \
  -H "Cookie: your-session-cookie"
```

Create Incident

```
curl -X POST http://localhost:3000/api/residents/[RESIDENT_ID]/incidents \
  -H "Content-Type: application/json" \
  -H "Cookie: your-session-cookie" \
  -d '{
    "type": "FALL_NO_INJURY",
    "severity": "MODERATE",
    "status": "REPORTED",
    "description": "Test incident",
    "occurredAt": "2024-12-08T14:30:00Z",
    "location": "Test Room",
    "reportedBy": "Test Reporter",
    "reportedAt": "2024-12-08T14:35:00Z",
    "actionsTaken": "Test actions",
    "followUpRequired": false
  }'
```

Verification Checklist

Assessments Tab ✅

- [ ] Tab appears in navigation
- [ ] Assessments list displays correctly
- [ ] Assessment cards show all key information
- [ ] Status badges have correct colors
- [ ] “New Assessment” button works
- [ ] Create modal opens and form works
- [ ] All 8 assessment types are available
- [ ] Assessment creation succeeds
- [ ] View modal displays complete information
- [ ] Edit modal pre-fills data correctly
- [ ] Update operation works
- [ ] Delete confirmation appears
- [ ] Delete operation works
- [ ] Empty state displays when no assessments
- [ ] Toast notifications appear for all actions
- [ ] Responsive design works on mobile/tablet

Incidents Tab ✅

- [ ] Tab appears in navigation
- [ ] Incidents list displays correctly
- [ ] Incident cards show all key information
- [ ] Severity badges have correct colors

- [ ] Status badges have correct colors
- [ ] "Report Incident" button works
- [ ] Create modal opens with all sections
- [ ] All 13 incident types are available
- [ ] Severity auto-selects based on type
- [ ] Incident creation succeeds
- [ ] View modal displays complete information
- [ ] Edit modal pre-fills data correctly
- [ ] Status change to RESOLVED shows resolution fields
- [ ] Update operation works
- [ ] Delete confirmation appears
- [ ] Delete operation works
- [ ] Empty state displays when no incidents
- [ ] Toast notifications appear for all actions
- [ ] Responsive design works on mobile/tablet

Integration ✅

- [ ] No regressions in Overview tab
- [ ] No regressions in Details tab
- [ ] Tab navigation preserves query parameter
- [ ] All Phase 1 features still work
- [ ] Photo upload still works
- [ ] Status actions still work
- [ ] Notes still work
- [ ] Compliance panel still works

Troubleshooting

Issue: "Field does not exist" errors

Solution: Regenerate Prisma Client:

```
npx prisma generate
```

Issue: Assessments/Incidents tab is empty but demo data was seeded

Solution:
1. Check if you're viewing the correct resident
2. Verify the API is returning data: Check browser DevTools → Network tab
3. Check for JavaScript errors in Console tab

Issue: Can't create assessment/incident - validation error

Solution:
1. Check browser Console for specific validation errors

2. Ensure required fields are filled (marked with *)
3. Verify datetime fields are in correct format

Issue: Migration fails

Solution:
1. Check database connection
2. Verify DATABASE_URL in .env file
3. Check migration file for syntax errors
4. Try: `npx prisma migrate reset` (⚠️ WARNING: Deletes all data!)

Success Criteria

Phase 2 is successfully deployed when:
- ✅ All Assessments Tab tests pass
- ✅ All Incidents Tab tests pass
- ✅ No regressions in Phase 1 features
- ✅ Demo data seeds successfully
- ✅ Responsive design works across devices
- ✅ All CRUD operations function correctly
- ✅ Audit logs are created for all operations
- ✅ Error handling and validation work properly

Next Steps

After successful testing:
1. Deploy to staging environment
2. Perform user acceptance testing (UAT)
3. Train staff on new features
4. Monitor for any issues
5. Collect feedback for future enhancements
6. Begin Phase 3 planning (Compliance & Family tabs)

Need Help?
- Check `PHASE_2_IMPLEMENTATION_SUMMARY.md` for detailed technical documentation
- Review inline code comments in components and API routes
- Check browser Console and Network tabs for errors
- Verify Prisma schema matches migration file

Happy Testing! 🎉