Residents Testing Checklist

Purpose: Comprehensive QA guide for Residents domain refresh
Last Updated: December 8, 2024
Scope: Priority 1 & 2 features

Test Environment Setup

Demo Accounts

Role	Email	Password	Access Level
Operator	`demo.operator@carelinkai.test`	`DemoUser123!`	Full CRUD for own homes
Family	`demo.family@carelinkai.test`	`DemoUser123!`	View-only for own residents
Admin	`demo.admin@carelinkai.test`	`DemoUser123!`	Full system access

Prerequisites

- [ ] Application deployed and running
- [ ] Database migrations applied
- [ ] Test accounts active and verified
- [ ] Browser console open (F12) to monitor errors

Test Suite 1: Navigation & Discovery

Test 1.1: Operator Sidebar Navigation

Objective: Verify Residents link is visible and functional

Steps:
1. Sign in as operator (`demo.operator@carelinkai.test`)
2. Locate sidebar menu
3. Find “Residents” link with Users icon
4. Click “Residents” link
5. Verify URL changes to `/operator/residents`
6. Verify “Residents” link has active highlighting

Expected Results:
- ✅ “Residents” link visible in sidebar between “Homes” and “Inquiries”
- ✅ Icon displays correctly (Users icon)

- ✅ Navigation works without page reload
- ✅ Active state highlighted correctly
- ✅ No console errors

Status: [ ] Pass [ ] Fail

Notes:

Test Suite 2: Status Action Buttons

Test 2.1: Admit Button

Objective: Test resident admission with confirmation flow

Preconditions:
- Resident exists with status “PENDING” or “INQUIRY”

Steps:
1. Sign in as operator
2. Navigate to `/operator/residents`
3. Click on a PENDING resident
4. Locate “Status Actions” section
5. Set effective date to today
6. Click “Admit” button
7. Verify confirmation modal appears
8. Click “Cancel” in modal
9. Verify modal closes, no action taken
10. Click “Admit” again
11. Click “Confirm” in modal
12. Verify loading spinner appears
13. Wait for completion

Expected Results:
- ✅ Admit button visible for non-ACTIVE residents
- ✅ Confirmation modal shows correct date
- ✅ Cancel button works without side effects
- ✅ Confirm button triggers API call
- ✅ Loading state disables button
- ✅ Success toast displays: “Resident admitted successfully”
- ✅ Status updates to “ACTIVE”
- ✅ Page refreshes with new status
- ✅ No console errors

Status: [ ] Pass [ ] Fail

Notes:

Test 2.2: Discharge Button

Objective: Test resident discharge workflow

Preconditions:
- Resident exists with status “ACTIVE”

Steps:
1. Sign in as operator
2. Navigate to an ACTIVE resident detail page
3. Set effective date to today
4. Click “Discharge” button
5. Verify confirmation modal appears
6. Read confirmation message
7. Click “Confirm”
8. Wait for completion

Expected Results:
- ✅ Discharge button visible only for ACTIVE residents
- ✅ Confirmation asks: “Are you sure you want to discharge…”
- ✅ Success toast: “Resident discharged successfully”
- ✅ Status updates to “DISCHARGED”
- ✅ Home occupancy decremented (verify on home detail page)
- ✅ No console errors

Status: [ ] Pass [ ] Fail

Notes:

Test 2.3: Mark Deceased Button

Objective: Test deceased status transition

Preconditions:
- Resident exists with status “ACTIVE”

Steps:
1. Navigate to an ACTIVE resident detail page
2. Click “Mark Deceased” button (red button)
3. Read confirmation warning
4. Click “Cancel” first
5. Click “Mark Deceased” again
6. Click “Confirm”
7. Verify results

Expected Results:
- ✅ Red color indicates destructive action
- ✅ Confirmation includes warning: “This action cannot be undone”
- ✅ Success toast: “Resident marked as deceased”
- ✅ Status updates to “DECEASED”
- ✅ Status actions disappear (no further transitions allowed)
- ✅ No console errors

Status: [ ] Pass [ ] Fail

Notes:

Test 2.4: Transfer Button

Objective: Test resident transfer between homes

Preconditions:
- Resident exists with status “ACTIVE”
- Operator has multiple homes

Steps:
1. Navigate to an ACTIVE resident detail page
2. Click “Transfer” button (blue)
3. Verify transfer dialog opens
4. Check that home list loads
5. Verify current home is not in list (or marked differently)
6. Select a different home
7. Verify selection highlights
8. Click “Cancel”
9. Verify dialog closes without action
10. Click “Transfer” again
11. Select a home
12. Click “Transfer” button in dialog
13. Wait for completion

Expected Results:
- ✅ Transfer button visible only for ACTIVE residents
- ✅ Dialog shows “Transfer Resident to Another Home” title
- ✅ Home list displays with names, locations, occupancy
- ✅ Loading state shows “Loading homes...”
- ✅ Radio buttons work correctly
- ✅ Transfer button disabled until home selected
- ✅ Success toast: “Resident transferred successfully”
- ✅ Resident’s homeId updated in database
- ✅ Old home occupancy decremented
- ✅ New home occupancy incremented
- ✅ Timeline shows transfer event
- ✅ No console errors

Status: [ ] Pass [ ] Fail

Notes:

Test Suite 3: Archive Functionality

Test 3.1: Archive Button Display

Objective: Verify archive button appears correctly

Steps:
1. Sign in as operator
2. Navigate to any resident detail page
3. Locate page header
4. Find “Archive” button (top-right area)

Expected Results:
- ✅ Archive button visible in header
- ✅ Grey/neutral color scheme
- ✅ Archive icon displayed
- ✅ Button not visible if resident already archived

Status: [ ] Pass [ ] Fail

Notes:

Test 3.2: Archive Workflow

Objective: Test archiving a resident

Steps:
1. Navigate to a non-archived resident
2. Click “Archive” button
3. Read confirmation dialog
4. Click “Cancel” first
5. Verify dialog closes, no action
6. Click “Archive” again
7. Click “Archive” in confirmation
8. Wait for completion

Expected Results:
- ✅ Confirmation asks: “Are you sure you want to archive [Name]?”
- ✅ Explanation text mentions “Show Archived” filter
- ✅ Success toast: “Resident archived successfully”
- ✅ Redirect to `/operator/residents` list page
- ✅ Archived resident NOT in default list
- ✅ `archivedAt` timestamp set in database
- ✅ Status set to “DISCHARGED” if not already
- ✅ No console errors

Status: [ ] Pass [ ] Fail

Notes:

Test 3.3: Show Archived Filter

Objective: Test viewing archived residents

Steps:
1. Navigate to `/operator/residents` list page
2. Verify archived resident NOT visible
3. Locate “Show Archived” checkbox in filters
4. Check the “Show Archived” checkbox
5. Click “Search” button
6. Verify archived residents appear
7. Look for “Archived” badge or indicator
8. Uncheck “Show Archived”
9. Click “Search”
10. Verify archived residents disappear

Expected Results:
- ✅ Archived residents hidden by default
- ✅ Checkbox labeled “Show Archived”
- ✅ Checking box adds `showArchived=true` to URL
- ✅ Archived residents visible when checked
- ✅ Archived residents have visual indicator
- ✅ Unchecking hides archived residents again
- ✅ Filter state persists in URL
- ✅ No console errors

Status: [ ] Pass [ ] Fail

Notes:

Test Suite 4: Medical Information

Test 4.1: Medical Fields Display

Objective: Verify medical info section in edit form

Steps:
1. Sign in as operator
2. Navigate to any resident
3. Click “Edit” link
4. Scroll to “Medical Information” section
5. Verify all fields present

Expected Results:
- ✅ Section has amber warning icon
- ✅ HIPAA compliance notice displayed
- ✅ Four textarea fields visible:
- Medical Conditions
- Current Medications
- Allergies

- Dietary Restrictions
- ✅ Each field has character counter
- ✅ Placeholders provide guidance
- ✅ Fields styled consistently with rest of form

Status: [ ] Pass [ ] Fail

Notes:

Test 4.2: Character Count Validation

Objective: Test character limits enforce correctly

Steps:
1. On edit form, locate “Medical Conditions” field
2. Type or paste text to verify counter updates
3. Try to exceed 2000 characters
4. Verify “Current Medications” has 2000 char limit
5. Verify “Allergies” has 1000 char limit
6. Verify “Dietary Restrictions” has 1000 char limit

Expected Results:
- ✅ Counter shows “0 / 2000 characters” initially
- ✅ Counter updates in real-time while typing
- ✅ Cannot type beyond limit (enforced by maxLength)
- ✅ Counter turns red near limit (if implemented)
- ✅ Medical Conditions: 2000 max
- ✅ Medications: 2000 max
- ✅ Allergies: 1000 max
- ✅ Dietary Restrictions: 1000 max

Status: [ ] Pass [ ] Fail

Notes:

Test 4.3: Save Medical Information

Objective: Test medical data persistence

Steps:
1. Fill in all four medical fields with sample data:
- Medical Conditions: “Type 2 Diabetes, Hypertension”
- Medications: “Metformin 500mg twice daily”
- Allergies: “Penicillin, shellfish”
- Dietary Restrictions: “Low sodium, diabetic-friendly”
2. Click “Save Changes”
3. Wait for success toast
4. Navigate away from edit page
5. Return to edit page
6. Verify all data persisted

Expected Results:
- ✅ Save button shows “Saving...” during request
- ✅ Success toast: “Resident updated successfully”
- ✅ Redirect to detail page
- ✅ Returning to edit shows saved data
- ✅ Data matches what was entered
- ✅ Character counters show correct counts
- ✅ Database has encrypted values (check directly if possible)
- ✅ No console errors

Status: [ ] Pass [ ] Fail

Notes:

Test 4.4: Clear Medical Fields

Objective: Test removing medical information

Steps:
1. Edit a resident with existing medical data
2. Clear all medical fields (delete all text)
3. Click “Save Changes”
4. Verify fields are empty in database
5. Return to edit page
6. Verify fields are empty in UI

Expected Results:
- ✅ Clearing fields sets values to NULL in database
- ✅ Save succeeds with empty fields
- ✅ No validation errors for empty medical fields
- ✅ Placeholder text reappears when empty

Status: [ ] Pass [ ] Fail

Notes:

Test Suite 5: Photo Upload

Test 5.1: Photo Upload Component Display

Objective: Verify photo upload UI renders correctly

Steps:
1. Sign in as operator
2. Edit any resident
3. Locate “Profile Photo” section (should be first)
4. Verify component layout

Expected Results:
- ✅ Section titled “Profile Photo”
- ✅ Circular 128x128px avatar displayed
- ✅ If no photo: Initials displayed (e.g., “JD” for John Doe)
- ✅ If photo exists: Photo displayed in circle
- ✅ “Upload Photo” button visible below avatar
- ✅ Info text: “Upload a photo (JPEG, PNG, or WebP). Maximum file size: 5MB.”
- ✅ No broken images or layout issues

Status: [ ] Pass [ ] Fail

Notes:

Test 5.2: Upload Valid Photo

Objective: Test successful photo upload

Preconditions:
- Have a valid JPEG/PNG/WebP file < 5MB ready

Steps:
1. On edit form, click “Upload Photo” button
2. Select a valid image file (JPEG, < 5MB)
3. Wait for upload
4. Verify preview appears
5. Check for success feedback

Expected Results:
- ✅ File picker opens with correct file type filters
- ✅ Loading spinner appears during upload
- ✅ Preview shows uploaded image in circle
- ✅ Success toast: “Photo uploaded successfully”
- ✅ “Change Photo” button replaces “Upload Photo”
- ✅ “Remove” button appears next to “Change Photo”
- ✅ Photo URL saved to database
- ✅ Photo file exists in `/public/uploads/residents/`
- ✅ No console errors

Status: [ ] Pass [ ] Fail

Notes:

Test 5.3: File Validation - Size Limit

Objective: Test 5MB size limit enforcement

Preconditions:
- Have a file > 5MB ready (or use browser dev tools to simulate)

Steps:
1. Click “Upload Photo”

2. Select a file > 5MB
3. Observe result

Expected Results:
- ✅ Error toast: “Image must be less than 5MB”
- ✅ Photo not uploaded
- ✅ No API request made (check Network tab)
- ✅ Previous photo (or initials) still displayed

Status: [ ] Pass [ ] Fail

Notes:

Test 5.4: File Validation - Type Check

Objective: Test file type restriction

Steps:
1. Click “Upload Photo”
2. Try to select a PDF, GIF, or other non-JPEG/PNG/WebP file
3. Observe result

Expected Results:
- ✅ File picker should only show JPEG/PNG/WebP files
- ✅ If wrong type selected: Error toast
- ✅ Error message: “Please select a JPEG, PNG, or WebP image”
- ✅ Photo not uploaded

Status: [ ] Pass [ ] Fail

Notes:

Test 5.5: Remove Photo

Objective: Test photo deletion

Preconditions:
- Resident has an uploaded photo

Steps:
1. Navigate to resident edit page with photo
2. Verify “Remove” button visible
3. Click “Remove” button
4. Click “OK” in browser confirmation (if present)
5. Wait for completion

Expected Results:
- ✅ Confirmation prompt: “Are you sure you want to remove this photo?”
- ✅ Loading state during deletion
- ✅ Success toast: “Photo removed successfully”
- ✅ Initials placeholder reappears

- ✅ “Upload Photo” button replaces “Change Photo”
- ✅ “Remove” button disappears
- ✅ photoUrl set to NULL in database
- ✅ Old photo file deleted from `/public/uploads/residents/`
- ✅ No console errors

Status: [ ] Pass [ ] Fail

Notes:

Test 5.6: Change Existing Photo

Objective: Test replacing an existing photo

Preconditions:
- Resident has an uploaded photo

Steps:
1. Navigate to edit page with existing photo
2. Click “Change Photo” button
3. Select a different image
4. Wait for upload
5. Verify old photo replaced

Expected Results:
- ✅ New photo uploaded successfully
- ✅ Old photo file deleted from filesystem
- ✅ Only one photo exists per resident
- ✅ Preview shows new photo immediately
- ✅ Success toast displayed

Status: [ ] Pass [ ] Fail

Notes:

Test Suite 6: Family Resident Detail Page

Test 6.1: Timeline Section

Objective: Verify family sees resident timeline

Steps:
1. Sign in as family (`demo.family@carelinkai.test`)
2. Navigate to `/family/residents`
3. Click on a resident
4. Locate “Timeline” section
5. Verify contents

Expected Results:
- ✅ Timeline section visible

- ✅ Shows appointments, notes, documents
- ✅ Events sorted by date (newest first)
- ✅ Each event has icon, title, date
- ✅ Empty state if no events: “No recent activity”
- ✅ No console errors

Status: [ ] Pass [ ] Fail

Notes:

Test 6.2: Compliance Summary

Objective: Verify compliance metrics visible to family

Steps:
1. On family resident detail page
2. Locate “Compliance Summary” section
3. Verify cards display

Expected Results:
- ✅ Four metric cards visible:
- Open (count)
- Completed (count)
- Due Soon (14 days) (count)
- Overdue (count)
- ✅ Numbers accurate (cross-check with operator view)
- ✅ Card layout clean and readable
- ✅ Help text: “Family can view status counts only”

Status: [ ] Pass [ ] Fail

Notes:

Test 6.3: Family-Visible Notes

Objective: Verify note visibility controls work

Steps:
1. As operator, create a note with visibility “FAMILY”
2. Create another note with visibility “INTERNAL”
3. Sign in as family
4. Navigate to resident detail
5. Check “Care Team Notes” section

Expected Results:
- ✅ “Care Team Notes (Family-visible)” heading
- ✅ FAMILY-visible note displays
- ✅ INTERNAL note does NOT display
- ✅ Note shows author name and date
- ✅ Note content truncated if long (240 chars)
- ✅ Empty state: “No notes available” if none

Status: [ ] Pass [ ] Fail

Notes:

Test 6.4: Contacts Section

Objective: Verify contacts display for family

Steps:
1. On family resident detail page
2. Locate “Contacts” section
3. Verify table layout

Expected Results:
- ✅ Table with columns: Name, Relationship, Phone, Email
- ✅ Primary contact has badge: “PRIMARY”
- ✅ Data displays correctly
- ✅ Empty state: “No contacts on file”
- ✅ Responsive table scrolls on mobile

Status: [ ] Pass [ ] Fail

Notes:

Test Suite 7: RBAC & Permissions

Test 7.1: Operator Scoping

Objective: Verify operators see only their residents

Preconditions:
- Two operators with different homes
- Each home has residents

Steps:
1. Sign in as Operator A
2. Navigate to `/operator/residents`
3. Note resident IDs/names displayed
4. Sign out
5. Sign in as Operator B
6. Navigate to `/operator/residents`
7. Verify different resident list

Expected Results:
- ✅ Operator A sees only residents in their homes
- ✅ Operator B sees only residents in their homes
- ✅ No overlap (unless they share a home)
- ✅ Direct URL access blocked:
- Operator A accessing Operator B’s resident returns 403

Status: [ ] Pass [ ] Fail

Notes:

Test 7.2: Family Access

Objective: Verify families see only their residents

Steps:
1. Sign in as family
2. Navigate to `/family/residents`
3. Verify only family's residents displayed
4. Try accessing another family's resident by URL
5. Verify 403 or redirect

Expected Results:
- ✅ Family sees only their own residents
- ✅ Cannot access others' residents via URL
- ✅ List filtered correctly
- ✅ No edit/delete options visible

Status: [ ] Pass [ ] Fail

Notes:

Test 7.3: Admin Access

Objective: Verify admin has full access

Steps:
1. Sign in as admin
2. Navigate to residents list
3. Verify can see all residents across all operators
4. Edit any resident
5. Verify save succeeds

Expected Results:
- ✅ Admin sees all residents (no filtering)
- ✅ Can edit any resident
- ✅ Can archive any resident
- ✅ All status actions work
- ✅ All features accessible

Status: [ ] Pass [ ] Fail

Notes:

Test Suite 8: Integration & Data Integrity

Test 8.1: Occupancy Management

Objective: Verify home occupancy updates correctly

Steps:
1. Note a home's current occupancy
2. Admit a resident to that home
3. Check home occupancy increased by 1
4. Discharge the resident
5. Check home occupancy decreased by 1
6. Transfer resident to different home
7. Check both homes' occupancy updated

Expected Results:
- ✅ Admit increases occupancy
- ✅ Discharge decreases occupancy
- ✅ Transfer updates both homes
- ✅ Occupancy never goes negative
- ✅ Occupancy never exceeds capacity

Status: [ ] Pass [ ] Fail

Notes:

Test 8.2: Timeline Event Creation

Objective: Verify status changes create timeline events

Steps:
1. Admit a resident
2. Navigate to resident detail page
3. Check timeline for admission event
4. Discharge the resident
5. Check timeline for discharge event
6. Transfer the resident
7. Check timeline for transfer event

Expected Results:
- ✅ Each status change creates a timeline event
- ✅ Events have correct type, title, date
- ✅ Events appear in chronological order
- ✅ Transfer event shows target home

Status: [ ] Pass [ ] Fail

Notes:

Test Suite 9: Mobile Responsiveness

Test 9.1: Responsive List Page

Objective: Test residents list on mobile

Steps:
1. Open `/operator/residents` on mobile (or resize browser to 375px)
2. Verify layout adapts
3. Test filters
4. Test search
5. Scroll through list

Expected Results:
- ✅ Table converts to cards or stacks vertically
- ✅ Filters accessible (may be collapsible)
- ✅ Search bar full-width
- ✅ No horizontal scroll
- ✅ Touch targets at least 44px
- ✅ Text readable without zoom

Status: [ ] Pass [ ] Fail

Notes:

Test 9.2: Responsive Detail Page

Objective: Test resident detail on mobile

Steps:
1. Open resident detail page on mobile
2. Verify sections stack vertically
3. Test status action buttons
4. Test archive button

Expected Results:
- ✅ Sections stack in single column
- ✅ Status action buttons stack vertically or wrap
- ✅ Archive button accessible
- ✅ All content readable
- ✅ No overlapping elements
- ✅ Modals/dialogs fit screen

Status: [ ] Pass [ ] Fail

Notes:

Test 9.3: Responsive Edit Form

Objective: Test edit form on mobile

Steps:
1. Open edit form on mobile
2. Test photo upload on mobile
3. Fill in medical fields
4. Save changes

Expected Results:
- ✅ Form fields stack vertically
- ✅ Photo upload works on mobile
- ✅ Textareas expand as needed
- ✅ Save button accessible
- ✅ Keyboard doesn't obscure fields
- ✅ No horizontal scroll

Status: [ ] Pass [ ] Fail

Notes:

Test Suite 10: Error Handling

Test 10.1: API Errors

Objective: Verify graceful error handling

Steps:
1. Turn off internet/WiFi
2. Try to admit a resident
3. Observe error handling
4. Restore internet
5. Try uploading invalid photo
6. Observe error message

Expected Results:
- ✅ Network errors show toast: "Failed to..."
- ✅ Button re-enables after error
- ✅ User can retry action
- ✅ No app crashes
- ✅ Error messages are user-friendly
- ✅ Console logs show technical details

Status: [ ] Pass [ ] Fail

Notes:

Test 10.2: Validation Errors

Objective: Test form validation

Steps:
1. Try to save resident with empty first name
2. Try to save with future date of birth
3. Try to upload 10MB photo

Expected Results:
- ✅ Required field errors prevent submit
- ✅ Date validation works
- ✅ File size errors before upload
- ✅ Error messages display clearly
- ✅ Fields highlighted in red

Status: [ ] Pass [ ] Fail

Notes:

Test Suite 11: Browser Compatibility

Test 11.1: Cross-Browser Testing

Browsers to test:
- [ ] Chrome (latest)
- [ ] Firefox (latest)
- [ ] Safari (latest)
- [ ] Edge (latest)
- [ ] Mobile Safari (iOS)
- [ ] Chrome Mobile (Android)

Key features to verify:
- Navigation works
- Status actions work
- Photo upload works
- Modals display correctly
- Styles render correctly

Status: [ ] Pass [ ] Fail

Notes:

Test Suite 12: Performance

Test 12.1: Page Load Times

Objective: Verify reasonable load times

Steps:
1. Open residents list page
2. Measure time to interactive

3. Open resident detail page
4. Measure time to interactive

Expected Results:
- ✅ List page loads < 2 seconds
- ✅ Detail page loads < 1.5 seconds
- ✅ Edit page loads < 1.5 seconds
- ✅ Photo upload completes < 3 seconds
- ✅ Status changes complete < 1 second

Status: [ ] Pass [ ] Fail

Notes:

Test Suite 13: Regression Testing

Test 13.1: Existing Features Unaffected

Objective: Verify no regressions in other areas

Features to verify still work:
- [ ] Operator dashboard
- [ ] Home management (list/create/edit)
- [ ] Inquiry management
- [ ] Aide marketplace
- [ ] Provider marketplace
- [ ] Family home search
- [ ] Messaging system
- [ ] Calendar/shifts
- [ ] Settings pages
- [ ] Admin analytics

Status: [ ] Pass [ ] Fail

Notes:

Test Suite 14: Automated Checks

Test 14.1: Build & Type Check

Commands to run:

```
cd /home/ubuntu/carelinkai
npm run type-check
npm run lint
npm run build
```

Expected Results:
- ✅ `type-check` : 0 errors
- ✅ `lint` : 0 errors
- ✅ `build` : succeeds
- ✅ No warnings (or only minor ones)

Status: [ ] Pass [ ] Fail

Output:

```
[Paste output here if failures]
```

Summary Report

Test Statistics

Category	Total Tests	Passed	Failed	Skipped
Navigation	1			
Status Actions	4			
Archive	3			
Medical Info	4			
Photo Upload	6			
Family Views	4			
RBAC	3			
Integration	2			
Mobile	3			
Error Handling	2			
Browsers	1			
Performance	1			
Regression	1			
Automated	1			
TOTAL	**36**			

Critical Issues Found

1.
2.
3.

Minor Issues Found

1.
2.
3.

Recommendations

- [ ] Ready for production
- [ ] Needs fixes before production
- [ ] Needs more testing

Sign-Off

Tester Name: _____
Date: _____
Signature: _____

Appendix: Test Data

Sample Medical Information

```
Medical Conditions:
- Type 2 Diabetes (diagnosed 2015)
- Hypertension (controlled with medication)
- Osteoarthritis in knees
- History of falls (2 in past year)

Current Medications:
- Metformin 500mg - twice daily with meals
- Lisinopril 10mg - once daily in morning
- Atorvastatin 20mg - once daily at bedtime
- Aspirin 81mg - once daily

Allergies:
- Penicillin (anaphylaxis)
- Shellfish (hives)
- Latex (rash)

Dietary Restrictions:
- Low sodium diet (<2000mg/day)
- Diabetic-friendly meals
- No shellfish
- Prefers soft foods due to dental issues
```

Sample Photos for Testing

- Valid JPEG: 500KB, 800x800px

- Valid PNG: 1.2MB, 1200x1200px
- Valid WebP: 300KB, 600x600px
- Invalid: 6MB JPEG (too large)
- Invalid: PDF document (wrong type)

End of Testing Checklist