

Feature #3 Tour Scheduling - Critical Bug Fix



Bug Report

Issue

Tour submission fails with error: "Something went wrong. Please try again later."

Impact: CRITICAL - Families cannot request tours, Feature #3 is non-functional

User Flow (Before Fix)

1. ✓ Family navigates to AI Match page
2. ✓ Family views home details
3. ✓ Family clicks "Schedule Tour" button
4. ✓ Tour modal opens with professional UI
5. ✓ Family selects date range
6. ✓ AI suggests optimal time slots
7. ✓ Family selects preferred time slot
8. ✓ Family adds notes (optional)
9. ✗ **FAILURE:** Click "Schedule Tour" → Error message appears
10. ✗ No tour created in database
11. ✗ "My Tours" page shows "No tours yet"



Root Cause Analysis

Technical Investigation

1. API Endpoint Analysis

File: `src/app/api/family/tours/request/route.ts`

Problem Code (Line 73 - BEFORE):

```
const tourRequest = await prisma.tourRequest.create({
  data: {
    familyId: family.id,
    homeId: home.id,
    operatorId: home.operatorId,
    requestedTimes: validatedData.requestedTimes.map((t) => new Date(t)), // ✗ BUG
    familyNotes: validatedData.familyNotes,
    status: "PENDING",
  },
});
```

2. Schema Definition

File: `prisma/schema.prisma` (Line 2482)

```

model TourRequest {
  id          String      @id @default(cuid())
  familyId    String
  homeId      String
  operatorId   String

  // Requested times from family (stored as array of DateTime)
  requestedTimes  Json      // ⚠ JSON type, not DateTime[]

  // ... other fields
}

```

3. Data Flow Analysis

Frontend (TourRequestModal.tsx):

```

// Line 134
requestedTimes: [selectedSlot] // Array of ISO strings
// Example: ["2025-08-02T10:00:00.000Z"]

```

Validation (route.ts):

```

// Line 17
requestedTimes: z.array(z.string().datetime()) // Validates ISO strings ✓

```

Database Storage (BEFORE):

```

// Line 73
requestedTimes: validatedData.requestedTimes.map((t) => new Date(t))
// Converts: ["2025-08-02T10:00:00.000Z"] → [Date object]
// ✗ Prisma JSON field cannot serialize Date objects

```

Root Cause

JSON Serialization Error: Prisma's `Json` field type expects JSON-compatible data (strings, numbers, arrays, objects). JavaScript `Date` objects are not JSON-serializable and must be converted to ISO strings before storage.

Error Chain:

1. API receives ISO strings from frontend ✓
 2. Zod validates ISO strings ✓
 3. API converts to Date objects ✗
 4. Prisma attempts to serialize Date objects to JSON ✗
 5. Serialization fails → Database insert fails ✗
 6. API returns 500 error ✗
 7. User sees generic error message ✗
-

✓ Solution

Code Changes

File: `src/app/api/family/tours/request/route.ts`

1. Fixed Data Conversion (Line 83)

```
// BEFORE:
requestedTimes: validatedData.requestedTimes.map((t) => new Date(t)),

// AFTER:
requestedTimes: validatedData.requestedTimes, // Keep as ISO strings for JSON field
```

2. Added Comprehensive Logging

Added logging at each step for debugging:

```
// Line 23: Request received
console.log("[Tour Request API] POST request received");

// Line 32: User authenticated
console.log("[Tour Request API] User authenticated:", session.user.id, session.user.role);

// Line 42: Request body
console.log("[Tour Request API] Request body:", JSON.stringify(body, null, 2));

// Line 45: Validation success
console.log("[Tour Request API] Data validated successfully");

// Line 70-76: Pre-creation data
console.log("[Tour Request API] Creating tour request with data:", {
  familyId: family.id,
  homeId: home.id,
  operatorId: home.operatorId,
  requestedTimes: validatedData.requestedTimes,
  familyNotes: validatedData.familyNotes,
});

// Line 106: Success
console.log("[Tour Request API] Tour request created successfully:", tourRequest.id);
```

3. Enhanced Error Handling (Lines 129-156)

```

catch (error) {
  console.error("[Tour Request API] Error:", error);

  // Log detailed error information
  if (error instanceof Error) {
    console.error("[Tour Request API] Error name:", error.name);
    console.error("[Tour Request API] Error message:", error.message);
    console.error("[Tour Request API] Error stack:", error.stack);
  }

  if (error instanceof z.ZodError) {
    console.error("[Tour Request API] Validation error details:", JSON.stringify(
      error.errors, null, 2));
    return NextResponse.json(
      { error: "Validation error", details: error.errors },
      { status: 400 }
    );
  }

  // Return detailed error message in development, generic in production
  const errorMessage = process.env.NODE_ENV === "development" && error instanceof Error
    ? error.message
    : "Failed to create tour request";

  return NextResponse.json(
    { error: errorMessage },
    { status: 500 }
  );
}

```

Impact Assessment

Before Fix

- ❌ Tour submissions: 0% success rate
- ❌ Database records: 0 tours created
- ❌ User experience: Complete feature failure
- ❌ Error visibility: Generic error messages only

After Fix




- ✅ Tour submissions: Should work end-to-end
 - ✅ Database records: Tours properly saved
 - ✅ User experience: Smooth tour request flow
 - ✅ Error visibility: Detailed logging for debugging
-

Testing Guide





Manual Testing Steps

Test 1: Successful Tour Submission


1. Login as Family user (Chris Smith / chris@example.com)

2. Navigate to "Find Care Home"
3. Click "View Details" on any home
4. Click "Schedule Tour" button
5. Select date range (e.g., today → +30 days)
6. Click "Next"
7. Select a suggested time slot
8. Click "Next"
9. Add notes (optional): "Looking forward to visiting!"
10. Click "Submit Request"
11. **Expected:**  Success message appears
12. **Expected:**  Modal shows "Tour Request Submitted!"
13. **Expected:**  Redirect to tours list



Test 2: Verify Database Record

1. Navigate to "My Tours" page
2. **Expected:**  New tour appears in list
3. **Expected:**  Status shows "Pending"
4. **Expected:**  Requested time is correct
5. **Expected:**  Home name is displayed

Test 3: Operator Notification (Future)

1. Login as Operator
2. Check notifications
3. **Expected:**  New tour request notification

Test 4: Error Handling

1. Try submitting tour with invalid date
2. **Expected:**  Validation error appears
3. Try submitting without selection
4. **Expected:**  "Please select a time slot" error

API Testing (Postman/curl)

```
# Test tour request creation
curl -X POST https://carelinkai.onrender.com/api/family/tours/request \
-H "Content-Type: application/json" \
-H "Cookie: <session-token>" \
-d '{
  "homeId": "cm4sho42f00014wy76q4n0zxu",
  "requestedTimes": ["2025-08-02T10:00:00.000Z"],
  "familyNotes": "Test tour request"
}'

# Expected Response:
{
  "success": true,
  "tourRequest": {
    "id": "cm4xxx...",
    "homeId": "cm4sho42f00014wy76q4n0zxu",
    "homeName": "Sunrise Senior Living",
    "status": "PENDING",
    "requestedTimes": ["2025-08-02T10:00:00.000Z"],
    "familyNotes": "Test tour request",
    "createdAt": "2025-12-16T..."
  }
}
```

Deployment

Commit

```
git add src/app/api/family/tours/request/route.ts
git commit -m "🐛 Fix: Tour submission bug - JSON serialization error"
git push origin main
```

Commit Hash: 7c99d78

Render Deployment

- **Status:** Deployed automatically via GitHub webhook
- **Build Time:** ~5-7 minutes
- **Health Check:** <https://carelinkai.onrender.com/health>

Verification Steps

1. Wait for Render deployment to complete
 2. Check Render logs for new tour requests
 3. Test tour submission as family user
 4. Verify tours appear in database
 5. Check "My Tours" page shows created tours
-

Rollback Plan

If issues persist after deployment:

Option 1: Revert Commit

```
git revert 7c99d78
git push origin main
```

Option 2: Deploy Previous Version

```
git reset --hard cb286b2 # Previous working commit
git push origin main --force
```

Option 3: Manual Fix

If tours are still failing, check:

1. Database connection (Prisma client)
2. Authentication (NextAuth session)
3. Permissions (FAMILY role has TOURS_REQUEST)
4. OpenAI API key (for AI suggestions)

Related Documentation

- **Feature Implementation:** `FEATURE_3_TOUR_SCHEDULING.md`
- **Database Schema:** `prisma/schema.prisma` (Lines 2475-2519)
- **API Route:** `src/app/api/family/tours/request/route.ts`
- **Frontend Component:** `src/components/tours/TourRequestModal.tsx`
- **AI Scheduler:** `src/lib/tour-scheduler/ai-tour-scheduler.ts`
- **Permissions:** `src/lib/permissions.ts` (Lines 50-57)

Success Criteria

- [x] **Root cause identified:** JSON serialization error
 - [x] **Fix implemented:** Keep ISO strings instead of Date objects
 - [x] **Logging added:** Comprehensive error tracking
 - [x] **Code committed:** Pushed to GitHub
 - [] **Deployment verified:** Waiting for Render build
 - [] **End-to-end test:** Family can submit tours
 - [] **Database verification:** Tours saved correctly
 - [] **UI confirmation:** “My Tours” shows tours
-



Next Steps

1. Monitor Render Deployment

- Check build logs
- Verify deployment success
- Monitor for errors

2. Test Tour Submission

- Login as family user
- Submit test tour request
- Verify success message

3. Verify Database

- Check TourRequest records
- Verify data format
- Confirm relationships

4. Update Documentation

- Mark bug as resolved
 - Update FEATURE_3_TOUR_SCHEDULING.md
 - Document troubleshooting tips
-




Author

Bug Fix: DeepAgent (Abacus.AI)

Date: December 16, 2025

Priority: CRITICAL

Status:  FIXED (Awaiting Deployment Verification)