

# Feature #3: AI Tour Scheduling Assistant

---

## Overview

---

Comprehensive tour scheduling system with AI-powered time suggestions, automated notifications, and calendar management for CareLinkAI platform.

## Completion Status

---

### COMPLETED COMPONENTS:

#### 1. Database Schema ✓

- **Location:** `prisma/schema.prisma`
- Added `TourRequest` model with full fields
- Added `TourSlot` model for availability management
- Created `TourStatus` enum (PENDING, CONFIRMED, COMPLETED, CANCELLED, RESCHEDULED, NO\_SHOW)
- Created `TourOutcome` enum (SHOWED\_UP, NO\_SHOW, CONVERTED, NOT\_CONVERTED)
- Added relations to Family, Operator, and AssistedLivingHome models
- **Migration:** `prisma/migrations/20251216225759_add_tour_scheduling_feature/migration.sql`

#### 2. AI Scheduling Service ✓

- **Location:** `src/lib/tour-scheduler/ai-tour-scheduler.ts`
- Implements `suggestOptimalTimes()` function
- Uses OpenAI GPT-4o-mini for intelligent suggestions
- Analyzes historical tour data (last 90 days)
- Considers:
  - Home availability and tour slots
  - Conversion rates by time/day
  - Popular times and patterns
  - Scheduled conflicts
  - Family preferences
- Returns top 5 suggestions with scores and reasoning
- Includes fallback logic when OpenAI unavailable

#### 3. Notification Service ✓

- **Location:** `src/lib/notifications/tour-notifications.ts`
- Console-based email logging (MVP approach)
- Functions:
  - `sendTourConfirmationEmail()` - Family & operator notifications
  - `sendTour24HourReminder()` - 24h before tour
  - `sendTour2HourReminder()` - 2h before tour
  - `sendTourCancellationEmail()` - Cancellation notices
  - `sendTourReschedulingEmail()` - Rescheduling updates
- Professional HTML email templates included

- Ready for production email service integration

#### 4. Permissions System ✓

- **Location:** `src/lib/permissions.ts`
- Added 7 new tour permissions:
  - `TOURS_REQUEST` - Family can request tours
  - `TOURS_VIEW` - View own tours
  - `TOURS_VIEW_ALL` - View all tours (operator/admin)
  - `TOURS_CONFIRM` - Confirm tour requests
  - `TOURS_RESCHEDULE` - Reschedule tours
  - `TOURS_CANCEL` - Cancel tours
  - `TOURS_MANAGE_SLOTS` - Manage availability slots
- Updated FAMILY role permissions
- Updated OPERATOR role permissions
- Full RBAC enforcement

#### 5. API Endpoints ✓

##### Family Endpoints:

1. `POST /api/family/tours/request` - Request new tour
  - Validates home exists
  - Creates tour request with status PENDING
  - Sends notifications
  - Returns tour details
1. `GET /api/family/tours/available-slots/[homeId]` - Get AI suggestions
  - Accepts date range parameters
  - Returns AI-analyzed optimal times
  - Includes scores and reasoning
2. `GET /api/family/tours` - List family's tours
  - Supports status filtering
  - Returns all tours for authenticated family
  - Includes home details

##### Operator Endpoints:

4. `POST /api/operator/tours/[id]/confirm` - Confirm tour
  - Sets confirmed time
  - Changes status to CONFIRMED
  - Sends confirmation emails
  - Adds operator notes
1. `POST /api/operator/tours/[id]/reschedule` - Reschedule tour
  - Updates confirmed time
  - Changes status to RESCHEDULED
  - Notifies all parties
  - Includes reason
2. `POST /api/operator/tours/[id]/cancel` - Cancel tour
  - Sets status to CANCELLED
  - Records cancellation details

- Notifies affected parties
  - Supports family or operator cancellation
3. GET /api/operator/tours - List operator's tours
- Filtered by operator ID
  - Supports status and home filtering
  - Includes family details

#### Shared Endpoint:

8. GET /api/tours/[id] - Get tour details
- Permission-based access control
  - Returns full tour information
  - Hides sensitive data based on role
- 



## PENDING COMPONENTS (To be completed):

---

### Frontend UI Components:

#### Family Side:

- [ ] **Tour Request Modal** (src/components/tours/TourRequestModal.tsx)
  - Multi-step wizard (Select times → Add notes → Confirm)
  - Display AI suggestions with scores
  - Time selection interface
  - Notes input field
- [ ] **Family Tours Page** (src/app/dashboard/tours/page.tsx)
  - List upcoming/past tours
  - Status badges
  - Cancel/reschedule buttons
  - Tour details view

#### Operator Side:

- [ ] **Operator Tours Dashboard** (src/app/operator/tours/page.tsx)
  - Tabs: Pending, Confirmed, Completed, Cancelled
  - List view with filters
  - Quick actions (Confirm, Decline, Reschedule)
  - Tour counts and analytics
- [ ] **Tour Detail Page** (src/app/operator/tours/[id]/page.tsx)
  - Family information
  - Home details
  - Requested times display
  - Confirm/decline/reschedule forms
  - Notes management
- [ ] **Calendar View Component** (optional enhancement)

- Visual calendar of scheduled tours
- Drag-and-drop rescheduling
- Conflict detection

## Integration:

- [ ] Add “Schedule Tour” button to home detail pages
- [ ] Add navigation links in sidebar menus
- [ ] Add tour count to operator dashboard
- [ ] Update home listing cards with tour CTA



## Testing Checklist:

- [ ] Test tour request flow (family perspective)
- [ ] Verify AI suggestions are reasonable and conflict-free
- [ ] Test tour confirmation (operator perspective)
- [ ] Test tour cancellation (both roles)
- [ ] Test tour rescheduling
- [ ] Verify notifications are sent correctly
- [ ] Check RBAC enforcement on all endpoints
- [ ] Test edge cases:
- [ ] Conflicting tour times
- [ ] Past date selections
- [ ] Missing required fields
- [ ] Unauthorized access attempts
- [ ] OpenAI API failure handling



## API Documentation:

### Family Tour Request

```
POST /api/family/tours/request
Body: {
  homeId: string,
  requestedTimes: string[], // ISO 8601 datetime strings
  familyNotes?: string
}
Response: {
  success: boolean,
  tourRequest: TourRequest
}
```

## Get AI Suggestions

```
GET /api/family/tours/available-slots/[homeId]?startDate=...&endDate=...
Response: {
  success: boolean,
  suggestions: Array<{
    dateTime: Date,
    dayOfWeek: string,
    timeSlot: string,
    score: number,
    reasoning: string
  }>
}
```

## Confirm Tour

```
POST /api/operator/tours/[id]/confirm
Body: {
  confirmedTime: string, // ISO 8601
  operatorNotes?: string
}
Response: {
  success: boolean,
  tour: TourRequest
}
```



## Environment Variables:

Required for AI features:

```
OPENAI_API_KEY=sk-...
NEXT_PUBLIC_OPENAI_API_KEY=sk-... # Optional fallback
```



## Deployment Notes:

### 1. Database Migration:

```
bash
npx prisma migrate deploy
```

### 2. Environment Setup:

- Ensure OpenAI API key is set in production
- Configure email service (currently console logging)

### 3. Prisma Client:

```
bash
npx prisma generate
```

### 4. Build & Deploy:

```
bash
```

```
npm run build
```

```
# Auto-deploys on Render when pushed to main
```



## Future Enhancements:

---

### 1. Email Integration:

- Replace console logging with SendGrid/AWS SES
- Add email templates management
- Track email delivery status

### 2. SMS Notifications:

- Add Twilio integration for SMS reminders
- Support SMS opt-in/opt-out

### 3. Calendar Sync:

- Google Calendar integration
- iCal export functionality
- Outlook calendar sync

### 4. Advanced Analytics:

- Tour conversion tracking
- Optimal time analysis
- No-show prediction
- Seasonal pattern detection

### 5. Automated Reminders:

- Background job scheduler (Bull, AWS SQS)
- Scheduled reminder delivery
- Follow-up automation

### 6. Virtual Tours:

- Video conferencing integration (Zoom, Google Meet)
- Virtual tour scheduling
- Recording and playback



## Demo Accounts:

---

Test the feature using:

- **Family:** demo.family@carelinkai.test / DemoUser123!
- **Operator:** demo.operator@carelinkai.test / DemoUser123!
- **Admin:** demo.admin@carelinkai.test / DemoUser123!



## Code Quality:

---

- ✓ Type-safe with TypeScript
- ✓ Zod validation on API endpoints
- ✓ Error handling and logging

- ✓ RBAC enforcement
  - ✓ Comprehensive comments
  - ✓ Follows project conventions
  - ✓ Database relations properly defined
  - ✓ Fallback logic for AI failures
- 



## Known Issues:

---

1. **Email Service:** Currently using console logging (MVP)
    - **Resolution:** Integrate production email service
  2. **Reminder Scheduling:** No background job scheduler yet
    - **Resolution:** Add Bull/AWS SQS for scheduled jobs
  3. **Time Zone Handling:** All times in UTC
    - **Resolution:** Add user timezone preferences
- 



## Support:

---

For questions or issues:

- GitHub Issues: [profyt7/carelinkai](https://github.com/profyt7/carelinkai)
  - Deployed URL: <https://carelinkai.onrender.com>
  - Documentation: See project README.md
- 

**Status:** Backend Complete, Frontend In Progress

**Last Updated:** December 16, 2025

**Version:** 1.0.0