



10-DAY ANGULAR IMPLEMENTATION PLAN

Phase 3 HBS - LuxuryStay Project

Objective: Complete all Angular-only modules in 10 days based on backend permission requirements

Backend Permission Model:

- Scopes like `ROOM_MANAGEMENT:WRITE` , `BOOKING:WRITE` , etc.
- Role-based access (super_admin, normal_admin, etc.)
- Permission-based feature access



DAILY BREAKDOWN

DAY 1: Foundation & Setup ⚙️

Theme: Architecture Setup & Route Guards

Tasks:

- 1. Create Route Guard Service** (2 hours)
 - Implement `PermissionGuard` based on backend scopes
 - Scopes to handle: `ROOM_MANAGEMENT:*` , `BOOKING:*` , `ADMIN:*` , etc.
 - Create `@HasPermission('SCOPE')` decorator
- 2. Setup Module Structure** (1.5 hours)
 - Create `shared/modules` directory
 - Create `@auth-guard` interceptor
 - Create permission resolver service
- 3. Create Core Guards & Interceptors** (1.5 hours)
 - `auth.guard.ts` - Check if user is logged in
 - `permission.guard.ts` - Check specific permission scopes
 - `admin.guard.ts` - Check if user is admin
 - HTTP interceptor for token attachment

4. **Setup Pipes & Utilities** (1 hour)

- Create `*appHasPermission` structural directive
- Create permission check utility functions

Deliverables:

- ☒ Route protection system
- ☒ Permission validation pipes & directives
- ☒ Core guards in place

Dev Days Remaining: 9

DAY 2: Admin Management Module

Theme: Admin User Management Interface

Scope: `ADMIN_CREATION:*` permissions

Tasks:

1. **Admin List Component** (2 hours)

- Display all admin users
- Filters: role, status, created date
- Search functionality
- Protected route: `ADMIN_CREATION:READ`

2. **Admin Creation Form** (2 hours)

- Form with: email, name, role selection
- Role dropdown (super_admin, normal_admin, content_admin, etc.)
- Protected button: `ADMIN_CREATION:WRITE`
- Form validation

3. **Admin Edit Component** (1 hour)

- Edit admin details
- Change role assignment
- Protected: `ADMIN_CREATION:WRITE`

Deliverables:

- ☒ Admin List page
- ☒ Admin Create form

- ☒ Admin Edit form with permission guards

Dev Days Remaining: 8

DAY 3: Room Management Module (Part 1)

Theme: Room CRUD Operations

Scope: ROOM_MANAGEMENT:* permissions

Tasks:

1. **Room List Component** (1.5 hours)
 - Display all rooms in table format
 - Columns: Room ID, Type, Price, Status, Availability
 - Filters: type, price range, status
 - Protected: ROOM_MANAGEMENT:READ (admin only view)
2. **Room Create Form** (2.5 hours)
 - Form fields: name, type, price, capacity, description
 - Image upload (use backend image service)
 - Room status selection
 - Protected: ROOM_MANAGEMENT:WRITE
3. **Amenity Selection Component** (1 hour)
 - Multi-select for amenities
 - Checkbox list of available amenities
 - Preview of selected amenities

Deliverables:

- ☒ Room List page (admin)
- ☒ Room Create form
- ☒ Amenity selector component

Dev Days Remaining: 7

DAY 4: Room Management Module (Part 2)

Theme: Room Details & Editing

Scope: ROOM_MANAGEMENT:* permissions

Tasks:

1. **Room Details Component** (2 hours)
 - Full room information display
 - Amenities list
 - Image gallery
 - Availability calendar
2. **Room Edit Form** (1.5 hours)
 - Edit all room properties
 - Update amenities
 - Update images
 - Protected: ROOM_MANAGEMENT:WRITE
3. **Room Delete/Archive** (1 hour)
 - Delete confirmation modal
 - Soft delete (status change to inactive)
 - Protected: ROOM_MANAGEMENT:DELETE
4. **Amenity Management** (0.5 hours)
 - Create amenity
 - Manage amenity list (admin)
 - Protected: ROOM_MANAGEMENT:WRITE

Deliverables:

- ☒ Room Details page
- ☒ Room Edit form
- ☒ Delete/Archive functionality
- ☒ Amenity CRUD

Dev Days Remaining: 6

DAY 5: Booking Management Module (Part 1)

Theme: Booking Creation & Customer Booking View

Scope: BOOKING:* permissions

Tasks:

1. **Customer Booking Page** (2 hours)

- Search rooms by filters (date, type, price)
- Room availability display
- Room cards with images
- Protected: `BOOKING:READ` for customers

2. **Book Room Form** (2 hours)

- Select check-in/out dates (date picker)
- Guest information
- Booking confirmation
- Protected: `BOOKING:WRITE`

3. **Booking Confirmation** (1 hour)

- Display booking summary
- Booking ID generation
- Email confirmation trigger

Deliverables:

- ☒ Room booking search page
- ☒ Book room form
- ☒ Booking confirmation page

Dev Days Remaining: 5

DAY 6: Booking Management Module (Part 2)

Theme: Admin Booking Management

Scope: `BOOKING:*` permissions

Tasks:

1. **Admin Booking List** (2 hours)

- All bookings with filters (status, customer, date)
- Search by booking ID or customer name
- Protected: `BOOKING:READ` (admin only)

2. **Admin Modify Booking** (1.5 hours)

- Edit booking dates, room, guest info
- Change booking status
- Protected: `BOOKING:WRITE`

3. **My Bookings (Customer)** (1 hour)

- Customer view of their bookings
- Edit own bookings
- Cancel booking (triggers refund)
- Protected: `BOOKING:READ` for own bookings

4. **Individual Booking Details** (0.5 hours)

- Full booking information
- Invoice display
- Status timeline

Deliverables:

- ☒ Admin booking list & management
- ☒ Customer my bookings page
- ☒ Booking detail view

Dev Days Remaining: 4

DAY 7: Payment & Refund Modules

Theme: Payment Processing & Refund Management

Scope: `BOOKING:WRITE` , `REFUND:*` permissions

Tasks:

1. **Payment Processing Page** (1.5 hours)

- Display total amount to pay
- Payment method selection
- Process payment button
- Protected: `BOOKING:WRITE`

2. **Payment History** (1 hour)

- List of all payments
- Filter by status, date, amount
- Invoice view/download
- Protected: `BOOKING:READ`

3. **Refund Request Form** (1 hour)

- Create refund request
- Select booking

- Enter refund reason
- Protected: REFUND:WRITE

4. **Admin Refund Management** (1 hour)

- All refund requests list
- Approve/Reject refunds
- Protected: REFUND:APPROVE

5. **My Refunds (Customer)** (0.5 hours)

- Track refund status
- View refund details

Deliverables:

- ☒ Payment processing page
- ☒ Payment history
- ☒ Refund request form
- ☒ Admin refund management
- ☒ Customer refund tracking

Dev Days Remaining: 3

DAY 8: Support & Notification Modules

Theme: Issue Management & Notifications

Scope: ISSUE:* , NOTIFICATION:* permissions

Tasks:

1. **Report Issue Form** (1 hour)

- Issue category selection
- Description input
- File attachments (if needed)
- Protected: ISSUE:WRITE

2. **Issue Management (Admin)** (1.5 hours)

- All issues list with status
- Filter by status, priority, category
- Edit issue status and priority
- Protected: ISSUE:MANAGE

3. **My Issues (Customer)** (1 hour)

- Track personal issues
- View issue updates
- Protected: `ISSUE:READ` (own issues)

4. **Notification Center** (1 hour)

- Real-time notification display (WebSocket integration)
- Mark as read/unread
- Filter by type
- Protected: `NOTIFICATION:READ`

5. **Toast Notification System** (0.5 hours)

- Integrate toast service for all components
- Success, error, warning notifications

Deliverables:

- ☒ Issue reporting system
- ☒ Admin issue management
- ☒ Customer issue tracking
- ☒ Notification center
- ☒ Toast notifications throughout app

Dev Days Remaining: 2

DAY 9: Reporting & Analytics

Theme: Reports, Audit Logs, Dashboard

Scope: `REPORT:*` , `AUDIT:READ` , `ADMIN:READ` permissions

Tasks:

1. **Report Generation Page** (1.5 hours)

- Select report type (booking, revenue, customer, etc.)
- Date range selector
- Filter options
- Generate report button
- Protected: `REPORT:WRITE`

2. **Report Viewer** (1.5 hours)

- Display reports in formatted view
- Charts and graphs (using Chart.js/ng2-charts)

- Export to PDF/Excel
- Protected: REPORT:READ

3. **Admin Dashboard** (1 hour)

- Key metrics (total bookings, revenue, customers)
- Recent bookings chart
- Occupancy rate chart
- Protected: ADMIN:READ

4. **Audit Logs Viewer** (0.5 hours)

- Activity log display
- Filter by user, action, date
- Protected: AUDIT:READ

Deliverables:

- ☒ Report generation system
- ☒ Report viewer with export
- ☒ Admin dashboard with analytics
- ☒ Audit logs viewer

Dev Days Remaining: 1

DAY 10: Polish, Testing & Deployment ✨

Theme: Quality Assurance, Integration & Final Touches

Tasks:

1. **Component Testing** (1.5 hours)

- Write unit tests for critical components
- Test permission guards
- Test form validations

2. **Integration Testing** (1 hour)

- Test API calls with backend
- Test permission scenarios
- Test role-based access

3. **UI/UX Polish** (1 hour)

- Fix responsive design issues
- Consistent styling across modules

- Loading states and skeletons

4. **Error Handling** (1 hour)

- Global error handler for API failures
- User-friendly error messages
- Fallback pages (404, 403, 500)




5. **Documentation** (0.5 hours)

- Component usage documentation
- API integration guide
- Permission reference guide

6. **Final Deployment Preparation** (0.5 hours)

- Build production bundle
- Environment configuration
- Final bug fixes








Deliverables:


-  All modules tested
-  Production-ready build
-  Complete documentation

Dev Days Remaining: 0 



FEATURE COMPLETION TRACKER

Module	Day	Status	Tests
Foundation & Guards	1		✓
Admin Management	2		✓
Room Management (CRUD)	3-4		✓
Booking Management	5-6		✓
Payment & Refund	7		✓
Issues & Notifications	8		✓
Reports & Audit	9		✓

Module	Day	Status	Tests
Testing & Polish	10		✓



PERMISSION SCOPES BY MODULE

DAY 1 - Foundation

- └─ No specific scopes (setup phase)

DAY 2 - Admin Management

- └─ ADMIN_CREATION:READ
- └─ ADMIN_CREATION:WRITE
- └─ ADMIN_CREATION:DELETE
- └─ ADMIN_CREATION:MANAGE

DAY 3-4 - Room Management

- └─ ROOM_MANAGEMENT:READ
- └─ ROOM_MANAGEMENT:WRITE
- └─ ROOM_MANAGEMENT:DELETE
- └─ ROOM_MANAGEMENT:MANAGE

DAY 5-6 - Booking Management

- └─ BOOKING:READ
- └─ BOOKING:WRITE
- └─ BOOKING:CANCEL

DAY 7 - Payment & Refund

- └─ BOOKING:WRITE (payment)
- └─ REFUND:WRITE
- └─ REFUND:APPROVE
- └─ REFUND:READ

DAY 8 - Issues & Notifications

- └─ ISSUE:READ
- └─ ISSUE:WRITE
- └─ ISSUE:MANAGE
- └─ NOTIFICATION:READ

DAY 9 - Reports

- └─ REPORT:READ
- └─ REPORT:WRITE
- └─ AUDIT:READ

DELIVERABLES SUMMARY

Angular Components to Create (50+):

- ☒ 12 Route Guard/Interceptor files
- ☒ 40+ Feature components (list, form, detail, admin views)
- ☒ 15+ Shared utility services
- ☒ 20+ Reusable UI components

Services to Implement:

- ☒ PermissionService
- ☒ RoomService
- ☒ BookingService
- ☒ PaymentService
- ☒ RefundService
- ☒ IssueService
- ☒ NotificationService
- ☒ ReportService
- ☒ AuditService

Modules Structure:

```
src/app/features/  
├─ admin-management/  
├─ room-management/  
├─ booking-management/  
├─ payment-refund/  
├─ issue-notification/  
├─ reports-analytics/  
└─ shared/  
    ├─ guards/  
    ├─ interceptors/  
    ├─ directives/  
    ├─ pipes/  
    └─ services/
```

KEY IMPLEMENTATION TIPS

Permission Checking Pattern:

```
// In component
canCreateRoom = this.permissionService.hasPermission('ROOM_MANAGEMENT:WRITE');

// In route guard
@HasPermission('ROOM_MANAGEMENT:READ')

// In template
<button *appHasPermission="'ROOM_MANAGEMENT:WRITE'">Create Room</button>
```

API Call Pattern:

```
// Backend returns 403 if permission denied
this.roomService.createRoom(data).subscribe(
  (response) => { /* success */ },
  (error) => {
    if (error.status === 403) {
      this.toastService.error('You do not have permission to create rooms');
    }
  }
);
```

Role-Based UI Pattern:

```
isAdmin = this.authService.user.role.includes('admin');
isSuperAdmin = this.authService.user.role === 'super_admin';
```

SUCCESS CRITERIA

- ✓ All 15 Angular-only modules implemented
- ✓ Permission guards on all protected routes
- ✓ CRUD operations for all data modules
- ✓ Responsive design for mobile & desktop

- ✓ Error handling & validation
- ✓ Unit tests (minimum 70% coverage)
- ✓ Integration with backend APIs
- ✓ Production-ready build



NOTES

1. **Backend Already Done:** All backend endpoints exist and are protected with permission scopes
2. **Permission Model:** Granular permission system (`MODULE:ACTION`)
3. **Role System:** `super_admin`, `normal_admin`, `content_admin`, `BACKUP_ADMIN`, `customer`
4. **Data Persistence:** All data stored in PostgreSQL via backend
5. **Caching:** Backend handles Redis caching (consider implementing client-side cache)
6. **Image Management:** Backend has image service (Cloudinary integration)



DAILY STAND-UP CHECKLIST

Each morning, verify:

- ☐ All previous day components integrated
- ☐ New routes added to routing module
- ☐ Permission guards attached
- ☐ API calls tested
- ☐ UI responsive on mobile
- ☐ Error messages user-friendly
- ☐ Components documented