🗓️ Progress Tracking - Smart Notification Service

This folder tracks day-wise progress for the Smart Notification Service project.

📊 Overall Progress

Day	Date	Focus Area	Status	Time Spent
Day 1	Nov 2, 2025	Foundation - Entities & Repositories	✅ Complete	4 hours
Day 2	TBD	DTOs & Service Layer	⏳ Pending	-
Day 3	TBD	Email Integration (SendGrid)	⏳ Pending	-
Day 4	TBD	Rate Limiting (Redis)	⏳ Pending	-
Day 5	TBD	REST API & Controllers	⏳ Pending	-
Day 6	TBD	Async Processing & Retry Logic	⏳ Pending	-
Day 7	TBD	Testing & Documentation	⏳ Pending	-
Day 8	TBD	Frontend Setup (React)	⏳ Pending	-
Day 9	TBD	Frontend UI Components	⏳ Pending	-
Day 10	TBD	Deployment & Final Testing	⏳ Pending	-

📈 Metrics Summary

Code Statistics

- **Total Files**: 7 Java files + 1 test file
- **Lines of Code**: ~600
- **Test Coverage**: 100% (4/4 tests passing)

- **Build Status**: ✅ SUCCESS

Database

- **Tables**: 2 (notifications, templates)
- **Indexes**: 5
- **Relationships**: 1 foreign key

Features Completed

- ✅ Core domain model
- ✅ Repository layer with custom queries
- ✅ Basic CRUD tests
- ⏳ Service layer (Day 2)
- ⏳ REST API (Day 5)
- ⏳ Email integration (Day 3)

🎯 Current Sprint: Backend Development (Days 1-7)

Completed ✅

- ☑ Day 1: Foundation setup with entities and repositories

In Progress 🔄

- ☐ Day 2: DTOs and service layer

Upcoming ⏳

- ☐ Day 3: Email integration
- ☐ Day 4: Rate limiting
- ☐ Day 5: REST API
- ☐ Day 6: Async processing
- ☐ Day 7: Testing

📁 Daily Progress Files

Day 1 - Foundation Setup

Status: ✅ Completed
Highlights:

- Created 3 enums (NotificationStatus, NotificationChannel, NotificationPriority)
- Implemented 2 entities (Notification, Template) with 26 total fields
- Built 2 repositories with 17 custom query methods
- Wrote 4 tests - all passing ✅
- Fixed H2 database JPQL compatibility issues

🔗 Quick Links

- [Main Project Plan](../../# 🔔 Smart Notification Service.md)
- Build Status
- Day 1 Progress

📝 Notes

Key Decisions

1. **Database**: Using H2 for development, PostgreSQL (Supabase) for production
2. **Architecture**: Starting with monolithic Spring Boot, can extract to microservices later
3. **Testing Strategy**: Write tests alongside implementation (not at the end)

Blockers

- None currently

Next Session

- Start Day 2: Create DTOs and implement service layer