**PDL IMS - Project Status & Roadmap**

**📊 Current Status: ~60% Complete**

**✅ COMPLETED MODULES (Phases 1-4)**

**Phase 1: Foundation ✓**

* [x] Project setup with Django 5.0.1
* [x] PostgreSQL database configuration
* [x] Environment setup (.env, settings)
* [x] Git repository initialized

**Phase 2: Core Module ✓**

* [x] Company model
* [x] Factory model
* [x] Department model
* [x] Location hierarchy (Warehouse → Building → Floor → Room → Rack → Bin)
* [x] Validation triggers for location depth
* [x] Soft delete support
* [x] Audit fields (created\_at, updated\_at, created\_by, etc.)

**Phase 3: Users & RBAC ✓**

* [x] Custom User model (AbstractBaseUser)
* [x] Role model (SuperAdmin, Manager, Storekeeper, etc.)
* [x] Permission model (granular permissions)
* [x] UserRole (many-to-many)
* [x] RolePermission (many-to-many)
* [x] StoreType (Yarn Store, IT Store, General Store, etc.)
* [x] UserStoreAccess (store-level permissions)
* [x] UserSession (session tracking)
* [x] Seed data command for roles & permissions
* [x] Helper functions for permission checks

**Phase 4: Inventory Master Data ✓**

* [x] Currency & Exchange Rates
* [x] Unit of Measurement (UOM)
* [x] Item Categories (hierarchical)
* [x] Items (comprehensive with stock control)
* [x] Item-UOM Conversions
* [x] Supplier master
* [x] Item-Supplier mapping
* [x] Supplier Evaluations
* [x] Buyers (RMG-specific)
* [x] Styles (RMG-specific)
* [x] Colors
* [x] Sizes
* [x] Status Master (centralized status management)
* [x] Seed data command for master data

**Phase 5: Procurement ✓**

* [x] Purchase Requests (PR) & Lines
* [x] Purchase Orders (PO) & Lines
* [x] Goods Received Notes (GRN) & Lines
* [x] QC Inspections
* [x] Stock Batches (FIFO valuation support)
* [x] Stock Movements (single source of truth)
* [x] Auto-number generation for PR, PO, GRN
* [x] Auto-calculation of totals, pending quantities
* [x] Helper functions (get\_current\_stock, get\_item\_valuation)

**Phase 6: Production & Operations ✓**

* [x] Work Orders with size/color breakdowns
* [x] Yarn & Accessory Requirements
* [x] Internal Requisitions & Lines
* [x] Stock Issues & Lines
* [x] Stock Transfers & Lines
* [x] Inventory Adjustments & Lines
* [x] Stocktaking Sessions & Lines
* [x] Auto-number generation
* [x] Pending quantity calculations

**Phase 7: Asset Management ✓ (JUST COMPLETED!)**

* [x] Asset model (IT equipment, machinery, serialized items)
* [x] Asset Assignment History
* [x] Asset Maintenance Records
* [x] Asset Depreciation Schedule
* [x] Auto-create assets from GRN (via signals)
* [x] Depreciation calculation (Straight-line, Declining Balance)
* [x] Warranty tracking
* [x] Maintenance scheduling
* [x] Django admin interface
* [x] Management commands:
  + [x] generate\_depreciation
  + [x] check\_maintenance\_due
  + [x] check\_warranty\_expiring
  + [x] asset\_report

**🚧 REMAINING MODULES (Phases 8-14)**

**Phase 8: Approval Workflows (NEXT PRIORITY) ⏳**

**Estimated Time: 3-4 days**

* [ ] ApprovalWorkflow model
* [ ] ApprovalWorkflowStep model
* [ ] Approval model (instance tracking)
* [ ] ApprovalHistory model
* [ ] Integration with existing models:
  + [ ] PR approval workflow
  + [ ] PO approval workflow
  + [ ] GRN approval workflow
  + [ ] Adjustment approval workflow
  + [ ] Transfer approval workflow
  + [ ] Asset assignment approval
* [ ] Approval engine logic
* [ ] Timeout & escalation handling
* [ ] Parallel approvals support
* [ ] Conditional approvals (value thresholds)
* [ ] Django admin interface
* [ ] Management commands

**Phase 9: RMG-Specific Operations ⏳**

**Estimated Time: 3-4 days**

**Yarn Store Operations:**

* [ ] YarnAllocation model
* [ ] YarnIssue & YarnIssueLine models
* [ ] YarnGatePass model
* [ ] KnittingProgram model

**Gray Fabric Operations:**

* [ ] GrayFabricReceipt model
* [ ] GrayFabricInspection model
* [ ] GrayFabricIssue model
* [ ] KnittingBill model

**Finish Fabric Operations:**

* [ ] FinishFabricReceipt model
* [ ] FinishFabricIssue model
* [ ] FabricLeftover model
* [ ] DyeingBill model

**Phase 10: Invoices & Payments ⏳**

**Estimated Time: 2 days**

* [ ] Invoice model
* [ ] Invoice verification & approval
* [ ] Payment tracking
* [ ] Outstanding payments report
* [ ] Supplier payment history

**Phase 11: Stock Returns ⏳**

**Estimated Time: 1-2 days**

* [ ] StockReturn model
* [ ] StockReturnLine model
* [ ] Return types:
  + [ ] Return to supplier
  + [ ] Return from department
  + [ ] Return from production
* [ ] Disposition tracking (restock, scrap, repair)

**Phase 12: Reports Module ⏳**

**Estimated Time: 4-5 days**

* [ ] ScheduledReport model
* [ ] ReportExecutionLog model
* [ ] Stock reports:
  + [ ] Stock Balance Report
  + [ ] Stock Movement Ledger
  + [ ] Stock Valuation Report (FIFO)
  + [ ] Aging Analysis
  + [ ] Low Stock Alert Report
  + [ ] Expiry Alert Report
* [ ] Procurement reports:
  + [ ] Pending Purchase Requests
  + [ ] Purchase Order Status
  + [ ] GRN Summary
  + [ ] Supplier Performance
  + [ ] Pending Approvals Dashboard
* [ ] Asset reports:
  + [ ] Asset Register
  + [ ] Depreciation Report
  + [ ] Maintenance Schedule
  + [ ] Warranty Expiry Report
  + [ ] Asset Assignment Report
* [ ] Production reports:
  + [ ] Work Order Status
  + [ ] Material Consumption
  + [ ] Production Efficiency
* [ ] Report scheduler
* [ ] Export functionality (PDF, Excel, CSV)
* [ ] Email delivery

**Phase 13: Supporting Infrastructure ⏳**

**Estimated Time: 2-3 days**

**Notifications:**

* [ ] Notification model
* [ ] UserNotificationPreference model
* [ ] Notification types:
  + [ ] Low stock alerts
  + [ ] Pending approvals
  + [ ] QC pending
  + [ ] Maintenance due
  + [ ] Warranty expiring
* [ ] Notification channels (email, system, SMS)
* [ ] Notification templates

**Audit Logs:**

* [ ] AuditLog model (immutable)
* [ ] Auto-logging via signals
* [ ] Change tracking (before/after values)
* [ ] User activity tracking
* [ ] Search & filter interface

**Attachments & Comments:**

* [ ] Attachment model (generic)
* [ ] Comment model (threaded comments)
* [ ] File upload handling
* [ ] Integration with all major entities

**System Settings:**

* [ ] SystemSetting model
* [ ] Company-level settings
* [ ] User preferences
* [ ] Settings UI in admin

**Phase 14: API Layer (REST) ⏳**

**Estimated Time: 5-7 days**

* [ ] Django REST Framework serializers for all models
* [ ] ViewSets for CRUD operations
* [ ] Custom endpoints for:
  + [ ] Dashboard statistics
  + [ ] Stock availability check
  + [ ] Approval workflows
  + [ ] Report generation
* [ ] Authentication (Token, JWT, Session)
* [ ] Permission classes
* [ ] Pagination & filtering
* [ ] API documentation (Swagger/OpenAPI)
* [ ] API testing suite

**🎨 FRONTEND DEVELOPMENT (Phase 15) ⏳**

**Estimated Time: 6-8 weeks**

**Technology Stack Decision:**

* **Option 1**: React + TypeScript + Tailwind CSS + shadcn/ui
* **Option 2**: Vue.js + TypeScript + Tailwind CSS + Vuetify
* **Option 3**: Next.js (React SSR) + TypeScript + Tailwind CSS

**Frontend Modules:**

**1. Authentication & Profile (1 week)**

* [ ] Login page
* [ ] Password reset
* [ ] User profile
* [ ] Session management

**2. Dashboard (1 week)**

* [ ] Stock overview widgets
* [ ] Pending approvals widget
* [ ] Low stock alerts
* [ ] KPI cards (total assets, total value, etc.)
* [ ] Recent activities
* [ ] Quick actions

**3. Procurement Module (1 week)**

* [ ] Purchase Request (create, list, approve, view)
* [ ] Purchase Order (create, list, track, send to supplier)
* [ ] GRN (receive goods, QC, post to stock)
* [ ] Supplier management

**4. Inventory Module (1 week)**

* [ ] Items master (CRUD)
* [ ] Stock balance view
* [ ] Stock movements history
* [ ] Inventory adjustments
* [ ] Location management

**5. Assets Module (3-4 days)**

* [ ] Asset register (list, filter, search)
* [ ] Asset details view
* [ ] Assignment management
* [ ] Maintenance scheduling
* [ ] Depreciation view

**6. Production Module (1 week)**

* [ ] Work orders (create, track, close)
* [ ] Internal requisitions
* [ ] Stock issues
* [ ] Transfers

**7. Reports Module (1 week)**

* [ ] Report library
* [ ] Pre-built reports
* [ ] Custom query builder
* [ ] Export center (PDF, Excel, CSV)
* [ ] Scheduled reports

**8. Admin Module (3-4 days)**

* [ ] User management
* [ ] Role management
* [ ] Store access management
* [ ] Approval workflows configuration
* [ ] System settings

**📈 Progress Summary**

**Backend Completion: ~60%**

* ✅ Core infrastructure: 100%
* ✅ Master data: 100%
* ✅ Procurement: 100%
* ✅ Production basics: 100%
* ✅ Asset management: 100%
* ⏳ Approval workflows: 0%
* ⏳ RMG operations: 0%
* ⏳ Reports: 0%
* ⏳ Notifications & audit: 0%
* ⏳ API layer: 0%

**Frontend Completion: 0%**

* ⏳ All frontend work pending

**Overall Project: ~30%**

* Backend: 60% complete
* Frontend: 0% complete
* Testing: 20% complete
* Documentation: 40% complete

**🎯 Recommended Next Steps**

**Immediate (This Week):**

1. ✅ **Install Asset Management Module** (Just completed!)
   * Follow INSTALLATION\_GUIDE.md
   * Run tests to verify
   * Populate with test data
2. ⏳ **Start Approval Workflows Module** (Next priority)
   * Design workflow structure
   * Create models
   * Build approval engine
   * Integrate with existing modules

**Short Term (Next 2 Weeks):**

1. ⏳ **Complete RMG Operations**
   * Yarn store ops
   * Gray/Finish fabric ops
   * Bills & payments
2. ⏳ **Build Reports Module**
   * Start with critical reports
   * Add export functionality
   * Build scheduler

**Medium Term (Next Month):**

1. ⏳ **Supporting Infrastructure**
   * Notifications system
   * Audit logs
   * Attachments & comments
2. ⏳ **API Layer**
   * REST API for all modules
   * API documentation
   * Testing suite

**Long Term (Next 2 Months):**

1. ⏳ **Frontend Development**
   * Choose technology stack
   * Build authentication
   * Develop core modules
   * Testing & deployment

**💾 Database Statistics**

**Tables Created: 45+**

* Core: 4 tables
* Users: 8 tables
* Inventory: 12 tables
* Procurement: 8 tables
* Production: 10 tables
* Assets: 4 tables

**Tables Remaining: 15-20**

* Approval: 4 tables
* RMG Operations: 9 tables
* Reports: 2 tables
* Notifications: 2 tables
* Audit: 2 tables
* Supporting: 3-5 tables

**Total Estimated Tables: 60-65**

**📦 Dependencies Status**

**Installed:**

* ✅ Django 5.0.1
* ✅ djangorestframework
* ✅ psycopg2 (PostgreSQL driver)
* ✅ python-decouple
* ✅ Pillow
* ✅ django-cors-headers
* ✅ django-filter
* ✅ python-dateutil (for assets)

**To Install:**

* ⏳ reportlab (for PDF reports)
* ⏳ openpyxl (for Excel export)
* ⏳ celery (for background tasks)
* ⏳ redis (for Celery broker)
* ⏳ django-celery-beat (for scheduled tasks)

**🧪 Testing Status**

**Unit Tests: 20%**

* ⏳ Core models: Pending
* ⏳ Users: Pending
* ⏳ Inventory: Pending
* ⏳ Procurement: Pending
* ⏳ Assets: Pending

**Integration Tests: 0%**

* ⏳ All pending

**API Tests: 0%**

* ⏳ All pending

**📝 Documentation Status**

**Technical Docs: 60%**

* ✅ Database schema (comprehensive)
* ✅ Model docstrings
* ✅ Admin docstrings
* ✅ Installation guides
* ⏳ API documentation (pending)
* ⏳ Deployment guide (pending)

**User Docs: 0%**

* ⏳ User manual
* ⏳ Admin guide
* ⏳ Training materials
* ⏳ Video tutorials

**🚀 Deployment Readiness: 15%**

**Current State:**

* ✅ Development environment setup
* ✅ Database migrations working
* ⏳ Production settings pending
* ⏳ Docker configuration pending
* ⏳ CI/CD pipeline pending
* ⏳ Server deployment pending
* ⏳ Backup strategy pending
* ⏳ Monitoring setup pending

**🎉 Achievements So Far**

1. **Solid Foundation**: Complete core infrastructure
2. **Comprehensive Data Model**: 45+ well-designed tables
3. **RBAC Implementation**: Role-based access control
4. **Auto-numbering**: All documents auto-generate numbers
5. **Stock Tracking**: FIFO-ready stock movement system
6. **Asset Management**: Full lifecycle tracking
7. **Soft Deletes**: Data preservation throughout
8. **Audit Trail**: Created/updated tracking on all models

**🎯 Success Criteria**

**For Beta Release:**

* [x] Core modules complete
* [x] Master data complete
* [x] Procurement complete
* [x] Basic production complete
* [x] Asset management complete
* [ ] Approval workflows complete
* [ ] Reports module complete
* [ ] API layer complete
* [ ] Frontend (basic) complete

**For Production Release:**

* [ ] All modules complete
* [ ] Full testing suite
* [ ] Complete documentation
* [ ] User training completed
* [ ] Performance optimization
* [ ] Security audit
* [ ] Backup & disaster recovery
* [ ] Production deployment
* [ ] Monitoring & alerting

**📞 Next Actions**

**Your Action Items:**

1. **Install Asset Management** (30 minutes)
2. # Follow INSTALLATION\_GUIDE.md
3. cd pdl\_ims\_backend
4. # Create directory structure
5. mkdir -p assets/management/commands
6. mkdir -p assets/migrations
7. # Copy files from artifacts
8. # ... (follow installation guide)
9. # Run migrations
10. python manage.py makemigrations assets
11. python manage.py migrate assets
12. # Test
13. python manage.py check\_maintenance\_due
14. **Test Asset Module** (30 minutes)
    * Create test assets
    * Test assignment workflow
    * Generate depreciation
    * Verify admin interface
15. **Decision Time** (Planning)
    * Choose next module: **Approval Workflows** (recommended)
    * Or: **Reports Module** (business value)
    * Or: **RMG Operations** (industry-specific)
16. **Review & Confirm** (15 minutes)
    * Review PROJECT\_STATUS.md
    * Confirm priorities
    * Set timeline expectations

**💡 Recommendations**

**Priority 1: Approval Workflows (CRITICAL)**

**Why First:**

* Needed by all other modules
* Core business process
* Blocks frontend development
* Required for production use

**Benefits:**

* PR/PO/GRN approvals automated
* Multi-level approval support
* Email notifications
* Audit trail

**Priority 2: Reports Module (HIGH VALUE)**

**Why Important:**

* Immediate business value
* Management visibility
* Compliance requirements
* Decision support

**Benefits:**

* Stock reports daily
* Procurement tracking
* Asset register
* Scheduled delivery

**Priority 3: RMG Operations (INDUSTRY-SPECIFIC)**

**Why Later:**

* Specific to garment industry
* Can work without it initially
* Complex workflows
* Requires domain expertise

**Benefits:**

* Complete yarn-to-garment tracking
* Knitting/dyeing operations
* Bill management
* Industry compliance

**🔮 Future Enhancements (Phase 16+)**

**Advanced Features:**

* [ ] Barcode/QR code scanning
* [ ] Mobile app (React Native/Flutter)
* [ ] Real-time notifications (WebSockets)
* [ ] Advanced analytics & BI dashboards
* [ ] Machine learning for demand forecasting
* [ ] Integration with accounting systems
* [ ] Supplier portal
* [ ] Customer portal
* [ ] Multi-language support
* [ ] Multi-currency advanced features

**Optimizations:**

* [ ] Database query optimization
* [ ] Caching layer (Redis)
* [ ] Background job processing (Celery)
* [ ] CDN for static assets
* [ ] Load balancing
* [ ] Database replication

**📊 Estimated Timeline to Production**

**Conservative Estimate:**

* **Backend Completion**: 4-6 weeks (remaining 40%)
  + Approval Workflows: 1 week
  + RMG Operations: 1 week
  + Reports: 1 week
  + Infrastructure: 1 week
  + API Layer: 1-2 weeks
* **Frontend Development**: 6-8 weeks
  + Setup & Auth: 1 week
  + Core modules: 4-5 weeks
  + Testing & polish: 1-2 weeks
* **Testing & QA**: 2-3 weeks
  + Unit tests
  + Integration tests
  + UAT (User Acceptance Testing)
  + Bug fixes
* **Documentation & Training**: 1-2 weeks
  + User manual
  + Admin guide
  + Training sessions
* **Deployment & Go-Live**: 1 week
  + Server setup
  + Data migration
  + Production deployment
  + Monitoring setup

**Total: 14-20 weeks (3.5-5 months)**

**Aggressive Estimate (with full-time work):**

**Total: 8-12 weeks (2-3 months)**

**🎓 Learning Resources**

**Django & DRF:**

* Official Django Docs: https://docs.djangoproject.com/
* DRF Tutorial: https://www.django-rest-framework.org/tutorial/quickstart/
* Django Best Practices: https://djangobook.com/

**Frontend (React):**

* React Official: https://react.dev/
* TypeScript Handbook: https://www.typescriptlang.org/docs/
* Tailwind CSS: https://tailwindcss.com/docs
* shadcn/ui: https://ui.shadcn.com/

**Deployment:**

* Docker: https://docs.docker.com/
* PostgreSQL: https://www.postgresql.org/docs/
* Nginx: https://nginx.org/en/docs/

**📈 Success Metrics**

**Development Metrics:**

* Code coverage: Target 80%+
* API response time: < 200ms
* Page load time: < 2s
* Database queries: Optimized (< 10 per request)

**Business Metrics:**

* Stock accuracy: 99%+
* Approval cycle time: Reduced by 70%
* Report generation: Automated
* Asset tracking: 100% visibility
* User adoption: 90%+ within 3 months

**🤝 Collaboration Workflow**

**For Each New Module:**

1. **Planning Phase**
   * Review database schema
   * Confirm requirements
   * Design models & relationships
2. **Development Phase**
   * Create models
   * Write admin interface
   * Add signals/hooks
   * Create management commands
3. **Testing Phase**
   * Write unit tests
   * Manual testing
   * Edge case testing
4. **Documentation Phase**
   * Update installation guide
   * Add usage examples
   * Update PROJECT\_STATUS.md
5. **Review & Integration**
   * Code review
   * Integration testing
   * Merge to main branch

**📁 Project Structure (Current)**

pdl\_ims\_backend/

├── core/ ✅ Complete

│ ├── models.py (Company, Factory, Department, Location)

│ ├── admin.py

│ └── migrations/

├── users/ ✅ Complete

│ ├── models.py (User, Role, Permission, StoreAccess)

│ ├── admin.py

│ └── management/

│ └── commands/

│ └── seed\_initial\_data.py

├── inventory/ ✅ Complete

│ ├── models.py (Item, Category, Supplier, Buyer, etc.)

│ ├── admin.py

│ └── management/

│ └── commands/

│ └── seed\_master\_data.py

├── procurement/ ✅ Complete

│ ├── models.py (PR, PO, GRN, QC, StockBatch, StockMovement)

│ └── admin.py

├── production/ ✅ Complete

│ ├── models.py (WorkOrder, Requisition, Issue, Transfer, etc.)

│ └── admin.py

├── assets/ ✅ Just Completed!

│ ├── models.py (Asset, Assignment, Maintenance, Depreciation)

│ ├── admin.py

│ ├── signals.py

│ └── management/

│ └── commands/

│ ├── generate\_depreciation.py

│ ├── check\_maintenance\_due.py

│ ├── check\_warranty\_expiring.py

│ └── asset\_report.py

├── reports/ ⏳ Empty (Next after approvals)

├── manage.py

├── requirements.txt

└── Documentation/

├── IMS\_Database\_Schema\_v1.1.txt

├── Environment\_Setup\_Manual.md

└── PROJECT\_STATUS.md (This file)

**🎯 Decision Point: What's Next?**

You have **3 options** for the next module:

**Option 1: Approval Workflows (RECOMMENDED) ✨**

**Time**: 3-4 days  
**Complexity**: Medium  
**Impact**: High  
**Blocks**: Frontend, Production use

**Why This:**

* Critical for all operations
* Enables multi-level approvals
* Required for compliance
* Unblocks other development

**Deliverables:**

* ApprovalWorkflow model
* ApprovalWorkflowStep model
* Approval & ApprovalHistory models
* Integration with PR/PO/GRN/Assets
* Admin interface
* Notification triggers

**Option 2: Reports Module (HIGH VALUE) 📊**

**Time**: 4-5 days  
**Complexity**: Medium-High  
**Impact**: Very High  
**Blocks**: Management visibility

**Why This:**

* Immediate business value
* Management needs reports ASAP
* Can start manually for now
* Export functionality useful

**Deliverables:**

* Stock Balance Report
* Stock Movement Report
* Valuation Report
* PO Status Report
* Asset Register
* Export to PDF/Excel
* Scheduled reports

**Option 3: RMG Operations (INDUSTRY-SPECIFIC) 👔**

**Time**: 3-4 days  
**Complexity**: High  
**Impact**: Medium  
**Blocks**: Complete RMG workflow

**Why This:**

* Specific to garment industry
* Completes production cycle
* Yarn-to-garment tracking
* Industry compliance

**Deliverables:**

* Yarn store operations
* Gray fabric operations
* Finish fabric operations
* Knitting/Dyeing bills

**🚦 My Recommendation**

**Start with Approval Workflows because:**

1. **Foundation First**: It's infrastructure that everything else needs
2. **Unblocks Progress**: Other modules depend on it
3. **Production Critical**: Can't go live without approvals
4. **Medium Complexity**: Achievable in 3-4 days
5. **High Impact**: Touches all major processes

**Then follow with:**

1. Reports Module (business value)
2. RMG Operations (complete the system)
3. Supporting Infrastructure (notifications, audit)
4. API Layer (enable frontend)
5. Frontend Development (user interface)

**📝 Final Checklist for Asset Module**

Before moving to next module, verify:

* [ ] Assets app installed in INSTALLED\_APPS
* [ ] Migrations created and applied
* [ ] Admin interface accessible at /admin/assets/
* [ ] Can create assets manually
* [ ] Can assign assets to users
* [ ] Can record maintenance
* [ ] Depreciation calculates correctly
* [ ] Management commands work
* [ ] Auto-creation from GRN works
* [ ] Tests pass

**🎊 Congratulations!**

You've completed **7 major modules** representing 60% of the backend!

The system now supports:

* ✅ Complete organizational structure
* ✅ User management & security
* ✅ Master data management
* ✅ Full procurement cycle
* ✅ Production operations
* ✅ Fixed asset tracking

**Ready to proceed with Approval Workflows?** 🚀

Let me know your decision and I'll start building the next module!