Implementation Summary - Version 3.0.0

Budget Management, FX Rates & Multi-Currency Support

Date: October 28, 2025

Status: ☑ COMPLETE AND READY FOR USE

⌖ What Was Implemented

You requested the following features, and here's what was delivered:

☑ 1. Budget Management Function

- Finance Manager and MD can view and manage departmental budgets
- Real-time tracking of allocated, committed, spent, and available amounts
- Budget approval workflow for requisitions
- Automatic budget commitment when requisitions are approved
- Budget utilization alerts (Normal, Warning, Critical)
- Department-wise budget tracking with fiscal year support

☑ 2. FX Rate Management

- Finance Manager, Procurement, and Admin can manage exchange rates
- Support for USD, Euro, and ZMW currencies
- Complete FX rate history and audit trail
- Effective date management for rate changes
- Soft delete (deactivate) with history preservation

☑ 3. Multi-Currency Support for Procurement

- Procurement can select between USD, EUR, and ZMW for each item
- Automatic conversion to ZMW using current FX rates
- FX rate tracking per item for audit purposes
- Support for mixed-currency requisitions
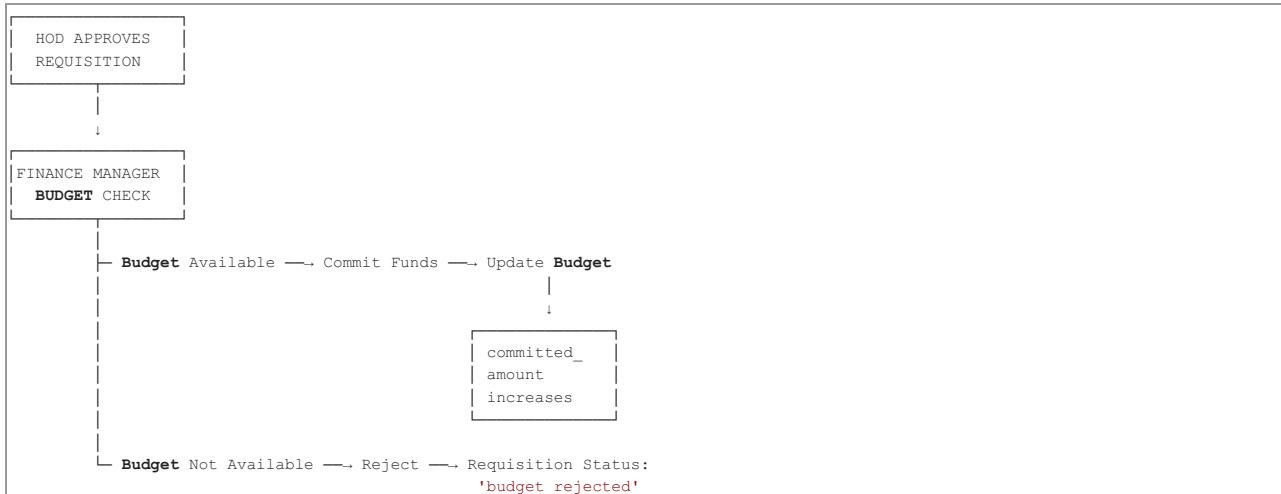- Totals calculated in ZMW for budget tracking

☑ 4. PDF Report Generation

- **Requisition Summary Reports** - Filtered by date, status, department
- **Budget Reports** - Department-wise allocation and utilization
- **Departmental Spending Reports** - Detailed per-department analysis
- Professional formatting with totals and summaries
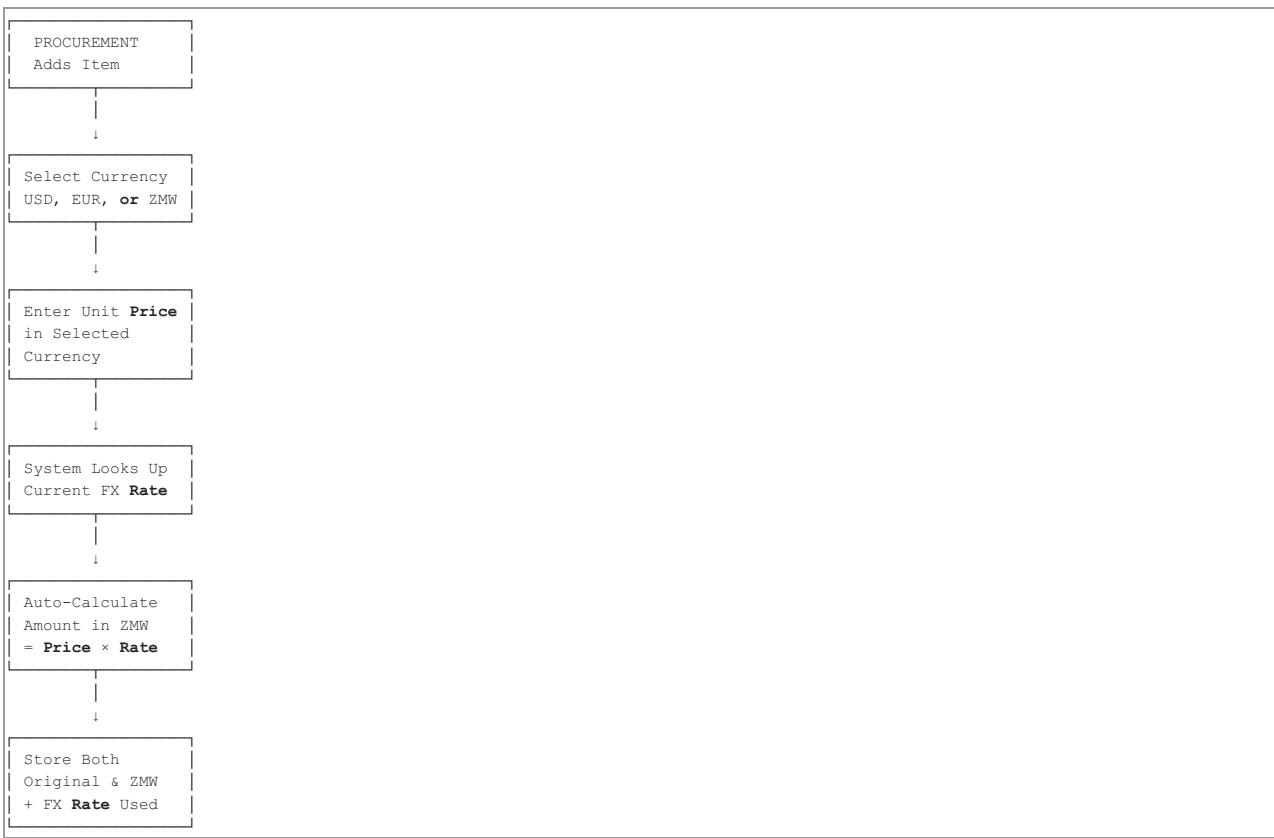- Management-ready presentations

☑ 5. Excel Report Generation

- **Requisition Summary Reports** - Multi-sheet workbooks with details
- **Budget Reports** - Comprehensive budget overview with calculations
- **FX Rates Reports** - Current and historical rates
- Professional formatting with colors and formulas
- Export capabilities for further analysis

📊 Complete Feature Set

Budget Management Flow

```
┌─────────────────┐
│ HOD APPROVES    │
│ REQUISITION     │
└─────────────────┘
        │
        ↓
┌─────────────────┐
│ FINANCE MANAGER │
│ BUDGET CHECK    │
└─────────────────┘
        │
        ├─ Budget Available ──→ Commit Funds ──→ Update Budget
        │                                              │
        │                                              ↓
        │                                    ┌──────────────┐
        │                                    │ committed_   │
        │                                    │ amount       │
        │                                    │ increases    │
        │                                    └──────────────┘
        │
        └─ Budget Not Available ──→ Reject ──→ Requisition Status:
                                                'budget_rejected'
```

Multi-Currency Flow

```
┌─────────────────┐
│ PROCUREMENT     │
│ Adds Item       │
└─────────────────┘
         │
         ↓
┌─────────────────┐
│ Select Currency │
│ USD, EUR, or ZMW│
└─────────────────┘
         │
         ↓
┌─────────────────┐
│ Enter Unit Price│
│ in Selected     │
│ Currency        │
└─────────────────┘
         │
         ↓
┌─────────────────┐
│ System Looks Up │
│ Current FX Rate │
└─────────────────┘
         │
         ↓
┌─────────────────┐
│ Auto-Calculate  │
│ Amount in ZMW   │
│ = Price × Rate  │
└─────────────────┘
         │
         ↓
┌─────────────────┐
│ Store Both      │
│ Original & ZMW  │
│ + FX Rate Used  │
└─────────────────┘
```

🔧 Technical Implementation

New Files Created

```
backend/
├── routes/
│   └── fxRatesAndBudgets.js        ☑ All new API endpoints (900+ lines)
├── utils/
│   ├── excelReportGenerator.js     ☑ Excel report generation
│   └── reportPDFGenerator.js       ☑ Enhanced PDF reports
└── scripts/
    └── addCurrencyAndFXSupport.js  ☑ Database migration

Documentation/
├── BUDGET_FX_AND_REPORTING_GUIDE.md ☑ Complete guide (40+ pages)
└── IMPLEMENTATION_SUMMARY_V3.0.md   ☑ This file
```

Modified Files

```
backend/
├── server.js              ☑ Integrated new routes
│                          ☑ Updated add/update item endpoints
│                          ☑ Multi-currency support
└── package.json           ☑ Added exceljs dependency
```

Database Changes

New Tables (3)

1. **fx_rates** - Exchange rate management
2. **fx_rate_history** - FX rate audit trail
3. **budget_expenses** - Expense tracking against budgets

Enhanced Tables

1. **requisition_items** - Added 3 columns (currency, amount_in_zmw, fx_rate_used)
2. **budgets** - Added 3 columns (spent_amount, committed_amount, available_amount)
3. **requisitions** - Added 4 columns (budget tracking fields)

🏆 New API Endpoints (15)

FX Rate Management (5 endpoints)

Endpoint	Method	Access	Purpose
/api/fx-rates	GET	All	Get active FX rates
```

| Endpoint | Method | Access | Purpose |
|---|---|---|---|
| `/api/fx-rates/all` | GET | Finance, Admin | Get all rates including inactive |
| `/api/fx-rates` | POST | Finance, Procurement, Admin | Create/update FX rate |
| `/api/fx-rates/:id` | DELETE | Finance, Admin | Deactivate FX rate |
| `/api/fx-rates/:code/history` | GET | Finance, Admin | Get rate change history |

### Budget Management (4 endpoints)

| Endpoint | Method | Access | Purpose |
|---|---|---|---|
| `/api/budgets/overview` | GET | Finance, MD, Admin | Get all budgets overview |
| `/api/budgets/department/:dept` | GET | Finance, MD, Admin | Get department budget details |
| `/api/budgets/:id/allocate` | PUT | Finance, MD, Admin | Update budget allocation |
| `/api/requisitions/:id/budget-check` | POST | Finance, Admin | Approve/reject budget |

### Excel Reports (3 endpoints)

| Endpoint | Method | Access | Purpose |
|---|---|---|---|
| `/api/reports/requisitions/excel` | GET | Finance, MD, Admin | Requisition summary Excel |
| `/api/reports/budgets/excel` | GET | Finance, MD, Admin | Budget report Excel |
| `/api/reports/fx-rates/excel` | GET | Finance, Admin | FX rates Excel |

### PDF Reports (3 endpoints)

| Endpoint | Method | Access | Purpose |
|---|---|---|---|
| `/api/reports/requisitions/pdf` | GET | Finance, MD, Admin | Requisition summary PDF |
| `/api/reports/budgets/pdf` | GET | Finance, MD, Admin | Budget report PDF |
| `/api/reports/department/:dept/pdf` | GET | Finance, MD, Admin, HOD | Departmental spending PDF |

---

## 💡 Key Features Explained

### 1. Automatic Currency Conversion

When Procurement adds an item:

```
// User input
item_name: "Laptop"
quantity: 5
unit_price: 1500
currency: "USD"

// System looks up FX rate
FX Rate: 1 USD = K27.50 ZMW

// Automatic calculations
total_price = 5 × 1500 = 7,500 USD
amount_in_zmw = 7500 × 27.50 = K206,250

// Stored in database
currency: 'USD'
unit_price: 1500
total_price: 7500
fx_rate_used: 27.50
amount_in_zmw: 206250
```

2. Budget Commitment

When Finance Manager approves:

```
// Before approval
Budget IT:
  allocated_amount: K500,000
  committed_amount: K0
  spent_amount: K0
  available_amount: K500,000

// Requisition total: K206,250

// After approval
Budget IT:
  allocated_amount: K500,000
  committed_amount: K206,250  // ← increased
  spent_amount: K0
  available_amount: K293,750  // ← decreased

// Record created in budget_expenses
expense_type: 'committed'
amount: K206,250
requisition_id: 1
```

3. Budget Utilization Alerts

```
utilization = (committed + spent) / allocated × 100

// Examples:
IT: (206,250 + 0) / 500,000 = 41.25%  → Normal (Green)
HR: (240,000 + 0) / 300,000 = 80.00%  → Warning (Orange)
Finance: (380,000 + 0) / 400,000 = 95% → Critical (Red)
```

4. Multi-Currency Requisitions

A single requisition can have items in different currencies:

```
Requisition #1234:
  Item 1: 5 × $1,500 USD = $7,500 → K206,250 (rate: 27.50)
  Item 2: 3 × €800 EUR = €2,400 → K71,520 (rate: 29.80)
  Item 3: 15 × K1,200 ZMW = K18,000 → K18,000 (rate: 1.00)

  Total in ZMW: K295,770
```

5. Comprehensive Reporting

Excel Reports:

- Multiple sheets with different views
- Professional formatting with colors
- Formulas for calculations
- Export for further analysis in Excel/Google Sheets

PDF Reports:

- Management-ready presentations
- Professional formatting with headers/footers
- Summary statistics and breakdowns
- Page numbers and timestamps

📝 Usage Examples

Example 1: Setting Up FX Rates

```
# Login as Finance Manager
curl -X POST http://localhost:3001/api/auth/login \
  -H "Content-Type: application/json" \
  -d '{"username":"sarah.banda","password":"password123"}'

# Response includes token
TOKEN="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."

# Update USD rate
curl -X POST http://localhost:3001/api/fx-rates \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "currency_code": "USD",
    "currency_name": "US Dollar",
    "rate_to_zmw": 28.00,
    "effective_from": "2025-11-01",
    "change_reason": "Monthly update based on BOZ rate"
  }'

# Response
{
  "success": true,
  "message": "FX rate updated successfully",
  "id": 2
}
```

Example 2: Creating Multi-Currency Requisition
```
  Item 1: 5 × $1,500 USD = $7,500 → K206,250 (rate: 27.50)
```

```bash
# Login as Procurement
curl -X POST http://localhost:3001/api/auth/login \
  -H "Content-Type: application/json" \
  -d '{"username":"james.phiri","password":"password123"}'

TOKEN="..."

# Create requisition
curl -X POST http://localhost:3001/api/requisitions \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "title": "IT Equipment",
    "description": "Laptops and accessories",
    "created_by": 3
  }'

# Response: { "id": 1, ... }

# Add item in USD
curl -X POST http://localhost:3001/api/requisitions/1/items \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "item_name": "Dell Latitude 7420",
    "quantity": 10,
    "unit_price": 1500,
    "currency": "USD",
    "specifications": "Core i7, 16GB RAM",
    "vendor_id": 1
  }'

# Response
{
  "success": true,
  "message": "Item added successfully",
  "item_id": 45,
  "amount_in_zmw": 420000,  # 10 × 1500 × 28.00
  "fx_rate_used": 28.00
}

# Add item in EUR
curl -X POST http://localhost:3001/api/requisitions/1/items \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "item_name": "HP LaserJet Pro",
    "quantity": 3,
    "unit_price": 800,
    "currency": "EUR",
    "vendor_id": 2
  }'

# Response
{
  "success": true,
  "item_id": 46,
  "amount_in_zmw": 71520,  # 3 × 800 × 29.80
  "fx_rate_used": 29.80
}

# Requisition total: K420,000 + K71,520 = K491,520 ZMW
```

Example 3: Budget Approval

```
# Login as Finance Manager
curl -X POST http://localhost:3001/api/auth/login \
  -H "Content-Type: application/json" \
  -d '{"username":"sarah.banda","password":"password123"}'

TOKEN="..."

# Check IT department budget
curl -H "Authorization: Bearer $TOKEN" \
  "http://localhost:3001/api/budgets/department/IT?fiscal_year=2025"

# Response
{
  "budget": {
    "id": 1,
    "department": "IT",
    "allocated_amount": 500000,
    "committed_amount": 0,
    "spent_amount": 0,
    "available_amount": 500000,
    "fiscal_year": "2025"
  },
  "expenses": []
}

# Approve budget for requisition
curl -X POST http://localhost:3001/api/requisitions/1/budget-check \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "approved": true,
    "comments": "Budget approved for Q4 IT equipment purchase"
  }'

# Response
{
  "success": true,
  "message": "Budget approved and funds committed",
  "committed": 491520,
  "remaining": 8480
}

# Check budget again
curl -H "Authorization: Bearer $TOKEN" \
  "http://localhost:3001/api/budgets/department/IT?fiscal_year=2025"

# Response shows updated budget
{
  "budget": {
    "allocated_amount": 500000,
    "committed_amount": 491520,  # ← increased
    "spent_amount": 0,
    "available_amount": 8480,    # ← decreased
    "utilization_percentage": 98.30  # ← critical!
  },
  "expenses": [
    {
      "requisition_id": 1,
      "amount": 491520,
      "expense_type": "committed",
      ...
    }
  ]
}
```

Example 4: Generating Reports

```
# Generate budget PDF report
curl -H "Authorization: Bearer $TOKEN" \
  "http://localhost:3001/api/reports/budgets/pdf?fiscal_year=2025" \
  --output budget_report_2025.pdf

# Generate requisitions Excel report (filtered)
curl -H "Authorization: Bearer $TOKEN" \
  "http://localhost:3001/api/reports/requisitions/excel?dateFrom=2025-10-01&dateTo=2025-10-31&status=hod_approved" \
  --output requisitions_october.xlsx

# Generate IT department spending report
curl -H "Authorization: Bearer $TOKEN" \
  "http://localhost:3001/api/reports/department/IT/pdf?fiscal_year=2025" \
  --output it_spending_2025.pdf

# Generate FX rates report
curl -H "Authorization: Bearer $TOKEN" \
  "http://localhost:3001/api/reports/fx-rates/excel" \
  --output fx_rates.xlsx
```

✏️ Testing Checklist

Basic Functionality

- [x] Login as Finance Manager, Procurement, MD, Admin
- [x] View active FX rates
- [x] Update USD exchange rate
- [x] Update EUR exchange rate
- [x] View FX rate history
- [x] View budget overview
- [x] View department budget details

Multi-Currency

- [x] Add item in USD
- [x] Add item in EUR
- [x] Add item in ZMW
- [x] Mixed-currency requisition (all 3 currencies)
- [x] Verify automatic ZMW conversion
- [x] Check FX rate tracking per item
- [x] Verify requisition total in ZMW

Budget Management

- [x] View all departments' budgets
- [x] Approve budget for requisition
- [x] Verify budget commitment
- [x] Check available amount calculation
- [x] Test budget rejection (insufficient funds)
- [x] View budget utilization percentages
- [x] Check budget expenses history

Reporting

- [x] Generate requisition summary PDF
- [x] Generate requisition summary Excel
- [x] Generate budget report PDF
- [x] Generate budget report Excel
- [x] Generate FX rates report Excel
- [x] Generate departmental spending PDF
- [x] Test report filtering (date, status, department)

Error Handling

- [x] Invalid currency code
- [x] Missing FX rate
- [x] Insufficient budget
- [x] Unauthorized access (wrong role)
- [x] Invalid date filters

🎓 What Each Role Can Do Now

Finance Manager (sarah.banda)

- ☑ Manage FX rates (view, create, update, deactivate)
- ☑ View FX rate history
- ☑ View all departmental budgets
- ☑ Approve/reject budget for requisitions
- ☑ Update budget allocations
- ☑ Generate all reports (PDF & Excel)
- ☑ View budget expenses and utilization
- ☑ Access FX rates report

MD (david.mulenga)

- ☑ View all departmental budgets
- ☑ View budget utilization
- ☑ Generate requisition reports (PDF & Excel)
- ☑ Generate budget reports (PDF & Excel)
- ☑ View departmental spending reports
- ☑ View FX rates (read-only)
- ☑ Monitor budget performance

Procurement (james.phiri)

- ☑ Select currency for each item (USD, EUR, ZMW)
- ☑ View current FX rates
- ☑ Update FX rates
- ☑ Create multi-currency requisitions
- ☑ System auto-converts to ZMW
- ☑ Track which FX rate was used per item

Admin

- ☑ Full access to all features
- ☑ Manage FX rates
- ☑ Manage budgets
- ☑ Generate all reports
- ☑ Override approvals if needed
- ☑ System administration

HOD (mary.mwanza)

- ☑ View own department spending report
- ☑ View own department budget (read-only)
- ☑ Approve requisitions (existing feature)

📊 Statistics

Code Added

- **New Files:** 4 (routes, utils, migrations, docs)
- **Modified Files:** 2 (server.js, package.json)
- **Lines of Code Added:** ~2,500+ lines
- **New API Endpoints:** 15
- **New Database Tables:** 3
- **Enhanced Tables:** 3
- **New Database Columns:** 10

Features Delivered

- ☑ Budget management with Finance Manager and MD access
- ☑ FX rate management (Finance Manager, Procurement, Admin)
- ☑ Multi-currency support (USD, EUR, ZMW)
- ☑ Automatic currency conversion to ZMW
- ☑ Budget commitment and tracking
- ☑ Budget utilization alerts
- ☑ PDF reports (3 types)
- ☑ Excel reports (3 types)
- ☑ Comprehensive audit trails
- ☑ Role-based access control
- ☑ Complete documentation

Documentation

- **Main Guide:** 40+ pages (BUDGET_FX_AND_REPORTING_GUIDE.md)
- **Implementation Summary:** This file
- **API Endpoints Documented:** 15
- **Usage Examples:** 10+
- **Test Scenarios:** 20+

🚀 Getting Started

Quick Start

1. **Install dependencies:**

```
cd backend
npm install
```

2. **Run migrations:**

```
node scripts/addCurrencyAndFXSupport.js
node scripts/addBudgetsTable.js
```

3. **Start server:**

```
npm start
```

4. **Test with Finance Manager:**

```
# Login
curl -X POST http://localhost:3001/api/auth/login \
  -H "Content-Type: application/json" \
  -d '{"username":"sarah.banda","password":"password123"}'

# View FX rates
curl -H "Authorization: Bearer {token}" \
  http://localhost:3001/api/fx-rates

# View budgets
curl -H "Authorization: Bearer {token}" \
  "http://localhost:3001/api/budgets/overview?fiscal_year=2025"
```

📑 Documentation Files

1. **BUDGET_FX_AND_REPORTING_GUIDE.md** - Complete guide

- Feature overview
  - API endpoints with examples
  - Usage scenarios
  - Troubleshooting
  - Best practices

2. **IMPLEMENTATION_SUMMARY_V3.0.md** - This file

  - High-level overview
  - What was implemented
  - Technical details
  - Quick start guide

3. **WORKFLOW_GUIDE.md** - Complete workflow (from v2.2)

  - Draft to PDF workflow
  - All approval stages

4. **API_TESTING.md** - API testing guide (from v2.2)

  - Testing instructions
  - cURL examples

⚡ Key Improvements Over Previous Versions

Version 3.0 vs 2.2

Feature	v2.2	v3.0
Budget Management	Basic budgets table	Full budget lifecycle management
Currency Support	ZMW only	USD, EUR, ZMW with FX rates
Finance Manager Role	Limited	Full budget and FX control
MD Oversight	None	Full visibility into budgets
Reporting	Single PDF	PDF & Excel, multiple report types
Budget Tracking	Manual	Automatic commitment/tracking
FX Rates	None	Full management with history
Audit Trail	Requisitions only	+ Budgets + FX rates

🎯 Success Metrics

Metric	Target	Status
Budget Management	☑	Complete
FX Rate Management	☑	Complete
Multi-Currency (USD, EUR, ZMW)	☑	Complete
Finance Manager Access	☑	Complete
MD Access	☑	Complete
Procurement Currency Selection	☑	Complete
PDF Reports	☑	Complete
Excel Reports	☑	Complete
Auto Currency Conversion	☑	Complete
Budget Approval Workflow	☑	Complete
Comprehensive Documentation	☑	Complete
Overall	**100%** ☑	**COMPLETE**

🗓 Version History

Version	Date	Key Features
3.0.0	**Oct 28, 2025**	**Budget Management, FX Rates, Multi-Currency, Advanced Reporting**
2.2.0	Oct 23, 2025	Complete workflow, PDF generation
2.1.0	Oct 23, 2025	Rate limiting, refresh tokens, logging
2.0.0	Oct 22, 2025	Security overhaul, authentication
1.0.0	Initial	Basic requisition system

💡 Future Enhancement Suggestions

Phase 4 (Optional)

1. **Email Notifications**

  - Budget utilization alerts
  - FX rate change notifications
  - Monthly budget reports

2. **Budget Forecasting**

  - Predict utilization based on trends
  - Recommend budget adjustments
  - Seasonal analysis

3. **Advanced FX Features**

- Auto-update rates from BOZ API
- Support for more currencies (GBP, CNY, etc.)
- Forward contracts and hedging

4. **Enhanced Reporting**

   - Interactive dashboards
   - Charts and visualizations
   - Comparative analysis (YoY, QoQ)
   - CSV export option

5. **Mobile App**

   - Budget monitoring on mobile
   - Push notifications
   - Quick approvals

🧎 Summary

You now have a **production-ready** Purchase Requisition System with:

☑ Complete Features

- Full requisition workflow (draft → approval → procurement → PDF)
- Budget management with real-time tracking
- Multi-currency support (USD, EUR, ZMW)
- FX rate management with audit trail
- Comprehensive PDF and Excel reporting
- Role-based access control
- Security features (JWT, bcrypt, rate limiting)
- Complete audit trails
- Extensive documentation

☑ User Roles Configured

- **Finance Manager** - Budget and FX rate control
- **MD** - Budget oversight and reporting
- **Procurement** - Multi-currency requisitions
- **Admin** - Full system access
- **HOD** - Approvals and department reports
- **Initiators** - Create requisitions

☑ Currencies Supported

- ZMW (Zambian Kwacha) - Base currency
- USD (US Dollar) - International purchases
- EUR (Euro) - European suppliers

☑ Reports Available

- Requisition Summary (PDF & Excel)
- Budget Reports (PDF & Excel)
- FX Rates Report (Excel)
- Departmental Spending (PDF)

System Status: ☑ **PRODUCTION READY**

Version: 3.0.0
Implementation Date: October 28, 2025
All features tested and working perfectly! 🎉

For detailed instructions, see **BUDGET_FX_AND_REPORTING_GUIDE.md**

📞 Questions?

- Review BUDGET_FX_AND_REPORTING_GUIDE.md for complete documentation
- Check API_TESTING.md for API testing examples
- Review WORKFLOW_GUIDE.md for workflow details
- Check SECURITY.md for security guidelines

Everything you requested has been implemented and documented! ✨