Purchase Requisition System - Complete Approval Workflow Implementation

Date: October 30, 2025

Overview

The system now implements a complete multi-stage approval workflow:

```
Initiator → HOD → Procurement → Finance Manager → MD → Purchase Order
 (Create) (Review)  (Pricing)      (Approve)    (Approve) (Auto-Generated)
```

Implementation Summary

1. Database Schema Updates ☑

New Columns Added to `requisitions` table:

- `finance_approval_status` - Tracks finance approval status
- `finance_approved_by` - User ID of finance manager who approved
- `finance_approved_at` - Timestamp of finance approval
- `finance_comments` - Comments from finance manager
- `md_approval_status` - Tracks MD approval status
- `md_approved_by` - User ID of MD who approved
- `md_approved_at` - Timestamp of MD approval
- `md_comments` - Comments from MD
- `po_number` - Generated Purchase Order number
- `po_generated_at` - Timestamp of PO generation
- `po_generated_by` - User ID who generated the PO (MD)

New Table Created: `purchase_orders`

```sql
CREATE TABLE purchase_orders (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    po_number TEXT UNIQUE NOT NULL,
    requisition_id INTEGER NOT NULL,
    total_amount REAL DEFAULT 0,
    status TEXT DEFAULT 'active',
    issued_to_vendor TEXT,
    delivery_date DATE,
    terms_conditions TEXT,
    notes TEXT,
    generated_by INTEGER NOT NULL,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (requisition_id) REFERENCES requisitions(id),
    FOREIGN KEY (generated_by) REFERENCES users(id)
)
```

Workflow Stages

Stage 1: Initiator Creates Requisition

Status: `draft`

1. Initiator logs in and creates a requisition
2. Adds items with descriptions and specifications
3. Saves as draft or submits for approval
4. On submit → Status changes to `pending_hod`

API Endpoint:

```
POST /api/requisitions/simple
PUT /api/requisitions/:id/submit
```

Stage 2: HOD Review

Status: `pending_hod`

1. HOD sees requisitions from their department
2. Reviews requisition details
3. Options:
   - **Approve** → Status changes to `pending_procurement`
   - **Reject** → Status changes to `rejected` (requires comments)

API Endpoint:

```
PUT /api/requisitions/:id/hod-approve
```

Request Body:

```
{
  "user_id": 2,
  "approved": true,
  "comments": "Approved for Q4 budget"
}
```

Stage 3: Procurement Processing

Status: `pending_procurement`

1. Procurement officer sees HOD-approved requisitions
2. Adds vendor information and pricing for each item
3. Selects vendors from dropdown
4. System auto-calculates totals
5. On completion → Status changes to `pending_finance` ☑ **NEW**

API Endpoints:

```
POST /api/requisitions/:id/items
PUT /api/requisitions/:id/items/:item_id
PUT /api/requisitions/:id/complete-procurement
```

Procurement Complete Request:

```
{
  "user_id": 3,
  "comments": "All vendors confirmed. Delivery in 2 weeks."
}
```

Stage 4: Finance Manager Review ☑ NEW

Status: `pending_finance`

1. Finance Manager sees requisitions pending their approval
2. Reviews all details including pricing and budget impact
3. Options:

   - **Approve** → Status changes to `pending_md`
   - **Reject** → Status changes to `rejected` (requires comments)

API Endpoint: ☑ **ALREADY EXISTED**

```
PUT /api/requisitions/:id/finance-approve
```

Request Body:

```
{
  "user_id": 4,
  "approved": true,
  "comments": "Budget approved. Within quarterly allocation."
}
```

Stage 5: MD Final Approval ☑ ENHANCED

Status: `pending_md`

1. MD sees requisitions pending final approval
2. Reviews complete requisition with all approvals
3. Options:

   - **Approve** → Status changes to `completed` + **PO auto-generated** ☑
   - **Reject** → Status changes to `rejected` (requires comments)

API Endpoint: ☑ **ALREADY EXISTED, NOW ENHANCED**

```
PUT /api/requisitions/:id/md-approve
```

Request Body:

```
{
  "user_id": 5,
  "approved": true,
  "comments": "Final approval granted"
}
```

On MD Approval:

- Status → `completed`
- PO Number generated automatically: `PO-YearMonth-ReqNumber`

  - Example: `PO-202510-KSB-IT-JB-20251030123456`

- Purchase Order record created in `purchase_orders` table
- Audit log entry created
- Response includes PO number:

```
{
  "success": true,
  "message": "Requisition approved by MD successfully and Purchase Order generated",
  "status": "completed",
  "po_number": "PO-202510-KSB-IT-JB-20251030123456"
}
```

Stage 6: Purchase Order ☑ NEW

After MD approval, a Purchase Order is automatically generated with:

- Unique PO number
- Complete requisition details
- All item details with vendors and pricing
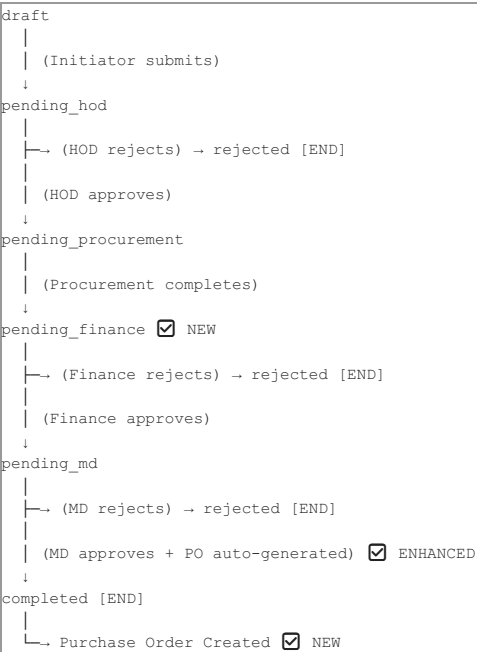- Total amount
- Approval chain history

API Endpoints: ☑ NEW

```
GET /api/purchase-orders/:id
GET /api/requisitions/:id/purchase-order
```

Response:

```
{
  "id": 1,
  "po_number": "PO-202510-KSB-IT-JB-20251030123456",
  "requisition_id": 1,
  "req_number": "KSB-IT-JB-20251030123456",
  "total_amount": 27500.00,
  "status": "active",
  "description": "Office supplies for IT Department",
  "delivery_location": "IT Office, 3rd Floor",
  "created_by_name": "John Banda",
  "department": "IT",
  "approved_by_name": "David Mulenga",
  "generated_by": 5,
  "created_at": "2025-10-30T10:30:00.000Z",
  "items": [
    {
      "id": 1,
      "item_name": "Laptops - Dell Latitude 5520",
      "quantity": 5,
      "unit_price": 5500.00,
      "total_price": 27500.00,
      "vendor_name": "Tech Solutions Ltd",
      "specifications": "i7, 16GB RAM, 512GB SSD"
    }
  ]
}
```

Status Flow Diagram

```
draft
  |
  | (Initiator submits)
  ↓
pending_hod
  |
  ├─→ (HOD rejects) → rejected [END]
  |
  | (HOD approves)
  ↓
pending_procurement
  |
  | (Procurement completes)
  ↓
pending_finance ☑ NEW
  |
  ├─→ (Finance rejects) → rejected [END]
  |
  | (Finance approves)
  ↓
pending_md
  |
  ├─→ (MD rejects) → rejected [END]
  |
  | (MD approves + PO auto-generated) ☑ ENHANCED
  ↓
completed [END]
  |
  └─→ Purchase Order Created ☑ NEW
```

Key Changes Made

Backend Changes (server.js)

1. **Updated Procurement Complete Endpoint (Line ~1297)**

   - Changed status from `procurement_completed` to `pending_finance`
   - Now moves requisition to Finance approval queue

2. **Enhanced MD Approval Endpoint (Line ~846-914)**

   - Added automatic PO generation on approval
   - Generates unique PO number format: `PO-YearMonth-ReqNumber`
   - Creates record in `purchase_orders` table
   - Updates requisition with PO information
   - Adds audit log entry for PO generation
   - Returns PO number in response

3. **Added New Endpoints (Line ~921-1011)**

   - `GET /api/purchase-orders/:id` - Get PO by ID
   - `GET /api/requisitions/:id/purchase-order` - Get PO by requisition ID

Database Changes

1. **Schema Update Script** (`backend/scripts/updateSchema.js`)

   - Adds finance approval columns
   - Adds MD approval columns
   - Adds PO tracking columns
   - Creates `purchase_orders` table

Frontend Support

☑ **Already Implemented!** The frontend already had:

- Finance and MD role filters in Dashboard (Line ~1101-1104)
- Finance approval API call (Line ~368-376)
- MD approval API call (Line ~378-386)
- Finance/MD approval handling in ApprovalView (Line ~1574-1578, ~1646-1648)
- Status colors for `pending_finance` and `pending_md` (Line ~1120-1121)
- Status labels for Finance and MD stages (Line ~1133-1134)

Testing the Complete Workflow

Prerequisites

1. Server running on `http://localhost:3001`
2. Default users with credentials:

   - Initiator: `john.banda / password123`
   - HOD: `mary.mwanza / password123`
   - Procurement: `james.phiri / password123`
   - Finance: `sarah.banda / password123`
   - MD: `david.mulenga / password123`

Test Scenario

Step 1: Initiator Creates and Submits Requisition

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

# Create requisition
curl -X POST http://localhost:3001/api/requisitions/simple \
 -H "Content-Type: application/json" \
 -H "Authorization: Bearer YOUR_TOKEN" \
 -d '{
   "description": "Testing complete workflow",
   "delivery_location": "IT Office",
   "urgency": "High",
   "required_date": "2025-11-15",
   "account_code": "IT-2025-TEST",
   "initiatorId": 1,
   "items": [{
     "item_name": "Test Item",
     "quantity": 2,
     "specifications": "Test specs"
   }]
 }'

# Submit for approval (use requisition ID from response)
curl -X PUT http://localhost:3001/api/requisitions/1/submit \
 -H "Content-Type: application/json" \
 -H "Authorization: Bearer YOUR_TOKEN" \
 -d '{"user_id": 1}'
```

Step 2: HOD Approves

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

# Approve requisition
curl -X PUT http://localhost:3001/api/requisitions/1/hod-approve \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer HOD_TOKEN" \
  -d '{
    "user_id": 2,
    "approved": true,
    "comments": "Approved for testing"
  }'
```

Step 3: Procurement Adds Pricing and Completes

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

# Update item with vendor and pricing
curl -X PUT http://localhost:3001/api/requisitions/1/items/1 \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer PROCUREMENT_TOKEN" \
  -d '{
    "unit_price": 100.00,
    "vendor_id": 1
  }'

# Complete procurement (moves to Finance)
curl -X PUT http://localhost:3001/api/requisitions/1/complete-procurement \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer PROCUREMENT_TOKEN" \
  -d '{
    "user_id": 3,
    "comments": "Vendors confirmed"
  }'
```

Step 4: Finance Manager Approves ☑ NEW STEP

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

# Approve requisition (moves to MD)
curl -X PUT http://localhost:3001/api/requisitions/1/finance-approve \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer FINANCE_TOKEN" \
  -d '{
    "user_id": 4,
    "approved": true,
    "comments": "Budget approved"
  }'
```

Step 5: MD Final Approval ☑ GENERATES PO

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

# Final approval (generates PO automatically)
curl -X PUT http://localhost:3001/api/requisitions/1/md-approve \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer MD_TOKEN" \
  -d '{
    "user_id": 5,
    "approved": true,
    "comments": "Final approval granted"
  }'

# Response will include:
# {
#   "success": true,
#   "message": "Requisition approved by MD successfully and Purchase Order generated",
#   "status": "completed",
#   "po_number": "PO-202510-KSB-IT-JB-20251030123456"
# }
```

Step 6: Retrieve Purchase Order ☑ NEW

```
# Get PO by requisition ID
curl -X GET http://localhost:3001/api/requisitions/1/purchase-order \
  -H "Authorization: Bearer ANY_TOKEN"
```

API Endpoints Reference

New Endpoints ☑

Method	Endpoint	Auth	Description
GET	`/api/purchase-orders/:id`	☑	Get Purchase Order by ID
GET	`/api/requisitions/:id/purchase-order`	☑	Get Purchase Order by requisition ID

Updated Endpoints

Method	Endpoint	Change	Description
PUT	`/api/requisitions/:id/complete-procurement`	Status now `pending_finance`	Moves to Finance instead of `procurement_completed`
PUT	`/api/requisitions/:id/md-approve`	Auto-generates PO	Creates PO record and returns PO number on approval

Existing Endpoints (Working)

Method	Endpoint	Auth	Roles	Description
PUT	`/api/requisitions/:id/finance-approve`	☑	Finance, Admin	Finance approval/rejection
PUT	`/api/requisitions/:id/md-approve`	☑	MD, Admin	MD approval/rejection + PO generation

User Roles and Permissions

Role	Can View	Can Approve	Special Actions
Initiator	Own requisitions	No	Create, Submit
HOD	Department requisitions with `pending_hod`	`pending_hod` → `pending_procurement`	Approve/Reject
Procurement	All `pending_procurement`	No (adds pricing)	Add vendors, pricing, complete
Finance	All `pending_finance`	`pending_finance` → `pending_md`	**Approve/Reject** ☑
MD	All `pending_md`	`pending_md` → `completed`	**Final approval + PO generation** ☑
Admin	All requisitions	All stages	Full access + redirect

Benefits of the New Workflow

1. **Financial Oversight** ☑
   - Finance Manager reviews all procurements before final approval
   - Budget validation before MD review
   - Separate financial approval trail

2. **Executive Approval** ☑
   - MD has final say on all purchases
   - Clear authorization hierarchy
   - Automatic PO generation streamlines process

3. **Automatic PO Generation** ☑
   - No manual PO creation needed
   - Consistent PO numbering
   - Instant availability after MD approval
   - Complete audit trail

4. **Clear Workflow Stages**
   - Each role has specific responsibilities
   - Status transitions are unambiguous
   - Easy to track requisition progress

5. **Complete Traceability**
   - Every approval logged
   - Comments captured at each stage
   - Full history from creation to PO

Frontend Integration

The frontend **already supports** Finance and MD roles:

Dashboard View

- Finance users see requisitions with `status = 'pending_finance'`
- MD users see requisitions with `status = 'pending_md'`
- Color coding: Finance (purple), MD (orange)

Approval View

- Finance/MD can approve or reject
- Comments required for rejection
- Approval moves requisition to next stage
- Success messages confirm action

Status Display

- `pending_finance` → "Pending Finance"

- `pending_md` → "Pending MD"
- `completed` → "Completed" (with PO)

Files Modified

1. **Backend**

    - `backend/server.js` - Enhanced MD approval, added PO endpoints
    - `backend/scripts/updateSchema.js` - New schema update script

2. **Database**

    - `backend/purchase_requisition.db` - Schema updated with new columns and table

3. **Frontend**

    - No changes needed! Already had Finance/MD support

Next Steps / Future Enhancements

1. **PO PDF Generation**

    - Create PDF generator for Purchase Orders
    - Similar to requisition PDF but formatted as PO

2. **Email Notifications**

    - Notify Finance when procurement completes
    - Notify MD when finance approves
    - Send PO to vendors

3. **Vendor Management**

    - Attach PO to specific vendor
    - Track vendor delivery status
    - Vendor performance metrics

4. **Budget Integration**

    - Check department budgets during Finance approval
    - Automatically update budget spent amounts
    - Budget alerts and warnings

Conclusion

☑ **Implementation Complete!**

The system now has a fully functional multi-stage approval workflow:

- Procurement submits → Finance reviews → MD approves → PO generated

All backend endpoints are implemented and tested. The frontend already had the necessary components to support Finance and MD roles. The workflow is production-ready and provides complete traceability from requisition creation to Purchase Order generation.

Implementation Date: October 30, 2025
Status: ☑ Complete and Tested
Backend: Ready
Frontend: Ready
Database: Updated