Purchase Orders Implementation - Complete

Date: October 30, 2025

Overview

Implemented a comprehensive Purchase Order system with role-based access control and PDF generation capabilities. POs are automatically created when MD approves requisitions and are accessible through a dedicated "Purchase Orders" section in the sidebar.

Features Implemented

1. Automatic PO Generation ☑

- **Trigger:** When MD approves a requisition
- **PO Number Format:** `PO-YearMonth-ReqNumber`
  - Example: `PO-202510-KSB-IT-JB-20251030123456`
- **Storage:** Saved in `purchase_orders` table
- **Audit Trail:** Logged in `audit_log` table

2. Role-Based Access Control ☑

Role	Access Level	Can See
Procurement	All POs	All Purchase Orders (for download/distribution)
Finance	All POs	All Purchase Orders (financial oversight)
MD	All POs	All Purchase Orders (executive view)
HOD	Filtered POs	Only POs from requisitions they approved
Initiator	Own POs	Only POs from their own requisitions
Admin	All POs	All Purchase Orders (full access)

3. PDF Generation ☑

- **Professional Format:** Company header, vendor details, items table
- **Content:**
  - PO Number and Date
  - Company Information (From)
  - Vendor Information (To)
  - Requisition Details
  - Items with quantities, unit prices, totals
  - Grand Total
  - Complete Approval Chain (HOD → Procurement → Finance → MD)
  - Generation timestamp
- **Security:** Role-based access validation before PDF generation
- **Filename:** `PO_{PO_NUMBER}.pdf`

4. Purchase Orders View ☑

- **Location:** Sidebar → "📄 Purchase Orders" menu item
- **Display:**
  - Table with PO Number, Requisition, Description, Department, Amount, Date
  - Download PDF button for each PO
  - Responsive design
  - Empty state with helpful messages
- **Functionality:**
  - One-click PDF download
  - Automatic filename generation
  - Error handling with user-friendly messages

Backend Implementation

API Endpoints

1. Get All Purchase Orders

```
GET /api/purchase-orders
Authorization: Bearer <token>
```

Role-Based Filtering:

- **Initiators:** Returns only POs from their requisitions (`created_by = user.id`)
- **HODs:** Returns only POs they approved (`hod_approved_by = user.id`)
- **Procurement/Finance/MD/Admin:** Returns all POs

Response:

```
[
  {
    "id": 1,
    "po_number": "PO-202510-KSB-IT-JB-20251030123456",
    "requisition_id": 5,
    "req_number": "KSB-IT-JB-20251030123456",
    "description": "Office supplies for IT Department",
    "department": "IT",
    "total_amount": 27500.00,
    "created_by": 1,
    "created_by_name": "John Banda",
    "approved_by_name": "David Mulenga",
    "created_at": "2025-10-30T10:30:00.000Z"
  }
]
```

2. Download PO as PDF

```
GET /api/purchase-orders/:id/pdf
Authorization: Bearer <token>
```

Access Control:

- Validates user has permission to view this PO
- Returns 403 if access denied

Response:

- `Content-Type: application/pdf`
- `Content-Disposition: attachment; filename="PO_{PO_NUMBER}.pdf"`
- Binary PDF stream

File Location: `backend/server.js` (Lines 1072-1321)

Frontend Implementation

Components

1. Purchase Orders View

Location: `frontend/app.js` (Lines 3044-3145)

Features:

- Loads POs on component mount
- Displays table with sortable columns
- Download button for each PO
- Loading state
- Empty state with role-specific messages

2. Sidebar Menu Item

Location: `frontend/app.js` (Line 982)

Configuration:

```
{
  id: 'purchase-orders',
  label: 'Purchase Orders',
  icon: '📄',
  show: ['initiator', 'hod', 'procurement', 'finance', 'md', 'admin'].includes(user.role)
}
```

API Integration

Location: `frontend/app.js` (Lines 206-217)

```
// Get all POs (filtered by backend based on role)
getPurchaseOrders: async () => {
  const res = await fetchWithAuth(`${API_URL}/purchase-orders`);
  if (!res.ok) throw new Error('Failed to fetch purchase orders');
  return res.json();
},

// Download PO PDF
downloadPOPDF: async (poId) => {
  const res = await fetchWithAuth(`${API_URL}/purchase-orders/${poId}/pdf`);
  if (!res.ok) throw new Error('Failed to download PO PDF');
  return res.blob();
}
```

PDF Format Details

Document Structure

```
┌─────────────────────────────────────────────────────────┐
│                   PURCHASE ORDER                        │
│               PO Number: PO-202510-...                  │
│               Date: October 30, 2025                    │
├─────────────────────────────────────────────────────────┤
│  From:                          To:                     │
│  Kabwe Sugar Brokerage Ltd      Vendor Name            │
│  P.O. Box 123, Kabwe            vendor@email.com        │
│  Zambia                         +260 123 456            │
├─────────────────────────────────────────────────────────┤
│  Requisition Details                                    │
│  Requisition #: KSB-IT-JB-...                          │
│  Description: Office supplies...                        │
│  Department: IT                                         │
│  Delivery Location: IT Office, 3rd Floor               │
│  Required Date: November 15, 2025                      │
│  Urgency: High                                         │
├─────────────────────────────────────────────────────────┤
│  Items                                                  │
│  Item                Qty  Unit Price      Total        │
│  ──────────────────────────────────────────────────────│
│  Laptops - Dell       5    ZMW 5,500.00   ZMW 27,500  │
│  Specs: i7, 16GB RAM, 512GB SSD                       │
│                                                         │
│  ──────────────────────────────────────────────────────│
│                Grand Total:              ZMW 27,500.00 │
├─────────────────────────────────────────────────────────┤
│  Approval Chain                                         │
│  Requested by: John Banda                              │
│  HOD Approved by: Mary Mwanza                          │
│  Procurement by: James Phiri                           │
│  Finance Approved by: Sarah Banda                      │
│  MD Approved by: David Mulenga                         │
├─────────────────────────────────────────────────────────┤
│  This is a computer-generated document.                │
│  Generated on: October 30, 2025 10:30 AM              │
└─────────────────────────────────────────────────────────┘
```

PDF Styling

- **Font:** Helvetica (standard PDF font)
- **Page Size:** Letter (8.5" x 11")
- **Margins:** 50 points all sides
- **Colors:** Black text, gray for secondary info
- **Layout:** Professional business document format

Database Schema

purchase_orders Table

```
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)
)
```

Complete Workflow

Requisition to PO Flow

```
1. Initiator: Create requisition
   ↓
2. HOD: Approve requisition
   ↓
3. Procurement: Add vendors & pricing
   ↓
4. Finance: Approve budget
   ↓
5. MD: Final approval
   ↓
6. SYSTEM: Auto-generate PO ☑ NEW
   ↓
7. Users: View & Download PO ☑ NEW
```

PO Number Generation

Format: `PO-{YearMonth}-{ReqNumber}`

Example:

- Requisition: `KSB-IT-JB-20251030123456`
- PO Number: `PO-202510-KSB-IT-JB-20251030123456`

Logic:

```
const now = new Date();
const yearMonth = `${now.getFullYear()}${String(now.getMonth() + 1).padStart(2, '0')}`;
const poNumber = `PO-${yearMonth}-${req.req_number}`;
```

User Experience

Procurement Role

Access: All Purchase Orders

Use Case:

1. Navigate to "Purchase Orders" in sidebar
2. See complete list of all POs
3. Download any PO as PDF
4. Distribute POs to vendors

Why: Procurement needs access to all POs to manage vendor relationships and order fulfillment.

Finance Role

Access: All Purchase Orders

Use Case:

1. Navigate to "Purchase Orders" in sidebar
2. See all POs for financial oversight
3. Download POs for records
4. Verify payment amounts

Why: Finance needs visibility into all POs for budget tracking and payment processing.

MD Role

Access: All Purchase Orders

Use Case:

1. Navigate to "Purchase Orders" in sidebar
2. Review all POs issued
3. Download for executive records
4. Track organizational spending

Why: MD approved these requisitions and needs visibility into the resulting POs.

HOD Role

Access: POs They Approved

Use Case:

1. Navigate to "Purchase Orders" in sidebar
2. See POs from requisitions they approved
3. Download for department records
4. Track department spending

Why: HODs need to see outcomes of requisitions they approved from their department.

Initiator Role

Access: Own POs Only

Use Case:

1. Navigate to "Purchase Orders" in sidebar

2. See POs from their requisitions
3. Download for reference
4. Confirm order details

Why: Initiators need to see if their requisitions resulted in POs and track order status.

Testing Instructions

Test 1: Complete Flow End-to-End

Step 1: Create and Approve Requisition

```
1. Login as initiator (john.banda)
2. Create requisition
3. Submit for approval
4. Logout

5. Login as HOD (mary.mwanza)
6. Approve requisition
7. Logout

8. Login as Procurement (james.phiri)
9. Add vendor and pricing
10. Submit to Finance
11. Logout

12. Login as Finance (sarah.banda)
13. Approve requisition
14. Logout

15. Login as MD (david.mulenga)
16. Approve requisition
17. Verify "PO generated" message appears
```

Step 2: View PO (Procurement)

```
1. Still logged in as MD (or login as procurement)
2. Click "📄 Purchase Orders" in sidebar
3. Should see newly created PO in table
4. PO Number should be in format: PO-202510-...
5. Click "⬇ Download PDF"
6. PDF should download with correct filename
7. Open PDF and verify all details are correct
```

Step 3: Test Role-Based Access

Initiator Access:

```
1. Logout and login as john.banda (initiator)
2. Click "Purchase Orders" in sidebar
3. Should see ONLY POs from their own requisitions
4. Download PDF - should work
```

HOD Access:

```
1. Logout and login as mary.mwanza (HOD)
2. Click "Purchase Orders"
3. Should see ONLY POs they approved as HOD
4. Download PDF - should work
```

Finance Access:

```
1. Logout and login as sarah.banda (finance)
2. Click "Purchase Orders"
3. Should see ALL POs
4. Download any PDF - should work
```

Security Features

1. Authentication Required

- All endpoints require valid JWT token
- Unauthorized users redirected to login

2. Role-Based Filtering

- Backend filters POs based on user role
- Frontend cannot bypass these restrictions

3. PDF Access Control

- Validates user permission before generating PDF
- Returns 403 Forbidden if access denied
- Checks:
    - Admin: Always allowed
    - MD/Finance/Procurement: Always allowed

- Initiator: Only if they created the requisition
    - HOD: Only if they approved the requisition

4. Audit Logging

- Every PDF download is logged in `audit_log`
- Tracks who downloaded which PO and when

Error Handling

Frontend

- Loading states while fetching data
- Error messages for failed API calls
- Graceful fallbacks for empty states
- User-friendly error alerts

Backend

- Try-catch blocks around all operations
- Detailed error logging
- Proper HTTP status codes
- Descriptive error messages

Files Modified

Backend

1. **server.js** (Lines 1072-1321)
    - Added `GET /api/purchase-orders` endpoint
    - Added `GET /api/purchase-orders/:id/pdf` endpoint
    - Role-based filtering logic
    - PDF generation with PDFKit

Frontend

1. **app.js** (Multiple locations)
    - Added PO API methods (Lines 206-217)
    - Added PurchaseOrders component (Lines 3044-3145)
    - Added sidebar menu item (Line 982)
    - Added view routing (Line 896)

Dependencies

Backend

- **pdfkit:** For PDF generation (already installed)
- **better-sqlite3:** For database queries (already installed)

Frontend

- No new dependencies required
- Uses existing React and fetch API

Known Limitations

1. **Single Vendor Per PO:**
    - Currently one PO per requisition
    - All items go to same vendor (first item's vendor)
    - Future: Split POs by vendor if needed
2. **PDF Customization:**
    - Company details hardcoded
    - Logo not included
    - Future: Add company logo and customizable header
3. **PO Status:**
    - All POs created with status 'active'
    - No workflow for PO modifications or cancellations
    - Future: Add PO lifecycle management

Future Enhancements

1. **PO Status Management:**
    - Mark POs as: Pending, Sent to Vendor, Acknowledged, Fulfilled, Cancelled
    - Track PO lifecycle

2. **Vendor Acknowledgment:**
   - Email PO to vendor automatically
   - Vendor portal to acknowledge receipt
   - Track vendor response
3. **Delivery Tracking:**
   - Mark items as received
   - Partial deliveries
   - Match deliveries to POs
4. **PO Amendments:**
   - Allow modifications to POs
   - Version tracking
   - Approval required for changes
5. **Multiple Vendors:**
   - Split requisitions into multiple POs
   - One PO per vendor
   - Cross-reference requisition
6. **Email Integration:**
   - Auto-email PO to procurement on generation
   - Email PO to vendor
   - CC relevant stakeholders
7. **Company Branding:**
   - Upload company logo
   - Customizable company details
   - Template selection
8. **PO Templates:**
   - Different templates for different types
   - International vs Domestic
   - Services vs Goods

Troubleshooting

Issue: "No Purchase Orders Found"

Possible Causes:

1. No requisitions have been approved by MD yet
2. User doesn't have access to existing POs (role-based)

Solution:

1. Complete a requisition through full workflow to MD approval
2. Check that requisition status is 'completed'
3. Verify PO was created in database:

```
SELECT * FROM purchase_orders;
```

Issue: PDF Download Fails

Possible Causes:

1. PDF generation error on backend
2. Missing vendor information
3. Network error

Solution:

1. Check backend logs for errors
2. Verify requisition has complete data
3. Check browser console for errors
4. Try different PO

Issue: Access Denied (403)

Possible Causes:

1. User doesn't have permission for that PO
2. Role not recognized

Solution:

1. Verify user role is correct
2. Check PO ownership/approval chain
3. Confirm backend filtering logic

Summary

☑ **Complete PO System Implemented!**

Features:

- Automatic PO generation on MD approval
- Role-based access control
- Professional PDF generation
- Dedicated Purchase Orders view
- Download functionality
- Complete audit trail

Access:

- Procurement: All POs
- Finance: All POs
- MD: All POs
- HOD: POs they approved
- Initiator: Own POs only
- Admin: All POs

Location: Sidebar → "📄 Purchase Orders"

PDF Format: Professional business document with full details and approval chain

---

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