

Bug Fixes Summary - October 30, 2025

Issues Fixed

1. Procurement to Finance Flow Not Working

Problem:

- Procurement officers were unable to submit requisitions to Finance Manager
- The `/procurement-update` endpoint was trying to update columns that didn't exist in the database

Root Cause:

The requisitions table was missing procurement-specific columns:

- `selected_vendor`
- `vendor_currency`
- `unit_price`
- `total_cost`
- `justification`
- `quantity`
- `procurement_status`
- `procurement_assigned_to`
- `procurement_completed_at`
- `procurement_comments`

Solution:

1. Created script `backend/scripts/addProcurementColumns.js` to add missing columns
2. Added rejection tracking columns:

- `rejected_by`
- `rejected_at`
- `rejection_reason`

Files Modified:

- `backend/scripts/addProcurementColumns.js` (NEW)
- Database schema updated

Testing:

```
# Run the schema update
cd backend && node scripts/addProcurementColumns.js
```

Result:

- Procurement officers can now successfully add vendor information and pricing
- Status correctly transitions from `pending_procurement` to `pending_finance`
- All procurement data is properly stored in the database

2. Draft Requisitions Not Editable

Problem:

- Initiators couldn't edit draft requisitions after creation
- Only option was to view draft but not modify it
- No "Update Draft" or "Submit" buttons for drafts

Solution:

Backend Changes:

Added new endpoint: `PUT /api/requisitions/:id/update-draft`

Features:

- Validates that requisition is in `draft` status before allowing edits
- Updates requisition fields: `description`, `quantity`, `required_date`, `urgency`
- Updates the associated item in `requisition_items` table
- Logs the update action in audit log
- Returns error if requisition is not a draft

File: `backend/server.js` (Lines 616-673)

Frontend Changes:

Enhanced ApproveRequisition component to support draft editing

New Features:

1. **Draft Detection:**

```
const isDraftEditable = user.role === 'initiator' &&
    req.status === 'draft' &&
    req.created_by === user.id;
```

2. **Editable Fields for Drafts:**

- Item Description (text input)
- Quantity (number input)
- Urgency (dropdown: Standard/Urgent/Critical)
- Justification (textarea)

3. New Buttons for Drafts:

- Submit for Approval (green button) - Submits draft to HOD
- Update Draft (blue button) - Saves changes without submitting

4. Read-Only View for Non-Drafts:

- Shows fields as text when requisition is not editable
- Maintains existing approval flow for other statuses

File: frontend/app.js (Lines 1678-1822, 1975-1987)

User Experience:

1. Initiator creates a draft requisition
2. Can view the draft and see editable fields highlighted in blue box
3. Can modify description, quantity, urgency, and justification
4. Can either:
 - Click "Update Draft" to save changes and keep as draft
 - Click "Submit for Approval" to send to HOD
5. Once submitted, requisition is no longer editable

Complete Workflow Now Working

Full Approval Chain:

1. Initiator: Create Draft → **Edit** Draft → Submit
2. HOD: Review → Approve/Reject
3. Procurement: Add Vendors & Pricing → Submit FIXED
4. Finance Manager: Review → Approve/Reject
5. MD: Final Approval → **Auto**-generate PO

Status Transitions:

```

draft (editable) 
  ↓ (initiator submits)
pending_hod
  ↓ (HOD approves)
pending_procurement
  ↓ (procurement completes)  FIXED
pending_finance
  ↓ (finance approves)
pending_md
  ↓ (MD approves + PO generated)
completed

```

API Endpoints Summary

New Endpoints Added:

Method	Endpoint	Purpose	Status
PUT	/api/requisitions/:id/update-draft	Update draft requisition	<input checked="" type="checkbox"/> NEW
GET	/api/purchase-orders/:id	Get PO by ID	<input checked="" type="checkbox"/> (from previous update)
GET	/api/requisitions/:id/purchase-order	Get PO by req ID	<input checked="" type="checkbox"/> (from previous update)

Fixed Endpoints:

Method	Endpoint	Issue	Fix
PUT	/api/requisitions/:id/procurement-update	Missing DB columns	Added procurement columns to schema

Database Schema Changes

New Columns Added to `requisitions` table:

Procurement Columns:

```

selected_vendor INTEGER
vendor_currency TEXT DEFAULT 'ZMW'
unit_price REAL DEFAULT 0
total_cost REAL DEFAULT 0
justification TEXT
quantity INTEGER
procurement_status TEXT DEFAULT 'pending'
procurement_assigned_to INTEGER
procurement_completed_at DATETIME
procurement_comments TEXT

```

Rejection Tracking:

```
rejected_by INTEGER  
rejected_at DATETIME  
rejection_reason TEXT
```

Approval Tracking (from previous update):

```
finance_approval_status TEXT DEFAULT 'pending'  
finance_approved_by INTEGER  
finance_approved_at DATETIME  
finance_comments TEXT  
md_approval_status TEXT DEFAULT 'pending'  
md_approved_by INTEGER  
md_approved_at DATETIME  
md_comments TEXT
```

Purchase Order Tracking (from previous update):

```
po_number TEXT  
po_generated_at DATETIME  
po_generated_by INTEGER
```

Testing Instructions

Test 1: Draft Editing

1. Login as Initiator:

```
Username: john.banda  
Password: password123
```

2. Create Draft:

- Click "Create Requisition"
- Fill in details
- Click "Save as Draft"

3. Edit Draft:

- Go to Dashboard
- Click "View" on the draft requisition
- You should see editable fields with blue background
- Modify any field
- Click "Update Draft"
- Verify changes are saved

4. Submit Draft:

- View the draft again
- Click "Submit for Approval"
- Verify status changes to pending_hod
- Verify requisition is no longer editable

Test 2: Complete Workflow

1. Initiator Creates and Submits:

```
User: john.banda / password123  
Actions: Create → Save → View → Submit  
Expected: Status = pending_hod
```

2. HOD Approves:

```
User: mary.mwanza / password123  
Actions: Review → Approve with comment  
Expected: Status = pending_procurement
```

3. Procurement Processes:

```
User: james.phiri / password123  
Actions: Add vendor, pricing → Submit  
Expected: Status = pending_finance 
```

4. Finance Approves:

```
User: sarah.banda / password123  
Actions: Review → Approve with comment  
Expected: Status = pending_md
```

5. MD Final Approval:

```
User: david.mulenga / password123  
Actions: Review → Approve  
Expected: Status = completed, PO generated
```

Files Modified

Backend:

1. backend/server.js
 - Added /api/requisitions/:id/update-draft endpoint (Lines 616-673)
 - Modified procurement complete logic (already done previously)
2. backend/scripts/addProcurementColumns.js (NEW)
 - Script to add missing database columns
 - Run once to update schema
3. backend/scripts/checkSchema.js (NEW)
 - Utility to verify database schema
 - Helpful for debugging

Frontend:

1. frontend/app.js
 - Enhanced ApproveRequisition component:
 - Added isDraftEditable flag (Line 1679)
 - Added handleUpdateDraft function (Lines 1681-1716)
 - Added handleSubmitDraft function (Lines 1718-1746)
 - Added conditional editable fields (Lines 1777-1831)
 - Updated button rendering (Lines 1975-1987)

Database:

1. backend/purchase_requisition.db
 - Schema updated with new columns
 - All existing data preserved

Known Limitations

1. **Single Item Support:**
 - The simplified requisition workflow currently supports one item per requisition
 - Multiple items are stored in `requisition_items` but the UI edits only the first one
2. **Required Date:**
 - Currently not editable in draft edit view (uses original value)
 - Can be enhanced in future
3. **Account Code:**
 - Set to null in simple requisition flow
 - Can be added if needed

Future Enhancements

1. **Multiple Items Editing:**
 - Allow adding/removing items in draft edit view
 - Show list of items with individual edit buttons
2. **Attachments:**
 - Allow uploading documents to drafts
 - Support for quotes, specifications, etc.
3. **Auto-Save:**
 - Automatically save draft changes every few seconds
 - Prevent data loss
4. **Draft Templates:**
 - Save frequently used requisition patterns
 - Quick-start new requisitions
5. **Collaborative Editing:**
 - Multiple users can comment on drafts
 - Request clarifications before submission

Deployment Checklist

- [x] Run schema update script on database
- [x] Restart backend server
- [x] Clear browser cache
- [x] Test with each user role
- [x] Verify complete workflow end-to-end
- [x] Check audit logs are being created
- [x] Verify PO generation still works

Rollback Plan

If issues occur:

1. Database Rollback:

```
-- Revert added columns (if needed)
ALTER TABLE requisitions DROP COLUMN selected_vendor;
ALTER TABLE requisitions DROP COLUMN vendor_currency;
-- ... (drop other new columns)
```

2. Code Rollback:

- Restore previous version of server.js
- Restore previous version of app.js

3. Quick Fix:

- Most changes are additive
- Old functionality still works
- Can disable new endpoints if needed

Support Information

Issue Tracking:

- Log issues in project documentation
- Note any database errors in backend/logs/

Common Issues:

1. "Only draft requisitions can be edited" error:

- Requisition status is not 'draft'
- User is not the creator
- Check requisition status in database

2. Missing vendor dropdown:

- Procurement columns not added to database
- Run addProcurementColumns.js script

3. Status not changing:

- Check audit log for errors
- Verify endpoint is being called
- Check browser console for errors

Summary

Both Issues Resolved!

1. **Procurement → Finance flow:** Fixed by adding missing database columns
2. **Draft editing:** Implemented with new endpoint and enhanced UI

The system now provides a complete, production-ready workflow from requisition creation through PO generation, with full support for draft editing and the complete approval chain.

Date: October 30, 2025

Status: Complete and Tested

Version: 3.1