HOD Review Features - Complete Guide

Date: October 23, 2025
Status: ☑ **IMPLEMENTED**

🎯 What Was Implemented

The HOD (Head of Department) review interface has been enhanced with the following features:

☑ 1. Approve and Decline Buttons

- **Two clear action buttons:**
  - 🟢 **"Approve"** - Approves the requisition
  - 🔴 **"Decline"** - Declines/rejects the requisition
- Both buttons are located in the modal footer
- Buttons are disabled until comments are provided

☑ 2. Mandatory Comments Section

- **Large textarea for comments**
- **Marked as required** with red asterisk (*)
- **Validation:** Both Approve and Decline actions require comments
- **Alert shown if clicked without comments:** "Please provide comments for your decision"
- Comments are sent to the backend and stored in the database

☑ 3. Editable Quantity Fields

- **HOD can modify item quantities** before approving
- Quantity fields shown as **editable input boxes** (number type)
- Only HOD sees editable fields - others see read-only values
- Changes are saved to database when requisition is approved

☑ 4. PDF Download After Approval

- **"Download PDF" button** appears after HOD approves
- Button location: Left side of modal footer
- **Also available to requisitioner** after approval
- Generates professional PDF document with all requisition details
- Auto-downloads with filename: `Requisition_{req_number}.pdf`

☑ 5. Status Flow

Once HOD approves:

- Status changes: `pending_hod` → `hod_approved`
- Requisition moves to **Procurement** stage
- PDF becomes downloadable
- Procurement can add vendors and pricing

💻 How to Use (HOD)

Step 1: Login as HOD

```
Username: mary.mwanza
Password: password123
```

Step 2: View Pending Requisitions

1. You'll see requisitions with status **"Pending HOD"**
2. Click **"View"** button on any requisition

Step 3: Review the Requisition

The modal shows:

- **Basic Information:** Title, description, location, requester
- **Items Table:**
  - Item name
  - **Editable quantity** (you can change this!)
  - Specifications
  - Unit price (if added by procurement)
  - Total price
  - Vendor (if selected)

Step 4: Adjust Quantities (Optional)

- Click in the **Qty** field for any item
- Change the quantity as needed
- System will save these changes when you approve

Step 5: Add Comments

- **Mandatory step!**
- Scroll to the **Comments section**
- Type your justification/decision reason
- Examples:
    - "Approved for Q4 budget. Urgent requirement."
    - "Declined. Budget not available for this quarter. Please resubmit in January."
    - "Approved but reduced quantity from 10 to 5 based on current needs."

Step 6: Make Decision

- Click **"✓ Approve"** to approve (buttons disabled until you add comments)
- Click **"✕ Decline"** to reject

Step 7: Download PDF (After Approval)

- If you approved, the requisition status becomes **"HOD Approved"**
- Click **"📄 Download PDF"** button (left side of footer)
- PDF will be generated and downloaded automatically

📋 Features in Detail

Feature 1: Comments Validation

Frontend Validation:

```
if (!comments.trim()) {
    alert('Please provide comments for your decision');
    return;
}
```

Button State:

- Buttons are **disabled** (grayed out) when comments field is empty
- Buttons are **enabled** (colored) when you type comments

Feature 2: Quantity Editing

Before (Original):

```
Item: Laptop
Quantity: 10  (read-only)
```

After (HOD View):

```
Item: Laptop
Quantity: [  5  ]  (editable input box)
```

How it works:

1. HOD changes quantity from 10 to 5
2. Clicks "Approve"
3. System updates item quantity in database
4. System approves requisition
5. New quantity (5) is used for procurement

Feature 3: PDF Download

When PDF is available:

- Status: `hod_approved`
- Status: `procurement_completed`
- Status: `completed`

What's included in PDF:

- Company header: "PURCHASE REQUISITION"
- Requisition number
- Basic details (location, date, requester, department)
- All items with quantities, specs, prices
- **Grand total** (sum of all item totals)
- Approval details
- Procurement information (vendors, pricing)
- Generated timestamp

🧪 Testing Guide

Test Scenario 1: Approve with Quantity Change

1. **Login as Initiator:**
    - Username: `john.banda`
    - Create requisition with:
        - Item: `Laptop`

        - Quantity: `10`
    - Click **"Submit for Approval"**

2. **Login as HOD:**
    - Username: `mary.mwanza`
    - View the requisition
    - **Change quantity from 10 to 5**
    - Add comment: `"Approved but reduced quantity based on current budget"`
    - Click **"✓ Approve"**

3. **Verify:**
    - Status changes to **"HOD Approved"**
    - Item quantity is now **5** (not 10)
    - Comment is saved
    - PDF download button appears

4. **Download PDF:**
    - Click **"📄 Download PDF"**
    - Open PDF and verify quantity shows **5**

Test Scenario 2: Decline Without Comments

1. **Login as HOD**
2. View a pending requisition
3. **Don't type any comments**
4. Try clicking **"✕ Decline"**

Expected Result:

- Alert appears: `"Please provide comments for your decision"`
- Requisition is **NOT declined**
- You must add comments first

Test Scenario 3: Decline With Comments

1. **Login as HOD**
2. View a pending requisition
3. Type comment: `"Budget not available. Resubmit next quarter."`
4. Click **"✕ Decline"**

Expected Result:

- Alert: `"Requisition declined successfully!"`
- Status changes to **"Rejected"**
- Rejection reason is stored
- Requisitioner can see the reason

Test Scenario 4: PDF Download After Approval

1. Create and submit requisition (as initiator)
2. Approve requisition (as HOD)
3. **Login as initiator** (`john.banda`)
4. View the approved requisition

Expected Result:

- **"📄 Download PDF"** button is visible
- Click it to download
- PDF contains all details

🔧 Technical Implementation

Frontend Changes (index.html)

1. Added State Variables:

```
const [comments, setComments] = useState('');
const [editedItems, setEditedItems] = useState(requisition.items || []);
const [downloadingPDF, setDownloadingPDF] = useState(false);
```

2. Quantity Update Function:

```
const updateItemQuantity = (index, newQuantity) => {
    const updated = [...editedItems];
    updated[index].quantity = parseInt(newQuantity) || 0;
    setEditedItems(updated);
};
```

3. Enhanced Approve Handler:

```
const handleApprove = async (approved) => {
    // Validate comments
    if (!comments.trim()) {
        alert('Please provide comments for your decision');
        return;
    }

    // Update quantities if changed (for HOD)
    if (approved && action === 'hod_approve') {
        for (let item of editedItems) {
            if (originalQuantity !== item.quantity) {
                await updateItemAPI(item);
            }
        }
    }

    // Send approval/decline with comments
    await fetch(`/api/requisitions/${id}/hod-approve`, {
        method: 'PUT',
        body: JSON.stringify({
            user_id: user.id,
            approved: approved,
            comments: comments
        })
    });
};
```

4. PDF Download Function:

```
const downloadPDF = async () => {
    const response = await fetch(`/api/requisitions/${id}/pdf`, {
        headers: { 'Authorization': `Bearer ${token}` }
    });
    const blob = await response.blob();
    const url = window.URL.createObjectURL(blob);
    const a = document.createElement('a');
    a.href = url;
    a.download = `Requisition_${req_number}.pdf`;
    a.click();
};
```

5. Editable Quantity Fields:

```
<td>
    {actionType === 'hod_approve' ? (
        <input
            type="number"
            className="form-control form-control-sm"
            value={item.quantity}
            min="1"
            onChange={(e) => updateItemQuantity(index, e.target.value)}
        />
    ) : (
        item.quantity
    )}
</td>
```

6. Comments Section:

```
{actionType === 'hod_approve' && (
    <div className="mb-4">
        <h6>Comments <span className="text-danger">*</span></h6>
        <textarea
            className="form-control"
            rows="3"
            placeholder="Please provide your comments..."
            value={comments}
            onChange={(e) => setComments(e.target.value)}
            required
        />
        <small className="text-muted">
            Comments are mandatory for both approval and decline
        </small>
    </div>
)}
```

7. Modal Footer Buttons:

```
<div className="modal-footer">
    {/* PDF Download Button */}
    {canDownloadPDF && (
        <button onClick={downloadPDF}>
            📄 Download PDF
        </button>
    )}

    {/* Approve/Decline Buttons */}
    {actionType === 'hod_approve' && (
        <>
            <button
                className="btn btn-success"
                onClick={() => handleApprove(true)}
                disabled={!comments.trim()}
            >
                ✓ Approve
            </button>
            <button
                className="btn btn-danger"
                onClick={() => handleApprove(false)}
                disabled={!comments.trim()}
            >
                ✕ Decline
            </button>
        </>
    )}

    <button className="btn btn-secondary" onClick={onClose}>
        Close
    </button>
</div>
```

Backend Changes (server.js)

1. Added HOD to Item Update Authorization:

```
app.put('/api/requisitions/:id/items/:item_id',
    authenticate,
    authorize('initiator', 'admin', 'procurement', 'hod'),  // Added 'hod'
    (req, res) => {
        // Update item quantity, specifications, etc.
    }
);
```

2. HOD Approval Endpoint Already Supported Comments:

```
app.put('/api/requisitions/:id/hod-approve',
    authenticate,
    authorize('hod', 'admin'),
    (req, res) => {
        const { user_id, approved, comments } = req.body;

        // Validate comments for rejection
        if (!approved && !comments) {
            return res.status(400).json({
                error: 'Comments required when rejecting a requisition'
            });
        }

        // Update requisition status
        const newStatus = approved ? 'hod_approved' : 'rejected';

        db.run(`UPDATE requisitions SET
            status = ?,
            rejection_reason = ?,
            rejected_by = ?,
            rejected_at = CURRENT_TIMESTAMP
            WHERE id = ?
        `, [newStatus, comments, user_id, id]);
    }
);
```

3. PDF Generation Endpoint:

```
app.get('/api/requisitions/:id/pdf',
    authenticate,
    (req, res) => {
        // Fetch requisition with items
        // Generate PDF using pdfGenerator utility
        // Return as downloadable file
    }
);
```

🗄️ Database Schema

Requisitions Table:

```
CREATE TABLE requisitions (
    id INTEGER PRIMARY KEY,
    req_number TEXT,
    status TEXT,  -- 'pending_hod', 'hod_approved', 'rejected'
    rejection_reason TEXT,  -- Stores HOD comments for rejection
    rejected_by INTEGER,
    rejected_at DATETIME,
    -- ... other fields
);
```

Requisition Items Table:

```
CREATE TABLE requisition_items (
    id INTEGER PRIMARY KEY,
    requisition_id INTEGER,
    item_name TEXT,
    quantity INTEGER,  -- Can be modified by HOD
    specifications TEXT,
    unit_price REAL,
    total_price REAL,
    vendor_id INTEGER
);
```

Success Criteria

- [x] HOD sees **Approve** and **Decline** buttons
- [x] **Comments section** is visible and mandatory
- [x] Buttons are **disabled without comments**
- [x] **Quantity fields are editable** for HOD
- [x] Quantity changes are **saved to database**
- [x] Comments are **stored with approval/decline**
- [x] **PDF download button** appears after approval
- [x] PDF downloads successfully with correct filename
- [x] **Requisitioner can download PDF** after HOD approval
- [x] Status changes correctly: `pending_hod` → `hod_approved`
- [x] Declined requisitions show **rejection reason**

Next Steps

For Requisitioner (After HOD Approval):

1. View approved requisitions
2. Download PDF for records
3. Track procurement progress

For Procurement (After HOD Approval):

1. View requisitions with status **"HOD Approved"**
2. Select vendors from dropdown
3. Enter unit prices
4. Complete procurement process

Files Modified

File	Changes
`frontend/index.html`	Added HOD review UI, comments section, quantity editing, PDF download
`backend/server.js`	Added 'hod' to item update authorization
`HOD_REVIEW_FEATURES.md`	This documentation

Summary

HOD Review Features are now complete!

☑ **Approve/Decline buttons** with proper labels
☑ **Mandatory comments** for all decisions
☑ **Editable quantities** before approval
☑ **PDF download** after approval
☑ **Status flow** to procurement after approval

Test it now:

1. Start servers (backend + frontend)
2. Login as `john.banda` → Create requisition
3. Login as `mary.mwanza` (HOD) → Review & Approve
4. Download PDF and verify all details!

Status: ☑ **READY FOR USE**
Last Updated: October 23, 2025