

📋 Purchase Requisition System

A comprehensive web-based system for managing purchase requisitions with multi-level approval workflows, vendor management, analytics, and PDF generation.

version 3.0
node >=14.0.0
license MIT

🚀 Quick Start (5 Minutes)

1. Clone the Repository

```
git clone https://github.com/YOUR-USERNAME/purchase-requisition-system.git  
cd purchase-requisition-system
```

2. Install Dependencies

```
cd backend  
npm install
```

3. Configure Environment

```
cp .env.example .env  
# Edit .env and change JWT SECRET to a secure random string
```

4. Initialize Database

```
node scripts/hashPasswords.js
```

5. Start the Application

```
npm start
```

6. Access the System

Open your browser and navigate to: <http://localhost:3001>

Default Login Credentials:

- **Admin:** admin / admin123
- **MD:** david.mulenga / password123
- **Finance:** sarah.banda / password123
- **Procurement:** james.phiri / password123
- **HOD:** mary.mwanza / password123
- **Initiator:** john.banda / password123

⚠ Change these passwords immediately after first login!

📋 Overview

A full-stack web application for managing purchase requisitions with a complete multi-tier approval workflow designed for KSB Zambia.

Complete Workflow

1. CREATE REQUISITION
 - ↳ Employee creates purchase request
2. HOD APPROVAL
 - ↳ Department Head reviews and approves/rejects
3. PROCUREMENT PROCESSING
 - ↳ Procurement team adds vendor quotes and pricing
4. HOD PRICING REVIEW
 - ↳ HOD reviews and confirms pricing
5. FINANCE APPROVAL
 - ↳ Finance Manager reviews budget and approves
6. MD APPROVAL
 - ↳ Managing Director gives final approval
7. PO GENERATION
 - ↳ System generates Purchase Order PDF
8. AUDIT TRAIL
 - ↳ All actions tracked with timestamps

❖ Key Features

❑ Requisition Management

- Create requisitions with detailed specifications
- Auto-generated requisition numbers (KSB-DEPT-INITIALS-TIMESTAMP)
- Multi-item support with quantities and descriptions
- Urgency levels (Standard, High, Emergency)
- Department categorization
- Account code assignment

❑ User Roles & Permissions

- **Initiator:** Create and track own requisitions
- **HOD:** Approve/reject department requisitions
- **Procurement:** Add vendor quotes and pricing
- **Finance Manager:** Review budget and approve spending
- **Managing Director (MD):** Final approval authority
- **Admin:** Full system access and user management

⌚ Financial Management

- Multi-currency support (ZMW, USD, EUR, GBP, ZAR)
- Real-time exchange rate management
- Budget tracking and reporting
- VAT calculation (16%)
- Department-wise spending analytics

❑ PDF Generation

- **Purchase Orders:** Professional PDFs with company branding
- **Requisitions:** Downloadable requisition documents
- Company logo support
- Role-based access control for downloads
- Automatic calculations (subtotal, VAT, grand total)

❑ Analytics & Reporting

- Executive overview dashboard
- Spending trends (daily/weekly/monthly)
- Department-wise breakdown
- Approval flow analytics
- Processing duration metrics
- Top vendors report
- Interactive charts (Chart.js)
- Export to CSV/Excel

🔒 Security Features

- JWT-based authentication with refresh tokens
- Password hashing (bcrypt)
- Role-based access control (RBAC)
- SQL injection prevention
- Input validation
- CORS protection
- Rate limiting
- Comprehensive audit logging

⌚ User Experience

- Light/Dark theme toggle
- Responsive design (mobile-friendly)
- Real-time notifications
- Intuitive navigation
- Status-based color coding
- Search and filtering

❖ Technology Stack

Backend

- **Runtime:** Node.js (v14+)
- **Framework:** Express.js 5.1.0
- **Database:** SQLite3 5.1.6
- **Authentication:** JWT (`jsonwebtoken`)
- **PDF Generation:** PDFKit
- **Security:** bcryptjs, express-validator
- **Environment:** dotenv

Frontend

- **Framework:** React 18 (CDN-based, no build step required)
- **Styling:** Tailwind CSS (CDN)
- **Charts:** Chart.js 4.4.0
- **Icons:** SVG-based

- **State Management:** React Hooks

Infrastructure

- SQLite database (file-based, no separate DB server needed)
- Single server deployment
- Static file serving
- PM2 or systemd for process management

📁 Project Structure

```

purchase-requisition-system/
├── backend/
│   ├── middleware/
│   │   ├── auth.js          # JWT authentication & RBAC
│   │   ├── validation.js    # Input validation
│   │   ├── errorHandler.js # Error handling
│   │   ├── rateLimiter.js   # Rate limiting
│   │   └── requestLogger.js # Request logging
│   ├── utils/
│   │   ├── auth.js          # Password hashing & tokens
│   │   ├── logger.js        # Winston logger
│   │   └── pdfGenerator.js  # PDF utilities
│   ├── scripts/
│   │   └── hashPasswords.js # Password hashing utility
│   ├── assets/
│   │   ├── logo.png         # Company logo (optional)
│   │   └── README.md        # Asset instructions
│   ├── server.js           # Main Express application
│   ├── database.js         # Database initialization
│   ├── .env.example         # Environment template
│   └── package.json         # Dependencies
├── frontend/
│   ├── index.html          # HTML entry point
│   └── app.js               # React application
└── .gitignore              # Git exclusions
└── README.md               # This file
└── DEPLOYMENT.md          # Deployment guide

```

🔧 Configuration

Environment Variables

Create a `.env` file in the `backend/` directory:

```

# Server Configuration
NODE_ENV=production
PORT=3001

# Database
DATABASE_PATH=./purchase_requisition.db

# JWT Configuration
JWT_SECRET=your-super-secret-random-string-change-this
JWT_EXPIRES_IN=15m
REFRESH_TOKEN_DAYS=7

# CORS
ALLOWED_ORIGINS=https://your-domain.com

# Security
BCRYPT_ROUNDS=10

# Rate Limiting
RATE_LIMIT_WINDOW_MS=900000
RATE_LIMIT_MAX_REQUESTS=100

```

Company Logo

To add your company logo to PDFs:

1. Prepare a PNG/JPG image (400x100px recommended, 4:1 aspect ratio)
2. Copy to `backend/assets/logo.png`
3. Restart the server

📦 Installation Guide

Prerequisites

- Node.js v14 or higher ([Download](https://nodejs.org/) (<https://nodejs.org/>))
- Git ([Download](https://git-scm.com/) (<https://git-scm.com/>))

Development Setup

```

# Clone repository
git clone https://github.com/YOUR-USERNAME/purchase-requisition-system.git
cd purchase-requisition-system

# Install backend dependencies
cd backend
npm install

# Configure environment
cp .env.example .env
nano .env # Edit with your settings

# Initialize database and hash passwords
node scripts/hashPasswords.js

# Start development server
npm start

```

The application will be available at:

- **Frontend:** <http://localhost:3001>
- **API:** <http://localhost:3001/api>

Production Deployment

See [DEPLOYMENT.md \(DEPLOYMENT.md\)](#) for detailed production deployment instructions including:

- Server setup (VPS, cloud)
- PM2 process management
- Nginx reverse proxy
- SSL/HTTPS configuration
- Docker deployment
- Database backups
- Monitoring

API Documentation

Authentication Endpoints

| Method | Endpoint | Description |
|--------|-------------------|----------------------|
| POST | /api/auth/login | User login |
| POST | /api/auth/refresh | Refresh access token |
| POST | /api/auth/logout | User logout |

Requisition Endpoints

| Method | Endpoint | Description | Access |
|--------|---------------------------|--------------------------|------------|
| GET | /api/requisitions | Get all requisitions | All |
| GET | /api/requisitions/:id | Get single requisition | All |
| POST | /api/requisitions | Create requisition | Initiator+ |
| PUT | /api/requisitions/:id | Update requisition | Various |
| GET | /api/requisitions/:id/pdf | Download requisition PDF | Role-based |

Purchase Order Endpoints

| Method | Endpoint | Description | Access |
|--------|------------------------------|-----------------|--------|
| GET | /api/purchase-orders | Get all POs | All |
| GET | /api/purchase-orders/:id | Get single PO | All |
| GET | /api/purchase-orders/:id/pdf | Download PO PDF | All |

Analytics Endpoints

| Method | Endpoint | Description | Access |
|--------|-------------------------------------|--------------------|---------------------------------|
| GET | /api/analytics/overview | Executive overview | Finance, MD, Admin |
| GET | /api/analytics/spending-trend | Spending trends | Finance, MD, Admin |
| GET | /api/analytics/department-breakdown | Dept analysis | Finance, MD, Admin |
| GET | /api/analytics/approval-flow | Approval funnel | Finance, MD, Admin |
| GET | /api/analytics/duration | Processing times | Finance, MD, Admin |
| GET | /api/analytics/top-vendors | Vendor metrics | Finance, MD, Admin, Procurement |

User Management Endpoints

| Method | Endpoint | Description | Access |
|--------|----------------|---------------|--------|
| GET | /api/users | Get all users | Admin |
| POST | /api/users | Create user | Admin |
| PUT | /api/users/:id | Update user | Admin |
| DELETE | /api/users/:id | Delete user | Admin |

Security

Implemented Security Measures

Authentication & Authorization

- JWT-based authentication with 15-minute access tokens
- Refresh tokens with 7-day expiration
- Role-based access control (RBAC)
- Password hashing with bcrypt (10 rounds)

API Security

- Rate limiting (100 requests per 15 minutes)
- Input validation on all endpoints
- SQL injection prevention (parameterized queries)
- CORS protection
- Request logging

Data Protection

- No sensitive data in error messages
- Environment variables for secrets
- Audit logging for all actions
- Session management

Best Practices

- HTTPS recommended for production
- .env file not committed to Git
- Stack traces hidden in production
- Regular security audits

Security Checklist Before Production

- [] Change all default passwords
- [] Update `JWT_SECRET` to a strong random string
- [] Configure `ALLOWED_ORIGINS` with your domain
- [] Enable HTTPS/SSL
- [] Set `NODE_ENV=production`
- [] Configure firewall
- [] Enable database backups
- [] Review user permissions

✍ Testing

Manual Testing Workflow

1. Create Requisition (as john.banda)
2. HOD Approval (as mary.mwanza)
3. Procurement Processing (as james.phiri)
4. HOD Pricing Review (as mary.mwanza)
5. Finance Approval (as sarah.banda)
6. MD Approval (as david.mulenga)
7. Download PO (any user)

Test User Accounts

All test accounts use password: `password123` (except admin: `admin123`)

| Username | Role | Department |
|---------------|-----------------|-------------|
| admin | Admin | IT |
| david.mulenga | MD | Management |
| sarah.banda | Finance Manager | Finance |
| james.phiri | Procurement | Procurement |
| mary.mwanza | HOD | Operations |
| john.banda | Initiator | Operations |

🗄 Database Schema

Main Tables

- `users`: User accounts and roles
- `requisitions`: Purchase requisitions
- `purchase_orders`: Generated POs
- `vendors`: Vendor information
- `departments`: Department list
- `fx_rates`: Exchange rates
- `budgets`: Budget allocations
- `audit_log`: Action history

Key Relationships

```
users (1) —— (N) requisitions
requisitions (1) —— (1) purchase_orders
vendors (1) —— (N) requisitions
departments (1) —— (N) users
```

Troubleshooting

Common Issues

Issue: Can't login

```
# Solution: Reset passwords  
cd backend  
node scripts/hashPasswords.js
```

Issue: Port 3001 already in use

```
# Solution: Find and kill the process  
netstat -ano | findstr :3001  
# Kill the PID shown  
taskkill /PID <PID> /F
```

Issue: Database locked

```
# Solution: Ensure only one instance is running  
pm2 stop all  
# OR check for zombie processes  
ps aux | grep node
```

Issue: CORS errors

```
# Solution: Update .env ALLOWED_ORIGINS  
ALLOWED_ORIGINS=http://localhost:3000,https://your-domain.com
```

License

MIT License - See LICENSE file for details

Contributing

1. Fork the repository
2. Create a feature branch (`git checkout -b feature/amazing-feature`)
3. Commit your changes (`git commit -m 'Add amazing feature'`)
4. Push to the branch (`git push origin feature/amazing-feature`)
5. Open a Pull Request

Support

For questions or issues:

1. Check [DEPLOYMENT.md \(DEPLOYMENT.md\)](#) for deployment help
2. Review this README
3. Check application logs
4. Open a GitHub issue

Version History

v3.0 (October 2025)

- Requisition PDF download with role-based access
- Company logo support for PDFs
- Compact single-page PO layout
- VAT calculations (16%)
- Updated KSB Zambia branding

v2.1 (October 2025)

- Analytics dashboard with Chart.js
- Export functionality (CSV, Excel)
- Interactive drill-down charts
- Date range presets
- Multi-currency support

v2.0 (October 2025)

- Complete approval workflow
- Purchase Order generation
- Vendor management
- Budget tracking
- Admin console

v1.0 (October 2025)

- Initial release
- Basic requisition management
- User authentication
- Role-based access

Built with ❤ for KSB Zambia

Last Updated: October 31, 2025 | Version: 3.0