

Purchase Requisition System - Deployment Guide

Quick Start Deployment (5 Minutes)

Prerequisites

- Node.js (v14 or higher)
- Git
- A server with port 3001 available

Step 1: Clone from GitHub

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

Step 2: Backend Setup

```
cd backend
npm install
```

Step 3: Configure Environment Variables

```
# Copy the example environment file
cp .env.example .env

# Edit .env file with your settings
# IMPORTANT: Change JWT_SECRET to a secure random string
# Update ALLOWED_ORIGINS with your production domain
```

Required .env changes:

```
NODE_ENV=production
JWT_SECRET=your-super-secret-random-string-here
ALLOWED_ORIGINS=https://your-domain.com
```

Step 4: Initialize Database

```
# Hash the default passwords
node scripts/hashPasswords.js

# Database will be created automatically on first run
```

Step 5: Start the Application

```
# Start backend server
npm start

# Server will run on http://localhost:3001
# Frontend is served from http://localhost:3001
```


Step 6: Access the Application

Open your browser and navigate to:

```
http://your-server-ip:3001
```

Default Login Credentials:

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

 **IMPORTANT:** Change these passwords immediately after first login!

Production Deployment Options

Option 1: Simple Server Deployment (VPS/Dedicated Server)

1. Using PM2 (Recommended for Production)

```
# Install PM2 globally
npm install -g pm2

# Start the application
cd backend
pm2 start server.js --name "purchase-requisition"

# Enable startup on boot
pm2 startup
pm2 save

# Monitor the application
pm2 status
pm2 logs purchase-requisition
```

2. Using systemd (Linux)

Create a **service** file: /etc/systemd/system/purchase-requisition.service

```
[Unit]
Description=Purchase Requisition System
After=network.target

[Service]
Type=simple
User=your-username
WorkingDirectory=/path/to/purchase-requisition-system/backend
ExecStart=/usr/bin/node server.js
Restart=on-failure
Environment=NODE_ENV=production

[Install]
WantedBy=multi-user.target
```

Enable and start the service:

```
sudo systemctl enable purchase-requisition
sudo systemctl start purchase-requisition
sudo systemctl status purchase-requisition
```

Option 2: Docker Deployment

Create a Dockerfile in the backend directory:

```
FROM node:18-alpine
WORKDIR /app
COPY package*.json ./
RUN npm ci --only=production
COPY . .
EXPOSE 3001
CMD ["node", "server.js"]
```

Create docker-compose.yml in the root directory:

```
version: '3.8'
services:
  backend:
    build: ./backend
    ports:
      - "3001:3001"
    environment:
      - NODE_ENV=production
    volumes:
      - ./backend/purchase_requisition.db:/app/purchase_requisition.db
      - ./backend/assets:/app/assets
    restart: unless-stopped
```

Deploy with Docker:

```
docker-compose up -d
```

Option 3: Cloud Platform Deployment

Heroku

```
# Install Heroku CLI
heroku create your-app-name

# Set environment variables
heroku config:set NODE_ENV=production
heroku config:set JWT_SECRET=your-secret-key

# Deploy
git push heroku main
```

DigitalOcean App Platform

1. Connect your GitHub repository

2. Select the `backend` directory as the source
3. Set environment variables in the dashboard
4. Deploy

AWS EC2

1. Launch an EC2 instance (Ubuntu recommended)
2. SSH into the instance
3. Follow "Simple Server Deployment" steps above
4. Configure security group to allow port 3001

Nginx Reverse Proxy Setup (Recommended)

This allows you to run the app on port 80/443 with SSL.

Install Nginx

```
sudo apt update
sudo apt install nginx
```

Configure Nginx

Create `/etc/nginx/sites-available/purchase-requisition:`

```
server {
    listen 80;
    server_name your-domain.com;

    location / {
        proxy_pass http://localhost:3001;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

Enable the site:

```
sudo ln -s /etc/nginx/sites-available/purchase-requisition /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl reload nginx
```

Add SSL with Let's Encrypt

```
sudo apt install certbot python3-certbot-nginx
sudo certbot --nginx -d your-domain.com
```

Update your `.env` file:

```
ALLOWED_ORIGINS=https://your-domain.com
```

Security Checklist

Before going to production, ensure:

- ☐ Changed `JWT_SECRET` to a strong random string
- ☐ Updated `ALLOWED_ORIGINS` with your production domain
- ☐ Changed all default user passwords
- ☐ Enabled HTTPS/SSL
- ☐ Configured firewall (allow only ports 80, 443, 22)
- ☐ Set `NODE_ENV=production`
- ☐ Database backups configured
- ☐ Monitoring and logging enabled
- ☐ Rate limiting configured properly

Adding Company Logo

To display your company logo on Purchase Orders and Requisitions:

1. Prepare your logo:
 - Format: PNG (transparent background recommended) or JPG
 - Size: 400x100 pixels (4:1 aspect ratio)
 - File size: Under 500KB
2. Upload to server:

```
# Copy your logo to the backend assets folder
cp /path/to/your/logo.png backend/assets/logo.png
```

3. Restart the application:

```
pm2 restart purchase-requisition
# OR
sudo systemctl restart purchase-requisition
```

The logo will automatically appear on all PDF downloads.

Database Backup

Manual Backup

```
# Backup the database
cp backend/purchase_requisition.db backend/purchase_requisition.db.backup

# Or with timestamp
cp backend/purchase_requisition.db backend/purchase_requisition.db.$(date +%Y%m%d_%H%M%S)
```

Automated Daily Backup (Linux)

Create a backup script /usr/local/bin/backup-requisition-db.sh:

```
#!/bin/bash
BACKUP_DIR="/var/backups/purchase-requisition"
DB_PATH="/path/to/purchase-requisition-system/backend/purchase_requisition.db"
DATE=$(date +%Y%m%d_%H%M%S)

mkdir -p $BACKUP_DIR
cp $DB_PATH $BACKUP_DIR/purchase_requisition.db.$DATE

# Keep only last 30 days of backups
find $BACKUP_DIR -name "*.db.*" -mtime +30 -delete
```

Add to crontab:

```
sudo chmod +x /usr/local/bin/backup-requisition-db.sh
crontab -e
# Add this line:
0 2 * * * /usr/local/bin/backup-requisition-db.sh
```

Troubleshooting

Application won't start

```
# Check if port 3001 is already in use
netstat -tulpn | grep 3001

# Check application logs
pm2 logs purchase-requisition
# OR
journalctl -u purchase-requisition -n 50
```

Database errors

```
# Check database file permissions
ls -l backend/purchase_requisition.db

# Ensure the backend user has write permissions
chmod 644 backend/purchase_requisition.db
```

CORS errors

Ensure your .env file has the correct domain:

```
ALLOWED_ORIGINS=https://your-actual-domain.com
```

Can't login

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

Monitoring

Check Application Status

```
# PM2
pm2 status

# Systemd
sudo systemctl status purchase-requisition

# Check logs
pm2 logs purchase-requisition --lines 100
```

Monitor Server Resources

```
# CPU and Memory usage
top
htop

# Disk space
df -h

# Application-specific
pm2 monit
```

Updating the Application

```
# Stop the application
pm2 stop purchase-requisition

# Pull latest changes
git pull origin main

# Update dependencies
cd backend
npm install

# Restart application
pm2 restart purchase-requisition
```

Support

- For issues or questions:
1. Check the troubleshooting section above
 2. Review logs: `pm2 logs purchase-requisition`
 3. Check GitHub issues
 4. Contact your system administrator

System Requirements

- Minimum:**
- CPU: 1 core
 - RAM: 512MB
 - Disk: 1GB
 - Node.js: v14+

- Recommended for Production:**
- CPU: 2+ cores
 - RAM: 2GB+
 - Disk: 10GB+
 - Node.js: v18+

Last Updated: October 31, 2025
Version: 3.0