# IT Requisition System - Deployment Guide

This guide provides step-by-step instructions for deploying the IT Requisition System on a new Windows server (e.g., a desktop PC or a server acting as a local host, compatible with QNAP NAS web server environments). It assumes a clean installation and aims to be easy to follow for anyone, including non-developers.

## 🎯 Goal

To successfully set up and run the IT Requisition System, making it accessible on your local network.

## 📋 Prerequisites

Before you begin, ensure the following are installed and configured on your **new server**:

1. **Python 3.x:** \* **Download:** Go to [python.org](https://www.python.org/downloads/windows/) and download the latest Python 3.x installer for Windows.
   * **Installation:** \* Run the installer.
     + **CRITICAL STEP:** On the first screen, **check the box that says "Add Python.exe to PATH"** before clicking "Install Now". This is essential for running Python commands from any directory.
     + Complete the installation.
   * **Verification:** Open Command Prompt and type python --version. You should see Python 3.x.x.
2. **Network IP Address:** \* Your server needs a stable local IP address (e.g., 192.168.0.3 or 192.168.1.105).
   * **To find your IP:** Open Command Prompt and type ipconfig. Look for "IPv4 Address" under your active network adapter (e.g., "Ethernet adapter Ethernet" or "Wireless LAN adapter Wi-Fi").
   * **Recommendation:** If possible, configure a static IP address for your server in your router settings to prevent it from changing.

## 📦 Application Structure

The application consists of two main parts: a frontend (HTML, CSS, JavaScript) and a backend (Python Flask API).

IT\_Requisition\_App/

├── backend/

│ ├── data/

│ │ └── requisitions.db <-- SQLite database (will be created automatically)

│ └── app.py <-- Python Flask backend API

├── frontend/

│ └── index.html <-- Main frontend application file

├── start\_it\_app.bat <-- Batch script to start both servers

├── requirements.txt <-- Python dependencies for Flask

├── app\_context\_history.json <-- Application version and feature history

├── migrate\_db.py <-- Database migration script

└── Deployment Guide.md <-- This guide

## 🚀 Deployment Steps

Follow these steps carefully to deploy the application:

### Step 1: Transfer Application Files

1. Create a new folder on your server, for example: C:\IT\_Requisition\_App.
2. Copy all the files and folders from the provided application package into this IT\_Requisition\_App folder.
   * Ensure the backend and frontend subdirectories are correctly placed.
   * **Ensure migrate\_db.py is in the root IT\_Requisition\_App/ directory.**

### Step 2: Install Python Dependencies

1. Open Command Prompt.
2. Navigate to your application's root directory:
3. cd C:\IT\_Requisition\_App
4. Install the required Python libraries using pip:
5. pip install -r requirements.txt
   * You should see messages indicating successful installation of Flask and Flask-Cors.

### Step 3: Configure Frontend API Base URL

1. Open the IT\_Requisition\_App\frontend\index.html file in a text editor (like Notepad, Notepad++, VS Code).
2. Locate the line that defines API\_BASE\_URL:
3. const API\_BASE\_URL = 'http://192.168.0.3:5000/api/requisitions';
4. **Replace 192.168.0.3 with the actual local IP address of your server** (the one you found in Prerequisites Step 2).
   * **Example:** If your server's IP is 192.168.1.105, change the line to:
   * const API\_BASE\_URL = 'http://192.168.1.105:5000/api/requisitions';
5. Save the index.html file.

### Step 4: Configure Firewall Rules

The application uses two ports:

* **Backend:** Port 5000 (for the Flask API)
* **Frontend:** Port 8000 (for the simple HTTP server)

You need to allow inbound TCP connections on these ports in your server's Windows Firewall.

1. Open **Windows Defender Firewall with Advanced Security**:
   * Press Win + R, type wf.msc, and press Enter.
   * Alternatively, search for "Windows Defender Firewall" in the Start Menu, then click "Advanced settings".
2. In the left pane, click **Inbound Rules**.
3. In the right pane, click **New Rule...**.
4. Follow the New Inbound Rule Wizard:
   * **Rule Type:** Select Port, then click Next.
   * **Protocol and Ports:** Select TCP. For "Specific local ports", enter 5000, 8000. Click Next.
   * **Action:** Select Allow the connection. Click Next.
   * **Profile:** Ensure Domain, Private, and Public are all checked (or at least Private and Public if this is a home/small office network). Click Next.
   * **Name:** Give the rule a descriptive name, e.g., IT Requisition App Ports. Click Finish.

### Step 5: Start the Application Servers

1. Navigate to your application's root directory (C:\IT\_Requisition\_App) in File Explorer.
2. **Double-click start\_it\_app.bat**.
   * This batch script will now:
     + First, run the migrate\_db.py script. This will happen in the main batch window and will complete before the next steps.
     + Then, open two more Command Prompt windows: one for the Flask backend (port 5000) and one for the simple Python HTTP frontend server (port 8000).
   * **Do NOT close the backend and frontend server windows.** They must remain open for the application to run. The main batch window will pause after launching them.

### Step 6: Access the Application

1. Open a web browser (Chrome, Firefox, Edge, etc.) on any computer on your local network.
2. In the address bar, type the following URL, replacing YOUR\_PC\_LOCAL\_IP with the actual IP address of your server:
3. http://YOUR\_PC\_LOCAL\_IP:8000/index.html
   * **Example:** If your server's IP is 192.168.0.3, you would go to http://192.168.0.3:8000/index.html.
4. The IT Requisition System should now load in your browser!

## 🛑 Stopping the Application

To stop the application, you need to close both server processes:

1. Go to the two Command Prompt windows that opened when you ran start\_it\_app.bat (one for Flask, one for http.server).
2. In each window, press Ctrl + C. You might be asked to confirm (Terminate batch job (Y/N)?). Press Y and Enter.
3. The windows will close.

## 💡 Important Notes & Troubleshooting

* **Automated Database Migrations:** The start\_it\_app.bat script now automatically runs migrate\_db.py every time you start the application. This ensures your requisitions.db is always up-to-date with the latest schema, preventing errors like "no column named display\_id".
* **Data Persistence:** All your requisition data is now stored in requisitions.db located in IT\_Requisition\_App\backend\data\. To back up your data, simply copy this .db file.
* **New Sequential Requisition IDs:** The system now generates sequential requisition numbers (e.g., RQ001, RQ002) for new requisitions.
  + The migrate\_db.py script will also populate sequential RQxxx numbers for any existing requisitions that don't have a display\_id yet, ensuring all records have a consistent display format.
* **IP Address Changes:** If your server's local IP address changes in the future, you **must** update the API\_BASE\_URL in index.html (Step 3) and restart the servers.
* **"File not found" (404) error:** Ensure you are running the http.server from the main IT\_Requisition\_App directory (where index.html is located), not from a subfolder. The start\_it\_app.bat script handles this automatically.
* **"Failed to submit requisition" / "Network Error":** \* Verify both Command Prompt windows are open and running without errors.
  + Double-check the IP address in index.html (Step 3) matches your server's current IP.
  + Ensure firewall rules (Step 5) are correctly configured.
* **"Submit Request" button not working / Modals not appearing:** \* Perform a "hard refresh" in your browser (Ctrl + F5 or Cmd + Shift + R on Mac) to ensure the latest index.html is loaded. Browser caching can sometimes prevent new JavaScript from loading.
* **New Features:** This version includes support for **IT Requisitions**, **Conference Room Requisitions**, and **Leave Requests**. The Manager Portal is now split to handle Leave Requests separately from IT/CR requests for better organization. Sequential requisition numbers (e.g., RQ001) are now used for display. The header now features a **centered "PES Requisition System" title** and a **top-left "Home" button** for easier navigation. The **password modal now redirects to the home page** if dismissed without successful authentication. The header layout is also **fully responsive for mobile views**.