THE CONTRACT BOARD

INSTALL GUIDE

Table of Contents

1. Introduction

Who Should Use This Guide?

This Installation Guide is written specifically for IT professionals, system administrators, and technical operations teams. It provides the step-by-step instructions required to deploy, configure, and initialize The Contract Board on your organization's internal infrastructure.

> **Looking for the User Manual?**
> If you are a legal professional or end-user looking for instructions on how to navigate the software's interface, please refer to the User Manual instead. This guide covers technical deployment only.

Technical Architecture

The Contract Board is a modern, lightweight web application designed for straightforward deployment. The core product is a dynamic, client-side application written in React.js, designed to be hosted on any standard web server capable of serving production-ready application files. It communicates with an accompanying PostgreSQL backend and database.

What Is in the Zip File?

When you download The Contract Board, you will receive a single compressed zip file containing everything needed for a complete deployment. Upon extraction, you will find the following core components:

- **/build (or /client)** — The compiled, production-ready React.js application files.
- **/server** — The backend application files, routing logic, and configuration templates.
- **/database** — The initialization scripts required to set up your database schema and tables.
- **install-check.sh** — A simple script to verify that your environment meets all necessary prerequisites before beginning installation.

2. System Prerequisites

Before deploying The Contract Board, ensure your hosting environment meets the following minimum requirements. The software is highly portable and can be hosted on a standalone server, a virtual machine, or within containerized environments.

Server & Runtime Environment

Requirement	Details
Operating System	Any modern OS capable of running Node.js (Linux, Windows Server, or macOS). Linux (e.g., Ubuntu, Debian, or RHEL) is strongly recommended for production environments.
Node.js	A recent LTS (Long Term Support) version of Node.js is required to run the backend server.

Database Requirements

Requirement	Details
PostgreSQL	The Contract Board uses PostgreSQL for robust, secure data storage. A PostgreSQL instance must be running — either locally on the same server or hosted remotely.
Database Credentials	The IT administrator will need the ability to create a new, empty database and a dedicated PostgreSQL user with privileges to run initialization scripts and modify schemas.

Network Requirements

Requirement	Details
Open Port	By default, the Node.js backend requires an open port to serve the web application — Port 80 for HTTP, or Port 443 for HTTPS if placed behind a reverse proxy such as NGINX or Apache.

3. Environment Setup

Step 1: Extract the Application Files

- Move the downloaded zip file to the directory on your server where you intend to host the application.
- Example locations: /var/www/thecontractboard on Linux, or C:\inetpub\thecontractboard on Windows.
- Extract the contents of the zip file. You will see the core folder structure emerge, including the frontend application, backend server files, and database initialization scripts.

Step 2: Database Preparation

- Open your PostgreSQL command-line tool or a graphical database manager such as pgAdmin.
- Create a new, completely empty database specifically for the application:

```
CREATE DATABASE contractboard_db;
```

- Ensure the PostgreSQL user account you plan to use has full read/write privileges and the ability to create tables within this new database.

Step 3: Configure Environment Variables

The Contract Board uses environment variables to securely connect to your organization's specific infrastructure without hard-coding sensitive information.

- Navigate to the backend server directory.
- Locate the sample configuration file (typically named .env.example) and duplicate it. Rename the new file to exactly .env.
- Open the .env file in a text editor and configure the following:
  - **Database Credentials** — Enter the PostgreSQL database name, user, password, host, and port (default: 5432) set up in Step 2.
  - **Server Port** — Specify the port for the backend Node.js server to listen on (e.g., PORT=3000 or PORT=8080).
  - **Security Secrets** — Provide secure random strings for the session cookie secret and the JSON Web Token (JWT) secret. These ensure user logins and sessions remain encrypted.

4. First-Time Initialization

Step 1: Initialize the Database Schema

The Contract Board requires a specific table structure to operate. A ready-to-use SQL script is included to build this structure automatically.

- Navigate to the /database folder in your extracted files.
- Execute the schema.sql file against your PostgreSQL database via the command line or pgAdmin:

```
psql -U your_user -d contractboard_db -f schema.sql
```

- This script will automatically create all necessary tables (such as users, contracts, and user_settings), and will insert a default configuration row into the system_settings table so the application has a baseline to read from on startup.

Step 2: Install Node Dependencies

Before the backend server can run, it needs to download its required code libraries.

- Open your terminal or command prompt and navigate to the /server directory.
- Run the following command:

```
npm install
```

- This will read the application's configuration and download all necessary security, database, and routing packages required to run the server.

Step 3: Start the Application Server

With the database ready and dependencies installed, you can now start the application.

- For a simple test run, execute:

```
npm start
# or: node server.js
```

> **Production Recommendation**
> For a live production environment, use a process manager like PM2 (e.g., pm2 start server.js) or a systemd service. This ensures the server automatically restarts if it crashes or if the physical machine reboots.

Step 4: Verification and First Login

- Open a web browser and navigate to the server's IP address or domain name (e.g., http://localhost:3000).
- You should see The Contract Board login screen.
- By default, the system allows initial account creation from the login screen via the "Register For An Account" link.

> **Important: First Account = System Administrator**
> The software is specifically designed so that the very first account created is automatically designated as a System Administrator. This means whoever creates the first account receives full administrative privileges, including the ability to manage settings and promote other users. Consider carefully whether the IT team or a Legal team member should create the first account. If IT creates it and does not need ongoing admin access, have Legal create the first account directly, or create the account and then promote a Legal colleague to System Administrator afterward via the Admin Settings page.

- Every account created after the first is automatically assigned as a Read-Only User until a System Administrator promotes them. There is no limit on the number of System Administrators or Read-Write Users that can exist simultaneously.

5. Backing Up the PostgreSQL Database

Manual Backups Required

> **Important**
> The Contract Board does not automatically back up your SQL data. Because the software is hosted entirely on your organization's own infrastructure, System Administrators are responsible for backing up the PostgreSQL database.

We highly recommend scheduling an automated daily backup using standard server tools — such as a cron job on Linux or Task Scheduler on Windows — to ensure your organization's data is consistently protected.

Reference Commands

The following standard command-line tools can be used to manage your data. These commands are also available for reference on the Administrator Tools page within the software.

Backup

```
pg_dump -Fc -d contractboard > contractboard_backup_$(date +%Y%m%d).dump
```

Restore (Destructive)

```
pg_restore -d contractboard --clean contractboard_backup_YYYYMMDD.dump
```

Docker Example

```
docker exec -t contractboard-db pg_dump -U postgres contractboard > backup.dump
```

6. Troubleshooting Common Installation Issues

The following table covers the most common errors encountered during first-time setup and their recommended resolutions.

"Missing Modules" / Server Crash on Startup

The Problem	You attempt to start the server, but the console immediately throws a "module not found" error or the application fails to boot entirely.
The Solution	This almost always means Node dependencies were not downloaded. Ensure you have navigated to the correct directory containing the package.json file and run npm install. This single command downloads all required components, including security libraries, database connectors, and icon packs.

Frontend Not Loading / "Connection Refused"

The Problem	The backend Node.js server is running without errors, but when you navigate to the application URL in your browser, the page is blank or fails to connect.
The Solution	Ensure you are serving the compiled production frontend, not a development server. Your web server (NGINX or Apache) must be configured to point to the /build (or /client) directory to serve the visual interface, while proxying API requests to the Node.js backend on its designated port.

Invalid Database Credentials

The Problem	The server starts, but logs show "Authentication failed" or "Unable to connect to database."
The Solution	Double-check your .env configuration file. Ensure the PostgreSQL username, password, and database name exactly match the database you created in Step 2. Also verify that the PostgreSQL service is actively running on your host machine.