Nonprofit Fund Accounting System v8.8

Installation Guide – VirtualBox + Windows 11

This guide walks you through spinning up a clean Windows 11 guest in Oracle VM VirtualBox, installing all prerequisites, cloning the v8.8 code-base, configuring PostgreSQL, and running the application locally. Every step has been tested on VirtualBox 7.x with a Windows 11 Pro 23H2 ISO.

---

1. Prerequisites & System Requirements

Host requirement	Minimum	Recommended
Host OS	Windows 10/11, macOS 12+, or Linux	—
CPU	4 cores	6 + cores with VT-x/AMD-V enabled
RAM	8 GB	16 GB (leave ≥ 8 GB for guest)
Disk space	40 GB free	80 GB SSD/NVMe
Software	Oracle VirtualBox 7.x, Windows 11 ISO, Git	—

⚠ Enable hardware virtualization (Intel VT-x/AMD-V) in the host BIOS/UEFI **before** you start.

---

2. VirtualBox Setup & Windows 11 Installation

1. **Download**
   • Oracle VM VirtualBox: https://www.virtualbox.org/wiki/Downloads
   • Windows 11 ISO: https://www.microsoft.com/software-download/windows11
2. **Create a new VM**
   - Name: `Win11-FundAcct-v8_8`
   - Type: *Microsoft Windows*, Version: *Windows 11 (64-bit)*
   - Memory size: **8192 MB** (or ½ of host RAM)
   - Virtual hard disk: **VDI**, dynamically allocated, **60 GB**.
3. **Adjust VM settings**
   - System ➔ Processor: 4 CPUs (enable PAE/NX).
   - Display ➔ Graphics Controller: *VBoxSVGA* ➔ enable 3D Acceleration.
   - Storage ➔ click empty optical drive ➔ **Choose a disk file…** ➔ select Windows 11 ISO.
   - Network ➔ Adapter 1: *Bridged Adapter* or *NAT* (either works).
4. **Boot & install Windows 11** – follow Microsoft out-of-box experience.
   Screenshot `01-oobe.png`: choose “Set up for personal use”.
   ❑ Create a local admin account, e.g. `fundadmin`.

---

3. Development Environment

Open **PowerShell (Admin)** inside the VM and run:

```
# 3-a. Install Chocolatey (package manager)
Set-ExecutionPolicy Bypass -Scope Process -Force; `
[System.Net.ServicePointManager]::SecurityProtocol = 3072; `
iex ((New-Object
        System.Net.WebClient).DownloadString('https://community.chocolatey.or
        install.ps1'))


# 3-b. Use choco to install tools
choco install -y git nodejs-lts postgresql16 vscode
```

Component	Version tested
Node.js	20 LTS
PostgreSQL	16.x
Git	2.44
VS Code	1.90

After PostgreSQL installer finishes it shows a **superuser password dialog** – set `postgres` → `P@ssw0rd!`.

---

4. Cloning the Repository & Installing Dependencies

```
# 4-a. Clone repo
cd $HOME\source
git clone https://github.com/tpfbill/nonprofit-fund-accounting.git
cd nonprofit-fund-accounting
git checkout v8.8      # ensure you are on the tag

# 4-b. Install Node dependencies
npm install            # installs server + client packages
```

Screenshot `02-git-clone.png` shows successful clone at commit `444f887`.

---

5. Database Configuration & Setup

1. **Create a DB user & database**

```
psql -U postgres -W       # password: P@ssw0rd!
CREATE ROLE funduser LOGIN PASSWORD 'fundpass';
```

```
CREATE DATABASE fundacct OWNER funduser;
\q
```

2. **Load schema & seed data**

```
cd .\database
psql -U funduser -d fundacct -f schema.sql
psql -U funduser -d fundacct -f seed-data.sql
```

3. **Configure connection string**

Create `.env` in project root:

```
PGHOST=localhost
PGPORT=5432
PGDATABASE=fundacct
PGUSER=funduser
PGPASSWORD=fundpass
```

---

6. Running the Application

In **two** PowerShell terminals:

```
# Terminal 1 – backend
cd nonprofit-fund-accounting
node server.js

# Terminal 2 – frontend (static)
npx http-server . -p 8080 --no-cache
```

Open `http://localhost:8080/index.html` in the guest browser.
Landing dashboard should load with sample metrics ("Dashboard cards loading…" ➔ populates).

---

7. Testing the Installation

Test	Expected result
Dashboard loads	Summary cards + charts appear without errors
Documentation tab	Opens `direct-docs.html`, styled white (no underline)
Fund Reports	Dropdown lists **all funds**
Inter-Entity Transfer wizard	Form loads, API calls succeed
Database status badge	Shows **Connected** (green)

Run `npm test` to execute unit tests (if present).

8. Troubleshooting

Symptom	Fix
"DB Offline" badge	Verify PostgreSQL running; check `.env` credentials; `psql` to test.
Dashboard tabs unresponsive	Ensure you served frontend on **8080** and backend on **3000**; clear browser cache (`Ctrl+Shift+R`).
Documentation tab pink/purple	Cached CSS—hard refresh or delete cached file at `%LOCALAPPDATA%\Microsoft\Edge\User Data\Default\Cache`.
Port 3000 already in use	`netstat -aon | find "3000"` ➔ `taskkill /PID <pid> /F`
`node-gyp` build fails (rare)	`npm install -g windows-build-tools` and retry.

9. Performance Optimizations

1. **Increase VM CPUs & RAM** in VirtualBox settings.
2. Enable **I/O APIC** and **Hyper-V Paravirtualization Interface** (System ➔ Acceleration).
3. Place the VM disk on SSD/NVMe.
4. Inside Windows guest:
    - Turn off **Real-time Protection** (Windows Security) for dev work.
    - Disable **Startup apps** via Task Manager.
5. Use **pgTune** or set `shared_buffers = 512MB` in `postgresql.conf`.

10. Security Considerations

- **Keep host & guest patched** – run Windows Update after installation.
- Store secrets in **`.env.local`** (never commit to Git).
- Restrict PostgreSQL to listener `127.0.0.1`.
- Change default passwords (`fundpass`, `P@ssw0rd!`) before production.
- Use **HTTPS reverse proxy** (e.g., Nginx) if exposing outside VM.
- Enable **VirtualBox snapshots** to roll back quickly after testing malware or risky data.

Appendix A – Useful Commands

```
# stop servers cleanly
pkill -f "http-server"
```

```
pkill -f "node server.js"

# backup database
pg_dump -U funduser -Fc fundacct > backup.dump

# restore
pg_restore -U funduser -d fundacct -c backup.dump
```

Enjoy your fully-functional Nonprofit Fund Accounting System v8.8! If you encounter issues not covered here, open an issue on GitHub or consult the `direct-docs.html` library inside the app for detailed PDF guides.