Nonprofit Fund Accounting System v8.8

Installation Guide – VirtualBox on Windows 11 (Ubuntu 24.04 LTS Guest)

> Architecture overview
> • **Host OS:** Windows 11 – where Oracle VM VirtualBox is installed
> • **Virtualisation:** VirtualBox 7.x running on the Windows 11 host
> • **Guest OS:** Ubuntu Desktop 24.04 inside the VirtualBox VM
> • **Application:** Non-profit Fund Accounting System v8.8 installed **inside the Ubuntu guest** under `/opt/nonprofit-fund-accounting`

The steps below walk through:
1. Installing / configuring VirtualBox on the Windows 11 host
2. Creating an Ubuntu 24.04 VM in VirtualBox
3. Installing prerequisite packages inside the Ubuntu guest
4. Cloning the repository, configuring PostgreSQL 16, loading schema / seed data
5. Running the application and verifying functionality

---

1 Prerequisites & Host Requirements

Host requirement	Minimum	Recommended
Host OS	Windows 10/11, macOS 12+, or Linux	—
CPU	4 cores with VT-x/AMD-V	6+ cores
RAM	8 GB	16 GB (allocate ≥ 6 GB to guest)
Disk space	40 GB free	80 GB SSD/NVMe
Software	Oracle VirtualBox ≥7.0, Ubuntu 24.04 ISO	—

⚠ Enable hardware virtualization (Intel VT-x/AMD-V) in BIOS/UEFI before proceeding.

---

2 VirtualBox VM Setup & Ubuntu 24.04 Installation

1. **Download**
   • VirtualBox: https://www.virtualbox.org/wiki/Downloads
   • Ubuntu 24.04 ISO: https://ubuntu.com/download/desktop

2. **Create a new VM**

   - Name: `Ubuntu24-FundAcct-v8_8`
   - Type: *Linux* → version *Ubuntu (64-bit)*
   - Memory: **6144 MB**
   - Processors: **4 vCPU** (System ➔ Processor)
   - Disk: **VDI**, dynamically allocated, **60 GB**

3. **Adjust settings**

   - Display ➔ Graphics Controller: **VBoxSVGA**, enable **3D Acceleration**
   - Storage ➔ Empty optical drive → **Choose a disk file…** select Ubuntu ISO
   - Network Adapter 1: **Bridged** or **NAT** (either works)

4. **Install Ubuntu 24.04** inside the VM

   - "Normal installation", enable third-party software (optional)
   - Disk setup: **Use entire disk** with **LVM** (default)
   - Username: **fundadmin** (sudo)
   - Reboot, login, and finish updates (`Software Updater`)

---

3 Install Prerequisite Packages

Open a **terminal in the guest** and run:

```
# Update system
sudo apt update && sudo apt -y upgrade

# Essential tools
sudo apt install -y git build-essential curl

# Node.js 20 LTS
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
sudo apt install -y nodejs
```

```
# PostgreSQL 16
echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb_release
        -cs)-pgdg main" | \
  sudo tee /etc/apt/sources.list.d/pgdg.list
curl -fsSL https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo
        apt-key add -
sudo apt update
sudo apt install -y postgresql-16

# Verify versions
node -v    # v20.x
npm  -v    # 10.x+
psql -V    # 16.x
```

---

4 Clone Repository & Install Node Dependencies

```
# Conventional location for third-party apps
sudo mkdir -p /opt && sudo chown "$USER":"$USER" /opt
cd /opt

# Clone & checkout v8.8 tag
git clone https://github.com/tpfbill/nonprofit-fund-accounting.git
cd nonprofit-fund-accounting
git checkout v8.8

# Install Node packages (frontend + backend)
npm install
```

---

5 Database Configuration & Initialization

5.1 Create Role & Database

```
sudo -u postgres psql <<'SQL'
CREATE ROLE npfadmin LOGIN PASSWORD 'npfa123';
CREATE DATABASE fund_accounting_db OWNER npfadmin;
\q
SQL
```

5.2 Load Schema & Seed Data

```
cd /opt/nonprofit-fund-accounting

# 1. Create the full schema (tables, constraints, minimal seed rows)
sudo -u postgres psql -d fund_accounting_db -f src/db/db-init.sql

# 2. Add **The Principle Foundation — Parent** entity
#    Creates the single top-level `TPF_PARENT` entity so child
       entities
#    can attach to it.
sudo -u postgres psql -d fund_accounting_db -f
       add_top_level_organization.sql

# 3. Add **The Principle Foundation** child entities
#    This script creates the three-tier structure underneath
       `TPF_PARENT`:
#      • TPF          — top-level parent
#      • TPF-ES       — middle tier (Environmental Services)
#      • IFCSN        — middle tier (Community Service Networks)
node add-tpf-hierarchy.js

# 4. Load rich test transactions that reference those entities /
       funds
sudo -u postgres psql -d fund_accounting_db -f test-data.sql
```

5.3 Application Environment File

Create `/opt/nonprofit-fund-accounting/.env`:

```
PGHOST=localhost
PGPORT=5432
PGDATABASE=fund_accounting_db
PGUSER=npfadmin
PGPASSWORD=npfa123
```

Optionally restrict PostgreSQL to localhost only (`/etc/postgresql/16/main/postgresql.conf`).

---

6 Running the Application

Open **two shells**:

```
# Shell 1 — backend API on port 3000
cd /opt/nonprofit-fund-accounting
node server.js

# Shell 2 — serve static frontend on port 8080
cd /opt/nonprofit-fund-accounting
npx http-server . -p 8080 --no-cache
```

In the guest browser visit **http://localhost:8080/index.html**.
Dashboard cards and charts should populate within a few seconds.

---

7 Testing Checklist

Test	Expected outcome
Dashboard loads	Summary cards + charts visible
Documentation tab	Opens `direct-docs.html`, no styling issues
Fund Reports	Fund dropdown lists all funds
Inter-Entity Transfer wizard	Form loads, API endpoints return 200
DB status badge	**Connected** (green)

Run `npm test` for automated unit tests (if included).

---

8 Troubleshooting

Issue	Resolution
DB Offline badge	`sudo systemctl status postgresql`; verify credentials in `.env`
Port 3000 in use	`sudo lsof -i:3000` then `kill <PID>`
CSS cache issues	Hard-refresh (Ctrl + F5) or clear browser cache
Node native build fails	`sudo apt install -y build-essential python3` and re-run `npm install`

---

9 Performance Optimisation

1. Allocate additional **vCPU/RAM** via VirtualBox settings.

2. Enable **Nested Paging**, **I/O APIC**, **KVM Paravirtualization** (System ➔ Acceleration).
3. Store VDI on SSD/NVMe; enable **discard/trim** if using dynamic disks.
4. PostgreSQL tuning:

   ```
   shared_buffers = 512MB
   work_mem       = 16MB
   ```

5. Enable **pg_preload_libraries = 'pg_stat_statements'** for performance insights.

---

10 Security Notes

- Update guest OS regularly: `sudo apt update && sudo apt upgrade`.
- Change default passwords before production.
- Keep `.env` out of version control; use `chmod 600` on the file.
- Configure UFW:

  ```
  sudo ufw allow 8080/tcp
  sudo ufw allow 3000/tcp
  sudo ufw enable
  ```

- Use **Nginx** reverse proxy with SSL if exposing outside the VM.
- Snapshot the VM after a successful install for easy rollback.

Appendix A – Useful Commands

```
# Stop servers
pkill -f http-server
pkill -f node

# Backup database
sudo -u postgres pg_dump -Fc fund_accounting_db > fundacct_$(date
        +%F).dump

# Restore
sudo -u postgres pg_restore -d fund_accounting_db -c
        fundacct_2024-07-15.dump
```

Enjoy your fully-functional Nonprofit Fund Accounting System v8.8 on Ubuntu 24.04!

For additional documentation refer to the in-app **Documentation** tab or the GitHub wiki.