Mr-
MoneyBags

Mr-MoneyBags v1.x

Installation Guide

Ubuntu 24.04 LTS in Oracle VirtualBox

August 2025

Table of Contents

0. **Document Scope & Audience**
1. **System Overview (v1.x)**
2. **Prerequisites & Host Requirements**
3. **Create the VirtualBox VM**
4. **Install Runtime Dependencies (Guest)**
5. **Application Installation**

5.1 Clone Repository

5.2 Create .env

5.3 Install Node Dependencies

6. **Database Setup – Master Schema**

6.1 Automated (recommended)

6.2 Manual

7. **Authentication & Security Configuration**
8. **Banking Modules Configuration**
9. **Run the Application**
10. **Comprehensive Verification Checklist**
11. **Troubleshooting**
12. **Performance Optimisation**
13. **Security Hardening**
14. **Appendix A – Useful Commands**

Mr-MoneyBags v1.x

Installation Guide – Ubuntu 24.04 LTS Guest in Oracle VirtualBox

Document version 1.x – August 2025

0 Document Scope & Audience

This guide walks you through a **clean, reproducible installation** of the **Mr-MoneyBags v1.x** fund-accounting system in an **Ubuntu 24.04 LTS** virtual machine running under **Oracle VM VirtualBox** on a Windows / macOS host.

It reflects every feature delivered through July 2025, including:

- Complete banking suite (Bank Reconciliation, Bank Deposits, Check Printing)
- Secure authentication (bcrypt-hashed passwords, Express-session, connect-pg-simple)
- Role-based access control & session persistence
- Modularised JavaScript front-end to prevent large-file corruption
- Master database schema (24 tables) plus **one-click sample-data loader** for immediate testing

1 System Overview (v1.x)

Layer	Key Components	Notes
Front-end	Modular HTML 5, CSS 3, **ES Modules**	Split per feature → faster load, avoids truncation
Back-end	Node 18, Express 5, REST API	server-modular.js autoloads routes
Database	PostgreSQL 16	master-schema.sql covers **24 tables**
Banking Modules	Bank Reconciliation (5 tables), Bank Deposits (2), Check Printing (2)	full CRUD & workflows
Security	bcrypt, express-session, connect-pg-simple	sessions stored in DB
Utilities	multer, csv-parser	file uploads (statements, checks)
Docs / Tests	Verification Procedures (295 cases)	see documentation page

2 Prerequisites & Host Requirements

Component	Minimum	Recommended
Host OS	Windows 10/11, macOS 13+	—
CPU	4 cores	6+ cores
RAM	8 GB (allocate ≥ 6 GB guest)	16 GB
Disk	40 GB free	80 GB SSD/NVMe
Software	VirtualBox 7.0+, Ubuntu 24.04 ISO	—

> Enable **VT-x / AMD-V** in BIOS before installing VirtualBox.

New runtime packages pulled inside the guest:
`build-essential curl git nodejs postgresql-16 postgresql-contrib`

---

3 Create the VirtualBox VM

1. **Download**
   - VirtualBox 7+: https://www.virtualbox.org/wiki/Downloads
   - Ubuntu 24.04 ISO: https://ubuntu.com/download/desktop
2. **New VM → Ubuntu24-MrMoneybags-v1x**
   - Linux 64-bit – 4 vCPU – 6144 MB RAM – 60 GB VDI (dynamic)
3. **Tweaks**
   - Graphics **VBoxSVGA**, enable **3-D Acceleration**
   - Network **Bridged** (preferred) or NAT
   - Attach ISO, start installer.

4. **Install Ubuntu (Normal); create user fundadmin with sudo.**

   ```
   sudo apt update && sudo apt -y upgrade
   sudo reboot
   ```

---

4 Install Runtime Dependencies (Guest)

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

# Node 18 LTS
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt install -y nodejs          # node 18.x  npm 10.x+

# 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 postgresql-client-16
```

Verify: `node -v` → 18.x, `npm -v` 10+, `psql -V` 16.x.

5 Application Installation

5.1 Clone Repository

```
sudo mkdir -p /opt && cd /opt
sudo git clone https://github.com/your-org/mr-moneybags-v1.x.git
sudo chown -R $USER:$USER mr-moneybags-v1.x
cd mr-moneybags-v1.x
```

5.2 Create .env

```
cat > .env <<'EOF'
# ── Database ──────────────────────────────────────────────────────────
PGHOST=localhost
PGPORT=5432
PGDATABASE=fund_accounting_db
PGUSER=npfadmin
PGPASSWORD=npfa123

# ── Server ───────────────────────────────────────────────────────────
PORT=3000
SESSION_SECRET=$(openssl rand -hex 32)
EOF
chmod 600 .env
```

5.3 Install Node Dependencies

```
npm ci
```

Key packages now included:

`express@5 pg bcrypt express-session connect-pg-simple multer csv-parser dotenv concurrently http-server`

6 Database Setup – Master Schema

6.1 Automated (recommended)

```
scripts/setup-ubuntu-database.sh
```

The script:

1. Starts PostgreSQL service.
2. Creates role **npfadmin / npfa123**.
3. Creates DB **fund_accounting_db**.
4. Runs **database/master-schema.sql** (24 tables, constraints, indexes).
5. Runs **database/load-sample-data.sql** (complete demo dataset).
6. Seeds admin / user accounts (bcrypt hashed).
7. Verifies connectivity using psql.

6.2 Manual

```
# 1. Role & DB
sudo -u postgres psql -f database/create-role-and-db.sql

# 2. Schema
sudo -u postgres psql -d fund_accounting_db -f database/master-schema.sql

# 3. Sample data
sudo -u postgres psql -d fund_accounting_db -f database/load-sample-data.sql
```

7 Authentication & Security Configuration

1. **Password Hashing** – handled automatically via `bcrypt`.
2. **Sessions** – stored in DB (`session` table) by `express-session` + `connect-pg-simple`.
3. **Role-Based Access** – default roles: `admin`, `user`.
    - Non-admin users cannot access Settings tab.
4. **Adjust session settings** (`src/middleware/auth.js`) if you need different TTL.

8 Banking Modules Configuration

Module	API Mount	Front-end File	Key Extras
Bank Reconciliation	`/api/bank-reconciliation`	`bank-reconciliation.html`	`multer`, `csv-parser`
Bank Deposits	`/api/bank-deposits`	`bank-deposits.html`	—
Check Printing	`/api/checks` & `/api/check-formats`	`check-printing.html`	`amount-to-words` util

No extra config required—routes auto-register from `server-modular.js`.
If you change mount paths, update corresponding JS modules in `src/js/`.

9 Run the Application

```
# Terminal 1 – API (port 3000)
npm start           # or: node server-modular.js

# Terminal 2 – Static front-end (port 8080) with cache busting
npx http-server . -p 8080 --no-cache
```

Browse to **http://localhost:8080/index.html**.
Login with seeded credentials:

- **admin / admin123** (Admin)
- **user / user123** (Standard)

10 Comprehensive Verification Checklist

#	Area	Steps	Expected
1	API Health	`curl :3000/api/health`	`{"status":"ok"}`
2	Auth Flow	Login → navigate modules → session persists	No re-login required
3	Bank Accounts	Settings ➜ Bank Accounts ➜ Add	New account row appears
4	Bank Deposits	Banking ➜ New Deposit	Real-time totals update
5	Check Printing	Queue two checks ➜ **Print Preview**	PDF preview renders
6	Reconciliation	Upload CSV statement ➜ Auto-match	Items status ✔
7	NACHA	Vendor Payments ➜ Batch ➜ Generate NACHA	`.ACH` file downloads
8	Role Control	Login as `user` → Settings tab	Hidden, 403 if forced
9	Sample Data	Dashboard cards	Pre-loaded balances show
10	Reports	Fund Reports ➜ Generate	Table & charts render

For full 295-case matrix see **Verification Procedure v1.x (Part 1 & 2)**.

11 Troubleshooting

Symptom	Remedy
"Failed to fetch reconciliations"	Check fetch URLs include `credentials: 'include'`; verify session cookie not blocked.
404 on `/api/check-formats`	Ensure **check-formats.js** route registered **before** `/api/checks/:id` in server-modular.js
Blank dropdowns (entities / funds)	Rerun **load-sample-data.sql**; restart API
`bcrypt` install fails	`sudo apt install -y build-essential python3` then `npm ci`
Port conflicts	`sudo lsof -i:3000,8080` → `kill <PID>`
Large SQL timeout	Edit `postgresql.conf` → `statement_timeout = 0` for initial load

12 Performance Optimisation

1. VM → enable **Nested Paging**, **KVM** paravirtualisation.
2. PostgreSQL tuning (`/etc/postgresql/16/main/postgresql.conf`):

   ```
   shared_buffers = 512MB
   work_mem       = 32MB
   maintenance_work_mem = 256MB
   ```

3. Allocate 2 GB Node heap if dealing with huge imports:
   `node --max-old-space-size=2048 server-modular.js`
4. Place repository on host SSD; enable VirtualBox **IO APIC**.

13 Security Hardening

- Change default passwords, regenerate **SESSION_SECRET**.
- Enforce HTTPS (reverse-proxy) for production.
- `ufw` firewall:

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

- Set `PGHBA.conf` to **md5** authentication, bind PostgreSQL to `127.0.0.1`.
- Regularly `apt upgrade` and `npm audit`.

14 Appendix A – Useful Commands

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

# Backup DB
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_2025-08-08.dump

# Rotate sessions table (purge expired)
psql -d fund_accounting_db -c "DELETE FROM session WHERE expire < now();"
```

Enjoy your fully-featured Mr-MoneyBags v1.x environment!

For further reference open the in-app **Documentation ➜ System Architecture**.

Mr-MoneyBags v1.x Installation Guide – August 2025