

HOSTELIX PRO

Fresh Start & Admin Setup Guide

How to reset the database, start fresh,
and set up admin accounts using `admin_setup.py`

Version 1.0 | February 2026
Database Reset • Admin Setup • First Run

Table of Contents

1. Stop the Backend Server
2. Delete the Old Database
3. Recreate the Database Tables
4. Set Up Admin Accounts (admin_setup.py)
5. Verify the Setup
6. Start Using the System
7. Quick Reference — One-Shot Reset

1. Stop the Backend Server

Before deleting the database, stop any running backend process:

- **Terminal:** Press **Ctrl + C** in the terminal running the server
- **Systemd service:** `sudo systemctl stop hostelixpro`
- **Docker:** `docker stop hostelixpro`

■■ WARNING: Deleting the database will **permanently erase ALL data** — users, students, fees, reports, routines, announcements, notifications, and audit logs. **Make a backup first** if you need any existing data.

■ IMPORTANT: All user accounts, usernames, and passwords are stored **only** in the database. When you delete the database file, **every account is permanently gone** — including all admin, teacher, routine manager, and student accounts. There is no separate credential store. You **must** run **admin_setup.py** to create new accounts before anyone can log in.

2. Delete the Old Database

SQLite (Default Local Setup)

The database file is at **backend/hostelixpro.db**:

Platform	Command
Windows (CMD)	<code>cd project\backend del hostelixpro.db</code>
Windows (PowerShell)	<code>cd project\backend Remove-Item hostelixpro.db</code>
macOS / Linux	<code>cd project/backend rm hostelixpro.db</code>

To also clear migration history (for a completely clean slate):

```
# Windows  
rmdir /s /q migrations\versions  
mkdir migrations\versions  
  
# macOS / Linux  
rm -rf migrations/versions/*
```

PostgreSQL

```
psql -U postgres  
DROP DATABASE hostelixpro;  
CREATE DATABASE hostelixpro OWNER your_user;  
\q
```

3. Recreate the Database Tables

Navigate to the backend and activate the virtual environment:

Platform	Commands
Windows (PowerShell)	cd project\backend .venv\Scripts\Activate.ps1
Windows (CMD)	cd project\backend .venv\Scripts\activate.bat
macOS / Linux	cd project/backend source .venv/bin/activate

Create all 11 database tables:

```
python create_tables.py
```

Expected output:

```
Creating tables...  
Done!
```

This creates: `users`, `students`, `fees`, `fee_structures`, `transactions`, `reports`, `routines`, `announcements`, `notifications`, `audit_logs`, `backup_meta`.

4. Set Up Admin Accounts

Use `admin_setup.py` to create admin accounts. The script is located at `backend/admin_setup.py`.

All Available Commands

Command	Mode	Description
python admin_setup.py	Interactive	Prompts for email, name, and password with validation
python admin_setup.py --quick	Quick	Creates default admin instantly (admin@hostelixpro.com)
python admin_setup.py --list	List	Shows all existing admin accounts
python admin_setup.py --help	Help	Displays usage information

Option A: Interactive Mode (Recommended)

```
python admin_setup.py
```

The script will prompt you step by step:



```
Enter admin email: admin@yourschool.com
Enter display name: Dr. Ahmed Khan
```

Password requirements:

- At least 8 characters
- At least 1 uppercase letter (A-Z)
- At least 1 lowercase letter (a-z)
- At least 1 digit (0-9)

```
Enter password: ****
```

```
Confirm password: ****
```

```
Create this admin account? (y/n): y
```

✓ Admin account created successfully!

ID: 1

Email: admin@yourschool.com

Name: Dr. Ahmed Khan

Option B: Quick Setup (For Testing)

```
python admin_setup.py --quick
```

This creates a default admin account:

Field	Value
Email	admin@hostelixpro.com
Password	Admin@1234
Display Name	System Administrator

■■ WARNING: Change the default password immediately after first login in production!

Option C: View Existing Admins

```
python admin_setup.py --list
```

Shows all admin accounts with their ID, email, name, and lock status.

5. Verify the Setup

Start the Backend

```
python app.py
```

Test Health Endpoint

```
curl http://localhost:3000/api/v1/health

# Expected:
# {"status": "healthy", "service": "Hostelix Pro API"}
```

Test Admin Login

```
curl -X POST http://localhost:3000/api/v1/auth/login \
-H "Content-Type: application/json" \
-d '{"email": "admin@hostelixpro.com", "password": "Admin@1234"}'
```

Expected: A response with **tx_id** and **otp_sent: true**.

■ **TIP:** Since email is likely not configured on a fresh setup, OTPs print to the backend terminal. Look for:
[DEV MODE] OTP for admin@hostelixpro.com: 482917

6. Start Using the System

After admin setup, configure the system in this order:

Step	Action	Where
1	Log in with admin account	Flutter App → Login page
2	Create teacher accounts	Admin → Users → Add User
3	Create routine manager accounts	Admin → Users → Add User
4	Set up fee structures	Admin → Fees → Fee Structures
5	Create announcements	Admin → Announcements
6	Students register via the app	Login page → Register
7	Admin approves students	Admin → Users → Pending
8	Assign students to teachers	Admin → Users → Assign

User Roles Quick Reference

Role	Access Level
Admin	Full system access — users, fees, reports, backups, audit logs
Teacher	View assigned students, approve reports, manage routines
Routine Manager	Manage walk/exit requests, return confirmations
Student	Submit reports, request exits, view fees, pay fees

7. Quick Reference — One-Shot Reset

Copy-paste this to do a **complete fresh start** in one go:

Windows (PowerShell)

```
cd project\backend
.venv\Scripts\Activate.ps1

# Delete old database
Remove-Item hostelixpro.db -ErrorAction SilentlyContinue

# Recreate tables
python create_tables.py

# Set up admin (interactive)
python admin_setup.py

# Start server
python app.py
```

macOS / Linux

```
cd project/backend
source .venv/bin/activate

# Delete old database
rm -f hostelixpro.db

# Recreate tables
python create_tables.py

# Set up admin (interactive)
python admin_setup.py

# Start server
python app.py
```

End of Document — Hostelix Pro Fresh Start & Admin Setup Guide v1.0