Hostel Management System (Django)

A simple Django-based hostel management app intended for **learning/demo** purposes. This fork prioritizes an **easy setup** on Windows & macOS/Linux and documents a few fixes for common legacy-dependency issues.

> **Heads-up**: This project uses **Django 2.2** and a few older packages. It runs reliably on **Python 3.9 / 3.10**. Newer Pythons (e.g., 3.13) remove some stdlib modules (`cgi`) that Django 2.2 expects.

Features (high level)

- Basic student & faculty management
- Authentication & Django admin
- Email hooks (for notifications) — configurable
- SQLite database by default

> Exact screens and flows vary by commit; this is a learning repo with basic CRUD and admin-driven workflows.

Tech Stack

- **Backend**: Django 2.2.x
- **DB**: SQLite (default)
- **Runtime**: Python 3.9 / 3.10 recommended

Repository Layout (key parts)

```
hostel-management/
├─ Home/                # main project app & settings
│  ├─ settings.py       # settings (email config described below)
│  └─ ...
├─ faculty/             # faculty app (expects faculty/demo.py for email
constants)
├─ student/             # student app
├─ manage.py
├─ requirements.txt     # pinned (cleaned) dependencies
└─ ...
```

Prerequisites

- **Python**: 3.9 or 3.10 (strongly recommended)
- **Git**
- Optional: a code editor (VS Code, PyCharm, etc.)

Why not Python 3.13?

Django 2.2 imports `cgi`, which was removed in Python 3.13. Use Python 3.9/3.10 to avoid runtime import errors.

Quick Start (Windows PowerShell)

```
# 1) Clone your fork
git clone https://github.com/<YOUR_USERNAME>/hostel-management.git
cd hostel-management

# 2) Create & activate a virtualenv with Python 3.9
py -3.9 -m venv venv
venv\Scripts\activate

# 3) Install dependencies
pip install --upgrade pip setuptools wheel
pip install -r requirements.txt

# 4) Set up email config (pick ONE option below; Option A is easiest)
#    Option A – Create two small files with email constants:
#    - Home/demo.py (optional, only if you see settings importing it)
#    - faculty/demo.py
#    Each should contain:
#        EMAIL_HOST_USER = "test@example.com"
#        EMAIL_HOST_PASSWORD = "password123"
#
#    Option B – Use environment variables instead (more secure):
#        setx EMAIL_HOST_USER "test@example.com"
#        setx EMAIL_HOST_PASSWORD "password123"

# 5) Database setup
python manage.py makemigrations
python manage.py migrate

# 6) Create an admin user (for /admin)
python manage.py createsuperuser

# 7) Run the server
```

```
python manage.py runserver
# Browse http://127.0.0.1:8000/
```

Quick Start (macOS / Linux)

```
# 1) Clone your fork
git clone https://github.com/<YOUR_USERNAME>/hostel-management.git
cd hostel-management

# 2) Create & activate a virtualenv with Python 3.9
python3.9 -m venv venv
source venv/bin/activate

# 3) Install dependencies
pip install --upgrade pip setuptools wheel
pip install -r requirements.txt

# 4) Email config (choose ONE)
# Option A – Create Home/demo.py and faculty/demo.py with:
#   EMAIL_HOST_USER = "test@example.com"
#   EMAIL_HOST_PASSWORD = "password123"
# Option B – Environment variables (recommended):
#   export EMAIL_HOST_USER="test@example.com"
#   export EMAIL_HOST_PASSWORD="password123"

# 5) Database setup
python manage.py makemigrations
python manage.py migrate

# 6) Admin user
python manage.py createsuperuser

# 7) Run
ython manage.py runserver
# Open http://127.0.0.1:8000/
```

Creating Accounts

Admin

Create with `createsuperuser` and log in at `http://127.0.0.1:8000/admin/`.

Students / Faculty

The simplest path is via **Django Admin**: 1. Log in to `/admin` with the superuser. 2. Find the **Student** and **Faculty** models and click **Add**. 3. Fill in the details and save.

(If the project exposes a public signup page, you can also register there — varies by branch.)

Configuration: Email

This project references two values for email auth: - `EMAIL_HOST_USER` - `EMAIL_HOST_PASSWORD`

You can provide them **either** via small `demo.py` files **or** via environment variables.

Option A — `demo.py` files (quickest)

Create these files if you see imports like `from .demo import ...`:

Home/demo.py (only if `Home/settings.py` imports it):

```
EMAIL_HOST_USER = "test@example.com"
EMAIL_HOST_PASSWORD = "password123"
```

faculty/demo.py (required if `faculty/views.py` imports it):

```
EMAIL_HOST_USER = "test@example.com"
EMAIL_HOST_PASSWORD = "password123"
```

> For safety, do **not** commit real credentials. Commit `demo.py.example` files instead and add `demo.py` to `.gitignore`.

Option B — Environment Variables (recommended)

`Home/settings.py` is set up to read from environment variables when present. Define: - `EMAIL_HOST_USER` - `EMAIL_HOST_PASSWORD`

On Windows (PowerShell):

```
setx EMAIL_HOST_USER "test@example.com"
setx EMAIL_HOST_PASSWORD "password123"
```

On macOS/Linux:

```
export EMAIL_HOST_USER="test@example.com"
export EMAIL_HOST_PASSWORD="password123"
```

> If you use env vars, remove or ignore any `from .demo import ...` lines — or keep the `try/except` pattern in `settings.py` so both methods work.

Corrected `requirements.txt`

Some forks include a bogus line `pkg-resources==0.0.0` that breaks Windows installs.

Use this cleaned set of pinned deps:

```
astroid==2.4.2
Django==2.2.11
django-phone-field==1.8.1
django-widget-tweaks==1.4.8
djangorestframework==3.11.0
isort==4.3.21
lazy-object-proxy==1.4.3
mccabe==0.6.1
Pillow==8.4.0
pylint==2.5.3
pytz==2020.1
six==1.15.0
sqlparse==0.3.1
toml==0.10.1
wrapt==1.12.1
```

> If you already have `requirements.txt`, replace its contents with the above.

Common Commands

```
# activate venv (Windows)
venv\Scripts\activate

# activate venv (macOS/Linux)
source venv/bin/activate
```

```
# runserver
python manage.py runserver

# make & apply migrations
python manage.py makemigrations
python manage.py migrate

# create admin
python manage.py createsuperuser
```

Troubleshooting

1) `ModuleNotFoundError: No module named 'django'` - Ensure your venv is active and run `pip install -r requirements.txt`.

2) `ModuleNotFoundError: No module named 'faculty.demo'` - Create `faculty/demo.py` with `EMAIL_HOST_USER` and `EMAIL_HOST_PASSWORD` (see above).

3) `No module named 'cgi'` **or similar import errors** - You are likely using Python 3.13+. Install Python 3.9 or 3.10 and recreate the venv.

4) Pillow install/build errors - Use `Pillow==8.4.0` and ensure `pip`, `setuptools`, and `wheel` are upgraded.

5) `pkg-resources==0.0.0` **error** - Remove that line from `requirements.txt` (it's invalid on Windows).

6) Emails not sending - Verify `EMAIL_HOST_USER` / `EMAIL_HOST_PASSWORD` and SMTP host/port/TLS settings in `settings.py`.

Security & Production

- This is a **learning** project. Django 2.2 is **end-of-life**; do not deploy as-is to production.
- Do not commit secrets. Use env vars or a `demo.py.example` with `.gitignore`d `demo.py`.

Contributing

PRs are welcome for docs, compatibility fixes (e.g., Django/LTS upgrades), and tests. Please open an issue to discuss larger changes.

License

For learning/demo only. If you plan to publish commercially, consider upgrading the stack to a supported Django version and adding a proper license.