introduction

This app is used for managing rental properties. It makes it convenient for Tenants to be able to access your property on line and make bookings. This saves time to avoid future tenants from walking around estates to physically find places to rent also there is a ranking mechanism which ranks the places through security, water availability and comments.

Houses are uploaded by caretakers or managers. They must include at least two photos,cell contact and location.

Future features such as us of Google Maps aren't yet implemented.

technicality

This web-app is written in pure Python and Python popular web-framework Django.

Requirements:

Python 2.7 or higher
Django 1.8 or higher
Mysql 5.5.4 or higher
Apache web server 2.2 or higher
Most preferably a Linux machine ie Ubuntu 14.04

log in to 127.0.0.1:8000/admin and
create to groups of users:

i)caretakers who have permissions to add remove delete plots and houses
ii)tenant group who have permissions to book and comment

in searching our houses and plots will be using a third-party extension Solr Apache with python binding haystack.
requirements:
Java Runtime Environment version 1.7 or higher you can download it from *http://www.oracle.com/technetwork/java/javase/downloads/index.html* .
Solr version 4.10.4 or get it from *http://archive.apache.org/dist/lucene/solr/*
django haystack through pip install django-haystack==2.4.0

web-app basics

Users register through two registration forms tenant registration and caretaker/manager registrations this automatically join them to their relevant groups and creates a profile for the user.

Once a caretaker/manager registers he/she adds a plot/estate to our portal as all houses must belong to an estate/plot. Then he/she adds house through a house description form. Then waits for an email notifying him/her about bookings where he/she can approve or ignore.

The normal user account(tenant) one views all listed houses and can book houses and add comments. Once a booking is approved by the caretaker all other bookings are invalidated.

Each user has a profile which contains useful information such as contacts,occupation and date of birth.

Useful commands

```
python manage.py runserver.
      Runs the django server on top of your web-server.
python manage.py makemigrations && migrate.
      performs database interactions
python manage.py shell.
      Provides a shell API for interaction and testing
python manage.py build_solr_schema.
      Provides xml code to be inputed in schema.xml in the core configurations
python manage.py rebuild_index
       indexes  your models to be serchable.
python manage.py update_index
       update after adding new items
```

Note all this commands are run from the root project folder of the application.