Real Estate Agency Portal Tutorial

The Real Estate Agency portal is a web portal in which the user can choose to select one among multiple cities in which the user can then have a look at the various properties available in that particular city.

The Real Estate Agents can list out their properties on the web portal and the users can visit this portal to view these properties and contact the real estate agents if they want to buy/rent a particular property.

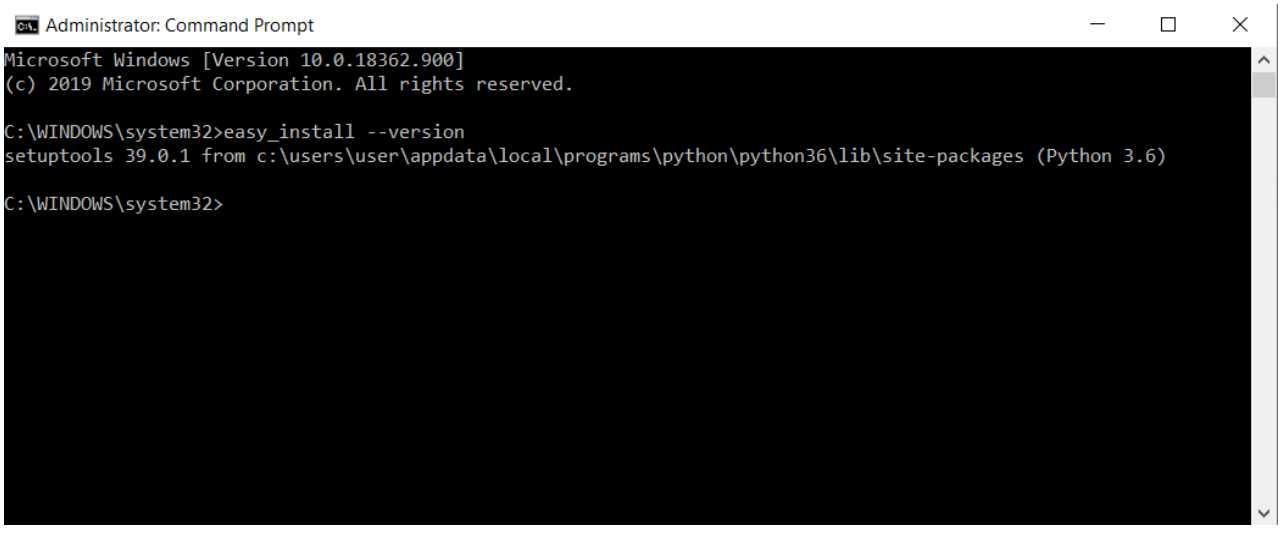
The following steps will help you to reverse engineer the project by consuming the code from the Github repository. The following are the steps to be followed to install Python, Django and make the required virtual environments and to run the code from the Github Repository.

1. Download and install Python

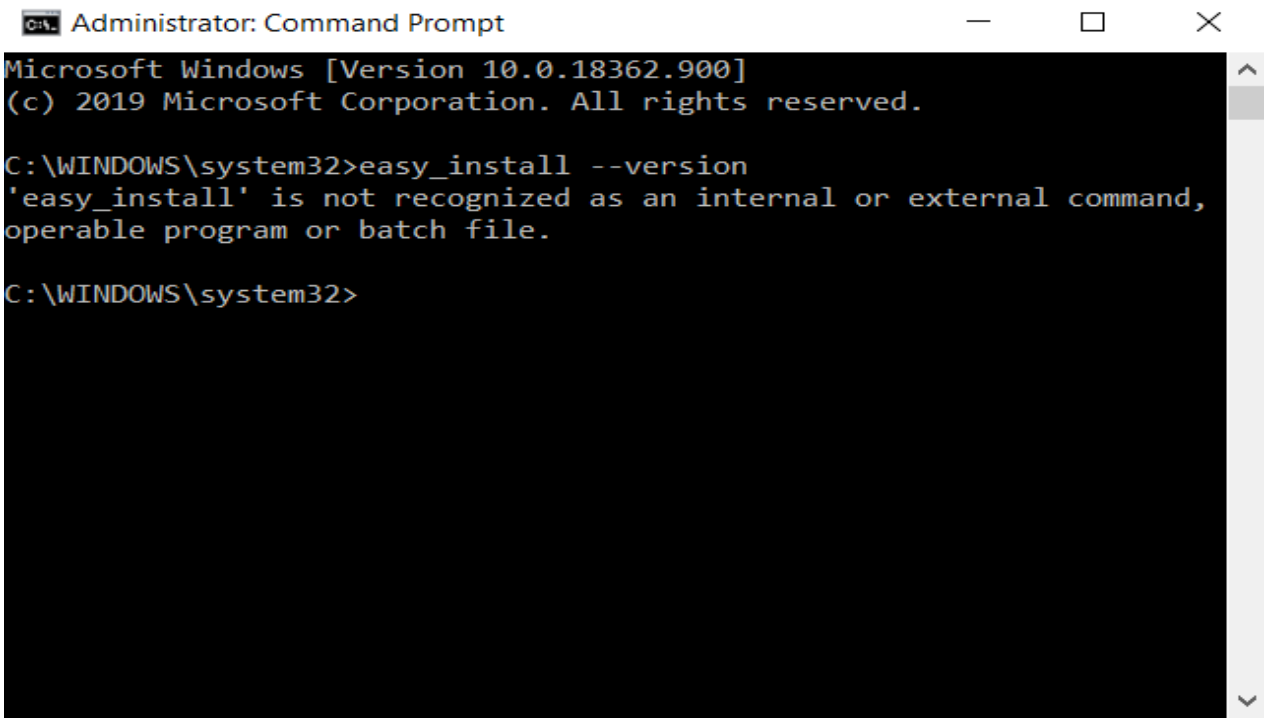
link- https://www.python.org/downloads/ and install it.

2. Install Virtualenv and Django

- Open command prompt (cmd) and run it as Administrator
- Type Command: easy_install --version



- You should get the output as seen in the above image



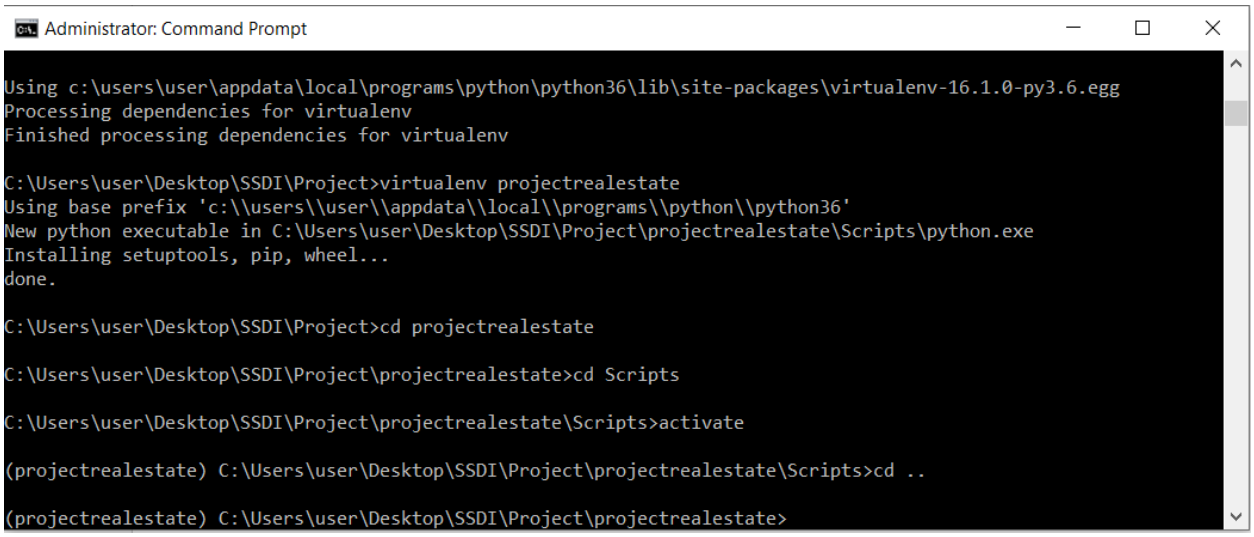
- If and when you run into the above error, you will need to setup the environment variable for easy_install which can be done using following steps
- Navigate to the folder where python is installed
- Go to the Scripts Folder
- Copy the file path for easy_install
- Right Click on My Computer and click properties
- Go to advanced system settings and click on environment variables
- Edit the path string and copy the filepath for easy_install
- Click ok and type command: easy_install --version

- Navigate to the folder where you want to create the virtual environment for the project

- Type command: easy_install virtualenv

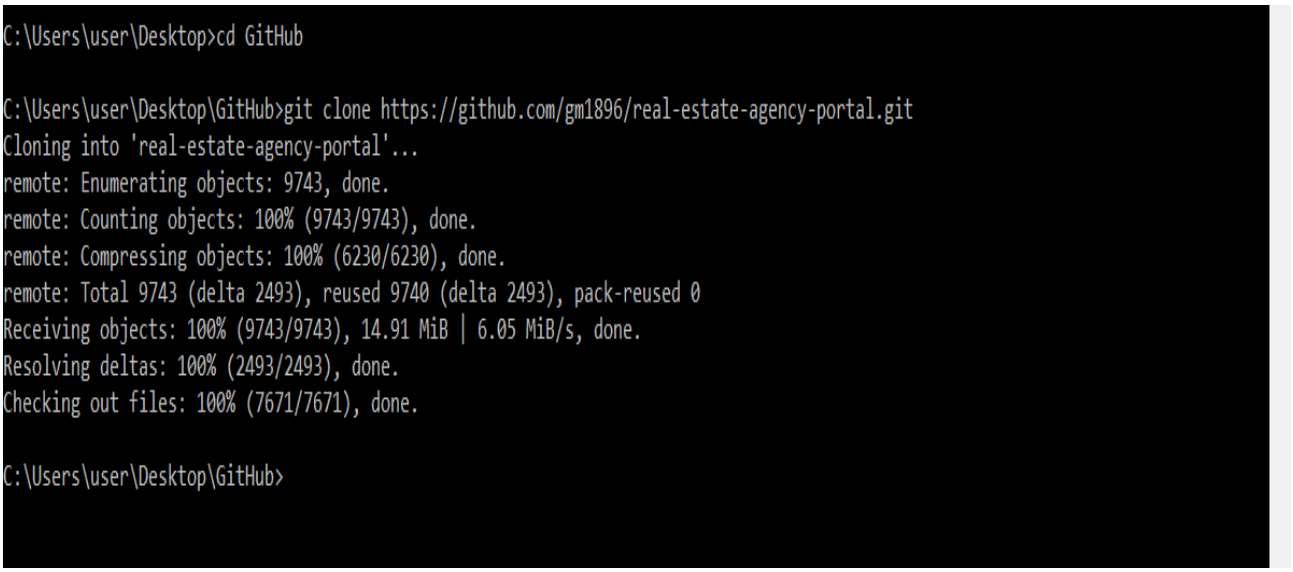
- Type command: virtualenv projectrealestate

- The above commands help us to create a virtual environment which needs to be activated as follows:



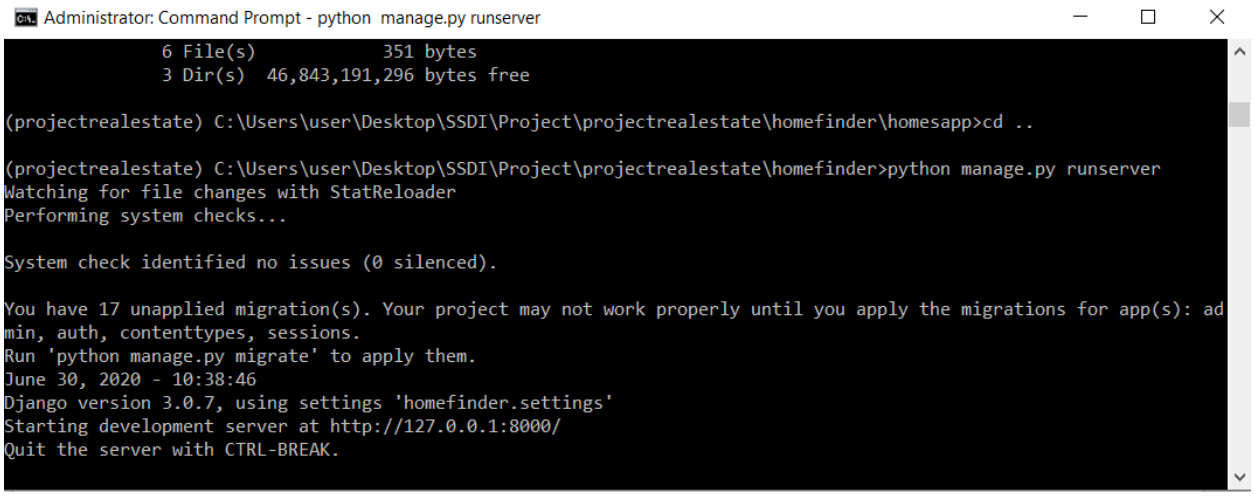
- Once we have activated the virtual environment as mentioned above then we can install Django in our virtual environment using the following command: pip install Django

- Once the virtual environment has been created clone the real-estate-agency-portal repository from this link as follows:

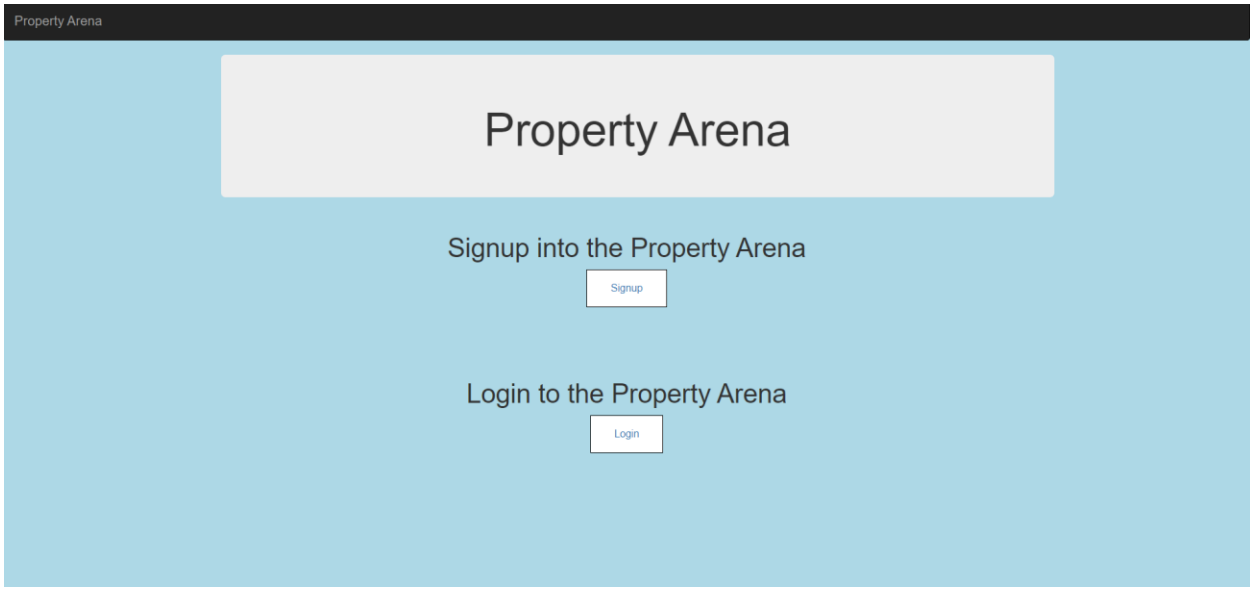


- The cloned repository contains the homefinder folder inside the real-estate-agency-portal folder which is to be copied and placed inside the virtual environment to form a structure similar to the cloned structure.
- This step has to be done because a new virtual environment will have to be created on a new machine.
- Now we copy the homefinder folder from the cloned Github folder and place it in the projectrealestate folder with the other generated files.
- As the required folder structure is now created on your local machine you just need to start the development server to access the project

3. **Start server using the following command:**



4. **Copy the server path into the browser to run the project and see the homepage of the application:**



5. From the homepage you can either signup into the portal or sign in to the portal if you have already a user of the application.

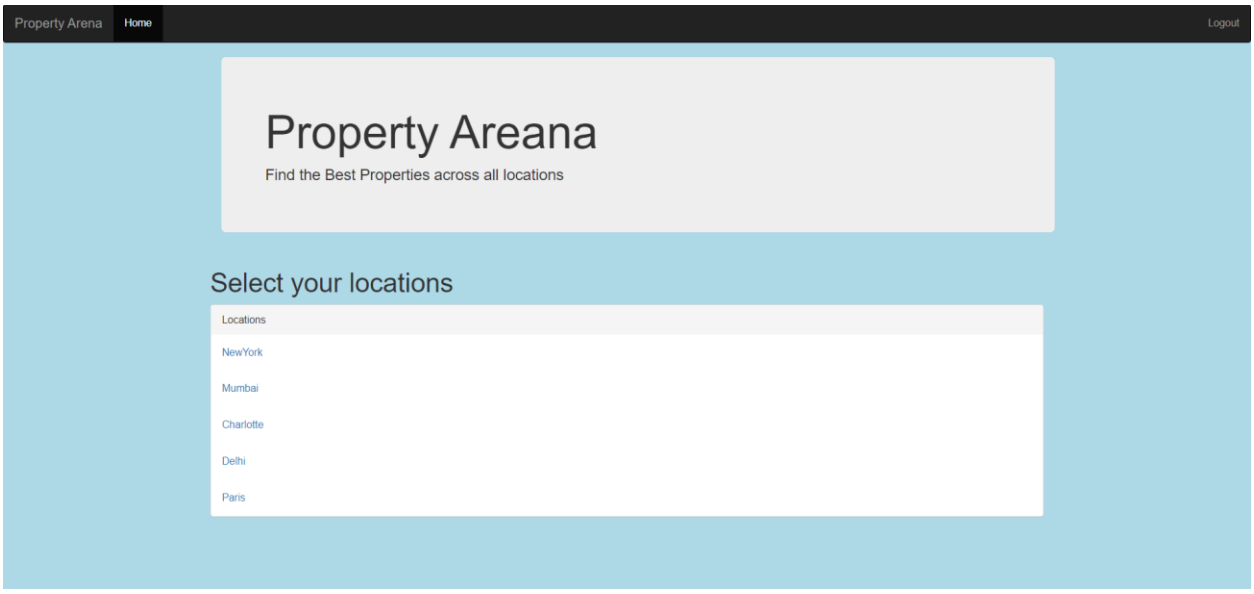
6. This project follows the MVC Architecture where we have the different views in the views.py folder which are displayed as per the url matching which is done by regular expressions from the urls.py file in homesapp.

7. Depending upon the url, the valid view is generated using the data from the data models and the respective HTML files for the particular views.

8. Following we can see three different views which are the Index View, Location View and Property View respectively which use underlying base templates for the styling which can be seen.

9. The application has a simple flow which can be seen once the above steps are followed and the application is run on a local system and then changes can be made appropriately to reverse engineer and take this as the base template for the project and improve the project by adding some additional functionality to the existing project

10. **Upon signing in you go to the main page of the application which lists out various locations as follows:**

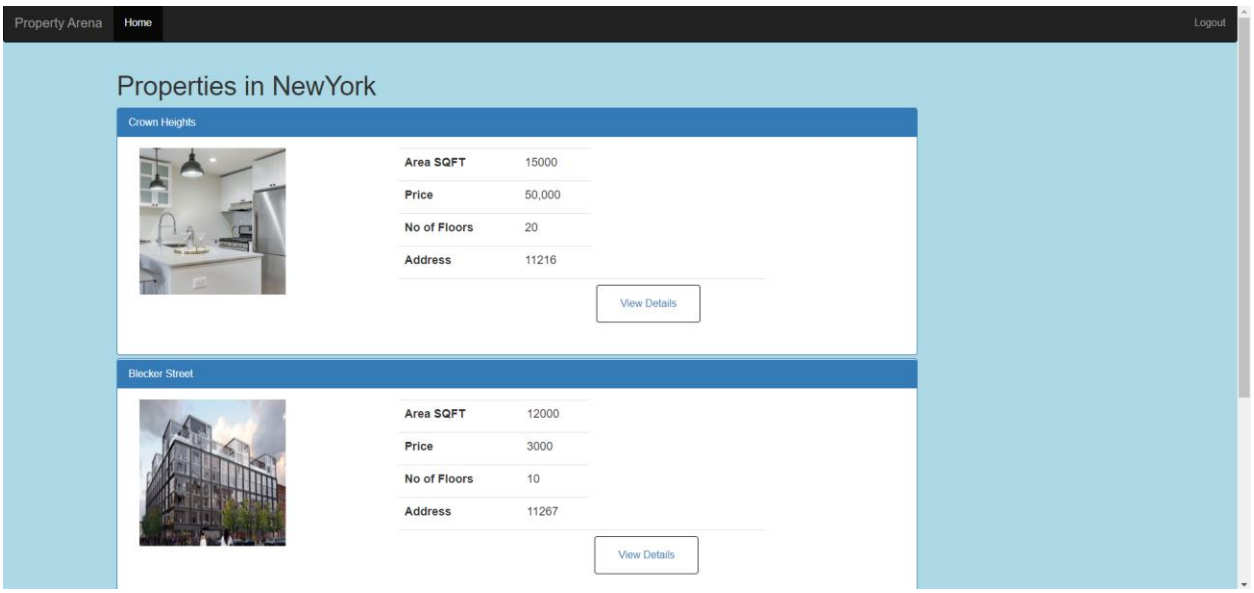


11. **You can study how the index.html is displaying different locations and you can make changes in the same as desired.**

homefinder > homesapp > templates > homesapp > index.html > ...

```
1  {% extends 'homesapp/base.html' %}
2  <html>
3      <head></head>
4          {% block body %}
5          <body>
6              <div class="container">
7                  <div class="container">
8                      <div class="jumbotron">
9                          <h1>Property Areana</h1>
10                         <p>Find the Best Properties across all locations</p>
11                     </div>
12             </div>
13
14             <centre><h1>Select your locations</h1></centre>
15             <div class="panel-group">
16                 <div class="panel panel-default">
17                     <div class="panel-heading">Locations</div>
18                     {% for location in object_list %}
19                     <div class="panel-body"><a href="/homesapp/{{location.id}}">{{location.location_name}}</a></div>
20                     {% endfor %}
21                 </div>
22             </div>
23         </body>
24         {% endblock %}
25 </html>>
26
27
```

12. **Every location has multiple properties linked to that particular location as follows:**



13. **You can study how the locationview.html is displaying different properties in given location and you can make changes in the same as desired.**

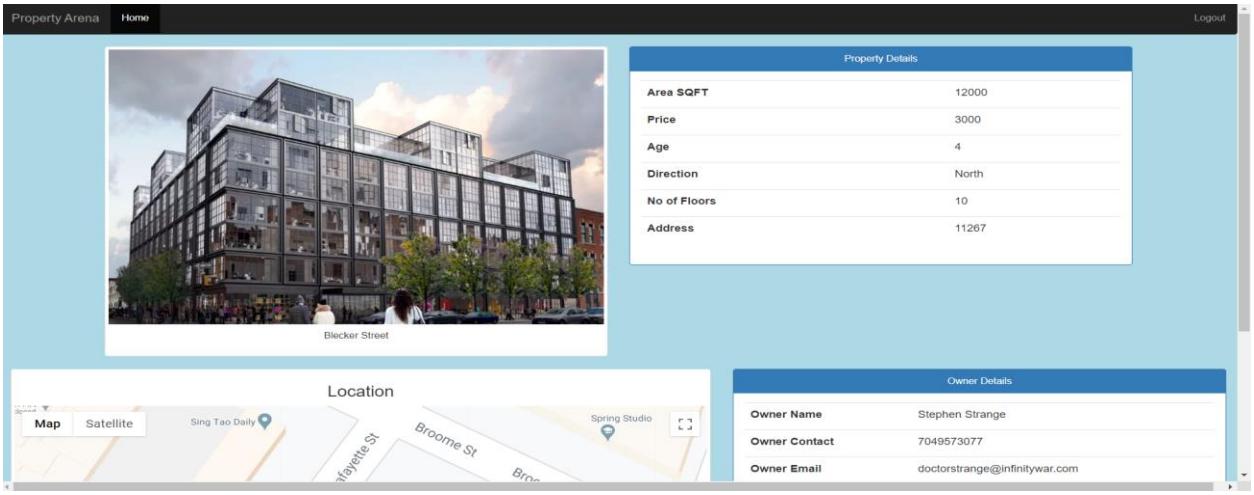
homefinder > homesapp > templates > homesapp > locationview.html > html > body > div.col-md-8 > div.panel-group > div.panel.panel-primary

```
{% extends 'homesapp/base.html' %}

{% block body  %}

<html>
    <head></head>
    <body>
        <div class="col-md-1">
        </div>
        <div class="col-md-8">
            <h1>Properties in {{location.location_name}}</h1>
            <div class ="panel-group">
                {% for place in location.property_set.all %}
                <div class ="panel panel-primary">
                    <div class="panel-heading">{{place.property_name}}</div>
                    <div class="panel-body">
                        <div class="col-md-4">
                            <img src="{{place.image_url}}" height="200" width="200">
                        </div>
                        <div class="col-md-6">
                            <table class="table">
                                <tbody>
                                    <tr>
                                        <td><label><b>Area SQFT</b></label></td>
                                        <td>{{place.area}}</td>
                                    </tr>
                                    <tr>
                                        <td><label><b>Price</b></label></td>
                                        <td>{{place.price}}</td>
                                    </tr>
```

14. **And Every Property has its own page to check out the details of the project as follows:**



15. **You can study how the propertyview.html is displaying different properties in given location and you can make changes in the same as desired.**

homefinder > homesapp > templates > homesapp > propertyview.html > html > head > style > html

```
{% extends 'homesapp/base.html' %}

{% block body  %}

<html>

    <head>
        <style>

         #map {
                height: 100%;
                width: 100%;
                }
        html,body{
            text-align: center;
            height:75%;
            width: 100%;
            margin: 0;
            padding:0;
        }
        </style>
    </head>

    <body>
        <div class="row">
            <div class="col-md-1">
            </div>

            <div class="col-md-5">
                <div class="thumbnail">
                    <img src="{{property.image_url}}" height="420" widht="420">
                    <div class ="caption">
                        <p>{{property.property_name}}</p>
                    </div>
                </div>
            </div>
```

Following the above tutorial one can recreate the project generate a virtual environment to hold the Django Project and replicate the project in order to understand how it has been developed and even change and refactor the code to make the application much more attractive with better UI and different kinds of additional functionalities can be added as required.