NewLine

Real Estate Management Web Application

Anshul Sharma

Project Document

Table of Contents

Real Estate Management Web Application

Project purpose

This website is an Online real estate management website through which a user can access information about the various properties and listings provided by the company which is handling the real estate business. The project is oriented for both the admin (in this case, the head of a company or whoever manages the website) and a user who would search for information about the listing. The admin can manage all the adding, updating, and deleting of the assets. The project effectively advertises the listings which are available and provides options to contact the realtor of the listing. The admin can update the information regarding the listing or the realtors who are present for the service. The system is very useful for companies that own apartments, hotels, villas, residential properties, and commercial properties for business purposes. Companies or individual agents can also advertise them properly.

Project scope

The world wide web has spread across millions of households, so naturally, Internet has become by far the best platform for real estate marketing and advertising today. There are a lot of real estate companies that advertise their property online. This application replicates the online presence of real estate marketing, and when deployed in a public server, has vast scope for growth.

The application is an online real estate management through which agents can maintain their property document keeping along with managing property registration and any user can access its information. The user can inform their agents regarding the property. The system is very useful for companies or builders that can post and edit their properties and personal information.

Project Development Approach

The Development Process can be stated in the following phases-

1) The objectives and Requirements of the project must be evaluated and reviewed.

2) The basic structure of the Project must be designed.

3) The development environment must be considered and stated.

4) Programming, configuration, building and coding must be done after the completion of the above steps.

5) The Application must be tested using the Development Server and automation testing.

Project Objectives and Requirements

The objectives of this project are the following-

- The Admin maintains the property and the realtors kept on the behalf. Admin (or superuser) of the website can add, delete, and update the properties and listings available alongside the information of the users of the website, realtors. The website is equipped with a login system.
- The user can browse through the listings advertised and can register to submit information regarding his/her interest in the listing.

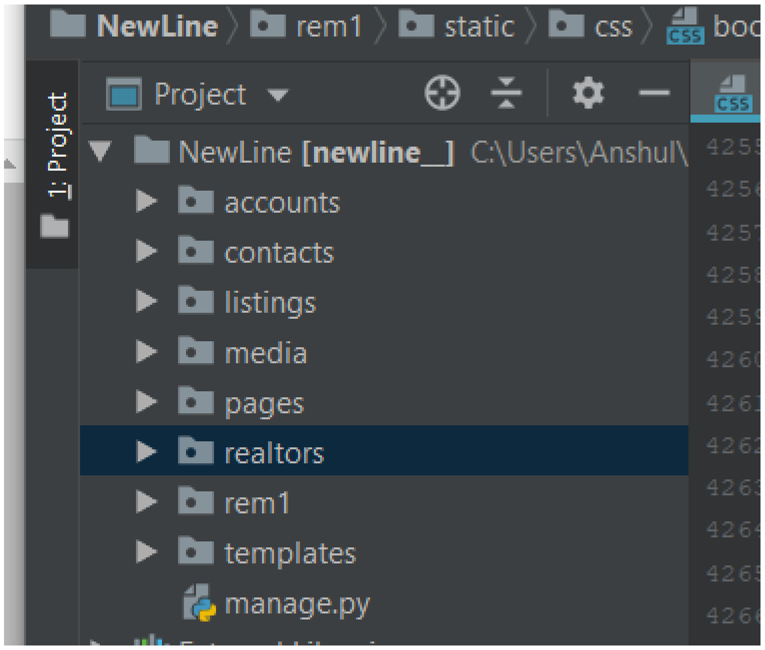
Based on the objectives, the following requirements can be demonstrated, and later used to implement the test strategy-

- The application should have a navigation bar, to allow the users to navigate through the following sections- Home, About, Featured Listings, Registration, and Login Page.
- Users should be able to register an account (if not registered already), by providing information in a form, such as FirstName, LastName, Username, Email and Password.
- Users should be able to access the listings and information related to them.
- The user must be able to send his/her information to the admin based on interest in a specific listing.

- Registration Page must not let invalid values pass through it.
- Validations must be present on the Login Page as well.
- The Validations must be a combination of server-side and client-side responses. It enables the login and registration modules to be more dynamic and responsive.
- Admin must have a separate panel dedicated to the admin's view of the application. A normal user must not be able to access it.

Project Design

The Following snapshot of the project directory can be considered-



```
NewLine 〉 rem1 〉 static 〉 css 〉 boc
1: Project
Project ▾
▼ NewLine [newline_] C:\Users\Anshul\
    ▶ accounts
    ▶ contacts
    ▶ listings
    ▶ media
    ▶ pages
    ▶ realtors
    ▶ rem1
    ▶ templates
      manage.py
```

The directories within the Main directory (accounts, contacts, listings, pages, realtors) have the main view files, for rendering and directing various pages in the web application. The mentioned directories have the render/view files that correspond to the first requirement of the project (The User must be able to navigate and explore the Login, Registration, Listings and Home page).

Back-end Approach – 'Accounts' has the view (the name for the request and response function file) for the Login and Registration. Moreover, it maps the model for Users in the database. 'Contacts' has the view for the model that relates to the user that made contact with the admin over a listing. **The ‘**Listings’ view file renders the page where the user can browse through listings. The directory ‘rem1’, named roughly after ‘Real estate management 1’, has the urls.py and settings.py (mentioned later). In simple terms, urls.py binds the path of the web pages in the web application together. For instance, if the web application’s main page runs on 127.0.0.1, the 127.0.0.1/admin redirects the web application to the admin panel page. In the settings file, paths and configurations to web page templates and databases are written and included in the code.

Project Development Environment

PyCharm is an IDE used for the project specifically for the Python language. It is developed by the Czech company **Jetbrains**. It provides code analysis, and a graphical and supports web development.

For Front-End Development-

- **HTTP:** Hypertext transfer protocol as a transaction or oriented client/server protocol between the web browser and web Server.
- **HTML/CSS:** Data presented needs to be organized such that it’s easily understandable to the users. Used extensively for Design and User Interface.
- **JavaScript/jQuery:** To create scripts that increase the response and dynamic aspect of the web application. Primarily used for Client-side scripting for this application, as most of the server-side responses are framed using Django.
- **Bootstrap:** The web application has a responsive design i.e. it can adapt its design to various screen sizes.

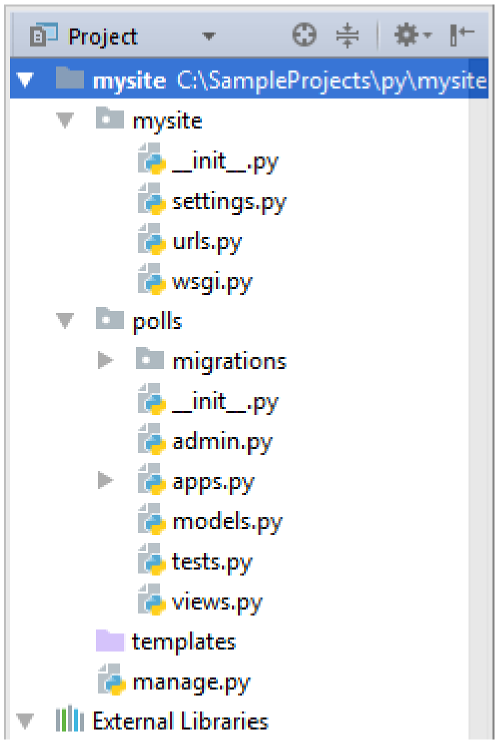
For Back-End Development-

- **Python**: The programming language that is utilized to configure data migrations and models during application development.
- **Django Framework:** Framework that supports the platform for the application development, which holds the models used in this project, and provides us with Application Development Server.

- **Postgresql:** A Database management system for assisting the maintainability of the website. As this website is dynamic, Postgresql is used to maintain the records of listings, realtors, and users. **PgAdmin,** a PostgreSQL tool for windows was used to make interaction with the database easier.
- **Psycopg2:** A python driver used as a driver for PostgreSQL for the application. The models will be mapped with the database using the driver.
- **Pillow**: A python library that acts as a bridge for the media files (png, bmp and jpeg) between the development server and the web application.

Django Project Structure

When a web application is developed using the Django framework, its structure can be described with the help of a simple diagram-



The following diagram shows the basic structure of a sample Django project.

Here mysite is a Django Project. Here, mysite is the name of the website which is under development. It has the following files-

- The **__init__.py** file tells Python that this Directory forms a Python package.

- **settings.py** contains the configurations for your Django project. Every Django project must have a settings file. It contains settings for the installed packages, databases, and paths to the static content of our website. It also indicates the integration of additional libraries used for development.
- **urls.py** contains project-level URL configurations. By default, this contains a single URL pattern for the admin. The paths to different web pages within the web application are assigned here.
- **wsgi.py** enables WSGI-compatible web servers to serve your project.

App- Here, an ***app*** refers to a sub-module of the project. It's self-sufficient and not intertwined with the other apps in the project such that, in theory, you could pick it up and plop it down into another project without any modification. A simple project might only have one app.

Models- A **model** is a class that represents relational data or collection in our DB, and where every attribute of the class is a field of the table or collection.
Views- A **view** function, or “view” for short, is simply a Python function that takes a web request and returns a web response. Enables the Server-side aspect of the Web Application. This response can be the HTML contents of a Web page, a redirect, a 404 error, an XML document, an image, etc. Example: You use view to create web pages, note that you need to associate a view to a URL to see it as a web page.

In the above diagram, polls are the name of the app. It has the following files-

- **models.py** is where the models for your app are located. Models map the attributes of the data to the database.
- The **migrations** folder is where Django stores migrations, maps the models, or changes to your database (if we have one connected to the project)
- **admin.py** is where you register your app’s models with the Django admin application.
- **apps.py** is a configuration file common to all Django apps.

- **tests.py** contains test procedures that will be run when testing your app.
- **views.py** is where the views for your app are located.

Other constituents of the project-

Manage.py- A command-line utility for executing Django commands from within your project.

The Development server is executed through the command-

(>> python manage.py runserver).

Manage.py is automatically created in each Django project.

Commands related to manage.py do tasks like migration(connecting model to the database), creating admin of the project, etc. The file itself is never edited.

Templates- This is the folder in which the templates of the Django project are located. Templates contain the HTML Content which is required to be generated by the website which is under development. A template contains the static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted.

Some of the important Django commands are-

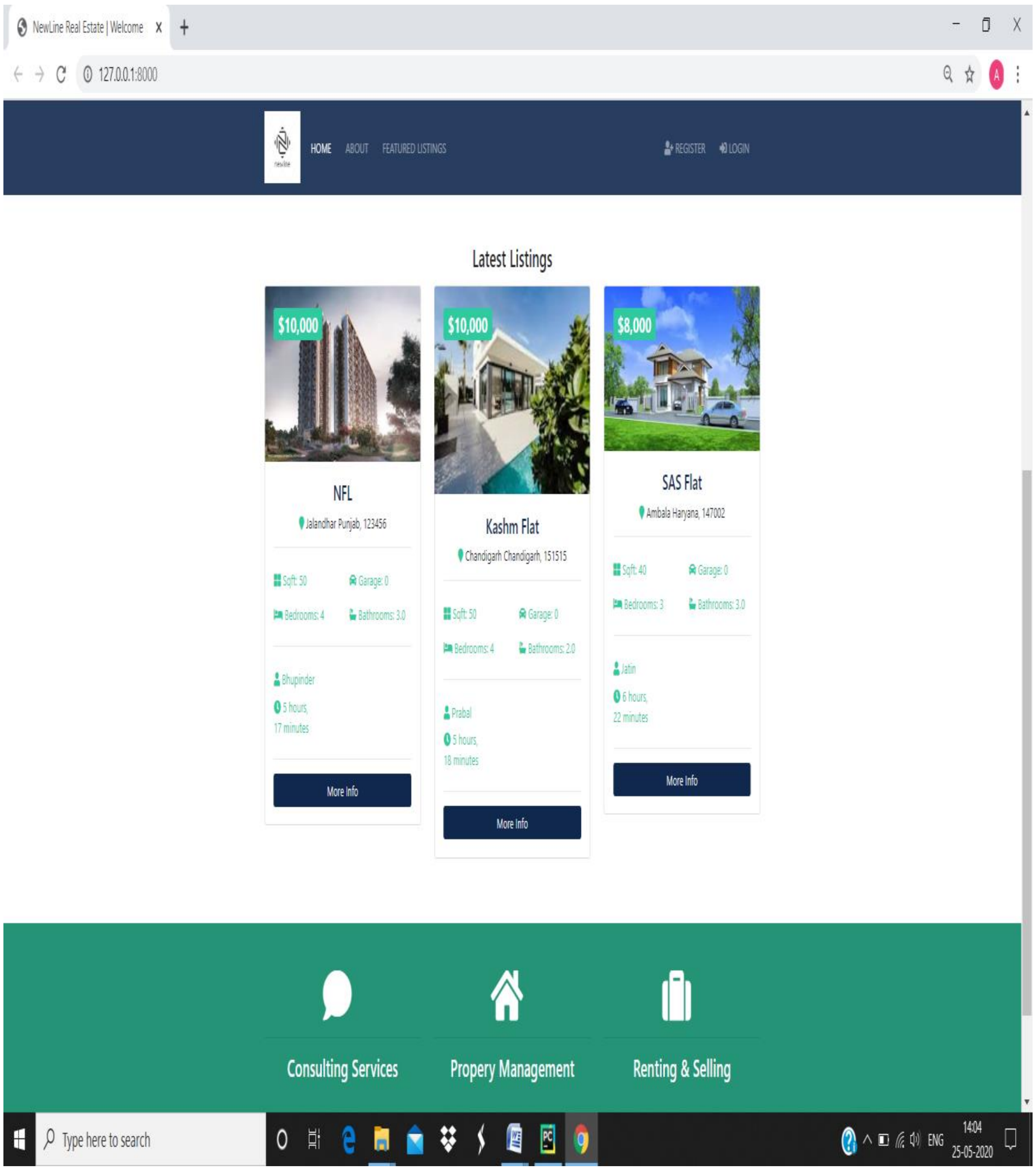
- **To start project** ->> Django-admin startproject project_name
- **To Create App-** >>python manage.py startappn app_name
- **To Execute the server-** >> python manage.py runserver
- **To create a superuser-** >>python manage.py createsuperuser
- **For migration-** >>python manage.py makemigrations
  >>python manage.py migrate

PROJECT DETAILS

The website is dynamic and the admin has control over which Listings are to be advertised. Admin can edit/create/delete Listings with the help of the Admin panel that is provided on the website.

When we run (>>python manage.py runserver) on the python terminal, the project will execute on the link 127.0.0.1:8000.

This is the **Home** page of the Website-



Latest Listings

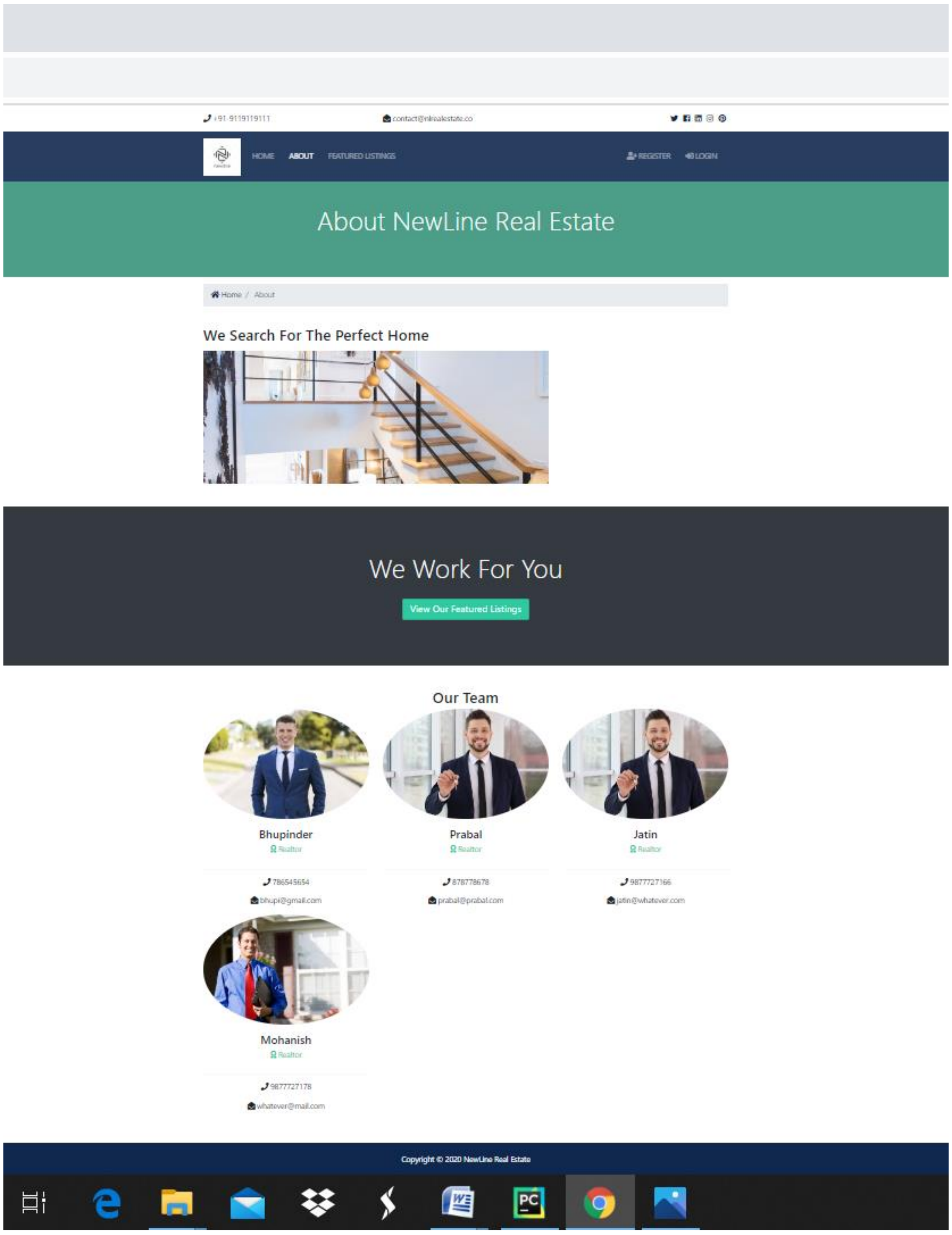
$10,000
NFL
Jalandhar Punjab, 123456
Sqft: 50 Garage: 0
Bedrooms: 4 Bathrooms: 3.0
Bhupinder
5 hours, 17 minutes

$10,000
Kashm Flat
Chandigarh Chandigarh, 151515
Sqft: 50 Garage: 0
Bedrooms: 4 Bathrooms: 2.0
Prabal
5 hours, 18 minutes

$8,000
SAS Flat
Ambala Haryana, 147002
Sqft: 40 Garage: 0
Bedrooms: 3 Bathrooms: 3.0
Jatin
6 hours, 22 minutes

It consists the list of the latest listings that's been added by the company. It has the following links on the navigation bar- HOME, ABOUT, FEATURED LISTINGS, REGISTER and LOGIN.

The **ABOUT** page-



+91-9119119111
contact@nlrealestate.co

HOME ABOUT FEATURED LISTINGS REGISTER LOGIN

About NewLine Real Estate

Home / About

We Search For The Perfect Home

We Work For You

View Our Featured Listings

Our Team

Bhupinder
Realtor
786545654
bhupi@gmail.com

Prabal
Realtor
878778678
prabal@prabal.com

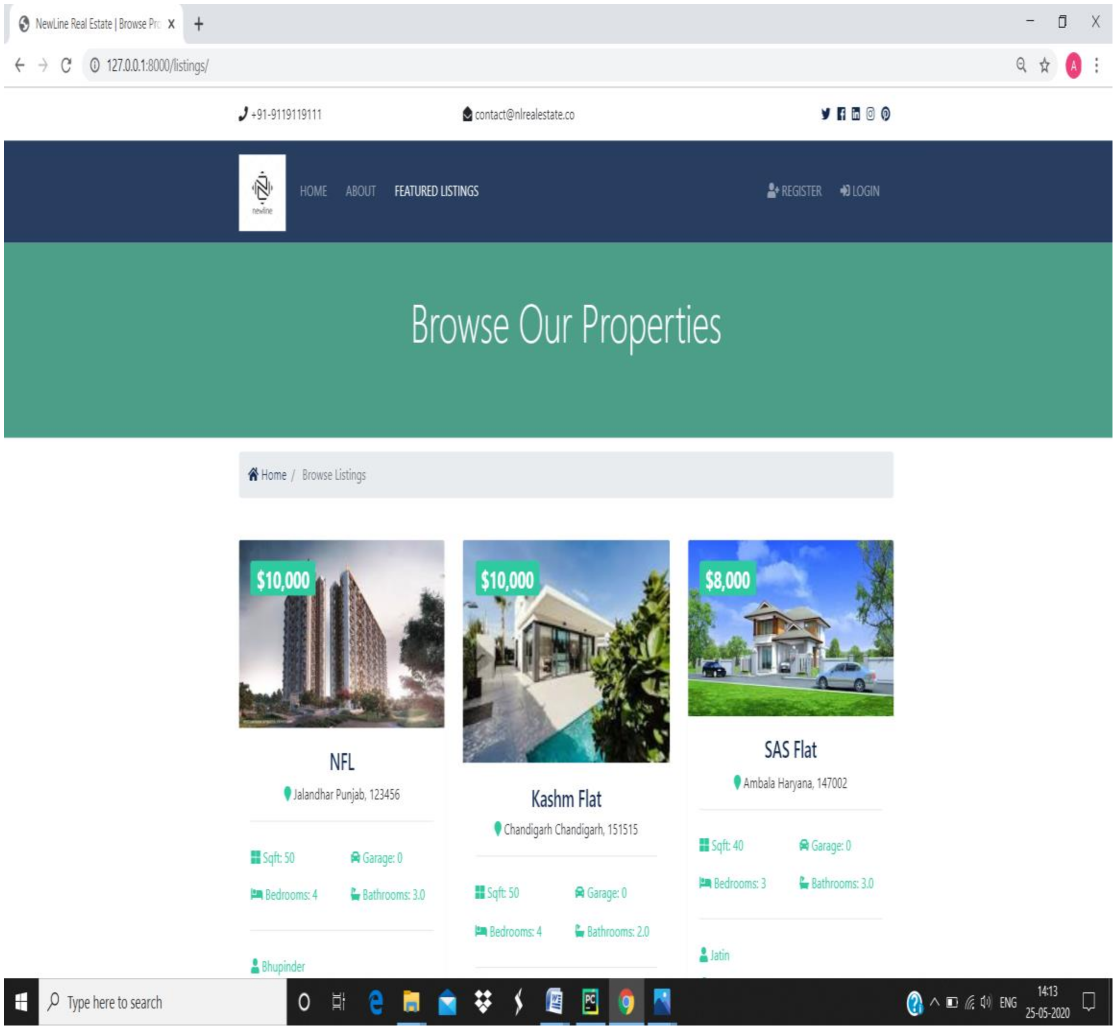
Jatin
Realtor
9877727166
jatin@whatever.com

Mohanish
Realtor
9877727178
whatever@mail.com

Copyright © 2020 NewLine Real Estate

It contains the information about the team of realtors associated with the company. This is dynamic and it can be edited by the admin in the admin panel (without diving into the code of the project).

Featured Listing webpage-



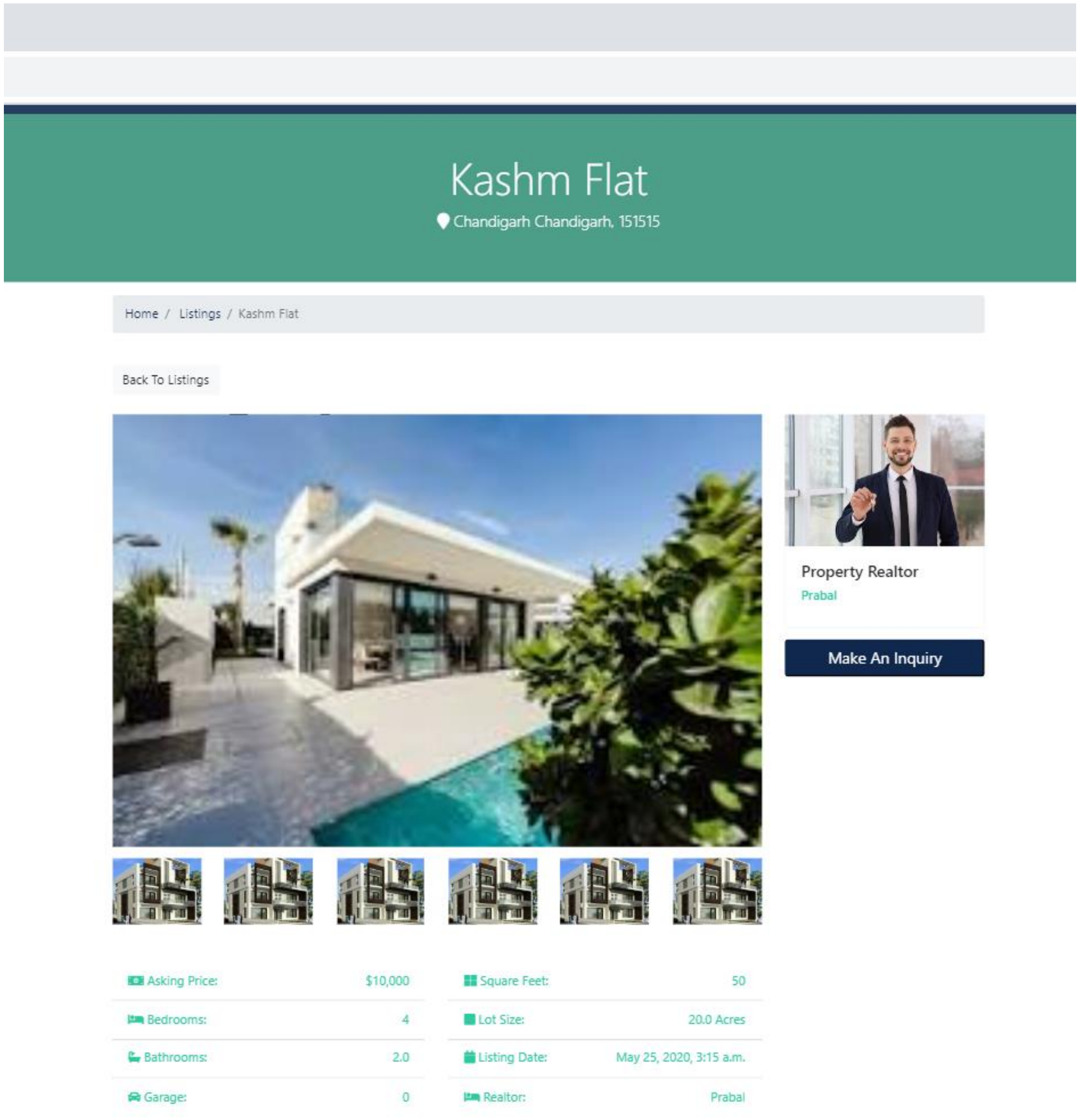
This webpage lists the listings which are made available by the company for the user to browse through their desired property.

This, along with listings and contacts is dynamic too as the admin or the owner can list out the properties which are available at the moment. So the admin can maintain, update, change and delete any listing according to him. Also, the prices can be updated as labeled on the listings.

The admin can add as many as listings required. The information of the name of the listing, price, address and no. of bedrooms is displayed alongside the image of the listing.

Upon clicking on any one of them, more details are shown about the listings.

Upon clicking on **more info**, the page displayed would be like-



Kashm Flat

Chandigarh Chandigarh, 151515

Home / Listings / Kashm Flat

Back To Listings

Property Realtor

Prabal

Make An Inquiry

Asking Price:	$10,000	Square Feet:	50
Bedrooms:	4	Lot Size:	20.0 Acres
Bathrooms:	2.0	Listing Date:	May 25, 2020, 3:15 a.m.
Garage:	0	Realtor:	Prabal

The things which are displayed here-

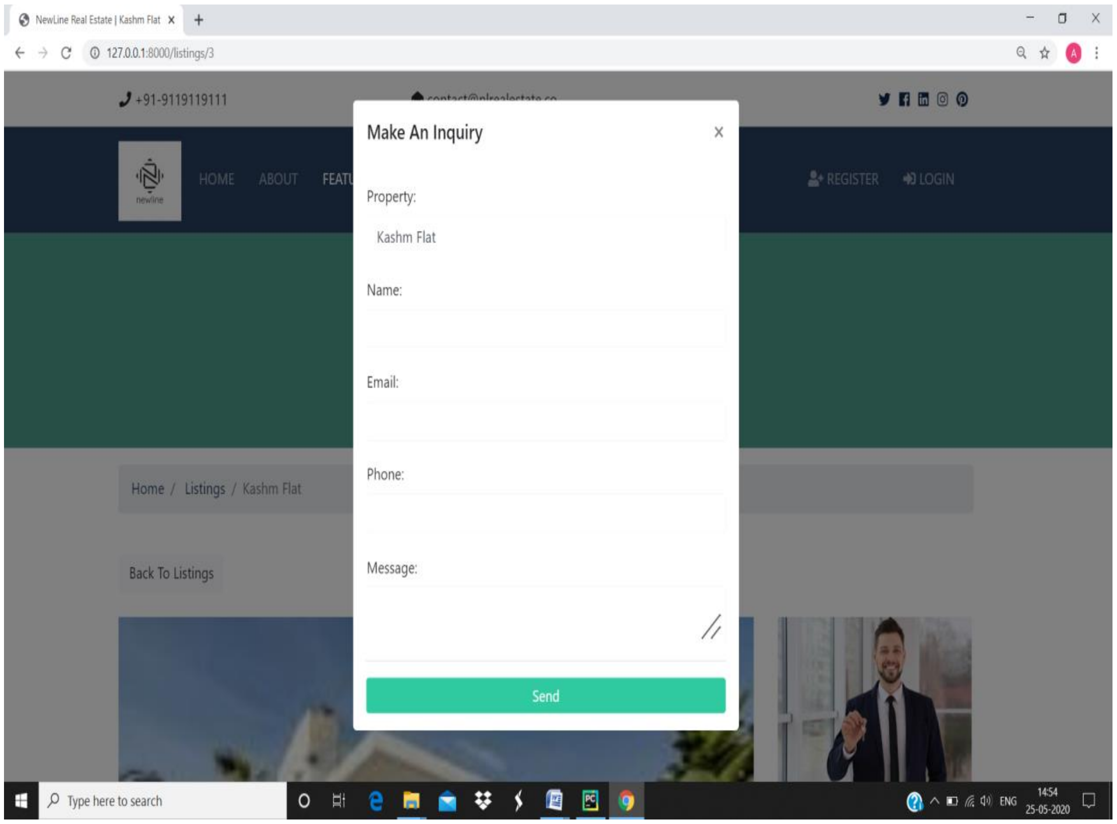
1. The property realtor of the particular listing.

2. Various images of the property.

3. Information regarding number of bedroom, bathrooms, garages, area, listing date and the address.

4. A button named “Make An Inquiry”. Clicking on it opens a form which asks for the user who is interested in checking out the property.

The form is like-


Make An Inquiry

Property:

Kashm Flat

Name:

Email:

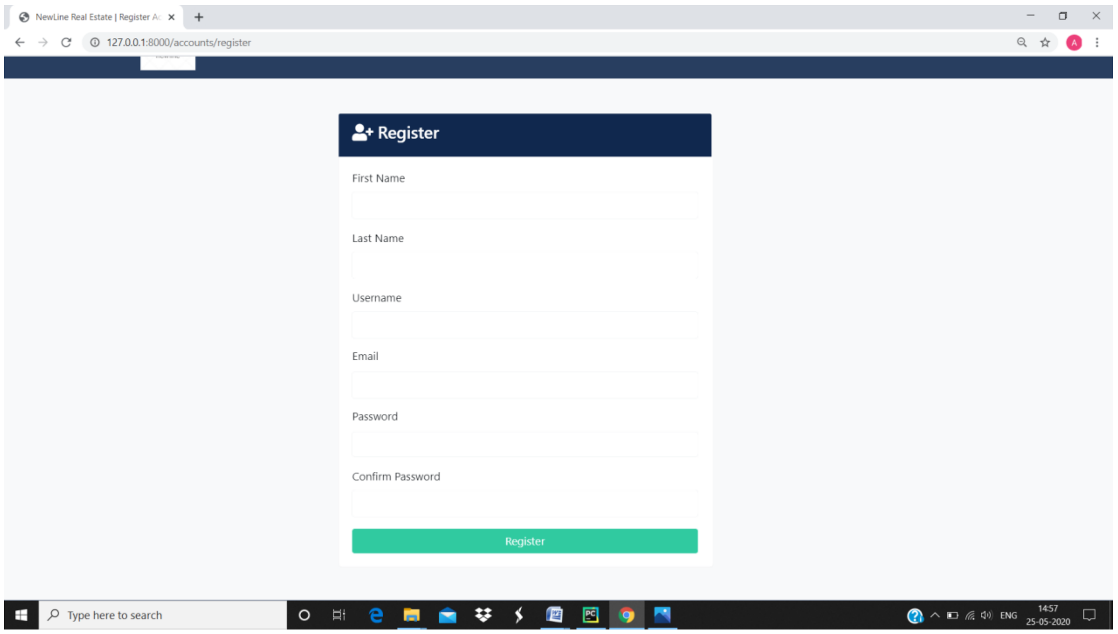
Phone:

Message:

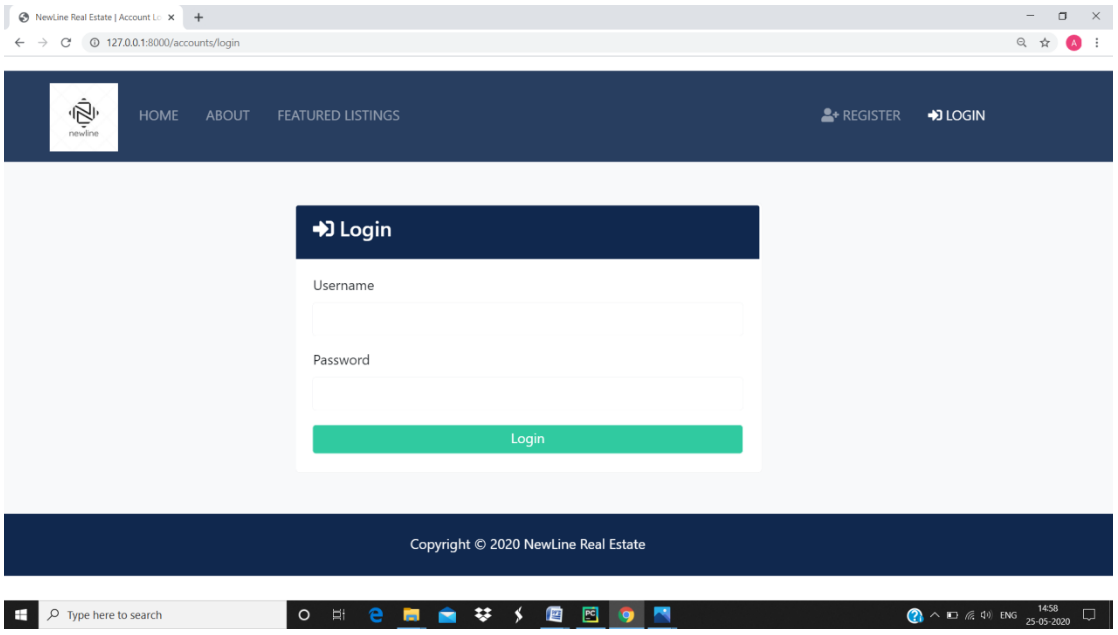
On clicking send, the information of the user who is interested in the property is submitted to the admin panel where the company admin can view the information of the interested user.

There are two more options in the navigation bar- Registration and Login.

Registration opens a form for a new user who is asked to fill his userID/username in order to directly login next time.



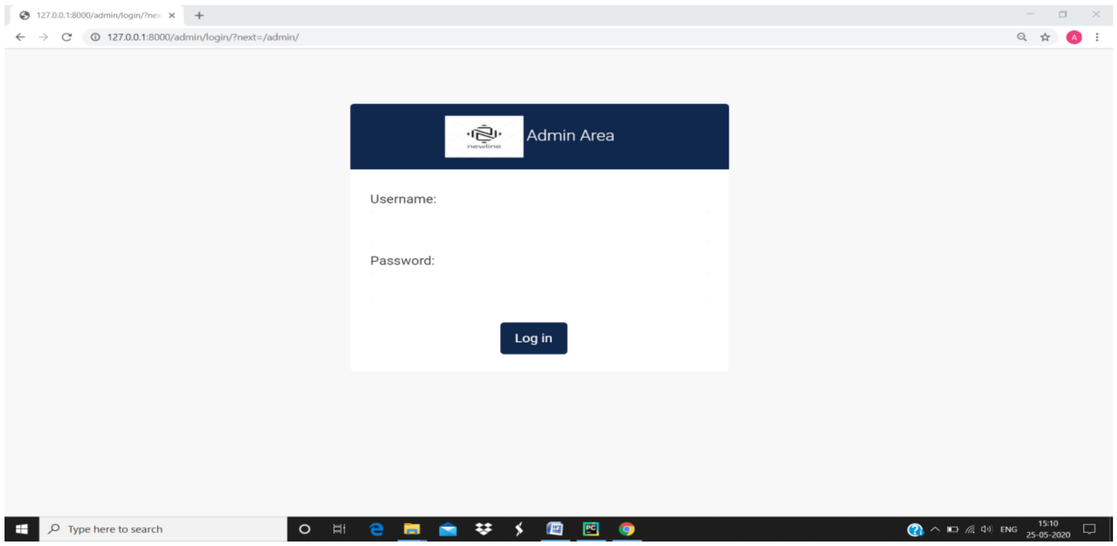
Upon filling the details, the user can directly login next time. Login page looks like this-


NewLine Real Estate | Account L
127.0.0.1:8000/accounts/login
newline
HOME
ABOUT
FEATURED LISTINGS
REGISTER
LOGIN
Login
Username
Password
Login
Copyright © 2020 NewLine Real Estate
Type here to search

The user has to enter the username and password. User can logout from the site anytime as login link is replaced by the logout link.

So this is the way the site is implemented for a user to browse through all the available properties of the company. Now, as the website is dynamic and the admin has to maintain it and do operations like adding more listings, updating them, deleting them based on whether they are available or not anymore. So with the help of the Django Framework, the website provides an admin panel over which the admin can regulate everything dynamic in the website. The admin panel can be opened by adding / admin to the url of the website. For eg- 127.0.0.1:8000/admin

This opens another username-password field but this field is only for superusers, not normal users.



127.0.0.1:8000/admin/login/?next=/admin/

Admin Area

Username:

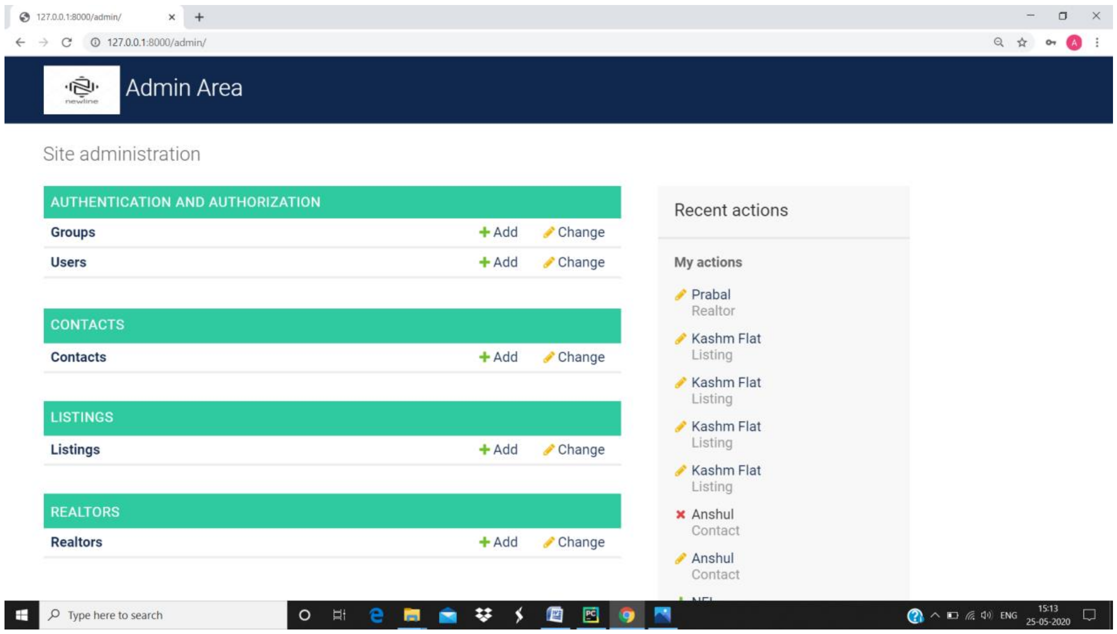
Password:

Log in

Type here to search

ENG 15:10 25-05-2020

Upon entering the superuser password (it is configured before executing command through python terminal, explained later) admin's area is opened and it looks like-



127.0.0.1:8000/admin/

Admin Area

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups + Add Change

Users + Add Change

CONTACTS

Contacts + Add Change

LISTINGS

Listings + Add Change

REALTORS

Realtors + Add Change

Recent actions

My actions

Prabal
Realtor

Kashm Flat
Listing

Kashm Flat
Listing

Kashm Flat
Listing

Kashm Flat
Listing

Anshul
Contact

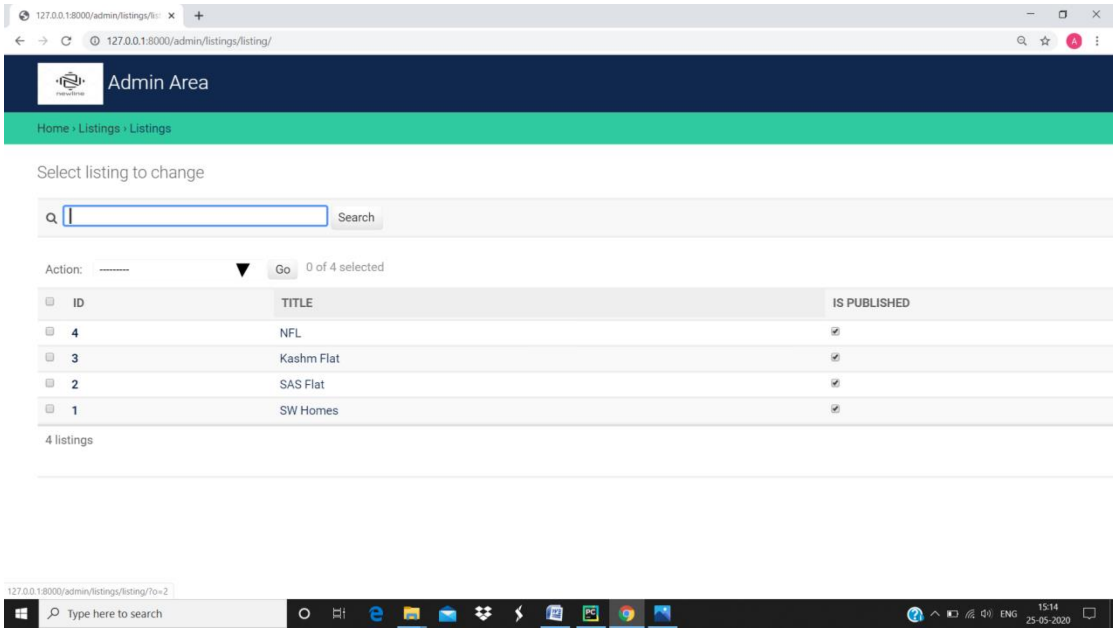
Anshul
Contact

Type here to search

ENG 15:13 25-05-2020

Record of all the listings, users of the website, Contacts made for inquiry and the information of realtors associated.

The list of all listers-



127.0.0.1:8000/admin/listings/listing/

Admin Area

Home › Listings › Listings

Select listing to change

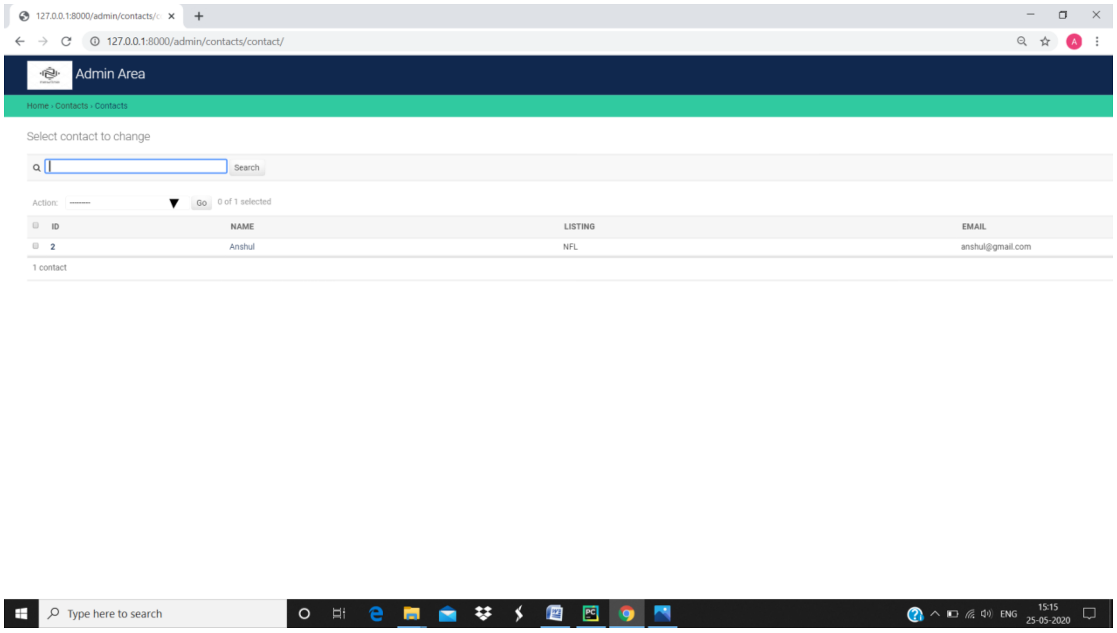
Action: --------- Go 0 of 4 selected

ID	TITLE	IS PUBLISHED
4	NFL	✓
3	Kashm Flat	✓
2	SAS Flat	✓
1	SW Homes	✓

4 listings

127.0.0.1:8000/admin/listings/listing/?o=2

The list of all Contacts-



127.0.0.1:8000/admin/contacts/contact/

Admin Area

Home › Contacts › Contacts

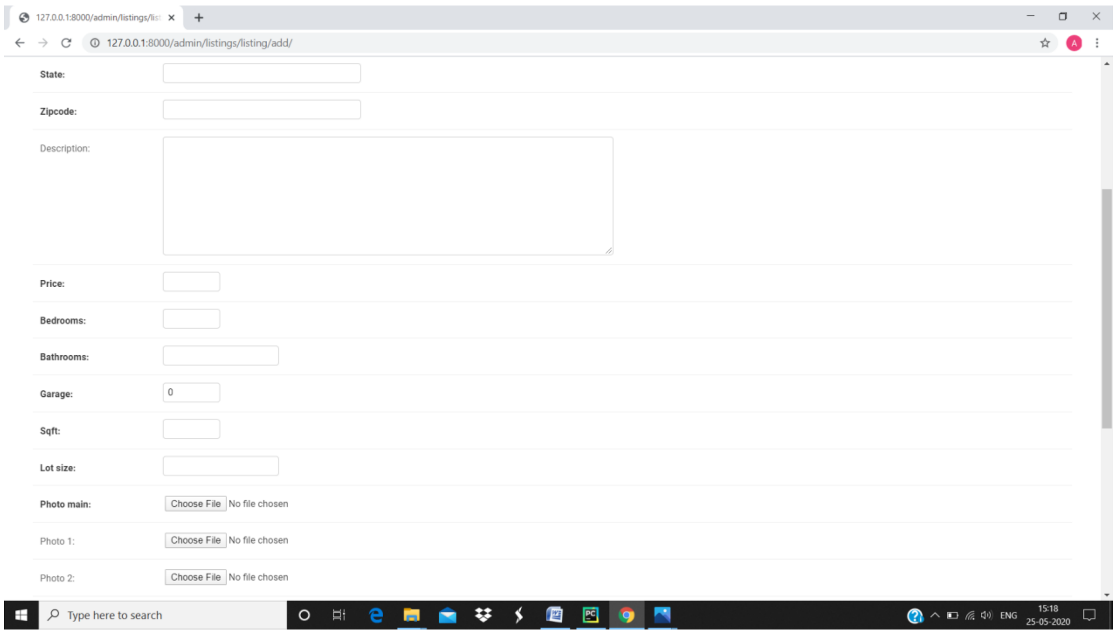
Select contact to change

Action: --------- Go 0 of 1 selected

ID	NAME	LISTING	EMAIL
2	Anshul	NFL	anshul@gmail.com

1 contact

There is an option for deleting, adding and updating over which the admin can easily control the content of the website. For eg- if the admin wishes to add another listing, upon clicking on add button over the listing, a form appears-



State:	
Zipcode:	
Description:	
Price:	
Bedrooms:	
Bathrooms:	
Garage:	0
Sqft:	
Lot size:	
Photo main:	Choose File No file chosen
Photo 1:	Choose File No file chosen
Photo 2:	Choose File No file chosen

We can even add photos alongside with description, address, title, no. of bedrooms for the new property etc.

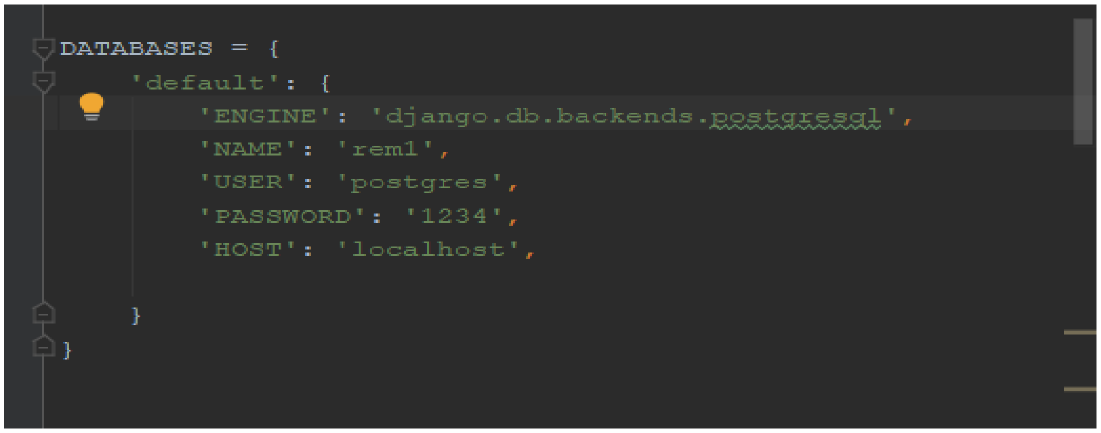
So as we can see, it is really easy for the admin to update the website.

About Database Connectivity-

As we have a database of the whole dynamic elements of the website, this was achieved with the help of PostgreSQL and its interface pgadmin. Listings, Contacts, and Realtors are classes that are defined in the models.py file. All the information shown are the objects of the models. That’s due to the ORM (Object relational mapping) the concept that transitions our object–oriented data into a relational form.

The **settings.py** file is present at NewLine/rem1/settings.py.

The database configuration was done on this file like –

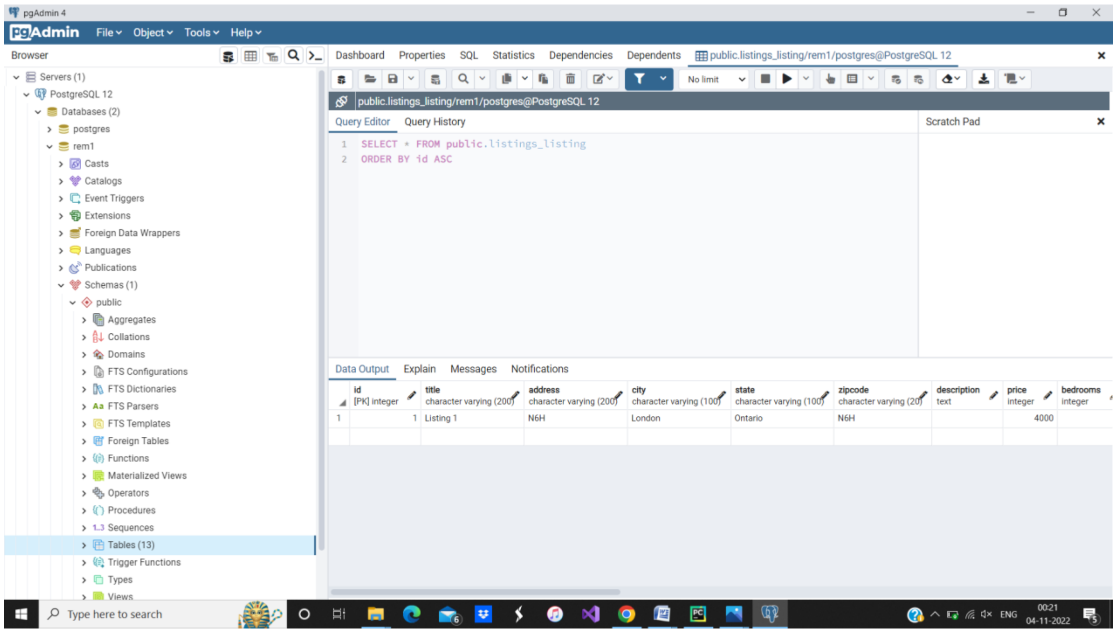


```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'rem1',
        'USER': 'postgres',
        'PASSWORD': '1234',
        'HOST': 'localhost',

    }
}
```

The database 'rem1' was created in pgadmin and after creating it the command used was- (>>python manage.py migrate) to make migrations(connecting models to the database).

Meanwhile at pgAdmin, the databases are present in the relational form-



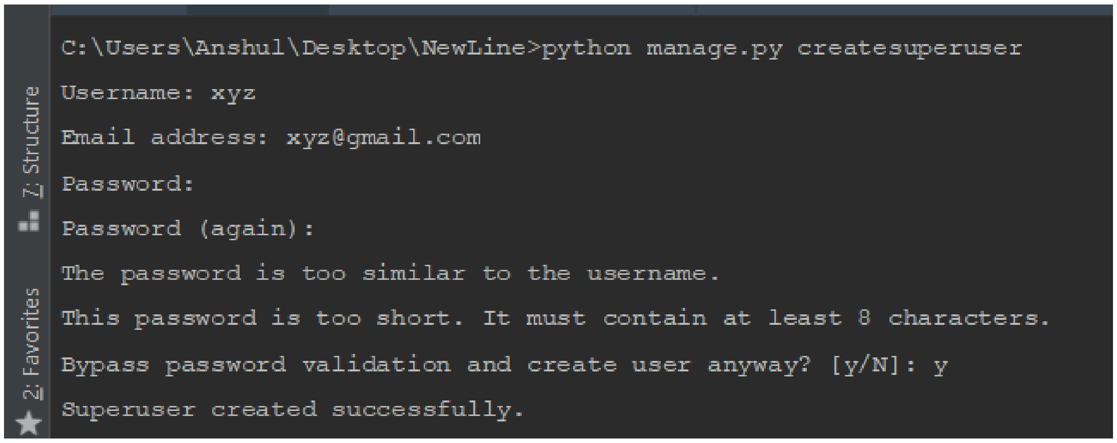
```
SELECT * FROM public.listings_listing
ORDER BY id ASC
```

	id [PK] integer	title character varying (200)	address character varying (200)	city character varying (100)	state character varying (100)	zipcode character varying (20)	description text	price integer	bedrooms integer
1	1	Listing 1	N6H	London	Ontario	N6H		4000	

Every model present in the Models.py file has a relational schema present in the pgAdmin under rem1 database schemas.

User Management

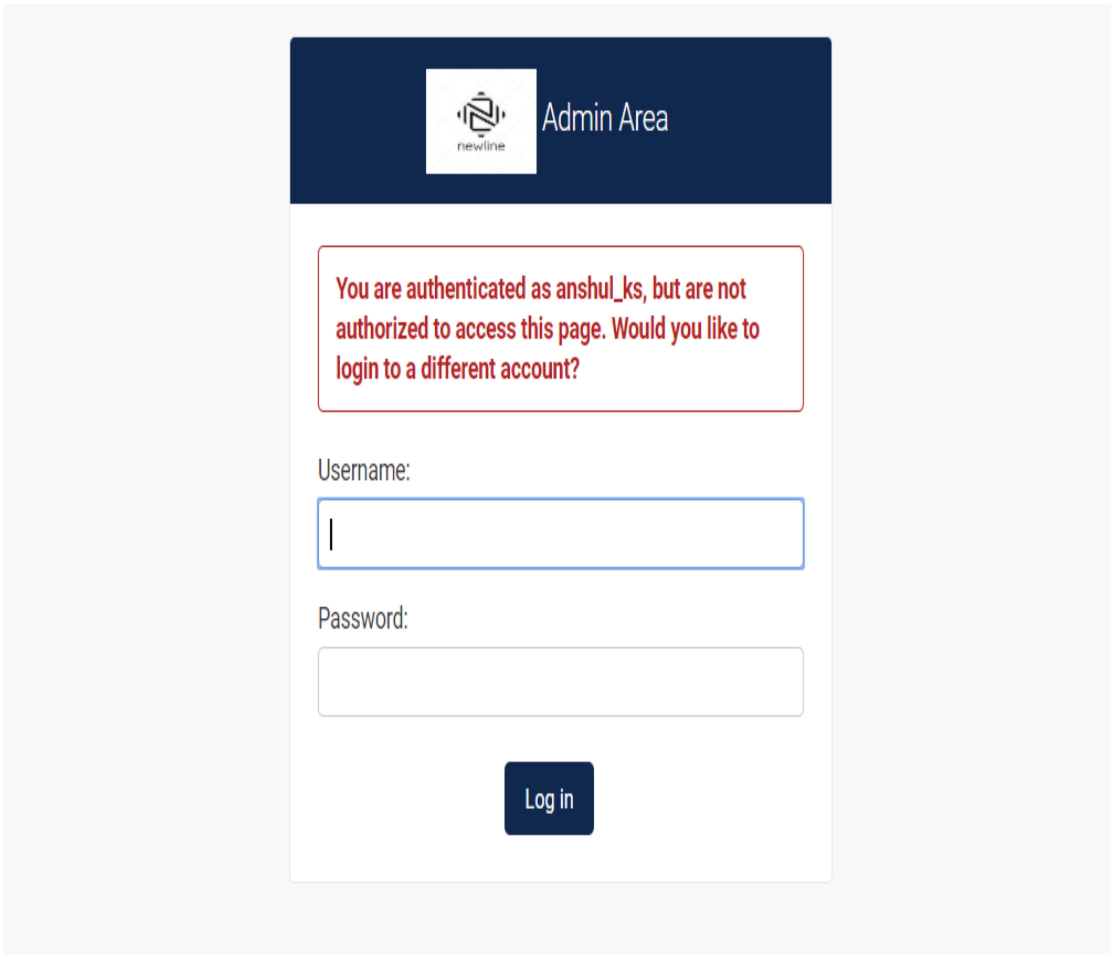
No other user can access the /admin panel because the admin panel is only for the superuser which was created in the python terminal.



```
C:\Users\Anshul\Desktop\NewLine>python manage.py createsuperuser
Username: xyz
Email address: xyz@gmail.com
Password:
Password (again):
The password is too similar to the username.
This password is too short. It must contain at least 8 characters.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

As we can see, we used the previously mentioned command >>python manage.py createsuperuser

Only usernames registered like this are the admin of the website. If a user registers with the Register link of the website navigation bar and tries to enter the /admin panel, the following error will occur-



You are authenticated as anshul_ks, but are not authorized to access this page. Would you like to login to a different account?

Additional Information-

- If the project is imported to another system's IDE, installation of the components and the environment must be verified.
- Django can be installed using the following command in the shell console – pip install Django.
- Pillow and Psycorg2 can be installed using –
- pip install psycopg2-binary
- pip install Pillow
- **Recommended**- always make migrations before running the Development Server.
- The lightbox (image gallery) used on the webpages was replaced by an open source lightbox code (Lightbox v2.10.0), originally created by Lokesh Dhakar (https://github.com/lokesh/lightbox2/blob/master/LICENSE), (Copyright 2007, 2018). This was done to learn and understand the potential optimization of jQuery.
- The testing on this application was done through Selenium and TestNg (Please check the Test Report for more information).

Important Links

Django Documentation, https://docs.djangoproject.com/en/4.1/

Introduction – Bootstrap, https://getbootstrap.com/docs/4.1/getting-started/introduction/