1. INTRODUCTION

This is a report comprising of the knowledge about the website developed and deployed using Django as a framework and python as backend language, HTML, CSS and Bootstrap as frontend languages and if we talk about deployment the AWS (Amazon Web Services) were used for example EC2 and AWS CDK etc.

This report contains a thorough information about the programming languages and the various platforms used in the production of the project.

So, this project is a web application developed in python using Django Framework whose objective is to serve the information of the properties available in town for buy and rent purpose to the users. Which ultimately satisfies the CRUD operations of web App. As it consists of basic user authentication and registrations process involving login as well. A user can Login by first registering itself in the website and the logging in user is redirected to the home page where he can show case his property by filling in basic information needed. And can see the property displayed by other users that makes it a common platform for the buyers and sellers and for rent as well. And the deployment is done on the ec2 instance of windows. An instance is virtual machine running on a server and we run the project there and makes it available for the internet.

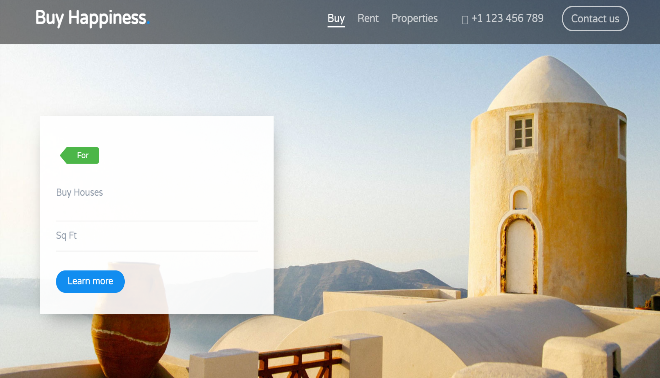
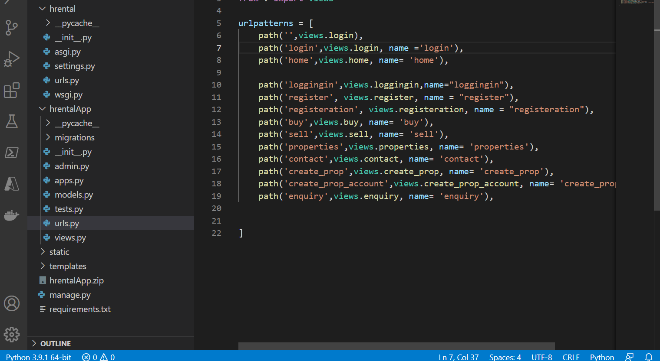


Fig 1.1 : - Home Page

1. Project Requirements

* Programming Environment: One would need a programming environment where he can develop the web application.

One of the environments I used is Visual Studio Code (VScode). This Vscode has all the languages plugins and we can develop anything of any language.

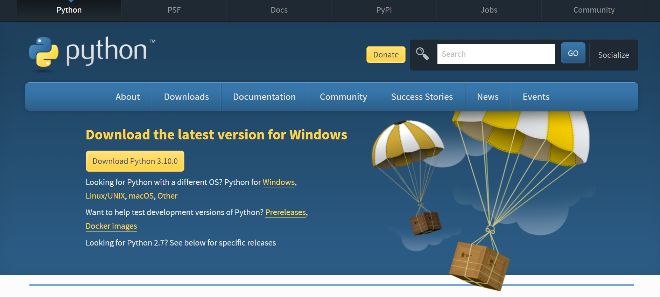


This Screenshots comprises of a code snippet from vscode.

* PYTHON

One should install python from python.org for windows. He can install the latest version of python. Add python to path is also a mandatory step while installing python globally in the machine.

And it is mandatory to install python because we would be using Django as web development Framework, and it is based on python totally, that makes python a mandatory requirement.



This Snippet showcases the website from where you can download python.

* **DJANGO**

Django is a python based web development Framework which provides security, models , views , template , authentication , database connectivity and local server facilities for our web application. Command to install Django:

*Pip install Django*

One can install Django from Django.org its official web site.

* **AWS**

One needs a AWS (Amazon Web Services) account which is free tier we used. Free tier is for students using for non-commercial projects which are academic projects like this one. In AWS account we create EC2 windows instance and some security groups and deploy the web application on that instance

* **MONGODB**

We used MONGODB as database for this web application. And we would need to install mongo atlas for the Gui representation of the database.

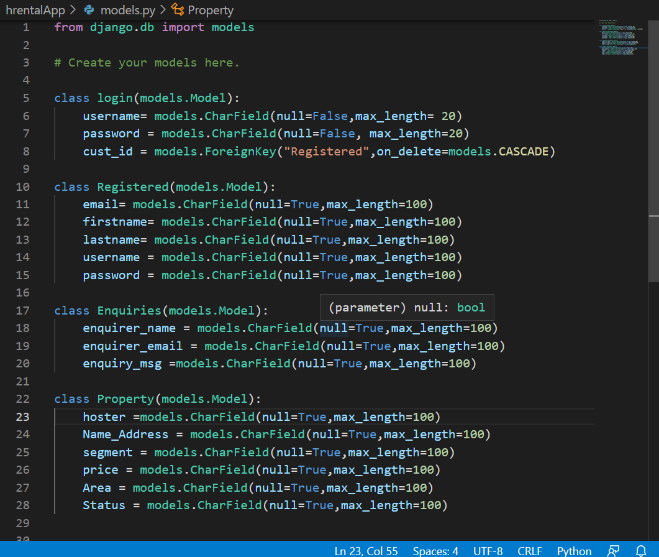
1. **ARCHITECTURE**

Django uses MVT architecture which stands for Model View Template.



**MODELS**

Models are the database schema definition which on migrations form the tables in the database



We used four models in models.py

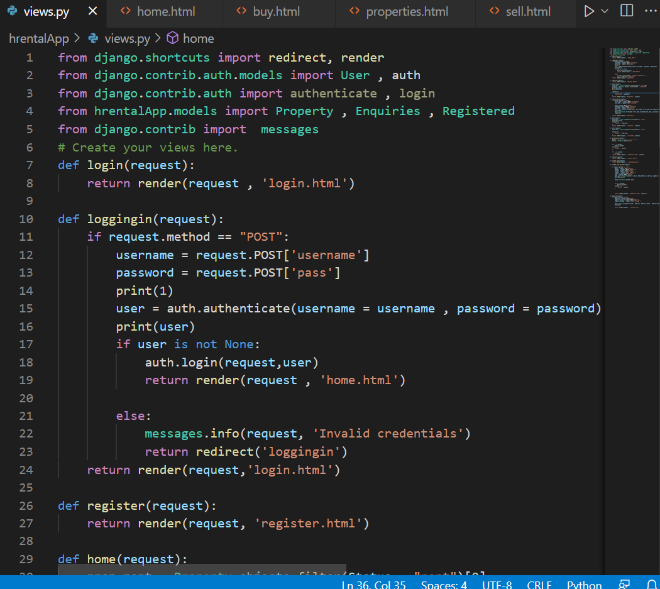
* Login
* Registration
* Enquiries
* Property

Actually, these models are classes holds the meta information about the schema of the database tables each model represents the table individually.

**VIEWS**

Views.py is a file consists of all the functions of the web application such as login, registration, enquiry, property etc.

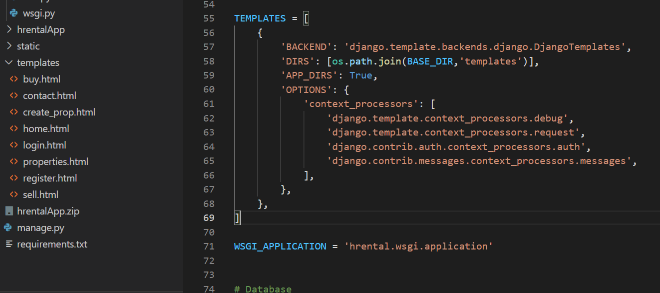
It holds all logics and form action , button action functionalities. So main logic of our web app lies here.It is defined in the App directory of the project file architecture. We import all the models in this using from and import commands so that d=functions can use them on the go.



**TEMPLATES**

In this project we used multiple HTML, CSS templates consisting of the html divisions and form divisions. And CSS is for styling of those elements of HTML, apart from these we use JAVASCRIPT and BOOTSTRAP for further designing of these templates like animations and element calls using JS(JavaScript). We used CDN for Fonts.

In Django Templates is a directory in projects working directory which is referred in settings.py of the project.

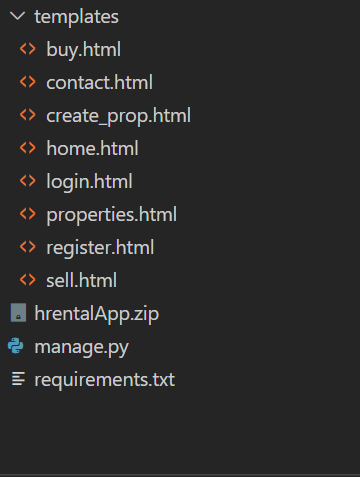


This snippet shows the Templates reference in the settings.py.

The various templates we used are as follows :-

* Login.html
* Register.html
* Home.html
* Buy.html
* Sell.html
* Properties.html
* Create\_prop.html
* Contact.html

This Snippet shows the multiple templates used in the web application.

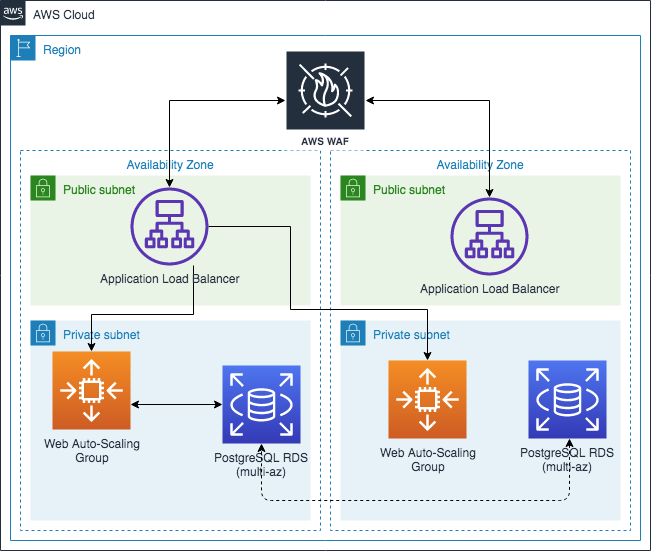


These templates are rendered by the view functions describes in views.py according to the need and flow of the web application.

**AWS & Django Architecture**

The cloud platform we used in this web application is AWS, we created an instance which has windows 2019 as operating system. We create security groups for opening the ports of the server for the request coming from the internet users.

This is the architecture diagram totally satisfies our web application.



So, the EC2 instance holds are whole project on the server we host our web app on the open port of HTTP which is port 80 from that port our web app is welcoming all the other user’s requests which is handled by the application load balancer which is a public subnet served by EC2 by default and when request enters private subnet it is handled by Web Auto-Scaling Group and interacts with the Database and serves the data on the html page as response of that request.

1. **AWS** **SERVICES**

Services used of AWS as in particularly was EC2.

* **EC2**

It is an amazon’s computer rental service as in it provides us a remote server with the operating systems (windows, ubuntu, Debian, CentOS, FreeBSD) with the desired configurations where we can configure the memory needed according to our needs. Along with the storage we can configure security groups like HTTP, HTTPS, DNS, RDP, RDS,

HTTP: - port 80

DNS: - port 443

RDS: - port 3389

* **Why EC2?**

Used EC2 out of experience and ease of instance creation an instance can be created within a few clicks which makes it faster and being a student its pocket friendly as we can user free tier in EC2 and have a basic configuration needed for the web App.

As other cloud services are expensive than AWS EC2. And they Does not provide free Student tier for Academics use. And for the deployment on AWS EC2 the internet has proven to be bigger community in the error resolving activities.

So, after all we used EC2 architecture for the deployment of web App on a cloud service.

1. **LIBRARIES**

* OS

The OS module in python is used for interacting with operating system. It comes under python’s utilities module. Which helps in the interaction with the operating system of the computer system.

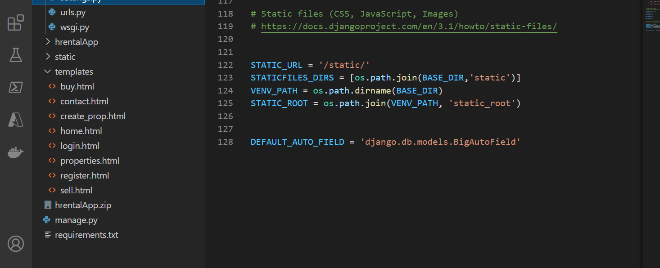
* Django-allauth

This helps in Django authentication as well third-party authentication.

Django-auth is used in this project for user authentication and login functionality.

1. **IMPLEMENTATION**

This whole web app is implemented using Django, as in process we have templates, static files which are used in templates as {% load static %} this means all the static files such as style.css images, fonts, JS files are loaded into html templates. Figure below gives a idea about this mechanism.



This whole process is done by a command which is :

*Python manage.py collectstatic*

After this command we simply run the web app using *python maange.py runserver .*

After this command the web app runs on local server on the address <http://127.0.0.0:8000>

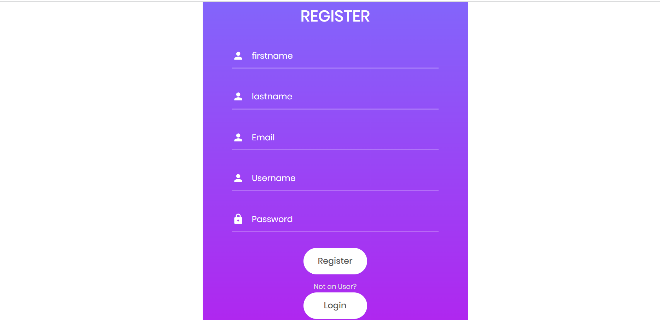
To run this web app on the instance and serve it to the internet would need a command *python manage.py runserver 0.0.0.0:80* this means the server is running on port 80 which is a global Http port from which internet users can access this web app.

This all information was about running this web app but how it works approximately everything is defined in the report apart from routing of the web pages defining the flow of this web application which will be defined in the next section.

1. **Integration**

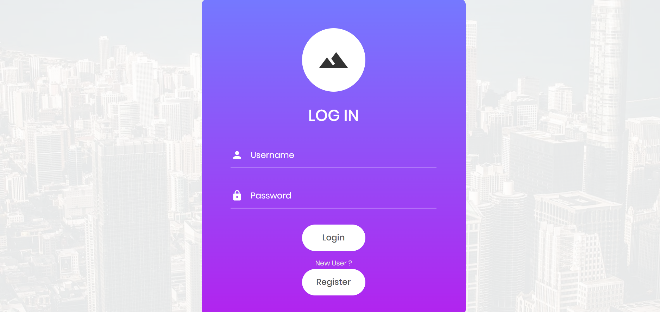
In integration we have multiple templates in the form of .html which are ready to be integrated according to the flow of the web app and use cases.

The flow here is that a user will register himself and will be redirect to the login page.

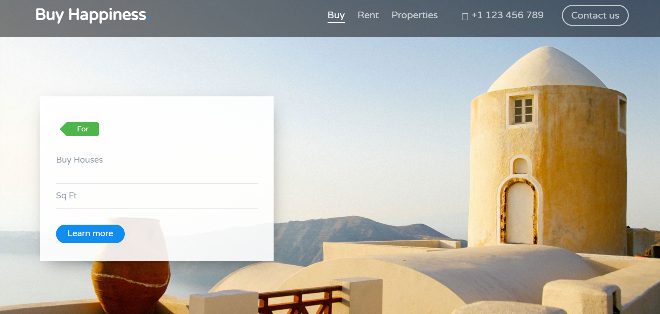


*Registration page*

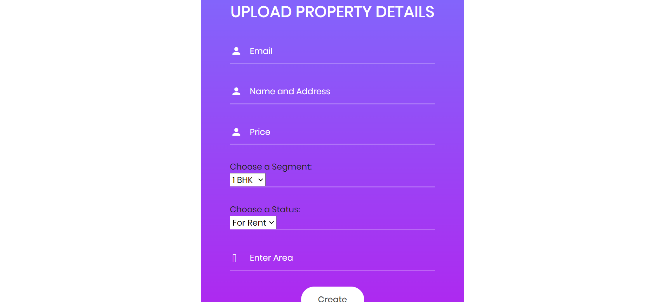
After registering user will be rendered to login page.



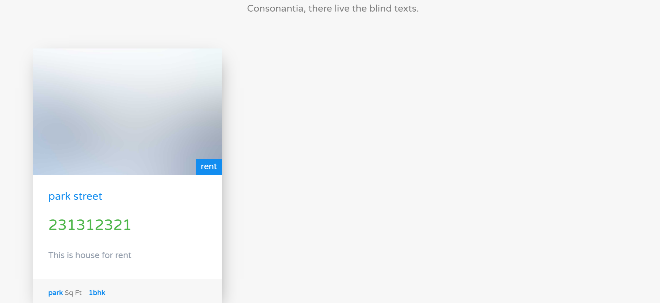
This login page will authenticate if the user is valid or not, if user is valid user will be rendered to home page.



After reaching home page user can create its own property to sell, buy or rent. Using this form.

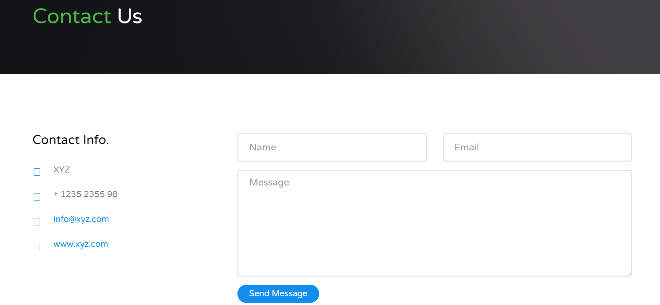


After creation of the property user can see its properties where all can see is on all property’s pages.

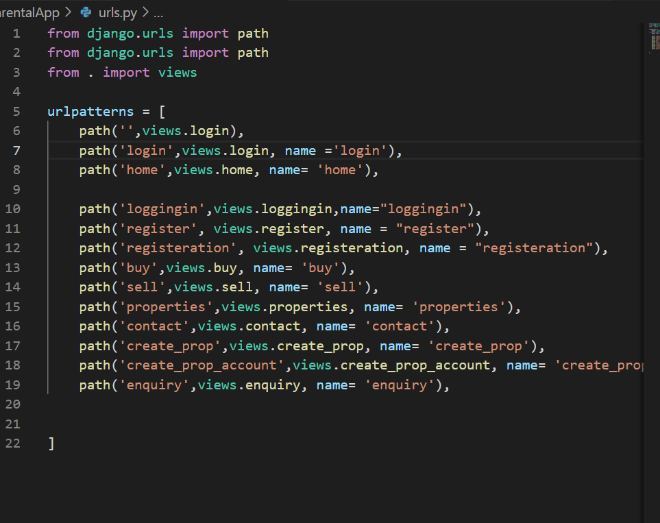


Here we can see the property user created .

If a user wants to enquire he can send enquiry using contact page.

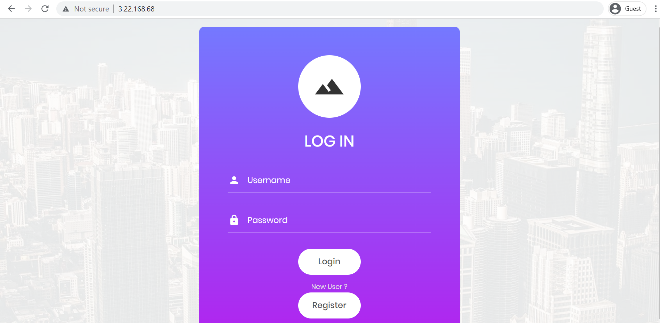


These all pages are linked using routing which is done in urls.py of the application.



1. **Conclusion**

So here we conclude that after hosting this web application on address 3.22.168.68



Django is a open source framework helped us a lot in achieving this stage of this project.