**Chapter 1: Introduction**

**1.1: Brief overview of the work**

**@param**

**1.2: Objective**

**@param**

**1.3: Scope**

**@param**

**1.4: Project Modules**

1.4.1: Selecting Dataset.

1.4.2: Data Processing.

1.4.3: Data Transformation.

1.4.4: Selecting Model.

1.4.5: Display the result.

**1.5: Project Hardware/Software Requirements**

1.5.1: Hardware Requirements

* Computer
* Local Server, etc

1.5.2 Software Requirements

* Python Language
* Basic of some libraries
* Django Framework
* GitHub

**Chapter 2: Literature Review**

**Chapter 3: System Analysis & Design**

**3.1 Comparison of Existing Application with our Project**

**@param**

**3.2 Feasibility Study**

A feasibility study is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable. It tells us whether a project is worth the investment.

3.2.1 Technical Feasibility:

This assessment focus on technical resource required for this system is available at organization or not. All the necessary technical resource exists at the organization to do what is suggested. An organization has sufficient memory storage available to hold the data required to use this system. After

This system has been developed it can be updated, but it will require someone with technical knowledge. This system also provides security to user’s data.

3.2.2 Operational Feasibility:

This assessment involves undertaking a study to analyze and determine whether and how well the organization’s needs can be met by completing the project. There will be sufficient support for the end user from management.

3.2.3 Economical Feasibility:

This assessment typically involves a cost/ benefits study of the project, helping organizations determine the cost and benefits associated with a project before financial resources are allocated.

As this system is not considered for a business purpose, no any pre-investment is required and also no any benefits has been calculated.

3.2.4 Legal Feasibility:

This assessment investigates whether any aspect of the proposed system conflicts with legal acts.

As this system is not considered for a business purpose, so that all the constraints that has to be followed by an respected organization while developing such system will not be hindrance for developer.

**3.3 Timeline Chart**

**3.4 Modules Description**

3.4.1: Selecting Dataset.

* One csv file that contains information about price of a house according to some parameters.

3.4.2: Data Processing.

* After having data in hand, it’s important to pre-process the data to transform raw data in a useful and efficient format.
* Data Cleaning: Dataset can have many irrelevant data and missing parts. So it is required to clean data.
  + Dropping Null values.
  + Filling Null values with mean.

3.4.3: Data Transformation.

* After data cleaning, need to transform data into appropriate form suitable for model training.
* Attribute Selection: In this strategy, new attributes are constructed from given set of attributes to help model.
  + Aggregate function can be used on two or more columns to have new column that may have highly positive or highly negative correlation with target value.

3.4.4: Selecting Model.

* Now data is all set to be trained. But How….??
* So, it is required to select an appropriate training method to train model.
* So, understanding relationship between various set of attributes using histograms, pie charts, bar graphs, etc can help one to select best method to train model.

3.4.5: Display the result.

* One simple website has been created to deploy this model, that takes input as parameters and predict house price accordingly.
* Django framework is used for same.

**3.5 Project SRS**

3.5.1 Use Case Diagram

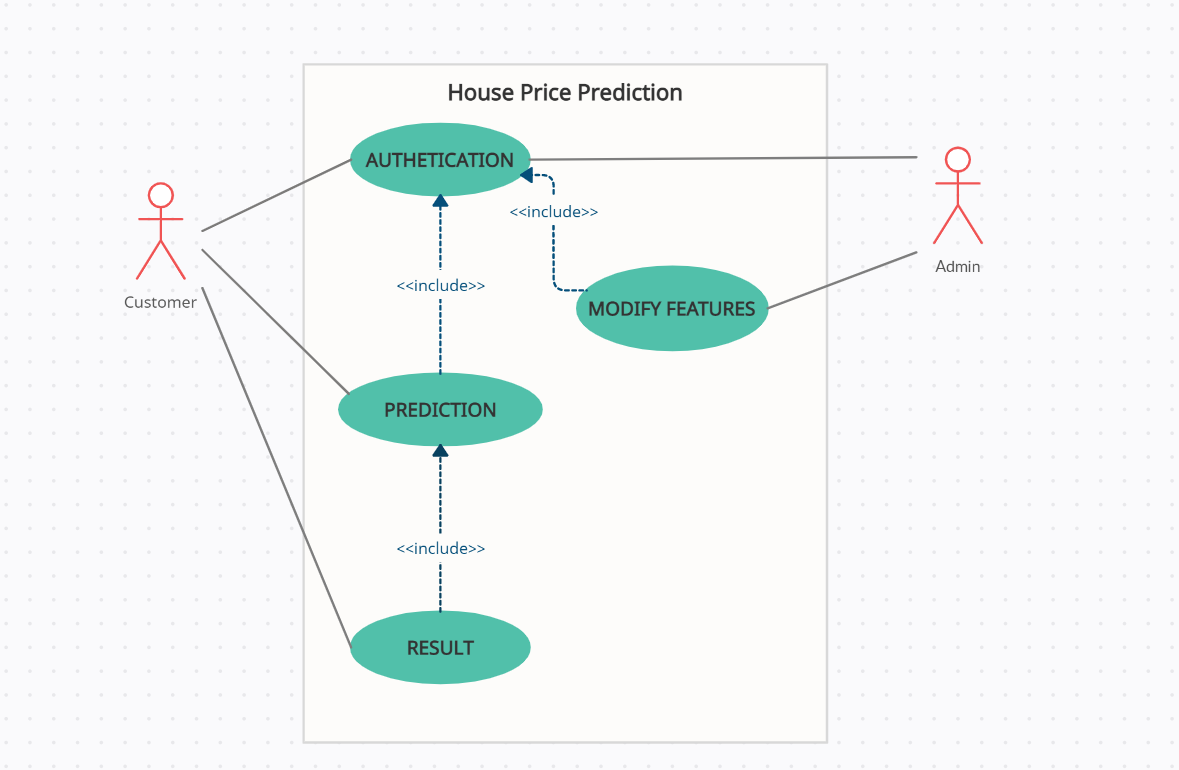
****

Figure: 3.1

3.5.2 Data Flow Diagram

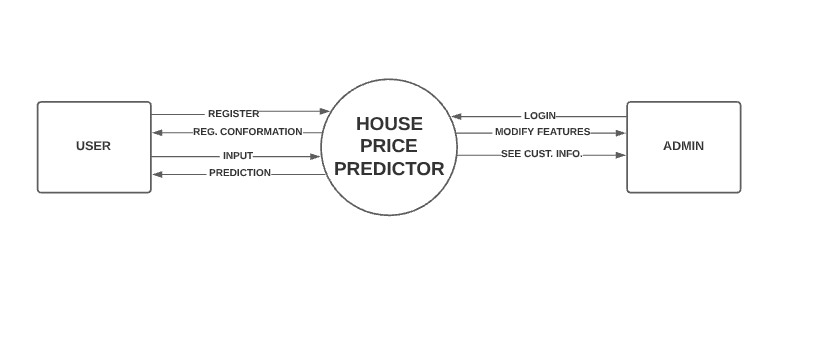
****

Figure: 3.2

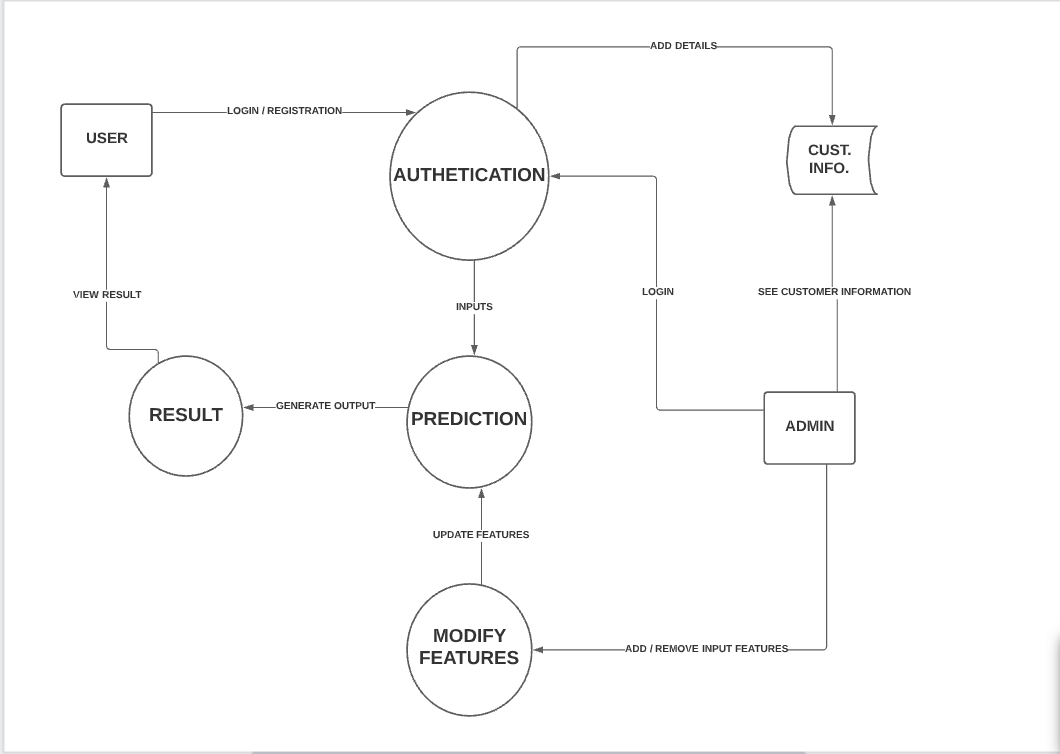
****

Figure: 3.3

3.5.3 Class Diagram

Figure: 3.4

3.5.4 Sequence Diagram

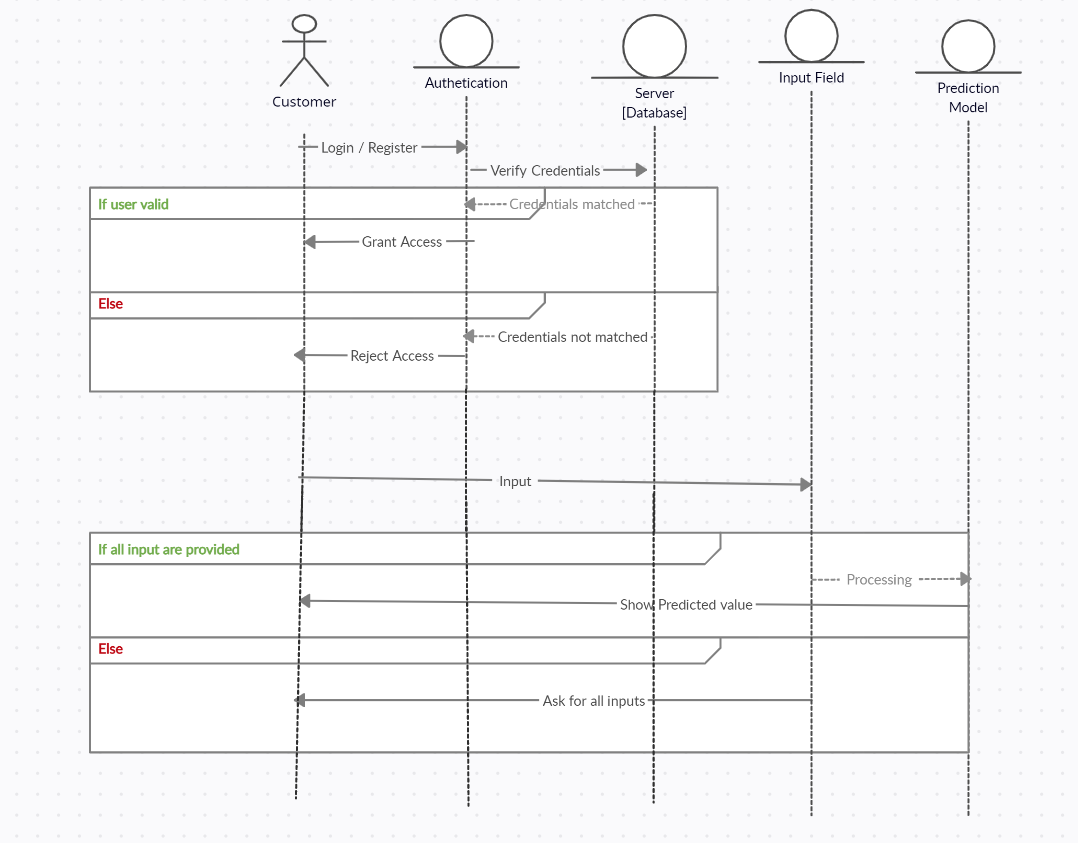


Figure: 3.5

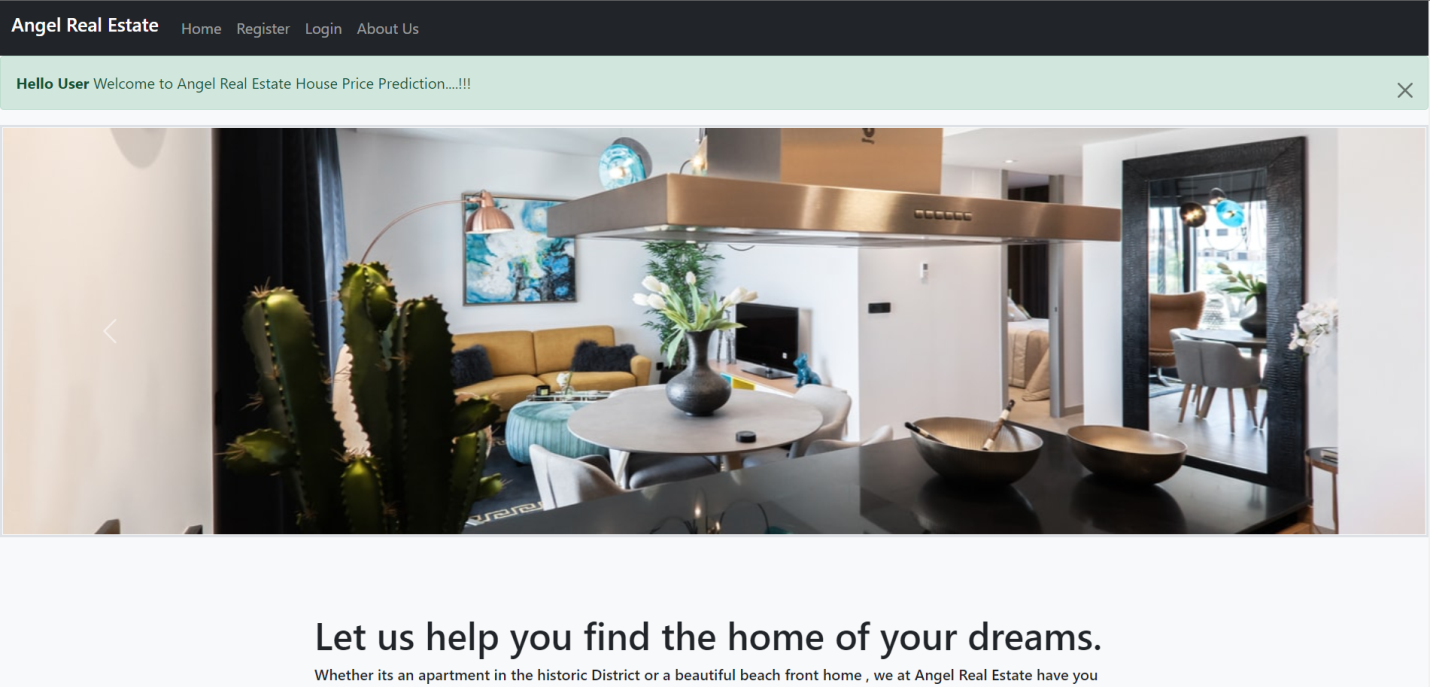
**Chapter 4: Implementation and Testing**

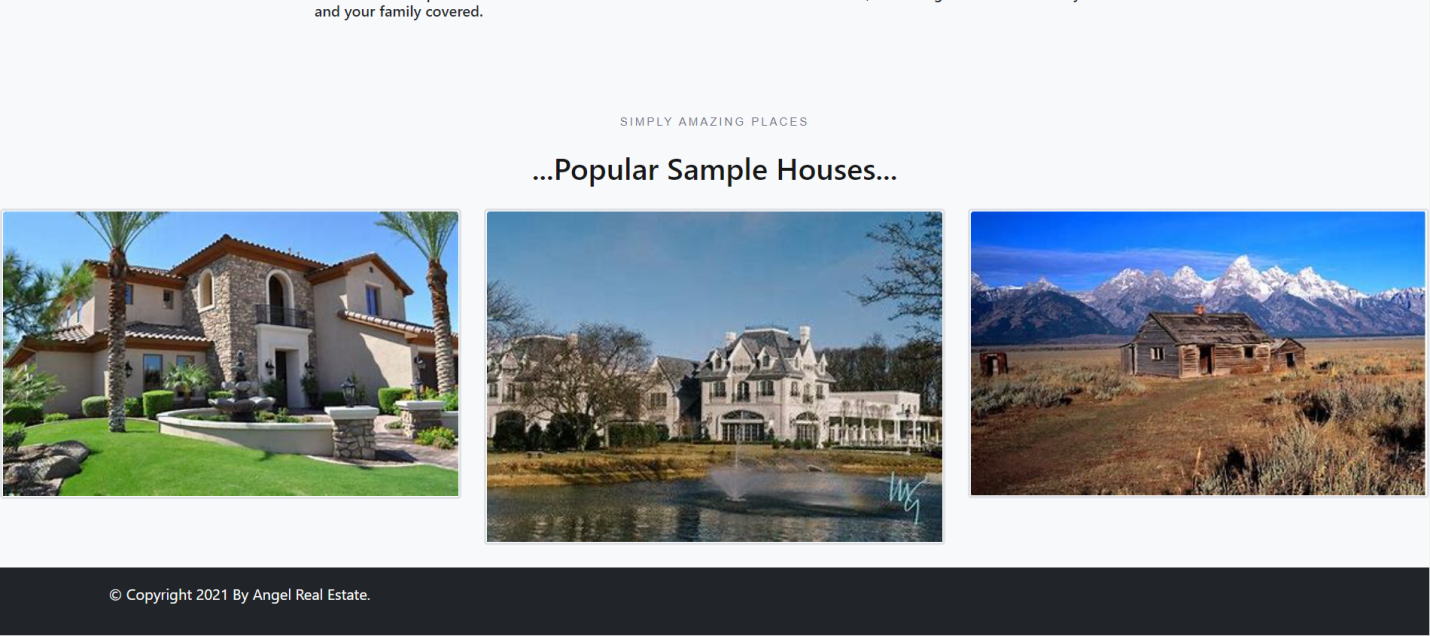
**4.1 User Interface**

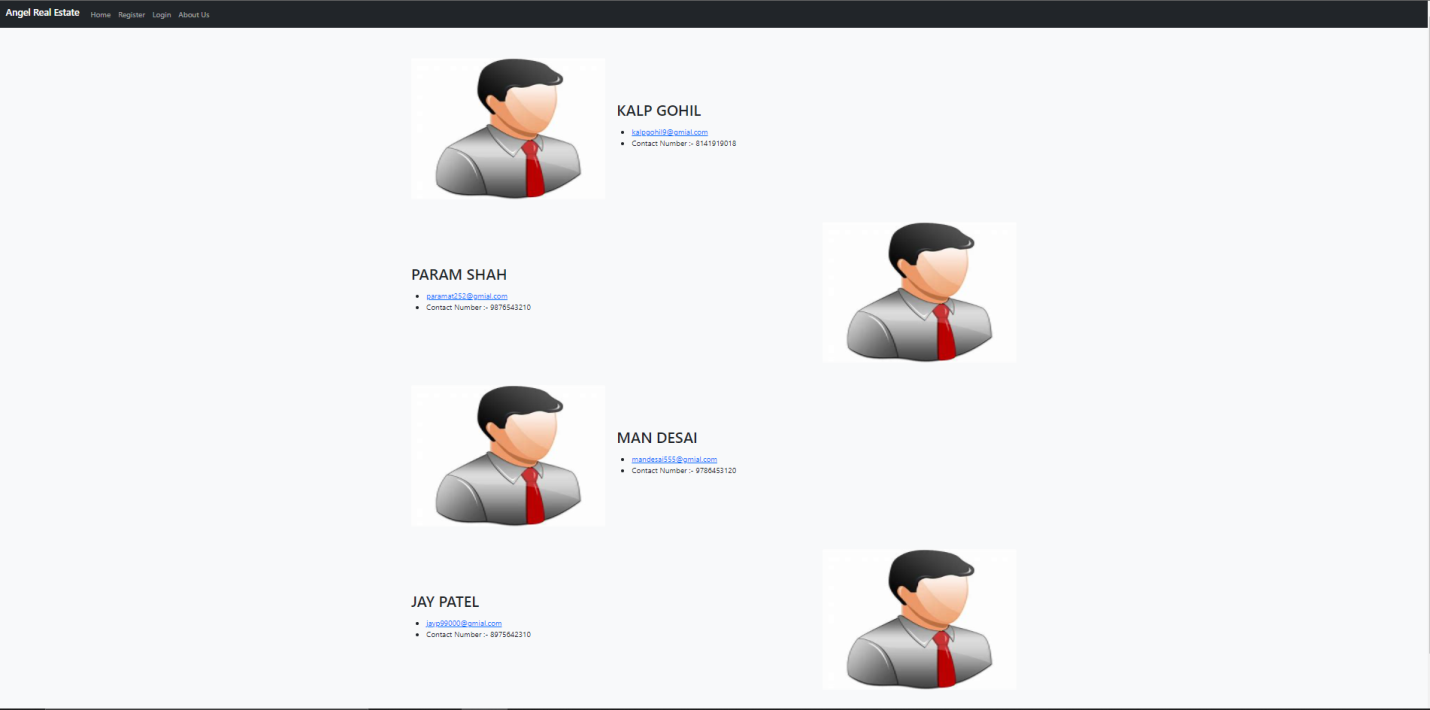
4.1.1 Home Page and About Page :--

Description: The very first User Interface in this project is Home page of website. It is a simple static page. It contains images and some quoted text.

Website also contain About page, which has information about all the contributors of this website.





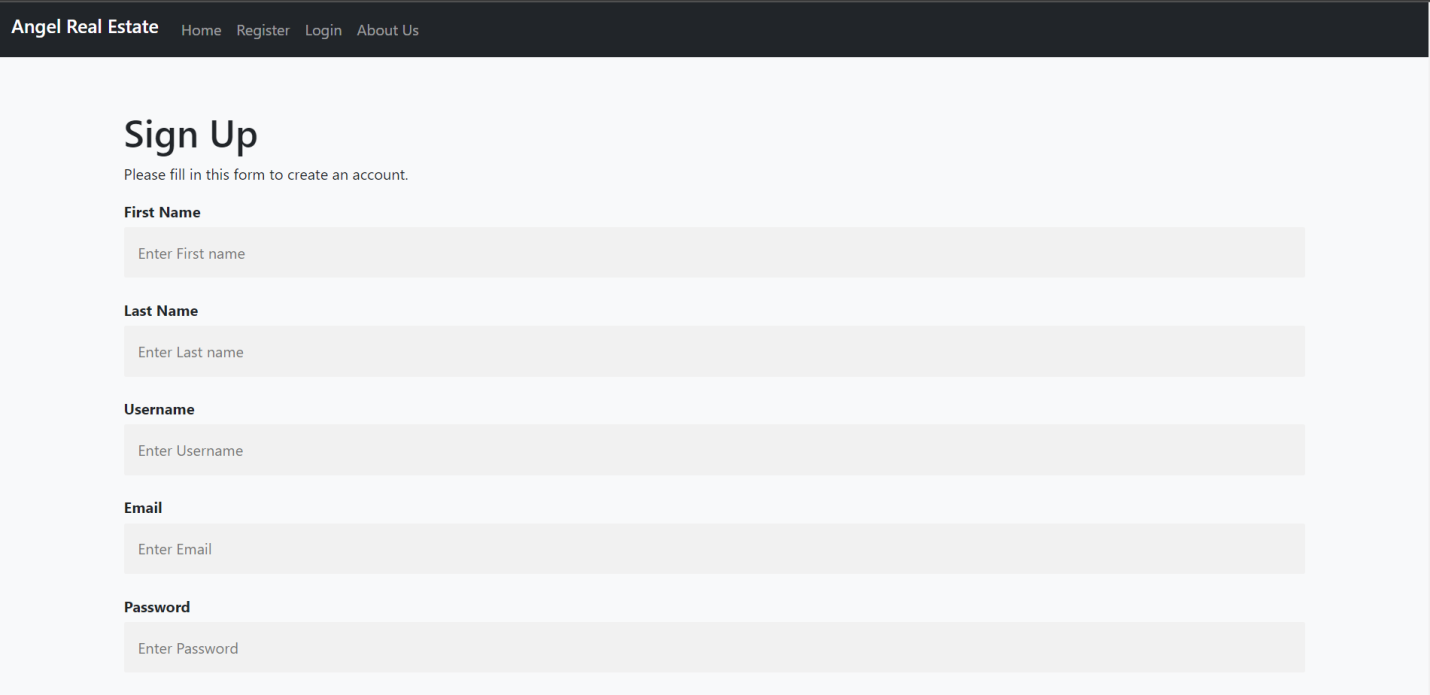


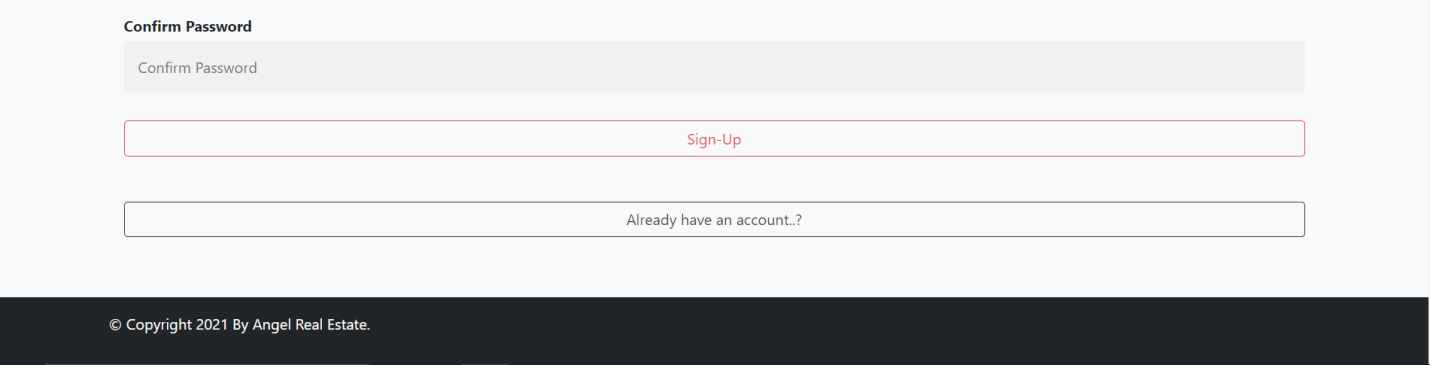
4.1.2 Sign Up/ Login:--

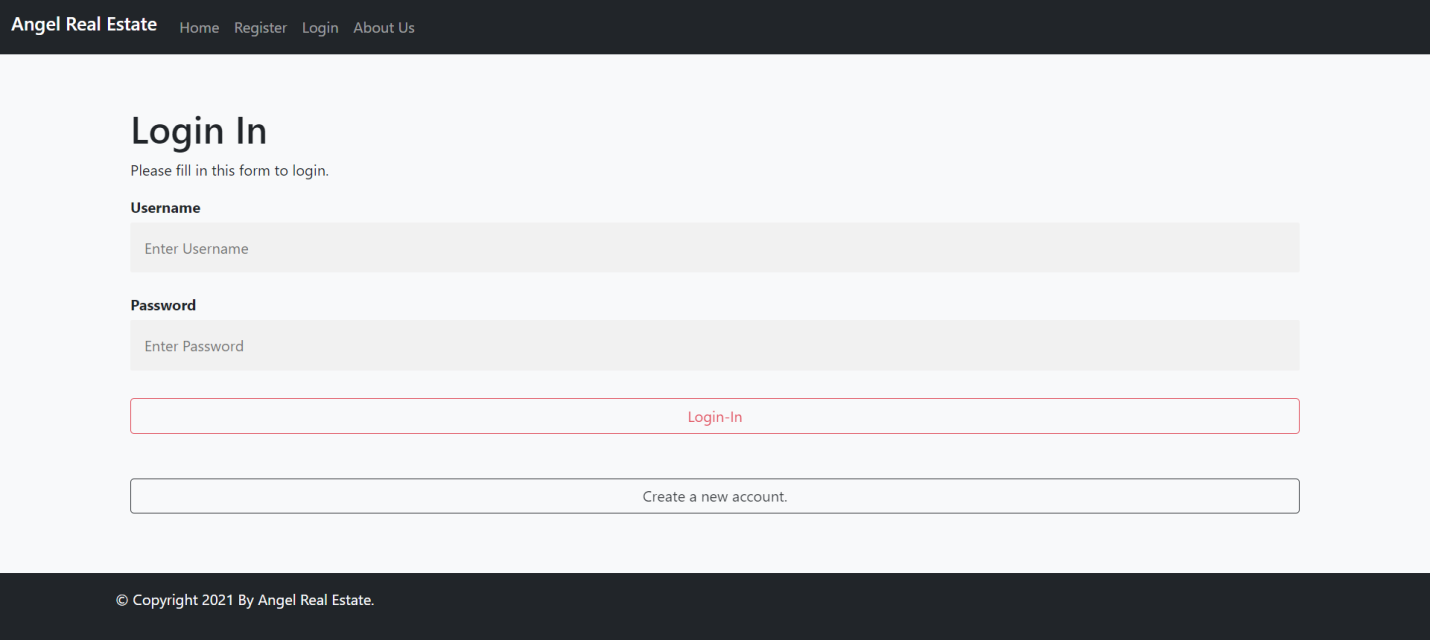
Description: While designing website for any purpose (like for public use or for business part), it is very important to design user sign up page. This will help to analyze our website performance.

In this, it will collect data of the user like first name, last name, email, etc. All this information will be stored in database provided by Django.

If this is a new user, he/she can register on the website and then he/she has all the access to the website. If user is already registered, then he/she can login directly.







Code:-

def register(request):

if request.method == "POST":

fname = request.POST['fname']

lname = request.POST['lname']

username = request.POST['username']

email = request.POST['email']

password1 = request.POST['password1']

password2 = request.POST['password2']

if password1 == password2:

if User.objects.filter(username=username):

messages.info(request, "Username already taken...!!")

return redirect('register')

elif User.objects.filter(email=email):

messages.info(request, "Email already taken...!!")

return redirect('register')

else:

user = User.objects.create\_user(username=username, email=email, password=password1,first\_name=fname, last\_name=lname)

user.save()

login(request, user)

messages.info(request, "You have successfully register. Please Login.")

return redirect('/login')

else:

messages.info(request, "Password does not match...!!")

return redirect('register')

else:

return render(request, 'register.html')

def logout1(request):

logout(request)

return redirect("/")

def login1(request):

if request.method == "POST":

username = request.POST['username']

password1 = request.POST['password1']

user = authenticate(username=username, password=password1)

if user is not None:

login(request, user)

return redirect('/')

else:

messages.info(request, "Invalid Credentials.")

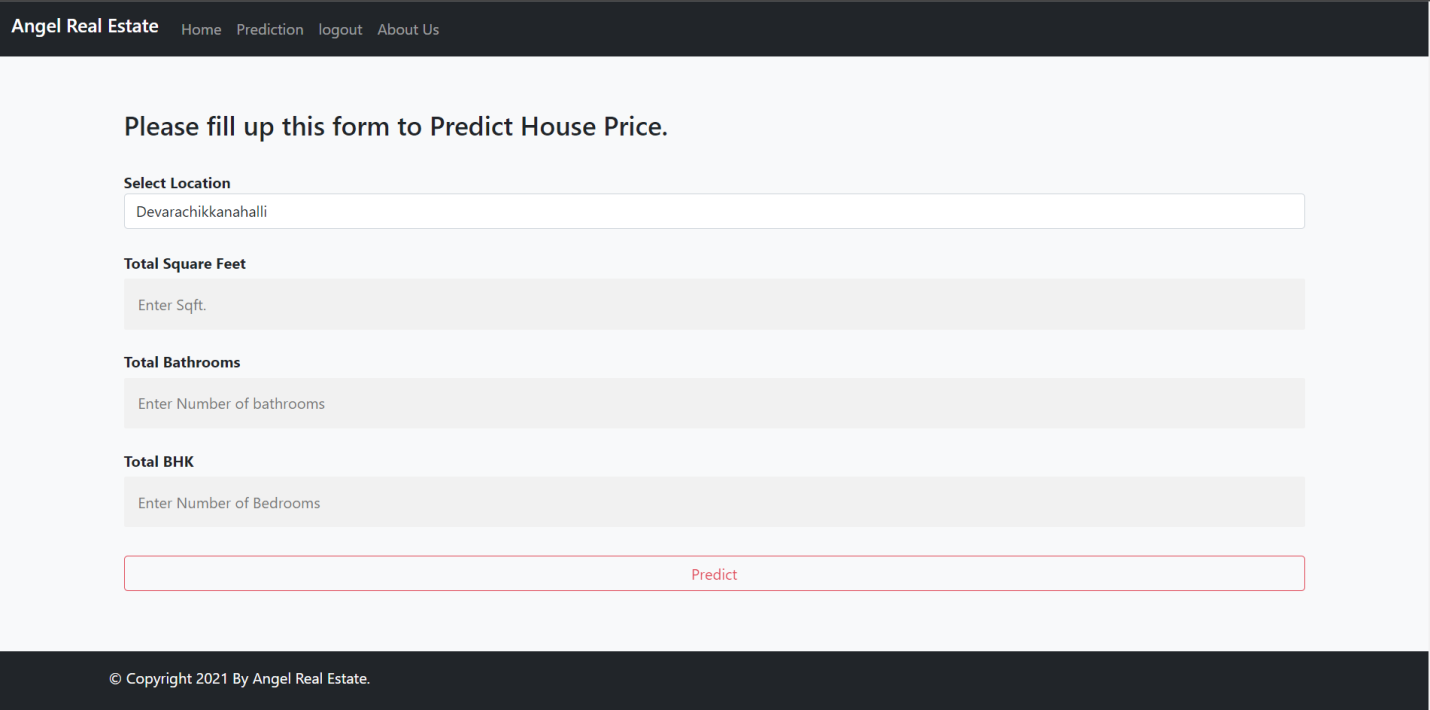
return redirect('login')

else:

return render(request, 'login.html')

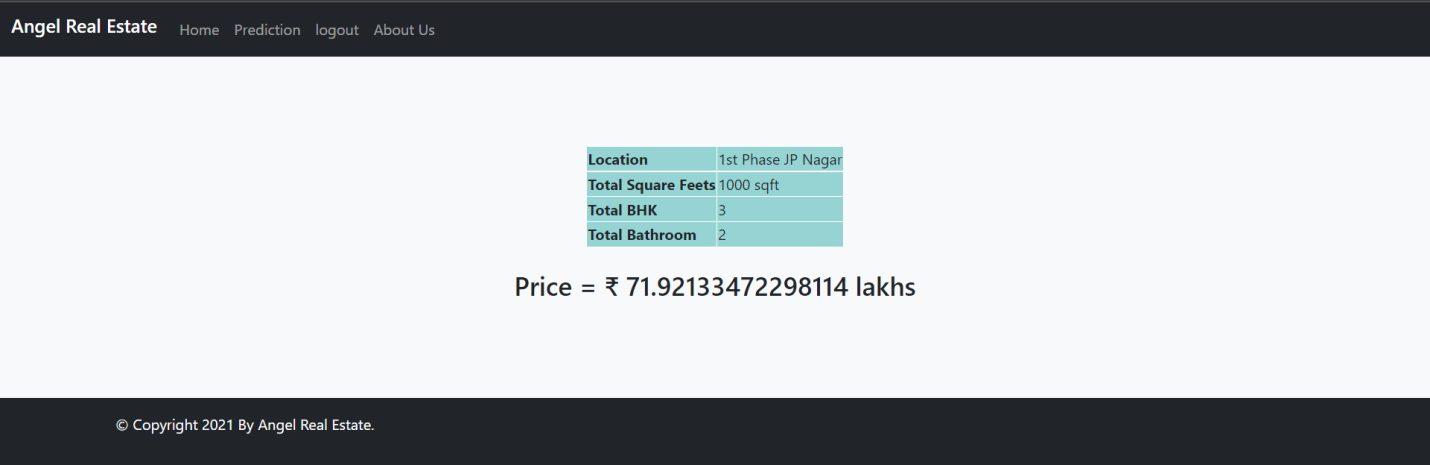
4.1.3 Prediction Page:--

Description: This page is like heart of the website. In this page there is 4 feature of house that has to be entered by user in order to predict price accordingly.



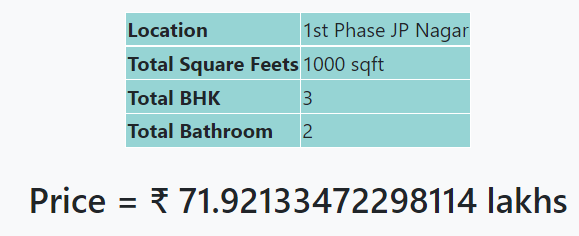
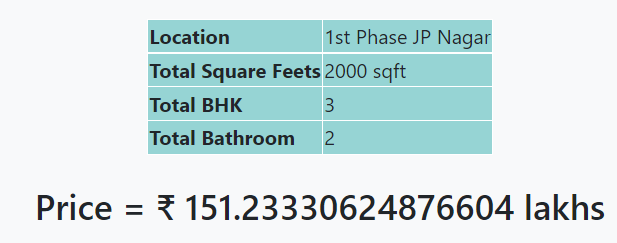
4.1.3 Result Page:--

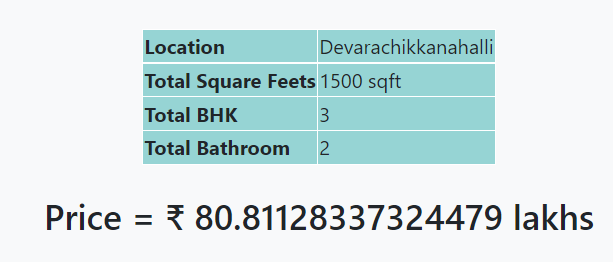
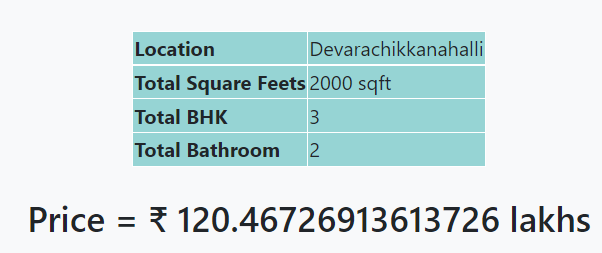
Description: This page is like heart of the website. In this page there is 4 feature of house that has to be entered by user in order to predict price accordingly.



**4.2 Testing using use cases**

Description: This trained model and website together has been tested on different inputs and it gives desire output.

From about, it’s clear that on different parameters, model performs accurately.

**Chapter 5: Conclusion and Future Work**

**@param**

**Chapter 6: References**