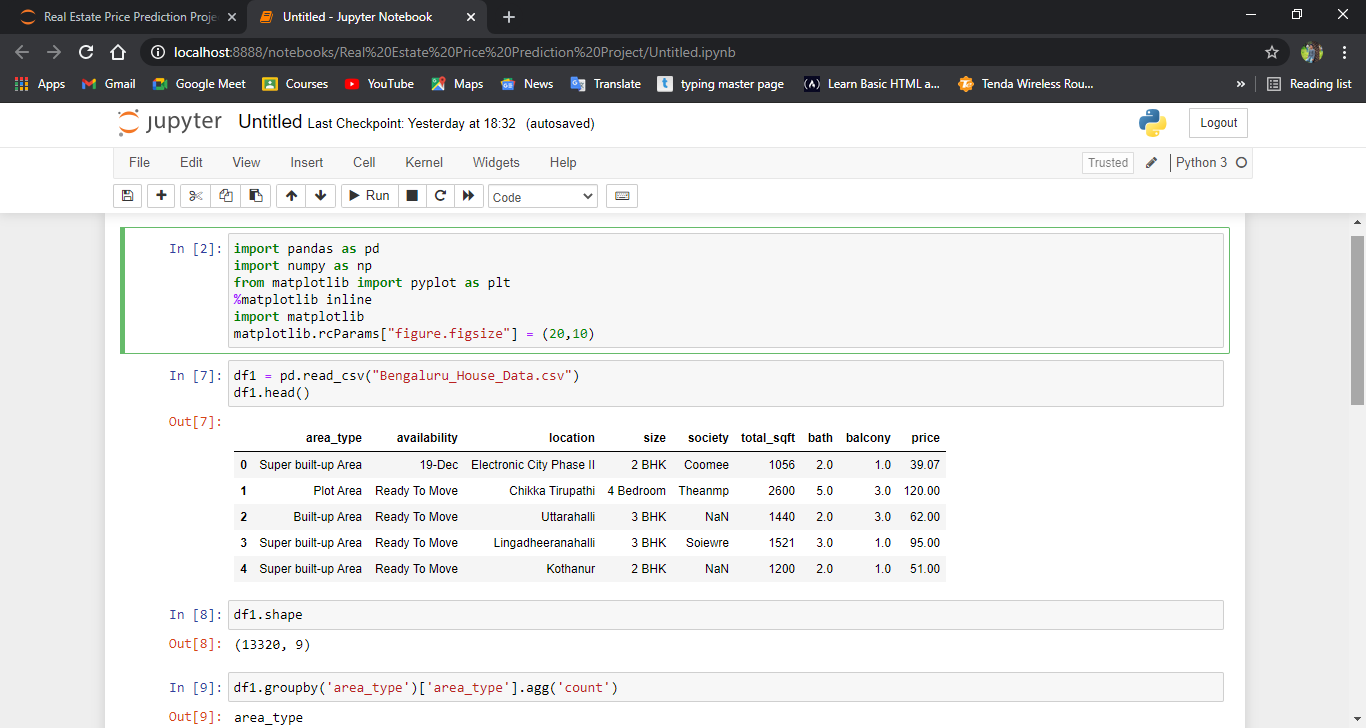
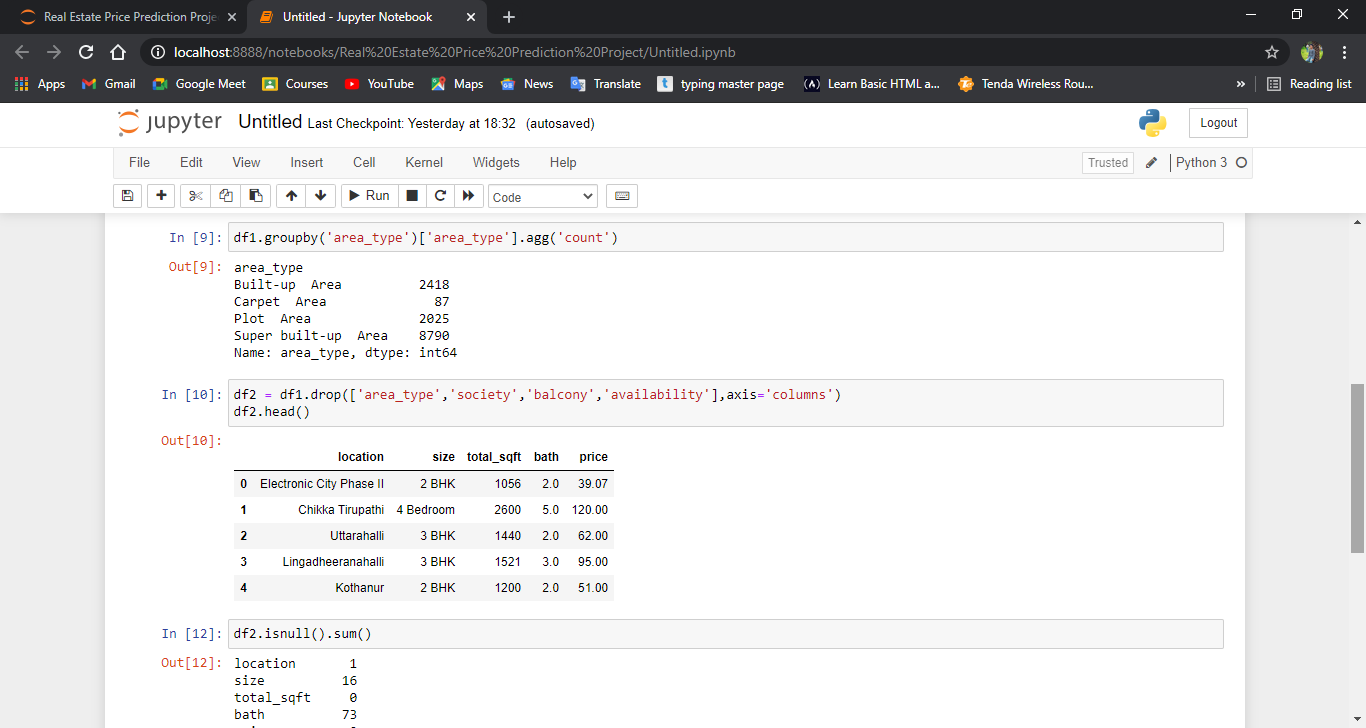
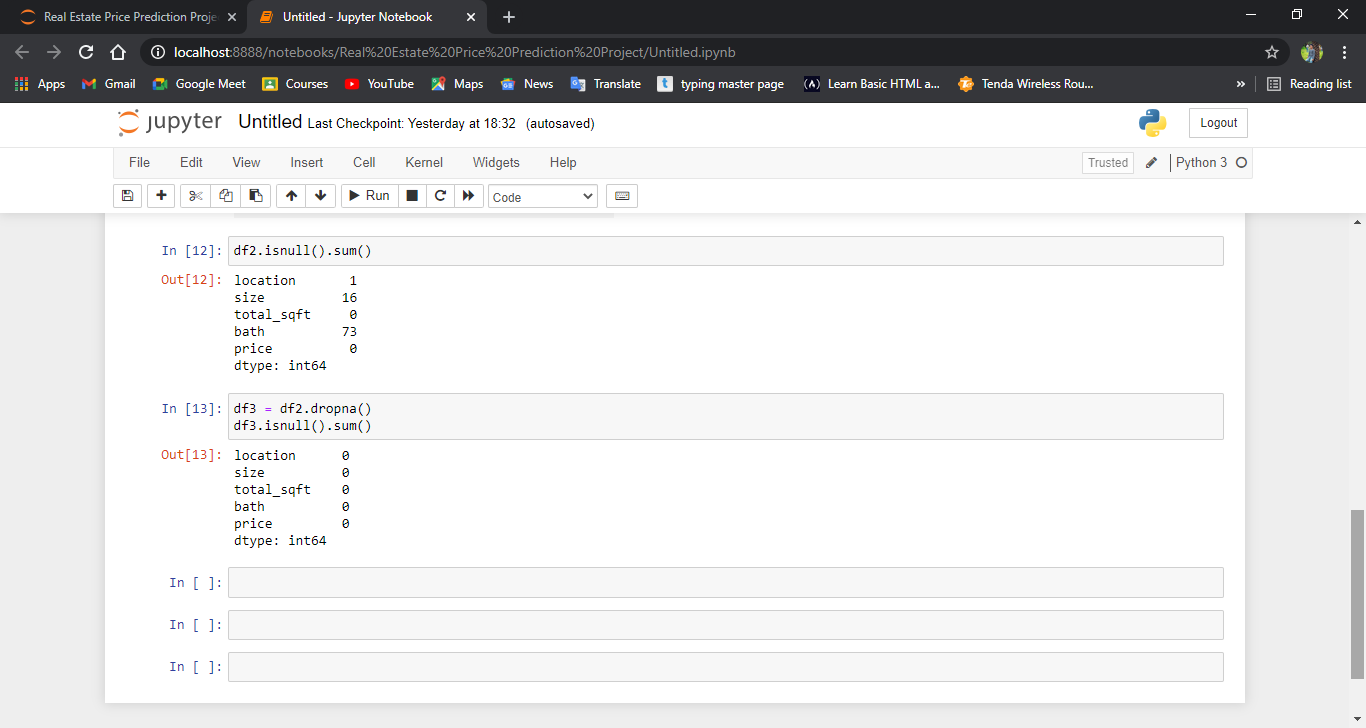
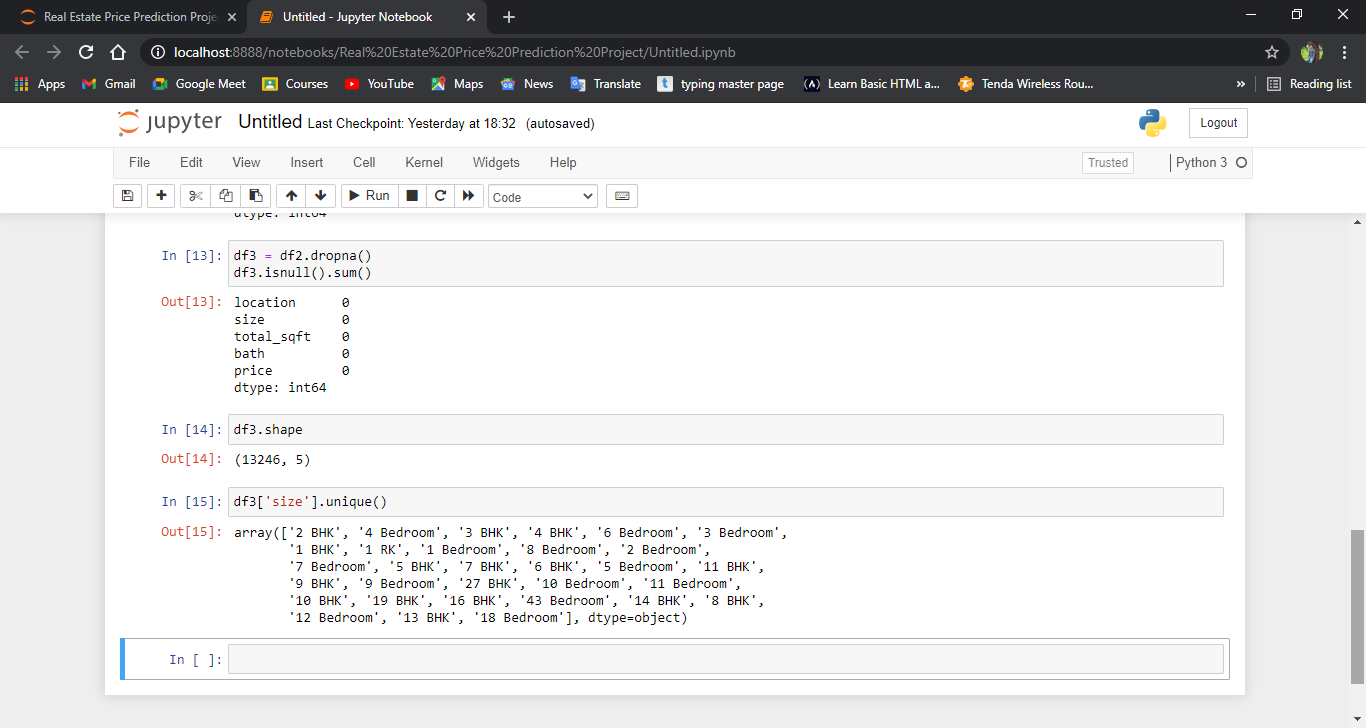
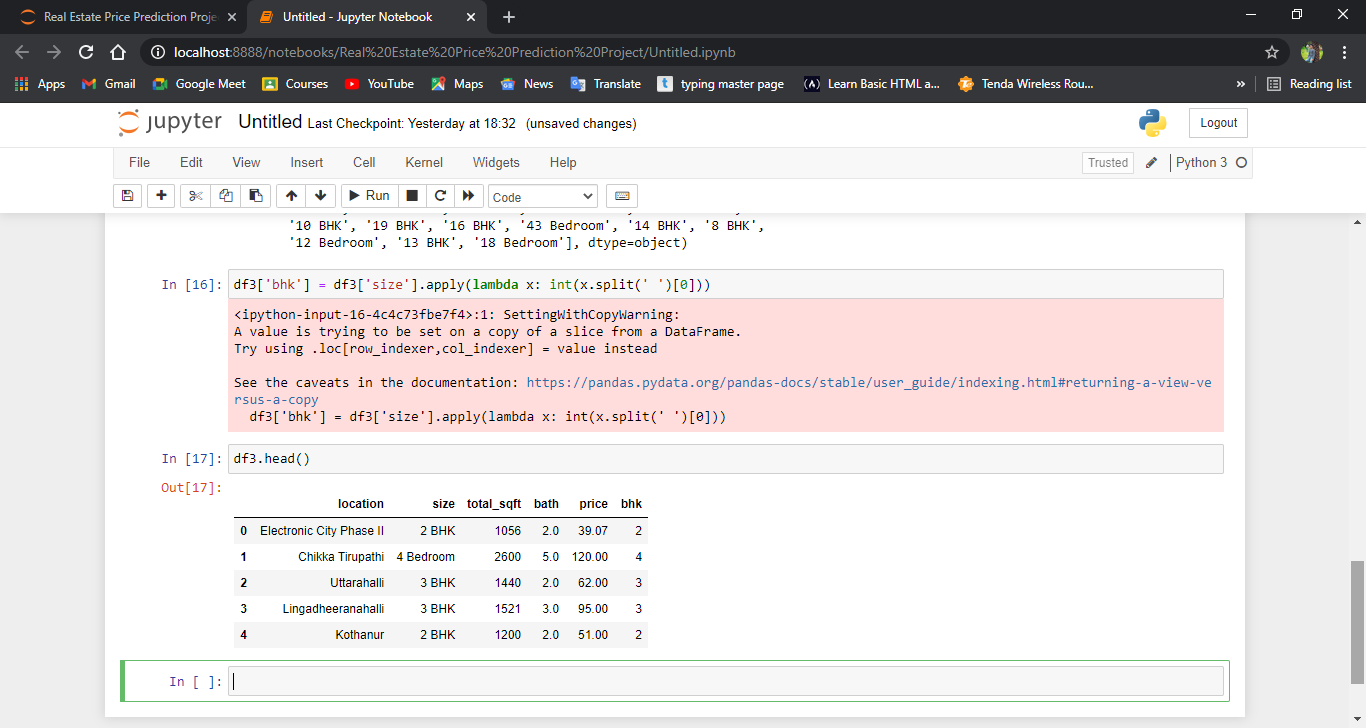
Step1:-Data Cleaning

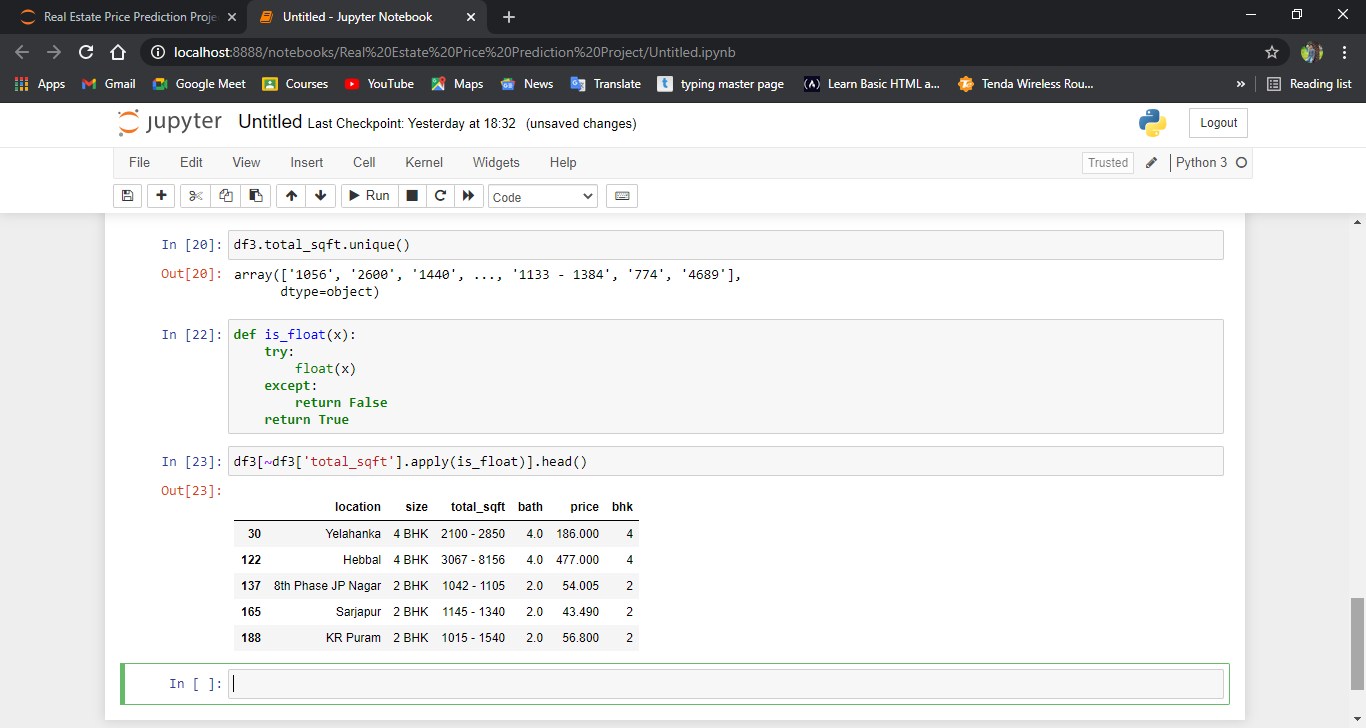


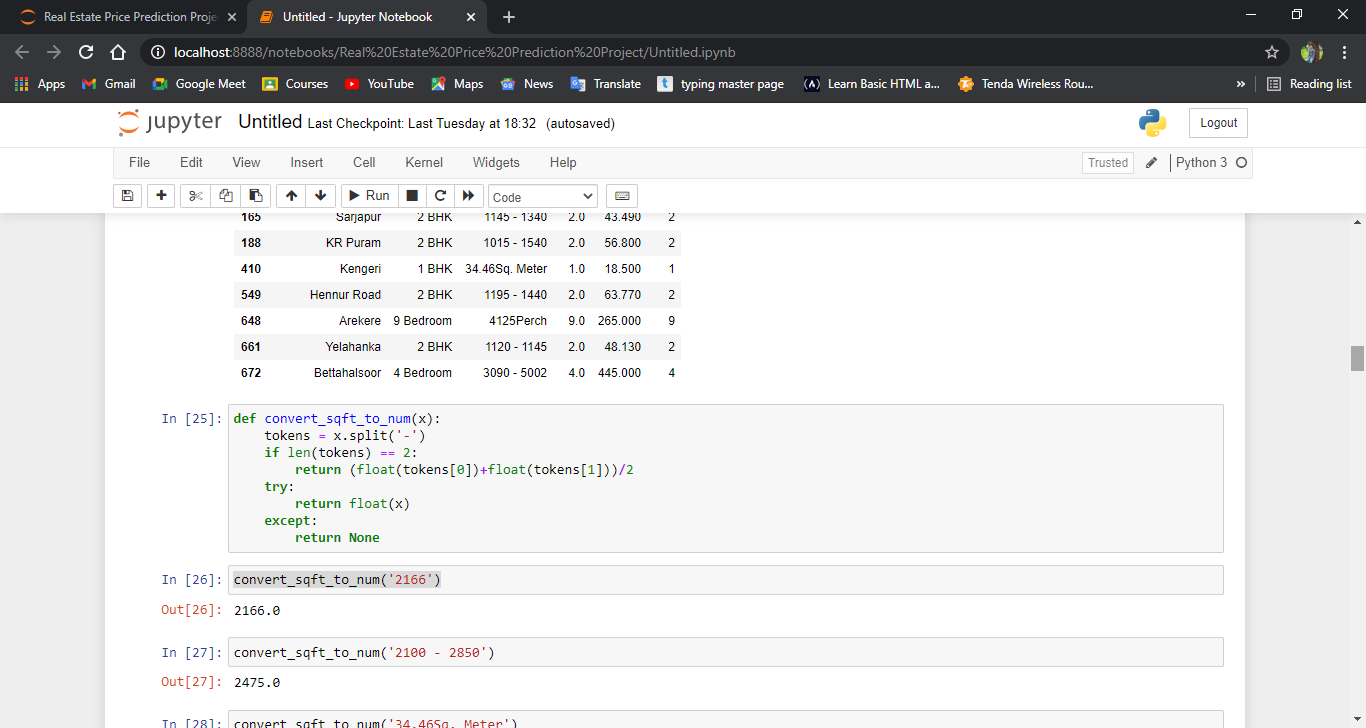


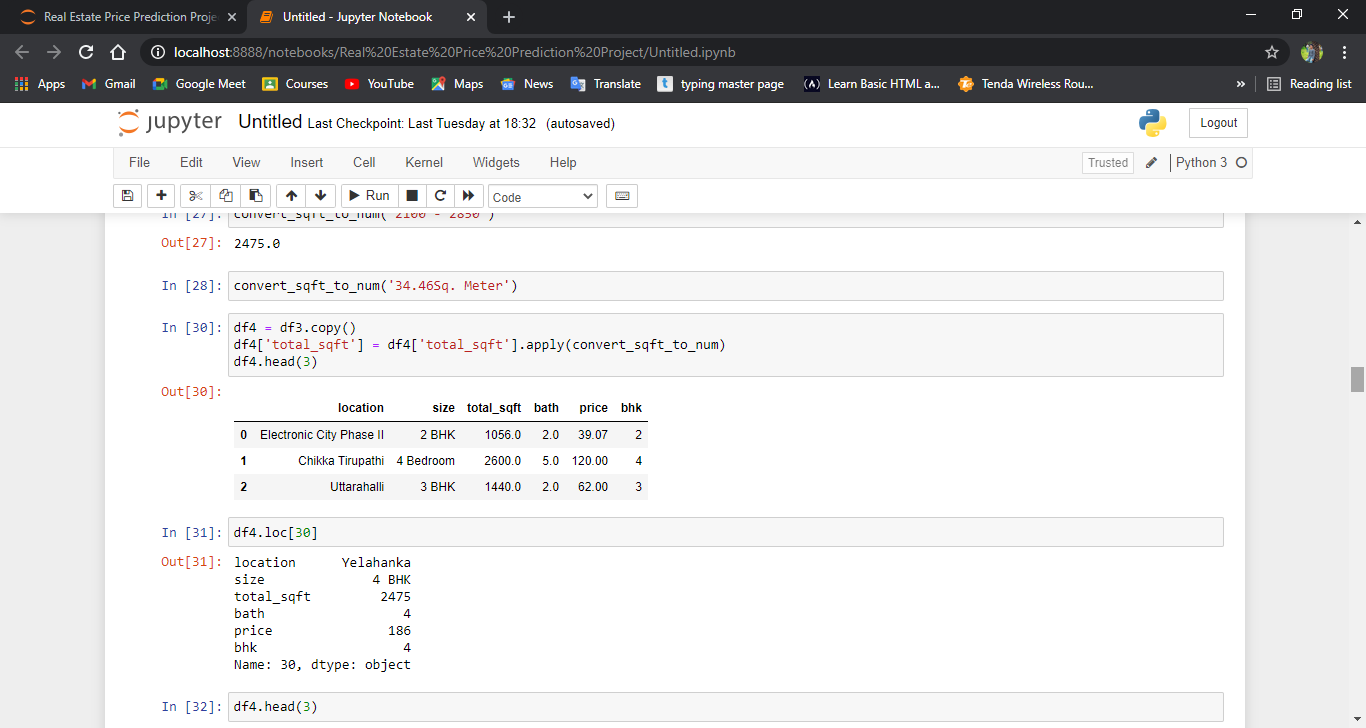
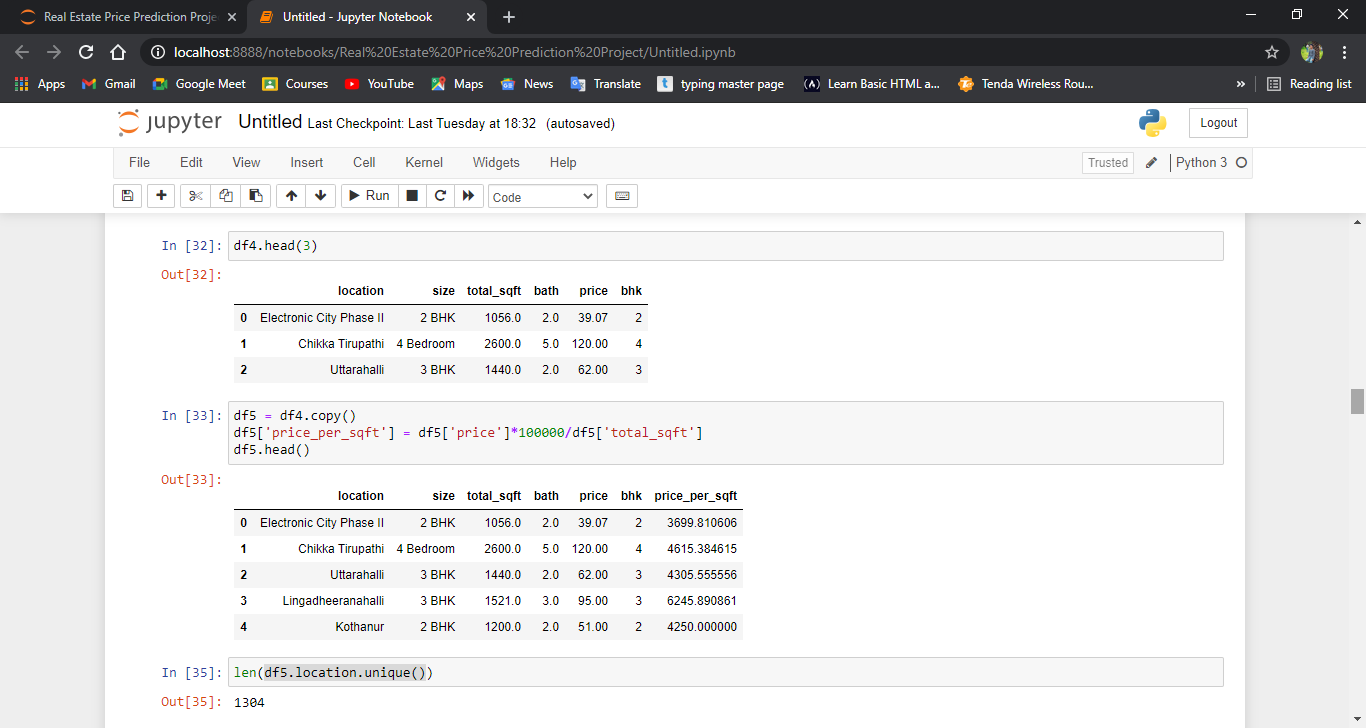


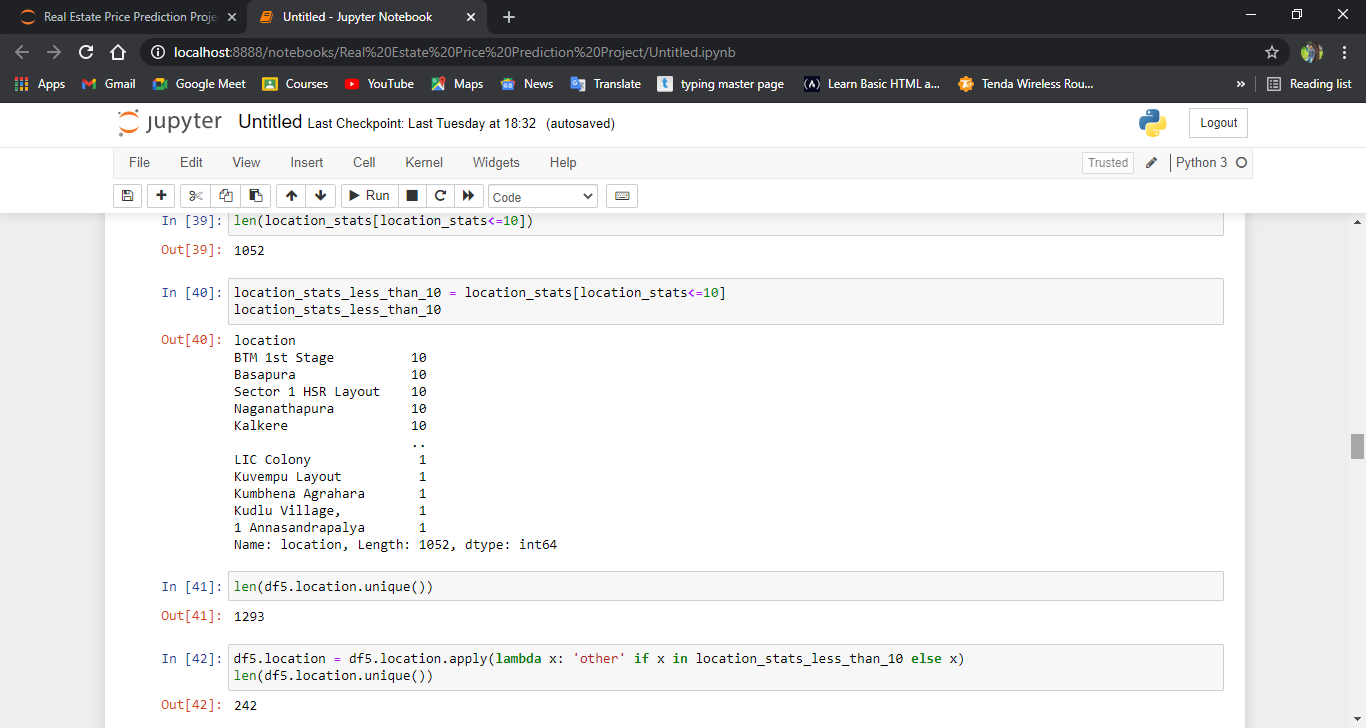


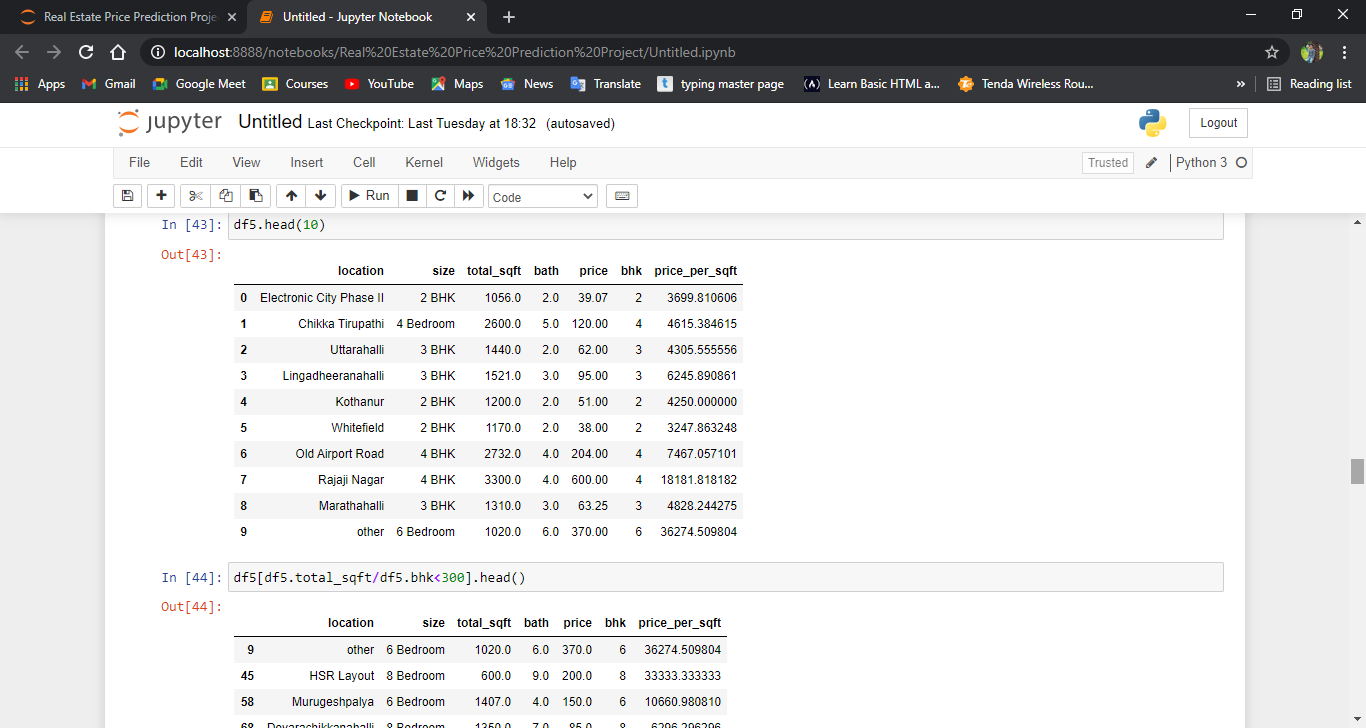
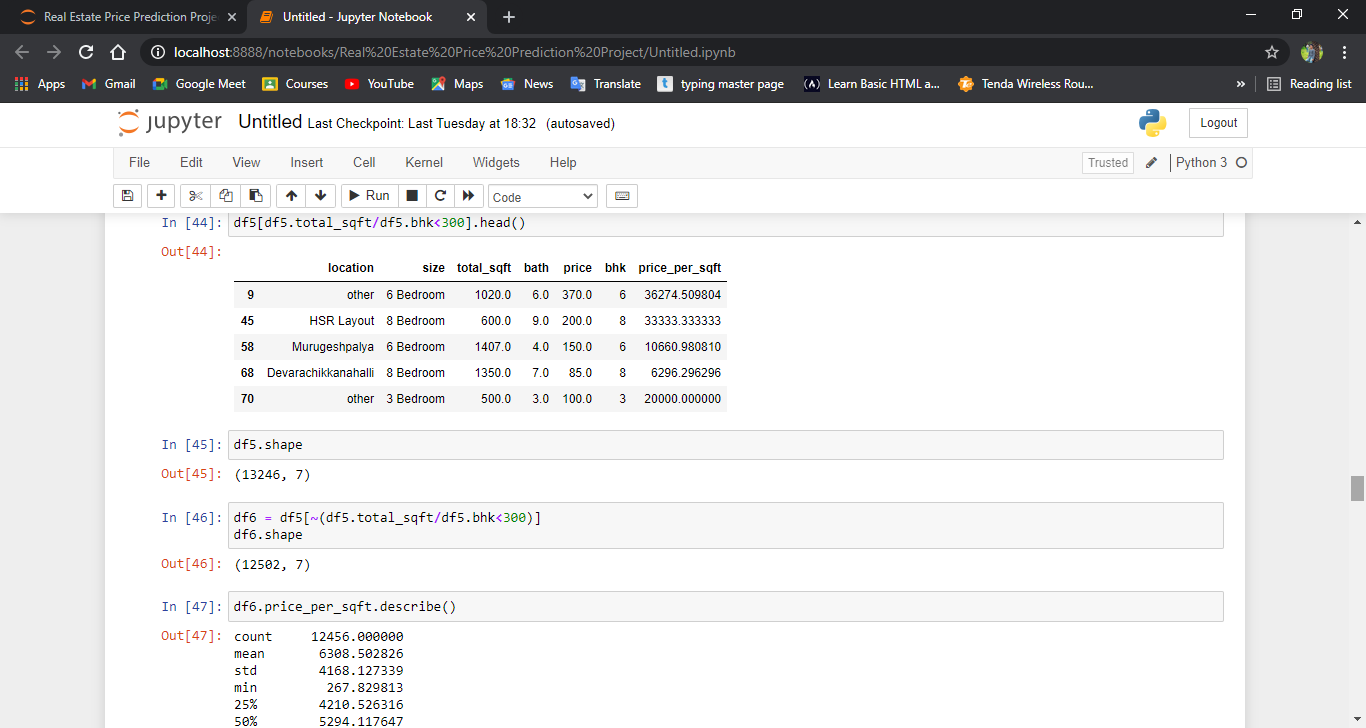
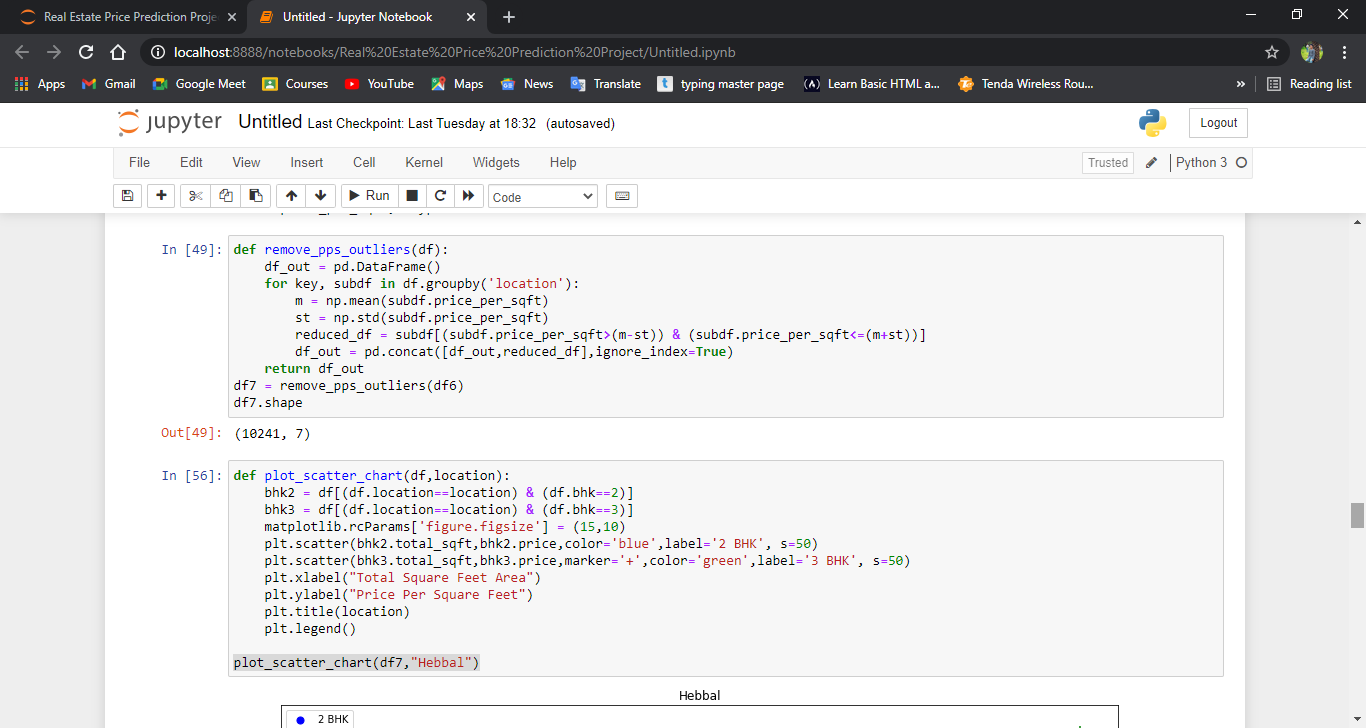
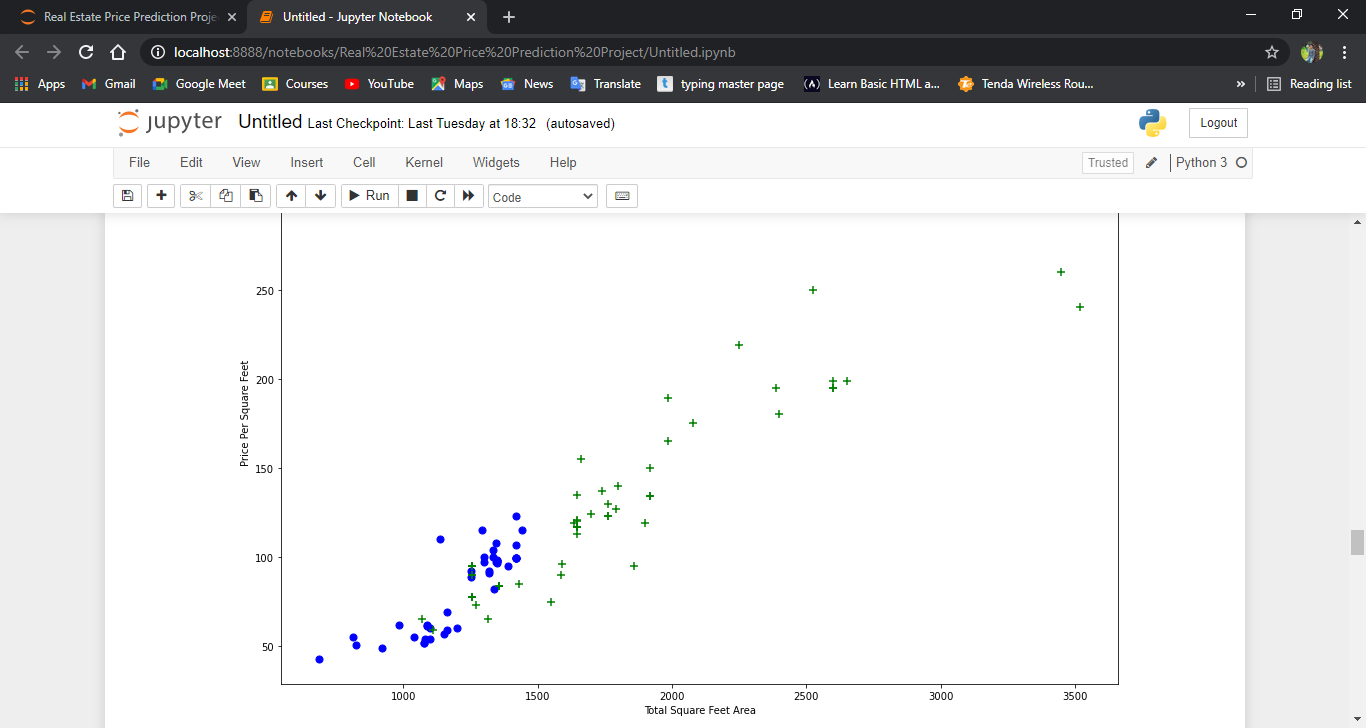
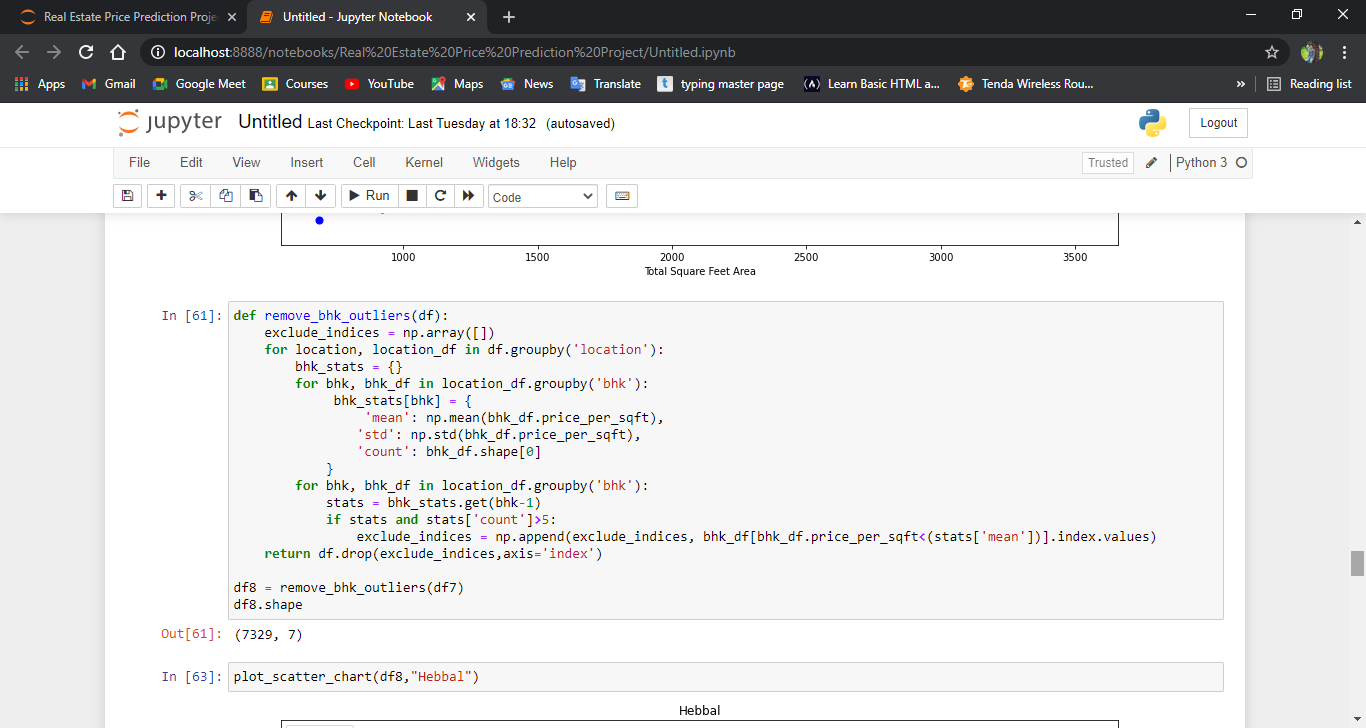
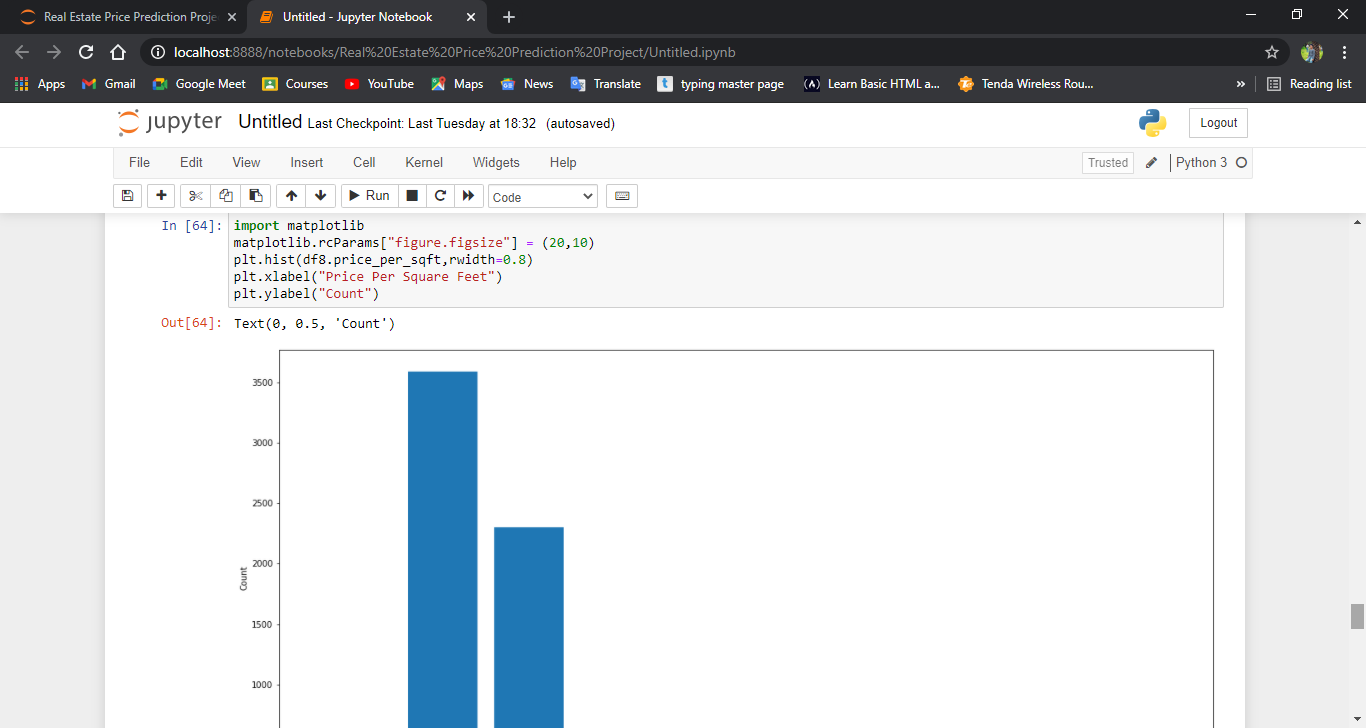
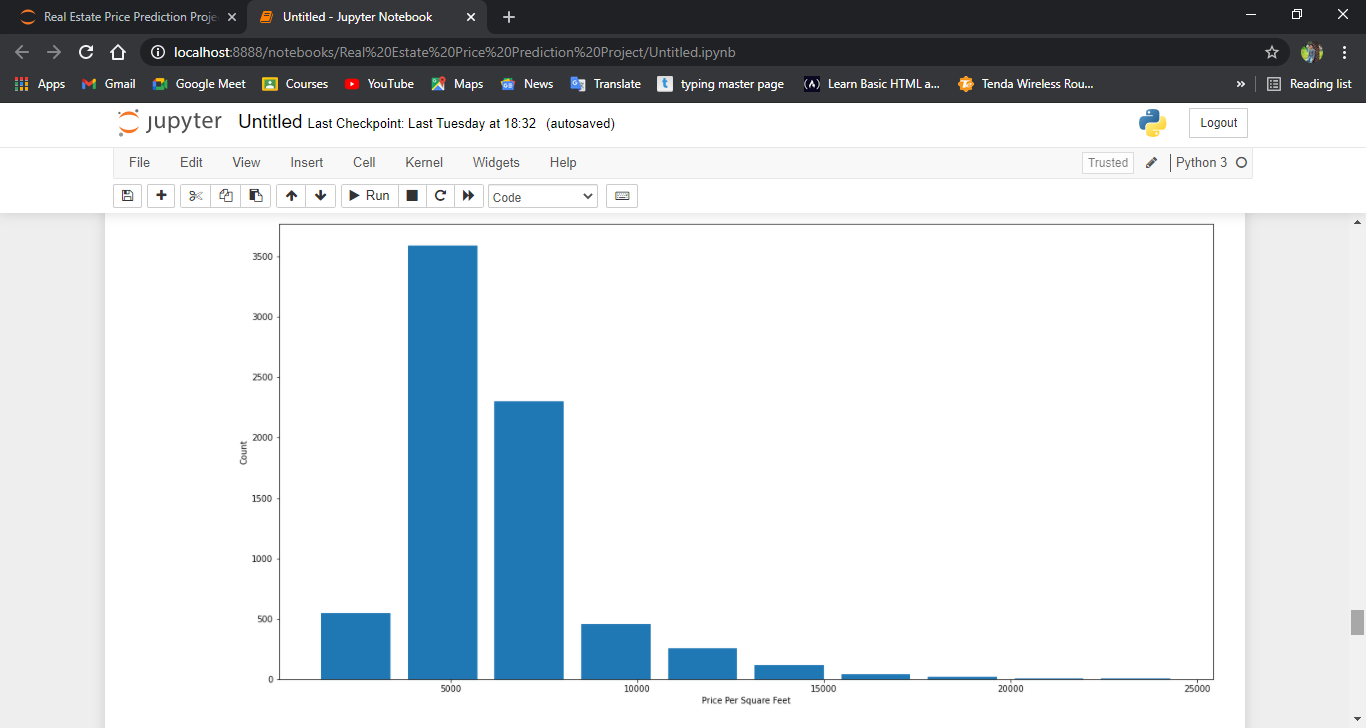
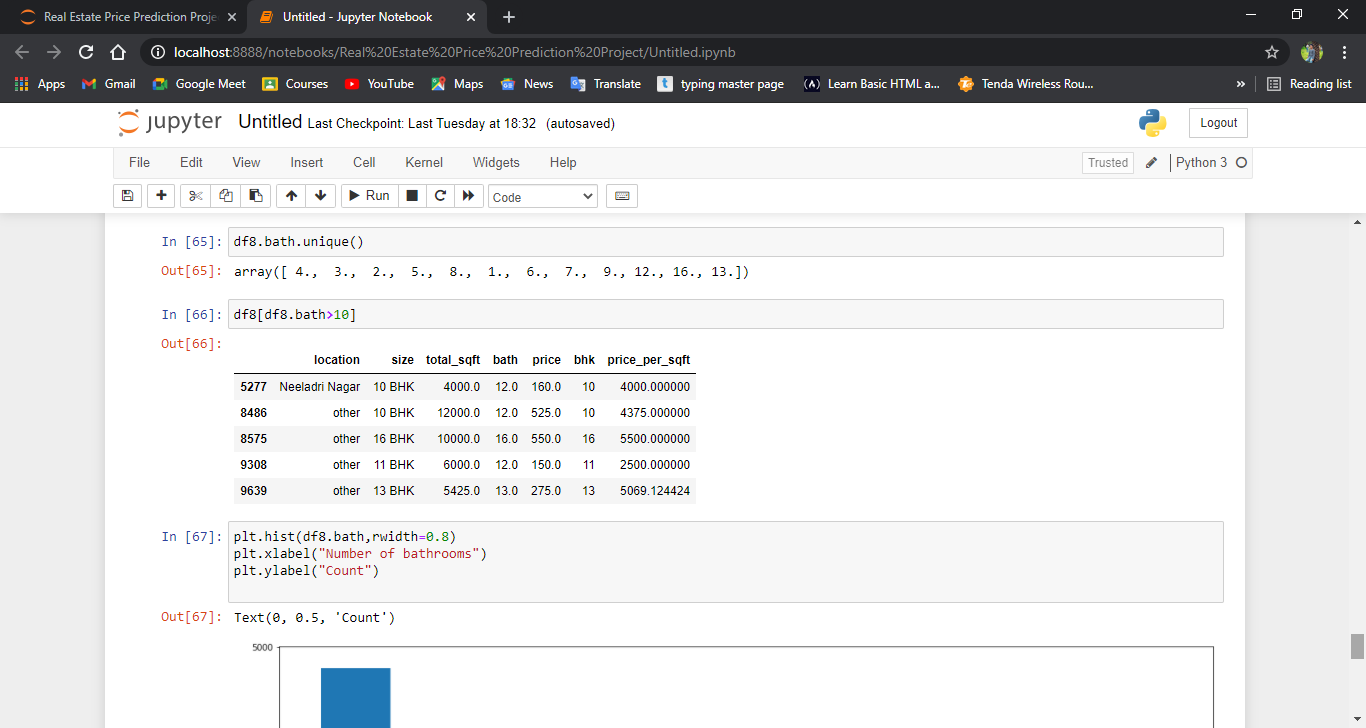
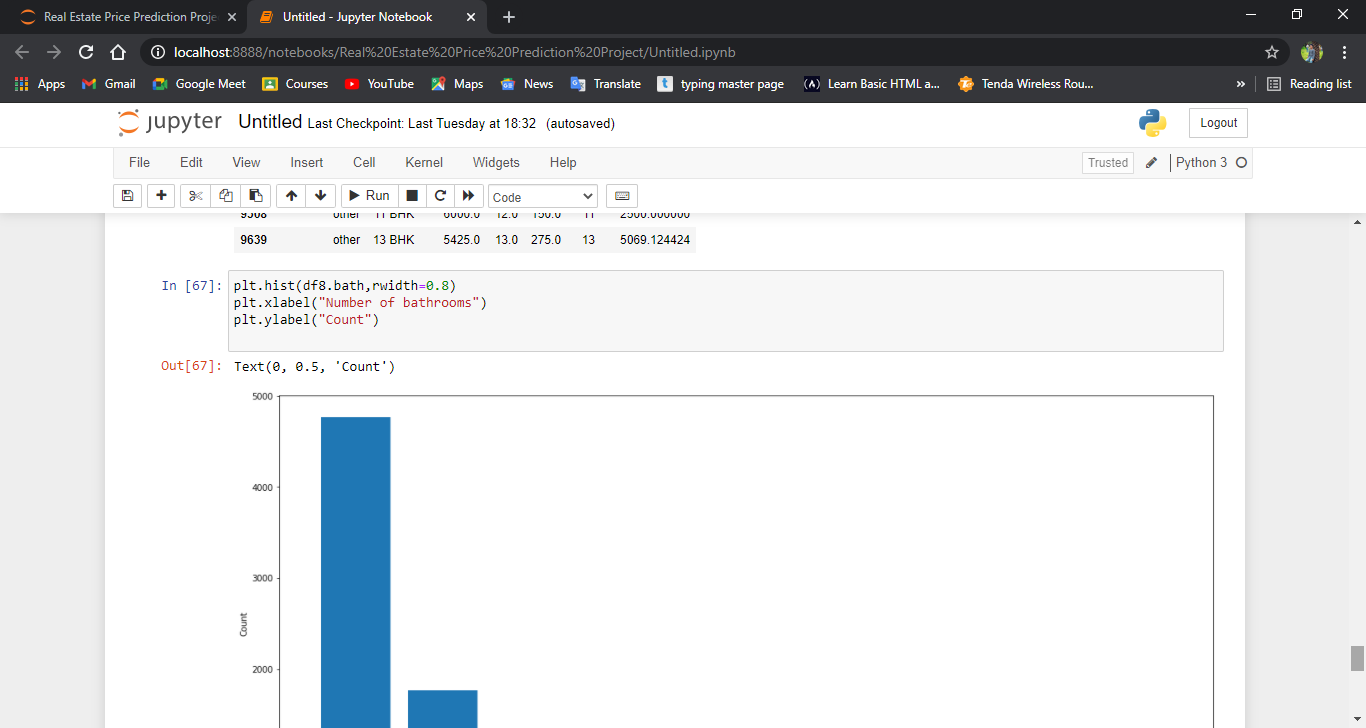
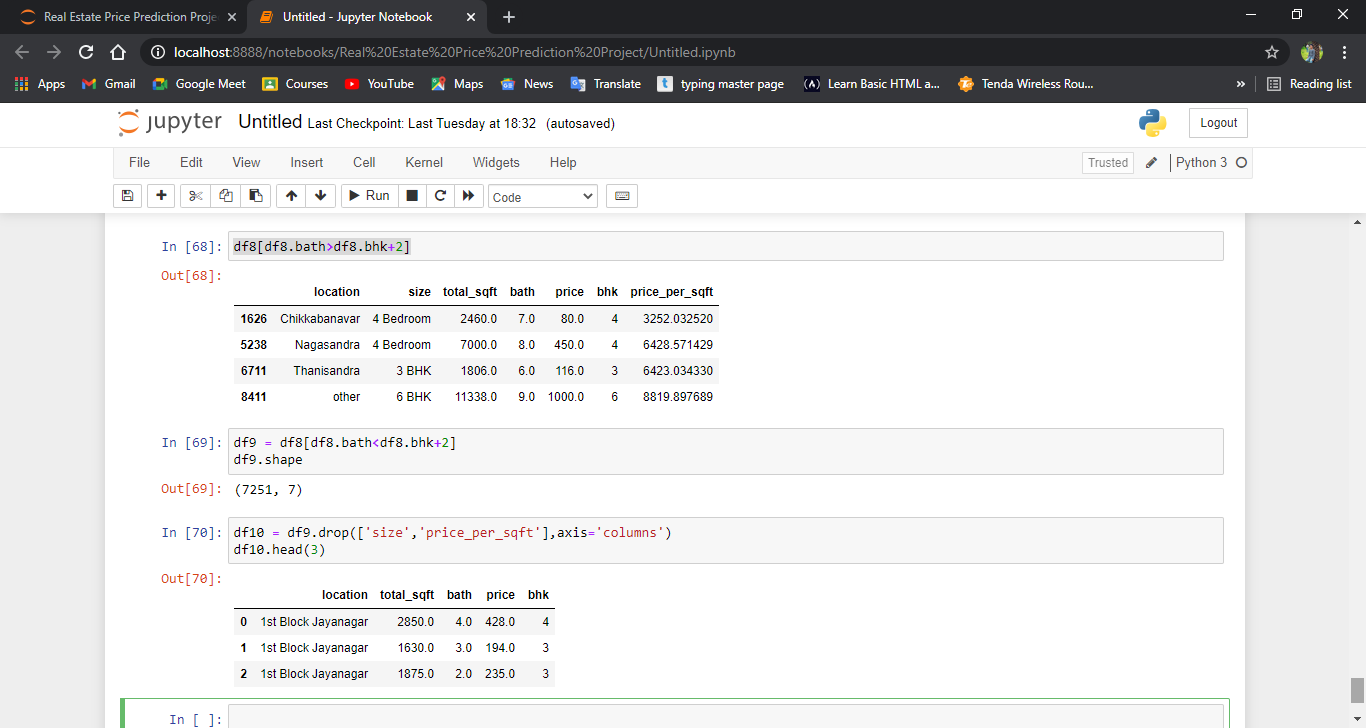


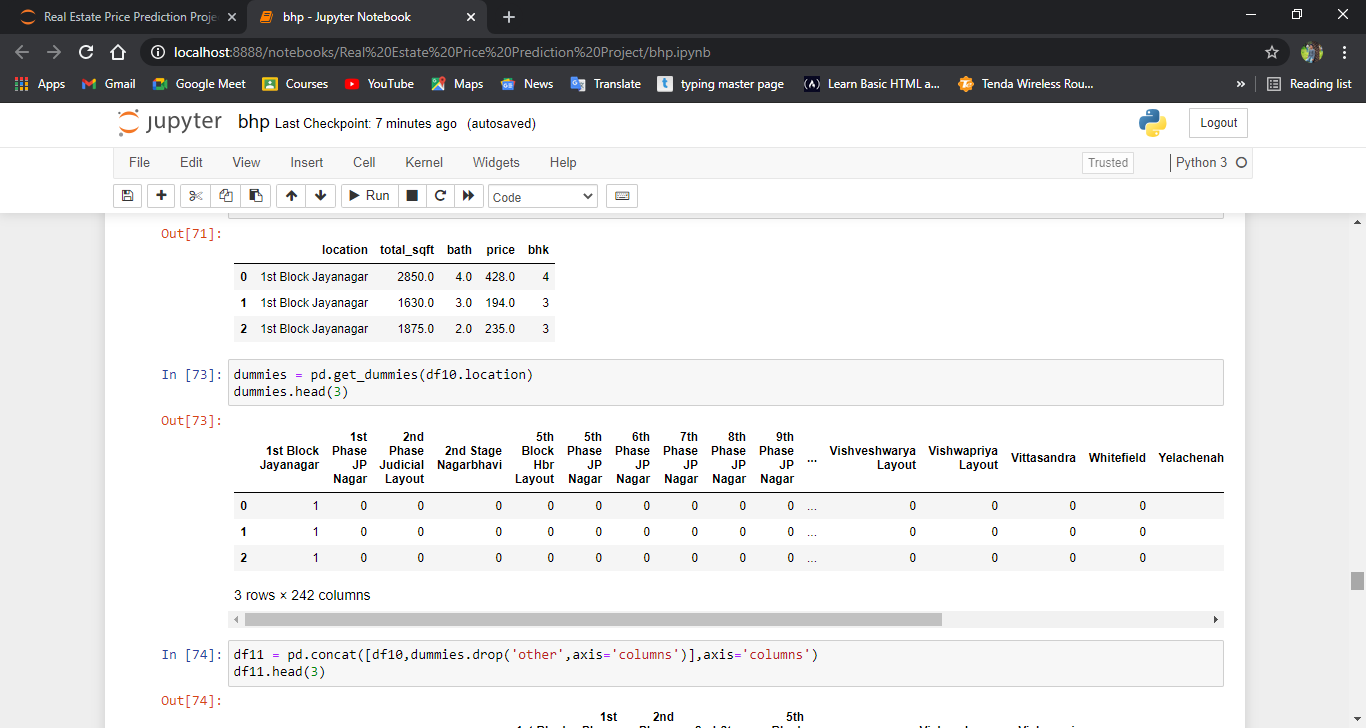


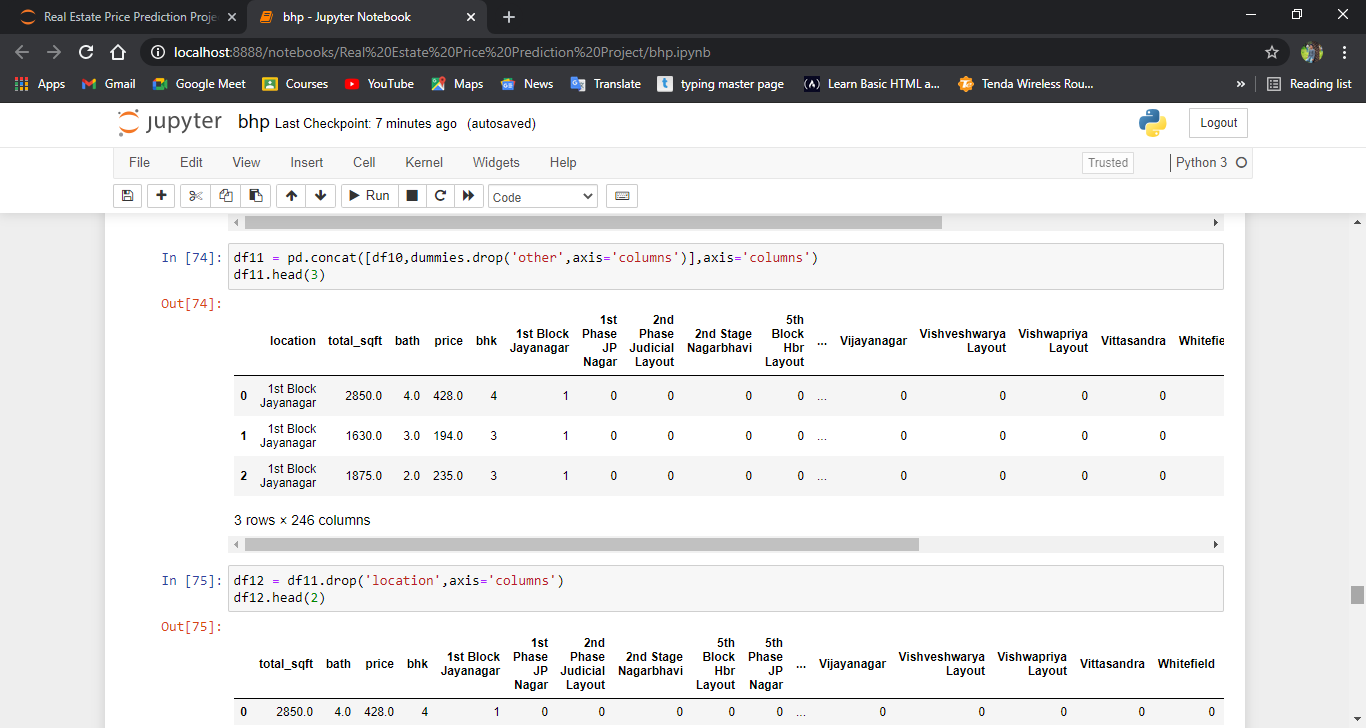
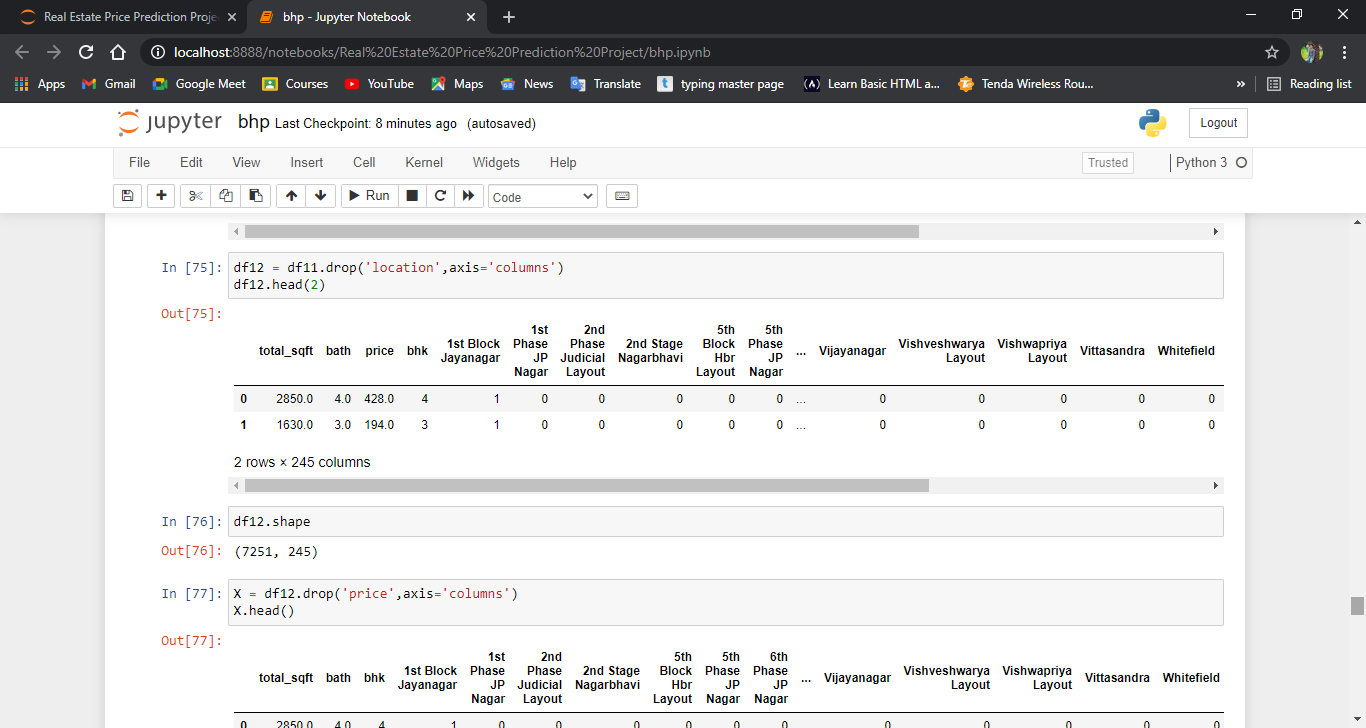


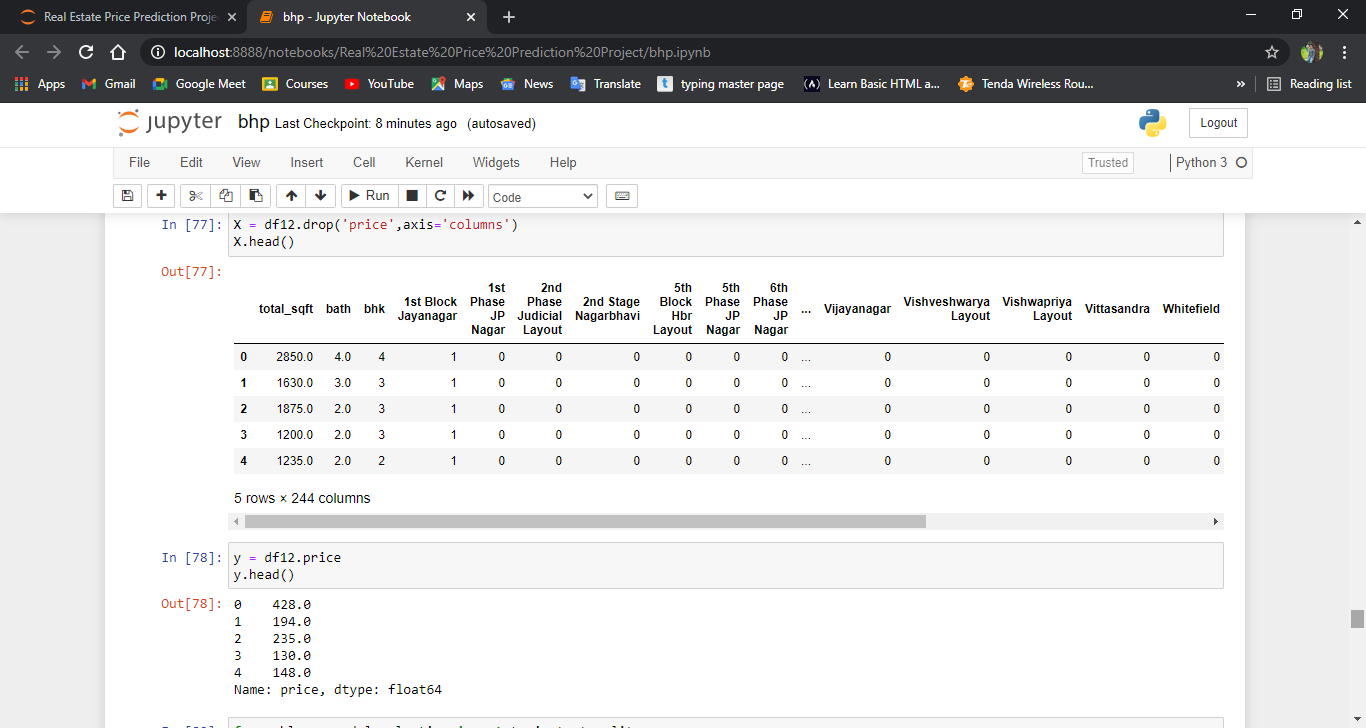
 

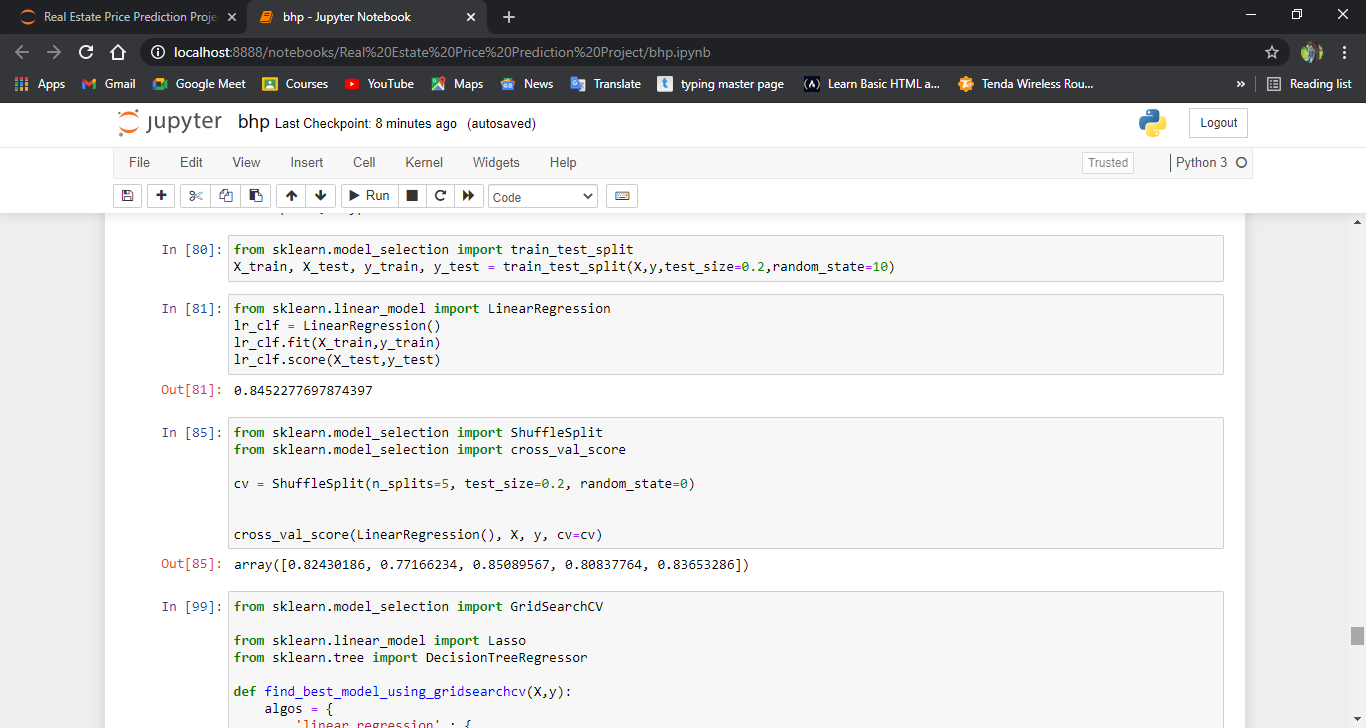
 

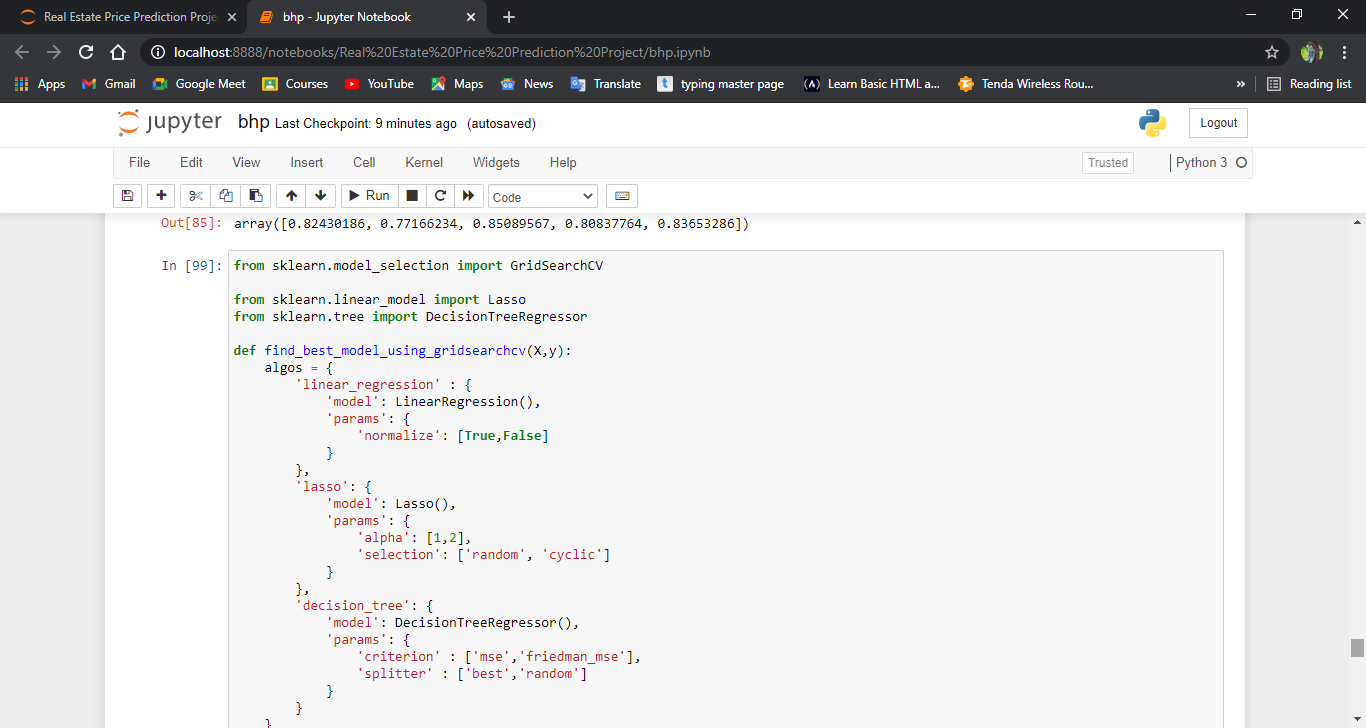
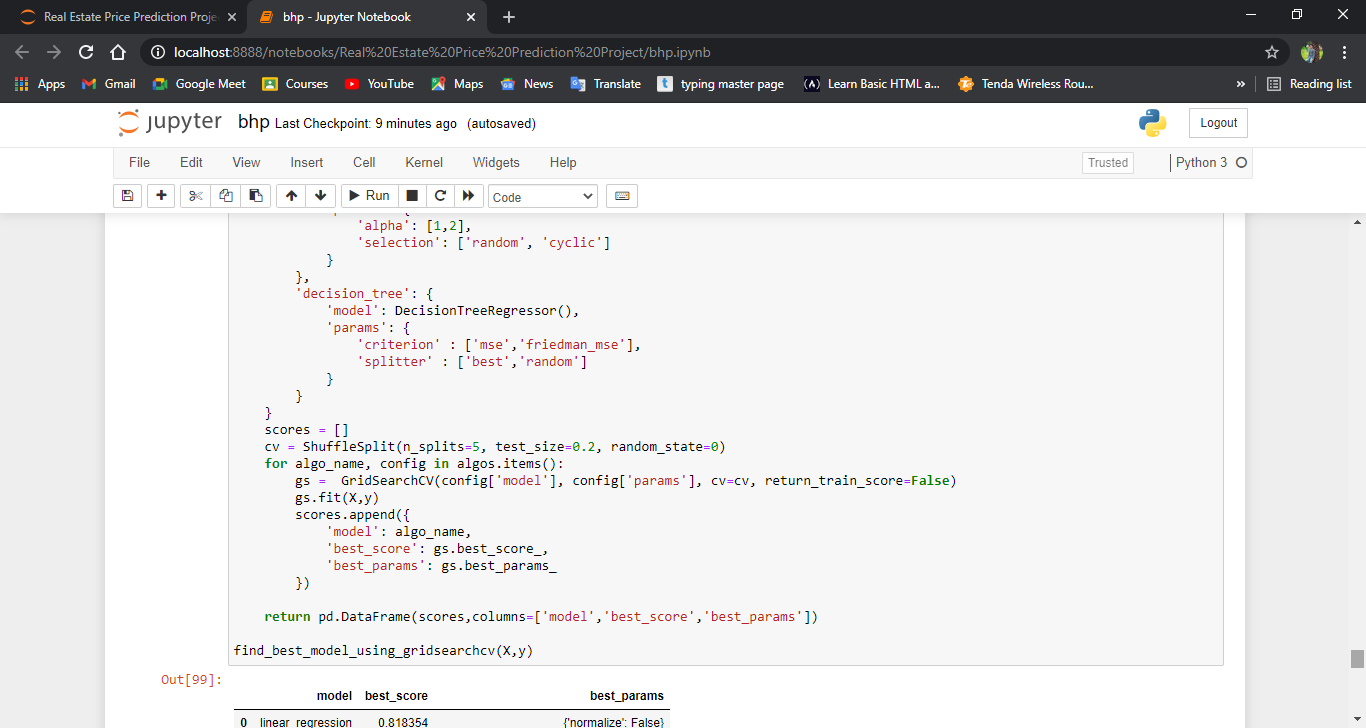
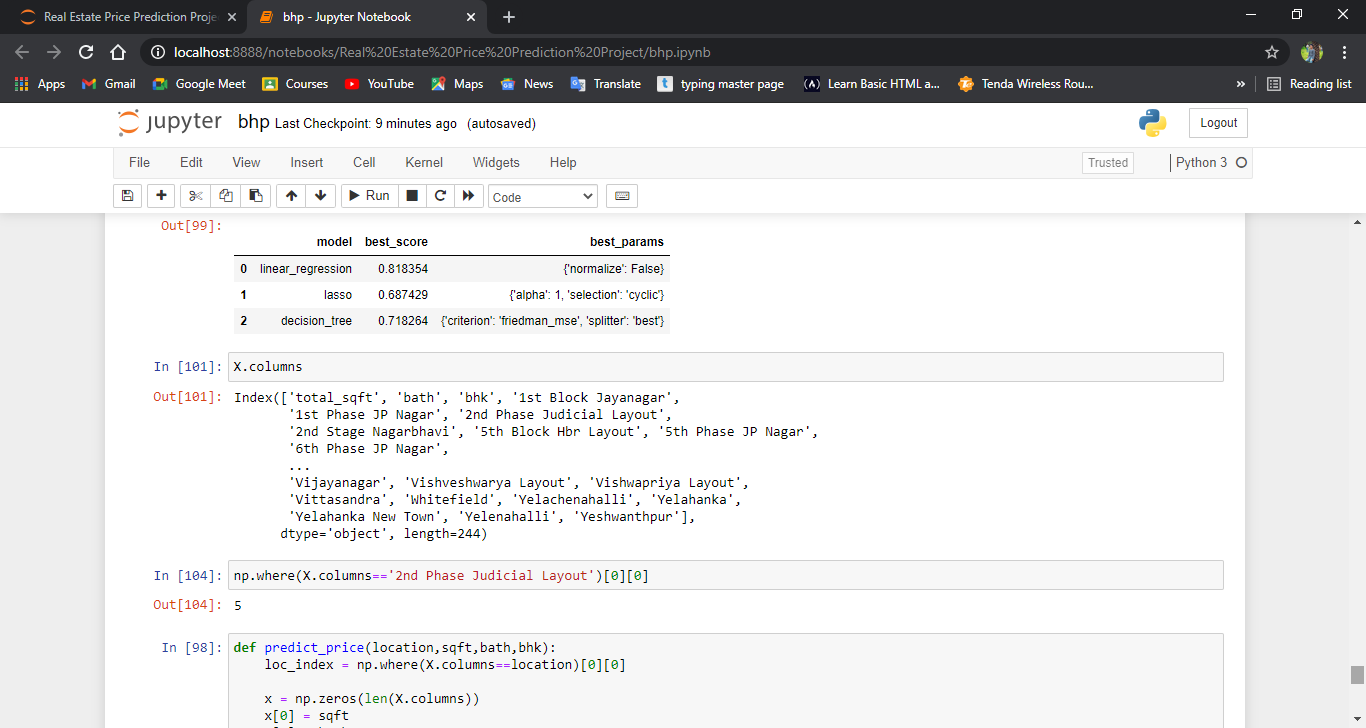
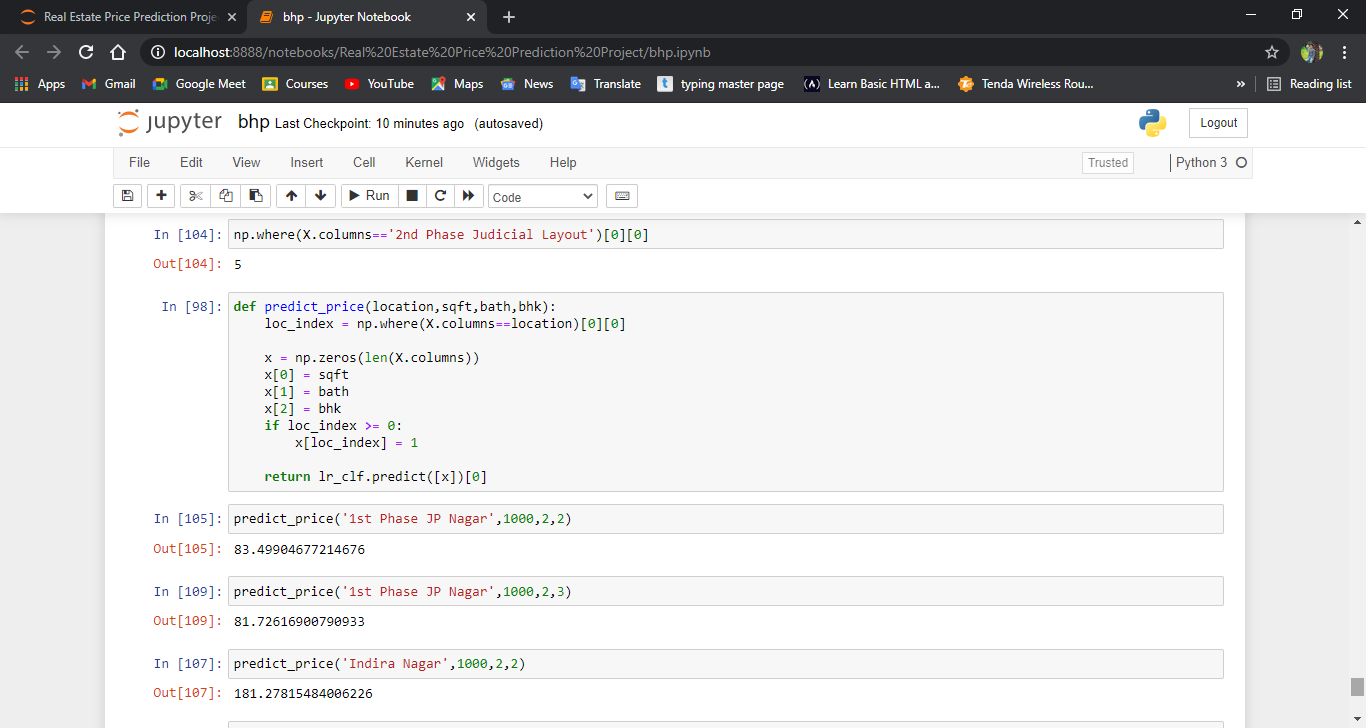
           

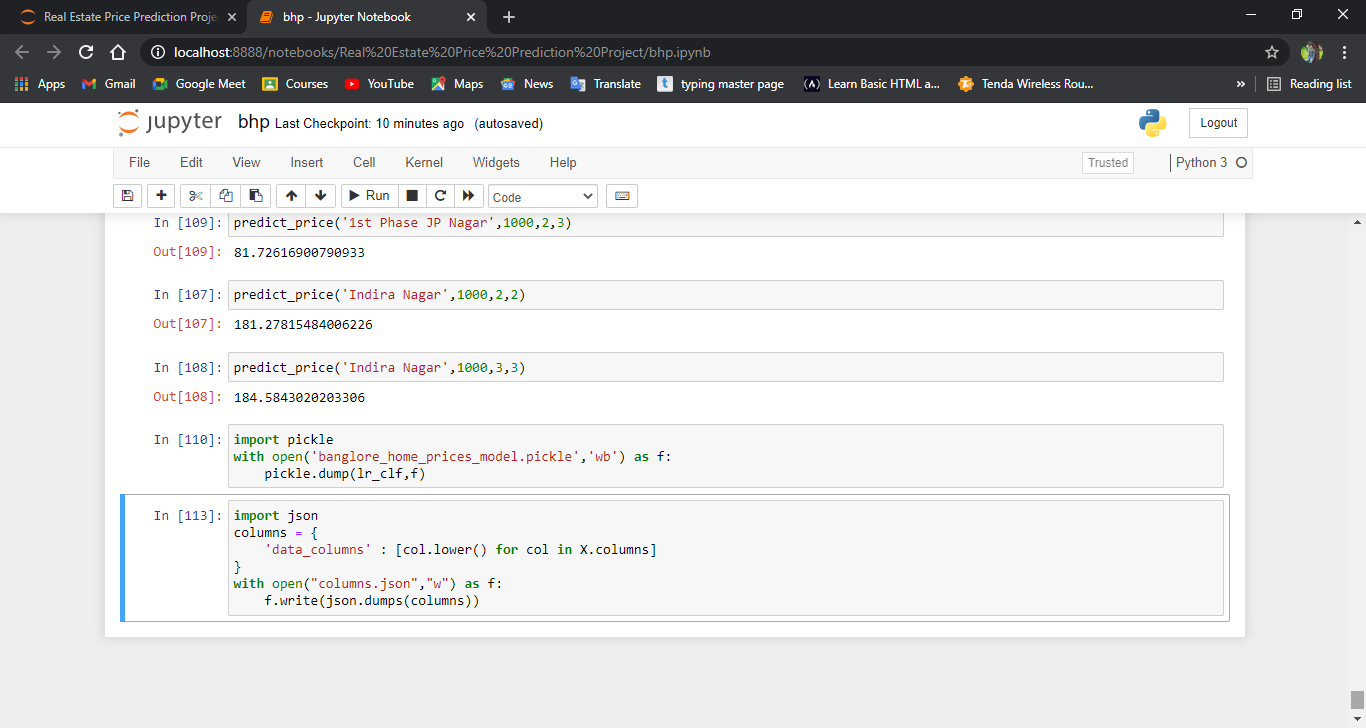




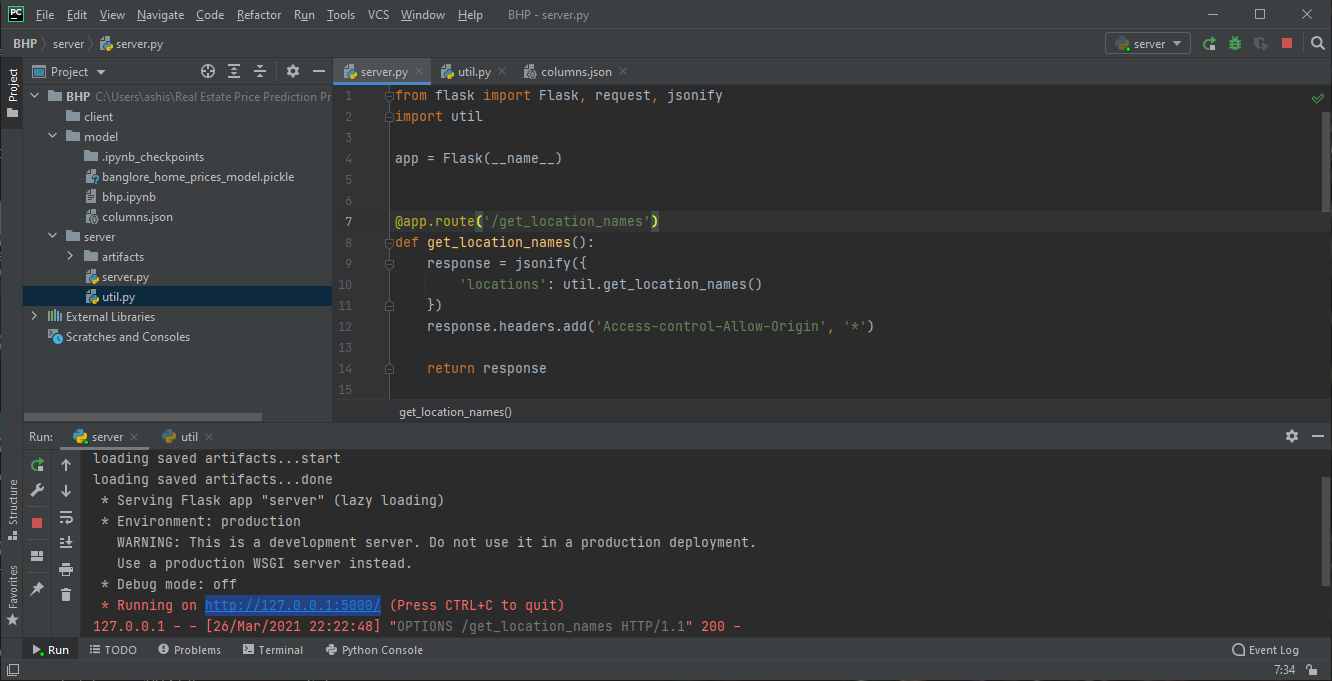


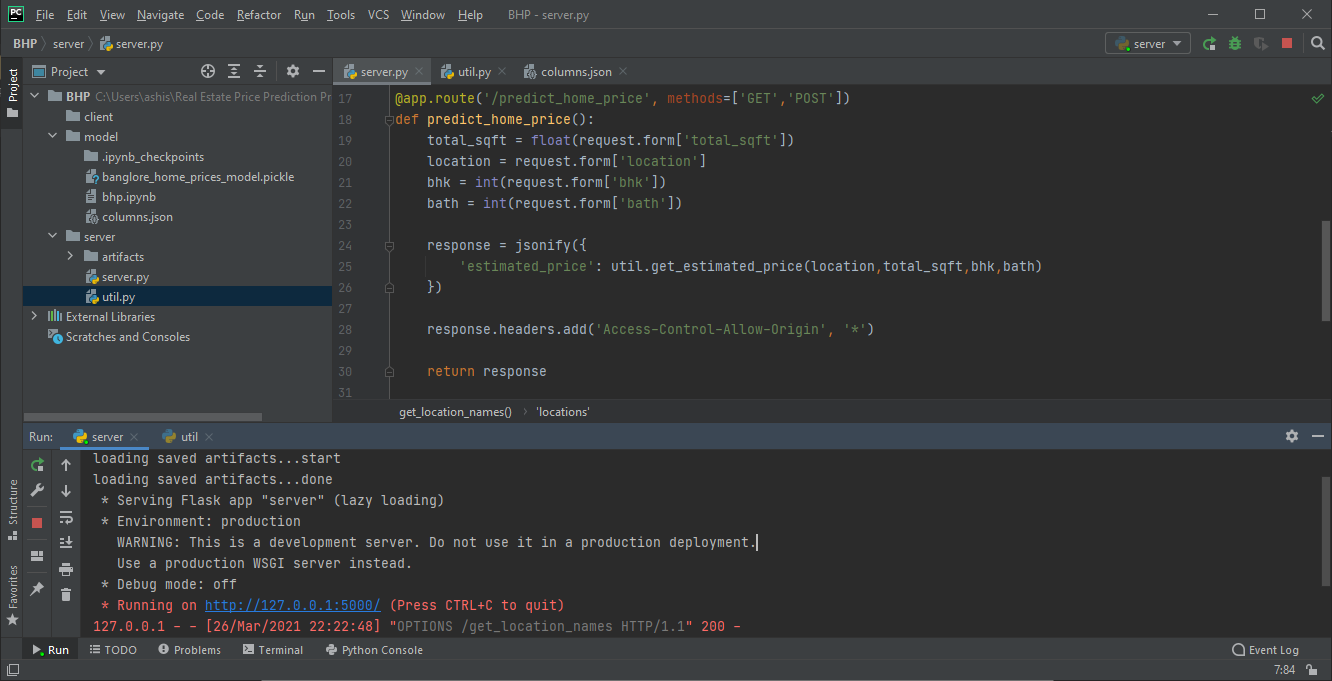
   



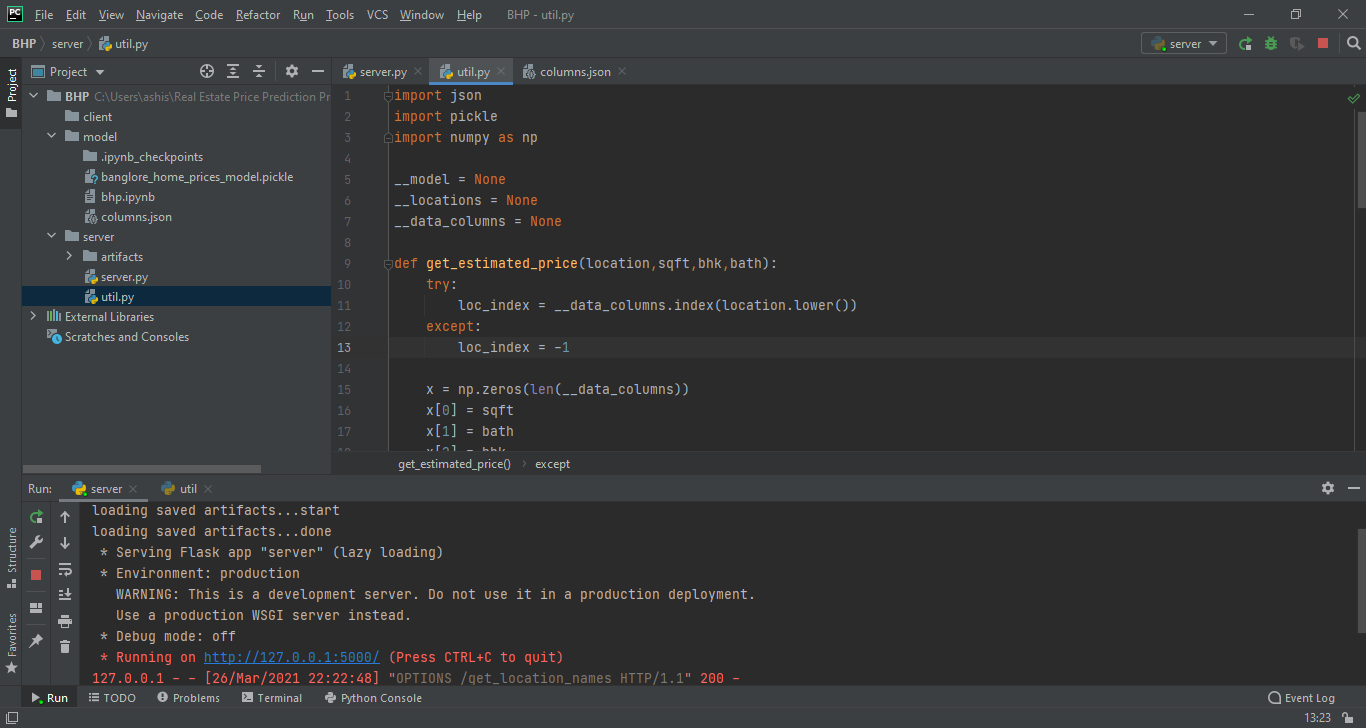
Python Flask Server:-

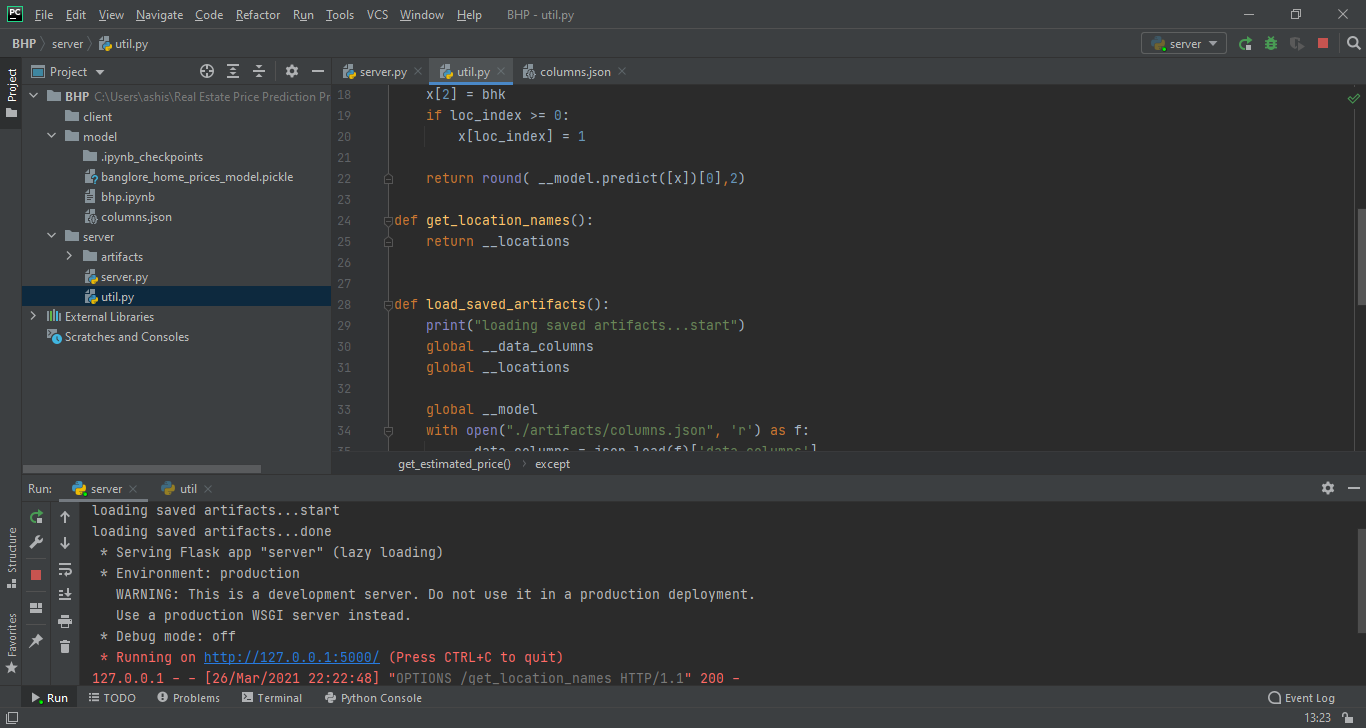
Server.py

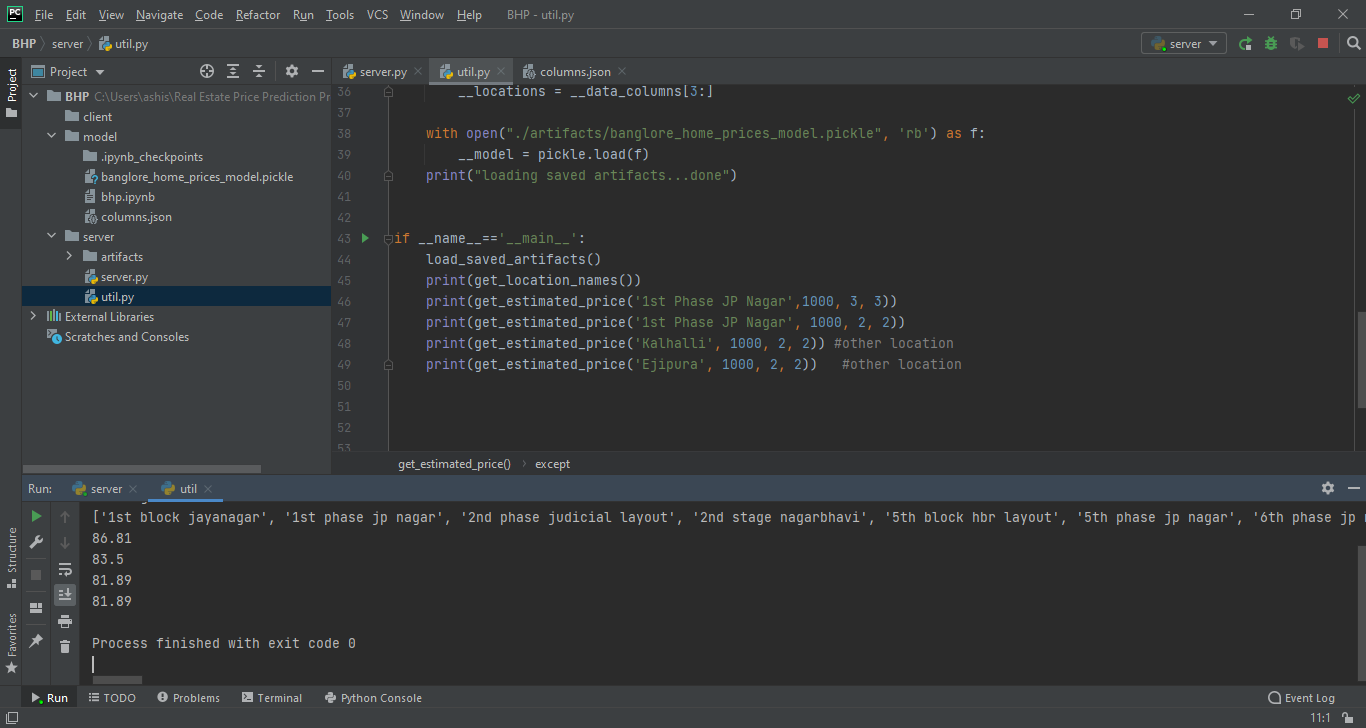




util.py







Python Flask Server Code:-

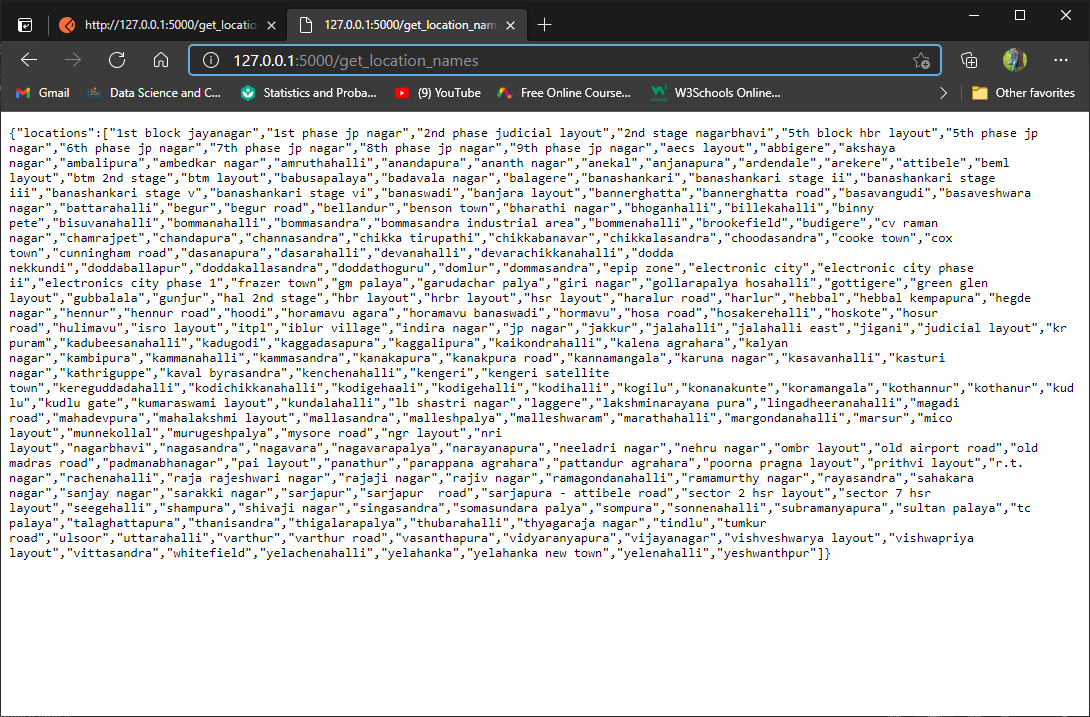
Server.py code:-

from flask import Flask, request, jsonify  
import util  
  
app = Flask(\_\_name\_\_)  
  
@app.route('/get\_location\_names', methods=['GET'])  
def get\_location\_names():  
 response = jsonify({  
 'locations': util.get\_location\_names()  
 })  
 response.headers.add('Access-Control-Allow-Origin', '\*')  
  
 return response  
  
@app.route('/predict\_home\_price', methods=['GET', 'POST'])  
def predict\_home\_price():  
 total\_sqft = float(request.form['total\_sqft'])  
 location = request.form['location']  
 bhk = int(request.form['bhk'])  
 bath = int(request.form['bath'])  
  
 response = jsonify({  
 'estimated\_price': util.get\_estimated\_price(location,total\_sqft,bhk,bath)  
 })  
 response.headers.add('Access-Control-Allow-Origin', '\*')  
  
 return response  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 print("Starting Python Flask Server For Home Price Prediction...")  
 util.load\_saved\_artifacts()  
 app.run()

util.py :-

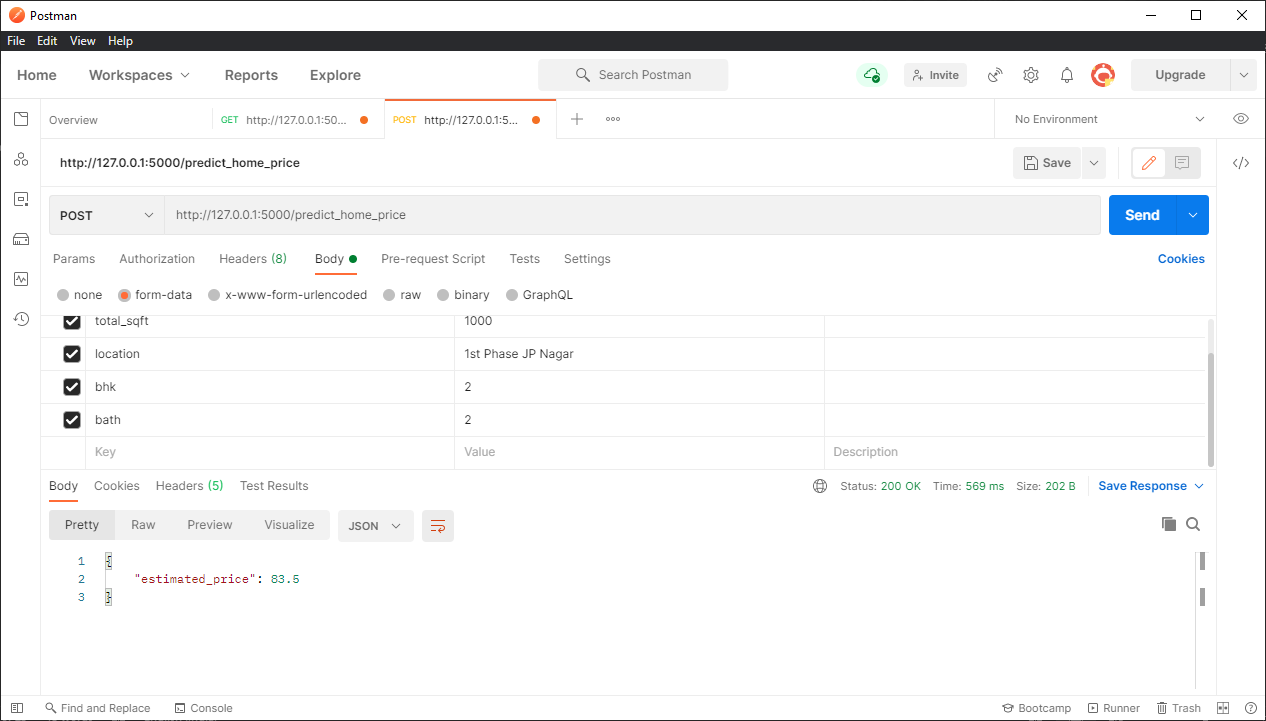
import pickle  
import json  
import numpy as np  
  
\_\_locations = None  
\_\_data\_columns = None  
\_\_model = None  
  
def get\_estimated\_price(location,sqft,bhk,bath):  
 try:  
 loc\_index = \_\_data\_columns.index(location.lower())  
 except:  
 loc\_index = -1  
  
 x = np.zeros(len(\_\_data\_columns))  
 x[0] = sqft  
 x[1] = bath  
 x[2] = bhk  
 if loc\_index>=0:  
 x[loc\_index] = 1  
  
 return round(\_\_model.predict([x])[0],2)  
  
  
def load\_saved\_artifacts():  
 print("loading saved artifacts...start")  
 global \_\_data\_columns  
 global \_\_locations  
  
 with open("./artifacts/columns.json", "r") as f:  
 \_\_data\_columns = json.load(f)['data\_columns']  
 \_\_locations = \_\_data\_columns[3:] # first 3 columns are sqft, bath, bhk  
  
 global \_\_model  
 if \_\_model is None:  
 with open('./artifacts/banglore\_home\_prices\_model.pickle', 'rb') as f:  
 \_\_model = pickle.load(f)  
 print("loading saved artifacts...done")  
  
def get\_location\_names():  
 return \_\_locations  
  
def get\_data\_columns():  
 return \_\_data\_columns  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 load\_saved\_artifacts()  
 print(get\_location\_names())  
 print(get\_estimated\_price('1st Phase JP Nagar',1000, 3, 3))  
 print(get\_estimated\_price('1st Phase JP Nagar', 1000, 2, 2))  
 print(get\_estimated\_price('Kalhalli', 1000, 2, 2)) # other location  
 print(get\_estimated\_price('Ejipura', 1000, 2, 2)) # other location

Server.py output:-



Postman Application Tool :-





Visual Studio Code:-

Code:-

App.html :-

<!DOCTYPE html>

<html>

<head>

    <title>Banglore Home Price Prediction</title>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

    <script src="app.js"></script>

    <link rel="stylesheet" href="app.css">

</head>

<body>

<div class="img"></div>

<form class="form">

    <h2>Area (Square Feet)</h2>

    <input class="area"  type="text" id="uiSqft" class="floatLabel" name="Squareft" value="1000">

    <h2>BHK</h2>

    <div class="switch-field">

        <input type="radio" id="radio-bhk-1" name="uiBHK" value="1"/>

        <label for="radio-bhk-1">1</label>

        <input type="radio" id="radio-bhk-2" name="uiBHK" value="2" checked/>

        <label for="radio-bhk-2">2</label>

        <input type="radio" id="radio-bhk-3" name="uiBHK" value="3"/>

        <label for="radio-bhk-3">3</label>

        <input type="radio" id="radio-bhk-4" name="uiBHK" value="4"/>

        <label for="radio-bhk-4">4</label>

        <input type="radio" id="radio-bhk-5" name="uiBHK" value="5"/>

        <label for="radio-bhk-5">5</label>

    </div>

    </form>

<form class="form">

    <h2>Bath</h2>

    <div class="switch-field">

        <input type="radio" id="radio-bath-1" name="uiBathrooms" value="1"/>

        <label for="radio-bath-1">1</label>

        <input type="radio" id="radio-bath-2" name="uiBathrooms" value="2" checked/>

        <label for="radio-bath-2">2</label>

        <input type="radio" id="radio-bath-3" name="uiBathrooms" value="3"/>

        <label for="radio-bath-3">3</label>

        <input type="radio" id="radio-bath-4" name="uiBathrooms" value="4"/>

        <label for="radio-bath-4">4</label>

        <input type="radio" id="radio-bath-5" name="uiBathrooms" value="5"/>

        <label for="radio-bath-5">5</label>

    </div>

        <h2>Location</h2>

    <div>

  <select class="location" name="" id="uiLocations">

    <option value="" disabled="disabled" selected="selected">Choose a Location</option>

        <option>Electronic City</option>

        <option>Rajaji Nagar</option>

  </select>

</div>

    <button class="submit" onclick="onClickedEstimatePrice()" type="button">Estimate Price</button>

    <div id="uiEstimatedPrice" class="result">  <h2></h2> </div>

</body>

</html>

app.css :-

@import url(https://fonts.googleapis.com/css?family=Roboto:300);

.switch-field {

    display: flex;

    margin-bottom: 36px;

    overflow: hidden;

}

.switch-field input {

    position: absolute !important;

    clip: rect(0, 0, 0, 0);

    height: 1px;

    width: 1px;

    border: 0;

    overflow: hidden;

}

.switch-field label {

    background-color: #e4e4e4;

    color: rgba(0, 0, 0, 0.6);

    font-size: 14px;

    line-height: 1;

    text-align: center;

    padding: 8px 16px;

    margin-right: -1px;

    border: 1px solid rgba(0, 0, 0, 0.2);

    box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.3), 0 1px rgba(255, 255, 255, 0.1);

    transition: all 0.1s ease-in-out;

}

.switch-field label:hover {

    cursor: pointer;

}

.switch-field input:checked + label {

    background-color: #a5dc86;

    box-shadow: none;

}

.switch-field label:first-of-type {

    border-radius: 4px 0 0 4px;

}

.switch-field label:last-of-type {

    border-radius: 0 4px 4px 0;

}

.form {

    max-width: 270px;

    font-family: "Lucida Grande", Tahoma, Verdana, sans-serif;

    font-weight: normal;

    line-height: 1.625;

    margin: 8px auto;

    padding-left: 16px;

    z-index: 2;

}

h2 {

    font-size: 18px;

    margin-bottom: 8px;

}

.area{

  font-family: "Roboto", sans-serif;

  outline: 0;

  background: #f2f2f2;

  width: 76%;

  border: 0;

  margin: 0 0 10px;

  padding: 10px;

  box-sizing: border-box;

  font-size: 15px;

  height: 35px;

  border-radius: 5px;

}

.location{

  font-family: "Roboto", sans-serif;

  outline: 0;

  background: #f2f2f2;

  width: 76%;

  border: 0;

  margin: 0 0 10px;

  padding: 10px;

  box-sizing: border-box;

  font-size: 15px;

  height: 40px;

  border-radius: 5px;

}

.submit{

  background: #a5dc86;

  width: 76%;

  border: 0;

  margin: 25px 0 10px;

  box-sizing: border-box;

  font-size: 15px;

    height: 35px;

    text-align: center;

    border-radius: 5px;

}

.result{

        background: #dcd686;

        width: 76%;

        border: 0;

        margin: 25px 0 10px;

        box-sizing: border-box;

        font-size: 15px;

        height: 35px;

        text-align: center;

}

.img {

  background: url('https://images.unsplash.com/photo-1564013799919-ab600027ffc6?ixlib=rb-1.2.1&auto=format&fit=crop&w=1350&q=80');

    background-repeat: no-repeat;

  background-size: auto;

  background-size:100% 100%;

  -webkit-filter: blur(5px);

  -moz-filter: blur(5px);

  -o-filter: blur(5px);

  -ms-filter: blur(5px);

  filter: blur(15px);

  position: fixed;

  width: 100%;

  height: 100%;

  top: 0;

  left: 0;

  z-index: -1;

}

body, html {

  height: 100%;

}

App.js :-

function getBathValue() {

  var uiBathrooms = document.getElementsByName("uiBathrooms");

  for(var i in uiBathrooms) {

    if(uiBathrooms[i].checked) {

        return parseInt(i)+1;

    }

  }

  return -1; // Invalid Value

}

function getBHKValue() {

  var uiBHK = document.getElementsByName("uiBHK");

  for(var i in uiBHK) {

    if(uiBHK[i].checked) {

        return parseInt(i)+1;

    }

  }

  return -1; // Invalid Value

}

function onClickedEstimatePrice() {

  console.log("Estimate price button clicked");

  var sqft = document.getElementById("uiSqft");

  var bhk = getBHKValue();

  var bathrooms = getBathValue();

  var location = document.getElementById("uiLocations");

  var estPrice = document.getElementById("uiEstimatedPrice");

  var url = "http://127.0.0.1:5000/predict\_home\_price"; //Use this if you are NOT using nginx which is first 7 tutorials

   // Use this if  you are using nginx. i.e tutorial 8 and onwards

  $.post(url, {

      total\_sqft: parseFloat(sqft.value),

      bhk: bhk,

      bath: bathrooms,

      location: location.value

  },function(data, status) {

      console.log(data.estimated\_price);

      estPrice.innerHTML = "<h2>" + data.estimated\_price.toString() + " Lakh</h2>";

      console.log(status);

  });

}

function onPageLoad() {

  console.log( "document loaded" );

  var url = "http://127.0.0.1:5000/get\_location\_names"; // Use this if you are NOT using nginx which is first 7 tutorials

  //var url = "/api/get\_location\_names"; // Use this if  you are using nginx. i.e tutorial 8 and onwards

  $.get(url,function(data, status) {

      console.log("got response for get\_location\_names request");

      if(data) {

          var locations = data.locations;

          var uiLocations = document.getElementById("uiLocations");

          $('#uiLocations').empty();

          for(var i in locations) {

              var opt = new Option(locations[i]);

              $('#uiLocations').append(opt);

          }

      }

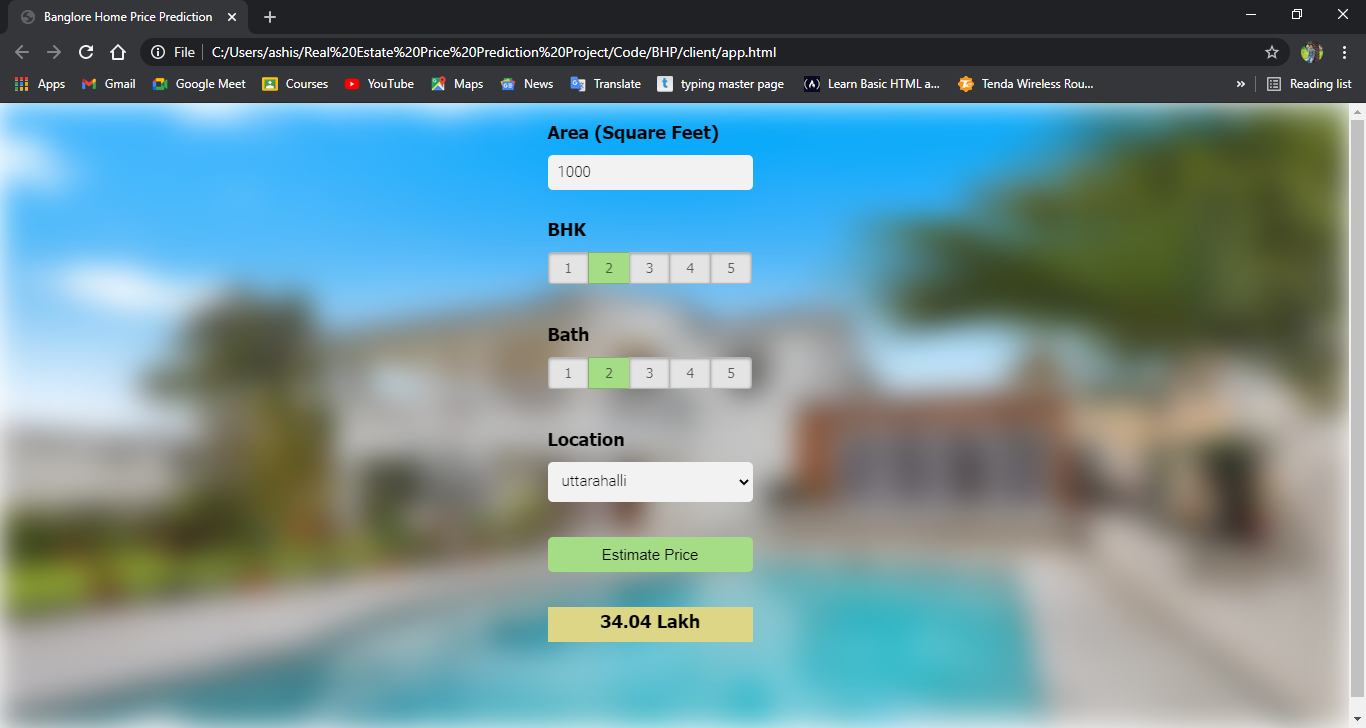
  });

}

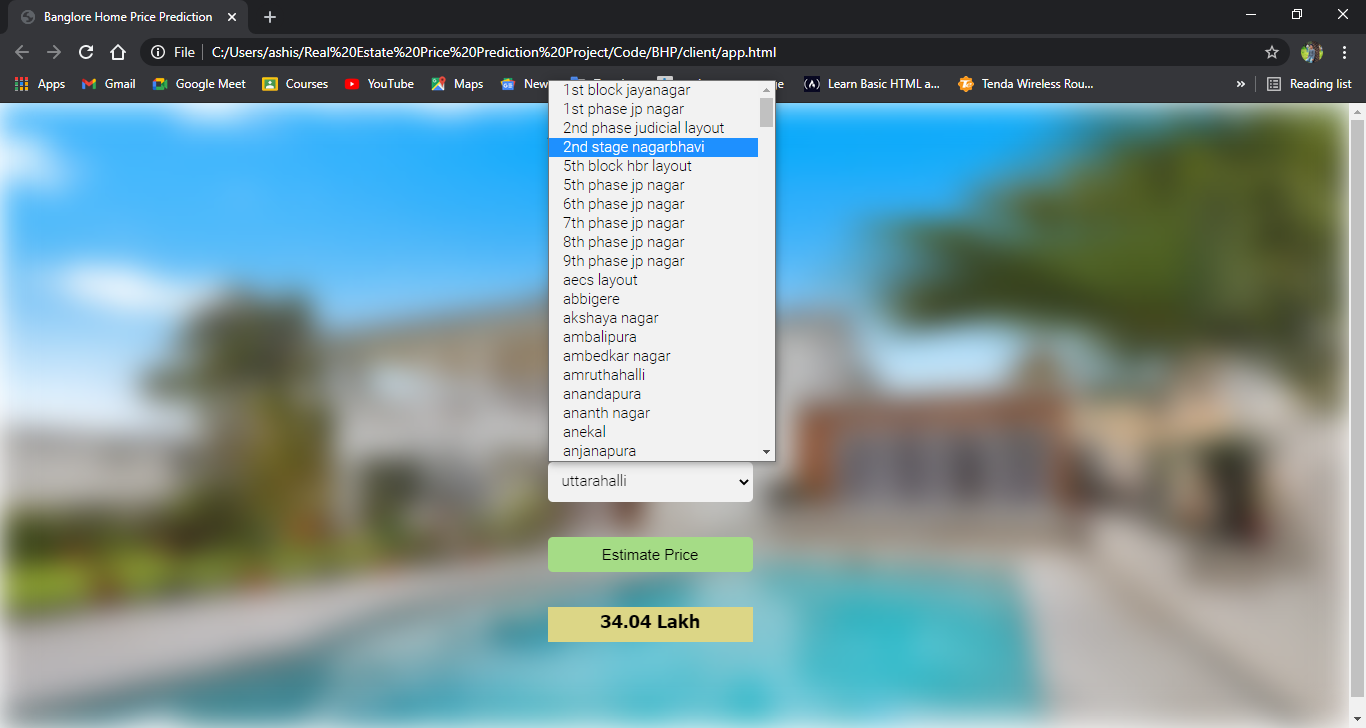
window.onload = onPageLoad;

Output screenshoot:-

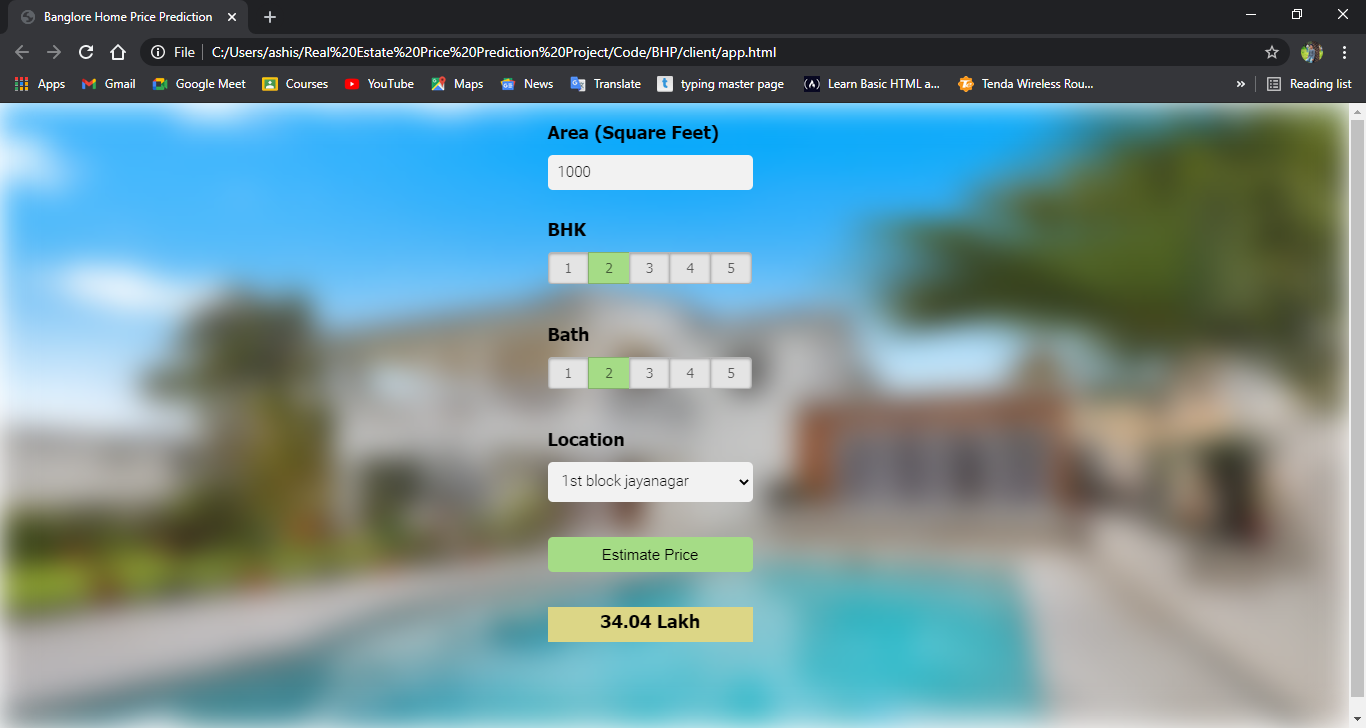
1)web Page:-



2) All Location scroll:-



3)Selection of sqft, bhk, bath:-



4)Price Prediction Depends upon bhk, bath,sqft, different location:-

