Online Auction System Using Django

Last Updated : 24 Sep, 2024

In this article, I am Going to Build an **Online auction system** Using Django. Physical auctions are limited by location but online auctions are accessible globally. It opens a broader range of buyers. It Can Connect the Seller, Buyer, and admin.

Tools & Technologies Used in this Project :

- Python
- Django
- Visual Studio Code
- HTML
- CSS
- Java Script

Implementation :

Follow the below steps to implement the discussed project :

Step 1: Download Visual Studio Code and install it in your Device.

Step 2: After Setup Visual Studio Code.Press (Cntrl + Shift + X) and Install Python & Django Extension in it.

Step 3: Open Command Promt and Create a Directory.

```
>Mkdir Online_Auction
>cd Online_Auction
>code .
```

Step 4: Open shell of VSCode and create and activate virtual environment using below commands.

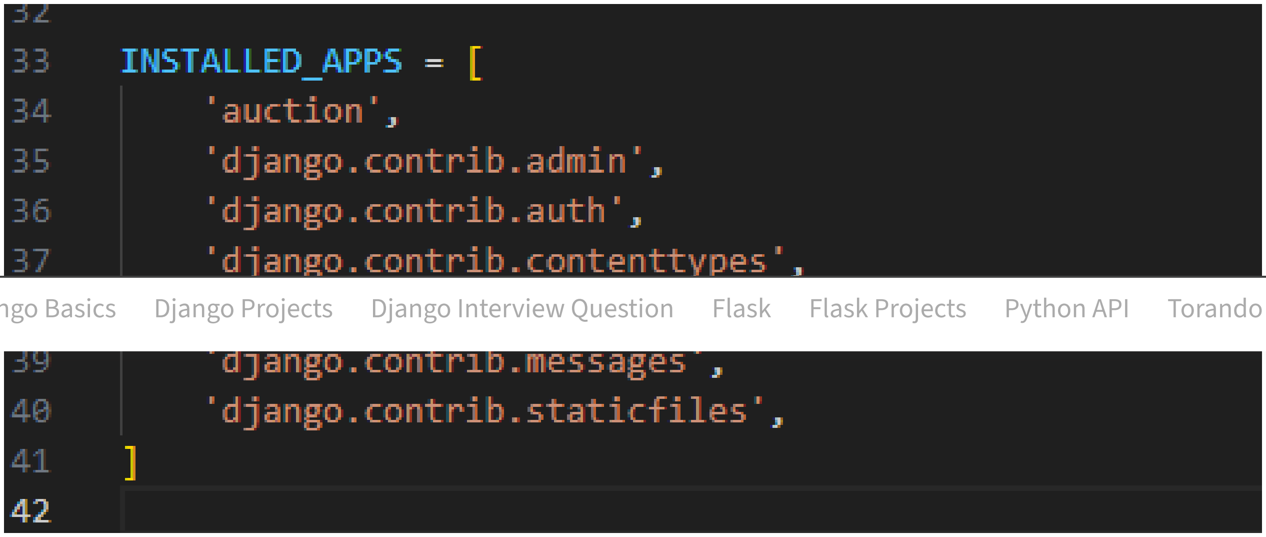
```
virtualenv venv
venv/Scripts/activate
```

```
pip install django
```

Step 4: Create the Project and app using the below commands.

```
django-admin startproject Online_Auction
cd Online_Auction
python manage.py startapp auction
```

Step 5: Add app name in Settings.py file. 'auction' or paste the below code in Settings.py file



```
INSTALLED_APPS = [
    'auction',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

Django Basics Django Projects Django Interview Question Flask Flask Projects Python API Torando Che

Settings.py File

Python

```python
# Online_Auction &gt; Online_Auction &gt; settings.py
import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))


# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/2.2/howto/deployment/che
```

```
cklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '_igtoud8=l*xejhiwtfz$ce-fu*%80&amp;^hybem91#g#)5j7td#z'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
    'auction',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'Online_Auction.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'Online_Auction.wsgi.application'


# Database
# https://docs.djangoproject.com/en/2.2/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}


# Password validation
# https://docs.djangoproject.com/en/2.2/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]


# Internationalization
# https://docs.djangoproject.com/en/2.2/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'Asia/Kolkata'

USE_I18N = True

USE_L10N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/2.2/howto/static-files/

STATIC_URL = '/static/'
MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR,'media')
```

Step :6 Paste the Below code in urls.py file in Online_Auction.

Python

```
# Online_Auction &gt; Online_Auction &gt; urls.py
from django.contrib import admin
from django.urls import path
from django.conf import settings
from django.conf.urls.static import static
from auction.views import *

urlpatterns = [
    path('admin/', admin.site.urls),
    path('',Home,name=&quot;home&quot;),
    path('user_home',Bidder_Home,name=&quot;user_home&quot;),
    path('trainer_home',Auction_User,name=&quot;trainer_home&quot;),
    path('login_user',Login_User,name=&quot;login_user&quot;),
    path('contact',Contact,name=&quot;conRtact&quot;),
    path('about',About,name=&quot;about&quot;),
    path('contact',Contact,name=&quot;contact&quot;),
    path('edit_profile',Edit_Profile,name=&quot;edit_profile&quot;),
    path('edit_profile1',Edit_Profile1,name=&quot;edit_profile1&quot;),
    path('logout', Logout, name=&quot;logout&quot;),
    path('login_admin',Login_Admin,name=&quot;login_admin&quot;),
    path('signup', Signup_User,name=&quot;signup&quot;),
    path('change_password',Change_Password,name=&quot;change_password&quot;),
    path('change_password1',Change_Password1,name=&quot;change_password1&quot;),
    path('admin_home', Admin_Home,name=&quot;admin_home&quot;),
    path('add_product', Add_Product,name=&quot;add_product&quot;),
    path('new_product', New_product,name=&quot;new_product&quot;),
    path('bidder_user', Bidder_User,name=&quot;bidder_user&quot;),
    path('user_delete/&lt;int:id&gt;', user_delete,name=&quot;user_delete&quot;),
    path('seller_user', Seller_User,name=&quot;seller_user&quot;),
    path('user_delete_seller/&lt;int:id&gt;', user_delete_seller,name=&quot;user_delete_seller&quot;),
    path('all_product2', All_product2,name=&quot;all_product2&quot;),
    path('profile', profile, name='profile'),
    path('result', result, name='result'),
    path('view_auction/&lt;int:pid&gt;', view_auction, name='view_auction'),
    path('particpated_user(&lt;int:pid&gt;)', Participated_user, name='particpated_user'),
    path('profile1', Profile1, name='profile1'),
    path('status(&lt;int:pid&gt;)', Change_status, name='status'),
    path('winner(&lt;int:pid&gt;)', Winner,name='winner'),
    path('winner2(&lt;int:pid&gt;)', Winner2,name='winner2'),
    path('winner1(&lt;int:pid&gt;)', Winner1,name='winner1'),
    path('start_auction(&lt;int:pid&gt;)', Start_Auction, name='start_auction'),
    path('view_category', view_category, name='view_category'),
    path('view_subcategory', view_subcategory, name='view_subcategory'),
    path('view_session_date', view_session_date, name='view_session_date'),
    path('view_session_time', view_session_time, name='view_session_time'),
    path('add_category', Add_Category, name='add_category'),
    path('add_subcategory', Add_SubCategory, name='add_subcategory'),
    path('add_session_date', Add_Session_date, name='add_session_date'),
    path('add_session_time', Add_Session_time, name='add_session_time'),
    path('bidding_status', Bidding_Status, name='bidding_status'),
    path('bidding_status2', Bidding_Status2, name='bidding_status2'),
    path('all_product', All_product, name='all_product'),
    path('edit_category(&lt;int:pid&gt;)', Edit_Category, name='edit_category'),
    path('product_detail(&lt;int:pid&gt;)', product_detail, name='product_detail'),
    path('edit_subcategory(&lt;int:pid&gt;)', Edit_SubCategory, name='edit_subcategory'),
    path('edit_session_date(&lt;int:pid&gt;)', Edit_Session_date, name='edit_session_date'),
    path('edit_session_time(&lt;int:pid&gt;)', Edit_Session_time, name='edit_session_time'),
    path('delete_category(&lt;int:pid&gt;)', delete_category, name='delete_category'),
    path('delete_subcategory(&lt;int:pid&gt;)', delete_subcategory, name='delete_subcategory'),
    path('delete_session_date(&lt;int:pid&gt;)', delete_session_date, name='delete_session_date'),
    path('delete_session_time(&lt;int:pid&gt;)', delete_session_time, name='delete_session_time'),
    path('load-courses/', load_courses, name='ajax_load_courses'),
    path('load-courses1/', load_courses1, name='ajax_load_courses1'),
    path('product_detail2(&lt;int:pid&gt;)', product_detail2, name='product_detail2'),
]+static(settings.MEDIA_URL,document_root=settings.MEDIA_ROOT)
```

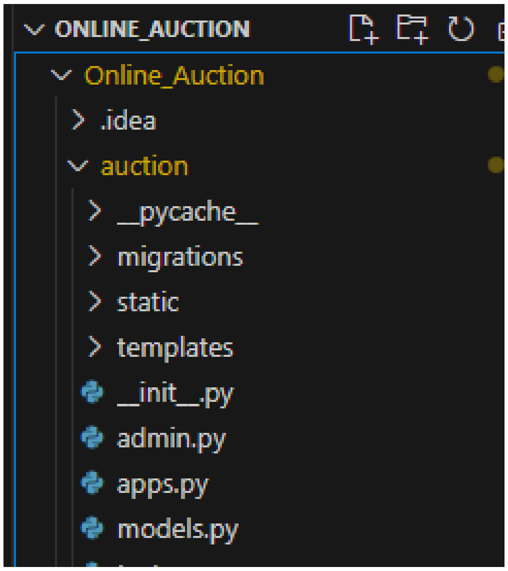
Step 7: Go to admin.py file and paste the below code.

Python

```
#Online_Auction &gt; auction &gt; admin.py
from django.contrib import admin
from .models import *

# Register your models here.'
admin.site.register(Bidder)
admin.site.register(Result)
admin.site.register(Payment)
admin.site.register(Member_fee)
admin.site.register(Status)
admin.site.register(Send_Feedback)
admin.site.register(Auction_User)
admin.site.register(Category)
admin.site.register(Sub_Category)
admin.site.register(Session_date)
admin.site.register(Session_Time)
admin.site.register(Product)
admin.site.register(Aucted_Product)
admin.site.register(Participant)
```

Step 9: Create two folders in auction one is "templates" , second is "static"



```
ONLINE_AUCTION
  Online_Auction
    .idea
    auction
      __pycache__
      migrations
      static
      templates
      __init__.py
      admin.py
      apps.py
      models.py
```

Online_Auction > auction > Static , Online_Auction > auction > templates

Step 10: Open views.py file from auction_app and paste the below code.

Python

```python
from django.shortcuts import render
from django.shortcuts import render, redirect
from django.http import HttpResponse
from auction_app.models import *
from datetime import date
today=date.today()
global_var = None # Initialize it with a default value
# Function to set the global variable to a dynamic value
def set_global_variable(value):
    global global_var
    global_var = value
# Function to access the global variable
```

```
def access_global_variable():
    return global_var
def maxbid(id):
    li=[]
    global maximum
    item = Item.objects.get(id=id)
    print(&quot;item id--&gt;&quot;,item.id)
    user = User.objects.all().values()
    # item = Item.objects.all().values()
    for i in user:
        a = i['bid_amt']
        print(&quot;--&gt;&quot;,a)
        li.append(a)
        maximum = max(li)
    print(li)
    print(&quot;maximum&quot;,maximum)
    item.max_bid = maximum
    item.save()
    print(&quot;Item Saved&quot;)
    return maximum
        # print(a)
        # li.append(i)
        # print(&quot;list--&gt;&quot;,li)
        # print(max(li))
        # return a
def bid(request):
    item = Item.objects.all()
    # user = User.objects.all()
    maxbid()
    # print(item.item_name.values)
    return render(request,'base.html',
{&quot;item&quot;:item})
def login(request):
    if request.method == 'POST':
        email = request.POST['email']
        print(email)
        password = request.POST['password']
        print(password)
        item = Item.objects.all()
        try:
            user = User.objects.get(email=email)
            print(&quot;userpass---
&gt;&quot;,user.password)
            if user is not None:
                if password == user.password:
                    global mailid
                    # print(&quot;Mail id here---&gt;&quot;,item.item_status)
                    set_global_variable(email)
                    mailid = email
                    mail= email
                    # print(&quot;Ye rhi dates--&gt;&quot;,item.start_date)
                    # print(&quot;ye print ho gyi---&gt;&quot;,mailid)
                    # print(&quot;outer funciton global ---&gt;&quot;,mail)
                    return render(request,'base.html',{&quot;item&quot;:item,&quot;mailid&quot;:mailid})
                else:
                    print(&quot;Wrong pass&quot;)
                    return HttpResponse(&quot;Wrong Password&quot;)
        except:
            return HttpResponse(&quot;email id Not exist&quot;)
            print(&quot;email id Not  exist&quot;)
    return render(request,'login.html')
def register(request):
    if request.method == 'POST':
        name = request.POST['name']
        print(name)
        email = request.POST['email']
        print(email)
        phone = request.POST['phone']
        print(phone)
        password = request.POST['password']
        print(password)
        user = User.objects.all().values()
        for i in user:
            flag=0
            dataemail = i['email']
            if dataemail == email:
                flag=1
                return HttpResponse(&quot;Email id Already Exist&quot;)
        if flag!=1:
            new=User.objects.create(name=name,email=email,phone=phone,
password=password )
            new.save()
            print(f&quot;User Create with email {email}&quot;)
            return HttpResponse(f&quot;User Created Successfully with email {email}&quot;)
    return render(request,'register.html')
def applybid(request,id):
    mail = access_global_variable()
    print(&quot;Yaha pe ye mail ai h applybid mei --&gt;&quot;,mail)
    item = Item.objects.get(id=id)
    user = User.objects.get(email=mail)
    print(&quot;user---&gt;&quot;,user.email)
    # for i in user:
    #     # print(&quot;i--&gt;&quot;,i)
    #     if i.email == mail:
    #         print(&quot;mail id--&gt;&quot;,i.email)
    if request.method == &quot;POST&quot;:
        maximum=request.POST['max_bid']
        item.max_bid = maximum
        user.bid_amt = maximum
        print(&quot;user.status---&gt;&quot;,user.status)
        user.status =&quot;Applied&quot;
        print(&quot;user.status---&gt;&quot;,user.status)
        item.save()
        user.save()
        print(&quot;Hogya save&quot;)
        maxbid(id)
        return redirect('/')
    return render(request,'applybid.html', {&quot;item&quot;:item,&quot;user&quot;:user})
def home(request):
    item = Item.objects.all()
    return render(request,'home.html', {&quot;item&quot;:item})
```

Step 11: Create the **base.html** file in templates folder and paste the below code.

HTML

```
&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
&lt;style&gt;
table {
  font-family: arial, sans-serif;
  border-collapse: collapse;
  width: 100%;
}

td, th {
  border: 1px solid #dddddd;
  text-align: left;
  padding: 8px;
}

tr:nth-child(even) {
  background-color: #dddddd;
}
a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 14px 25px;
  text-align: right;
  text-decoration: none;
  display: inline-block;
}

a:hover, a:active {
  background-color: Blue;
}
&lt;/style&gt;
&lt;/head&gt;
&lt;body&gt;

&lt;center&gt;&lt;h2&gt;Available Auction&lt;/h2&gt;&lt;/center&gt;
&lt;a href=&quot;{% url 'home' %}&quot; target=&quot;_blank&quot;&gt;Logout&lt;/a&gt;
{{mailid}}

&lt;table&gt;
```

```
  &lt;thread&gt;
  &lt;tr&gt;
    &lt;th&gt;SNo.&lt;/th&gt;
    &lt;th&gt;Product Name&lt;/th&gt;
    &lt;th&gt; Bid Amount &lt;/th&gt;
    &lt;th&gt;Description&lt;/th&gt;
    &lt;th&gt;Start Date&lt;/th&gt;
    &lt;th&gt;End Date&lt;/th&gt;
    &lt;th&gt;Action &lt;/th&gt;
  &lt;/tr&gt;
&lt;/thread&gt;
{% for i in item %}
  &lt;tr&gt;
    &lt;td&gt;{{forloop.counter}}&lt;/td&gt;
    &lt;td&gt;  {{i.item_name}}&lt;/td&gt;
    &lt;td&gt;  {{i.basic_amount}}&lt;/td&gt;
    &lt;td&gt;  {{i.desc}}&lt;/td&gt;
    &lt;td&gt;  {{i.start_date}}&lt;/td&gt;
    &lt;td&gt;  {{i.end_date}}&lt;/td&gt;
    &lt;td&gt;&lt;a href=&quot;{% url 'applybid' i.id %}&quot; target=&quot;_blank&quot;&gt;Apply&lt;/a&gt;&lt;/td&gt;
    {% comment %} &lt;td&gt;&lt;a href=&quot;{% url 'applybid' i.id %}&quot;&gt;&lt;button class=&quot;btn btn-primary&quot;&gt; Apply&lt;/button&gt;&lt;/a&gt; &lt;/td&gt; {% endcomment %}

    {% endfor %}
  &lt;/tr&gt;
&lt;/table&gt;

&lt;/body&gt;
&lt;/html&gt;
```

Step 12: Create the **home.html** file in templates folder and paste the below code.

HTML

```
&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
```

```
&lt;style&gt;
table {
  font-family: arial, sans-serif;
  border-collapse: collapse;
  width: 100%;
}

td, th {
  border: 1px solid #dddddd;
  text-align: left;
  padding: 8px;
}

tr:nth-child(even) {
  background-color: #dddddd;
}
    a:link, a:visited {
      background-color: #f44336;
      color: white;
      padding: 14px 25px;
      text-align: right;
      text-decoration: none;
      display: inline-block;
    }

    a:hover, a:active {
      background-color: Blue;
    }
    &lt;/style&gt;
&lt;/head&gt;
&lt;body&gt;

&lt;center&gt;&lt;h2&gt;Available Auction&lt;/h2&gt;&lt;/center&gt;
&lt;a href=&quot;{% url 'login' %}&quot; target=&quot;_blank&quot;&gt;Login&lt;/a&gt;
&lt;table&gt;
  &lt;thread&gt;
  &lt;tr&gt;
    &lt;th&gt;SNo.&lt;/th&gt;
    &lt;th&gt;Product Name&lt;/th&gt;
    &lt;th&gt; Bid Amount &lt;/th&gt;
    &lt;th&gt; Max Bid &lt;/th&gt;
    &lt;th&gt;Results &lt;/th&gt;
  &lt;/tr&gt;
&lt;/thread&gt;
{% for i in item %}
  &lt;tr&gt;
    &lt;td&gt;{{forloop.counter}}&lt;/td&gt;
    &lt;td&gt;  {{i.item_name}}&lt;/td&gt;
    &lt;td&gt;  {{i.basic_amount}}&lt;/td&gt;
    &lt;td&gt;{{i.max_bid}}&lt;/td&gt;
    &lt;td&gt;&lt;/td&gt;

    {% endfor %}
  &lt;/tr&gt;
&lt;/table&gt;

&lt;/body&gt;
&lt;/html&gt;
```

Step 13: Create the **login.html** file in templates folder and paste the below code.

HTML

```
&lt;!DOCTYPE html&gt;
&lt;html lang=&quot;en&quot;&gt;
&lt;head&gt;
    &lt;meta charset=&quot;UTF-8&quot;&gt;
    &lt;meta name=&quot;viewport&quot; content=&quot;width=device-width, initial-scale=1.0&quot;&gt;
    &lt;title&gt;Login Page&lt;/title&gt;
    &lt;style&gt;
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 0;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
        }

        .container {
            background-color: white;
            padding: 20px;
            border-radius: 5px;
            box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.2);

        }

        input[type=&quot;text&quot;], input[type=&quot;password&quot;] {
            width: 90%;
            padding: 7px;
            margin: 8px 0;
            border: 1px solid #ccc;
            border-radius: 3px;
        }

        button {
            background-color: #007BFF;
            color: white;
            padding: 10px 20px;
            border: none;
            border-radius: 3px;
            cursor: pointer;
        }

        button:hover {
            background-color: #0056b3;
        }

        a {
            text-decoration: none;
            color: #007BFF;
        }
    &lt;/style&gt;
&lt;/head&gt;
&lt;body&gt;
    &lt;div class=&quot;container&quot;&gt;
        &lt;center&gt;&lt;h2&gt;LOGIN PAGE&lt;/h2&gt;&lt;/center&gt;
        &lt;form method=&quot;post&quot;&gt;
        {% csrf_token %}
            &lt;label for=&quot;email&quot;&gt;Email:&lt;/label&gt;&lt;br&gt;
            &lt;input type=&quot;text&quot; id=&quot;email&quot; name=&quot;email&quot; required&gt;&lt;br&gt;&lt;br&gt;

            &lt;label for=&quot;password&quot;&gt;Password:&lt;/label&gt;&lt;br&gt;
            &lt;input type=&quot;password&quot; id=&quot;password&quot; name=&quot;password&quot; required&gt;&lt;br&gt;&lt;br&gt;

            &lt;center&gt;&lt;button type=&quot;submit&quot;&gt;Login&lt;/button&gt;&lt;/center&gt;
        &lt;/form&gt;
        &lt;p&gt;Don't have an account? &lt;a href=&quot;{% url 'register'%}&quot;&gt;Register here&lt;/a&gt;&lt;/p&gt;
    &lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;
```

Step 14: Create the **register.html** file in templates folder and paste the below code.

HTML

```
&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
    &lt;title&gt;User Registration&lt;/title&gt;
    &lt;style&gt;
        body {
            font-family: Arial, Helvetica, sans-serif;
            background-color: #f2f2f2;
        }
        .container {
            background-color: #fff;
            border-radius: 5px;
            box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
            padding: 20px;
            margin: 0 auto;
            max-width: 400px;
        }
        .form-group {
            margin-bottom: 15px;
        }
        label {
            display: block;
            font-weight: bold;
        }
        input[type=&quot;text&quot;],
        input[type=&quot;email&quot;],
        input[type=&quot;tel&quot;],
        input[type=&quot;password&quot;] {
            width: 100%;
            padding: 7px;
            margin-bottom: 10px;
            border: 1px solid #ccc;
            border-radius: 3px;
        }
        input[type=&quot;submit&quot;] {
            background-color: #007BFF;
            color: #fff;
            border: none;
            border-radius: 3px;
            padding: 10px 20px;
            cursor: pointer;
        }
    &lt;/style&gt;
&lt;/head&gt;
&lt;body&gt;
    &lt;div class=&quot;container&quot;&gt;
        &lt;center&gt;&lt;h2&gt;User Registration&lt;/h2&gt;&lt;/center&gt;
        &lt;form action=&quot;&quot; method=&quot;POST&quot;&gt;
        {% csrf_token %}
            &lt;div class=&quot;form-group&quot;&gt;
                &lt;label for=&quot;name&quot;&gt;Name:&lt;/label&gt;
                &lt;input type=&quot;text&quot; id=&quot;name&quot; name=&quot;name&quot; required&gt;
            &lt;/div&gt;
            &lt;div class=&quot;form-group&quot;&gt;
                &lt;label for=&quot;email&quot;&gt;Email:&lt;/label&gt;
                &lt;input type=&quot;email&quot; id=&quot;email&quot; name=&quot;email&quot; required&gt;
            &lt;/div&gt;
            &lt;div class=&quot;form-group&quot;&gt;
                &lt;label for=&quot;phone&quot;&gt;Phone:&lt;/label&gt;
                &lt;input type=&quot;tel&quot; id=&quot;phone&quot; name=&quot;phone&quot; required&gt;
            &lt;/div&gt;
            &lt;div class=&quot;form-group&quot;&gt;
                &lt;label for=&quot;password&quot;&gt;Password:&lt;/label&gt;
                &lt;input type=&quot;password&quot; id=&quot;password&quot; name=&quot;password&quot; required&gt;
            &lt;/div&gt;
            &lt;center&gt;&lt;input type=&quot;submit&quot; value=&quot;Register&quot;&gt;&lt;/center&gt;
        &lt;/form&gt;
    &lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;
```

Step 15: Create the **applybid.html** file in templates folder and paste the below code.

HTML

```
&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
    &lt;title&gt;Auction Item Form&lt;/title&gt;
    &lt;style&gt;
        body {
            font-family: Arial, Helvetica, sans-serif;
            background-color: #f2f2f2;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            margin: 0;
        }

        .form-container {
            background-color: #ffffff;
            border: 1px solid #ccc;
            border-radius: 5px;
            padding: 100px;
            box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
        }

        .form-container h2 {
            text-align: center;
        }

        .form-container label,
        .form-container input,
        .form-container textarea {
            display: block;
            margin-bottom: 15px;
        }

        .form-container input,
        .form-container textarea {
            width: 100%;
            padding: 10px;
            border: 1px solid #ccc;
            border-radius: 3px;
        }

        .form-container input[type=&quot;submit&quot;] {
            background-color: #007BFF;
            color: #fff;
            border: none;
            border-radius: 3px;
            padding: 10px 20px;
            cursor: pointer;
        }
    &lt;/style&gt;
&lt;/head&gt;
&lt;body&gt;
    &lt;div class=&quot;form-container&quot;&gt;
        &lt;h2&gt;Auction Participation Form &lt;/h2&gt;
        &lt;form method=&quot;POST&quot;&gt;
        {% csrf_token %}
            &lt;div&gt;
```

```
                    &lt;label for=&quot;item_name&quot;&gt;Item Name:&lt;/label&gt;
                    &lt;input type=&quot;text&quot; id=&quot;item_name&quot; name=&quot;item_name&quot; placeholder=&quot;{{item.item_name}}&quot; readonly&gt;
                &lt;/div&gt;
                &lt;div&gt;
                    &lt;label for=&quot;description&quot;&gt;Description:&lt;/label&gt;
                    &lt;textarea id=&quot;description&quot; name=&quot;description&quot; rows=&quot;3&quot; placeholder=&quot;{{item.desc}}&quot; readonly&gt;&lt;/textarea&gt;
                &lt;/div&gt;
                &lt;div&gt;
                    &lt;label for=&quot;basic_amount&quot;&gt;Basic Amount:&lt;/label&gt;
                    &lt;input type=&quot;number&quot; id=&quot;basic_amount&quot; name=&quot;basic_amount&quot; placeholder=&quot;{{item.basic_amount}}&quot; readonly&gt;
                &lt;/div&gt;
                &lt;div&gt;
                    &lt;label for=&quot;start_date&quot;&gt;Start Date:&lt;/label&gt;
                    &lt;input type=&quot;text&quot; id=&quot;start_date&quot; name=&quot;start_date&quot; placeholder=&quot;{{item.start_date}}&quot; readonly&gt;
                &lt;/div&gt;
                &lt;div&gt;
                    &lt;label for=&quot;end_date&quot;&gt;End Date:&lt;/label&gt;
                    &lt;input type=&quot;text&quot; id=&quot;end_date&quot; name=&quot;end_date&quot; placeholder=&quot;{{item.end_date}}&quot; readonly&gt;
                &lt;/div&gt;
                &lt;div&gt;
                    &lt;label for=&quot;max_bid&quot;&gt;Max Bid:&lt;/label&gt;
                    &lt;input type=&quot;number&quot; id=&quot;max_bid&quot; name=&quot;max_bid&quot; min=&quot;{{item.basic_amount}}&quot; required&gt;
                &lt;/div&gt;
                &lt;div&gt;
                    &lt;input type=&quot;submit&quot; value=&quot;Submit&quot;&gt;
                &lt;/div&gt;
            &lt;/form&gt;
```

```
        &lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;
```

Step :16 Open the Shell and run the following commands

```
Python manage.py makemigrations
python manage.py migrate
python manage.py runserver
```

Step: 17 Open the Browser and run the server.

Useful Resource: Building an auction system can be quite complex. For those looking to further explore advanced Django projects, the Django Web Development Course is a valuable resource.

Are you ready to elevate your web development skills from foundational knowledge to advanced expertise? Explore our Mastering Django Framework - Beginner to Advanced Course on GeeksforGeeks, designed for aspiring developers and experienced programmers. This comprehensive course covers everything you need to know about Django, from the basics to advanced features. Gain practical experience through **hands-on projects** and real-world applications, mastering essential Django principles and techniques. Whether you’re just starting or looking to refine your skills, this course will empower you to build sophisticated web applications efficiently. Ready to enhance your web development journey? Enroll now and unlock your potential with Django!

H harsh... 1

Previous Article **Next Article**

Joke Application Project Using Django Framework

Similar Reads

Real-Time Auction Platform using Node and Express.js

The project is a Real-Time Auction Platform developed using Node.js Express.js and MongoDB database for storing details where users can browse different...

12 min read

Online Markdown Editor using Django

In this article, we will guide you through the process of creating an Online Markdown Editor using Django. This powerful tool allows users to seamlessly...

5 min read

Online Survey Tool using Django

In this article, we'll help you make an Online Survey Tool using Django. In our tool, only admin users can create forms to keep things in control and ensure...

6 min read

Online Resume Builder using Django

In this article, we will create an online resume builder web app. We will start by filling out a form with our personal information. Once the form is completed, we...

9 min read

Build an Authentication System Using Django, React and Tailwind

In this article, we will guide you in building an Authentication system using React and Tailwind with the Django Framework that includes features like sign up, log...

15+ min read

Anonymous Feedback System using Django

In web development, getting feedback from users is super important for making websites better. One way to do that is by having a feedback form where users c...

5 min read

Voting System Project Using Django Framework

Project Title: Pollster (Voting System) web application using Django frameworkType of Application (Category): Web application. Introduction: We wil...

14 min read

College Management System using Django - Python Project

In this article, we are going to build College Management System using Django and will be using dbsqlite database. In the times of covid, when education has...

15+ min read

User Authentication System using Django

In this article, we will explore the process of creating a secure and efficient user authentication system in Django, a high-level Python web framework. Building...

10 min read

Movie Recommendation System using Django

In this article, we will guide you through the process of creating a comprehensive Movie Recommendation System with the added functionality of user...

8 min read

Article Tags : Project Django Django-Projects

Corporate & Communications Address:-
A-143, 9th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305) | Registered Address:- K 061,
Tower K, Gulshan Vivante Apartment,
Sector 137, Noida, Gautam Buddh
Nagar, Uttar Pradesh, 201305

Company

About Us
Legal
In Media
Contact Us

Languages

Python
Java
C++
PHP

Advertise with us

GFG Corporate Solution

Placement Training Program

GeeksforGeeks Community

GoLang

SQL

R Language

Android Tutorial

Tutorials Archive

DSA

Data Structures

Algorithms

DSA for Beginners

Basic DSA Problems

DSA Roadmap

Top 100 DSA Interview Problems

DSA Roadmap by Sandeep Jain

All Cheat Sheets

Data Science & ML

Data Science With Python

Data Science For Beginner

Machine Learning

ML Maths

Data Visualisation

Pandas

NumPy

NLP

Deep Learning

Web Technologies

HTML

CSS

JavaScript

TypeScript

ReactJS

NextJS

Bootstrap

Web Design

Python Tutorial

Python Programming Examples

Python Projects

Python Tkinter

Web Scraping

OpenCV Tutorial

Python Interview Question

Django

Computer Science

Operating Systems

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

Software Development

Software Testing

DevOps

Git

Linux

AWS

Docker

Kubernetes

Azure

GCP

DevOps Roadmap

System Design

High Level Design

Low Level Design

UML Diagrams

Interview Guide

Design Patterns

OOAD

System Design Bootcamp

Interview Questions

Inteview Preparation

Competitive Programming

Top DS or Algo for CP

Company-Wise Recruitment Process

Company-Wise Preparation

Aptitude Preparation

Puzzles

School Subjects

Mathematics

GeeksforGeeks Videos

DSA