



Receipt Print with GUI using Django

Last Updated : 30 Aug, 2024

In this article, we will guide you through creating a Receipt Print GUI. To achieve this, we first establish a login system, allowing sellers to create accounts through registration and subsequent login. Once logged in, we provide a straightforward form where sellers input the name, price, and quantity of materials. After clicking the "Add" button, the system calculates the total by multiplying the price and quantity, and the sum is displayed in a table at the bottom. Sellers can conveniently manage their data by accessing options to delete or update entries with a single click. When ready, they can generate bills effortlessly by clicking the "Generate Bill" button. Finally, sellers can log out with a simple click on the "Logout" button.

Receipt Print with GUI using Django

Sellers can conveniently manage their data by accessing options to delete or update entries with a single click. When ready, they can generate bills effortlessly by clicking the "Generate Bill" button. Finally, sellers can log out with a simple click on the "Logout" button.

To install [Django](#) follow these [steps](#).

Starting the Project Folder

To start the project use this command

```
django-admin startproject core
cd core
```

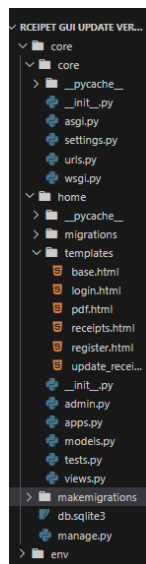
To start the app use this command

```
python manage.py startapp home
```

Now add this app to the 'settings.py'

```
INSTALLED_APPS = [  
    "django.contrib.admin",  
    "django.contrib.auth",  
    "django.contrib.contenttypes",  
    "django.contrib.sessions",  
    "django.contrib.messages",  
    "django.contrib.staticfiles",  
    "home",  
]
```

File Structure



file Structure

Setting Necessary Files

models.py: This Django model defines a Receipt with fields for user association, name, price, quantity, and total. It includes a foreign key to the User model and sets default values for the fields.

Python



```
1 from django.db import models  
2 from django.contrib.auth.models import User  
3  
4 class Receipt(models.Model):  
5     user = models.ForeignKey(User,  
        on_delete=models.SET_NULL, null=True, blank=True)
```

```
        name = models.CharField(max_length=100,
default='something')
7     price = models.IntegerField(default=0)
8     quantity = models.IntegerField(default =0)
9     total = models.IntegerField(default=0)
```

views.py: The functions in the view file collectively build a Django web application for managing receipt data, user accounts, and generating PDFs.

[Django Basics](#) [Django Projects](#) [Django Interview Question](#) [Flask](#) [Flask Projects](#) [Python API](#) [Torando](#) [Che](#)



```
1  from django.shortcuts import render, redirect
2  from .models import Receipt
3  from django.http import HttpResponse, JsonResponse,
    HttpResponseRedirect
4  from django.contrib import messages
5  from django.contrib.auth import login, authenticate
6  from django.contrib.auth.decorators import login_required
7  from django.contrib.auth.models import User
8  from django.contrib.auth import logout
9
10 @login_required(login_url='/login/')
11 def receipts(request):
12     if request.method == 'POST':
13         data = request.POST
14         name = data.get('name')
15         price = data.get('price')
16         quantity = data.get('quantity')
17         total = float(price) * int(quantity)
18
19         Receipt.objects.create(
20             name = name,
21             price=price,
22             quantity=quantity,
23             total=total
24         )
25         return redirect('/')
26
27     queryset = Receipt.objects.all()
28     if request.GET.get('search'):
29         queryset = queryset.filter(
30             name__icontains=request.GET.get('search'))
```

```
32         # Calculate the total sum
33         total_sum = sum(receipt.total for receipt in queryset)
34         context = {'receipts': queryset, 'total_sum':
total_sum}
35         return render(request, 'receipts.html', context)
36 @login_required(login_url='/login/')
37 def update_receipt(request, id):
38     queryset = Receipt.objects.get(id=id)
39
40     if request.method == 'POST':
41         data = request.POST
42         name = data.get('name')
43         price = data.get('price')
44         quantity = data.get('quantity')
45         total = float(price) * int(quantity)
46
47         queryset.name = name
48         queryset.price = price
49         queryset.quantity = quantity
50         queryset.total = total
51         queryset.save()
52         return redirect('/')
53
54     context = {'receipt': queryset}
55     return render(request, 'update_receipt.html', context)
56
57 @login_required(login_url='/login/')
58 def delete_receipt(request, id):
59     queryset = Receipt.objects.get(id=id)
60     queryset.delete()
61     return redirect('/')
62
63 def login_page(request):
64     if request.method == 'POST':
65         try:
66             username = request.POST.get('username')
67             password = request.POST.get('password')
68             user_obj =
User.objects.filter(username=username)
69             if not user_obj.exists():
70                 messages.error(request, 'Username not
found')
71                 return redirect('/login/')
```

```
        user_obj = authenticate(username=username,
password=password)
73         if user_obj:
74             login(request, user_obj)
75             return redirect('receipts')
76             messages.error(request, "Wrong
Password");)
77             return redirect('/login/')
78         except Exception as e:
79             messages.error(request, "Something went
wrong");)
80             return redirect('/register/')
81         return render(request, "login.html");)
82
83 def register_page(request):
84     if request.method == "POST";:
85         try:
86             username = request.POST.get('username')
87             password = request.POST.get('password')
88             user_obj =
User.objects.filter(username=username)
89             if user_obj.exists():
90                 messages.error(request, "Username is
taken");)
91                 return redirect('/register/')
92                 user_obj =
User.objects.create(username=username)
93                 user_obj.set_password(password)
94                 user_obj.save()
95                 messages.success(request, "Account
created");)
96                 return redirect('/login')
97             except Exception as e:
98                 messages.error(request, "Something went
wrong");)
99                 return redirect('/register')
100         return render(request, "register.html");)
101
102 def custom_logout(request):
103     logout(request)
104     return redirect('login')
105
106 @login_required(login_url='/login/')
107 def pdf(request):
108     if request.method == 'POST':
```

```
data = request.POST
110     name = data.get('name')
111     price = data.get('price')
112     quantity = data.get('quantity')
113     total = float(price) * int(quantity)
114
115     Receipt.objects.create(
116         name = name,
117         price=price,
118         quantity=quantity,
119         total=total
120     )
121     return redirect('pdf')
122
123     queryset = Receipt.objects.all()
124
125     if request.GET.get('search'):
126         queryset = queryset.filter(
127             name__icontains=request.GET.get('search'))
128
129
130     total_sum = sum(receipt.total for receipt in queryset)
131
132     context = {'receipts': queryset, 'total_sum':
total_sum}
133     return render(request, 'pdf.html', context)
```

Creating GUI

login.html: This HTML code is for a basic login page within a Bootstrap-styled container. It includes fields for a username and password, a "Login" button, and a link to create a new account. The page uses some custom CSS for styling and includes external libraries for additional styling.

HTML

```
1 <!doctype html>
2 <html lang="en">
3
4 <head>
```

```

        <!-- Required meta tags -->
        <meta charset="utf-8">
        <meta name="viewport"
content="width=device-width, initial-scale=1">
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist
/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfSpd3yD65VohhpooC0mL
ASjC" crossorigin="anonymous">
        <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css">
        <title>Job Portal</title>
    </head>
    <style>
        .text{
            color: green;
            font-weight: bold;
            font-family: 'Times New Roman', Times, serif;
        }
    </style>
    <body><br><br><br><br>

        <br><br>

        <div class="container mt-4 bg-white col-md-3
card shadow p-3" id="log">
            <div class="login-form">
                {% if messages %}
                {% for message in messages %}
                <div class="alert alert-success {{
message.tags }}" mt-4" role="alert">
                    {{ message }}
                </div>
                {% endfor %}
                {% endif %}
                <form action=""
method="post">
                    {% csrf_token %}
                    <h3 class="text text-
center">GeeksforGeeks </h3>

```

```

        <h4 class="text-center"
style="font-family: 'Times New Roman', Times,
serif;"> Login </h4>
39         <div class="form-group">
40             <input type="text"
class="form-control" name="username"
placeholder="Username" required
41                 style="background-color: #fff;
border: 1px solid #ddd; border-radius: 5px; padding:
10px;">
42         </div>
43         <div class="form-group mt-
2">
44             <input type="password"
class="form-control" name="password"
placeholder="Password" required
45                 style="background-color: #fff;
border: 1px solid #ddd; border-radius: 5px; padding:
10px;">
46         </div>
47         <div class="form-group mt-
2">
48             <button class="btn btn-success
btn-block" style="margin-left:
138px;">Login ???</button>
49         </div>
50         <br>
51     </form>
52     <p class="text-center"
style="color: #555;"><a href="{% url
'register' %}" style="color:
#007bff;">Create an
53         Account.</a></p>
54     </div>
55 </div>
56
57 </body>
58
59 </html>

```

register.html: This HTML code is for a simple registration page. It includes fields for a username and password, a "Register" button, and a link to log in. The page uses Bootstrap and Font Awesome for styling and includes custom CSS for additional styling.

HTML



```
1  <!doctype html>;
2  <html lang="en">;
3
4  <head>;
5      <!-- Required meta tags -->;
6      <meta charset="utf-8">;
7      <meta name="viewport"
content="width=device-width, initial-scale=1">;
8      <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist
/css/bootstrap.min.css" rel="stylesheet"
9          integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfSpd3yD65VohhpooC0mL
ASjC" crossorigin="anonymous">;
10     <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css">;
11     <title>Job Portal</title>;
12 </head>;
13
14 <body>;
15
16     <style>;
17         .text{
18             color: green;
19             font-weight: bold;
20             font-family: 'Times New Roman', Times, serif;
21         }
22
23     </style>;
24
25     <body>;
26         <br>;
27         <br><br><br><br><br><br>;
28         <div class="container mt-4  bg-white mx-
auto col-md-3 card shadow p-3">;
29             <div class="login-form">;
30                 {% if messages %}
31                 {% for message in messages %}
32                 <div class="alert alert-success {{
message.tags }}" role="alert">;
```

```

    {{ message }}
34         </div>
35     {% endfor %}
36 {% endif %}
37     <form action=""
method=""post>
38         {% csrf_token %}
39         <h3 class="text text-
center">GeeksforGeeks </h3>
40         <h4 class="text-center"
style="font-family: 'Times New Roman', Times,
serif;">Register </h4>
41         <div class="form-group">
42             <input type="text"
class="form-control" name="username"
placeholder="Username" required>
43         </div>
44         <div class="form-group mt-
2">
45             <input type="password"
class="form-control" name="password"
placeholder="Password" required>
46         </div>
47         <div class="form-group mt-
2">
48             <button class="btn btn-
success btn-block" style="margin-left:
117px;">Register ???</button>
49         </div>
50         <br>
51     </form>
52     <p class="text-center"><a
href="{% url 'login' %}">Log In
</a></p>
53     </div>
54 </div>
55
56 </body>
57
58 </html>

```

receipt.html: This HTML template extends a base template, and it creates a form for users to add receipts with item details. It also displays a list of existing receipts in a table, allowing users to delete or update them. The page includes

styling with Bootstrap and custom CSS for aesthetics and layout. Additionally, there are options to generate a bill and log out.

HTML

```

1 {% extends "base.html" %}
2 {% block start %}
3
4 <link
  href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/
  css/bootstrap.min.css";
  rel="stylesheet">
5 <style>
6   .text{
7     color: green;
8     font-weight: bold;
9     font-family: 'Times New Roman', Times, serif;
10  }.ok{
11     color: white;
12     text-decoration: none;
13  }
14  .ok:hover{
15     color: white;
16     text-decoration: none;
17  }
18 </style>
19
20 <div class="container mt-5 col-6">
21   <form class="col-6 mx-auto card p-3 shadow-
  lg" method="post"
  enctype="multipart/form-data">
22     {% csrf_token %}
23     <h3 class="text text-center">
  GeeksforGeeks </h3>
24     <br>
25     <h4 style="font-family: 'Times New Roman',
  Times, serif;">Create Receipt??? </h4>
26     <hr>
27     <div class="form-group">
28       <label
  for="exampleInputEmail1">Item Name
  </label>
29       <input type="text"
  name="name" class="form-control">

```

```

required&gt;
30         &lt;/div&gt;
31         &lt;div class=&quot;form-group&quot;&gt;
32             &lt;label
for=&quot;exampleInputEmail1&quot;&gt;Price &lt;/label&gt;
33             &lt;input name=&quot;price&quot;
type=&quot;number&quot; class=&quot;form-control&quot;
required&gt;
34             &lt;/div&gt;
35         &lt;div class=&quot;form-group&quot;&gt;
36             &lt;label
for=&quot;exampleInputPassword1&quot;&gt;Quantity
&lt;/label&gt;
37             &lt;input name=&quot;quantity&quot;
class=&quot;form-control&quot; type=&quot;number&quot;
required&gt;
38             &lt;/div&gt;
39             &lt;button type=&quot;submit&quot; class=&quot;btn
btn-success&quot;&gt;Add Data ???&lt;/button&gt;
40         &lt;/form&gt;
41         &lt;hr&gt;
42         &lt;div class=&quot;class mt-5&quot;&gt;
43             &lt;form action=&quot;&quot;&gt;
44                 &lt;button class=&quot;btn btn-primary&quot;&gt;
&lt;a class=&quot;ok&quot; href=&quot;{% url 'pdf'
%}&quot;&gt;Generate Bill &lt;/a&gt;&lt;/button&gt;
45                 &lt;button class=&quot;btn btn-danger&quot;&gt;
&lt;a class=&quot;ok&quot; href=&quot;{% url 'logout'
%}&quot;&gt;Logout &lt;/a&gt;&lt;/button&gt;
46             &lt;/form&gt;
47
48         &lt;table class=&quot;table mt-6&quot;&gt;
49             &lt;thead&gt;
50                 &lt;tr&gt;
51                     &lt;th scope=&quot;col&quot;&gt;S.No.
&lt;/th&gt;
52                     &lt;th scope=&quot;col&quot;&gt;Item
name &lt;/th&gt;
53                     &lt;th scope=&quot;col&quot;&gt;Price
&lt;/th&gt;
54                     &lt;th
scope=&quot;col&quot;&gt;Quantity &lt;/th&gt;
55                     &lt;th scope=&quot;col&quot;&gt;Total
&lt;/th&gt;
56                     &lt;th
scope=&quot;col&quot;&gt;Actions ???&lt;/th&gt;

```

```

        </tr>
58     </thead>
59     <tbody>
60         {% for receipt in receipts %}
61         <tr>
62             <th scope="row">
        {{forloop.counter}}</th>
63             <td>{{receipt.name}}</td>
64             <td> &#x20B9;
        {{receipt.price}}</td>
65             <td>
        {{receipt.quantity}}</td>
66             <td> &#x20B9;
        {{receipt.total}}</td>
67             <td>
68                 <a
        href="/delete_receipt/{{receipt.id }}">
        class="btn btn-danger m-2">Delete </a>
69                 <a
        href="/update_receipt/{{receipt.id }}">
        class="btn btn-success">Update </a>
70             </td>
71         </tr>
72         {% endfor %}
73     </tbody>
74     <th colspan="4"
        class="text-right">Total Sum :</th>
75     <td>{{ total_sum }}</td>
76     </tr>
77 </tbody>
78 </table>
79 </div>
80 {% endblock %}

```

update_receipt.html: This HTML template is for updating data in a web application. It includes a form for editing details related to an item, with fields pre-filled using existing data. The page uses Bootstrap for styling and some custom CSS. Users can update the data by clicking the **"Update Data"** button.

HTML



```

1 {% extends "base.html" %}
2 {% block start %}

```



```

4  <link
    href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2
    /css/bootstrap.min.css"
    rel="stylesheet">
5  <style>
6      .text{
7          color: green;
8          font-weight: bold;
9          font-family: 'Times New Roman', Times, serif;
10     }
11 </style>
12
13 <div class="container mt-5 col-5">
14
15     <form class="col-6 mx-auto card p-3 shadow-
    lg" method="post"
    enctype="multipart/form-data">
16         {% csrf_token %}
17         <h3 class="text text-center">
    GeeksforGeeks </h3>
18
19         <h4 style="font-family: 'Times New Roman',
    Times, serif; font-size:20px;">Update Data
    </h4>
20
21         <div class="form-group">
22             <label for="exampleInputEmail1">Item
    name </label>
23             <input type="text"
    name="name" value="{{receipt.name}}"
    class="form-control" required>
24         </div>
25         <div class="form-group">
26             <label
    for="exampleInputEmail1">Price </label>
27             <input name="price"
    type="number" value="
    {{receipt.price}}" type="text"
    class="form-control" required>
28         </div>
29         <div class="form-group">
30             <label
    for="exampleInputPassword1">Quantity
    </label>

```

```

        <input name="quantity"
type="number" value="
{{receipt.quantity}}" class="form-control
32         required
33         <br>
34
35         <button type="submit" class="btn
btn-danger">Update Data </button>
36     </form>
37
38
39 </div>
40
41 {% endblock %}

```

pdf.html: This HTML template creates a receipt for a purchase with the following features:

- It includes Bootstrap for styling and custom CSS.
- Displays company details, item information, and a total sum.
- Provides a "Generate Receipt" button that triggers the print functionality when clicked.
- JavaScript is used to extract the receipt content and initiate printing.

The page offers a printable receipt for users to save or use for their records.

HTML

```

1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
7      <title>Receipt</title>
8
9      <!-- Add Bootstrap CSS Link -->
10     <link
href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.2/dis

```

```
t/css/bootstrap.min.css"
rel="stylesheet">
11     <style>
12         .invoice {
13             text-align: center;
14             margin-left: 33%;
15         }
16
17         .o123 {
18             margin-top: 3%;
19             margin-left: 1%;
20         }
21
22         .company {
23             margin-left: 80%;
24             margin-top: -20%;
25         }
26
27         .amount {
28             margin-left: 76%;
29         }
30
31         .total {
32             margin-left: 70%;
33         }
34
35         .sign {
36             margin-top: 3%;
37             margin-left: -13%;
38         }
39
40         .receipt-container {
41             padding: 20px;
42             border: 2px solid #000;
43             border-radius: 10px;
44             background-color: #fff;
45             margin-top: 30px;
46         }
47
48         .receipt-header {
49             text-align: center;
50         }
51
```



```
53     .receipt-title {
54         font-size: 24px;
55         color: #333;
56     }
57     .receipt-details {
58         font-size: 16px;
59         color: #555;
60         margin-top: 20px;
61     }
62
63     .receipt-table {
64         width: 100%;
65         margin-top: 20px;
66         border-collapse: collapse;
67     }
68
69     .receipt-table th,
70     .receipt-table td {
71         border: 1px solid #ddd;
72         padding: 8px;
73         text-align: left;
74     }
75
76     .receipt-total {
77         font-size: 20px;
78         font-weight: bold;
79         color: #333;
80         margin-top: 20px;
81     }
82
83     .receipt-footer {
84         margin-top: 20px;
85         font-size: 16px;
86         color: #555;
87     }
88
89     .btn {
90         margin-left: 48%;
91         margin-top: 2%;
92     }
93
94
```

```

        h2 {
96             color: green;
97
98         }
99
100        img {
101            width: 200px;
102            height: 150px;
103        }
104        #date{
105            margin-left: 120px;
106
107        }
108        </style>
109    </head>
110
111    <body>
112
113        <div id="mn" class="container
receipt-container col-8">
114            <div class="receipt-header">
115                
117            </div>
118
119
120            <div class="row" >
121                <div class="col-xl-10"
id="date">
122
123                    <br><br>
124                    <br><br>
125                    <ul class="list-unstyled company
float-end">
126                        <li style="font-size: 20px;
" > <b>Address :</b></li>
127                        <li>Noida,Uttar
Pradesh</li>
128                        <li>1800 258 4458</li>
129                        <li>geeksforgeeks.org</li>
130                    </ul>
131                </div>

```

```

133         </div>
134         <br><br>
135         <table class='receipt-table'>
136             <thead>
137                 <tr>
138                     <th>Description</th>
139                     <th>Amount/Piece</th>
140                     <th>Quantity</th>
141                     <th>Total</th>
142             </thead>
143             <tbody>
144
145                 {% for receipt in receipts %}
146                 <tr>
147
148                     <td>{{receipt.name}}</td>
149                     <td> &#x20B9;
150                     {{receipt.price}}</td>
151                     <td>{{receipt.quantity}}</td>
152                     <td> &#x20B9;
153                     {{receipt.total}}</td>
154
155                 </tr>
156                 {% endfor %}
157
158             </tbody>
159         </table>
160         <p class='receipt-total'>Total Sum:
161         &#x20B9;{{ total_sum }}</p>
162         <div class='receipt-footer'>
163             <p>Thank you for your purchase!</p>
164         </div>
165
166         <button onclick='myfun('mn')'
167         type='button' class='btn btn-success btn1
168         &#x20B9;Generate Receipt</button>
169
170         <!-- Add Bootstrap JS and jQuery -->
171         <script src='https://code.jquery.com/jquery-
172         3.5.1.slim.min.js'></script>

```

```
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.2/dist
/js/bootstrap.min.js"></script>
171    <script type="text/javascript">
172        function myfun(pa) {
173            let d = document.getElementById(pa).innerHTML;
174            document.body.innerHTML = d;
175            window.print();
176        }
177    </script>
178 </body>
179
180 </html>
```

base.html: In this html file we write the code of the table and some basic margin padding which we apply in some html files for that we joint some html file with base.html file

HTML

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
7      <title>{{page}}</title>
8
9      <style>
10         table {
11             width: 80%;
12             margin: 20px auto;
13             border-collapse: collapse;
14         }
15
16         th,
17         td {
18             padding: 10px;
19             text-align: left;
20             border: 1px solid #ccc;
```

```
    }
22
23     th {
24         background-color: #f2f2f2;
25     }
26
27     tr:nth-child(even) {
28         background-color: #f2f2f2;
29     }
30
31     tr:hover {
32         background-color: #ddd;
33     }
34     </style>
35 </head>
36
37 <body>
38
39     {% block start %}
40     {% endblock %}
41
42     <script>
43         console.log('Hey Django')
44     </script>
45 </body>
46
47 </html>
```

urls.py : below are the urls.py file those connect html file with views.py file

Python



```
1  from django.contrib import admin
2  from django.urls import path
3  from home import views
4
5  urlpatterns = [
6      path('logout/', views.custom_logout, name="logout"),
7      path('pdf/', views.pdf , name='pdf'),
8      path('admin/', admin.site.urls),
9      path('login/' , views.login_page, name='login'),
10     path('register/', views.register_page, name='register'),
```

```
12     path('', views.receipts, name='receipts'),
13     path('update_receipt/<id>', views.update_receipt,
14           name='update_receipt'),
15     path('delete_receipt/<id>', views.delete_receipt,
16           name='delete_receipt'),
17 ]
```

admin.py: Here we are registering our models.

Python



```
1 from django.contrib import admin
2 from .models import *
3 from django.db.models import Sum
4
5 admin.site.register(Receipt)
```

Deployment of the Project

Run these commands to apply the migrations:

```
python3 manage.py makemigrations
```

```
python3 manage.py migrate
```


Run the server with the help of following command:


```
python3 manage.py runserver
```

Output

GeeksforGeeks


Register


Register 

[Log In](#) 

GeeksforGeeks

Login

Login 


[Create an Account](#) 

GeeksforGeeks

Create Receipt


Price

Quantity


Add Data 

Generate Bill

Logout

S.No.	Item name	Price	Quantity	Total	Actions 
1	DS/Other	₹50	2	₹100	<div>DeleteUpdate</div>

base.html



Address :
Noida,Uttar Pradesh
1800 258 4458
geeksforgeeks.org

Description	Amount/Piece	Quantity	Total
DS/Other	₹ 50	2	₹ 100
Data Structure & Algorithm	₹ 300	2	₹ 600
Machine Learning Project	₹ 350	1	₹ 350
Advanced DSA	₹ 500	2	₹ 1000

Total Sum: ₹2050
Thank you for your purchase!

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!



Next Article

Recipe Meal Planner using Django

Similar Reads

Image Gallery GUI using Django

Django, a powerful web framework for Python, makes it relatively easy to create a robust and user-friendly Image Gallery Graphical User Interface (GUI) for your...

4 min read

Python | Simple GUI calculator using Tkinter

Prerequisite: Tkinter Introduction, lambda function Python offers multiple options for developing a GUI (Graphical User Interface). Out of all the GUI methods,...

6 min read

Python | ToDo GUI Application using Tkinter

Prerequisites : Introduction to tkinter Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, Tkinter is...

5 min read

GUI chat application using Tkinter in Python

Chatbots are computer program that allows user to interact using input methods. The Application of chatbots is to interact with customers in big enterprises or...

7 min read

Build a Basic Form GUI using CustomTkinter module in Python

In this article, we are going to learn how to build a simple form GUI using CustomTkinter in Python. CustomTkinter: It is an extension of the Tkinter modul...

7 min read

Build GUI Application Pencil Sketch from Photo in Python

In this article, we will cover how to convert images to watercolor sketches and pencil sketch Linux. Sketching and painting are used by artists of many disciplin...

5 min read

Create a new Django project in Pycharm using Pycharm Terminal

PyCharm is one of the most popular Python-IDE developed by JetBrains used for performing scripting in Python language. PyCharm provides many useful featur...

2 min read

A Guide to Sending Data Using Cache in Django

In this article, we aim to comprehensively explore the functionality of sending data using cache in Django by providing a well-structured example...

6 min read

Blog Post Recommendation using Django

In this article, we will guide you through the creation of a blog post recommendation system using Django. Our article covers the integration of a...

10 min read

Count number of Faces from Image using Django

This article outlines the process of creating a face-counting feature in images through a Simple Django Web App. The project allows users to count the numb...

6 min read

Article Tags :

[Django](#)

[Geeks Premier League](#)

[Project](#)

[Geeks Premier League 2023](#)



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
Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

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

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

System Design

High Level Design

Languages

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

Interview Preparation

Competitive Programming

Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

GeeksforGeeks Videos

DSA
Python
Java
C++
Web Development
Data Science
CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved