



# Calculate CGPA using Python Django

Last Updated : 06 Feb, 2024

---

This article demonstrates how to build a CGPA calculator and display the results using Django. Initially, we establish a login system to ensure security. Users must first register for an account and then log in to access the calculator. After logging in, users can input their subject names, CGPA, and credits, essential information typically obtained from their college. Once all data is entered, clicking on the "Generate Result" button will display the result, which can be printed in PDF format.

## What is CGPA Calculator ?

A CGPA (Cumulative Grade Point Average) calculator is a tool used to determine a student's overall academic performance based on the grades obtained in various courses or subjects over a specific period. It assigns a numerical value, known as the Grade Point, to each grade received, and the CGPA is calculated as the average of these grade points.

## Calculate College CGPA and Print Results using Django

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

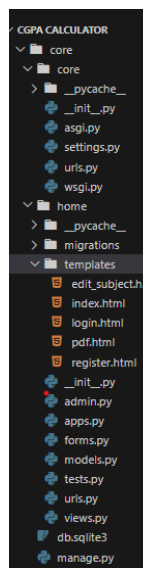
```
python manage.py startapp home
```

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",  
]
```

Django Views Model Template Forms Jinja Python SQLite Flask Json Postman Interview Ques

## File Structure



## Setting Necessary Files

**models.py** : This Django model, named "Subject," includes fields for user (linked to the built-in User model), subject name, grade (with predefined choices), and credit. The '.\_\_str\_\_' method customizes the string representation of an instance.

### Python3

```
1 from django.db import models  
2 from django.contrib.auth.models import User  
3
```

```

class Subject(models.Model):
    user = models.ForeignKey(User,
on_delete=models.SET_NULL, null=True, blank=True)
    name = models.CharField(max_length=255)
    grade = models.CharField(max_length=2, choices=[('S',
'S'), ('A', 'A'), ('B', 'B'), ('C', 'C'), ('D', 'D'), ('F',
'F')])
    credit = models.PositiveIntegerField()

    def __str__(self):
        return f"{self.name} - Grade: {self.grade}, Credit:
{self.credit}"

```

**views.py :** In below code, Django application includes views for a CGPA calculator, subject editing, subject deletion, and a result view. It also has user authentication functionalities, including a login page, registration page, and a logout function. The CGPA calculator computes the CGPA based on subjects and their grades, while the result view displays the result in a PDF template. User authentication is implemented with login, registration, and logout functionalities. Error messages are displayed for incorrect login attempts or registration issues.

### Python3

```

1 from django.shortcuts import render, redirect,
  get_object_or_404
2 from .models import Subject
3 from .forms import SubjectForm
4 from django.http import HttpResponse, JsonResponse,
  HttpResponseRedirect
5 from django.contrib import messages
6 from django.contrib.auth import login, authenticate
7 from django.contrib.auth.decorators import login_required
8 from django.contrib.auth.models import User
9 from django.contrib.auth import logout
10
11 # Mapping of letter grades to numeric values
12 GRADE_POINTS = {'A': 9, 'S': 10, 'B': 8, 'C': 7, 'D': 6,
  'F': 0}
13 @login_required(login_url='/login/')
14 def cgpa_calculator(request):

```

```
subjects = Subject.objects.all()
16 form = SubjectForm()
17
18 if request.method == 'POST':
19     form = SubjectForm(request.POST)
20     if form.is_valid():
21         form.save()
22         return redirect('cgpa_calculator')
23
24 # Calculate CGPA
25 total_credits = 0
26 total_grade_points = 0
27
28 for subject in subjects:
29     total_credits += subject.credit
30     total_grade_points += subject.credit *
GRADE_POINTS.get(subject.grade, 0)
31
32 if total_credits != 0:
33     cgpa = total_grade_points / total_credits
34 else:
35     cgpa = 0
36
37 context = {
38     'subjects': subjects,
39     'form': form,
40     'cgpa': cgpa,
41 }
42
43 return render(request, 'index.html', context)
44
45
46 @login_required(login_url='/login/')
47 def edit_subject(request, subject_id):
48     subject = get_object_or_404(Subject, id=subject_id)
49
50     if request.method == 'POST':
51         form = SubjectForm(request.POST, instance=subject)
52         if form.is_valid():
53             form.save()
54             return redirect('cgpa_calculator')
55     else:
56         form = SubjectForm(instance=subject)
```

```
58     context = {
59         'form': form,
60         'subject_id': subject_id,
61     }
62
63     return render(request, 'edit_subject.html', context)
64
65 @login_required(login_url='/login/')
66 def delete_subject(request, subject_id):
67     subject = get_object_or_404(Subject, id=subject_id)
68     subject.delete()
69     return redirect('cgpa_calculator')
70
71
72 @login_required(login_url='/login/')
73 def result(request):
74     subjects = Subject.objects.all()
75     form = SubjectForm()
76
77     if request.method == 'POST':
78         form = SubjectForm(request.POST)
79         if form.is_valid():
80             form.save()
81             return redirect('cgpa_calculator')
82
83     # Calculate CGPA
84     total_credits = 0
85     total_grade_points = 0
86
87     for subject in subjects:
88         total_credits += subject.credit
89         total_grade_points += subject.credit *
GRADE_POINTS.get(subject.grade, 0)
90
91     if total_credits != 0:
92         cgpa = total_grade_points / total_credits
93     else:
94         cgpa = 0
95
96     context = {
97         'subjects': subjects,
98         'form': form,
```

```
        'cgpa': cgpa,
100     }
101
102     return render(request, 'pdf.html', context)
103
104
105 #login page for user
106 def login_page(request):
107     if request.method == "POST":
108         try:
109             username = request.POST.get('username')
110             password = request.POST.get('password')
111             user_obj =
User.objects.filter(username=username)
112             if not user_obj.exists():
113                 messages.error(request, "Username not
found")
114                 return redirect('/login/')
115             user_obj = authenticate(username=username,
password=password)
116             if user_obj:
117                 login(request, user_obj)
118                 return redirect('receipts')
119                 messages.error(request, "Wrong Password")
120                 return redirect('/login/')
121             except Exception as e:
122                 messages.error(request, "Something went
wrong")
123                 return redirect('/register/')
124         return render(request, "login.html")
125
126
127 #register page for user
128 def register_page(request):
129     if request.method == "POST":
130         try:
131             username = request.POST.get('username')
132             password = request.POST.get('password')
133             user_obj =
User.objects.filter(username=username)
134             if user_obj.exists():
135                 messages.error(request, "Username is
taken")
136                 return redirect('/register/')
```

```

        user_obj =
    User.objects.create(username=username)
138         user_obj.set_password(password)
139         user_obj.save()
140         messages.success(request, "Account created")
141         return redirect('/login')
142     except Exception as e:
143         messages.error(request, "Something went
wrong")
144         return redirect('/register')
145     return render(request, "register.html")
146
147
148 #logout function
149 def custom_logout(request):
150     logout(request)
151     return redirect('login')

```

**forms.py** : The "SubjectForm" in Django is a ModelForm linked to the "Subject" model. It includes fields for subject name, credit, and grade, enabling easy creation and modification of Subject instances through user interfaces.

### Python3

```

1  from django import forms
2  from .models import Subject
3
4  class SubjectForm(forms.ModelForm):
5      class Meta:
6          model = Subject
7          fields = ['name', 'credit', 'grade']

```

**project/urls.py** : The main 'urls.py' file in this Django project includes paths for the admin interface and includes URLs from the 'home.urls' module using the 'include' function, promoting modularity and organization in the project's URL configuration.

### Python3

```

1  from django.contrib import admin

```

```
from django.urls import include, path

3
4 urlpatterns = [
5     path('admin/', admin.site.urls),
6     path('', include('home.urls')),
7 ]
```

**app/urls.py** : The 'urls.py' file in the 'home' app of this Django project includes paths for views like 'cgpa\_calculator', 'edit\_subject', 'delete\_subject', 'result', 'login\_page', 'custom\_logout', and 'register\_page'. These paths correspond to various functionalities, defining the URL patterns for navigation within the application.

### Python3

```
1 from django.urls import path
2 from .views import *
3
4 urlpatterns = [
5     path('', cgpa_calculator, name='cgpa_calculator'),
6     path('edit/<int:subject_id>/', edit_subject,
7         name='edit_subject'),
8     path('delete/<int:subject_id>/', delete_subject,
9         name='delete_subject'),
10    path('result/', result, name='result'),
11
12    path('login/' , login_page, name='login'),
13    path('logout/', custom_logout, name="logout"),
14    path('register/', register_page, name='register'),
15 ]
```

## Creating GUI

**login.html** : This HTML file implements a simple login form for a CGPA Calculator web application using Bootstrap. It includes fields for username and password, with messages displayed for login outcomes. The form is styled



using Bootstrap classes for a clean and responsive design, and there's a link to the registration page.

## 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/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmL
      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>CGPA Calculator</title>
12
13 </head>
14 <body>
15     <br>
16     <br>
17     <br>
18
19     <div class="container mt-4 bg-white col-md-3 card
      shadow p-3 " id="log">
20         <div class="login-form">
21             {% if messages %}
22             {% for message in messages %}
23                 <div class="alert alert-success {{ message.tags
                }} mt-4" role="alert">
24                     {{ message }}
25                 </div>
26             {% endfor %}
27             {% endif %}
28             <form action="" method="post">
29                 {% csrf_token %}
```

```

    <h4 class="text-center" style="font-family:
'Times New Roman', Times, serif;"> Login </h4>
31     <div class="form-group">
32         <input type="text" class="form-control"
name="username" placeholder="Username" required
33             style="background-color: #fff;
border: 1px solid #ddd; border-radius: 5px; padding: 10px;">
34     </div>
35     <div class="form-group mt-2">
36         <input type="password" class="form-
control" name="password" placeholder="Password" required
37             style="background-color: #fff;
border: 1px solid #ddd; border-radius: 5px; padding: 10px;">
38     </div>
39     <div class="form-group mt-2">
40         <button class="btn btn-light btn-block"
style="margin-left: 138px;border: 0.5px solid
black">Login</button>
41     </div>
42     <br>
43 </form>
44     <p class="text-center" style="color: #555;"><a
href="{% url 'register' %}" style="color: #007bff;">Create
an
45         Account.</a></p>
46     </div>
47 </div>
48
49 </body>
50
51 </html>

```

**register.html** : This HTML file implements a registration form for a CGPA Calculator web application using Bootstrap. It includes fields for username and password, with messages displayed for registration outcomes. The form is styled using Bootstrap classes for a clean and responsive design, and there's a link to the login page for existing users. The "Register" button has a style attribute for additional customization.

## HTML



```
1 <!doctype html>
```



```

1  <html lang="en">
2
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/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfSpd3yD65VohhpU0mL
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>CGPA Calculator</title>
12 </head>
13 <body>
14     <body>
15         <br><br><br><br>
16
17         <div class="container mt-4 bg-white mx-auto col-
md-3 card shadow p-3">
18             <div class="login-form">
19                 {% if messages %}
20                 {% for message in messages %}
21                     <div class="alert alert-success {{
message.tags }}" role="alert" >
22                         {{ message }}
23                     </div>
24                 {% endfor %}
25                 {% endif %}
26                 <form action="" method="post">
27                     {% csrf_token %}
28                     <h4 class="text-center" style="font-
family: 'Times New Roman', Times, serif;"> Register </h4>
29                     <div class="form-group">
30                         <input type="text" class="form-
control" name="username" placeholder="Username" required>
31                     </div>
32                     <div class="form-group mt-2">
33                         <input type="password" class="form-
control" name="password" placeholder="Password" required>
34                     </div>

```

```

36         <div class="form-group mt-2">
            <button class="btn btn-light btn-
block" style="margin-left: 117px;border: 0.5px solid
black">Register</button>
37         </div>
38         <br>
39     </form>
40     <p class="text-center" ><a href="{% url
'login' %}">Log In </a></p>
41 </div>
42 </div>
43
44 </body>
45
46 </html>

```

**index.html** : This HTML file provides a clean and user-friendly web interface for a CGPA Calculator. It includes a form to add subjects, a table to display existing subjects with their grades and credits, and buttons for editing or deleting subjects. The styling is done using a combination of inline styles and embedded CSS. Additionally, there's a logout button at the bottom for user convenience.

## HTML



```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width,
initial-scale=1.0">
6      <title>CGPA Calculator</title>
7  </head>
8  <style>
9      body {
10         font-family: Arial, sans-serif;
11         margin: 20px;
12     }
13     #calculator {
14         max-width: 600px;
15         margin: 0 auto;

```

```
padding: 20px;
border: 1px solid #ccc;
border-radius: 5px;
}

h1{
    color: #333;
    text-align: center;
    font-family: 'Times New Roman', Times, serif;
}

form {
    margin-bottom: 20px;
}

button {
    padding: 10px;
    color: black;
    border: none;
    font-size: 13px;
    font-weight: bold;
    border-radius: 3px;
    cursor: pointer;
}

table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 20px;
}

th, td {
    border: 1px solid #ddd;
    padding: 10px;
    text-align: left;
}

th {
    background-color: #d9dad9d7;
    color: rgb(97, 93, 93);
}

#cgpa {
    font-weight: bold;
}

a{
    color: black;
}

</style>
```

```

<body>
60 <div id="calculator">
61     <h1 style="color: green;">GeeksforGeeks</h1>
62     <h2 style="font-family: 'Times New Roman', Times, serif;
text-align:center;">CGPA Calculator</h2>
63     <br>
64     <form method="post" action="{% url 'cgpa_calculator'
%}">
65         {% csrf_token %}
66         {{form.as_p}}
67         <button type="submit">Add Subject</button>
68         <button> <a href="{% url 'result' %}">Generate
Result</a></button>
69     </form>
70     <br>
71     <table>
72         <thead>
73             <tr>
74                 <th>Subject</th>
75                 <th>Grade</th>
76                 <th>Credit</th>
77                 <th>Actions</th>
78             </tr>
79         </thead>
80         <tbody>
81             {% for subject in subjects %}
82                 <tr>
83                     <td style="font-family: 'Times New Roman',
Times, serif;">{{ subject.name }}</td>
84                     <td style="font-family: 'Times New Roman',
Times, serif;">{{ subject.grade }}</td>
85                     <td style="font-family: 'Times New Roman',
Times, serif;">{{ subject.credit }}</td>
86                     <td>
87                         <button><a href="{% url 'edit_subject'
subject.id %}">Edit</a></button>
88                         <button><a href="{% url 'delete_subject'
subject.id %}">Delete</a></button>
89                     </td>
90                 </tr>
91             {% endfor %}
92         </tbody>
93     </table>
94 </div>
95 <br>

```

```
<button style="margin-left: 400px;"> <a class="ok" href="{%  
url 'logout' %}">Logout </a></button>  
97 </body>  
98 </html>
```

**edit\_subject.html** : This HTML file offers a compact and centered layout for editing a subject. It includes a form with input fields, a "Save Changes" button linked to the 'edit\_subject' view, and a "Cancel" button redirecting to the 'cgpa\_calculator' view. Styling is kept minimal for a clean and focused editing experience.

## 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>Document</title>  
8 </head>  
9  
10 <body>  
11     <style>  
12         body {  
13             font-family: Arial, sans-serif;  
14             margin: 20px;  
15         }  
16  
17         #calculator {  
18             max-width: 400px;  
19             margin: 0 auto;  
20             padding: 20px;  
21             border: 1px solid #ccc;  
22             border-radius: 5px;  
23         }  
24  
25         h2 {  
26             font-family: 'Times New Roman', Times, serif;  
27     }
```

```

29         button {
30             padding: 10px;
31             color: black;
32             border: none;
33             font-size: 13px;
34             font-weight: bold;
35             border-radius: 3px;
36             cursor: pointer;
37         }
38
39         a {
40             color: black;
41         }
42     </style>
43
44
45     <div id="calculator">
46         <br>
47
48         <h2 style="text-align: center;">Edit Subject</h2>
49         <br>
50         <form method="post" action="{% url 'edit_subject'
subject_id=subject_id %}">
51             {% csrf_token %}
52             {{ form.as_p }}
53
54             <button type="submit">Save Changes</button>
55         </form>
56         <br>
57         <button><a href="{% url 'cgpa_calculator'
58         %}">Cancel</a></button>
59     </div>
60 </body>
61
62 </html>

```

**pdf.html** : This HTML file represents a CGPA Calculator result page with a table displaying subjects, grades, and credits. It calculates and displays the CGPA. The "Generate Receipt" button triggers a JavaScript function to copy the content of the 'calculator' div, open a new window, and print the receipt. The



styling is clean, and the jsPDF library is included for potential PDF generation, although not currently used in the provided code.

## 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>CGPA Calculator</title>
8     <!-- Add jsPDF library -->
9     <script
    src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.4.0/jspdf.umd.min.js"></script>
10 </head>
11 <style>
12     body {
13         font-family: Arial, sans-serif;
14         margin: 20px;
15     }
16     .calculator {
17         max-width: 600px;
18         margin: 0 auto;
19         padding: 20px;
20         border: 1px solid #ccc;
21         border-radius: 5px;
22     }
23
24     form {
25         margin-bottom: 20px;
26     }
27
28     table {
29         width: 100%;
30         border-collapse: collapse;
31         margin-top: 20px;
32     }
33
34     th,
35     td {
```

```

        border: 1px solid #ddd;
37     padding: 10px;
38     text-align: left;
39 }
40
41 th {
42     color: rgb(97, 93, 93);
43 }
44
45 #cgpa {
46     font-weight: bold;
47 }
48
49 button {
50     padding: 10px;
51     color: black;
52     border: none;
53     font-size: 13px;
54     font-weight: bold;
55     border-radius: 3px;
56     cursor: pointer;
57     margin-left: 655px;
58 }
59 </style>
60 <body>
61     <div id="mn" class="calculator" >
62         
63         <br>
64         <br>
65         <table>
66             <thead>
67                 <tr>
68                     <th>Subject</th>
69                     <th>Grade</th>
70                     <th>Credit</th>
71                 </tr>
72             </thead>
73             <tbody>
74                 {% for subject in subjects %}
75                 <tr>
76                     <td style="font-family: 'Times New
Roman', Times, serif;">{{ subject.name }}</td>

```

```

        <td style="font-family: 'Times New
Roman', Times, serif;">{{ subject.grade }}</td>
78        <td style="font-family: 'Times New
Roman', Times, serif;">{{ subject.credit }}</td>
79    </tr>
80    {% endfor %}
81    </tbody>
82    </table>
83    <h3 style="font-family: 'Times New Roman', Times,
serif;">CGPA: <span id="cgpa">{{cgpa}}</span></h3>
84    </div>
85    <br>
86    <button onclick="myfun('mn')" type="button" class="btn
btn-success btn1 ">Generate Receipt</button>
87    <script type="text/javascript">
88        function myfun(pa) {
89            let d = document.getElementById(pa).innerHTML;
90            document.body.innerHTML = d;
91            window.print();
92        }
93    </script>
94 </body>
95 </html>

```

## 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 :**

## Register

Register

[Log In](#)

## Login

Login

[Create an Account.](#)

## GeeksforGeeks

### CGPA Calculator

Name:


Credit:

Grade:

Add Subject

Generate Result

Subject	Grade	Credit	Actions
---------	-------	--------	---------



Subject	Grade	Credit
Science	S	4
Maths	B	5
English	A	3
Hindi	C	5

**CGPA: 8.352941176470589**

Generate Receipt

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!



harsh...



### Previous Article

Simple Calculator in Python Socket  
Programming

### Next Article

## Similar Reads

### Adding Tags Using Django-Taggit in Django Project

Django-Taggit is a Django application which is used to add tags to blogs, articles etc. It makes very easy for us to make adding the tags functionality to our django...

2 min read

## How to customize Django forms using Django Widget Tweaks ?

Django forms are a great feature to create usable forms with just few lines of code. But Django doesn't easily let us edit the form for good designs. Here, we...

3 min read

## Integrating Django with Reactjs using Django REST Framework

In this article, we will learn the process of communicating between the Django Backend and React js frontend using the Django REST Framework. For the sake...

15+ min read

## Styling Django Forms with django-crispy-forms

Django by default doesn't provide any Django form styling method due to which it takes a lot of effort and precious time to beautifully style a form. django-crispy...

1 min read

## How to Fix Django "ImportError: Cannot Import Name 'six' from..."

After Django 3.0, django.utils.six was removed. The "ImportError: Cannot Import Name 'six' from 'django.utils'" error occurs because of the dependency issue. Thi...

3 min read

## E-commerce Website using Tailwind and React using Django

Our e-commerce website, "ECOM," aims to offer users a smooth online shopping experience. We use Tailwind CSS for styling, React for interactive user interface...

6 min read

## Weather app using Django | Python

In this tutorial, we will learn how to create a Weather app that uses Django as backend. Django provides a Python Web framework based web framework that...

2 min read

## Change Object Display Name using \_\_str\_\_ function - Django Models | Python

How to Change Display Name of an Object in Django admin interface? Whenever an instance of model is created in Django, it displays the object as ModelName...

2 min read

## Python Django - Test Driven Development of Web API using DRF & Docker

We are going to create a to-do list web API using Django rest framework, docker and also going to write different tests for different functionalities in our code...

11 min read

## Creating multiple user types and using proxy models in Python Django

In this article, we will understand the concept of the proxy model and the implementation of multiple user types depending on the requirements of your...

9 min read

Article Tags :

[Python](#)[Django](#)[Python Django](#)

Practice Tags :

[python](#)

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](#)

### Languages

[Python](#)[Java](#)[C++](#)[PHP](#)[GoLang](#)

GFG Corporate Solution  
Placement Training Program  
GeeksforGeeks Community

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  
Physics

GeeksforGeeks Videos

DSA  
Python



Chemistry	Java
Biology	C++
Social Science	Web Development
English Grammar	Data Science
Commerce	CS Subjects
World GK	

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved