Generate a PDF in Django

Last Updated : 25 Sep, 2023

With the correct frameworks and an organized approach, it is simple to generate PDF files in a Django web application using **Python**. Following the techniques indicated in this article, you may enable users to download data in these popular formats, improving the usability and functionality of your **Django** application.

Generating a PDF file in Python and Django

Step 1: Create a **Virtual Environment**.

Step 2: Create a new **Django project** and app, and move to the folder.

```
django-admin startproject test_project
cd test_project
python manage.py startapp test
```

Step 3: Register your app

```python
# Application definition

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

Step 4: Create views.py

Here is the code explanation of the views:

- Import necessary modules.
- **home function**: It is a simple input form which take Title, Author and, Publish Year as an Input, that is saved in the database.
- **generate_pdf function**: This view function is called when a user requests a PDF. It uses generate_pdf_file to create a PDF and returns it as a downloadable attachment with the filename "book_catalog.pdf."
- **generate_pdf_file function**: Imports BytesIO from io to create an in-memory buffer for the PDF. Initializes a PDF canvas (p) on the buffer. Retrieves all books from the Book model. Adds "Book Catalog" as the PDF title at coordinates (100, 750). Iterates through the books: Adds book details (title, author, year) to the PDF at decreasing Y-coordinates. Each book's details are printed with a vertical offset of -20 units. Calls showPage() to end the current page and save() to finish creating the PDF. The buffer is reset and returned. It now contains the PDF content.

Python3

```python
# test/views.py
from django.http import FileResponse
from reportlab.pdfgen import canvas
from .models import Book
from django.shortcuts import redirect, render
from .forms import BookForm


def home(request):
    if request.method == 'POST':
        form = BookForm(request.POST)
        if form.is_valid():
            form.save()
    # Redirect to PDF generation after adding a book
            return redirect('home')
    else:
        form = BookForm()
    return render(request, 'myapp/create_user_profile.html',
                  {'form': form})


def generate_pdf(request):
    response = FileResponse(generate_pdf_file(),
                            as_attachment=True,
                            filename='book_catalog.pdf')
    return response
```

```python
def generate_pdf_file():
    from io import BytesIO

    buffer = BytesIO()
    p = canvas.Canvas(buffer)

    # Create a PDF document
    books = Book.objects.all()
    p.drawString(100, 750, "Book Catalog")

    y = 700
    for book in books:
        p.drawString(100, y, f"Title: {book.title}")
        p.drawString(100, y - 20, f"Author: {book.author}")
        p.drawString(100, y - 40, f"Year: {book.publication_year}")
        y -= 60

    p.showPage()
    p.save()

    buffer.seek(0)
    return buffer
```

Step 5: Create model.py

This code represents a simple [Django model](#) for storing information about books, including their title, author, and publication year.

Python3

```python
# test/models.py
from django.db import models

class Book(models.Model):
    title = models.CharField(max_length=100)
    author = models.CharField(max_length=100)
    publication_year = models.PositiveIntegerField()

    def __str__(self):
        return self.title
```

Step 6: Create a form.py

This code sets up a [Django form](#), BookForm, that is connected to the Book model and includes specific fields for creating or editing book instances with title, author, and publication year attributes.

Python3

```python
# test/forms.py
from django import forms
from .models import Book


class BookForm(forms.ModelForm):
    class Meta:
        model = Book
        fields = ['title', 'author', 'publication_year']
```

Step 7: Create test/urls.py

This is to set in the URLs of the app.

Python3

```python
# test/urls.py
from django.urls import path
from . import views

urlpatterns = [
    path('', views.home, name='home'),
    path('generate-pdf/', views.generate_pdf, name='generate_pdf'),
]
```

Step 8: Update urls.py

This is to set in the Urls of the project folder.

Python3

```python
from django.contrib import admin
from django.urls import path, include
```

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('test.urls')),
]
```

Step 9: Create a templates/create_user_profile.html

HTML

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Generate PDF</title>
</head>
<body>
    <h1>Generate and Download PDF</h1>
    <h2>Add a New Book</h2>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Add Book</button>
    </form>
    <p><a href="{% url 'generate_pdf' %}" download>Download Book Catalog (PDF)</a
<br>
    <!-- <p><a href="{% url 'generate_pdf' %}" download>Download Book Catalog (CSV

</body>
</html>
```

Step 10: Apply Migration

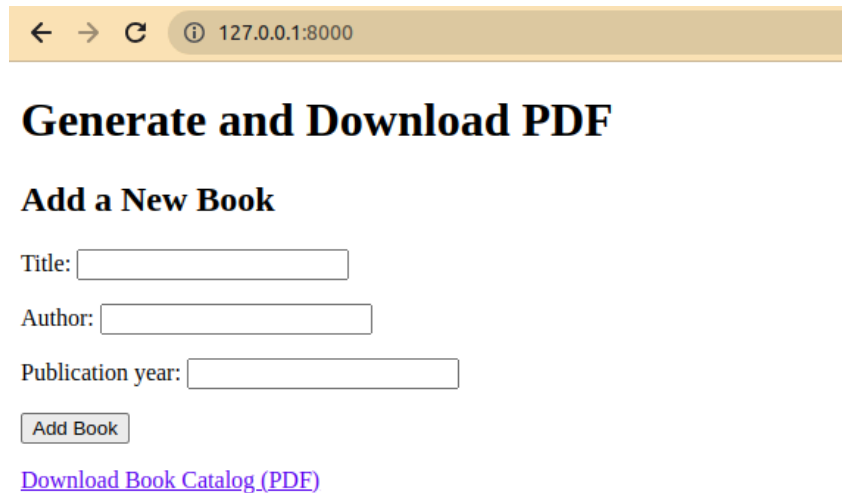By the following commands you can migrate the data into the database.

python manage.py makemigrations
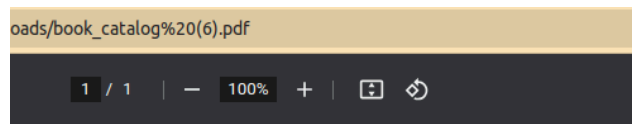
python manage.py migrate

Deploy project

python manage.py runserver

Output: Input form

The downloaded PDF

Learn in a distraction-free environment with refined, **high-quality content** and **35+ expert-led tech courses** to help you crack any interview. From programming languages and DSA to web development and data science, GeeksforGeeks Premium has you covered!

Choose GeeksforGeeks Premium today and also get access to **Unlimited**

A	amit...	GE

Next Article

Similar Reads

Send PDF File through Email using pdf-mail module

pdf_mail module is that library of Python which helps you to send pdf documents through your Gmail account. Installing Library This module does not come built-...

2 min read

Interact with PDF with PDF ChatBot

PDFs are widely used for sharing and viewing documents across various platforms and devices. Working with PDFs can sometimes be difficult and time-...

5 min read

Generate PDF in ElectronJS

ElectronJS is an Open Source Framework used for building Cross-Platform native desktop applications using web technologies such as HTML, CSS, and JavaScrip...

8 min read

Generate PDF File in Android using Kotlin

Most of the applications provide users with the facility of bills to be downloaded from the mobile applications. This type of application gets the data from the API...

7 min read

Generate PDF File in Android using Jetpack Compose

There are many apps in which data from the app is provided to users in the downloadable PDF file format. So in this case we have to create a PDF file from...

10 min read

How to generate PDF file using jsPDF library ?

In most web-based projects or applications, we need report generation for showing some relevant data which includes PDF conversion. There are many P...

3 min read

Generate .pdf from RMarkdown file with R

R is a popular programming language used in the field of data analysis and statistical computing. One of its key strengths is its ability to produce high-quali...

4 min read

How to Generate a PDF file in Android App?

There are many apps in which data from the app is provided to users in the downloadable PDF file format. So in this case we have to create a PDF file from...

6 min read

How to generate PDF file and add TrueType fonts using PHP ?

In this article, we will learn how to generate PDF files and add new TrueType fonts with PHP by using FPDF. It is a free PHP class that contains many function...

2 min read

How to generate PDF file using PHP ?

In this article, we will learn how to generate PDF files with PHP by using FPDF. It is a free PHP class that contains many functions for creating and modifying PDF...

2 min read

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

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

Software Development

Software Testing

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

Chemistry

Biology

Social Science

English Grammar

Commerce

World GK

GeeksforGeeks Videos

DSA

Python

Java

C++

Web Development

Data Science

CS Subjects