Django Code for generating PDF with the data which is we entered in form.

```
# models.py


class Resume(models.Model):
  FirstName= models.CharField(max_length=200)
  last_name = models.CharField( max_length=100)
  age=models.IntegerField()
  address=models.CharField(max_length=200)
  marks=models.DecimalField(max_digits=5, decimal_places=2)


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

Description: Defines a Django model `Resume` with fields for first name, last name, age, address, and marks.

```html
<!-- generate_pdf_form.html -->

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Generate PDF Form</title>
</head>
<body>
  <h1>Generate PDF Form</h1>
  <form action="{% url 'generate_pdf' %}" method="post">
    {% csrf_token %}
    <!-- Form fields for first name, last name, age, address, marks go here -->
```

```
        <label for="first_name">First Name:</label>
        <input type="text" id="first_name" name="first_name" required><br>

        <label for="last_name">Last Name:</label>
        <input type="text" id="last_name" name="last_name" required><br>

        <label for="age">Age:</label>
        <input type="number" id="age" name="age" required><br>

        <label for="address">Address:</label>
        <input type="text" id="address" name="address" required><br>

        <label for="marks">Marks:</label>
        <input type="number" id="marks" name="marks" required><br>

        <button type="submit">Generate PDF</button>
    </form>
</body>
</html>
```

Description: HTML template for a form to input data for generating a PDF resume.

```python
# views.py

def my_view(request):
    if request.method == 'POST':
        form = Myform(request.POST)
        if form.is_valid():
```

```
            # Create a new Resume object with form data
            resume = Resume(
                FirstName=form.cleaned_data['FirstName'],
                last_name=form.cleaned_data['last_name'],
                age=form.cleaned_data['age'],
                address=form.cleaned_data['address'],
                marks=form.cleaned_data['marks']
            )
            # Save the resume object to the database
            resume.save()
            # Redirect to the generate_pdf URL
            return redirect('generate_pdf')  # Assuming you named the URL pattern 'generate_pdf'
    else:
        form = Myform()
    return render(request, 'generate_pdf_form.html', {'form': form})


def success_page(request):
    return HttpResponse("Success! Your data has been submitted.")


def generate_pdf(request):
    if request.method == 'POST':
        # Extract form data
        first_name = request.POST.get('first_name')
        last_name = request.POST.get('last_name')
        age = request.POST.get('age')
        address = request.POST.get('address')
        marks = request.POST.get('marks')

        # Generate PDF
        pdf_buffer = generate_pdf_document(first_name, last_name, age, address, marks)
```

```
        if pdf_buffer:
            response = HttpResponse(pdf_buffer, content_type='application/pdf')
            response['Content-Disposition'] = 'attachment; filename="resume.pdf"'
            return response
        else:
            # If PDF generation failed, return an error response
            return HttpResponse("Error generating PDF", status=500)
    elif request.method == 'GET':
        # If accessed via GET, return a method not allowed response
        return HttpResponseNotAllowed(['POST'])



def generate_pdf_document(first_name, last_name, age, address, marks):
    buffer = io.BytesIO()
    pdf = canvas.Canvas(buffer)
    pdf.drawString(100, 750, "Resume")
    pdf.drawString(100, 700, f"First Name: {first_name}")
    pdf.drawString(100, 680, f"Last Name: {last_name}")
    pdf.drawString(100, 660, f"Age: {age}")
    pdf.drawString(100, 640, f"Address: {address}")
    pdf.drawString(100, 620, f"Marks: {marks}")
    pdf.save()
    buffer.seek(0)
    return buffer
```

Description: Defines Django views for rendering the form, processing form submissions, generating PDFs, and displaying a success page.

```python
# urls.py
```

```
from django.urls import path
from .views import my_view, success_page, generate_pdf

urlpatterns = [
    path('my_view/', my_view, name='myview'),
    path('generate_pdf/', generate_pdf, name='generate_pdf'),
    path('success_page/', success_page, name='success_page'),
]
```

Description: Maps URL patterns to corresponding view functions for the application.