**Project Setup and Roadmap**

1. Since I am developing my project inside the virtual environment so first let’s create a virtual environment using the command

*python -m venv name\_of\_venv (in our case I have set it as venv only)*

1. Install restframework library with the help of pip inside venv

*pip install django djangorestframework psycopg2-binary*

*pip install psycopg*

1. Install all the necessary libraries using pip

*pip freeze > requirements.txt*

1. Now we start the project using the command and change directory to our project

django-admin startproject ResumeParser

cd ResumeParser

1. Create a jango app ‘resume’ using the command

*python manage.py startapp resume*

1. Create a new database using the postgresql and grant all permissions to it

*psql -U postgres*

*CREATE DATABASE hariom;*

*GRANT ALL PRIVILEGES ON DATABASE hariom TO resume\_parser;*

*GRANT ALL PRIVILEGES ON SCHEMA public TO resume\_parser;*

1. We have to update the setting.py in the ResumeParser

* *In INSTALLED\_APPS we have to add the 'resume', 'rest\_framework',*
* *In the DATABASES we have to update the name, user and password as created.*

1. Create a class Candidate inside the models.py of resume app having the fields

first\_name, email, mobile\_number

*from django.db import models*

*class Candidate(models.Model):*

*first\_name = models.CharField(max\_length=100)*

*email = models.EmailField()*

*mobile\_number = models.CharField(max\_length=15)*

1. Create a Serializer for the Candidate model in the resume app using the rest framework

*from rest\_framework import serializers*

*from .models import Candidate*

*class CandidateSerializer(serializers.ModelSerializer):*

*class Meta:*

*model = Candidate*

*fields = ['first\_name', 'email', 'mobile\_number']*

1. Make migrations and migrate to ensure all the changes made

*python manage.py makemigrations resume*

*python manage.py migrate*

1. Create a view.py file to handle resume extraction

from django.http import HttpResponse

from rest\_framework.response import Response

from rest\_framework.views import APIView

from rest\_framework import status

import os

import logging

import re

from django.conf import settings

from .models import Candidate

from .serializers import CandidateSerializer

import spacy

import pdfplumber

from django.shortcuts import render

from docx import Document

logger = logging.getLogger(\_\_name\_\_)

def homepage(request):

    return render(request, 'homepage.html')

class ResumeExtractView(APIView):

    def post(self, request):

        file = request.FILES.get('resume')

        if not file:

            return Response({'error': 'No file uploaded'}, status=status.HTTP\_400\_BAD\_REQUEST)

        # Ensure MEDIA\_ROOT directory exists

        if not os.path.exists(settings.MEDIA\_ROOT):

            os.makedirs(settings.MEDIA\_ROOT)

        file\_path = os.path.join(settings.MEDIA\_ROOT, file.name)

        try:

            # Save the file temporarily

            with open(file\_path, 'wb') as f:

                for chunk in file.chunks():

                    f.write(chunk)

            # Load spaCy model

            nlp = spacy.load('en\_core\_web\_sm')

            # Process the resume based on file type

            text = ''

            first\_name = ''

            if file.name.lower().endswith('.pdf'):

                # Handle PDF files

                with pdfplumber.open(file\_path) as pdf:

                    for page in pdf.pages:

                        # Extract text, font size, and other text properties

                        for char in page.chars:

                            if char['size'] > 14:

                                first\_name += char['text']

                        text += page.extract\_text()

            elif file.name.lower().endswith('.docx'):

                # Handle DOCX files

                doc = Document(file\_path)

                for para in doc.paragraphs:

                    for run in para.runs:

                        if run.font.size and run.font.size.pt > 14:

                            first\_name += run.text

                    text += para.text

            else:

                # Unsupported file type

                return Response({'error': 'Unsupported file type. Please upload a PDF or DOCX file.'}, status=status.HTTP\_400\_BAD\_REQUEST)

            # Apply spaCy NLP processing

            doc = nlp(text)

            # Extract other details like email and mobile number

            email = ''

            mobile\_number = ''

            # Regex patterns

            email\_pattern = re.compile(r'[a-zA-Z0-9.\_%+-]+@gmail\.com')

            phone\_pattern = re.compile(r'\b\d{10}\b')

            # Extract email and phone number using regex

            email\_matches = email\_pattern.findall(text)

            if email\_matches:

                email = email\_matches[0]

            phone\_matches = phone\_pattern.findall(text)

            if phone\_matches:

                mobile\_number = phone\_matches[0]

            # Create a Candidate object

            candidate = Candidate.objects.create(

                first\_name=first\_name.strip()[:100],

                email=email,

                mobile\_number=mobile\_number

            )

            # Serialize the Candidate object

            serializer = CandidateSerializer(candidate)

            return Response(serializer.data, status=status.HTTP\_201\_CREATED)

        except Exception as e:

            return Response({'error': f'Internal server error: {e}'}, status=status.HTTP\_500\_INTERNAL\_SERVER\_ERROR)

1. Add URL Routing by creating a urls.py inside the resume app

from django.urls import path

from .views import ResumeExtractView, homepage

urlpatterns = [

    path('', homepage, name='homepage'),

    path('api/extract\_resume/', ResumeExtractView.as\_view(), name='extract\_resume'),

]

Include the app URLs in the main ResumeParser/urls.py

*from django.contrib import admin*

*from django.conf import settings*

*from django.urls import path, include*

*from django.conf.urls.static import static*

*from resume.views import homepage  # import your new view*

*urlpatterns = [*

*path('admin/', admin.site.urls),*

*path('', include('resume.urls')),  # This includes the URLs from the resume app*

*]*

*if settings.DEBUG:*

*urlpatterns += static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)*

1. I have created a homepage for uploading resume using HTML and CSS

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Upload Resume</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            background-color: #acf9e2;

            margin: 0;

            padding: 0;

            display: flex;

            justify-content: center;

            align-items: center;

            height: 100vh;

        }

        .container {

            background-color: white;

            padding: 20px;

            border-radius: 8px;

            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

            max-width: 400px;

            width: 100%;

            text-align: center;

        }

        h1 {

            color: #333;

            font-size: 24px;

            margin-bottom: 20px;

        }

        form {

            display: flex;

            flex-direction: column;

            align-items: center;

        }

        input[type="file"] {

            padding: 10px;

            border-radius: 4px;

            border: 1px solid #ccc;

            margin-bottom: 20px;

            width: 100%;

        }

        button {

            padding: 10px 20px;

            background-color: #4CAF50;

            color: white;

            border: none;

            border-radius: 4px;

            cursor: pointer;

            font-size: 16px;

        }

        button:hover {

            background-color: #45a049;

        }

        @media (max-width: 600px) {

            .container {

                max-width: 90%;

            }

            h1 {

                font-size: 20px;

            }

        }

    </style>

</head>

<body>

    <div class="container">

        <h1>Resume Parser</h1>

        <form action="/api/extract\_resume/" method="POST" enctype="multipart/form-data">

            <input type="file" name="resume" accept=".pdf,.doc,.docx" required>

            <button type="submit">Upload Resume</button>

        </form>

    </div>

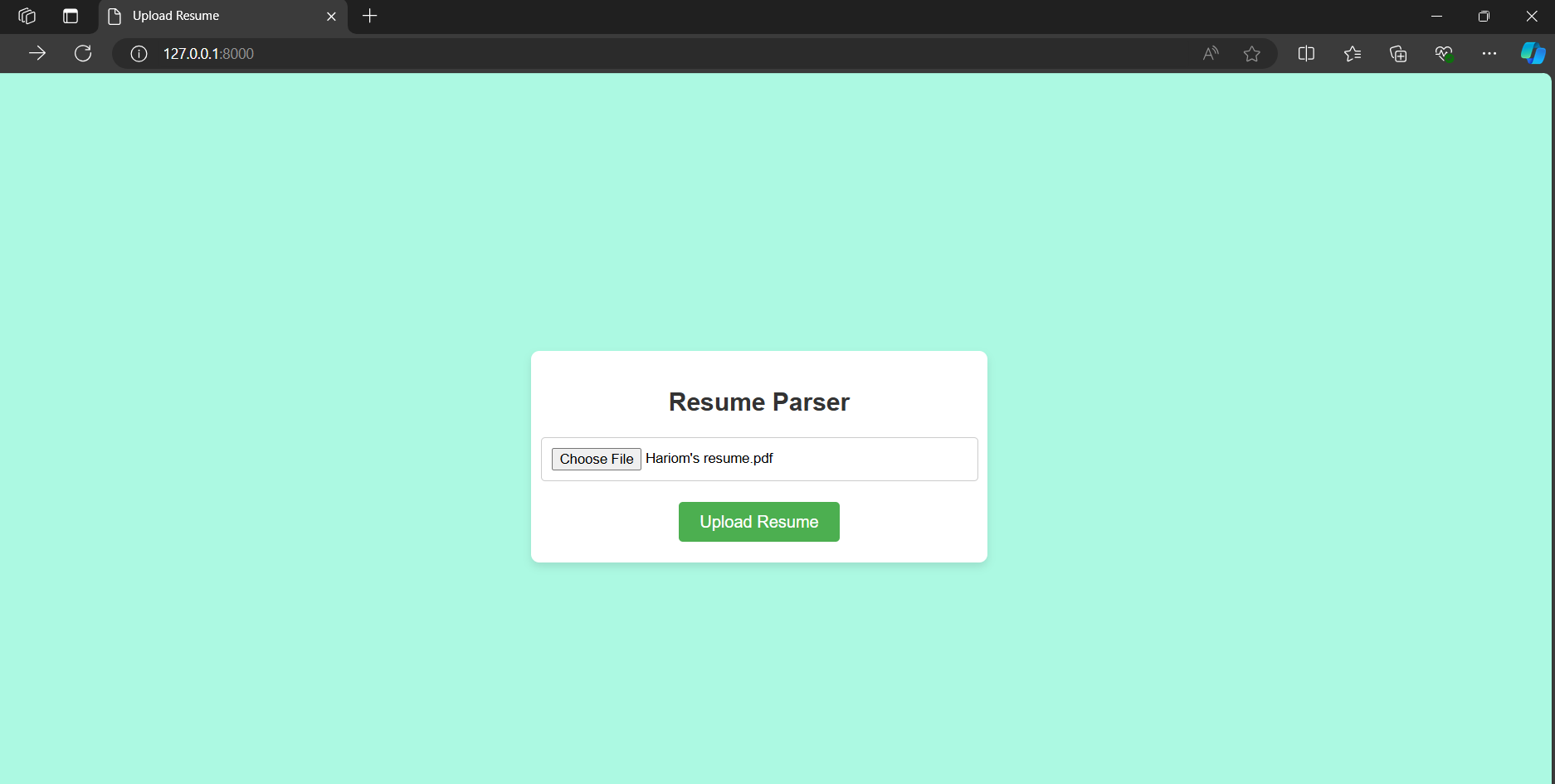
</body>

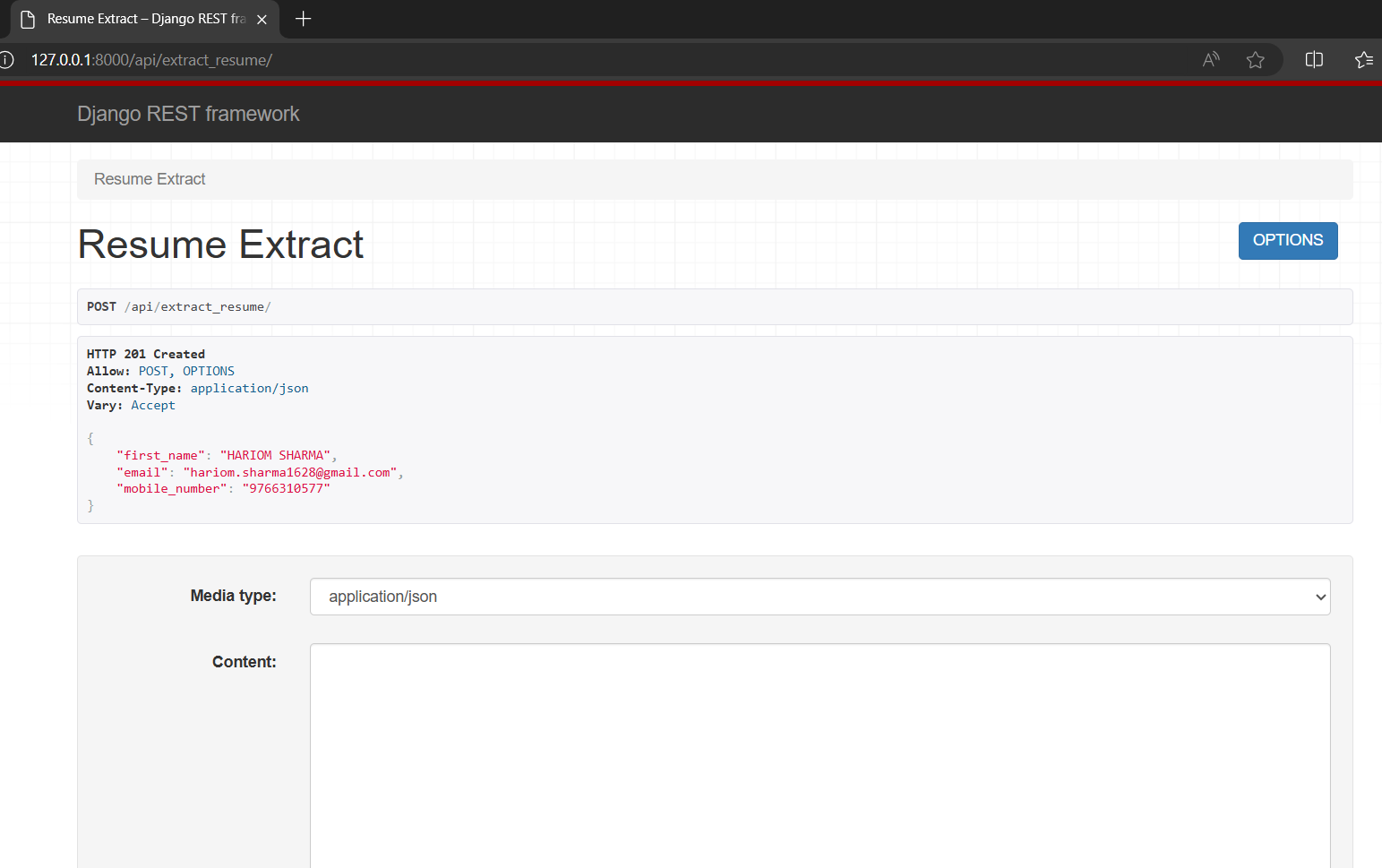
</html>

1. After successfully completing all the setup carefully we run python server

*python manage.py runserver*

1. Output





1. Testing the API Endpoint using the Postman

