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)

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

   pip install django djangorestframework psycopg2-binary

   pip install psycopg

3. Install all the necessary libraries using pip

   pip freeze > requirements.txt

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

   django-admin startproject ResumeParser

   cd ResumeParser

5. Create a jango app 'resume' using the command

   python manage.py startapp resume

6. 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;

7. 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.*

8. 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)
```

9. 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']
```

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

```
python manage.py makemigrations resume
python manage.py migrate
```

11. 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
```

```python
            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)
```

12. 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)
```

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

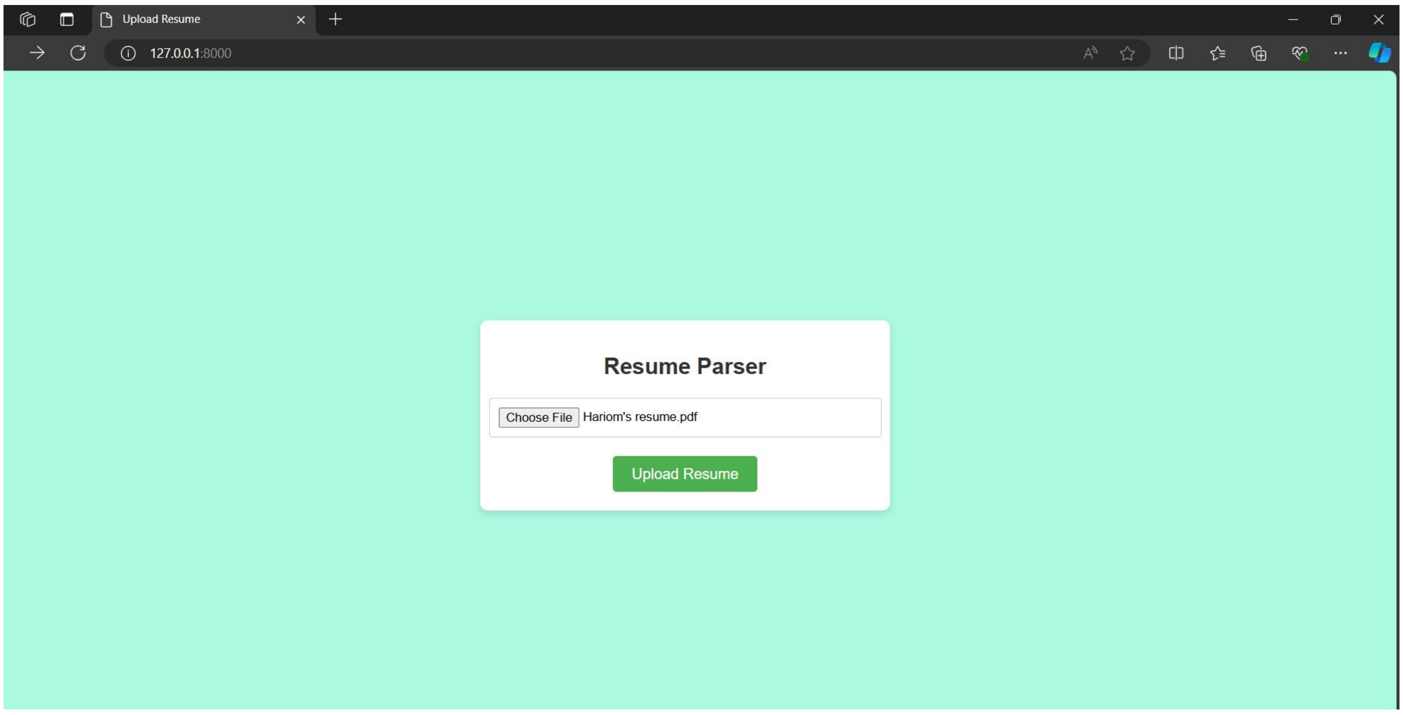
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Upload Resume</title>
    <style>
```

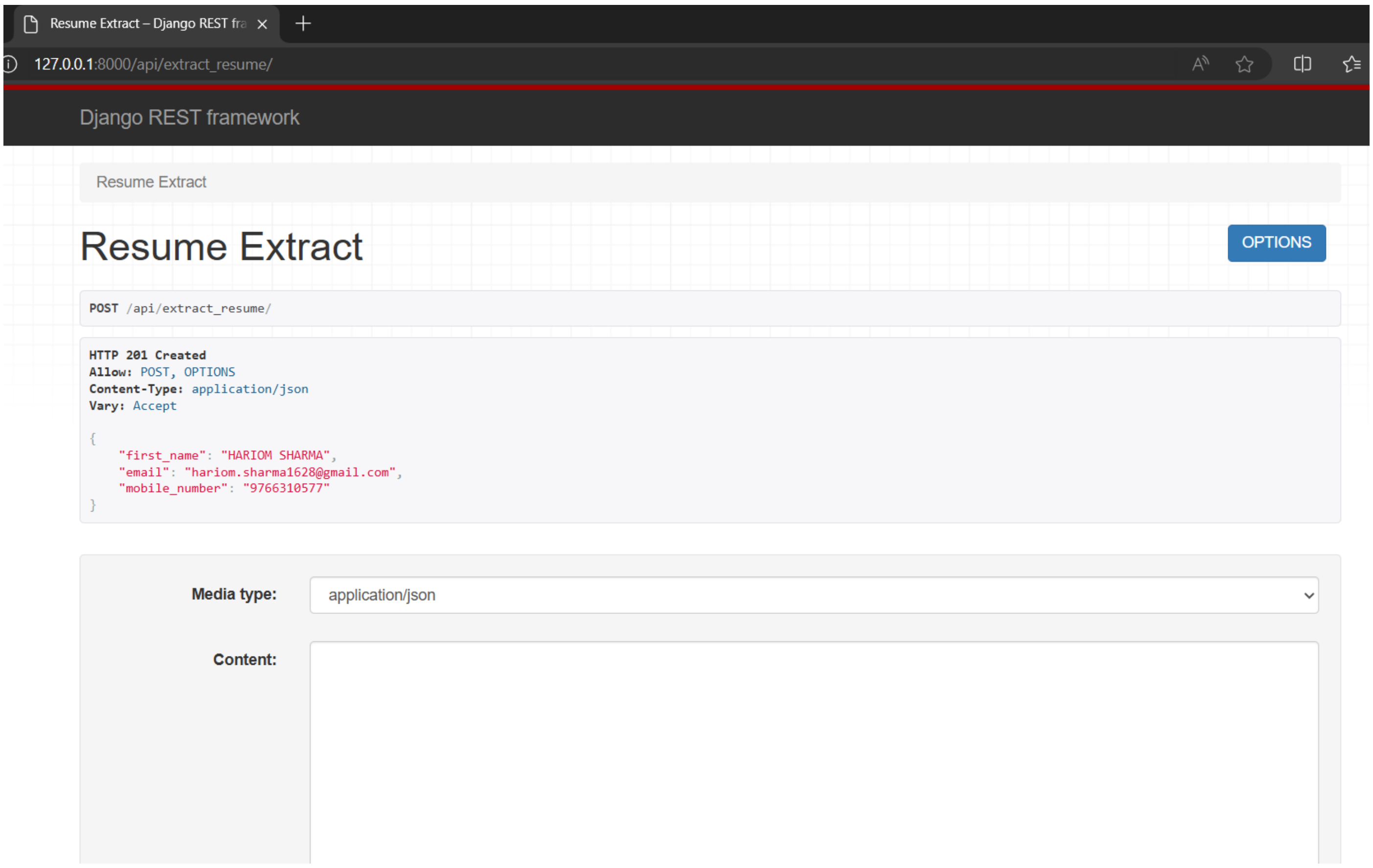
```css
        body {
            font-family: Arial, sans-serif;
            background-color: #acf9e2;
            margin: 0;
            padding: 0;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
        }

        .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>
```

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

python manage.py runserver

15. Output




Resume Extract – Django REST fra
127.0.0.1:8000/api/extract_resume/
Django REST framework
Resume Extract
OPTIONS

Resume Extract

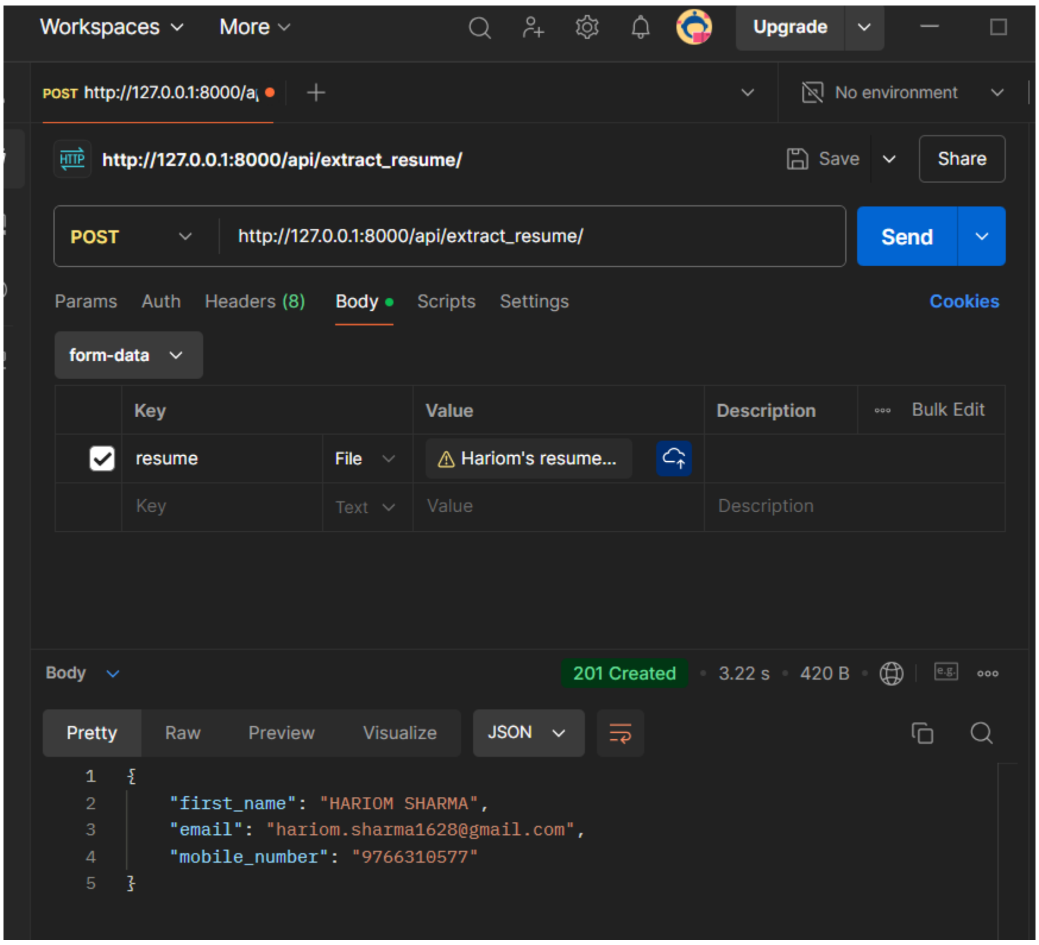
```
POST /api/extract_resume/
```

```
HTTP 201 Created
Allow: POST, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "first_name": "HARIOM SHARMA",
    "email": "hariom.sharma1628@gmail.com",
    "mobile_number": "9766310577"
}
```

Media type: application/json

Content:

16. Testing the API Endpoint using the Postman



http://127.0.0.1:8000/api/extract_resume/

POST http://127.0.0.1:8000/api/extract_resume/

form-data

Key		Value	Description
resume	File	Hariom's resume...	
Key	Text	Value	Description

Body — 201 Created · 3.22 s · 420 B — JSON

```
{
    "first_name": "HARIOM SHARMA",
    "email": "hariom.sharma1628@gmail.com",
    "mobile_number": "9766310577"
}
```