

Resume Analyser

```
"use client";
import React, { useState, useRef } from "react";
import { Button } from "@/components/ui/button";
import {
  Dialog,
  DialogContent,
  DialogDescription,
  DialogHeader,
  DialogTitle,
  DialogTrigger,
} from "@/components/ui/dialog";
import {
  FileText,
  Upload,
  CheckCircle2,
  AlertTriangle,
  TrendingUp,
  Loader2,
  ArrowRight,
  FileCheck,
  Zap,
} from "lucide-react";
import axios from "axios";
import { ResumeAnalysisResponse } from "@/type";
import { utils_service } from "@/context/AppContext";
// import toast from "react-hot-toast";
const ResumeAnalyzer = () => {
  const [open, setOpen] = useState(false);
  const [file, setFile] = useState<File | null>(null);
  const [loading, setLoading] = useState(false);
  const [response, setResponse] = useState<ResumeAnalysisResponse | null>(null);
  const fileInputRef = useRef<HTMLInputElement>(null);
  const handleFileSelect = (e: React.ChangeEvent<HTMLInputElement>) => {
    const selectedFile = e.target.files?.[0];
    if (selectedFile) {
      if (selectedFile.type !== "application/pdf") {
        // toast.error("Please upload a PDF file");
        alert("Please upload a PDF file");
        return;
      }
      if (selectedFile.size > 5 * 1024 * 1024) {
        // toast.error("File size should be less than 5MB");
        alert("File size should be less than 5MB");
        return;
      }
      setFile(selectedFile);
    }
  };

  const convertToBase64 = (file: File): Promise<string> => {
    return new Promise((resolve, reject) => {
```

```

const reader = new FileReader();
reader.readAsDataURL(file);
reader.onload = () => resolve(reader.result as string);
reader.onerror = (error) => reject(error);
});
};

const analyzeResume = async () => {
if (!file) {
// toast.error("Please upload a resume");
alert("Please upload a resume");
return;
}
setLoading(true);
try {
const base64 = await convertToBase64(file);
const { data } = await axios.post(
`${utils_service}/api/utils/resume-analyser`,
{
pdfBase64: base64,
}
);
setResponse(data);
// toast.success("Resume analyzed successfully!");
alert("Resume analyzed successfully!");
} catch (error: any) {
// toast.error(error.response?.data?.message || "Failed to analyze resume");
alert(error.response?.data?.message || "Failed to analyze resume");
console.log(error);
} finally {
setLoading(false);
}
};

const resetDialog = () => {
setFile(null);
setResponse(null);
setOpen(false);
if (fileInputRef.current) {
fileInputRef.current.value = "";
}
};

const getScoreColor = (score: number) => {
if (score >= 80) return "text-green-600";
if (score >= 60) return "text-yellow-600";
return "text-red-600";
};

const getScoreBgColor = (score: number) => {

if (score >= 80) return "bg-green-100 dark:bg-green-900/30";
if (score >= 60) return "bg-yellow-100 dark:bg-yellow-900/30";
return "bg-red-100 dark:bg-red-900/30";
};

const getPriorityColor = (priority: string) => {
if (priority === "high")

```

```

return "bg-red-100 dark:bg-red-900/30 text-red-600 border-red-200 dark:border-red-800";

if (priority === "medium")
return "bg-yellow-100 dark:bg-yellow-900/30 text-yellow-600 border-yellow-200 dark:border-
yellow-800";

return "bg-blue-100 dark:bg-blue-900/30 text-blue-600 border-blue-200 dark:border-blue-
800";

};
return (
<div className="max-w-7xl mx-auto px-4 py-16 bg-secondary/30">
<div className="text-center mb-12">
<div

className="inline-flex items-center gap-2 px-4 py-2 rounded-full border bg-red-
50 dark:bg-red-950/30 mb-4"

>
<FileCheck size={16} className="text-red-500" />
<span className="text-sm font-medium">AI-Powered ATS Analysis</span>
</div>
<h2 className="text-3xl md:text-4xl font-bold mb-4">
Optimize Your Resume for ATS
</h2>
<p className="text-lg opacity-70 max-w-2xl mx-auto mb-8">
Get instant feedback on your resume's compatibility with Applicant
Tracking Systems
</p>
<Dialog open={open} onOpenChange={setOpen}>
<DialogTrigger asChild>
<Button size="lg" className="gap-2 h-12 px-8">
<FileText size={18} />
Analyze My Resume
<ArrowRight size={18} />
</Button>
</DialogTrigger>
<DialogContent className="max-w-4xl max-h-[90vh] overflow-y-auto">
{!response ? (
<>
<DialogHeader>
<DialogTitle className="text-2xl flex items-center gap-2">
<FileText className="text-red-500" />
Upload Your Resume
</DialogTitle>
<DialogDescription>
Upload your resume in PDF format to get an instant ATS
compatibility analysis
</DialogDescription>
</DialogHeader>
<div className="space-y-4 py-4">
<div
onClick={() => fileInputRef.current?.click()}

```

```

className="border-2 border-dashed rounded-lg p-12 text-center cursor-
pointer hover:border-blue-500 transition-colors"

>
<div className="flex flex-col items-center gap-4">
<div
className="h-16 w-16 rounded-full bg-blue-100 dark:bg-blue-900/30
flex items-center justify-center"
>
<Upload size={32} className="text-blue-600" />
</div>
<div>

<p className="font-medium mb-1">
{file ? file.name : "Click to upload your resume"}
</p>

<p className="text-sm opacity-60">
PDF format only, maximum 5MB
</p>
</div>
{file && (

<div className="flex items-center gap-2 text-green-600">
<CheckCircle2 size={18} />
<span className="text-sm font-medium">
File uploaded successfully
</span>
</div>
)}
</div>
</div>
<input
ref={fileInputRef}
type="file"
accept="application/pdf"
onChange={handleFileSelect}
className="hidden"
/>
<Button
onClick={analyzeResume}
disabled={loading || !file}
className="w-full h-11 gap-2"
>
{loading ? (
<>
<Loader2 size={18} className="animate-spin" />
Analyzing Your Resume...
</>
) : (

<>
<Zap size={18} />

```

```

Analyze Resume
</>
)}}
</Button>
</div>
</>
) : (
<>
<DialogHeader>
<DialogTitle className="text-2xl flex items-center gap-2">
<FileCheck className="text-red-500" />
Your Resume Analysis
</DialogTitle>
</DialogHeader>
<div className="space-y-6 py-4">
  { /* Overall Score */ }
  <div
    className={ `p-6 rounded-lg ${getScoreBgColor(
      response.atsScore
    )} border-2`}
    >
    <div className="text-center">
      <p className="text-sm font-medium opacity-70 mb-2">
        ATS Compatibility Score
      </p>
      <div
        className={ `text-6xl font-bold ${getScoreColor(
          response.atsScore
        )} ` }
        >
        {response.atsScore}
      </div>
      <p className="text-sm opacity-70 mt-2">out of 100</p>
    </div>
  </div>
  { /* Summary */ }
  <div
    className="p-4 rounded-lg bg-blue-50 dark:bg-blue-950/30 border
    border-blue-200 dark:border-blue-800"
    >
    <p className="text-sm leading-relaxed">
      {response.summary}
    </p>
  </div>
  { /* Score Breakdown */ }
  <div>
    <h3 className="text-lg font-semibold mb-3 flex items-center gap-2">
      <TrendingUp size={20} className="text-red-500" />
      Detailed Score Breakdown
    </h3>

```

```

<div className="grid md:grid-cols-2 gap-3">
  {Object.entries(response.scoreBreakdown).map(
    ([key, value]) => (
      <div key={key} className="p-4 rounded-lg border">
        <div className="flex items-center justify-between mb-2">
          <p className="font-semibold capitalize">{key}</p>
          <span
            className={`text-lg font-bold ${getScoreColor(
              value.score
            )}}`>
            {value.score}%
          </span>
        </div>
      </div>

      <p className="text-xs opacity-70">
        {value.feedback}
      </p>
    </div>
  )
  )}
</div>
</div>
{/* Strengths */}
<div
  className="p-4 rounded-lg bg-green-50 dark:bg-green-950/30 border
  border-green-200 dark:border-green-800"
  >
  <h3 className="font-semibold mb-3 flex items-center gap-2">
    <CheckCircle2 size={18} className="text-green-600" />
    What Your Resume Does Well
  </h3>
  <ul className="space-y-2">
    {response.strengths.map((strength, index) => (
      <li
        key={index}
        className="text-sm flex items-start gap-2"
        >
        <span className="text-green-600 mt-0.5">✓</span>
        <span>{strength}</span>
      </li>
    ))}
  </ul>
</div>
{/* Suggestions */}
<div>
  <h3 className="text-lg font-semibold mb-3 flex items-center gap-2">
    <AlertTriangle size={20} className="text-red-500" />
    Recommendations for Improvement
  </h3>
  <div className="space-y-3">
    {response.suggestions.map((suggestion, index) => (

```

```
<div key={index} className="p-4 rounded-lg border">
<div className="flex items-start justify-between gap-3 mb-2">
<h4 className="font-semibold text-sm">
{suggestion.category}
</h4>
<span
className={`text-xs px-2 py-1 rounded-full border
${getPriorityColor(suggestion.priority)}}`
>
{suggestion.priority}
</span>
</div>

<div className="space-y-2 text-sm">

<div>
<span className="font-medium opacity-70">
Issue:{" "}
</span>

<span className="opacity-80">

{suggestion.issue}
</span>
</div>
<div>

<span className="font-medium opacity-70">
Fix:{" "}
</span>

<span className="opacity-80">
{suggestion.recommendation}
</span>
</div>
</div>
</div>
)))}
</div>
</div>
<Button
onClick={resetDialog}
variant="outline"
className="w-full"
>
Analyze Another Resume
</Button>
</div>
</>
))}
</DialogContent>
</Dialog>
</div>
```

```
</div>
);
};
export default ResumeAnalyzer;
```

About Page

```
import { Button } from "@components/ui/button";
import Link from "next/link";
import React from "react";
import { ArrowRight } from "lucide-react";

const About = () => {
  return (
    <div className="min-h-screen">
      {/* Mission Section */}
      <section className="container mx-auto px-4 py-12 md:py-16">
        <div className="max-w-4xl mx-auto">
          {/* Image */}
          <div className="flex justify-center mb-8">
            
          </div>
        </div>

        {/* Content */}
        <div className="text-center space-y-6">
          <h1 className="text-3xl md:text-4xl lg:text-5xl font-bold">
            Our Mission At Hire
            <span className="text-red-500">Heaven</span>
          </h1>

          <p
            className="text-lg md:text-xl leading-relaxed opacity-90 max-w-3xl mx-auto"
          >
            At HireHaven, we're dedicated to revolutionizing the job search
            experience. Our mission is to create meaningful connections
            between talented individuals and forward-thinking companies,
            fostering growth and success for both.
          </p>
        </div>
      </section>

      {/* CTA Section */}
      <section className="py-16 md:py-20 bg-secondary">
        <div className="container mx-auto px-4">
          <div className="max-w-3xl mx-auto text-center space-y-6">
            <h2 className="text-3xl md:text-4xl lg:text-5xl font-bold">
              Ready to find your dream job?
            </h2>

```



```
    <p className="text-lg md:text-xl opacity-80">
      Join thousands of successful job seekers on HireHeaven
    </p>
    <div className="pt-4">
      <Link href="/jobs">
        <Button size="lg" className="gap-2 h-12 px-8 text-base">
          Get Started
          <ArrowRight size={18} />
        </Button>
      </Link>
    </div>
  </div>
</div>
</section>
</div>
);
};

export default About;
```