

mock-interview-assistant

Project Folder Structure

mock-interview-assistant/

|— backend/

| |— server.js

| |— routes/

| | |— resume.js

| | |— interview.js

| | |— jobs.js

| |— services/

| | |— amazonQService.js

| | |— atsParser.js

| | |— scoringEngine.js

| |— models/

| | |— Candidate.js

| | |— Interview.js

| |— config/

| |— db.js

```
|
|— frontend/
|   |— src/
|       |— App.jsx
|       |— components/
|           |— ResumeUpload.jsx
|           |— InterviewPanel.jsx
|           |— VoiceRecorder.jsx
|           |— JobRecommendations.jsx
|           |— pages/
|               |— Dashboard.jsx
|               |— Results.jsx
|           |— public/
|               |— index.html
|
|— package.json
|— README.md
```

```
# backend/server.js !  
  
// Basic Express setup  
const express = require("express");  
const multer = require("multer");  
const cors = require("cors");  
  
const { parseResume } =  
  require("./services/atsParser");  
const { generateQuestions, evaluateAnswers } =  
  require("./services/amazonQService");  
const { fetchJobs } = require("./services/jobsAPI");  
  
const app = express();  
app.use(cors());  
app.use(express.json());  
  
// Multer for file uploads  
const upload = multer({ dest: "uploads/" });
```

```
// Upload & parse resume
app.post("/api/upload-resume",
upload.single("resume"), async (req, res) => {
  try {
    const skills = await parseResume(req.file.path);
    res.json({ success: true, skills });
  } catch (err) {
    console.error(err);
    res.status(500).json({ success: false, error:
"Failed to parse resume" });
  }
});
```

```
// Generate interview questions
app.post("/api/questions", async (req, res) => {
  const { role } = req.body;
  if (!role) return res.status(400).json({ error: "Role
required" });
```

```
const questions = await generateQuestions(role);  
res.json({ success: true, questions });  
});
```

```
app.post("/api/evaluate", async (req, res) => {  
  const { answers } = req.body;  
  if (!answers) return res.status(400).json({ error:  
    "Answers required" });
```

```
  const results = await evaluateAnswers(answers);  
  res.json({ success: true, results });  
});
```

```
app.get("/api/jobs", async (req, res) => {  
  const skills = req.query.skills ?  
    req.query.skills.split(",") : [];  
  const jobs = await fetchJobs(skills);  
  res.json({ success: true, jobs });  
});
```

```
app.listen(5000, () => console.log("Server running  
on port 5000"));
```

```
#backend/services/atsParser.js
```

```
const fs = require("fs");
```

```
async function parseResume(filePath) {  
  const content = fs.readFileSync(filePath, "utf8");  
  const skills = ["JavaScript", "React", "Node.js",  
"AWS"];  
  return skills;  
}
```

```
module.exports = { parseResume };
```

```
# backend/services/amazonQService.js !
```

```
async function generateQuestions(role) {
```

```
if (role.toLowerCase() === "frontend developer")
{
    return [
        "Explain the difference between React state
and props.",
        "How would you optimize a React app for
performance?",
    ];
} else if (role.toLowerCase() === "backend
developer") {
    return [
        "What are the benefits of using
microservices?",
        "Explain connection pooling in databases.",
    ];
}

return ["Tell me about yourself.", "What
motivates you to work in this role?"];
}
```

```
async function evaluateAnswers(answers) {  
  return answers.map((ans, idx) => ({  
    questionId: idx + 1,  
    answer: ans,  
    score: Math.floor(Math.random() * 10) + 1,  
    feedback: "Good attempt, improve clarity."  
  }));  
}
```

```
module.exports = { generateQuestions,  
  evaluateAnswers };
```

#backend/services/jobsAPI.js !

```
async function fetchJobs(skills) {  
  return [  
    { title: "Frontend Engineer", company:  
      "TechCorp", skills: ["React", "JavaScript"], location:  
      "Remote" },
```



```
    { title: "Backend Developer", company:
    "InnoSoft", skills: ["Node.js", "AWS"], location:
    "Bangalore" },

    { title: "Fullstack Developer", company:
    "Cloudify", skills: ["React", "Node.js", "AWS"],
    location: "Hyderabad" }

];
}
```

```
module.exports = { fetchJobs };
```

#FRONT-END !

-----frontend/src/App.jsx !

```
import React, { useState } from "react";
```

```
import ResumeUpload from
"./components/ResumeUpload";
```

```
import InterviewPanel from
"./components/InterviewPanel";
```

```
import JobRecommendations from
"./components/JobRecommendations";
```

```
function App() {
  const [skills, setSkills] = useState([]);
  const [questions, setQuestions] = useState([]);
  const [results, setResults] = useState([]);
  const [jobs, setJobs] = useState([]);

  return (
    <div className="p-6 max-w-4xl mx-auto">
      <h1 className="text-2xl font-bold text-center
mb-6">
        Intelligent Mock Interview Assistant
      </h1>

      <ResumeUpload setSkills={setSkills} />
      <InterviewPanel
        skills={skills}
```

```
    setQuestions={setQuestions}
    questions={questions}
    setResults={setResults}
  />
  <JobRecommendations skills={skills}
jobs={jobs} setJobs={setJobs} />
</div>
);
}
```

```
export default App;
```

```
#ResumeUpload.jsx !
```

```
import React, { useState } from "react";
```

```
import axios from "axios";
```

```
function ResumeUpload({ setSkills }) {
```

```
  const [file, setFile] = useState(null);
```

```
const uploadResume = async () => {  
  if (!file) return alert("Please select a resume  
first");
```

```
  const formData = new FormData();  
  formData.append("resume", file);
```

```
  const res = await  
  axios.post("http://localhost:5000/api/upload-  
resume", formData);  
  setSkills(res.data.skills);  
};
```

```
return (  
  <div className="p-4 border rounded mb-4">  
    <h2 className="font-semibold mb-2">Upload  
Resume</h2>  
    <input type="file" onChange={(e) =>  
setFile(e.target.files[0])} />  
    <button
```

```
      className="bg-blue-600 text-white px-4 py-1  
rounded ml-2"
```

```
      onClick={uploadResume}
```

```
>
```

```
    Upload
```

```
</button>
```

```
</div>
```

```
);
```

```
}
```

```
export default ResumeUpload;
```

```
#InterviewPanel.jsx !
```

```
import React, { useState } from "react";
```

```
import axios from "axios";
```

```
function InterviewPanel({ skills, setQuestions,  
questions, setResults }) {
```

```
const [answers, setAnswers] = useState([]);
```

```
const getQuestions = async () => {  
  const res = await  
  axios.post("http://localhost:5000/api/questions",  
  {  
    role: skills.includes("React") ? "Frontend  
Developer" : "Backend Developer",  
  });  
  setQuestions(res.data.questions);  
};
```

```
const submitAnswers = async () => {  
  const res = await  
  axios.post("http://localhost:5000/api/evaluate", {  
    answers,  
  });  
  setResults(res.data.results);  
};
```

```
return (  
  <div className="p-4 border rounded mb-4">  
    <h2 className="font-semibold mb-2">Mock  
Interview</h2>  
    <button  
      className="bg-green-600 text-white px-4 py-  
1 rounded"  
      onClick={getQuestions}  
    >  
      Get Questions  
    </button>  
  
    {questions.map((q, idx) => (  
      <div key={idx} className="mt-3">  
        <p>{q}</p>  
        <input  
          className="border px-2 py-1 w-full"  
          onChange={(e) => {  
            const newAnswers = [...answers];
```

```
        newAnswers[idx] = e.target.value;
        setAnswers(newAnswers);
    }}
/>
</div>
)))}
```

```
{questions.length > 0 && (
  <button
    className="bg-blue-600 text-white px-4 py-
1 rounded mt-3"
    onClick={submitAnswers}
  >
    Submit Answers
  </button>
)}
</div>
);
}
```



```
export default InterviewPanel;
```

```
#JobRecommendations.jsx !
```

```
import React from "react";
```

```
import axios from "axios";
```

```
function JobRecommendations({ skills, jobs,  
setJobs }) {
```

```
  const getJobs = async () => {
```

```
    const res = await
```

```
    axios.get("http://localhost:5000/api/jobs", {
```

```
      params: { skills: skills.join(",") },
```

```
    });
```

```
    setJobs(res.data.jobs);
```

```
  };
```

```
  return (
```

```
<div className="p-4 border rounded">
  <h2 className="font-semibold mb-2">Job
Recommendations</h2>
  <button
    className="bg-purple-600 text-white px-4
py-1 rounded"
    onClick={getJobs}
  >
    Fetch Jobs
  </button>
  <ul className="mt-3">
    {jobs.map((job, idx) => (
      <li key={idx} className="border p-2 mb-2
rounded">
        <strong>{job.title}</strong> at
{job.company} <br />
        Skills: {job.skills.join(", ")} <br />
        Location: {job.location}
      </li>
    ))}
  </ul>
</div>
```


</div>

);

}

export default JobRecommendations;