

PROJECT DOCUMENTATION

AI-POWERED INTERVIEWER

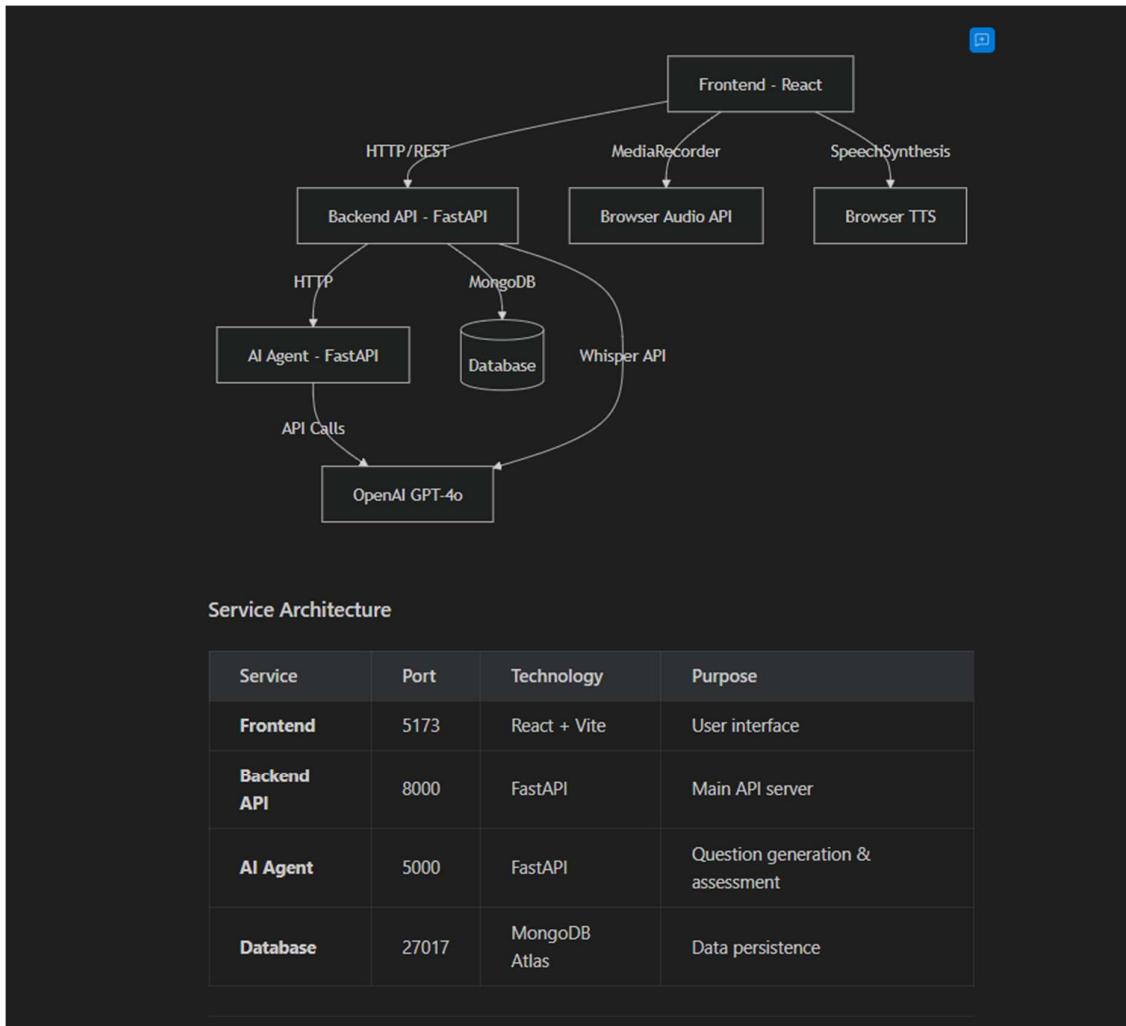
Project Overview

An AI-powered interview platform that conducts **fully voice-based technical interviews**. The system analyzes resumes, generates contextual questions, transcribes voice responses using Whisper API, and provides comprehensive candidate assessments.

Key Capabilities

- Resume parsing and analysis
 - AI-generated contextual interview questions
 - Voice-only interview (no typing required)
 - Real-time speech-to-text transcription
 - Text-to-speech for questions
 - Automated candidate assessment with ratings
 - Comprehensive interview reports
-

Architecture



Features

1. Resume Processing

- PDF upload and text extraction
- Automatic chunking for context
- Candidate profile extraction (name, email, skills, experience)
- Seniority level detection

2. Voice Interview System

- **Audio Recording:** Browser MediaRecorder API with 100ms timeslice
- **Microphone Selection:** Automatic device enumeration
- **Minimum Duration:** 4-second recording requirement with live timer
- **Transcription:** OpenAI Whisper API (whisper-1 model)
- **Text-to-Speech:** Browser SpeechSynthesis API

- **Audio Quality:** Echo cancellation, noise suppression, auto gain control

3. Interview Flow

1. User uploads resume
2. System extracts profile and generates first question
3. AI speaks question (TTS)
4. User records voice answer (min 4 seconds)
5. Whisper transcribes answer
6. User reviews and submits transcription
7. AI generates next contextual question
8. Repeat until interview complete
9. Generate comprehensive assessment

4. Assessment System

- **Overall Score:** 0-100 rating
 - **Hiring Recommendation:** "Definitely Hire!", "Proceed with caution", or "Don't hire"
 - **Strengths:** List of demonstrated strengths
 - **Improvement Areas:** Areas needing development
 - **Next Steps:** Actionable recommendations
-

Technology Stack

Frontend

- **Framework:** React 18
- **Build Tool:** Vite
- **Styling:** Tailwind CSS
- **HTTP Client:** Axios
- **Audio:** MediaRecorder API, SpeechSynthesis API
- **Routing:** React Router

Backend API

- **Framework:** FastAPI
- **Language:** Python 3.11+
- **Database:** MongoDB (Motor async driver)
- **Authentication:** JWT

- **File Processing:** PyPDF2
- **Configuration:** Pydantic Settings

AI Agent

- **Framework:** FastAPI
- **LLM:** OpenAI GPT-4o
- **Prompting:** LangChain ChatPromptTemplate
- **Speech-to-Text:** OpenAI Whisper API
- **Structured Output:** Pydantic models

Database

- **Type:** MongoDB Atlas (Cloud)
 - **Collections:** users, interview_sessions, interview_answers
 - **ODM:** Motor (async)
-

Setup & Installation

Prerequisites

- Python 3.11+
- Node.js 18+
- MongoDB Atlas account
- OpenAI API key

Backend Setup

```
# Navigate to backend
cd backend

# Create virtual environment
python -m venv venv
venv\Scripts\activate # Windows

# Install dependencies
pip install -r requirements.txt

# Configure environment

# Create .env file with:
MONGO_URI="mongodb+srv://username:password@cluster.mongodb.net/"
```

```
DB_NAME="ai_interview_db"
AI_AGENT_URL="http://localhost:5000"
JWT_SECRET="your-secret-key"
OPENAI_API_KEY="sk-proj-..."
# Run backend
python -m uvicorn app.main:app --reload --port 8000
```

AI Agent Setup

```
# Navigate to AI agent
cd backend/ai-agent
# Create virtual environment
python -m venv venv
venv\Scripts\activate
# Install dependencies
pip install -r requirements.txt
# Configure environment
# Create .env file with:
OPENAI_API_KEY="sk-proj-..."
PORT=5000
# Run AI agent
python app.py
```

Frontend Setup

```
# Navigate to frontend
cd frontend/frontend/Chat_Agent
# Install dependencies
npm install
# Run development server
npm run dev
```

Access Application

- Frontend: <http://localhost:5173>
 - Backend API: <http://localhost:8000/docs>
 - AI Agent: <http://localhost:5000/health>
-

API Documentation

Backend API Endpoints

Authentication

POST /api/auth/signup

POST /api/auth/login

Resume Management

POST /api/resume/upload

GET /api/resume/{userId}

Interview Flow

POST /api/interview/start

POST /api/interview/init/{sessionId}

POST /api/interview/answer/{sessionId}

POST /api/interview/transcribe

Voice Transcription

POST /api/interview/transcribe

Content-Type: multipart/form-data

Body:

- audio: File (webm, mp3, wav, mpeg, ogg)

Response:

```
{  
  "text": "transcribed text",  
  "success": true  
}
```

AI Agent Endpoints

Resume Parsing

POST /parse-resume

{

 "userId": "string",

 "resumeText": "string",

 "chunks": ["string"]

}

Interview Questions

POST /init-interview

{

 "sessionId": "string",

 "resumeText": "string",

 "chunks": ["string"]

}

POST /next-question

{

 "sessionId": "string",

 "resumeText": "string",

 "chunks": ["string"],

 "currentQuestionNumber": 1,

 "currentAnswer": "string"

}

Assessment Generation

POST /generate-assessment

{

 "sessionId": "string",

 "resumeText": "string",

 "chunks": ["string"],

 "transcript": [

 {"question": "...", "answer": "..."}
]

```
],
  "seniorityLevel": "Mid-Senior"
}
```

Response:

```
{
  "assessment": {
    "candidate_score_percent": "85",
    "hiring_recommendation": "Definitely Hire!",
    "strengths": [...],
    "improvement_areas": [...],
    "next_steps": ...
  }
}
```

Database Schema

Collections

users

```
{
  "_id": ObjectId,
  "email": "user@example.com",
  "password": "hashed_password",
  "name": "John Doe",
  "resumeProfile": {
    "extracted_text": "...",
    "chunks": [..., ...],
    "candidate_first_name": "John",
    "candidate_last_name": "Doe",
    "candidate_email": "john@example.com",
    "candidate_linkedin": "linkedin.com/in/johndoe",
    "experience": "5 years in software development",
    "skills": ["Python", "React", "AWS"],
  }
}
```

```
        "seniority_level": "Mid-Senior"
    },
    "createdAt": ISODate
}

interview_sessions
{
    "_id": ObjectId,
    "userId": "user_id",
    "status": "completed", // "active" | "completed"
    "createdAt": ISODate,
    "completedAt": ISODate,
    "assessment": {
        "candidate_score_percent": "85",
        "hiring_recommendation": "Definitely Hire!",
        "strengths": ["Strong technical knowledge", "Clear communication"],
        "improvement_areas": ["System design depth"],
        "next_steps": "Proceed to technical round"
    }
}

interview_answers
{
    "_id": ObjectId,
    "sessionId": "session_id",
    "questionNumber": 1,
    "question": "Tell me about yourself",
    "answer": "I am a software engineer..."
}
```

```
"createdAt": ISODate,  
"updatedAt": ISODate  
}  
}
```

