# Resume Analyzer API

A mini-backend service for analyzing resumes, built with Node.js, Express, and MongoDB, with integration to the Google Gemini API for LLM processing. This API authenticates users, extracts structured data from unstructured resume text, encrypts sensitive fields, and stores/retrieves applicant data from a MongoDB Atlas instance.

# Features

- JWT-based Authentication:

Authenticate using hard-coded credentials and receive a JSON Web Token for securing subsequent API calls.

- Resume Enrichment:

Send raw resume text to extract structured data (e.g., name, email, education, experience, skills, summary) via the Google Gemini API. Sensitive data (name & email) is encrypted before storage and decrypted for display.

- Resume Search:

Search stored resumes by applicant name (case-insensitive).

- Secure and Modular:

Uses environment variables for secrets, a modular code structure (separate folders for models, routes, middleware, and utilities), and robust error handling.

Technologies Used

- Node.js & Express: For building RESTful APIs.

- MongoDB & Mongoose: For non-relational data storage (using MongoDB Atlas free tier).

- JWT: For authentication and authorization.

- Crypto (Node.js): For AES-128 encryption/decryption.

- Google Gemini API: For extracting structured data from raw resume text.

- @google/generative-ai: Official Gemini API client for Node.js.

- Nodemon: For development auto-restarts.

Getting Started

Prerequisites

- [Node.js](https://nodejs.org/) installed on your machine.

- A MongoDB Atlas account and a connection string.

- Access to the Google Gemini API and your API key.

- Git installed for version control.

Installation

1. Clone the Repository:

git clone https://github.com/rsphoenix01/resume-parser.git

cd resume-backend

**Install Dependencies:**

bash

Copy

npm install

npm install @google/generative-ai

Create a .env File in the Root Directory:

# MongoDB connection string (Cloud version, e.g., MongoDB Atlas)

MONGODB\_URI="your\_mongodb\_connection\_string"

# Secret key for signing JSON Web Tokens

JWT\_SECRET=your\_jwt\_secret

# Encryption key (16 characters for AES-128)

ENC\_KEY=abcdefghijklmnop

# Initialization Vector for encryption (16 characters)

IV=abcdefghijklmnop

# Your Google Gemini API key

GEMINI\_API\_KEY=your\_gemini\_api\_key

# Port number (optional, defaults to 3000)

PORT=3000

## **API Endpoints**

### **1. Authentication API**

* **Endpoint:** POST /api/auth

**Request Body:**json  
Copy  
{

"username": "roshan",

"password": "admin123"

}

**Response:**Returns a JSON object with a JWT token:  
json  
Copy  
{

"JWT": "<your\_jwt\_token\_here>"

}

### **2. Resume Enrichment API**

* **Endpoint:** POST /api/enrich

**Headers:**Include your JWT in the Authorization header as:  
makefile  
Copy  
Authorization: Bearer <your\_jwt\_token\_here>

**Request Body:**json  
Copy  
{

"raw\_text": "Your raw resume text here..."

}

**Response:**Returns structured JSON extracted from the resume, e.g.:  
json  
Copy  
{

"name": "Scarlett Emerson",

"email": "scarlett.emerson@hollywoodstudios.com",

"education": {

"degree": "Bachelor of Fine Arts",

"branch": "Film Production",

"institution": "UCLA",

"year": "2015"

},

"experience": {

"job\_title": "Assistant Director",

"company": "Paramount Pictures",

"start\_date": "June 2018",

"end\_date": "current"

},

"skills": ["Cinematography", "editing", "screenwriting"],

"summary": "Passionate about filmmaking."

}

### **3. Resume Search API**

* **Endpoint:** POST /api/search
* **Headers:**Include the JWT in the Authorization header.

**Request Body:**json  
Copy  
{

"name": "Scarlett"

}

* **Response:**Returns an array of applicant records matching the search query.