

OPTEHIRE - Resume Analysis and Feedback System

Overview

OPTEHIRE is an AI-driven platform designed to assist job seekers in optimizing their resumes for specific job roles. By leveraging advanced AI models, OPTEHIRE provides position-specific feedback, including a SWOT analysis (Strengths, Weaknesses, Opportunities, Threats) and key metrics, to help users craft impactful resumes and discover relevant job opportunities. The system features a Flask-based backend API and a React-based frontend, offering a seamless, user-friendly experience for resume improvement and job matching.

Features

- **Resume Upload:** Users can upload PDF, TXT, or DOCX resume files through an intuitive interface.
- **Position-Specific Feedback:** Generate tailored SWOT analysis and metrics (grammar score, hiring probability, skill coverage, ATS compatibility) for roles like Software Engineering, Data Science, or custom positions using AI models.
- **Feedback Display:** View detailed feedback in a points format, with a dashboard for metrics and a loader animation (using `react-loader-spinner`) during data fetching.
- **Job Listing Exploration:** Easily navigate to explore job listings via a dedicated button.
- **Timeout Handling:** Includes 10-second timeout protection for API requests, with user-friendly error messages.
- **Responsive UI:** Features a clean, modern layout designed with Tailwind CSS, including background images and responsive design.
- **CORS Support:** Backend supports CORS for `localhost:3000` and `localhost:5173`, enabling smooth frontend-backend communication.

Installation

Follow these steps to set up OPTEHIRE on your local machine:

Prerequisites

- **Python 3.8+** for the backend.
- **Node.js 14+** and **npm** for the frontend.
- Git for version control.

Backend Setup

1. Clone the repository:

```
git clone <repository-url>
cd optehire/backend/AI-DIY-Factory/Resume ATS Builder
```

2. Create a virtual environment and activate it:

```
python -m venv venv
source venv/bin/activate # On Windows: venv\Scripts\activate
```

3. Install backend dependencies:

```
pip install -r requirements.txt
```

4. Configure environment variables in `.env` (e.g., API keys for AI models like `crewai` and `langchain-openai`).

Frontend Setup

1. Navigate to the frontend directory:

```
cd optehire/frontend
```

2. Install Node.js dependencies:

```
npm install
```

3. Ensure Tailwind CSS is configured in your project (follow React Tailwind setup if not already included).

Usage

Running the Application

1. Start the backend server:

```
python resume_swot_analyse.py
```

The backend will run on `http://localhost:5000` by default.

2. Start the frontend development server:

```
npm start
```

The frontend will run on `http://localhost:3000` or `http://localhost:5173` (depending on your configuration).

Example Workflow

- **Uploading a Resume:** Visit the frontend, click "Upload Resume," and select a PDF, TXT, or DOCX file.
- **Submitting a Job Role:** Specify a role (e.g., "Web Development") and click "Submit Role" to generate feedback.
- **Fetching Feedback:** Wait for the loader to complete, then view the SWOT analysis and metrics on the feedback dashboard.
- **Exploring Jobs:** Click "Explore Jobs" to navigate to job listings.

Project Structure

Backend

Located at `c:\Users\SEC\Desktop\optehire\backend\AI-DIY-Factory\Resume ATS Builder\`, the backend includes:

- `resume_swot_analyse.py`: Main Flask API file for processing resumes and generating feedback.
- `agents.py`, `receiver.py`, `tasks.py`, `utils.py`: Supporting Python scripts for AI logic and utilities.
- `requirements.txt`: Dependency list.
- `data/`, `db/`: Directories for data storage and databases.
- `.env`: Environment configuration.

Frontend

Located at `c:\Users\SEC\Desktop\optehire\frontend\src\pages\`, the frontend includes:

- `feedback.jsx`: Main React component for displaying feedback and metrics.
- `explorejobs.jsx`, `Home.jsx`, `landing.jsx`: Pages for job exploration, home, and landing.
- `assets/`: Directory for UI images (e.g., background images, loader animations).
- `components/`: Reusable React components (e.g., `CommonNavbar.jsx`, `footer.jsx`).
- `public/`, `src/`: Standard React project structure with Tailwind CSS styling.

Technologies

- **Backend:** Python, Flask, `crewai`, `langchain-openai`, `flask-cors`.
- **Frontend:** React, Tailwind CSS, `react-loader-spinner`.
- **Other:** Git for version control, `Node.js`, `npm`.

Contributors

Frontend Development

Manoj MV - Responsible for building and maintaining the React-based user interface.

UI/UX Design

Aparna RB - Designed the responsive, user-friendly interface with Tailwind CSS and modern styling.

AI Development

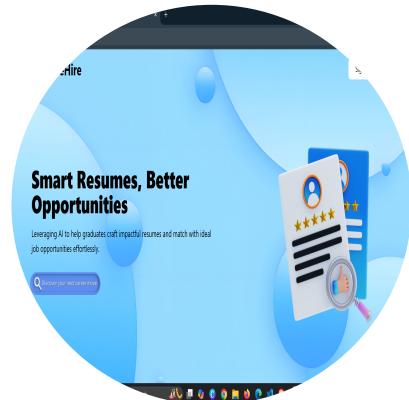
Vijis Durai R - Developed the AI-driven backend logic using CrewAI and LangChain OpenAI for resume analysis and feedback generation.

Job Fetching

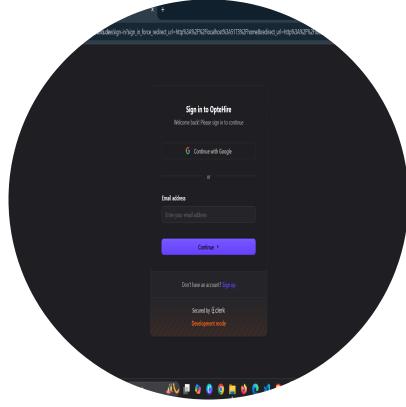
Dinesh Kumaraa K - Implemented the job listing exploration feature, enabling users to discover relevant opportunities.

Screenshots

Landing Page



Authentication Page



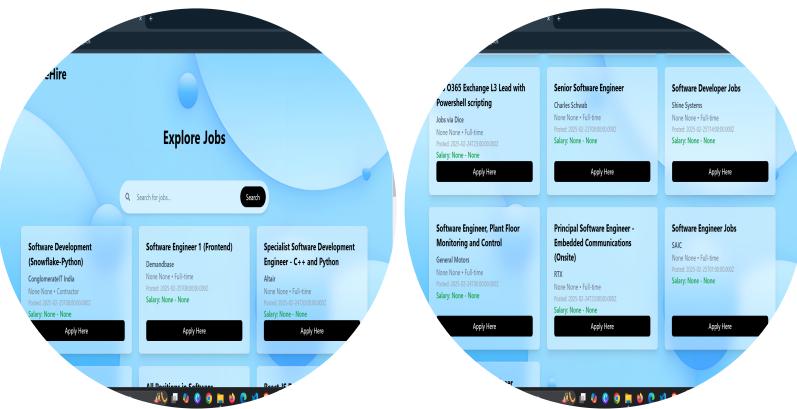
User Dashboard



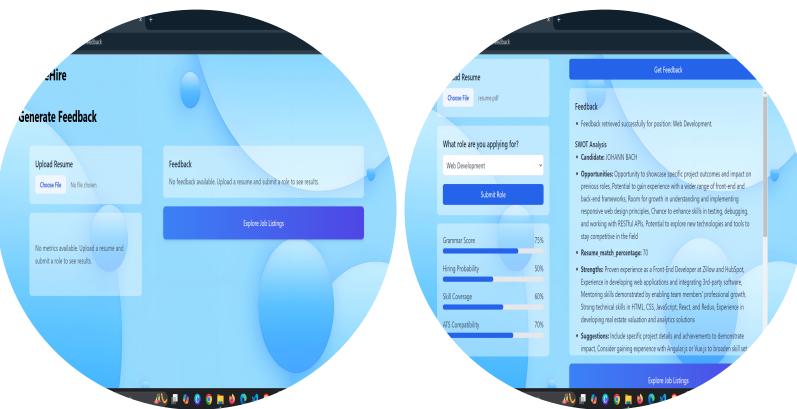
Home Page



Job Exploration Pages



Feedback Page



Backend AI Process

