# AI-Powered Resume Analyzer (React + FastAPI)

**Ayan Naskar**

**DT**

**Github:** https://github.com/ayan-naskar/ResumeAnalyzer

## Problem Statement

Develop a resume analysis system with AI-based text processing.

## Backend Implementation

### Technologies Used

* FastAPI for API development
* SQLAlchemy for database management
* PostgreSQL as the database

### Features

* Resume Upload: Users upload resumes, which are processed and stored in the database.
* Store Resume Data: Extracted details like name, email, skills, and experience are saved.
* Job Recommendations: Fetches job suggestions based on stored skills. (The list of jobs were predefined)

### Project Structure

The backend follows an MVC-inspired structure with separate modules for models, routes, controllers, and database connections.

backend/

│── app/

│ ├── main.py

│ ├── models/

│ │ ├── resume.py

│ ├── db/

│ │ ├── database.py

│ ├── controllers/

│ │ ├── resume\_controller.py

│ ├── repositories/

│ │ ├── resume\_repository.py

│ ├── routes/

│ │ ├── resume\_routes.py

### API Endpoints

* POST /upload/ – Uploads and processes resumes
* POST /store-resume/ – Saves resume details if not already in the database
* GET /recommend-jobs/{email} – Fetches job recommendations for a user

### Database

The Resume model stores name, email, phone, skills, experience, and education in a relational PostgreSQL database.

## Frontend Implementation

### Project Structure

/resume-app

│── /public

│── /src

│ │── /components

│ │ │── DragDropResume.tsx

│ │ │── EditableResumeFields.tsx

│ │ │── JobRecommendations.tsx

│ │── /pages

│ │ │── Home.tsx

│ │── /services

│ │ │── api.ts

│ │── App.tsx

│ │── main.tsx

│── package.json

│── tsconfig.json

│── index.html

### Screenshots of Working Project

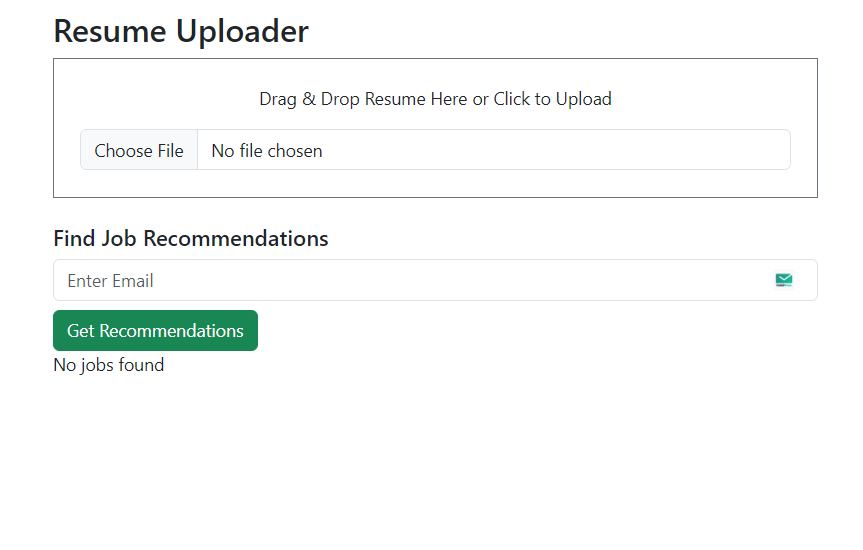
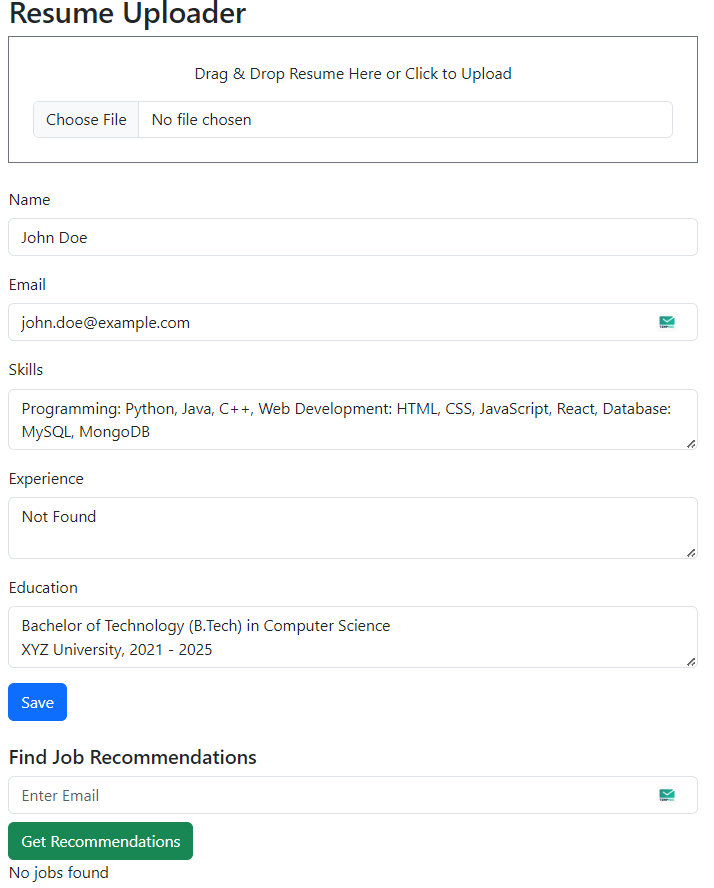


Figure 2 Home Page after dropping a resume

Figure 1 Home Page

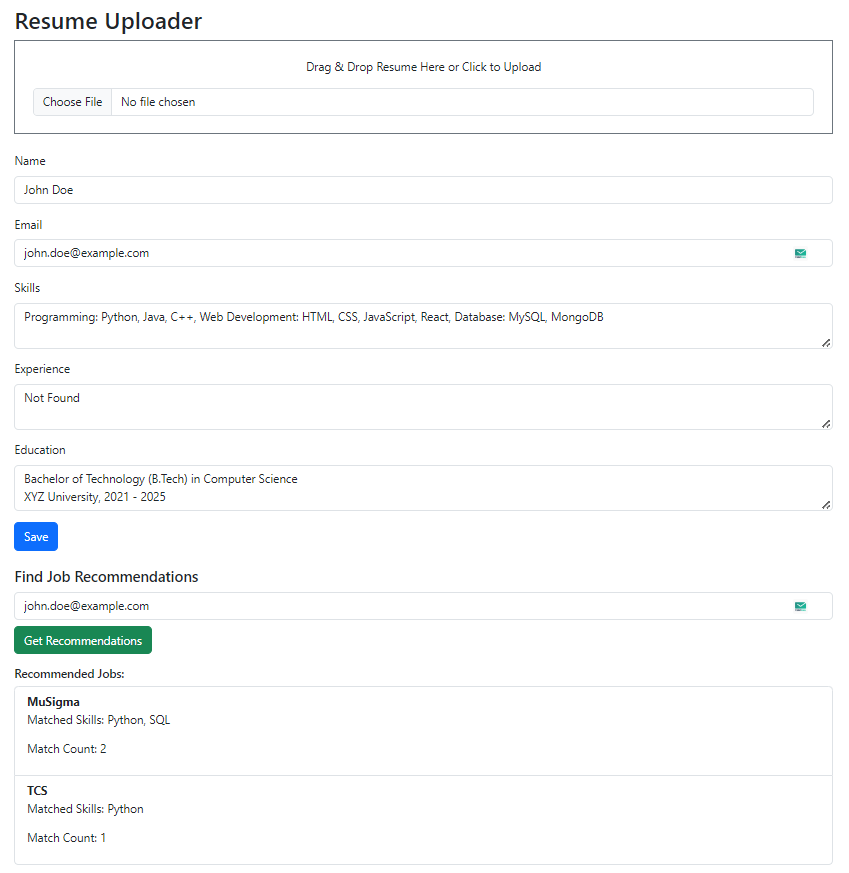


Figure 3 Home Page after asking for Job Recommendations

## Relevant Questions

How did you handle different resume formats and inconsistent data?

* Checked if it is docx or pdf, based on that, file is read and data is extracted.
* If data stored in pdf is not consistent, I have done regex and spacy NER methods to extract data

How did you optimize entity extraction for accuracy?

* First, I used Simple Regex but it didn’t give good results. So used Spacy NER. Did hit and trial of what worked best and went with it.