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

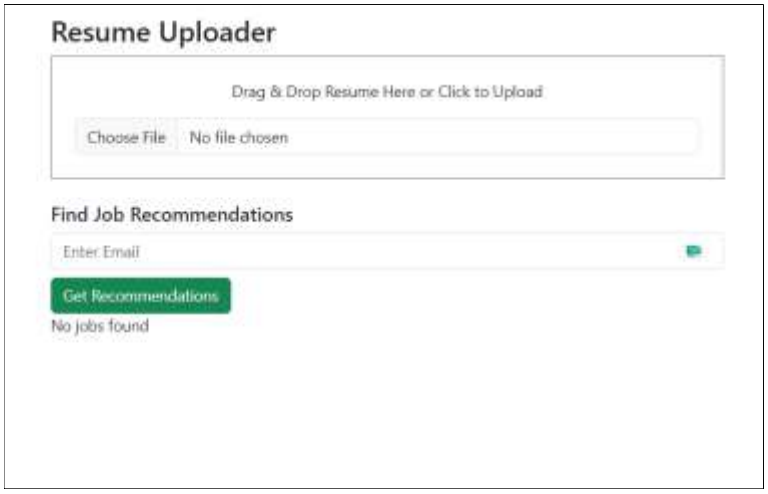
The Resume model stores name, email, phone, skills, experience, and education in a relational PostgreSQL database.

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


Resume Uploader

Drag & Drop Resume Here or Click to Upload

Choose File No file chosen

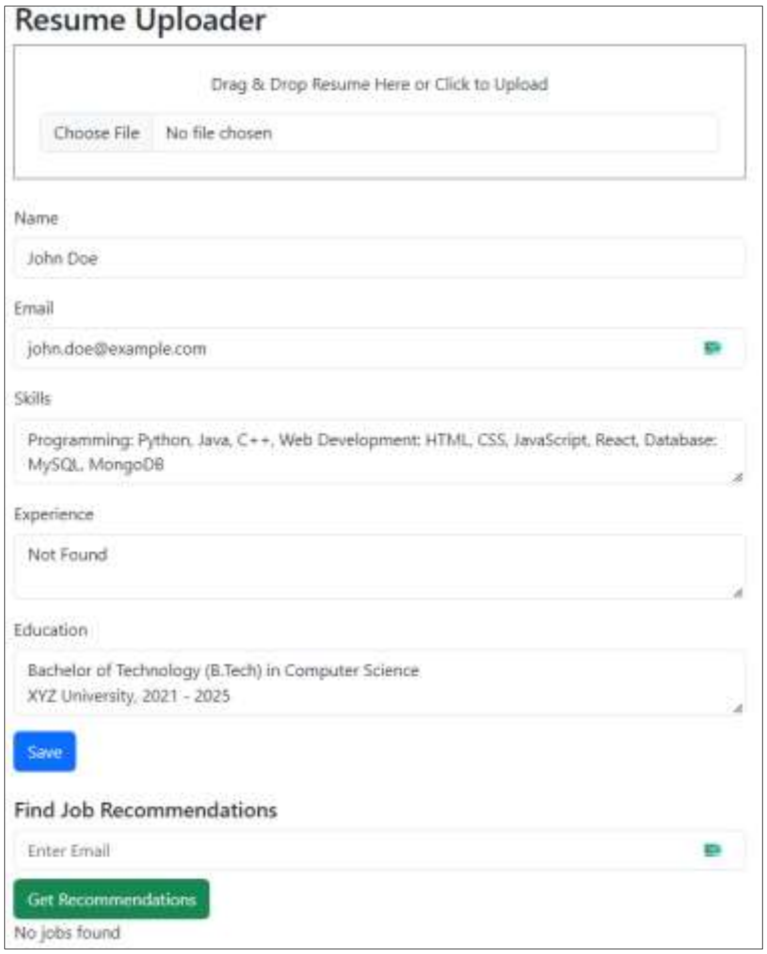
Find Job Recommendations

Enter Email

Get Recommendations

No jobs found

Figure 1 Home Page


Resume Uploader

Drag & Drop Resume Here or Click to Upload

Choose File No file chosen

Name

John Doe

Email

john.doe@example.com

Skills

Programming: Python, Java, C++, Web Development: HTML, CSS, JavaScript, React, Database: MySQL, MongoDB

Experience

Not Found

Education

Bachelor of Technology (B.Tech) in Computer Science
XYZ University, 2021 - 2025

Save

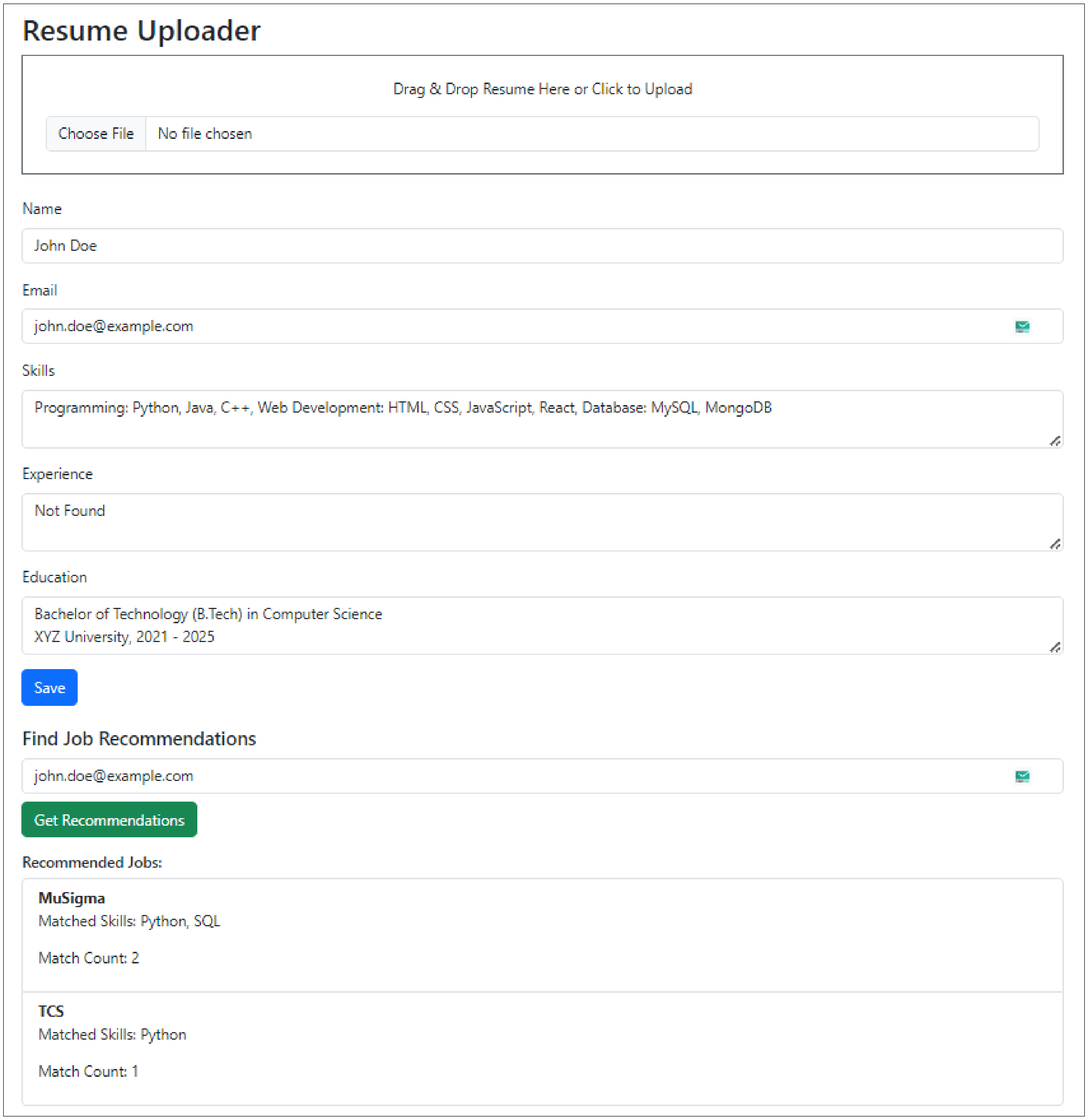
Find Job Recommendations

Enter Email

Get Recommendations

No jobs found

Figure 2 Home Page after dropping a resume



Resume Uploader

Drag & Drop Resume Here or Click to Upload

Choose File No file chosen

Name: John Doe

Email: john.doe@example.com

Skills: Programming: Python, Java, C++, Web Development: HTML, CSS, JavaScript, React, Database: MySQL, MongoDB

Experience: Not Found

Education: Bachelor of Technology (B.Tech) in Computer Science
XYZ University, 2021 - 2025

Save

Find Job Recommendations

john.doe@example.com

Get Recommendations

Recommended Jobs:

MuSigma
Matched Skills: Python, SQL

Match Count: 2

TCS
Matched Skills: Python

Match Count: 1

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.