Abstract

Finding the right resources during a job search can often be overwhelming for candidates. This project, *Job Seeker Resource Portal*, provides a centralized web platform that organizes essential tools and guidance in one place. Built with Flask and HTML, the portal categorizes resources into resume preparation, interview tips, career guidance, and job search platforms. The system is lightweight, user-friendly, and deployable online for easy access, helping job seekers streamline their preparation process.

Problem Statement

Job seekers often face challenges in navigating multiple platforms to find reliable information about resumes, interviews, and career development. The lack of a centralized hub leads to wasted time and inconsistent guidance. This project addresses that gap by creating a single, accessible platform where curated job-related resources are organized into categories, providing job seekers with a streamlined and structured experience.

Project Overview

Purpose

- **Objective**: Provide a centralized resource hub for job seekers.
- **Functionality**:
    - Web-based portal displaying job search resources.
    - Organized categories: Resume Building, Interview Preparation, Career Guidance, and Job Portals.
    - Accessible locally and via online deployment.

Key Features

- **Web Interface**: Simple and clean homepage (index.html).
- **Categorized Resources**: Direct links to job search tools and learning platforms.
- **Backend Support**: Flask server to handle page routing.
- **Deployment Ready**: Runs locally and deploys on Render with Gunicorn.
- **Live Demo**: https://job-seeker-resourse.onrender.com

Author

- **Name**: Varshini Kavisetti
- **GitHub**: https://github.com/varshinikavisetti/job-seeker-resourse

---

Frontend

- **Technologies**: HTML, CSS.
- **Pages**:
  - index.html → Displays categorized job seeker resources.
- **Design Goal**: Lightweight, accessible, and easy navigation.

Backend

- **Framework**: Flask.
- **Server Setup**:
  - app.py handles backend routing and serves the HTML template.
- **Deployment**:
  - Gunicorn as WSGI server.
  - Hosted on Render.

Project Structure

/Job_Seeker_resourse

```
├── app.py              # Flask backend server
├── requirements.txt    # Python dependencies
├── templates/
│   └── index.html      # Homepage UI
├── static/
│   └── style.css       # Optional stylesheet
├── README.md           # Documentation
```

Setup and Usage

Setup

git clone https://github.com/varshinikavisetti/job-seeker-resourse.git

cd job-seeker-resourse

```
python -m venv venv
venv\Scripts\activate      # Windows
# or source venv/bin/activate for Linux/Mac
pip install -r requirements.txt
```

Run Locally

```
python app.py
```

- Access at: http://127.0.0.1:5000

Deploy on Render

- Push repo to GitHub.
- Create a new Web Service on Render.
- Set **Start Command**:
- gunicorn app:app
- Live URL provided after deployment.

Additional Notes

- **License**: MIT (can be added).
- **Future Improvements**: Add login, personalized dashboards, or API-based live job listings.
- **Use Case**: Ideal for students, job seekers, and career guidance platforms.