

**Superior University Gold Campus**

**PAI Lab Project**

Name: Syed Ejiz Ul Hassan Kazmi

Roll. No: SU92-BSDSM-F23-025

Section: BSDS-4A

Instructor: Sir Rasikh

## **Resume Analyzer Web Application Using**

## **Flask and Machine Learning**

### ****Objective:****

The main goal of this project is to develop a web application that can:

* Predict the job category of a candidate based on their resume.
* Extract technical and soft skills from uploaded resumes.
* Provide skill-based filtering for recruiters.
* Support role-based access for admin and users.

## **Technologies Used:**

* **Programming Language:** Python
* **Web Framework:** Flask
* **Frontend:** HTML, CSS (with Jinja templating)
* **Machine Learning Model:** Random Forest Classifier
* **NLP Tools:** NLTK (stopwords, lemmatization)
* **File Handling:** PyPDF2
* **Storage:** CSV, JSON
* **Model Persistence:** Joblib
* **Resume Upload:** PDF format

## **Modules and Functionalities:**

### ****1. User Authentication & Role Management****

* Users can register and log in with a selected role (admin or user).
* User data is stored in users.csv.

### ****2. Resume Upload & Parsing****

* Users (with user role) upload a PDF resume.
* PDF is parsed using **PyPDF2**.
* Text is cleaned and preprocessed with **regex** and **NLTK**.
* A **machine learning model** predicts the most suitable job category.

### ****3. Skill Extraction****

* A large predefined list of skills is matched against the text.
* Extracted skills are stored with the user's resume information.

### ****4. Admin Dashboard****

* Admins can view all uploaded resumes and associated skills.
* Admins can filter resumes based on job category and required skills using a category-skills mapping generated from the training dataset.

### ****5. Data Storage****

* Resume files are stored in static/resumes/.
* Parsed resume data is stored in parsed\_resumes.json.
* User accounts are stored in users.csv.

## **Machine Learning Details:**

* The model is trained on UpdatedResumeDataSet.csv using text data labeled with job categories.
* The resume text is vectorized using **TfidfVectorizer**.
* A **Random Forest classifier** is trained and saved using joblib.

## **Key Python/Flask Concepts Used:**

* @app.route: Defines URL routes.
* redirect(): Used to redirect users after login or upload.
* render\_template(): Renders HTML pages with dynamic content.
* session: Stores login details across pages.
* secure\_filename(): Ensures filenames are safe for saving.
* request.files: Handles uploaded files.
* joblib.load(): Loads the trained ML model and vectorizer.

## **Security Considerations:**

* Input sanitization is done for text.
* Users are stored securely in CSV with roles.
* Only PDF files are allowed for upload.

## **Possible Enhancements:**

* Add password hashing for better security.
* Store data in a database (e.g., SQLite or PostgreSQL).
* Add resume feedback or score based on skill match.
* Email notifications to candidates or admin.

## **Conclusion:**

This project is a complete machine learning and web-based application that helps in automating resume screening. It demonstrates practical usage of Python, Flask, machine learning, and NLP.