**AI RESUME ANALYZER – FINAL PROJECT DOCUMENTATION**

**Project Overview**

The AI Resume Analyzer is a Streamlit-built interactive web tool that uses Natural Language Processing (NLP) methods to evaluate and grade user-uploaded resumes in PDF format.

**The system**:  
1. Extracts important information from resumes.  
2. Predicts the experience level and job domain of the applicant.  
3. Suggests suitable courses and other abilities.  
4. Evaluates the resume according to the completeness of its material.  
5. Offers materials for interview preparation and customized resume writing advice.  
6. Include a user input feedback module.  
7. Includes a CSV export feature and an analytics-rich admin dashboard.

**Important Features:**  
**Resume Parsing:** Uses sophisticated parsing libraries to extract personal information, education, employment history, and skills from submitted resumes.  
**Candidate profiling:** Uses extracted skills to predict job domains including UI/UX, Web Development, Android, iOS, and Data Science.  
**Experience Level Assessment:** An evaluation of a candidate's experience level establishes whether they are fresh, intermediate, or experienced.  
**Skill Recommendations:** Makes recommendations for extra training and abilities related to the anticipated employment domain.  
**Resume scoring:** Assesses resumes according to the inclusion of important parts such as Projects, Education, Experience, and Objective.  
**Feedback Gathering:** Enables users to rate and comment on the app.  
**Admin Dashboard:** Shows comprehensive user data, domain forecasts, and interactive graphic summaries of user input.  
**Bonus Advice:** Offers well-chosen videos for interview and resume preparation.

**Descriptions of Functions**:  
*get\_csv\_download\_link(df, filename, text)* is a helper function.  
Creates a download link for a CSV file containing a Pandas DataFrame.  
  
*pdf\_reader(file)*  
Pulls text from a PDF file for examination.  
  
*show\_pdf(file\_path)*  
Uses an integrated viewer to display a PDF in the Streamlit app.  
  
*course\_recommender(course\_list)*Shows a predetermined list of suggested courses with a chosen number.  
  
*insert\_data(...)*Adds information about user resume analysis to the user\_data MySQL table.  
  
*insertf\_data(...)*   
Adds user input to the user\_feedback MySQL table.

**Main Application Flow (run() function)  
User Section:**  
1. Gathers user information.  
2. Enables the uploading of resumes in PDF format.  
3. Extracts resume information, parses it, and displays it.  
4. Predicts experience level and domain.  
5. Recommends courses and skills.  
6. Scores the sections of resumes.  
7. Stores information in a database.  
8. Provides interview and resume writing advice.

**Section for Feedback:**  
1. Rating and commenting feedback form.  
2. Database-stored feedback.  
3. Shows the overall results in charts.

**About Section:**  
Shows the credits, usage guide, and project goal.

**Admin Section:**  
1. Secure login (admin/admin@resume-analyzer by default).  
2. Shows all user information and comments in tables.  
3. CSV export capability.  
4. Ratings, domains, experience levels, and geographic regions usage analytics charts.

**Database Setup:**

MySQL server must be running locally.

Database CV with tables user\_data and user\_feedback will be created automatically.

**Run Application:**streamlit run main\_app.py

**Navigation:**

**User**: View results and suggestions, upload a resume.  
**Feedback**: Leave comments and ratings about the app.  
**About**: Find out more about the undertaking.  
**Admin**: Export and view analytics (requires login).

**Notes and Known Problems:**  
1. Case-sensitive substring matching is used for resume scoring; lowercase versions might not be picked up.  
2. Admin credentials are hardcoded; they ought to be protected with a database or environment variables.  
3. To conserve space, uploaded files should be deleted on a regular basis.  
4. Admin panel execution problem: More debugging is needed because the admin area does not currently operate as intended.

**Future Improvements:**  
1. Increase the coverage of career domains.  
2. Use cutting-edge NLP models for parsing.  
3. Include user authentication.  
4. DOCX and TXT formats are supported.  
5. Turn on resume editing within the app.

**Connection to Course Content**

This project demonstrates:

**Imperative Programming** — procedural logic and modular functions.

**Object-Oriented Concepts** — structuring of helper functions and data handling.

**NLP Techniques** — tokenization, parsing, keyword matching.

**Database Integration** — MySQL for persistent storage.

**Data Visualization** — using Plotly for interactive charts.

**Application Screens**

**A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a web page

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.**