**Software Development Documentation**

**Project Name:** AI-Powered Resume Screening & Bias Reduction System  
**Team Members:** Ranya Saidi, Saif Eddine Ben Hadj Kacem  
**Course:** Data Science Projects in Business  
**Submission Date:** May 2025

**1. Project Overview**

This project addresses the inefficiencies and biases in traditional resume screening by developing an AI-powered solution that automates resume parsing, ranks candidates based on skills, and reduces unconscious bias through anonymized evaluation. The system integrates machine learning with a user-friendly dashboard for recruiters.

**2. Functional Requirements**

* Parse resumes using Natural Language Processing (NLP)
* Anonymize sensitive information (name, gender, ethnicity)
* Vectorize resume text using TF-IDF
* Score candidates using a Logistic Regression model
* Display ranked candidate profiles in a Streamlit dashboard
* Allow feedback and future integration with testing modules

**3. Non-Functional Requirements**

* Cloud-hosted (AWS or GCP) for scalability
* Real-time or batch processing with low latency
* GDPR-compliant handling of personal data
* Monitorable, retrainable, and easy to update
* Simple user interface with clear AI explanations

**4. System Architecture**

* **Frontend:** Streamlit-based dashboard
* **Backend:** Python application with Scikit-learn and SpaCy
* **Model Storage:** Pickled files (model.pkl, vectorizer.pkl)
* **Cloud Deployment:** AWS EC2 + S3 (or equivalent)
* **CI/CD:** GitHub + (optional) CodePipeline or GitHub Actions

**5. Technology Stack**

| **Component** | **Tool/Library** |
| --- | --- |
| Programming Language | Python 3.9 |
| NLP Processing | SpaCy, Scikit-learn, NLTK |
| Vectorization | TF-IDF (TfidfVectorizer) |
| Model | Logistic Regression |
| UI Framework | Streamlit |
| Data Storage | CSV / S3 Buckets |
| Model Persistence | joblib (.pkl files) |
| Deployment | AWS EC2 |
| Visualization (Testing) | Matplotlib, Seaborn |

**6. Data Flow**

1. Resume uploaded or read from dataset
2. Resume anonymized and preprocessed (lowercased, tokenized, cleaned)
3. Vectorized using TF-IDF
4. Passed to trained Logistic Regression model
5. Ranked output sent to Streamlit dashboard

**7. Security & Compliance**

* All data anonymized prior to model inference
* Personal identifiers removed using SpaCy entity recognition
* Complies with GDPR principles: purpose limitation, data minimization, and confidentiality
* Resume files are not stored permanently unless anonymized

**8. Testing & Validation**

* Unit Tests: Resume cleaner, vectorizer, model scorer
* Integration Tests: End-to-end processing (resume → rank)
* Performance Metrics: Accuracy, F1-score, bias audit
* Human Evaluation: Recruiter review of output quality

**9. Model Performance Summary**

* **Algorithm:** Logistic Regression
* **Accuracy:** ~82% on validation data
* **Bias Testing:** Model trained on anonymized features to reduce demographic influence
* **Explainability:** Model output includes keyword match indicators

**10. Deployment Plan Summary**

* **Environment:** AWS EC2 (inference), Streamlit dashboard
* **Security:** HTTPS, encryption, IAM access controls
* **Monitoring:** Logs, feedback loop for retraining
* **Update Schedule:** Model retraining every 1–3 months with new data

**11. User Roles**

* **Recruiters**: End users of the dashboard
* **Data Team**: Maintains and monitors the model
* **HR/Diversity Team**: Reviews compliance and fairness
* **Business Sponsors**: Oversee ROI and adoption metrics

**12. Change Management Strategy**

* Internal training for dashboard use
* Pilot rollout with selected recruiters
* Documentation provided for handover and onboarding
* Stakeholder feedback loop included in post-deployment phase

**13. Known Limitations**

* Model currently uses resume content only (no cover letter or social data)
* Bias mitigation is based on removal, not adversarial learning
* Limited support for multilingual resumes
* System assumes structured resumes or structured parsing

**14. Future Work**

* Integrate skill-based test results and pre-screening quizzes
* Implement a fairness-aware learning algorithm (e.g., reweighing or adversarial debiasing)
* API integration with Applicant Tracking Systems (ATS)
* Add support for multilingual and audio/video resumes