PARSE | MATCH | ENRICH

Templatization / Design Resume

with RChilli

June 2024 | Version 1.0

Purpose Statement

This document details integrating **Templatization / Design Resume** use case leveraging RChilli Plugin – Resume Templater API.

Disclaimer:

Licensed Materials - Property of RChilli

For information about RChilli trademarks, copyrights, and patents, refer to the RChilli Intellectual Property page (https://www.rchilli.com/) on the RChilli website. All other trademarks and registered trademarks are the property of their respective holders.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of RChilli Inc. Under the law, reproducing includes translating into another language or format.

Every effort has been made to ensure that the information in this manual is accurate. RChilli Inc. is not responsible for printing or clerical errors. Information in this document is subject to change without notice.

If you have any comments or feedback on our documentation, please send them to us at support@rchilli.com

Contents

Introduction

The **Templatization / Design Resume** feature improves nonbiased hiring practices by automatically creating redacted and templatized resumes. This also works great for branding resumes and removing contact information for staffing and recruiting firms.

The process involves getting the resume from the candidate, redacting it using the RChilli Templater Plugin API, which returns a redacted and templatized document. This document is then stored in the database. When a recruiter checks the resumes for a job opening, they will get the redacted resume in the same template.

Impact:

- Speeds up the candidate review process by up to 70%

Key Components and Flow

1. **Resume Reception:** The candidate interacts with the interface and uploads their resume.
   a. **Action:** The candidate submits their resume.
   b. **Response:** The resume is uploaded to the server for processing.
2. **Redaction & Templatization Using RChilli API:**
   a. **Action:** The system sends the resume to the RChilli Templater API.
   b. **Response:** The API processes the resume, redacts sensitive information, and applies the predefined template. The redaction will be applied to fields, as mentioned below:
      i. Candidate Name
      ii. Candidate Email
      iii. Address
      iv. Phone Number
      v. Employment and Education Details
3. **Storing Redacted and Templatized Resume:**
   a. **Action:** The system stores the redacted and templatized resume in the database.
   b. **Response:** The document is saved and indexed for future reference.
4. **Return Redacted Document:**
   a. **Action**: Recruiters access the redacted resumes when reviewing candidates for job openings.
   b. **Response:** Recruiters receive redacted and templatized resumes for review.

Benefits

1. **Nonbiased Hiring:** Reduces bias-related incidents by 80%.
2. **Efficiency:** Speeds up the candidate review process by up to 70%, improving overall productivity.
3. **Branding:** Helps staffing and recruiting firms maintain consistent resume formats and remove unnecessary contact information.

User Stories

1. **Upload Resume**
    a. As a candidate, I want to upload my resume so that I can apply for a job without manually entering my information.
2. **Redact and Templatize Resume**
    a. As a system, I need to redact and templatize the uploaded resume to ensure uniformity and protect candidate privacy.
3. **Store Redacted Resume**
    a. As a system, I need to store the redacted resume in the database to keep a record of the redacted information.
4. **Retrieve Redacted Resume**
    a. As a recruiter, I want to access redacted and templatized resumes to make unbiased hiring decisions.

Example Scenario

1. **Scenario:** A candidate named Alex submits a resume through the application portal.
2. **Action:** The system sends Alex's resume to the RChilli Templater Plugin API.
3. **Response:** The API redacts personal information, applies a template, and returns the processed resume.
4. **Outcome:** The system stores the processed resume, and when a recruiter searches for candidates, they receive the redacted and templatized version of Alex's resume.

User Acceptance Testing (UAT)

The **Templatization / Design Resume** UAT plan ensures the feature meets user requirements and business goals. It validates that candidates can apply for jobs efficiently by uploading their resumes, which are then redacted & templatized properly, stored in the database and return the same templatized resume.

UAT Scenarios

UAT Scenario 1: Resume Redaction and Templatization

- **Objective:** Verify that the system correctly redacts and templatizes uploaded resumes.
- **Preconditions:** Candidate has a resume file ready for upload.
- **Steps:**
  - Navigate to the resume submission page.
  - Upload a resume file.
  - Check the processed resume in the database.
- **Expected Result:** The resume is redacted, templatized, and stored correctly.
- **Acceptance Criteria:**
  - The processed resume has no personal information.
  - The processed resume follows the predefined template.

UAT Scenario 2: Recruiter Access to Redacted Resumes

- **Objective:** Ensure recruiters can access redacted and templatized resumes.
- **Preconditions**: Processed resumes are stored in the database.
- **Steps:**
  - Log in as a recruiter.
  - Search for candidates.
  - View the resumes of shortlisted candidates.
- **Expected Result:** Recruiters receive redacted and templatized resumes.

- **Acceptance Criteria:**
  - Resumes are consistently redacted.
  - Resumes are in the predefined template format.

UAT Scenario 3: Error Handling for Unsupported File Formats

- **Objective:** Test the system's response to unsupported file formats.
- **Preconditions:** Candidate attempts to upload an unsupported file format.
- **Steps:**
  - Navigate to the resume submission page.
  - Upload a file in an unsupported format (e.g., Zip, CSV).
- **Expected Result:** The system displays an error message.
- **Acceptance Criteria:**
  - The system does not accept unsupported formats.
  - An appropriate error message is shown to the user.

UAT Scenario 4: Performance Testing for High Volume Uploads

- **Objective:** Verify the system performance with high volume resume uploads.
- **Preconditions:** Bulk upload of multiple resumes.
- **Steps:**
  - Navigate to the bulk resume upload page.
  - Upload a zip file containing multiple resumes.
- **Expected Result:** All resumes are processed without performance degradation.
- **Acceptance Criteria:**
  - System processes all resume in the bulk upload.
  - Processing time is within acceptable limits.

UAT Scenario 5: Data Integrity Check

- **Objective**: Ensure that the data remains consistent and accurate after processing.
- **Preconditions**: Resume is uploaded and processed.

- **Steps**:
    - Upload a resume.
    - Check the processed resume for accuracy in redacted information.
- **Expected Result**: Data remains accurate and consistent.
- **Acceptance Criteria**:
    - All information remains accurate post-redaction and templating.

UAT Scenario 6: Security Testing

- **Objective**: Verify that sensitive information is securely handled and stored.
- **Preconditions**: Resume contains sensitive information.
- **Steps**:
    - Upload a resume with sensitive information.
    - Monitor the handling and storage process.
- **Expected Result**: Sensitive information is redacted and securely stored.
- **Acceptance Criteria**:
    - Sensitive information is not exposed.
    - Data is securely stored in the database.

Test Cases

Test Case 1: Resume Submission Test

- **Objective:** Verify that a resume can be submitted successfully.
- **Preconditions:** Candidate has a resume file ready for submission.
- **Steps:**
  a. Submit the resume through the job application portal.
  b. Monitor the process to ensure the resume is sent to the RChilli redaction API.
- **Expected Result:** Resume submission complete without errors.
- **Pass Criteria:** Resume is submitted and sent to the RChilli redaction API successfully.

Test Case 2: Redaction and Templatization Test

- **Objective:** Verify that the system redacts and templatizes resumes correctly.
- **Preconditions:** Uploaded resume is ready for processing.
- **Steps:**
  a. Upload a resume.
  b. Check the processed resume in the database.
- **Expected Result:** Resume is redacted and templatized.
- **Pass Criteria:** Redacted resume includes correctly redacted fields & follows the predefined template.

Test Case 3: Recruiter Resume Access Test

- **Objective:** Verify that recruiters can access redacted and templatized resumes.
- **Preconditions**: Processed resumes are stored in the database.
- **Steps:**
  a. Log in as a recruiter.
  b. Search for candidates.
  c. View candidate resumes.

- **Expected Result:** Recruiters see redacted and templatized resumes.
- **Pass Criteria:** Recruiters receive correctly processed resumes.

Test Case 4: Unsupported Resume Format Handling Test

- **Objective:** Verify that the system handles unsupported resume formats gracefully.
- **Preconditions:** Resume with an unsupported format.
- **Steps:**
  a. Submit a resume with an unsupported format through the job application portal.
  b. Monitor the process to see how the system handles the format.
- **Expected Result:** Appropriate error message is generated.
- **Pass Criteria:** System does not crash, and clear error messages are provided.

Test Case 5: High Volume Resume Processing Test

1. **Objective:** Ensure the system can handle a high volume of resume submissions efficiently.
2. **Preconditions:** Multiple resumes submitted to the system.
3. **Steps:**
   a. Submit multiple resumes through the job application portal.
   b. Monitor processing time and system response.
4. **Expected Result:** All resumes are processed within acceptable time limits**.**
5. **Pass Criteria:** Processing time is within acceptable limits, and system performance remains stable.

Test Case 6: Data Accuracy Test

- **Objective**: Ensure the accuracy of redacted and templatized information.
- **Preconditions**: Resume is uploaded and processed.
- **Steps**:
  a. Upload a resume.
  b. Verify the redacted and templatized information.

- **Expected Result**: Information is accurate and consistent.
- **Test Data**: Sample resumes with various details.
- **Pass Criteria**: All information remains accurate post-processing.

Test Case 7: Resume Size Limit Handling

- **Objective**: Ensure system handles large resume files properly.
- **Preconditions**: Candidate attempts to upload a large resume file.
- **Steps**:
  a. Try to upload a resume file exceeding the maximum size limit.
- **Expected Result**: System displays an error message regarding file size.

- **Pass Criteria**: Error message is shown, and upload is prevented.

Note: - Please create the other error code test cases according to the Resume Templater API, click here

Miscellaneous

The integration depends on your application workflow. There might be different steps involved in your integration which may vary from application to application.

This document indicates basic steps which are involved, you can review your application and add other UAT, and test cases based upon your needs.

Integration Estimation

Disclaimer: This estimation is based on a sample application with simple workflow. This may vary depending on the complexity of your application and the skill set/experience of the team involved in the integration.

1. **Development (16hr – 22hr)**
    a. Upload Resume: 2hr
    b. Redact and Templatize using RChilli API: 3hr
    c. Store Processed Resume in Database: 3hr
    d. Retrieve and Display Redacted Resume for Recruiters: 4-5hr
    e. Unit Test Cases: 2-3hr
    f. Bug Fixing and Optimization: 2-4hr
2. **QA (5hr - 7hr)**
    a. UAT: 1-2hr
    b. Executing Test Cases: 4-5hr

Appendix

Resume Templater API Details:

https://docs.rchilli.com/kc/Resume_Templater_plugin

Get API key and Endpoints:

https://docs.rchilli.com/kc/Resume_Templater_plugin_endpoint_para

Postman Integration and Sample Code:

https://documenter.getpostman.com/view/6003669/Szmh1GQX?version=latest#0cb8fa37-28c1-4e2b-9ea0-75d130a7dc41

Resume Templator API Error Code:

https://docs.rchilli.com/kc/Resume_Templater_plugin_error_codes

Sample Codes:

https://docs.rchilli.com/kc/Resume_Templater_plugin_sample_codes