

Software Requirements Specification (SRS)

Project Title: AI-Powered Talent Scout – Resume Analyzer + Interview Scheduler

Author: Simran Shaikh

1. Introduction

1.1 Purpose

The purpose of this document is to outline the software requirements for the "AI-Powered Talent Scout" application. This software aims to automate the recruitment workflow, providing resume parsing, candidate evaluation, and interview scheduling capabilities using AI and modern web technologies.

1.2 Scope

The AI-Powered Talent Scout is a web-based application that helps recruiters and HR professionals manage their hiring pipeline efficiently. Key functionalities include:

- Parsing resumes (PDF format)
- Matching candidate skills to job requirements
- Scheduling interviews via Jitsi Meet
- Sending automated email invites
- Providing feedback on skill gaps for unqualified candidates

This application is designed for small to medium-sized enterprises looking for cost-effective, automated recruitment solutions.

1.3 Definitions, Acronyms, and Abbreviations

- **SRS:** Software Requirements Specification
- **UI:** User Interface
- **SMTP:** Simple Mail Transfer Protocol
- **PDF:** Portable Document Format
- **Jitsi:** Open-source video conferencing solution
- **HR:** Human Resources
- **Streamlit:** Python framework for building web apps

1.4 References

- Streamlit Documentation: <https://docs.streamlit.io>
 - PyPDF2 Library: <https://pypdf2.readthedocs.io>
 - Jitsi Meet API Docs: <https://jitsi.github.io/handbook/>
 - Python SMTP Library: <https://docs.python.org/3/library/smtplib.html>
-

2. Overall Description

2.1 Product Perspective

This system is a standalone application that integrates third-party services like Gmail (SMTP) and Jitsi for video conferencing. It provides a modular architecture allowing easy updates to roles and skillsets.

2.2 Product Functions

- **Resume Upload:** Users can upload candidate resumes in PDF format.
- **Resume Parsing:** Extracts text and identifies skill keywords.
- **Skill Matching:** Compares resume content to job-specific skills.
- **Auto-Scheduling:** If matched, creates Jitsi meeting link and email.
- **Manual Scheduling:** User-specified interview date and time.
- **Email Notifications:** Sends formatted invites via SMTP.
- **Skill Gap Feedback:** Identifies and displays missing skills for rejected candidates.

2.3 User Characteristics

- **HR Professionals:** Familiar with basic technology, responsible for uploading resumes and scheduling interviews.
- **Candidates:** Receive email invites and attend interviews through Jitsi Meet.

2.4 Constraints

- Internet connectivity is required
- Resumes must be in PDF format
- Requires Gmail account with App Password setup
- All communication must adhere to privacy and data security standards

2.5 Assumptions and Dependencies

- Users have Python 3.8+ installed
 - Users follow correct Gmail setup instructions for SMTP
 - Jitsi Meet service is available and accessible
-

3. Specific Requirements

3.1 Functional Requirements

- **FR1:** The system shall allow uploading of one or more resumes in PDF format.
- **FR2:** The system shall parse uploaded PDFs using the PyPDF2 library.
- **FR3:** The system shall match parsed skills against job role configurations in config.py.
- **FR4:** If a candidate meets skill requirements, a Jitsi link shall be generated automatically.
- **FR5:** The system shall send interview invites via email using Gmail SMTP.
- **FR6:** The system shall display missing skills for candidates not meeting requirements.
- **FR7:** The system shall allow HR users to manually schedule interviews.

3.2 Non-Functional Requirements

- **NFR1:** The UI shall load within 2 seconds on a high-speed internet connection.
 - **NFR2:** Emails shall be sent within 1 second of interview scheduling.
 - **NFR3:** The application shall provide a responsive and user-friendly UI.
 - **NFR4:** Skill-matching algorithms shall be case-insensitive and robust.
-

4. External Interface Requirements

4.1 User Interface

The Streamlit interface includes:

- Resume Analyzer section with file upload
- Interview Scheduler for both auto and manual modes
- About page with project details and instructions

4.2 Hardware Interfaces

- Standard computing device with internet access

- No specific hardware dependencies

4.3 Software Interfaces

- Python 3.8+
- Streamlit 1.13.0
- PyPDF2 2.11.1
- Gmail SMTP
- Jitsi Meet API

4.4 Communication Interfaces

- HTTPS for all communications
- SMTP for email services

5. System Architecture

5.1 Modules

- app.py: Main Streamlit application
- utils/config.py: Defines job roles and required skills
- utils/resume_parser.py: Parses text from PDFs
- utils/jitsi_scheduler.py: Schedules meetings via Jitsi
- utils/email_sender.py: Sends interview emails

5.2 Data Flow

User -> Upload Resume -> Parser -> Skill Matcher

|

[Qualified]--> Jitsi + Email Scheduler

[Unqualified]--> Skill Gap Feedback

6. Installation and Setup Guide

6.1 Prerequisites

- Python 3.8 or higher
- Gmail account (with App Password enabled)

6.2 Setup Instructions

```
git clone https://github.com/SimranShaikh20/AI-Powered-Talent-Scout.git
```

```
cd AI-Powered-Talent-Scout
```

```
pip install -r requirements.txt
```

```
streamlit run app.py
```

6.3 Gmail Configuration

- Enable 2-Step Verification
 - Generate App Password for "Mail"
 - Use the generated password in email_sender.py
-

7. Future Enhancements

- Integration with ATS platforms like Greenhouse or Lever
 - NLP-based advanced resume scoring
 - Admin dashboard with analytics and candidate tracking
 - Multiple language support for international hiring
-

8. Appendices

- Resume format must be PDF
- Jitsi Meet API usage is subject to fair usage policy
- Gmail App Passwords must be stored securely