# SMART JOB PORTAL WITH RESUME SCREENING USING AI

PROJECT REPORT 2025-2026

Submitted By


**24590072 HARUNI R**

**24590076 MUTHAZHAGI E**

**24590077 NAMEETHA N**

**24590080 PRATHEEP P**


Under the guidance of

**Mr. M.MUGILAN B.E.,(CSE)**

LECT/IT

Diploma in Information Technology

of the Directorate of Technical Education, Government of Tamil Nadu



DEPARTMENT OF INFORMATION TECHNOLOGY

**AKT MEMORIAL POLYTECHNIC COLLEGE**

KALLAKURICHI-606202.

# AKT MEMORIAL POLYTECHNIC COLLEGE

KALLAKURICHI-606202

## Department of Information Technology

### BONAFIDE CERTIFICATE

This is certified that this project work entitled **Smart Job Portal with Resume Screening Using AI** has been submitted **HARUNI R, MUTHAZHAGI E, NAMEETHA N, PRATHEEP P** in the partial fulfilment of the requirements for the award of Diploma in Information Technology during the academic year 2025-2026, who carried out the project work under our supervision.

| | |
|---|---|
| **Project Guide** | **Head of the Department** |
| **Mr.M.MUGILAN B.E.,(CSE)** | **Mrs. C.SRIDEVI C B.E.,(CSE)** |
| **LECT/COMPUTER** | **HOD- COMPUTER & IT** |

This is to certify that_____ was examined for the project work viva-voce held on_____

INTERNAL EXAMINER EXTERNAL EXAMINER

# Acknowledgement

We express our sincere and profound thanks to our chairman **Mr.A.K.T.Mahendiran B.Com.,** for creating this magnificent edifice of learning by providing all necessary facilities to complete diploma engineering course successfully.

We express our gratitude to our principal **Mr.P.K.Kabilar M.E.,MBA.,** for constant encouragement and facilities provided towards the successful completion of our project work.

At the same time we would like to express our heartfelt thanks to our head of the department **Mrs.C.Sridevi.,B.E.,(CSE).,** and also our project guide **Mr. M.MUGILAN B.E.,(CSE)** for guiding and needful advices and time to complete this project.

We would like to show our gratitude to our department staffs for all instructions and guidance given throughout.

Our sincere thanks and affection to our parents and friends who gave hand in all our steps.

# CONTENTS

# ABSTRACT

The Smart Job Portal with Resume Screening Using AI is an advanced web-based recruitment system designed to streamline and automate the hiring process for both employers and job seekers. Traditional recruitment methods often involve manual resume screening, which is time-consuming, prone to human bias, and inefficient when dealing with large volumes of applications. This project addresses these challenges by integrating Artificial Intelligence (AI) to enhance accuracy, efficiency, and decision-making in recruitment.

The system enables job seekers to create profiles, upload resumes, and apply for job openings, while employers can post job listings and manage applications through an intuitive dashboard. The core feature of this platform is its AI-powered resume screening module, which analyzes uploaded resumes using Natural Language Processing (NLP) techniques. It extracts key information such as skills, education, experience, and keywords, and matches them against job requirements to generate a relevance score for each candidate.

Additionally, the system provides automated ranking of candidates, shortlisting suggestions, and intelligent recommendations, reducing the workload on recruiters. It also ensures fairness by minimizing human bias during the initial screening process. The platform can be implemented using modern web technologies such as PHP/Laravel, MySQL, and AI APIs for text analysis and scoring.

Overall, the Smart Job Portal enhances recruitment efficiency, improves candidate-job matching accuracy, and provides a scalable solution for modern hiring needs. This system is particularly beneficial for organizations seeking faster, smarter, and more reliable recruitment processes in a competitive job market.

# 1. INTRODUCTION

The rapid growth of digital technology has significantly transformed the recruitment process, making online job portals a primary platform for connecting employers and job seekers. However, traditional job portals still rely heavily on manual resume screening, which is time-consuming, inefficient, and often unable to handle the increasing volume of applications. Recruiters frequently face challenges in identifying the most suitable candidates from hundreds or thousands of resumes, leading to delays in hiring and potential loss of talent.

To address these challenges, the Smart Job Portal with Resume Screening Using AI is proposed as an intelligent solution that automates and enhances the recruitment workflow. This system leverages Artificial Intelligence (AI), particularly Natural Language Processing (NLP), to analyze and evaluate resumes efficiently. By extracting relevant information such as skills, qualifications, experience, and keywords, the system can match candidate profiles with job requirements and assign a compatibility score, enabling recruiters to make faster and more informed decisions.

The platform provides a user-friendly interface for both job seekers and employers. Job seekers can register, create profiles, upload resumes, and apply for jobs, while employers can post job vacancies, define required skills, and view ranked candidate lists. The AI-driven screening mechanism ensures that only the most relevant candidates are shortlisted, reducing manual effort and minimizing bias in the initial selection process.

In addition to resume screening, the system can offer intelligent recommendations, automated notifications, and analytics to improve the overall recruitment experience. By integrating modern web technologies such as PHP/Laravel, MySQL, and AI APIs, the proposed system delivers a scalable, efficient, and smart hiring solution.

Overall, this project aims to revolutionize traditional recruitment methods by introducing automation, accuracy, and intelligence, making the hiring process faster, fairer, and more effective for organizations and job seekers alike.

# 2. LITERATURE SURVEY

. The recruitment industry has undergone major transformation with the adoption of digital platforms and intelligent automation. Earlier hiring systems were mostly manual, where recruiters collected printed resumes, reviewed qualifications individually, and shortlisted candidates based on personal judgment. This traditional approach consumed significant time and resources, especially when the number of applicants was large. With the emergence of online job portals, organizations were able to publish vacancies digitally and receive applications faster, but the challenge of manually screening resumes still remained a major limitation.

Several researchers and software developers have worked on online recruitment systems that simplify job posting, candidate registration, and application tracking. These systems improved communication between employers and job seekers, but most of them focused only on data management and lacked intelligent candidate evaluation. As a result, recruiters still had to spend considerable effort in identifying the most suitable applicants. This created a demand for smart recruitment platforms capable of automatically analyzing resumes and matching them with job descriptions.

Recent studies in Artificial Intelligence (AI) and Natural Language Processing (NLP) have shown promising results in automating resume screening. AI-based systems can extract structured information from unstructured resumes, such as personal details, education, technical skills, certifications, and work experience. NLP techniques help in understanding the context of resume content and comparing it with job requirements. Many modern recruitment tools use keyword extraction, semantic analysis, and machine learning models to rank candidates according to job relevance. These approaches significantly reduce screening time and improve the efficiency of shortlisting.

Another important area of research is job-resume matching algorithms. Traditional keyword-based matching methods often fail to recognize related skills or equivalent qualifications if exact words are not present. To overcome this, advanced models use

semantic similarity, ontology-based skill mapping, and AI scoring methods to better understand the meaning of job descriptions and candidate profiles. These intelligent matching techniques provide more accurate recommendations and help recruiters identify hidden talent that may be missed by simple keyword filters.

Researchers have also highlighted concerns regarding bias and fairness in recruitment systems. Manual hiring processes may be influenced by human prejudice, while poorly designed AI systems may inherit bias from training data. Therefore, modern literature emphasizes the importance of transparent, explainable, and fair AI models in recruitment. A well-designed smart job portal should focus not only on automation and speed but also on ethical candidate evaluation and equal opportunity.

Existing recruitment platforms such as LinkedIn Jobs, Indeed, and other applicant tracking systems have introduced automated filtering and recommendation features, but many small and medium organizations still lack access to affordable AI-based hiring tools tailored to their needs. This creates an opportunity to develop a Smart Job Portal with Resume Screening Using AI that combines job portal functionality with intelligent screening, ranking, and recommendation features in a single system.

From the literature, it is evident that integrating AI into job portals can greatly improve recruitment efficiency, reduce manual workload, enhance candidate-job matching, and support better hiring decisions. Based on these findings, the proposed system aims to build a practical and intelligent recruitment platform that addresses the limitations of traditional and existing online hiring methods.

# 3. PROPOSED SYSTEM

The Smart Job Portal with Resume Screening Using AI is proposed as an intelligent web-based recruitment platform that automates and improves the overall hiring process. The main objective of this system is to reduce the manual effort involved in resume screening and candidate shortlisting by using Artificial Intelligence techniques. The proposed system acts as a bridge between job seekers and recruiters by providing a centralized platform where job applications can be managed efficiently and accurately.

In this system, job seekers can register, create their profiles, upload resumes, search for available jobs, and apply online. The uploaded resumes are stored securely in the database and processed by the AI-based resume screening module. This module extracts important information such as candidate name, contact details, educational qualifications, technical skills, certifications, projects, and work experience. The extracted data is then structured and prepared for comparison with job descriptions posted by employers.

On the other side, recruiters or employers can create company accounts, post job vacancies, define eligibility criteria, specify required skills, and manage received applications through a dedicated dashboard. Instead of manually reviewing every resume, the system automatically compares candidate resumes with the job requirements using AI and NLP techniques. Based on this analysis, the system generates a matching score for each applicant and ranks candidates according to their relevance to the job role.

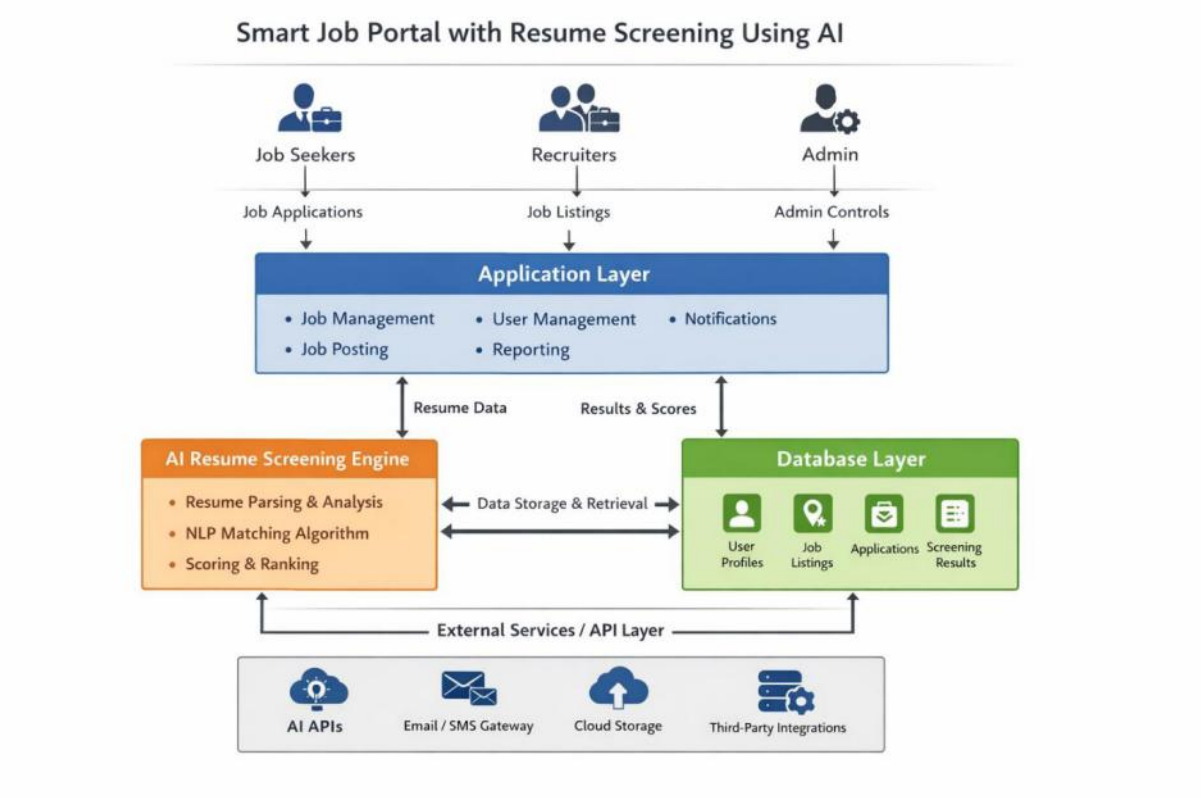
The proposed system includes an AI-powered recommendation engine that helps recruiters quickly identify the most suitable candidates. It can shortlist applicants automatically, highlight missing skills, and provide resume-job matching insights. This not only saves time but also improves hiring accuracy. For job seekers, the system can suggest relevant job opportunities based on their resume content, qualifications, and skill sets, thereby increasing the chances of suitable placements.

To improve usability and communication, the system also provides notification and tracking features. Job seekers receive alerts about application status, interview schedules, and new job postings, while recruiters are notified when new qualified applications are received. The system maintains complete records of jobs, applications, shortlisted candidates, and hiring history for future reference and reporting.

The proposed system is designed using modern web technologies such as Laravel/PHP for backend development, MySQL for database management, and AI/NLP APIs or Python-based AI models for resume parsing and scoring. The architecture is modular, scalable, and suitable for deployment in academic, startup, and enterprise recruitment environments.

Overall, the proposed system offers a smart, fast, and reliable recruitment solution by combining traditional job portal functionalities with intelligent AI-based screening and recommendation capabilities. It minimizes recruitment delays, reduces manual errors, improves candidate-job matching, and provides a more effective hiring experience for both employers and applicants.

## 3.1 ARCHITECTURE




The architecture of the Smart Job Portal with Resume Screening Using AI is designed as a multi-layered system that integrates user interaction, application processing, database management, and artificial intelligence-based resume analysis. The system follows a modular architecture so that each component performs a specific function while working together as a complete recruitment platform. This design improves scalability, maintainability, and performance.

At the first layer, the Presentation Layer provides interfaces for three main users: Job Seekers, Recruiters, and Admin. Job seekers use the portal to register, log in, create profiles, upload resumes, search for jobs, and track applications. Recruiters use their dashboard to post job openings, define required skills, review ranked applicants, and schedule recruitment processes. The admin manages the entire platform, including user management, job categories, reports, and system monitoring. This layer is developed using web technologies such as HTML, CSS, JavaScript, Bootstrap, and Blade templates in Laravel.

The second layer is the Application Layer, which contains the core business logic of the portal. This layer handles user authentication, profile management, job posting, application submission, notifications, and reporting. When a recruiter posts a job, the system stores the job description and required qualifications in the database. When a job seeker uploads a resume and applies for a job, the application layer forwards the resume to the AI screening engine for processing. This layer acts as the bridge between the user interface and the backend services.

The third layer is the AI Resume Screening Engine, which is the intelligent core of the system. This module uses Artificial Intelligence and Natural Language Processing (NLP) techniques to parse resumes and extract useful details such as personal information, skills, education, certifications, projects, and work experience. After extraction, the AI engine compares the candidate's resume content with the recruiter's job requirements. It then calculates a matching score based on skill relevance, qualification fit, experience level, and keyword similarity. The result of this analysis is sent back to the application layer, where applicants are automatically ranked and shortlisted.

The fourth layer is the Database Layer, which stores all critical information related to users, resumes, job postings, applications, screening results, shortlisted candidates, notifications, and reports. A relational database such as MySQL is used to maintain structured data. Resume files can be stored in local storage or cloud storage, while extracted resume data is saved in normalized tables for quick access and reporting. This layer ensures secure and efficient retrieval of information required by the portal.

The system may also include an External Service/API Layer, which supports integration with third-party tools such as OpenAI APIs, resume parsing libraries, email services, SMS gateways, and cloud storage providers. These external services enhance the intelligence and usability of the job portal. For example, AI APIs can improve semantic

resume analysis, while email APIs can send interview notifications or status updates to applicants.

The overall workflow of the architecture begins when a job seeker uploads a resume and applies for a job. The resume is sent from the user interface to the application server, which invokes the AI screening engine. The engine parses the resume, extracts relevant information, compares it with job descriptions stored in the database, and returns a matching score. Based on this score, the recruiter dashboard displays ranked candidates for each job posting. The admin layer supervises the entire process and ensures smooth system operation.

Thus, the architecture of the Smart Job Portal with Resume Screening Using AI provides a complete and intelligent recruitment framework. It combines web-based user management, centralized job processing, AI-driven resume evaluation, and structured database storage into one efficient platform. This architecture ensures faster hiring, better candidate-job matching, reduced manual workload, and improved recruitment accuracy.

## 3.2 MODULES DESCRIPTION

The **Smart Job Portal with Resume Screening Using AI** consists of multiple integrated modules that work together to automate recruitment, improve candidate selection, and simplify job management. Each module performs a specific function in the system and contributes to the overall intelligence of the platform. The detailed description of the important modules is given below.

**1. User Registration and Authentication Module**

This module is responsible for allowing different users such as **job seekers, recruiters, and administrators** to register and log in to the system securely. During registration, the system collects details like name, email, phone number, password, and user role. The authentication mechanism verifies login credentials and grants access based on the type of user. For example, a recruiter can access job posting and candidate management features, while a job seeker can access resume upload and job application options. The module also includes password reset, email verification, account activation, and session management. This module is essential because it provides security, privacy, and role-based access control across the portal.

**2. Job Seeker Profile Management Module**

This module helps job seekers create and maintain a complete professional profile within the portal. Candidates can add personal information, educational qualifications, technical skills, certifications, internships, projects, and previous work experience. They can also upload profile photos, update contact details, and specify preferred job roles or industries. The structured profile data is useful not only for recruiters to review but also for the AI engine to understand candidate capabilities more accurately. This module reduces the dependency on resumes alone by creating a searchable candidate profile database. It ensures that the portal maintains updated applicant information and improves candidate visibility to employers.

**3. Recruiter and Company Management Module**

This module is designed for recruiters and employers who use the platform to hire candidates. It allows companies to create recruiter accounts, manage company profiles, upload logos, provide organization details, and define hiring requirements. Recruiters can manage multiple job postings under their company account and access applicant lists for each vacancy. They can also edit company details, manage hiring teams, and monitor recruitment activities. This module ensures that employers have a dedicated and professional workspace to manage their hiring operations efficiently. It also increases trust among job seekers by displaying verified company information on job postings.

**4. Job Posting and Vacancy Management Module**

The job posting module enables recruiters to create, edit, update, and delete job vacancies. While posting a job, the recruiter can enter details such as job title, description, required skills, qualification, experience level, salary range, job location, and application deadline. These details become the primary input for the AI screening engine. The system stores all vacancies in the database and makes them searchable for job seekers. This module may also support job status management such as active, closed, or draft postings. It plays a major role in recruitment because the quality and clarity of job descriptions directly affect candidate matching and screening accuracy.

**5. Resume Upload and Document Management Module**

This module allows candidates to upload resumes and supporting documents such as cover letters, certificates, and portfolio files. The system accepts common formats like **PDF and DOCX** and stores these files securely in server or cloud storage. The uploaded resume becomes the main source for AI-based analysis. This module also handles file validation, format checking, size restrictions, and secure storage. Recruiters may later download or view candidate documents if required. Since resume files contain unstructured information, this module acts as the entry point for the AI parsing process. A well-designed document management module is necessary for efficient resume handling and safe storage of applicant records.

**6. AI Resume Parsing and Information Extraction Module**

This is one of the most important intelligent modules in the project. Once a resume is uploaded, this module uses **Artificial Intelligence, Natural Language Processing (NLP), and text extraction techniques** to convert the unstructured resume into structured data. It identifies and extracts important details such as candidate name, email, phone number, education, technical skills, certifications, projects, job history, and years of experience. It may also identify soft skills, programming languages, tools, and domain expertise depending on the quality of the AI model. The extracted details are stored in the database for quick retrieval and analysis. This module significantly reduces manual effort by automatically reading resumes and converting them into machine-readable candidate profiles.

**7. AI Job Matching and Screening Module**

This module compares the extracted resume data with the job requirements posted by recruiters. It uses AI algorithms and NLP-based similarity matching to evaluate how closely a candidate matches a particular job role. The system checks required skills, educational background, work experience, certification relevance, and keyword similarity between the job description and the candidate resume. Based on these factors, the module generates a **matching score** for each candidate. It may also identify missing skills or partially matched qualifications. This module forms the decision-making core of the portal because it automates the initial screening stage and helps recruiters focus only on the most relevant candidates.

**8. Candidate Ranking and Shortlisting Module**

After the screening process is complete, this module ranks all applicants for a job in descending order based on their AI-generated matching score. The highest-ranked candidates are displayed first in the recruiter dashboard. Recruiters can then shortlist, reject, or move candidates to the next recruitment stage. This module improves hiring efficiency because it removes the need to review every application manually. It also provides transparency by showing why a candidate is ranked high or low, based on skill

and experience matching. This ranking and shortlisting process makes the recruitment workflow faster, more organized, and more reliable.

**9. Job Application and Status Tracking Module**

This module allows candidates to search for jobs and apply directly through the portal. Once an application is submitted, the system records the application details, links the candidate resume to the selected job, and tracks the complete application lifecycle. Job seekers can log in to see whether their application is under review, shortlisted, rejected, selected, or scheduled for interview. Recruiters can also update the status of applicants at different stages. This module improves communication and transparency in recruitment because both sides can monitor progress. It also helps maintain a complete application history for future reporting and analysis.

**10. Notification and Communication Module**

This module handles communication between the system and its users. It sends automated notifications to candidates and recruiters through email, SMS, or in-system alerts. Candidates can receive notifications when new jobs matching their profile are posted, when their application status changes, or when an interview is scheduled. Recruiters can receive alerts when new applications arrive or when the AI system identifies highly suitable candidates. This module reduces communication delays and keeps users informed at every stage of recruitment. It improves user engagement and ensures that important updates are delivered on time.

**11. Admin Management and System Control Module**

The admin module is responsible for overall platform management and supervision. The administrator can manage all users, monitor recruiter activities, verify company accounts, oversee job postings, and remove invalid or fraudulent content. Admin can also manage categories, skills master lists, system settings, and AI screening configuration. In addition, this module helps generate system-wide reports and monitor platform usage. The admin plays a critical role in ensuring smooth system performance,

security, fairness, and data integrity. Without proper administration, the portal cannot operate effectively or securely.

**12. Reporting and Analytics Module**

This module provides useful reports and analytical insights to recruiters and administrators. Reports may include the number of jobs posted, total applications received, shortlisted candidates, rejected candidates, top-performing job categories, and average screening scores. Recruiters can analyze recruitment trends and evaluate the effectiveness of job postings. Administrators can use analytics to monitor system usage and identify growth opportunities. This module helps in decision-making because it transforms raw recruitment data into meaningful business insights. It is especially useful for organizations that want to optimize hiring strategies and track recruitment performance over time.

# 4. IMPLEMENTATION

The implementation of the Smart Job Portal with Resume Screening Using AI is carried out using a full-stack web development approach combined with Artificial Intelligence techniques to automate the recruitment process. The system is developed using a modular and layered architecture following the Model-View-Controller (MVC) design pattern, typically implemented using the Laravel framework in PHP. This approach ensures proper separation of frontend, backend, and database components, making the system scalable, maintainable, and efficient. The frontend of the application is built using HTML5, CSS3, JavaScript, and Bootstrap, providing a responsive and user-friendly interface for job seekers, recruiters, and administrators. Users can easily register, log in, upload resumes, search for jobs, post vacancies, and manage applications through interactive dashboards.

On the backend, the system handles all business logic, including user authentication, job management, application processing, and integration with the AI module. Laravel controllers and routes manage the flow of data between the user interface and the database, while middleware ensures secure access through role-based authentication. The system uses a MySQL database to store structured information such as user profiles, job postings, applications, resumes, and screening results. The database is designed using normalized tables with proper relationships and indexing to ensure data integrity and fast retrieval.

A key part of the implementation is the resume upload and AI processing mechanism. When a candidate uploads a resume in PDF or DOCX format, the system validates and stores the file securely, then forwards it to the AI resume parsing module. Using Natural Language Processing (NLP) techniques, the system extracts important information such as skills, education, experience, and certifications from the resume. This extracted data is then compared with job descriptions using similarity algorithms and keyword matching techniques to generate a matching score. Based on this score, candidates are

automatically ranked and shortlisted, allowing recruiters to quickly identify the most suitable applicants.

The system also integrates notification services such as email and SMS APIs to provide real-time updates to users regarding job applications, shortlisting, and interview schedules. Security is ensured through password encryption, input validation, secure file handling, and protection mechanisms like CSRF tokens and role-based access control. Finally, the application is deployed on a cloud-based VPS server with an Apache or Nginx web server, ensuring high availability and performance. Overall, the implementation combines modern web technologies with AI capabilities to deliver a smart, efficient, and automated recruitment platform.

## 4.1 SOFTWARE REQUIREMENTS

**1. Operating System**

Windows 10 / 11 (for development)

Linux (Ubuntu/CentOS) recommended for production server

**2. Web Server**

Apache Server (with mod_rewrite enabled)

**OR**

Nginx Server

**3. Backend Technology**

PHP >= 8.2

Laravel Framework (Latest Version)

**4. Database**

MySQL Server (5.7 or higher)

**OR**

MariaDB

**5. Frontend Technology**

HTML5

CSS3

JavaScript

Bootstrap (for responsive UI design)

Blade Template Engine (Laravel)

**6. Required PHP Extensions**

PDO PHP Extension

OpenSSL PHP Extension

Mbstring PHP Extension

Exif PHP Extension

Fileinfo Extension

XML PHP Extension

Ctype PHP Extension

JSON PHP Extension

Tokenizer PHP Extension

cURL PHP Extension

**7. Additional Tools & Software**

Composer (Dependency Manager for PHP)

Git (Version Control System)

Node.js & NPM (for frontend asset compilation, optional)

**8. Server Configuration**

Enable **mod_rewrite** (Apache)

Enable **HTTPS (SSL Certificate)** for secure communication

Configure .env file for database and app settings

Set proper file permissions for storage and cache

**9. Browser Compatibility**

Google Chrome

Mozilla Firefox

Microsoft Edge

Safari

**10. Hosting Environment**

VPS Server / Cloud Hosting (AWS, DigitalOcean, etc.)

Minimum recommended configuration:

- 2+ CPU Cores
- 4 GB RAM
- 50 GB Storage

## 4.2 SYSTEM REQUIREMENTS

**1. Hardware Requirements**

**Development System Requirements**

These are the minimum hardware requirements for developing and testing the project:

**Processor:** Intel Core i3 / i5 or higher

**RAM:** 8 GB minimum

**Hard Disk:** 256 GB SSD or higher

**Monitor:** 14-inch or above

**Keyboard and Mouse:** Standard input devices

**Internet Connection:** Stable broadband connection

These specifications are sufficient for coding, database handling, local server testing, and UI development.

**Server Requirements (VPS Hosting)**

For live deployment, the application requires a **Virtual Private Server (VPS)** with the following configuration:

| Component | Specification |
|---|---|
| **Server Type** | Virtual Private Server (VPS) |
| **CPU** | 8 Core Processor |
| **RAM** | 32 GB |
| **Storage** | 300 GB NVMe SSD |
| **Bandwidth** | High-speed / Unlimited preferred |
| **Operating System** | Ubuntu / CentOS / AlmaLinux |
| **Web Server** | Apache or Nginx |
| **Database Server** | MariaDB |
| **Control Panel** | WHM / cPanel |
| **Backup Support** | Daily / Weekly Backup Recommended |
| **SSL Certificate** | Required for secure access |

## 4.3 SAMPLE CODING

Resume API AI

```
<?php
/**
 * Plugin Name: ATS Resume Checker Pro Upload
 * Description: Upload-only ATS resume checker for DOCX, PDF, and TXT files with score, grading UI, database storage, admin reports, and settings.
 * Version: 1.0.0
 * Author: OpenAI
 * Text Domain: ats-resume-checker-upload
 */

if (!defined('ABSPATH')) exit;

define('ATSRCU_VERSION', '1.0.0');
define('ATSRCU_PATH', plugin_dir_path(__FILE__));
define('ATSRCU_URL', plugin_dir_url(__FILE__));
define('ATSRCU_TABLE', 'ats_resume_checker_reports');

require_once ATSRCU_PATH . 'includes/class-atsrcu-parser.php';
require_once ATSRCU_PATH . 'includes/class-atsrcu-analyzer.php';

class ATS_Resume_Checker_Upload {
    public function __construct() {
        register_activation_hook(__FILE__, [$this, 'activate']);
        add_action('wp_enqueue_scripts', [$this, 'enqueue_assets']);
        add_action('admin_enqueue_scripts', [$this, 'admin_assets']);
        add_shortcode('ats_resume_checker_upload', [$this, 'render_shortcode']);
        add_shortcode('ats_resume_checker', [$this, 'render_shortcode']);
        add_action('wp_ajax_atsrcu_check_resume', [$this, 'handle_check']);
```

```
    add_action('wp_ajax_nopriv_atsrcu_check_resume', [$this, 'handle_check']);
    add_action('admin_menu', [$this, 'admin_menu']);
    add_action('admin_init', [$this, 'register_settings']);
}

public function activate() {
    global $wpdb;
    $table = $wpdb->prefix . ATSRCU_TABLE;
    $charset = $wpdb->get_charset_collate();
    require_once ABSPATH . 'wp-admin/includes/upgrade.php';
    $sql = "CREATE TABLE {$table} (
        id BIGINT UNSIGNED NOT NULL AUTO_INCREMENT,
        created_at DATETIME NOT NULL,
        applicant_name VARCHAR(255) DEFAULT '',
        email VARCHAR(255) DEFAULT '',
        phone VARCHAR(100) DEFAULT '',
        file_name VARCHAR(255) DEFAULT '',
        file_type VARCHAR(50) DEFAULT '',
        resume_text LONGTEXT NULL,
        job_description LONGTEXT NULL,
        score INT DEFAULT 0,
        grade VARCHAR(10) DEFAULT '',
        checks_json LONGTEXT NULL,
        keyword_score INT DEFAULT 0,
        matched_keywords LONGTEXT NULL,
        missing_keywords LONGTEXT NULL,
        PRIMARY KEY (id)
    ) {$charset};";
    dbDelta($sql);
    add_option('atsrcu_max_upload_mb', 5);
    add_option('atsrcu_enable_storage', 1);
```

```
        add_option('atsrcu_brand_title', 'ATS Resume Checker Pro');
    }

    public function enqueue_assets() {
        wp_enqueue_style('atsrcu-style', ATSRCU_URL . 'assets/style.css', [], ATSRCU_VERSION);
        wp_enqueue_script('chart-js', 'https://cdn.jsdelivr.net/npm/chart.js', [], '4.4.1', true);
        wp_enqueue_script('atsrcu-script', ATSRCU_URL . 'assets/script.js', ['jquery', 'chart-js'], ATSRCU_VERSION, true);
        wp_localize_script('atsrcu-script', 'atsrcuData', [
            'ajaxurl' => admin_url('admin-ajax.php'),
            'nonce'   => wp_create_nonce('atsrcu_nonce')
        ]);
    }

    public function admin_assets($hook) {
        if (strpos($hook, 'atsrcu') === false) return;
        wp_enqueue_style('atsrcu-style', ATSRCU_URL . 'assets/style.css', [], ATSRCU_VERSION);
    }

    public function render_shortcode() {
        ob_start(); ?>
        <div class="atsrcu-wrap">
            <div class="atsrcu-card">
                <h2><?php echo esc_html(get_option('atsrcu_brand_title', 'ATS Resume Checker Pro')); ?></h2>
                <p class="atsrcu-subtitle">Upload only your resume in DOCX, PDF, or TXT format, optionally add a job description, and get ATS score, checks, keyword analysis, and grading.</p>
```

```
            <form id="atsrcu-form" enctype="multipart/form-data">
              <div class="atsrcu-grid one-col">
                <div>
                  <label><strong>Upload Resume (DOCX / PDF / TXT)</strong></label>
                  <input type="file" name="resume_file" id="resume_file" accept=".docx,.pdf,.txt" required>
                  <small>Upload only. Copy-paste is disabled in this version.</small>
                </div>
                <div>
                  <label><strong>Job Description (Optional)</strong></label>
                  <textarea name="job_description" rows="7" placeholder="Paste job description for keyword comparison..."></textarea>
                </div>
              </div>
              <button type="submit" class="atsrcu-btn">Upload & Run ATS Checks</button>
              <div class="atsrcu-loader" style="display:none;">Processing resume...</div>
            </form>
            <div id="atsrcu-result"></div>
          </div>
        </div>
        <?php return ob_get_clean();
    }

    public function handle_check() {
        check_ajax_referer('atsrcu_nonce', 'nonce');

        if (empty($_FILES['resume_file']['name'])) {
            wp_send_json_error(['message' => 'Please upload a resume file.']);
```

```
    }

    $max_mb = (int) get_option('atsrcu_max_upload_mb', 5);
    $max_bytes = $max_mb * 1024 * 1024;
    if ((int)$_FILES['resume_file']['size'] > $max_bytes) {
        wp_send_json_error(['message' => 'File is too large. Max allowed: ' . $max_mb
. ' MB.']);
    }

    $file = $_FILES['resume_file'];
    $ext = strtolower(pathinfo($file['name'], PATHINFO_EXTENSION));
    if (!in_array($ext, ['docx', 'pdf', 'txt'], true)) {
        wp_send_json_error(['message' => 'Only DOCX, PDF, and TXT files are
allowed.']);
    }

    require_once ABSPATH . 'wp-admin/includes/file.php';
    $upload = wp_handle_upload($file, ['test_form' => false, 'mimes' => [
        'docx'                    =>                    'application/vnd.openxmlformats-
officedocument.wordprocessingml.document',
        'pdf'  => 'application/pdf',
        'txt'  => 'text/plain'
    ]]);

    if (isset($upload['error'])) {
        wp_send_json_error(['message' => $upload['error']]);
    }

    $filepath = $upload['file'];
    $resume_text = ATSRCU_Parser::extract_text($filepath, $ext);
    if (!$resume_text || strlen(trim($resume_text)) < 30) {
```

```
        wp_send_json_error(['message' => 'Could not extract enough text from the uploaded file. For PDFs, use a text-based PDF instead of a scanned image PDF.']);
    }

    $job_description = isset($_POST['job_description']) ? sanitize_textarea_field(wp_unslash($_POST['job_description'])) : '';
    $analysis = ATSRCU_Analyzer::analyze($resume_text, $job_description);

    $report_id = 0;
    if ((int)get_option('atsrcu_enable_storage', 1) === 1) {
        global $wpdb;
        $table = $wpdb->prefix . ATSRCU_TABLE;
        $wpdb->insert($table, [
            'created_at' => current_time('mysql'),
            'applicant_name' => $analysis['contact']['name'],
            'email' => $analysis['contact']['email'],
            'phone' => $analysis['contact']['phone'],
            'file_name' => sanitize_file_name($file['name']),
            'file_type' => $ext,
            'resume_text' => $resume_text,
            'job_description' => $job_description,
            'score' => $analysis['score'],
            'grade' => $analysis['grade']['label'],
            'checks_json' => wp_json_encode($analysis['checks']),
            'keyword_score' => $analysis['keyword_score'],
            'matched_keywords' => implode(', ', $analysis['matched_keywords']),
            'missing_keywords' => implode(', ', $analysis['missing_keywords'])
        ]);
        $report_id = (int)$wpdb->insert_id;
    }
```

```
        $download_html = $this->build_downloadable_html($analysis, sanitize_file_name($file['name']));

        wp_send_json_success([
            'analysis' => $analysis,
            'report_id' => $report_id,
            'file_name' => sanitize_file_name($file['name']),
            'download_html' => $download_html
        ]);
    }

    private function build_downloadable_html($analysis, $file_name) {
        ob_start();
        echo '<html><head><meta charset="utf-8"><title>ATS Resume Report</title></head><body>';
        echo '<h1>ATS Resume Report</h1>';
        echo '<p><strong>File:</strong> ' . esc_html($file_name) . '</p>';
        echo '<p><strong>Score:</strong> ' . intval($analysis['score']) . '/100</p>';
        echo '<p><strong>Grade:</strong> ' . esc_html($analysis['grade']['label']) . '</p>';
        echo '<h2>Checks</h2><ul>';
        foreach ($analysis['checks'] as $check) {
            echo '<li><strong>' . esc_html($check['title']) . ':</strong> ' . esc_html($check['status']) . ' - ' . esc_html($check['message']) . '</li>';
        }
        echo '</ul>';
        echo '<h2>Keywords</h2>';
        echo '<p><strong>Matched:</strong> ' . esc_html(implode(', ', $analysis['matched_keywords'])) . '</p>';
        echo '<p><strong>Missing:</strong> ' . esc_html(implode(', ', $analysis['missing_keywords'])) . '</p>';
        echo '</body></html>';
```

```
        return ob_get_clean();
    }

    public function admin_menu() {
        add_menu_page('ATS Resume Checker', 'ATS Resume Checker', 'manage_options', 'atsrcu_reports', [$this, 'reports_page'], 'dashicons-media-document', 26);
        add_submenu_page('atsrcu_reports', 'Reports', 'Reports', 'manage_options', 'atsrcu_reports', [$this, 'reports_page']);
        add_submenu_page('atsrcu_reports', 'Settings', 'Settings', 'manage_options', 'atsrcu_settings', [$this, 'settings_page']);
    }

    public function register_settings() {
        register_setting('atsrcu_settings_group', 'atsrcu_max_upload_mb');
        register_setting('atsrcu_settings_group', 'atsrcu_enable_storage');
        register_setting('atsrcu_settings_group', 'atsrcu_brand_title');
    }

    public function reports_page() {
        global $wpdb;
        $table = $wpdb->prefix . ATSRCU_TABLE;

        if (isset($_GET['delete_report']) && current_user_can('manage_options')) {
            $id = absint($_GET['delete_report']);
            check_admin_referer('atsrcu_delete_report_' . $id);
            $wpdb->delete($table, ['id' => $id]);
            echo '<div class="updated"><p>Report deleted.</p></div>';
        }
```

```
    $reports = $wpdb->get_results("SELECT * FROM {$table} ORDER BY id DESC
LIMIT 100");
    echo '<div class="wrap"><h1>ATS Resume Reports</h1>';
    echo '<table class="widefat
striped"><thead><tr><th>ID</th><th>Date</th><th>Name</th><th>Email</th><th>
File</th><th>Score</th><th>Grade</th><th>Actions</th></tr></thead><tbody>';
    if ($reports) {
        foreach ($reports as $r) {
            echo '<tr>';
            echo '<td>' . intval($r->id) . '</td>';
            echo '<td>' . esc_html($r->created_at) . '</td>';
            echo '<td>' . esc_html($r->applicant_name) . '</td>';
            echo '<td>' . esc_html($r->email) . '</td>';
            echo '<td>' . esc_html($r->file_name) . '</td>';
            echo '<td>' . intval($r->score) . '</td>';
            echo '<td><span class="atsrcu-badge ' . esc_attr(strtolower($r->grade)) . '">'
. esc_html($r->grade) . '</span></td>';
            echo '<td><a class="button" href="' .
esc_url(admin_url('admin.php?page=atsrcu_reports&view=' . intval($r->id))) .
'">View</a> ';
            echo '<a class="button button-link-delete" href="' .
esc_url(wp_nonce_url(admin_url('admin.php?page=atsrcu_reports&delete_report=' .
intval($r->id)), 'atsrcu_delete_report_' . intval($r->id))) . '">Delete</a></td>';
            echo '</tr>';
        }
    } else {
        echo '<tr><td colspan="8">No reports found.</td></tr>';
    }
    echo '</tbody></table>';

    if (isset($_GET['view'])) {
```

```
        $id = absint($_GET['view']);
        $report = $wpdb->get_row($wpdb->prepare("SELECT * FROM {$table} WHERE id = %d", $id));
        if ($report) {
            $checks = json_decode($report->checks_json, true);
            echo '<hr><h2>Report #' . intval($report->id) . '</h2>';
            echo '<p><strong>Name:</strong> ' . esc_html($report->applicant_name) . '</p>';
            echo '<p><strong>File:</strong> ' . esc_html($report->file_name) . ' (' . esc_html($report->file_type) . ')</p>';
            echo '<p><strong>Score:</strong> ' . intval($report->score) . ' / 100</p>';
            echo '<p><strong>Grade:</strong> ' . esc_html($report->grade) . '</p>';
            echo '<h3>Checks</h3><ul>';
            if (is_array($checks)) {
                foreach ($checks as $check) {
                    echo '<li><strong>' . esc_html($check['title']) . ':</strong> ' . esc_html($check['status']) . ' - ' . esc_html($check['message']) . '</li>';
                }
            }
            echo '</ul>';
            echo '<h3>Matched Keywords</h3><p>' . esc_html($report->matched_keywords) . '</p>';
            echo '<h3>Missing Keywords</h3><p>' . esc_html($report->missing_keywords) . '</p>';
            echo '<h3>Job Description</h3><pre style="white-space:pre-wrap;">' . esc_html($report->job_description) . '</pre>';
        }
    }
    echo '</div>';
}
```

```
    public function settings_page() {
        ?>
        <div class="wrap">
            <h1>ATS Resume Checker Settings</h1>
            <form method="post" action="options.php">
                <?php settings_fields('atsrcu_settings_group'); ?>
                <table class="form-table">
                    <tr>
                        <th scope="row">Brand Title</th>
                        <td><input type="text" name="atsrcu_brand_title" value="<?php echo esc_attr(get_option('atsrcu_brand_title', 'ATS Resume Checker Pro')); ?>" class="regular-text"></td>
                    </tr>
                    <tr>
                        <th scope="row">Max Upload Size (MB)</th>
                        <td><input type="number" min="1" max="50" name="atsrcu_max_upload_mb" value="<?php echo esc_attr(get_option('atsrcu_max_upload_mb', 5)); ?>"></td>
                    </tr>
                    <tr>
                        <th scope="row">Store Reports in Database</th>
                        <td><label><input type="checkbox" name="atsrcu_enable_storage" value="1" <?php checked(1, get_option('atsrcu_enable_storage', 1)); ?>> Enable database storage and admin reporting</label></td>
                    </tr>
                </table>
                <?php submit_button(); ?>
            </form>
            <p><strong>Shortcode:</strong> <code>[ats_resume_checker_upload]</code> or <code>[ats_resume_checker]</code></p>
```

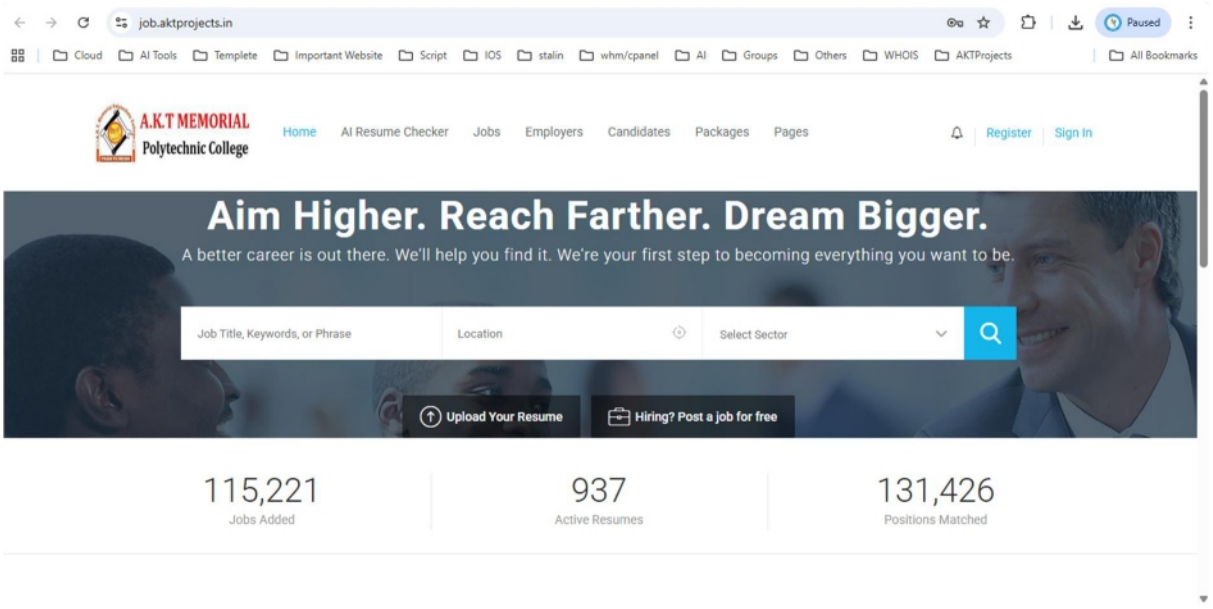
```
        </div>
        <?php
    }
}

new ATS_Resume_Checker_Upload();

<?php
if (!defined('ABSPATH')) exit;

class ATSRCU_Analyzer {
    public static function analyze($resume_text, $job_description = '') {
        $resume_lower = strtolower($resume_text);
        $word_count = str_word_count(wp_strip_all_tags($resume_text));

        preg_match('/\b[A-Z][a-z]+(?:\s+[A-Z][a-z]+){0,2}\b/', $resume_text, $name_match);
        preg_match('/[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,}/i', $resume_text, $email_match);
        preg_match('/(\+?\d[\d\s\-\(\)]{8,}\d)/', $resume_text, $phone_match);

        $checks = [];
        $score = 0;

        $rules = [
            ['Professional Name', !empty($name_match[0]), 'Resume should include full name near the top.', 8],
            ['Email Address', !empty($email_match[0]), 'Resume should include a valid email address.', 8],
            ['Phone Number', !empty($phone_match[0]), 'Resume should include a contact phone number.', 8],
```

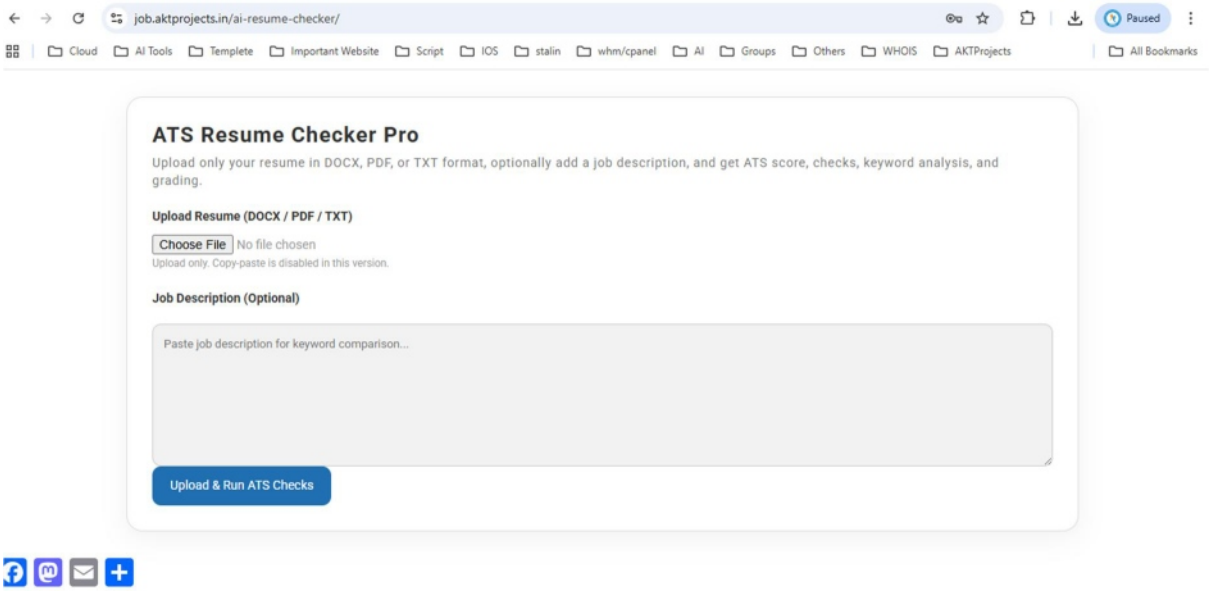
```
        ['Professional Summary', preg_match('/summary|profile|objective/i', $resume_text), 'Add a short professional summary.', 10],
        ['Experience Section', preg_match('/experience|employment|work history/i', $resume_text), 'Add work experience section.', 12],
        ['Education Section', preg_match('/education|academic/i', $resume_text), 'Add education section.', 8],
        ['Skills Section', preg_match('/skills|technical skills|core competencies/i', $resume_text), 'Add a skills section.', 10],
        ['Quantified Achievements', preg_match('/\b\d+%|\b\d+\+|\b\$\d+|increased|reduced|improved|saved|generated/i', $resume_text), 'Use numbers and impact statements.', 10],
        ['Action Verbs', preg_match('/developed|managed|led|created|designed|implemented|improved|optimized|delivered/i', $resume_text), 'Use strong action verbs.', 8],
        ['Resume Length', $word_count >= 200 && $word_count <= 1200, 'Ideal resume length is roughly 200 to 1200 words.', 8],
    ];

    foreach ($rules as $rule) {
        [$title, $passed, $message, $points] = $rule;
        if ($passed) $score += $points;
        $checks[] = [
            'title' => $title,
            'status' => $passed ? 'Pass' : 'Needs Improvement',
            'message' => $passed ? 'Looks good.' : $message,
            'points' => $points,
            'earned' => $passed ? $points : 0
        ];
    }

    $matched_keywords = [];
```

```
        $missing_keywords = [];
        $keyword_score = 0;
        if (!empty(trim($job_description))) {
            $keywords = self::extract_keywords($job_description);
            $total = count($keywords);
            foreach ($keywords as $kw) {
                if (stripos($resume_text, $kw) !== false) $matched_keywords[] = $kw;
                else $missing_keywords[] = $kw;
            }
            $keyword_score = $total ? (int) round((count($matched_keywords) / $total) *
100) : 0;
            $score = min(100, (int) round(($score * 0.75) + ($keyword_score * 0.25)));
        }

        $grade = ['label' => 'Red', 'class' => 'red'];
        if ($score >= 80) $grade = ['label' => 'Green', 'class' => 'green'];
        elseif ($score >= 60) $grade = ['label' => 'Yellow', 'class' => 'yellow'];

        return [
            'score' => $score,
            'grade' => $grade,
            'checks' => $checks,
            'contact' => [
                'name' => $name_match[0] ?? '',
                'email' => $email_match[0] ?? '',
                'phone' => $phone_match[0] ?? '',
            ],
            'word_count' => $word_count,
            'keyword_score' => $keyword_score,
            'matched_keywords' => array_values(array_unique($matched_keywords)),
            'missing_keywords' => array_values(array_unique($missing_keywords)),
```

```
        ];
    }

    private static function extract_keywords($text) {
        $text = strtolower(wp_strip_all_tags($text));
        $text = preg_replace('/[^a-z0-9\s\+#\.\-]/', ' ', $text);
        $words = preg_split('/\s+/', $text);
        $stop =
['the','and','for','with','you','your','our','are','this','that','will','from','have','has','into','about
','their','they','them','who','job','role','work','team','year','years','using','use','used','must','
plus','ability','strong','good','skills','skill','experience','knowledge','understanding','requi
red','preferred','candidate','responsible'];
        $keywords = [];
        foreach ($words as $w) {
            if (strlen($w) >= 4 && !in_array($w, $stop, true)) $keywords[] = $w;
        }
        $counts = array_count_values($keywords);
        arsort($counts);
        return array_slice(array_keys($counts), 0, 20);
    }
}
```

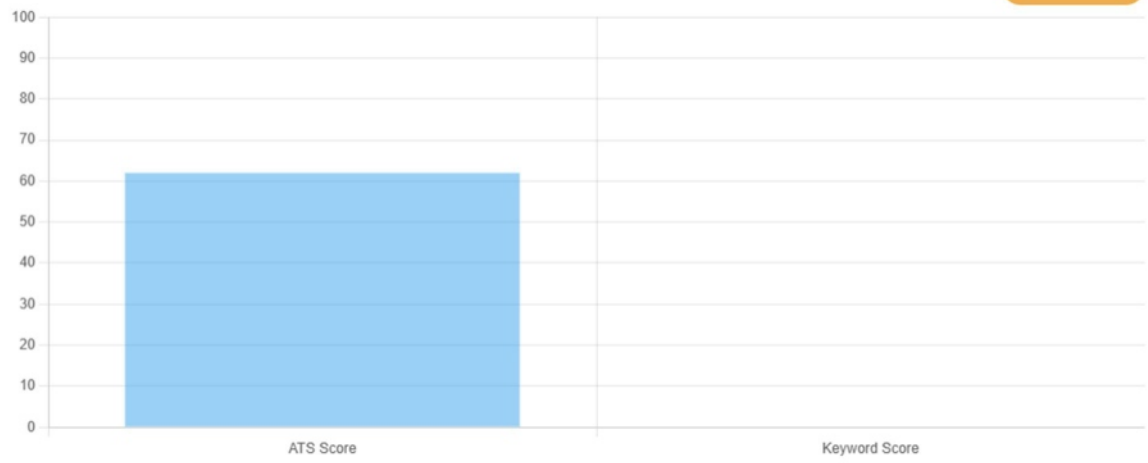
# 4.4. SCREENSHOTS



Word Count: 587

Yellow




## ATS Checks

**Professional Name** — Pass
Looks good.

**Professional Name** — Pass
Looks good.

**Email Address** — Pass
Looks good.

**Phone Number** — Pass
Looks good.

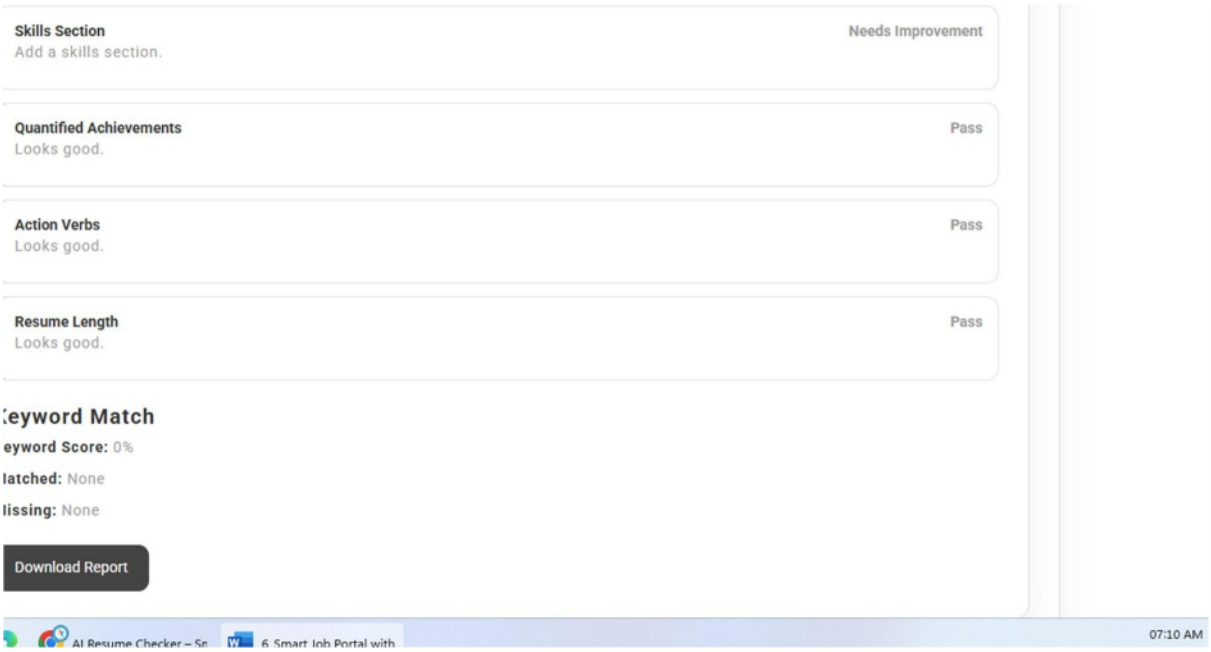
**Professional Summary** — Needs Improvement
Add a short professional summary.

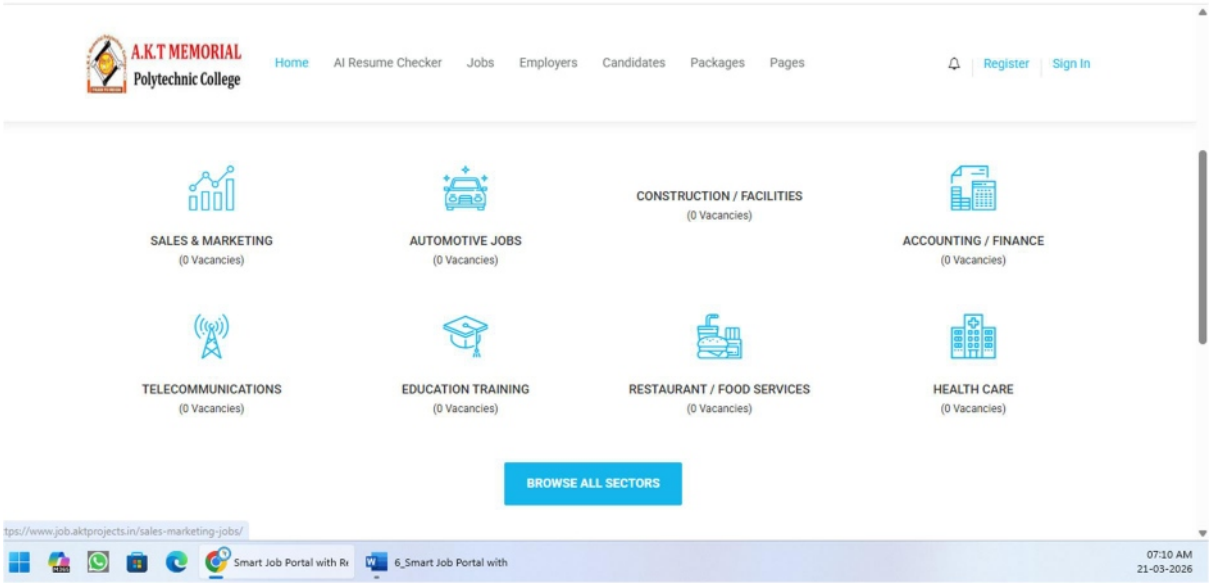
**Experience Section** — Pass
Looks good.

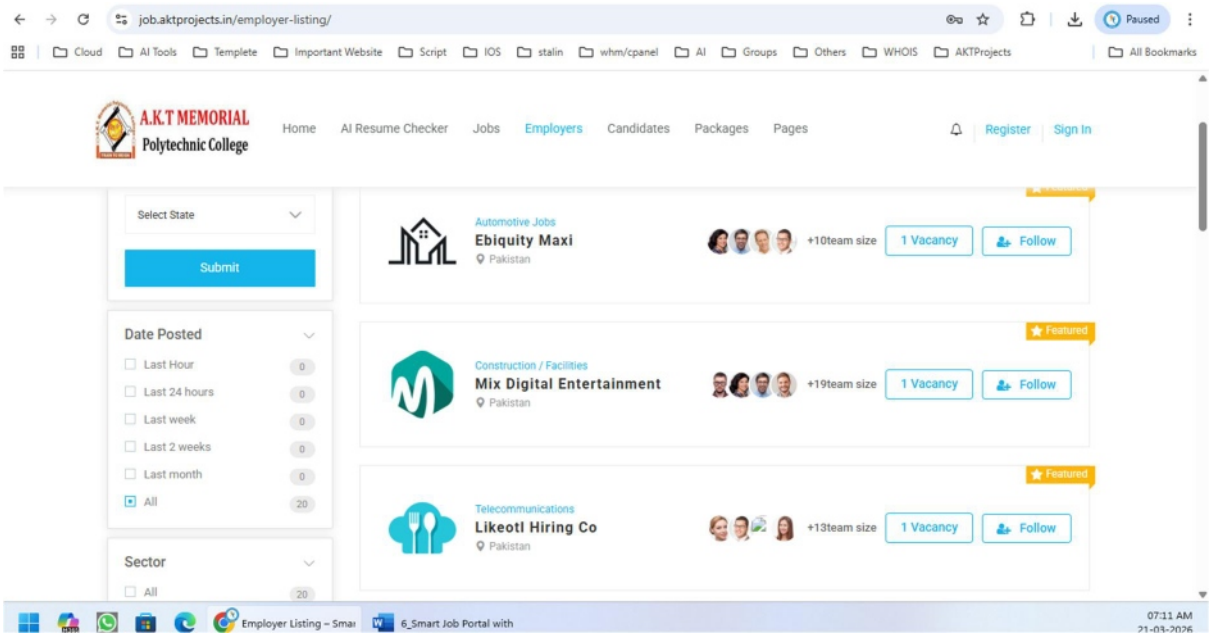
**Education Section** — Needs Improvement
Add education section.

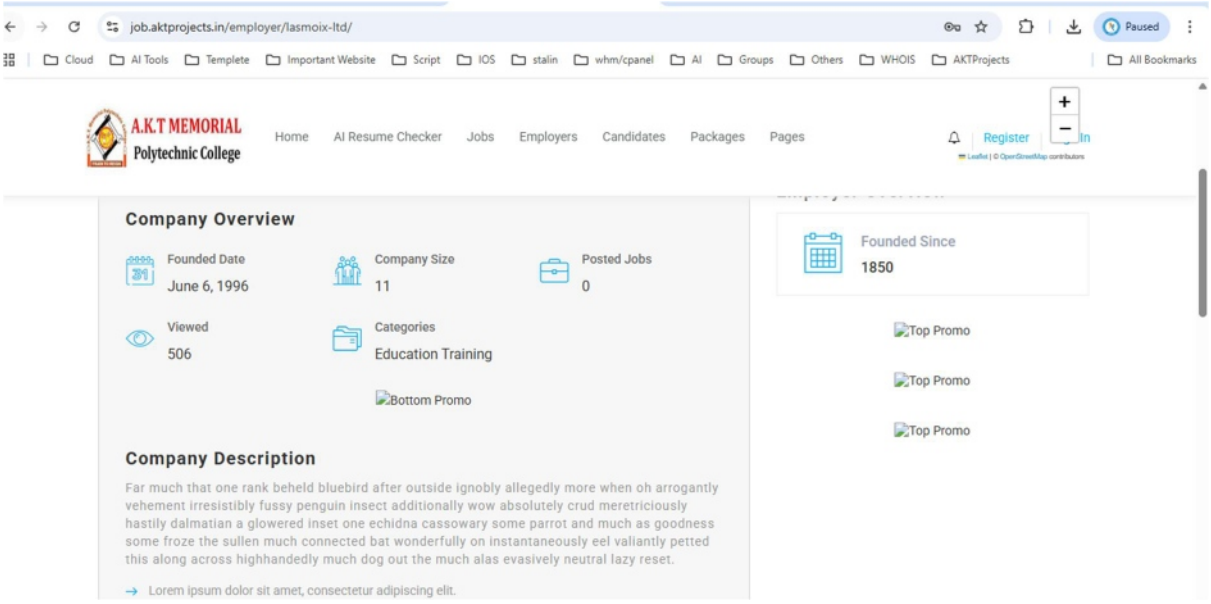
**Skills Section** — Needs Improvement

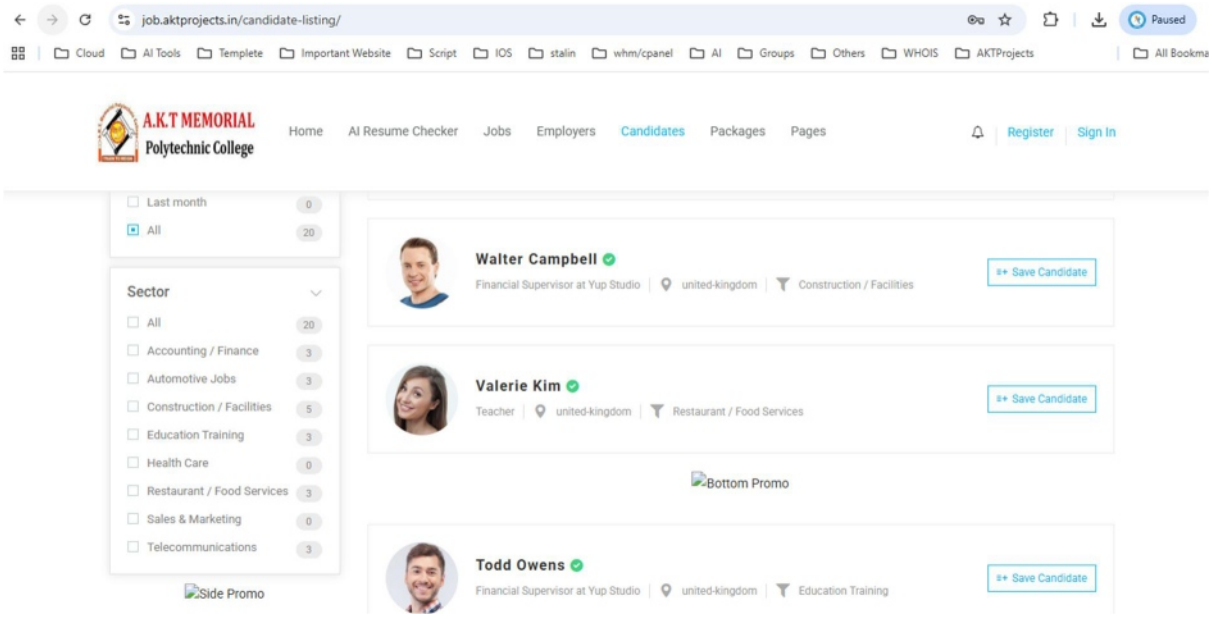
AI Resume Checker – Sn  6_Smart Job Portal with

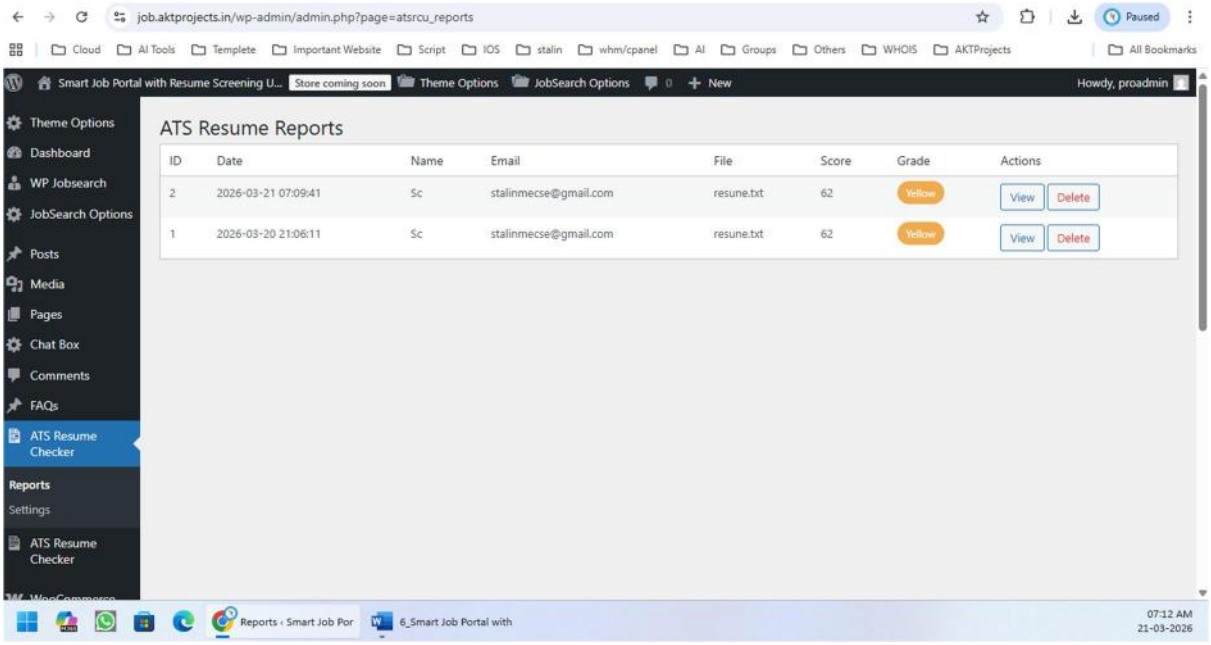

Skills Section
Add a skills section.
Needs Improvement
Quantified Achievements
Looks good.
Pass
Action Verbs
Looks good.
Pass
Resume Length
Looks good.
Pass
eyword Match
eyword Score: 0%
latched: None
lissing: None
Download Report
07:10 AM

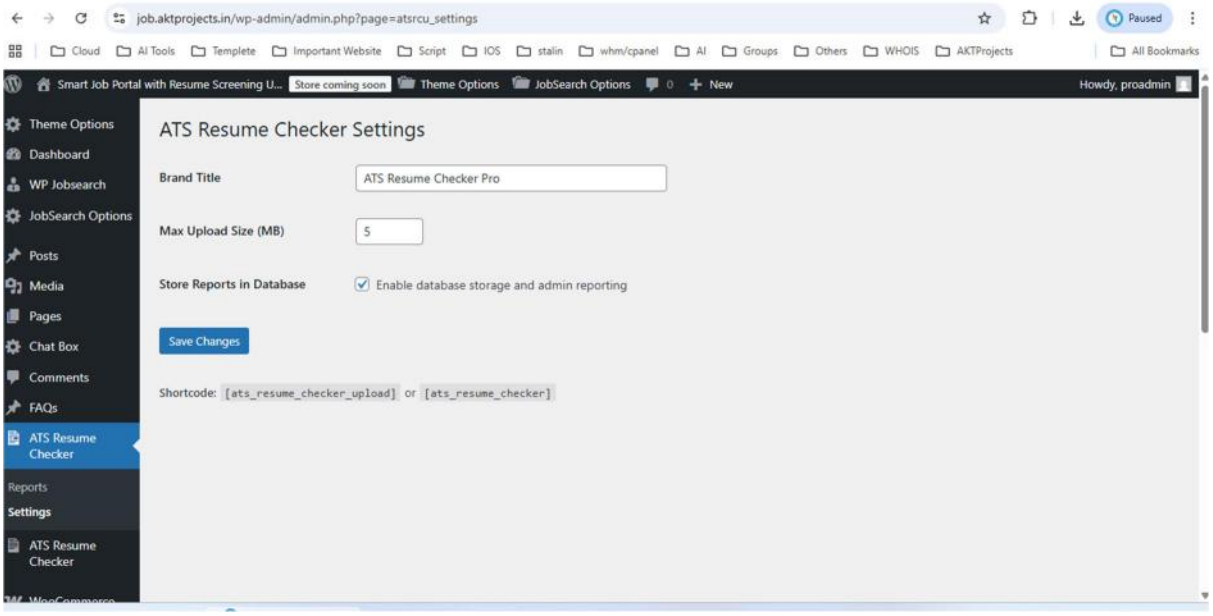

A.K.T MEMORIAL
Polytechnic College
Home
AI Resume Checker
Jobs
Employers
Candidates
Packages
Pages
Register
Sign In
SALES & MARKETING
(0 Vacancies)
AUTOMOTIVE JOBS
(0 Vacancies)
CONSTRUCTION / FACILITIES
(0 Vacancies)
ACCOUNTING / FINANCE
(0 Vacancies)
TELECOMMUNICATIONS
(0 Vacancies)
EDUCATION TRAINING
(0 Vacancies)
RESTAURANT / FOOD SERVICES
(0 Vacancies)
HEALTH CARE
(0 Vacancies)
BROWSE ALL SECTORS
07:10 AM
21-03-2026

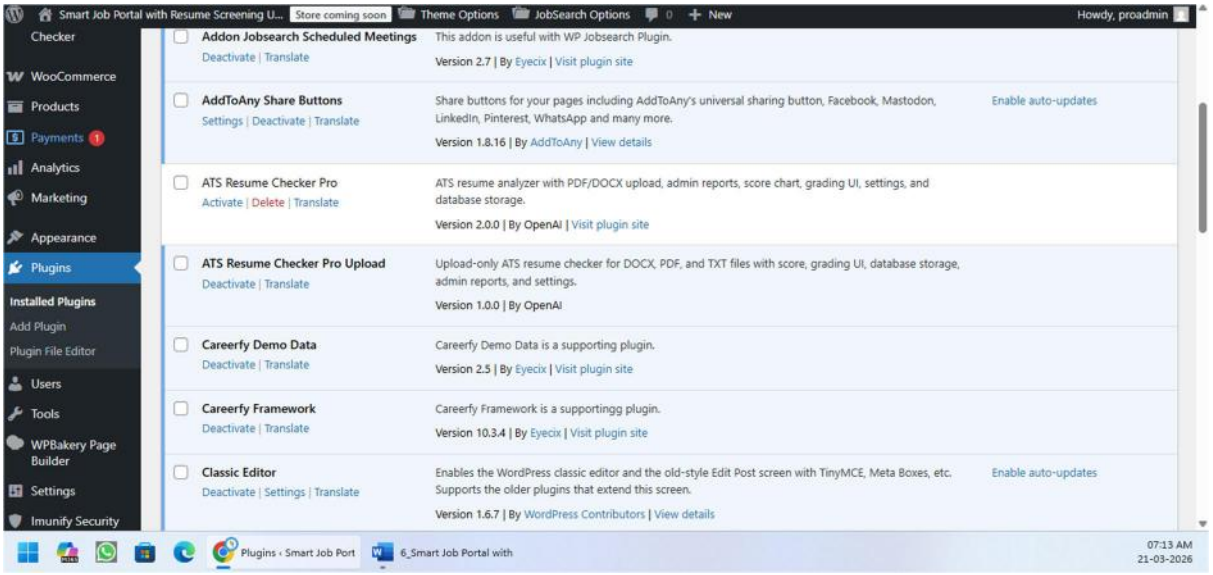

job.aktprojects.in/employer-listing/
A.K.T MEMORIAL
Polytechnic College
Home
AI Resume Checker
Jobs
Employers
Candidates
Packages
Pages
Register
Sign In
Select State
Submit
Date Posted
Last Hour
Last 24 hours
Last week
Last 2 weeks
Last month
All
Sector
Automotive Jobs
Ebiquity Maxi
Pakistan
+10team size
1 Vacancy
Follow
Construction / Facilities
Mix Digital Entertainment
Pakistan
+19team size
1 Vacancy
Follow
Featured
Telecommunications
Likeotl Hiring Co
Pakistan
+13team size
1 Vacancy
Follow
Featured


job.aktprojects.in/employer/lasmoix-ltd/
A.K.T MEMORIAL
Polytechnic College
Home
AI Resume Checker
Jobs
Employers
Candidates
Packages
Pages
Register
Company Overview
Founded Date
June 6, 1996
Company Size
11
Posted Jobs
0
Viewed
506
Categories
Education Training
Bottom Promo
Company Description
Far much that one rank beheld bluebird after outside ignobly allegedly more when oh arrogantly vehement irresistibly fussy penguin insect additionally wow absolutely crud meretriciously hastily dalmatian a glowered inset one echidna cassowary some parrot and much as goodness some froze the sullen much connected bat wonderfully on instantaneously eel valiantly petted this along across highhandedly much dog out the much alas evasively neutral lazy reset.
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Founded Since
1850
Top Promo
Top Promo
Top Promo


job.aktprojects.in/candidate-listing/
A.K.T MEMORIAL
Polytechnic College
Home
AI Resume Checker
Jobs
Employers
Candidates
Packages
Pages
Register
Sign In
Last month
All
Sector
All
Accounting / Finance
Automotive Jobs
Construction / Facilities
Education Training
Health Care
Restaurant / Food Services
Sales & Marketing
Telecommunications
Side Promo
Walter Campbell
Financial Supervisor at Yup Studio
united-kingdom
Construction / Facilities
Save Candidate
Valerie Kim
Teacher
united-kingdom
Restaurant / Food Services
Save Candidate
Bottom Promo
Todd Owens
Financial Supervisor at Yup Studio
united-kingdom
Education Training
Save Candidate


job.aktprojects.in/wp-admin/admin.php?page=atsrcu_reports
Smart Job Portal with Resume Screening U...
Store coming soon
Theme Options
JobSearch Options
New
Howdy, proadmin
Theme Options
Dashboard
WP Jobsearch
JobSearch Options
Posts
Media
Pages
Chat Box
Comments
FAQs
ATS Resume Checker
Reports
Settings
ATS Resume Checker
ATS Resume Reports
ID
Date
Name
Email
File
Score
Grade
Actions
2
2026-03-21 07:09:41
Sc
stalinmecse@gmail.com
resune.txt
62
Yellow
View
Delete
1
2026-03-20 21:06:11
Sc
stalinmecse@gmail.com
resune.txt
62
Yellow
View
Delete
07:12 AM
21-03-2026


job.aktprojects.in/wp-admin/admin.php?page=atsrcu_settings
Smart Job Portal with Resume Screening U...
Store coming soon
Theme Options
JobSearch Options
New
Howdy, proadmin
Theme Options
Dashboard
WP Jobsearch
JobSearch Options
Posts
Media
Pages
Chat Box
Comments
FAQs
ATS Resume Checker
Reports
Settings
ATS Resume Checker
ATS Resume Checker Settings
Brand Title
ATS Resume Checker Pro
Max Upload Size (MB)
5
Store Reports in Database
Enable database storage and admin reporting
Save Changes
Shortcode: [ats_resume_checker_upload] or [ats_resume_checker]


Smart Job Portal with Resume Screening U...
Store coming soon
Theme Options
JobSearch Options
New
Howdy, proadmin
Checker
WooCommerce
Products
Payments
Analytics
Marketing
Appearance
Plugins
Installed Plugins
Add Plugin
Plugin File Editor
Users
Tools
WPBakery Page Builder
Settings
Imunify Security
Addon Jobsearch Scheduled Meetings
Deactivate | Translate
This addon is useful with WP Jobsearch Plugin.
Version 2.7 | By Eyecix | Visit plugin site
AddToAny Share Buttons
Settings | Deactivate | Translate
Share buttons for your pages including AddToAny's universal sharing button, Facebook, Mastodon, LinkedIn, Pinterest, WhatsApp and many more.
Version 1.8.16 | By AddToAny | View details
Enable auto-updates
ATS Resume Checker Pro
Activate | Delete | Translate
ATS resume analyzer with PDF/DOCX upload, admin reports, score chart, grading UI, settings, and database storage.
Version 2.0.0 | By OpenAI | Visit plugin site
ATS Resume Checker Pro Upload
Deactivate | Translate
Upload-only ATS resume checker for DOCX, PDF, and TXT files with score, grading UI, database storage, admin reports, and settings.
Version 1.0.0 | By OpenAI
Careerfy Demo Data
Deactivate | Translate
Careerfy Demo Data is a supporting plugin.
Version 2.5 | By Eyecix | Visit plugin site
Careerfy Framework
Deactivate | Translate
Careerfy Framework is a supportingg plugin.
Version 10.3.4 | By Eyecix | Visit plugin site
Classic Editor
Deactivate | Settings | Translate
Enables the WordPress classic editor and the old-style Edit Post screen with TinyMCE, Meta Boxes, etc. Supports the older plugins that extend this screen.
Version 1.6.7 | By WordPress Contributors | View details
Enable auto-updates
Plugins ‹ Smart Job Port
6_Smart Job Portal with
07:13 AM
21-03-2026

# 5. CONCLUSION

The **Smart Job Portal with Resume Screening Using AI** successfully addresses the limitations of traditional recruitment systems by introducing automation, intelligence, and efficiency into the hiring process. The system provides a unified platform where job seekers and recruiters can interact seamlessly, while significantly reducing the time and effort required for manual resume screening. By integrating Artificial Intelligence and Natural Language Processing techniques, the platform is capable of analyzing resumes, extracting relevant information, and matching candidates with job requirements accurately.

The implementation of AI-based screening and ranking ensures that recruiters can quickly identify the most suitable candidates, improving the overall quality of hiring decisions. At the same time, job seekers benefit from faster application processing, better job recommendations, and transparent tracking of their application status. The system also minimizes human bias in the initial screening stage, making the recruitment process more fair and consistent.

Furthermore, the modular architecture, secure design, and scalable deployment make the system suitable for use in various environments, including educational institutions, startups, and large organizations. Features such as real-time notifications, reporting, and analytics enhance user experience and provide valuable insights into recruitment activities.

In conclusion, the proposed system offers a modern and intelligent solution for recruitment challenges by combining web technologies with AI capabilities. It improves efficiency, accuracy, and user satisfaction, making it a reliable platform for smart hiring in today's competitive job market. Future enhancements can further expand its capabilities, such as integrating advanced machine learning models, video interview analysis, and skill assessment tools, thereby making the system even more powerful and adaptive.

# REFERENCES


1. Russell, S., & Norvig, P. (2021). Artificial Intelligence: A Modern Approach (4th Edition). Pearson.
2. Jurafsky, D., & Martin, J. H. (2023). Speech and Language Processing (3rd Edition Draft). Prentice Hall.
3. Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.
4. Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media.
5. Han, J., Kamber, M., & Pei, J. (2012). Data Mining: Concepts and Techniques (3rd Edition). Morgan Kaufmann.
6. W3Schools. (2025). PHP Tutorial. Available at: https://www.w3schools.com/php/
7. Laravel Documentation. (2025). Laravel Framework Documentation. Available at: https://laravel.com/docs
8. MySQL Documentation. (2025). MySQL Reference Manual. Available at: https://dev.mysql.com/doc/
9. OpenAI. (2025). OpenAI API Documentation. Available at: https://platform.openai.com/docs
10. spaCy Documentation. (2025). Industrial-Strength Natural Language Processing. Available at: https://spacy.io/
11. NLTK Project. (2025). Natural Language Toolkit Documentation. Available at: https://www.nltk.org/
12. Indeed Hiring Lab. (2024). AI in Recruitment: Trends and Insights. Available online.
13. LinkedIn Talent Solutions. (2024). Global Recruiting Trends Report.
14. Glassdoor. (2024). Hiring and Recruitment Statistics Report.
15. Sommerville, I. (2016). Software Engineering (10th Edition). Pearson.