

**RESUME VERIFICATION SYSTEM**

**A MINI-PROJECT REPORT**

*Submitted by*

**LOGA PRABHA GB      240701290**

**SHREMATHI K      240701504**

*in partial fulfillment of the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**



**RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI**

**An Autonomous Institute**

**CHENNAI**

**NOVEMBER 2025**

## **BONAFIDE CERTIFICATE**

Certified that this project “**RESUME VERIFICATION SYSTEM**” is the bonafide work of “**LOGA PRABHA GB, SHREMATHI K**” who carried out the project work under my supervision.

### **SIGNATURE**

**Dr. V. JANANEE**

**ASSISTANT PROFESSOR SG**

Dept. of Computer Science and Engg,  
Rajalakshmi Engineering College  
Chennai

This mini project report is submitted for the viva voce examination to be held on

---

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## **ABSTRACT**

In today's recruitment landscape, verifying the authenticity and completeness of resumes has become a tedious and error-prone task. The Resume Verification System aims to automate resume analysis using a database-driven application that validates the structure, keywords, and contents of resumes.

This desktop application, built using Java Swing and MySQL, allows users to upload, paste, or analyze their resume contents. It performs smart text-based validation such as checking for essential fields (name, email, phone number, education, and skills), ensures proper word count and keyword presence, and generates a comprehensive score and verification summary. All results and resumes are stored in a MySQL database for easy retrieval and analytics.

This project demonstrates efficient frontend-backend integration, database connectivity, and content verification logic, making resume evaluation simpler, automated, and reliable for academic or business use.

## ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman **MR. S. MEGANATHAN** and the chairperson **DR. M.THANGAM MEGANATHAN** for their timely support and encouragement.

We are greatly indebted to our respected and honorable principal **Dr. S.N. MURUGESAN** for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by our Head Of The Department **Dr. E.M. MALATHY** and our Deputy Head Of The Department **Dr. J. MANORANJINI** for being ever supporting force during our project work

We also extend our sincere and hearty thanks to our internal guide **Dr. V. JANANEE** , for her valuable guidance and motivation during the completion of this project.

## TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
1.1	INTRODUCTION	8
1.2	SCOPE OF THE WORK	8
1.3	PROBLEM STATEMENT	8
1.4	AIM AND OBJECTIVES OF THE PROJECT	8
<b>2</b>	<b>SYSTEM SPECIFICATIONS</b>	<b>9</b>
2.1	HARDWARE SPECIFICATIONS	9
2.2	SOFTWARE SPECIFICATIONS	9
<b>3</b>	<b>MODULE DESCRIPTION</b>	<b>10</b>
<b>4</b>	<b>CODING</b>	<b>11</b>
<b>5</b>	<b>SCREENSHOTS</b>	<b>16</b>
<b>6</b>	<b>CONCLUSION AND FUTURE ENHANCEMENT</b>	<b>18</b>
<b>7</b>	<b>REFERENCES</b>	<b>19</b>

## LIST OF FIGURES

<b>FIGURE NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
<b>5.1</b>	<b>LOGIN SCREEN</b>	<b>13</b>
<b>5.2</b>	<b>DASHBOARD - MAIN INTERFACE</b>	<b>13</b>
<b>5.3</b>	<b>FILE UPLOAD DIALOG</b>	<b>14</b>
<b>5.4</b>	<b>RESUME CONTENT EDITOR</b>	<b>14</b>
<b>5.5</b>	<b>VERIFICATION IN PROGRESS</b>	<b>15</b>
<b>5.6</b>	<b>VERIFICATION RESULTS POPUP</b>	<b>15</b>
<b>5.7</b>	<b>ANALYTICS SUMMARY (SCORE, WORD COUNT, KEYWORDS)</b>	<b>16</b>
<b>5.8</b>	<b>RESUME HISTORY TABLE</b>	<b>16</b>
<b>5.9</b>	<b>DETAILED RESUME VIEW</b>	<b>17</b>

<b>5.10</b>	<b>DATABASE - USERS TABLE</b>	<b>17</b>
<b>5.11</b>	<b>DATABASE - RESUMES TABLE</b>	<b>18</b>
<b>5.12</b>	<b>DATABASE - VERIFICATION RESULTS TABLE</b>	<b>18</b>

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 INTRODUCTION**

The Resume Verification System is a database-driven desktop application designed to validate and verify resumes efficiently. Traditional manual resume screening often leads to inconsistencies and errors. This system automates verification by scanning resume text for crucial details such as contact info, education, skills, and experience, ensuring data completeness and authenticity.

### **SCOPE OF THE WORK**

This application is designed for HR professionals, educational institutions, and online job platforms to evaluate resumes. It can later be extended to integrate AI-based scoring models, PDF parsing, and email verification tools.

### **PROBLEM STATEMENT**

With the digital evolution and high job application volumes, fake or incomplete resumes are common. Manual verification consumes excessive time and effort. A semi-automated, scalable tool is needed to verify resume content based on structure and required key points.

### **1.4 AIM AND OBJECTIVES OF THE PROJECT**

The main aims of this project are:

- To automate resume verification using a rule-driven algorithm.
- To store and manage user data and resume analysis results in a MySQL database.



- To develop a friendly Java Swing GUI for users to upload and analyze resumes.
- To enhance evaluation accuracy through keyword and field validation.

## CHAPTER 2

### SYSTEM SPECIFICATIONS

#### 2.1 HARDWARE SPECIFICATIONS

Component	Specification
Processor	Intel i5 or higher
RAM	8 GB Minimum
Storage	256 GB SSD / 1 TB HDD
Display	1366x768 or higher

#### 2.2 SOFTWARE SPECIFICATIONS

COMPONENT	SPECIFICATION

OPERATING SYSTEM	WINDOWS 10 / LINUX / MacOS
FRONT END	JAVA SWING
BACK END	MYSQL DATABASE
PROGRAMMING LANGUAGE	JAVA (JDK 17+)
TOOLS USED	INTELLIJ IDEA, MYSQL WORKBENCH
LIBRARIES	JDBC CONNECTOR, SWING COMPONENTS

## **CHAPTER 3**

### **MODULE DESCRIPTION**

This application consists of two modules. When the program runs, it will ask for a confirmation to the login window. The person who interacts can login as an Administrator or as a User. The description of the modules are as follows:

#### **1. LOGIN MODULE**

Provides authentication for admin and user accounts using database-stored credentials. Ensures secure login with email and password.

#### **2. DASHBOARD MODULE**

After login, users can upload or paste resumes for analysis. It includes buttons to load files, analyze content, and view verification summaries.

#### **3. VERIFICATION MODULE**

Analyzes resumes for required fields:

Personal Info (name, email, phone)

Education details

Skill keywords

Experience details

Calculates a final score out of 100 based on content quality and completeness.

#### **4 . DATABASE MODULE**

Handles storing:

- User credentials
- Uploaded resumes and details
- Analysis results

Uses relational design:

- `users(user_id, email, password, role)`
- `resumes(resume_id, user_id, file_name, resume_text, word_count)`
- `verification_results(result_id, resume_id, overall_score, keyword_hits, education_keywords, skill_keywords, status)`

## 5. HISTORY AND REPORT MODULE

Displays all past verifications, allows searching/viewing previous entries, and provides a detailed summary of each resume.

## SAMPLE CODING

### 4.1 Database Connection Code

// DatabaseConnection.java - Establishing MySQL Connection

```
import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DatabaseConnection {

    private static final String URL = "jdbc:mysql://localhost:3306/resume_verifier";

    private static final String USER = "root";

    private static final String PASSWORD = "";

    private static Connection connection = null;

    public static Connection getConnection() {

        try {

            if (connection == null || connection.isClosed()) {

                Class.forName("com.mysql.cj.jdbc.Driver");

                connection = DriverManager.getConnection(URL, USER, PASSWORD);

                System.out.println("✅ Database connected successfully!");

            }

        } catch (ClassNotFoundException e) {

            System.err.println("❌ MySQL JDBC Driver not found!");

            e.printStackTrace();

        } catch (SQLException e) {
```

```

        System.err.println(" ✖ Database connection failed!");

        e.printStackTrace();

    }

    return connection;

}

}

```

## 4.2 User Authentication Code

// UserDao.java - User Login Authentication

```

import java.sql.*;

public class UserDao {

    public static User authenticateUser(String email, String password) {

        Connection conn = DatabaseConnection.getConnection();

        String query = "SELECT * FROM users WHERE email = ? AND password = ?";

        try {

            PreparedStatement pst = conn.prepareStatement(query);

            pst.setString(1, email);

            pst.setString(2, password);

            ResultSet rs = pst.executeQuery();

            if (rs.next()) {

                User user = new User();

```

```

        user.setUserId(rs.getInt("user_id"));

        user.setUsername(rs.getString("username"));

        user.setEmail(rs.getString("email"));

        user.setRole(rs.getString("role"));

        return user;

    }

    } catch (SQLException e) {

        e.printStackTrace();

    }

    return null;

}

}

```

### 4.3 Resume Verification Algorithm

```

// Verification Logic - Email Detection using Regex

private boolean checkEmail(String text) {

    Pattern emailPattern = Pattern.compile(

        "[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}"

    );

    Matcher matcher = emailPattern.matcher(text);

    return matcher.find();

}

```

```
// Keyword Counting Algorithm

private int countKeywords(String text, String[] keywords) {

    text = text.toLowerCase();

    int count = 0;

    for (String keyword : keywords) {

        if (text.contains(keyword.toLowerCase())) {

            count++;

        }

    }

    return count;

}
```

#### 4.4 Saving Resume to Database

```
// ResumeDAO.java - Saving Resume Data

public static int saveResume(Resume resume) {

    Connection conn = DatabaseConnection.getConnection();

    String query = "INSERT INTO resumes (user_id, file_name, file_path, " +

        "resume_text, word_count) VALUES (?, ?, ?, ?, ?)";

    try {

        PreparedStatement pst = conn.prepareStatement(query,

            Statement.RETURN_GENERATED_KEYS);

        pst.setInt(1, resume.getUserId());

        pst.setString(2, resume.getFileName());

    }
```



```

        pst.setString(3, resume.getFilePath());

        pst.setString(4, resume.getResumeText());

        pst.setInt(5, resume.getWordCount());

        int rowsAffected = pst.executeUpdate();

        if (rowsAffected > 0) {

            ResultSet rs = pst.getGeneratedKeys();

            if (rs.next()) {

                return rs.getInt(1); // Return generated resume_id

            }

        }

    } catch (SQLException e) {

        e.printStackTrace();

    }

    return -1;

}

```

## 4.5 Score Calculation Logic

```

// Scoring Algorithm

private VerificationResult performVerification(String resumeText) {

    VerificationResult result = new VerificationResult();

    int score = 20; // Base score

    // Word count scoring (max 15 points)

    int wordCount = resumeText.split("\\s+").length;

```

```
if (wordCount > 500) score += 15;

else if (wordCount > 300) score += 10;

else if (wordCount > 150) score += 5;

// Email and Phone (10 points each)

if (checkEmail(resumeText)) score += 10;

if (checkPhone(resumeText)) score += 10;

// Education keywords (15 points)

String[] educationKeywords = {"B.Tech", "M.Tech", "University",

                               "Degree", "Engineering"};

int eduCount = countKeywords(resumeText, educationKeywords);

if (eduCount > 0) score += 15;

// Skills (20 points)

String[] skillKeywords = {"Java", "Python", "SQL", "MySQL"};

int skillCount = countKeywords(resumeText, skillKeywords);

if (skillCount >= 5) score += 20;

else if (skillCount >= 3) score += 10;

score = Math.min(score, 100); // Cap at 100

result.setOverallScore(score);

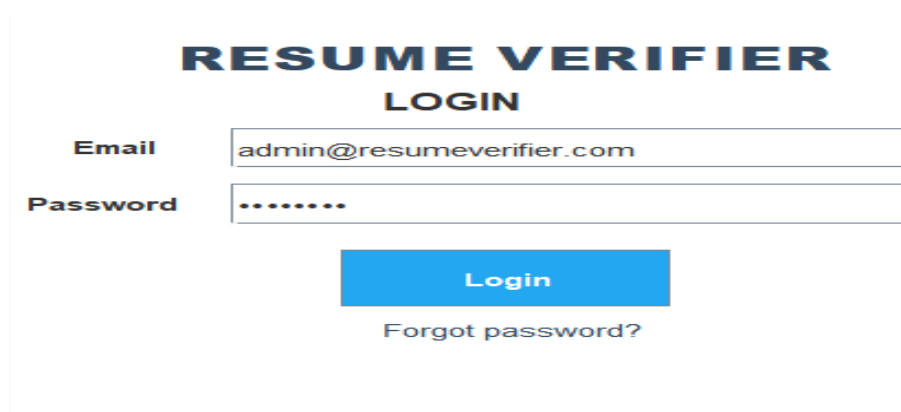
return result;

}
```

## CHAPTER 5

### SCREEN SHOTS

Fig 5.1 Introduction page



The screenshot displays the login interface for 'RESUME VERIFIER'. The title 'RESUME VERIFIER' is in a large, bold, blue font, with 'LOGIN' in a smaller, bold, black font directly below it. There are two input fields: the first is labeled 'Email' and contains the text 'admin@resumeverifier.com'; the second is labeled 'Password' and contains seven dots. Below these fields is a blue rectangular button with the word 'Login' in white. At the bottom, there is a link that says 'Forgot password?'.

<b>RESUME VERIFIER</b>	
<b>LOGIN</b>	
<b>Email</b>	<input type="text" value="admin@resumeverifier.com"/>
<b>Password</b>	<input type="password" value="....."/>
<input type="button" value="Login"/>	
<a href="#">Forgot password?</a>	

Resume Verifier

## RESUME ANALYTICS DASHBOARD

[View History](#) [Logout](#)

**File Path:** No file selected. Please use Browse or Paste content below. [Browse...](#)

**Resume Content Editor**

Paste your resume content here or use the 'Browse' button above...

[Analyze Resume](#)

Status: Ready to load or paste content.

**Verification Summary**

Overall Score	Total Words	Keywords Found
---------------	-------------	----------------

Detailed Verification Results

[Close](#)

**Verification Results:**

**Overall Score: 85/100**

**Status: Excellent**

**Word Count: 75**

**Education Keywords: 3**

**Skill Keywords: 18**

**Has Email: Yes**

**Has Phone: Yes**

**Has Education: Yes**

**Has Experience: Yes**

**Comments: Strong resume with good content coverage.**

[OK](#)

RESUME HISTORY					
<div>~ Back to Dashboard</div>					
All Uploaded Resumes					
ID	File Name	Word Count	Score	Status	Upload Date
6	pasted_content.txt	75	85/100	Excellent	2025-10-25 13:22:22.0
5	pasted_content.txt	75	85/100	Excellent	2025-10-25 12:18:46.0
4	pasted_content.txt	75	85/100	Excellent	2025-10-25 12:17:03.0
3	pasted_content.txt	315	95/100	Excellent	2025-10-25 12:05:50.0
2	resume_img_real.jpg	1517	40/100	Fair	2025-10-25 11:48:20.0
1	resume_img_real.jpg	1517	40/100	Fair	2025-10-25 11:48:05.0

## CHAPTER 6

### CONCLUSION AND FUTURE ENHANCEMENT

The Resume Verification System facilitates automated resume validation using keyword and field detection. It simplifies HR and academic screening processes and ensures consistent, data-driven evaluation.

Future Enhancements:

- Integration with PDF/Text extractors.
- AI/NLP-based scoring.
- Direct HR portal integration.
- Export reports to PDF.

## REFERENCES

1. <https://www.javatpoint.com/java-swing>
2. <https://www.mysqltutorial.org/>
3. <https://www.geeksforgeeks.org/jdbc-connectivity-in-java/>
4. <https://www.w3schools.com/sql/>
5. <https://docs.oracle.com/en/java/>