



Resume Upload and Search Web Application

Submitted by

Name: Mehak Kaur, 22csu117

Shefali Khera, 22csu426

Shreya, 22csu515

DS-2

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,
SCHOOL OF ENGINEERING AND TECHNOLOGY

THE NORTHCAP UNIVERSITY
GURUGRAM-122017

Table of Contents

TITLE	PAGE NO
1. Abstract	2
2. Introduction	3
3. Problem Statement and Objective	4
4. Dataset Overview	5
5. Data Preprocessing and Parsing Logic	6
6. System Design and Architecture	7–8
7. User Interface Design	9
8. Features and Functionalities	10
9. Implementation Details	11–12

Abstract

This project presents the development of a web-based **Resume Upload and Search Application** that automates resume parsing and enables keyword-based searching to assist recruiters in finding ideal candidates efficiently. Candidates can upload resumes in .docx format, which are parsed using the **Apache POI** library to extract structured fields such as name, skills, experience, and education. The parsed data is stored in a **MySQL database**, enabling recruiters to perform fast and precise keyword-based searches to identify candidates matching specific job requirements.

This application is designed to **minimize manual effort**, reduce errors, and speed up the recruitment process. Traditional methods of manually screening resumes are time-consuming and inefficient, especially when dealing with large volumes. By leveraging automation, the system allows recruiters to focus more on strategic decision-making rather than administrative tasks.

The backend is implemented using **Java Servlets (J2EE)** for scalability and robustness, while the frontend leverages **JSP, HTML, CSS, and JavaScript** to provide a clean and responsive user interface. Key system modules include candidate registration and login, resume upload and parsing, admin verification, recruiter-based search, and server health monitoring.

Additionally, the application includes a **real-time health check API** to ensure high availability and reliability. The system also supports role-based access control to ensure security and privacy of candidate data. With modular architecture and clean separation of responsibilities, the application is scalable and ready for future enhancements, including PDF parsing, bulk resume uploads, and integration with machine learning models for intelligent ranking and recommendation.

This solution holds promise for deployment in HR departments, recruitment agencies, and job portals, significantly improving their operational efficiency and enabling **data-driven hiring decisions**.

1. Introduction

1.1 Background

Recruiters often sift through a large number of resumes to shortlist suitable candidates, a task that is both time-consuming and inefficient when done manually. In many organizations, there is no centralized system to parse, store, and search resume content, which leads to delays in hiring and missed opportunities.

1.2 Objective

The main goal of the Resume Upload and Search Web Application is to automate the process of resume management by:

- Allowing candidates to upload resumes securely.
- Parsing .docx files to extract relevant data using Apache POI.
- Storing parsed data in a structured relational database.
- Allowing recruiters to perform keyword-based searches.
- Enabling admins to verify and manage candidate authenticity.
- Monitoring server health to ensure uptime and availability.

2. Problem Statement and Objective

2.1 Problem Statement

Recruiters struggle with:

- Manual handling of resumes.
- Inability to quickly filter or search based on skills/experience.
- Lack of centralized storage or automated parsing capabilities.

2.2 Proposed Solution

An end-to-end web application that supports:

- Secure candidate login and resume upload.
- Automatic parsing of resumes and data storage.
- Efficient search using keywords for skills, education, etc.
- Role-based access for candidates, recruiters, and admins.
- System health monitoring to ensure operational reliability.

3. Dataset Overview

Since this project deals with user-uploaded resumes, there is no static dataset. However, during development, sample .docx resumes were created with varying content structures to simulate real-world scenarios.

Each resume typically contains:

- Name
- Contact Details
- Skills
- Education
- Work Experience

Parsed fields are stored in the following database structure:

Field	Data Type
Name	VARCHAR
Email	VARCHAR
Skills	TEXT
Experience	TEXT
Education	TEXT
Resume_File	BLOB/VARCHAR
Verified_Status	BOOLEAN

4. Data Preprocessing and Parsing Logic

4.1 Apache POI for Parsing

- Apache POI XWPFDocument and XWPFPParagraph classes are used to read .docx resumes.
- Each paragraph is scanned for keywords (e.g., “Skills”, “Experience”).
- Text following these headers is stored into respective database fields.

4.2 Data Cleaning

- Remove extra whitespaces, special characters.
- Normalize skill keywords (e.g., Java, JAVA → java).
- Ensure email and contact info are extracted using regex.

5. System Design and Architecture

5.1 Overall Architecture

- Frontend: JSP, HTML5, CSS3, JavaScript
- Backend: Java Servlets (J2EE), JDBC
- Database: MySQL
- Server: Jetty Embedded Server
- Resume Parser: Apache POI
- Others: CORS, Health Check Endpoint

5.2 Module Breakdown

Module	Description
Authentication	User login/registration with session handling.
Resume Management	Uploading, parsing, and storing resume content.
Recruiter Search	Search by keywords (skills, education, etc.).
Admin Panel	Candidate verification, database viewing.
Health Monitor	Endpoint to verify server availability.

6. User Interface Design

- Login/Registration Page: Simple forms for new users and returning users.
- Candidate Dashboard: Upload .docx resume, view parsing results.
- Recruiter Dashboard: Search by keyword, view matched candidates.
- Admin Panel: Verify users, view status.
- Mobile Responsiveness: Achieved using CSS Flexbox/Grid.

The screenshot shows the homepage of the QuickHire platform. At the top, there is a navigation bar with the logo 'QuickHire' and three buttons: 'Candidate Login', 'Candidate Register', and 'Recruiter Login'. Below the navigation bar, the main heading 'Find Your Dream Job' is displayed in green, followed by the sub-instruction 'Upload your resume and let recruiters find you!'. A descriptive text explains that QuickHire uses advanced AI to parse resumes and match users with opportunities. There are two calls-to-action: 'Get Started' (green button) and 'Sign In' (blue button). To the right, there is a large green icon of a person's head and shoulders. Below this, three service cards are shown in a grid:

- Upload Resume**: An icon of a resume with an upward arrow. Below it, text says: 'Upload your resume in PDF or DOCX format and let our intelligent parser extract your skills and experience.'
- Get Discovered**: An icon of a magnifying glass. Below it, text says: 'Recruiters can search for specific skills and find your profile if you're a match.'
- Get Shortlisted**: An icon of a checkmark inside a circle. Below it, text says: 'Recruiters can shortlist your profile and get in touch for interviews.'

7. Features and Functionalities

Feature	Description
Resume Upload	Upload .docx files via secure UI.
Resume Parsing	Apache POI extracts Name, Skills, Experience, etc.
Keyword Search	Recruiters can find candidates using specific terms.
Candidate Verification	Admins validate the authenticity of data.
Health Check API	Confirms server is "UP" or "DOWN".
Session Management	Role-based session tracking and secure access.

8. Implementation Details

8.1 Technologies Used

- Java Servlets and JSP (MVC architecture)
- JDBC for database interactions
- Apache POI for file parsing
- Jetty as embedded lightweight server
- MySQL for structured data storage

9. Results and Testing

9.1 Manual Testing

Test Case	Input	Expected Result	Status
Upload Valid Resume .docx file		Fields extracted	<input checked="" type="checkbox"/> Pass
Search "Python"	Keyword	Show relevant candidates	<input checked="" type="checkbox"/> Pass
Invalid Login	Wrong credentials	Show error	<input checked="" type="checkbox"/> Pass
Admin Verification	Candidate ID	Status updated	<input checked="" type="checkbox"/> Pass

9.2 Performance Metrics

- Parsing speed: ~0.5s per resume
- Search query response time: < 100ms
- Zero downtime recorded during test runs

10. Conclusion and Future Scope

Conclusion

The Resume Upload and Search Web Application automates resume parsing and simplifies candidate discovery for recruiters. With keyword-based searching and structured storage, the system reduces manual workload and enhances decision-making speed.

Future Enhancements

- Support for PDF Parsing

- ML/NLP-based Resume Ranking
- Bulk Resume Uploads
- REST APIs for external integration
- Recruiter Notifications and Email Alerts

11.Demo

1. REGISTER/ LOGIN



Create a Candidate Account

Join QuickHire and get discovered by recruiters

Full Name

Enter your full name

Email Address

Enter your email

Password

Create a password

Confirm Password

Confirm your password

2. UPLOAD YOUR RESUME

The screenshot shows the QuickHire Candidate Dashboard. At the top, there's a header with the QuickHire logo, a welcome message "Welcome, Shefali Khera", a "Back to Home" button, and a "Logout" button. On the left, there's a sidebar with a profile picture, the name "Shefali Khera", and the title "Candidate Dashboard". Below the sidebar, there's a green navigation bar with "Dashboard" selected. The main content area has a title "Dashboard" with two cards: "Resume Status" (green icon, status: "Resume uploaded and parsed") and "Application Status" (blue icon, status: "Not reviewed yet"). Below these cards is a section titled "Upload Your Resume" with instructions: "Upload your resume in PDF or DOCX format. Our system will parse it and extract relevant information." It includes a "Choose File" button, a file input field showing "No file chosen...", and a green "Upload Resume" button.

3. RESUME PARSED SUCCESSFULLY

The screenshot shows the QuickHire Candidate Dashboard after a resume has been uploaded. It features two main status cards: "Resume Status" (green icon, status: "Resume uploaded and parsed") and "Application Status" (blue icon, status: "Pending"). Below these cards is a section titled "Upload Your Resume" with instructions: "Upload your resume in PDF or DOCX format. Our system will parse it and extract relevant information." It includes a "Choose File" button, a file input field showing "No file chosen...", and a green "Upload Resume" button. At the bottom, there's a green success message box with the title "Resume Parsed Successfully!" and the text: "Your resume has been uploaded and parsed. Click on \"View My Resume\" to see the extracted information."

4. VIEW YOUR RESUME OR REUPLOAD

The screenshot shows the QuickHire Candidate Dashboard. At the top, there is a logo and the text "QuickHire". To the right, it says "Welcome, Shefali Khera" and has a "Back to Home" button. On the left, there is a sidebar with a profile picture, the name "Shefali Khera", and the title "Candidate Dashboard". Below this are links for "Dashboard" and "View My Resume". The main content area is titled "My Resume". It contains sections for "Personal Information" (Name: Shefali Khera, Email: khera.shefali20@gmail.com, Phone: 9718932467), "Education" (The NorthCap University, Bachelor of Technology - BTech, Computer Science Sep 2022 - Jun 2026 HansRaj Model School Licenses & Certification), and "Work Experience" (Oct 2023 - Present (5 months) Public Outreach CRY - Child Rights and You, Jul 2023 - Present (8 months) Head Management OPTICA NCU).

5. FOR RECRUITER, LOG IN AND SEE RESUMES

The screenshot shows the QuickHire Recruiter Dashboard. At the top, there is a logo and the text "Admin". To the right, it says "Recruiter Dashboard". On the left, there is a sidebar with a profile picture, the name "Admin", and the title "Recruiter Dashboard". Below this are links for "All Resumes", "Shortlisted", and "Rejected". The main content area is titled "Resume Search". It includes a search bar with placeholder text "Search by skills, experience, education..." and a "Search" button. Below the search bar, there is a "Filter by:" section with buttons for "All", "Shortlisted", "Rejected", and "Pending". It also shows "8 results found". The results are listed in three columns under "WORK EXPERIENCE". Each result includes an email address (mehak.kaur1705@gmail.com), a phone number (9871872385), and a list of skills. The first result is labeled "Shortlisted", the second is "Rejected", and the third is "Shortlisted".

6. SEARCH A KEY WORD TO FIND MATCHING RESUMES

The screenshot shows a resume search interface. At the top, there is a search bar with the query "java". Below the search bar, there are filter options: "All", "Shortlisted", "Rejected", and "Pending". It indicates "4 results found". The results are listed in three cards:

- Shefali Khera** (Pending)
Email: khera.shefali20@gmail.com
Phone: +91 9718932467
Skills
- SHEFALI KHERA** (Rejected)
Email: khera.shefali20@gmail.com
Phone: +91 9718932467
Skills
- Shefali Khera** (Pending)
Email: khera.shefali20@gmail.com
Phone: +91 9718932467
Skills

7. ACCEPT OR REJECT RESUMES

The screenshot shows a recruiter dashboard for an "Admin" user. On the left, there is a sidebar with navigation options: "All Resumes", "Shortlisted" (selected), and "Rejected" (highlighted with a green background). On the right, there is a search interface with a search bar containing "java". Below the search bar, there are filter options: "All", "Shortlisted", "Rejected" (selected), and "Pending". The results are shown in a card:

SHEFALI KHERA (Rejected)

Email: khera.shefali20@gmail.com
Phone: +91 9718932467

Skills

python mysql azure technical skills
tableau excel microsoft azure
jupyter pycharm visual studio
ms office data mining
machine learning data science
data analysis data structures

VIDEO LINK: https://drive.google.com/drive/folders/1jttgZBodo9K-XxhJ_svEdi38cwAHDCn5?usp=sharing

12. References

1. Apache POI Documentation – <https://poi.apache.org/>
2. Java Servlet API – <https://docs.oracle.com/javaee/7/api/>
3. Jetty Server Docs – <https://www.eclipse.org/jetty/>
4. MySQL JDBC – <https://dev.mysql.com/doc/connector-j/>
5. W3Schools – <https://www.w3schools.com/>
6. GitHub Docs – <https://docs.github.com/>