

XML Parser for Email Resume System

Vivek Koli ¹, Anum Khan², Rajasi Pawaskar ³ Prof.Supriya Solaskar ⁴

¹195.vivekkoli@gmail.com

²khananum632@gmail.com

³rajasipawaskar7@gmail.com

⁴supriya147s@gmail.com Prof. Supriya Solaskar, Dept. of IT, Atharva College of Engineering, Maharashtra, India

ABSTRACT-

XML (eXtensible Markup Language) is a text based language derived from Standard Generalized Markup Language (SGML) that allows storage and exchange of information between applications, and for which it is now considered the standard means. It was designed to provide flexible information identification in web documents.. Parsing is a core operation performed on any XML document before it can be navigated, queried, or manipulated. By parsing an XML document, we can choose to view only some specific data, instead of all the data from the original input XML document. We can apply this principle to the very commonplace process of job application. Job applicants usually have to prepare a resume consisting of all relevant personal information and educational qualifications. These resumes are sent to the company's e-mail, where the company recruiters go through each and every e-mail to select suitable candidates for the interview. This approach can be very tedious for a larger number of applicants. The proposed work intends to provide all applicants with a common online form which will consist of all the necessary fields, viz. name, age, e-mail, etc. The applicant form, which is sent to a server and stored in .xml format, will be parsed through using Java code. The recruiters can set filters according to the needs of the job position using the provided user interface, and the results will be displayed accordingly.

Key Words: XML, DOM, AJAX, JSON.

1. INTRODUCTION

The recruitment process today as we know it is typically as follows: When an organization is hiring, they specify the requirements of the position through their chosen medium of announcement or advertisement, and invite eligible applicants to send their resumes via email to the specified company email address. Resume documents are usually sent as email

attachments in commonly used document formats such as Microsoft Word Document (.doc, .docx), text (.txt) or Portable Document Format (PDF). While it is expected that all responding applicants will consider their eligibility before responding, it is not uncommon that an ineligible individual should still send in an application in the hope of getting the job. It is also mostly the case that there are some applicants whose qualifications or experience instantly make them highly desirable candidates for a job interview, while others seem like less interesting choices. However, the only way to know this for sure is for the recruiter to open every applicant email, read the email, download the attached resume, and examine it, before finally deciding if the applicant seems worthy or not. While this may seem like a doable task on a small scale, such a situation becomes problematic in case of large scale organizations or highly reputed companies offering coveted positions, because the number of applying candidates is simply vast. It is very difficult for the recruiter to go through each and every parameter of all the attached resumes and then select a few candidates for a job interview.

2. METHODOLOGY

This project includes creation of a web application that provides separate pages for applicant and recruiter. The applicant page allows uploading of data through text fields, and a resume attachment. The attachment is automatically sent as a mail to the required company email address. The registration data of the user is stored in a relational database, and the resume data is stored in .xml format. During the selection process, qualification criteria can be defined and eligible or interesting applicants can be determined. Once a set of candidates has been selected, it is possible to manually check the automatically generated emails containing their resumes, and scrutinize them in detail. For the purpose of this project, it is being considered that the recruitment needs are those of an I.T. Company.

3. SYSTEM DESIGN

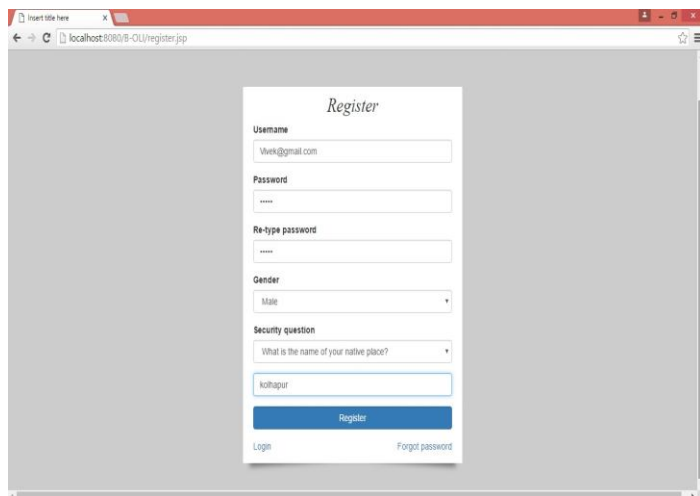


Fig1. Registration page

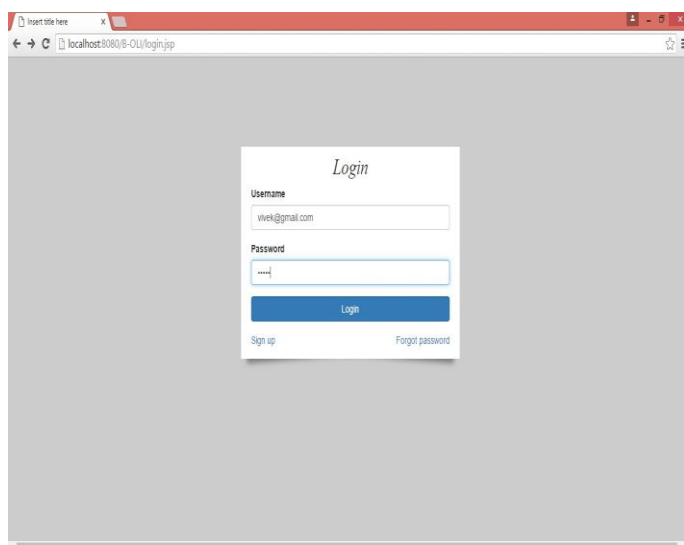


Fig 2. Login page

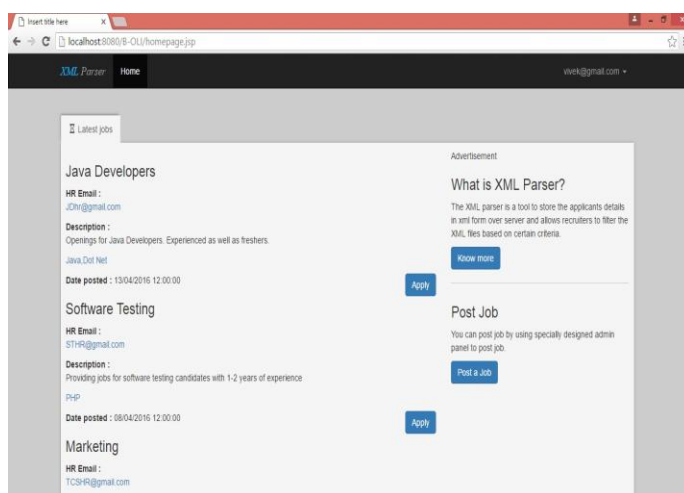


Fig3. User homepage

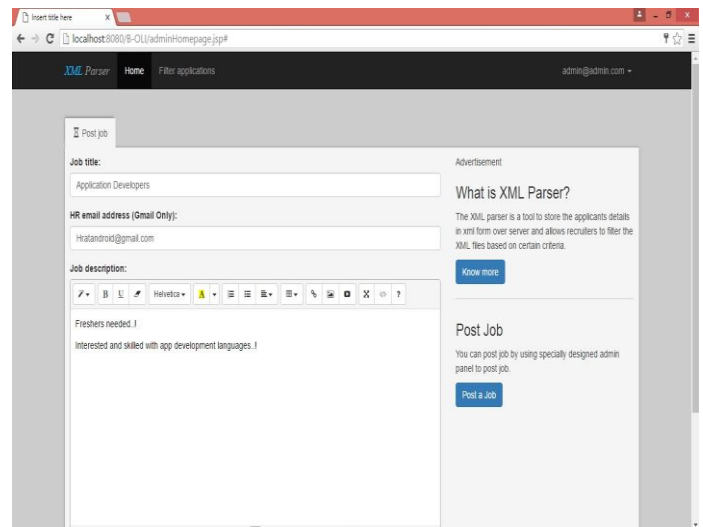


Fig4. Admin homepage

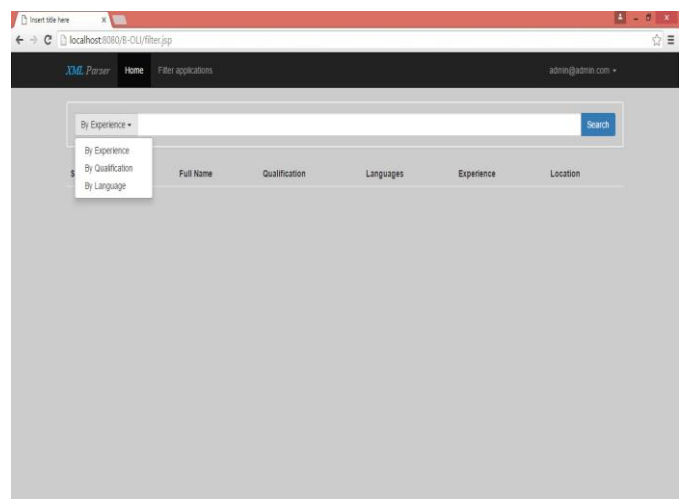


Fig5. Filter application

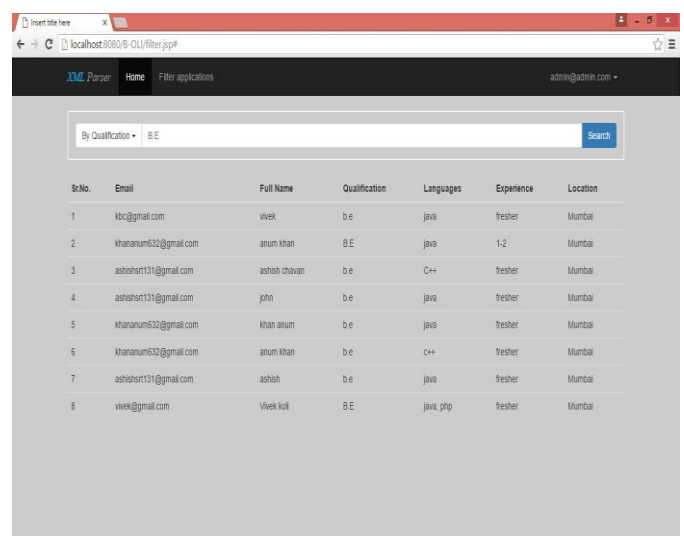


Fig5. Working of filter

4. PROPOSED SYSYTEM

The proposed system aims to develop a GUI in web application for an employee website. A registration form is to be filled by the applicant and then login to view the available job openings. The applicant will select and apply for the job and attach his full resume with it. The application will be submitted and stored in XML format on the server. On the recruiter side, the HR will login and set the filters according to his/her requirements. A list of shortlisted candidates will be displayed based on the filters set. The list of eligible candidates is obtained through XML parsing. Once the recruiter has obtained a filtered list of interesting/eligible candidates, he/she can check the company inbox for the e-mail with that particular candidates unique identifier in the subject field. That e-mail will contain the resume uploaded by the candidate at the time of application.

Applicant Side:

1. The Applicant registers with the portal and then logs in to the portal.
2. Now the Applicant can view details of posted vacancies by various organizations and upload his/her resume.
3. Applicant will address this upload to the official e-mail of desired company.

Recruiter Side:

1. The Recruiter registers company details with the portal and then logs in to the portal.
2. The Recruiter posts the job requirement.
3. She/he can view details of the Applicants and shortlist candidates for the job interview by setting the desired

5. IMPLEMENTATION

The proposed system has two sides, they are, the applicant side and the recruiter side. On the applicant side is a registration form which needs to be filled by the job applicant. After registration, the details of the applicant will be stored in database to authenticate the user when he logs in. After login is complete, a list of job openings will be displayed to the applicant. The applicant can apply to any of them based on his/her interests. After clicking on the 'apply' button, the user will be given a form wherein he has to fill all the requirements such as experience and qualification. After filling the details, the applicant has to attach his resume and submit the form. The form is stored in XML format on the server and a message is displayed about

the successful submission of the form. On the recruiter side, the HR logs in as an admin. He/she then posts job openings with the required filters such as experience of 1-2 years and so on. A list of shortlisted candidates will be displayed. If the recruiter finds any candidate interesting and apt for the job, he/she can check the company inbox for the e-mail with that particular candidates unique identifier in the subject field. He/she can then decide whether or not an interview should be scheduled for him/her. The system architecture is shown in the figure below:

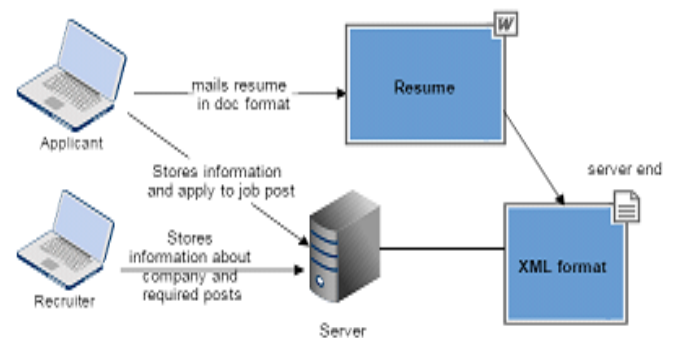


Fig. System architecture

6. TESTING AND EVALUATION

The testing procedure for this project consisted of the following steps:

- 1) Set the values of sender and receiver emails, and sender password from 'Mailer.java'
- 2) Create a registration
- 3) After registration, login to create an application
- 4) Fill the application form; upload a document (i.e. the 'resume') to the form, and finally submit the application
- 5) Open the login page once again and login as admin
- 6) On the filter page, select the criteria that had been entered in the job application
- 7) Check if the displayed result contains the name entered during registration

Although in the actual code the sender and receiver address strings are dummy addresses, it was necessary to replace these with functional email addresses during testing. Also necessary was a functional internet connection and firewall access for all involved programs. This would ensure that the email is sent properly. The testing results were found to be satisfactory. No errors were encountered during the process, and no changes to the code were seen necessary.

7. CONCLUSION

Parsing an XML document provides a quick way to view data we specifically need. Sorting through a large number of emails for the purpose of candidate selection is inconvenient. The presence of a centralized system to automate this process can simplify the task and reduce the burden on human resources. An XML parser based system for handling resumes provides a fast and convenient way to shortlist applicants/candidates based on the information they have provided and skills required for the job. The system considerably automates the tedious manual task of candidate selection and thus forms a novel solution for an existing problem.

REFERENCES

- [1] International Journal of Computer Engineering and Technology (IJCET), ISSN 0976-6367(Print),ISSN 0976 - 6375(Online), Volume 5, Issue 3, March (2014).
- [2] V. Deshmukh, G. Bamnote, "XML Parsing using Various Data Structures," IJCSA., vol. 6, no. 2 (2013, 400).
- [3] S. Taylor, P. Watters, "Trustworthy E-mail Using Secure XML Web Service," in Proc. 7thIEEE Int. Conf.E-commerce Technology., 2005.