WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

ONLINE JOB PORTAL

Deepika Guntu[1], J Anuradha Rao[2], Renuka Rajimwale[3], Ileen Singh[4]

[1]Department of Computer Science & Engineering, Shri Shankaracharya Technical Campus - India
[2]Chhattisgarh Swami Vivekanand Technical University, Technical university in Bhilai - India
(Session: 2024-25)
deepika1411g@gmail.com[1], anuradha.rao.sstc@gmail.com[2], renukarajimwale1608@gmail.com[3], ileensingh14@gmail.com[4]

ABSTRACT:

The Online Job Portal is a web application aimed at simplifying the job search and recruitment process. It provides an intuitive interface where students can explore job opportunities based on technology, location, and job category. Recruiters can post job openings and search for candidates that match their requirements. The portal offers distinct login functionalities for both user types and supports profile management, job history tracking, and real-time updates of new openings. With the integration of filters and personalized user experience, the system bridges the gap between job seekers and employers, making the process more efficient, transparent, and streamlined.

Introduction:

In the current age of digitalization, job search and recruitment have evolved from traditional methods such as newspaper advertisements and word-of-mouth references to more dynamic, scalable, and interactive online job portals. The Online Job Portal project is designed to act as a platform where students (job seekers) can browse, search, and apply for jobs based on their preferences, and recruiters can post job openings and find suitable candidates. This portal allows users to register and log in either as students or as recruiters. Students can create and update their profiles, including uploading resumes and listing their skills, while recruiters can post job openings, view applicants, and shortlist candidates. The platform includes advanced search and filter functionalities to enable quick access to relevant job opportunities. Additionally, users can browse their activity history for easy reference.

Problem Statement:

Finding a job or hiring the right candidate is often a tedious and inefficient process, particularly for fresh graduates and small recruiters. Key challenges include:
Lack of centralized job postings tailored to specific technologies or fields.
Difficulty in filtering jobs based on user preferences such as location, experience, and skill set.
Inadequate tools for recruiters to manage applications, sort candidates, and track hiring.
Absence of a customizable platform that serves both job seekers and recruiters with equal focus.
Hence, there is a pressing need for a solution that provides an integrated platform for managing job searches and recruitments with personalized and efficient tools.

Literature Review:

Several online job portals like Naukri.com, Indeed, LinkedIn, and Glassdoor offer features for job searches and recruitments. These platforms focus on large-scale hiring with advanced AI-based recommendations. However, they often lack customization for smaller or domain-specific needs, and may not cater well to fresh graduates or niche recruiters. Our portal draws inspiration from their best practices while focusing on simplicity, user-specific customization, and academic integration, especially suited for college students and startups or mid-size recruiters..

Objective:

The primary objective of the Online Job Portal is to create an efficient and user-friendly platform that bridges the gap between job seekers and recruiters. This system aims to allow students to easily search and apply for jobs based on various criteria such as job category, preferred location, and relevant technologies. At the same time, it provides recruiters with the tools to post job openings, filter applicants, and manage their hiring process effectively. By offering dedicated login systems for both students and recruiters, the portal ensures personalized access and functionality for each user type. Additionally,

the platform includes features like profile management, resume uploads, browsing history, and real-time job updates, all designed to enhance user experience and streamline the job application and recruitment process.

Methodology:

The development of the Online Job Portal follows the Agile Software Development Life Cycle (SDLC) to ensure continuous feedback and iterative improvement. The process begins with requirement analysis, where the features needed by both students and recruiters are identified. This is followed by system design, which includes creating wireframes, database schemas, and use case diagrams. Frontend development involves building responsive user interfaces using HTML, CSS, and JavaScript or React, while the backend is developed using technologies like Node.js or Django to handle server-side logic and APIs. The portal uses a relational or NoSQL database such as MongoDB to store user information, job posts, applications, and other relevant data. Secure authentication mechanisms are implemented to allow role-based login for students and recruiters. Advanced job filtering features are added to refine search results based on user preferences.

Purpose:

The purpose of this research and development project is to create a comprehensive and interactive Online Job Portal that simplifies and enhances the job-seeking and hiring process for both applicants and recruiters. The portal serves as a bridge to connect talents with opportunities in a more efficient and accessible manner. The key purposes are categorized as follows:.

1. To Simplify Job Search for Candidates- The portal allows users to easily search for jobs using multiple filters such as job category, preferred location, and technology. This targeted approach helps candidates find relevant jobs faster and more efficiently.
2. To Provide Latest Job Updates**-** The system includes a section for displaying the latest job openings and updates, ensuring users are informed about new opportunities as they become available.
3. To Enable Dual User Roles- The platform supports separate login systems for both students (job seekers) and recruiters, offering role-specific access and functionalities to improve user experience and streamline interaction.
4. 4.Organize and Track Job Browsing History- A browsing feature is included to help users revisit previously viewed job postings, allowing for better decision-making and follow-up actions.
5. To Enhance Job Categorization- Jobs are well-categorized based on various fields and sectors, making it easier for users to explore opportunities in their are of interest or specialization.
6. To Facilitate Profile Management- The portal includes a personal profile section for job seekers where they can add and manage their name, contact details, skills, email, and resume—making it easier for recruiters to assess candidate qualifications.5.4 Energy Management System (EMS).
7. To Improve Recruiter-Candidate Interaction- By allowing recruiters to post jobs and view candidate profiles, the portal fosters smoother communication and faster hiring processes.
8. To Offer Secure Account Features- Functions like logout and session management ensure user accounts are kept secure, maintaining the confidentiality of personal information.Smart Meter.

Technology Stack Justification

1. Frontend (Client-Side) Technology Used: React.js

Justification: React is a component-based JavaScript library ideal for building dynamic, responsive UIs. Virtual DOM improves performance for user interactions (job listings, filters, dashboards). Strong ecosystem (Redux, React Router) supports complex single-page applications (SPAs). Large community and reusable components enhance productivity and maintainability.

2. Backend (Server-Side) Technology Used: Node.js with Express.js

Justification:

Non-blocking I/O and event-driven architecture suits real-time features like job notifications and chat. JavaScript on both frontend and backend provides a consistent development experience (full-stack JS). Express.js is minimal, flexible, and ideal for building RESTful APIs for job listings, user profiles, etc.

3. Database Primary Database: MongoDB (NoSQL)

Justification:Schema-less design supports flexible data models for users, employers, jobs, and applications. JSON-like documents align naturally with JavaScript/Node.js stack. Good for fast development and scaling unstructured or semi-structured data. Alternative (if relational DB needed): PostgreSQL Useful if strict relational integrity is needed (e.g., normalized schema for jobs ↔ applicants ↔ companies). Advanced querying and indexing capabilities.

4. Authentication Technology Used: JWT (JSON Web Tokens) with Passport.js or Firebase Auth

Justification: JWT allows stateless, scalable authentication and secure session handling. Easily integrates with role-based access for job seekers, employers, and admins.

5. Cloud Storage (Optional) Technology Used: AWS S3 / Cloudinary
Justification: Used for storing resumes, company logos, and job-related documents. Scalable, secure, and easily integrated with the backend.
6. Search & Filtering Technology Used: Elasticsearch (optional enhancement)
Justification: Full-text search and advanced filtering over large job datasets. Improves user experience with fast, relevant search results.
7. Hosting & Deployment Frontend: Vercel / Netlify / AWS Amplify
Backend: Render / Heroku / AWS EC2 Database: MongoDB Atlas / AWS RDS
Justification: Cloud platforms offer scalability, CI/CD, SSL, and easy version control integration. Quick to deploy for MVPs, and flexible for future scaling.
8. Version Control and Collaboration Technology Used: Git + GitHub
Justification: Standard for source control and team collaboration. GitHub Actions can be used for automated CI/CD.

Features and Functionalities

1. 1.User Registration and Login- Student Login: Students/job seekers can register and log in to access job search features, update their profile, and apply for jobs. Recruiter Login: Recruiters can log in to post job openings, browse student profiles, and manage applications.

2. Job Search with Advanced Filters- Keyword Search: Users can search jobs using specific keywords.
Filter Options: Jobs can be filtered by: Preferred Location, Job Category (e.g., IT, Finance, Marketing), Required Technology/Skills, Job Type (Full-time, Part-time, Internship)

3. Latest Job Updates- Real-time updates and listings of the latest job opportunities on the home or jobs page.
Option to sort by most recent postings.

4. 4. Job Categories- Jobs are grouped under various categories or industries. Easy navigation for users to explore opportunities in their field of interest.

5. 5. Job Details Page- Each job post includes full details like:
Company Name, Job Description, Required Skills, Salary Range, Application Deadline.

6. 6. User Profile Management- Students can add/update:
Name, Mobile Number, Email, Skills and Experience, Upload Resume (PDF/Doc), Recruiters can update company profile and job postings.

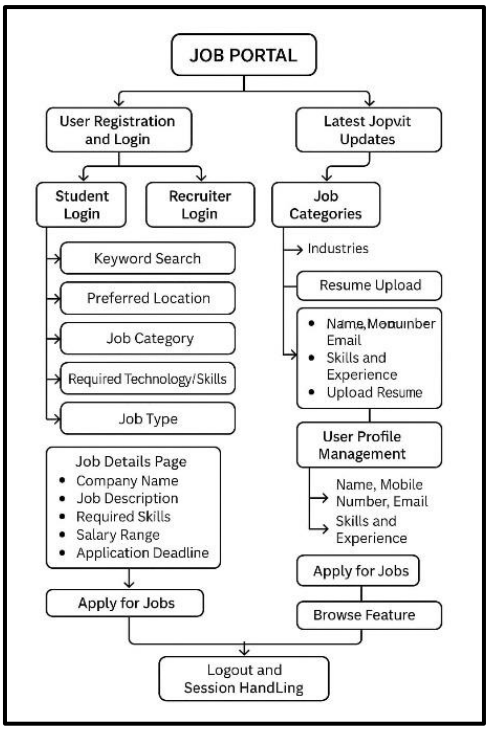
7. 7. Resume Upload- Students can upload and manage their resumes, which recruiters can view during the selection process.

8. Job History (Browse Feature)- Keeps track of previously viewed job listings. Allows users to revisit jobs they were interested in.

9. Apply for Jobs- Students can apply for jobs directly through the portal. Applications are stored and visible to recruiters.

10. Logout and Session Handling- Secure logout functionality. Proper session handling to prevent unauthorized access.

Fig.1 Flow diagram of project



Software Used

The development of the Online Job Portal was carried out using modern, scalable, and efficient technologies. Each software tool and framework was selected based on its relevance, performance, and suitability for web application development.

1. React.js (Frontend Framework)

Why Used: React was chosen for building the user interface due to its component-based architecture, reusability, and efficient rendering using the Virtual DOM. It allows for a dynamic and responsive user experience, which is essential for features like job filtering, form inputs, and interactive navigation.

2. Node.js (Backend Runtime Environment)

Why Used: Node.js was used to handle the backend logic because of its non-blocking, event-driven architecture, which provides high performance and scalability. It's well-suited for real-time data handling and integrates seamlessly with MongoDB.

3. Express.js (Backend Framework)

Why Used: Express was used as a lightweight framework on top of Node.js to simplify routing, middleware integration, and API development. It helps manage HTTP requests for login, job applications.

4. MongoDB (Database)

Why Used: MongoDB, a NoSQL database, was chosen for storing user profiles, job postings, resumes, and application history. Its schema-less nature allows flexibility in handling various data types, making it ideal for a platform that deals with different user roles.

5. Visual Studio Code (Code Editor)

Why Used: VS Code is a powerful and widely-used code editor with support for extensions, Git integration, and debugging tools, which enhanced productivity during development.

6. Git and GitHub (Version Control)

Why Used: Git was used for version control to manage changes in the codebase. GitHub provided a remote repository for collaboration and project backup.

Conclusion:

The Online Job Portal project successfully achieves its aim of providing a reliable, user-friendly, and efficient platform for connecting job seekers with recruiters. By offering essential features like categorized job listings, advanced search filters, profile management, resume uploads, and dual login access, the system streamlines the entire recruitment process. It not only benefits students by helping them find relevant job opportunities, but also supports recruiters in identifying the right candidates with minimal effort. The system reduces manual processes, increases accessibility, and enhances user interaction, making it a practical and scalable solution for modern hiring needs.

Future Work:

- Integrate AI-based job recommendations using student profiles, skillsets and previous applications.
- Add predictive analytics to suggest career paths based on academic performance and project history.
- Enable real-time integration with college ERP or LMS to auto-fetch student data and validate eligibility.
- Implement resume scoring using Natural Language Processing (NLP) to match candidates with job descriptions.
- Enhance recruiter dashboards with analytics to track application success, hiring timelines, and applicant quality.

REFERENCES:

1. Werk-An Online Job Portal (International Journal of Research Publication and Reviews, Vol. 5, Issue 6, June 2024) https://ijrpr.com/uploads/V5ISSUE6/IJRPR30532.pdf?utm_source=chatgpt.com
2. Comparative Study of Job Portals in India by Shaista Haleem and Dr. Firdous Ahmad Shah International Journal of Research in Engineering, Science and Management https://www.ijresm.com/Vol.2_2019/Vol2_Iss5_May19/IJRESM_25_41.pdf
3. Online Job Recruitment System by Adeyemo S. A., 2020 International Journal of Computer Applications Technology and Research (IJCATR) https://www.ijcat.com/archives/volume9/issue5/ijcatr09051007.pdf
4. React.js Official Documentation https://reactjs.org/docs/getting-started.html
5. Node.js Documentation https://nodejs.org/en/docs/
6. Express.js Documentation https://expressjs.com/en/starter/installing.html
7. MongoDB Documentation – Databases and Collections https://www.mongodb.com/docs/manual/core/databases-and-collections/
8. *JWT (JSON Web Tokens) Introduction and Guide* https://jwt.io/introduction/
9. *Git & GitHub Documentation*

https://docs.github.com/en

10. *Cloudinary for File Uploads – Documentation*
https://cloudinary.com/documentation/upload_images
11. *Vercel Deployment Guide for React Applications*
https://vercel.com/docs/concepts/projects/overview
12. *A Review on E-Recruitment Portals and Their Impact on Employment Generation* by Dr. P. Selvi and A. Ramya
International Journal of Advanced Science and Technology (IJAST)
http://sersc.org/journals/index.php/IJAST/article/view/25526
13. *Job Portal using MERN Stack – Case Study and Source Code* by Chandan Verma
https://github.com/chand1012/mern-job-board
14. *UI/UX Design Principles for Web Applications* by Usability.gov
https://www.usability.gov/what-and-why/user-interface-design.html
15. *Elasticsearch Official Guide*
https://www.elastic.co/guide/en/elasticsearch/reference/index.html
16. Implementing Efficient Job Portal: REST API Design and Role-Based Access Control by K. Abhay, Kunal Juvvala & M. Manivardhan