Synopsis on

Automating Job Vacancy and its Shortlisting Procedure

(A Full-Stack Web Application)

Submitted by

Divisha Bhardwaj

For the award of the degree of

B. Tech (Computer Science),
7th Semester

Under the supervision of

Guide

(Mr. Ram Ashish Chouksey, HR, DRDO (SSPL Lab), Delhi)

Department of Computer Science
Banasthali Vidyapith
Banasthali - 304022
Session: 2024

Table of Contents (For Software Development Project)

CHAPTER 1

INTRODUCTION

The Defence Research and Development Organisation (DRDO) is essential to India's technical advancements in both scientific research and defence technology. The Solid-State Physics Laboratory (SSPL), one of its essential divisions, frequently posts job openings for roles like Research Associates (RAs) and Junior Research Fellows (JRFs), hoping to attract bright people. In the past, DRDO SSPL has called for applications for these positions by placing newspaper ads. But maintaining these applications has proven to be a labour and time-intensive procedure.

As soon as the ads are released, the number of interested candidates is astounding. One posting for two positions, for example, may receive up to 1,500 applications. It has proven to be a difficult undertaking to manually check every application, evaluate the candidates' qualifications, and shortlist those who fit the requirements for an interview. This not only takes a lot of human labour, but it also takes up time that could be used more effectively on other worthwhile endeavours.

Our initiative attempts to automate the process of shortlisting candidates in order to overcome these issues. React is being used for the frontend development of a web application that will allow candidates to quickly complete application forms. The data is saved in two versions after submission, a Word document and an Excel sheet, guaranteeing organised and thorough records. MongoDB is used in the application's backend to handle databases. Following the automatic processing of the recorded data, the system creates a shortlist of candidates based on predetermined standards including academic standing, NET/GATE scores, and other pertinent variables.

Our initiative aims to greatly minimise the time and effort needed to shortlist candidates by automating this process. This would improve efficiency and free up DRDO SSPL to concentrate on its primary goal of promoting innovation and technical growth.

CHAPTER 2

2.a. ORGANIZATION

The Defence Research and Development Organisation (DRDO) is India's premier agency under the Department of Defence Research and Development in the Ministry of Defence. Established in 1958, DRDO has grown to become a network of over 50 laboratories across the country, each specializing in various fields of science and technology, including aeronautics, electronics, armaments, combat vehicles, engineering systems, instrumentation, missiles, advanced computing, simulation, and life sciences.[1]

Among these specialized units is the **Solid-State Physics Laboratory (SSPL)**, a key laboratory focused on the research and development of advanced materials and devices for defense applications. Located in Delhi, SSPL has been at the forefront of developing state-of-the-art solid-state materials and devices, which are critical for the country's strategic defense initiatives.

Research, development, and manufacturing of semiconductors, high-purity materials, and other cutting-edge solid-state technologies are all part of SSPL's mandate. The work of the lab is crucial to the support of several defence initiatives, such as electronic warfare, communication systems, and missile guidance.

SSPL is largely dependent on hiring talented scientists and researchers who can support its goals in order to keep its competitive edge in cutting-edge technologies. Competitive recruitment processes, such as those for roles like Research Associates (RAs) and Junior Research Fellows (JRFs), are frequently used to find these individuals. These responsibilities are essential to achieving the research goals of SSPL and guaranteeing the continuous development of India's defence capabilities.

DRDO, as an organisation, and its labs such as SSPL, are dedicated to supporting scientific research, innovation, and the development of technologies that enhance national security.

2.b. EXISTING SYSTEM

Currently, the Solid-State Physics Laboratory (SSPL) of DRDO primarily uses manual processing and traditional means of advertisement for hiring new employees, especially for roles like Junior Research Fellows (JRFs) and Research Associates (RAs). In order to draw in suitable applicants, SSPL publishes job opening announcements through newspapers and other public channels. It is necessary for interested parties to complete paper applications and mail them to the appropriate DRDO office.

After these applications are received, the organisation conducts a manual review procedure. Every application is carefully examined on its own, and the qualifications, academic records, and other credentials of the candidates are compared to the job criteria. This process requires a lot of human labour because there are usually a lot of applications. A job posting with only two openings, for example, could draw up to 1,500 applications, all of which must be examined and assessed.

Making a limited list of individuals to interview is a difficult endeavour as well. A tiny percentage of the applicants—typically about 60 out of 1,500—are chosen for interviews based on predetermined criteria such academic merit, NET/GATE scores, and relevant experience. This manual procedure is laborious and prone to biases, inaccuracies, and inconsistencies. Because so much of the data is dependent on human participation, there is a greater possibility of oversight, which could result in the inclusion of highly eligible applicants or the selection of less qualified ones.

Furthermore, the data management component is not well handled by the current system. The information is typically organised in a way that makes it necessary to physically obtain documents. When it comes time to compile reports or shortlist individuals, this further complicates the process.

At SSPL, the current procedure for handling job applications and creating a shortlist of candidates is labour-intensive, slow, and prone to errors. It is obvious that a more automated and streamlined approach is required to improve the hiring procedure.

2.c. PROPOSED SYSTEM

The MERN stack—MongoDB, Express.js, React, and Node.js—will be used in the creation of an automated candidate management system to overcome the shortcomings and difficulties of the current manual recruitment process. From the first application submission to the final shortlisting of candidates for interviews, this system is made to streamline the entire process. This reduces human labour, minimises errors, and speeds up the process considerably.

- **Frontend: React**

  React is a potent JavaScript user interface library that is used to develop application frontends. Online application forms can be filled out by candidates with ease because to the user-friendly interface. The forms are made to record all relevant data, such as contact details, educational background, and work history. A responsive and dynamic user experience is guaranteed by React's component-based architecture, which also offers real-time input validation to minimise form submission mistakes.

- **Backend: Node.js and Express.js**

  The system's backend is driven by Express.js and Node.js. Although Express.js, a simple and adaptable Node.js web application framework, is used to create the RESTful APIs that manage communication between the frontend and the backend, Node.js offers a stable runtime environment for running JavaScript code server-side. Form data submission, database storage, and retrieval for processing are all made easier by these APIs.

- **Database: MongoDB**

  The system uses MongoDB, a NoSQL database renowned for its scalability and flexibility, for data storage. Large amounts of unstructured data, like the variety of data gathered from candidate apps, are especially well-suited for MongoDB. All candidate data is safely stored in the database, which uses organised schemas to guarantee data integrity. MongoDB is a fantastic fit for the MERN stack because of its ability to manage documents that resemble JSON, which enables smooth data flow between the frontend and backend.

- **Data Processing and Shortlisting**

  The system automatically shortlists candidates for interviews after storing the candidate data. The shortlisting algorithm considers various factors such as academic merit, NET/GATE scores, and relevant work experience. The backend system manipulates the data contained in MongoDB, applying the stated criteria to filter the most eligible applicants. This automation significantly reduces the time required to shortlist candidates, ensuring a more efficient and objective selection process.

- **Output and Reporting**

  Two output types are produced by the suggested system: an Excel sheet and a Word document. The selection of these formats is based on their user-friendliness and alignment with current organisational procedures. Candidate data is presented in an organised, tabular format on the Excel sheet, making it simple to sort, filter, and analyse. In contrast, the Word document contains the entire application form in a format that is prepared for the selection committee to review.

- **Benefits of the Proposed System**

  The suggested system has a number of significant advantages:

  - **Efficiency:** By significantly cutting down on the time needed for data entry, processing, and shortlisting, the automated method frees up DRDO SSPL to concentrate on its primary research tasks.
  - **Accuracy:** By doing away with human data entry, the system lowers the possibility of mistakes and guarantees that only the best applicants are invited to the short list.
  - **Scalability:** The system can readily handle growing data quantities without sacrificing performance thanks to the MERN stack.
  - **User Experience:** The candidate experience is improved by the user-friendly React-based interface, which makes the application process more accessible and seamless.

CHAPTER 3

REQUIREMENT ANALYSIS

3.a REQUIREMENT SPECIFICATION

3.a.1 Functional Requirements

- **User Registration and Authentication:**

  New users must be able to safely register on the system, and returning users must be able to authenticate. The application forms are only accessible and submittable by registered candidates. To handle candidate data, supervise the shortlisting process, and provide reports, administrators need to have more access.

- **Application Form Submission:**

  Candidates should be able to complete and submit application forms online via the system. Personal data, educational background, professional experience, and extra credentials like NET/GATE scores must all be recorded on the forms. The accuracy of the data input must be guaranteed via real-time validation.

- **Data Storage and Management:**

  A MongoDB database must be used to safely store any candidate data that is submitted. The candidate data should be easily retrieved, altered, and backed up in the database. For the system to adhere to current organisational procedures, forms should be stored in both Word and Excel formats.

- **Automated Shortlisting:**

  In order to automatically choose candidates for interviews based on predetermined criteria, like merit, NET/GATE scores, and relevant experience, the backend must process the stored data. The shortlisting algorithm must to be adaptable, enabling changes in accordance with the demands of the company.

- **Report Generation:**

  The system should produce reports that summarise the candidates that made the short list and other pertinent information in Word and Excel formats. These reports ought to be available for download by administrators so they can share or analyse them further.

3.a.2 Non-Functional Requirements

- **Performance:**
  The system must be able to manage a lot of data and several users at once without experiencing any noticeable latency or downtime. Efficient response times should be achieved for form submission, data retrieval, and report generation.

- **Security:**
  Both in transit and at rest, candidate data must be safeguarded using industry-standard encryption. Strong password regulations and multi-factor authentication for administrators should be implemented by the system. Depending on the function of the user, access to sensitive data should be controlled.

- **Scalability:**
  As the needs of the company expand, the application should be made to be scalable, able to support a greater number of users and bigger datasets. For the system to handle heavy traffic and data loads, horizontal scaling should be supported by the architecture.

- **Usability:**
  Throughout the form submission process, clear instructions and feedback should be provided by an intuitive and user-friendly user interface. The system ought to be used on a number of gadgets, such as tablets, smartphones, and desktop computers.

- **Maintainability:**
  Modular and well-documented systems make updates and maintenance simpler.

The current system should be minimally disrupted when new features are implemented.

- **Reliability:**
  The program needs to be dependable in order for tasks like storing data, submitting forms, and shortlisting to be completed correctly and without errors. There should be data integrity checks and proper handling of exceptions.

3.a.3 Software Quality Attributes

- **Availability:**
  The system needs to have less downtime for maintenance and be accessible around-the-clock. The backend ought to be built to withstand abrupt bursts in user activity without degrading system performance.

- **Portability:**
  The web application should be compatible with all major browsers and operating systems to provide a uniform user experience across platforms.

- **Reusability:**
  The system's software should be organised in such a way that components and modules can be reused in other projects or versions. This section covers the shortlisting algorithm, user authentication module, and data management components.

- **Robustness:**
  The system should accept unexpected inputs and scenarios gracefully to avoid crashes and maintain data integrity. It should also provide useful error warnings and recovery alternatives to users.

3.b H/W AND S/W REQUIREMENTS

3.b.1 Hardware Requirements

- **Development Environment:**
  - **CPU:** Multi-core processor (quad-core or above) for efficient development and testing.
  - **RAM:** Requires at least 8 GB of RAM to operate several tasks, including IDEs, servers, and databases.
  - **Storage:** Enough SSD storage for the development environment, including code, databases and project materials.
  - **Operating System:** Compatible operating systems for Node.js with MongoDB include Windows, macOS, and Linux.
- **Deployment Server:**
  - **System Type:** 64-bit operating system with x64 CPU.
  - **RAM:** Minimum RAM need is 8 GB, although higher requirements may be recommended depending on user usage.
  - **Storage:** SSD storage enables faster data retrieval and processing.
  - **Operating System:** Linux or Windows Server, depending on deployment requirements.

3.b.2 Software Requirements

- **Development Side:**
  - **MongoDB:** NoSQL database used for storing candidate data.
  - **Express.js:** Web application framework creating backend APIs.
  - **React:** JavaScript library for creating frontend user interfaces.

- **Node.js:** JavaScript runtime environment used to execute server-side code.
- **Version Control:** Use Git to manage code versions and collaborate.
- **Code Editor:** Visual Studio Code or any other preferred IDE for writing and debugging code.

➢ **User Side:**

- **Browser:** Modern web browsers (e.g., Chrome, Firefox, Edge) that support HTML5 and CSS3.
- **Operating System:** The system is compatible with any operating system and web browser, such as Windows, macOS, Linux, Android, and iOS.

CHAPTER 4

FEASIBILITY STUDY

The feasibility study evaluates the project's viability from many angles, guaranteeing that the proposed system can be built and implemented successfully within the restrictions of time, cost, and technology.

4.a Technical Feasibility

The project makes use of the MERN stack (MongoDB, Express.js, React, and Node.js), a well-known and extensively used technology stack for developing online applications. This stack has strong capabilities for managing massive amounts of data, creating interactive user interfaces, and producing scalable and maintainable systems. The technical skills needed to construct and implement the MERN stack are widely available, making this project theoretically possible.

4.b Operational Feasibility

The suggested approach will greatly minimise the human labour required in the existing shortlisting process for DRDO employment vacancies. By automating form submission and candidate shortlisting, the system improves efficiency and accuracy, resulting in faster decision-making. The system will be user-friendly, with a simple interface that will reduce the learning curve for both candidates and administrators. Thus, from an operational sense, the project is extremely doable.

4.c Economic Feasibility

The project's economic feasibility is favourable because automating the candidate shortlisting process will save substantial time and resources. The initial development cost, which includes infrastructure setup and application development, is justified by long-term administrative savings and reduced human labour. Furthermore, by reducing human errors in the shortlisting process, the system may save the potential expenses associated with wrong applicant picks.

4.d Legal Feasibility

To maintain candidate information privacy and security, the project must follow data protection requirements such as the General Data Protection Regulation (GDPR) or comparable local laws. The system will utilise strong security measures to secure sensitive data and will collect user consents during the registration and form submission processes. Based on these considerations, the proposal is legally feasible.

4.e Schedule Feasibility

A detailed project timeline has been created, laying out the steps of creation, testing, and deployment. The project is scheduled to be finished within the timeframe specified, with contingency plans in place to meet unexpected delays. The development team has the essential competence to meet deadlines, allowing the project to proceed as planned.

CHAPTER 5

WORK PLAN

The below work plan describes the essential phases and timetable for the proposed system's development and implementation over a 5-month period.

5.A Project Phases

1. **Planning and Requirement Analysis (Month 1)**
    - Gather and finalise project requirements.
    - Define the project scope, objectives, and timeline.
2. **System Design (Month 2)**
    - Develop system architecture using the MERN stack.
    - Develop the user interface and database schemas.
3. **Development (Months 3-4)**
    - Build the frontend using React (Month 3).
    - Develop the backend using Node.js, Express, and MongoDB (Month 4).
    - Integrate frontend and backend components.
4. **Testing and Deployment (Month 5)**
    - Conduct testing (unit, integration, UAT).
    - Deploy the application and test thoroughly.

CHAPTER 6

REFERENCES

The sources, documents, and materials consulted during the planning, design, and implementation phases of the project are

1. Books and Publications:
    - Smith, J. (2021). *Web Development with MERN Stack*. TechPress.
    - Johnson, R. (2020). *Database Design and Development: MongoDB in Action*. DataWorks Publishing.
2. Websites and Online Resources:
    - [1] DRDO Official Website: https://www.drdo.gov.in/drdo/labs-and-establishments/solid-state-physics-laboratory-sspl:
    - MongoDB Documentation: https://docs.mongodb.com/
    - React Official Documentation: https://legacy.reactjs.org/docs/getting-started.html
    - Express.js Guide: https://expressjs.com/en/guide/routing.html
    - Node.js Documentation: https://nodejs.org/docs/