Job Portal System Project Report

Completed on September 8, 2025

1 Introduction

The Job Portal System is a modern website that helps job seekers and employers connect easily. Users can sign up, create detailed profiles, upload resumes, and manage their skills. Job seekers can apply for jobs, and employers can post and manage job listings. The system uses secure email and password login to keep accounts safe. It has a clean, user-friendly design made with Tailwind CSS and Font Awesome icons. The site works well on different screen sizes, like phones and laptops. This project report explains how the system was built, the tools used, and the steps followed. It also highlights the main results and features of the platform.

2 Abstract

The Job Portal System is a platform that helps job seekers and employers connect easily. Job seekers can make detailed profiles by adding their skills, resumes, and job preferences. Employers can create and manage job posts to find the right candidates. One helpful feature allows users to add or remove skills with just a click. There's also a profile completion checklist that shows progress with visual indicators, making it easy to see what's missing. The system includes a dark mode for better visibility and user comfort. It's designed using Tailwind CSS for styling and Font Awesome for icons. There are two simple roles in the system: Applicant and Employer. Technical problems like circular dependencies and database issues were solved by improving the code and database structure. In the end, the system is user-friendly, works well, and can be expanded with more features in the future.

3 Tools Used

The following tools and technologies were utilized to build the Job Portal System:

- **Spring Boot**: Backend framework for RESTful APIs, user authentication, and business logic.
- **Thymeleaf**: Server-side templating engine for dynamic HTML rendering.
- **MySQL**: Relational database for storing user profiles, job listings, and application data.
- **Tailwind CSS**: Utility-first CSS framework (via CDN) for responsive and modern UI design.
- **Font Awesome**: Icon library (via CDN) for enhancing UI with icons (e.g., checkmarks, save).
- **Java**: Programming language for backend development.
- **Maven**: Dependency management tool for project libraries.
- **Spring Security**: Framework for email/password authentication.
- **MySQL Workbench**: For database schema management and query execution.

4 Steps Involved in Building the Project

The development of the Job Portal System involved the following key steps:

1. **Project Setup**: Initialized a Spring Boot project with Maven, adding dependencies for Spring Web, Thymeleaf, Spring Security, and Spring Data JPA. Configured MySQL database in application.properties.
2. **Database Design**: Designed the user table with fields for fullname, email, password, role (Applicant or Employer), skills, and profile attributes (e.g., bio, resumePath).

Added user_skills table for dynamic skills storage.

3. **User Authentication**: Implemented email/password login using Spring Security, ensuring secure access for Applicants and Employers.
4. **Profile Management**: I created a ProfileController to let users view and edit their profiles. I also made two web pages using Thymeleaf one for viewing the profile (view.html) and one for editing it (edit.html). These pages are styled with Tailwind CSS to look good on all screen sizes and use Font Awesome for icons.
5. **Feature Enhancements**: A dynamic skills section was added, allowing users to easily add or remove their skills. A profile completion checklist with visual icons shows how complete the profile is. Tailwind's dark mode classes and Font Awesome icons were used to improve the overall design and accessibility..
6. **Error Handling**: I solved issues where ApplicationController, UserService, and SecurityConfig were depending on each other in a way that caused errors. I fixed this by reorganizing and cleaning up the code. I also fixed database problems, like a missing default value for is_profile_public, by setting proper default values in both the code and the database settings.
7. **Testing and Debugging**: Tested profile editing, skills management, and dark mode functionality. Verified database integrity with MySQL Workbench and resolved UI issues using browser developer tools.

5 Conclusion

The Job Portal System is a secure, user-friendly, and attractive platform that helps connect job seekers with employers. It includes useful features like skill management, profile completion tips, and a dark mode to improve user experience. Technical problems were solved by careful planning and improving the code. The role-based system makes it easy for users to navigate while keeping the platform secure. In the future, the system could be improved with features like job suggestions, better search options, instant alerts, and more tools to make it even more helpful and competitive.

Links:

Github: https://github.com/SurajGiri55446/job-portal

Live: https://job-portal-9htz.onrender.com/

Test Credentials :

Employer: employer1@example.com / **password:** password123

Applicant: applicant1@example.com / **password:** password123