

Vacancy Web application

FULLSTACK DEVELOPER FINAL ASSIGNMENT

Accountability document

Attemp-3

Student name: Mustafa Dogan

3/7/2025

Vacancy Web is a Java spring-boot react web application that allows the system owner to post the vacancy and let people apply for the position.

Table of contents

Table of contents	2
List of diagrams	3
List of Table.....	3
1 Introduction.....	4
2 Scope	4
3 Personal technical choice	4
3.1 Programming Language: Java	4
3.2 Front-end Framework: React	4
3.3 Database: PostegrsSQL	4
3.4 IDE: JetBrains IntelliJ IDEA.....	5
3.5 Version Control System: Git	5
3.6 Authentication and Authorization: JSON Web Tokens (JWT)	5
3.7 Build Management: Maven	5
3.8 Server-side JavaScript runtime: Node.js	5
3.9 Java Framework: Spring Boot	5
3.10 Modeling Language: Unified Modeling Language (UML)	6
3.11 MVC pattern	6
3.12 React JS.....	6
3.13 RESTful APIs (Representational State Transfer APIs).....	6
3.14 ORM (Object-Relational Mapping) tool	6
3.15 PosgresSQL Database.....	7
3.16 Authorization and authentication using Spring Security	7
3.17 React hooks	7

3.18 Centralized exception handler 7

3.19 DTO (Data Transfer Object) 8

3.20 JWT Request filtering 8

3.21 JAVA Spring Boot 8

4 Limitations 8

5 Future enhancements 9

6 Project link backend and frontend project on GitHub..... 9

7 Conclusion 9

List of diagrams

No table of figures entries found.

List of Table

No table of figures entries found.

1 Introduction

Today vacancy web apps are one of the essential online tools where employees can register, post and search jobs (openings). It basically helps job seekers to find and apply for relevant jobs. Vacancy based web applications can provide an easy platform to search available job positions based on interest and skills, enabling them to submit their resume directly through our website. Here I have developed a vacancy Web application using Java spring-boot react.

2 Scope

This accountability document describes the technical choices made when developing our application. Our application is designed to register employees, let them respond to vacancies and upload resumes.

3 Personal technical choice

These are personal technical choices made during the project development. The below technical choices were made to optimize and improve the development of our application. These choices provide a stable, scalable and secure environment for our users.

3.1 Programming Language: Java

Java was chosen as the primary programming language because of its robustness, flexibility and cross-platform support. It is also a widely used language for business applications, making it easier to find developers who are familiar with the language.

3.2 Front-end Framework: React

React was chosen as the front-end framework because of its efficiency and flexibility. React allows us to quickly and easily develop interactive and responsive user interfaces.

3.3 Database: PostgreSQL

PostgreSQL was chosen as the database because of its stability, scalability and ease of use. PostgreSQL is a widely used database and offers a rich set of data management features.

3.4 IDE: JetBrains IntelliJ IDEA

JetBrains IntelliJ IDEA was chosen as the integrated development environment (IDE) because of its ease of use and extensive functionality. IntelliJ IDEA provides a fast and efficient way to write, debug, and test Java and React code. It also supports several Java frameworks, including Spring Boot, making it easy to develop and deploy Java applications. IntelliJ IDEA is known for its user-friendly interface and many productivity-enhancing features, making it an ideal choice for our Java and React development.

3.5 Version Control System: Git

Git was chosen as the version control system because of its ease of use, flexibility, and widespread support in the software development community. Git allows us to collaborate effectively on the development of the application and track the version history.

3.6 Authentication and Authorization: JSON Web Tokens (JWT)

JWT was chosen as the authentication and authorization technology because of its efficiency and security. JWT allows us to securely establish and verify the identity of users, without the need for repeated authentication.

3.7 Build Management: Maven

Maven was chosen as the build management tool because of its ease of use and extensive functionality. Maven allows us to build, test, and manage our application in an efficient and standard way. Maven makes it easy to manage dependencies and configure automatic builds, making our development process more efficient.

3.8 Server-side JavaScript runtime: Node.js

Node.js was chosen as the server-side JavaScript runtime because of its speed and efficiency. Node.js uses an asynchronous and event-driven architecture, which allows it to handle large volumes of requests without sacrificing performance. This makes Node.js suitable for real-time applications and developing fast and scalable applications.

3.9 Java Framework: Spring Boot

Spring Boot was chosen as the Java framework because of its speed and flexibility. Spring Boot provides a simplified way to develop and deploy Java applications. It offers standard configuration, automatic configuration, and simplified startup procedures, which reduces

development time and effort. In addition, Spring Boot has extensive support for various technologies, making it easy to achieve integrations with other systems and tools.

3.10 Modeling Language: Unified Modeling Language (UML)

UML was chosen as a modeling language because of its versatility and worldwide recognition. UML allows us to visually model the architecture, components and processes of our application. This makes it easier to better understand our application, document it and collaborate with other developers and stakeholders. Furthermore, UML is an industry standard, which means that many developers and organizations are familiar with this modeling language.

3.11 MVC pattern

I found it is a more efficient way to develop because It separates entire project logic into three important units (Model, Controller and View) that ensure modular structure in terms of scalability (by adding or updating features) without disturbing the entire code. Additionally it is easier to maintain code, support RESTful APIs, integrate frontend and backend, testing and scalability.

3.12 React JS

Today React is an excellent choice for modern, scalable and dynamic web site development. It provides faster performance using DOM (Document Object Model) to update and render. I can re-use components very easily in multiple interfaces. Another thing I found is that it simplifies website development by decreasing duplication

3.13 RESTful APIs (Representational State Transfer APIs)

It makes excellent communication between frontend and backend and also supports multiple platforms (i.e. mobile, IoT device, web) without modifications. Especially I found it to be lightweight and increase our website performance. This will especially help deal with the high volume of data.

3.14 ORM (Object-Relational Mapping) tool

It is more beneficial because of these reasons

- This helps me to simplify database operation, I can utilize save(), findById() and delete() methods for database operation instead of writing SQL queries.

- I can easily define object oriented principles like one-to-many, many-to-many, generic value etc... using java classes.
- It is not required to manage code (such as connection, mapping and data transactions), ORM automatically handles it, which increases my project development speed.
- It was easier to perform INSERT, UPDATE and DELETE operation without writing SQL queries.
- It automatically ensured transaction management like rollback if errors, consistency, unique value etc...

3.15 PostgreSQL Database

I used PostgreSQL database because it is open-source and available at free of cost so further it can help to decrease website deployment cost. It is compatible with the Spring Boot framework. It supports every modern database feature (i.e. primary key, foreign key, check constraint, unique constraints, types etc)

3.16 Authorization and authentication using Spring Security

Project utilized JWT authentication and authorization for user authentication because it is a more secure, scalable and stateless way to manage user access. This authentication is more convenient for modern applications which need API integration, multiple platform support and seamless communication between services.

3.17 React hooks

It helps to improve vacancy web app frontend especially to develop dynamic, modern and responsive interface through useReducer and useState. It enables me to fetch and manage data seamlessly from the RESTful API. Project handles API Calls, changes UI State dynamically and also helps to manage side effects. Here it is also used to simplify frontend and develop clear code.

3.18 Centralized exception handler

A Centralized Exception Handler is usually measured better than a normal exception handler in Java Spring Boot web application because it ensures every exception is handled in a consistent manner and provides uniform error responses. Here the project defines an exception handler applied on the entire project, there is no need to write try-catch blocks

repeatedly. This makes code more readable, decreases duplication, is easy to maintain, and keeps error handling business logic separate. Thus it helps to enhance development experience and make debugging easy.

3.19 DTO (Data Transfer Object)

The project used DTO to transfer information between controller and service layer, it helps to optimize data through encapsulating only necessary fields instead of entire entity objects. This ensures controllers receive only required information and avoid data overload APIs. It also prevents exposing database sensitive information such as fields, database structure.

3.20 JWT Request filtering

It is utilized to intercept incoming user requests to validate and verify JWTs authorization purpose. This is utilized as part of security configuration to make sure only authenticated users can access vacancy web application resources. It validates the token on every user request and enables applications to scale horizontally without worrying about shared session state. This prevent unauthorized access through validating signature (i.e. ensure token is not expired, tampered or created by unauthorized source)

3.21 JAVA Spring Boot

Vacancy web application developed using java spring boot because it is fast, reliable and developer friendly. Additionally it provide numerous features such as 1) automatically configuration and setup dependent components (i.e. Tomcate) and we can focus on project logic, 2) Pre-configured library reduce setup time for general tasks and increase overall development speed, 3) it support flexible software architecture that is MVC pattern, 4) It comes with embedded server, it makes easy test, deploy and portability, 5) it provide higher security integration such as OAuth2, JWT.

4 Limitations

Given below are known limitations in our current vacancy web portal.

- Currently developed website with a very basic interface design, it doesn't have more pictures, content, graphics, and visual appeal.

- Website focuses on generic jobs, if it doesn't categorise jobs in specific sectors, industries, or freelance.
- Websites don't filter or check fake or misleading job posts further; it can lead to scams. This can waste candidate time.
- Due to insufficient time and resources, the project was tested with few users and requests.

5 Future enhancements

Given below are some updates that can help to improve user experience.

- ✓ Further requires updating frontend design with more visual appeal to enhance user interest.
- ✓ System should automatically take the database back at a specific date and time.
- ✓ Develop mobile applications (iOS and Android platform) with push notification for certain alerts, job status update and recruitment messages.
- ✓ Provide more specific results on the dashboard using AI (Artificial Intelligence) tools.
- ✓ Implement features to conduct an interview through video conference with the selected candidates.
- ✓ Support multiple languages to increase web portal usage globally.

6 Project link backend and frontend project on GitHub

Github Project link: <https://github.com/dd0gan/vacancy/>

I submit for second chance after reviewing feedback, so I have also submitted 'Template second chance feedback' into submission zip.

7 Conclusion

The vacancy web application developed using Java Spring Boot and React JS Successfully demonstrates a robust, user friendly, scalable application design to bridge the gap between

job employers and job seekers. Through integrating the latest backend technology with responsive front-end (React) technology, this web application offers an efficient platform for managing job posting, users and their interaction.

In conclusion the given project serves as a proof of concept of building full-stack web applications with modern technologies. It can be further extended with additional features to cater industry specific needs, creating valuable tools for employment service and recruitment organizations.

