PLACEMENT MANAGEMENT SYSTEM

A project report submitted in partial fulfilment of the requirement for the degree

submitted by

SHAIK MISBA (R170607)

Under the esteemed guidance of

B LINGAMURTHY
Software Engineer

Department of Computer Science and Engineering

Rajiv Gandhi University of Knowledge and Technologies(RGUKT),

R.K.Valley, Kadapa, Andra Pradesh.

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES, (AP Government Act 18 of 2008)
IIIT RK VALLEY, RGUKT-AP,Vempalli, Kadapa,
Andhrapradesh-516330.

CERTIFICATE FOR EXAMINATION

This is to certify that I have examined the thesis entitled **Placement Management System** submitted by **S.Misba(R170607)** hereby accord my approval of it as a study carried out and presented in a manner required for its acceptance in partial fulfilment for the award of Bachelor of Technology degree for which it has been submitted. This approval does not necessarily endorse or accept every statement made, opinion expressed or conclusions drawn, as recorded in this thesis. It only signifies the acceptance of this thesis for the purpose for which it has been submitted.

EXAMINER

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES, (AP Government Act 18 of 2008)

IIIT RK VALLEY, RGUKT-AP, Vempalli, Kadapa, Andhrapradesh-516330.

CERTIFICATE FOR PROJECT COMPLETION

This is to certify that I have examined the thesis entitled submitted by **S.MISBA(R170607)** under our guidance and supervision for the partial fulfilment for the degree of **Bachelor of Technology** in Computer Science and Engineering during the academic session September 2022 – April 2023 at AP IIIT RGUKT RK -Valley. To the best of my knowledge, the results embodied in this dissertation work have not been submitted to any university or institute for the award of any degree or diploma.

Project Internal Guide,

B. Lingamurthi sir,

Software Engineer

Head of the department,

Mr. N. Satyanandaram,

HOD of CSE

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES, (AP Government Act 18 of 2008)
IIIT RK VALLEY, RGUKT-AP,Vempalli, Kadapa, Andhrapradesh-516330.

DECLARATION

S.Misba (R170607) hereby declare that the project report entitled "P**LACEMENT MANAGMENT SYSTEM**" done by under the guidance of Mr **B.LINGAMURTHI sir** is submitted in partial fulfilment for the degree of Bachelor of Technology in Computer Science and Engineering during the academic session September 2022 – April 2023 at RGUKT-RK Valley. I also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from any websites are mentioned in the references. To the best of my knowledge, the results embodied in this dissertation work have not been submitted to any university or institute for the award of any degree or diploma.

S.MISBA (R170607)

ACKNOWLEDGEMENT

I would like to express our deep sense of gratitude & respect to all those people behind the screen who guided, inspired and helped us crown all our efforts with success. I wish to express our gratitude to our project guide **Mr.B.Lingamurthi sir** for her valuable guidance at all stages of study, advice, constructive suggestions, supportive attitude and continuous encouragement, without which it would not be possible to complete this project.

We would also like to extend our deepest gratitude & reverence to the HOD of the computer science and engineering **Mr.N. Satyanandaram** RGUKT RK-Valley and the Director of RGUKT, **Prof. K. Sandhyarani** for their constant support and encouragement.

Last but not least I express my gratitude to my parents for their constant source of encouragement and inspiration for me to keep my morals high.

Table of contents

ABSTRACT

The placement management system is a web-based application developed using Spring Boot and HTML. This system is designed to help educational institutions manage their campus recruitment process more efficiently. It provides a platform for students to register and apply for job opportunities, while employers can post job openings and review applicants' profiles. The system is built using Spring Boot, a popular Java framework for developing web applications, which provides features such as dependency injection, data access, and web services. The front-end of the system is developed using HTML and Thymeleaf, a template engine for Java that allows for the easy creation of dynamic web pages. The placement management system features a user-friendly interface that allows students to easily register and upload their profiles, which can be viewed by potential employers.

Placement coordinators can create accounts, post job openings, and review applicants' profiles to find suitable candidates. Overall, the placement management system provides an efficient and streamlined process for educational institutions to manage their campus recruitment, benefiting both students and employers.

PROBLEM STATEMENT

The Placement Management System is designed to address several problems associated with the traditional campus recruitment process. Some of these problems include:

- ✗ Inefficiency: Traditional recruitment processes can be time-consuming and inefficient, with manual paperwork and a lack of centralized communication channels leading to delays and mistakes.
- ✗ Limited Reach: Traditional campus recruitment processes are limited by geography and other factors, making it difficult for employers to access a broad pool of talented candidates.
- ✗ Poor Visibility: Job postings may not reach a wide audience, and students may not be aware of all the available job opportunities.
- ✗ Lack of Coordination: Traditional recruitment processes may lack coordination and standardization, making it difficult to manage the recruitment process and evaluate candidate suitability.
- ✗ High Cost: Traditional recruitment processes can be expensive, with costs associated with advertising, travel, and other expenses.

INTRODUCTION

The Placement Management System is a web-based application designed to assist educational institutions in managing their campus recruitment process effectively. The system provides a platform for students to register, apply for job opportunities, and create profiles while allowing employers to post job openings and review applicants' profiles. The system is built using Spring Boot and HTML, which provides a flexible and scalable platform for developing web applications. The front-end of the system is developed using Thymeleaf, a Java-based template engine that allows for the creation of dynamic web pages. The system features a user-friendly interface that simplifies the process of registering and applying for jobs, while employers can easily post job openings and review applicants' profiles. This report provides an overview of the system's design, implementation, and features, as well as the benefits it provides to educational institutions and students.

EXISTING SYSTEM

The existing system for campus recruitment management can be inefficient and difficult to manage. The existing system often involves manual data entry and management, which can be time-consuming, prone to errors, and difficult to analyse. The lack of automation, integration, and standardization can lead to errors, inconsistencies, and delays in the recruitment process.

PROPOSED SYSTEM

the proposed system for campus recruitment management is designed to be efficient, user-friendly, and scalable. The automation, integration, and standardization provided by the system can lead to improved efficiency, better communication, and increased access to job opportunities for students. The system also provides analytics and reporting tools that can help employers and educational institutions to continuously improve the recruitment process.

PURPOSE

The purpose of the Placement Management System is to streamline and optimize the campus recruitment process. The system provides a centralized platform for managing job postings, student applications, and communication channels between employers and students. By doing so, the system aims to contribute to the growth and development of both students and businesses, benefiting both the economy and society as a whole.

SCOPE

The scope of the Placement Management System project is to deliver a fully functional and user-friendly application that streamlines and optimizes the campus recruitment process. The system will provide a centralized platform for job postings, student applications, and communication channels, enabling employers to access a broad pool of talented candidates and students to find job opportunities more efficiently.

ADVANTAGES

The Placement Management System offers several advantages to educational institutions, employers, and students. Some of the key advantages of the system are as follows:

- Improved Efficiency:
- Increased Reach
- Enhanced Communication
- User-Friendly Interface
- Cost-Effective

DISADVANTAGES

Overall, while the Placement Management System has several advantages, it is important to consider potential disadvantages and risks before implementation. Adequate planning, training, and security measures can help to mitigate these issues and ensure a successful implementation.

- Technical Issues
- Training Requirements
- Security Risks
- Limited Accessibility
- Resistance to Change

REQUIREMENT SPECIFICATION

HARDWARE REQUIREMENT:

Client side:

Ram	512 MB
Hard disk	10 GB
Processor	1.0 GHz

Server side:

Ram	1 GB
Hard disk	20 GB
Processor	2.0 GHz

SOFTWARE REQUIREMENT:

Front end	HTML, CSS, Bootstrap
Server side Language	Sring Boot and Apache maven
Database Server	H2
Web Browser	Firefox , Google Chrome or any compatible browser
Operating System	Ubuntu, Windows or any equivalent OS
Software	Xampp

SOFTWARE TOOLS:

Springboot:

Spring Boot is an open-source Java-based framework used for creating stand-alone, production-grade applications quickly and easily. It is built on top of the popular Spring framework and provides a streamlined development experience, allowing developers to focus on writing business logic instead of boilerplate code. Spring Boot comes with many built-in features, such as auto-configuration, which automatically configures application components based on the classpath and allows developers to get up and running quickly. Spring Boot also includes embedded servers, which eliminate the need for developers to deploy their applications to an external server. Additionally, it includes support for a wide range of data sources, security protocols, and other integrations.

Thymleaf:

Thymeleaf is a Java-based server-side template engine for web and stand-alone environments, designed for creating dynamic web applications. It is an open-source and highly extensible framework that allows developers to create HTML, XML, and other types of templates using natural template language with standard syntax. Thymeleaf provides an elegant way of building templates that can be rendered on the server-side, making it a popular choice for Java-based web applications. One of its key features is its ability to work seamlessly with Spring Framework, allowing developers to easily integrate Thymeleaf into their Spring-based applications.

MYSQL:

MySQL is a popular open-source relational database management system that can be used in Spring Boot applications. In Spring Boot, MySQL can be used to store and manage data for a wide range of applications, including web applications, microservices, and RESTful APIs. Spring Boot provides built-in support for MySQL through the Spring Data JPA module, which allows developers to easily perform CRUD (create, read, update, delete) operations on MySQL databases using simple Java classes and annotations. Spring Boot also provides automatic configuration of the database connection, reducing the amount of boilerplate code that developers need to write.

DEV-TOOLS:

Spring Boot DevTools is a set of additional tools and utilities that can be added to a Spring Boot project as a dependency. DevTools provides a range of features that can improve the development experience, including automatic restarts, live reloads, and enhanced debugging. The DevTools dependency includes several sub-modules, such as:

- ✗ Automatic Restart:
- ✗ Live Reload
- ✗ Enhanced Debugging
- ✗ Cache Support:

SPRING SECURITY:

Spring Security is a powerful and highly customizable security framework for Java-based web applications. It is designed to provide authentication, authorization, and other security features to Spring-based applications, and is built on top of the Spring Framework. Its extensive range of security features, combined with its flexibility and customizability, make it a popular choice for developers working on Java-based web applications.

Spring Security provides a wide range of security features, including:

- ✗ Authentication
- ✗ Authorization
- ✗ Session Management
- ✗ Cross-Site Request Forgery (CSRF) Protection
- ✗ Security Headers

HTML:

HTML stands for Hypertext Markup Language. It is a standard markup language used to create and structure web pages. HTML uses various tags and attributes to define the structure and content of a web page, such as headings, paragraphs, links, images, and forms. In the above project, HTML is used to create the user interface for the web application. Specifically, HTML is used to create the registration and login forms for students, as well as the various pages and forms used by the admin interface. HTML is used in combination with Thymeleaf, a server-side Java template engine, to dynamically generate the content of the web pages based on data retrieved from the Spring Boot application.

APACHE TOMCATE SERVER:

Apache Tomcat is an open-source web server and servlet container that is widely used for deploying and running Java-based web applications, including Spring Boot applications. It is a lightweight and scalable server that provides a secure and reliable platform for hosting web applications. In the context of Spring Boot, Apache Tomcat is often used as the embedded web server. This means that instead of deploying the application to a separate web server like Apache HTTP Server or Nginx, the web server is packaged and run inside the Spring Boot application itself. This allows for easy deployment and management of the application, as it can be run as a stand-alone executable JAR file.

H2 Database:

H2 database is an open-source relational database management system written in Java. It is a lightweight, fast, and embeddable database that can be used in Java applications. H2 database supports SQL and JDBC standards, and it provides several features such as transactions, multi-version concurrency control, and data types for storing different data formats. One of the main advantages of H2 database is its simplicity and ease of use. It can be easily embedded in Java applications and does not require any installation or configuration. It also provides an in-memory mode that allows developers to create and manipulate databases entirely in memory, which can be useful for testing or prototyping.

DESIGN AND ANALYSIS

Design Introduction:

The design and analysis part of the project involves various steps and considerations, each of which is critical to the success of the project. By carefully designing and implementing each component of the system, and by rigorously testing and deploying the application, developers can create a high-quality and reliable web application that meets the needs of the users.

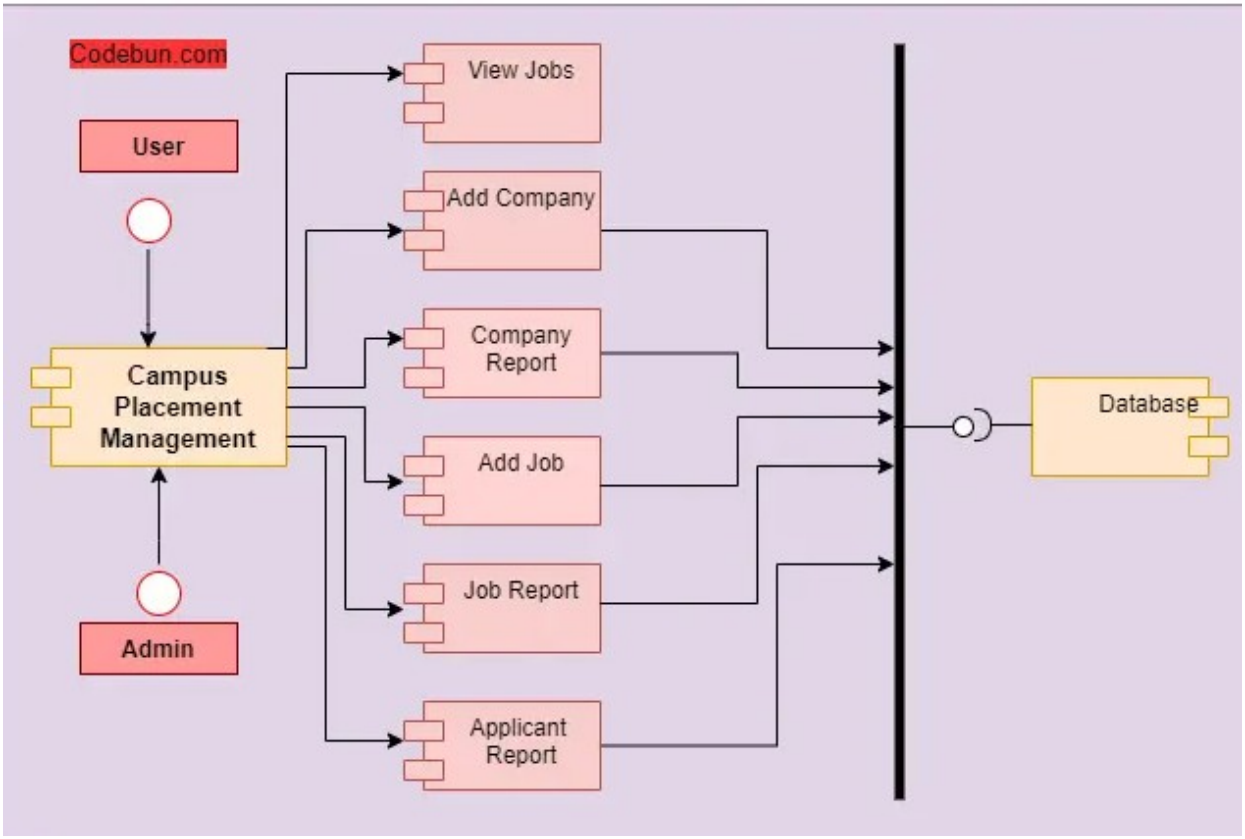
The design and analysis part of the above project involves various steps and considerations, such as:

- ✗ Understanding the requirements
- ✗ Creating the system architecture
- ✗ Selecting the technology stack
- ✗ Creating the database schema
- ✗ Designing the user interface
- ✗ Testing and deployment

Component Diagram:

The component diagram would show the relationships and dependencies between these components, such as the dependency between the presentation layer and the controller layer, the dependency between the controller layer and the service layer, and the dependency between the service layer and the data access layer. It would also show the interfaces that the components expose to each other and the external services they consume. Overall, the component diagram would provide a high-level view of the system architecture and the components involved in its operation. A component diagram for a placement management system using Spring Boot and HTML would illustrate the high-level architecture and structure of the system's components and how they interact with each other. The main components in such a system might include:

- ✗ Presentation Layer
- ✗ Controller Layer
- ✗ Service Layer
- ✗ Data Access Layer
- ✗ Security Layer
- ✗ External Services



Use Case Diagram:

A use case diagram for a placement management system using Spring Boot and HTML would illustrate the various actors that interact with the system and the use cases that describe their interactions. The main actors in such a system might include:

1. Student
2.Admin

Some of the use cases that these actors might perform in the system include:

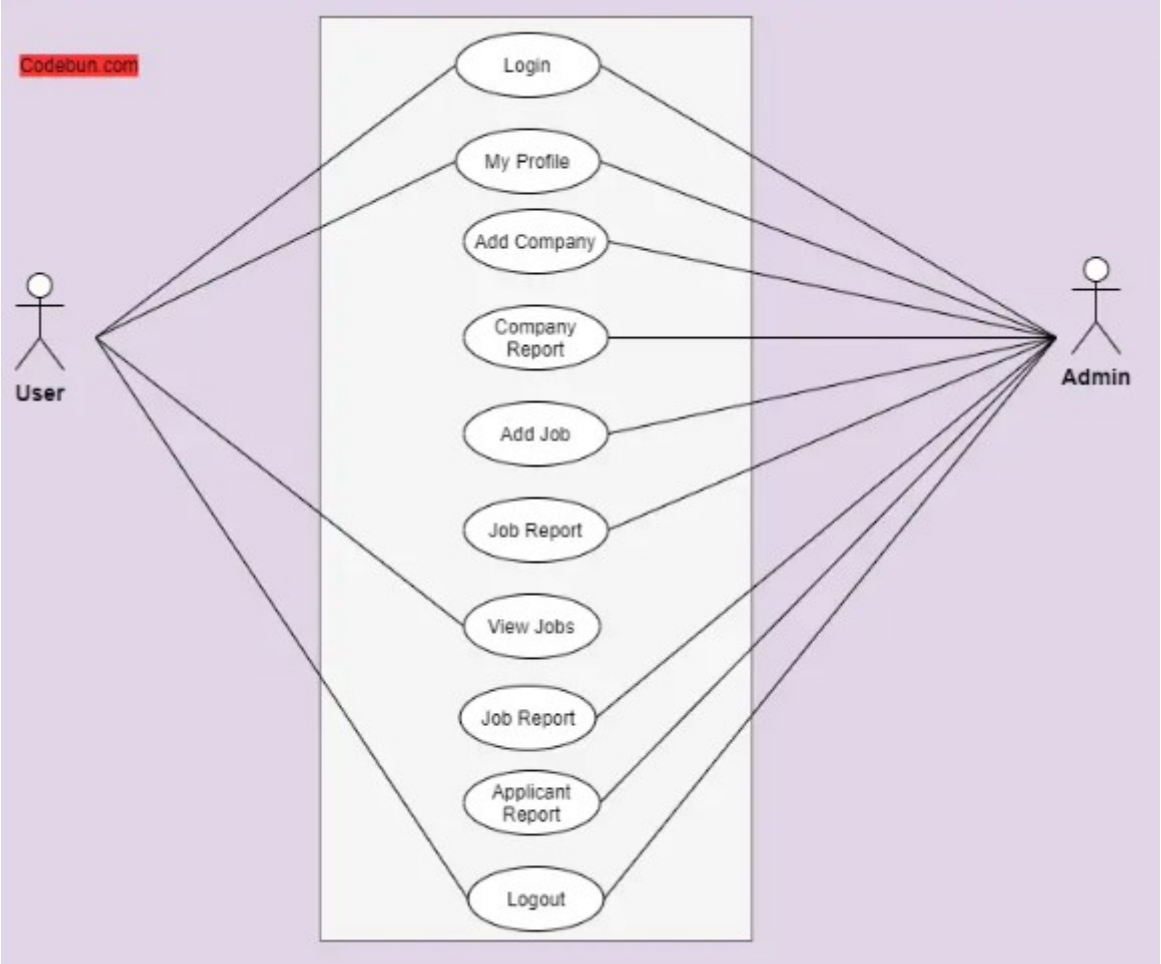
1. Student:

- ✗ Register
- ✗ Login
- ✗ Search Jobs
- ✗ Apply for Job

2.Admin:

- ✗ Approve Job Postings
- ✗ Manage Student Registrations
- ✗ Manage Company Registrations
- ✗ View Job Applications

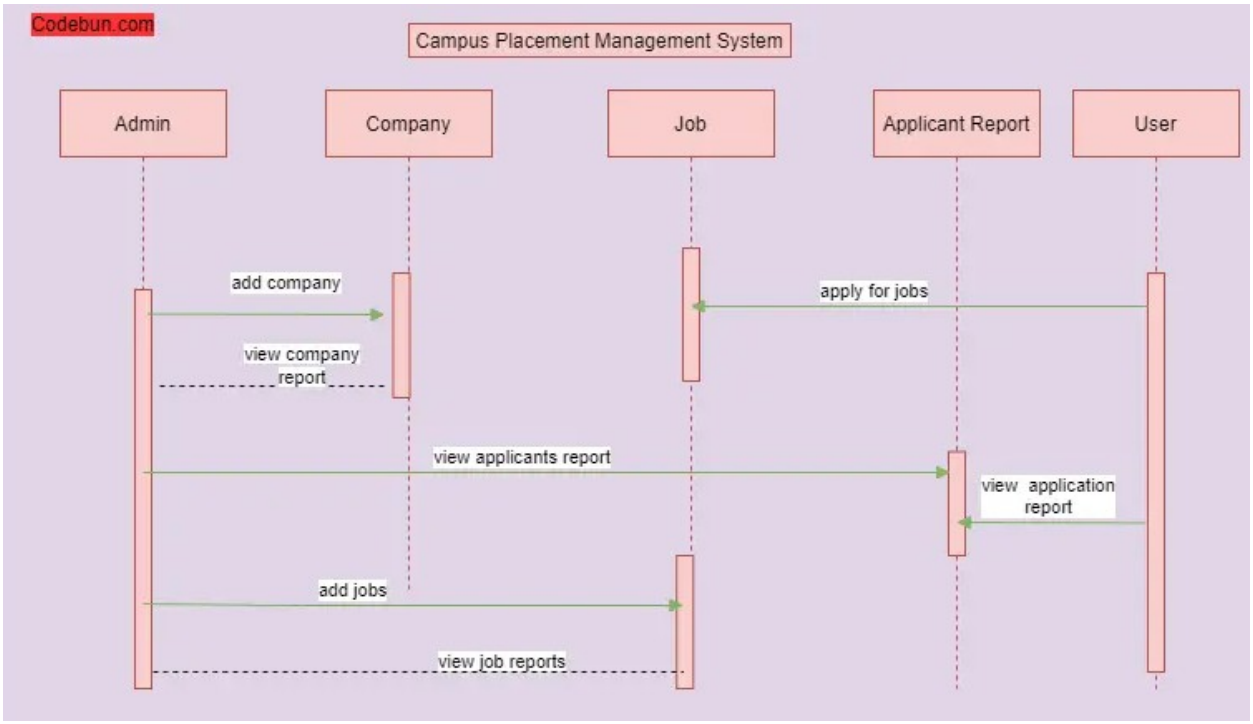
The use case diagram would show the relationships and interactions between these actors and use cases, such as the association between the student actor and the apply for job use case or the association between the company actor and the post job use case. It would also show the generalization relationships between actors, such as the generalization of the student and company actors from the user actor. Overall, the use case diagram would provide a high-level view of the system's functionality and the different actors involved in its operation.



Sequence Diagram:

A sequence diagram for a placement management system using Spring Boot and HTML would illustrate the interactions between different actors and components in the system during a particular scenario or use case. For example, the following sequence diagram shows the interactions between a student, the login controller, and the database when the student logs in to the system.

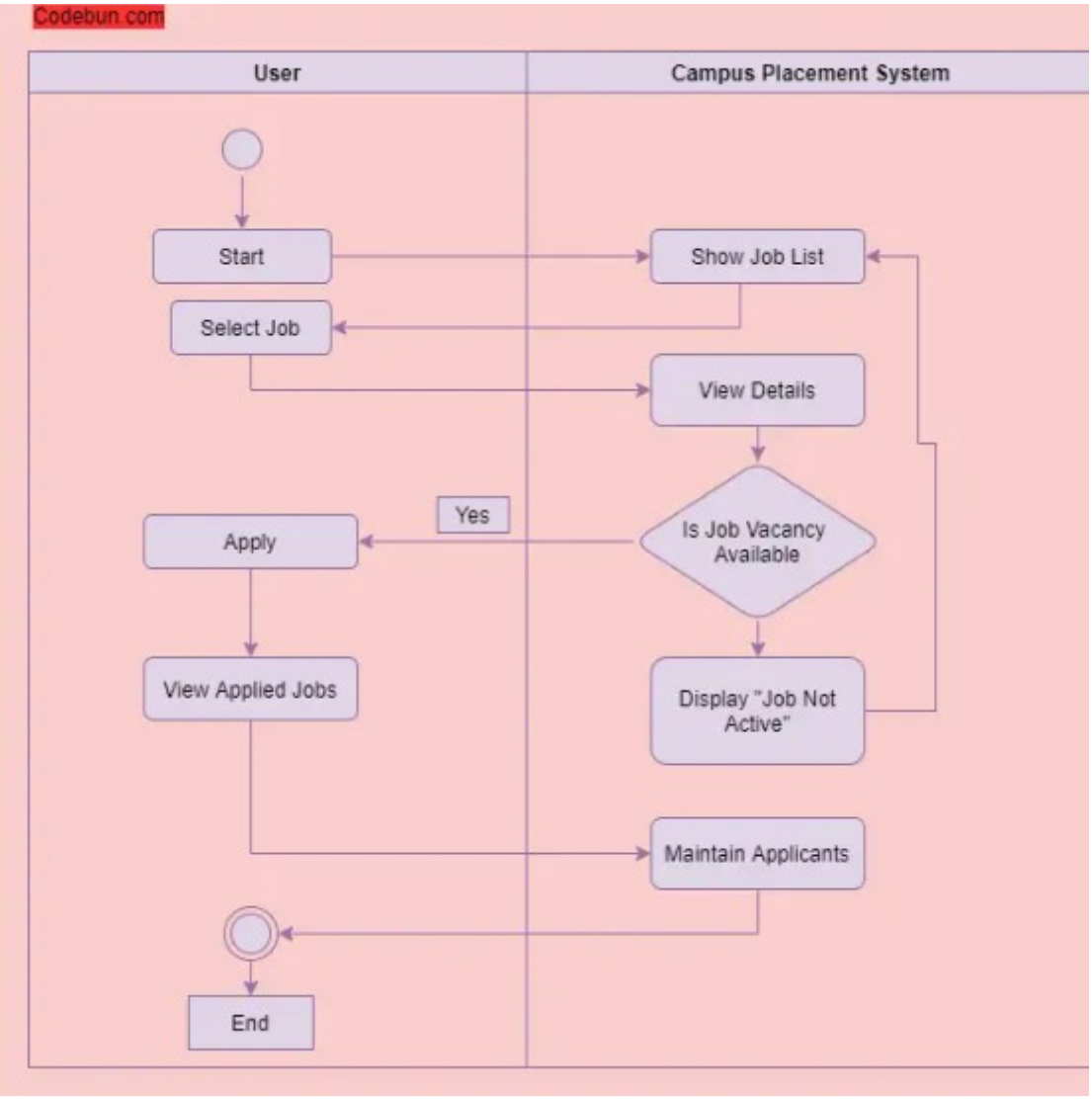
In this sequence diagram, the student enters their login credentials, which are then passed to the login controller. The login controller verifies the credentials by querying the database. If the credentials are valid, the database returns a confirmation to the login controller, which then redirects the student to the home page. If the credentials are invalid, the login controller may return an error message to the student.



Activity Diagram:

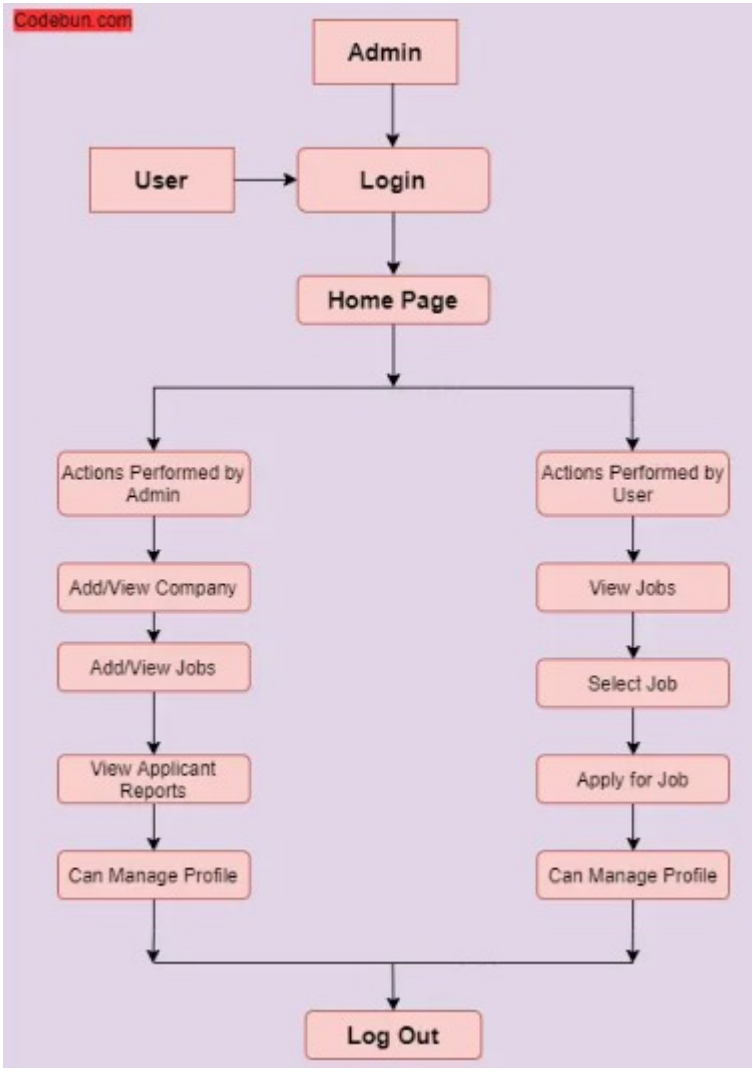
An activity diagram for a placement management system using Spring Boot and HTML would illustrate the flow of activities and actions involved in a particular use case or scenario. For example, the following activity diagram shows the steps involved in a student registering for the system. In this activity diagram, the student begins by entering their registration details into a form. They then submit the form to the system, which checks whether the form is valid. If the form is valid, the system saves the student details to the database and redirects the student to the login page. If the form is not valid, the system displays an error message to the student.

Other activity diagrams might illustrate the flow of activities involved in a student logging in to the system, a company posting a job opening, or an admin approving a job posting. In each case, the activity diagram would show the steps involved in the process, highlighting the decisions and actions that need to be taken by the relevant actors and components.



Data Flow Diagram:

Assuming you meant "data flow diagram", a data flow diagram (DFD) for a placement management system using Spring Boot and HTML would illustrate the flow of data between different components and actors in the system. Here is an example of a simple DFD for a student registering for the system:



In this DFD, the student inputs their registration details into a form, which is then transmitted to the system. The system stores the student details in a database, which can then be used for various purposes, such as authentication or matching with job postings.

Other DFDs might illustrate the flow of data involved in a student logging in to the system, a company posting a job opening, or an admin approving a job posting. In each case, the DFD would show the flow of data between the relevant components and actors, highlighting the dependencies and interactions between different parts of the system.

IMPLEMENTATION ANS SYSTEM DESIGN

The implementation and system testing of a placement management system using Spring Boot and HTML involves several key steps:

1. **Develop the code**: The first step in implementing the system is to develop the code. This involves writing the software that implements the different components and subsystems, as well as integrating them together into a cohesive whole.

2. **Unit testing**: Once the code has been written, the next step is to perform unit testing. This involves testing individual components and subsystems in isolation to ensure that they function correctly.

3. **Integration testing:** After unit testing, the next step is to perform integration testing. This involves testing the system as a whole, to ensure that all the components and subsystems work together correctly.

4. **System testing:** The next step is to perform system testing. This involves testing the system from the perspective of the end user, to ensure that it meets the functional and non-functional requirements.

5. **User acceptance testing:** Once system testing is complete, the next step is to perform user acceptance testing. This involves having actual users test the system to ensure that it meets their needs and is easy to use.

6. **Performance testing:** In addition to functional testing, it's important to perform performance testing. This involves testing the system under different load conditions to ensure that it can handle the expected number of users and transactions.

7. **Security testing:** Finally, it's important to perform security testing. This involves testing the system for vulnerabilities and ensuring that appropriate security measures are in place to protect user data.

By following these steps, you can ensure that your placement management system is implemented correctly and thoroughly tested, resulting in a high-quality, reliable system that meets the needs of your users.

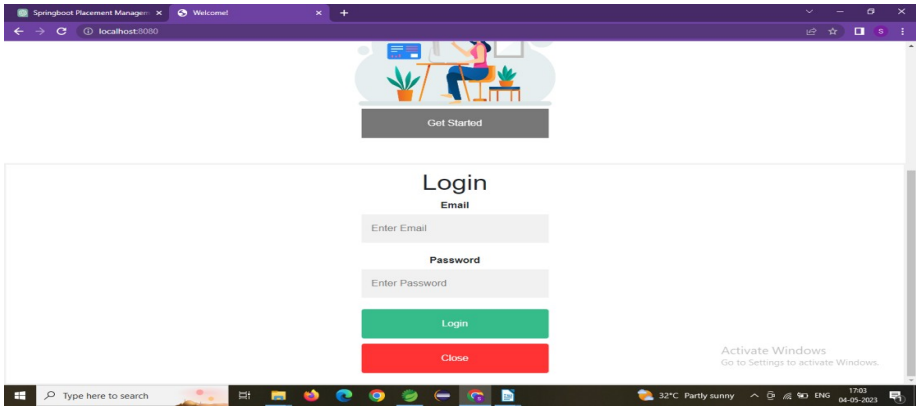
EVALUATION

Project URL:http://localhost:8080/login

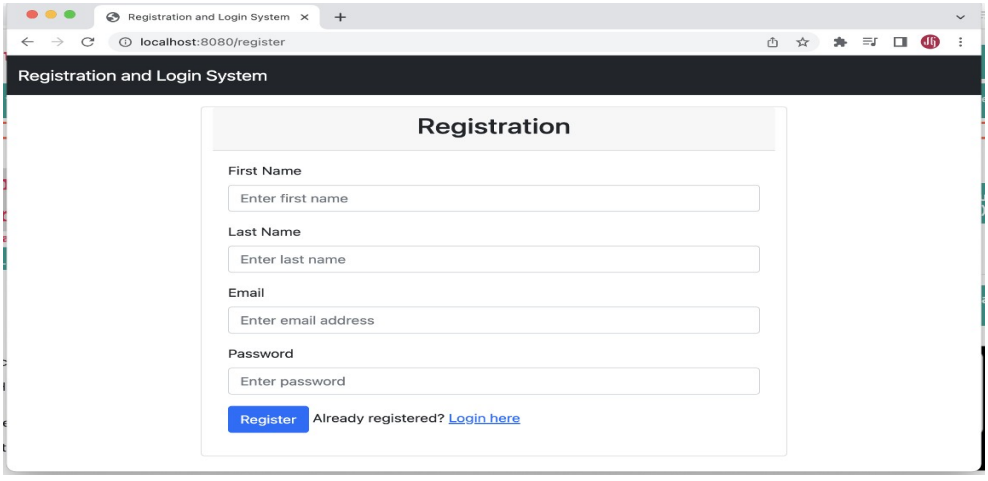
The evaluation of a placement management system using Spring Boot and HTML involves assessing the system's performance and functionality against the requirements and goals set out in the project scope. This can be done through various means, such as

1. User feedback
2. Performance metrics
3. Compliance testing
4. Security testing
5. Maintenance and suppor

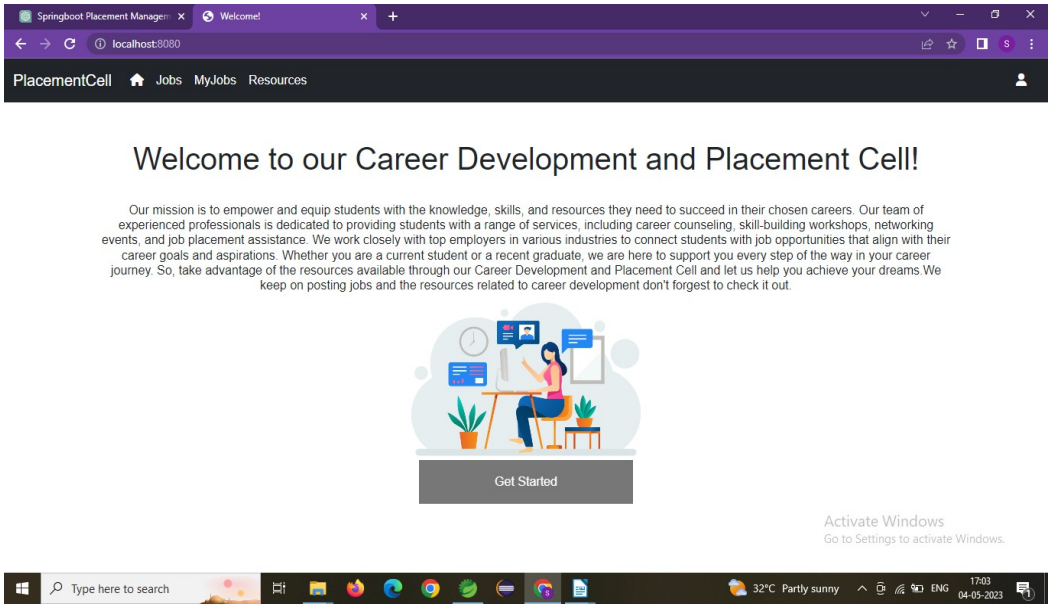
Login Page:



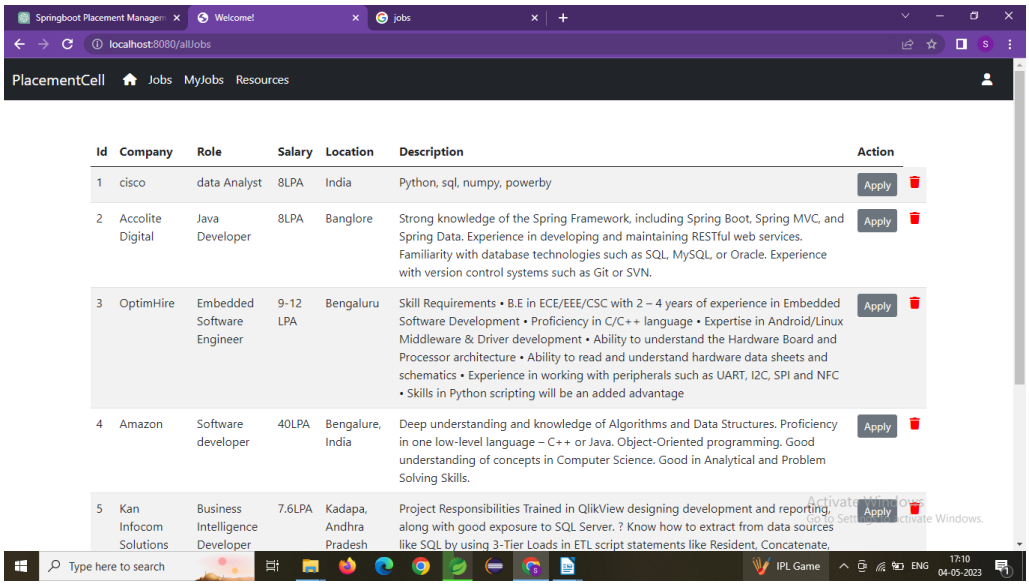
Registration Page:



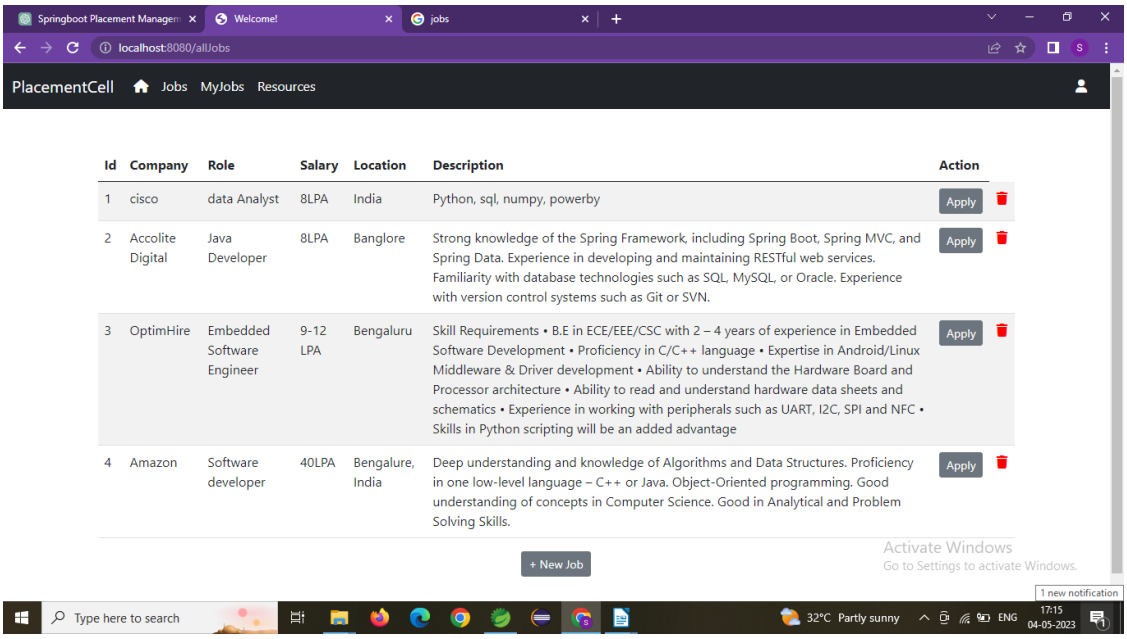
Dash Board:



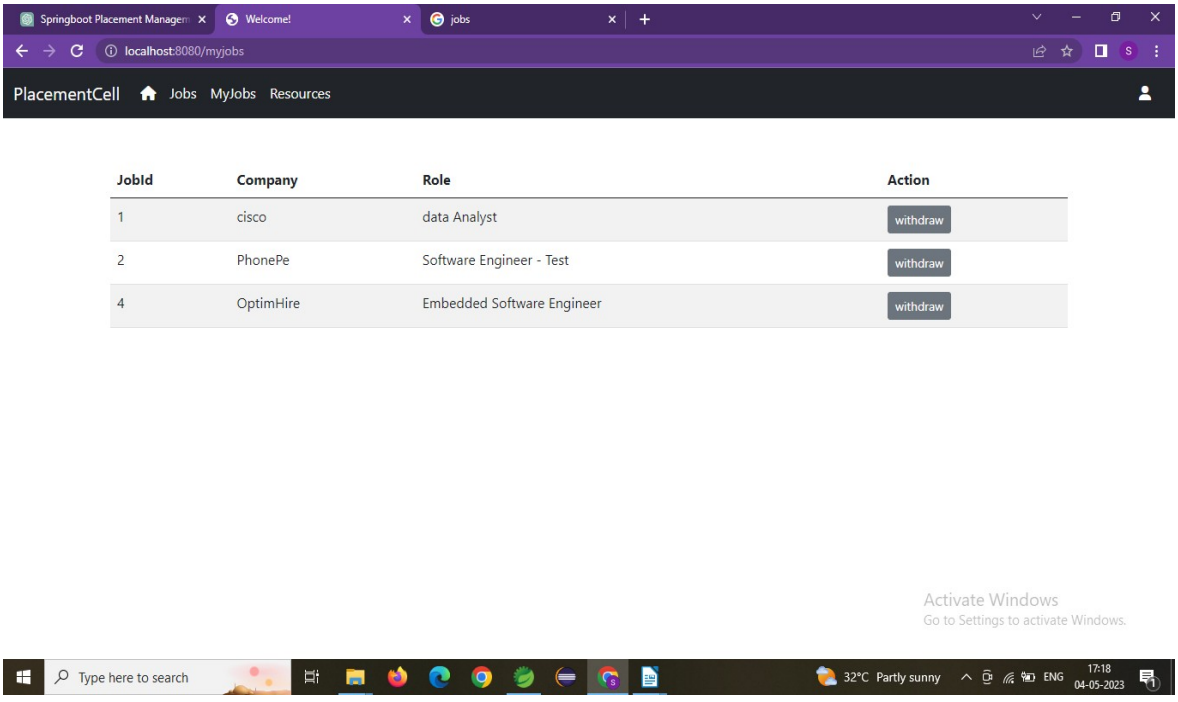
Jobs List:



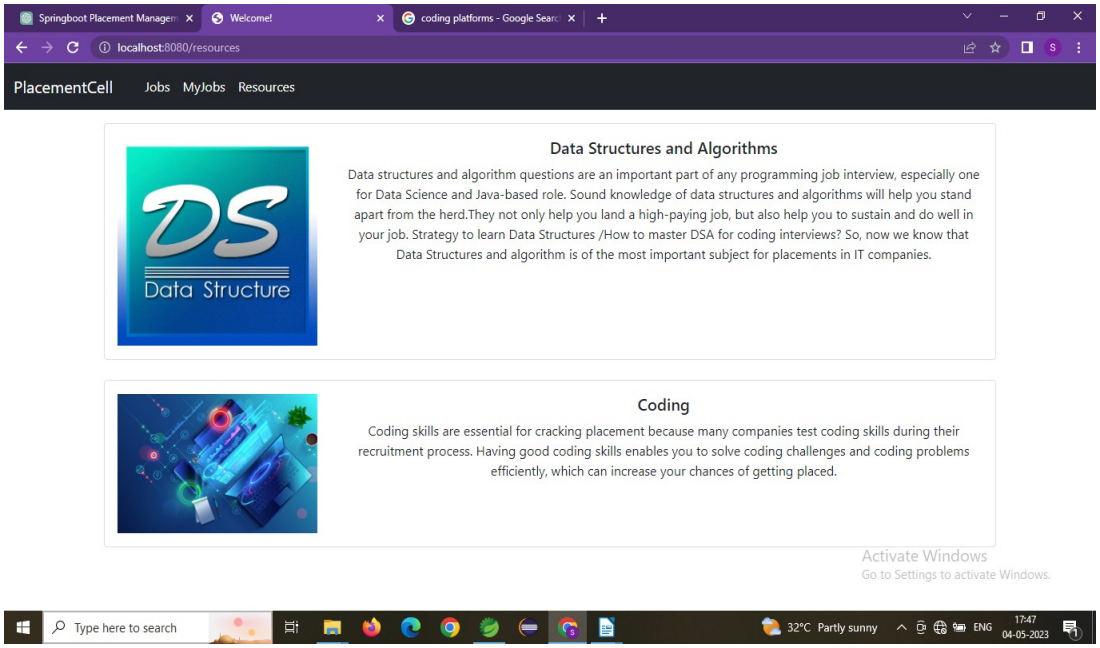
Add Jobs:



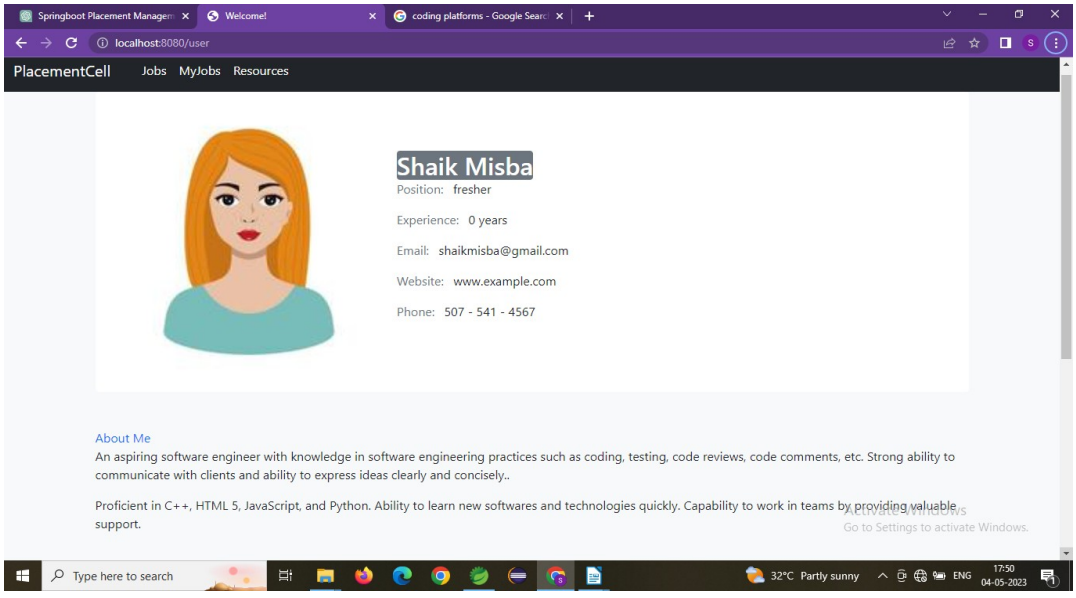
Student Applied Jobs:



Resources:



User profile:



CONCLUSION

The placement management system using Spring Boot and HTML is a comprehensive solution for managing the placement process of an educational institution. It provides features for students to register, upload their resumes, and view job opportunities posted by the recruiters.

The recruiters can also register, post job opportunities, and view the resumes of the students.The project makes use of various technologies such as Spring Boot, Thymeleaf, MySQL, Apache Tomcat, Spring Security, and H2 Database. The implementation of the project involved the analysis and design phase, followed by the implementation and testing phase.

The project has several advantages, including a user-friendly interface, efficient management of placement activities, secure login system, and automated resume screening. However, it also has some limitations, such as limited scalability, dependency on external libraries, and potential data security issues.

Overall, the placement management system can be a valuable tool for educational institutions to streamline their placement processes and provide better opportunities for students.

REFERENCES:

Here are some references that you can use for this project: Spring Boot documentation: https://spring.io/projects/spring-boot

- Thymeleaf documentation: https://www.thymeleaf.org/documentation.html
- MySQL documentation: https://dev.mysql.com/doc/
- Apache Tomcat documentation: https://tomcat.apache.org/tomcat-9.0-doc/index.html
- Spring Security documentation: https://spring.io/projects/spring-security
- H2 Database documentation: https://www.h2database.com/html/main.html
- Object-Oriented Analysis and Design Using UML: https://www.amazon.com/Object-Oriented-Analysis-Design-Using-UML/dp/0471324215
- Agile Software Development, Principles, Patterns, and Practices: https://www.amazon.com/Software-Development-Principles-Patterns-Practices/dp/0135974445
- Head First Design Patterns: https://www.amazon.com/Head-First-Design-Patterns-Brain-Friendly/dp/0596007124
- Java Persistence with Hibernate: https://www.amazon.com/Java-Persistence-Hibernate-Christian-Bauer/dp/1932394885

Additionally, you can also refer to online resources such as blogs, video tutorials, and forums to get help and guidance on specific topics related to the project.