Phase-3 Practice Project

Searching for a Specific User and Updating User Information

writeup

Introduction:
The process of searching for a specific user and updating their information is a common requirement in many web applications. This write-up provides an overview of a solution using Spring Boot, a popular Java framework, along with XML configuration, HTML views, Java servlets, and Spring Boot code.

XML Configuration:
The XML configuration file, typically named application-context.xml, sets up the application context and defines necessary beans and configurations. It includes component scanning, view resolvers, and other essential settings for the Spring Boot application.

HTML View:
The HTML view, represented by the search.jsp file, provides a user interface to search for a specific user. It includes a form with an input field for the username and a search button. The form submits a POST request to the /search endpoint when the user clicks the button.

Java Servlet:
The UserController class acts as the servlet for handling user-related operations. It is annotated with @Controller and contains methods to handle different requests. The searchUser() method receives the POST request from the search form, searches for the user by invoking the UserService, and displays the user's information on the "update" view if found. If the user is not found, it returns to the search form with an error message. The updateUser() method handles the POST request for updating the user's information.

UserService:
The UserService interface defines the contract for user-related operations. It typically includes methods such as getUserByUsername() and updateUser() that are implemented by a service class. In a real-world application, the service class would interact with a database or any other data source to perform the actual operations.

Conclusion:
Searching for a specific user and updating their information is a common functionality in web applications. This write-up outlined a solution using Spring Boot along with XML configuration, HTML views, Java servlets, and Spring Boot code. By following the provided example, developers can create a robust and scalable application that enables users to search for and

update user information efficiently. However, it's important to adapt the solution to fit the specific requirements and architectural considerations of the target application.