

1.	Creating Entity Class:
	<ul style="list-style-type: none"> • Create an entity class (e.g., User) to represent the user data. • Define attributes like ID, name, email, etc. • Use annotations to map this class to the database table.
2.	Creating a DAO (Data Access Object):
	<ul style="list-style-type: none"> • Create a DAO class to handle database operations for the User entity. • Implement methods for retrieving, updating, and validating user data.
3.	Creating a Service Class:
	<ul style="list-style-type: none"> • Create a service class that provides a layer of abstraction between the controller and the DAO. • Implement methods for user validation, retrieval, and update.
4.	Creating a JSP Page for User ID Input:
	<ul style="list-style-type: none"> • Create a JSP page (e.g., ".jsp") with a form to input the user ID. • Include form validation for the user ID field. • On submission, make a POST request to the controller.
5.	Creating a Controller for User ID Validation:
	<ul style="list-style-type: none"> • Create a controller (e.g., UserController) to handle the POST request from the .jsp page. • Validate the user ID. If it's invalid, redirect back to the .jsp page with an error message. • If the user ID is valid, retrieve user details from the database using the service layer.
6.	Creating a JSP Page for User Details Editing:
	<ul style="list-style-type: none"> • Create a JSP page to display user details in an editable form. • Populate the form with the user's data obtained from the database. • Include form fields for updating user information.
7.	Creating a Controller for User Details Update:
	<ul style="list-style-type: none"> • Create another controller method in UserController to handle the POST request from the .jsp page. • Retrieve data from the form fields. • Update the user details in the database using the service layer. • Redirect to a confirmation JSP page on successful update.
8.	Creating a Confirmation JSP Page:
	<ul style="list-style-type: none"> • Create a JSP page to display a success message after the user details are updated.
9.	Mapping and Configuration:
	<ul style="list-style-type: none"> • Configure the request mappings in your framework (e.g., Spring MVC) to route requests to the appropriate controllers. • Ensure the JSP views are correctly configured in your application's configuration.