1.Define the User entity with properties like ID, name, and email.

2.Create a UserDao interface with methods for retrieving, saving, updating, and deleting user data.

3.Implement the UserDao interface using Hibernate to perform CRUD operations on the User entity.

4.Create a UserController class with methods to handle HTTP requests related to user data.

5.Configure the necessary Spring and Hibernate configurations for the application.

6.Create a JSP page to display user details, including ID, name, and email.

7.Create a JSP page for editing user data, with fields for name and email.

8.Configure the servlet mapping for the Spring DispatcherServlet in the web deployment descriptor (web.xml).

9.Start the web server and deploy the application.

10.Handle requests to retrieve user data by invoking the getUserById() method from the UserDao implementation.

11.Display the retrieved user data on the user details JSP page.

12.Handle requests to edit user data by displaying the edit form JSP page and pre-populating it with the user's current data.

13.Retrieve the updated user data from the form submission and update the corresponding user record in the database using the updateUser() method from the UserDao implementation.

14.Redirect the user back to the user details page to display the updated data.