NSBT

SPRING DAY 5 & 6 ASSIGHMENT

Spring MVC

1. Create the First Spring MVC application to print “Hello World” on the JSP page. Create “index.jsp” with hyperlink to the Controller and return a JSP Page as response which prints “Hello World”
   Make the application first with XML configuration and then with Java configuration.

2. Change the above application by sending the name of the User as a Query String parameter to the link created in “index.jsp” page. Retrieve this name with the help of @RequestParam annotation and then send it to the display.jsp page for printing. The response must look like as mentioned below:

   Hello <username>

   Example:

   Index.jsp contains the below hyperlink:

   <a href=”sayHello.do?username=Ajay”> Say Hello </a>

   The output which need to be printed on display.jsp is:

   Hello Ajay

3. Change the above application to return the response with the help of ModelAndView object.

4. In the above application, check for the difference between forward and redirect. How will you do the same and what will be the impact. Maintain the difference in a readMe.txt file.

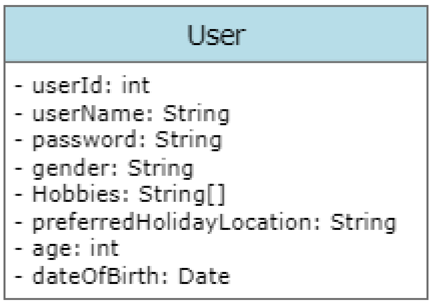
5. In the above application, create another controller to access Request Headers with the help of @RequestHeader annotation. Display different response based on the value of ‘referrer’ request header. If header value is null, it should take the user to error page else to home page.

6. In the same application, create a form to take the details of Customer to register the same. The details which need to be accepted are in the as per the class given in Spring Day – 1 Question – 1. Use POST method to submit the details. Create a Controller to accept the values using @RequestParam annotation, Use the same service and repository layer created in Question-4 of Spring Day -3 assignment and display the inputs taken in the Success.jsp page.

7. In the same form create two Submit buttons with the two values – Add and Update. When user clicks on Add Button – A new customer is registered. When user clicks on Update button – the customer details with the given customer id are updated.

8. In the same application as above, create a controller to fetch the names of all Customers. Display the list of Customer Names with each name acting as a hyperlink. When one hyperlink is clicked, the complete details of that customer are displayed. Demonstrate the use of @PathVariable annotation to create the application.

9. Create an application to create a form for User Registration. The User Bean is as per the class diagram:



   a. Bind the User Bean with the User Registration Form using Spring Tags. The userId and age will not be displayed on the screen. The Screen objects for each attribute is as follows:

Field	Screen Object
Username	Text box
Password	Password
Gender	Radio button
Hobbies	Check box
preferredHolidayLocation	Drop Down
Age	number
DateOfBirth	Date

   b. Use ModelAttribute to retrieve the User details filled in the form, generate userId and calculate age based on the dateOfBirth and display the complete User details on a display.jsp page. If the age is less than 21, the request is forwarded to an error page.

   c. Use propertyEditor for dateOfBirth field.

   d. While displaying the form, the values for Hobbies and preferredHolidayLocation must be prefilled with the dynamic data coming from the server.

10. Create an application to calculate the EMI by providing the required information. The EMI formula can be used from Java – Day 1 assignment. If the details are valid, EMI is calculated otherwise it throws an Exception. Write code to demonstrate the various ways available for Exception Handling.