**Exercise Objective(s):** *Using config parameters*

**Exercise:**

*Create a web application with a servlet that displays user name and password. Validate the user name and password using the information provided in the config parameters . If the user name or password is invalid displays user name and password again till maximum X times where X is specified as context parameter.*

**Solution Guidance (if applicable):** *Implement using Annotation and web.xml.*

**Exercise Objective(s):** *Using config parameters*

**Exercise:**

*Create a Servlet that acts as a filter . This servlet does not allow IPs that are forbidden which are specified as config parameters. If the remote address represents the IP specified in the config parameters, display a forbidden access page. Otherwise, a success access page message is displayed.*

**Solution Guidance (if applicable):** *HttpServletRequest ‘s getRemoteAddr*

**ServletCommunication**

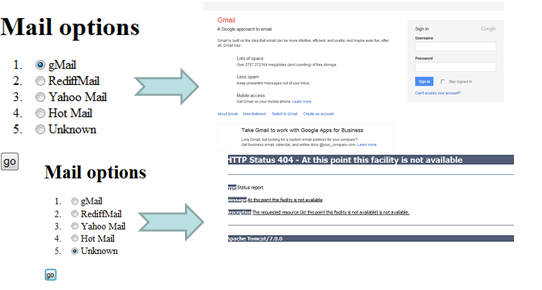
**Exercise Objective(s):** *Redirecting*

**Exercise:**

*Write a web application that does the following:*

* *HTML page displays email options of which one is unknown.*

*If unknown option is an 404 error page is displayed.*



**Exercise Objective(s):** *forward and include*

**Exercise:**

*User can choose to go to flowers page or cards page from the home page and choose the items displayed in the page. From cards (or flowers) page he/she can go to flowers(or cards) page and bill page. In the bill page amount for the items selected in the cards page and flowers page is listed. In case neither flower nor card is selected, then an error message is displayed along with the home page.*

**Exercise Objective(s):** *forward and include*

**Exercise:**

*Add the filter servlet that you created in the last session to the above application. All the request would now pass through this filter servlet. If the remote address represents the IP specified in the config parameters, display a forbidden access page. Otherwise, any page requested by the user is displayed.*

**Exercise Objective(s):** *reading from a passive resource*

**Exercise:**

* *Display the product catalog that is provided in the text file with the prices in a web page using a servlet.*
* *.*

**Connecting to database**

**Exercise Objective(s):** *connecting to database from a web page*

**Exercise:**

*Write a program to create a table called training with the following attributes,*

* *Sap\_ID*
* *Stud\_name*
* *Stream*
* *Percentage*

*Insert all your batch mates’ details and create a web page that displays the details of the table.*

*The config information about the database should be supplied through the web.xml.*

**Exercise Objective(s):** *connecting to database from a web page*

**Exercise:**

*Create a page with the list of energy drinks like boost, horlicks, complain, bournvita, and pediasure available in a shop. Let the user select any number of varieties he wants and provide a text box for each variety to specify the quantity. On completion of the selection instruct the user to press the Submit button. Maintain a database with the following fields like Item name, Quantity available, Price, Expiry date etc. On clicking the submit button, display the details to the user for the items that he has selected to continue shopping.*

**Session Handling**

**Exercise Objective(s):** *Using hidden fields*

**Exercise:**

* *Rewrite the shopping card application written in the previous example using hidden field to maintain state.*

**Exercise Objective(s):** *Using HttpSession*

**Exercise:**

*A table contains 10 yes/no questions. When user takes exam , 3 questions are randomly picked from the table and displayed. Maximum of 3 mins is given for the exam. Each question is displayed in a separate page. If user can use next and prev buttons to navigate. The first question page should have prev button and last question page should not have next button. There should be a finish button in every question page. If user has answered a question, then the user’s answer must be shown when the question page is displayed. When user clicks on the finish button or if the time is beyond 3 mins, the result page with marks must be displayed.*

*Write a servlet based web application to do this.*

**Listeners**

**Exercise Objective(s):** *Using context listeners*

**Exercise:**

* *All the attributes that are added to the context must also be maintained in the database table with the name and value. Similarly they must be removed when the attribute is removed.*

**Exercise Objective(s):** *Using session listeners*

**Exercise:**

* *For the exam web application that you created, if user accidently closes the browser, then he must be able to resume the exam. To provide this facility, when ever any attribute is added or updated in the session, it must be added to the database too. Similarly when ever it is removed, it must be removed. Apart from that status of the exam must also be maintained.*

*On requesting for exam page, if the exam status is incomplete, then the last question that user attempted must be shown. User must be able to resume seamlessly.*

**Introduction to JSP**

**Exercise Objective(s):**  *Form data handling using JSP*

**Exercise:**

*C Create a JSP that displays 2 text boxes. Accept two numbers and print the addition, subtraction, multiplication and division of those numbers.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Exercise Objective(s):**  *Form data handling using JSP*

**Exercise:**

* *Display a JSP registration form that requests the user to enter user’s name, email id, date of birth, address, phone number. On submitting the form, a servlet must validate the data. In case the data is invalid, the JSP page must be displayed again indicating the error fields. This time the registration form must be prefilled with the data that user has entered previously. If the data is valid then*

1. *If a record already exists in the table, ask the user if the records have to be updated. If user click yes, the display the JSP registration page with successful updation after updating the database. If user clicks no, then fetch the record from the database and display the record in the JSP registration page.*
2. *If the record does not exist then insert the data and show the page with the relevant data and successful insertion message.*

*The database parameters must be in the web.config*

**Exercise Objective(s):**  *Handling error*

**Exercise:**

* *In the previous exercise, in case an error occurs, on account of a non-number being entered or if an attempt to divide by 0 occurs the page should automatically be redirected to error page.*

**Exercise Objective(s):**  *using multiple JSP*

**Exercise:**

* *There are 2 pages- Images and Cards. The user can select a card style (border thickness, border color, font size and color) and add personal message in the Cards page. He /She then can select image from the Images page. Finally on clicking “Preview” a page with the image inside the boundary as specified by the card style and message in the font and color specified is displayed. One final submit, email details of the user to whom this has to be sent is prompted. On clicking “Send” , “Sent successfully” page is displayed.*

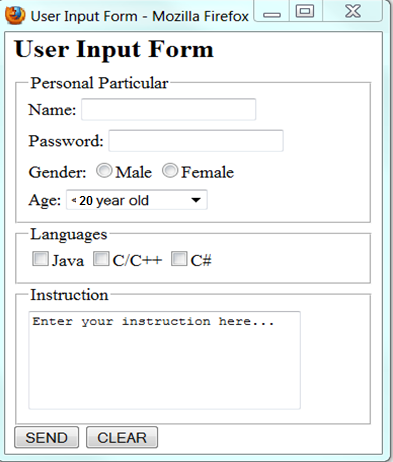
**JSP Action tags/Standard Actions**

**Exercise Objective(s):**  *using standard actions*

**Exercise:**

*Create a form as shown.*

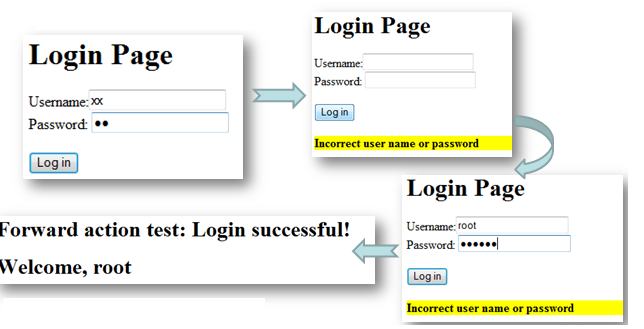
*Create a bean for the input form and when “Send” button is clicked, it should display the user values in a table in the next page*



**Exercise Objective(s):**  *using standard actions*

**Exercise:**

* *User enters login and password which is validated. If invalid values are provided, login page is displayed with error message, otherwise welcome page is displayed.*



**Exercise Objective(s):**  *using JSP scriplets, standard actions, directives*

**Exercise:**

* *Stamp collectors can use this site to post details of stamp that they would like to auction. Interested collectors can bid to purchase the stamp. Each item should be available for 3 days. At the end of three days person whose bid amount is highest, gets the stamp.*

**Custom Tags**

**Exercise Objective(s):**  *using simple custom tag without body*

**Exercise:**

* *Write a custom tag called Hello Tag which will print "Hello! Welcome to the world of custom tags" wherever <custom:hello/> would be used.*
* *What happens if the tag with invoked with a body?*