**ADVANCE JAVA ASSINGNMENT**

**JDBC**

**01.** Create a program to create table named student having columns [ rno, name, marks, date of birth].

**02.** Create a program to insert record in the student table.

**03.** Create a program to select all of the student records.

**04.** Create a program to display the student records in sorted order of marks [ highest to lowest ].

**05.** Create a program to display no. of student records present in the table.

**06.** Create a program to perform CRUD operations for student [ rno, name, marks, date of birth ] using Statement. Take values from keyboard.

**07.** Create a program to search student record on the basis of given roll no.

**08.** Create a program to display average marks obtained by the student.

**09.** Create a program to perform CRUD operation for student [ rno,name,marks,date of birth ] using PreparedStatement. Take values from keyboard.

**10.** Make a program that displays number of columns present in the given table.

**11.** Create a program to call a stored procedure, created to insert student record [ rno, name, marks].

**12.** There is need to store employee information [like : employee id, name, salary, join date] and employee's address information [like : address id, city, country, employee id]. Employee and Address information should be stored in separate tables. Create a program which can insert employee and address records. First store employee record and then address record. Make sure, if the address record is not saved then employee transaction should be rollbacked.

**13.** Select the employee records with their address.

**14.** Select the address of an employee whose employee id is given.

**15.** Select all the employees who are in the given city.

**16.** Select the employee who gets highest salary.

**17.** Select all the employees who has experience more than 5 years.

**18.** Create an event driven program as follows to manage the daily expenses

**a.** create a table expense with columns expense\_id, item, price, purchase\_date to store expense record.

**b.** create a java bean class named dto.Expense with properties expenseId, item, price, purchaseDate.

**c.** create a dao interface dao.ExpenseDao that has method declaration for insert, update, delete and select the expense record.

**d.** create a dao class dao.ExpenseDaoImple that implements ExpenseDao and overrides all methods.

**e.** create p01.Main class to demonstrate CRUD operation written in DAO class.

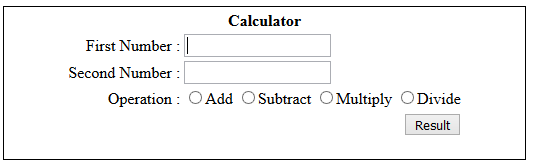
**SERVLET ASSIGNMENT**

**19.** Create a servlet that prints WELCOME TO SERVLET WORLD using html's heading tag.

**20.** Create a servlet that displays system date in the dd/mm/yyyy format.

**21.** Create a servlet that prints following error message using red color font.“CREATING GUI IS COMPLEX USING SERVLET”.

**22.** Create an html page having two buttons first labeled with GET REQUEST DEMO and second with POST REQUEST DEMO. Create a Servlet that overrides both doGet and doPost methods. After clicking first button doGet method should be called that prints “YOU SELECTED GET TYPE REQUEST”. After clicking second button doPost method should be called that prints “YOU SELECTED POST TYPE REQUEST”.

**23.** Create an html page that has a textfield to take the user name. After submitting the page Servlet should be called that prints Welcome message with entered name like WELCOME SARJE.

**24.** Create an html page Calci.html that has a GUI to take two numbers, to make choice from addition, subtraction, multiplication or division and a submit button. After submitting the form, request should be generated for ArithmeticServlet. This servlet should get two numbers entered by the user and his choice. Perform the requested operation and display the result in the same servlet.

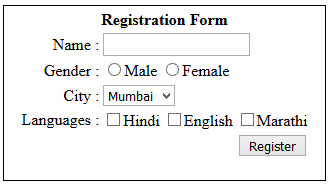
**25.** Create an html page to take the name. Take this entered user name on a servlet. Create another servlet that prints the name taken by the first servlet.

**26.** Create a Servlet that prints number of visits by a particular user.

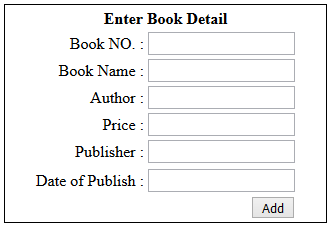
**27.** Create a Servlet that prints the *real* path corresponding to the given *virtual* path.

**28.** Do context parameter entry in web.xml file for jdbc information like jdbc driver class name, url, username, password. Create one servlet that reads these context parameter values from web.xml file and prints them.

**29.** Create a Servlet that prints header information like host (server/domain), user-agent (web browser), mime type in the separate lines.

**30.** Create a Servlet named RegistraionFormServlet that has GUI for registration form to take information like name (textfield), gender(radio button), city(combobox), languages (checkbox) and submit button.

After submitting this form request should go to RegistraionServlet. This servlet should check if all the information are filled. If filled then response should be a success message otherwise response should be redirected to the RegistraionFormServlet to complete the information. Make sure the previously entered information are retained.

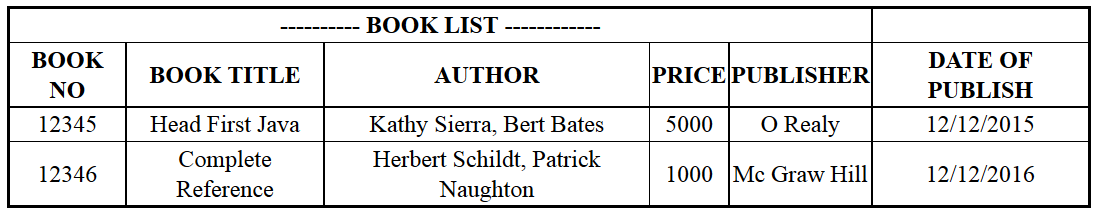
**31.** Create an html page book\_form.html that has a GUI to take book information like book no., book title, author name, publisher name, year of publish and a submit button.

After submitting the form, request should be generated for AddServlet. This servlet should get all book information entered by the user and this information should be stored in the data base table named book. This Servlet should have one link BOOK LIST. When this link is clicked request should be generated for BookListServlet. This servlet should display all book record present in the book database table.

**32.** Create a program to manage GuestBook having properties guestId, guestName, date and message.

a) Create a GuestbookServlet that is mapped to guestbook to display the current guestbook messages.

b) Add a form at the bottom allowing users to submit a new guestbook message. Like below



**JSP [Java Server Pages]**

**33.** Write a JSP code to design a web page that accepts a user name and greet user hello <unm>.

**34.** Write a JSP code to design a web page which accepts username and password and compare it against static username and password, display appropriate message valid/invalid user on same page.

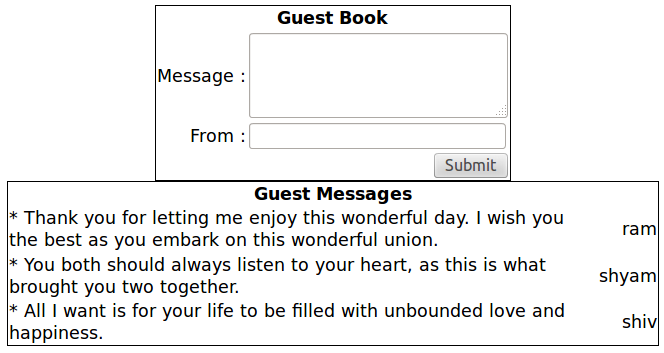
**35.** Write a JSP code to design a web page which accepts username and age from user if age is less than 18 print message “not eligible for vote” else “eligible for vote”.

**36**. Create a Login module using JSPtostorecredential(s) inside Cookie with age of 30 minutes, and do also needful code where user does not has to enter username/password if he is restarting the page within 30 minutes.

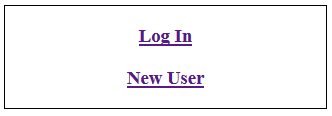
**37.** Write a JSP code which accepts username and nickname from user. At first visit, display message "Hello Username" and for the next successive request, display "Hello nickname".

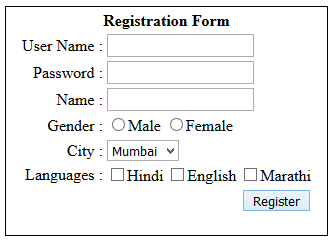
**38.** Create a JSP program for accepting registration details of a singer for singing competition and store in the database table, if the record already exist, display appropriate message.

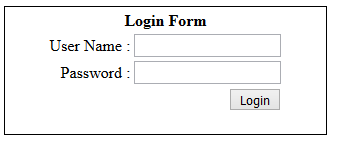
**39.** Write a JSP code to display list of blood donors with details for particular blood group. (Assume a donor table with donor name, blood group, hemoglobin and candidate no columns)

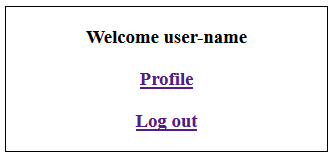
**40.** Create a program to manage GuestBook having properties message and name. a) Create a jsp that displays the current guestbook messages. b) Add a form at the top allowing users to submit a new guestbook message. Like following

**41.** Create an application using JSP that has following pages & conditions:

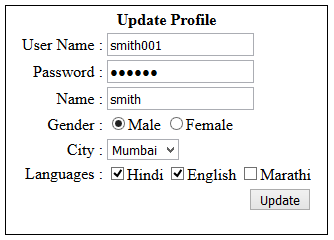
a) index.jsp (which contains link for registration and login)

b) register.jsp (which takes name, username, password, gender, languages and city; store registration data to database table and move to login.jsp page)

c) login.jsp (which validates user credentials (Username/Password) with Database table; redirect to welcome.jsp by maintaining Session if user is authenticated; else display Invalid Credential(s) message on current page)

d) welcome.jsp (which says "Hello <username>!" at the top of page and provides links for Profile and Logout; ensure about that user will not be able to open welcome.jsp page without authentication process)

e) profile.jsp (which displays complete information about user in edit mode and can be updated by user by clicking on Update button after saving profile data, redirect to welcome.jsp page.



f) When user clicks on Logout link available on welcome.jsp destroy running Session and redirect to index.html

**42.** Develop a Product Management web Application using MVC architecture by following these steps.

User Interface (as shown in the figure below) should be created to add product into database table and to display existing product information from database table.

If the user clicks Add button without providing all of the information data should not be added to the database table and the UI must retain the provided information.

If the record is stored in the database table then same user interface should appear with all the empty fields and new record in the list. Records should appear in the descending order of the purchase date.

