Pune Institute of Computer Technology, Pune
Department of Computer Engineering

Web Technology Lab

Class: TE **AY: 2018-19**

Additional Practice Assignments

1. Create an **HTML** page to accept customer's information for the following with appropriate form elements: Customer Name, Address, Gender, Languages known, City, submit and reset button. On submit event store the customer information into **XML** file. Use XSLT.

2. Create a web page using **HTML & CSS** for online registration for international seminar. The participants can be students, faculty members, professional, company/firm representatives from different countries. Simulate payment gateway and show payment successful message. Validate the fields using **JavaScript / jQuery**.

3. Design **HTML** (CSS is optional) form to accept five subject marks of at least five different students. Use **JavaScript / jQuery / XML** to store the marks and calculate the result. Display results and grades of all students using HTML **table**.

4. Create a specimen of a corporate web page (**HTML, CSS**). Divide the browser screen into two frames. The frame on the left will be a menu consisting of hyperlinks. Clicking on any one of these links will lead to a new page, which must open in the target frame at right hand side. In menu create two links, first link that will open a page that displays the company profile, its business & its products. The second link will display that contact address of the company.

5. Create **Employee.xml** file which contains name, id, department, gross salary of an employee. Use this **XML** file with **JavaScript / jQuery** to calculate and display the department wise average salary paid. Provide 2-3 extra similar statistics of an employee based on **XML** file contents.

6. Prepare **Login.html** having username, password and login button. Button click event shall redirect you to **Process.jsp**. At Process.jsp you will notify a message "Login successful" with one **Logout** button. Clicking on Logout button shall redirect you to Login.html. Validate and invalidate **session** at necessary places.

7. Create an application using **JavaScript** and **JSP** to take food order from a customer and generate its bill. (At least one web page to accept order and one for billing)

8. Design an income tax calculator using **Servlets** which takes Name and gross income of an individual as an input and calculates tax payable by an individual based on following income tax slabs: (up to 2.5 Lacs: 0% tax, above 2.5 lacs upto 5 lacs: 10% tax, 5 lacs above: 20% tax)

9. Create **product.html** form to accept products details. Fetch these details into **Servlet First.java** and manage product_id as a part of a **session**. Redirect this session to **Servlet Second.java** using **RequestDispatcher**.

10. Design a **calculator** which can perform addition, subtraction, multiplication and division of two numbers using **AngularJS**.

11. Design a currency converter application using **AngularJS**, which converts INR to USD and vice versa.

12. Create an application using **PHP**, which takes Student_rollno, student_name, student_dept as an input on **HTML** form and stores these details into **MySQL** table.

13. A **MySQL** tables contains Student_rollno, student_name, Student_class, student_dept student_percentage. Create a **PHP** application which fetches data from this table and displays it in sorted order according to percentage.

14. Use **Spring Core** and **Spring Context** to create and fetch **students** records like studentName,studentRollNo, studentBranch etc.

15. Demonstrate **Spring Dependency Injection** example. Following classes and interfaces can be used: interface **Coach** with **getDailyWorkout**(), class **BaseballCoach**, **CricketCoach** that implements **Coach**. Another interface **Fortune** with **getDailyFortune**(). Class **FortuneService** that implements Fortune. Inject the dependency of **FortuneService** instance into **BaseballCoach** and **CricketCoach**. Configure Spring using XML file / Annotations.

16. Demonstrate **Spring IoC** bean example. Create an **Employee** bean Spring BeanFactory. Bean should have eId, eName, eSalary, eDept. Use bean **scope** as **prototype**.