Questions for Web Tech File

HTML, CSS, Bootstrap

1. **Personal Portfolio Webpage**
   Create a multi-section portfolio page with an intro, projects, and contact form. Style it with colors, fonts, and basic layout using CSS.
2. **Responsive Navigation Bar**
   Build a top navigation bar that changes layout on smaller screens (hint: use bootstrap).
3. **Styled Table Showcase**
   Create an HTML table to display product pricing or student grades and style it with borders, padding, and alternating row colors.
4. **Profile Card Grid**
   Make a layout with three profile cards using CSS Grid. Each card includes an image, name, title, and a short description.
5. **Login Form with Validation Styles**
   Design a login form using HTML and style it with CSS. Include hover effects, focus styles, and error message placeholders.
6. **Build a Blog Post Layout**
   Make a single post page with headings, images, paragraphs, and tags. Use CSS to give the page a clean, readable look.
7. **CSS Animations and Transitions**
   Create buttons or elements that animate on hover or click—like color transitions, sliding menus, or fade-ins.
8. **Newsletter Signup Page**
   Make a form with fields for name and email. Use CSS to style input fields and add simple responsiveness.

Java Script

1. **Interactive Calculator**
   Create a simple calculator that performs addition, subtraction, multiplication, and division based on user input.
2. **To-Do List App**
   Build a dynamic to-do list where users can add and remove tasks. Add optional features like marking tasks as completed.
3. **Form Validation**
   Write JavaScript that validates a form (e.g., checks if the email input contains "@", and that password fields match).
4. **Color Changer**
   Make a page where clicking a button changes the background color to a random color.
5. **Real-Time Clock**
   Create a digital clock that updates every second and displays the current time in hours, minutes, and seconds.

Servlet

1. **Form Data Processing**
   Create an HTTP servlet that accepts user input from a form (e.g., name and email) and displays the submitted data on a new page.
2. **Session Tracking**
   Write a servlet that tracks the number of times a user has visited the site using HTTP sessions. Display a welcome message that changes based on visit count.
3. **Redirect vs Forward**
   Create two servlets: one that uses sendRedirect() and another that uses RequestDispatcher.forward(). Demonstrate the difference in behavior between the two.

JSP

1. **User Registration System**
   Create a JSP page with a form that collects user details (name, email, password). On submission, store the data in a MySQL database using JDBC.
2. **Login Authentication**
   Build a login page that checks user credentials against a database. If valid, redirect to a welcome page; otherwise, show an error message.
3. **Display Records from Database**
   Write a JSP page that connects to a database and displays a list of products (or students) in an HTML table using a loop.
4. **Update and Delete Operations**
   Create a page that lists records with “Edit” and “Delete” buttons. Implement functionality to update or delete the selected record using JSP and JDBC.
5. **Search Functionality**
   Design a search form that allows users to enter a keyword. On submission, display matching records from the database using a LIKE query.