

Web Programming List for Practicals

1. **Create a basic web page using HTML elements like headings, paragraphs, images, and links.**
Example: Build the homepage of a personal blog, including a welcome message, profile image, and navigation links to different sections like "About Me," "Blog Posts," and "Contact."
2. **Design a personal portfolio webpage using HTML5 semantic elements.**
Example: Develop a digital resume with sections like "Experience," "Skills," and "Projects," utilizing <header>, <section>, and <footer> to structure the content.
3. **Build a web page to display a product catalog using HTML tables and lists.**
Example: Create a catalog page for an online electronics store displaying products like smartphones and laptops with their features (price, specifications) using a table layout.
4. **Create a contact form using HTML with text fields, radio buttons, checkboxes, and a submit button.**
Example: Develop a "Contact Us" page for a service-based business (e.g., web design agency), collecting user details like name, service inquiry type, and preferred contact method.
5. **Create a web page to display a photo gallery using HTML and image elements.**
Example: Build a travel blog gallery where users can browse through various destination images, each with a title or description.
6. **Design a simple HTML page with internal CSS for styling headings and paragraphs.**
Example: Create an "About Us" page for a small business, using CSS to style different sections (history, mission, values) with distinct colors and fonts.
7. **Apply CSS to style an HTML page with properties like color, font-size, margin, padding, and borders.**
Example: Style an e-commerce product page, ensuring each product card has a margin between them, padded content, and proper color schemes for the title and price.
8. **Create a responsive web page layout using CSS Grid and Flexbox.**
Example: Design a multi-column layout for an online magazine that adjusts the number of columns based on the screen size, showcasing articles with images and summaries.
9. **Implement a navigation bar using HTML and CSS with hover effects.**
Example: Create a top menu for a restaurant website with links to "Home," "Menu," "Gallery," and "Reservations," and add a hover effect to highlight the selected section.
10. **Build a responsive webpage using Bootstrap and create a login form.**
Example: Develop a login page for a fitness application, ensuring it adapts to mobile devices using Bootstrap's grid system, with the form centrally aligned.
11. **Use Bootstrap to create a responsive grid layout for a portfolio webpage.**
Example: Design a personal portfolio showcasing your work, such as project screenshots and descriptions, with a grid layout that adapts to different screen sizes.
12. **Apply external CSS to style a webpage that showcases different types of typography.**

Example: Create a blog post page and use CSS to style headings, body text, blockquotes, and footnotes in unique but harmonious typography styles.

13. Use CSS animations to create a hover effect on buttons.

Example: Add a “Sign Up” button to an event registration page, where the button changes color and size when the user hovers over it.

14. Add a background image and style it using CSS.

Example: Design the homepage of a bakery website with a background image of a bakery, adjusting its position, repeat, and opacity for readability of text.

15. Use CSS to create a pricing table for different subscription plans.

Example: Build a subscription plan comparison for a SaaS product, highlighting features of Basic, Pro, and Enterprise plans with distinct colors and layouts.

16. Create a basic JavaScript program that performs arithmetic operations.

Example: Develop a simple web calculator that allows users to perform basic arithmetic (addition, subtraction, multiplication, and division) on two input numbers.

17. Add interactivity to a web page using JavaScript events (e.g., mouse hover, click).

Example: Implement a dynamic FAQ section for a website where clicking on each question reveals the answer, and hovering over the question highlights it.

18. Create a web application for BMI Calculation using JavaScript.

Example: Build a health tool for users to input their height and weight, and then calculate and display their Body Mass Index (BMI).

19. Use JavaScript to validate form inputs such as email and phone numbers.

Example: Develop a user registration form for a job portal and use JavaScript to ensure that the email is in the correct format and that the phone number contains only digits.

20. Implement a simple JavaScript clock that shows the current time.

Example: Create a live clock for a dashboard page of a stock trading platform that updates every second.

21. Create a to-do list web application using JavaScript for adding/removing tasks.

Example: Develop a task manager app where users can add daily tasks, mark them as completed, and remove them from the list.

22. Use JavaScript to dynamically add, remove, and modify HTML elements.

Example: Create a shopping cart for an online store where users can dynamically add items, remove them, and adjust the quantity in real-time.

23. Implement form validation using JavaScript for a user registration form.

Example: Build a registration page for a conference that validates user inputs like name, email, and chosen sessions before submission.

24. Use JavaScript to create a simple image slideshow on a web page.

Example: Create a rotating image banner for a real estate website to showcase different property images.

25. Implement a dropdown menu using JavaScript and CSS.

Example: Create a multi-level navigation menu for a clothing store where hovering over “Women” shows options like “Dresses,” “Tops,” and “Shoes.”

26. Create a responsive navigation bar using Bootstrap and JavaScript.

Example: Design a website for a cafe with a collapsible navigation bar on smaller screens, using Bootstrap’s navbar component.

27. Develop a simple e-commerce webpage using Bootstrap components.

Example: Create a product listing page for an online clothing store using Bootstrap cards to display products in a grid layout.

28. Create a modal popup form using Bootstrap and JavaScript.

Example: Design a subscription popup form that appears when users visit a blog page, offering them the chance to subscribe to a newsletter.

29. Build a simple blog webpage using HTML, CSS, and Bootstrap.

Example: Create a personal blog page with a sidebar for categories, a main area for posts, and a footer using Bootstrap's grid layout.

30. Create a simple React application that displays a list of items.

Example: Develop a grocery list app in React where users can add items to a list and mark them as purchased.

31. Use React to create a dynamic to-do list application.

Example: Implement a task management app in React with the ability to add tasks, set deadlines, and mark them as complete.

32. Create a contact form in React and validate inputs using JavaScript.

Example: Build a contact form for a business website in React that checks for valid email addresses and mandatory fields.

33. Develop a weather application using React and an open API.

Example: Create a React app that fetches and displays the current weather for a user's location using a weather API.

34. Implement a web page using PHP to display form data submitted by a user.

Example: Build a simple feedback form where users submit comments, and the page displays the comments in a list.

35. Connect a PHP application to a MySQL database and perform CRUD operations.

Example: Create an admin panel for managing blog posts, allowing users to add, edit, delete, and view posts stored in a MySQL database.

36. Create a registration form using PHP and store the user data in MySQL.

Example: Develop a user sign-up page for a membership site, storing user information (name, email, password) in a MySQL database.

37. Create a REST API with Node.js and Express to manage products in a store.

Example: Build an API for a bookstore that allows clients to view, add, update, and delete book records.

38. Implement a Node.js application that connects to MongoDB to store and retrieve data.

Example: Create a simple blog backend where users can write posts that are stored in a MongoDB database and retrieved dynamically.

39. Create a dynamic blog application using the Django web framework.

Example: Develop a blog site using Django where users can post articles, leave comments, and view archives.

40. Implement a user authentication system using Django and MySQL.

Example: Build a login system for a web app where users can register, log in, and access protected pages based on their credentials.