

QUESTION BANK SYSTEM

PAPER VIII: WEB TECHNOLOGIES

Semester IV — Computer Science / Applications

UNIT I: Internet Protocols, WWW, and HTML Basics

A. Short Answer Questions (1 Mark Each)

1. Define the term 'Internet'.
2. Name any two common Internet communication protocols.
3. What does WWW stand for in the context of web technologies?
4. Who invented the World Wide Web and in which year?
5. Give full form of HTTP.
6. Mention any two common HTTP request methods.
7. What is the fundamental purpose of a Web Browser?
8. Give two popular examples of modern web browsers.
9. Define a Web Server.
10. Mention two popular examples of operational web servers.
11. What is the primary difference between a static web page and a dynamic web page?
12. Briefly state the core working principle of the WWW.
13. What is front-end development?
14. What is back-end development?
15. Define client-server communication.
16. Write down the full form of HTML.
17. State any one major characteristic of HTML.
18. Write the fundamental root tag of an HTML document.
19. Which HTML tag contains meta-information like titles and stylesheets?
20. Name the tag inside which all visible content of an HTML document is structured.

B. Conceptual Questions (2 Marks Each)

1. Explain the historical evolution of the Internet in brief.
2. Differentiate between HTTP and HTTPS protocols.
3. Explain the specific roles of GET and POST methods in HTTP.
4. Describe the basic architecture of a typical Client-Server interaction model.
5. Illustrate the basic skeleton structure of an HTML5 document.
6. Differentiate between an HTML element and an HTML attribute with examples.
7. How do you create an ordered list and an unordered list in HTML?
8. Show the syntax for inserting a hyperlink and an image into an HTML page.
9. Explain how row spacing and column widths are managed in HTML tables using attributes.
10. What is the role of the HTML 'action' and 'method' attributes within a form element?
11. Discuss the importance and function of web server software like Apache or Nginx.
12. What are block-level and inline elements in HTML? Give examples of each.
13. Explain the purpose of the tag in modern web design.
14. How are audio and video files integrated into HTML without third-party plugins?
15. Explain how a text area field differs from a normal standard text input field in forms.

C. Analytical & Programming Questions (5 Marks Each)

1. Describe the core structural components and working pipeline of the World Wide Web.
2. Elaborate on the differences between front-end and back-end development technologies and architectures.
3. List and explain five essential text formatting tags available in HTML along with code outputs.
4. Discuss the concept of client-server architecture in web systems, highlighting how a browser parses strings.
5. Explain the implementation of tables in HTML, detailing tags like , , , and attributes like colspan/rowspan.
6. Provide a detailed review of HTML multimedia elements, explaining the syntax, formats, and control attributes for audio/video.
7. Sketch out a comprehensive HTML Form containing text inputs, radio buttons, checkboxes, dropdown menus, and submit triggers.
8. Explain HTML frames and iframes, clarifying how target links can load content into specific windows or frames.
9. Discuss the structural layout of an HTML page, explaining semantic tags like , , , and .
10. Write down step-by-step instructions and code to build a complete corporate contact web page with mapped hyperlink paths.

D. Comprehensive & Design Questions (8 Marks Each)

1. Explain the entire lifecycle of an HTTP request-response cycle starting from URL input in a browser to rendering text elements, naming all participating server layers.

2. Create a fully structured, multi-input registration form in HTML that includes user validation containers, dropdown selectors, fieldsets, text areas, and password attributes with neat inline comments.
3. Discuss the evolution of front-end vs back-end technologies. Map out a detailed ecosystem chart showing how HTML, CSS, client-side scripts, databases, and server scripts collaborate.
4. Design a complex academic results page using advanced HTML table styling, showcasing merging cells via rowspan/colspan, header blocks, custom captions, and nested lists inside cells.
5. Write an extensive guide explaining the history of protocols, comparing HTTP/1.1, HTTP/2, and HTTP/3 mechanisms, detailing how multi-connection streams optimize rendering performance.

UNIT II: Cascading Style Sheets (CSS) & Layout Design

A. Short Answer Questions (1 Mark Each)

1. What does CSS stand for?
2. Mention two distinct advantages or benefits of using CSS in web design.
3. Write the basic syntax pattern of a CSS rule.
4. Define a CSS selector.
5. Name three ways to inject or apply CSS into an HTML document.
6. What is Inline CSS?
7. What is Internal or Embedded CSS?
8. Write the syntax for creating a comment inside a CSS document.
9. Which property is used to change the text color of an element in CSS?
10. Name the property used to set the background color of an HTML page.
11. Which property controls the text alignment inside an element?
12. List two font-related properties available in CSS.
13. What are the four main components of the CSS Box Model?
14. Define the term 'Padding' in the context of the CSS box model.
15. What is the fundamental difference between Margin and Padding?
16. Give an example of a CSS Pseudo-class.
17. Give an example of a CSS Pseudo-element.
18. Which property is used to configure scrolling behavior or manage text overflow?
19. What does the 'display: none;' property do to an element?
20. Name the CSS property used to create a dropdown navigation panel toggle link.

B. Conceptual Questions (2 Marks Each)

1. Compare External CSS with Internal CSS regarding project scale efficiency.
2. Explain Element selectors, ID selectors, and Class selectors in CSS.
3. Describe how to set custom list-style types and positions using CSS.
4. Explain the box model components: Content, Padding, Border, and Margin.
5. Differentiate between pseudo-classes and pseudo-elements with examples.
6. Write a short snippet showing how to implement a sticky or fixed navigation bar.
7. Explain the 'display: block;' and 'display: inline;' property behaviors.
8. How do you create a text shadow effect using standard CSS properties?

9. Explain how border-radius is used to create circular element containers.
10. Discuss how background images are positioned and repeated via CSS properties.
11. What is the purpose of the 'overflow: scroll;' configuration?
12. Explain the use of the descendant selector space character in compound CSS rules.
13. Illustrate the syntax for controlling hover states on hyperlink anchor tags.
14. What are CSS font stacks, and why are fallback fonts specified?
15. Explain how table borders can be combined or separated using the border-collapse property.

C. Analytical & Programming Questions (5 Marks Each)

1. Explain the CSS Box Model in detail, drawing a descriptive architecture block diagram showing padding, border, and margin spaces.
2. Discuss the three methods of integrating CSS styles with code syntax examples for inline, internal, and external approaches.
3. Elaborate on different types of CSS Selectors including universal, attribute, grouping, combinators, and contextual structures.
4. Design a clean, responsive top-level horizontal navigation bar using CSS block modifications, custom padding, and font styles.
5. Explain the properties used to format tables in CSS, focusing on striping rows, handling cell paddings, and alignment rules.
6. Discuss typography formatting in CSS, detailing font-family, font-size, font-weight, text-transform, and text-indent rules.
7. What are pseudo-classes? Explain the configuration of interactive states for anchor links (:link, :visited, :hover, :active).
8. Describe the 'display' property options in CSS, outlining the mechanics of block, inline, inline-block, flex, and grid frameworks.
9. Explain how vertical and horizontal dropdown menus are implemented using purely CSS positioning and display alterations.
10. Write down a complete CSS stylesheet snippet that modifies a standard HTML card structure with custom shadow, margins, and borders.

D. Comprehensive & Design Questions (8 Marks Each)

1. Design an extensive CSS template framework from scratch that builds a three-column web layout with custom header, sidebar, primary view, and footer containers using advanced positioning and alignment properties.
2. Write a comprehensive review on the CSS Box Model. Provide complete code sets demonstrating how padding, margins, borders, box-sizing, and background positions dynamically alter real layout sizes.
3. Build a fully operational, multi-level hover dropdown navigation bar in HTML/CSS, explaining the code trick behind hiding and revealing secondary block elements.

4. Discuss CSS structural hierarchies, selector specificity weights, cascade ordering, and the rule of style inheritance across complex nested elements.
5. Construct a sample layout that demonstrates advanced CSS formatting techniques across lists, forms, and custom buttons, including transition animations for hover states.

UNIT III: Client-Side Scripting via JavaScript & DOM

A. Short Answer Questions (1 Mark Each)

1. What is JavaScript?
2. List any one key feature of JavaScript.
3. State a major benefit of using JavaScript in client-side web development.
4. Write down the HTML tag used to enclose JavaScript statements.
5. Name the built-in popup box function used to flash warning messages.
6. What value does a 'confirm()' box return if a user clicks on the Cancel button?
7. Which popup function takes direct text string inputs from site users?
8. Mention one method used to output text statements directly into document pipelines.
9. What is the role of 'console.log()' in JavaScript environments?
10. Name any three primitive data types available in core JavaScript.
11. How do you declare a variable in JavaScript?
12. Write the single-line comment symbol used in JavaScript code blocks.
13. Name two arithmetic operators supported in JavaScript equations.
14. Give an example of a logical comparison operator in JavaScript.
15. Write the basic syntax structure of an 'if...else' condition block.
16. Which loop structure checks condition vectors only after executing its internal body once?
17. Name the statement used to instantly abort a loop sequence.
18. What does DOM stand for?
19. Which built-in object handles system times and calendar dates in JavaScript?
20. Name an execution event handler associated with form submit buttons.

B. Conceptual Questions (2 Marks Each)

1. Differentiate between alert(), confirm(), and prompt() boxes with expressions.
2. Explain how variables are declared using 'var', 'let', and 'const' keywords.
3. Differentiate between '==' and '===' comparison operators in JavaScript logic.
4. Describe the basic features and applications of arrays in client scripts.
5. Explain how user-defined functions are structured and executed in JavaScript.
6. Write a short segment showing how to instantiate and access components of a Date object.
7. What is event handling? Explain the utility of the 'onclick' event property.
8. Show how JavaScript can read the text input value of a form field using DOM models.

9. Explain conditional assignment using the ternary operator pattern.
10. Briefly describe two built-in mathematical functions provided by the Math object.
11. Discuss loop navigation alterations using the 'continue' keyword syntax.
12. How do you dynamically change the color of an HTML text container using DOM attributes?
13. What is an anonymous function block? Provide a simple conceptual example.
14. Explain string concatenation syntax using both plus operators and template literal tokens.
15. Describe how the 'onsubmit' event is utilized to interrupt form routing during evaluations.

C. Analytical & Programming Questions (5 Marks Each)

1. Discuss JavaScript features, highlighting its position as an interpreted, cross-platform, client-side scripting language.
2. Explain the methods available for displaying outputs in JavaScript, covering document.write(), console.log(), alert(), and innerHTML.
3. Elaborate on control flow loops in JavaScript, comparing the performance and execution metrics of for, while, and do-while systems.
4. Write a JavaScript program to find the largest number out of a user-provided array list using simple evaluation loops.
5. Discuss the structural hierarchy of the Document Object Model (DOM), showing how elements are accessed and updated via scripts.
6. Explain string manipulation techniques in JavaScript using functions like sub_str(), charAt(), toUpperCase(), and indexOf().
7. Write a script that checks a form input text block to verify if it is empty, flashing custom alerts based on evaluation flags.
8. Discuss structural functions in JavaScript, clarifying differences between built-in utilities and user-defined function parameters.
9. Create an interactive digital counter using HTML elements and a JavaScript event listener structure that modifies DOM value fields.
10. Explain how system time is checked using Date objects to print custom morning/evening text strings on webpage loads.

D. Comprehensive & Design Questions (8 Marks Each)

1. Write a complete validation script for an HTML registration form that verifies email string patterns, confirms matching passwords, and enforces age boundary criteria before routing data.
2. Explain the Document Object Model (DOM) comprehensively. Detail the methods used to fetch elements by ID, class, or tag, and how to programmatically create and append new HTML elements.
3. Write a JavaScript program to implement an interactive calculator layout that evaluates arithmetic string streams on button presses and updates output boxes.

4. Discuss the concept of multi-branch logic and iterations. Write a program that takes integer inputs via prompts, detects if the input is a prime value, and logs output arrays.
5. Explain events and handlers in JavaScript. Map out a comprehensive reference guide explaining keyboard events, mouse movement tracking, form updates, and focus shifts.

UNIT IV: Server-Side Web Development using PHP & MySQL

A. Short Answer Questions (1 Mark Each)

1. What does PHP stand for?
2. State one principal feature of PHP as a server-side language.
3. Write the opening and closing syntax tags for embedding PHP commands.
4. How do you declare a variable in a PHP code block?
5. Which statement is widely used to print text strings onto output documents?
6. Name any two primitive scalar data types supported in PHP.
7. How do you declare a constant parameter in PHP?
8. Write the string concatenation operator symbol used in PHP.
9. Name two types of arrays available within PHP structures.
10. Which statement terminates loop iterations or switch match flows in PHP?
11. Write the syntax template for a 'foreach' iteration loop in PHP.
12. Name a built-in string handling function used to count character length.
13. Which function is used to fetch the current system date inside a script?
14. What superglobal array collects form parameters routed via the POST method?
15. Name the built-in library function used to initiate database connections to MySQL.
16. What function runs SQL query statements against connected database schemas?
17. Write the SQL statement template keyword used to insert records into data tables.
18. Which function fetches a data row from an active query collection as an associative array?
19. What PHP function is used to close an open database stream safely?
20. How do you write a single-line text comment inside a PHP script block?

B. Conceptual Questions (2 Marks Each)

1. Compare print and echo statements in PHP regarding execution metrics.
2. Explain Indexed Arrays and Associative Arrays with initialization examples.
3. Describe how constants are declared and consumed using the 'define()' function.
4. Explain conditional evaluation handling via 'if-elseif-else' structures.
5. How does the 'foreach' loop optimize the traversal of associative data maps?
6. Show how PHP code strings are neatly embedded within plain HTML container blocks.
7. Explain the function and structural purpose of the '\$_GET' superglobal matrix array.
8. Write a snippet that connects a script to a database schema, handling connection errors.

9. Explain how to find the total length and word count of a text variable using built-in utilities.
10. Describe the execution workflow of user-defined functions that accept optional parameter inputs.
11. What is the role of mathematical functions like 'rand()' and 'sqrt()' in PHP scripts?
12. Show the SQL syntax template string for updating table rows based on conditional inputs.
13. How does the 'mysqli_fetch_assoc()' routine work inside record extraction loops?
14. Explain why security precautions require form sanitization procedures before query compilation.
15. Describe how a script collects multi-line text input blocks from a form text area.

C. Analytical & Programming Questions (5 Marks Each)

1. Discuss server-side execution features of PHP, detailing how pages are compiled on servers before routing plain HTML text back to clients.
2. Explain array configurations in PHP, covering indexed, associative, and multi-dimensional matrices with structural syntax examples.
3. Write a PHP script that parses a target numeric variable to verify if it is an armstrong value using iteration loops.
4. Discuss built-in string utilities in PHP, highlighting the usage and outputs of strlen(), strrev(), strpos(), and explode().
5. Write a comprehensive program segment showing how data entries routed from an HTML form are fetched and assigned to PHP variables.
6. Explain step-by-step how to establish a MySQL database connection, compile a creation query, and build a data table via PHP scripts.
7. Describe conditional loop control systems in PHP, detailing while, do-while, for, and foreach frameworks with syntax templates.
8. Write a custom PHP function that accepts an integer array, sorts it, and returns the computed arithmetic mean.
9. Explain database manipulation commands in PHP, clarifying the procedures for inserting records and reading transaction status flags.
10. Discuss date and time processing in PHP, explaining how date formatting parameters are used to present customized time headers.

D. Comprehensive & Design Questions (8 Marks Each)

1. Design a comprehensive server-side system in PHP that receives web form inputs, sanitizes data parameters, establishes a MySQL connection, registers a user record, and prints an HTML confirmation grid.
2. Write an extensive guide explaining array handling in PHP. Construct a program that loops through multi-dimensional associative arrays of catalog records to print structured tables.
3. Build a complete CRUD (Create, Read, Update, Delete) dashboard pipeline script in PHP that queries a database, modifies fields based on URL queries, and outputs status updates.

4. Discuss the integration architecture of PHP and HTML. Explain data passing mechanics via superglobals, detailing cookie state tracking, session management pipelines, and form actions.
5. Write a comprehensive PHP module implementing secure authentication verification routines that validate form strings against hashed keys, managing login routing steps safely.