K. K. Wagh Institute of Engineering Education and Research, Nashik

(An Autonomous Institute from A. Y. 2022-23)

MODEL ANSWER KEY

Exam Seat No.:

End-Sem Examination-I	Winter 2025
Academic Year: 2025-26	Sem: I
Class: FYMCA	Program: MCA
Branch Code:09	Pattern: 2024
Name of Course: Web Technologies	Course Code: 2409513
Max. Marks: 60	Duration: 2:30 Hrs.

Instructions: Candidates should read carefully the instructions printed on the Question Paper and on the cover page of the Answer Book, which is provided for their use.

(*keep space*)

1. This question paper contains 2 page(s).
2. Answer to each new question is to be started on a new page.
3. Assume suitable data wherever required, but justify it.
4. Draw the neat labelled diagrams, wherever necessary.
5. The last columns indicates the Course Outcome

Q. No.	Details	Max. Marks	CO No.	BT Level
Q.1	a) Describe how HTML5 is used in modern web development and explain in what ways it improves the user experience compared to older versions of HTML. . (6 Marks) Answer: Solution: HTML5 revolutionized web development by introducing a more robust and user-friendly approach to Structuring and presenting web content. Key enhancements include: • **Semantic Elements:** Elements like <header>, <footer>, <article>, and <section> provide meaningful structure, improving SEO and accessibility. • **Multimedia Support:** HTML5 supports embedding audio (<audio>) and video (<video>) directly without third-party plugins, offering a seamless user	[6]	CO1	L2

	experience. • **Graphics & Animation:** The <canvas> element and integration with SVG enable dynamic graphics and animations directly in the browser. • **Form Enhancements:** New input types (e.g., email, date) and built-in validation reduce reliance on JavaScript and improve form usability. • **Local Storage:** HTML5's Web Storage APIs (localStorage, sessionStorage) allow web applications to store data on the client side, offering better performance and offline capabilities. • **Cross-Platform Compatibility:** Designed with mobile and responsive design in mind, HTML5 ensures consistent performance across devices. These features collectively enhance the interactivity, responsiveness, and functionality of modern web applications, setting the foundation for rich, user-centric experiences.			

Q.2	a) Write and apply a JavaScript function to determine whether a given number is even or odd, and demonstrate how the logic works with an example.
	
	(6 Marks)
	Answer:
	
	Solution: function checkEvenOdd(num) {
	if (num % 2 === 0) {
	return "Even";
	} else {
	return "Odd";
	}
	}
	
	Explanation:
	
	• The function checkEvenOdd accepts a number num.
	• It uses the modulo operator (%) to determine the remainder when dividing num by 2.
	• If num % 2 === 0, it means the number is divisible by
		[6]	CO2	L3

	2, so it's even. Otherwise, it's odd. • The if...else control structure applies conditional logic to decide the result.			
Q.3	a) Use real-world scenarios to illustrate how XML and JSON can be applied, and explain how their advantages and disadvantages affect their usage in those situations. (8 Marks) Solution: **Feature** / **XML** / **JSON** **Advantages**: - Rich metadata with attributes; - Schema validation with DTD/Schema; - Suitable for complex document structures / - Lightweight and easy to read; - Faster parsing and smaller size; - Natively compatible with JavaScript **Disadvantages**: - Verbose syntax; - Requires parsing overhead; - Complex to handle in browsers / - Limited metadata support; - Lacks robust validation like DTD; - Not as human-readable for non-technical users **Explanation:** • XML is ideal for complex document structures and data integrity checks (via DTD/Schema). • JSON excels in lightweight data exchange, particularly for web APIs and dynamic applications. • The choice depends on project needs: XML for robust data modeling, JSON for speed and simplicity. **OR** b) Describe JSON documentation. Illustrate how JSON elements are documented with acceptable values, data types, and nesting rules. Provide an example of documentation for a JSON object. (8 Marks)	[16]	CO3	L3

Answer:

- Solution: **JSON documentation** explains the structure, elements, data types, and allowed values of a JSON object.
- It ensures that developers understand how to create or consume JSON data.
- **Key documentation aspects:**
  - Element names and descriptions
  - Data types (string, number, boolean, array, object)
  - Acceptable values (e.g., enums, ranges)
  - Nesting and hierarchy rules

Example Documentation:

```
json
CopyEdit
{
  "Student": {
    "Name": "string (Required) - Full name of the student",
    "ID": "number (Required) - Unique student ID",
    "Department": "string (Optional) - Academic department",
    "Email": "string (Optional) - Email address"
  }
}
```

Explanation:

- This describes the Student object with each property's type and whether it's required or optional.
- Acceptable values and descriptions clarify how to use the JSON structure, ensuring data consistency.

c) Develop a sample JSON file for a library system and demonstrate how it can be applied in practice. Then, explain how its structure and usage differ from an equivalent XML representation.

CO3 | L3

Answer:

Solution: json
CopyEdit
{

```
  "Library": [
    {
     "Title": "JSON Basics",
     "Author": "John Doe",
     "ISBN": "9876543210",
     "Year": 2024,
     "Language": "English"
    }
  ]
}
```

Comparison:

- Concise, no closing tags.
- Uses key-value pairs, arrays for collections.

OR

d) What is JSON? Demonstrate the working of JSON with a practical example. . (8 marks)

Answer:

Solution:

Definition: JavaScript Object Notation (JSON) is a lightweight data format using key-value pairs and arrays.

- **Working:**
  - Data stored as objects (curly braces) and arrays (square brackets).
  - Easily parsed into objects in JavaScript using JSON.parse().
- **Example:**

json
CopyEdit

```
{
  "name": "Alice",
  "age": 25,
  "skills": ["Python", "Java"]
}
```

- **Explanation:** Web APIs and configurations use JSON for lightweight, fast data exchange.

Q.4	a) How would you implement the control statements in PHP with a focus on if-else, switch, and for each, using suitable example. (8 marks)
	
	Answer:
	Solution:
	
	• **if-else**:
	
	php
	CopyEdit
	if($role == 'admin') {
	echo "Access granted.";
	} else {
	echo "Access denied.";
	}
	
	• **switch**:
	
	php
	CopyEdit
	switch($role) {
	case 'admin': echo "Admin Panel"; break;
	case 'editor': echo "Editor Panel"; break;
	default: echo "User Panel"; break;
	}
	
	• **foreach**:
	
	php
	CopyEdit
	$users = ["Alice" => "admin", "Bob" => "editor"];
	foreach($users as $name => $role) {
	echo "$name is $role
	";
	}
	
	• **Application**: Control statements manage flow in user-based systems, ensuring the right content and permissions are displayed.
	
	OR
	b) Write and apply a PHP script that inserts form data (name and email) into a MySQL database, and demonstrate how the
		[16]	CO4	L3

flow of execution and error handling work in this process.
(8 marks)

Solution:

```
php
CopyEdit
$conn = new mysqli("localhost", "root", "", "mydb");
$name = $_POST['name'];
$email = $_POST['email'];
$sql = "INSERT INTO users (name, email) VALUES
('$name', '$email')";
if($conn->query($sql)) {
  echo "User added.";
} else {
  echo "Error: ".$conn->error;
}
$conn->close();
```

- **Flow**:
  - Form data ($_POST) captured.
  - SQL INSERT query formed.
  - Checks for errors and outputs messages.
  - Closes connection.

c) How to create and use a PHP array concept which helps to store student scores and calculate the average using a loop.
(8marks)
Solution:

```
php
CopyEdit
$scores = [85, 90, 78, 92];
$total = 0;
foreach($scores as $score) {
  $total += $score;
}
$average = $total / count($scores);
echo "Average score: $average";
```

- **Explanation**: Loops through array elements, sums scores, calculates average. Useful in student portals.

CO4 | L3

OR

d) Design a sample MySQL database schema for a small application, and then implement it in PHP by writing and executing SQL queries to insert, retrieve, and update data. Demonstrate the implementation with working PHP code and explain how each part of the code applies the schema and performs the database operations. (8marks)

Solution:

Let's create a simple employees table to store employee details:

Column	Data Type	Description
id	INT (Primary Key)	Auto-incremented ID
name	VARCHAR(100)	Employee's full name
email	VARCHAR(100)	Employee's email
department	VARCHAR(50)	Department name
salary	DECIMAL(10,2)	Employee salary

Step 2: Create the Database and Table in MySQL

sql
CopyEdit

```
CREATE DATABASE company;

USE company;

CREATE TABLE employees (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100) NOT NULL,
  email VARCHAR(100) NOT NULL,
  department VARCHAR(50) NOT NULL,
  salary DECIMAL(10,2) NOT NULL
);
```

Step 3: PHP Script to Connect to MySQL

php
CopyEdit

```
<?php
// Database connection parameters
$servername = "localhost";
```

```
$username = "root";
$password = "";
$dbname = "company";

// Create connection
$conn = new mysqli($servername, $username, $password,
$dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
```

Step 4: Insert Data into the employees Table

```
php
CopyEdit
<?php
// Assuming connection $conn is established (see Step 3)

// Insert employee data
$name = "John Doe";
$email = "john.doe@example.com";
$department = "Sales";
$salary = 55000.00;

$sql = "INSERT INTO employees (name, email, department,
salary) VALUES (?, ?, ?, ?)";
$stmt = $conn->prepare($sql);
$stmt->bind_param("sssd", $name, $email, $department,
$salary);

if ($stmt->execute()) {
    echo "New employee inserted successfully.";
} else {
    echo "Error inserting employee: " . $stmt->error;
}

$stmt->close();
?>
```

Step 5: Retrieve Data from the employees Table

```
php
CopyEdit
<?php
// Fetch all employees
$sql = "SELECT id, name, email, department, salary FROM
employees";
```

```
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // Output data of each row
    while($row = $result->fetch_assoc()) {
        echo "ID: " . $row["id"] . " - Name: " . $row["name"] .
            " - Email: " . $row["email"] . " - Department: " .
$row["department"] .
            " - Salary: $" . $row["salary"] . "<br>";
    }
} else {
    echo "No employees found.";
}
?>
```

Step 6: Update Data in the employees Table

```
php
CopyEdit
<?php
// Update salary of an employee by ID
$employee_id = 1;  // For example, update employee with id =
1
$new_salary = 60000.00;

$sql = "UPDATE employees SET salary = ? WHERE id = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("di", $new_salary, $employee_id);

if ($stmt->execute()) {
    echo "Employee salary updated successfully.";
} else {
    echo "Error updating salary: " . $stmt->error;
}

$stmt->close();
$conn->close();
?>
```

Summary:

- Step 1 & 2: Create database and table schema.
- Step 3: Connect PHP to MySQL.
- Step 4: Insert data using prepared statements for security.
- Step 5: Retrieve and display data.
- Step 6: Update existing data securely.

Q.5	a) Explain how ASP processes a client request. Illustrate the role of core ASP objects in managing user input. Provide an example of a simple ASP form processing. (8 marks) Answer: Solution: ASP (Active Server Pages) processes client requests on the server side. When a client sends a request (usually via HTTP), the IIS server hands it over to the ASP engine which parses the ASP script embedded in the webpage. The ASP engine executes the script, interacts with server resources (databases, files), and generates HTML output sent back to the client browser. **Core ASP Objects:** • **Request:** Collects data sent by the client (form data, query strings). • **Response:** Sends output to the client browser. • **Server:** Provides utility methods, such as URL encoding, script mapping. • **Session:** Stores user-specific information between requests. • **Application:** Shares data among all users. **Example:** asp CopyEdit <form method="post" action="process.asp"> Name: <input type="text" name="username" /> <input type="submit" value="Submit" /> </form> In process.asp: asp CopyEdit <% Dim userName userName = Request.Form("username") Response.Write("Hello, " &	[16]	CO5	L3

	Server.HTMLEncode(userName)) %> **Explanation:** The form submits data via POST. ASP uses Request.Form to access the user input and safely outputs it using Server.HTMLEncode to avoid script injection. **OR** b) Describe the stages of the JSP lifecycle and apply your understanding by explaining how these stages help in designing efficient JSP pages with an example. (8 marks) Answer: Solution: The JSP lifecycle consists of several stages: 1. **Translation:** The JSP file is translated into a Java servlet. This happens only once unless the JSP is modified. 2. **Compilation:** The generated servlet is compiled into bytecode. 3. **Loading:** The servlet class is loaded into the JVM. 4. **Instantiation:** The servlet instance is created. 5. **Initialization (jspInit):** Servlet initialization method is called once. 6. **Request Handling (_jspService):** Handles each client request. This is where the main JSP code runs. 7. **Destruction (jspDestroy):** Called once before servlet is unloaded. **Application:** Understanding these stages helps optimize JSP pages by: • Placing expensive initialization code in jspInit (runs once). • Minimizing code in the request handling part to improve response time. • Managing resources effectively using jspDestroy. • Avoiding frequent reloads by proper caching and compilation strategies.			
	c) Apply your understanding of the .NET Framework by identifying the features that support rapid web application development in ASP.NET Web Forms and MVC, and demonstrate how the page lifecycle differs when these two		CO5	L3

	approaches are used in real development scenarios. (8marks) Solution: **Key .NET Framework features for web development:** • **Common Language Runtime (CLR):** Manages execution of .NET programs, including memory management and security. • **Rich class libraries:** Extensive APIs for networking, data access, UI, and security. • **ASP.NET Web Forms:** Event-driven model that abstracts HTML supports drag-and-drop controls, state management via ViewState. • **ASP.NET MVC:** Follows Model-View-Controller architecture; offers full control over HTML, better for testability and clean separation of concerns. • **Built-in security and authentication mechanisms.** **Lifecycle differences:** • **Web Forms Lifecycle:** Complex, includes stages like Initialization, Load ViewState, Load PostBack Data, Event Handling, Rendering, and Unload. Supports server-side controls and ViewState for maintaining state between postbacks. • **MVC Lifecycle:** Simpler, request goes through Routing → Controller → Action Method → View → Response. No ViewState; developers handle state explicitly. More control and flexibility over HTML output. **OR** d) Illustrate the role of the form tag in ASP and how it supports server-side scripting in web applications. (8marks) Solution: • The <form> tag defines an HTML form for user input. • In ASP, the form must use method="post" or method="get" to send data to the server. • ASP scripts use Request.Form or Request.QueryString objects to access this data. • The form tag is crucial because it enables server-side			

	scripts to receive user input and process it, e.g., login credentials or search queries. Example: asp CopyEdit <form method="post" action="process.asp"> <input type="text" name="username" /> <input type="submit" /> </form> .			