**By Maha Rana**

# CS521 Finals Subjective Solved

## CS521 - Short Questions & Answers

### Short Questions

1. **Update SQL Query**

```
UPDATE Categories SET CategoryName = 'Web' WHERE
CategoryName = 'Business';
```

1. Reference: Week 11, Topic 91

2. **PHP Methods for:**
   - **Removing session variable:**

     ```
     session_unset(); // removes all session
     variables
     ```

   - **Destroying session:**

     ```
     session_destroy(); // ends the session
     ```

2. Reference: Week 13, Topic 114

3. **Using `$_FILES` in PHP**

   PHP uses the `$_FILES` array to handle file uploads. Key steps:

   - HTML form should use: `enctype="multipart/form-data"`

- Use `$_FILES['fileInputName']['name'], ['type'], ['tmp_name'], ['error'], ['size']` for file details
- Example:

```
if ($_FILES['myFile']['error'] ==
UPLOAD_ERR_OK) {

move_uploaded_file($_FILES['myFile']['tmp_name'
], 'uploads/' . $_FILES['myFile']['name']);
}
```

Reference: Week 10, Topic 84

4. **Benefits of AJAX**
   - Asynchronous communication with server
   - Reduces page reloads
   - Improves user experience
   - Allows partial page updates
     Reference: Week 8, Topic 72

5. **What are sockets and how they differ from traditional HTML communication**
   - **Socket**: Enables full-duplex communication between browser and server.
   - **Difference**:
     - Traditional HTTP: request-response model
     - Web Sockets: real-time, bi-directional communication
       Reference: Week 15, Topic 134

6. **Different Types of Form Validation**
   - **Client-side (JavaScript):**
     - Input format, required fields, data type checks
   - **Server-side (PHP):**
     - Required info, correct data type, range check, regex
   - **Using PHP filters**

```
filter_var($email, FILTER_VALIDATE_EMAIL);
```

7. Reference: Week 12, Topics 107–108

**By Maha Rana**

## Long Questions

1. **Benefits of using Transactions (with Example)**
   - **Benefits**:
     - Ensures **data integrity**
     - Allows **rollback** if an error occurs
     - Multiple queries act as one atomic operation
   - **Example**:

```
$pdo->beginTransaction();
$pdo->query("INSERT INTO Categories
(CategoryName) VALUES ('Philosophy')");
$pdo->query("INSERT INTO Categories
(CategoryName) VALUES ('Art')");
$pdo->commit();
```

     If error occurs, use `$pdo->rollback();`
     Reference: Week 11, Topic 96

---

2. **Syntax to Define Array in PHP + Types of Arrays**
   - **Indexed Array**:

```
$colors = array("Red", "Green", "Blue");
```

   - **Associative Array**:

```
$ages = array("Peter"=>35, "John"=>40);
```

   - **Multidimensional Array**:

```
$students = array(
  array("Ali", 20),
  array("Sara", 22)
);
```

3. Reference: Week 9, Topic 78

**3. Difference between File Handling in PHP and Node.js**

**By Maha Rana**

| Feature | PHP | Node.js |
| --- | --- | --- |
| Syntax | `fopen()`, `fgets()`, `fclose()` | `fs.readFile()`, `fs.writeFile()` |
| Blocking/Non-blocking | Blocking | Non-blocking (asynchronous) |
| File Upload Handling | Built-in with `$_FILES` | Requires libraries like `multer` |

Reference: Week 10 vs Week 15

4. **PHP Code for Adding Two Variables and Display Result**

```
<?php
$a = 10;
$b = 20;
$sum = $a + $b;
echo "The sum is: " . $sum;
?>
```

## Short Questions

1. **PHP: Class Creation**

```
class Artist {
    function __construct($firstName, $lastName, $city,
$birth, $death=null) {
        $this->firstName = $firstName;
        $this->lastName = $lastName;
        $this->birthCity = $city;
        $this->birthDate = $birth;
        $this->deathDate = $death;
    }
}
```

Then instantiate using:

```
$picasso = new Artist("Pablo", "Picasso", "Malaga",
"Oct 25,1881", "Apr 8,1973");
```

2. **JavaScript: File Include Syntax**

```
<script src="filename.js"></script>
```

Used in HTML to include JavaScript files.

3. **SQL Query with Greater Than Condition**

```
SELECT * FROM books WHERE price > 100;
```

This fetches all books with price greater than 100.

4. **Ways to Access Variables in JavaScript**

- Direct access: `var a = 10; console.log(a);`
- Object property: `obj.a`
- Array index: `arr[0]`
- DOM: `document.getElementById('id').value`

5. **Benefits of AJAX**

- No full page reload
- Asynchronous communication
- Faster response
- Improved user experience

6. **Node.js Usage**

- Runs JavaScript outside the browser
- Used for server-side development
- Non-blocking I/O

- Used with MongoDB, Express.js
  **Real-world use:** LinkedIn and PayPal adopted Node.js for performance and efficiency.

---

## ✅Long Questions

1. **Types of Input Validation**

- **Required Info**: Must fill fields
- **Correct Data Type**: Numbers, strings, emails
- **Correct Format**: Use of regex
- **Range Check**: e.g., age 18–60
- **Comparison**: password == confirm password
- **Custom Validation**: Based on logic
  Validation must happen **client-side (JS)** & **server-side (PHP)**

2. **PHP: Array Sum Using Loop**

```
<?php
$nums = array(10, 20, 30, 40);
$sum = 0;
foreach($nums as $value) {
    $sum += $value;
}
echo "Total sum is: " . $sum;
?>
```

3. **JavaScript: Event Listener for Message Print**

```
<button id="showMsg">Click Me</button>
<p id="msg"></p>

<script>
document.getElementById("showMsg").addEventListener("cl
ick", function() {
    document.getElementById("msg").innerText = "You
clicked the button!";
```

```
});
</script>
```

Event listener helps in separating structure and behavior.

4. **Define: Digital Signature**
   A **digital signature** is an electronic form of a signature that validates the authenticity and integrity of a digital message or document.

- Created using a private key
- Verified using the sender's public key
- Ensures **authentication**, **integrity**, and **non-repudiation**

➡ commonly used in **HTTPS**, **e-documents**, **email encryption**

## ✓Short Questions

1. **Node.js ki Working**

- Runs JavaScript outside the browser using **Chrome V8 engine**
- **Single-threaded, non-blocking I/O model**
- Used for **server-side development**
- Can handle thousands of requests simultaneously
  Example from slide:

```
var http = require('http');
var server = http.createServer(function (req, res) {
  res.writeHead(200, {"Content-Type": "text/plain"});
  res.write("Hello this is our first node.js
application\n");
  res.end();
});
server.listen(7000, "localhost");
console.log("server running at
http://127.0.0.1:7000/");
```

---

2. **Session Destroy and Session End Code (PHP)**

```
session_unset();     // Unsets session variables
session_destroy();   // Destroys the session
```

3. **AJAX k Benefits**

- Asynchronous data exchange without page reload
- Faster performance
- Partial page update
- Better user experience

4. **Class Create in PHP**

```
class Artist {
    function __construct($firstName, $lastName, $city,
$birth, $death = null) {
        $this->firstName = $firstName;
        $this->lastName = $lastName;
        $this->birthCity = $city;
        $this->birthDate = $birth;
        $this->deathDate = $death;
    }
}
$picasso = new Artist("Pablo", "Picasso", "Malaga",
"Oct 25,1881", "Apr 8,1973");
```

5. **SQL Query for Showing Data**

```
SELECT * FROM Categories ORDER BY CategoryName;
```

This fetches all records from `Categories` table.

6. **jQuery ko Webserver me Include karne ke 2 Methods**

1. **CDN** (external link):

```
<script src="https://code.jquery.com/jquery-
3.6.0.min.js"></script>
```

2. **Local file include**:

**By Maha Rana**

```
<script src="js/jquery.min.js"></script>
```

## ✅Long Questions

1. **File Handling Difference: PHP vs Node.js**

| Feature | PHP | Node.js |
|---|---|---|
| Access Type | Blocking I/O (fopen, fgets) | Non-blocking I/O ( `fs.readFile()` ) |
| File Upload | Built-in via `$_FILES` | Use middleware (e.g., Multer) |
| File Processing | Streams and full-memory access | Callback-based, asynchronous |
| Common Use | Easy file read/write for forms | Server-based file APIs |

2. **Validate Input and Alert if Empty**

```
<input type="text" id="inputuser">
<script>
function validate() {
  var val = document.getElementById("inputuser").value;
  if(val == "") {
    alert("Please insert the input");
  }
}
</script>
<button onclick="validate()">Check</button>
```

Client-side validation using JavaScript.

3. **PHP Code to Sum Array Elements**

```
<?php
$nums = array(10, 20, 30, 40);
$sum = 0;
foreach($nums as $value) {
    $sum += $value;
}
```

```
echo "Sum is: " . $sum;
?>
```

4. **Session Expiry Time One Day (PHP Code)**

```
// Must be set before session_start()
ini_set('session.gc_maxlifetime', 86400); // 86400
seconds = 1 day
session_set_cookie_params(86400);
session_start();
```

This ensures the session and cookie last one full day.

## ✅Short Questions

1. **SQL Query for Displaying Data**

```
SELECT * FROM Categories ORDER BY CategoryName;
```

This query fetches all categories sorted alphabetically.

2. **Difference: Valid XML vs Well-Formed XML**

- **Well-Formed XML**:
  - Follows syntax rules: single root element, properly nested tags, attributes in quotes, etc.
- **Valid XML**:
  - Must be **well-formed AND** conform to a **DTD or Schema**.

3. **jQuery `animate()`**

```
$("#box").animate({
  left: '250px',
  opacity: '0.5'
}, 1000);
```

Used to animate CSS properties like position, size, or opacity.

4. **PHP Class Creation**

```
class Artist {
  function __construct($firstName, $lastName) {
    $this->firstName = $firstName;
    $this->lastName = $lastName;
  }
}
$picasso = new Artist("Pablo", "Picasso");
```

5. **SEO and Its Techniques**

- **White Hat SEO**:
  - Title/meta-data
  - Alt tags, image names
  - Original content
  - Sitemap, good layout
- **Black Hat SEO**:
  - Keyword stuffing
  - Cloaking
  - Hidden links

6. **jQuery Introduction**

- A fast, lightweight JS library
- Simplifies DOM manipulation
- Syntax: `$(selector).action();`
- Example:

```
$(document).ready(function(){
  $("p").click(function(){
    $(this).hide();
  });
});
```

## ✅Long Questions (All Coding)

1. **Class Code in PHP**

```
class Student {
  public $name;
  function __construct($n) {
```

```
    $this->name = $n;
  }
  function showName() {
    echo "Name: " . $this->name;
  }
}
$std = new Student("Ali");
$std->showName();
```

2. **Event Listener Approach**

```
<button id="clickMe">Click Me</button>
<script>
document.getElementById("clickMe").addEventListener("cl
ick", function(){
  alert("Button clicked!");
});
</script>
```

Used for separating JavaScript logic from HTML.

3. **Create Class Button with Event Listener**

```
<button id="createBtn">Create Class</button>
<script>
document.getElementById("createBtn").addEventListener("
click", function(){
  class Person {
    constructor(name) {
      this.name = name;
    }
    show() {
      alert("Name: " + this.name);
    }
  }
  const p1 = new Person("Sara");
  p1.show();
});
</script>
```

4. **Setting Cookie with Expiry Time in PHP**

```
// setcookie(name, value, expiry_time, path)
setcookie("user", "Ali", time() + (86400), "/"); //
86400 = 1 day
```

Cookie will expire after 24 hours.

## ✅Coding-Based Questions

1. **Add Two Numbers in PHP**

```
<?php
$a = 10;
$b = 20;
$sum = $a + $b;
echo "Sum is: " . $sum;
?>
```

Basic addition example using PHP.

2. **`if` Statement in PHP**

```
<?php
$marks = 60;
if ($marks >= 50) {
  echo "Pass";
} else {
  echo "Fail";
}
?>
```

Conditional control is covered under Topic 74

3. **GET and POST in PHP**

```
// GET Method Example
<form method="GET" action="process.php">
  <input type="text" name="name">
  <input type="submit">
</form>
```

```
// POST Method Example
<form method="POST" action="process.php">
  <input type="text" name="email">
  <input type="submit">
</form>
```

Explained in Topic 81–82

4. **Validation (MCQs + Coding Concepts)**

- **Client-side (JavaScript)**
- **Server-side (PHP using `filter_var()`)**

```
$email = "abc@mail.com";
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
  echo "Invalid email format";
}
```

Covered in Topic 107

5. **XML Questions (MCQs & Valid/Well-formed)**

- **Well-formed XML**
  - Correct nesting, single root, closing tags
- **Valid XML**
  - Well-formed + conforms to DTD or Schema

```
<art>
  <painting id="1">
    <title>Balcony</title>

<artist><name>Manet</name><nationality>France</national
ity></artist>
    <year>1868</year>
    <medium>Oil on canvas</medium>
  </painting>
</art>
```

6. **SEO Techniques Question**

- **White Hat**:
  - Use of proper titles, meta tags, keyword density, sitemaps
- **Black Hat**:
  - Keyword stuffing, hidden text, cloaking

7. **PHP Array and Sum**

```
<?php
$numbers = array(10, 20, 30, 40);
$sum = 0;
foreach ($numbers as $num) {
  $sum += $num;
}
echo "Total Sum: $sum";
?>
```

Covered in Topic 78

8. **jQuery `.animate()` Example**

```
$("#box").animate({
  left: '250px',
  opacity: '0.5'
}, 1000);
```

Used for DOM animations — covered in Topic 72

**✅Short Questions**

1. **SEO Techniques**
   **White Hat SEO:**

- Title and meta-data
- Original content
- Descriptive URLs
- Image alt text
- Sitemap and proper layout

**Black Hat SEO:**

- Keyword stuffing
- Cloaking
- Hidden links
- Link farming

2. **AJAX Properties**

- Asynchronous: No page reload
- XMLHttpRequest: Behind the scenes
- Partial page updates
- Faster performance

3. **Form Validation Names (PHP filters)**

- `FILTER_VALIDATE_EMAIL`
- `FILTER_VALIDATE_URL`
- `FILTER_VALIDATE_INT`
- `FILTER_VALIDATE_REGEXP`
- `FILTER_SANITIZE_STRING`, etc.

4. **SQL Query**

```
SELECT * FROM Categories ORDER BY CategoryName;
```

Used to fetch all records from the `Categories` table sorted alphabetically

5. **Use of `$_FILES` in PHP**

- Handles file uploads via POST method
- Must use `enctype="multipart/form-data"` in `<form>`
- Access file info like:

```
$_FILES['file']['name']
$_FILES['file']['tmp_name']
$_FILES['file']['error']
```

- Validate type and move file using `move_uploaded_file()`

**✅Long Questions**

**By Maha Rana**

1. **PHP Code Example**

```
<?php
$a = 10;
$b = 20;
$sum = $a + $b;
echo "Sum is: $sum";
?>
```

Simple addition using PHP variables

2. **JavaScript Event Handler**

```
<button id="btn">Click Me</button>
<script>
document.getElementById("btn").addEventListener("click"
, function() {
  alert("Button clicked!");
});
</script>
```

Used to handle DOM events in JS

3. **Concept of Transactions in PHP**

```
$pdo = new PDO($connString, $user, $pass);
try {
    $pdo->beginTransaction();
    $pdo->query("INSERT INTO Categories (CategoryName)
VALUES ('Philosophy')");
    $pdo->query("INSERT INTO Categories (CategoryName)
VALUES ('Art')");
    $pdo->commit();
} catch (Exception $e) {
    $pdo->rollback();
}
```

Used to ensure data integrity — all or nothing execution of multiple SQL statements

4. **Convert XML to JSON (Conceptual PHP Code)**
   Assume we have an XML file:

```
<student>
  <name>Ali</name>
  <age>22</age>
</student>
```

Conversion in PHP:

```
$xml = simplexml_load_file("student.xml");
$json = json_encode($xml);
echo $json;
```

This turns XML data into JSON format using `simplexml_load_file()` and `json_encode()`

5. **Difference between web Sockets and traditional http?**

| Feature | Traditional HTTP | WebSockets |
|---|---|---|
| Communication Type | Request-response | Full-duplex (bi-directional) |
| Connection | Opens and closes with every request | Stays open after handshake |
| Use Case | Static websites, form submissions | Real-time apps like chat, games |
| Latency | Higher, as connection resets each time | Lower, constant open connection |

📘 **Slide Reference**: Week 15, Topic 134 – WebSocket vs Traditional HTTP

## ✅Long Questions – Coding

### 1PHP: Add Two Numbers and Display Result

```
<?php
$num1 = 15;
$num2 = 25;
$sum = $num1 + $num2;
echo "Sum is: $sum";
?>
```

 **Slide Reference**: PHP Basics – Week 9

**2JavaScript: Show Message on Mouse Click**

```
<button id="clickBtn">Click Me</button>

<script>
document.getElementById("clickBtn").addEventListener("c
lick", function() {
  alert("Hello! Button was clicked.");
});
</script>
```

 **Slide Reference**: Event Listener Approach – Week 7

**By Maha Rana**

**May Allah grant you success, ease your efforts, and bless you with wisdom and perseverance. Keep your faith strong, for with prayer, every challenge becomes an opportunity. And please remember me in your prayers.**

**I hope this file helps you a lot; I have tried my best to find the most relevant and authenticated answers for the questions. However, in some cases, the questions might slightly differ from the ones in your slides. I recommend that every student kindly verify the answers on their own before using them. I am not responsible for any discrepancies. Best wishes for Finals!**