Department of Computing and Information Systems (CIS)
Web Engineering Lab – Final Examination
Instructor: Md. Moniruzzaman Hemal **SOLUTION MANUAL** **Total Marks: 40**

SET A - Answer Key

Q1. In the Model-View-Controller framework, how does the View fundamentally differ from the Model?
Answer: The View is the presentation layer (the HTML interface) and contains zero business or database logic; it is completely "dumb". The Model handles all database connections, executes SQL queries, and applies the application's business rules.

Q2. In PHP, which specific operator is used to concatenate variables and strings together?
Answer: The dot (`.`) operator is used for concatenation in PHP.

Q3. When using the `mysqli_prepare()` function, what is the critical security purpose of using `?` placeholders?
Answer: The `?` placeholders prevent SQL Injection attacks by ensuring that user inputs are treated strictly as data rather than executable SQL commands.

Q4. What happens on the server side if a user submits a form where an HTML checkbox is left completely unchecked?
Answer: The browser does not send that variable to the server at all. Attempting to access it directly via `$_POST` without checking first will trigger an "Undefined Index" warning.

Q5. When building a REST API, why must developers utilize `file_get_contents('php://input')` instead of the standard `$_POST` array to capture incoming JSON payloads?
Answer: The `$_POST` superglobal is designed exclusively to read data from standard HTML forms. `php://input` acts as a raw stream that catches JSON payloads sent by mobile apps or modern front-end frameworks.

Q6. When checking form submissions, what exactly does the PHP `isset()` function verify?
Answer: The `isset()` function checks whether a specific variable (such as a form input or button click) has been declared and is not null.

Q7. How can you dynamically inject a PHP variable, such as `$bg_color`, directly into an HTML inline style attribute?
Answer: By embedding PHP echo tags directly inside the HTML attribute quotes. Example: `<html style="background-color: <?php echo $bg_color; ?>;">`.

Q8. What are the specific parameters required by `mysqli_connect()` to link to a database? Additionally, what is the difference between `mysqli_connect_error()` and `mysqli_error()`?
Answer:

- **Parameters:** Server name, username, password, and database name.
- **Difference:** `mysqli_connect_error()` catches failures during the initial login phase. `mysqli_error()` retrieves specific error messages if an SQL query fails *after* connecting.

Q9. In REST APIs, what does the HTTP status code `201` explicitly signify compared to `200`?
Answer: `200 OK` means the request was successful. `201 Created` specifically signifies that a request (usually `POST`) was successful and a brand-new resource was created on the server.

Q10. Why is the `true` parameter critical when using `json_decode()` to process JSON for MySQL database insertion?
Answer: Passing `true` forces the function to translate the JSON into a PHP Associative Array (e.g., `$input['name']`), which is necessary for array-based backend database logic. Without it, it becomes a PHP Object.

Q11. If a developer runs an `UPDATE` or `DELETE` SQL query but forgets the `WHERE id=$id` clause, what catastrophic event occurs?
Answer: The database applies the command universally, updating or deleting every single record in the entire table.

Q12. How does the `die()` function differ from `mysqli_close()`?
Answer: `die()` is an emergency stop that prints an error message and immediately halts the entire PHP script. `mysqli_close()` gracefully closes the database connection at the end of a successful script.

Q13. How does `mysqli_fetch_assoc()` behave when called repeatedly inside a loop?
Answer: It acts like pulling papers from a stack. Each iteration pulls the next single row of data from the result set and converts it into an associative array until no rows remain.

Q14. Between `PUT` and `POST`, which HTTP verb is mathematically considered "Idempotent"? Compare `PUT` VS `PATCH`.
Answer: `PUT` is mathematically idempotent. **Comparison:** `PUT` replaces the entire resource completely. `PATCH` applies partial modifications to specific fields of an existing resource.

Q15. What specific visual rendering issue does the `border-collapse: collapse;` CSS property solve?
Answer: It solves the "double lines" issue in HTML tables, merging the table boundary and individual cell boundaries into one single, clean border line.

Q16. [**10 Marks**] Write a complete PHP script that accepts a positive integer from a variable `$num`. Your script must first mathematically reverse the number, and then logically check if this **reversed number** is a **Prime Number**. Print an appropriate message.
Answer:

```
<?php
$num = 123;

$rev = 0;
while ($num > 0) {
    $rev = $rev * 10 + $num % 10;
    $num = (int)($num / 10);
}

$prime = 1;
for ($i = 2; $i < $rev; $i++) {
    if ($rev % $i == 0) {
        $prime = 0;
        break;
    }
}

echo "Reversed: $rev <br>";
echo ($prime && $rev > 1) ? "Prime" : "Not Prime";
?>
```

Department of Computing and Information Systems (CIS)
Web Engineering Lab – Final Examination
Instructor: Md. Moniruzzaman Hemal **SOLUTION MANUAL** **Total Marks: 40**

SET B - Answer Key

Q1. Why must the `session_start()` function be written as the very first line of a PHP file before any other operations?
Answer: It must be called before any HTML or output is printed. If the server has already sent HTML headers to the browser, it cannot send the session cookie headers, resulting in a "Headers already sent" fatal error.

Q2. What is the "Golden Rule" mapping an HTML form input to its corresponding PHP `$_POST` array key?
Answer: The specific string defined in the HTML `name` attribute directly dictates the exact array key used in PHP (e.g., HTML `name="fname"` maps to PHP `$_POST['fname']`).

Q3. In a REST API, when you attempt to send JSON data back to a client, what specific header must be sent?
Answer: `header('Content-Type: application/json');`. Without it, the browser defaults to `text/html` and won't process the payload as a JSON object.

Q4. How can you dynamically inject a PHP variable, such as `$bg_color`, directly into an HTML inline style attribute?
Answer: By embedding PHP echo tags inside the HTML attribute.
Example: `<html style="background-color: <?php echo $bg_color; ?>;">`.

Q5. If a user cannot see or edit a hidden input field (`type="hidden"`), what is its practical purpose?
Answer: It securely passes required backend tracking data (such as User IDs or Database row IDs) along with a form submission without cluttering the user interface.

Q6. What specific array variable does PHP use to read data that was appended to the URL as query parameters?
Answer: The `$_GET` superglobal array.

Q7. How do you print out the specific MySQL error message if an insertion query fails in PHP?
Answer: You concatenate the `mysqli_error($conn)` function to your error output string to retrieve the exact backend rejection reason.

Q8. What is the functional purpose of the `target="_blank"` attribute when creating anchor (`<a>`) links?
Answer: It forces the browser to open the linked document in a completely new tab or window, preventing the user from leaving your current web application.

Q9. Why does an API developer use a tool like Postman instead of a normal web browser to test `POST` and `PUT` requests?
Answer: Standard web browser address bars only trigger `GET` requests. Postman allows developers to select specific verbs (`POST`, `PUT`) and manually attach raw JSON payloads to the body.

Q10. In the MVC framework, what is the primary role of the Model?
Answer: The Model acts as the application's logic layer. It communicates securely with the database, evaluates business rules, formats SQL queries, and returns validation responses to the Controller.

Q11. Compare `PUT` & `PATCH`.
Answer: `PUT` is used to replace an entire resource completely. `PATCH` is used to apply partial modifications, updating only the specific fields provided in the payload.

Q12. What do the HTTP status codes `401` and `403` respectively indicate?
Answer: `401 Unauthorized` means the client is not logged in. `403 Forbidden` means the client is logged in, but lacks the necessary admin permissions to access the endpoint.

Q13. Why does the `colspan` attribute exist in HTML tables, and how does it manipulate cells?
Answer: It allows a single table cell (`<td>` or `<th>`) to merge horizontally, stretching its width across the space of multiple columns.

Q14. Describe how the `mysqli_fetch_assoc()` function retrieves data row by row from a `SELECT` query result.
Answer: It pulls the next single row of data from the result set and converts it into an associative array. When called repeatedly in a loop, it pulls row after row until none remain.

Q15. When querying a database within the Model layer of an MVC application, what is the specific security purpose of using `?` placeholders with `mysqli_prepare()`?
Answer: The `?` placeholders prevent SQL Injection attacks by isolating user input from the executable SQL command.

Q16. [**10 Marks**] Write a complete PHP script that accepts a positive integer from a variable `$num`. Use a loop to reverse the number entirely, and then logically check if the original number is a **Palindrome**. Print an appropriate message.
Answer:

```
<?php
$num = 121; // Example input
$temp = $num;
$rev = 0;

// 1. Reversing the number mathematically
while ((int)$temp > 0) {
    $rem = $temp % 10;
    $rev = ($rev * 10) + $rem;
    $temp = (int)($temp / 10);
}

// 2. Checking if the original number matches the reversed number
if ($num == $rev) {
    echo "Reversed is $rev. $num is a Palindrome.";
} else {
    echo "Reversed is $rev. $num is NOT a Palindrome.";
}
?>
```