INSTRUCTION ON SUBMISSION OF ASSIGNMENT

- This assignment is worth 20 marks of the Final Term Grade
- This assignment is the replacement of Central Quiz
- This assignment will help you prepare for the final exam
- **The submission should be handwritten**
- **Use A4 papers (write on both sides)**
- **Use the official cover page for AIUB Assignments (Found in AIUB Portal) as the first page**
- Keep the answers as short as possible
- Answer all questions with proper explanations
- For coding questions, write complete, working code with comments
- Partial credit will be given for partially correct answers with proper justification
- Any sort of plagiarism, if detected, will render your submission invalid and you will be marked 0
- You may use external resources and AI tools to help with the assignment, but make sure you understand the solution

DEADLINE: 11th January (Sunday) – 12 PM (on the day of the Final Exam)

(*No submission will be accepted after deadline. Early submissions are welcome.***)**

Submit the assignment document in my office (If I am not available keep it on the floor)

Office room no: **DN0227C**

The next pages contain the questions…

Question 1
Consider an e-commerce application where users can search for products, add them to cart, and checkout. For each of the following operations, identify which MVC component(s) would be primarily responsible and explain why:

a) Calculating the total price including tax and discounts
b) Rendering the shopping cart page with product thumbnails
c) Processing the "Add to Cart" button click
d) Querying the database for products matching search criteria
e) Validating that quantity is a positive integer before updating cart

Question 2
Design a mini-application using MVC pattern:

Scenario: A student grade management system where teachers can:

- View all students in a class
- Add a new grade for a student
- Update an existing grade
- Calculate and display class average

Write pseudo-code or actual PHP code showing:

- The Model class with at least 3 methods
- The Controller logic for handling "add grade" request
- Explain what the View would display (you don't need to write full HTML)

Question 3
Explain the concept of "separation of concerns" in MVC. Why is it problematic to write database queries directly in your HTML view files? Provide a specific example of how mixing these concerns could cause maintenance issues in a real project.

Question 4
Write complete HTML and JavaScript code for the following scenario:

Create a "Real-time Comment Preview" feature where:

- User types a comment in a textarea
- As they type, the comment is sent to a PHP script (`preview.php`) that processes it (e.g., converts URLs to links, escapes HTML)
- The processed preview appears below the textarea in real-time
- Use AJAX to send data every 2 seconds (implement debouncing)
- Display a loading indicator while waiting for response

Your code must include:

- Complete HTML form structure
- JavaScript with proper XMLHttpRequest handling
- All necessary event handlers
- Error handling for failed requests

Question 5
Analyze the following code and answer the questions:

```
function checkData() {
    var xhr = new XMLHttpRequest();
    xhr.open("POST", "validate.php", true);
    xhr.onreadystatechange = function() {
        if (xhr.readyState == 4 && xhr.status == 200) {
            var response = xhr.responseText;
            document.getElementById("result").innerHTML = response.message;
        }
    };
    xhr.setRequestHeader("Content-Type", "application/json");
    xhr.send(JSON.stringify({username: "john", age: 25}));
}
```

a) Identify the bug in this code and explain why it will cause an error
b) What would happen if we forgot to set the Content-Type header?
c) Rewrite the code to include proper error handling for network failures
d) Explain what `readyState` values 0, 1, 2, and 3 represent

Question 6
Compare and contrast AJAX vs traditional form submission. Create a table showing at least 5 differences, then describe two scenarios where traditional form submission would be MORE appropriate than AJAX, with justification.

Question 7
Write complete PHP code for a "Shopping Cart" system with the following requirements:

- Users can add items (name, price, quantity) to cart
- Cart data must persist across page reloads
- Implement these functions:
  - `addToCart($itemName, $price, $quantity)`
  - `removeFromCart($itemName)`
  - `updateQuantity($itemName, $newQuantity)`
  - `getCartTotal()`
  - `clearCart()`
- Use sessions appropriately
- Include proper session initialization and data validation

Question 8
Explain the following security concepts:

a) What is session hijacking and how does it work?
b) How does setting `HttpOnly` flag on cookies help prevent XSS attacks?
c) Why should you never store passwords directly in cookies or sessions?
d) Write the PHP code to create a secure cookie that:

- Expires in 30 days
- Is accessible only via HTTP (not JavaScript)
- Is only sent over HTTPS connections
- Is restricted to the `/admin` path

Question 9
Trace the following code and explain what happens at each visit:

```
session_start();
if (!isset($_SESSION['visit_count'])) {
    $_SESSION['visit_count'] = 1;
    setcookie('first_visit', date('Y-m-d H:i:s'), time() + 86400);
} else {
    $_SESSION['visit_count']++;
}

if (isset($_COOKIE['first_visit'])) {
    echo "First visited: " . $_COOKIE['first_visit'];
}
echo " Total visits this session: " . $_SESSION['visit_count'];
```

What will be displayed on:

- First visit?
- Second visit (same browser session)?
- Third visit (after closing browser and reopening)?
- Visit after 2 days?

Question 10

Predict the output of the following code snippets. Show your reasoning:

```
// Snippet 1
$x = null;
$y = "";
$z = 0;
echo ($x ?? "A") . " ";
echo ($y ?? "B") . " ";
echo ($z ?? "C") . " ";
echo ($w ?? "D");
// Snippet 2
$fruits = ["apple", "banana", "cherry", "date"];
array_pop($fruits);
$fruits[] = "elderberry";
echo count($fruits) . " " . $fruits[2];
// Snippet 3
$text = "  Hello World  ";
$result = trim($text);
echo strlen($result) . " vs " . strlen($text);
// Snippet 4
$str = "abcdefgh";
echo substr($str, -3, 2) . " ";
echo substr($str, 2, -2);
```

Question 11

Explain the differences between `include`, `include_once`, `require`, and `require_once`. For each of the following scenarios, which one would you use and why?

a) Loading a configuration file that defines database credentials
b) Loading a utility functions file that might be needed multiple times in different included files
c) Loading a template file for page header that should appear exactly once
d) Loading an optional analytics script that won't break the site if missing

Question 12

Write a function `processUserData($data)` that:

- Accepts an associative array with keys: 'name', 'email', 'age', 'hobbies' (array)
- Validates that name is not empty and contains only letters and spaces
- Validates that email is valid format
- Validates that age is between 13 and 120
- Returns an array with 'valid' (boolean) and 'errors' (array of error messages)
- If valid, also return a 'formatted' key with a string like: "Name: John (25 years) - Email: john@example.com - Hobbies: reading, gaming"

Question 13
Compare MySQLi and PDO by creating a detailed comparison table covering:

- Database support
- Parameter binding syntax
- Error handling approaches
- Prepared statement syntax
- Transaction handling

Then write the SAME database operation (inserting a user with username, email, and password) using both MySQLi and PDO with prepared statements.

Question 14
Analyze this vulnerable code:

```
$username = $_POST['username'];
$password = $_POST['password'];
$query = "SELECT * FROM users WHERE username='$username' AND password='$password'";
$result = mysqli_query($conn, $query);
```

a) Show how an attacker could exploit SQL injection here
b) Rewrite this code securely using prepared statements
c) Explain why prepared statements prevent SQL injection