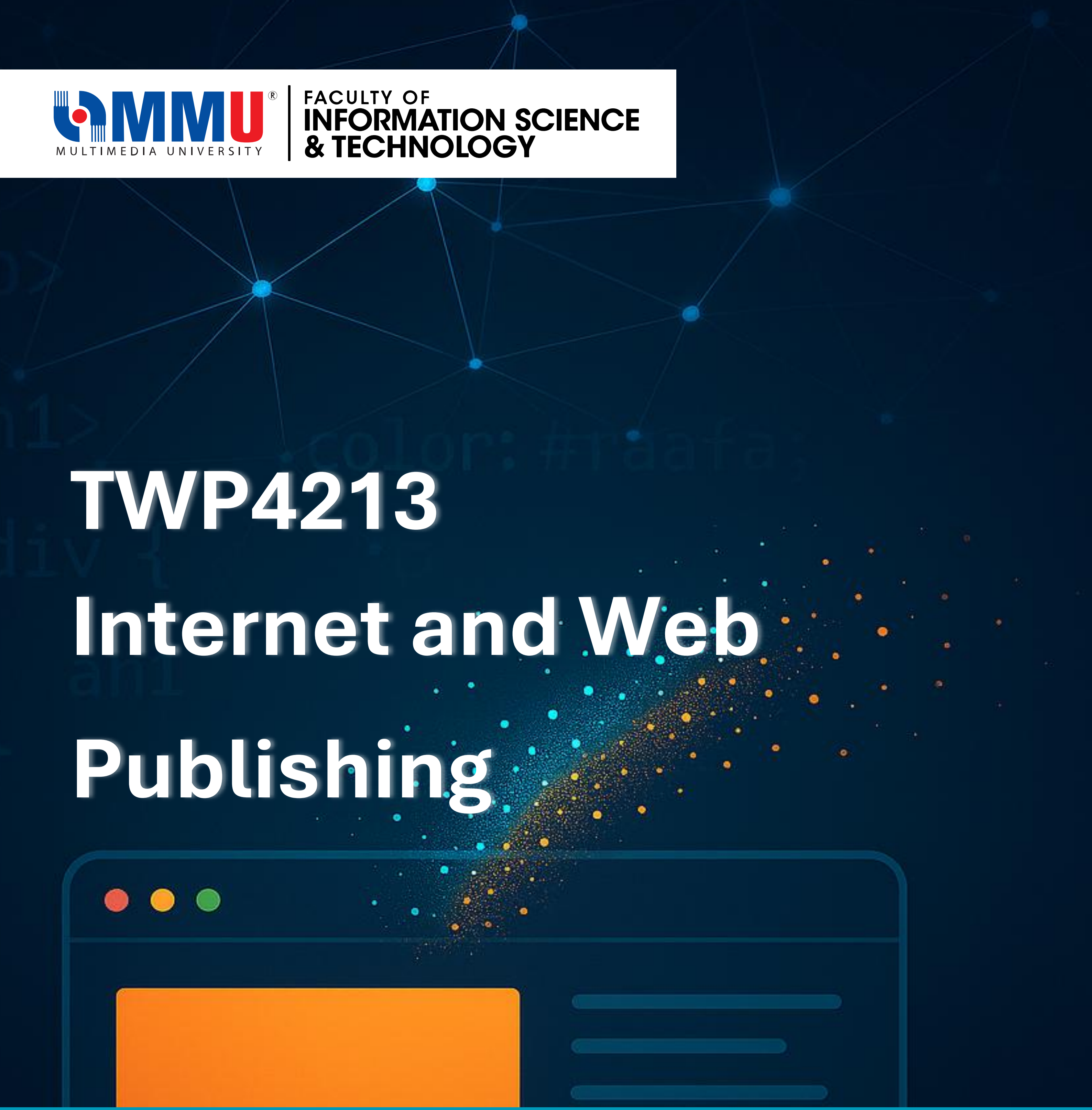
FACULTY OF
INFORMATION SCIENCE
& TECHNOLOGY



TWP4213 Internet and Web Publishing

Lab 09

PHP Working with Database

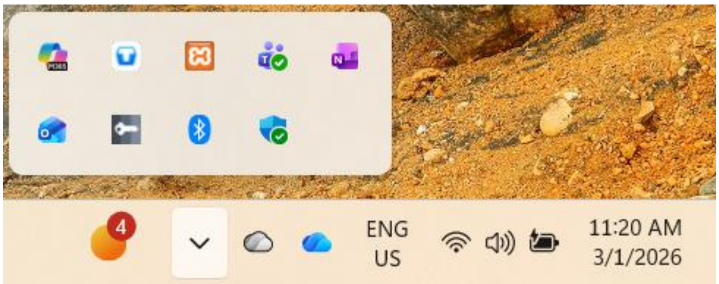
Question 1 – Configuring XAMPP

In this exercise, you will be able to:

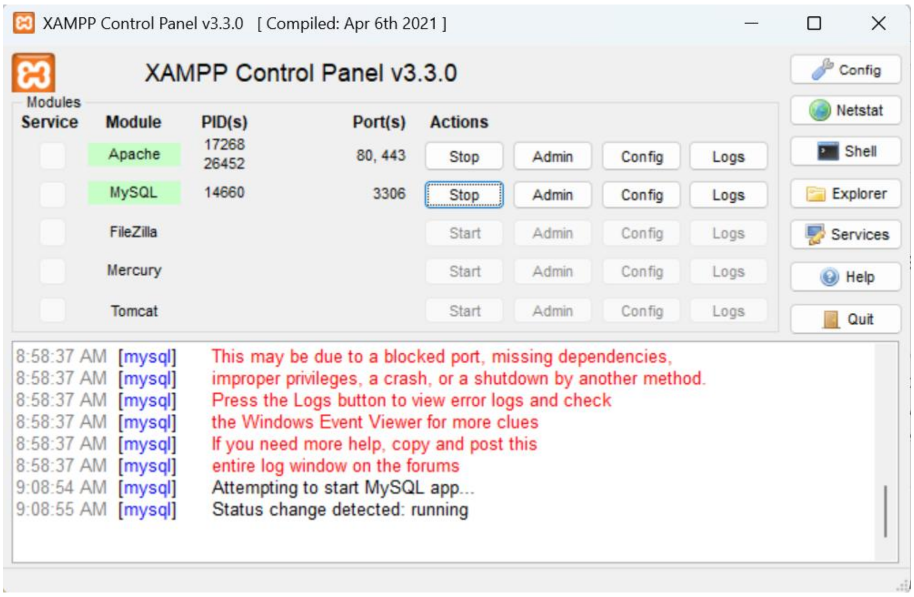
- Implement prepared statements to retrieve aggregate value.
- Build a server-side filtering mechanism using GET parameters to search products.
- Join related tables and render results securely in an HTML table

Follow the following steps to complete the code.

a) Check the taskbar and ensure that XAMPP is running. If XAMPP icon is there just double click the icon to open it. Do not run more than one XAMPP instances. This can lead to corruption of XAMPP files. If there is no XAMPP icon than use the Search in the taskbar to find and run XAMPP.



b) In the XAMPP Control Panel, make sure both Apache and MySQL are running.


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

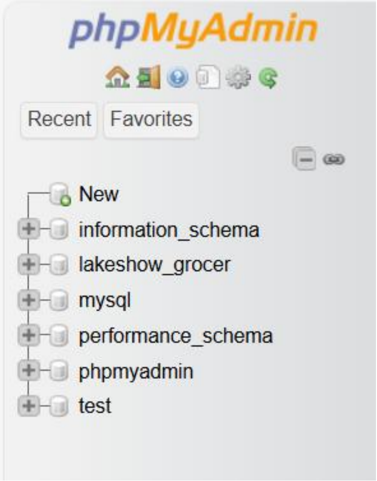
Service	Module	PID(s)	Port(s)	Actions			
	Apache	17268 26452	80, 443	Stop	Admin	Config	Logs
	MySQL	14660	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
8:58:37 AM  [mysql]   This may be due to a blocked port, missing dependencies,
8:58:37 AM  [mysql]   improper privileges, a crash, or a shutdown by another method.
8:58:37 AM  [mysql]   Press the Logs button to view error logs and check
8:58:37 AM  [mysql]   the Windows Event Viewer for more clues
8:58:37 AM  [mysql]   If you need more help, copy and post this
8:58:37 AM  [mysql]   entire log window on the forums
9:08:54 AM  [mysql]   Attempting to start MySQL app...
9:08:55 AM  [mysql]   Status change detected: running
```

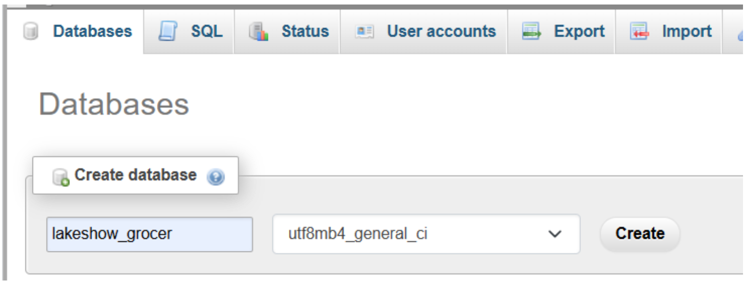
c) Download Lab 09 materials zip file and extract the content.

d) Open phpMyAdmin by clicking the **Admin** button for MySQL in the XAMPP control panel.

e) Create a new database call **lakeshow_grocer**.



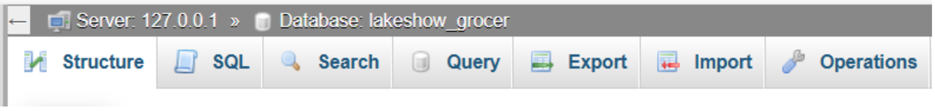
S



lakeshow_grocer

utf8mb4_general_ci

f) Once created, click the lakeshow_grocer database and click the Import tab.



g) Choose the **lakeshow_grocer.sql** file to import and click the import button. This will import all the table structures and data necessary for the lab exercise. Study the table structure especially the table names and column names.

h) Open Lab 09 Q1.php. This is an admin page where admin can view all the products stored in the database. As admin, we need quick information about the status of all products.

i) There are three critical information that will be displayed for the admin. The code is given for counting the total number of active products but for the other two, you need to write the code on your own. Do not use the same variable name.

Note that the connection code is stored in **config.php** file inside the **includes** folder.

- Total number of active products

```
$countproductsql = "SELECT COUNT(*) FROM product WHERE status = 'active'";
$stmtcount = $conn->prepare($countproductsql);
$stmtcount->execute();
$stmtcount->bind_result($totalActiveProducts);
$stmtcount->fetch();
$stmtcount->close();
```

- Total number of inactive Products (write your own code)

- Low stock Products (stock quantity of less than 10) (write your own code)

j) For the search or filter, we need to populate the dropdown list from category name stored in the database. Write the following code.

```
$categoriesQuery = "SELECT * FROM category ORDER BY Category_Name";
$categoriesResult = $conn->query($categoriesQuery);

while ($category = $categoriesResult->fetch_assoc()): ?>
<option& value="
<?php echo htmlspecialchars($category['Category_Name']); ?>" >
<?php echo htmlspecialchars($category['Category_Name']); ?>
</option>
<?php endwhile; ?>
```

k) To display the product, write the following code inside the <tbody> element.

```
$filter = "";
$productsQuery = "SELECT product.*, category.Category_Name FROM product
INNER JOIN category
ON product.Category_ID = category.Category_ID
WHERE 1" . $filter;

$productsQuery .= " ORDER BY product.Product_ID";
$productsResult = $conn->query($productsQuery);

while($product = $productsResult->fetch_assoc()): ?>
     <tr>
         <td><?php echo $product['Product_ID']; ?></td>
         <td><?php if ($product['Product_Picture']): ?>
             <img src="uploads/products/
             <?php echo $product['Product_Picture']; ?>"
             alt="Product Image" class="product-thumbnail">
             <?php else: ?>
             <div class="no-image">No image</div>
             <?php endif; ?>
         </td>
         <td><?php echo $product['Product_Name']; ?></td>
         <td>RM <?php echo number_format($product['Product_Price'],
2); ?></td>
         <td><?php echo $product['Product_Stock']; ?></td>
         <td><?php echo htmlspecialchars($product['Category_Name'] ??
'Uncategorized'); ?></td>
         <td><a href="EditProduct.php?id=
              <?php echo $product['Product_ID']; ?>"
              class="btn btn-edit" style="text-decoration: none;">
<i class="fas fa-edit"></i> Edit</a>

<form method="POST" action="UpdateProductStatus.php" style="display:
inline;">
<?php if ($product['status'] === 'Active'): ?>
<input type="hidden" name="product_id"
value="<?php echo $product['Product_ID']; ?>">
<input type="hidden" name="inactive_product" value="Inactive">
<button type="submit" class="btn btn-delete"
onclick="return confirm('Are you sure you want to mark this product as
inactive?');">
<i class="fas fa-trash"></i> Make Inactive</button>

<?php else: ?>
<input type="hidden" name="product_id" value="<?php echo
$product['Product_ID']; ?>">
<input type="hidden" name="active_product" value="Active">
<button type="submit" class="btn btn-primary" onclick="return confirm('Are
you sure you want to restore this product to be active?');">
<i class="fas fa-undo"></i> Activate Product</button>
<?php endif; ?>
</form>
</td>
</tr>
<?php endwhile; ?>
```

l) To make the search function work, we need to add additional criteria to where clause based on search criteria that user chose. User can choose based on product name, low stock or based on category. User can choose any combination of these three criteria. Code will be given to add search functionality based on stock level and based on product name as shown below. Add the code just below the **$filter** variable initialization.

```
if (isset($_GET['stock']) && !empty($_GET['stock'])) {
   if ($_GET['stock'] == 'low') {
       $filter .= " AND product.Product_Stock < 10";
   } elseif ($_GET['stock'] == 'out') {
       $filter .= " AND product.Product_Stock = 0";
   } elseif ($_GET['stock'] == 'in') {
       $filter .= " AND product.Product_Stock >= 10";
   }
}

 if (isset($_GET['productname']) && !empty($_GET['productname'])) {
    $filter .= " AND product.Product_Name LIKE
               '%".$_GET['productname']."%'";
                        }
```

m) Write your code to add search functionality based on category name.

Question 2 - Update product status

In this exercise, you will be able to:

- Implement product status changes securely with prepared statements.
- Redirect with user feedback using query strings.

Follow the following steps to complete the web page.

a) Create a new file called **UpdateProductStatus.php** file.

b) Include the config file.

c) Copy the following PHP code to check whether Make Inactive button is being submitted. If yes, then use a prepared statement to update the product status to Inactive. Then it go back to the previous file passing a querystring to show an alert message to the user.

```
if ($_SERVER['REQUEST_METHOD'] === 'POST' &&
isset($_POST['inactive_product'])) {
    $productId = $_POST['product_id'];

    $updateStmt = $conn->prepare("UPDATE product SET status = 'inactive'
WHERE Product_ID = ?");
    $updateStmt->bind_param("i", $productId);

    if ($updateStmt->execute()) {
         header("Location: Lab 09 Q1.php?message=deactivated");;
    } else {
         header("Location: Lab 09 Q1.php?message=error");r;
    }
    exit();
}
```

d) Write a similar code to change the product status to active.

e) Open the Lab 09 Q1.php file and add the following code to display the alert message. Modify the code so that it will show a proper alert message when the product has been deactivated or if there is an error.

```
<?php if (isset($_GET['message'])): ?>
    <div class="alert alert-success" id="successAlert">
       <?php if ($_GET['message'] == "activated"):
        echo "The product status has been activated";

?>
        </div>

    <?php endif; ?>
    </div>

<?php endif; ?>
```

Question 3 - Add Product

In this exercise, you will be able to:

- Check for duplicates using a parameterized SELECT and handle conditional logic to prevent duplicate product names.
- Insert new products with properly typed parameters using **bind_param()** and handle success/failure redirects.

Follow the following steps to complete the web page.

a) Create a new file called **AddProduct.php** file.

b) Include the config file.

c) Copy the following PHP code to check whether Add Product button is being submitted. The remaining code will be written inside this code block.

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {


}
```

d) Use the following code to check for any duplicate in the product name.

```
    $productname = trim($_POST['product_name']);

    $checkQuery = "SELECT Product_ID FROM product WHERE Product_Name = ?";
    $checkStmt = $conn->prepare($checkQuery);
    $checkStmt->bind_param('s', $productname);
    $checkStmt->execute();
    $checkStmt->bind_result($checkResult);
    $checkStmt->close();

    if ($checkResult > 0) {
        $nameError = 'A product with this name already exists.';

    }
```

e) If the name of the product has not been used that we can insert the product data in the database.

```
else {

        $categoryid = $_POST['category_id'];
        $price = $_POST['price'];
        $stock = $_POST['stock'] ;
        $description = trim($_POST['description']);

        $query = "INSERT INTO product (Product_Name, Category_ID,
Product_Price, Product_Stock, Product_Description, status)
                  VALUES (?, ?, ?, ?, ?, 'Active')";
        $addproductstmt = $conn->prepare($query);
        $addproductstmt->bind_param('sidis',
        $productname, $categoryid, $price, $stock, $description);



        if ($addproductstmt->execute()) {
            header("Location: Lab 09 Q1.php?message=success");
            exit();
```

```
                } else {
                    header("Location: Lab 09 Q1.php?message=error");
                    exit();
                }
                $addproductstmt->close();
            }
        }
```