Name: **FRANCHESKA FAITH B. BATALLA** Subject: **WEB SYSTEMS DEVELOPMENT**

Course/Year: **BSIS 2-A** Activity: **TROUBLE SHOOTING**

SCENARIO 1: Using $_POST instead of $_GET

Problem: Undefined index if no POST request; trying to read ID from URL with POST.
Solution: Use $_GET['id'] instead of $_POST['id'].

Explanation: The script failed because "?id=3" is coming from the URL and can be accessed using $GET not $POST. Kaya instead of using $_POST in line 3, it should be $GET.
There was also an incorrect column name in line 5 "id". It should be "student_id" cause that is what's in the database.

SCENARIO 2: Missing quotes in SQL when using POST

Problem: SQL error Unknown column 'Ana'; $fname not quoted.
Solution: Add quotes around the value

Explanation: Ang mga string values dapat ay enclosed sa quotes otherwise SQL will be interpreting the text as a column name and will result to "unknown column" kasi wala naman sa database.

SCENARIO 3: SQL injection vulnerability

Problem: Users can input 1 OR 1=1 and retrieve all records
Solution: Use prepared statements

Explanation: Using prepared statements are better, it ensures safe and tama yung input before running because if the value is directling nilagay sa query ay pwedeng maging risky for the SQL injection

SCENARIO 4: Forgetting to validate empty post field

Problem: Blank rows inserted if form is empty
Solution: Validate input before inserting

Explanation: If there are blank data to submit it could mess up the database. It is important to always check or validate para maiwasan yung errors

SCENARIO 5: Wrong key name in POST

Problem: Typo $_POST['emial'] → undefined index.
Solution: Correct the variable name

Explanation: Super strict ang naming, if it is in upper case or upper case and spelling it must be the same, it is it's unique name. Since mali yung spelling nung POST key, nagkaroon ng error and di makuha ning PHP ang value kaya undefined index

SCENARIO 6: Unsafe direct use of GET in DELETE

Problem: Unsafe delete, user can delete everything
Solution: Convert ID to an integer or use prepared statement

Explanation: Using raw GET value can enable user to delete all records. Using integers can limit kung ano lang pwede madelete nung user.

SCENARIO 7: Query fails but script continues

Problem: SQL error occurs, but "Updated!" still prints and missing quotes.
Solution: Add quotes and check if the query succeded

Explanation: Palaging icheck if the query succeeded para may proper report at maiwasan yung mga errors.

SCENARIO 8: Missing mysqli_fetch_assoc loop

Problem: The first record is the only one that prints.
Solution: Loop through the results

Explanation: Unless ilagay sa loop, once lng magafetch ang mysqli_fetch_assoc kaya need i-loop para lumabas lahat ng records.

SCENARIO 9: Using GET but link send POST

Problem: Link send GET but PHP uses POST
Solution: Change to GET

Explanation: There is an error because the link send an id from URL kaya dapat $GET na lng ang gamitin to avoid error at magmatch behavior.

SCENARIO 10: Wrong variable used in SQL

Problem: Error $aeg
Solution: Correct variable name

Explanation: Typographical error like in scenario 5

SCENARIO 11: Mismatched method (expets POST but form sends GET)

Problem: Form sends GET, PHP expects POST
Solution: Match methods in form and PHP

Explanation: Di nagamatch yung methods kaya nagkakaerror. Dapat palitan ang isa kasi magkaiba ang POST and GET, dapat pareho.

SCENARIO 12: Numeric GET used inside quotes

Problem: ID is numeric but quoted
Solution: Remove quotes

Explanation: Using quotes means text or string, yung ID is number so alsin yung quotes.

SCENARIO 13: Missing WHERE clause in UPDATE

Problem: All rows updated instead of one
Solution: Add WHERE clause

Explanation: Kulang yung ung nakalagay, kung wala yung WHERE sa UPDATE maapektuhan lahat ng rows sa table.

SCENARIO 14: Using POST array incorrectly

Problem: Undefined index and missing quotes
Solution: Use proper syntax

Explanation: Typo ulit. Kailangan ng quotes ng rray elements tas nakaenclose in brackets properly

---

SCENARIO 15: Get parameter used inside SQL without sanitation

Problem: Large page numbers can crash MySQL
Solution: Validate and limit page numbers

Explanation: Kung sobrang laki yung page number possible na magcrash yung database, kaya need i-limit and i-validate.