Arjiannah Carmelle A. Borleo **Web Systems BSIS 2- A** **Sir Reymar Llagas**

Troubleshooting Activity Answers:

Scenario 1

When I checked the code, I realized na we were trying to get the value from POST, but the ID is actually coming from the URL, which uses GET. Kaya yung sinsasabi nung script 'undefined index'. So I- rereplace natin yung $_POST ng $_GET.

Scenario 2

Na found out ko na yung error nangyari kasi wala yung string sa loob ng quotes. MySQL read the name 'Ana' like a column name, not a value. Nung nag add ako ng single quotes around $fname, naintindihan na nung MySQL na string siya.

Scenario 3

Naisipan ko na parang alanganin kasi pwedeng may mag type ng 1 or 1=1 at makuha lahat ng records. Kaya pwedeng gumamit tayo ng prepared statement with bind_param() which forces the value to be treated as a number. This removes the injection risk.

Scenario 4

I added a simple check using !empty() para yung form hindi mag insert ng blank arrows. If the fields are empty, the program tells the user kaysa mag save ng bad data.

Scenario 5

I noticed na yung variable was using emial instead of email. That tiny typo caused the entire error. I corrected the spelling para it matches the form's input name.

Scenario 6

Kagaya nung scenario 3, parang alanganin mag delege kasi someone could enter 1 OR 1=1 in the URL and ma- wipe yung table. I fixed it by using intval() para only real numbers lang accepted.

Scenario 7

I added error checking. Dati, it always printed 'Updated!' even when the query failed due to missed quotes. Ngayon it shows the actual MySQL error if something is wrong. This is more convenient.

Scenario 8

Sa old cold na fetch niya lang yung first row kasi hindi siya nagloop. I added a while loop para it prints all the students' emails.

Scenario 9

The link uses GET (view.php?id=3), so using POST will always fail. Swinitch ko siya sa $_GET, para ma- match yung talagang value na tig send.

Scenario 10

Mali yung spelling na $aeg instead of $age ang nalagay. It's just a small typo but it breaks the whole query din. I

replaced it with the correct variable.

Scenario 11

The form sends GET data, but the backend expects POST, so the value never arrives. Either the form or PHP must match. I fixed both options.

Scenario 12

IDs are numbers, kapag nilagay sila sa

quotes, MySQL reads them na parang string. It still works, but it's inefficient.

I tried to removed the quotes and casted it to int.

Scenario 13

Without a WHERE clause, inuupdate ng MySQL yung entire table. I added a specific condition so only the chosen student's email is updated.

Scenario 14

Yung original code hindi niya tig wrap the array keys in quotes and also hindi niya din tig quote yung values. Inayus natin siya bu using {$data['key']} properly inside the string.

Scenario 15

If someone types a huge number like ?page=999999999, MySQL might struggle. Need natin i- validate yung page number, forced it to be an integer, and set limits so mag stay lang siya sa safe range.