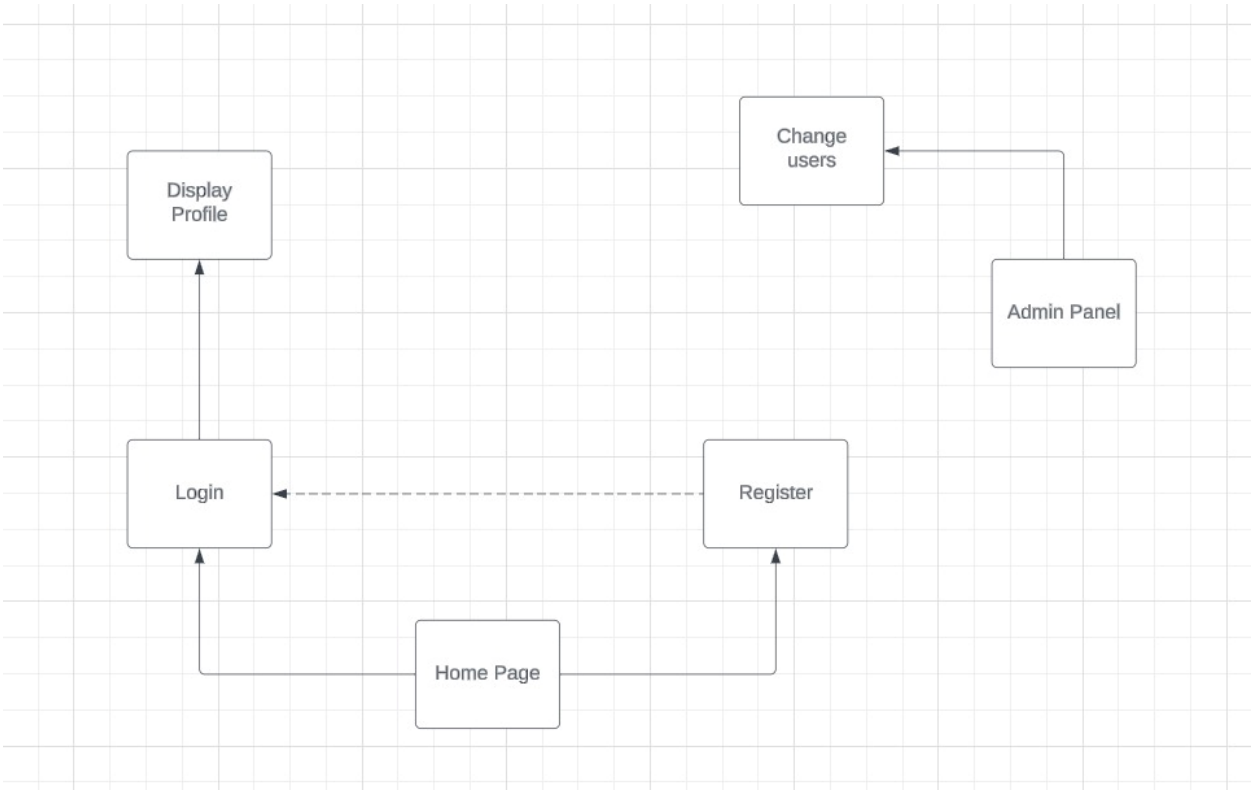
Mini-Project

Database Security

1- Abdelrahman Khaled Abdullah -2021170914
2- Abdelrahman Mohamed Yehia Sohsah-2021170916
3- Ziad Mahmoud Gomaa-2021170911
4- Mohamed Waleed Soliman-20201701823

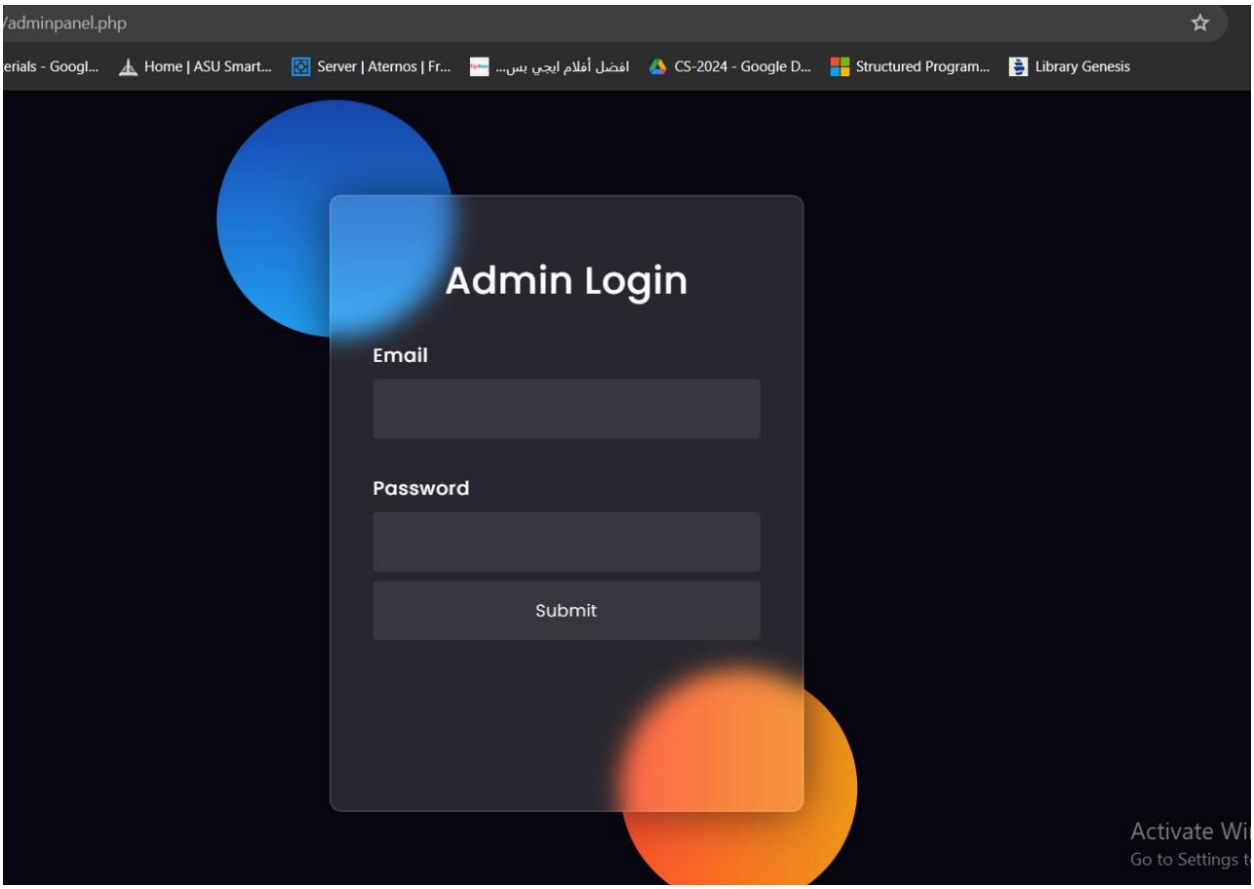
1-Scope Determination and Data Flow Analysis:

We have a mini bank system that has a register and login pages and an admin panel and display profile and a home page in the beginning.

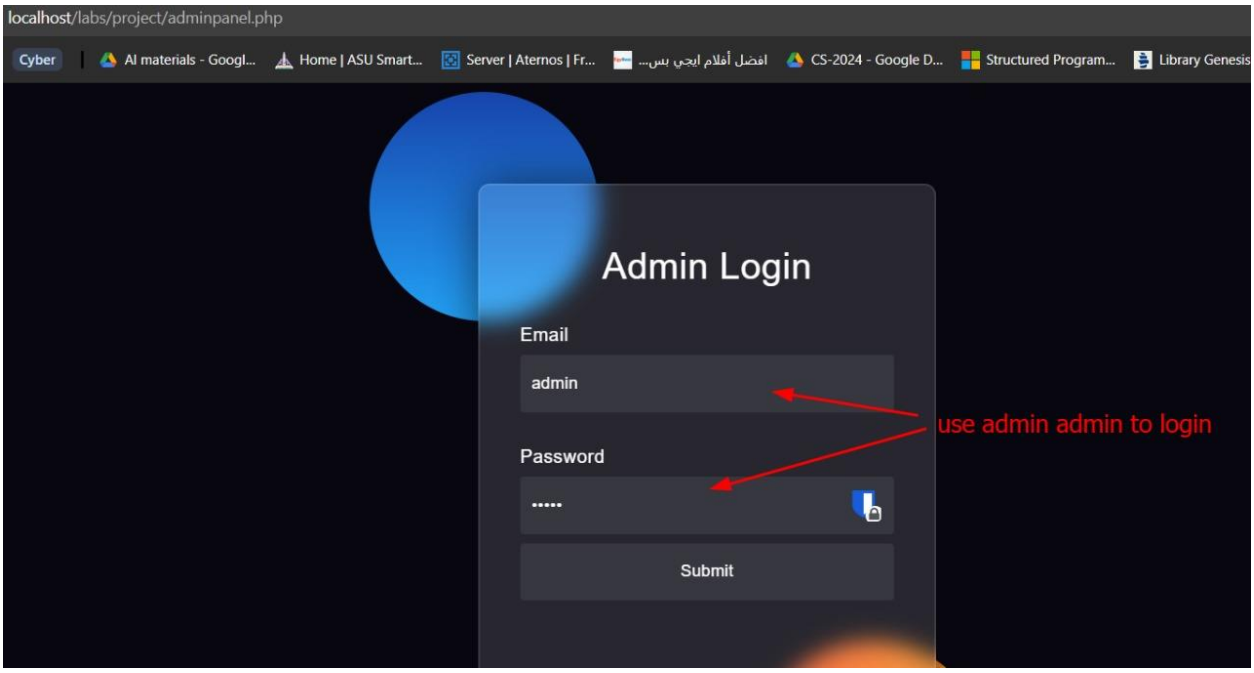


2-Simplified Gap Analysis:

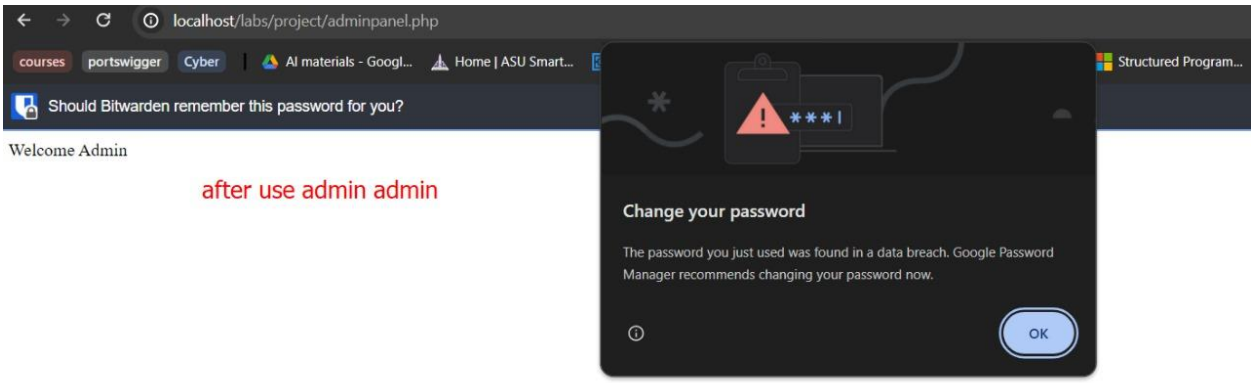
1- Admin panel : after fuzzing the web site we found an admin panel



Firstly, we tried the default username and password for any admin panel which is : (username = admin , Password= admin)

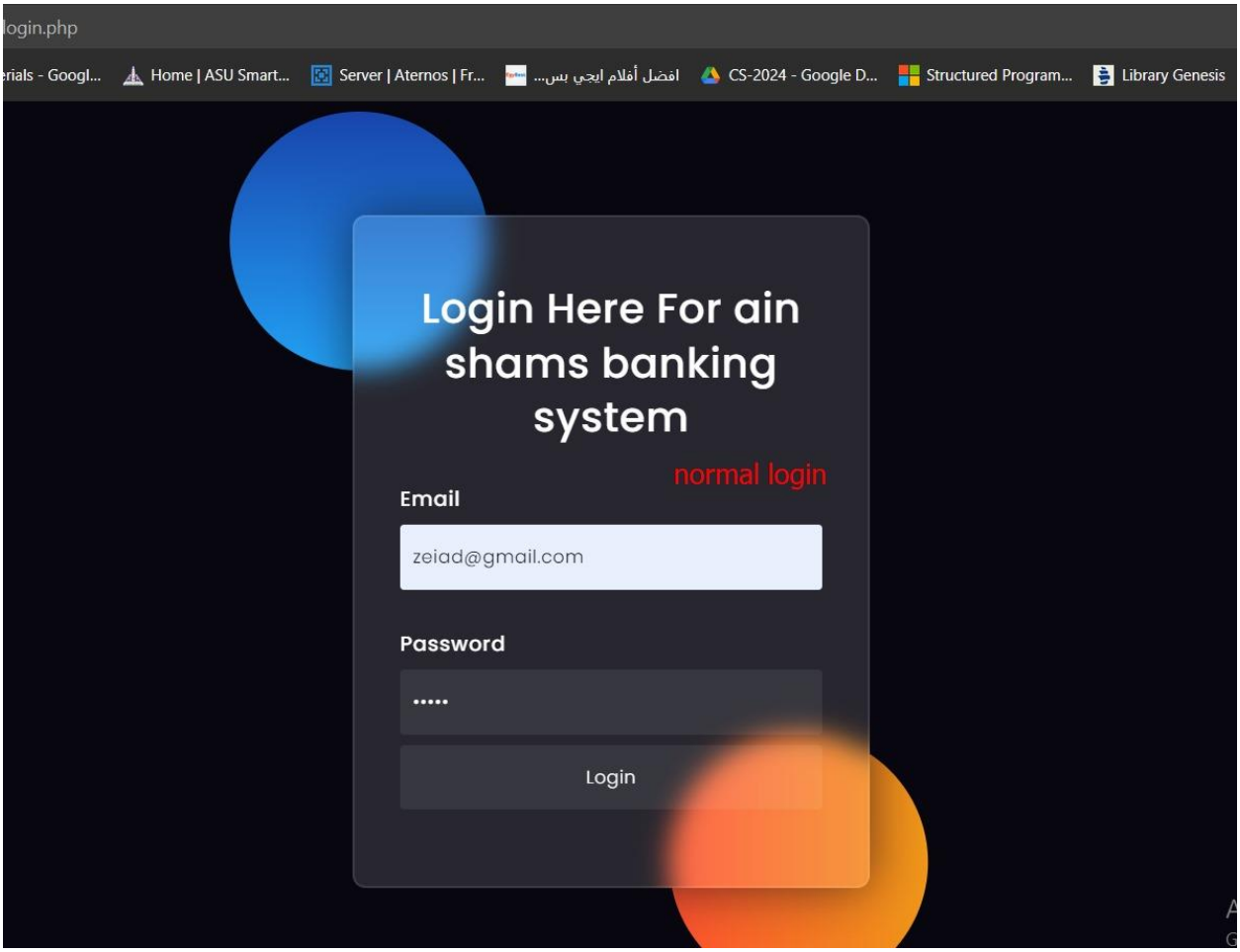


And here we tried the default credentials and it worked.

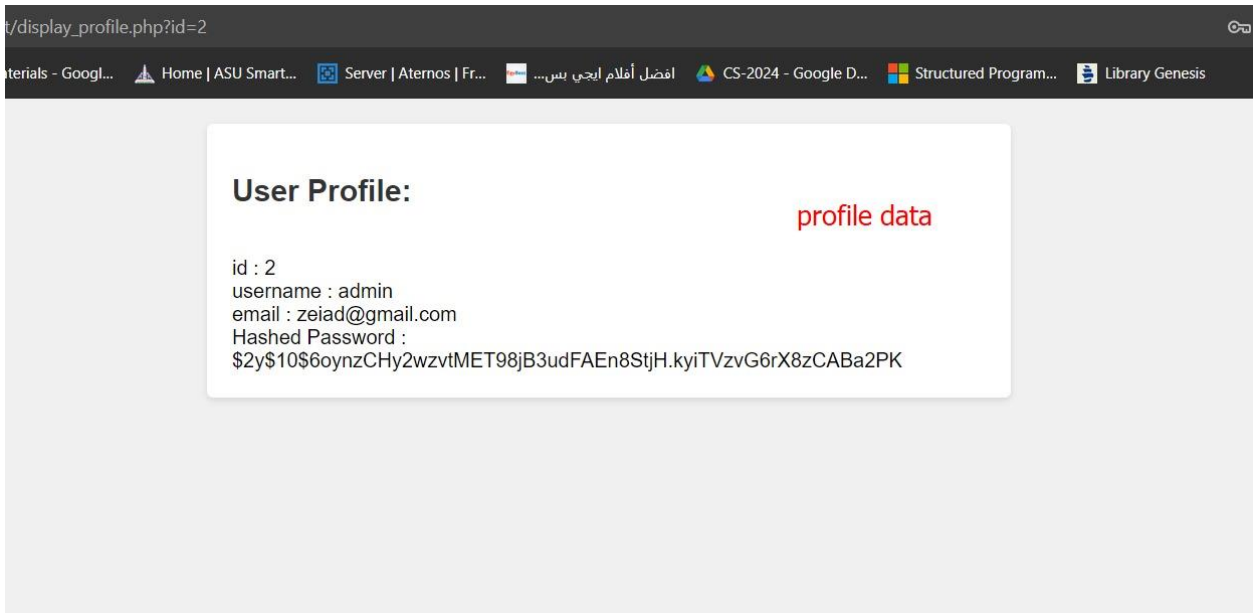


And this is all from requirement 2: Do not use vendor-supplied defaults for system passwords and other security parameters.

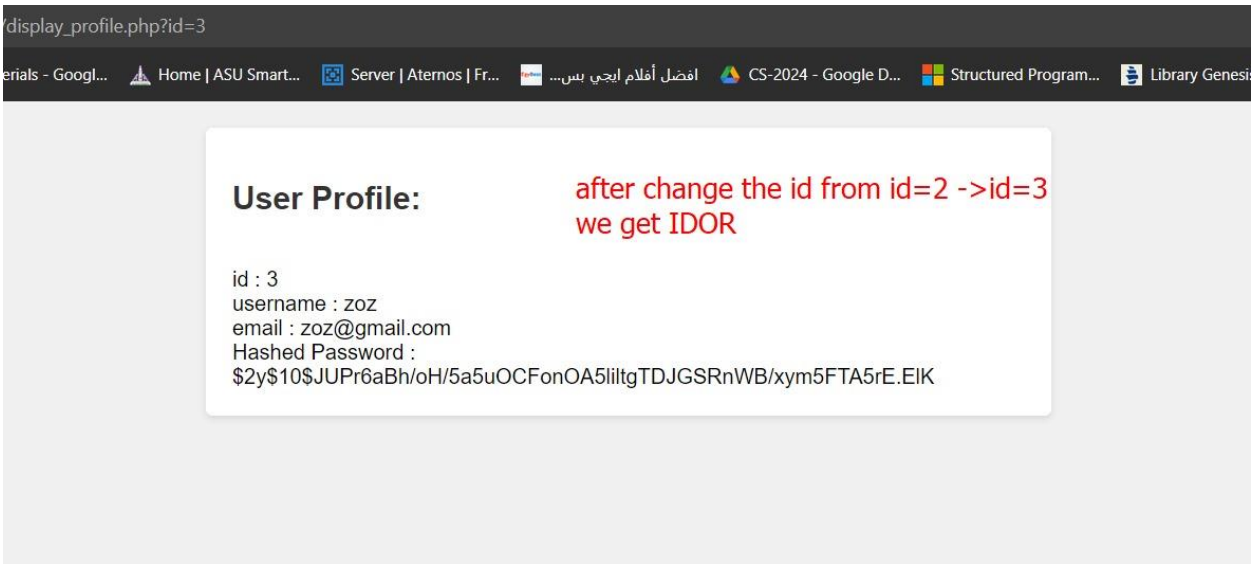
2- Login page ,Display profile (IDOR):



After logging in as a normal user we got redirected to the display profile page.



As you see we have a parameter called ID in the url so we will test for IDOR.



Now we have an IDOR so we can view other users' sensitive data.

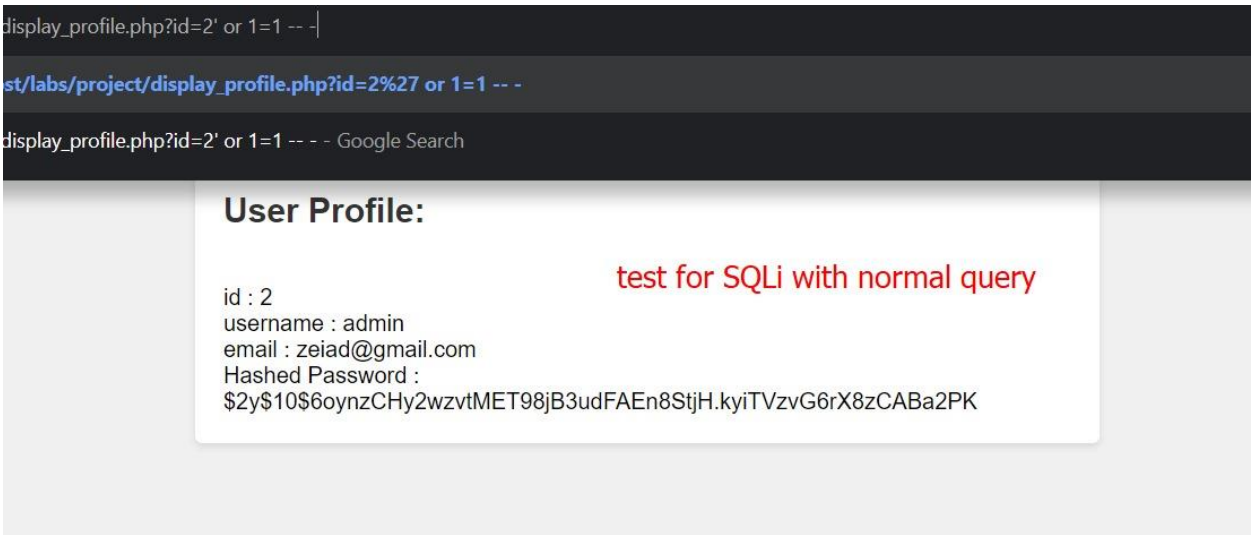
This is requirement 6.5.8 Improper access control (such as insecure direct object references, failure

to restrict URL access, directory traversal, and failure to restrict user access to

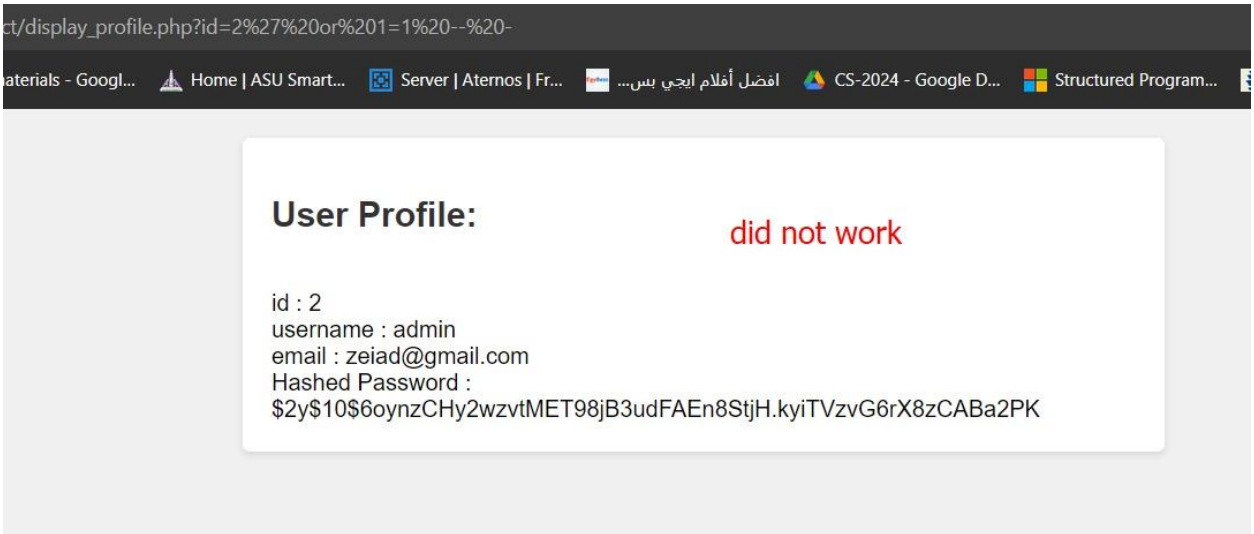
functions).

3-Display profile (SQLi):

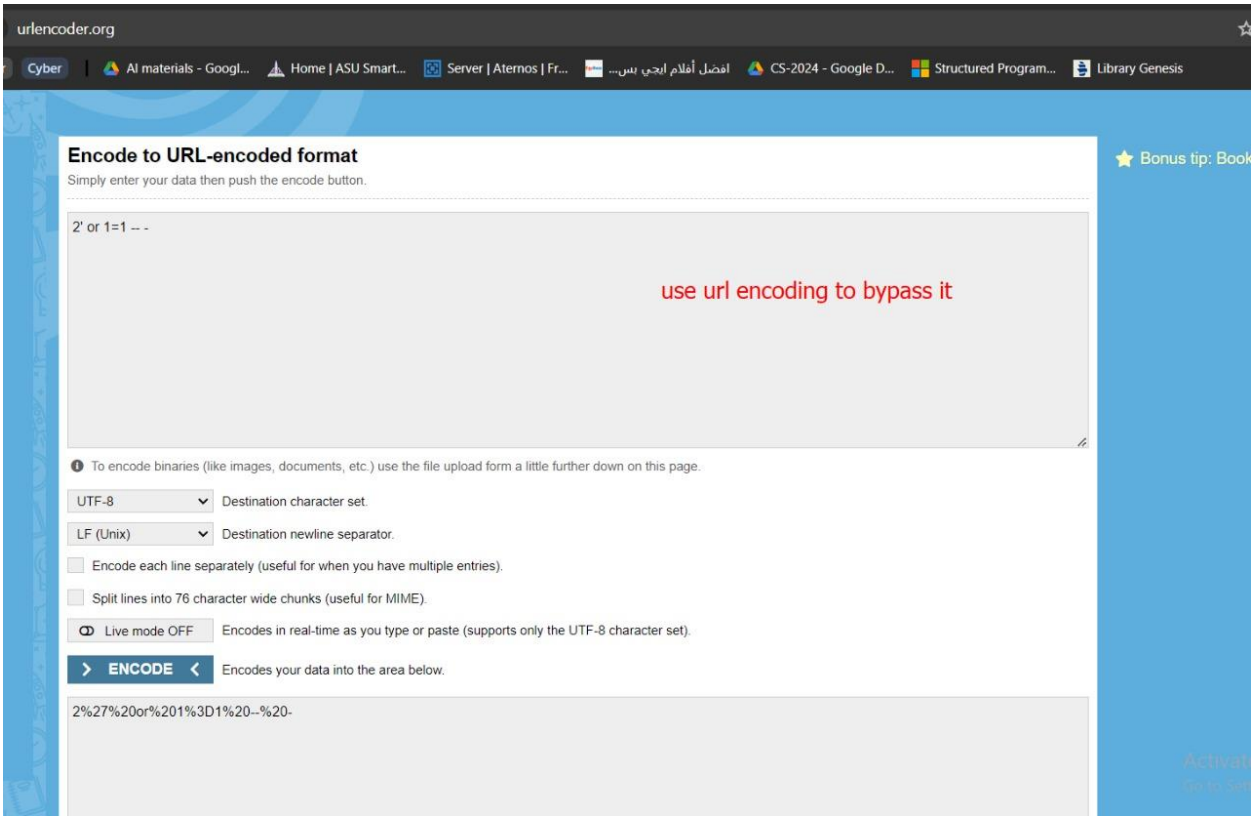
After we tested the ID parameter we suspect that there might be another vulnerability.

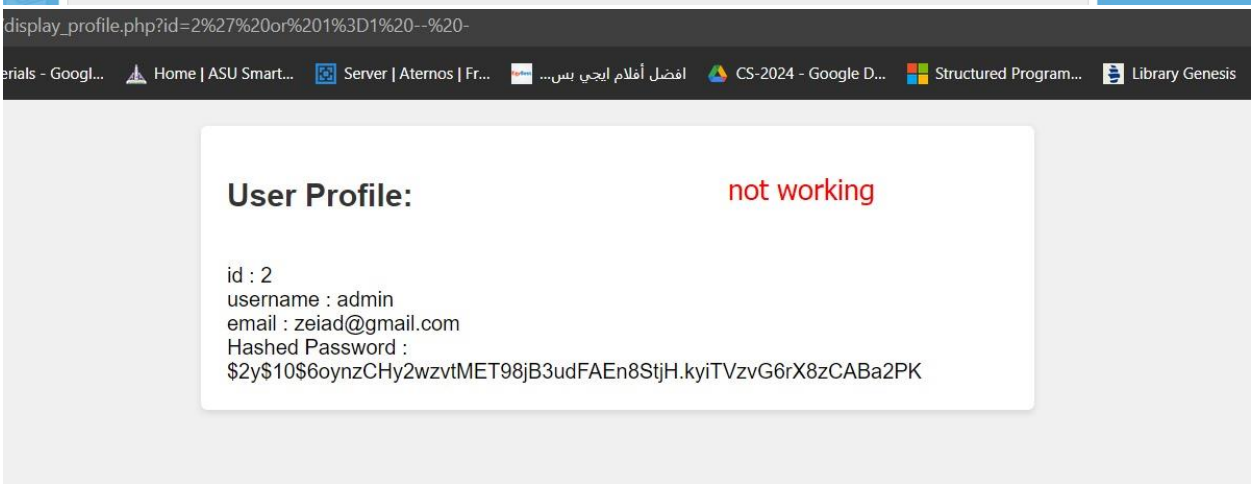


So we tried to test for SQLi with a normal query but it didn't work.



We then tried to bypass it with single url encoding but it also didn't work.





We then tried double url encoding and it worked.

urlencoder.org

Cyber | AI materials - Googl... | Home | ASU Smart... | Server | Aternos | Fr... | افضل أفلام ايجي بس... | CS-2024 - Google D... | Structured Program... | Library Genesis

Encode to URL-encoded format

Simply enter your data then push the encode button.

2%27%20or%201%3D1%20--%20-

use double url encoding to bypass it

To encode binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 — Destination character set.

LF (Unix) — Destination newline separator.

Encode each line separately (useful for when you have multiple entries).

Split lines into 76 character wide chunks (useful for MIME).

Live mode OFF — Encodes in real-time as you type or paste (supports only the UTF-8 character set).

ENCODE — Encodes your data into the area below.

2%2527%2520or%25201%253D1%2520--%2520-

/display_profile.php?id=2%2527%2520or%25201%253D1%2520--%2520-

erials - Googl... | Home | ASU Smart... | Server | Aternos | Fr... | افضل أفلام ايجي بس... | CS-2024 - Google D... | Structured Program... | Library Genesis

User Profile:

it worked we have SQLi

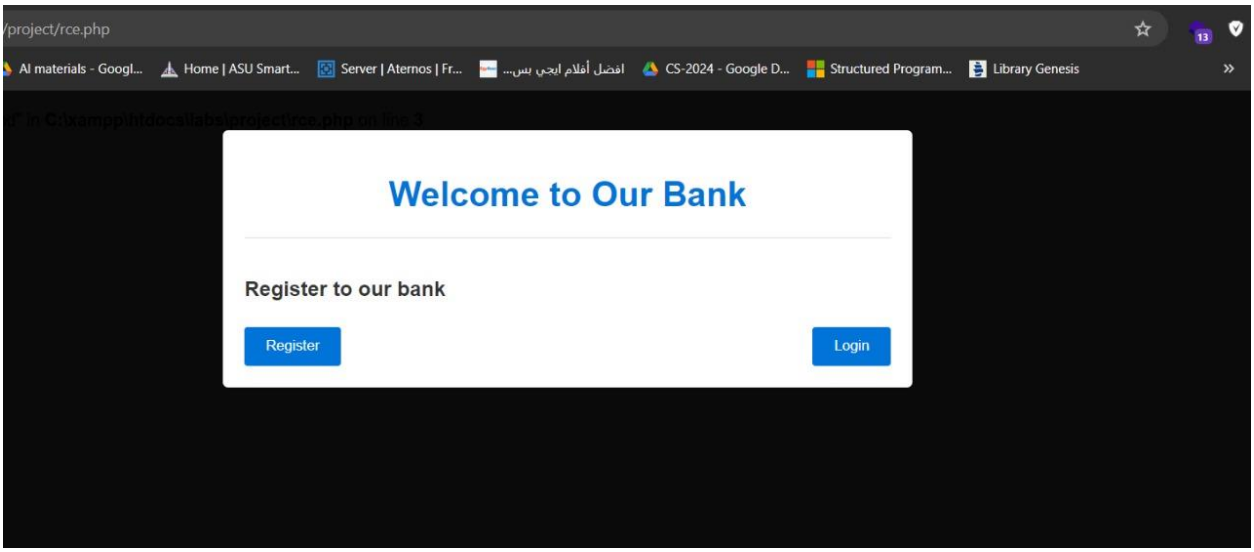
id : 2
username : admin
email : zeiad@gmail.com
Hashed Password :
$2y$10$6oynzCHy2wzvtMET98jB3udFAEn8StjH.kyiTVzvG6rX8zCABa2PK
id : 3
username : zoz
email : zoz@gmail.com
Hashed Password :
$2y$10$JUPr6aBh/oH/5a5uOCFonOA5liltgTDJGSRnWB/xym5FTA5rE.ElK
id : 4
username : [value-3]
email : [value-2]
Hashed Password : [value-4]

We were then able to view all the user and their data so now we have SQLi. This is requirement 6.5.1 Injection flaws, particularly SQL injection. Also consider OS Command Injection,

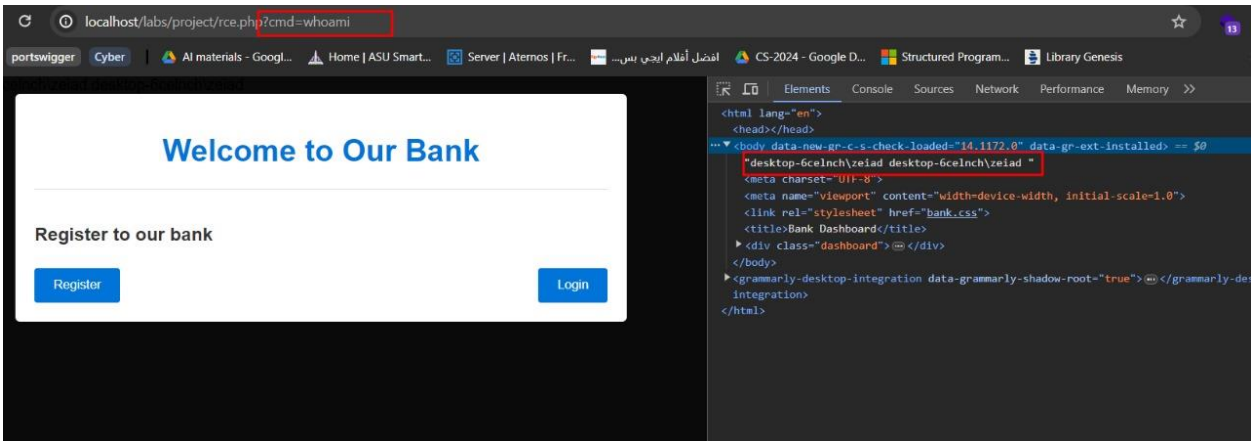
LDAP and XPath injection flaws as well as other injection flaws

4-Home page (OS injection - RCE):

When you visit our website you'll be directed to our home page.



After fuzzing for suspicious parameters we found a parameter called cmd so we tested for OS injection and it was successful.



As you can see we can now display the username of the host. So we can escalate our privilege to get reverse shell.

This violates both requirements 6.5.1 Injection flaws, particularly SQL injection. Also consider OS Command Injection,

LDAP and XPath injection flaws as well as other injection flaws and 6.5.8 Improper access control (such as insecure direct object references, failure

to restrict URL access, directory traversal, and failure to restrict user access to

functions).

3- Recommendations for Compliance Improvement:

1- Change all default credentials and replace it with strong passwords according to the PCI password policy and regularly change passwords at least once every 90 days.
2- Set proper access control.
3- Input validation and sanitization and using parameterized query.
4- Delete unused parameters and make sure to validate every input along with sanitizing it.
5- Regularly test security systems and processes.
6- Define application-layer penetration tests to include, at a minimum, the vulnerabilities listed in requirement 6.5